



# **UNIVERSIDAD NACIONAL DE LOJA**

## **ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES**

### **CARRERA DE INGENIERÍA EN SISTEMAS**

**“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SUB SISTEMA DE SERVICIOS WEB (WEB SERVICES) BASADO EN UNA BIBLIOTECA DE MÉTODOS DE RECUPERACIÓN DE INFORMACIÓN DEL SISTEMA DE GESTIÓN ACADÉMICA DE LA UNIVERSIDAD NACIONAL DE LOJA, QUE PERMITA LA COMUNICACIÓN E INTEROPERABILIDAD CON APLICACIONES RELACIONADAS A FINES ACADÉMICOS”**

**Tesis de Grado previo la obtención  
del Título de Ingeniero en Sistemas**

#### **AUTORES:**

Daysi Carolina Ordóñez González

Silvana de los Ángeles Pacheco Ludeña

#### **DIRECTOR:**

Ing. René Rolando Elizalde Solano

**LOJA- ECUADOR**

**2011**



## CERTIFICACIÓN

Ing. René Rolando Elizalde Solano

DOCENTE DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA

CERTIFICA:

Haber dirigido, revisado y corregido en todas sus partes el desarrollo de la tesis de Ingeniería en Sistemas titulada: **“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SUB SISTEMA DE SERVICIOS WEB (WEB SERVICES) BASADO EN UNA BIBLIOTECA DE MÉTODOS DE RECUPERACIÓN DE INFORMACIÓN DEL SISTEMA DE GESTIÓN ACADÉMICA DE LA UNIVERSIDAD NACIONAL DE LOJA, QUE PERMITA LA COMUNICACIÓN E INTEROPERABILIDAD CON APLICACIONES RELACIONADAS AFINES ACADÉMICOS”**, con autoría de Daysi Carolina Ordoñez González y Silvana de los Ángeles Pacheco Ludeña. En razón de que la misma reúne a satisfacción los requisitos de fondo y forma, exigidos para una investigación de este nivel, autorizo su presentación, sustentación y defensa ante el tribunal designado para el efecto.

Loja, Junio del 2011

.....  
**Ing. René Rolando Elizalde Solano**

**DIRECTOR DE TESIS**



## AUTORÍA

Los conceptos e ideas plasmadas en el presente trabajo de tesis de investigación han sido elaborados bajo razonamiento y discernimiento de las autoras, por lo que se declaran autoras legítimas de éste.



## DECLARACIÓN DE AUTORIDAD

Daysi Carolina Ordóñez González y Silvana de los Ángeles Pacheco Ludeña, autoras intelectuales del presente trabajo de tesis de investigación, autorizan a la Universidad Nacional de Loja, hacer uso del mismo en lo que estime conveniente.

.....  
Daysi Carolina Ordóñez González

.....  
Silvana de los Ángeles Pacheco Ludeña



## AGRADECIMIENTO

Queremos expresar nuestro agradecimiento sincero y estima profunda, a la Universidad Nacional de Loja, al Área de La Energía, las Industrias y los Recursos Naturales no Renovables y en particular a la Carrera de Ingeniería en Sistemas por acogernos como estudiantes, agradecer también a cada uno de los docentes quienes día a día supieron impartir y moldear nuevos conocimientos, contribuyendo con nuestra preparación para que seamos profesionales propositivos que den solución a las problemáticas de la sociedad. De la misma manera al Ing. Patricio Valarezo director del Departamento de Desarrollo de Software de la Universidad Nacional de Loja quien conjuntamente con el Ing. René Elizalde director del trabajo de investigación, se constituyeron en los pilares principales en la dirección y revisión de tesis.



## DEDICATORIA

A Dios por permitirme cumplir con esta meta tan importante en mi vida, en especial a mis padres quienes día tras día y gracias a su esfuerzo me brindan su apoyo para superarme, a mis hermanas y amigos por la ayuda y colaboración para desarrollar y finalizar el presente trabajo.

**Daysi Carolina Ordóñez González.**

A Dios, por concederme la bendición de la vida, a mis padres porque tomada de su mano comencé mi aprendizaje en la vida y ahora todo lo que soy se lo debo a su ejemplo de tenacidad, perseverancia y valor, a mis hermanos por apoyarme siempre, a mis amigos y a todas aquellas personas que contribuyeron y respaldaron para llevar a efecto la culminación de mis estudios universitarios con este trabajo.

**Silvana de los Ángeles Pacheco Ludeña.**



# RESUMEN



## 1. RESUMEN

El trabajo de tesis realizado comprende un sub sistema de servicios web (web services) basado en una biblioteca de métodos de recuperación de información del Sistema de Gestión Académica de la Universidad Nacional de Loja, que permita la comunicación e interoperabilidad con aplicaciones que tengan fines académicos independientemente del lenguaje de programación y plataforma en el que hayan sido desarrolladas logrando así una interrelación efectiva y confiable.

La metodología de desarrollo de software aplicada fue XP (Programación Extrema), que se basa en un modelo ágil, rápido, flexible y de calidad, con un diseño simple y trabajo directo con el usuario. Cumple con 4 fases principales: **Planificación (Análisis):** Donde se analizó qué herramientas se utilizaría y se recopiló los requerimientos del sub sistema. **Diseño simple:** Se definió el modelado que fue migrado del Sistema de Gestión Académica y funcionamiento de la aplicación. **Desarrollo:** Se codificaron cada uno de los métodos de recuperación de información, se integró el código diariamente, se efectuaron pruebas a nivel de desarrolladores y se subieron los cambios realizados al repositorio central de la Unidad de Desarrollo de Software, una vez comprobado el funcionamiento correcto del mismo. **Pruebas:** En esta fase se configuró y se desarrollaron componentes para una aplicación que permitió comprobar que el SGAWeb services o biblioteca de métodos funciona correctamente, asegurando y garantizando su óptimo desempeño.

El web services permite que otras aplicaciones o clientes desarrollados en lenguajes de programación como Ruby, Perl, PHP, Java, etc., puedan obtener datos del Sistema de Gestión Académica como: información académica de estudiantes, de docentes, planes de estudio, ofertas académicas, periodos lectivos, áreas de la universidad, número de estudiantes matriculados, aprobados y reprobados, etc. Todo esto de forma eficaz, segura y fiable.

Los interesados en hacer uso de este servicio deberán solicitar a la Unidad de Desarrollo de Software de la Universidad, que se les asigne un usuario y clave, los mismos que permitirán el acceso a este, debido a que el sistema se encuentra restringido con el





módulo de autenticación básica de Apache. El cliente que acceda al SGAWebservices deberá enviar la información de acuerdo a como se encuentra indicada en las páginas de documentación, en las cuales se realiza un descripción detallada de cada uno de los métodos.

Para el desarrollo del proyecto se utilizó el framework de desarrollo **TurboGears 2.0** el cual maneja como lenguaje de programación **Python**, mismo que permitió la elaboración de los diferentes servicios de este proyecto, la integración de TurboGears con otros paquetes como: **SQLAlchemy** colaboró a la migración correcta del modelo de clases del Sistema de Gestión Académica, debido a que el propio está desarrollado en la versión de TurboGears1.1, este proceso fue necesario para que se puedan realizar las consultas en la base de datos y obtener la información deseada; el paquete **TGWebservices** permitió la generación automática del documento WDSL de cada categoría del SGAWebservices donde se encuentran publicados cada uno de los métodos; el código fuente de este paquete permitió mejorar el trabajo de tesis en vista de que se acopló convenientemente a la necesidades presentadas; el módulo **http\_proxy** para deployar con Apache, y **Json** para dar formato a la salida de los métodos del webservice, entre otros.

El software desarrollado cubre con las necesidades de varios tesis y desarrolladores que requieren obtener información del Sistema de Gestión Académica de manera sencilla, oportuna, rápida, confiable y efectiva.



## SUMMARY

The thesis work involves a sub system made of web services (web services) based on a library of methods of information retrieval from the Academic Management System of the Universidad Nacional de Loja, to enable communication and interoperability with applications that have academic purposes regardless of programming language and platform that have been developed thus achieving an effective and reliable relationship.

The software development methodology applied was XP (Extreme Programming), which is based on a model agile, fast, flexible and quality with a simple and direct work with the user. Meets 4 main phases: planning (Analysis): Where you use what tools are analyzed and compiled the system requirements of sub. Simple: We defined the model that was migrated from the Academic Management System and application performance. Development: We coded each of the methods of information retrieval became part of the code daily, were tested at the developer and got on the changes to the central repository of the Software Development Unit, after checking the correct operation it. Testing: This phase will set up and developed components for an application that revealed that the method library SGAWeb services or working properly, securing and ensuring optimal performance.

The web services allow other applications and clients developed in programming languages like Ruby, Perl, PHP, Java, etc.. May obtain data from the Academic Management System as academic information of students, teachers, curricula, academic offerings , teaching periods, areas of the university, number of students enrolled, approved and disapproved, and so on. All this effectively, safely and reliably.

Those interested in using this service must apply to the Software Development Unit of the University, to be assigned a username and password, they will allow access to this, because the system is restricted to the module Apache basic authentication. The customer accesses the SGAWeb services shall send the information accordingly as is indicated in the pages of documentation, which makes a detailed description of each method.



For the project we used the TurboGears 2.0 development framework which handles Python programming language, which made it possible the development of the various services of this project, TurboGears integration with other packages such as SQLAlchemy contributed to the successful migration the class model of Academic Management System, because the self is developed in TurboGears1.1 version, this process was necessary so that they can perform queries on the database and obtain the desired information, the package TGWebservices allowed the automatic generation of WSDL document for each category are published SGAWeb services where each of the methods, the source of the package improved the thesis given that docked to the needs presented properly, the module for http\_proxy deployer with Apache, and Json to format the output of the methods of webservice, among others.

The software developed covering the needs of several postgraduate students and developers who require information Academic Management System in a simple, timely, fast, reliable and effective.



# ÍNDICE



## 2. ÍNDICE

### 2.1. ÍNDICE DE CONTENIDOS

CONTENIDO	Pág.
CERTIFICACIÓN.....	i
AUTORÍA.....	ii
DECLARACIÓN DE AUTORIDAD.....	iii
AGRADECIMIENTO.....	iv
DEDICATORIA.....	v
1. RESUMEN.....	1
2. ÍNDICE.....	6
2.1. ÍNDICE DE CONTENIDOS.....	<b>7</b>
2.2. ÍNDICE DE FIGURAS.....	13
2.3. ÍNDICE DE TABLAS.....	18
3. INTRODUCCIÓN.....	20
4. METODOLOGÍA.....	24
4.1. MÉTODOS.....	25
4.1.1. Método deductivo.....	25
4.1.2. Método inductivo.....	25
4.1.3. Método Analítico.....	25
4.1.4. Método Científico.....	26
4.1.5. Método de Desarrollo de Software XP.....	26
4.1.5.1. Planificación.....	26
4.1.5.2. Diseño.....	27
4.1.5.3. Desarrollo.....	27
4.1.5.4. Pruebas.....	28
4.2. TÉCNICAS.....	29
4.2.1. Lectura comprensiva.....	29
4.2.2. Entrevista.....	29
4.2.3. Encuesta.....	29
5. FUNDAMENTACIÓN TEÓRICA.....	30
CAPÍTULO I.....	35
1. SISTEMA OPERATIVO LINUX.....	35
1.1. Historia y origen de Linux.....	35
1.2. Características del Sistema.....	35
1.3. Distribución Ubuntu.....	36
1.4. Características de Ubuntu.....	37
1.5. Organización de Paquetes.....	37



1.6. Requisitos para instalar Ubuntu ..... 38

CAPÍTULO II..... 40

2. LENGUAJE DE PROGRAMACION PYTHON..... 40

2.1. Definición..... 40

2.2. Características del lenguaje..... 40

CAPÍTULO III..... 42

3. SISTEMA DE CONTROL DE VERSIONES..... 42

3.1. Definición..... 42

3.2. Ventajas..... 42

3.3. Directorios de Subversion..... 43

3.3.1. Trunk..... 43

3.3.2. Tag..... 43

3.3.3. Branches..... 43

CAPÍTULO IV..... 44

4. PROGRAMACIÓN EXTREMA..... 44

4.1. Definición..... 44

4.2. Funcionamiento..... 44

4.3. Actividades..... 44

4.3.1. Planificación..... 45

4.3.2. Diseño..... 45

4.3.3. Codificación..... 45

4.3.4. Pruebas..... 46

4.4. Ventajas e inconvenientes..... 46

CAPÍTULO V..... 48

5. TURBOGEARS..... 48

5.1. Definición..... 48

5.2. Componentes de TurboGears..... 48

CAPÍTULO VI..... 50

6. APACHE..... 50

6.1. Definición..... 50

6.2. Características de Apache..... 50

6.3. Módulos de Apache..... 51

6.4. Mod\_WSGI..... 52

6.5. Mod\_auth\_mysql..... 52

6.6. Mod\_Proxy..... 54

6.6.1. Mod\_proxy Reverse Proxy..... 54

6.6.2. Ejemplo de mod\_proxy Reverse Proxy..... 55

CAPÍTULO VII..... 56

7. BASE DE DATOS..... 56

7.1. SQLAlchemy..... 56



7.1.1. Definición..... 56

7.1.2. Filosofía de SQLAlchemy..... 56

7.2. SQLite..... 57

7.2.1. Definición..... 57

7.2.2. Características..... 57

7.3. PostgreSQL..... 58

7.3.1. Definición..... 58

7.3.2. Características..... 58

7.3.2.1. Alta concurrencia..... 58

7.3.2.2. Amplia variedad de tipos nativos..... 58

7.3.3. Características Generales..... 59

7.3.4. Funciones..... 60

CAPÍTULO VIII..... 61

8. TGWEB SERVICES..... 61

8.1. Definición..... 61

8.2. Características..... 61

8.3. Ejemplo de Servidor..... 61

8.4. Ejemplo de Cliente..... 62

CAPÍTULO IX..... 63

9. WEB SERVICES..... 63

9.1. Historia de los Servicios Web..... 63

9.2. Definición..... 64

9.3. Características..... 65

9.4. Arquitectura de los Servicios Web..... 66

9.5. Funcionamiento de los Web Services..... 66

9.6. Pila de Protocolo de los Web Services..... 67

9.6.1. Servicio de Transporte..... 68

9.6.1.1. HTTP..... 68

9.6.1.2. SMTP..... 68

9.6.1.3. FTP..... 68

9.6.2. Mensajería XML..... 69

9.6.2.1. XML-RCP..... 69

9.6.2.2. SOAP..... 69

9.6.3. Descripción del Servicio..... 69

9.6.3.1. WSDL..... 69

9.6.4. Descubrimiento de servicios..... 70

9.6.4.1. UDDI..... 70

9.7. MENSAJERÍA XML..... 70

9.7.1. Sintaxis XML..... 71

9.8. SOAP..... 72



9.8.1. Ventajas de SOAP.....	73
9.8.2. Funcionamiento de SOAP.....	74
9.8.3. Modelo de intercambio de mensajes .....	75
9.9. WSDL.....	76
9.9.1. Documento WSDL.....	76
9.9.2. Elementos de WSDL.....	77
9.9.2.1. Elemento type .....	77
9.9.2.2. Elemento message.....	77
9.9.2.3. Elemento portType.....	77
9.9.2.4. Elemento binding.....	77
9.9.2.5. Elemento service.....	77
9.9.2.6. Elementos de Extensibilidad.....	78
CAPÍTULO X.....	79
10. LINESURVEY.....	79
10.1. Características internacionales.....	79
10.2. Administración de Encuestas.....	80
10.2.1. Cómo se construye una encuesta en LimeSurvey.....	81
10.2.2. Cómo se publica una encuesta.....	81
10.2.3. Cómo se recolectan los resultados.....	81
10.3. Procesos por los que pasa la encuesta.....	82
6. EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN.....	84
7. DESARROLLO DE LA PROPUESTA ALTERNATIVA.....	88
7.1. PRESENTACIÓN DE RESULTADOS MEDIANTE LA METODOLOGÍA XP.....	90
7.1.1. Análisis y Diseño.....	90
7.1.1.1. Obtención de Requerimientos.....	91
7.1.1.2. Modelo del Dominio.....	95
7.1.1.3. Diagrama de Casos de Uso por paquetes.....	96
7.1.1.4. Desarrollo de Historias de Usuario y Diagramas de Secuencia.....	99
7.1.1.5. Modelo de Clases Final.....	169
7.1.1.6. Modelo Entidad Relación.....	170
7.1.1.7. Diagrama de Despliegue.....	171
7.1.2. Instalación y Configuración de Herramientas para el Desarrollo del SGAWebServices.....	172
7.1.2.1. Instalación del sistema de control de versiones subversión.....	172
7.1.2.2. Verificar la versión de Python instalada.....	173
7.1.2.3. Instalación de setuptools.....	174
7.1.2.4. Instalación de Gestor de Base de Datos PostgreSQL.....	175
7.1.2.5. Instalación la herramienta virtualenv.....	175
7.1.2.6. Instalar dependencias de Python.....	176
7.1.2.7. Instalación de framework Turbogears2.0.....	178





7.1.2.8.	Instalación del paquete TGWebServices 2.0.0a2.....	181
7.1.2.9.	Creación del Proyecto en Turbogears2.0.....	182
7.1.2.10.	Agregación del proyecto al repositorio de la Unidad de Desarrollo de Software de la Universidad Nacional de Loja.....	185
7.1.3.	Migración de Clases y Métodos del Modelo del Sistema de Gestión Académica al SGAWebServices con Turbogears 2.0.....	188
7.1.4.	Construcción de los Servicios de la Biblioteca de Métodos de Recuperación de Información.....	190
7.1.4.1.	Verificación del funcionamiento de los métodos.....	193
7.1.4.2.	Diseño y construcción de los templates de documentación del SGAWebServices.....	195
7.1.5.	Deployar Turbogears con Apache2.....	208
7.1.5.1.	Instalación de Apache2.....	208
7.1.5.2.	Instalación del módulo proxy_ http.....	208
7.1.5.3.	Creación y configuración del archivo sgawsproxy.....	209
7.1.6.	Implementación de la Autenticación Básica de Apache a través del Mod_Auth_Mysql.....	210
7.1.6.1.	Instalación de Mysql.....	210
7.1.6.2.	Instalación del módulo libapache2-mod-auth- mysql.....	212
7.1.6.3.	Creación y configuración de la base de datos para usuarios del SGAWebServices.....	213
7.1.6.4.	Configuración del sitio sgawsproxy para la autenticación básica con apache.....	216
7.1.6.5.	Comprobar el proceso de autenticación básica.....	218
7.2.	CONFIGURAR EL SISTEMA DE ENCUESTAS DE LA UNIVERSIDAD NACIONAL DE LOJA QUE PRUEBA EL FUNCIONAMIENTO DE MÉTODOS DEL SGAWEB SERVICES..	220
7.2.1.	Instalar php.....	220
7.2.2.	Uso de Lime Survey.....	221
7.2.3.	Configurar Lime Survey.....	221
7.2.4.	Instalación de Lime Survey.....	222
7.2.5.	Acceso a la plataforma por primera vez.....	222
7.2.6.	Configuración de la plataforma de acuerdo a los requerimientos del sistema de encuestas.....	224
7.2.7.	Creación de una plantilla HTML para la encuesta.....	224
7.2.8.	Creación de la Encuesta Socioeconómica para estudiantes.....	225
7.2.8.1.	Diagrama de Casos de Uso.....	225
7.2.8.2.	Desarrollo de Historias de Usuario y Diagramas de Actividades.....	226
7.3.	INSTALACIÓN Y CONFIGURACIÓN DEL SGAWEB SERVICES EN EL SERVIDOR.....	243
7.3.1.	Crear un nuevo usuario con los permisos necesarios en el servidor donde se va a levantar el servicio.....	243



7.3.2. Instalar y configurar los paquetes y programas requeridos en el servidor para poder levantar el WebServices..... 244

7.3.3. Creación de entorno virtual e instalación de programas dentro del mismo..... 246

7.3.4. Bajar del repositorio la aplicación y configurar archivos específicos para poder levantar el proyecto..... 247

7.3.5. Levantar el proyecto SGAWebServices..... 251

7.3.6. Crear usuarios del SGAWebServices 251

7.4. PRUEBAS Y VALIDACIÓN..... 254

7.4.1. Prueba de Validación Aplicadas al Sistema..... 254

7.4.2. Prueba de Usabilidad Aplicadas al Sistema..... 256

8. VALORACIÓN TÉCNICO-ECONÓMICA-AMBIENTAL..... 266

8.2. RECURSOS HUMANOS..... 267

8.3. RECURSOS MATERIALES..... 267

8.4. RECURSOS TÉCNICOS..... 268

8.5. RECURSOS TECNOLÓGICOS..... 268

8.6. RESUMEN DEL PRESUPUESTO..... 269

9. CONCLUSIONES..... 270

10. RECOMENDACIONES..... 272

11. BIBLIOGRAFÍA Y REFERENCIAS..... 274

11.1. LIBRO..... 275

11.2. RECURSOS DE INTERNET..... 275

12. ANEXOS..... 278

12.1. ANEXO #1: Documentación de la Tecnología Web Services..... 279

12.2. ANEXO #2: Formato de Encuesta para Recolección de Requerimientos..... 306

12.3. ANEXO # 3: Formato de Encuesta Socioeconómica proporcionada por el IDICE..... 308

12.4. ANEXO #4: Formato de Encuesta de Prueba de Validación..... 322

12.5. ANEXO #5: Formato de Encuesta de Prueba Usabilidad de la Aplicación..... 324

12.6. ANEXO #6: Contenido de Script de creación de Usuarios del SGAWebservices..... 327

12.7. ANEXO #7: Anteproyecto de Tesis..... 328

12.8. ANEXO #8: Listado de métodos migrados del sgaservices del Sistema de Gestión Académica al SGAWebservices..... 371

12.9. ANEXO # 9 Certificación de la sección de software de UNL..... 372



## 2.2. ÍNDICE DE FIGURAS

<b>FIGURAS</b>	<b>Pág.</b>
Fig.1: Funcionamiento del mod_proxy.....	54
Fig.2: Funcionamiento del mod_proxy Reverse Proxy.....	54
Fig.3: Arquitectura de Servicios Web.....	66
Fig.4: Funcionamiento de un Web Services.....	67
Fig.5: Funcionamiento de SOAP.....	74
Fig.6: Estructura de Mensaje SOAP.....	75
Fig.7: Administración de Encuestas Limesurvey .....	80
Fig.8: Modelo del Dominio.....	95
Fig.9: Diagrama de Casos de Uso de Información Académica.....	96
Fig.10: Diagrama de Casos de Uso de Información Personal.....	96
Fig.11: Diagrama de Casos de Uso de Información Institucional.....	97
Fig.12: Diagrama de Casos de Uso de Validación.....	97
Fig.13: Diagrama de Casos de Uso de Información Estadística.....	98
Fig.14: Diagrama de Secuencia del método sgaws_periodos_lectivos().....	100
Fig.15: Diagrama de Secuencia del método sgaws_ofertas_academicas( id _periodo).....	102
Fig.16: Diagrama de Secuencia del método sgaws_fechas_matriculaoa(id_oferta).....	104
Fig.17: Diagrama de Secuencia del método sgaws_estadoestudiantes_paralelo(id_paralelo).....	106
Fig.18: Diagrama de Secuencia del método sgaws_plan_estudio (id_paralelo)	108
Fig.19: Diagrama de Secuencia del método sgaws_carga_horaria_docente (id_oferta, cedula_docente).....	110
Fig.20: Diagrama de Secuencia del método sgaws_notas_estudiante (cedula, id_carrera, id_oferta).....	112
Fig.21: Diagrama de Secuencia del método sgaws_carreras_estudiante(cedula).....	114
Fig.22: Diagrama de Secuencia del método sgaws_reporte_matricula (id_oferta,cedula).....	116
Fig.23: Diagrama de Secuencia del método sgaws_lista_areas().....	118
Fig.24: Diagrama de Secuencia del método sgaws_datos_area(sigla).....	120
Fig.25: Diagrama de Secuencia del método sgaws_carreras_area(siglas).....	122
Fig.26: Diagrama de Secuencia del método sgaws_datos_carreras (id_oferta)	124
Fig.27: Diagrama de Secuencia del método sgaws_modulos_carrera (id_oferta, id_carrera).....	126
Fig.28: Diagrama de Secuencia del método sgaws_paralelos_carrera (id_oferta, id_carrera).....	128
Fig.29: Diagrama de Secuencia del método sgaws_validar_docente(cedula, clave).....	130



Fig.30: Diagrama de Secuencia del método sgaws\_validar\_estudiante (cedula, clave)..... 132

Fig.31: Diagrama de Secuencia del método sgaws\_datos\_docente(cedula). 134

Fig.32: Diagrama de Secuencia del método sgaws\_datos\_estudiante (cedula). 136

Fig.33: Diagrama de Secuencia del método sgaws\_datos\_usuario(cedula).... 138

Fig.34: Diagrama de Secuencia del método sgaws\_nmatriculados\_oferta(id\_oferta)..... 140

Fig.35: Diagrama de Secuencia del método sgaws\_nmatriculados\_area (id\_oferta, siglas)..... 142

Fig.36: Diagrama de Secuencia del método sgaws\_nmatriculados\_carrera (id\_oferta, id\_carrera)..... 144

Fig.37: Diagrama de Secuencia del método sgaws\_nmatriculados\_modulo (id\_oferta, id\_modulo)..... 146

Fig.38: Diagrama de Secuencia del método sgaws\_nmatriculados\_paralelo(id\_paralelo)..... 148

Fig.39: Diagrama de Secuencia del método sgaws\_naprobados\_oferta (id\_oferta)..... 150

Fig.40: Diagrama de Secuencia del método sgaws\_naprobados\_area (id\_oferta, siglas)..... 152

Fig.41: Diagrama de Secuencia del método sgaws\_naprobados\_carrera (id\_oferta, id\_carrera)..... 154

Fig.42: Diagrama de Secuencia del método sgaws\_naprobados\_modulo (id\_oferta, id\_modulo)..... 156

Fig.43: Diagrama de Secuencia del métodosgaws\_naprobados\_paralelo(id\_paralelo)..... 158

Fig.44: Diagrama de Secuencia del método sgaws\_nreprobados\_oferta (id\_oferta)..... 160

Fig.45: Diagrama de Secuencia del método sgaws\_nreprobados\_area (id\_oferta, siglas)..... 162

Fig.46: Diagrama de Secuencia del método sgaws\_nreprobados\_carrera (id\_oferta, id\_carrera)..... 164

Fig.47: Diagrama de Secuencia del métodosgaws\_nreprobados\_modulo (id\_oferta, id\_modulo)..... 166

Fig.48: Diagrama de Secuencia del método sgaws\_nreprobados\_paralelo (id\_paralelo)..... 168

Fig.49: Diagrama de Clases Final..... 169

Fig.50: Modelo Entidad Relación..... 170

Fig.51: Diagrama de despliegue..... 171

Fig.52: Instalación de subversion..... 172

Fig.53: Confirmación de instalación de subversion..... 173

Fig.54: Finalización de proceso de la instalación de subversion..... 173



Fig.55: Versiones de Python instaladas.....	174
Fig.56: Instalación de postgresql.....	175
Fig.57: Instalación de virtualenv.....	176
Fig.58: Finalización de instalación de virtualenv.....	176
Fig.59: Instalación de python-dev.....	177
Fig.60: Finalización de instalación de virtualenv.....	177
Fig.61: Instalación de build-essential.....	178
Fig.62: Finalización de instalación de build-essential.....	178
Fig.63: Ubicarse al directorio TG2.....	179
Fig.64: Creación del entorno virtual tg2env.....	179
Fig.65: Moverse en el directorio tg2env.....	180
Fig.66: Activación del entorno virtual tg2env.....	180
Fig.67: Instalación de Turbogears2.0.....	181
Fig.68: Instalación de TGWebservices.....	181
Fig.69: Crear el proyecto en Turbogears2.0.....	182
Fig.70: Especificación del nombre del proyecto sgaws.....	182
Fig.71: Especificación nombre del paquete y si requiere autenticación el proyecto.....	183
Fig.72: Mover al directorio del proyecto.....	183
Fig.73: Activación de los paquetes de Turbogears.....	184
Fig.74: Levantar el proyecto creado.....	184
Fig.75: Proyecto levantado correctamente .....	185
Fig.76: Visualización grafica del proyecto levantado.....	185
Fig.77: Importación del proyecto al repositorio.....	186
Fig.78: Contenido del archivo .svnignore.....	186
Fig.79: Agregación del archivo .svnignore al repositorio.....	187
Fig.80: Ejemplo de commit en svn.....	187
Fig.81: Ejemplo de resultados de un método del SGAWebservices.....	194
Fig.82: Ejemplo de error por no enviar todos los datos de un método.....	194
Fig.83: Ejemplo de error por enviar mal el id_carrera.....	194
Fig.84: Ejemplo de error cuando la carrera no pertenece a la oferta académica	195
Fig.85: Ejemplo de error por enviar mal el id de la oferta.....	195
Fig.86: Imagen del template de inicio del SGAWebservices.....	196
Fig.87: Imagen del menú del SGAWebservices.....	197
Fig.88: Imagen del template de Presentación del SGAWebservices.....	197
Fig.89: Listado de métodos de categoría la Información Académica.....	198
Fig.90: Listado de métodos de categoría de Información Institucional.....	198
Fig.91: Listado de métodos de categoría de Información Personal.....	198
Fig.92: Listado de métodos de categoría de Validación.....	199
Fig.93: Listado de métodos de categoría de Información Estadística.....	199
Fig.94: Descripción de la categoría de Información Académica.....	200



Fig.95: Ejemplo de descripción de método de Información Académica.....	201
Fig.96: Descripción de la categoría de Información Institucional.....	201
Fig.97: Ejemplo de descripción de método de Información Institucional.....	202
Fig.98: Descripción de la categoría de Información Personal.....	202
Fig.99: Ejemplo de descripción de método de Información Personal.....	203
Fig.100: Descripción de la categoría de Validación.....	203
Fig.101: Ejemplo de descripción de método de Validación.....	204
Fig.102: Descripción de la categoría de Información Estadística.....	204
Fig.103: Ejemplo de descripción de método de Información Estadística.....	205
Fig.104: Descripción de Documento WSDL.....	205
Fig.105: Ejemplo de Documento WSDL.....	206
Fig.106: Descripción de cómo usar el SGAWeb services.....	207
Fig.107: Instalación de apache2.....	208
Fig.108: Contenido del archivo sgawsproxy.....	209
Fig.109: Instalación de Mysql.....	210
Fig.110: Conceder permisos de instalación.....	211
Fig.111: Configuración de Mysql.....	211
Fig.112: Finalización del proceso de instalación y configuración de Mysql....	212
Fig.113: Instalación del libapache2-mod-auth-mysql.....	212
Fig.114: Finalización del proceso de instalación del libapache2-mod-auth- mysql.....	213
Fig.115: Consola de Mysql.....	213
Fig.116: Creación de la base de datos de usuarios.....	214
Fig.117: Creación de la tabla de usuarios.....	214
Fig.118: Permisos para usuario administrador de la base de datos.....	215
Fig.119: Sección de código agregada al sgawsproxy para autenticación.....	216
Fig.120: Autenticación al SGAWeb services en el navegador web.....	219
Fig.121: Imagen de error por cancelar Autenticación.....	219
Fig.122: Ingreso a la plataforma Lime Survey.....	223
Fig.123: Vista principal para configuración de Lime Survey.....	223
Fig.124: Configuración de Lime Survey.....	224
Fig.125: Plantilla HTML del Sistema de Encuestas.....	225
Fig.126: Diagrama de Casos de Uso del Sistema de Encuestas.....	226
Fig.127: Diagrama de Actividades Administrar encuesta Socio-Económica....	228
Fig.128: Ejemplo para crear encuesta.....	229
Fig.129: Formulario para crear una encuesta.....	230
Fig.130: Formulario para crear una sección de la encuesta.....	231
Fig.131: Formulario para crear una pregunta de la encuesta.....	232
Fig.132: Resumen de la encuesta creada.....	232
Fig.133. Script sgauth.php	234
Fig.134. Script datos_estudiantes.php	235



Fig.135: Diagrama de Actividades Autenticar Estudiante..... 238

Fig.136: Formulario de Autenticación del Estudiante..... 239

Fig.137: Autenticación del Estudiante..... 239

Fig.138: Página de Bienvenida de la Encuesta..... 240

Fig.139: Diagrama de Actividades Responder Encuesta..... 242

Fig.140: Formulario de la Encuesta..... 243

Fig.141: Ejemplo de edición de archivo sudoers..... 244

Fig.142: Ejemplo de acceso a un servidor en forma remota..... 245

Fig.143: Configuración del Archivo sgawsproxy en el servidor..... 250

Fig.144. Ejecución del script adminMysql.py..... 252

Fig.145. Ingreso de nombre usuario..... 252

Fig.146. Ingreso de la clave del usuario..... 253

Fig.147. Creación correcta del usuario..... 253

Fig.148: Resultados Pregunta#1 para Prueba de Usabilidad..... 257

Fig.149: Resultados Pregunta#2 para Prueba de Usabilidad..... 258

Fig.150: Resultados Pregunta#3 para Prueba de Usabilidad..... 260

Fig.151: Resultados Pregunta#4 para Prueba de Usabilidad..... 264

Fig.152: Resultados Pregunta#5 para Prueba de Usabilidad..... 265



## 2.3. ÍNDICE DE TABLAS

<b>TABLAS</b>	<b>Pág.</b>
Tabla. 1: Historia de Usuario #1.....	99
Tabla. 2: Historia de Usuario #2.....	101
Tabla. 3: Historia de Usuario #3.....	103
Tabla. 4: Historia de Usuario #4.....	105
Tabla. 5: Historia de Usuario #5.....	107
Tabla. 6: Historia de Usuario #6.....	109
Tabla. 7: Historia de Usuario #7.....	111
Tabla. 8: Historia de Usuario #8.....	113
Tabla. 9: Historia de Usuario #9.....	115
Tabla. 10: Historia de Usuario #10.....	117
Tabla. 11: Historia de Usuario #11.....	119
Tabla. 12: Historia de Usuario #12.....	121
Tabla. 13: Historia de Usuario #13.....	123
Tabla. 14: Historia de Usuario #14.....	125
Tabla. 15: Historia de Usuario #15.....	127
Tabla. 16: Historia de Usuario #16.....	129
Tabla. 17: Historia de Usuario #17.....	131
Tabla. 18: Historia de Usuario #18.....	133
Tabla. 19: Historia de Usuario #19.....	135
Tabla. 20: Historia de Usuario #20.....	137
Tabla. 21: Historia de Usuario #21.....	139
Tabla. 22: Historia de Usuario #22.....	141
Tabla. 23: Historia de Usuario #23.....	143
Tabla. 24: Historia de Usuario #24.....	145
Tabla. 25: Historia de Usuario #25.....	147
Tabla. 26: Historia de Usuario #26.....	149
Tabla. 27: Historia de Usuario #27.....	151
Tabla. 28: Historia de Usuario #28.....	153
Tabla. 29: Historia de Usuario #29.....	155
Tabla. 30: Historia de Usuario #30.....	157
Tabla. 31: Historia de Usuario #31.....	159
Tabla. 32: Historia de Usuario #32.....	161
Tabla. 33: Historia de Usuario #33.....	163
Tabla. 34: Historia de Usuario #34.....	165
Tabla. 35: Historia de Usuario #35.....	167
Tabla. 36: Historia de Usuario #1 del Sistema de Encuestas.....	227
Tabla. 37: Historia de Usuario #2 del Sistema de Encuestas.....	236





Tabla. 37: Historia de Usuario #3 del Sistema de Encuestas.....	240
Tabla. 38: Resultados Pregunta#1 para Prueba de Usabilidad.....	257
Tabla. 39: Resultados Pregunta#2 para Prueba de Usabilidad.....	258
Tabla. 40: Resultados Pregunta#4 para Prueba de Usabilidad.....	263
Tabla. 41: Resultados Pregunta#5 para Prueba de Usabilidad.....	264
Tabla. 42: Recursos Humanos.....	267
Tabla. 43: Recursos Materiales.....	268
Tabla. 44: Recursos Técnicos.....	268
Tabla. 45: Recursos Tecnológicos.....	268
Tabla. 46: Resumen Presupuesto.....	269



# INTRODUCCIÓN



### 3. INTRODUCCIÓN

La presente trabajo de investigación está orientada a dar solución a los problemas de integración de información que se presentan durante el desarrollo de nuevas aplicaciones que se realizan en la Universidad Nacional de Loja, debido que esta institución dispone de un Sistema de Gestión Académica (SGA) el cual posee características como: permitir el control adecuado de información académica, manejar información de estudiantes, docentes, planes de estudio, administración de acreditaciones, administración de asistencias, proceso de matriculación, proceso de admisiones, entre otras.

Es por esto que se ha creído conveniente el desarrollo de una biblioteca de métodos de recuperación de información basada en la tecnología de un WebService, el mismo que se lo ha realizado con herramienta como: Python, SQLAlchemy, Postgresql, y otros componentes específicos del Sistema de Gestión Académico, agregando una capa de transacciones basadas en un estándar como es SOAP, cuyo propósito principal es centralizar los datos, ofreciendo mayor flexibilidad a la hora de utilizarlos, reutilizar las herramientas que ya han sido desarrolladas por esta unidad, construir una red de servicios alrededor del mismo, aprovechar las prestaciones que el sistema presenta y arrimar los esfuerzos para convertirlo en una herramienta práctica para todos los desarrolladores que busquen construir programas relacionados con los aspectos académicos de esta entidad.

La Metodología utilizada permitió la planificación de tareas, las mismas que se realizaron para la obtención de información y desarrollo del software, con el fin de entregar como resultado un producto de calidad que llene la expectativas del usuario.

La Fundamentación Teórica recopila toda la información relacionada con respecto al proyecto y las herramientas utilizadas para la construcción del mismo.

La Evaluación del Objeto de Investigación da a conocer la situación actual del Sistema de Gestión Académica de la Universidad Nacional de Loja y las actividades llevadas a

cabo para el cumplimiento de los objetivos planteados. Por tal razón, es preciso dar a conocer cuáles son los objetivos de nuestro trabajo investigativo:

Objetivo General:

Analizar, diseñar e implementar un sub sistema de servicios web (web services) basado en una biblioteca de métodos de recuperación de información del sistema de gestión académica de la Universidad Nacional de Loja, que permita la comunicación e interoperabilidad con aplicaciones relacionadas a fines académicos.

Objetivos Específicos:

- Realizar el estudio y documentación acerca de servicios web y el Sistema de Gestión Académica, que nos permitirá adquirir conocimientos sobre el funcionamiento, requerimientos, conceptos y herramientas que se utilizan para el desarrollo del mismo.
- Analizar, diseñar e implementar la biblioteca de métodos de recuperación de información que formaran parte del sub sistema de servicios web.
- Realizar una aplicación en un lenguaje de programación diferente al utilizado por el Sistema de Gestión Académica, con la finalidad de realizar las pruebas necesarias al producto desarrollado y depurar cualquier tipo de error.
- Documentar el funcionamiento y manejo de los métodos desarrollados para la recuperación de información, orientada a los usuarios que harán uso del SubSistema de Servicios Web.

El Desarrollo de la Propuesta Alternativa, detalla explícitamente las actividades y procesos realizados con la finalidad de dar solución a los problemas que implican la falta de un webservices para la centralización de información, consta de un informe detallado de las etapas del desarrollo del software, a través de la metodología de desarrollo de software XP.

Valoración técnico-económica-ambiental en donde se expone principalmente los recursos invertidos para el desarrollo y culminación del presente trabajo.



Conclusiones, que dan una valoración del grado de cumplimiento de los objetivos trazados y los resultados alcanzados.

Recomendaciones en donde se brindan consejos pertinentes, que se podrían tomar en cuenta en el desarrollo de proyectos de esta índole.

Bibliografía: libros, revistas y páginas de internet consultados. Y por último los Anexos en donde se detalla información complementaria y también necesaria para el proyecto.



# METODOLOGÍA

## 4. METODOLOGÍA

La metodología permitió describir, analizar y valorar críticamente el desarrollo del proyecto, puesto que los métodos constituyen un conjunto de procedimientos lógicos a través de los cuales se plantean los problemas científicos, se ponen a prueba las hipótesis y los instrumentos de trabajo investigados. Los métodos y técnicas utilizados fueron los siguientes:

### 4.1. MÉTODOS

Se utilizaron los siguientes métodos para el desarrollo del proyecto:

#### 4.1.1. Método Deductivo

El método deductivo permitió deducir por medio del razonamiento lógico; que información y documentación serían la necesaria para lograr un conocimiento que fortalezca el desarrollo del proyecto, es decir; comprobar que al partir de ideas previamente establecidas como principios generales, se lograría aplicarlo al caso individual y comprobar así la validez del mismo

#### 4.1.2. Método Inductivo

Utilizado en la observación, análisis y clasificación de la información necesaria para el proyecto, así como para la definición correcta de los requerimientos del software.

#### 4.1.3. Método Analítico

Se aprovechó este método para realizar un análisis del objeto en estudio, para efectuar una investigación a fondo de los problemas, causas y consecuencias que se presentaban por la falta de un Sub Sistema de Servicios Web (Web Services), que permita la comunicación con otras aplicaciones desarrolladas en lenguajes de programación y plataformas diferentes a las del Sistema de Gestión Académica cuyo objetivo sea con fines académicos

#### 4.1.4. Método Científico

Permitió establecer las actividades o pasos que se llevaron a cabo con la finalidad de alcanzar los objetivos planteados, mediante el uso de herramientas confiables y conocimientos claramente analizados y concebidos, todo esto para la construcción y correcta definición de funcionamiento del SGAWeb services.

#### 4.1.5. Método de Desarrollo de Software XP

Esta metodología de desarrollo de software fue empleada en vista de que es muy ágil y permite adaptar cualquier cambio que se realice durante el progreso de desarrollo de una aplicación; esto permitió tener éxito en el desarrollo del proyecto. Se utilizó el lenguaje de Modelado Unificado UML para crear y definir los diferentes componentes del sistema. Se utilizó el patrón de arquitectura de software que separa los datos de la interfaz de usuario, y la lógica de control en tres componentes distintos, el modelo, la vista y el controlador que se utiliza frecuentemente en aplicaciones web. Además se utilizó Software Libre que es un excelente recurso para disponer de código fuente de valiosas herramientas de desarrollo. En la programación extrema el proceso es más práctico que teórico.

Esta metodología está compuesta por cuatro fases principales, cada una de las cuales permitió que el desarrollo del proyecto se logre de manera rápida y adecuada, tomando en consideración el tiempo y los requerimientos recolectados. A continuación detallamos cada una de estas fases

##### 4.1.5.1. Planificación

Se estableció un diálogo permanente entre los posibles usuarios y los desarrolladores del sistema. Lo que en primera instancia permitió determinar:

##### A los usuarios y desarrolladores

- **Ámbito:** El problema que el software va a resolver.
- **Prioridad:** Los servicios a desarrollarse en primera instancia.

##### A los desarrolladores

- **Estimaciones:** El tiempo que llevaría implementar una característica o un servicio



- **Procesos:** Organización del equipo de trabajo y distribución del tiempo de desarrollo.
- **Programación detallada:** Problemas que se deberían resolver en caso de presentarse y documentación clara de cada proceso

Además se fundamentó en un análisis permanente, recolección de requerimientos, estudio y definición de la información requerida para el desarrollo del proyecto logrando analizar la mejor decisión a tomar y la selección de herramientas que permitieron la optimización de los procesos que se llevaron a cabo para el desarrollo de la biblioteca de métodos, con la finalidad de establecer de manera clara: el ámbito y prioridad de los servicios.

De igual manera se definieron las historias de usuario en base a los procesos que se llevarían a cabo.

#### 4.1.5.2. Diseño

En esta etapa se realizaron diseños simples y sencillos de cada uno de los servicios que ofrece el sistema procurando conseguir un esquema fácilmente entendible e implementable, tomando en consideración que el software cumpla con las siguientes reglas:

- Funcione con todas las pruebas.
- No tenga lógica duplicada.
- Manifieste cada finalidad importante para los programadores

También se efectuaron las especificaciones y explicaciones por parte de los desarrolladores de cómo iba a funcionar el sistema diagrama de casos de uso, historias de usuario, modelos de dominio, diagramas de secuencia, diagrama de clase final, el diagrama de despliegue y el modelo entidad relación necesarios para una mejor comprensión de los procesos.

#### 4.1.5.3. Desarrollo

En esta fase se llevó a efecto la traducción de las especificaciones realizadas en el diseño a código fuente, creando cada componente del sistema y documentando el software implementado. El código se integró como mínimo una vez al día, y se realizaron las pruebas sobre la totalidad del código desarrollado hasta funcione al 100%,

para finalmente subir los cambios al repositorio de proyectos de la Unidad de Desarrollo de Software de la universidad.

La programación del proyecto se la realizó tomando en cuenta dos criterios propuestos por la metodología XP.

- **Programación en pareja.-** Esto implica que los programadores participen en un esfuerzo combinado de desarrollo realizando una acción que el otro no está haciendo actualmente, esto permitió que ambos programadores posean conocimiento de las tareas realizadas a lo largo del desarrollo, logrando un código más eficiente y de calidad.
- **Recodificación.-** Que implica la posibilidad de modificar el código fuente, para que exista mayor comprensión del mismo y se puedan realizar cambios a futuro si fuere necesario, desde luego sin cambiar su comportamiento ni perder su funcionalidad.

Asimismo es necesario indicar que todo el SGAWebServices fue creado con software libre.

#### 4.1.5.4. Pruebas

En esta fase el sistema fue probado, para asegurar y garantizar que funciona correctamente. Se aplicaron las pruebas que permitieron llegar a la conclusión de que el sistema cumple con todos los requerimientos recolectados a lo largo del proyecto. Las pruebas aplicadas director y desarrolladores de la Unidad de Desarrollo de Software de la universidad, docentes, egresados, estudiantes de la carrera de Ingeniería en Sistemas misma fueron:

- **Prueba de Validación.-** Aplicada director de la Unidad de Desarrollo de Software de la universidad, que permitió verificar la correcta funcionalidad del sistema.
- **Pruebas de Usabilidad.-** Aplicada a los usuarios del sistema que permitieron evaluar la utilidad y robustez del sistema.

Esta metodología promueve los valores de comunicación y simplicidad. Los individuos e interacciones son más importantes que los procesos y herramientas, el funcionamiento del software es más importante que la documentación exhaustiva.

## 4.2. TÉCNICAS

Se utilizaron las siguientes técnicas para la recolección de datos e información:

### 4.2.1. Lectura comprensiva

Permitió obtener un conocimiento ordenado y sistemático de los hechos e ideas relacionadas con el tema objeto de estudio, además sirvió para comprender correctamente como efectuar la implementación del proyecto.

### 4.2.2. Entrevista

Esta técnica permitió obtener los requerimientos de manera directa de las personas encargadas del Sistema de Gestión Académica con la finalidad de que la aplicación cubra todas las necesidades requeridas.

### 4.2.3. Encuesta

Permitió la recolección de requerimientos de los posibles usuarios de la biblioteca de métodos (web services). Dando a conocer la información considerada para ellos la más necesaria y utilizada, para el desarrollo de aplicaciones a fines académicos.



# FUNDAMENTACIÓN TÉORICA



## **5. FUNDAMENTACIÓN TEÓRICA**

### **CAPÍTULO I**

#### **1. SISTEMA OPERATIVO LINUX**

- 1.1. Historia y origen de Linux**
- 1.2. Características del Sistema**
- 1.3. Distribución Ubuntu**
- 1.4. Características de Ubuntu**
- 1.5. Organización de Paquetes**
- 1.6. Requisitos para instalar Ubuntu**

### **CAPÍTULO II**

#### **2. LENGUAJE DE PROGRAMACION PYTHON**

- 2.1. Definición**
- 2.2. Características del lenguaje**

### **CAPÍTULO III**

#### **3. SISTEMA DE CONTROL DE VERSIONES**

- 3.1. Definición**
- 3.2. Ventajas**
- 3.3. Directorios de Subversion**
  - 3.3.1. Trunk**
  - 3.3.2. Tag**
  - 3.3.3. Branches**

### **CAPÍTULO IV**

#### **4. PROGRAMACIÓN EXTREMA**

- 4.1. Definición**
- 4.2. Funcionamiento**
- 4.3. Actividades**
  - 4.3.1. Planificación**
  - 4.3.2. Diseño**



**4.3.3. Codificación**

**4.3.4. Pruebas**

**4.4. Ventajas e inconvenientes**

**CAPÍTULO V**

**5. TURBOGARS**

**5.1. Definición**

**5.2. Componentes de TurboGears**

**CAPÍTULO VI**

**6. APACHE**

**6.1. Definición**

**6.2. Características de Apache**

**6.3. Módulos de Apache**

**6.4. Mod\_WSGI**

**6.5. Mod\_auth\_mysql**

**6.6. Mod\_Proxy**

**6.6.1. Mod\_proxy Reverse Proxy**

**6.6.2. Ejemplo de mod\_proxy Reverse Proxy**

**CAPÍTULO VII**

**7. BASE DE DATOS**

**7.1. SQLAlchemy**

**7.1.1. Definición**

**7.1.2. Filosofía de SQLAlchemy**

**7.2. SQLite**

**7.2.1. Definición**

**7.2.2. Características**

**7.3. PostgreSQL**

**7.3.1. Definición**

**7.3.2. Características**

**7.3.2.1. Alta concurrencia**



**7.3.2.2. Amplia variedad de tipos nativos**

**7.3.3. Características Generales**

**7.3.4. Funciones**

**CAPÍTULO VIII**

**8. TGWEB SERVICES**

**8.1. Definición**

**8.2. Características**

**8.3. Ejemplo de Servidor**

**8.4. Ejemplo de Cliente**

**CAPÍTULO IX**

**9. WEB SERVICES**

**9.1. Historia de los Servicios Web**

**9.2. Definición**

**9.3. Características**

**9.4. Arquitectura de los Servicios Web**

**9.5. Funcionamiento de los Web Services**

**9.6. Pila de Protocolo de los Web Services**

**9.6.1. Servicio de Transporte**

**9.6.1.1. HTTP**

**9.6.1.2. SMTP**

**9.6.1.3. FTP**

**9.6.2. Mensajería XML**

**9.6.2.1. XML-RCP**

**9.6.2.2. SOAP**

**9.6.3. Descripción del Servicio**

**9.6.3.1. WSDL**

**9.6.4. Descubrimiento de servicios**

**9.6.4.1. UDDI**

**9.7. MENSAJERÍA XML**

**9.7.1. Sintaxis XML**



## **9.8. SOAP**

### **9.8.1. Ventajas de SOAP**

### **9.8.2. Funcionamiento de SOAP**

### **9.8.3. Modelo de intercambio de mensajes**

## **9.9. WSDL**

### **9.9.1. Documento WSDL**

### **9.9.2. Elementos de WSDL**

#### **9.9.2.1. Elemento type**

#### **9.9.2.2. Elemento message**

#### **9.9.2.3. Elemento portType**

#### **9.9.2.4. Elemento binding**

#### **9.9.2.5. Elemento service**

#### **9.9.2.6. Elementos de Extensibilidad**

## **CAPÍTULO X**

### **10. LINESURVEY**

#### **10.1. Características internacionales**

#### **10.2. Administración de Encuestas**

##### **10.2.1. Cómo se construye una encuesta en LimeSurvey**

##### **10.2.2. Cómo se publica una encuesta**

##### **10.2.3. Cómo se recolectan los resultados**

#### **10.3. Procesos Por Los Que Pasa La Encuesta**



## CAPÍTULO I

### 1. SISTEMA OPERATIVO LINUX

#### 1.1. Historia y origen de Linux<sup>1</sup>

Es un sistema operativo completamente gratuito de la creación de Linus B. Torvalds. A comienzos de la década de 1990, Torvalds quiso crear su propio sistema operativo para su proyecto de graduación. Linus Torvalds intentó desarrollar una versión de UNIX que pudiera utilizarse en una arquitectura de tipo 80386. El primer clon de UNIX que funcionó en un PC fue Minix, escrito por Andrew Tanenbaum. Éste era un sistema operativo mínimo que se podía usar en un PC. Después, Linus Torvalds decidió ampliar las posibilidades de Minix al desarrollar lo que se convertiría en Linux. Entusiasmados con esta iniciativa, diversas personas contribuyeron para ayudar a Linus Torvalds a hacer de su sistema una realidad. En 1991, la primera versión del sistema salió al mercado. En marzo de 1992 se distribuyó la primera versión, la cual no tenía prácticamente ningún error.

Al aumentar la cantidad de desarrolladores que trabajaban en el sistema, éste integró rápidamente nuevos desarrollos gratuitos de herramientas disponibles en sistemas UNIX comerciales. Después, comenzaron a aparecer nuevas herramientas para Linux con una velocidad increíble. La originalidad de este sistema radica en el hecho de que Linux no se desarrolló con fines comerciales. De hecho, no se copió ni una sola línea de código de los sistemas UNIX originales. Por lo tanto, una vez creado, todos pueden usar Linux gratuitamente e incluso pueden mejorarlo.

#### 1.2. Características del Sistema

Linux es un sistema operativo similar a los sistemas UNIX que puede ejecutarse en diferentes plataformas de hardware: x86 (es decir, plataformas basadas en Intel, AMD, etc.), Sparc, PowerPC, Alpha, ARM, etc. El sistema Linux también funciona en equipos personales y en consolas de juegos o en asistentes personales. Por lo tanto, Linux es un sistema con plataformas múltiples. También admite múltiples usuarios, tareas múltiples

---

<sup>1</sup> KIOSKEA.NET, [<http://es.kioskea.net/contents/linux/linintro.php3>], [Consulta: Octubre-2010]

y procesadores múltiples. A Linux se lo considera un sistema fiable, sólido y potente. Incluso puede ejecutarse con muy pocos recursos en equipos de muy bajo rendimiento.

En particular, el sistema Linux posee las siguientes ventajas:

- Es compatible con los estándares de Internet, es decir, los protocolos TCP/IP, que son los protocolos utilizados en Internet. Linux es una herramienta gratuita para crear redes locales, conectarse a Internet y configurar servidores.
- Mayor seguridad debida a la transparencia de su código fuente y la respuesta de la comunidad ni bien se anuncian las vulnerabilidades.
- Separación de los espacios de la memoria y del disco en combinación con una administración específica de los derechos que permiten la administración de una gran cantidad de usuarios con un nivel mínimo de riesgo.
- Un núcleo que se puede configurar por completo según el hardware del equipo en el que se instaló el sistema para maximizar el rendimiento.

### 1.3. Distribución Ubuntu<sup>2</sup>

Ubuntu es una distribución GNU/Linux que ofrece un sistema operativo predominantemente enfocado a ordenadores de escritorio aunque también proporciona soporte para servidores. Basada en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad de uso, la libertad de uso, los lanzamientos regulares y la facilidad en la instalación. Ubuntu está patrocinado por Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.

El nombre de la distribución proviene del concepto zulú y xhosa de ubuntu, que significa "humanidad hacia otros" o "yo soy porque nosotros somos". Ubuntu es un movimiento sudafricano encabezado por el obispo Desmond Tutu, quien ganó el Premio Nobel de la Paz en 1984 por sus luchas en contra del Apartheid en Sudáfrica. Mark Shuttleworth, favorecedor del proyecto, se encontraba muy familiarizado con la corriente. Tras ver similitudes entre los ideales de los proyectos GNU, Debian y en general con el movimiento del software libre, decidió aprovechar la ocasión para difundir los ideales de Ubuntu. El eslogan de Ubuntu - "Linux para seres humanos" (en inglés "Linux for Human Beings") – resume una de sus metas principales: hacer de Linux un sistema operativo más accesible y fácil de usar.

---

<sup>2</sup> DOC.UBUNTU-ES.ORG, [[http://doc.ubuntu-es.org/Sobre\\_Ubuntu](http://doc.ubuntu-es.org/Sobre_Ubuntu)], [Diciembre - 2009]

Cada seis meses se publica una nueva versión de Ubuntu la cual recibe soporte por parte de Canonical.

#### 1.4. Características de Ubuntu

Ubuntu está basado en la distribución Debian GNU/Linux y soporta oficialmente dos arquitecturas de hardware: Intel x86, AMD64. Sin embargo ha sido portada extraoficialmente a cinco arquitecturas más: PowerPC, SPARC (versión "alternate"), IA-64, Playstation 3 y HP PA-RISC. Al igual que casi cualquier distribución basada en Linux, Ubuntu es capaz de actualizar a la vez todas las aplicaciones instaladas en la máquina a través de repositorios, a diferencia de otros sistemas operativos comerciales, donde esto no es posible. Esta distribución ha sido y está siendo traducida a numerosos idiomas, y cada usuario es capaz de colaborar voluntariamente a esta causa, a través de Internet. Los desarrolladores de Ubuntu se basan en gran medida en el trabajo de las comunidades de Debian, GNOME y KDE (como es el caso de las traducciones).

#### 1.5. Organización de Paquetes<sup>3</sup>

Ubuntu divide todo el software en cuatro secciones, llamadas componentes, para mostrar diferencias en licencias y la prioridad con la que se atienden los problemas que informen los usuarios. Estos componentes son: main, restricted, universe y multiverse. Por omisión, se instala una selección de paquetes que cubre las necesidades básicas de la mayoría de los usuarios. Los paquetes del componente universe de Ubuntu generalmente se basan en los paquetes de la rama inestable (Sid) y en el repositorio experimental de Debian.

- **main:** contiene solamente los paquetes que cumplen los requisitos de la licencia de Ubuntu, y para los que hay soporte disponible por parte de su equipo. Éste está pensado para que incluya todo lo necesario para la mayoría de los sistemas Linux de uso general.
- **restricted:** contiene el programa soportado por los desarrolladores de Ubuntu debido a su importancia, pero que no está disponible bajo ningún tipo de licencia

---

<sup>3</sup> ENCICLOPEDIA LIBRE WIKIPEDIA, [[http://es.wikipedia.org/wiki/Ubuntu#\\_El\\_software\\_incluido](http://es.wikipedia.org/wiki/Ubuntu#_El_software_incluido)], [Consulta: Julio-2010]

libre para incluir en main. En este lugar se incluyen los paquetes tales como los controladores propietarios de algunas tarjetas gráficas, como por ejemplo, los de ATI y NVIDIA. El nivel de la ayuda es más limitado que para main, puesto que los desarrolladores puede que no tengan acceso al código fuente.

- **universe:** contiene una amplia gama de programas, que pueden o no tener una licencia restringida, pero que no recibe apoyo por parte del equipo de Ubuntu sino por parte de la comunidad. Esto permite que los usuarios instalen toda clase de programas en el sistema guardándolos en un lugar aparte de los paquetes soportados: main y restricted.
- **multiverse:** contiene los paquetes sin soporte debido a que no cumplen los requisitos de software libre.

## 1.6. Requisitos para instalar Ubuntu

Los requisitos mínimos recomendados deberían permitir ejecutar una instalación de Ubuntu. Por lo general, se puede ejecutar Ubuntu en hardware más antiguo de lo especificado, aunque el rendimiento necesariamente va a ser menor.

La mayoría de los usuarios tienen riesgo de acabar en frustración si ignoran estas sugerencias.

- Procesador: 1 GHz x86.
- Memoria RAM: 512 MB.
- Disco Duro: 5 GB (para una instalación completa con swap incluida).
- Tarjeta gráfica VGA y monitor capaz de soportar una resolución de 1024x768.
- Lector de CD-ROM o tarjeta de red.
- Tarjeta de sonido.
- Conexión a Internet.

Los efectos de escritorio, proporcionados por Compiz, se activan por defecto en las siguientes tarjetas gráficas:



- Intel (i915 o superior, excepto GMA 500, nombre en clave "Poulsbo")
- NVidia (con su controlador propietario)
- ATI (a partir del modelo Radeon HD 2000 puede ser necesario el controlador propietario).



## CAPÍTULO II

### 2. LENGUAJE DE PROGRAMACION PYTHON

#### 2.1. Definición<sup>4</sup>

Python es un lenguaje de programación interpretado e interactivo, capaz de ejecutarse en una gran cantidad de plataformas. Fue creado por Guido van Rossum en 1990. Python es habitualmente comparado a TCL, Perl, Scheme, Java y Ruby. Actualmente, Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation.

Guido van Rossum, más conocido como Guido, creó Python, un lenguaje de programación de scripting, la “oposición leal” a Perl, lenguaje con el cual mantiene una rivalidad amistosa. Los usuarios de Python consideran a éste mucho más limpio y elegante para programar. Python permite dividir el programa en módulos reutilizables desde otros programas Python. Viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas.

Python es un lenguaje interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. También es una calculadora muy útil.

#### 2.2. Características del lenguaje<sup>5</sup>

- **Propósito general.-** Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.

---

4 SOLODISENIO, [<http://www.solodisenio.com/lenguaje-de-programacion-python/>], [Abril-2010]  
5DESARROLLOWEB.COM, [<http://www.desarrolloweb.com/articulos/1325.php>], [Consulta: Marzo-2010]

- **Multiplataforma.-** Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- **Interpretado.-** Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.
- **Interactivo.-** Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
- **Orientado a Objetos.-** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- **Funciones y librerías.-** Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.
- **Sintaxis clara.-** Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave begin y end. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

## CAPÍTULO III

### 3. SISTEMA DE CONTROL DE VERSIONES

#### 3.1. Definición<sup>6</sup>

Un sistema de control de versiones es un software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, como una novela, o el código fuente de un programa.

Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local. Esto permite compartir los cambios sobre un mismo conjunto de ficheros. Además, el repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de necesidad.

#### 3.2. Ventajas<sup>7</sup>

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- La creación de ramas y etiquetas es una operación más eficiente; Tiene costo de complejidad constante como en CVS.
- Se envían sólo las diferencias en ambas direcciones.
- Puede ser servido, mediante Apache, sobre WebDAV/DeltaV. Esto permite que clientes WebDAV utilicen Subversion en forma transparente.
- Maneja eficientemente archivos binarios.
- Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.

---

<sup>6</sup> DIT.UPM.ES, [[http://polaris.dit.upm.es/~rubentb/docs/subversion/Tutorial Subversion/index.html](http://polaris.dit.upm.es/~rubentb/docs/subversion/Tutorial%20Subversion/index.html)], [Enero - 2010]  
<sup>7</sup> IMAPAX, [<http://www.imapax.com/soluciones/joomla/que-es-subversion-svn.html>], [Mayo-2010]





- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos.

### **3.3. Directorios de Subversion**

Los directorios que tiene subversion son los siguientes:

#### **3.3.1. Trunk**

Directorio de un repositorio bajo el cual se encuentran los directorios y ficheros de nuestro proyecto.

#### **3.3.2. Tag**

Directorio bajo el cual se almacenan las diferentes versiones de nuestro software.

#### **3.3.3. Branches**

Directorio donde se almacenan ramas de nuestro proyecto. Una rama es una desviación que se toma a partir de una versión del software para desarrollar alguna funcionalidad en paralelo con el desarrollo principal para su posterior integración.

## CAPÍTULO IV

### 4. PROGRAMACIÓN EXTREMA

#### 4.1. Definición<sup>8</sup>

Es un enfoque de la ingeniería de software formulado por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

La Programación Extrema es uno de los llamados procesos o metodologías ágiles (ProcesoAgil) de desarrollo de software. Consiste en un conjunto de prácticas que a lo largo de los años han demostrado ser las mejores prácticas de desarrollo de software, llevadas al extremo, fundamentadas en un conjunto de valores.<sup>9</sup>

#### 4.2. Funcionamiento

La programación extrema está compuesta por una serie de prácticas y actividades. Las prácticas que componen la programación extrema se pueden agrupar en cuatro grandes bloques: plan, diseño, codificación y pruebas.

Sin embargo, estos bloques no deben realizarse en orden, si no que cada uno consta de una serie de actividades, y todas ellas se irán realizando de manera evolutiva.

#### 4.3. Actividades

Las actividades son las siguientes:

---

<sup>8</sup> ENCICLOPEDIA LIBRE WIKIPEDIA, [[http://es.wikipedia.org/wiki/ Programacion%C3%B3n\\_Extrema](http://es.wikipedia.org/wiki/Programacion%C3%B3n_Extrema)], [Consulta: Julio-2010]

<sup>9</sup> PROGRAMACIONEXTREMA, [<http://www.programacionextrema.org>], [Enero - 2010]

#### 4.3.1. Planificación

- Se escriben *Historias de usuarios*, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen test de aceptación para el *Historias de usuarios* y permita hacer una estimación de tiempo de desarrollo del mismo.
- Se crea un plan de lanzamiento (*release planning*), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usará como base los *Historias de usuarios*, participando el cliente en la elección de los que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.
- El desarrollo se divide en iteraciones, cada una de las cuales comienza con un plan de iteración para el que se eligen las *Historias de usuarios* a desarrollar y las tareas de desarrollo.

#### 4.3.2. Diseño

- Se eligen los diseños más simples que funcionen.
- Se elige una metáfora del sistema para que el nombrado de clases, etcétera, siga una misma línea, facilitando la reutilización y la comprensión del código.

#### 4.3.3. Codificación

- El cliente está siempre disponible, a ser posible cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP (escribe las historias de usuario con la ayuda de los desarrolladores, participa en la elección de los que entrarán en el plan de lanzamientos, prueba pequeños lanzamientos, participa en las pruebas de funcionalidad). La idea es usar el tiempo del cliente para estas tareas en vez de para que cree una detalladísima especificación de requisitos, y evitar la entrega de un producto peor que le hará perder tiempo.
- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.

- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro qué objetivos tiene que cumplir lo que se va a codificar.
- La programación del código se realizará en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores integrando código.
- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes periodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades habrán sido detectadas enseguida.
- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo
- Se deja la optimización para el final

#### 4.3.4. Pruebas

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- Cuando se encuentra un error de codificación o *bug*, se desarrollan pruebas para evitar volver a caer en el mismo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las *historias de usuario* elegidas para la iteración, y son “pruebas de caja negra”, en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que la correspondiente historia de usuario se ha completado.

#### 4.4. Ventajas e inconvenientes

- **Ventaja:** Cuatro ojos ven más que dos. Al trabajar de dos en dos, el código será de mayor calidad desde el mismo momento de crearlo y tendrá menos fallos.



- **Inconveniente:** Para un programador experto puede resultar tedioso tener a un novato a su lado permanentemente.
- **Ventaja:** Los programadores novatos aprenderán de los expertos al emparejarse con ellos.
- **Inconveniente:** El programador experto no aprende y su trabajo se ve ralentizado.
- **Ventaja:** Si una pareja realiza un trozo de código susceptible de ser reutilizado en el proyecto, hay dos programadores que lo saben y que lo reutilizarán cuando puedan (ya que saben cómo funciona), enseñándolo a sus nuevos compañeros. De esta manera el conocimiento del código ya hecho se propaga de forma natural entre todos los programadores del equipo.
- **Ventaja:** El estilo de programación tiende a unificarse.
- **Inconveniente:** La mejora o cambios en el estilo de programación puede resultar más complejo

## CAPÍTULO V

### 5. TURBOGARS

#### 5.1. Definición<sup>10</sup>

**TurboGears** es un megaframework para desarrollo web, soporta una serie componentes propios y de terceros, incluyendo templates (además del oficial se puede usar otros vía plug-ins), AJAX, servidor (mapeador) web, manejo de formularios, widgets (con navegador incluido), interfaz para diseñar y administrar la base de datos, i18n con interfaz web para hacer las traducciones (y hasta para comenzar con una traducción automática de google!), autenticación/permisos, creación simple de interfaz administrativa (ABM simple), etc.

TurboGears está diseñado bajo la arquitectura modelo-vista-controlador al igual que Struts o Ruby on Rails, diseñado para un rápido desarrollo de aplicaciones web en Python y más fácil de mantener.

TurboGears es sólo el pegamento que une una gran cantidad de proyectos independientes que componen las capas del stack web, que bajo la filosofía Unix, hacen una sola cosa y la hacen bien.

#### 5.2. Componentes de TurboGears<sup>11</sup>

Los componentes de TurboGears son los siguientes:

- **SQLAlchemy (Modelo)** - Define la estructura de tablas de base de datos del usuario y cómo los relacionan con los objetos de Python usuario el controlador puede interactuar.
- **Genshi (Vistas)** - Define las plantillas para el código HTML o XHTML que el usuario va a generar. Aquí es donde el usuario define el usuario front-end del cliente interactuar con ellos.
- **Pylones (Contralor)** - Este middleware se encargará del usuario de servicios de fondo la lógica y se conecta a todos los usuarios del servidor web para ofrecer

---

10 YUKEI.NET, [<http://jesaavedra.opentelematics.org/?p=35>], [Mayo-2010]

11 TURBOGARS.ORG, [<http://www.turbogears.org>], [Abril-2010]



seguridad de datos en la web. Por defecto, el Paster servidor web se utiliza, pero el usuario también puede conectarse a Apache, Nginx, o cualquier otro servidor web compatible WSGI.

- **Repoze** - Repoze.who Repoze.what y se utilizan para manejar la seguridad (autenticación y autorización, respectivamente). Cuando los elementos definitorios de control del usuario para ser expuestos en la web, predicados repoze definir quién puede acceder a ellos y en qué condiciones.
- **Reproductores Tosca** - Es la principal librería de widgets para crear formas y GUIs complejo. Tosca por defecto generará simples formularios HTML, pero también puede ser utilizado como un middleware para conectarse a reproductores de JavaScript más avanzado y herramientas. A diferencia de TurboGears 1.x, ya no hay un preferido / integrado de bibliotecas JavaScript.

## CAPÍTULO VI

### 6. APACHE

#### 6.1. Definición<sup>12</sup>

Apache es un servidor web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. Estos voluntarios se conocen como el Apache Group. Además del Apache Group, cientos de personas han contribuido al proyecto con código, ideas y documentación. Estos voluntarios se conocen como el Apache Group. Además del Apache Group, cientos de personas han contribuido al proyecto con código, ideas y documentación.

#### 6.2. Características de Apache<sup>13</sup>

Las características de Apache tienen inserción y utilización en varios ámbitos tanto como empresariales, tecnológicos y educativos:

1. Fundamentalmente corre sobre una multitud de plataformas y Sistemas Operativos.
2. Apache ofrece tecnología libre y de código abierto, otorgándole una transparencia y dando la posibilidad de conocer que es lo que realmente estamos instalando.
3. Apache es un servidor Web altamente configurable y de diseño modular, capaz de ampliar su funcionalidad y calidad de servicios.
4. Apache trabaja en conjunto con gran cantidad de Lenguajes de Programación interpretados como PHP, Perl, soporte con CGI, Java, JSP y otros lenguajes de

---

<sup>12</sup>XOLUCIONESINFORMATICAS,[[http://www.xolucionesinformaticas.com/index.php?option=com\\_content&view=article&id=77&Itemid](http://www.xolucionesinformaticas.com/index.php?option=com_content&view=article&id=77&Itemid)], [Enero-2010]

<sup>13</sup> APLICACIONESEMPRESARIALES.COM , [<http://www.aplicacionesempresariales.com/apache-el-servidor-web-mas-reconocido.html>], [Junio - 2010]



script, el complemento ideal para los sitios web dinámicos que vemos en la actualidad.

5. Es posible configurar y personalizar cada uno de los mensajes de error que se pueden producir por la utilización del servidor.
6. Contar con los archivos Log, en donde registra gran cantidad de información global del sistema, errores producidos en un determinado tiempo, en la cual estos archivos son de gran importancia para los administradores de sistemas y pueden influenciar de alguna manera las políticas de seguridad debido a la gran cantidad de información que contiene.

Otra particularidad propia de Apache y que está muy ligada a su pensamiento y filosofía libre, es que al ser tan popular y utilizado, es posible encontrar gran cantidad de documentos, ejemplos y ayuda en Internet en todos los idiomas.

### 6.3. Módulos de Apache<sup>14</sup>

Apache fue desarrollado altamente modular por tal motivos contamos con una gran cantidad de módulos escritos y que aportan al complemento y la expansión funcional de este Servidor.

Un Módulo en Apache es una manera de agrupar y Modularizar ciertos funcionamientos para el Servidor, existen una gran cantidad de Módulos para utilizarse con Apache.

Una de las principales razones de emplear módulos en Apache, es que no toda instalación requiere de las mismas funcionalidades, esto es, una instalación que utilice PHP probablemente no requiera de Tomcat (Java), o bien posiblemente no todas las instalaciones requieran de Virtual Hosting.

Por lo tanto, si fueran incluidas todas las funcionalidades posibles en una versión única de Apache, esto lo haría sumamente pesado en cuanto a requerimientos de Memoria RAM y espacio en Disco Duro, por esto se opta por modularizar e incluir solo lo necesario

La modularización está aún más acentuada en Apache 2. En esta versión, el servidor se ocupa de un número muy reducido de tareas mientras que el resto se realiza a través de módulos. Esto se lleva a tal extremo que incluso el procesamiento de HTTP tiene lugar

---

<sup>14</sup> OSMOSISLATINA, [<http://www.osmosislatina.com/apache/modulos.htm>], [Enero - 2010]

a través de módulos. Por lo tanto, Apache 2 no debe ser necesariamente un servidor web; también puede asumir otras tareas muy distintas a través de módulos diferentes.

#### 6.4. Mod\_WSGI<sup>15</sup>

Mod\_wsgi es un módulo Apache que proporciona una interfaz compatible con WSGI para acoger las aplicaciones basadas en la web de Python en Apache. El adaptador está escrito completamente en código C en contra de la C Runtime Apache y para alojar aplicaciones WSGI dentro de Apache tiene una sobrecarga menor que utilizando los adaptadores WSGI para mod\_python o CGI.

#### 6.5. Mod\_auth\_mysql<sup>16</sup>

El mod\_auth\_mysql, este módulo permite gestionar la autenticación y autorización, almacenando los usuarios y los grupos en tablas de Mysql, de forma que incluso se pueden gestionar mediante un panel de control.

Para utilizar el módulo es necesario disponer de al menos una tabla donde se almacenen los usuarios y otra donde se almacenen los grupos. Para ello se puede crear una tabla usuarios con los campos, nombre y password, y otra tabla llamada grupos con los campos grupo, usuario, almacenando una entrada por cada usuario que tenga un grupo.

La configuración es la siguiente:

En primer lugar será necesario establecer el servidor, el usuario y la contraseña de la conexión con la base de datos, esto se hará escribiendo la siguiente línea en el fichero httpd.conf, y reemplazando los valores por los correctos:

**Auth\_MySQL\_Info <host><user><password>**

Para configurar el módulo será necesario establecer los valores de las siguientes directivas del módulo:

---

15 RKBLOG, [[http://www.rkblog.rk.edu.pl/w/p/mod\\_wsgi/](http://www.rkblog.rk.edu.pl/w/p/mod_wsgi/)], [Julio-2010]

16 DESARROLLOWEB.COM , [<http://www.desarrolloweb.com/articulos/2511.php>], [Consulta: Marzo-2010]



- **Auth\_MySQL\_DB:** nombre de la base de datos.
- **Auth\_MySQL\_Password\_Table:** nombre de la base de datos que contiene los nombres de usuario y las contraseñas.
- **Auth\_MySQL\_Username\_Field:** nombre del campo de la tabla que contiene al usuario.
- **Auth\_MySQL\_Password\_Field:** nombre del campo de la tabla que contiene la contraseña del usuario.
- **Auth\_MySQL\_Group\_Table:** nombre de la tabla que contiene los usuarios y los grupos a los que pertenecen.
- **Auth\_MySQL\_Group\_Field:** nombre del campo de la tabla que contiene el nombre del grupo
- **Auth\_MySQL\_Empty\_Passwords on/off:** para determinar si se aceptan usuarios con contraseñas vacías.
- **Auth\_MySQL\_Encryption\_Types:** para especificar el tipo de encriptación que se utilizará para almacenar la clave del usuario dentro de la base de datos.
- **Auth\_MySQL\_Non\_Persistent on/off:** para especificar si se desean utilizar conexiones persistentes.

Además será necesario establecer las siguientes directivas de core para su correcto funcionamiento:

- **AuthName:** con el nombre del dominio.
- **AuthType:** establecido a Basic

## 6.6. Mod\_Proxy<sup>17</sup>

Apache cuenta con el modulo proxy el cual es un complemento para hacer de gateway/proxy (no como un router, pero si como un intermediario antes de llegar hacia el destino final).

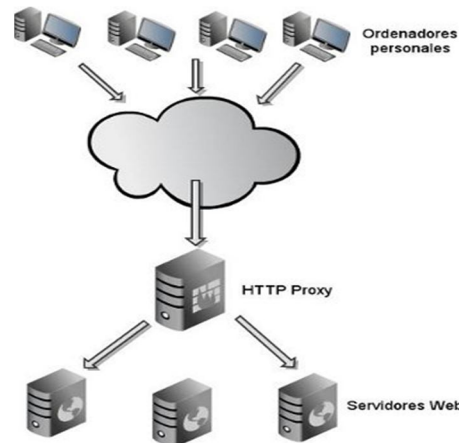


Fig.1: Funcionamiento del mod\_proxy

### 6.6.1. Mod\_proxy Reverse Proxy<sup>18</sup>

El funcionamiento del Reverse Proxy es totalmente transparente para el cliente como si fuera un Web Server único, el cliente hace los Requests del contenido en el espacio de nombres dentro de Reverse Proxy y es él quien decide si entregar estos contenidos o no, luego los contenidos son devueltos hacia el gateway y mostrados al usuario final.



Fig.2: Funcionamiento del mod\_proxy Reverse Proxy

<sup>17</sup> KAMUS BLOG, [[http://www.korporation.cl/blog/2009/03/acceso-filtrado-con-apache-y-mod\\_proxy/](http://www.korporation.cl/blog/2009/03/acceso-filtrado-con-apache-y-mod_proxy/)], [Julio-2010]

<sup>18</sup> RM-RF.ES, [<http://rm-rf.es/configurar-apache-como-reverse-proxy-proxy-inverso/>], [Febrero-2010]

Un *reverse proxy* es un servidor proxy instalado en el domicilio de uno o más servidores web. Todo el tráfico entrante de Internet y con el destino de uno de esos servidores web pasa a través del servidor proxy. Hay varias razones para instalar un “reverse proxy”:

- **Seguridad:** el servidor proxy es una capa adicional de defensa y por lo tanto protege los servidores web.
- **Cifrado / Aceleración SSL:** cuando se crea un sitio web seguro, habitualmente el cifrado SSL no lo hace el mismo servidor web, sino que es realizado por el “reverse proxy”, el cual está equipado con un hardware de aceleración SSL (Security Sockets Layer).
- **Distribución de Carga:** el “reverse proxy” puede distribuir la carga entre varios servidores web. En ese caso, el “reverse proxy” puede necesitar reescribir las URL de cada página web (traducción de la URL externa a la URL interna correspondiente, según en qué servidor se encuentre la información solicitada).
- **Caché de contenido estático:** Un “reverse proxy” puede descargar los servidores web almacenando, contenido estático como imágenes u otro contenido gráfico.

### 6.6.2. Ejemplo de mod\_proxy Reverse Proxy<sup>19</sup>

```
ProxyRequests Off
<Proxy *>
Order deny,allow
Allow from all
</Proxy>
ProxyPass /foo http://foo.example.com/bar
ProxyPassReverse /foo http://foo.example.com/bar
```

---

<sup>19</sup> APACHE SOFTWARE FOUNDATION , [http://http.apache.org/docs/2.0/es/ mod/mod\_proxy.html], [Enero - 2010 ]

## CAPÍTULO VII

### 7. BASE DE DATOS

#### 7.1. SQLAlchemy

##### 7.1.1. Definición<sup>20</sup>

SQLAlchemy es el kit de herramientas de SQL Python y relacional de objetos Mapper que da a los desarrolladores de aplicaciones toda la potencia y flexibilidad de SQL.

Proporciona una gama completa de patrones bien conocidos persistencia a nivel de empresa, diseñado para el acceso de base de datos eficiente y de alto rendimiento, adaptado a un lenguaje simple y Pythonic dominio.

##### 7.1.2. Filosofía de SQLAlchemy

SQLAlchemy no considera que las bases de datos sólo como colecciones de cuadros, sino que los ve como motores de álgebra relacional. Su objeto mapper relacional permite a clases para asignar la base de datos en más de una forma. SQL construye no sólo tienes que seleccionar las tablas de sólo-que también puede seleccionar de combinaciones, subconsultas, y los sindicatos. Así, las relaciones de base de datos y modelos de dominio objeto puede ser disociada limpiamente desde el principio, lo que permite a ambas partes a desarrollar todo su potencial.

El objetivo principal de SQLAlchemy es cambiar la manera de pensar sobre las bases de datos y SQL.

Lo más importante es SQLAlchemy no es sólo un ORM. Su capa de abstracción de datos permite la construcción y manipulación de expresiones SQL en una plataforma de manera agnóstico, y ofrece fácil de usar y súper objetos de resultados, así como creación de la tabla y la reflexión utilidades esquema. Ningún objeto de mapeo relacional de lo que está implicado hasta de importar el ORM paquete.

---

<sup>20</sup> SQLALCHEMY.ORG, [<http://www.sqlalchemy.org/>], [Abril-2010]

## 7.2. SQLite

### 7.2.1. Definición<sup>21</sup>

SQLite es una base de datos muy similar a la conocida Access del mundo Windows pero a diferencia de esta posee una serie de ventajas que la hacen interesante de aplicar. Para comenzar es multiplataforma y cumple con los estándares (en su mayoría) SQL92 por lo que su sintaxis y forma de uso casi no posee curva de aprendizaje a los conocedores de SQL y sus MySQL, porque además como este último (implementación de mysql en php) sqlite también está contemplada en el tratamiento dinámico de php profusamente.

SQLite se puede usar en modo ventana de comandos (Shell) o embebido en aplicaciones de código (casi cualquier lenguaje de programación directamente o indirectamente) por ejemplo c, c++, bash etc. En aplicaciones se puede utilizar con OpenOffice mediante la aplicación de los drivers ODB adecuados.

SQLite es un motor que trabaja embebido con la aplicación que lo use (en el servidor, en programas escritos por nosotros, etc) y otra de sus ventajas es la portabilidad.

SQLite es ideal para trabajar con volúmenes medianos o pequeños de información, de manera ágil y eficiente. Aunque sus diseñadores aducen que es posible manejar bases de datos de 2 terabytes sin mayores inconvenientes

### 7.2.2. Características<sup>22</sup>

- **Tamaño:** SQLite tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- **Rendimiento de base de datos:** SQLite realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- **Portabilidad:** se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- **Estabilidad:** SQLite es compatible con ACID, reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.
- **SQL:** implementa un gran subconjunto de la ANSI – 92 SQL estándar, incluyendo sub-consultas, generación de usuarios, vistas y triggers.

---

21 SCRIBD, [<http://www.scribd.com/doc/7253619/SQLite-Introduccion>], [Abril-2010]

22 APLICACIONESEMPRESARIALES.COM, [<http://www.aplicacionesempresariales.com/sqlite-el-motor-de-base-de-datos-agil-y-robusto.html>], [Junio – 2010]

- **Interfaces:** cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, Ruby, Tcl, groovy, etc.
- **Costo:** SQLite es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.

### 7.3. PostgreSQL<sup>23</sup>

#### 7.3.1. Definición

**PostgreSQL** es un servidor de base de datos objeto relacional libre, liberado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

#### 7.3.2. Características

Algunas de sus principales características son, entre otras:

##### 7.3.2.1. Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

##### 7.3.2.2. Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.

---

<sup>23</sup> ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/PostgreSQL>], [Consulta: Julio-2010]



- Direcciones MAC.
- Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL.

### 7.3.3. Características Generales

- Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (*foreign keys*).
- Disparadores (*triggers*): Un disparador o *trigger* se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:
  - El nombre del disparador o *trigger*
  - El momento en que el disparador debe arrancar
  - El evento del disparador deberá activarse sobre...
  - La tabla donde el disparador se activará
  - La frecuencia de la ejecución
  - La función que podría ser llamada

Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (*triggers*).

- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Soporte para transacciones distribuidas. Permite a PostgreSQL integrarse en un sistema distribuido formado por varios recursos (p.ej, una base de datos PostgreSQL, otra Oracle, una cola de mensajes IBM MQ JMS y un ERP SAP) gestionado por un servidor de aplicaciones donde el éxito ("commit") de la transacción global es el resultado del éxito de las transacciones locales.

#### 7.3.4. Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Los disparadores (*triggers* en inglés) son funciones enlazadas a operaciones sobre los datos.

Algunos de los lenguajes que se pueden usar son los siguientes:

- Un lenguaje propio llamado PL/PgSQL (similar al PL/SQL de oracle).
- C.
- C++.
- Java PL/Java web.
- PL/Perl.
- pI PHP.
- PL/Python.
- PL/Ruby.
- PL/sh.
- PL/Tcl.
- PL/Scheme.
- Lenguaje para aplicaciones estadísticas R por medio de PL/R.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés).

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados" (stored procedures en inglés).

## CAPÍTULO VIII

### 8. TGWEBSERVICES<sup>24</sup>

#### 8.1. Definición

TGWebServices proporciona una super API simple para crear servicios web que están disponibles a través de SOAP, HTTP, XML, y JSON HTTP. El API SOAP WSDL se genera automáticamente para el Python utilizado e incluso genera suficiente información para diferentes tipos de lenguajes con tipos estáticamente (Java y C #) para generar el código en el extremo de cliente.

#### 8.2. Características

- Manera más fácil de exponer una API de servicios Web
- Apoya SOAP, HTTP + XML, JSON + HTTP
- Productos envueltos documento / literal SOAP, que es el formato más ampliamente compatibles
- Proporciona suficiente información de tipo de lenguajes con tipos estáticamente para generar interfaces usables convenientemente
- También aceptan las instancias de sus clases como entrada
- Funciona con TurboGears 2.x

#### 8.3. Ejemplo de Servidor<sup>25</sup>

```
from tgwebservices.controllers import WebServicesRoot, wsexpose,
wsvalidate
class MyWSController(WebServicesRoot):
    @wsexpose(int)
    @wsvalidate(int, int)
    def multi(self, a, b):
    return a*b
```

24 PYTHON.ORG, [http://pypi.python.org/pypi/TGWebServices/2.0.0a2], [Noviembre-2009]

25 TURBOGEARS.ORG, [http://docs.turbogears.org/1.0/WebServices\_with\_TGWebServices], [Noviembre-2009]



## 8.4. Ejemplo de Cliente

```
#!/php (-)
<?php
ini_set("soap.wsdl_cache_enabled", 0);
$client = new
SoapClient('http://192.168.1.1:8080/myservices/soap/api.wsdl');
print_r($client->multi(3, 4));
?>
```

## CAPÍTULO IX

### 9. WEB SERVICES

#### 9.1. Historia de los Servicios Web

“En los primeros computadores corría un solo programa a la vez, pero en la medida que en un mismo computador podían correr varios programas al mismo tiempo, surgió la necesidad de contar con mecanismo de comunicación entre ellos, esto se llamó comunicación Task to Task y, este mecanismo a evolucionado debido que los computadores conforman redes. Por tanto, esta comunicación debe poder efectuarse entre un programa X, que corre en el computador Alfa, y otro programa Y, que corre en el computador Beta.

Para que esta comunicación funcione, primero debe existir un medio de comunicación entre el computador Alfa y el computador Beta; esto lo ha resuelto la Internet. Y segundo, el programa X debe saber conversar con el programa X. Para que esto ocurra el programador a cargo de X debe conocer de Y. A su vez el programador a cargo de Y debe conocer de X, por lo menos en los que se refiere al intercambio de datos. Esto hace que si no hay acuerdo entre el programador de X y el programador de Y, no hay comunicación posible.

La magia de los Web Services está en que el programador de X puede crear un Web Service para transferir datos sin necesidad de conocer al programador Y, ni a los programas que éste tiene a cargo. De modo que quien quiera recibir los datos solo necesita usar el Web Service y punto. Esto significa que pueden existir transferencias de datos entre distintas aplicaciones que funcionan en varios computadores, con distintos sistemas operativos, lenguajes de programación (PHP, C#, Java.), y que pertenezcan a diferentes empresas o instituciones.<sup>26</sup>

“Anteriormente se habían realizado intentos de crear estándares pero fracasaron o no tuvieron el suficiente éxito, algunos de ellos son DCOM y CORBA, por ser

---

26 DESARROLLOWEB.COM , [http://www.desarrolloweb.com/articulos/1883.php], [Consulta: Marzo-2010]



dependientes de la implementación del vendedor DCOM - Microsoft, y CORBA - ORB. Otro gran problema es que se hacía uso de RPC (Remote Procedure Call) para realizar la comunicación entre diferentes nodos. Es por esto que en 1999 se comenzó a plantear un nuevo estándar, al cual se lo terminó utilizando XML, SOAP, WSDL, y UDDI.<sup>27</sup>

## 9.2. Definición<sup>28</sup>

Web Services describe una forma estandarizada de integrar aplicaciones WEB mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet, permiten a las organizaciones intercambiar datos sin necesidad de conocer los detalles de sus respectivos Sistemas de Información.

A diferencia de los modelos Cliente/Servidor, tales como un servidor de páginas Web, los Web Services no proveen al usuario una interfaz gráfica (GUI). En vez de ello, comparten la lógica del negocio, los datos y los procesos, por medio de una interfaz de programas a través de la red. Es decir conectan programas, por tanto son componentes de software que no interactúan directamente con los usuarios.

Los Web Services permiten a distintas aplicaciones, de diferentes orígenes, comunicarse entre ellos sin necesidad de escribir programas costosos, esto porque la comunicación se hace con XML. Los Web Services no están ligados a ningún Sistema Operativo o Lenguaje de Programación. Por ejemplo, un programa escrito en Java puede conversar con otro escrito en Pearl; Aplicaciones Windows puede conversar con aplicaciones Unix. Por otra parte los Web Services no necesitan usar browsers (Explorer) ni el lenguaje de especificación HTML.

El modelo de computación distribuida de los Web Services permite la comunicación de aplicación a aplicación. Por ejemplo, la aplicación que procesa las órdenes de compra se puede comunicar con el sistema de inventarios, tal que este último le puede informar a la aplicación de compras cuales ítems deben comprarse por estar bajo su nivel mínimo. Dado el nivel integración que proveen para las aplicaciones.

---

27 MSAFFIRIO, [<http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>], , [Abril-2010]

28 GRUPO DE INVESTIGACIÓN FIEC – ESPOL, [ <http://blog.espol.edu.ec/taws/category/web-services/> ] , [Enero - 2010]

### 9.3. Características<sup>29</sup>

La tecnología de web services posee varias características entre las cuales se pueden citar:

- **Interoperabilidad:** Los Servicios Web se pueden ser consumir por clientes de otras plataformas.
- **Acceso externo desde Internet:** realizan una buena gestión para los accesos que provienen de clientes de Internet.
- **Tipos de datos de las Interfaces:** Los tipo de datos definidos para los Servicios Web se corresponde con los tipos de datos definidos por la mayoría de lenguajes de programación.
- **Uso de los estándares de Internet:** Los servicios Web utilizan los estándares de Internet y evitan, en la medida de lo posible, reinventar soluciones a problemas que ya están resueltas.
- **Soporte de cualquier lenguaje:** La implementación de un Servicio Web no está ligada a un particular lenguaje de programación. Esta es una gran ventaja frente a otras tecnologías como Java RMI, que está completamente ligada al uso de lenguaje Java, haciendo realmente difícil hacer una llamada a un objeto Java desde un objeto Visual Basic o Perl. De este modo, un cliente puede implementar o usar un Servicio Web independientemente del lenguaje de programación en el que fue implementado.
- **Soporte para cualquier infraestructura de componentes distribuidas:** Los Servicios Web no están ligados a una arquitectura de componentes en particular. Los protocolos facilitan a nivel base la comunicación entre las distintas infraestructuras de objetos distribuidos. Por este motivo, únicamente es necesario preocuparse del desarrollo y utilización de Servicios Web.

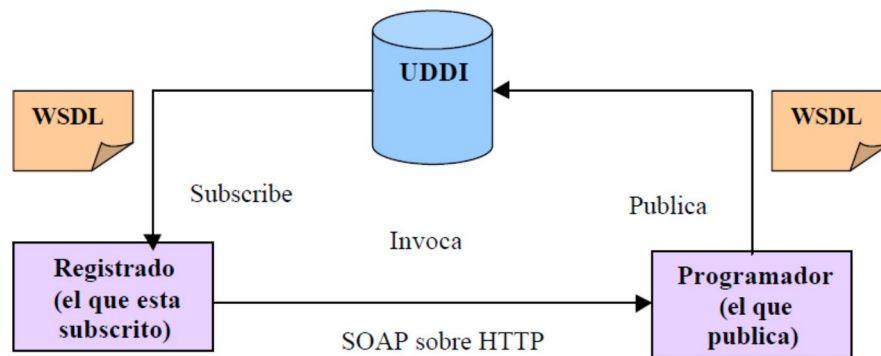
---

29 GXTECHNICAL, [http://www.gxtechnical.com/gxdlsp/pub/GeneXus/Internet/TechnicalPapers/Web\_Services.htm], [Noviembre-2009]

#### 9.4. Arquitectura de los Servicios Web

En una arquitectura de Web Services hay dos partes claramente diferenciables, el modo de utilizar un Web Services y cómo desarrollarlo. Existen varias partes implicadas y pasos necesarios para publicar un Web Service ya desarrollado y cómo puede ser utilizado, los cuales se detallan a continuación.

- El programador desarrolla el Web Service
- El programador describe el Web Service en un fichero WSDL
- El programador publica el Web Service en un directorio como UDDI
- La persona suscrita al directorio busca el Web Service
- La persona suscrita al directorio invoca el servicio con SOAP
- La persona suscrita al directorio recibe la respuesta mediante SOAP.



**Fig.3: Arquitectura de Servicios Web**

La arquitectura necesaria para el desarrollo de Web Services es la de un servidor que contenga las herramientas adecuadas para el soporte al desarrollo de este tipo de tecnología. Estas herramientas proporcionan el entorno de desarrollo de Web Services y la gestión de invocaciones de los servicios Web.

#### 9.5. Funcionamiento de los Web Services

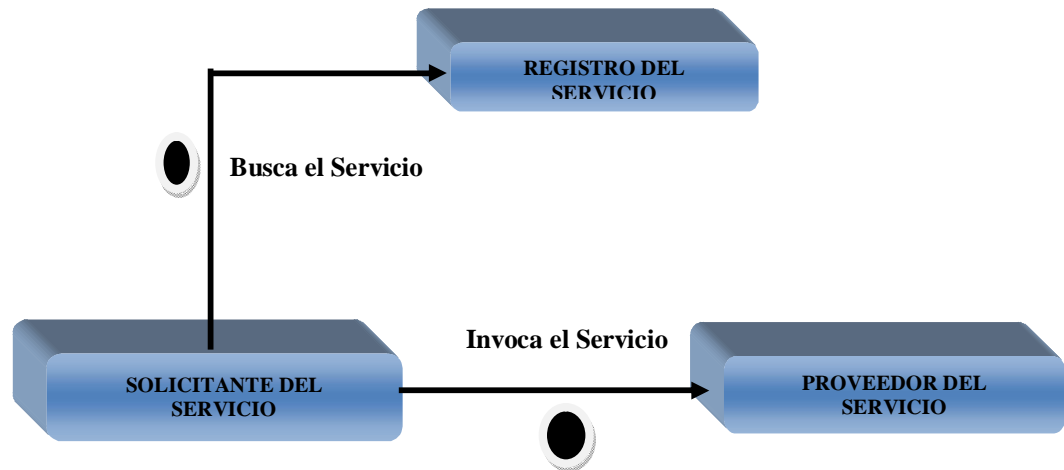
Existen tres principales funciones dentro de la arquitectura de un web services:

- **Proveedor de Servicio:** Este es el proveedor del web services. El proveedor del servicio está encargado de implementar el servicio, así hacerlo disponible en



Internet.

- **Solicitante del Servicio:** Este es cualquier consumidor del web services. El solicitante utiliza un servicio existente en el web al abrir una conexión de red y enviar una petición con XML.
- **Registro del Servicio:** Este es un directorio lógicamente centralizado de servicios. El registro provee un repositorio central donde los desarrolladores pueden publicar nuevos servicios, así como encontrar los existentes. Él, por lo tanto, sirve como una cámara de compensación centralizada para compañías y sus servicios.



**Fig. 4: Funcionamiento de un Web Services**

## 9.6. Pila de Protocolo de los Web Services

“La Pila de protocolos para Servicios Web es una colección de protocolos para redes de Computadores que son utilizados para definir, localizar, implementar y hacer que un Servicio Web interactúe con otro. La Pila de Protocolos para servicios está comprendida principalmente por cuatro áreas:

### 9.6.1. Servicio de Transporte

Responsable del transporte de mensajes entre las Aplicaciones de red y los protocolos en los cuales se incluyen protocolos tales como:<sup>30</sup>

30 ENCICLOPEDIA LIBRE WIKIPEDIA, [[http://es.wikipedia.org/wiki/Web\\_Services\\_Protocol\\_Stack](http://es.wikipedia.org/wiki/Web_Services_Protocol_Stack)], [Consulta: Julio-2010]

### 9.6.1.1. HTTP

El protocolo de transferencia de hipertexto es el protocolo usado en cada transacción de la Web (www). Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición se lo conoce como agente del usuario. A la información transmitida se la llama recurso y se la identifica mediante un URL. Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, etc.

### 9.6.1.2. SMTP

Protocolo Simple de Transferencia de Correo, es un protocolo de la capa de aplicación. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (teléfonos móviles, etc.). Está definido en el RFC 2821 y es un estándar oficial de Internet.

### 9.6.1.3. FTP

Protocolo de Transferencia de Archivos, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

El Servicio FTP es ofrecido por la capa de Aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y el 21.

## 9.6.2. “Mensajería XML

Responsable por la codificación de mensajes en un formato común XML así que ellos puedan ser entendidos en cualquier extremo de una conexión de red. Actualmente, esta área incluye protocolos tales como:”<sup>31</sup>

### 9.6.2.1. “XML-RCP

Es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. Es un protocolo muy simple

---

31 DESARROLLOWEB.COM, [<http://www.desarrolloweb.com/articulos/460.php>], [Consulta: Diciembre-2009]

ya que sólo define unos cuantos tipos de datos y comandos útiles, además de una descripción completa de corta extensión. La simplicidad del XML-RPC está en contraste con la mayoría de protocolos RPC que tiene una documentación extensa y requiere considerable soporte de software para su uso.”<sup>32</sup>

#### **9.6.2.2. “SOAP**

Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP está diseñado para llevar acabo intercambios de información en XML para sistemas altamente distribuidos, en específico: Internet.”<sup>33</sup>

#### **9.6.3. “Descripción del Servicio**

Usado para describir la interfaz pública de un Servicio Web específico.”<sup>34</sup>

##### **9.6.3.1. “WSDL**

Es un formato XML que se utiliza para describir servicios Web, escribe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.”

#### **9.6.4. “Descubrimiento de servicios**

Centraliza servicios en un registro común tal que los servicios Web de la red puedan publicar su localización y descripción, y hace que sea fácil descubrir que servicios están disponibles en la red.”

##### **9.6.4.1. “UDDI**

Son las siglas del catálogo de negocios de Internet denominado Universal Description, Discovery and Integration. UDDI es uno de los estándares básicos de los servicios Web

---

32 ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/XML-RPC>], [Consulta: Julio-2010]

33 ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/SOAP>], [Consulta: Julio-2010]

34 ENCICLOPEDIA LIBRE WIKIPEDIA, [[http://es.wikipedia.org/wiki/Web\\_Services\\_Protocol\\_Stack](http://es.wikipedia.org/wiki/Web_Services_Protocol_Stack)], [Consulta: Julio-2010]

cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.”<sup>35</sup>

## 9.7. MENSAJERÍA XML<sup>36</sup>

"XML es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Vamos a ver a lo largo de varios capítulos una introducción al mundo XML, es decir, al lenguaje así como a las tecnologías que trabajan con él, sus usos, ventajas y modos de llevar a cabo las tareas.

XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. El XML juega un papel importantísimo en este mundo actual, que tiende a la globalización y la compatibilidad entre los sistemas, ya que es la tecnología que permitirá compartir la información de una manera segura, fiable, fácil. Además, XML permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

XML, o mejor dicho, el mundo XML no es un lenguaje, sino varios lenguajes, no es una sintaxis, sino varias y no es una manera totalmente nueva de trabajar, sino una manera más refinada que permitirá que todas las anteriores se puedan comunicar entre sí sin problemas, ya que los datos cobran sentido. XML es interesante en el mundo de Internet y el e-bussiness, ya que existen muchos sistemas distintos que tienen que comunicarse entre sí, pero como se ha podido imaginar, interesa por igual a todas las ramas de la informática y el tratamiento de datos, ya que permite muchos avances a la hora de trabajar con ellos.

---

35 ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/UDDI>], [Consulta: Agosto-2010]

36 DESARROLLOWEB.COM, [<http://www.desarrolloweb.com/articulos/450.php>], [Consulta: Noviembre-2009]

### 9.7.1. Sintaxis XML<sup>37</sup>

Tiene una sintaxis extremadamente sencilla, el XML es un 10% del SGML (Lenguaje de Marcado Generalizado), porque las normas que tiene son muy simples. Se escribe en un documento de texto ASCII, igual que el HTML y en la cabecera del documento se tiene que poner el texto:

```
<?xml version="1.0"?>
```

En el resto del documento se deben escribir etiquetas como las de HTML, las etiquetas que nosotros queramos, por eso el lenguaje se llama XML, lenguaje de etiquetas extendido. Las etiquetas se escriben anidadas, unas dentro de otras.

```
<ETIQ1>...<ETIQ2>...</ETIQ2>...</ETIQ1>
```

Cualquier etiqueta puede tener atributos. Le podemos poner los atributos que queramos.

```
<ETIQ atributo1="valor1" atributo2="valor2"...>
```

Los comentarios de XML se escriben igual que los de HTML.

```
<!-- Comentario -->
```

Para definir qué etiquetas y atributos debemos utilizar al escribir en XML tenemos que fijarnos en la manera de guardar la información de una forma estructurada y ordenada.

Por ejemplo, si deseamos guardar la información relacionada con una película en un documento XML podríamos utilizar un esquema con las siguientes etiquetas.

```
<?xml version="1.0"?>
<PELICULA nombre="El Padrino" año=1985>
<PERSONAL>
</DIRECTOR nombre="Georgie Lucar">
```

---

37 W3C, [<http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>], [Enero-2010]

```
</INTERPRETE nombre="Marlon Brando" interpreta-a="Don Corleone">
</INTERPRETE nombre="Al Pacino" interpreta-a="Michael Corleone">
</PERSONAL>
</ARGUMENTO descripción="Película mafias sicilianas en Estados Unidos">
</PELICULA>
```

Se han puesto las etiquetas que se han creído conveniente para poner este ejemplo y se lo anidado de manera que la etiqueta más grande es la PELICULA y dentro de ella tenemos el PERSONAL y el ARGUMENTO.

A su vez, dentro de PERSONAL se ha puesto al DIRECTOR como a los actores (INTERPRETE).

## 9.8. SOAP<sup>38</sup>

SOAP es un protocolo para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP. Está basado en XML, esto facilita la lectura por parte de los humanos.

Existen múltiples tipos de modelos de mensajes en SOAP pero, por lejos, el más común es el RPC, en donde un nodo de red (el cliente) envía un mensaje de solicitud a otro nodo (el servidor) y el servidor inmediatamente responde el mensaje al cliente.

Los mensajes SOAP, son independientes del sistema operativo, y pueden transportarse en varios protocolos de Internet como: SMTP, MIME y HTTP.

SOAP al principio significaba Simple Object Access Protocol, luego fue Service Oriented Architecture Protocol, pero actualmente es simplemente SOAP.

El acrónimo inicial, fue dejado de lado en la versión 1.2, cuando se volvió una recomendación de la W3C el 24 de junio de 2003, porque su nombre daba a confusión.

---

38 ALESA , [<http://www.alegsa.com.ar/Dic/soap.php>], [Diciembre - 2009]

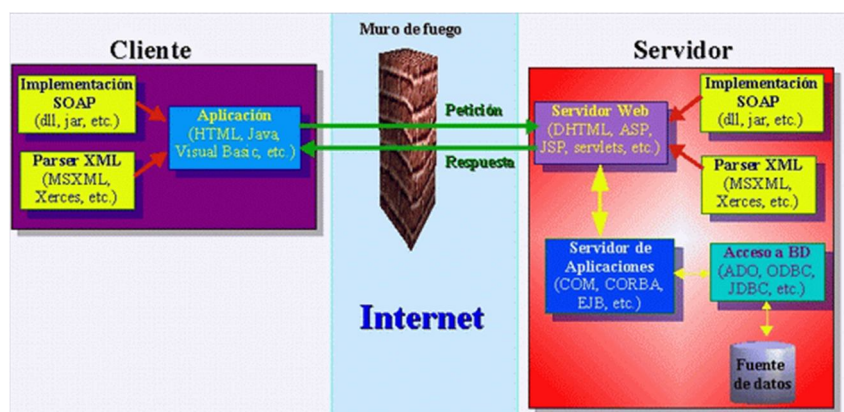
### 9.8.1. Ventajas de SOAP<sup>39</sup>

- **No está asociado con ningún lenguaje:** los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista pero los desarrolladores responsables de mantener antiguas aflicciones heredadas podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación, como en Java, y la plataforma como Microsoft .Net.
- **No se encuentra fuertemente asociado a ningún protocolo de transporte:** La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- **No está atado a ninguna infraestructura de objeto distribuido** La mayoría de los sistemas de objetos distribuidos se pueden extender, y ya lo están alguno de ellos para que admitan SOAP.
- **Aprovecha los estándares existentes en la industria:** Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML.
- **Permite la interoperabilidad entre múltiples entornos:** SOAP se desarrolló sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dicho estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas.

---

39 DESARROLLOWEB.COM , [http://www.desarrolloweb.com/articulos/1557.php], [Consulta: Marzo-2010]

### 9.8.2. Funcionamiento de SOAP<sup>40</sup>



**Fig. 5: Funcionamiento de SOAP**

La especificación SOAP menciona que las aplicaciones deben ser independientes del lenguaje de desarrollo, por lo que las aplicaciones cliente y servidor pueden estar escritas con HTML, DHTML, Java, Visual Basic u otras herramientas y lenguajes disponibles. Lo importante es tener alguna implementación de SOAP y enlazar sus librerías con la aplicación. Aunque esto no es estrictamente necesario, es preferible trabajar usando dichas librerías, con el fin de no reescribir un código ya probado.

Las peticiones con el uso del protocolo HTTP emplean el comando POST para transmitir información entre el cliente y el servidor.

Por otra parte el término Object en el nombre significa que se adhiere al paradigma de la programación orientada a objetos.

SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí.

Estos mensajes constan de 3 secciones: envelope, header y body.

<sup>40</sup> MONOGRAFÍAS, [<http://www.monografias.com/trabajos29/protocolo-acceso/protocolo-acceso.shtml>], [Abril-2010]





**Fig.6: Estructura de Mensaje SOAP**

Dónde:

- **Envelope(envoltura):** Es el elemento raíz del mensaje para describir su contenido y la forma de procesarlo.
- **Header(encabezado):** Es la información de identificación del contenido. Un grupo de reglas de codificación para expresar las instancias de tipos de datos definidos por la aplicación.
- **Body(cuerpo):** Es el contenido del mensaje. Una convención para representar las llamadas y las respuestas a procedimientos remotos.

### 9.8.3. Modelo de intercambio de mensajes

En el modelo de intercambio de mensajes tenemos:

- Los mensajes SOAP son transmisiones unidireccionales desde un emisor a un receptor.
- Se suelen combinar mensajes para implementar patrones, como petición/respuesta.
- Las implementaciones SOAP se pueden optimizar para explotar las características específicas de sistemas de red concretos.

## 9.9. WSDL<sup>41</sup>

WSDL son las siglas en inglés de "Lenguaje de Descripción de Servicios Web" (o "Web Services Description Language"), un lenguaje que está basado en XML y que permite la descripción de los servicios web desplegados. WSDL se utiliza también para la localización y ubicación de estos servicios en Internet.

Un documento WSDL no es más que un documento XML que describe ciertas características propias de un servicio web, así como su localización y aquellos parámetros y métodos que soporta.

### 9.9.1. Documento WSDL

Un documento WSDL define un servicio web utilizando a tal fin elementos XML, como:

- <portType> para las operaciones que proporciona el servicio web
- <message> para los mensajes que utiliza por el servicio web
- <types> para los tipos de datos que utiliza el servicio web
- <binding> para los protocolos de comunicaciones que utiliza el servicio web

Un documento WSDL tiene una estructura semejante a la siguiente:

```
<definitions>
  <types>los tipos de datos.. </types>
  <message> las definiciones del mensaje...</message>
  <portType> las definiciones de operación ...</portType>
  <binding> las definiciones de protocolo...</binding>
</definitions>
```

---

41 CIBERNETIA, [http://www.cibernetia.com/manuales/servicios\_web/4\_wsdl.php], [Febrero-2010]

### 9.9.2. Elementos de WSDL<sup>42</sup>

Los elementos del WSDL son los siguientes:

#### 9.9.2.1. Elemento type

El elemento Types contiene información de esquema referenciado en el documento WSDL. El sistema de tipos predeterminado que admite WSDL es de esquema de XML. Si se usa esquema de XML para definir los tipos que contiene el elemento Types el elemento schema aparecerá inmediatamente como elemento hijo.

#### 9.9.2.2. Elemento message

El elemento Message proporciona una abstracción común para el paso de mensajes entre el cliente y el servidor. Como puede utilizar múltiples formatos de definición de esquema en documento WSDL es necesario de disponer de un mecanismo común de identificar los mensajes. El elemento Message proporciona este nivel común de abstracción al que se hará referencia en otras partes del documento WSDL.

#### 9.9.2.3. Elemento portType

El elemento portType contiene un conjunto de operaciones abstractas que representan los tipos de correspondencia que pueden producirse entre el cliente y el servidor. Para los Servicios Web de estilo RPC se puede pensar en un portType como una definición de internas en donde cada método se puede definir como una operación.

#### 9.9.2.4. Elemento binding

El elemento binding contiene las definiciones de la asociación de un protocolo como SOAP a un determinado bindingType. Las definiciones binding especifican detalles de formatos del mensaje y el protocolo. Por ejemplo, la información de asociación especifica si se puede acceder a una instancia de un portType de forma RPC.

#### 9.9.2.5. Elemento service

Un servicio es un grupo de puertos relacionados y se definen en el elemento service. Un puerto es un extremo concreto de un Servicio Web al que se hace referencia por una

---

42 DESARROLLOWEB.COM , [<http://www.desarrolloweb.com/articulos/1581.php>], [Consulta: Marzo-2010]



dirección única. Los puertos que se definen en determinado servicio son independientes. Por ejemplo, la salida de un puerto que no puede utilizarse como una entrada de otro.

#### **9.9.2.6. Elementos de Extensibilidad**

Los elementos de extensibilidad se utilizan para representar determinadas tecnologías. Por ejemplo, se puede utilizar los elementos de extensibilidad para especificar el idioma en que se utiliza en el esquema de los elementos types.

## CAPÍTULO X

### 10. LINESURVEY<sup>43</sup>

**LimeSurvey** (anteriormente **PHPSurveyor**) es una aplicación open source para la aplicación de encuestas en línea, escrita en PHP y que utiliza bases de datos MySQL, PostgreSQL o MSSQL. Esta utilidad brinda la posibilidad a usuarios sin conocimientos de programación el desarrollo, publicación y recolección de respuestas de sus encuestas. Las encuestas incluyen ramificación a partir de condiciones, plantillas y diseño personalizado usando un sistema de plantillas web, y provee utilidades básicas de análisis estadístico para el tratamiento de los resultados obtenidos. Las encuestas pueden tener tanto un acceso público como un acceso controlado estrictamente por las claves que pueden ser utilizadas una sola vez (tokens) asignadas a cada persona que participa en la encuesta. Además los resultados pueden ser anónimos, separando los datos de los participantes de los datos que proporcionan, inclusive en encuestas controladas.

#### 10.1. Características internacionales

LimeSurvey está disponible en más de idiomas y dialectos, y utiliza la codificación de caracteres UTF-8. Entre las traducciones primarias se incluyen: albanés, euskera, chino, croata, danés, holandés, finés, francés, gallego, alemán, griego, húngaro, hebreo, italiano, japonés, portugués, ruso, serbio, esloveno, español, sueco, y muchas otras traducciones parciales

Es una plataforma de servicios en línea es una página web que permite el acceso diferenciado de los/as usuarios/as según su nivel de responsabilidad para la realización de una tarea compartida: los/as usuarios/as con privilegios de administración tendrán acceso a las diferentes opciones de gestión de la herramienta (según los permisos que se les haya conferido), sin embargo la mayoría de las personas que accederán a la página web únicamente podrán utilizar la herramienta (en este caso completar una encuesta), pero sin posibilidad de alterar aspecto alguno de la misma.

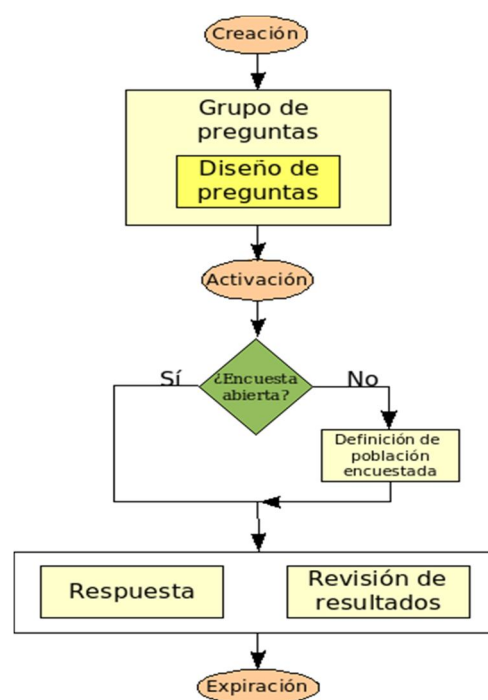
---

<sup>43</sup> ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/LimeSurvey>], [Consulta: Febrero-2011]

Para distinguir entre los diferentes roles se realiza una autenticación por medio de un nombre y una contraseña, del mismo modo que cualquier otro servicio de internet, y será responsabilidad de la administración técnica de la plataforma el delegar los diferentes privilegios a los usuarios que sean creados.

## 10.2. Administración de Encuestas<sup>44</sup>

Se puede esquematizar el trabajo global que será realizado sobre la plataforma en la siguiente ilustración.



**Fig.7: Administración de Encuestas Limesurvey**

Como puede apreciarse, el trabajo de encuestar a través de la plataforma es muy simple, lo que permitirá enfocar el esfuerzo en la tarea más complicada intelectualmente y a la vez más útil de todo el proceso: el análisis de los resultados y la toma de decisiones.

A continuación se explica a grandes rasgos el flujo de trabajo para la creación y administración de encuestas en línea con esta herramienta.<sup>45</sup>

44 LIMESURVEY.ORG, [http://docs.limesurvey.org/tikiindex.php?page=Administraci%C3%B3n+de+Encuestas], [Enero-2011]

45 CSIRC, [http://csirc.ugr.es/manualLimeSurvey\_CSIRC\_v5.pdf], [Enero-2011]

### **10.2.1. Cómo se construye una encuesta en LimeSurvey**

El formato habitual de una encuesta está compuesto por grupos que contienen las preguntas; sin embargo, al momento de crear la encuesta se puede cambiar esta estructura por el formato pregunta por pregunta o todas las preguntas en una sola página.

Las preguntas que componen la encuesta pueden ser de diversos tipos (respuesta simple, respuesta múltiple, valoración del 1 al 10, respuestas abiertas, etcétera). También se pueden ramificar las encuestas, es decir, presentar preguntas de acuerdo a las respuestas a preguntas anteriores.

### **10.2.2. Cómo se publica una encuesta**

Una vez terminada la encuesta, ésta debe activarse y se debe decidir si ésta será abierta o privada. En LimeSurvey se cuenta con la opción de habilitar un enlace que se puede hacer público para el llenado anónimo de la encuesta, o el mecanismo de generación de claves (tokens) para un grupo de personas, del cual se debe conocer su nombre y correo electrónico.

Con el listado de personas se puede enviar un correo electrónico de invitación que incluya la dirección específica asociada a los usuarios, e inclusive enviar recordatorios posteriormente.

El contenido del mensaje de invitación y el de recordatorio pueden ser personalizados.

### **10.2.3. Cómo se recolectan los resultados**

Aunque esta es la parte más deficiente en cuanto a funcionalidades, ya que no hay reportes muy elaborados disponibles en la misma plataforma, LimeSurvey si permite que los resultados de las encuestas puedan observarse vía web o exportarse en formato CSV (legible desde Microsoft Excel u OpenOffice.org Calc) para que puedan ser analizados.

### 10.3. Procesos por los que pasa la encuesta<sup>46</sup>

A lo largo del tiempo de vida de una encuesta, ésta pasará por diferentes procesos:

La **primera tarea** importante a llevar a cabo será realizar un diseño lo más completo posible, no sólo del listado de preguntas y posibles respuestas de la encuesta, sino un diseño global de la encuesta web, es decir, diseñar la/s ubicación/es de cada elemento en el proceso de la encuesta. Lo idóneo es agrupar las preguntas por grupos, para poder luego analizar los resultados más cómodamente.

Así, iremos creando los grupos, con sus respectivas preguntas, eligiendo el tipo de cada pregunta de entre los disponibles, y especificando las posibles respuestas de las mismas. Más adelante se especificará los posibles tipos de preguntas, así como las posibles respuestas y la posibilidad de elegir preguntas condicionadas en otras respuestas.

Otro aspecto importante del diseño será decidir el tipo de acceso que tendrán los usuarios a la encuesta. De igual forma, se explicará más detalladamente las posibilidades que nos ofrece esta plataforma.

La **segunda tarea**, una vez que hemos realizado todo el diseño y creación de la encuesta, será la de activar dicha encuesta para que sea accesible por los usuarios. Desde el momento en que habilitemos la encuesta, se crearán todas las estructuras necesarias en la base de datos para poder ir almacenando los resultados de los usuarios. En el proceso de activación es muy importante el tipo de acceso que hayamos elegido para nuestra encuesta. Si es un acceso público, no habrá ningún tipo de problema, cada usuario guardará sus datos en el sistema. Pero, si por el contrario, es un acceso restringido, se tendrá en cuenta la población definida (usuarios que podrán acceder a la encuesta) para archivar los resultados.

La **tercera tarea** será la de recoger los resultados obtenidos y hacer el estudio pertinente. El sistema permite, como ya explicaremos más adelante, exportar dichos datos a otras plataformas estadísticas, así como estudiar las gráficas comparativas de los resultados según los grupos de preguntas definidos.

---

46 EL MUNDO ES OPEN SOURCE, [<http://blogs.antartec.com/opensource/2010/03/limesurvey-todo-lo-que-necesitas-para-encuestas-en-linea/>], [Enero-2011]





La **cuarta tarea**, y última, será la de expirar la encuesta, es decir, deshabilitar el acceso de los usuarios a la misma. Como podremos ver más adelante existen dos formas de “deshabilitar” una encuesta; bien expirándola, o bien desactivándola.



# EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN



## 6. EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN

El uso de aplicaciones web se ha constituido en el nuevo recurso para la prestación de servicios y soporte informático a las instituciones, empresas y sociedad en general, es por ello que es evidente la importancia del uso de tecnologías web para el desarrollo de aplicaciones informáticas.

Además es necesario considerar que para el desarrollo de aplicaciones web, se existen una gran cantidad de tecnologías, muchas de ellas incompatibles entre sí, dificultando la integración de aplicaciones o sistemas informáticos, por esta razón surge la idea de hacer uso de servicios web (Web Services) que faciliten la intercomunicación e interoperabilidad entre diferentes arquitecturas, las cuales son la base principal dentro del desarrollo Web.

Un web services constituye un conjunto de protocolos cuya utilidad específica es permitir el intercambio de información entre aplicaciones, aun en el caso de que éstas estén programadas sobre tecnologías y lenguajes diferentes. Aunque los web services no son sino una más de las tecnologías que pueden implicar los proyectos de las instituciones públicas o las empresas debido a sus ventajas y cada vez mayor utilización en la sociedad informática en el desarrollo de software.

La perspectiva de la sociedad universitaria actual demanda que las instituciones de educación superior posean soluciones y mecanismos de administración y control de los procesos académicos, aprovechado los avances recientes de la tecnología informática, ya que dentro del proceso administrativo y gestión universitaria existen diferentes actividades que deben ser realizadas mediante soluciones informáticas adaptadas a las características de la institución y al sistema académico vigente.

Actualmente la Universidad Nacional de Loja cuenta con un Sistema de Gestión Académica, automatizado, integrado y adaptado a las especificaciones del Sistema Académico Modular por Objetos de Transformación SAMOT, desarrollado bajo el paradigma del software libre sobre la plataforma Linux, con el propósito de llevar una mejor organización en las actividades que realiza diariamente, con el objetivo de



permitir el control adecuado de información, en cuanto a lo referente a: planificación académica, matriculación, gestión de docentes, gestión de estudiantes, control académico, entre otros, son algunas de las múltiples prestaciones que brinda este sistema. Pero es importante destacar que este carece de un web services o biblioteca de métodos completa que le permita recuperar información del mismo, así como también integrar aplicaciones que no estén contempladas o desarrolladas bajo sus lineamientos técnicos y de planificación. Por esta razón surge la necesidad de desarrollar una biblioteca de métodos de recuperación de información basada en la tecnología web services la cual aporte la solución adecuada a este problema presentado.

Para poder dar cumplimiento a los objetivos planteados se realizó la investigación y análisis adecuados acerca del funcionamiento del Sistema de Gestión Académica y cada uno de los módulos que lo componen, en especial el modelo entidad-relación así como también la base de datos utilizados para la administración y almacenamiento de la información, además se llevó a cabo un estudio detallado mediante consultas e investigaciones sobre la tecnología de los Web Services con sus diferentes protocolos y herramientas para el análisis, diseño e implementación del sistema SGAWebServices.

El análisis de la biblioteca de métodos se lo realizó detalladamente tomando en cuenta las necesidades más primordiales que fueron proporcionadas por los posibles usuarios de ésta, estos requerimientos para mayor comprensión han sido descritos mediante el uso de las historias de usuarios, instrumento fundamental dentro de la Metodología de Desarrollo de Software XP en la cuales se puntualizó cada uno los métodos que formarían parte del SGAWebServices, para el diseño se utilizó la herramienta de modelado de UML que facilitó el desarrollo de los diferentes diagramas que posteriormente se utilizaron para la codificación e implementación de la aplicación, empleando el framework TurboGears2.0 para el desarrollo, también se utilizaron diferentes herramientas y protocolos para mejorar la implementación del Subsistema de métodos de la recuperación de información SGAWebServices para el Sistema de Gestión Académica de la Universidad Nacional de Loja.

Una vez desarrollada la aplicación fue necesario probar que la misma estaba funcionando correctamente, para lo cual se realizó la instalación, configuración y



modificación de la aplicación open source LimeSurvey desarrollada en PHP, que servirá para realizar una encuesta socioeconómica en línea a estudiantes de la Universidad Nacional de Loja. La encuesta creada gracias a esta aplicación permitió probar el funcionamiento del SGAWeb services debido a que utiliza dos de sus métodos, para esto fue necesaria la creación de un cliente web en PHP que acceda al mismo, probando así el correcto funcionamiento del servicio web y depurando cualquier error presentado.

Finalmente para dar fiel cumplimiento a todos los objetivos planteados se realizó la documentación clara, acerca del funcionamiento y manejo de los métodos desarrollados, indicando y explicando lo que es y hace la aplicación, la clasificación realizada por categorías de acuerdo al tipo de información que retorna cada método o servicio, y la descripción de los mismos en cuanto a los datos que deben enviar y los datos que van a obtener de cada servicio.



# DESARROLLO DE LA PROPUESTA ALTERNATIVA



## 7. DESARROLLO DE LA PROPUESTA ALTERNATIVA

La Universidad Nacional de Loja cuyo objetivo es la formación de profesionales propositivos capaces de resolver problemas dentro del entorno en el que se desenvuelven, con sólida base científico-técnica. Por esta razón se ha creído conveniente aportar con soluciones prácticas a los problemas que se presentan al surgir la necesidad de que la información fluya y sea accesible para quienes necesiten hacer uso de la información contenida en el Sistema de Gestión Académica (SGA) de la Universidad Nacional de Loja, así como que dicha información mantenga su validez e integridad.

En vista de que la recuperación de información desde aplicaciones que estén desarrolladas en lenguajes de programación o sistemas operativos diferentes al del SGA, se ve afectada por la inexistencia de servicios web que permitan la comunicación e interoperabilidad entre ellas. Por esta razón se optó por utilizar la tecnología de los Web Services, la misma que pretende mantener centralizados los datos con mayor flexibilidad a la hora de manejar dicha información por quien la necesite, sin tener que duplicar misma.

En base a lo expuesto anteriormente, se ha desarrollado de un sub sistema de servicios web basado en una biblioteca de métodos de recuperación de información del Sistema de Gestión Académica.

La Universidad Nacional de Loja cuyo objetivo es la formación de profesionales propositivos capaces de resolver problemas dentro del entorno en el que se desenvuelven, con una sólida base científico-técnica. Por esta razón hemos creído conveniente aportar con soluciones prácticas a los problemas que se presentan al surgir la necesidad de que la información fluya y sea accesible para quienes necesiten hacer uso de la información contenida en el Sistema de Gestión Académica (SGA) de la Universidad Nacional de Loja, así como que dicha información mantenga su validez e integridad.

En vista de que la recuperación de información desde nuevas aplicaciones que están desarrolladas en lenguajes de programación o sistemas operativos diferentes al del SGA, se ve afectada, debido a la inexistencia de servicios web que permitan la comunicación e interoperabilidad entre ellas. Razón por la cual se optó por la utilización de la tecnología de los Web Services, la misma que pretende mantener centralizados los datos con una mayor flexibilidad a la hora de utilizar dicha información por parte de quien la necesite, sin tener que duplicar información.

En base a lo expuesto anteriormente, se ha desarrollado de un sub sistema de servicios web basado en una biblioteca de métodos de recuperación de información del SGA

## 7.1. PRESENTACIÓN DE RESULTADOS MEDIANTE LA METODOLOGÍA XP

### 7.1.1. Análisis y Diseño

El desarrollo del SGAWeb services se realizó con el apoyo de la metodología de desarrollo XP, la misma que se basa en un modelo ágil, rápido, flexible y de calidad, diseño simple, pruebas continuas y trabajo directo con el usuario. Las personas e interacciones son más importantes que los procesos y herramientas, que el software funcione es más importante que documentación exhaustiva.

El proceso inició con la Planificación, donde se realizó una entrevista con el Ing. Patricio Valarezo Director del Proyecto denominado Sistema de Gestión Académica (SGA) de la Universidad Nacional de Loja, lo cual sirvió para fortalecer y aclarar el alcance, ámbito y prioridad del proyecto. En esta fase se analizó la situación actual del SGA frente a la problemática presentada haciendo un estudio del funcionamiento y características del mismo, se realizó un estudio y documentación de lo que son y hacen los web services o servicios web (**ver Anexo# 1**). Así mismo se llevaron a cabo encuestas dirigidas a posibles usuarios de la aplicación, para obtener los requerimientos que se constituyeron en la base para el desarrollo del SGAWeb services, en la creación del modelo de dominio, elaboración de casos de uso e historias de usuario, seguidamente se continuó con un diseño simple mediante la definición definitiva del modelo del dominio, de acuerdo a las entidades que forman parte del modelo del Sistema de Gestión Académica, se elaboraron también diagramas de secuencia, el



diagrama de clases final y modelo entidad relación. A continuación se detalla lo expuesto anteriormente:

#### 7.1.1.1. Obtención de Requerimientos

En el proceso de análisis se realizó una encuesta (**ver Anexo# 2**) para determinar y definir los métodos de la biblioteca de servicios que serían desarrollados, las mismas que fueron aplicadas a estudiantes, docentes y egresados de la Carrera de Ingeniería en Sistemas y a desarrolladores de la Unidad de Desarrollo de Software de la Universidad Nacional de Loja, con la finalidad de conocer si tenían conocimientos acerca de la tecnología de servicios web y qué información desearían obtener del Sistema de Gestión Académica de la Universidad Nacional de Loja, para el desarrollo de nuevas aplicaciones que tengan fines académicos.

Luego de aplicadas las encuestas se determinó a través del análisis de las mismas, los servicios que la mayoría de encuestados les interesaba obtener, llegando de esta manera a determinar los métodos o funciones que se implementarían a lo largo del desarrollo del proyecto. A continuación se listan todos los servicios requeridos:

- Obtener una lista de todas las áreas de la universidad
- Obtener los datos de una área
- Obtener las carreras de una área
- Obtener los datos de las carreras de la universidad
- Obtener todos los módulos de una carrera
- Obtener todos los paralelos de una carrera
- Verificar el usuario y clave de un docente
- Verificar el usuario y clave de un estudiante
- Obtener los datos personales de un docente
- Obtener los datos personales de un estudiante
- Verificar si un usuario es docente o estudiante y devolver los datos del mismo
- Obtener todos los periodos lectivos
- Obtener todas ofertas académicas de un periodo lectivo
- Obtener las fechas de matrículas ordinaria, extraordinaria y especial



- Ver el estado de los estudiantes de un paralelo: aprobado, reprobado o matriculado
- Obtener el plan de estudio de un paralelo
- Obtener la carga horaria de un docente
- Obtener las notas de cada unidad de un estudiante
- Conocer las carreras en las que está inscrito un estudiante
- Obtener la nota final y el porcentaje de asistencia de un estudiante en un módulo
- Obtener en número de estudiantes matriculados en una oferta académica
- Obtener en número de estudiantes matriculados en una área
- Obtener en número de estudiantes matriculados en una carrera
- Obtener en número de estudiantes matriculados en un modulo
- Obtener en número de estudiantes matriculados en un paralelo
- Obtener en número de estudiantes aprobados en una oferta académica
- Obtener en número de estudiantes aprobados en una área
- Obtener en número de estudiantes aprobados en una carrera
- Obtener en número de estudiantes aprobados en un modulo
- Obtener en número de estudiantes aprobados en un paralelo
- Obtener en número de estudiantes reprobados en una oferta académica
- Obtener en número de estudiantes reprobados en una área
- Obtener en número de estudiantes reprobados en una carrera
- Obtener en número de estudiantes reprobados en un modulo
- Obtener en número de estudiantes reprobados en un paralelo

#### 7.1.1.2. Modelo del Dominio

Tras la determinación de requerimientos, se realizó el estudio del modelo de clases del Sistema de Gestión Académica (SGA) con la finalidad de crear el modelo del dominio para el SGAWeb services en base a las necesidades requeridas.

#### Clases

- Area
- Arancel
- ArancelAcademico
- CarreraPrograma



- Componente
- ComponenteAcademico
- DatosPersonales
- Docente
- EstadoEstudiante
- EstadoMatricula
- Estudiante
- Expediente
- Group
- HorarioSemana
- Jornada
- Matricula
- Modulo
- Nivel
- OfertaAcademica
- OfertaComponente
- Papeleta
- Paralelo
- Parametro
- PeriodoLectivo
- PeriodoClase
- PeriodoMatricula
- PeriodoNoLaborado
- PlanEstudio
- RegistroAcreditacion
- RegistroAsistencia
- TipoPlanEstudioEx
- TipoPlanEstudio
- TipoPlanEstudioNone
- TipoPlanEstudioConTaller
- TipoPlanEstudioConTallerDetalle
- TipoPlanEstudioSumatoria
- TipoPlanEstudioPorPuntos



- TipoPlanEstudioPromedio
- TipoPlanEstudioAdmisiones
- Unidad
- User

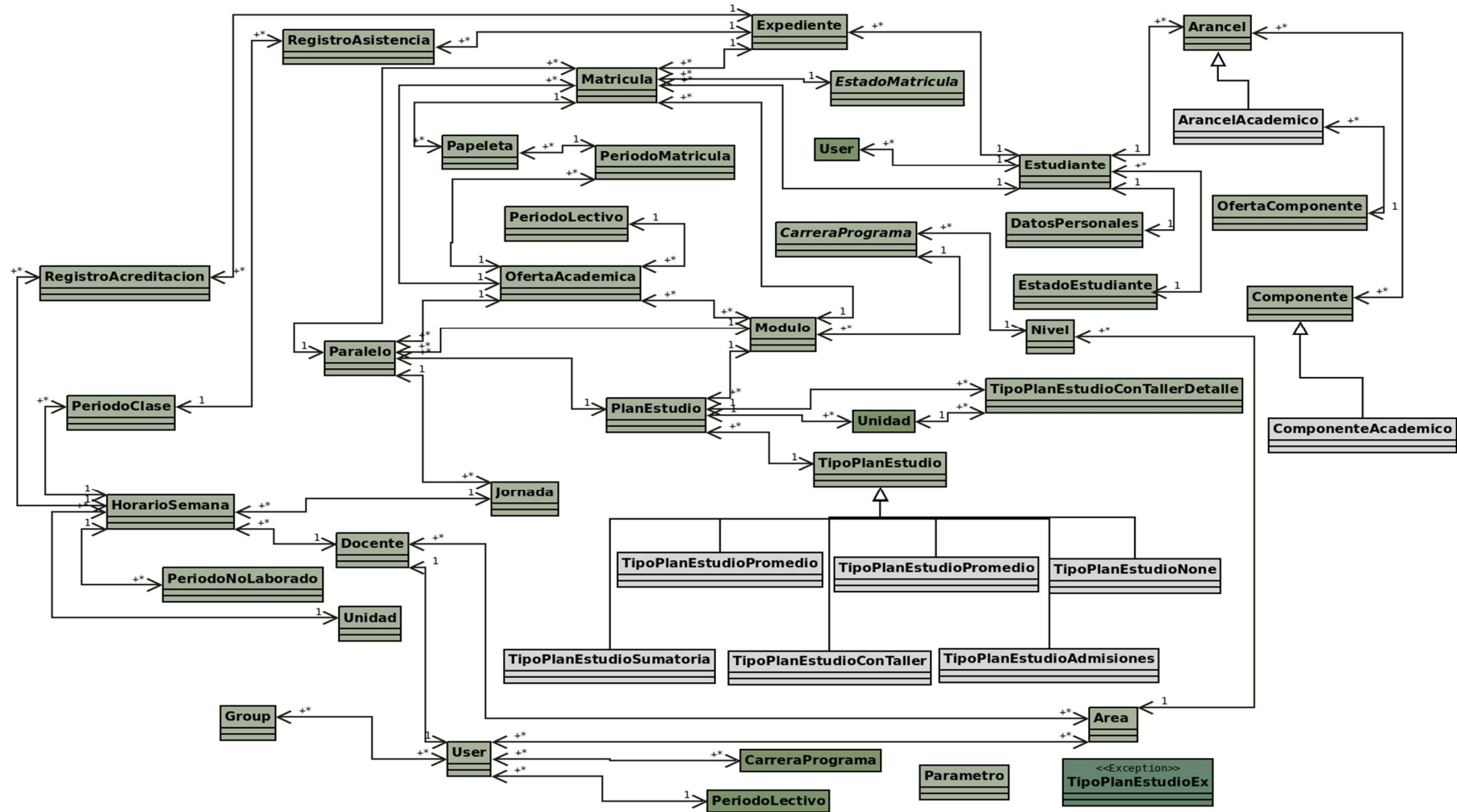
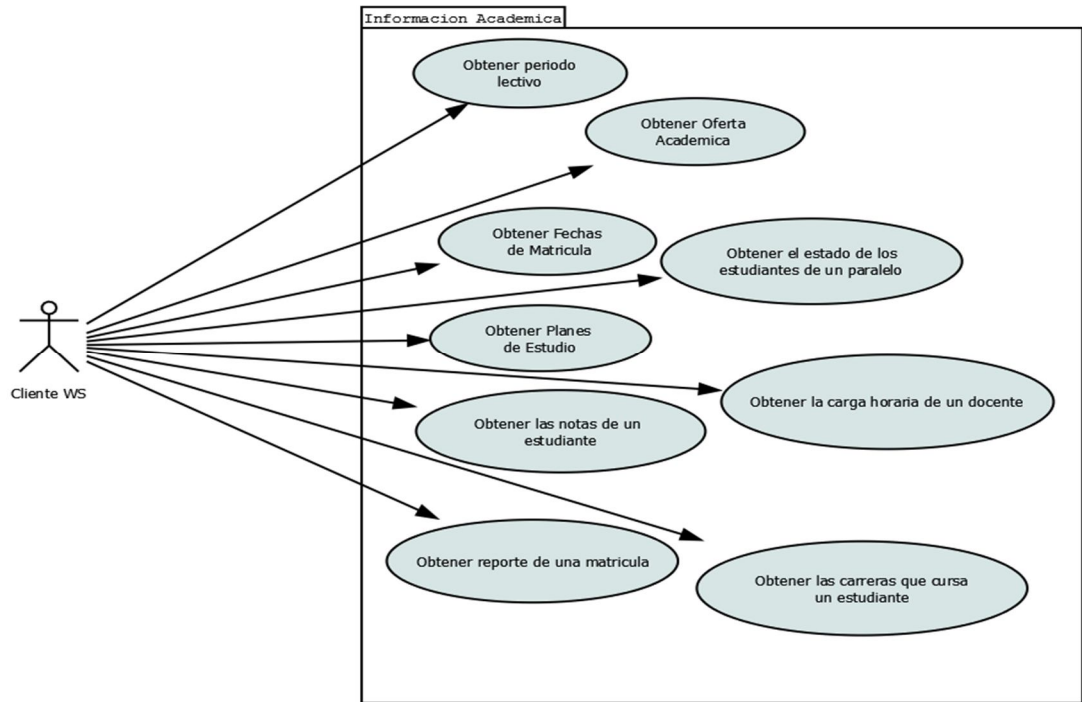


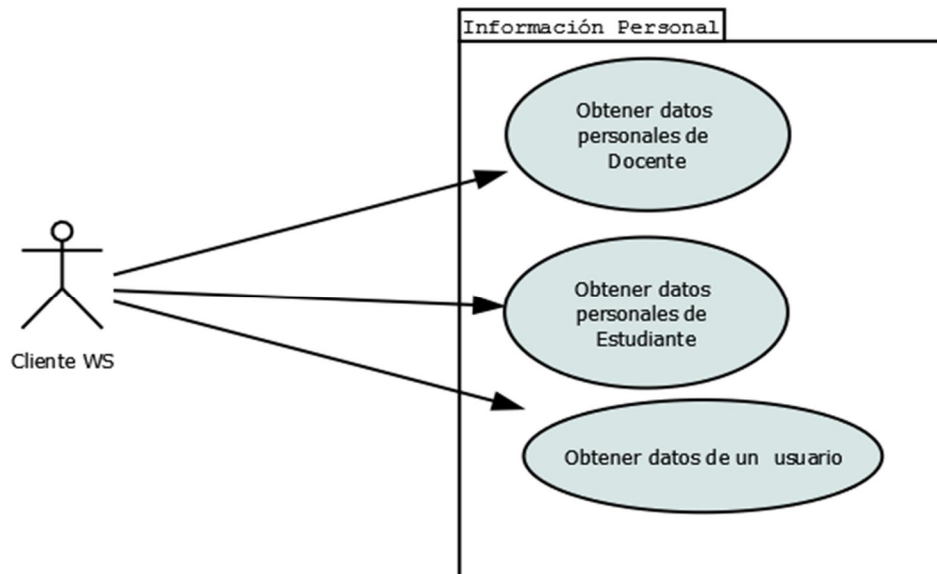
Fig.8: Modelo del Dominio

### 7.1.1.3. Diagrama de Casos de Uso por paquetes

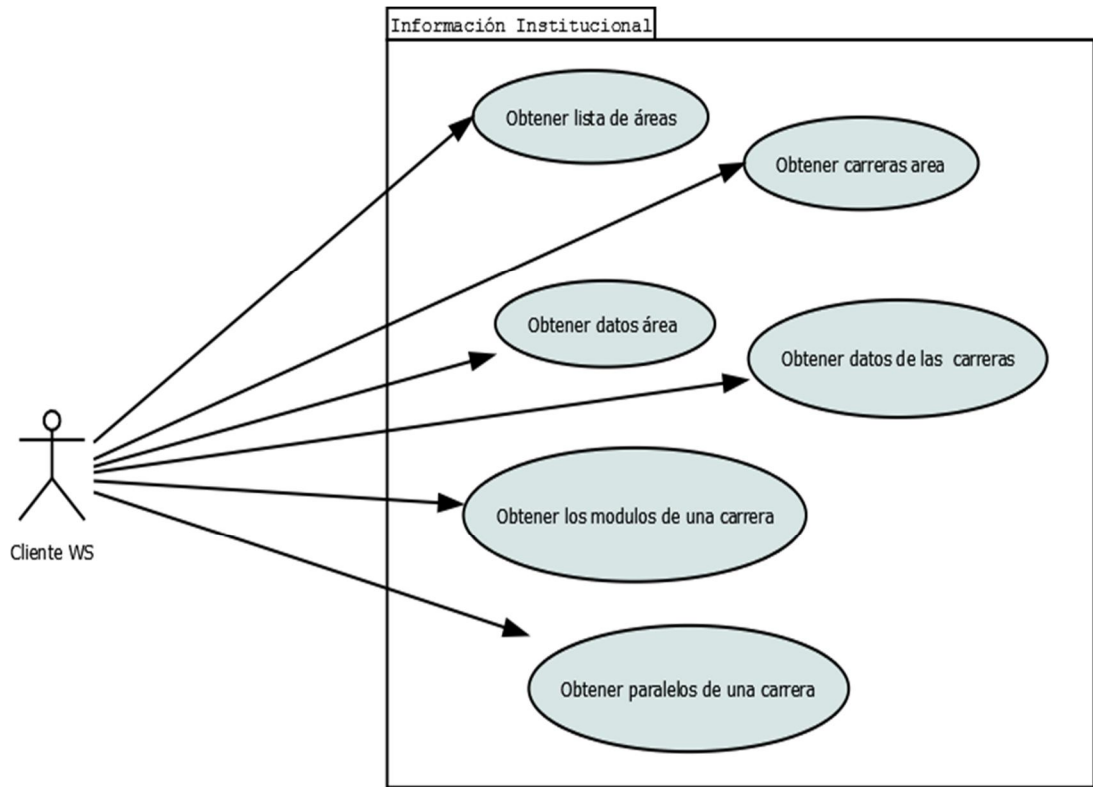
Frente a las ventajas que ofrece la metodología de desarrollo XP y su flexibilidad se realizó la definición de casos de uso para la identificación de procesos necesarios para el desarrollo de la aplicación. A continuación el diagrama de Casos de Uso por paquetes:



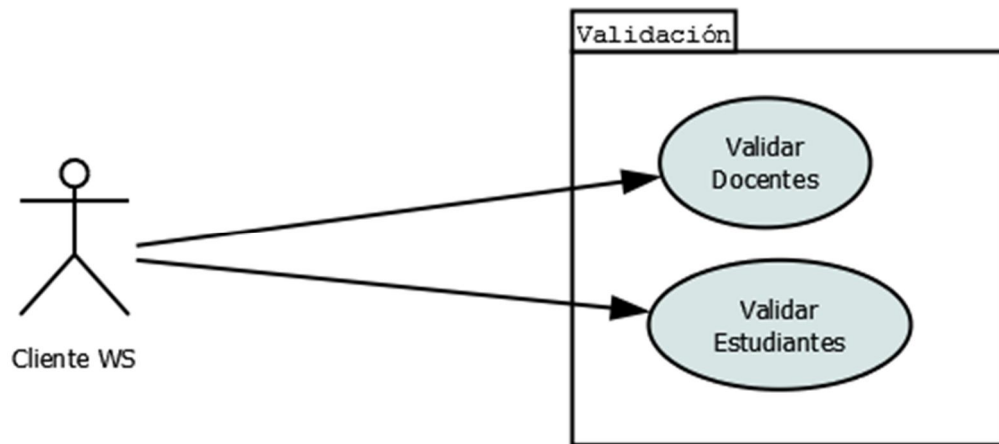
**Fig.9: Diagrama de Casos de Uso de Información Académica**



**Fig.10: Diagrama de Casos de Uso de Información Personal**



**Fig.11: Diagrama de Casos de Uso de Información Institucional**



**Fig.12: Diagrama de Casos de Uso de Validación**

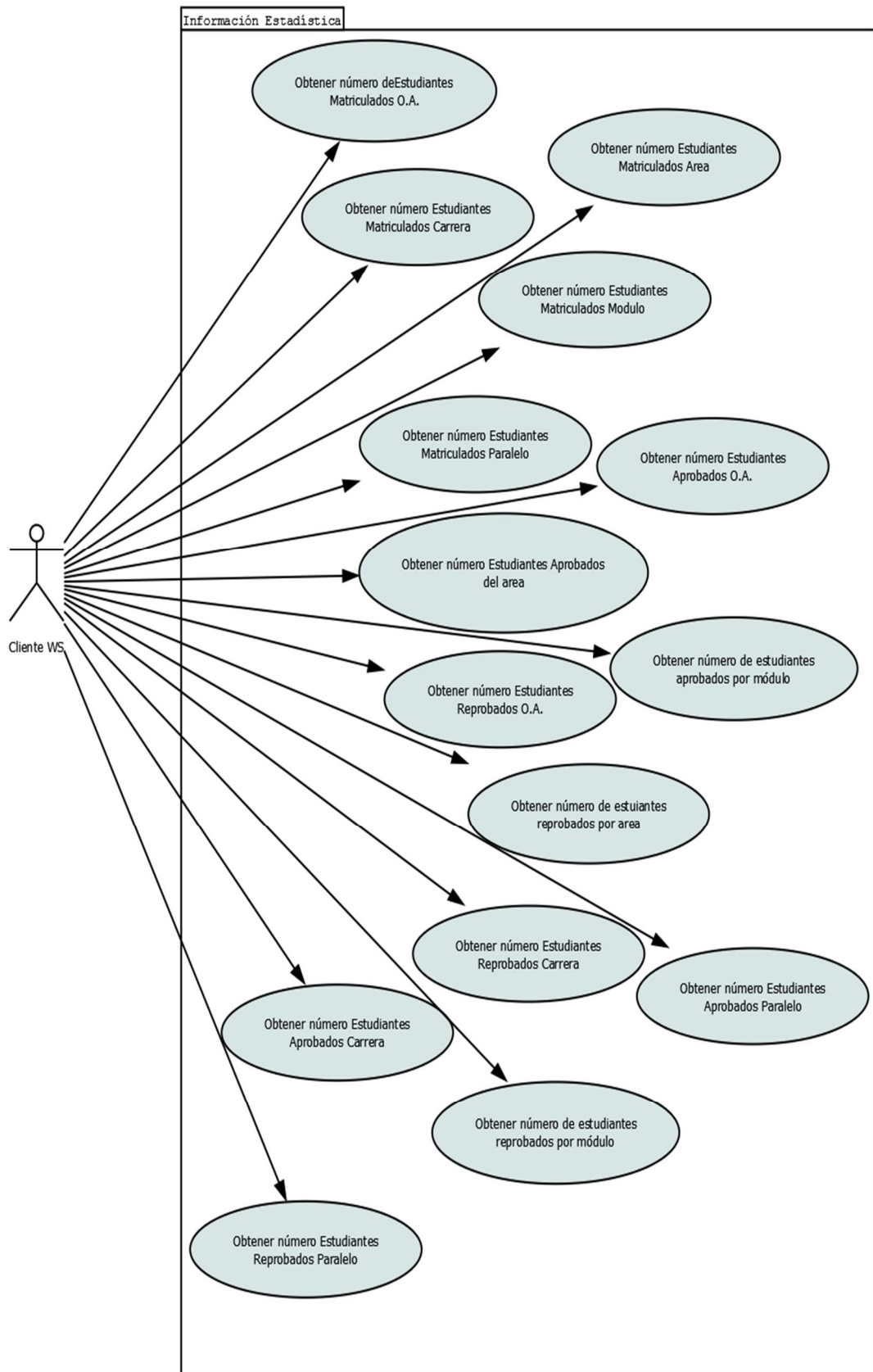


Fig.13: Diagrama de Casos de Uso de Información Estadística



#### 7.1.1.4. Desarrollo de Historias de Usuario y Diagramas de Secuencia

HISTORIA DE USUARIO		
<b>Nombre de Historia:</b> Obtener Periodos Lectivos		
<b>Nº:</b> 1	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El Cliente WS solicita los periodos lectivos existente en el sistema y el WS devuelve el listado de los mismos con su id y descripción.		
<b>Observaciones:</b> Este método no recibe parámetros de entrada: <code>sgaws_periodos_lectivos()</code>		

**Tabla. 1: Historia de Usuario #1**

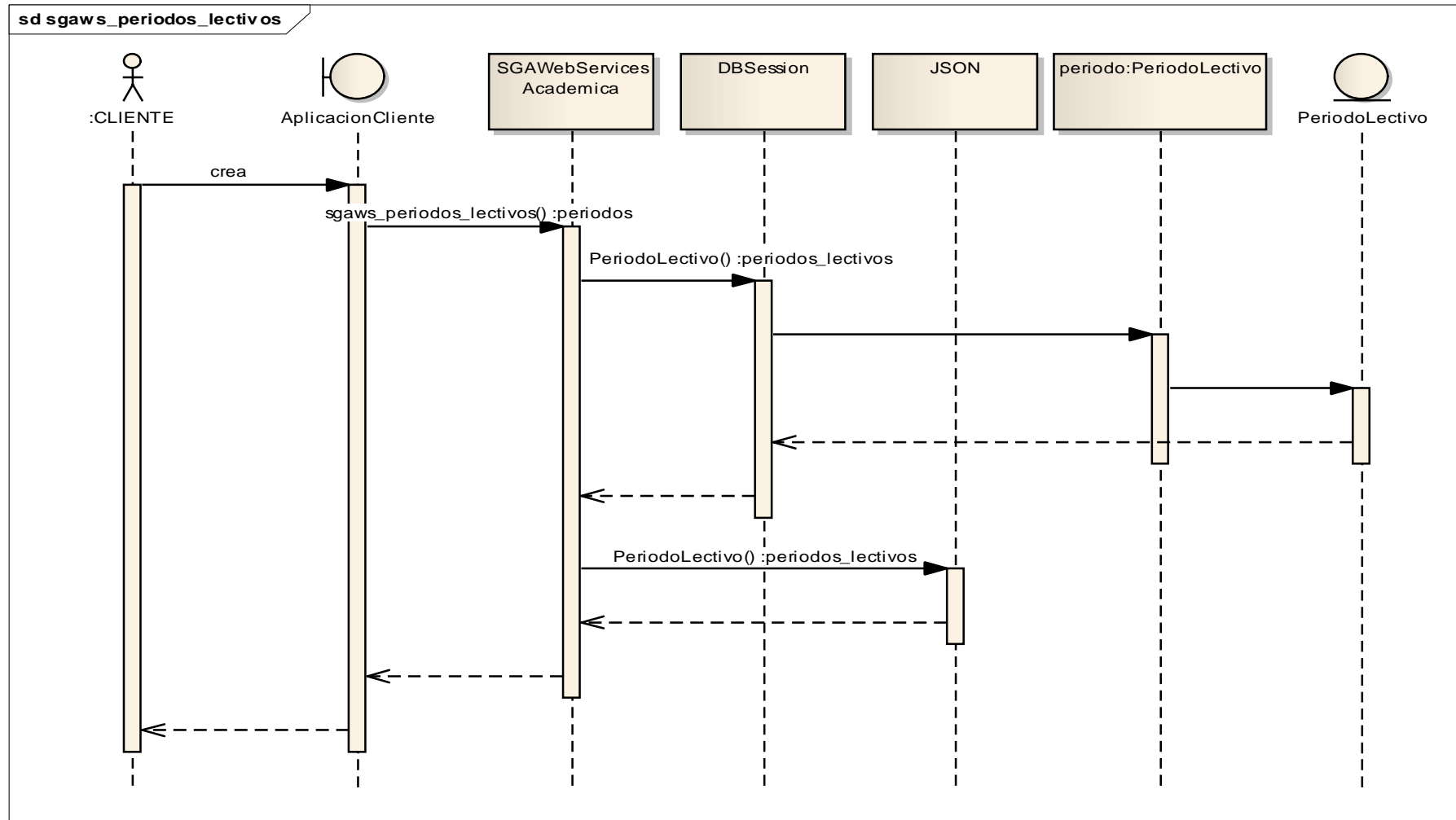


Fig.14: Diagrama de Secuencia del método sgaws\_periodos\_lectivos()



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener Ofertas Académicas de un Periodo Lectivo		
<b>Nº:</b> 2	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El Cliente WS solicita las ofertas académicas de un periodo lectivo, para lo cual envía el id del periodo lectivo, el WS devuelve una lista de ofertas académicas con su id, descripción, fecha de inicio de clase y fecha de fin de clases que dispone ese periodo lectivo.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id del periodo lectivo: <b>sgaws_ofertas_academicas( id_periodo)</b>		

Tabla. 2: Historia de Usuario #2

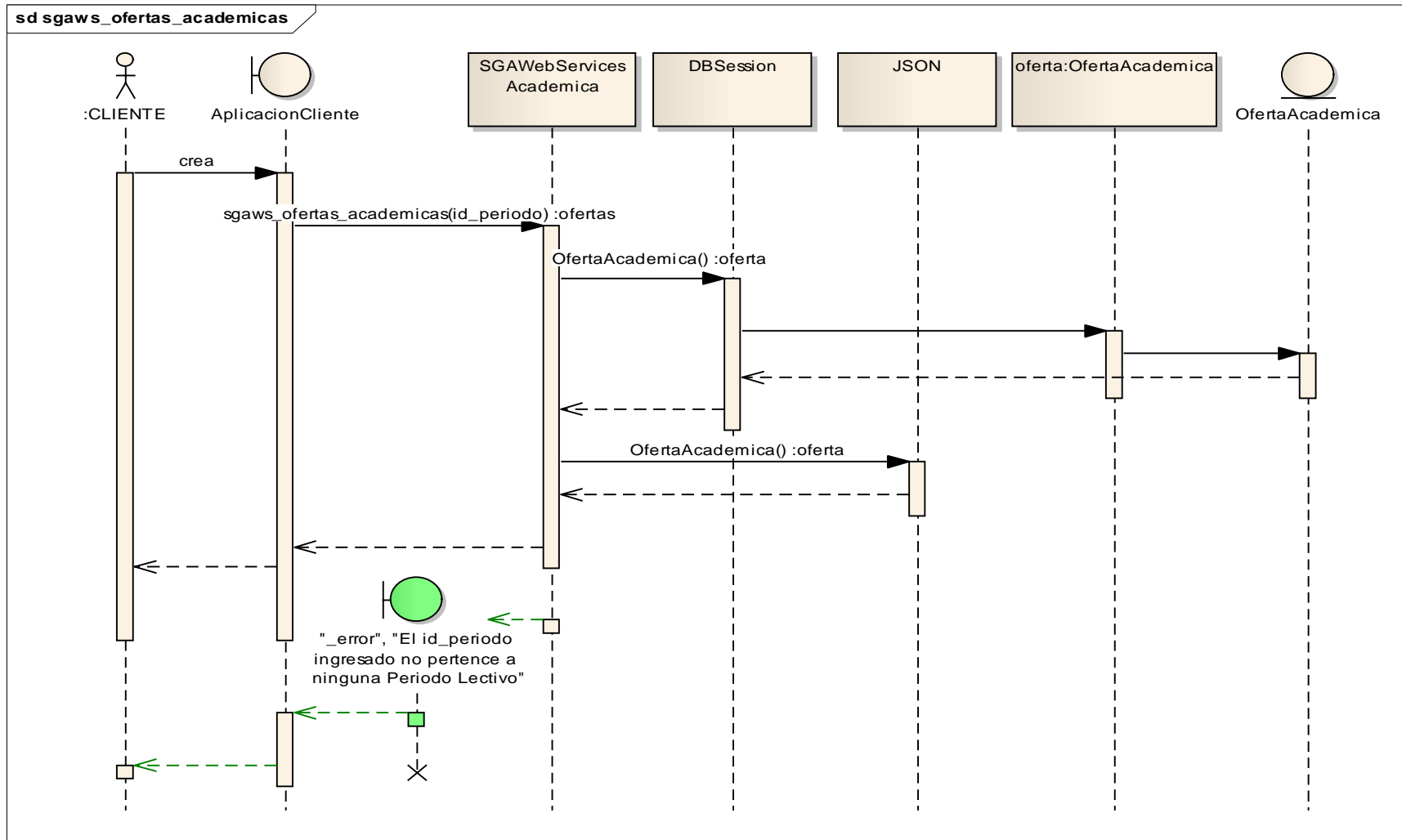


Fig.15: Diagrama de Secuencia del método sgaws\_ofertas\_academicas ( id\_periodo)

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener fechas de matrículas de una Oferta Académica		
<b>Nº:</b> 3	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El Cliente WS solicita las fechas de matrículas de una oferta académica, para lo cual envía el id de la oferta, el ws retorna tanto la fecha de inicio como también la fecha de fin de las matrículas de una oferta académica y su respectivo tipo de periodo: ordinaria, extraordinaria y especial.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica: <code>sgaws_fechas_matriculaoa(id_oferta)</code>		

**Tabla. 3: Historia de Usuario #3**

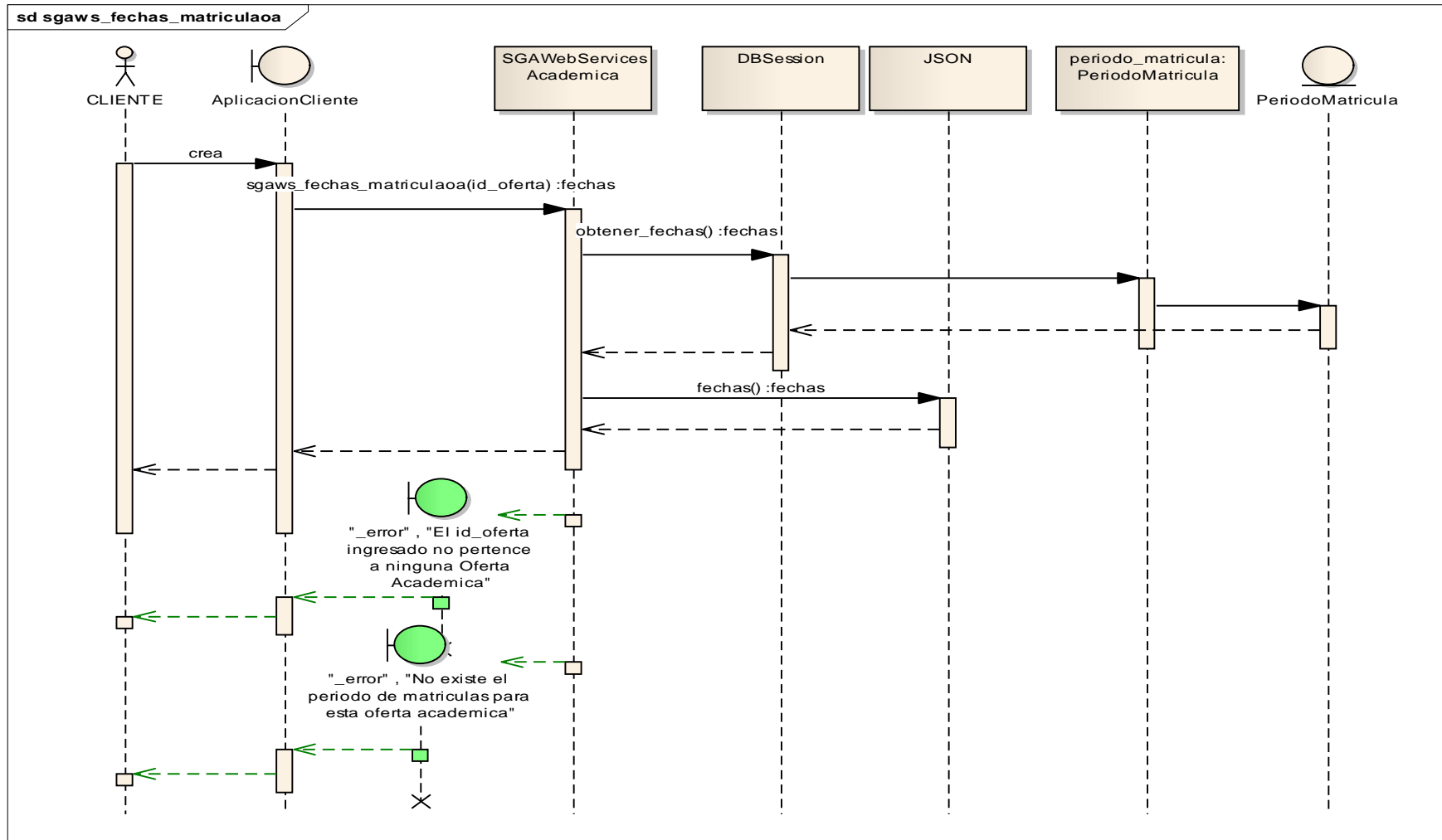


Fig.16: Diagrama de Secuencia del método sgaws\_fechas\_matriculaoa(id\_oferta)

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener estado de estudiantes Aprobados y Reprobados o matriculados de un paralelo.		
<b>Nº:</b> 4	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 3
<b>Descripción:</b> El cliente ws para obtener el estado de estudiantes de un paralelo, envía el id del paralelo del que desea obtener dicha información. El ws retorna un listado con la descripción de la oferta académica, nombre de la carrera, nombre y número de módulo, nombre de paralelo, id de matrícula, nombres, apellidos, cédula y el estado de la matrícula de los estudiantes de un paralelo.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id del paralelo: <b>sgaws_estadoestudiantes_paralelo(id_paralelo)</b>		

**Tabla. 4: Historia de Usuario #4**

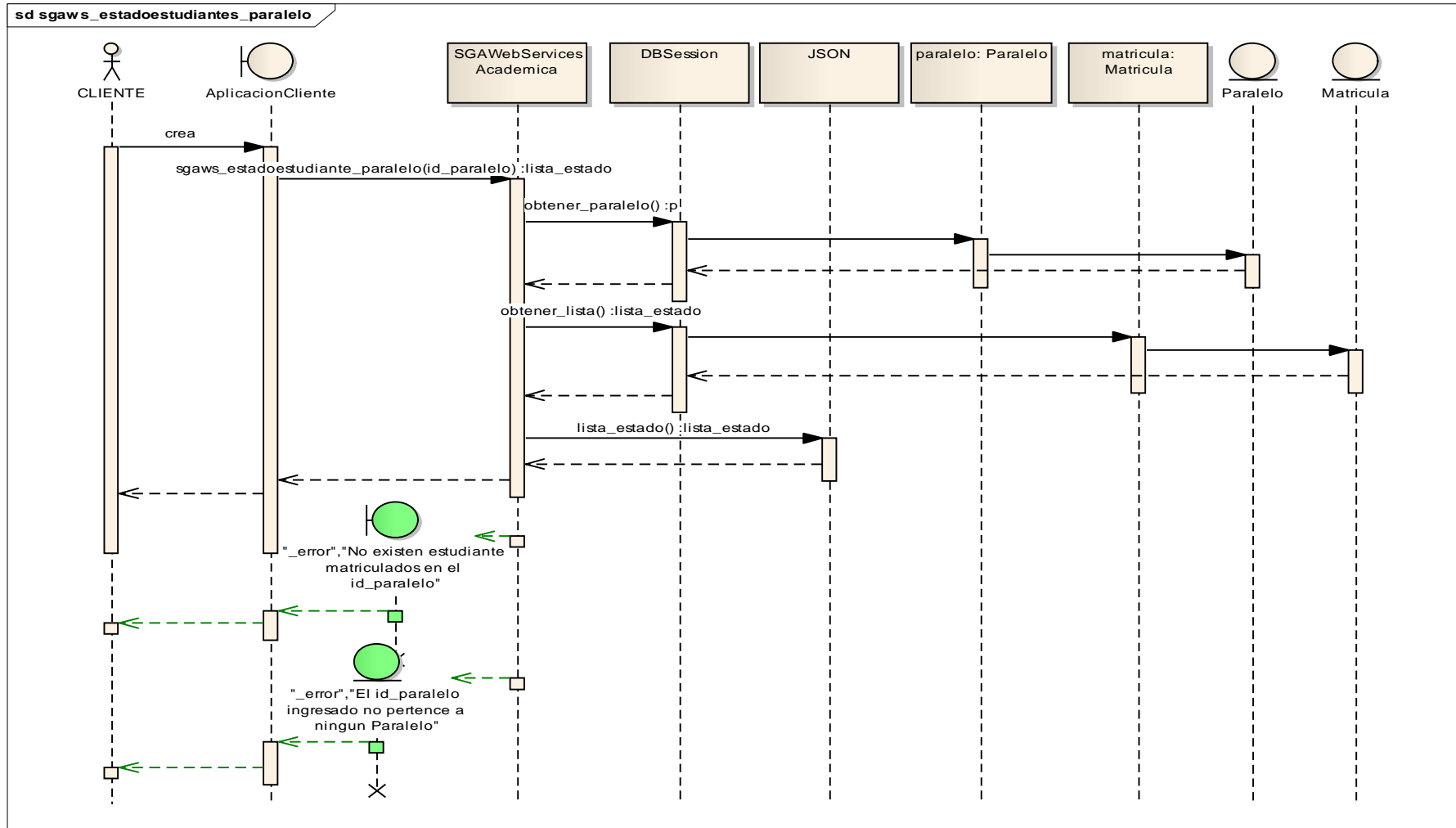


Fig.17: Diagrama de Secuencia del método sgaws\_estadoestudiantes\_paralelo(id\_paralelo)



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener plan de estudio asignado a un paralelo		
<b>Nº:</b> 5	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Alta	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 3
<b>Descripción:</b> El cliente WS solicita el plan de estudio asignado a un paralelo mediante el envío del id del paralelo, el Web Services retorna el plan de estudio solicitado con el nombre de la carrera, nombre y número de módulo, nombre y sección del paralelo, nombre del plan de estudio, nombre de unidades, duración, créditos y obligatoriedad de cada unidad correspondientes a la información ingresada.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id del paralelo: <code>sgaws_plan_estudio(id_paralelo)</code>		

**Tabla. 5: Historia de Usuario #5**

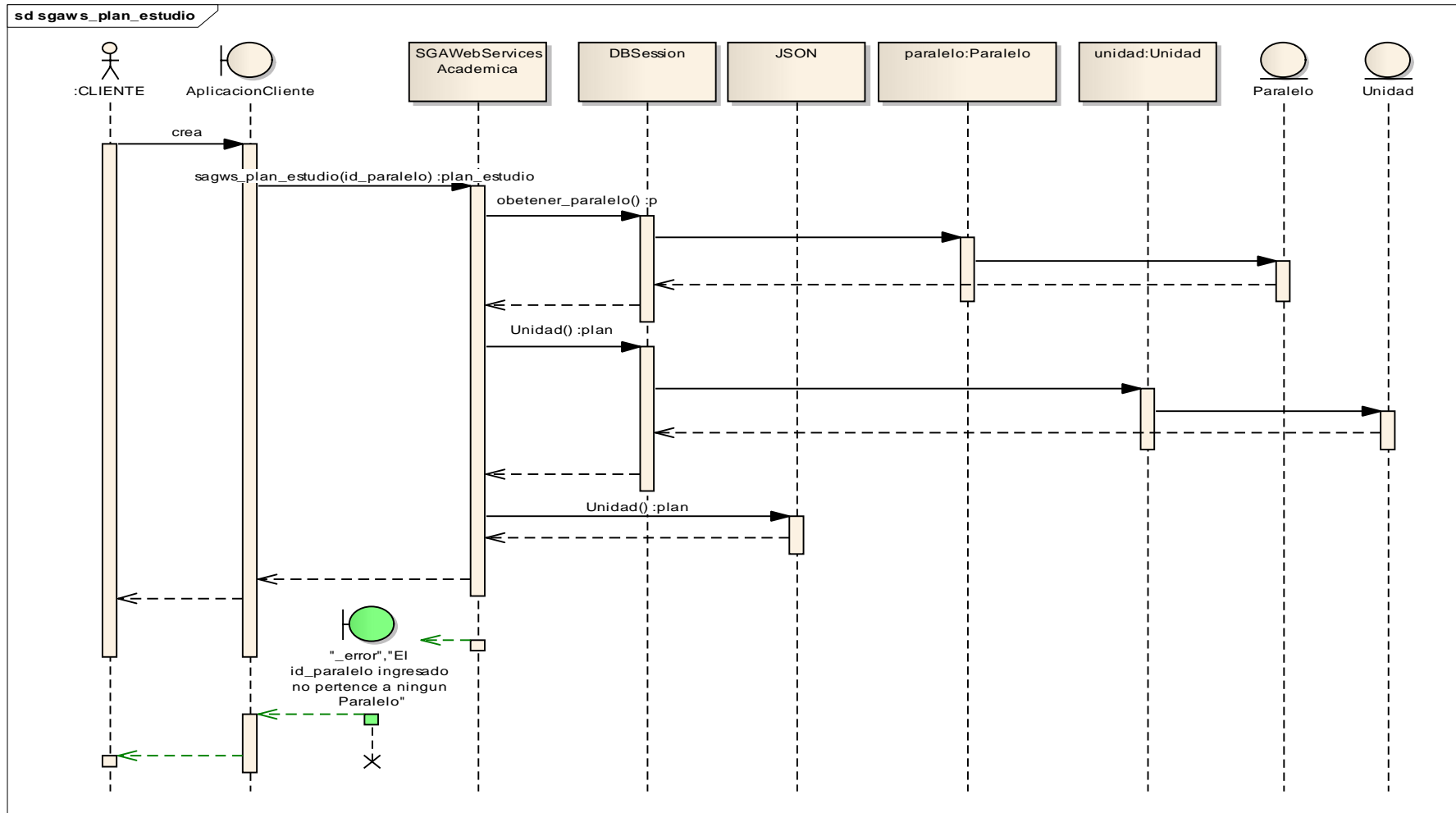


Fig.18: Diagrama de Secuencia del método `sgaws_plan_estudio(id_paralelo)`



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener Carga Horaria de Docentes en una oferta académica		
<b>Nº:</b> 6	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 4
<b>Descripción:</b> El cliente ws para obtener la carga horaria de un docente, debe enviar el id de la oferta académica y cédula del docente del que desea obtener dicha información, el ws responderá la petición hecha devolviendo: el título, nombres y apellidos del docente así como también el nombre de la carrera, las unidades, la duración, el número de modulo y el nombre del paralelo.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id del paralelo: <code>sgaws_carga_horaria_docente(id_oferta, cedula_docente)</code>		

Tabla. 6: Historia de Usuario #6

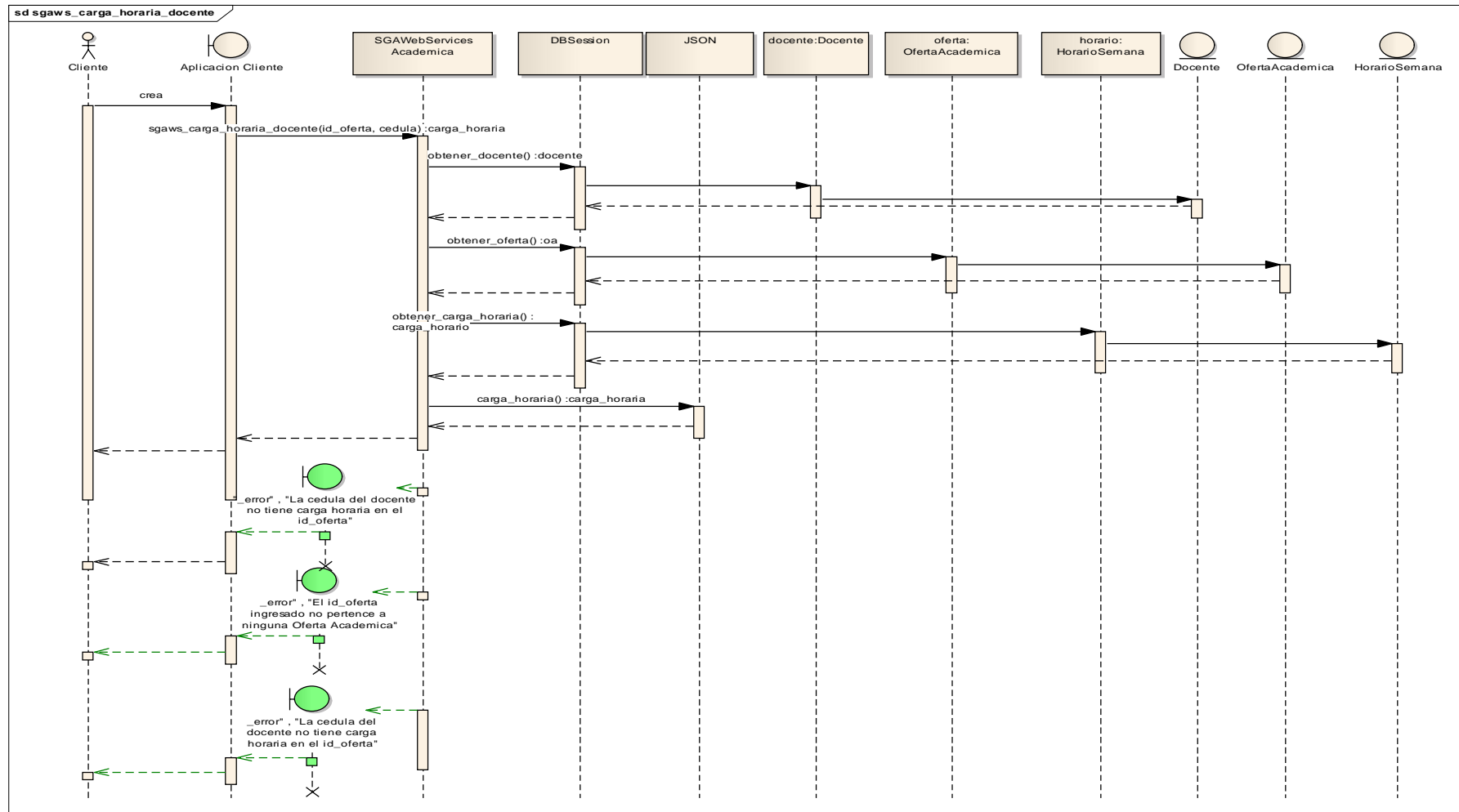


Fig.19: Diagrama de Secuencia del métodosgaws\_carga\_horaria\_docente(id\_oferta, cedula\_docente)



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener notas de cada unidad de un estudiante en una oferta académica		
<b>Nº:</b> 7	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Prioridad:</b> Media	<b>Prioridad:</b> Media
<b>Descripción:</b> El cliente WS envía la cédula, el id de la carrera y el id de la oferta académica del estudiante del que desea conocer las notas y el Web Services retorna los nombres, apellidos, número de módulo, paralelo, nombre de la carrera, modalidad, estado de la matrícula, nombre de la unidad y nota de la unidad.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada la cedula del estudiante, el id de la carrera e id de la oferta académica: <b>sgaws_notas_estudiante(cedula, id_carrera, id_oferta)</b>		

**Tabla. 7: Historia de Usuario #7**

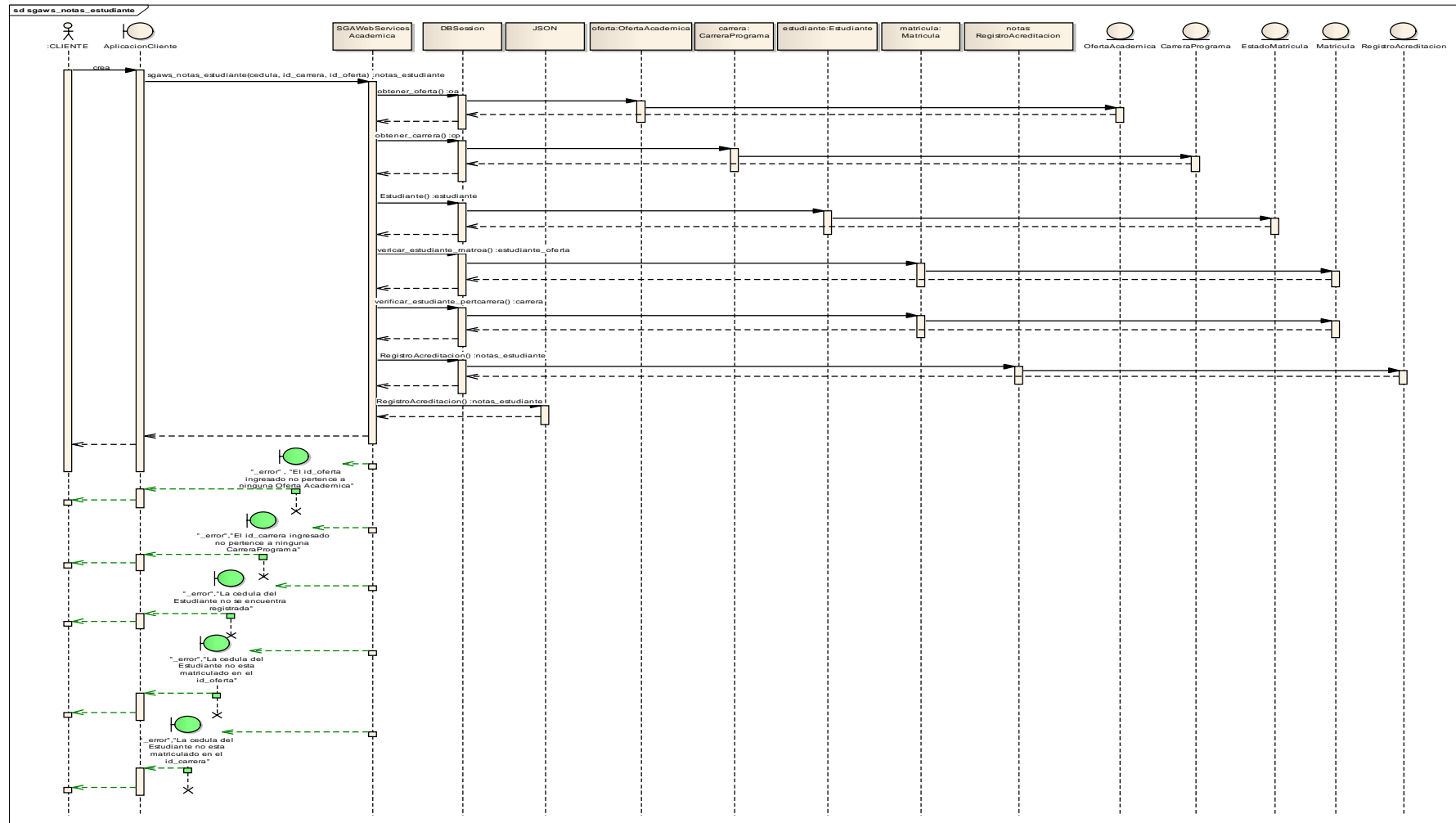


Fig.20: Diagrama de Secuencia del método sgaws\_notas\_estudiante(cedula, id\_carrera, id\_oferta)



HISTORIA DE USUARIO		
<b>Nombre de Historia:</b> Obtener carreras de un estudiante		
<b>Nº:</b> 8	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Prioridad:</b> Media	<b>Prioridad:</b> Media
<b>Descripción:</b> El cliente WS para obtener las carreras en las que está un estudiante envía la cedula del estudiante del que desea conocer dicha información y el Web Services retorna los nombres, apellidos, cedula del estudiante, nombre de la carrera y modalidad de la carrera en la que se encuentra.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada la cedula del estudiante: <b>sgaws_carreras_estudiante(cedula)</b>		

**Tabla. 8: Historia de Usuario #8**

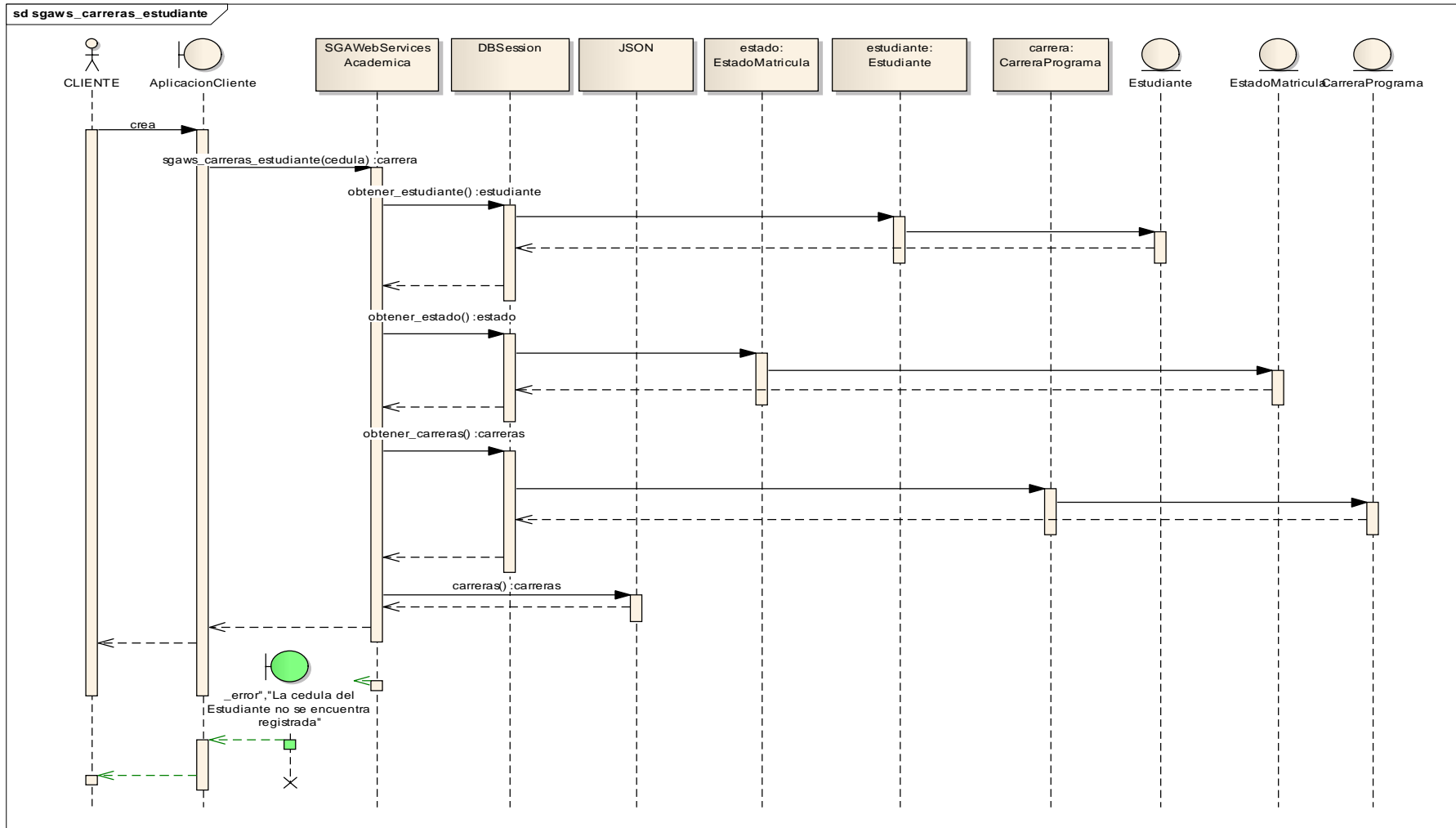


Fig.21: Diagrama de Secuencia del método sgaws\_carreras\_estudiante(cedula)





<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener el reporte de la matrícula de un estudiante en una oferta académica.		
<b>Nº:</b> 9	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Prioridad:</b> Media	<b>Prioridad:</b> Media
<b>Descripción:</b> El cliente WS para obtener el reporte de la matrícula de un estudiante en una oferta académica y la cedula del estudiante que se desea obtener dicha información, el Web Services retorna, la descripción de la oferta académica, los nombres del estudiante, apellidos, la nota registrada en esa matricula, el porcentaje de asistencia y el estado de la matrícula.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y la cedula del estudiante: <b>sgaws_reporte_matricula(id_oferta,cedula)</b>		

**Tabla. 9: Historia de Usuario #9**

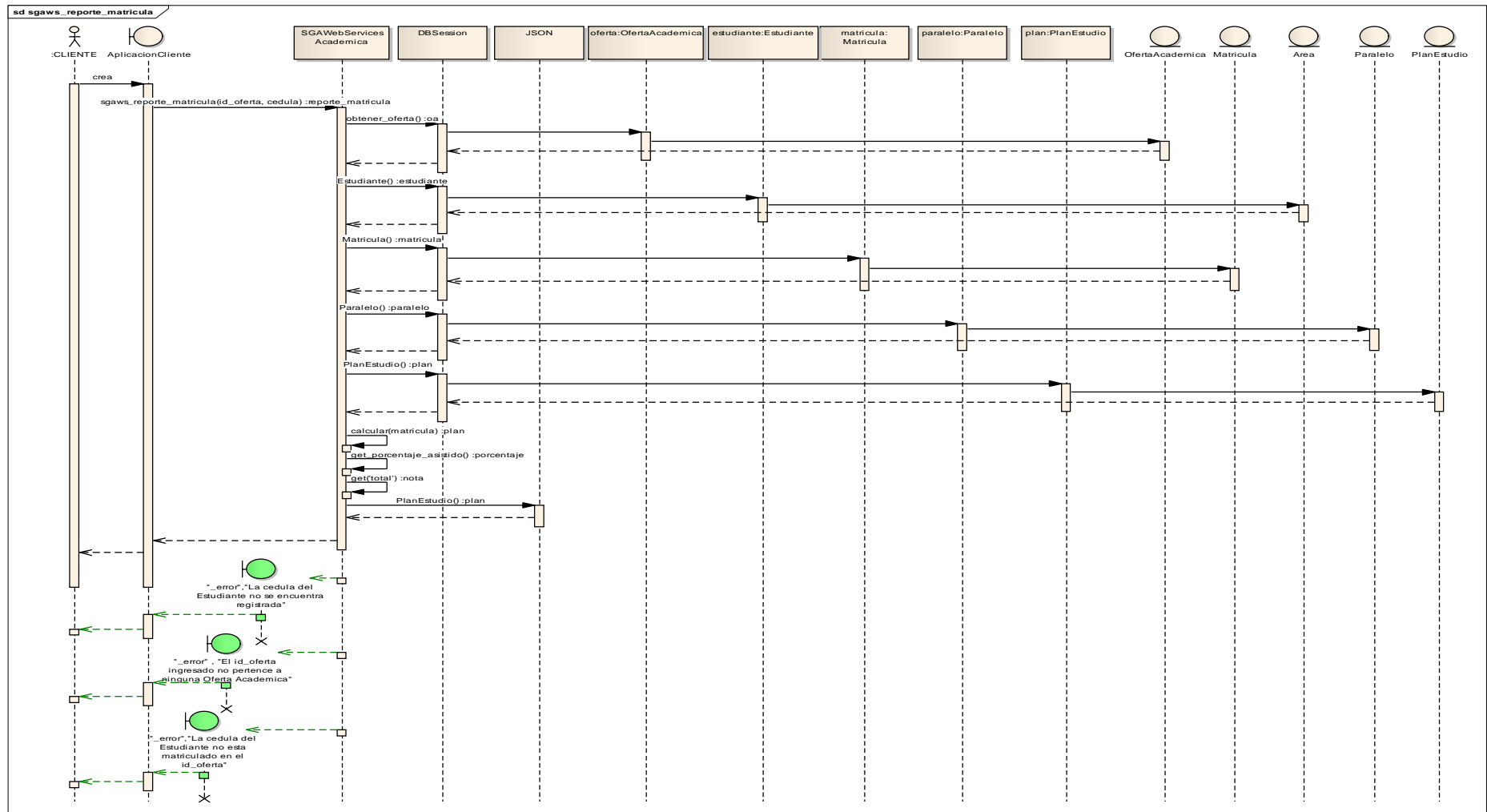


Fig.22: Diagrama de Secuencia del método `sgaws_reporte_matricula(id_oferta,cedula)`



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener lista de áreas		
<b>Nº:</b> 10	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Baja	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El cliente WS realiza una petición al SGAWebServices de las áreas de la universidad con sus datos respectivos, el WS retorna una todas las áreas contenidas en el Sistema de Gestión Académica con datos como: siglas, secretario abogado, director, y nombre.		
<b>Observaciones:</b> Este método no recibe parámetros de entrada: <code>sgaws_lista_areas()</code>		

**Tabla.10: Historia de Usuario #10**

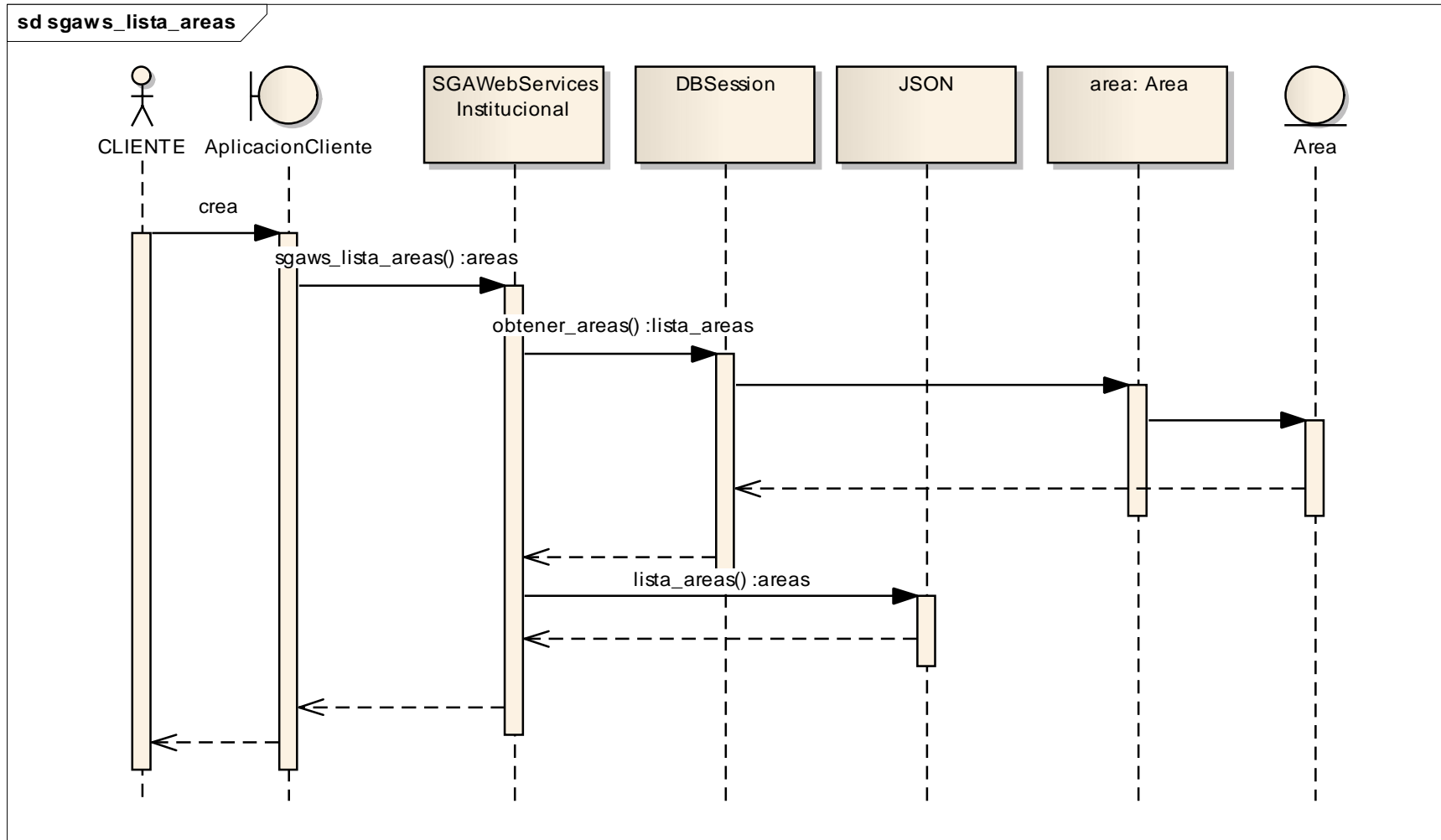


Fig.23: Diagrama de Secuencia del método `sgaws_lista_areas()`



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener datos área		
<b>Nº:</b> 11	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Baja	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El cliente WS solicita al SGAWebServices los datos del área para lo cual ingresa su siglas y el WS retorna los datos del área: siglas, nombre, secretario abogado y el director		
<b>Observaciones:</b> Este método tiene como parámetro de entrada las siglas del Área: <code>sgaws_datos_area(sigla)</code>		

**Tabla. 11: Historia de Usuario #11**

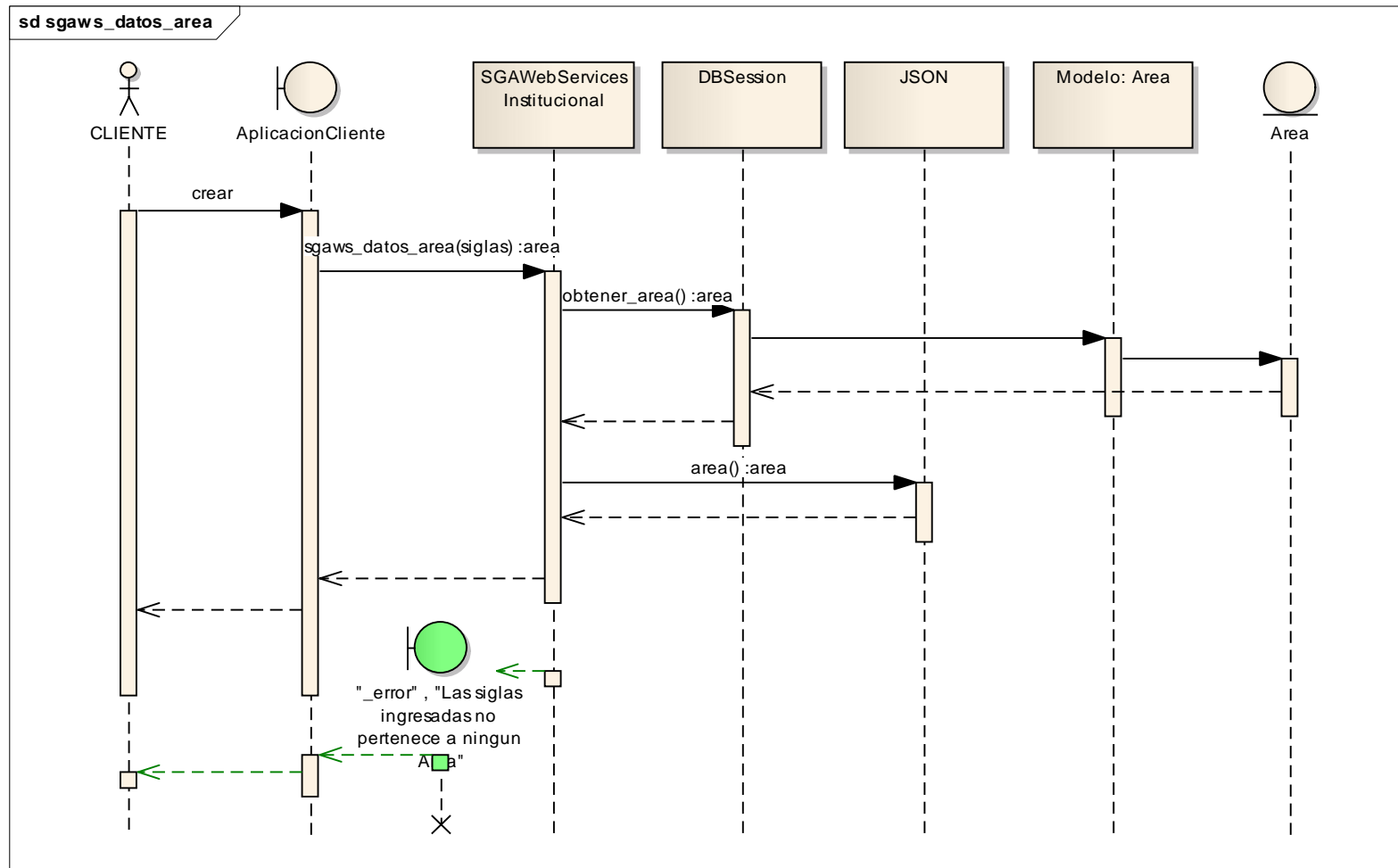


Fig.24: Diagrama de Secuencia del método sgaws\_datos\_area(sigla)

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener las carreras existentes en una área		
<b>Nº:</b> 12	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Baja	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El cliente WS solicita al SGAWebServices todas las carreras de un área para lo cual ingresa sus siglas y el WS retorna el: título, el nombre, la modalidad, el nivel y el cupo de cada una de las carreras del área a consultar.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada las siglas del Área: <b>sgaws_carreras_area(siglas)</b>		

**Tabla. 12: Historia de Usuario #12**

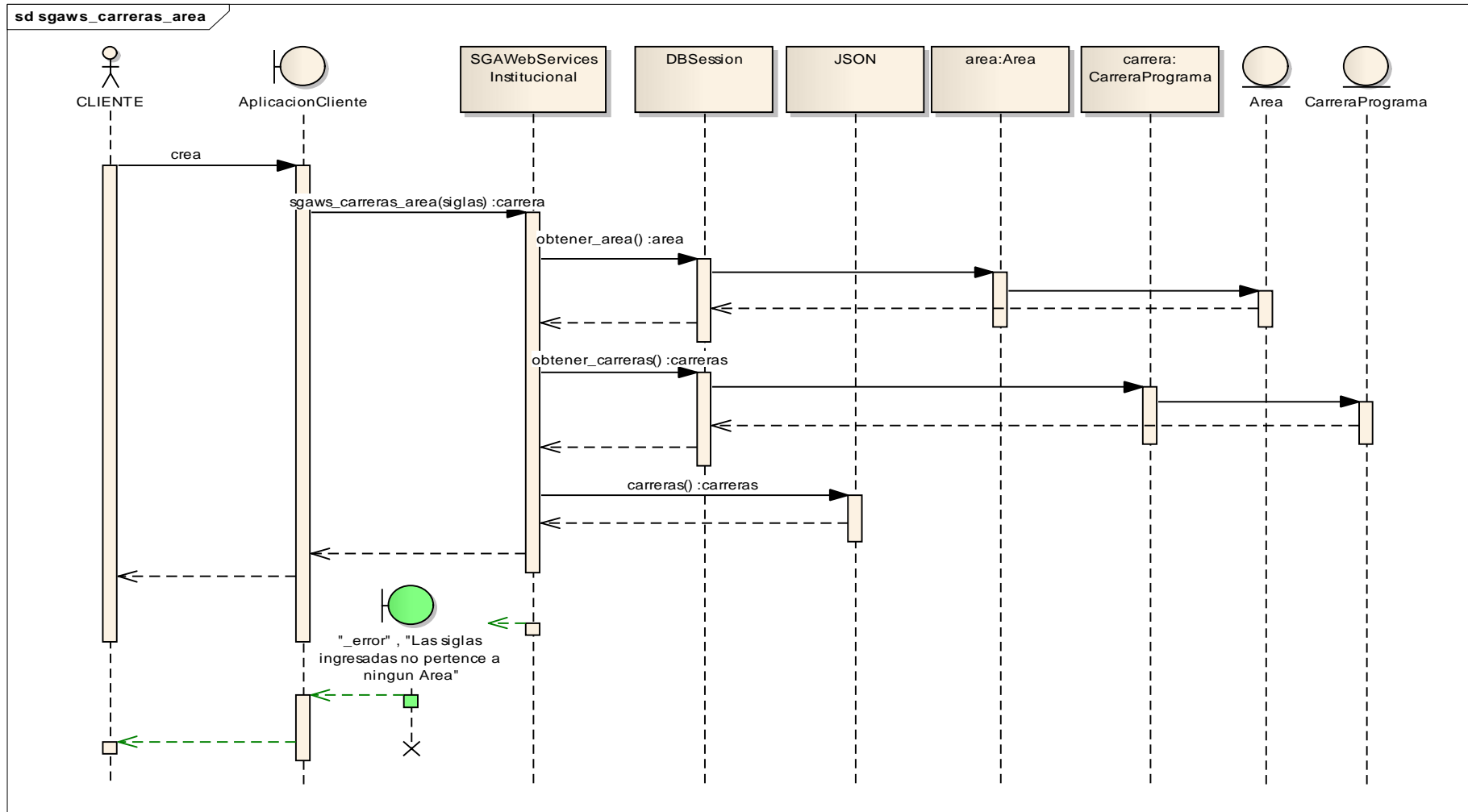


Fig.25: Diagrama de Secuencia del método sgaws\_carreras\_area(siglas)





HISTORIA DE USUARIO		
<b>Nombre de Historia:</b> Obtener datos de las carreras vigentes en una Oferta Académica		
<b>Nº:</b> 13	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El clientes solicita al SGAWebServices para conocer las carreras disponibles en una oferta académica para lo cual envía el id de la oferta académica de la cual desea obtener dicha información, el WS devuelve las carreras ofertadas con su id, nombre y modalidad		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica: <b>sgaws_datos_carreras(id_oferta)</b>		

Tabla. 13: Historia de Usuario #13

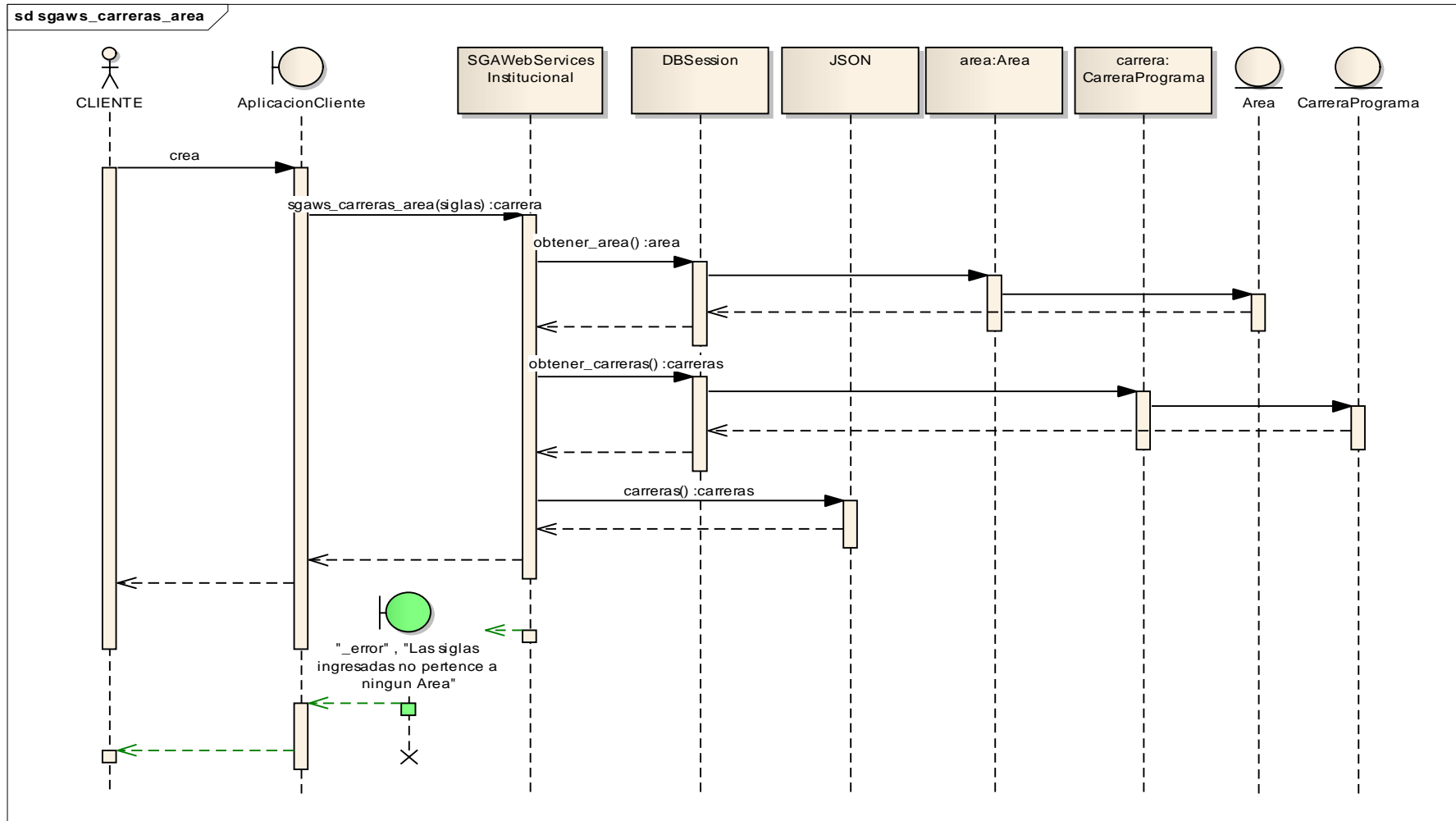


Fig.26: Diagrama de Secuencia del método sgaws\_datos\_carreras(id\_oferta)

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener módulos de una carrera en una Oferta Académica		
<b>Nº:</b> 14	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El cliente WS para conocer los módulos disponibles envía el id de la oferta académica y el de la carrera de la cual desea obtener dicha información, el WS devuelve una lista con la oferta, el nombre de la carrera, y los id, nombres y número de todos los módulos correspondientes a la información ingresada.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y el id de la carrera: <code>sgaws_modulos_carrera(id_oferta, id_carrera)</code>		

**Tabla. 14: Historia de Usuario #14**

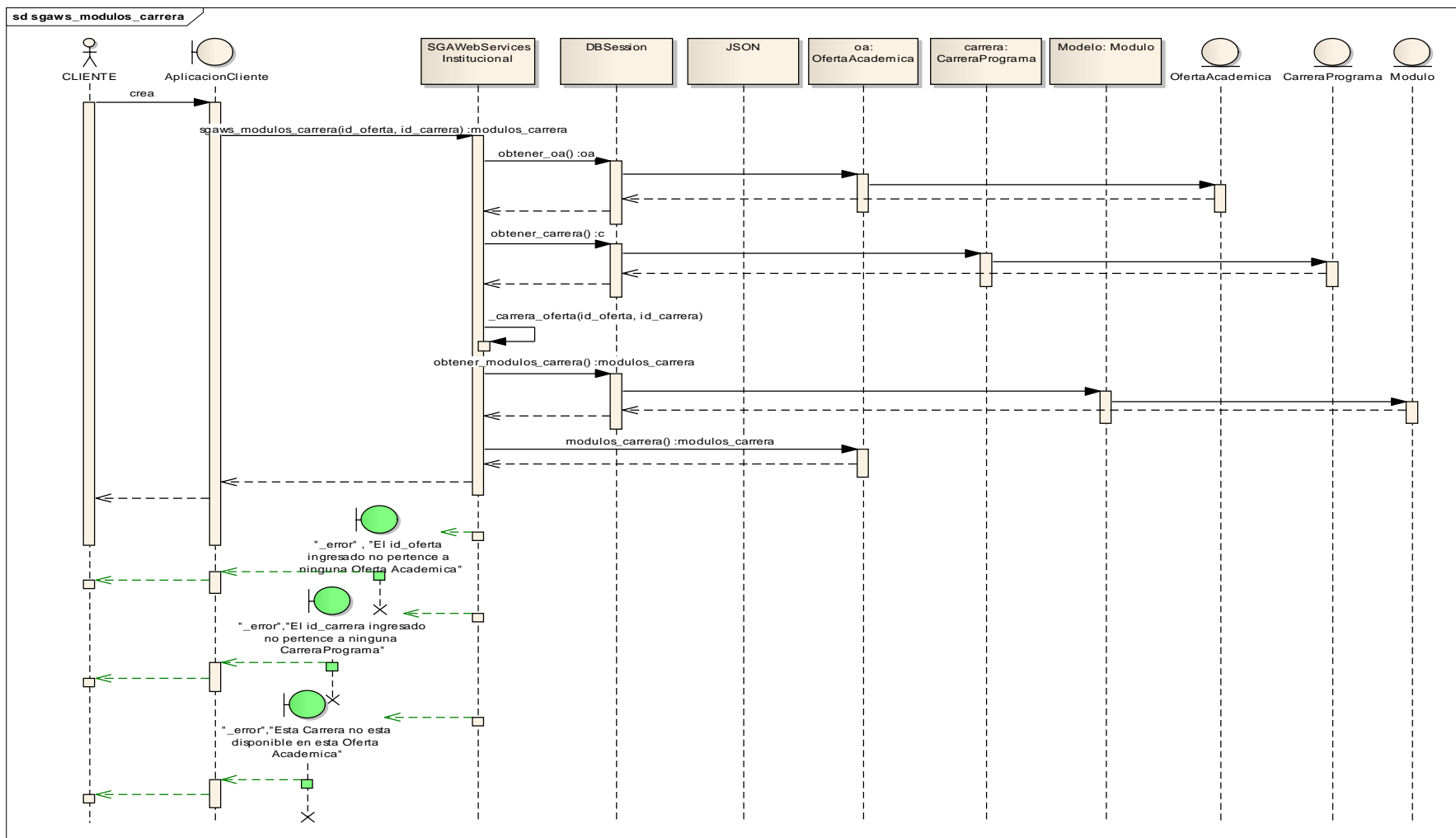


Fig.27: Diagrama de Secuencia del método sgaws\_modulos\_carrera(id\_oferta, id\_carrera)

HISTORIA DE USUARIO		
<b>Nombre de Historia:</b> Obtener paralelos de una carrera en un oferta académica		
<b>Nº:</b> 15	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El cliente WS para conocer los paralelos disponibles de una carrera en una oferta académica, envía el id de la oferta académica y la carrera, de la cual desea obtener dicha información, el WS devuelve una lista con la descripción de la oferta académica, el nombre de la carrera y los datos de los paralelos tales como: id, sección, nombre, así como también el número id del módulo al que corresponde.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y el id de la carrera: <code>sgaws_paralelos_carrera(id_oferta, id_carrera)</code>		

Tabla. 15: Historia de Usuario #15

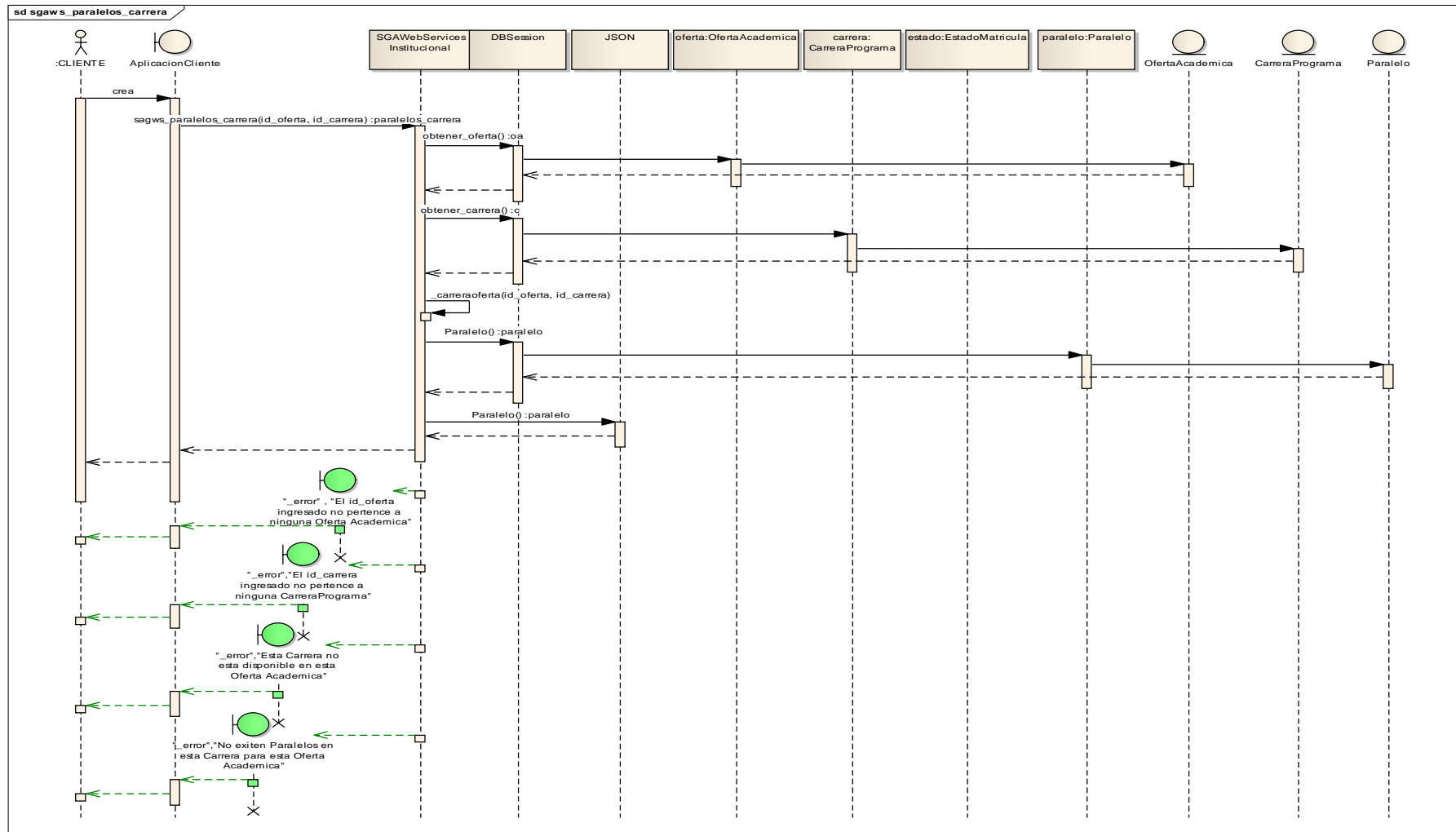


Fig.28: Diagrama de Secuencia del método sgaws\_paralelos\_carrera(id\_oferta, id\_carrera)



HISTORIA DE USUARIO		
<b>Nombre de Historia:</b> Validar docentes		
<b>Nº:</b> 16	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Alta	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El cliente WS envía la cedula y la clave del docente y el Web Services valida si el usuario se encuentra o no en el sistema.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada la cedula del docente y la clave del mismo: <code>sgaws_validar_docente(cedula, clave)</code>		

Tabla. 16: Historia de Usuario #16

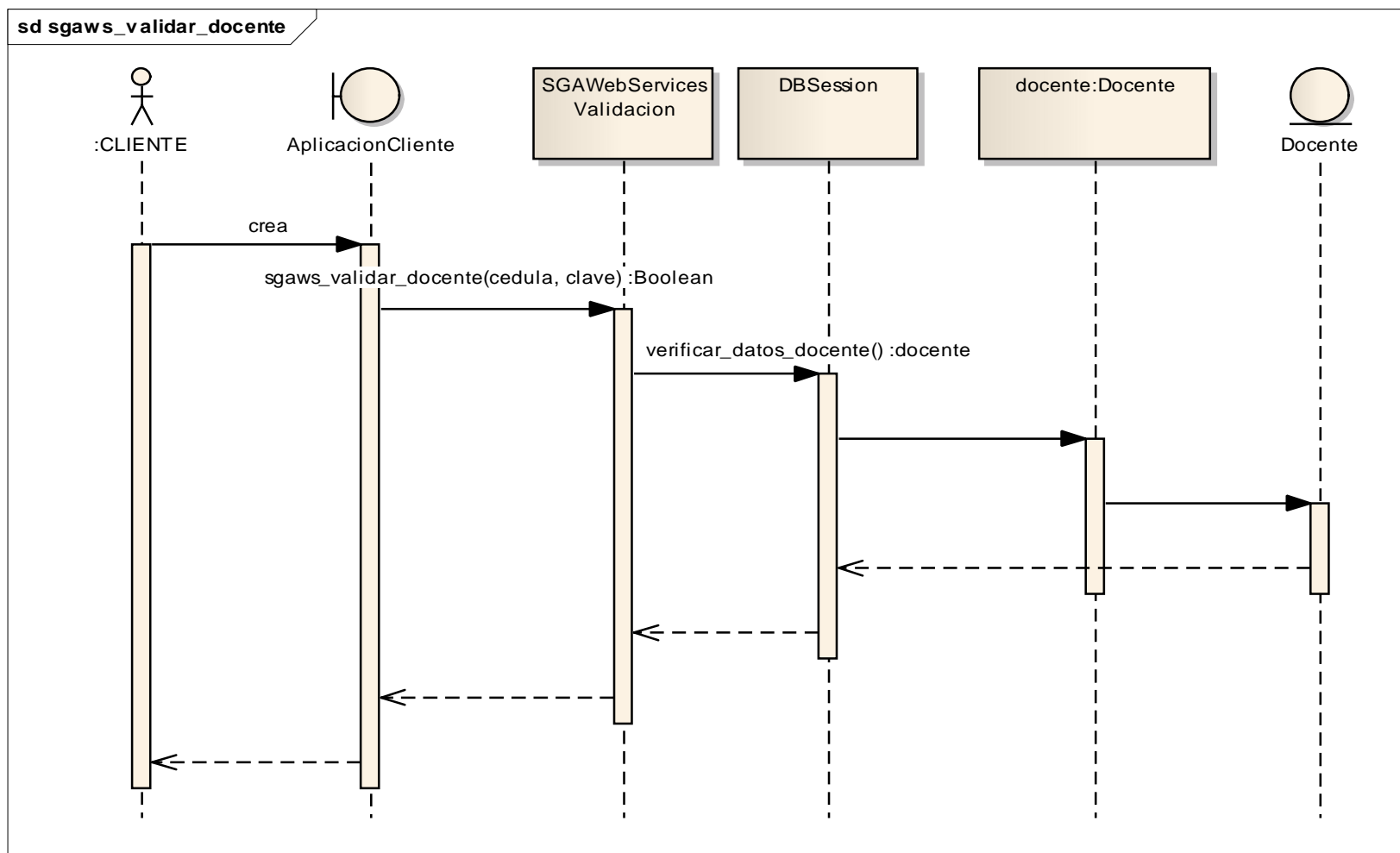


Fig.29: Diagrama de Secuencia del método sgaws\_validar\_docente(cedula, clave)





HISTORIA DE USUARIO		
<b>Nombre de Historia:</b> Validar estudiante		
<b>Nº:</b> 17	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Alta	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 2
<b>Descripción:</b> El cliente WS envía la cedula y la clave del estudiante y el Web Services valida si el usuario se encuentra o no en el sistema.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada la cedula del estudiante y la clave del mismo: <b>sgaws_validar_estudiante (cedula, clave)</b>		

**Tabla. 17: Historia de Usuario #17**

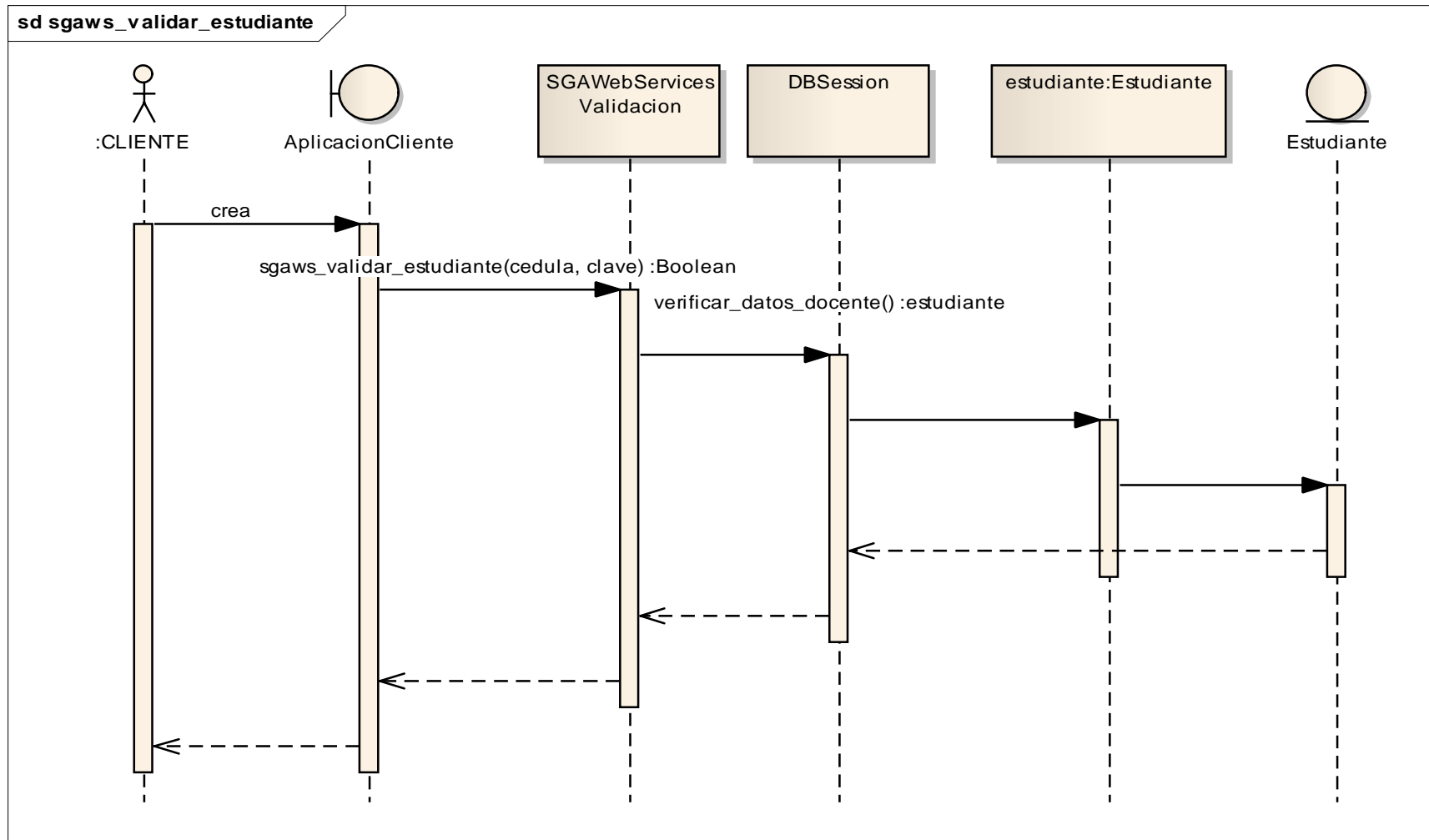


Fig.30: Diagrama de Secuencia del método `sgaws_validar_estudiante(cedula, clave)`



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener Datos Personales de Docente		
<b>Nº:</b> 18	<b>Usuario:</b> Cliente ws	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 3
<b>Descripción:</b> El cliente WS para obtener los datos personales de un docente, envía al ws la cédula del mismo y este le retorna los nombres, apellidos, cédula, título y categoría laboral.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada la cedula del docente: <b>sgaws_datos_docente(cedula)</b>		

**Tabla. 18: Historia de Usuario #18**

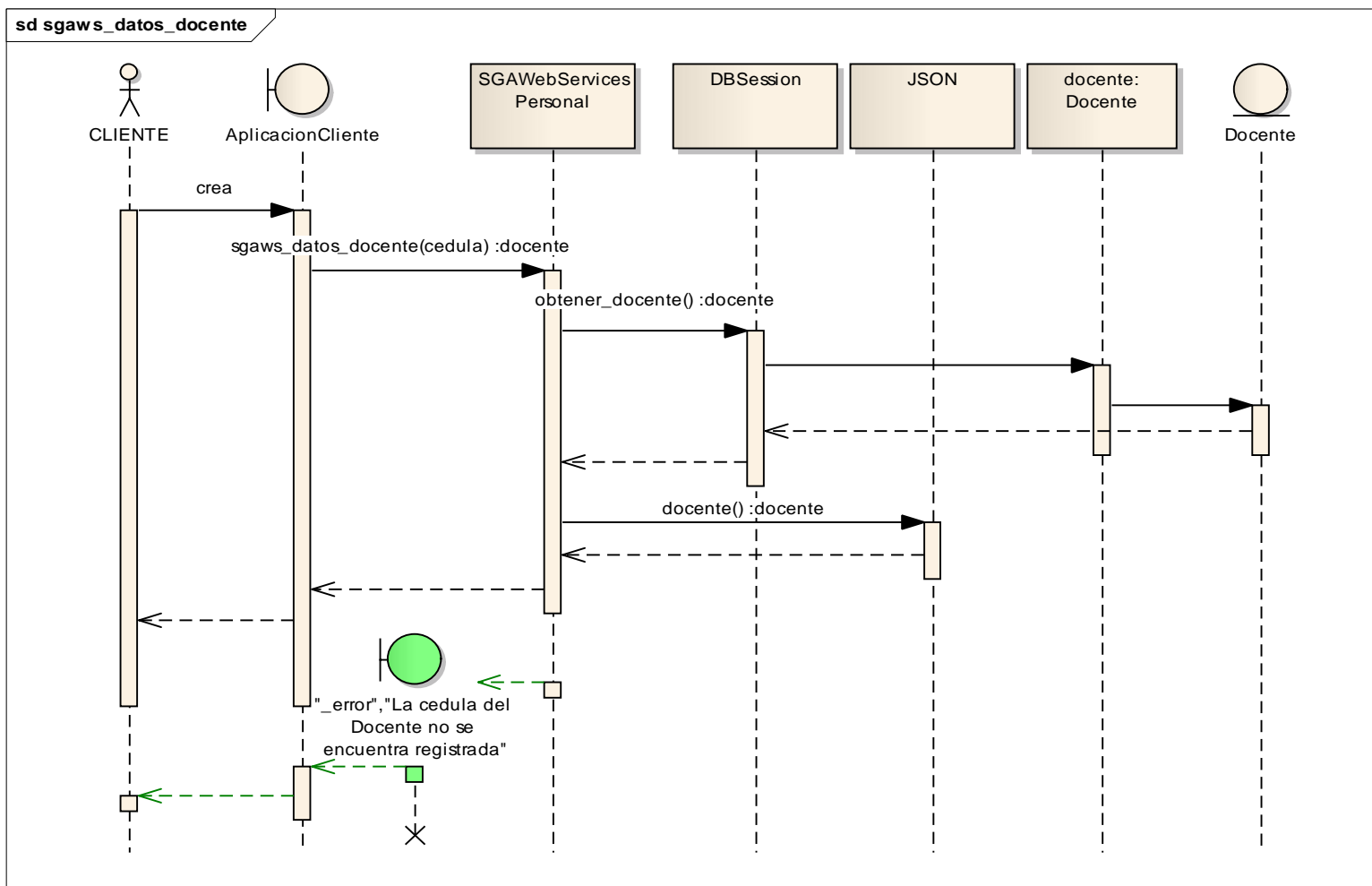


Fig.31: Diagrama de Secuencia del método sgaws\_datos\_docente(cedula)

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener Datos Personales de Estudiante		
<b>Nº:</b> 19	<b>Usuario:</b> Cliente ws	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 3
<b>Descripción:</b> El cliente ws para obtener los datos personales del estudiante, envía al ws la cédula del mismo y este le retorna los nombres, apellidos, fecha de nacimiento, teléfono, celular, dirección actual, país actual, ciudad actual, email y genero.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada la cedula del estudiante: <b>sgaws_datos_estudiante(cedula):</b>		

**Tabla. 19: Historia de Usuario #19**

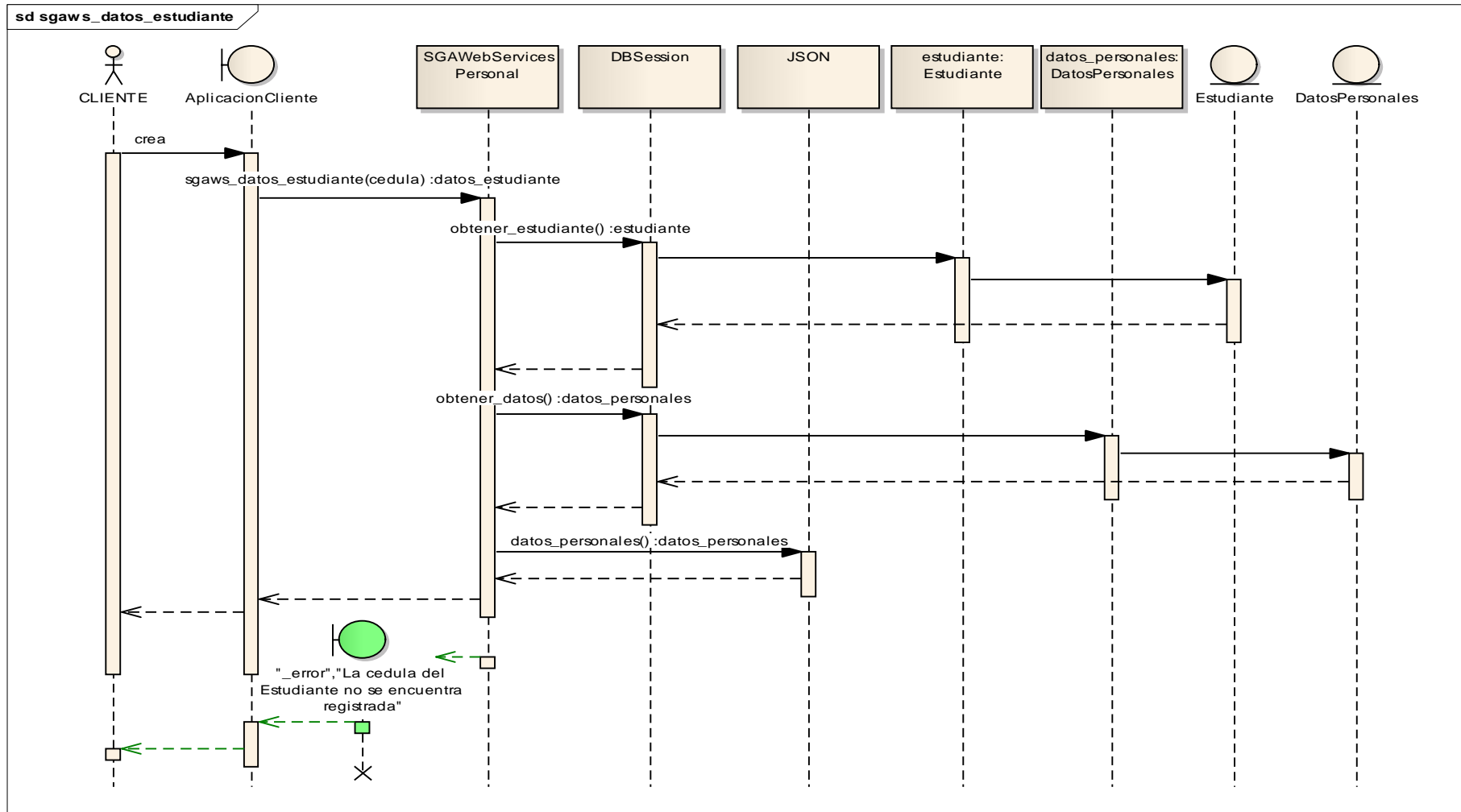


Fig.32: Diagrama de Secuencia del método sgaws\_datos\_estudiante(cedula)



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener los datos de un usuario indicando si es estudiante, docente o ambos.		
<b>Nº:</b> 20	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Prioridad:</b> Media	<b>Prioridad:</b> Media
<b>Descripción:</b> El cliente WS para obtener los datos del usuario, envía la cedula del usuario, el web services verifica si este usuario es un docente, un estudiante o ambos y retorna los nombres, apellidos y cedula del usuario de su respectiva categoría en la que se encuentra registrado en el sistema.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada la cedula del estudiante: <code>sgaws_datos_usuario(cedula)</code>		

**Tabla. 20: Historia de Usuario #20**

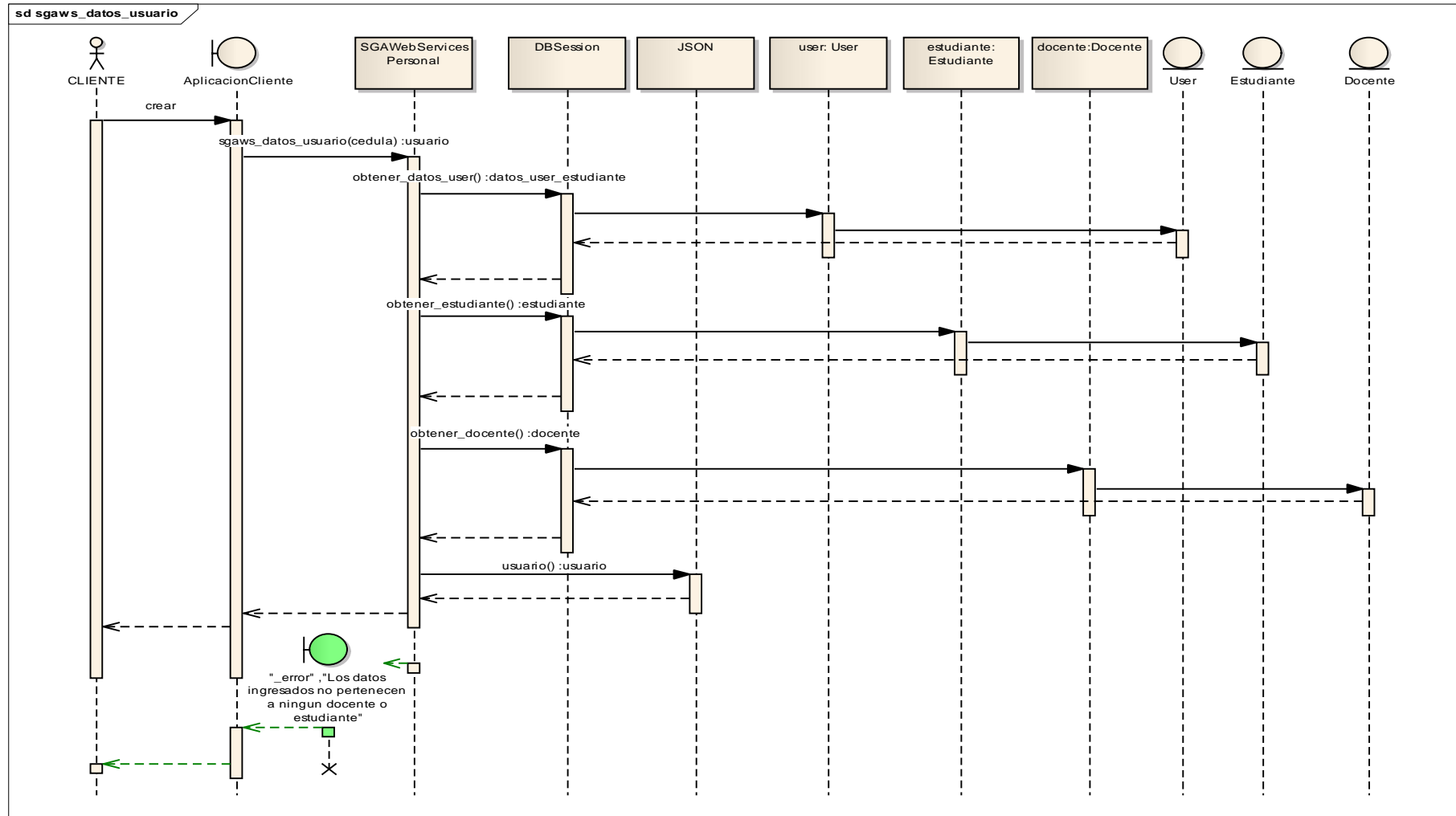


Fig.33: Diagrama de Secuencia del método sgaws\_datos\_usuario(cedula)



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiante matriculados en una oferta académica		
<b>Nº:</b> 21	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 4
<b>Descripción:</b> Para obtener el número de estudiantes matriculados en una oferta académica por el cliente ws, para esto debe enviar el id de la oferta académica y el ws devuelve una lista que contiene el id, la descripción de la oferta y el número de matriculados.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica: <code>sgaws_nmatriculados_oferta(id_oferta)</code>		

**Tabla. 21: Historia de Usuario #21**

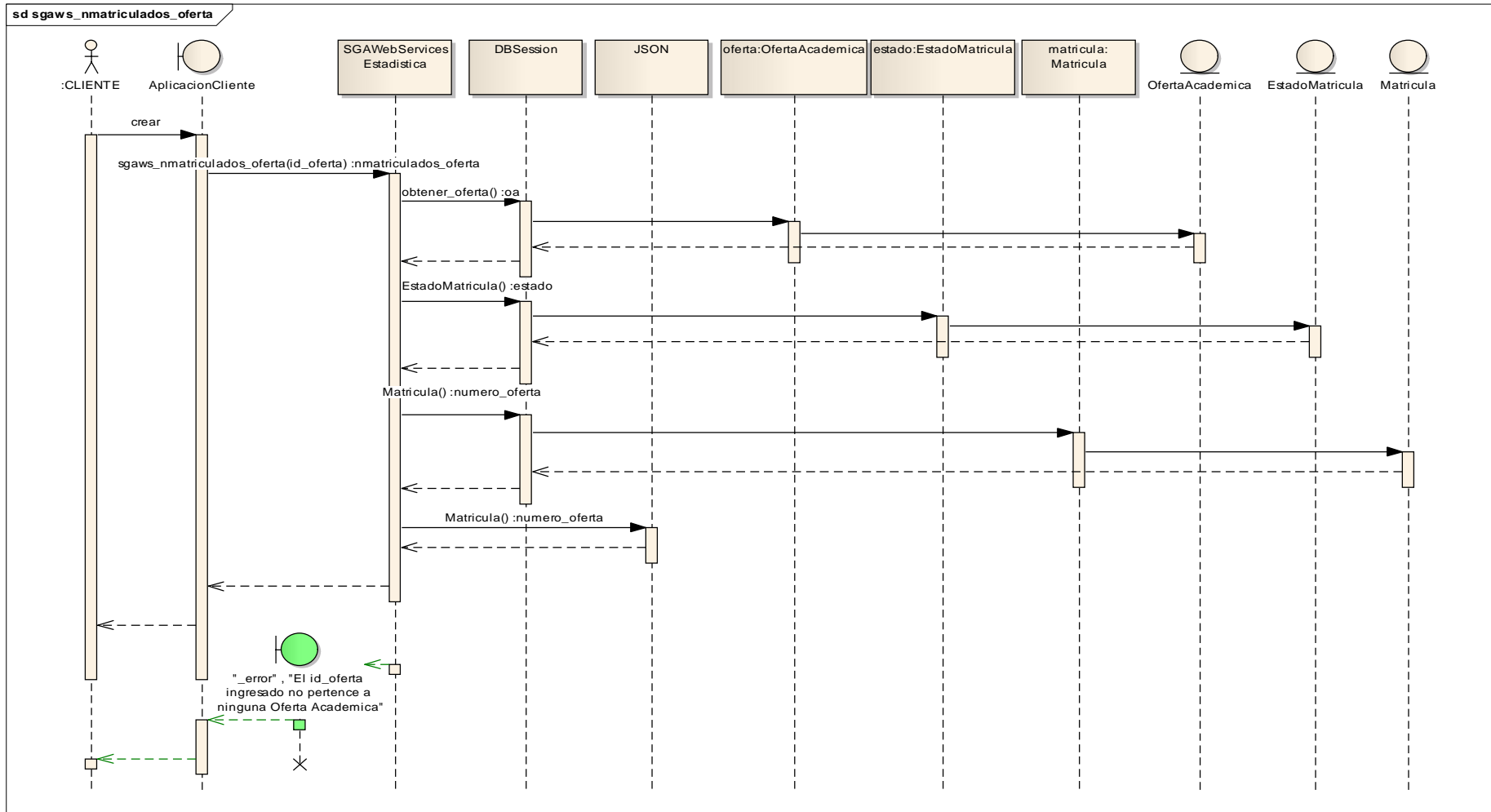


Fig.34: Diagrama de Secuencia del método sgaws\_nmatriculados\_oferta(id\_oferta)

HISTORIA DE USUARIO		
<b>Nombre de Historia:</b> Obtener número de estudiantes matriculados por área en una oferta académica		
<b>Nº:</b> 22	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 4
<b>Descripción:</b> Para obtener el número de estudiantes matriculados por área en una oferta académica por el cliente ws, debe enviar el id de la oferta académica y las siglas del área de la cual desea obtener dicha información y el ws devuelve la descripción de la oferta, el nombre del área y el número de matriculados en esa área.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y las siglas del área: <code>sgaws_nmatriculados_area(id_oferta, siglas)</code>		

Tabla. 22: Historia de Usuario #22

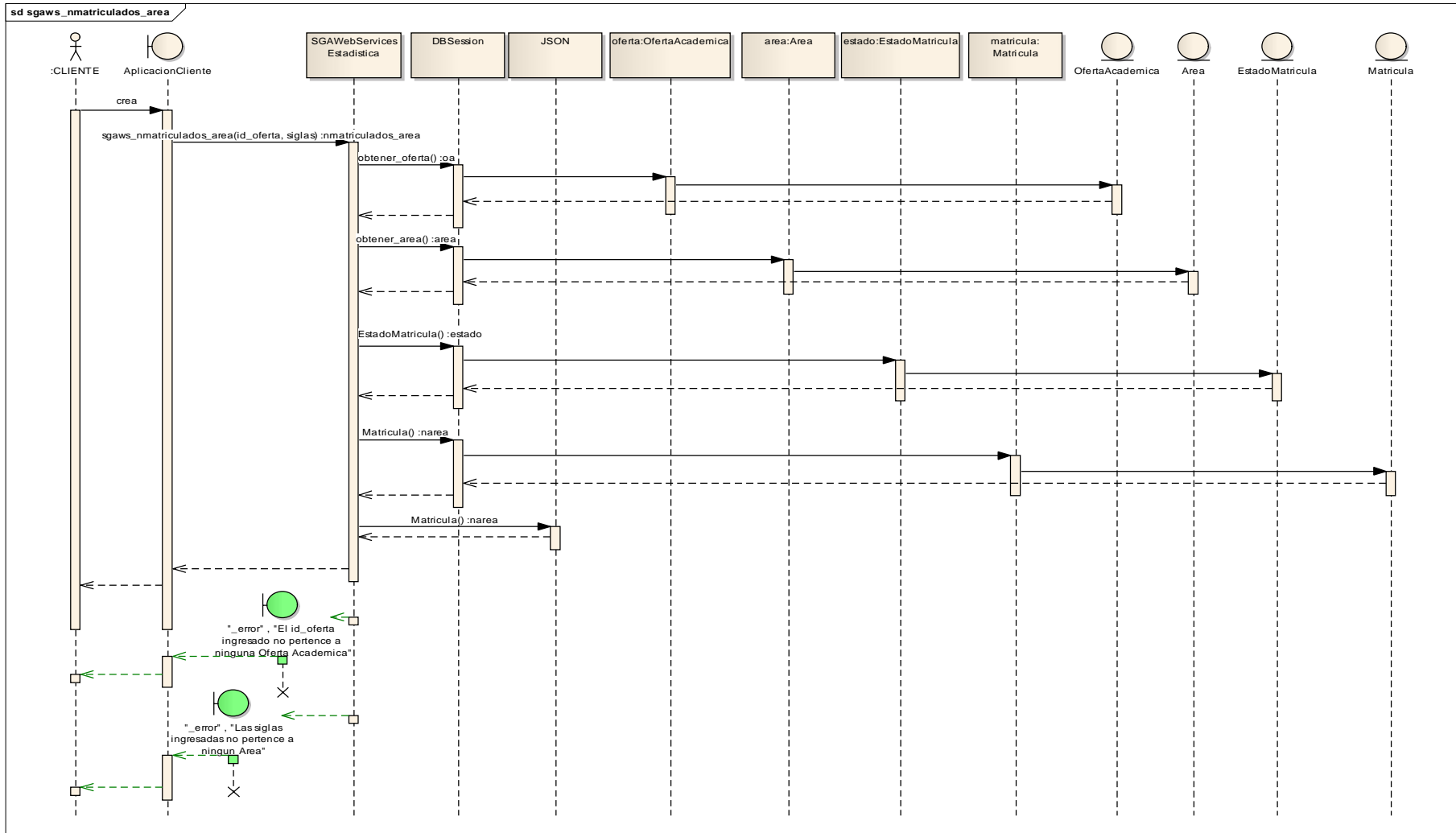


Fig.35: Diagrama de Secuencia del método sgaws\_nmatriculados\_area(id\_oferta, siglas)

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes matriculados por carrera en una oferta académica		
<b>Nº:</b> 23	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 4
<b>Descripción:</b> Se genera una petición del número de estudiantes matriculados por carrera en una determinada oferta académica por el cliente ws, para esto debe enviar el id de la oferta académica, y de la carrera de la cual desea obtener dicha información y el ws retorna una lista en donde retorna la descripción de la oferta, el nombre de la Carrera y el número de matriculados en esa carrera		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y el id de la carrera: <b>sgaws_nmatriculados_carrera (id_oferta, id_carrera)</b>		

**Tabla. 23: Historia de Usuario #23**

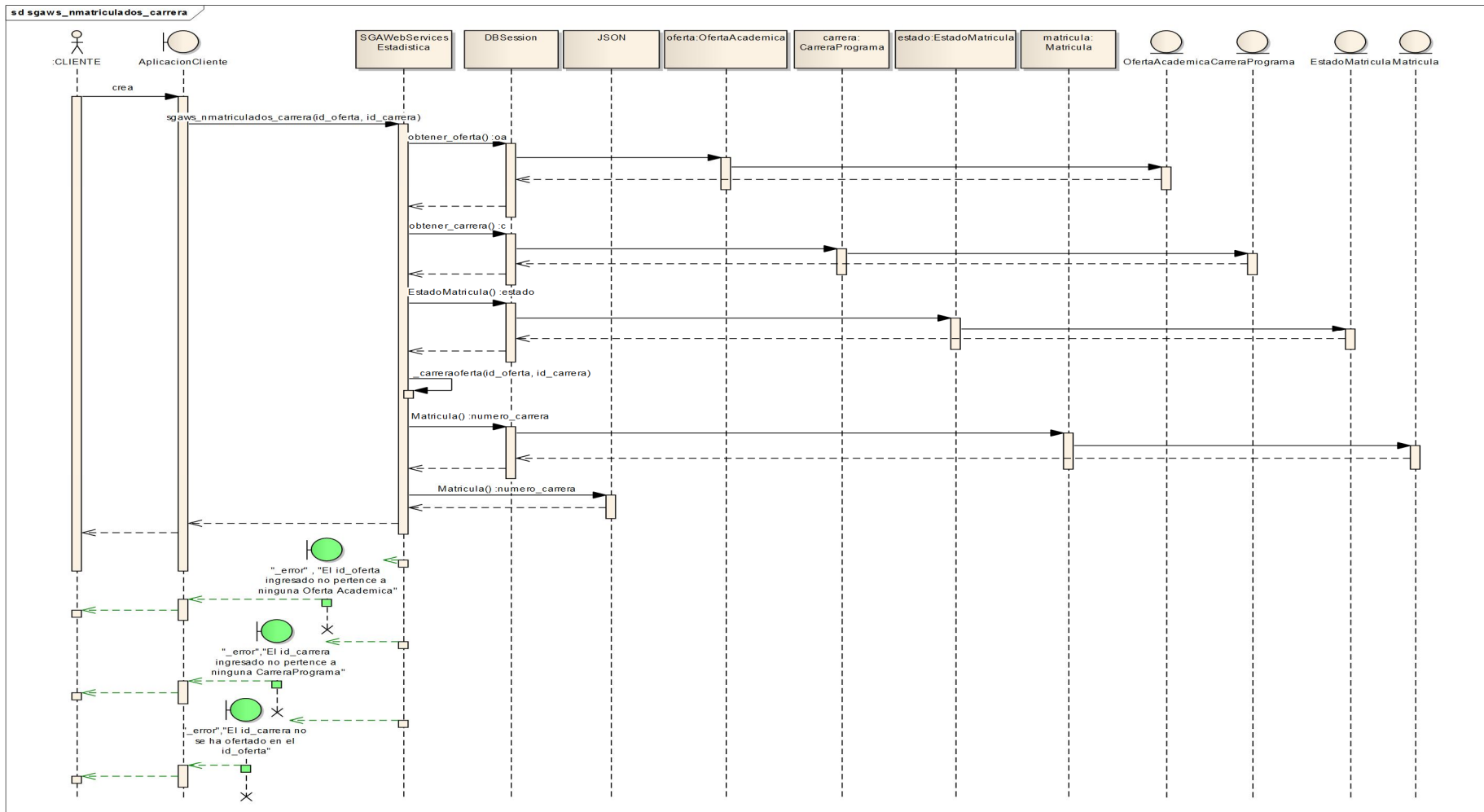


Fig.36: Diagrama de Secuencia del método `sgaws_nmatriculados_carrera (id_oferta, id_carrera)`



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes matriculados por modulo en una oferta académica		
<b>Nº:</b> 24	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 5
<b>Descripción:</b> Se genera una petición del número de estudiantes matriculados por carrera en una determinada oferta académica por el cliente ws, para esto debe enviar el id de la oferta académica, y del módulo de la cual desea obtener dicha información y el ws retorna una lista en donde se indica la descripción de la oferta ,el nombre de la carrera, el número del módulo y el número de matriculados en ese modulo		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y el id del módulo: <b>sgaws_nmatriculados_modulo(id_oferta, id_modulo)</b>		

**Tabla. 24: Historia de Usuario #24**

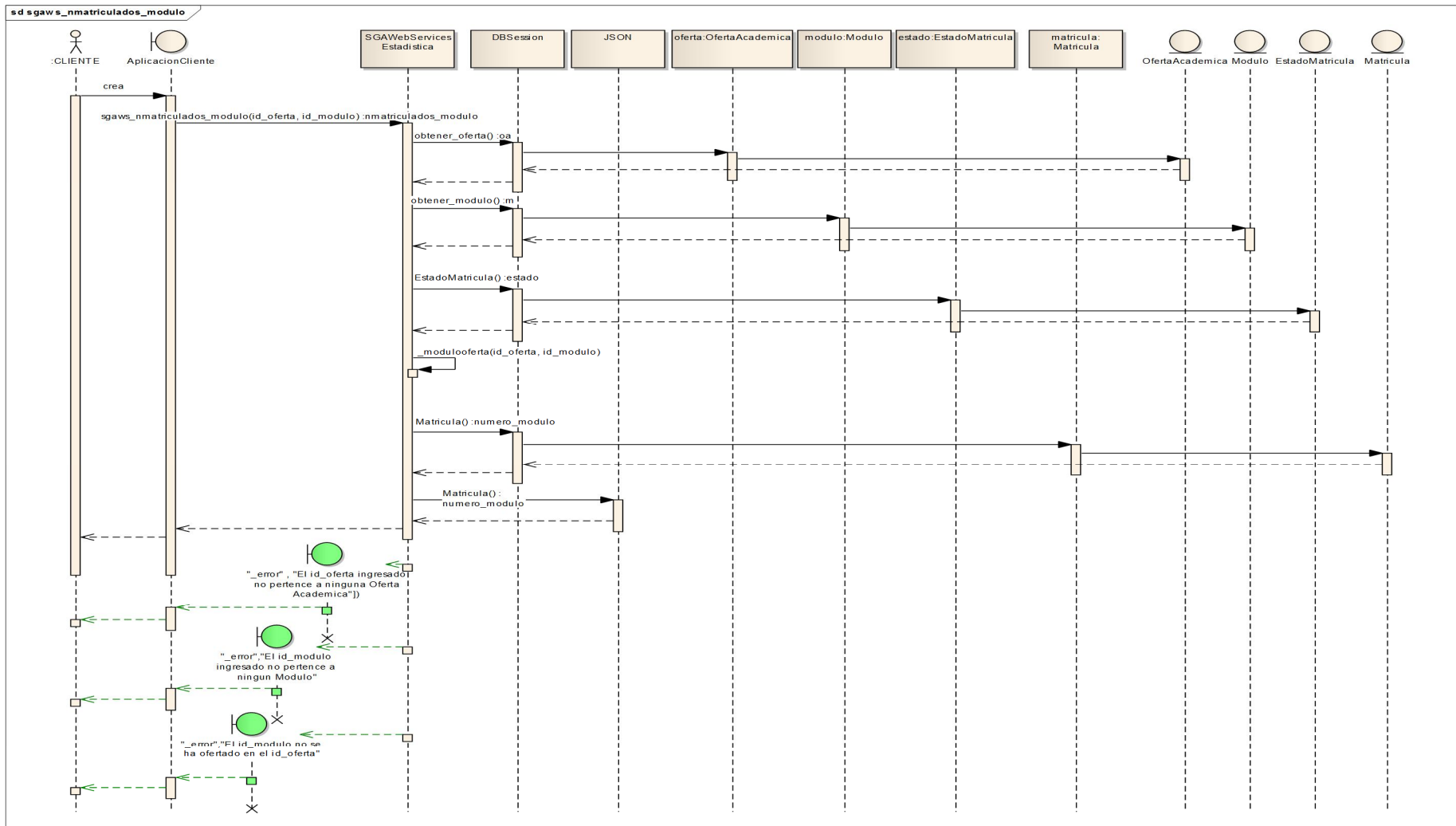


Fig.37: Diagrama de Secuencia del método sgaws\_nmatriculados\_modulo(id\_oferta, id\_modulo)





<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes matriculados por paralelo en una oferta académica		
<b>Nº:</b> 25	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 5
<b>Descripción:</b> Se genera una petición del número de estudiantes matriculados por paralelo en una determinada oferta académica por el cliente ws, para esto debe enviar el id del paralelo de la cual desea obtener dicha información y el ws retorna una lista en donde se indica la descripción de la oferta, el nombre de la Carrera, el número de modulo, el nombre del paralelo y el número de matriculados en ese paralelo.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id del paralelo: <code>sgaws_nmatriculados_paralelo(id_paralelo)</code>		

Tabla. 25: Historia de Usuario #25

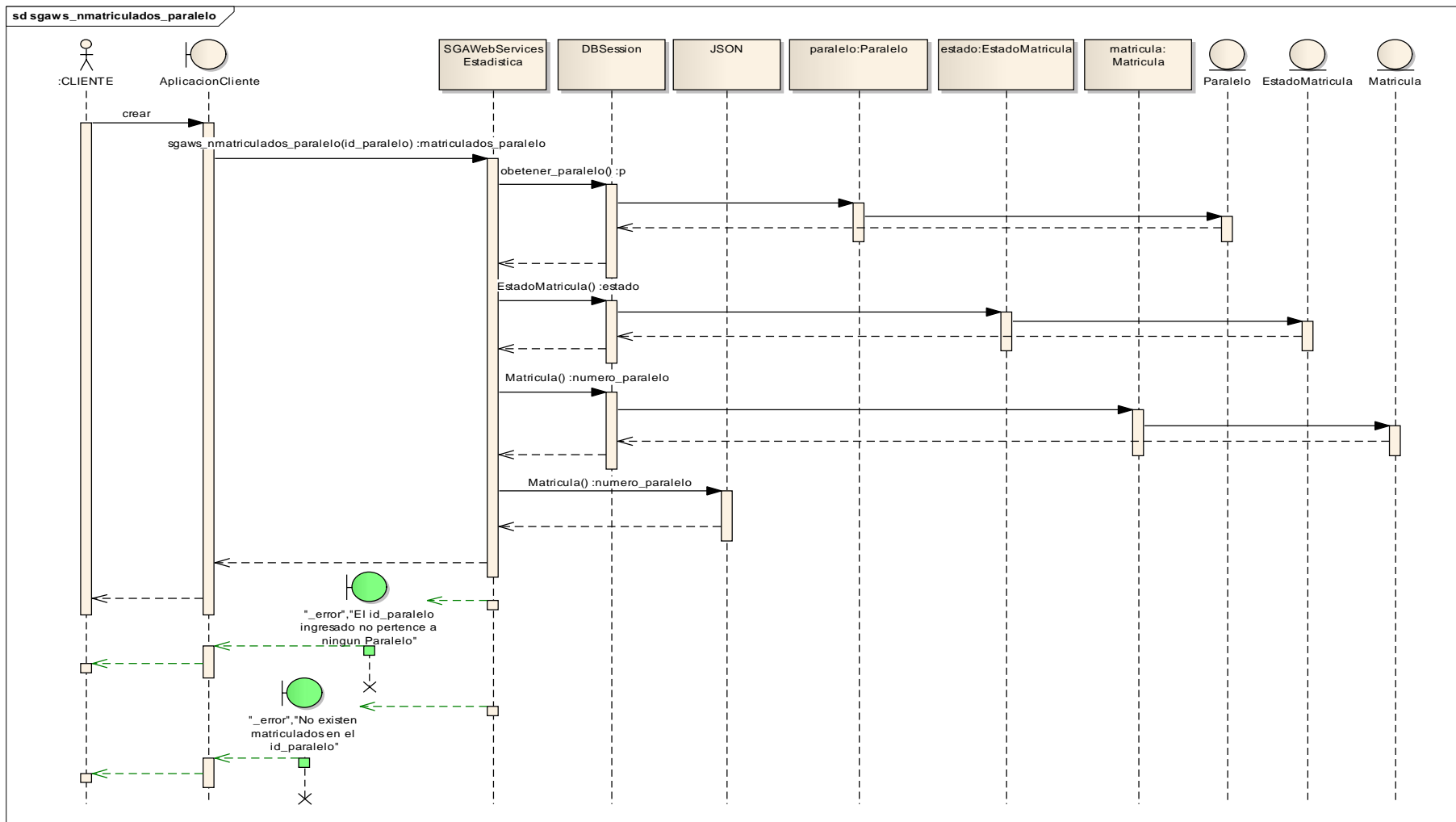


Fig.38: Diagrama de Secuencia del método `sgaws_nmatriculados_paralelo(id_paralelo)`

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes aprobados en una oferta académica		
<b>Nº:</b> 26	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 5
<b>Descripción:</b> Para obtener el número de estudiantes aprobados en una oferta académica por el cliente ws, para esto debe enviar el id de la oferta académica y el ws devuelve una lista que contiene el id, la descripción de la oferta y el número de aprobados.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica: <code>sgaws_naprobados_oferta(id_oferta)</code>		

**Tabla. 26: Historia de Usuario #26**

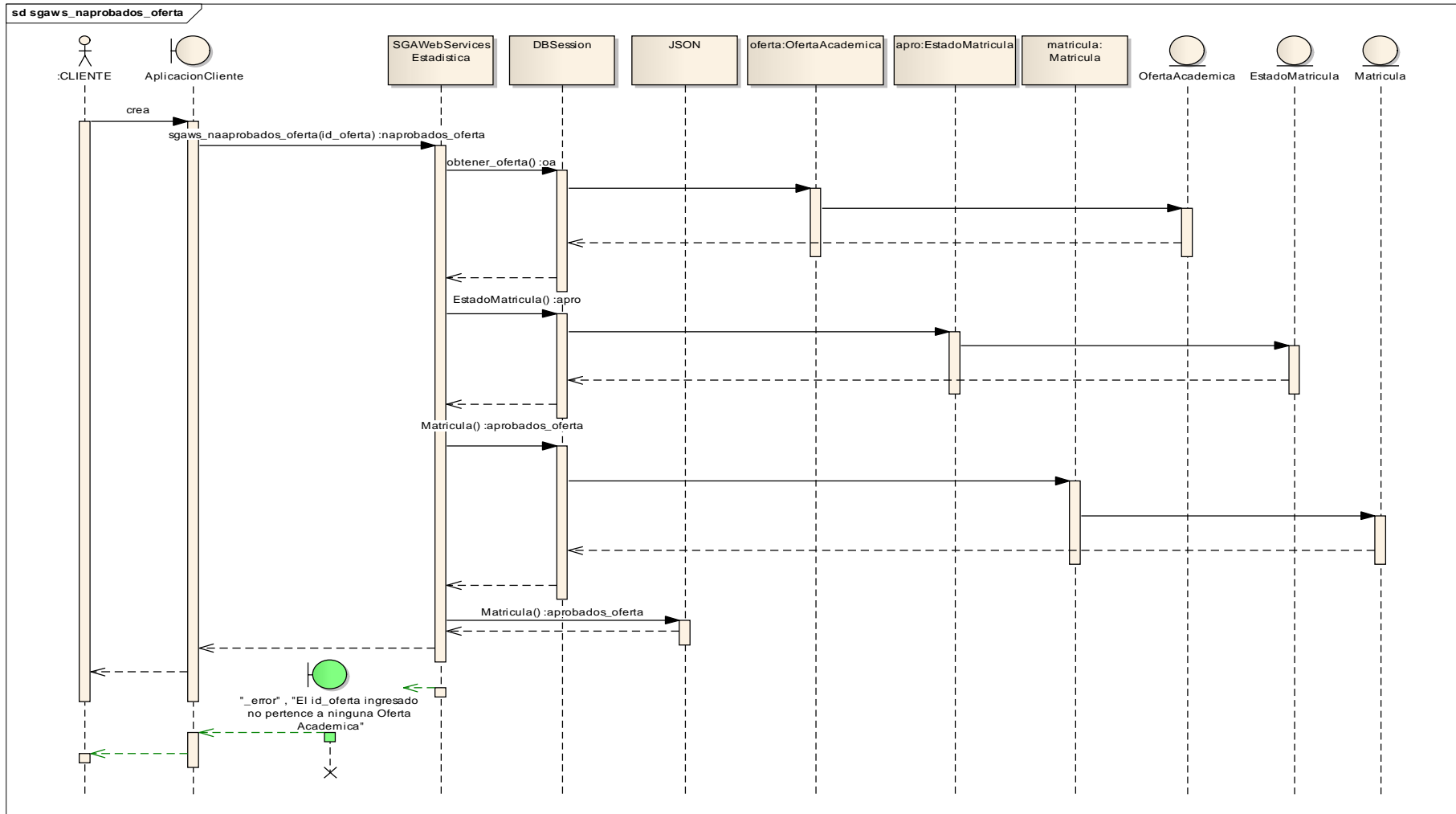


Fig.39: Diagrama de Secuencia del método sgaws\_naprobados\_oferta(id\_oferta)

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes aprobados por área en una oferta académica		
<b>Nº:</b> 27	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 4
<b>Descripción:</b> Para obtener el número de estudiantes aprobados por área en una oferta académica por el cliente ws, debe enviar el id de la oferta académica y las siglas del área de la cual desea obtener dicha información y el ws devuelve la descripción de la oferta, el nombre del área y el número de aprobados en esa área.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y las siglas del área: <b>sgaws_naprobados_area(id_oferta, siglas)</b>		

**Tabla. 27: Historia de Usuario #27**

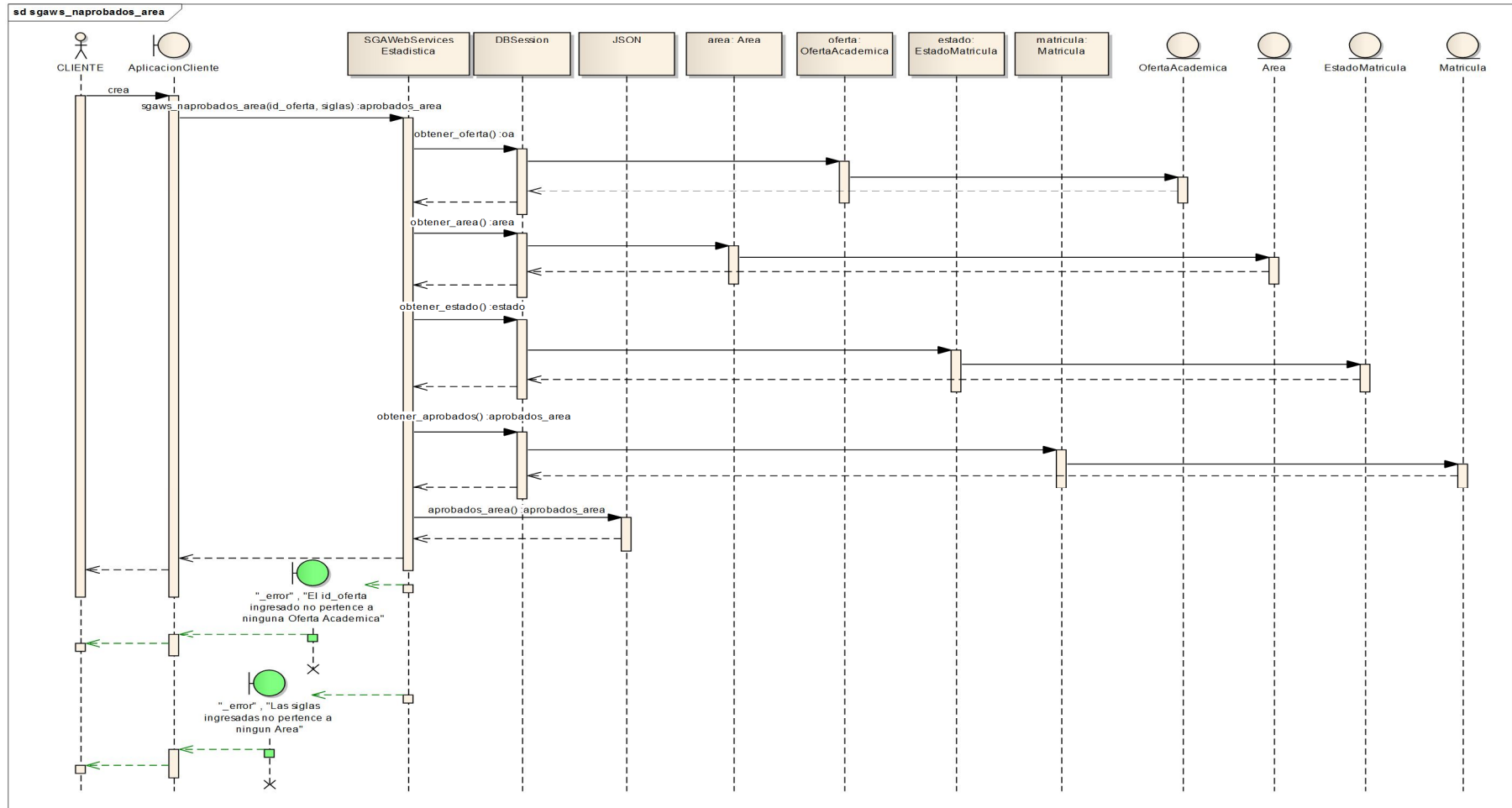


Fig.40: Diagrama de Secuencia del método sgaws\_naprobados\_area(id\_oferta, siglas)

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número estudiantes aprobados por carrera en una oferta académica		
<b>Nº:</b> 28	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 5
<b>Descripción:</b> Se genera una petición del número de estudiantes aprobados por carrera en una determinada oferta académica por el cliente ws, para esto debe enviar el id de la oferta académica, y de la carrera de la cual desea obtener dicha información y el ws retorna una lista en donde retorna la descripción de la oferta, el nombre de la Carrera y el número de aprobados en esa carrera		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y el id de la carrera: <b>sgaws_naprobados_carrera (id_oferta, id_carrera)</b>		

**Tabla. 28: Historia de Usuario #28**

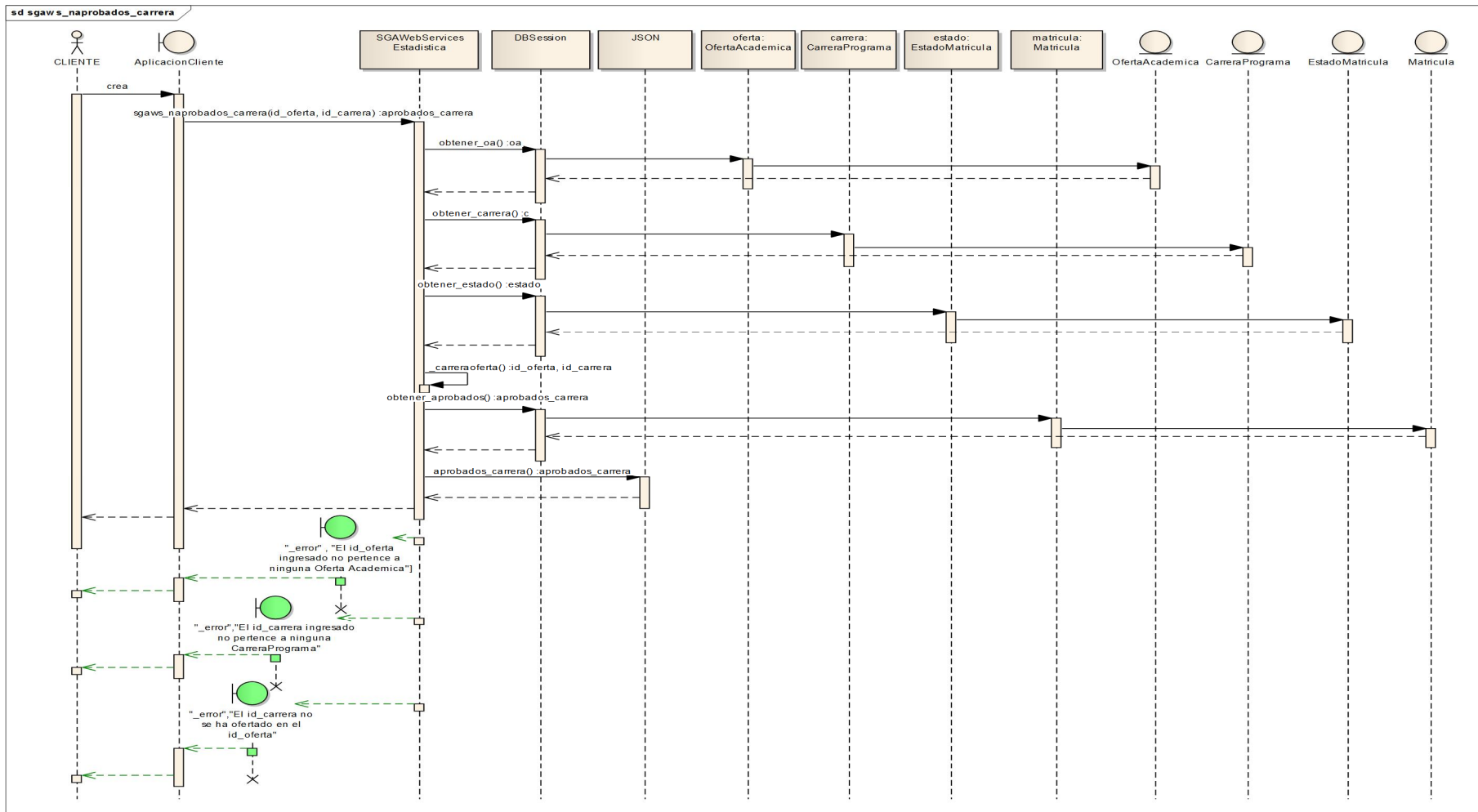


Fig.41: Diagrama de Secuencia del método sgaws\_naprobados\_carrera (id\_oferta, id\_carrera)



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes aprobados por modulo en una oferta académica		
<b>Nº:</b> 29	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 5
<b>Descripción:</b> Se genera una petición del número de estudiantes aprobados por carrera en una determinada oferta académica por el cliente ws, para esto debe enviar el id de la oferta académica, y del módulo de la cual desea obtener dicha información y el ws retorna una lista en donde se indica la descripción de la oferta ,el nombre de la carrera, el número del módulo y el número de aprobados en ese modulo		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y el id del módulo: <code>sgaws_naprobados_modulo(id_oferta, id_modulo)</code>		

**Tabla. 29: Historia de Usuario #29**

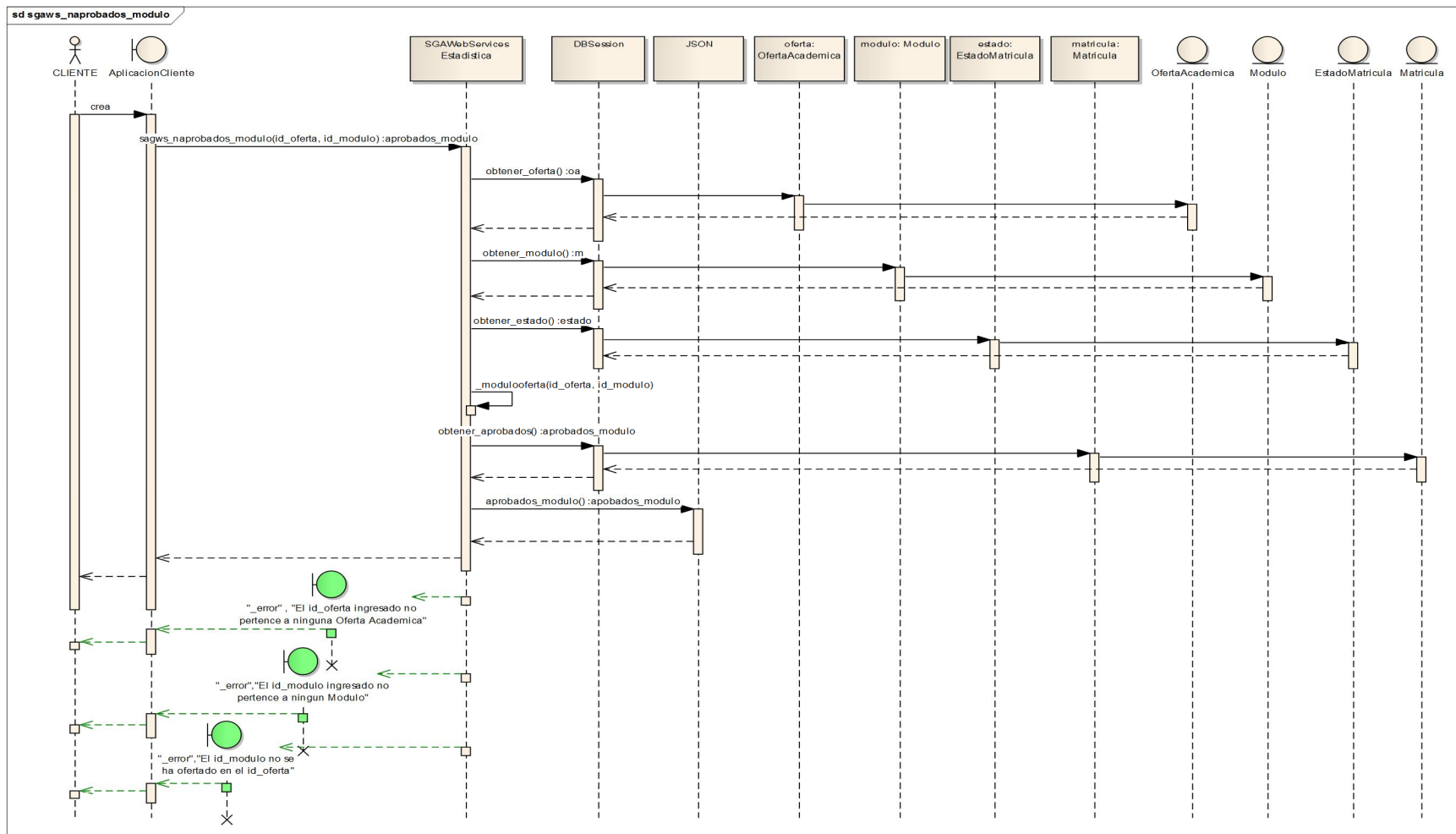


Fig.42: Diagrama de Secuencia del método sgaws\_naprobados\_modulo(id\_oferta, id\_modulo)



HISTORIA DE USUARIO		
<b>Nombre de Historia:</b> Obtener número de estudiantes aprobados por paralelo en una oferta académica		
<b>Nº:</b> 30	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Prioridad:</b> Media	<b>Prioridad:</b> Media
<b>Descripción:</b> Se genera una petición del número de estudiantes aprobados por paralelo en una determinada oferta académica por el cliente ws, para esto debe enviar el id del paralelo de la cual desea obtener dicha información y el ws retorna una lista en donde se indica la descripción de la oferta, el nombre de la Carrera, el número de modulo, el nombre del paralelo y el número de aprobados en ese paralelo.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id del paralelo: <b>sgaws_naprobados_paralelo(id_paralelo)</b>		

Tabla. 30: Historia de Usuario #30

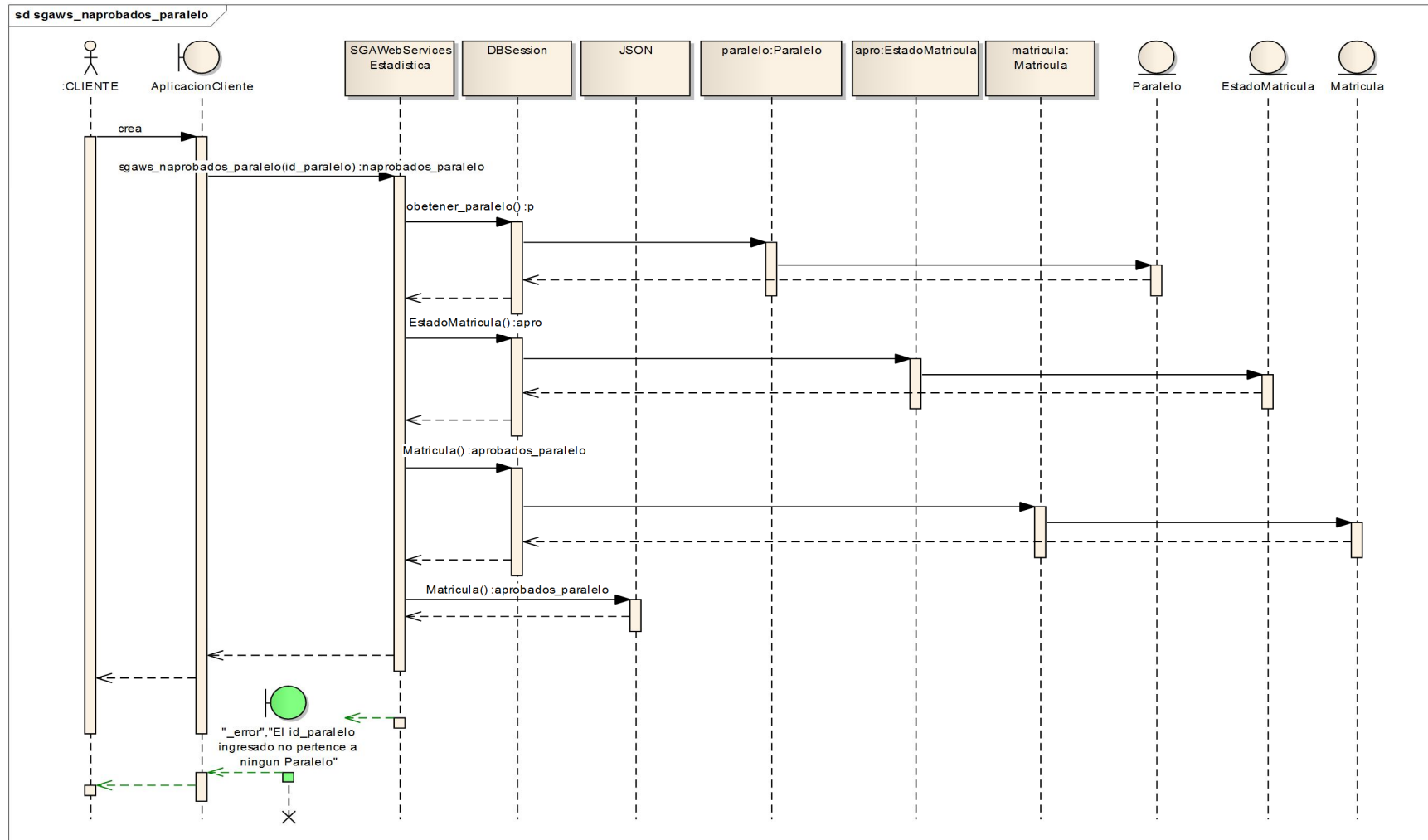


Fig.43: Diagrama de Secuencia del método sgaws\_naprobados\_paralelo(id\_paralelo)

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes reprobados en una oferta académica		
<b>Nº:</b> 31	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 5
<b>Descripción:</b> Para obtener el número de estudiantes reprobados en una oferta académica por el cliente ws, para esto debe enviar el id de la oferta académica y el ws devuelve una lista que contiene el id, la descripción de la oferta y el número de reprobados.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica: <b>sgaws_nreprobados_oferta(id_oferta)</b>		

**Tabla. 31: Historia de Usuario #31**

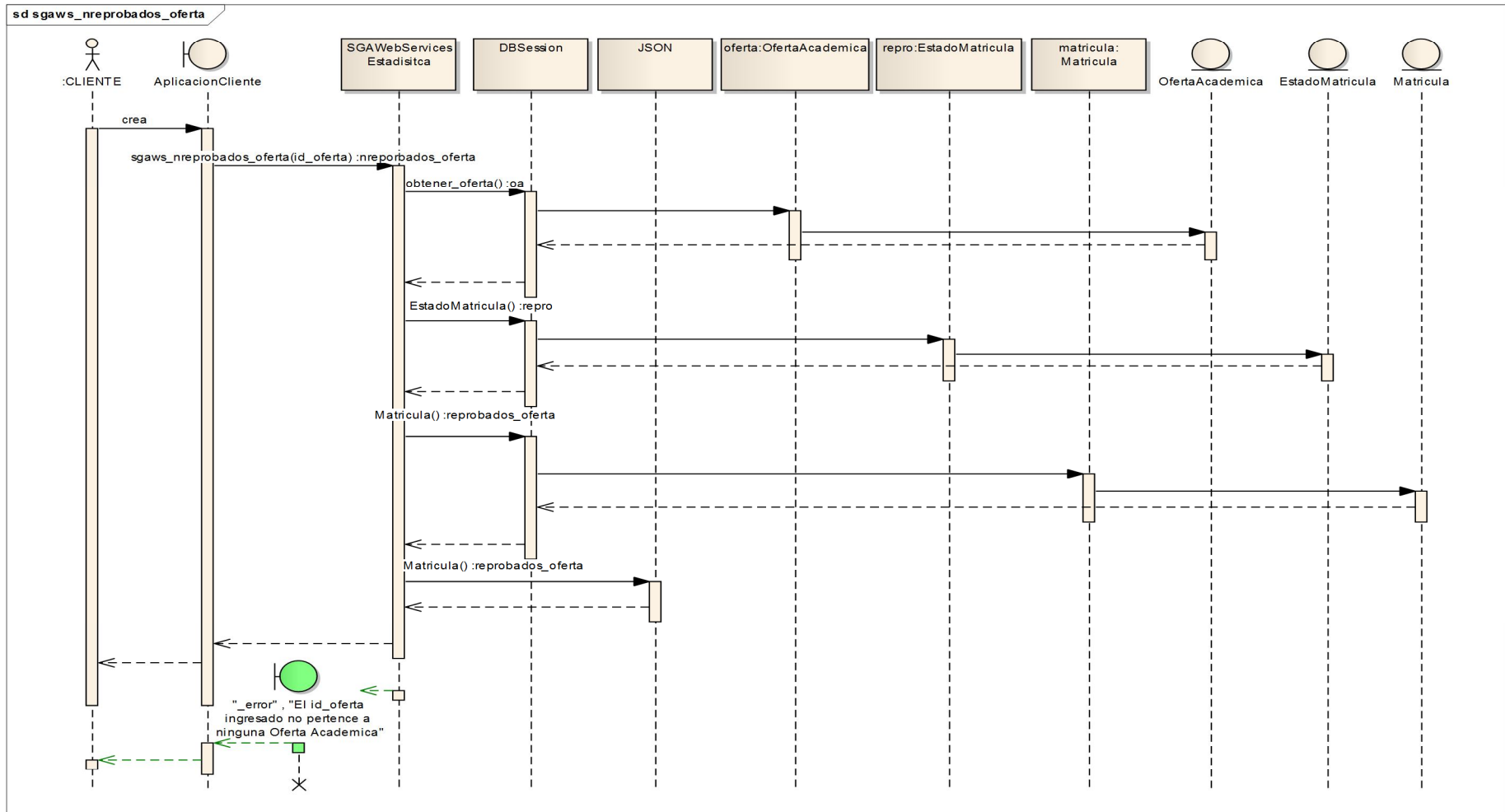


Fig.44: Diagrama de Secuencia del método sgaws\_nreprobados\_oferta(id\_oferta)



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes reprobados por área en una oferta académica		
<b>Nº:</b> 32	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 4
<b>Descripción:</b> Para obtener el número de estudiantes reprobados por área en una oferta académica por el cliente ws, debe enviar el id de la oferta académica y las siglas del área de la cual desea obtener dicha información y el ws devuelve la descripción de la oferta, el nombre del área y el número de reprobados en esa área.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y las siglas del área: <b>sgaws_nreprobados_area(id_oferta, siglas)</b>		

**Tabla. 32: Historia de Usuario #32**

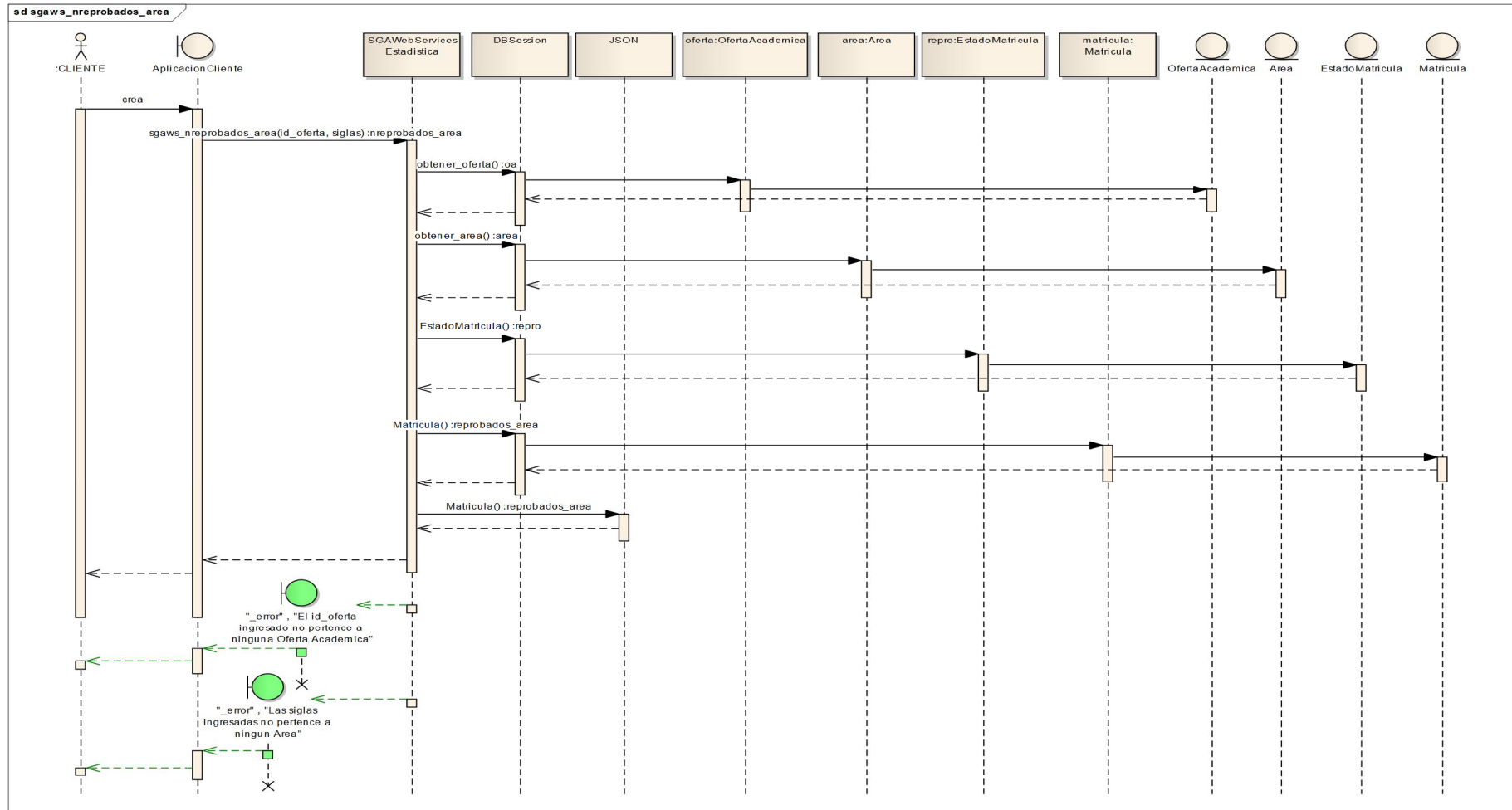


Fig.45: Diagrama de Secuencia del método sgaws\_nreprobados\_area(id\_oferta, siglas)



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número estudiantes reprobados por carrera en una oferta académica		
<b>Nº:</b> 33	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 5
<b>Descripción:</b> Se genera una petición del número de estudiantes reprobados por carrera en una determinada oferta académica por el cliente ws, para esto debe enviar el id de la oferta académica, y de la carrera de la cual desea obtener dicha información y el ws retorna una lista en donde retorna la descripción de la oferta, el nombre de la Carrera y el número de reprobados en esa carrera		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y el id de la carrera: <b>sgaws_nreprobados_carrera (id_oferta, id_carrera)</b>		

**Tabla. 33: Historia de Usuario #33**

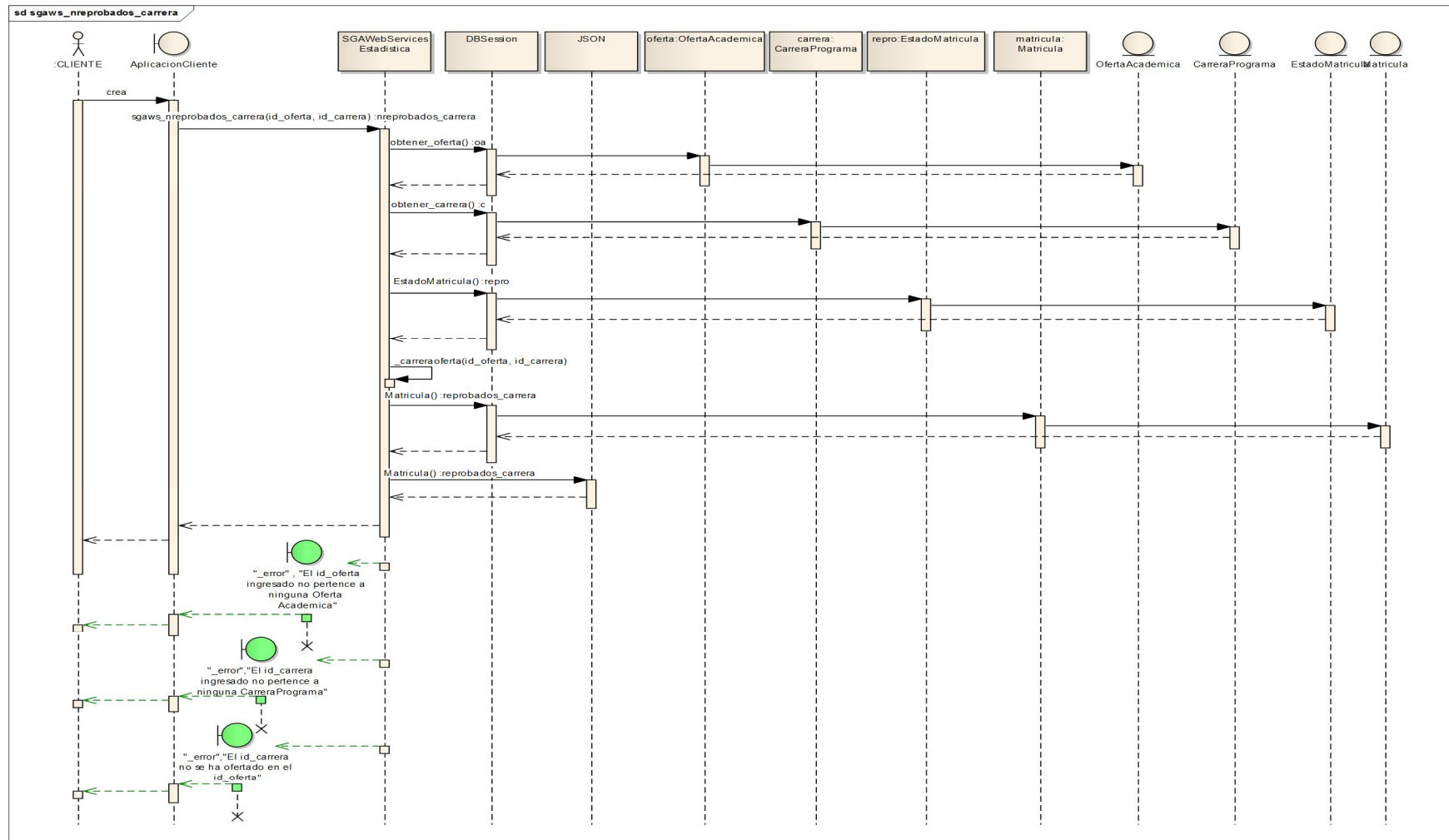


Fig.46: Diagrama de Secuencia del método sgaws\_nreprobados\_carrera (id\_oferta, id\_carrera)



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes reprobados por modulo en una oferta académica		
<b>Nº:</b> 34	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Puntos(estimación):</b> 3 días	<b>Iteración:</b> 5
<b>Descripción:</b> Se genera una petición del número de estudiantes reprobados por carrera en una determinada oferta académica por el cliente ws, para esto debe enviar el id de la oferta académica, y del módulo de la cual desea obtener dicha información y el ws retorna una lista en donde se indica la descripción de la oferta ,el nombre de la carrera, el número del módulo y el número de reprobados en ese modulo		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id de la oferta académica y el id del módulo: <code>sgaws_nreprobados_modulo(id_oferta, id_modulo)</code>		

Tabla. 34: Historia de Usuario #34

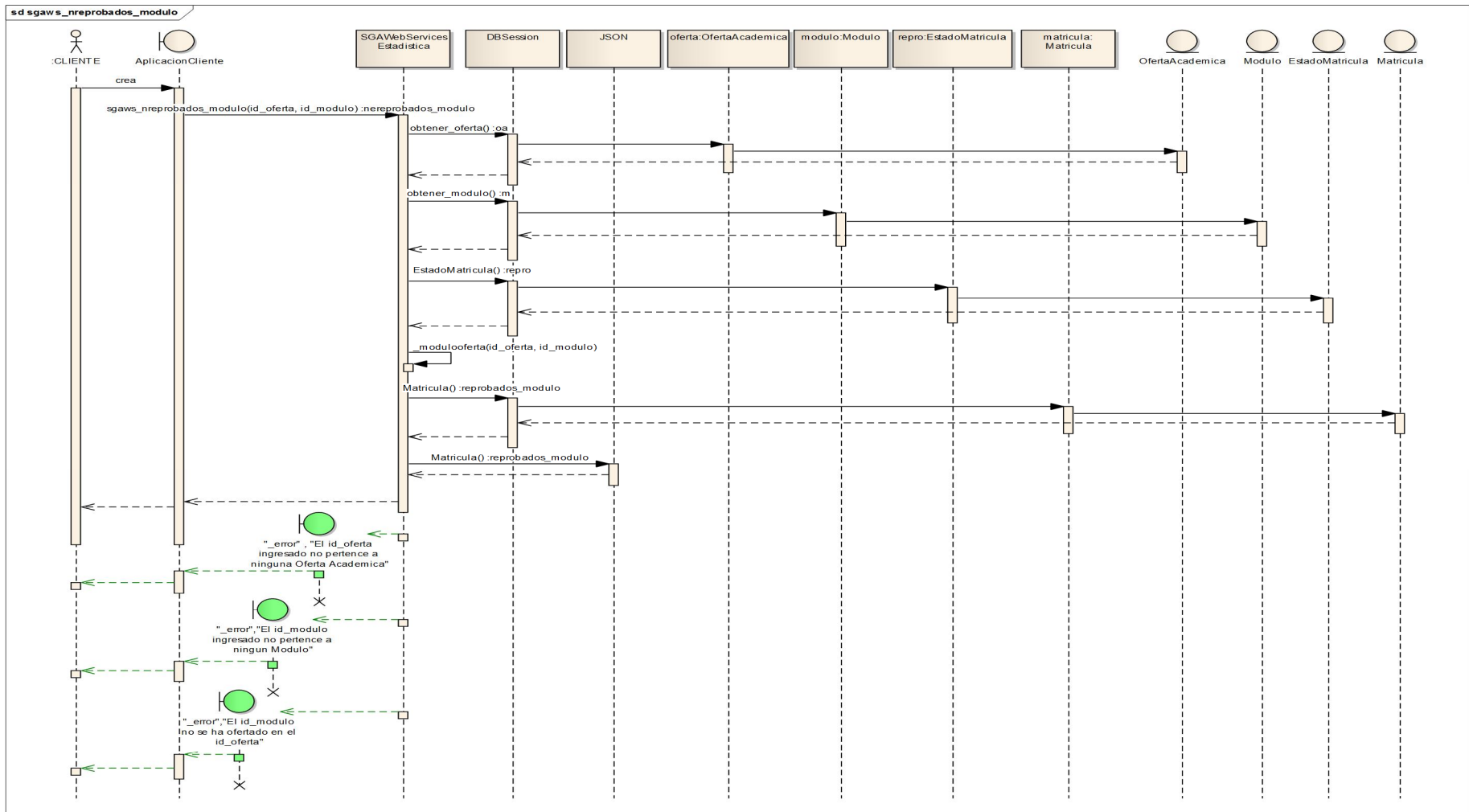


Fig.47: Diagrama de Secuencia del método sgaws\_nreprobados\_modulo(id\_oferta, id\_modulo)



<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Obtener número de estudiantes reprobados por paralelo en una oferta académica		
<b>Nº:</b> 35	<b>Usuario:</b> Cliente WS	
<b>Prioridad:</b> Media	<b>Prioridad:</b> Media	<b>Prioridad:</b> Media
<b>Descripción:</b> El cliente ws para obtener el número de estudiantes reprobados por paralelo en una determinada oferta académica debe enviar el id del paralelo de la cual desea obtener dicha información y el ws retorna una lista en donde se indica la descripción de la oferta, el nombre de la Carrera, el número de modulo, el nombre del paralelo y el número de reprobados en ese paralelo.		
<b>Observaciones:</b> Este método tiene como parámetro de entrada el id del paralelo: <b>sgaws_nreprobados_paralelo(id_paralelo)</b>		

Tabla. 35: Historia de Usuario #35





7.1.1.5. Modelo de Clases Final

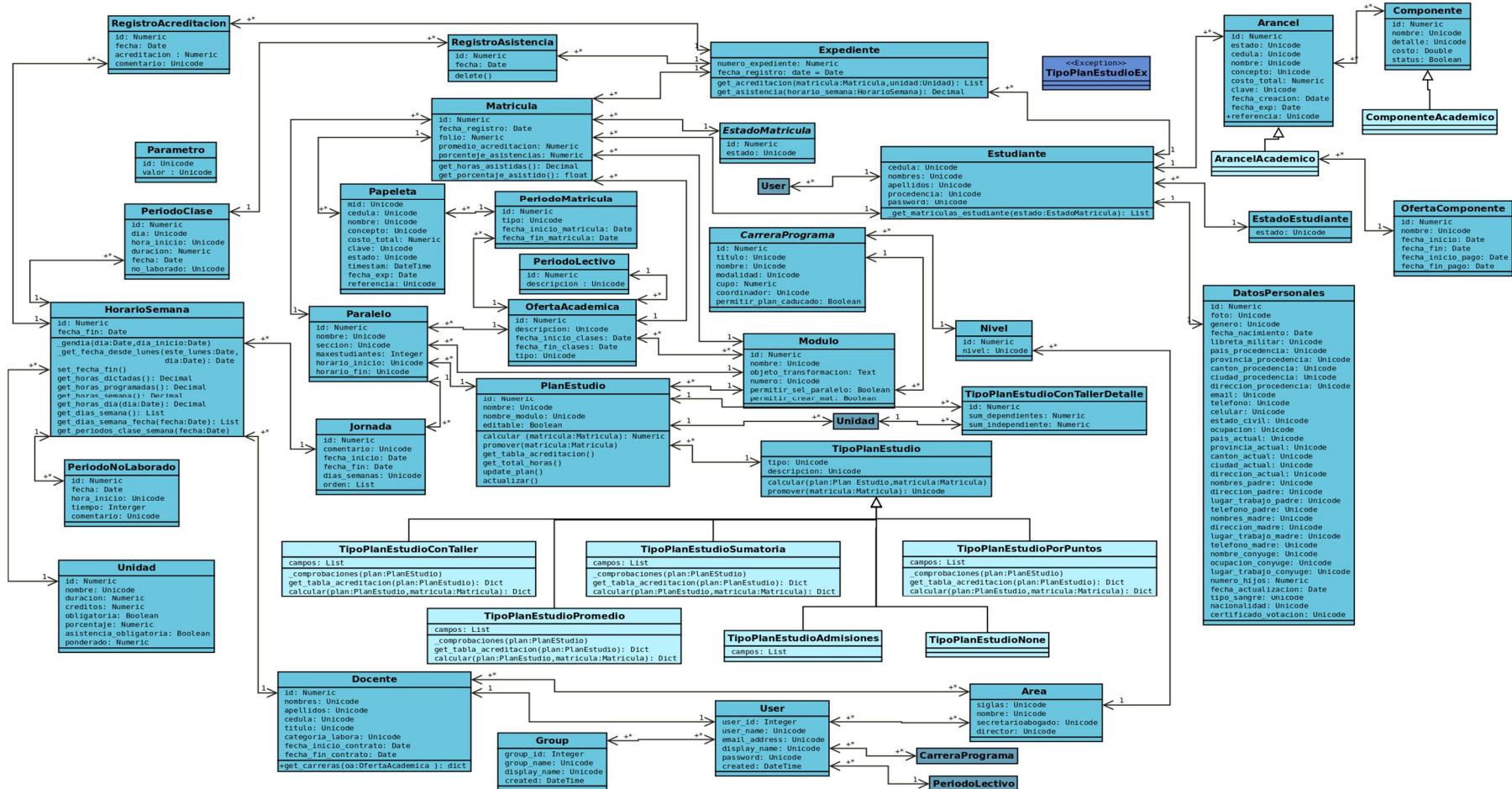


Fig.49: Diagrama de Clases Final





### 7.1.1.7. Diagrama de Despliegue

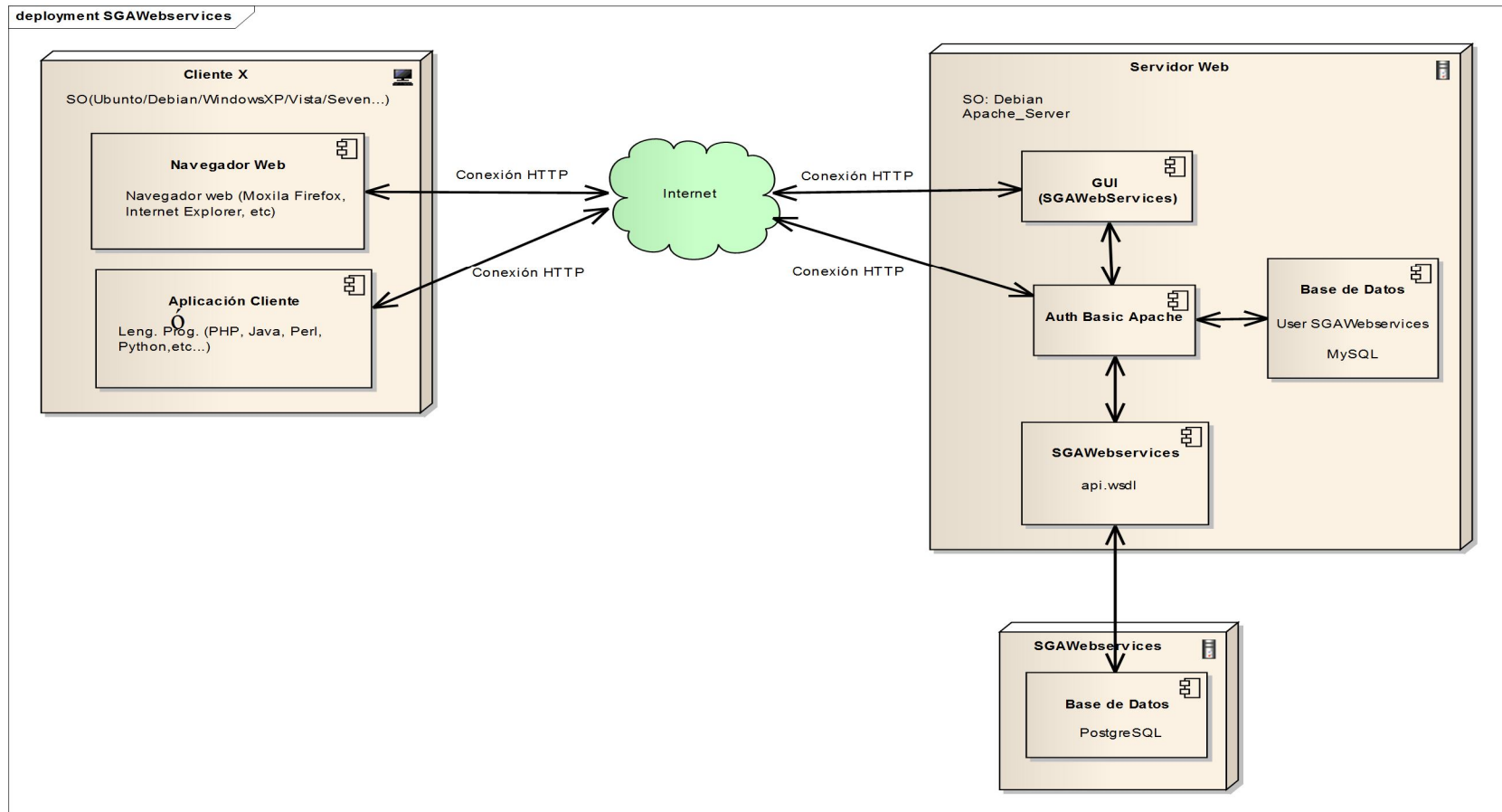


Fig.51: Diagrama de despliegue

### 7.1.2. Instalación y Configuración de Herramientas para el Desarrollo del SGASWebServices

Una vez realizado en análisis y diseño se realizaron las instalaciones y configuraciones necesarias que permitieron la codificación del SGASWebServices. El código se integró y fue subido al repositorio de proyectos de la Unidad de Desarrollo de Software de la Universidad Nacional de Loja, esto como mínimo una vez al día, luego de realizadas las pruebas a nivel de desarrolladores sobre la totalidad de los métodos del webservice hasta que funcione adecuadamente.

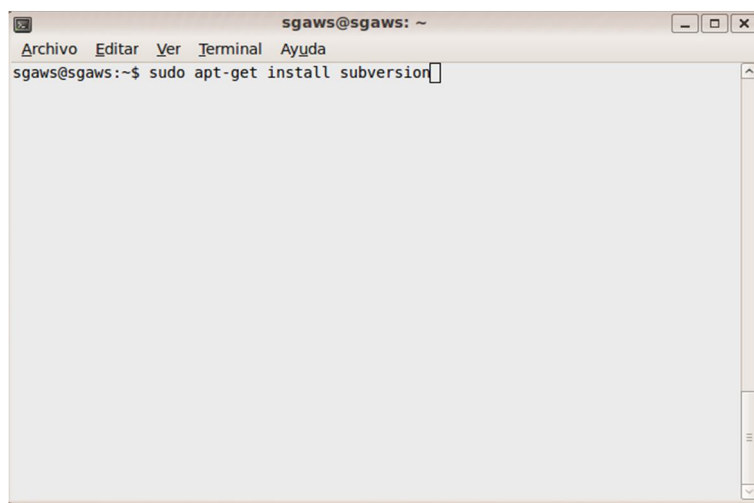
A continuación se detallan las instalaciones y configuraciones realizadas:

#### 7.1.2.1. Instalación del sistema de control de versiones subversión

**Subversion** herramienta que sirvió para subir el proyecto y los cambios hechos al mismo al repositorio de software de la Unidad de Desarrollo de Software para tener un control de las modificaciones efectuadas a lo largo del desarrollo del mismo, el proceso de instalación llevado a cabo fue el siguiente:

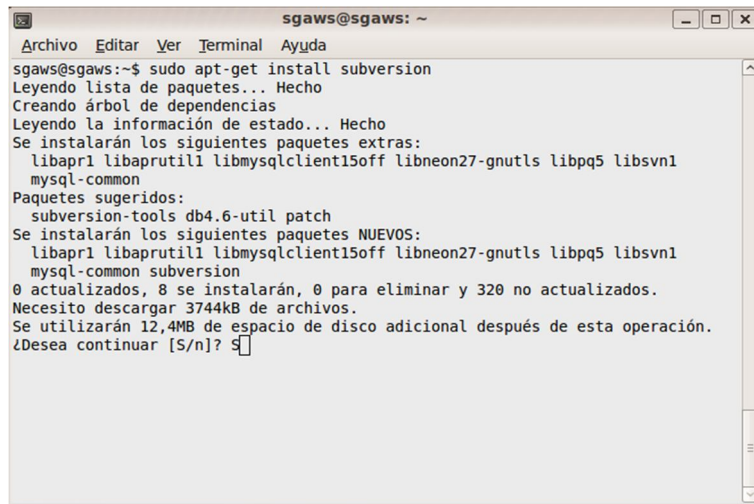
1. Ejecutar en consola o terminal el siguiente comando:

```
sudo apt-get install subversion
```



**Fig.52: Instalación de subversion**

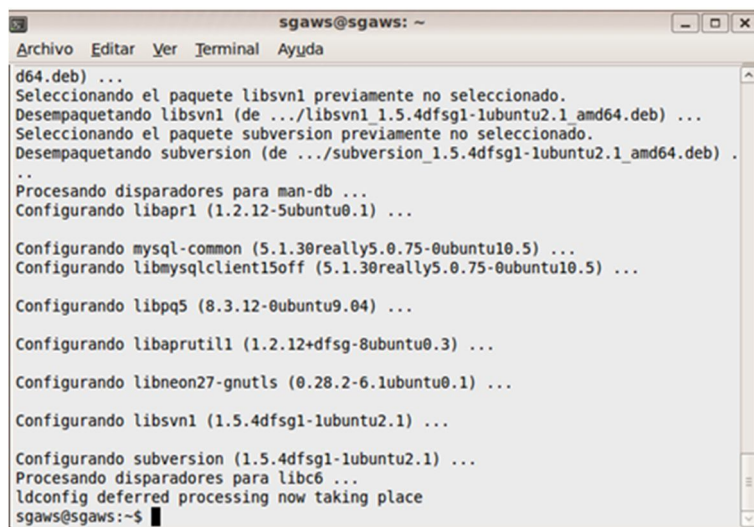
## 2. Confirmar la instalación:



```
sgaws@sgaws: ~
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~$ sudo apt-get install subversion
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 libapr1 libaprutil1 libmysqlclient15off libneon27-gnutls libpq5 libsvn1
mysql-common
Paquetes sugeridos:
 subversion-tools db4.6-util patch
Se instalarán los siguientes paquetes NUEVOS:
 libapr1 libaprutil1 libmysqlclient15off libneon27-gnutls libpq5 libsvn1
mysql-common subversion
0 actualizados, 8 se instalarán, 0 para eliminar y 320 no actualizados.
Necesito descargar 3744kB de archivos.
Se utilizarán 12,4MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? S
```

**Fig.53: Confirmación de instalación de subversion**

## 3. Esperar a que finalice el proceso de la instalación:



```
sgaws@sgaws: ~
Archivo Editar Ver Terminal Ayuda
d64.deb) ...
Seleccionando el paquete libsvn1 previamente no seleccionado.
Desempaquetando libsvn1 (de ../libsvn1_1.5.4dfsg1-1ubuntu2.1_amd64.deb) ...
Seleccionando el paquete subversion previamente no seleccionado.
Desempaquetando subversion (de ../subversion_1.5.4dfsg1-1ubuntu2.1_amd64.deb) .
..
Procesando disparadores para man-db ...
Configurando libapr1 (1.2.12-5ubuntu0.1) ...

Configurando mysql-common (5.1.30really5.0.75-0ubuntu10.5) ...
Configurando libmysqlclient15off (5.1.30really5.0.75-0ubuntu10.5) ...

Configurando libpq5 (8.3.12-0ubuntu9.04) ...

Configurando libaprutil1 (1.2.12+dfsg-8ubuntu0.3) ...

Configurando libneon27-gnutls (0.28.2-6.1ubuntu0.1) ...

Configurando libsvn1 (1.5.4dfsg1-1ubuntu2.1) ...

Configurando subversion (1.5.4dfsg1-1ubuntu2.1) ...
Procesando disparadores para libc6 ...
ldconfig deferred processing now taking place
sgaws@sgaws:~$
```

**Fig.54: Finalización de proceso de la instalación de subversion**

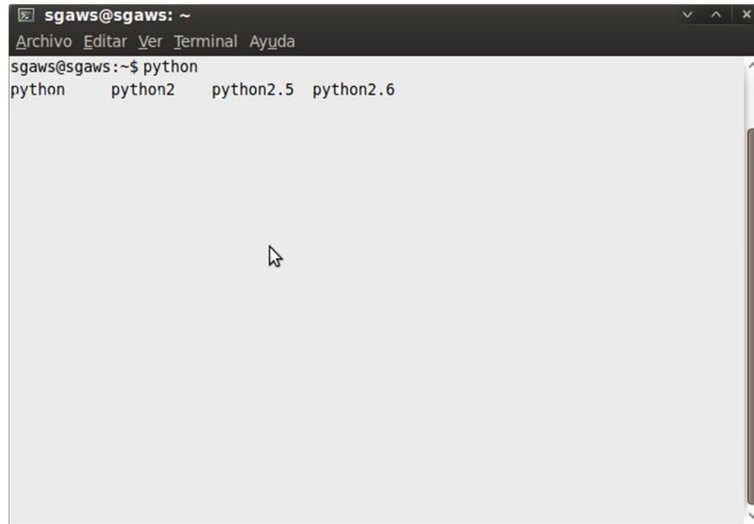
### 7.1.2.2. Verificar la versión de Python instalada

Para el desarrollo de la aplicación se optó por el sistema operativo Ubuntu 9.10, en vista de que se procuró el uso de sólo software libre.

Primeramente se verificó las versiones de Python instaladas y se seleccionó la versión de Python 2.6, para realizar esto realizar lo siguiente:

1. Poner en consola lo siguiente:

```
$ python
```



**Fig.55: Versiones de Python instaladas**

### 7.1.2.3. Instalación de setuptools

Setuptools es una colección de programas para mejorar las herramientas de distribución de Python, que permiten crear y distribuir paquetes de Python, incluyendo dependencias para su instalación. Permite el acceso inmediato e instalación transparente de miles de paquetes, automáticamente encuentra y baja de internet las dependencias, para instalarlas o actualizarlas todo esto mediante la herramienta easyinstall, además permite crear Python Eggs, que son paquetes de Python empaquetados en un sólo archivo para su distribución e incluye archivos de configuración y todos los archivos que forman parte del directorio de trabajo, sin necesidad de listarlos individualmente o crear archivos de manifiesto.

Su instalación es muy sencilla como se muestra a continuación:

1. Ejecutar en consola lo siguiente:

```
wget http://peak.telecommunity.com/dist/ez_setup.py | sudo python
```

2. Confirmar que se realizó la instalación ejecutando:

```
python
>>> import setuptools
>>> setuptools.__version__
'0.6c9'
```

#### 7.1.2.4. Instalación de Gestor de Base de Datos PostgreSQL

PostgreSQL permitió el manejo de los datos del Sistema de Gestión Académica para obtener la información requerida en cada uno de los métodos. Para la instalación de PostgreSQL realizar lo siguiente:

1. Ejecutar en consola el siguiente comando:

```
sudo apt-get install postgresql postgresql-client
sudo apt-get install postgresql-common pgadmin3
```

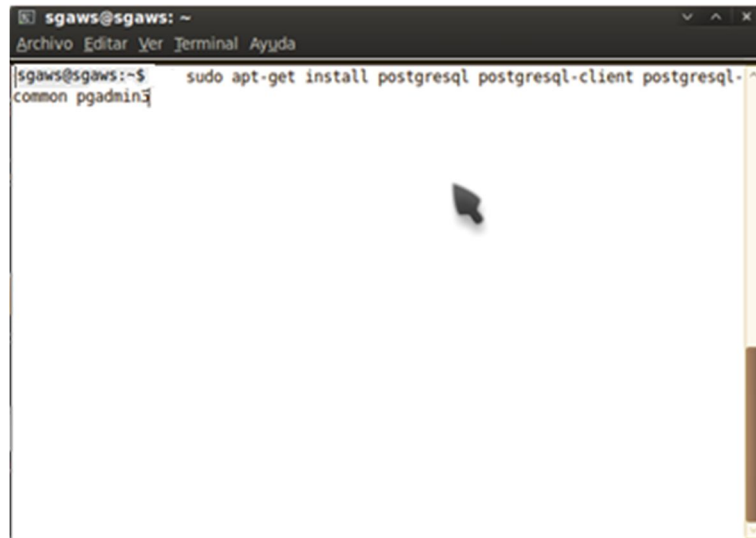


Fig.56: Instalación de postgresql

#### 7.1.2.5. Instalación la herramienta virtualenv

Virtualenv utilizada para crear un entorno aislado de python, para instalar paquetes sin modificar el python instalado en el sistema, sino que solo en el entorno. Para su instalación se realizó lo siguiente:

1. Ejecutar en consola el siguiente comando:

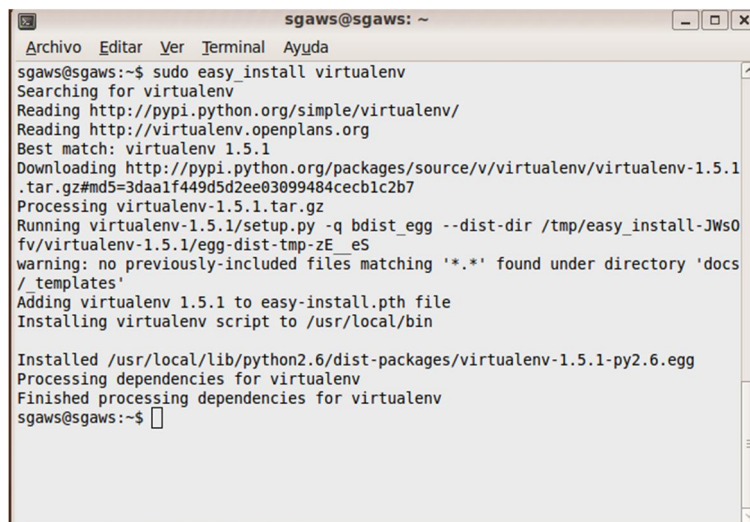
```
sudo easy_install virtualenv
```



```
sgaws@sgaws: ~  
Archivo Editar Ver Terminal Ayuda  
sgaws@sgaws:~$ sudo easy_install virtualenv
```

**Fig.57: Instalación de virtualenv**

2. Esperar a que finalice la instalación:



```
sgaws@sgaws: ~  
Archivo Editar Ver Terminal Ayuda  
sgaws@sgaws:~$ sudo easy_install virtualenv  
Searching for virtualenv  
Reading http://pypi.python.org/simple/virtualenv/  
Reading http://virtualenv.openplans.org  
Best match: virtualenv 1.5.1  
Downloading http://pypi.python.org/packages/source/v/virtualenv/virtualenv-1.5.1  
.tar.gz#md5=3daa1f449d5d2ee03099484cecb1c2b7  
Processing virtualenv-1.5.1.tar.gz  
Running virtualenv-1.5.1/setup.py -q bdist_egg --dist-dir /tmp/easy_install-Jws0  
fv/virtualenv-1.5.1/egg-dist-tmp-zE_es  
warning: no previously-included files matching '*.*' found under directory 'docs  
' templates'  
Adding virtualenv 1.5.1 to easy-install.pth file  
Installing virtualenv script to /usr/local/bin  
  
Installed /usr/local/lib/python2.6/dist-packages/virtualenv-1.5.1-py2.6.egg  
Processing dependencies for virtualenv  
Finished processing dependencies for virtualenv  
sgaws@sgaws:~$
```

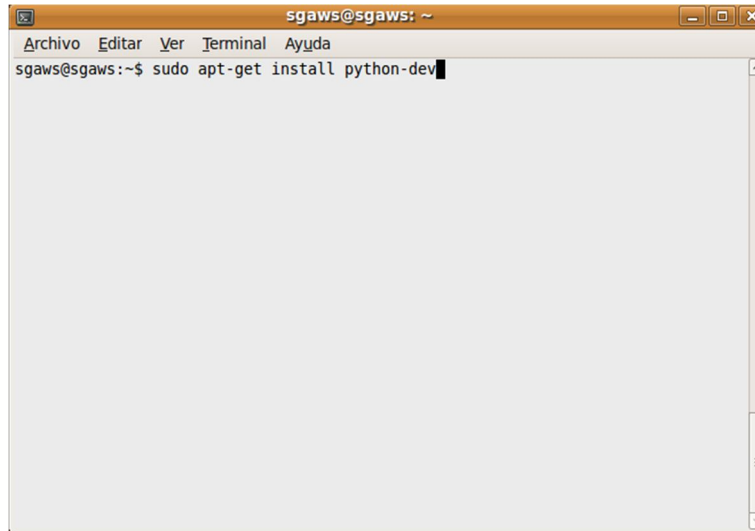
**Fig.58: Finalización de instalación de virtualenv**

#### 7.1.2.6. Instalar dependencias de Python

Versiones derivadas de Debian requiere la instalación de las siguientes dependencias `python-dev` y `build-essential`, el proceso fue el siguiente:

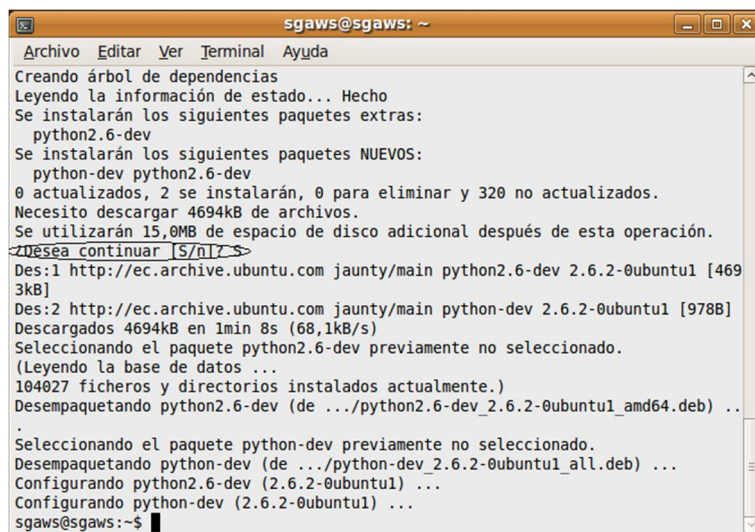
1. Para instalar `python-dev` ejecutar en consola el siguiente comando:

```
sudo apt-get install python-dev
```



**Fig.59: Instalación de python-dev**

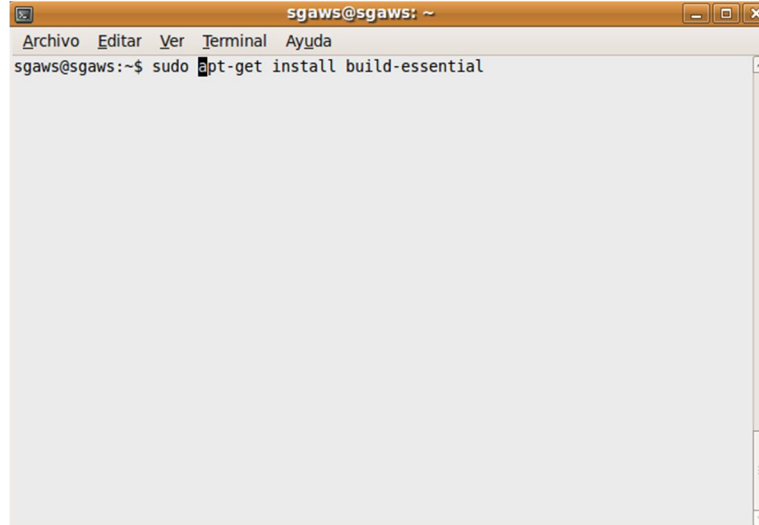
2. Confirmar la instalación y esperar a que esta finalice:



**Fig.60: Finalización de instalación de virtualenv**

3. Instalar build-essential ejecutando el siguiente comando:

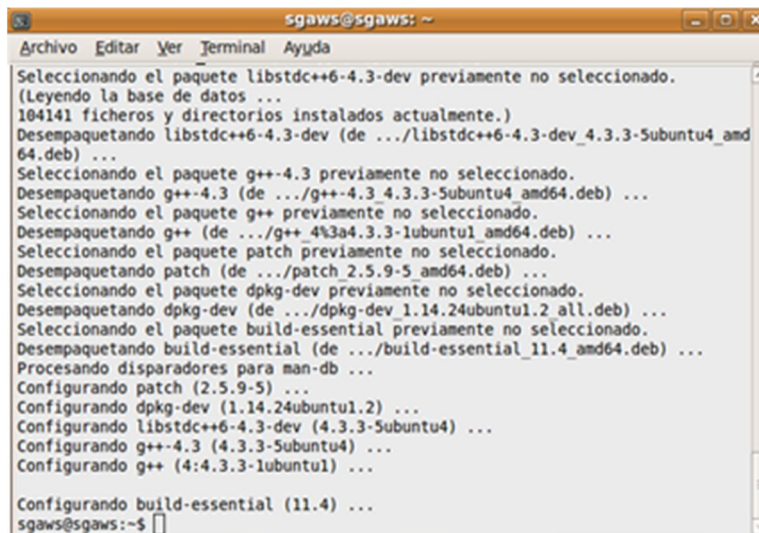
```
sudo apt-get install build-essential
```



```
sgaws@sgaws: ~  
Archivo Editar Ver Terminal Ayuda  
sgaws@sgaws:~$ sudo apt-get install build-essential
```

**Fig.61: Instalación de build-essential**

4. Confirmar la instalación y esperar a que esta finalice:



```
sgaws@sgaws: ~  
Archivo Editar Ver Terminal Ayuda  
Seleccionando el paquete libstdc++6-4.3-dev previamente no seleccionado.  
(Leyendo la base de datos ...  
104141 ficheros y directorios instalados actualmente.)  
Desempaquetando libstdc++6-4.3-dev (de ../libstdc++6-4.3-dev_4.3.3-Subuntu4_amd64.deb) ...  
Seleccionando el paquete g++-4.3 previamente no seleccionado.  
Desempaquetando g++-4.3 (de ../g++-4.3_4.3.3-Subuntu4_amd64.deb) ...  
Seleccionando el paquete g++ previamente no seleccionado.  
Desempaquetando g++ (de ../g++_4%3a4.3.3-lubuntu1_amd64.deb) ...  
Seleccionando el paquete patch previamente no seleccionado.  
Desempaquetando patch (de ../patch_2.5.9-5_amd64.deb) ...  
Seleccionando el paquete dpkg-dev previamente no seleccionado.  
Desempaquetando dpkg-dev (de ../dpkg-dev_1.14.24ubuntu1.2_all.deb) ...  
Seleccionando el paquete build-essential previamente no seleccionado.  
Desempaquetando build-essential (de ../build-essential_11.4_amd64.deb) ...  
Procesando disparadores para man-db ...  
Configurando patch (2.5.9-5) ...  
Configurando dpkg-dev (1.14.24ubuntu1.2) ...  
Configurando libstdc++6-4.3-dev (4.3.3-Subuntu4) ...  
Configurando g++-4.3 (4.3.3-Subuntu4) ...  
Configurando g++ (4:4.3.3-lubuntu1) ...  
Configurando build-essential (11.4) ...  
sgaws@sgaws:~$
```

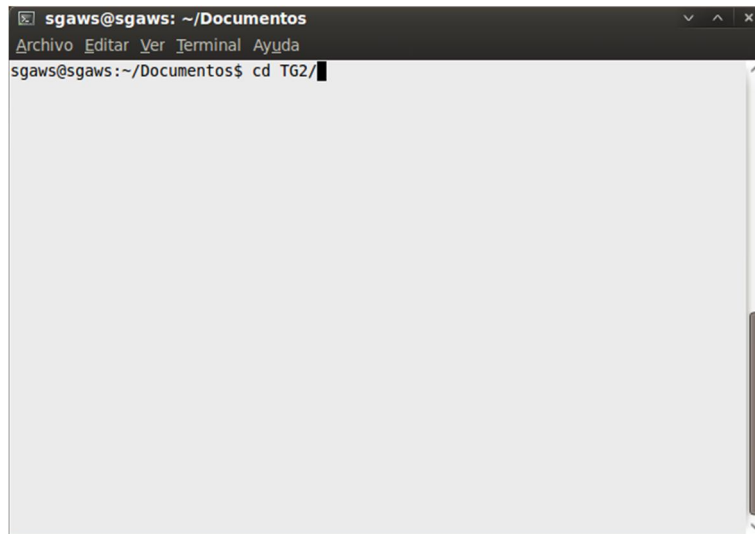
**Fig.62: Finalización de instalación de build-essential**

### 7.1.2.7. Instalación de framework Turbogears2.0

Para la instalación de Turbogears2.0 fue necesario que se haya instalado previamente: Python, setuptools, bases de datos, dependencias de python y virtualenv. Los pasos para la instalación realizados fueron los siguientes:

1. Ubicarse en el directorio donde se va a crear el entorno virtual para la instalación de TurboGears2.0 ejecutando:



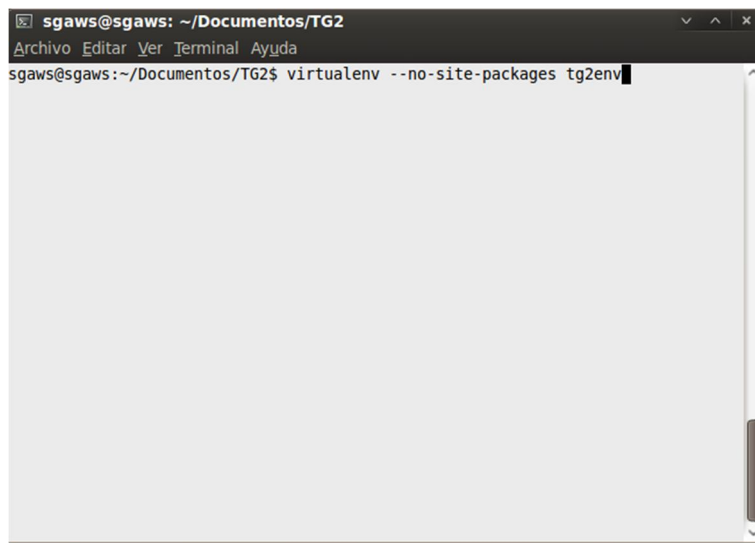


```
sgaws@sgaws: ~/Documentos
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~/Documentos$ cd TG2/
```

**Fig.63: Ubicación en el directorio TG2**

2. Crear un entorno virtual denominado *tg2env* ejecutando el siguiente comando:

```
virtualenv --no-site-packages tg2env
```

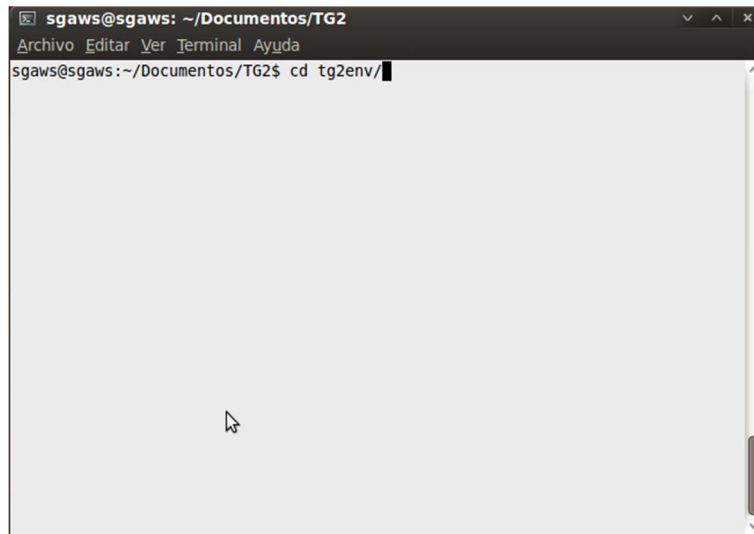


```
sgaws@sgaws: ~/Documentos/TG2
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~/Documentos/TG2$ virtualenv --no-site-packages tg2env
```

**Fig.64: Creación del entorno virtual tg2env**

3. Ubicarse en el directorio en que se creó el entorno virtual y activarlo ejecutando el siguiente comando:

```
cd tg2env/
source bin/activate
```



```
sgaws@sgaws: ~/Documentos/TG2
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~/Documentos/TG2$ cd tg2env/
```

**Fig.65: Moverse en el directorio tg2env**

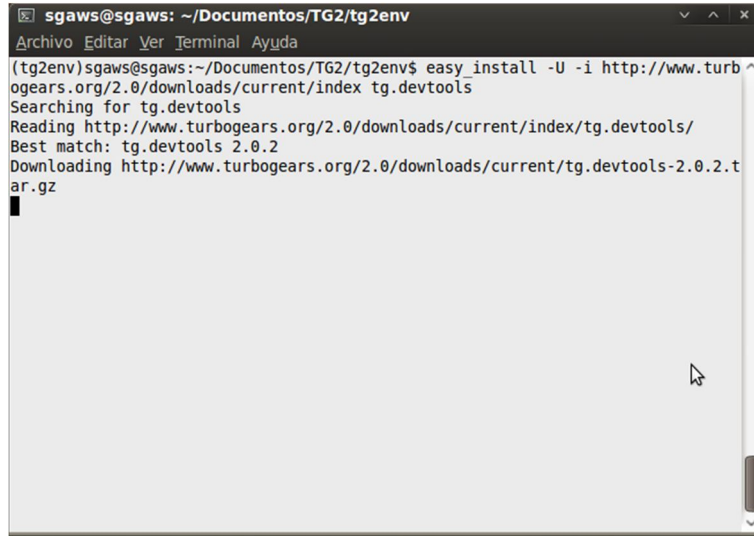


```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~/Documentos/TG2/tg2env$ source bin/activate
(tg2env) sgaws@sgaws:~/Documentos/TG2/tg2env$
```

**Fig.66: Activación del entorno virtual tg2env**

4. Instalar Turbogears2 ejecutando lo siguiente en el entorno virtual creado:

```
(tg2env)$ easy_install -U -i http://www.turbogears.org/2.0/downloads/current/index tg.devtools
```



```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda
(tg2env)sgaws@sgaws:~/Documentos/TG2/tg2env$ easy_install -U -i http://www.turbogears.org/2.0/downloads/current/index tg.devtools
Searching for tg.devtools
Reading http://www.turbogears.org/2.0/downloads/current/index/tg.devtools/
Best match: tg.devtools 2.0.2
Downloading http://www.turbogears.org/2.0/downloads/current/tg.devtools-2.0.2.tar.gz
```

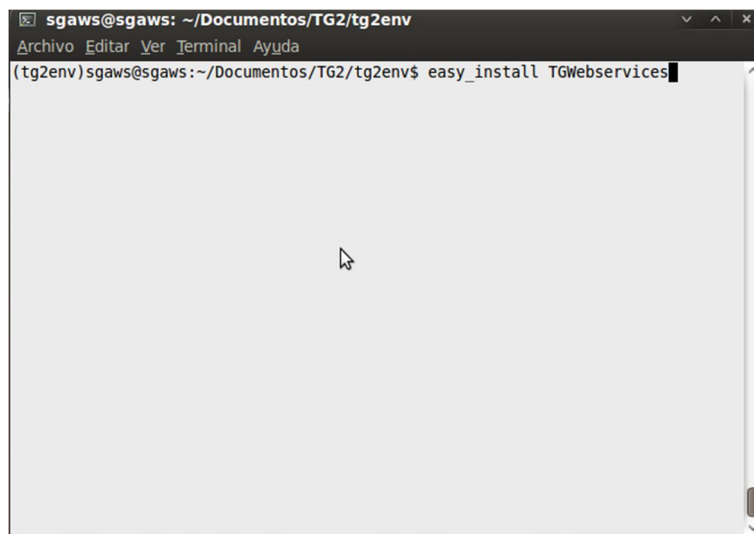
**Fig.67: Instalación de Turbogears2.0**

#### 7.1.2.8. Instalación del paquete TGWebServices 2.0.0a2

El paquete **TGWebServices 2.0.0a2** fue utilizado para la construcción de la biblioteca de métodos debido a que éste sirve para el desarrollo de servicios web. Para instalarlo se realizó el siguiente procedimiento:

1. Ejecutar el siguiente comando dentro del entorno virtual *tg2env*:

```
(tg2env)$ easy_installTGWebServices
```



```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda
(tg2env)sgaws@sgaws:~/Documentos/TG2/tg2env$ easy_install TGWebServices
```

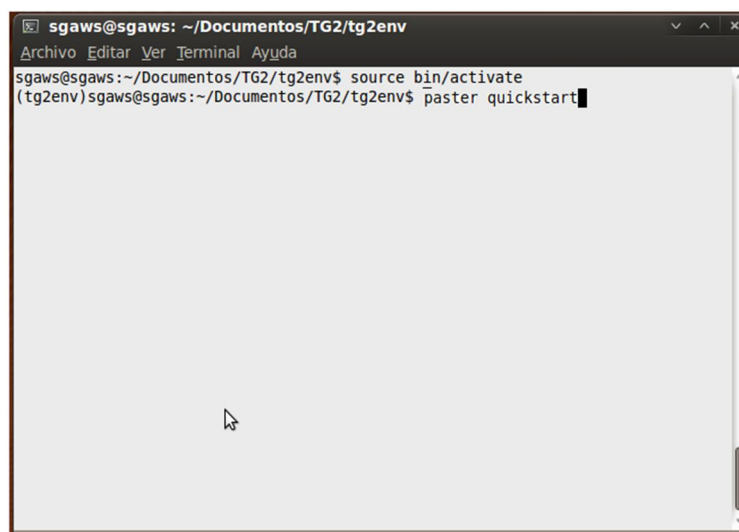
**Fig.68: Instalación de TGWebServices**

### 7.1.2.9. Creación del Proyecto en Turbogears2.0

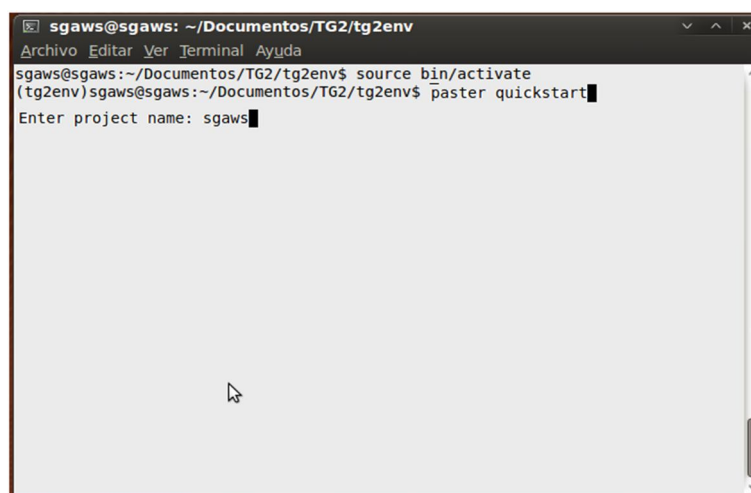
Una vez instaladas las herramientas necesarias dentro del entorno virtual, se realizó la creación del proyecto en Turbogears2.0 de la siguiente manera:

1. Ejecutar el siguiente comando para la creación de un proyecto:

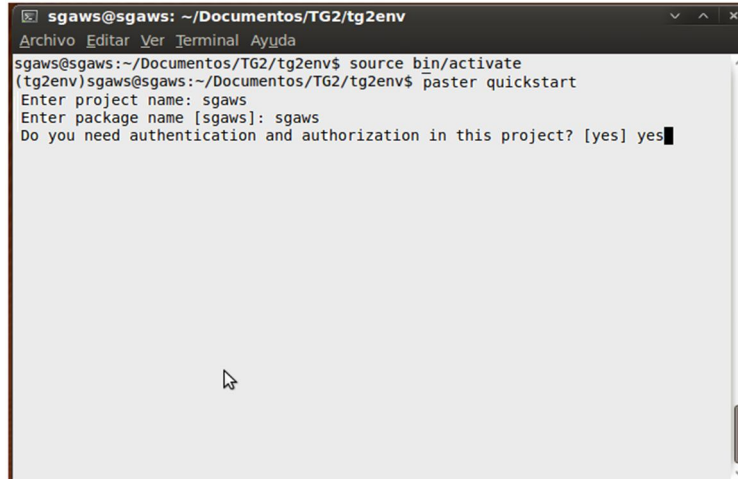
```
(tg2env)$ paster quickstart
```



**Fig.69: Crear el proyecto en Turbogears2.0**



**Fig.70: Especificación del nombre del proyecto sgaws**

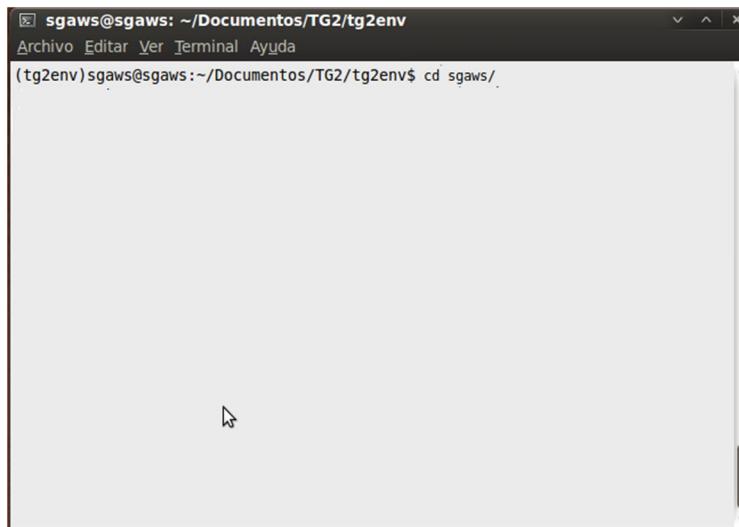


```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~/Documentos/TG2/tg2env$ source bin/activate
(tg2env)sgaws@sgaws:~/Documentos/TG2/tg2env$ paster quickstart
Enter project name: sgaws
Enter package name [sgaws]: sgaws
Do you need authentication and authorization in this project? [yes] yes
```

**Fig.71: Especificación de parámetros para la creación del proyecto**

2. Ubicarse al directorio del proyecto como se muestra en la siguiente figura:

```
cd sgaws/
```

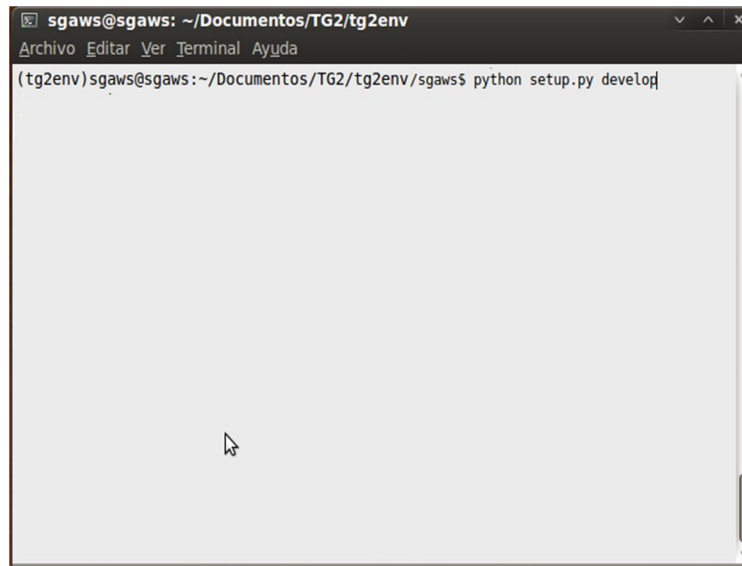


```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda
(tg2env)sgaws@sgaws:~/Documentos/TG2/tg2env$ cd sgaws/
```

**Fig.72: Mover al directorio del proyecto**

3. Ejecutar el siguiente comando para indicar que el proyecto debe trabajar con los paquetes instalados en el entorno virtual:

```
python setup.py develop
```

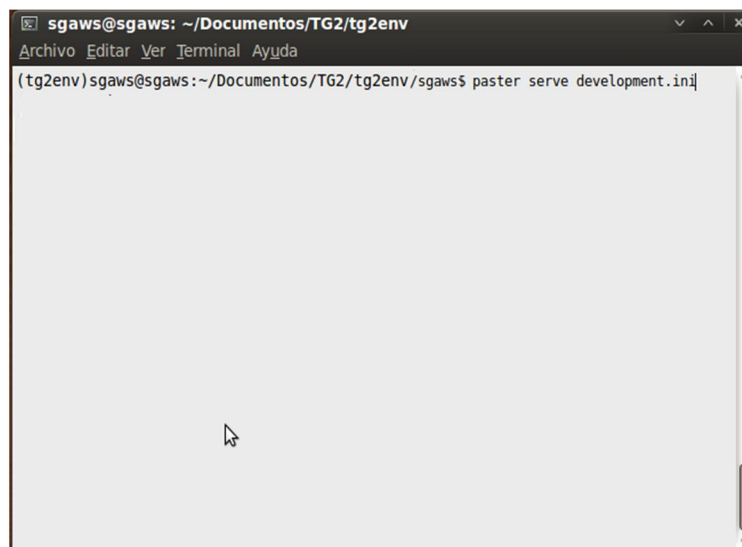


```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda
(tg2env)sgaws@sgaws:~/Documentos/TG2/tg2env/sgaws$ python setup.py develop
```

**Fig.73: Activación de los paquetes de Turboears**


4. Levantar el proyecto, ejecutando el siguiente comando:

```
paster serve development.ini
```



```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda
(tg2env)sgaws@sgaws:~/Documentos/TG2/tg2env/sgaws$ paster serve development.ini
```

**Fig.74: Levantar el proyecto creado**



```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda

(tg2env) sgaws@sgaws:~/Documentos/TG2/tg2env/sgaws$ paster serve development.ini
/home/sgaws/Documentos/TG2/tg2env/lib/python2.6/site-packages/PEAK_Rules-0.5a1.dev_r
2582-py2.6.egg/peak/rules/indexing.py:220: DeprecationWarning: object.__new__() takes
no parameters
Starting server in PID 2326.
serving on http://127.0.0.1:8080
```

Fig.75: Proyecto levantado en consola

5. Verificar en el navegador la creación correcta del proyecto, indicando en el mismo la dirección **http://localhost:8080** para que se levanta localmente como se muestra en la siguiente figura:

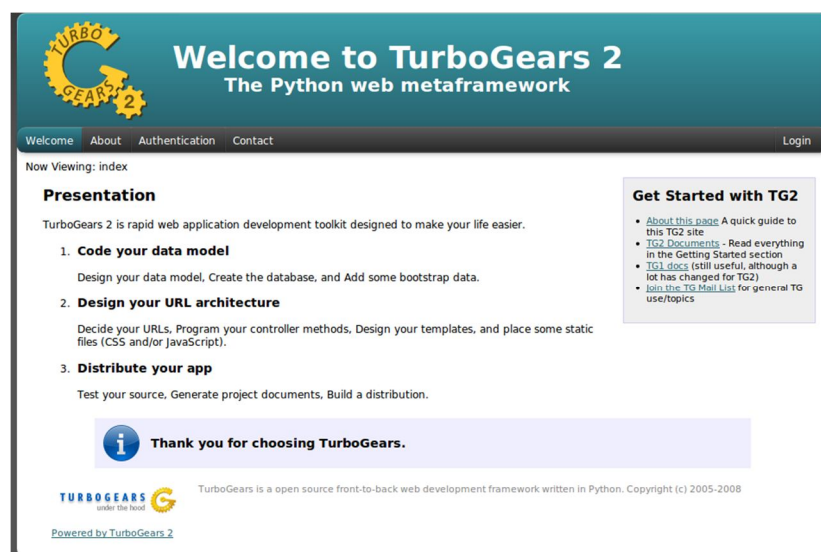


Fig.76: Visualización gráfica del proyecto levantado

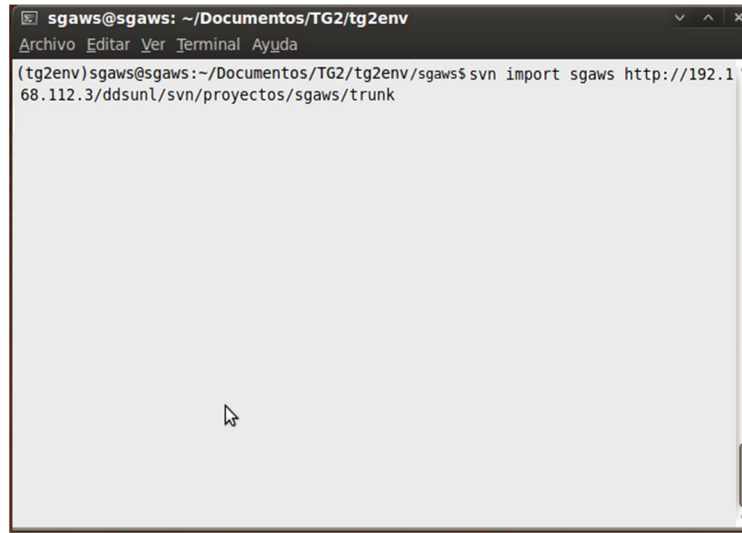
#### 7.1.2.10. Agregación del proyecto al repositorio de la Unidad de Desarrollo de software de la Universidad Nacional de Loja

Luego de creado el proyecto fue subido al repositorio de la Unidad de Desarrollo de software de la Universidad Nacional de Loja con la ayuda del sistema de control de

versiones subversión, para llevar un control de los cambios realizados al mismo y un respaldo de la información. El proceso que se llevó a cabo es el siguiente:

1. Importar el proyecto al repositorio como se muestra a continuación:

```
svn import sgaws http://192.168.112.3/svn/proyectos/sgaws/trunk
```



**Fig.77: Importación del proyecto al repositorio**

2. Crear un archivo denominado **.svnignore** con las extensiones de los archivos que se desea que no se agreguen al repositorio como se muestra a continuación:

```
*.pyc  
*.log  
*.sqlite  
*~  
*.swp  
development.ini  
*.egg  
,
```

**Fig.78: Contenido del archivo .svnignore**

3. Agregar el archivo **.svnignore** al repositorio en nuestro proyecto ejecutando el siguiente comando:

```
svn add .svnignore
```



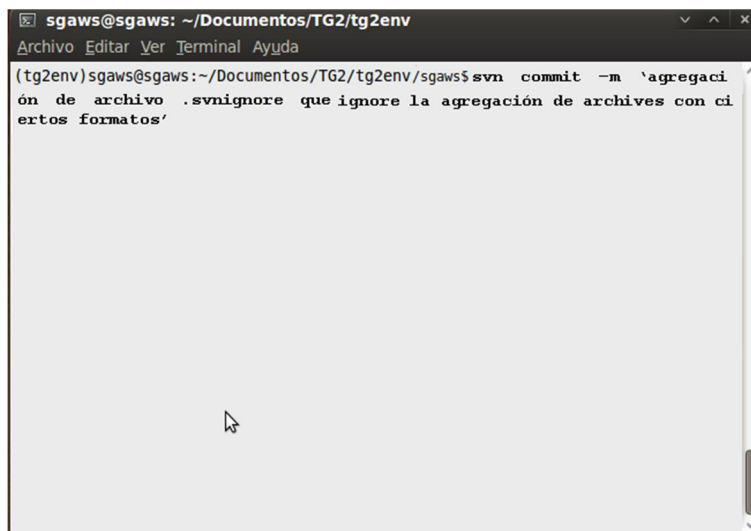


```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda
(tg2env) sgaws@sgaws:~/Documentos/TG2/tg2env/sgaws$ svn add .svnignore
```

**Fig.79: Agregación del archivo .svnignore al repositorio**

4. Finalmente subir los cambios realizados localmente en el proyecto al repositorio, como se muestra a continuación:

```
svn commit -m 'agregación de archivo .svnignore que ignore la agregación de archivos con ciertos formatos'
```



```
sgaws@sgaws: ~/Documentos/TG2/tg2env
Archivo Editar Ver Terminal Ayuda
(tg2env) sgaws@sgaws:~/Documentos/TG2/tg2env/sgaws$ svn commit -m 'agregación de archivo .svnignore que ignore la agregación de archivos con ciertos formatos'
```

**Fig.80: Ejemplo de commit en svn**

### 7.1.3. Migración de Clases y Métodos del Modelo del Sistema de Gestión Académica al SGAWebservices con Turbogears 2.0

Luego de creado el proyecto y subido al repositorio y en vista de que en el análisis realizado al model.py del Sistema de Gestión Académico se identificaron las clases y métodos objetos a migración en base a los servicios que presta la biblioteca de métodos SGAWebservices, se llevó a efecto esta actividad creándose el modelo.py para la aplicación. Este proceso fue necesario en vista a que el sistema académico está desarrollado en Turbogears1 y nuestro proyecto estaba encaminado al desarrollo en una versión más actualizada Turbogears2.0 el mismo que ofrece mejores particularidades y paquetes de desarrollo, sin dejar a un lado la perspectiva de que las tecnologías avanzan cada día. A continuación se presentan las clases y métodos migrados:

#### Clases y Métodos

- Area
- Arancel
- ArancelAcademico
- CarreraPrograma
- Componente
- ComponenteAcademico
- DatosPersonales
- Docente:
  - get\_carreras()
- EstadoEstudiante
- EstadoMatricula
- Estudiante:
  - \_get\_matriculas\_estudiante()
- Expediente:
  - get\_acreditacion()
  - get\_asistencias()
- Group
- HorarioSemana:
  - \_getdia()



- get\_fecha\_desde\_lunes()
- set\_fecha\_fin()
- get\_fecha\_fin()
- get\_horas\_dictadas()
- get\_horas\_programadas()
- get\_horas\_semana()
- get\_horas\_dia()
- get\_dias\_semana()
- get\_dias\_semana\_fecha()
- get\_periodos\_clase\_semana()
- Jornada
- Matricula:
  - get\_horas\_asistidas()
  - get\_porcentaje\_asistido()
- Modulo
- Nivel
- OfertaAcademica
- OfertaComponente
- Papeleta
- Paralelo
- Parametro
- PeriodoLectivo
- PeriodoClase
- PeriodoMatricula
- PeriodoNoLaborado
- PlanEstudio:
  - calcular()
  - promover()
  - get\_tabla\_acreditacion()
  - get\_total\_horas, update\_plan()
  - actualizar()
- RegistroAcreditacion
- RegistroAsistencia:

- delete()
- TipoPlanEstudioEx
- TipoPlanEstudio:
  - calcular()
  - promover()
- TipoPlanEstudioNone
- TipoPlanEstudioConTaller:
  - \_comprobaciones()
  - get\_tabla\_creditacion, calcular()
- TipoPlanEstudioConTallerDetalle
- TipoPlanEstudioSumatoria:
  - \_comprobaciones()
  - get\_tabla\_acreditacion()
  - calcular ()
- TipoPlanEstudioPorPuntos:
  - \_comprobaciones()
  - get\_tabla\_acreditacion()
  - calcular()
- TipoPlanEstudioPromedio:
  - \_comprobaciones()
  - get\_tabla\_acreditacion()
  - calcular()
- TipoPlanEstudioAdmisiones
- Unidad
- User

#### **7.1.4. Construcción de los Servicios de la Biblioteca de Métodos de Recuperación de Información**

Luego de realizada la migración de las entidades necesaria para el desarrollo de la aplicación y bien definidos los métodos que se implementaron, se procedió a clasificarlos dependiendo de la salida o información de retorno es decir lo que se debía obtener como respuesta al ser invocada cada uno de las funciones y además de acuerdo al servicio que este presta. Llegando así a la siguiente clasificación:

**Información Académica:**

En esta categoría se seleccionaron y se clasificaron todas las funciones que retornan datos académicos de la institución universitaria, docentes y alumnos pertenecientes a la misma.

- sgaws\_periodos\_lectivos
- sgaws\_ofertas\_academicas
- sgaws\_fechas\_matriculasa
- sgaws\_estadoestudiantes\_paralelo
- sgaws\_plan\_estudio
- sgaws\_carga\_horaria\_docente
- sgaws\_notas\_estudiante
- sgaws\_carreras\_estudiante
- sgaws\_reporte\_matricula

**Información Institucional:**

En esta categoría se han colocado las funciones cuya salida nos retornan datos institucionales de la Universidad Nacional de Loja:

- sgaws\_lista\_areas
- sgaws\_datos\_area
- sgaws\_carreras\_area
- sgaws\_datos\_carreras
- sgaws\_modulos\_carrera
- sgaws\_paralelos\_carrera

**Validación:**

Categoría en la que se clasifico los métodos que permiten validar un docente o un estudiante para verificar si existe dentro de la base de datos del Sistema de Gestión Académica.

- sgaws\_validar\_docente
- sgaws\_validar\_estudiante

**Información Personal:**

En esta categoría se han ubicado los métodos que devuelven información personal de un docente o un estudiante según la cedula ingresada.

- sgaws\_datos\_docente
- sgaws\_datos\_estudiante
- sgaws\_datos\_usuario

**Información Estadística:**

Finalmente en esta categoría se encuentran las funciones de las cuales se obtienen como resultados los datos estadísticos de cada uno de las ofertas académicas que se han promocionado durante la existencia del Sistema de Gestión Académica.

- sgaws\_nmatriculados\_oferta
- sgaws\_nmatriculados\_area
- sgaws\_nmatriculados\_carrera
- sgaws\_nmatriculados\_modulo
- sgaws\_nmatriculados\_paralelo
- sgaws\_naprobados\_oferta
- sgaws\_naprobados\_area
- sgaws\_naprobados\_carrera
- sgaws\_naprobados\_modulo
- sgaws\_naprobados\_paralelo
- sgaws\_nreprobados\_oferta
- sgaws\_nreprobados\_area
- sgaws\_nreprobados\_carrera
- sgaws\_nreprobados\_modulo
- sgaws\_nreprobados\_paralelo

Una vez llevada a cabo la clasificación de las funciones, se realizó la codificación de cada uno de los métodos o servicios, los mismos que fueron probados a nivel de desarrolladores con la finalidad de comprobar que funcionaban adecuadamente y poder subir el código al repositorio de proyectos.

#### 7.1.4.1. Verificación del funcionamiento de los métodos

Para comprobar el correcto funcionamiento de los métodos fue necesario el desarrollo de un cliente web en cualquier lenguaje de programación esto a través de consola. Otra forma de realizar la comprobación fue de forma gráfica por medio de un navegador web, el proceso llevado a cabo fue el siguiente:

**a) Revisar el documento wsdl para conocer el funcionamiento de cada método del SGAWebservices.**

Lo primero fue escoger uno de los métodos objeto de comprobación para revisar el documento **api.wsdl** de acuerdo a la categoría en que se encuentra el mismo, debido a que es aquí donde se indican los parámetros de entradas la función que se utilice, así como también lo que nos retorna esta. Este documento de formato XML se muestra digitando en la barra de direcciones del navegador web (Mozilla Firefox) la siguiente dirección según el método y categoría a la que pertenece el mismo en este caso:

```
http://localhost:8080/sgaws/wsinstitucional/soap/api.wsdl
```

**b) Enviar correctamente los parámetros de entrada para cada método.**

Una vez revisado el documento **api.wsdl** y después de haber verificado lo que requiere cada método para su funcionamiento se procede a digitar en la barra de navegación la dirección con el nombre de la función que se desea probar que está funcionando correctamente, si los parámetros que se deben enviar son correctos de acuerdo al método invocado se obtendrán los resultados esperados como se muestra en la siguiente figura:

```
http://localhost:8080/sgaws/wsinstitucional/sgaws_modulos_carrera?id_oferta=12&id_carrera=34
```

```
Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

-<result>
  ["Oferta", "Pregrado Marzo 2009 - Julio 2009", "Ingeniería en Geología Ambiental y Ordenamiento Territorial", [[66, "EL ENTORNO GEOLÓGICO AMBIENTAL", "4"], [68, "CARACTERIZACIÓN DE CONTAMINANTES", "6"], [70, "PREVENCIÓN Y EVALUACIÓN DE RIESGOS NATURALES", "8"], [72, "GESTIÓN AMBIENTAL", "10"], [64, "LAS PROBLEMÁTICAS DE LA REALIDAD DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES, SUS REFERENTES TEÓRICOS CONCEPTUALES DENTRO DEL CAMPO FUNDAMENTAL DE LA CIENCIA Y SU INCIDENCIA EN LA FORMACIÓN INTEGRAL Y PROFESIONAL", "2"]]]
</result>
```

**Fig.81: Ejemplo de resultados de un método del SGAWebsservices**

Dado el caso que se cometan algunos errores al momento de enviar los parámetros del método, se mostrarán varios mensajes de acuerdo al error cometido como se muestra en las siguientes figuras:

1. Si no se envían todos los datos:

```
Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

-<result>
  ["_error", "Por favor verifique la conexión o los datos que han sido ingresados"]
</result>
```

**Fig.82: Ejemplo de error por no enviar todos los datos de un método**

2. Para el método que se ha tomado como ejemplo, cuando se envía mal el id\_carrera:

```
Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

-<result>
  ["_error", "El id_carrera ingresado no pertenece a ninguna CarreraPrograma"]
</result>
```

**Fig.83: Ejemplo de error por enviar mal el id\_carrera**



3. Cuando la carrera no pertenece a la oferta académica enviada:

```
Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.
```

```
-<result>  
  ["_error", "Esta Carrera no esta disponible en esta Oferta Academica"]  
</result>
```

**Fig.84: Ejemplo de error cuando la carrera no pertenece a la oferta académica**

4. Cuando el id de la oferta no es el correcto:

```
Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.
```

```
-<result>  
  ["_error", "El id_oferta ingresado no pertenece a ninguna Oferta Academica"]  
</result>
```

**Fig.85: Ejemplo de error por enviar mal el id de la oferta**

#### 7.1.4.2. Diseño y construcción de los templates de documentación del SGAWebservices

En los templates de documentación se especifica en varias páginas el funcionamiento, parámetros de entrada y lo que retornará cada método del webservices de acuerdo a las categorías que se han establecido, la publicación de los documentos wsdl, la especificación del protocolo SOAP y dos ejemplos de clientes webservices. Se estableció un estilo propio con CSS, se creó una página donde se indica brevemente que hace y para que sirve el SGAWebservices y el login donde el cliente podrá acceder.



**Fig.86: Imagen del template de inicio del SGAWebservices**

Se crearon varios templates con la documentación y un menú para acceder a cada uno de ellos como se detalla a continuación:

- a. La Presentación del SGAWebservices
- b. Los Servicios
  - Listado de Categorías en las que fueron divididos los métodos
  - La Descripción de cada Categoría: Información Académica, Información Institucional, Información Personal, Validación e Información Estadística.
  - El Documento WSDL
- c. Cómo usar el SGAWebservices

Como se muestra en la siguiente figura:



Fig.87: Imagen del menú del SGAWebservices

#### a. Presentación del SGAWebservices

En la página de presentación se indica que es y que hace un servicio web (web services) y las ventajas que ofrece hacer uso de ellos:



Fig.88: Imagen del template de Presentación del SGAWebservices

## b. Los Servicios

En el menú de Servicios se indica la **Lista de Categorías** en la que fueron divididos los métodos del SGAWebservices como se muestra en las siguientes figuras:



**SERVICIOS**

El Sistema de Gestión Académico de la Universidad Nacional de Loja, posee un sin número de información que puede ser aprovechada por aplicaciones que tengan fines académicos, por esta razón la biblioteca de métodos o servicios **SGAWebservices** ofrece distintas funciones, utilizando la tecnología de Web Services. Los servicios están agrupados por categorías de acuerdo a la información que retornan.

Categorías:

**Académica**

Métodos que retornan información académica

- sgaws\_periodos\_lectivos
- sgaws\_ofertas\_academicas
- sgaws\_fechas\_matriculaoa
- sgaws\_estadoestudiantes\_paralelo
- sgaws\_plan\_estudio
- sgaws\_carga\_horaria\_docente
- sgaws\_notas\_estudiante
- sgaws\_carreras\_estudiante
- sgaws\_reporte\_matricula

**Fig.89: Listado de métodos de categoría de Información Académica**

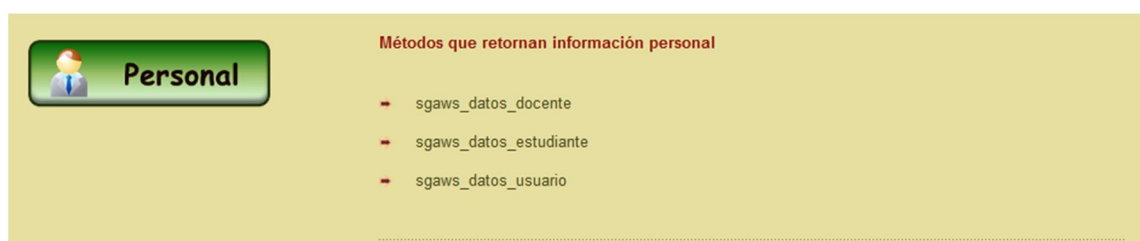


**Institucional**

Métodos que retornan información de la institución

- sgaws\_lista\_areas
- sgaws\_datos\_area
- sgaws\_carreras\_area
- sgaws\_datos\_carreras
- sgaws\_modulos\_carrera
- sgaws\_paralelos\_carrera

**Fig.90: Listado de métodos de categoría de Información Institucional**



**Personal**

Métodos que retornan información personal

- sgaws\_datos\_docente
- sgaws\_datos\_estudiante
- sgaws\_datos\_usuario

**Fig.91: Listado de métodos de categoría de Información Personal**



**Fig.92: Listado de métodos de categoría de Validación**



**Fig.93: Listado de métodos de categoría de Información Estadística**

Además se muestra la **Descripción de Servicios** que proporciona el SGAWebsservices, es decir la documentación de cada una de las categorías donde se listan cada uno de los métodos y conjuntamente se describe la forma de usarlos, se indica los parámetros que se deben enviar para obtener la información deseada y los datos que retornan.


En las siguientes figuras se muestra cada categoría y de uno de sus métodos para ejemplificar la forma como están descritos todos los métodos de las categorías.

## Descripción de servicios de Información Académica:

### Información Académica

En esta categoría se encontrarán los métodos o servicios relacionados a la información académica.  
Como datos de:

- Estudiantes
- Docentes



Listado de métodos de esta categoría:

Para observar la descripción de cada servicio pulsar sobre el:

sgaws_periodos_lectivos:	Retorna una lista de periodos lectivos
sgaws_ofertas_academicas:	Retorna una lista de ofertas académicas en un periodo lectivo
sgaws_fechas_matriculaoa:	Retorna las fechas de matrículas ordinaria, extraordinaria y especial de una oferta académica
sgaws_estadoestudiantes_paralelo:	Retorna la lista de estudiantes de un paralelo con el estado de matrícula ( aprobada, reprobado, matriculada), de acuerdo a la oferta académica, módulo, carrera y paralelo
sgaws_plan_estudio:	Retorna el plan de estudio asignado o un paralelo en una oferta académica
sgaws_carga_horaria_docente:	Retorna la carga horaria asignada a un docente en una oferta académica
sgaws_notas_estudiante:	Retorna las notas de un estudiante en cada unidad, la carrera, módulo, paralelo al que pertenece y estado de la matrícula en la oferta académica
sgaws_carreras_estudiante:	Retorna los datos de un estudiante y las carreras en las que está matriculado
sgaws_reporte_matricula:	Retorna los datos de un estudiante, oferta académica, nota y porcentaje de asistencia y estado de la matrícula del estudiante en esa oferta

Tesis  
SGAWebServices UNL  
Daisy Ordóñez - Silvana Pacheco

**Fig.94: Descripción de la categoría de Información Académica**

sgaws\_ofertas\_academicas(id\_periodo):

Parametros que recibe el método:	
id_periodo	Número que identifica a un periodo lectivo

Una vez introducidos los parámetros de entrada y procesados, el servicio retorna una lista de ofertas académicas, de acuerdo al id de período lectivo ingresado

Datos que retorna:	
id oferta	Número que identifica una oferta académica
descripción	Denominación que tiene una oferta académica
fecha_inicio_clases	Fecha de inicio de clases en una oferta académica
fecha_fin_clases	Fecha de fin de clases en una oferta académica

Tesis  
SGAWebsservices UNL  
Daisy Ordóñez - Silvana Pacheco

Fig.95: Ejemplo de descripción de método de Información Académica

Descripción de servicios de Información Institucional:

Información Institucional

En esta categoría se encontrarán los métodos o servicios relacionados con la información institucional. Como datos de:

- Areas
- Carreras
- Módulos
- Paralelos

Listado de métodos de esta categoría:

Para observar la descripción de cada servicio pulsar sobre el:

sgaws_lista_areas	Retorna una lista de las áreas que conforman la universidad
sgaws_datos_area	Retorna los datos de las siglas del área ingresada
sgaws_carreras_area	Retorna las carreras que forman parte de una área
sgaws_datos_carreras	Retorna una lista de carreras con sus datos en una oferta académica
sgaws_modulos_carrera	Retorna los datos de todos los módulos de una carrera en una oferta académica
sgaws_paralelos_carrera	Retorna todos los paralelos de una carrera en una oferta académica

ws/datosarea

Tesis  
SGAWebsservices UNL  
Daisy Ordóñez - Silvana Pacheco

Fig.96: Descripción de la categoría de Información Institucional

sgaws\_datos\_area(siglas):

Parametros que recibe el método:	
siglas	Siglas que representan el área de la que se desea obtener los datos

Una vez introducidos los parámetros de entrada y procesados, el servicio retorna los datos del área, de acuerdo a la sigla ingresada:

Datos que retorna:	
siglas	Sigla del área ingresada
nombre	Denominación que tiene el área
secretario abogado	Nombre del Secretario Abogado del área
director	Nombre del Director del área

Tesis  
SGAWebsservices UNL  
Daisy Ordóñez - Silvana Pacheco

Fig.97: Ejemplo de descripción de método de Información Institucional

### Descripción de servicios de información personal:

Información Personal

En esta categoría se encontrarán los métodos o servicios relacionados con información personal de:

- Datos de docentes
- Datos de estudiantes

Listado de métodos de esta categoría:

Para observar la descripción de cada servicio pulsar sobre el:

sgaws_datos_docente:	Retorna los datos o información del docente, que corresponden a la cédula ingresada
sgaws_datos_estudiante:	Retorna los datos o información del estudiante, que corresponden a la cédula ingresada
sgaws_datos_usuario:	Retorna los datos del usuario ingresado e indica el tipo de usuario estudiante / docente o ambos que corresponden a la cédula ingresada

Fig.98: Descripción de la categoría de Información Personal



sgaws\_datos\_usuario(cedula):

Parametros que recibe el método:	
cedula	Número de cédula que identifica a un usuario

Una vez introducidos los parámetros de entrada y procesados, el servicio retorna una lista con los datos del usuario del cual se ha ingresado la cédula, sea este estudiante, docente o ambos:

Datos que retorna:	
tipo de usuario	Tipo de usuario sea docente, estudiante o ambos docente-estudiante
nombres	Nombres del usuario del que se ingreso la cédula
apellidos	Apellidos del usuario del que se ingreso la cédula
cedula	Número de cédula que identifica al usuario

Tesis  
SGAWebsservices UNL  
Daisy Ordóñez - Silvana Pacheco

Fig.99: Ejemplo de descripción de método de Información Personal

### Descripción de servicios de validación:

Validación

En esta categoría se encontrarán los métodos o servicios relacionados con la validación de:

- Docentes
- Estudiantes

Listado de métodos de esta categoría:

Para observar la descripción de cada servicio pulsar sobre el:

sgaws_validar_docente:	Retorna Verdadero o Falso si el docente esta en la base de datos del Sistema de Gestión Académica
sgaws_validar_estudiante:	Retorna Verdadero o Falso si el estudiante esta en la base de datos del Sistema de Gestión Académica

restudiante

Fig.100: Descripción de la categoría de Validación

sgaws\_validar\_estudiante(cedula, clave)

Parametros que recibe el método:	
cedula	Número de cédula que identifica a un estudiante
clave	Password del estudiante en el Sistema de Gestión Académica

Una vez introducidos los parámetros de entrada y procesados, el servicio retorna un valor de true o false si el estudiante se encuentra o no en el sistema

Datos que retorna:	
True	Valor que indica si el estudiante se encuentra en el sistema
False	Valor que indica si el estudiante no se encuentra en el sistema

Tesis  
SGAWebsservices UNL  
Daisy Ordóñez - Silvana Pacheco

Fig.101: Ejemplo de descripción de método de Validación

Descripción de servicios de información estadística:

Información Estadística

En esta categoría se encontrarán los métodos o servicios relacionados con información estadística.  
Como:

- Número de estudiantes matriculados
- Número de estudiantes aprobados
- Número de estudiantes reprobados

Listado de métodos de esta categoría:

Para observar la descripción de cada servicio pulsar sobre el:

sgaws_nmatriculados_oferta:	Devuelve el número de estudiantes matriculados en una oferta académica
sgaws_nmatriculados_area:	Devuelve el número de estudiantes matriculados en una area en una oferta académica
sgaws_nmatriculados_carrera:	Devuelve el número de estudiantes matriculados en una carrera en una oferta académica
sgaws_nmatriculados_modulo:	Devuelve el número de estudiantes matriculados en un módulo en una oferta académica
sgaws_nmatriculados_paralelo:	Devuelve el número de estudiantes matriculados en un paralelo en una oferta académica
sgaws_naprobados_oferta:	Devuelve el número de estudiantes aprobados en una oferta académica
sgaws_naprobados_area:	Devuelve el número de estudiantes aprobados en una area en una oferta académica
sgaws_naprobados_carrera:	Devuelve el número de estudiantes aprobados en una carrera en una oferta académica
sgaws_naprobados_modulo:	Devuelve el número de estudiantes aprobados en un módulo en una oferta académica
sgaws_naprobados_paralelo:	Devuelve el número de estudiantes aprobados en un paralelo en una oferta académica
sgaws_nreprobados_oferta:	Devuelve el número de estudiantes reprobados en una oferta académica
sgaws_nreprobados_area:	Devuelve el número de estudiantes reprobados en una area en una oferta académica
sgaws_nreprobados_carrera:	Devuelve el número de estudiantes reprobados en una carrera en una oferta académica
sgaws_nreprobados_modulo:	Devuelve el número de estudiantes reprobados en un módulo en una oferta académica
sgaws_nreprobados_paralelo:	Devuelve el número de estudiantes reprobados en un paralelo en una oferta académica

Fig.102: Descripción de la categoría de Información Estadística

sgaws\_nreprobados\_modulo(id\_oferta, id\_modulo):

Parametros que recibe el método:	
id_oferta	Número que identifica a una oferta académica
id_modulo	Número que identifica a un módulo

Resultados que se obtienen:

Luego de introducidos los parámetros de entrada y procesados el web service retorna una lista con la descripción de la oferta, el nombre de la carrera, el número del módulo y el número de estudiantes reprobados en ese módulo.

Tesis  
SGAWebservices UNL  
Daisy Ordóñez - Silvana Pacheco

**Fig.103: Ejemplo de descripción de método de Información Estadística**

Finalmente dentro del menú de Servicios se encuentran el **Documentos WSDL** donde se muestra una descripción de éste documento, que no es más que un catálogo donde se describen y se listan los servicios web del SGAWebservices, convirtiéndose en el manual de operaciones del mismo, además se creó un enlace al documento wsdl de cada categoría, en la siguiente figura se muestra este template:

Descripción del servicio WSDL

El WSDL es un formato XML donde se describen los web services, es la interfaz pública de los mismos. Indica la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Puede usar SOAP para hacer la llamada a una de las funciones listadas en el mismo.

El WSDL es el manual de operación del web service, porque indica cuales son las funciones que provee el Servicio web y los tipos de datos necesarios para la utilización del mismo. Para observar el wsdl de cada categoría del SGAWebservices seleccione la misma, además encontrará el wsdl de los métodos migrados.

WSDL - Información Académica    WSDL - Información Institucional    WSDL - Información Personal

WSDL - Validación    WSDL - Información Estadística    WSDL - Métodos Migrados

<s/wsacademica/soap/api.wsdl>

**Fig.104: Descripción de Documento WSDL**

Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

```
-<wsdl:definitions name="SGAWebServicesAcademica" targetNamespace="http://ws.unl.edu.ec/sgaws/wsacademica/soap/">
  -<wsdl:types>
    -<xsd:schema elementFormDefault="qualified" targetNamespace="http://ws.unl.edu.ec/sgaws/wsacademica/soap/types">
      -<xsd:element name="sgaws_carga_horaria_docente">
        -<xsd:complexType>
          -<xsd:sequence>
            <xsd:element name="id_oferta" type="xsd:string"/>
            <xsd:element name="cedula" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      :
    -<wsdl:message name="sgaws_carga_horaria_docenteRequest">
      <wsdl:part name="parameters" element="types:sgaws_carga_horaria_docente"/>
    </wsdl:message>
    -<wsdl:message name="sgaws_carga_horaria_docenteResponse">
      <wsdl:part name="parameters" element="types:sgaws_carga_horaria_docenteResponse"/>
    </wsdl:message>
    :
  -<wsdl:binding name="SGAWebServicesAcademica_Binding" type="tns:SGAWebServicesAcademica_PortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    -<wsdl:operation name="sgaws_carga_horaria_docente">
      <soap:operation soapAction="sgaws_carga_horaria_docente"/>
      -<wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      -<wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    :
  -<wsdl:service name="SGAWebServicesAcademica">
    <wsdl:documentation>WSDL File for SGAWebServicesAcademica</wsdl:documentation>
    -<wsdl:port binding="tns:SGAWebServicesAcademica_Binding" name="SGAWebServicesAcademica_PortType">
      <soap:address location="http://ws.unl.edu.ec/sgaws/wsacademica/soap/">
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

**Fig.105: Ejemplo de Documento WSDL**

#### d. Cómo usar el SGAWebServices

En la figura siguiente se describe el protocolo que se debe utilizar para el intercambio de información, además se presentan dos ejemplos de clientes web en Python y Php utilizados para la invocación del SGAWebServices, también existe la posibilidad de descargar el código de los clientes:

## SOAP

Para usar el SGAWebservices de la Universidad Nacional de Loja, lo primero que tiene que asegurarse es que el lenguaje de programación que utiliza para construir su aplicación cliente está preparado para enviar y recibir mensajes según el estándar SOAP. Incluso en el caso de que no tenga una librería específica, es imprescindible la posibilidad de enviar y recibir mensajes XML vía HTTP. Independientemente de cómo se haga la solicitud, las respuestas son en XML formateado con JSON.

SOAP es un protocolo elaborado para facilitar la llamada remota de funciones a través de Internet, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web.



## Ejemplos de Clientes

Ejemplos prácticos de clientes para llamar un método del SGAWebservices.

Ejemplo de cliente en PHP [click aquí](#)

```
cliente.php
1 <?php
2 #wsdl de la categoría de información institucional debido a que se invoca un método de
  esta
3
4 #Descomentar la siguiente línea si tiene una conexión directa a internet y no tiene proxy
5 $cliente = new SoapClient("http://ws.unl.edu.ec/sgaws/wsinstucional/soap/
  api.wsdl",array('login'=> "usuario", 'password' => "password"));
6
7 #Utilizar la siguiente línea cuando utilice un proxy en la red
8 $cliente = new SoapClient("http://ws.unl.edu.ec/sgaws/wsinstucional/soap/api.wsdl",array
  ('proxy_host'=> "proxy de la red",'proxy_port' => 8080,'login'=> "usuario", 'password' =>
  "password"));
9
10 $parametros = array('sigla'=>'ASH');
11 $servidor = $cliente->sgaws_datos_area($parametros);
12 print_r ($servidor);
13
14 ?>
```

Ejemplo de cliente en Python [click aquí](#)

```
cliente.py
1 #!/usr/bin/env python
2 #-*- coding:utf-8 -*-
3 from SOAPpy import Config, HTTPTransport, SOAPAddress,
4 class myHTTPTransport (HTTPTransport):
5     username = None
6     passwd = None
7
8     @classmethod
9     def setAuthentication(cls,u,p):
10         cls.username = u
11         cls.passwd = p
12
13     def call(self, addr, data, namespace, soapaction=None, encoding=None,http_proxy=None,
14             config=Config):
15         if not isinstance(addr, SOAPAddress):
16             addr=SOAPAddress(addr, config)
17         if self.username != None:
18             addr.user = self.username+"."+self.passwd
19         return HTTPTransport.call(self, addr, data, namespace, soapaction,encoding, http_proxy,
20                                 config)
21
22 if __name__ == '__main__':
23     #wsdl de la categoría de información personal debido a que se invoca un método de esta |
24     wsdlFile = 'http://user:password@ws.unl.edu.ec/sgaws/wspersonal/soap/api.wsdl'
25     myHTTPTransport.setAuthentication('user', 'password')
26     server = WSDL.Proxy(wsdlFile, transport=myHTTPTransport)
27     print server.sgaws_datos_estudiante(cedula='1234567890')
```

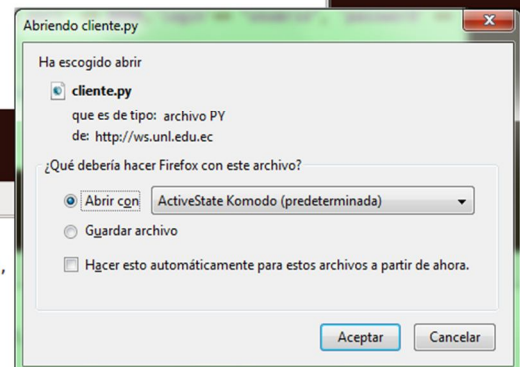


Fig.106: Descripción de cómo usar el SGAWebservices

### 7.1.5. Deployar Turbogears con Apache2

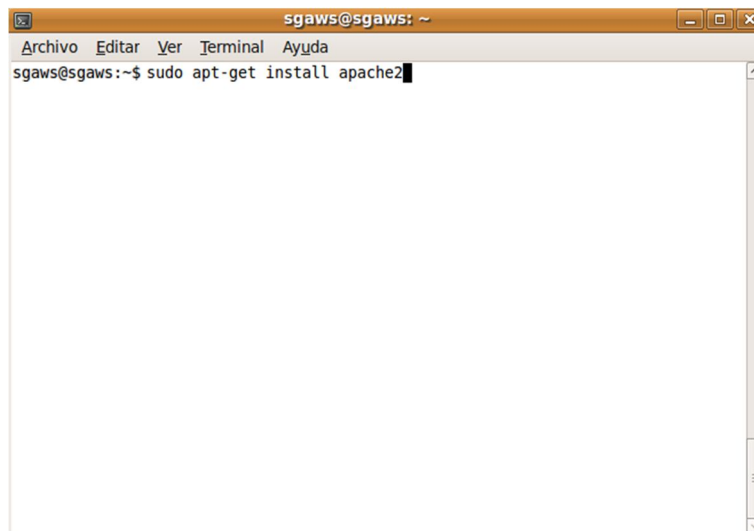
El proceso de levantar o deployar el proyecto en el servidor de aplicaciones Apache, permitió establecer el nivel de seguridad a la aplicación a través de la Autenticación Básica con apache2. A continuación se detalla el proceso llevado a cabo:

#### 7.1.5.1. Instalación de Apache2

El proceso llevado para instalar apache2 es el siguiente:

1. Ejecutar el siguiente comando en consola:

```
sudo apt-get install apache2
```



**Fig.107: Instalación de apache2**

#### 7.1.5.2. Instalación del módulo proxy\_http

Para instalar el módulo de apache necesario para la deployment con turbogears se realizó lo siguiente:

1. Ejecutar en consola el siguiente comando:

```
sudo apt-get install libapache2-mod-proxy-html
```

2. Activar el módulo instalado como se muestra a continuación:

```
sudo a2enmod proxy  
sudo a2enmod proxy_http
```

### 7.1.5.3. Creación y configuración del archivo sgawsproxy

Para que el proyecto desarrollado en turbogears pueda levantarse en el servidor de aplicaciones apache2 fue necesario llevar a cabo los siguientes pasos.

1. Crear un nuevo sitio para apache denominado sgawsproxy

El archivo contiene lo siguiente:

```
Listen 81
NameVirtualHost *:81
<VirtualHost *:81>

    ServerAdmin webmaster@localhost

    DocumentRoot /home/sgaws/webservice/sgaws/sgaws/publi
    #if using awstats
    CustomLog /var/log/apache2/sgaws.log combined

    <Directory /home/sgaws/webservice/sgaws/sgaws/public>
        Order allow,deny
        Allow from all
    </Directory>

    #ProxyPass /inicio.html !
    ProxyPass /sgaws/ http://127.0.0.1:8080/
    ProxyPassReverse /sgaws/ http://127.0.0.1:8080/

</VirtualHost>
```

**Fig.108: Contenido del archivo sgawsproxy**

2. Copiar el sitio creado en el directorio de apache2 ejecutando en consola lo siguiente:

```
sudo cp sgawsproxy /etc/apache2/sites-avalible
```

3. Activar el sitio creado ejecutando lo siguiente:

```
sudo a2ensite sgawsproxy
```

Después de realizada la deployment del framework con apache se procedió a instalar el software necesario para facilitar la administración de usuarios del SGAWebServices, a los cuales se les otorgara los permisos respectivos para acceder al mismo.

### 7.1.6. Implementación de la Autenticación Básica de Apache a través del Mod\_Auth\_Mysql

Para llevar a efecto la implementación de la Autenticación Básica fue preciso realizar la instalación y configuración de los paquetes necesarios para este proceso mediante la utilización del módulo mod\_auth\_mysql, controlando de esta manera el acceso al SGAWebServices.

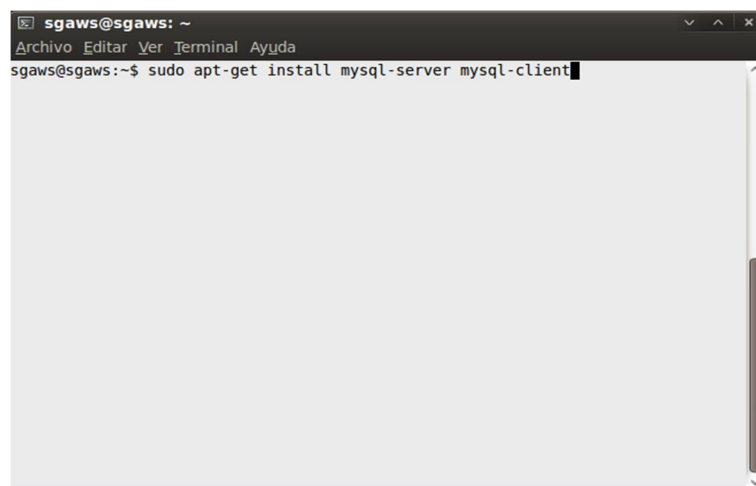
El proceso llevado acabo fue el siguiente:

#### 7.1.6.1. Instalación de Mysql

Para instalar el gestor de base de datos Mysql con los paquetes respectivos de realizaron los siguientes pasos:

1. Ejecutar en consola el siguiente comando:

```
sudo apt-get install mysql-server mysql-client
```



**Fig.109: Instalación de Mysql**

2. Conceder el permiso respectivo para que la instalación ocupe el espacio designado dentro del disco en el cual se va a instalar los paquetes de este gestor.



```
sgaws@sgaws: ~
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~$ sudo apt-get install mysql-server mysql-client
[sudo] password for sgaws:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 libdbd-mysql-perl libdbi-perl libhtml-template-perl libnet-daemon-perl
 libplrpc-perl mysql-client-5.0 mysql-server-5.0 mysql-server-core-5.0
Paquetes sugeridos:
 dbishell libipc-sharedcache-perl mysql-doc-5.0 tinyca mailx
Se instalarán los siguientes paquetes NUEVOS:
 libdbd-mysql-perl libdbi-perl libhtml-template-perl libnet-daemon-perl
 libplrpc-perl mysql-client mysql-client-5.0 mysql-server mysql-server-5.0
 mysql-server-core-5.0
0 actualizados, 10 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 37,0MB de archivos.
Se utilizarán 114MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? █
```

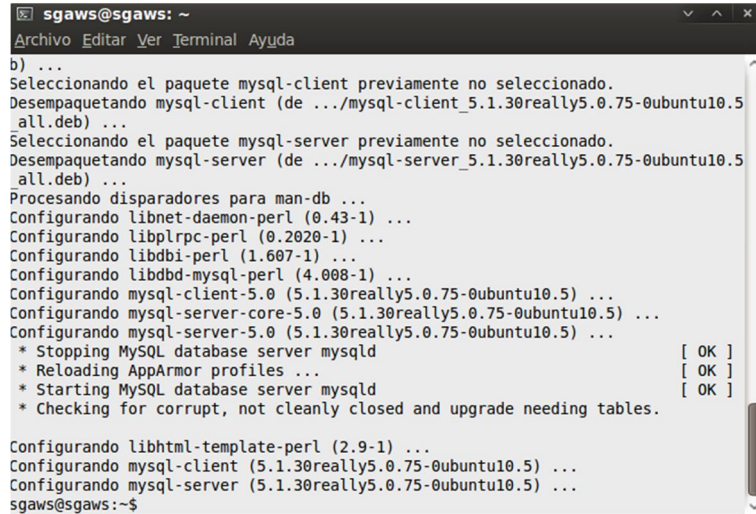
**Fig.110: Conceder permisos de instalación**

3. Configurar el administrador principal, y dar un nombre de usuario administrador de Mysql y contraseña la cual es solicitada por dos ocasiones

```
sgaws@sgaws: ~
Archivo Editar Ver Terminal Ayuda
Configuración de paquetes
Configuración de mysql-server-5.0
Se recomienda que configure una contraseña para el usuario «root»
(administrador) de MySQL, aunque no es obligatorio.
No se modificará la contraseña si deja el espacio en blanco.
Nueva contraseña para el usuario «root» de MySQL:
****
<Aceptar>
```

**Fig.111: Configuración de Mysql**

4. Finalmente esperar que culmine la instalación de mysql con su configuración respectiva



```
sgaws@sgaws: ~
Archivo Editar Ver Terminal Ayuda
b) ...
Seleccionando el paquete mysql-client previamente no seleccionado.
Desempaquetando mysql-client (de ../mysql-client_5.1.30really5.0.75-0ubuntu10.5
all.deb) ...
Seleccionando el paquete mysql-server previamente no seleccionado.
Desempaquetando mysql-server (de ../mysql-server_5.1.30really5.0.75-0ubuntu10.5
all.deb) ...
Procesando disparadores para man-db ...
Configurando libnet-daemon-perl (0.43-1) ...
Configurando liblprc-perl (0.2020-1) ...
Configurando libdbi-perl (1.607-1) ...
Configurando libdbd-mysql-perl (4.008-1) ...
Configurando mysql-client-5.0 (5.1.30really5.0.75-0ubuntu10.5) ...
Configurando mysql-server-core-5.0 (5.1.30really5.0.75-0ubuntu10.5) ...
Configurando mysql-server-5.0 (5.1.30really5.0.75-0ubuntu10.5) ...
* Stopping MySQL database server mysqld [ OK ]
* Reloading AppArmor profiles ... [ OK ]
* Starting MySQL database server mysqld [ OK ]
* Checking for corrupt, not cleanly closed and upgrade needing tables.

Configurando libhtml-template-perl (2.9-1) ...
Configurando mysql-client (5.1.30really5.0.75-0ubuntu10.5) ...
Configurando mysql-server (5.1.30really5.0.75-0ubuntu10.5) ...
sgaws@sgaws:~$
```

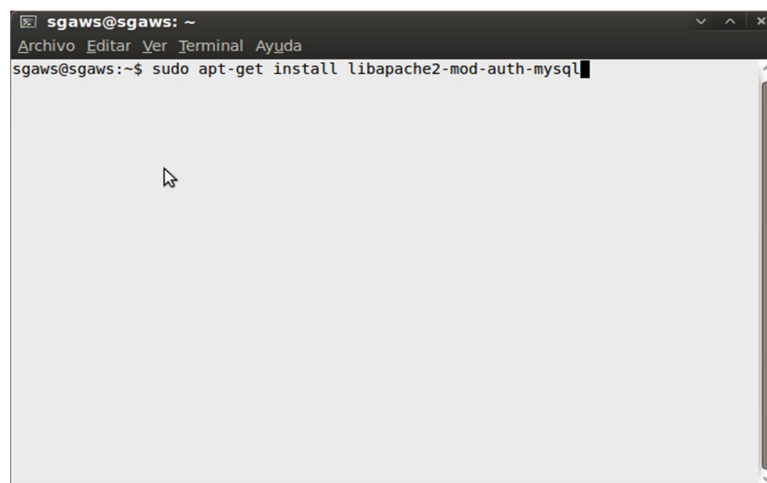
Fig.112: Finalización del proceso de instalación y configuración de Mysql

### 7.1.6.2. Instalación del módulo libapache2-mod-auth-mysql

Concluida la instalación del gestor de base de datos MySQL, se procedió a efectuar la instalación del módulo de Apache denominado **mod\_auth\_mysql**, el mismo que permitió llevar a cabo la autenticación básica de apache a través de una base de datos en mysql. Para realizar la instalación del módulo de autenticación de apache se realizó lo siguiente:

1. Ejecutar en el terminal el siguiente comando:

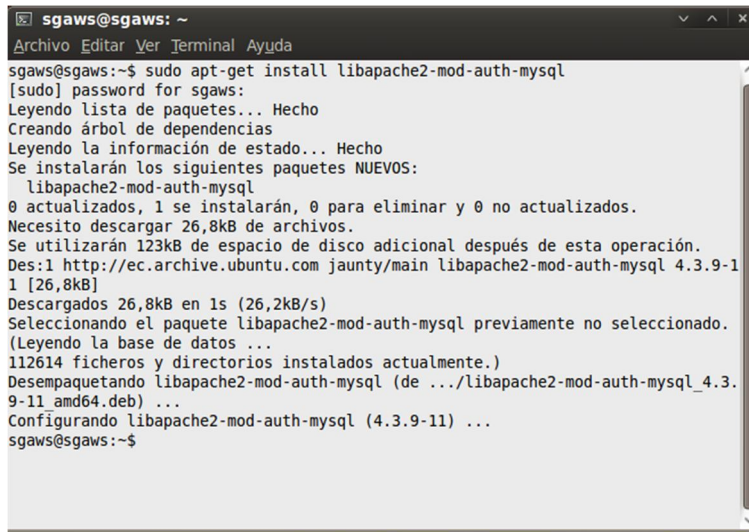
```
sudo apt-get install libapache2-mod-auth-mysql
```



```
sgaws@sgaws: ~
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~$ sudo apt-get install libapache2-mod-auth-mysql
```

Fig.113: Instalación del libapache2-mod-auth-mysql

2. Esperar que culmine la instalación



```
sgaws@sgaws: ~
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~$ sudo apt-get install libapache2-mod-auth-mysql
[sudo] password for sgaws:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
 libapache2-mod-auth-mysql
0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 26,8kB de archivos.
Se utilizarán 123kB de espacio de disco adicional después de esta operación.
Des:1 http://ec.archive.ubuntu.com jaunty/main libapache2-mod-auth-mysql 4.3.9-1
1 [26,8kB]
Descargados 26,8kB en 1s (26,2kB/s)
Seleccionando el paquete libapache2-mod-auth-mysql previamente no seleccionado.
(Leyendo la base de datos ...
112614 ficheros y directorios instalados actualmente.)
Desempaquetando libapache2-mod-auth-mysql (de ../libapache2-mod-auth-mysql_4.3.
9-11 amd64.deb) ...
Configurando libapache2-mod-auth-mysql (4.3.9-11) ...
sgaws@sgaws:~$
```

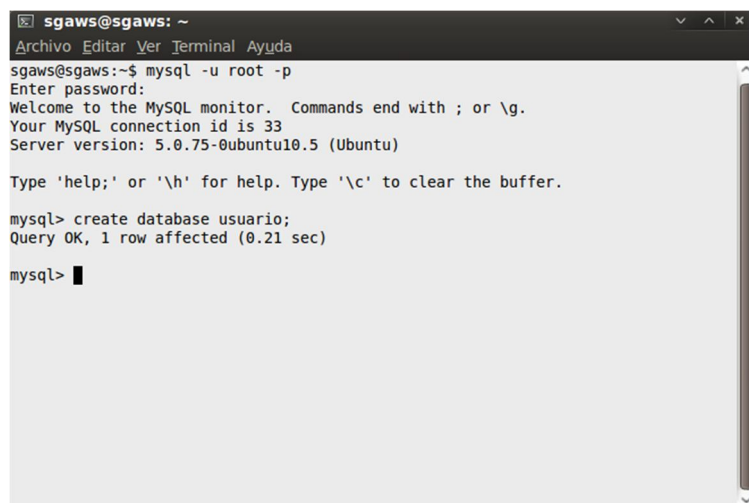
**Fig.114: Finalización del proceso de instalación del libapache2-mod-auth-mysql**

### 7.1.6.3. Creación y configuración de la base de datos para usuarios del SGAWebsservices.

Una vez instalados los programas necesarios para realizar la autenticación básica de apache con mysql, se procedió a efectuar la configuración necesaria que permita administrar la base de datos de los usuarios del sgaws. El proceso llevado a cabo fue el siguiente:

1. Abrir una consola de mysql mediante el comando:

```
mysql -u root -p
```



```
sgaws@sgaws: ~
Archivo Editar Ver Terminal Ayuda
sgaws@sgaws:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
Server version: 5.0.75-0ubuntu10.5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

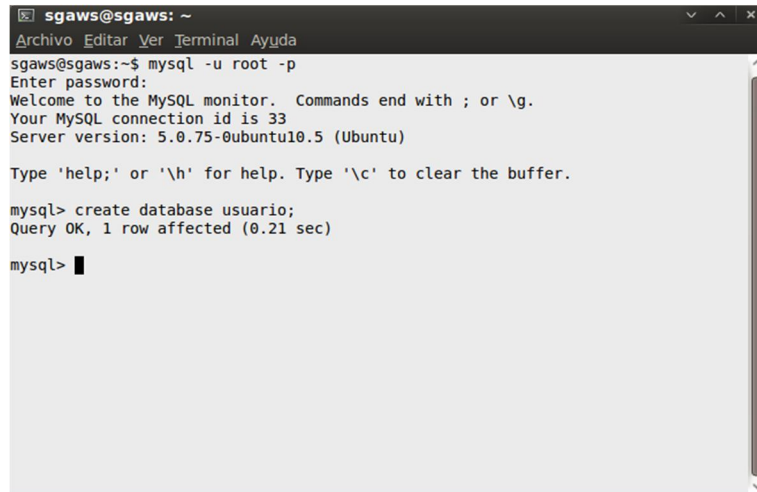
mysql> create database usuario;
Query OK, 1 row affected (0.21 sec)

mysql> █
```

**Fig.115: Consola de Mysql**

2. Digitar el password solicitado por **mysql**, para que se abra la consola de este gestor de datos, en la cual se procede a crear la base de datos **usuario**, como se muestra en la siguiente línea de código:

```
create database usuario;
```

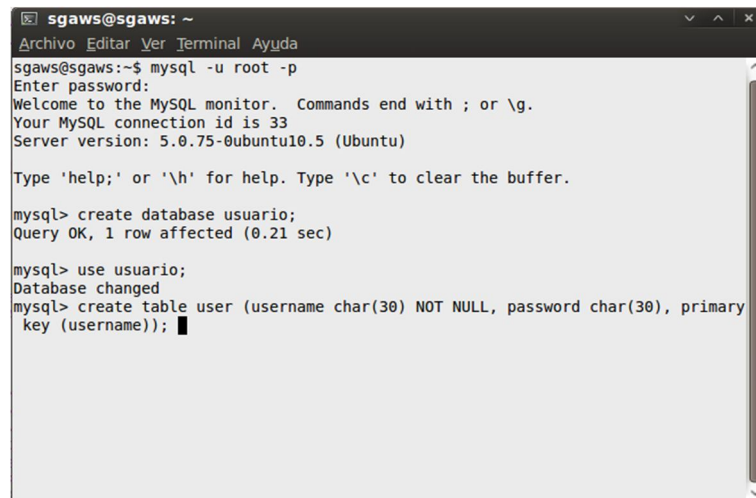


```
sgaws@sgaws: ~  
Archivo Editar Ver Terminal Ayuda  
sgaws@sgaws:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 33  
Server version: 5.0.75-0ubuntu10.5 (Ubuntu)  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql> create database usuario;  
Query OK, 1 row affected (0.21 sec)  
  
mysql> █
```

**Fig.116: Creación de la base de datos de usuarios**

3. Crear la tabla de usuarios denominada **user** en donde se registran los usuarios que harán uso de estos métodos del SGAWebsservices. Esta tabla se creará mediante el comando de **mysql**:

```
create table user (username char(30) NOT NULL, password  
char(30), primary key (username));
```



```
sgaws@sgaws: ~  
Archivo Editar Ver Terminal Ayuda  
sgaws@sgaws:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 33  
Server version: 5.0.75-0ubuntu10.5 (Ubuntu)  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql> create database usuario;  
Query OK, 1 row affected (0.21 sec)  
  
mysql> use usuario;  
Database changed  
mysql> create table user (username char(30) NOT NULL, password char(30), primary  
key (username)); █
```

**Fig.117: Creación de la tabla de usuarios**

La tabla creada consta de dos campos, uno denominado **username**, que es en donde se guarda el nombre del usuario que ingresa al SGAWebsservices después de haber sido aceptado y asignado los permisos por el encargado del SGA, cabe señalar que esta campo es único, es decir el nombre del usuario no se puede repetir, en la siguiente columna **password** se registra la clave del usuario que se ha sido ingresado. Para ingresar un usuario a la base de datos es necesario:

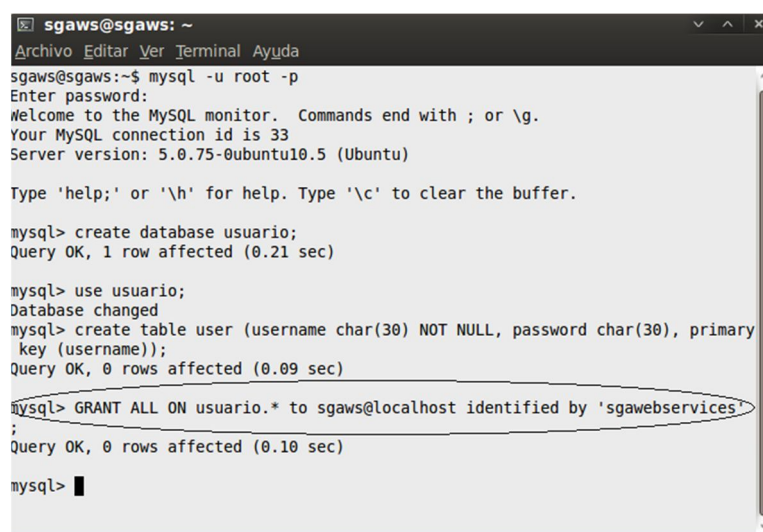
4. Ejecutar en la consola de mysql el código siguiente:

```
insert into user values('sgaws', 'sgawebsservices');
```

Para que la autenticación se lleve a cabo satisfactoriamente es necesario crear un usuario con todos los permisos en mysql el cual se encarga de la administración y gestión del acceso a la tabla donde se almacenaran lo usuarios de SGAWebsservices para que apache pueda utilizar los registros guardados.

5. Guardar el usuario mediante la ejecución del siguiente comando mysql:

```
GRANT ALL ON usuario.* to sgaws@localhost identified by 'sgawebsservices';  
flush privileges;
```



```
sgaws@sgaws: ~  
Archivo Editar Ver Terminal Ayuda  
sgaws@sgaws:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 33  
Server version: 5.0.75-0ubuntu10.5 (Ubuntu)  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql> create database usuario;  
Query OK, 1 row affected (0.21 sec)  
  
mysql> use usuario;  
Database changed  
mysql> create table user (username char(30) NOT NULL, password char(30), primary  
key (username));  
Query OK, 0 rows affected (0.09 sec)  
  
mysql> GRANT ALL ON usuario.* to sgaws@localhost identified by 'sgawebsservices'  
;  
Query OK, 0 rows affected (0.10 sec)  
  
mysql> █
```

Fig.118: Permisos para usuario administrador de la base de datos

La sección de **GRANT ALL ON** otorga todos los permisos a la base de datos **usuario**, para el nuevo usuario **sgaws** y el cual va estar identificado este por la clave **sgawebservices**.

#### 7.1.6.4. Configuración del sitio **sgawsproxy** para la autenticación básica con **apache**.

Una vez terminada la configuración de la base de datos y el usuario en Mysql para administración, se procedió a realizar cambios dentro del archivo **sgawsproxy** creado para la deployment de apache con turbogears, a este archivo se agregó un nuevo segmento de código con las cambios pertinentes del módulo para que la autenticación básica a través de mysql se lleve a cabo con éxito. Como se muestra en la siguiente figura:

```
<Location /sgaws>
  AuthType Basic
  AuthName "SGAWEBSSERVICES"
  AuthBasicAuthoritative Off
  AuthUserFile /dev/null
  AuthGroupFile /dev/null
  Auth_MYSQL on
  AuthMySQL_Authoritative Off
  Auth_MYSQL_DB usuario
  AuthMySQL_Host localhost
  Auth_MYSQL_User sgaws
  Auth_MYSQL_Password sgawebservices
  AuthMySQL_Password_Table user
  AuthMySQL_Username_Field username
  Auth_MYSQL_Password_Field password
  AuthMySQL_Empty_Passwords Off
  AuthMySQL_Encryption_Types Plaintext
  AllowOverride AuthConfig
  Order allow,deny
  Allow from all
  Options All
  <Limit GET POST PUT /home/sgaws/websservice/sgaws/sgaws>
  Require valid-user
  </Limit>
</Location>
```

**Fig.119:** Sección de código agregada al **sgawsproxy** para autenticación

En donde las directivas principales configuradas fueron:

- **AuthType Basic:** Indica que la autenticación será la básica de apache.
- **AuthName "SGAWEBSSERVICES":** Indica el nombre que tendrá en el mensaje en donde se pedirá la autenticación del usuario.
- **AuthBasicAuthoritative Off:** Aquí especificamos que **no** queremos la autenticación clásica (Basic)
- **AuthUserFile /dev/null:** Como no vamos a aplicar la autenticación básica directamente no hay una ruta de directorio del archivo en donde se guarda los usuarios registrados.
- **AuthGroupFile /dev/null:** Al no aplicar la autenticación básica no hay una ruta de directorio del archivo en donde se guarda los grupos creados.

Posteriormente se realizó la configuración de las directivas del mod\_auth\_mysql en donde se realizaron los cambios de:

- **Auth\_MYSQL on:** Indica que se va a realizar la autenticación con el módulo de apache de auth\_mysql.
- **Auth\_MYSQL\_DB usuario:** Indica el nombre de la base de datos en donde se creará la tabla para el registro de usuarios.
- **AuthMYSQL\_Host localhost:** determinan el servidor donde reside la base de datos a usar.
- **Auth\_MYSQL\_User sgaws:** es el nombre del usuario mysql que se utilizará para que apache pueda leer en la base de datos donde están registrados los usuarios.
- **Auth\_MYSQL\_Password sgawebsservices:** es el password del usuario mysql que se utilizará para que apache pueda leer en la base de datos donde están registrados los usuarios.
- **AuthMYSQL\_Password\_Table user:** Indica el nombre de la tabla que contendrá a los usuarios registrados en mysql
- **AuthMYSQL\_Username\_Field username:** Indica el nombre del campo de la tabla que contiene el username de los usuarios.
- **Auth\_MYSQL\_Password\_Field password:** Indica el nombre del campo de la tabla que contiene los password de los usuarios.

- **AuthMySQL\_Empty\_Passwords Off:** Esta directiva indica que no se aceptan usuarios con contraseñas vacías.
- **AuthMySQL\_Encryption\_Types Plaintext:** Especifica el tipo de encriptación que se utilizará para almacenar la clave del usuario dentro de la base de datos, en este caso es texto plano.

Igualmente se configuró el orden de acceso que tendrán los usuarios en la autenticación básica.

- **Order allow,deny:** Primero se evalúa Allow. Se deniega acceso a cualquier host que primero no esté indicado en Allow o que sí lo esté en Deny. El acceso se deniega por defecto.
- **Allow from all:** Permitir el acceso a todos que se autenticuen.
- **Options All:** Acepta todas las opciones excepto MultiViews
- **<Limit GET POST PUT>/home/sgaws/webservice/sgaws/sgaws/  
Require valid-user  
</Limit>:** Se indica que se requiere validar para acceder a la directiva indicada.

#### 7.1.6.5. Comprobar el proceso de autenticación básica

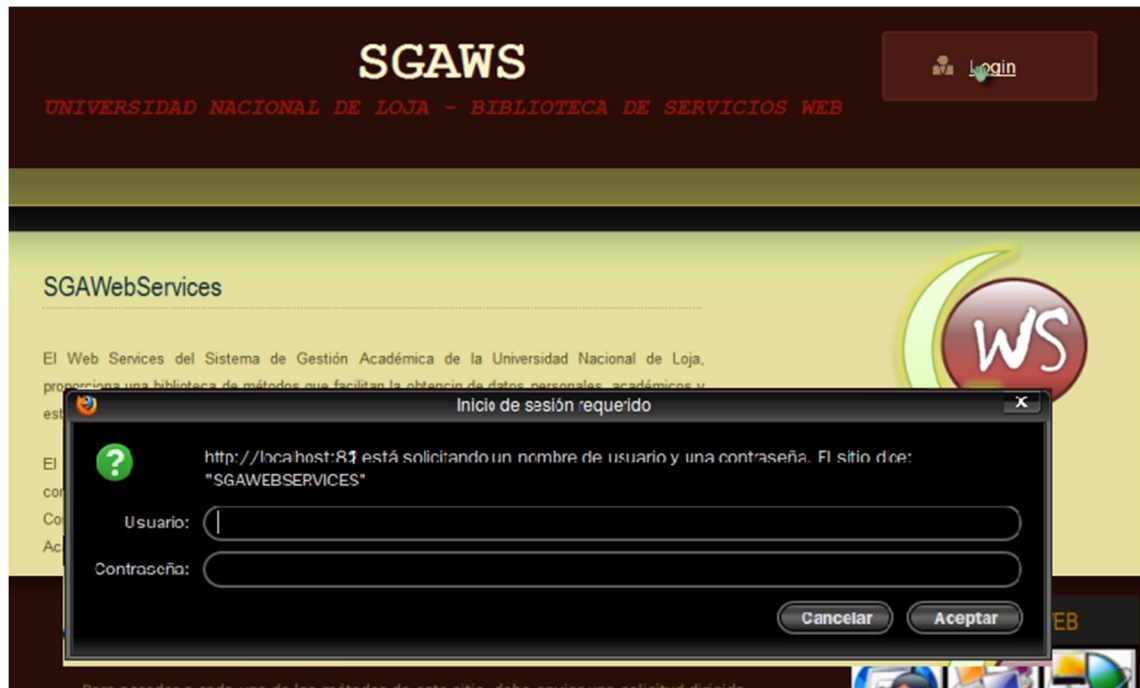
Luego de realizado el proceso de autenticación básica que se encarga de la seguridad del SGAWservices, debe existir el registro de un usuario en la base de datos de mysql y con los permisos respectivos, para acceder al sistema lo cual permitirá verificar el correcto funcionamiento. A continuación en la siguiente figura se muestra cómo se lleva a cabo el proceso de autenticación:

1. Digitar en el navegador web los siguiente:

```
http://localhost:81/
```

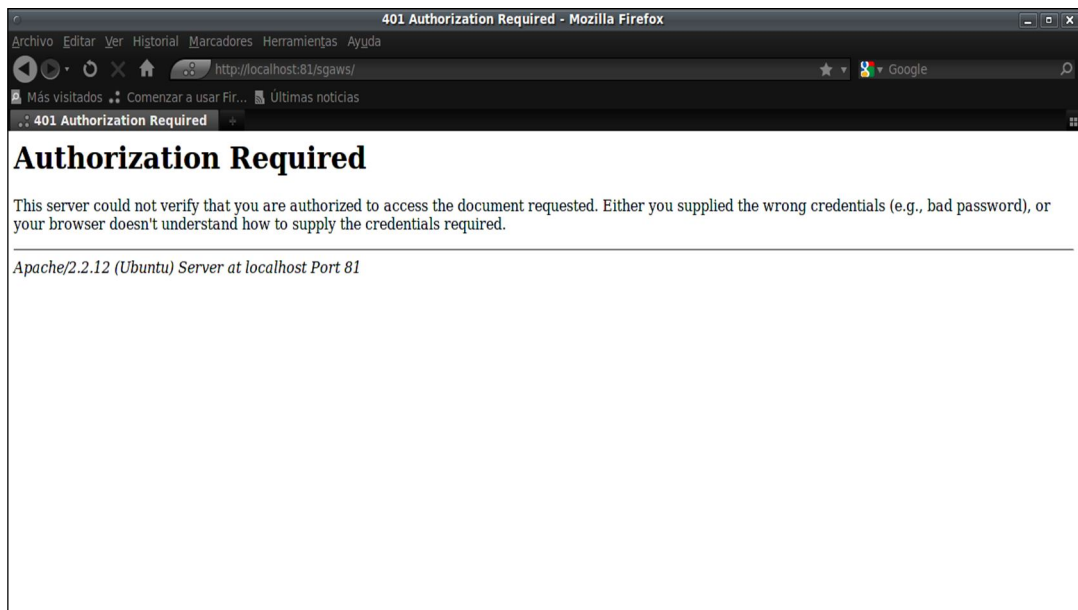
2. Dando click en la palabra login, se podrá ingresar el usuario y contraseña para la autenticación en el sistema como se muestra en la siguiente figura y una vez ingresados los datos correctamente se tendrá acceso SGAWservices.





**Fig.120: Autenticación al SGAwebservices en el navegador web**

3. En caso de cancelar la autenticación, saldrá una página web en la que se indica que la aplicación requiere autenticarse y deberá de ingresar obligatoriamente los datos de su usuario para hacer uso del SGAWebservices



**Fig.121: Imagen de error por cancelar Autenticación**

## 7.2. CONFIGURAR EL SISTEMA DE ENCUESTAS DE LA UNIVERSIDAD NACIONAL DE LOJA QUE PRUEBA EL FUNCIONAMIENTO DE MÉTODOS DEL SGAWEBSSERVICES

Para el cumplimiento de uno de los objetivos del proyecto y en vista de que el Instituto de Investigaciones Socioeconómicas (IDISE) de la Universidad Nacional de Loja requería el desarrollo de una encuesta en línea para obtener datos socioeconómicos de los estudiantes y resultados estadísticos de dicha encuesta, se procedió al desarrollo de dicha encuesta para probar y verificar el correcto funcionamiento del SGAWebServices depurando cualquier tipo de error.

Tras la investigación de software que permita el manejo de encuestas se determinó que LimeSurvey un sistema para la administración de encuestas en línea desarrollado en PHP constituía la mejor opción, por lo que se procedió a la configuración e instalación del mismo, con la finalidad de probar, asegurar y garantizar que el proyecto funciona correctamente. El proceso comienza con:

### 7.2.1. Instalar php

Instalación del lenguaje de programación PHP y algunos de sus paquetes debido a que LimeSurvey está basado en el mismo. El proceso se llevó a cabo fue el siguiente:

1. Ejecutar en consola el siguiente comando:

```
sudo apt-get install php5
```

2. Instalar el paquete para gestionar gráficos dinámicamente para php5 ejecutando:

```
sudo apt-get install php5-gd
```

3. Instalar el paquete que permite conectar php5 con mysql, ejecutando el siguiente comando:

```
sudo apt-get install php5-mysql
```

4. Instalar el módulo de apache2 para php5 ejecutando:

```
sudo apt-get install libapache2-mod-php5
```

### 7.2.2. Uso de Lime Survey

Para hacer uso de Lime Survey se relazaron los siguientes pasos:

- **Descargar el archivo comprimido de LimeSurvey**

Visitar el sitio de SourceForge para descargar la última versión de LimeSurvey: <http://sourceforge.net/projects/limesurvey/>. Descargar y guardar en la ubicación deseada según la configuración del servidor web usado.

- **Descomprimir el archivo comprimido de LimeSurvey**

Descomprimir el archivo .zip en el siguiente directorio /var/www/ asegurándose que se descomprima la estructura de directorios y los archivos.

- **Cambiar los permisos en algunos directorios**

Algunos directorios deben tener permisos de escritura para el servidor web, a fin de que se pueda importar encuestas y archivos de encuestados así como la creación y modificación de plantillas. Estos son:

- +limesurvey/tmp+-
- -+limesurvey/templates+-

Esto se realizó ejecutando el siguiente comando:

```
sudo chmod -r o+w limesurvey/tmp
sudo chmod -r o+w limesurvey/templates
```

El resto de archivos y directorios deben tener permiso sólo de lectura

### 7.2.3. Configurar Lime Survey

Para configurar LimeSurvey fue necesario editar el archivo config.php que se encuentra en el directorio '/var/www/limesurvey/', además es aquí donde se definieron las variables globales que se utilizaran dentro de la aplicación.

#### Parámetros de configuración

- **\$databasetype:** Tipo de Base de Datos. En nuestro caso "mysql"
- **\$databaselocation:** Dirección de red donde esté el servidor de Bases de Datos. Para nosotras "localhost".

- **\$databaseport:** Puerto usado por el servidor de BD. En la mayoría de los casos "default" es lo adecuado.
- **\$databasename:** El nombre de la base de datos que contendrá las tablas de LimeSurvey. "limesurvey"
- **\$databaseuser:** El nombre de usuario para acceder a la base de datos
- **\$databasepass:** La contraseña del usuario antes dicho
- **\$dbprefix:** Prefijo que tendrán las tablas de la base de datos. En nuestro caso en blanco ya que utilizamos una base de datos en exclusiva para LimeSurvey.
- **\$rooturl:** Parámetro que contiene la dirección web (URL) de los scripts de LimeSurvey. Usaremos el valor por defecto, es decir "http://{\$\_SERVER['SERVER\_NAME']}/limesurvey"
- **\$rootdir:** Se indica la ruta en el disco duro a sus scripts. Igual que en el caso anterior, utilizaremos el valor por defecto `dirname(__FILE__)`.
- **\$defaultuser:** Usuario administrador por defecto.
- **\$defaultpass:** Contraseña inicial del usuario administrador.
- **\$debug:** Tiene valores 1 o 0. Sirve para poder ver los errores que se presentan. Nosotros los hemos fijado en 1.

#### 7.2.4. Instalación de Lime Survey

Una vez realizadas las configuraciones necesarias se instaló LimeSurvey colocando en el navegador la siguiente dirección "<http://localhost/limesurvey/admin/install>", donde se pidió crear la base de datos y/o crear las tablas de LimeSurvey dentro de la base de datos especificada el archivo de configuración. Seguidamente se mostró el mensaje indicando que se ha creado la estructura de tablas exitosamente. Finalmente por motivos de seguridad se renombró el directorio "[/var/www/limesurvey/admin/install](#)".

#### 7.2.5. Acceso a la plataforma por primera vez

Posteriormente de realizada la instalación y las configuraciones necesarias, se colocó en el navegador la siguiente dirección <http://localhost/limesurvey/admin/admin.php>, para acceder a la pantalla de autenticación de la plataforma, donde se indica el usuario y contraseña, que se definió en el archivo `config.php` y el idioma en el que se desea trabajar en este caso español. Como se muestra en la figura siguiente:



**Fig.122: Ingreso a la plataforma Lime Survey**

Luego de llevado a efecto el proceso de autenticación se analizó los componentes principales de Lime Survey como se muestra en la siguiente figura:



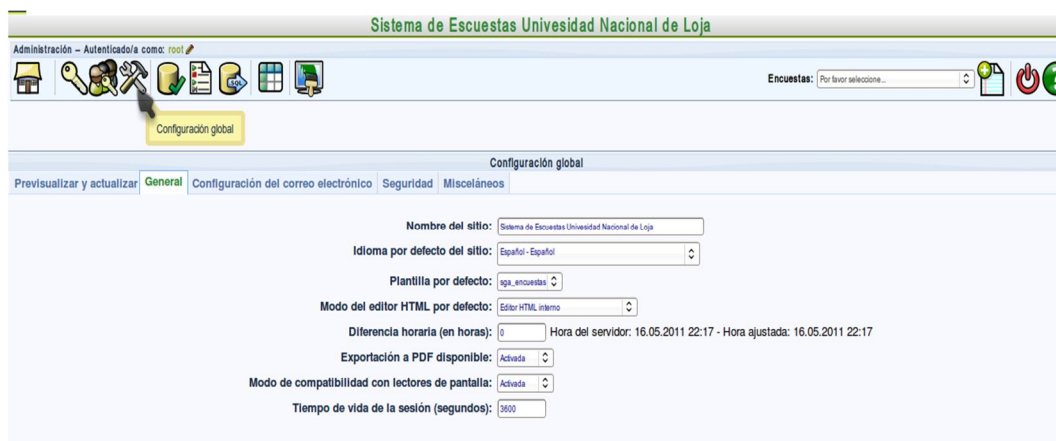
**Fig.123: Vista principal para configuración de Lime Survey**

1. **Volver a la pantalla de inicio:** Vuelve a esta pantalla desde cualquier lugar.
2. **Opciones del sistema:** Permite ajustar el idioma con el que se desea visualizar la plataforma y ver algunas estadísticas básicas sobre la misma.
3. **Listado de encuestas:** Muestra un listado de las encuestas elaboradas en la plataforma. Pulsando sobre el nombre de la encuesta creada podremos acceder a la administración propia a esa encuesta.
4. **Editor de plantillas:** Permite que la encuesta puede ser mostrada según diferentes estilos visuales (colores, bordes de cuadros, tipos de letra, etc.). LimeSurvey ofrece varios estilos, sin embargo es posible crear estilos propios.
5. **Lista de selección de encuesta:** Donde se selecciona una de las encuestas creadas, al seleccionarla nos redirige a la página de administración de la misma.
6. **Creación de nueva encuesta:** Abre el formulario que es necesario para la creación de una encuesta.
7. **Salida:** Cierra la sesión de trabajo.

- 8. Ayuda en línea:** Permite ver la ayuda al respecto de la pantalla que se esté visualizando.

### 7.2.6. Configuración de la plataforma de acuerdo a los requerimientos del sistema de encuestas

Una vez analizados los componentes de la plataforma se procedió a realizar las configuraciones necesarias de la misma, definiendo los parámetros principales como: Nombre del sitio, idioma por defecto del sitio, plantilla usada, manejo de editor de HTML, diferencia horaria, permitir exportación en pdf, tiempo de vida de la sesión, entre otros, como se muestra en la siguiente figura:



**Fig.124: Configuración de Lime Survey**

### 7.2.7. Creación de una plantilla HTML para la encuesta

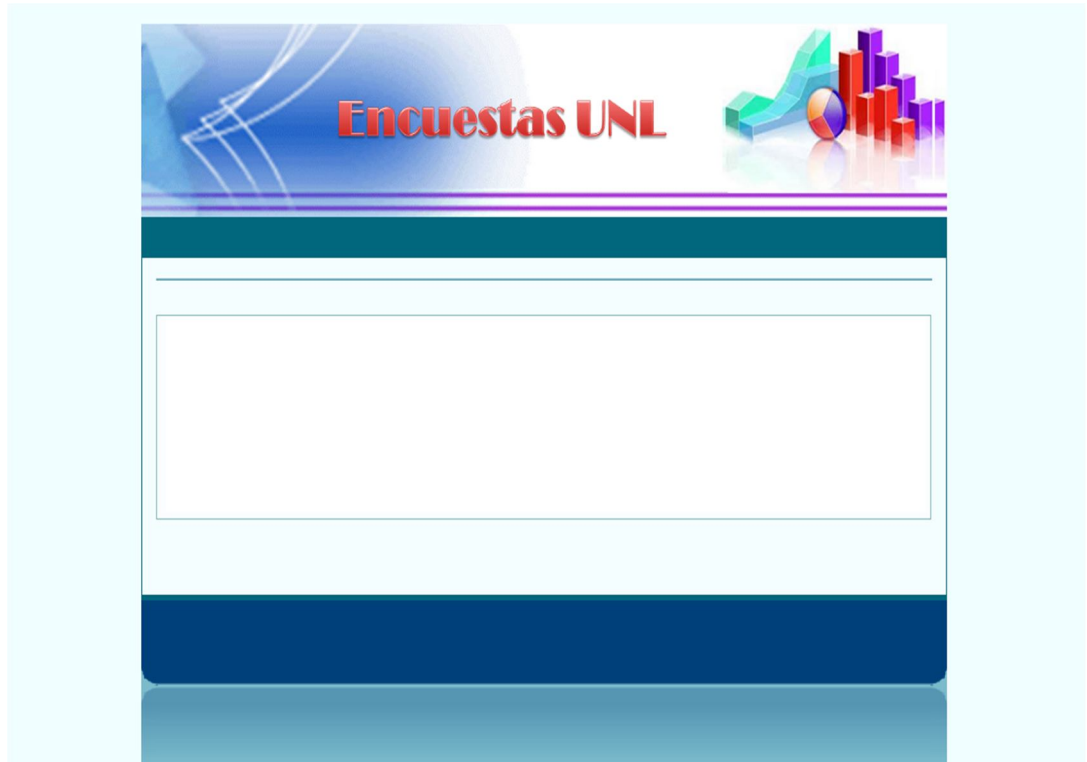
Otro de los procesos realizados fue la creación de una plantilla propia que se adapte al tipo de encuesta realizada, utilizando el Editor de plantillas de Lime Survey, ya que gracias a este componente la encuesta puede ser mostrada según diferentes estilos visuales (colores, bordes de cuadros, tipos de letra, etc.).

Se realizó una copia de una plantilla ya creada y se la modificó de acuerdo a las necesidades de la encuesta, la nueva plantilla denominada sga\_encuestas con un estilo propio para la misma. Esta plantilla se aplicó a los siguientes templates:

- Página de Bienvenida
- Página de Autenticación
- Página de Preguntas
- Página de Enviar Encuesta

- Página de Encuesta Completada
- Página de Borrar respuestas
- Página de Guardar Encuesta

A continuación se muestra el estilo creado:



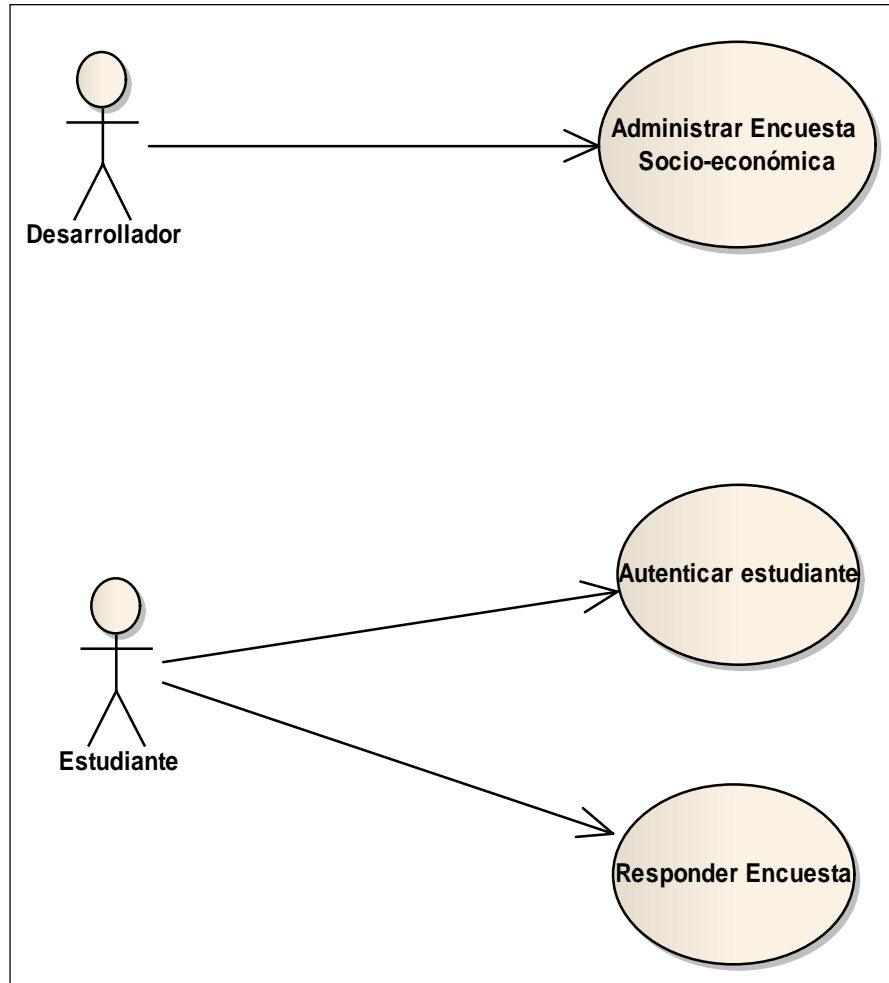
**Fig.125: Plantilla HTML del Sistema de Encuestas**

### **7.2.8. Creación de la Encuesta Socioeconómica para estudiantes**

El proceso inició con el análisis de las necesidades del Instituto de Investigaciones Socioeconómicas (IDISE) de la Universidad Nacional de Loja que requería el desarrollo de una encuesta en línea para obtener datos socioeconómicos de los estudiantes y cuyos resultados estadísticos puedan ser manejados por ellos posteriormente. El formato de la encuesta fue proporcionada por dicho departamento (**ver Anexo# 3**).

#### **7.2.8.1. Diagrama de Casos de Uso**

Luego analizados los requerimientos se determinaron los siguientes casos de uso:



**Fig.126: Diagrama de Casos de Uso del Sistema de Encuestas**

#### **7.2.8.2. Desarrollo de Historias de Usuario y Diagramas de Actividades**

Los casos de uso determinados se constituyeron en las Historias de Usuario del Sistema de Encuestas donde se describen los procesos llevados a cabo, son utilizadas en vista de que se utiliza la metodología de desarrollo de software XP (Programación Extrema).

A continuación el desarrollo de la historia de usuario #1:



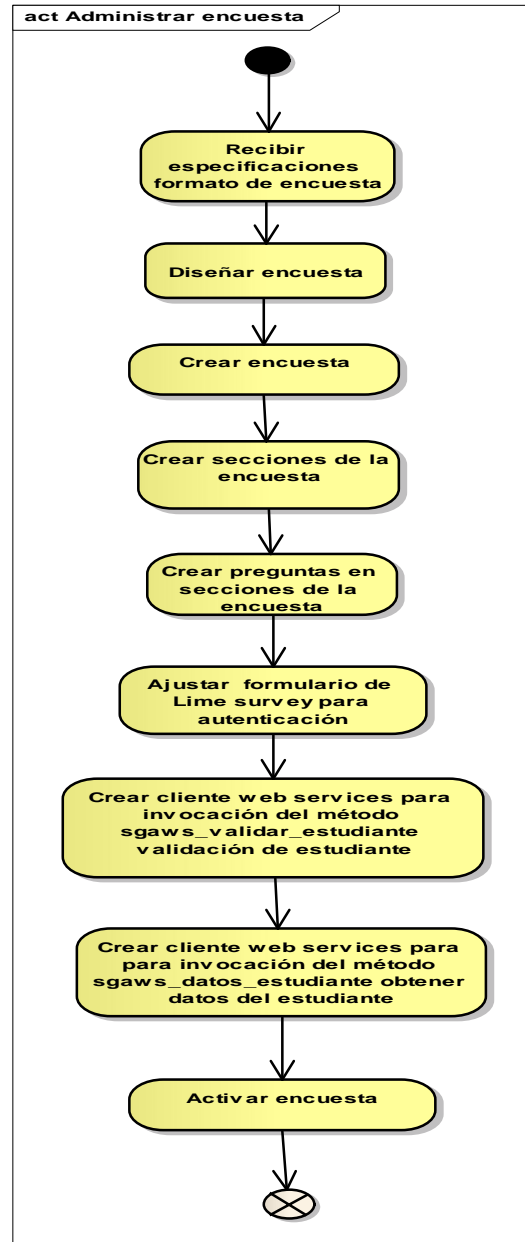
<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Administrar encuesta Socio-Económica		
<b>Nº:</b> 1	<b>Usuario:</b> Desarrollador	
<b>Prioridad:</b> Alta	<b>Puntos(estimación):</b> 3 semanas	<b>Iteración:</b> 3
<b>Descripción:</b> Crear la encuesta dentro de la plataforma de LimeSurvey, de acuerdo a las especificaciones indicadas por el del Instituto de Investigaciones Socioeconómicas (IDISE) de la Universidad Nacional de Loja y construcción del procesos para llevar a cabo la interacción entre la encuesta creada y el SGAWebservices, a través del uso de dos de sus servicios el sgaws_validar_estudiante para la autenticación y sgaws_carreras_estudiante para obtener los datos del mismo.		
<b>Observaciones:</b>		

**Tabla. 36: Historia de Usuario #1 del Sistema de Encuestas**

### Diagrama de Actividades

Para el diseño del diagrama de actividades fue necesario tener claro el siguiente proceso llevado a cabo:

- Recibir especificaciones del formato de la encuesta
- Diseñar encuesta
- Crear encuesta
- Crear secciones de la encuesta
- Crear preguntas en secciones de la encuesta
- Ajustar formulario de Lime survey para autenticación
- Crear cliente web services para invocación del método sgaws\_validar\_estudiante validación de estudiante
- Crear cliente web services para para invocación del método sgaws\_datos\_estudiante obtener datos del estudiante
- Activar encuesta



**Fig.127: Diagrama de Actividades Administrar encuesta Socio-Económica**

Con el análisis y diseño realizado se llevo a cabo el siguiente proceso:

#### a. Creación de la Encuesta

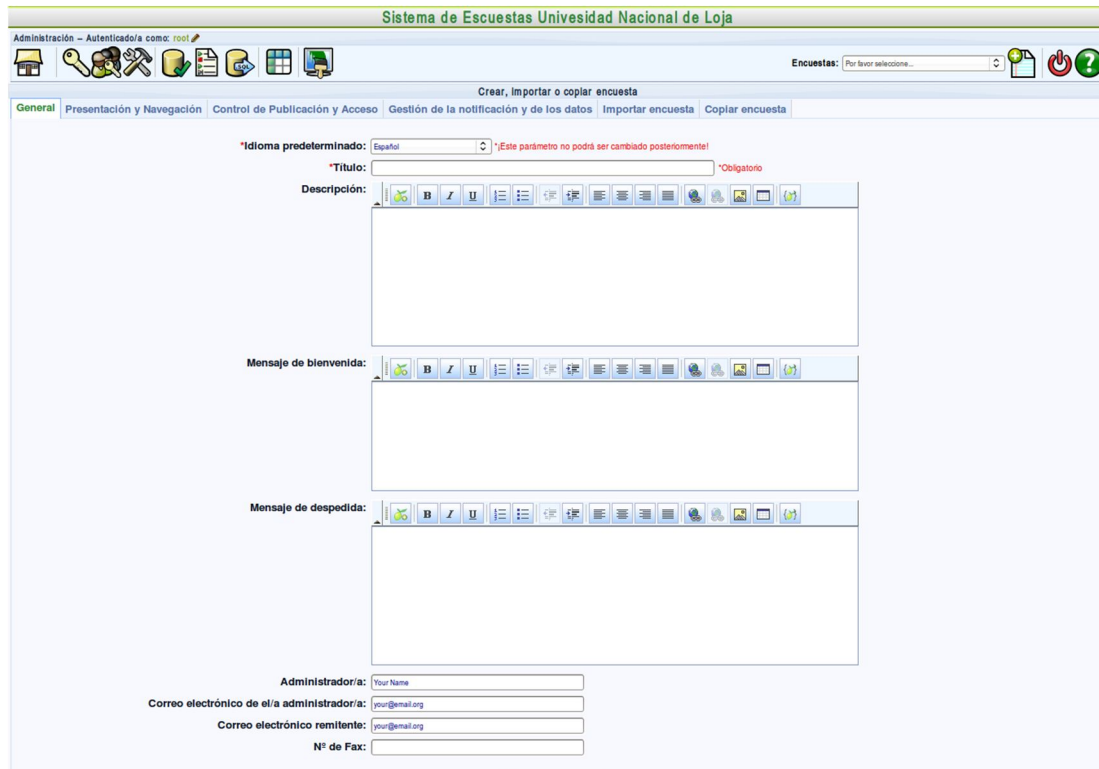
Para crear la encuesta se seleccionó el botón  de la plataforma Lime Survey como se muestra en la figura siguiente:



**Fig.128: Ejemplo para crear encuesta**

Posteriormente se indicó los parámetros necesarios para la creación de la encuesta, en base a los requerimientos de la encuesta:

- Idioma utilizado por defecto.
- Título o nombre de la encuesta.
- Descripción acerca de la finalidad de la encuesta
- Saludo de bienvenida e instrucciones generales para la realización de la encuesta
- Tipo de planteamiento de la encuesta. En este caso Grupo por grupo: donde se presentarán todas las preguntas pertenecientes al mismo grupo en cada pantalla. Es el formato más habitual.
- Plantilla que utilizará la encuesta en nuestro caso la plantilla sga\_encuestas diseñada anteriormente
- Permitir guardar las respuestas dadas en una encuesta así no haya concluido la misma.
- Mostrar botón anterior que permite antes de concluir con la encuesta, retroceder a las preguntas respondidas anteriormente para corregir alguna respuesta dada.
- Indicar Expiración y fecha si se tiene fecha de caducidad predefinida.



**Fig.129: Formulario para crear una encuesta**

## b. Creación de Secciones


Una vez creada la encuesta se procedió a crear las secciones de la misma, de acuerdo al formato proporcionado.

Las secciones creadas fueron las siguientes:

- Información General
- Antecedentes personales y socio-económicos
- Antecedentes Educativos
- Recursos de Apoyo
- Participación
- Salud Mental

Para crear una sección es necesario especificar el título o nombre de la sección y si se desea una descripción de la misma.

Como se muestra en la siguiente figura:



**Fig.130: Formulario para crear una sección de la encuesta**

### c. Creación de preguntas

Una vez creada la sección es necesario agregarle preguntas que formaran parte de esta sección de la encuesta.

Para crear una pregunta es necesario especificar:

- Un código de pregunta
- La pregunta en si
- Ayuda acerca de la pregunta si se requiriera
- Tipo de pregunta, ya que limesurvey nos ofrece una gama de opciones en este aspecto.
- Especificar en qué sección de la encuesta va la pregunta
- Indicar si es obligatorio o no que se responda la pregunta
- Indicar si requiere algún tipo de validación
- Finalmente la posición de la pregunta

En la siguiente figura se muestra como crear una pregunta:

**Fig.131: Formulario para crear una pregunta de la encuesta**

Es importante especificar que cuando se crea la encuesta es almacenada en la base de datos de la plataforma Lime Surver con un identificador que la diferencie de las demás, que servirá para hacer la llamada en el navegador a dicha encuesta, el link para llamar la encuesta creada es el siguiente:

<http://localhost/limesurvey/index.php?sid=21148&lang=es>. Lime Survey ofrece un resumen de la encuesta creada como se muestra en la siguiente figura:

**Fig.132: Resumen de la encuesta creada**



### Crear el cliente SOAP para la autenticación de estudiantes

Una vez creada la encuesta, fue necesario realizar el proceso para la autenticación del estudiante para que puedan acceder y participar de la encuesta, para lo cual se desarrolló un cliente webservices que invoca al método **sgaws\_validar\_estudiante** que pertenece a la categoría de validación del SGAWebservices y cuya dirección es *<http://ws.unl.edu.ec/sgaws/wsvalidacion/soap/api.wsdl>*, para esto se creó un script `sgauth.php` el cual se lo realizó utilizando el protocolo SOAP para el lenguaje de programación php y con la autenticación básica de apache.

```
<?php
require_once(dirname(__FILE__).'/config-defaults.php');
require_once(dirname(__FILE__).'/common.php');
require_once(dirname(__FILE__).'/config.php');

try {

$cliente = new SoapClient("http://ws.unl.edu.ec/sgaws/wsvalidacion/soap/
api.wsdl",array('login'=>$usersgaws,'password'=>$passwordsgaws));
$parametros = array('cedula'=>$_POST["usuario"],'clave'=>$_POST["token"]);
$servidor = $cliente->sgaws_validar_estudiante($parametros);

//print_r ($servidor);
$error = 0;
if ($servidor->result=="true"){
    // obtener el id del survey
    if (!isset($surveyid))
    {
        $surveyid=returnnglobal('sid');
    }
    else
    {
        //This next line ensures that the $surveyid value is never anything
but a number.
        $surveyid=sanitize_int($surveyid);
    }
    $cliente = new SoapClient("http://ws.unl.edu.ec/sgaws/wspersonal/soap/
api.wsdl",array('login'=>$usersgaws,'password'=>$passwordsgaws));
    $parametros = array('cedula'=>$_POST["usuario"]);
    $servidor = $cliente->sgaws_datos_estudiante($parametros);
    $resultado = json_decode($servidor->result);
    $cedula = $resultado[0];
    $nombres = $resultado[1];
    $apellidos = $resultado[2];
    $email = $resultado[9];
    //echo "DEBUG: ".$cedula.$nombres;
    //primero comprobar si existe antes de hacer el insert
    $ntquery = "SELECT * FROM {$dbprefix}tokens_{$surveyid} WHERE
token='{$cedula}'";
    $ntresult = $connect->Execute($ntquery); //Checked

    if(!$ntresult->RecordCount()){
        $error = 1;
        $query = "INSERT INTO {$dbprefix}tokens_{$surveyid}\n"
        . "(firstname, lastname, email, emailstatus, token)\n"
        . "VALUES (?, ?, ?, ?, ?)";
        $result = $connect->Execute($query, array($nombres,
$apellidos,"$cedula@unl.edu.ec",'OK', $cedula));
    }
    header("Location: index.php?sid=$surveyid&lang=es&token=$cedula");
}
}
else{
    if (!isset($surveyid)){
        $surveyid=returnnglobal('sid');
    }
    else
    {
        //This next line ensures that the $surveyid value is never anything
but a number.
        $surveyid=sanitize_int($surveyid);
    }
    $error==0;
    if ($error==0){
        #crear un mensaje de usuario y clave incorrercta
        echo "<script> alert (\"Usuario o Clave incorrectas. Por favor
vuelva intentarlo\"); </script>";
        echo "<script language=Javascript> location.href=\"http://localhost/
limesurvey/index.php?sid=21148&lang=es\"; </script>";
        die();
    }
}
} catch (Exception $e) {

    echo "<script> alert (\"ERROR EN LA CONEXIÓN\"); </script>";
    echo "<script language=Javascript> location.href=\"http://localhost/
limesurvey/index.php?sid=21148&lang=es\"; </script>";
    die();
}
?>
```

Fig.133. Script sgaauth.php



Así mismo se creó otro script `datos_estudiantes.php` con el cliente respectivo para invocar el método `sgaws_carreras_estudiante`, el cual pertenece a la categoría de información académica cuya dirección es <http://ws.unl.edu.ec/sgaws/wsacademica/soap/api.wsdl>, mediante el cual se puede obtener los nombres, apellidos y carreras que el estudiante curso en caso de que la autenticación de este se haya realizado efectivamente para poder ser presentados en la pantalla de la encuesta respectiva.

```
<?php
require_once(dirname(__FILE__).'/config.php');

function datos_est(){
    global $usersgaws, $passwordsgaws, $proxy, $puerto;
    $cliente = new SoapClient("http://ws.unl.edu.ec/sgaws/wsacademica/soap/
api.wsdl",array('login'=>$usersgaws,'password'=>$passwordsgaws));
    $parametros = array('cedula'=>$_GET["token"]);
    $servidor = $cliente->sgaws_carreras_estudiante($parametros);
    $resultado = json_decode($servidor->result);
    return $resultado;
}

$datos= datos_est();
$espacio = " ";
$nombre = strtoupper($datos[1]);
$apellido = strtoupper($datos[0]);
$c = $datos[3];
for($i=0;$i<count($c);$i++){
    $carrera=$c[$i];
    for ($j=0;$j<count($carrera);$j++){
        if($j==2){
            if($carrera[2]=='presencial'){
                $carreraprese=strtoupper($carrera[1]);

            }

        }
    }
}

print("<h3>UNIVERSIDAD NACIONAL DE LOJA </h3>
<p>Señor estudiante:</p>
Cédula: $datos[2] <br/>
Nombres: $nombre $espacio $apellido <br/>
Carrera:$carreraprese
<p>Lea atentamente cada pregunta y conteste con la mayor
veracidad.</p>");
?>
```

Fig.134. Script `datos_estudiantes.php`

Estos script se los integró posteriormente al sistema de encuesta para que uso del SGASWebServices sea realizado efectivamente.

Finalmente se procedió a activar la encuesta para que se pueda acceder y participar de

la misma, esto dando click en el botón , que es el encargado de activarla.

A continuación el desarrollo de la historia de usuario #2:

HISTORIA DE USUARIO		
<b>Nombre de Historia:</b> Autenticar estudiante a través del SGASWebServices		
<b>Nº:</b> 2	<b>Usuario:</b> Estudiante	
<b>Prioridad:</b> Alta	<b>Puntos(estimación):</b> 2semanas	<b>Iteración:</b> 3
<b>Descripción:</b> El estudiante ingresa a la página de la encuesta e ingresa su usuario y clave de su perfil de estudiantes del Sistema de Gestión Académica el sistema invoca al método sgaws_validar_estudiante del SGASWebServices para verificar datos ingresados, si los datos son correctos y existe conexión el sistema con la cedula ingresada invoca la método sgaws_carreras_estudiante para obtener dicha información, el sistema redirecciona a la página de bienvenida de la encuesta y presenta nombres, apellidos y cedula del estudiante e inicia la encuesta; caso contrario el si los datos no son los correctos o ya participo de la encuesta el sistema muestra tres tipos de mensajes de acuerdo al caso presentado como: los datos no son correctos, usted ya participo de la encuesta o no existe conexión.		
<b>Observaciones:</b>		

**Tabla. 37: Historia de Usuario #2 del Sistema de Encuestas**

### Diagrama de Actividades

Para el diseño del diagrama de actividades fue necesario tener claro el siguiente proceso:



- Ingresar a la página de la encuesta donde se presentará el formulario de autenticación
- Ingresar usuario y clave
- El sistema invoca el servicio al webservices con los datos ingresados en el formulario
- El sistema verifica conexión, si no existe conexión finaliza el proceso e indica que no existe conexión y en caso de que exista
- El sistema verifica si los datos son correctos, en caso de que si lo sean:
  - El sistema crea una entrada a la tabla de tokens de la base de datos tomando como token al usuario o cedula
  - Con la cedula del estudiante ingresada se invoca al servicio sgaws\_carreras\_estudiante del SGAWebservices para obtener nombres apellidos del estudiante
  - Verificar si la cedula se encuentra en la base de datos si es el caso no agregar los datos del estudiante a la base de datos, sino
  - Agregar datos del encuestado a la base de datos
  - Lime Survey redirecciona al usuario a la página de bienvenida
  - Se presentan los nombres, apellidos y cédula del estudiante en la página de bienvenida
  - Inicia la encuesta
- Si los datos nos son los correctos el sistema muestra un mensaje que indica que los datos ingresados son erróneos o que el estudiante ya participo de la encuesta.

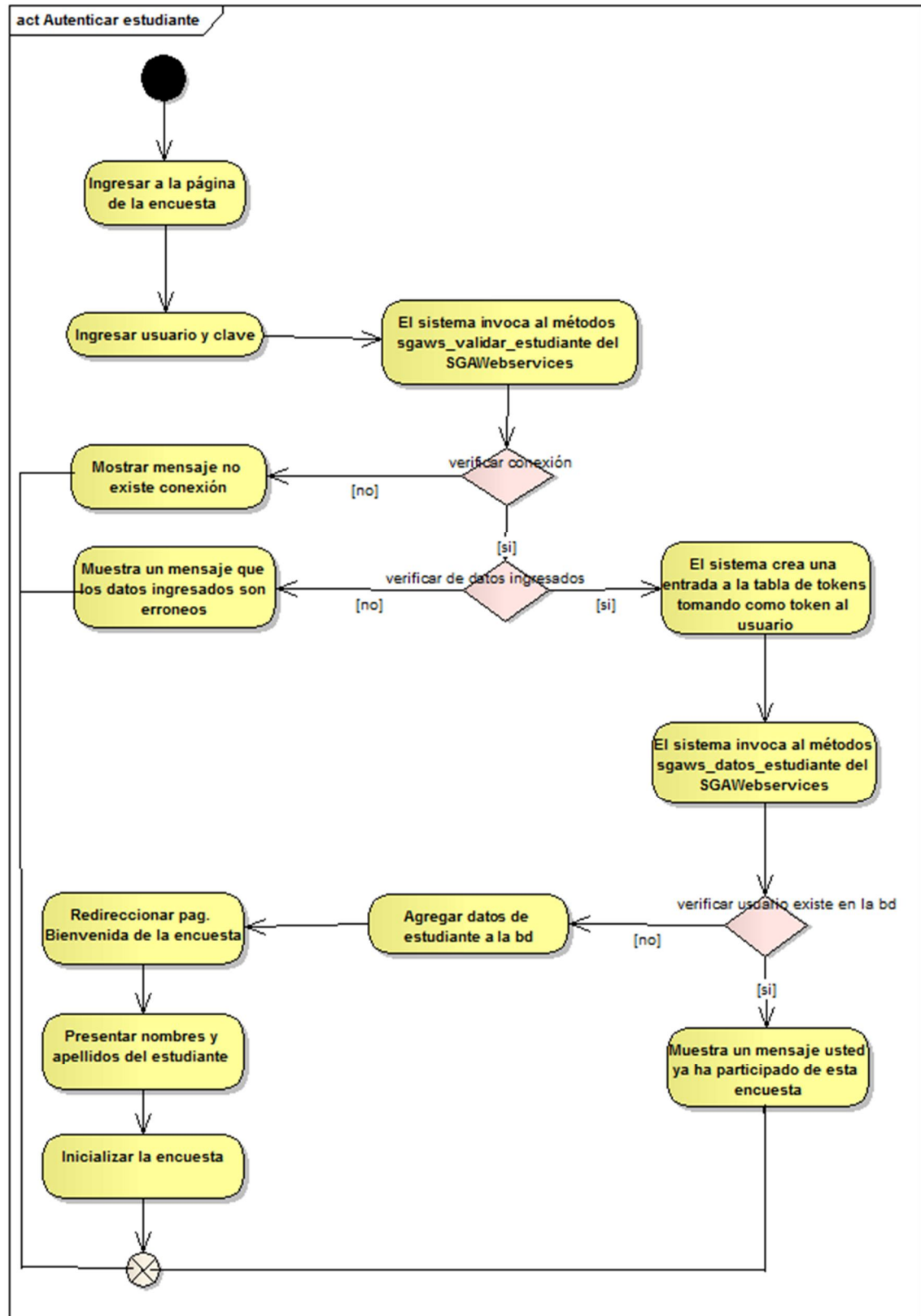


Fig.135: Diagrama de Actividades Autenticar Estudiante

Con el análisis y diseño realizado se llevo a cabo el siguiente proceso:

**a. Ingresar a la página de la encuesta donde se presentará el formulario de autenticación**

El estudiante ingresa al link <http://localhost/limesurvey/index.php?sid=21148&lang=es> que redirecciona a la página de la encuesta donde se muestra el formulario para autenticación del estudiante, como se observa en la siguiente figura:



**Fig.136: Formulario de Autenticación del Estudiante**

**b. Ingresar usuario y clave**

El estudiante ingresa su usuario y clave para que se realice la autenticación como se muestra en la siguiente figura:



**Fig.137: Autenticación del Estudiante**

Si no existiera conexión se muestra un mensaje indicado el problema. Por el contrario si existe conexión el sistema valido los datos, se muestra la página de

bienvenida con los datos del usuario autenticado e inicia la encuesta. Como se muestra en la siguiente figura:



**Fig.138: Página de Bienvenida de la Encuesta**

A continuación el desarrollo de la historia de usuario #3:

<b>HISTORIA DE USUARIO</b>		
<b>Nombre de Historia:</b> Responder encuesta		
<b>Nº:</b> 3	<b>Usuario:</b> Estudiante	
<b>Prioridad:</b> media	<b>Puntos(estimación):</b> 1 día	<b>Iteración:</b> 1
<p><b>Descripción:</b></p> <p>El estudiante inicia la encuesta contestando una a una las preguntas presentadas en cada sección de la encuesta, el estudiante debe contestar todas las preguntas de cada sección sino no podrá pasar a la siguiente debido a que el sistema le mostrará un mensaje indicando que debe contestar todas las preguntas, el estudiante irá interactuando con cada condición que tengan alguna preguntas, además el estudiante tendrá la opción de salir y reiniciar la encuesta en caso que así lo quisiera, de la misma forma podrá guardar sus respuestas hasta una sección dada para posteriormente continuar con la misma. Una vez que el estudiante haya concluido con la encuesta se mostrará un mensaje de agradecimiento por participar de la misma.</p>		
<b>Observaciones:</b>		

**Tabla. 38: Historia de Usuario #3 del Sistema de Encuestas**



## Diagrama de Actividades

Para el diseño del diagrama de actividades de esta historia de usuario fue necesario el siguiente proceso:

- El estudiante inicia la encuesta
- El estudiante contesta las preguntas en cada sección de la encuesta
- El sistema verifica si todas las preguntas de la sección fueron contestadas, si no fueron contestadas:
  - Mostrar un mensaje indicando que debe contestar todas
- Si contesta todas la preguntas continuar con la siguiente sección
  - Además el estudiante tendrá la opción de salir y reiniciar la encuesta es decir borrar todas las respuestas contestadas hasta la sección que desee
  - Así mismo podrá guardar sus respuestas hasta una sección dada para posteriormente continuar con la misma
- Finalmente una vez concluida la encuesta se mostrará un mensaje de agradecimiento por participar de la misma.

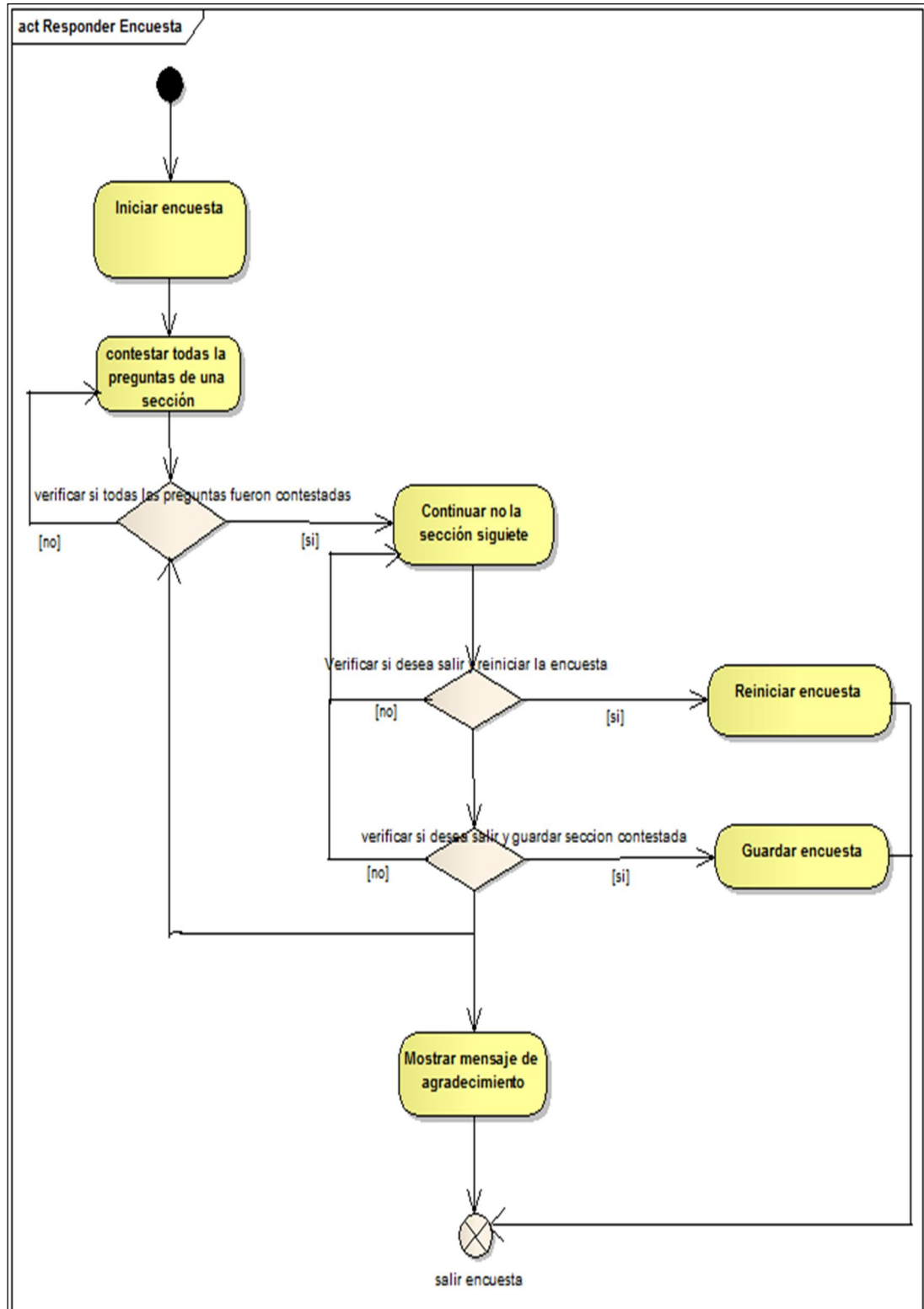


Fig.139: Diagrama de Actividades Responder Encuesta



El formulario de la encuesta se muestra en la siguiente figura:



**Fig.140: Formulario de la Encuesta**

### **7.3. INSTALACIÓN Y CONFIGURACION DEL SGAWEBSERVICES EN EL SERVIDOR**

Después de haber realizado las pruebas locales y verificar el correcto funcionamiento se realizó la configuración del servidor en donde se levantó el proyecto SGAWebServices, para esto se efectuaron las siguientes actividades.

#### **7.3.1. Crear un nuevo usuario con los permisos necesarios en el servidor donde se va a levantar el servicio.**

El procedimiento llevado a cabo para crear un nuevo usuario dentro de uno de los servidores por parte del encargado este proceso, siendo importante aclarar que el

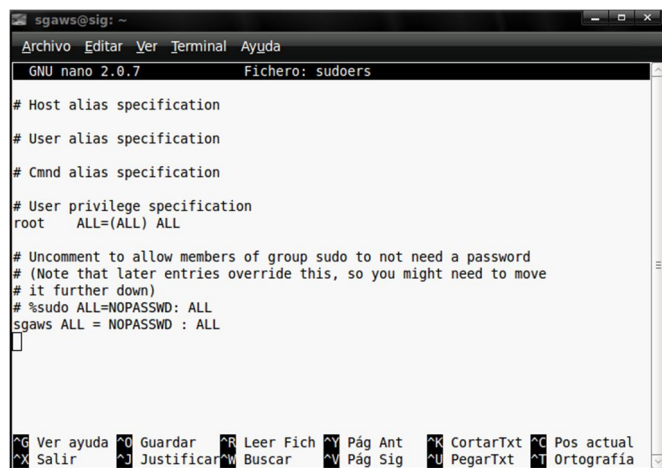
sistema operativo del servidor donde se levantó la aplicación fue Linux en la distribución de **DEBIAN 5.0**.

1. Ejecutar en consola el siguiente comando:

```
adduser sgaws
```

2. Crear un usuario con el nombre **sgaws** con su respectiva clave
3. Dar los permisos necesarios a este nuevo usuario

Con el usuario **sgaws** creado y con todos los permisos otorgados se realizaron las instalaciones y configuraciones que se requirieron para levantar los servicios web en este servidor, para lo cual se editó el archivo **sudoers** que se encuentra en **/etc/** con el editor **nano**, agregando en este al nuevo usuario con los permisos respectivos, como se muestra en la siguiente figura:



```
sgaws@sig: ~
Archivo Editar Ver Terminal Ayuda
GNU nano 2.0.7 Fichero: sudoers

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL

# Uncomment to allow members of group sudo to not need a password
# (Note that later entries override this, so you might need to move
# it further down)
# %sudo ALL=NOPASSWD: ALL
sgaws ALL = NOPASSWD : ALL

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```

**Fig.141: Ejemplo de edición de archivo sudoers**

### 7.3.2. Instalar y configurar los paquetes y programas requeridos en el servidor para poder levantar el WebServices.

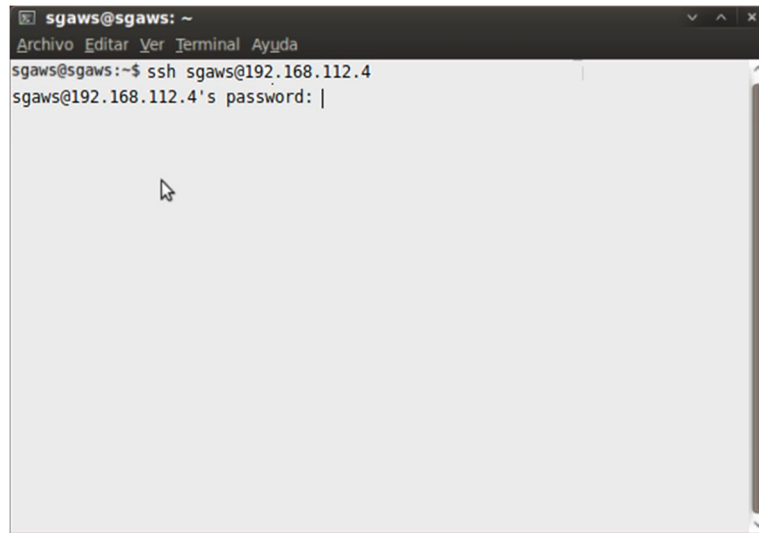
Luego de haber sido creado el usuario para levantar el SGAWebsservices, se accedió al máquina del servidor por medio del comando **ssh**, que es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente, el cual fue instalado en la maquina desde la que se accedió. El proceso fue el siguiente:

1. Ejecutar en consola en siguiente comando para instalar ssh en la máquina local:

```
apt-get install ssh
```

2. Para acceder al servidor desde la máquina local. Ejecutar en el terminal lo siguiente:

```
sshsgaws@192.168.112.4
```



**Fig.142: Ejemplo de acceso a un servidor en forma remota**

En donde

- ssh es el protocolo por el cual se va a ingresa
- sgaws es el nombre del usuario con el cual se va a ingresar al servidor
- 192.168.112.4 es la ip con la que se encuentra el servidor
- Password es la clave del Usuario sgaws creada

Una vez dentro del servidor se instalaron y configuraron los programas necesarios, todo esto como super-usuario del sistema.

Las instalaciones de programas y módulos necesarios fueron los siguientes:

- Instalación del Sistema de Control de Versiones(subversion):

```
apt-get install subversion subversion-tools
```

- Instalación de Apache:

```
apt-get install apache2
```

- Instalación de Mysql:

```
apt-get install mysql-server mysql-client mysql-admin  
mysql-query-browser
```

- Instalación del módulo para autenticación con mysql, que servirá para la autenticación básica con apache y el sgaws.

```
apt-get install libapache2-mod-auth-mysql
```

- Activación del módulo de mysql para autenticación con apache

```
apache2a2enmod auth_mysql
```

- Instalación del paquete que servirá para la conexión con la base de datos del SGA

```
apt-get install postgresql-server-dev-8.3
```

- Instalación de paquete Json

```
apt-get install python-json
```

- Instalación de virtualenv para crear el entorno virtual

```
sudo easy_install virtualenv
```

### 7.3.3. Creación de entorno virtual e instalación de programas dentro del mismo.

Ya instalados los programas requeridos en el servidor se creó el entorno virtual en donde se instaló todos los programas necesarios para poder levantar la aplicación dentro del mismo, para lo cual se llevó a cabo el siguiente proceso.

1. En el directorio **home/sgaws/** se creó el entorno virtual denominado **webservices**:

```
virtualenv --no-site-packageswebservices
```

2. Se activó el entorno virtual recién creado:

- Primero fue necesario ubicarse dentro del entorno virtual que se creó

```
cd webservices
```

- Finalmente se activó el entorno virtual ejecutando el siguiente comando:

```
source bin/actíivate
```

3. Se realizó la instalación de los paquetes necesarios dentro del entorno virtual para poder levantar la aplicación, las mismas que fueron:

- Instalación de Turbogears 2.0 para lo cual se lo realizó con el siguiente comando:

```
easy_install -i http://www.turbogears.org/2.0/downloads/  
current/index tg.devtools
```

- Instalación del paquete TGWebservices ejecutando:

```
easy_install TGWebServices 2.0
```

- Instalación del paquete SQLAlchemy-0.5.8

```
easy_install SQLAlchemy-0.5.8
```

- Instalación del paquete psycopg2

```
easy_install psycopg2
```

### 7.3.4. Bajar del repositorio la aplicación y configurar archivos específicos para poder levantar el proyecto.

Seguidamente se procedió a bajar del repositorio de proyectos de la Unidad de Desarrollo de Software de la Universidad Nacional de Loja cuya dirección es *http://192.168.112.3/svn/ddsunl/proyectos/sgaws/trunk* la aplicación, esto se realizó llevando a cabo los siguientes pasos:

1. Crear el directorio donde se va a bajar el proyecto de la siguiente manera:

```
mkdir /home/sgaws/webservices/sgaws
```

2. Bajar el código del repositorio en el entorno virtual creado en **/home/sgaws/webservices/sgaws** ejecutando:

```
svn co http://192.168.112.3/svn/ddsunl/proyectos/sgaws/trunk .
```

3. Copiar el archivo `development.ini` que se encuentra dentro del directorio `install` del proyecto:

```
cp/home/sgaws/webservices/sgaws/install/development.ini  
/home/sgaws/webservices/sgaws/ .
```

4. Ejecutar el siguiente comando para activar los paquetes de `turbogears` instalados:

```
python setup.py develop
```

5. Ejecutar el script `setupws.py` que se encuentra dentro del directorio `install`:

```
python setupws.py install
```

El script permite crear la base de datos que servirá para guardar los usuarios que tendrán acceso al SGAWebservices. La creación de usuario requiere lo siguiente:

- Entrar al gestor de base de datos mysql ejecutando:

```
mysql -u root -p
```

- Crear la base de datos usuario:

```
create database usuario;
```

- Declarar que la base de datos creada es sobre donde se va a trabajar

```
use usuario;
```

- Crear la tabla user para almacenar los usuarios del SGAWwbservices:

```
create table user (username char(30) NOT NULL, password char(30), primary key (username));
```

- Crear y dar los permisos necesarios al usuario mysql que se ha creado

```
GRANT ALL ON usuario.*to sgaws@localhost identified by 'sgawebsservices';  
flush privileges;
```

6. Copiar el archivo sgawsproxy ubicado en el directorio install mediante el siguiente comando al directorio **/etc/apache2/sites-available/**:

```
cp /home/sgaws/webservices/sgaws/install/sgawsproxy /etc/  
apache2/sites-available/ .
```

7. Modificar las rutas del archivo sgawsproxy de acuerdo a la ubicación donde se encuentre el proyecto.

En la siguiente imagen se observa la configuración que debe tener el archivo sgawsproxy, indicando el directorio correcto en donde se encuentra el proyecto, además el puerto en el que se levanta, en este caso en el puerto 83.

```
Listen 83
NameVirtualHost *:83
<VirtualHost *:83>
    ServerAdmin webmaster@localhost
    DocumentRoot /home/sgaws/websservice/sgaws/sgaws/public
    CustomLog /var/log/apache2/sgaws.log combined

    <Directory /home/sgaws/websservice/sgaws/sgaws/public>
        Order allow,deny
        Allow from all
    </Directory>

    ProxyPass /sgaws/ http://127.0.0.1:8089/
    ProxyPassReverse /sgaws/ http://127.0.0.1:8089/
    <Location /sgaws>
        AuthType Basic
        AuthName "SGAWEBSSERVICES"
        AuthBasicAuthoritative Off
        AuthUserFile /dev/null
        AuthGroupFile /dev/null
        Auth_MYSQL on
        AuthMySQL_Authoritative Off
        AuthMySQL_DB usuario
        AuthMySQL_Host localhost
        AuthMySQL_User sgaws
        AuthMySQL_Password sgawebsservices
        AuthMySQL_Password_Table user
        AuthMySQL_Username_Field username
        AuthMySQL_Password_Field password
        AuthMySQL_Empty_Passwords Off
        AuthMySQL_Encryption_Types Plaintext
        AllowOverride AuthConfig
        Order allow,deny
        Allow from all
        Options All
        <Limit GET POST PUT
            /home/sgaws/websservice/sgaws/sgaws>
            Require valid-user
        </Limit>
    </Location>
</VirtualHost>
```

**Fig.143: Configuración del Archivo sgawsproxy en el servidor**

8. Activar el modulo proxy con el siguiente comando

```
a2enmod proxy_http
```

9. Finalmente activar el archivo de configuración sgawsproxy

```
a2ensite sgawsproxy
```



### 7.3.5. Levantar el proyecto SGAServices

Finalmente para que el proceso esté completo, se puso en funcionamiento el proyecto levantando SGAServices de la siguiente forma:

1. Se reinició el servidor apache para establecer las configuraciones realizadas anteriormente ejecutando el siguiente comando:

```
sudo /etc/init.d/apache2 restart
```

2. Por último se levantó el proyecto dentro del entorno virtual que se encuentra en el directorio `/home/sgaws/webservices/sgaws/` ejecutando lo siguiente:

```
paster serve development.ini
```

Para verificar que el proyecto se ha levantado correctamente en un navegador web se colocó la dirección del servidor y el proxy configurado, como se muestra a continuación:

```
http://192.168.112.3:83
```

Posteriormente se tramitó un dominio público para que el SGAServices pueda ser accedido a través del internet por los usuarios, cuya dirección es:

```
http://ws.unl.edu.ec
```

### 7.3.6. Crear usuarios del SGAServices

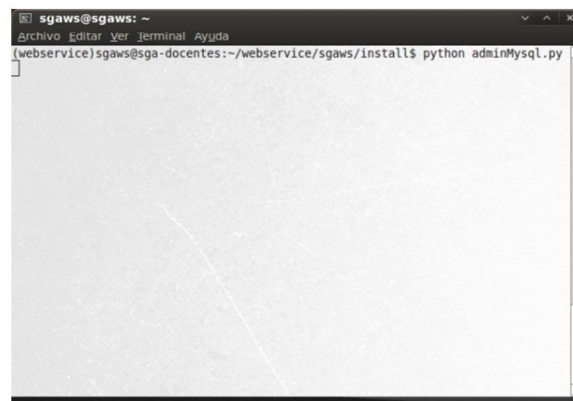
Para la creación de usuarios del sgaws se ha realizado un script en Python denominado **adminMySQL.py** (Ver Anexo # 6), el cual realiza concepción, verificación e ingreso de usuarios a la base de datos creada en mysql para este objetivo. Este archivo

inicialmente está ubicado en la carpeta install del proyecto descargado desde el repositorio.

Los pasos que se llevan a cabo para la creación de los usuarios mediante la ejecución del script son los siguientes:

- Ejecutar el script mediante el siguiente comando:

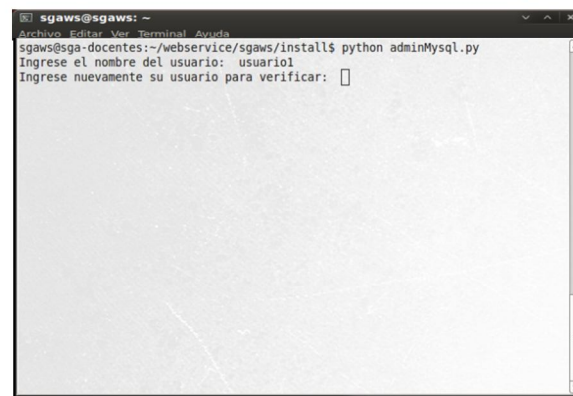
```
python adminMysql.py
```



**Fig.144. Ejecución del script adminMysql.py**

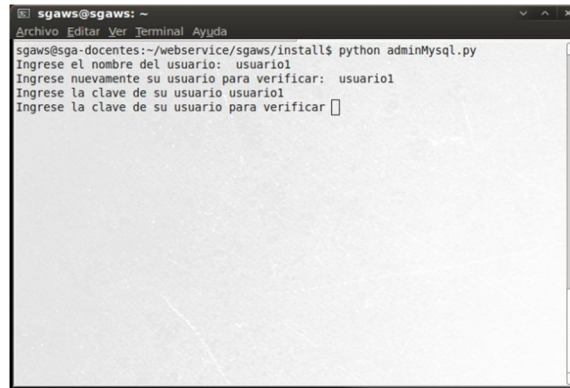
Este comando permite que desde el Shell del sistema se ingrese el nombre del usuario con su respectiva contraseña

- Ingreso del nombre del usuario con su verificación respectiva.



**Fig.145. Ingreso de nombre usuario**

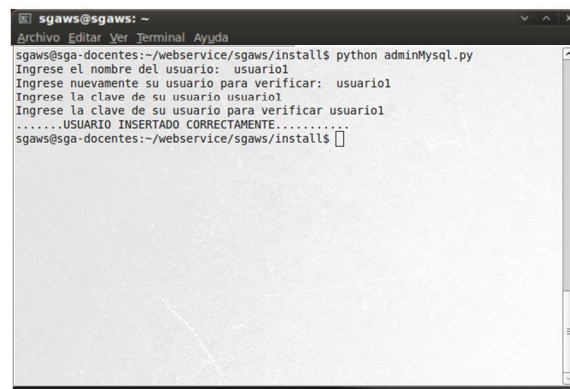
- Ingreso de la clave con su respectiva verificación



```
sgaws@sgaws: ~  
Archivo Editar Ver Terminal Ayuda  
sgaws@sga-docentes:~/webservice/sgaws/install$ python adminMySQL.py  
Ingrese el nombre del usuario: usuario1  
Ingrese nuevamente su usuario para verificar: usuario1  
Ingrese la clave de su usuario usuario1  
Ingrese la clave de su usuario para verificar: 
```

Fig.146. Ingreso de la clave del usuario

- Finalmente el usuario se crea con sus respectivos datos



```
sgaws@sgaws: ~  
Archivo Editar Ver Terminal Ayuda  
sgaws@sga-docentes:~/webservice/sgaws/install$ python adminMySQL.py  
Ingrese el nombre del usuario: usuario1  
Ingrese nuevamente su usuario para verificar: usuario1  
Ingrese la clave de su usuario usuario1  
.....USUARIO INSERTADO CORRECTAMENTE.....  
sgaws@sga-docentes:~/webservice/sgaws/install$ 
```

Fig.147. Creación correcta del usuario

## 7.4. PRUEBAS Y VALIDACIÓN

Una de las etapas más importantes dentro de las fases de desarrollo de software son las pruebas de validación, que fueron aplicadas para verificar el rendimiento de la aplicación, estas permitieron descubrir errores existentes y realizar las correcciones respectivas. Esta fase también permitió verificar la facilidad para poder administrar el sistema por parte del Jefe del Departamento de Desarrollo de Software de la Universidad Nacional de Loja. Las pruebas fueron aplicadas en dos niveles de personal, 1 encuesta fue realizada al administrador del Sistema de Gestión Académica y 15 encuestas también se efectuaron a los posibles usuarios del SGAWebServices. Las encuestas aplicadas fueron las siguientes:

### 7.4.1. Prueba de Validación Aplicadas al Sistema

La prueba de validación fue realizada por el Jefe del Departamento de Desarrollo de Software de la Universidad Nacional de Loja el Ing. Patricio Valarezo, lo que permitió comprobar que el sistema desarrollado cumple con los requisitos establecidos, la viabilidad de mantenimiento, así como también la facilidad para administrar el mismo.

#### Análisis de Prueba

Culminado con el periodo de pruebas se tabuló los datos obtenidos de la encuesta aplicada al Ing. Patricio Valarezo (**ver Anexo# 4**), obteniendo los siguientes resultados:

#### Encuesta realizada al Ing. Patricio Valarezo

**1. Considera usted que la tecnología que se ha utilizado para el desarrollo de la biblioteca de métodos (TurboGears2.0, TGWebServices) es:**

Excelente ( x )

#### Interpretación de los Resultados:

Una vez aplicada la encuesta al encargado del SGA, se determinó que la tecnología utilizada en el desarrollo de la biblioteca de métodos es excelente, ya que se lo realizó con herramientas dinámicas para los desarrollos web, mismos que son fáciles de manejar y de libre distribución.

**2. Considera usted que el proceso de administración del SGAWebServices (ejemplo: creación de nuevos métodos, modificación de los métodos, etc ) es:**

Fácil ( x )

**Interpretación de los Resultados:**

Una vez aplicada la encuesta al administrador del SGA, se determinó que la administración del SGAWebServices es fácil, esto debido a que se ha implementado el sistema con la misma tecnología y herramientas utilizadas por parte de los miembros del Departamento de Software de la Universidad Nacional de Loja, facilitando de esta manera un mejor mantenimiento del mismo.

**3. La creación y manipulación de los usuarios del SGAWebServices con mysql es:**

Fácil ( x )

**Interpretación de los Resultados:**

Después de aplicada la encuesta al administrador del SGA, se determinó que la creación y manipulación de los usuarios que tendrán acceso al SGAWebServices con mysql es fácil, ya que se ha proporcionado un script (**ver Anexo# 6**) que realiza de manera interactiva la creación y verificación de los usuarios, para que estos puedan ser registrados en la base de datos que almacenará a los diferentes consumidores del SGAWebServices.

**4. La información obtenida del SGAWebServices para los usuarios del Sistema consideras usted que es:**

Importante ( x )

**Interpretación de los Resultados:**

Mediante la encuesta aplicada al administrador del SGA, se determinó que la información obtenida mediante el SGAWebServices para los usuarios del Sistema es importante, ya que se ha recopilado mediante la biblioteca de métodos los datos más necesarios para que quienes necesiten hacer uso de éste puedan obtenerlos y dar a conocer a través de la implementación de sus proyectos respectivos, evitando de esta

manera redundar datos ya contenidos en el Sistema de Gestión Académica en otras aplicaciones.

**5. Considera usted que el sistema cumple con la finalidad con que fue planteado inicialmente?:**

Si ( x )

**Interpretación de los Resultados:**

Con el resultado obtenido después de haber sido aplicada la encuesta al administrador del SGA, se determinó que la este proyecto si cumple con los la finalidad con se lo planteó inicialmente, ya que se han cubierto todos los requerimientos que al iniciar la implementación fueron solicitados por parte de los miembros del Sistema de Gestión Académica así como también de los posibles usuarios de esta aplicación.

**6. Considera usted que la seguridad aplicada para el acceso al WebServices es:**

Excelente ( x )

**Interpretación de los Resultados:**

Una vez que se aplicó la encuesta al administrador del SGA, se ha podido establecer que la seguridad aplicada para el acceso al WebServices es excelente, porque este proyecto solo ofrece el servicio de consultas al Sistema de Gestión Académica, estando este restringido para realizar manipulaciones a la base de datos del sistema, es por ello que la seguridad mediante el módulo de autenticación básica de apache es lo más conveniente para este caso.

**7.4.2. Prueba de Usabilidad Aplicadas al Sistema**

Las pruebas de usabilidad fueron realizadas por los posibles usuarios del SGAWebServices, estos fueron tanto tesistas de la Carrera de Ingeniería en Sistema que requieren para el desarrollo de su proyecto datos académicos de la universidad, así como también fue dirigida a los miembros del Sistema de Gestión Académica de la Universidad Nacional de Loja que se encargan del desarrollo de software. Esta prueba permitió comprobar y medir la interacción de los usuarios con el software desarrollado, además ayudó para aplicar nuevas optimizaciones al software.

### Análisis de Prueba

Luego de haber culminado con el periodo de pruebas se procedió a tabular los datos obtenidos de las encuestas (ver Anexo# 5) aplicadas a 15 personas entre desarrolladores y testistas, los resultados obtenidos fueron los siguientes:

### Encuesta aplicados a los posibles usuarios del SGAWebServices

#### 1. Considera usted que la información obtenida del Sistema de Gestión Académica mediante el SGAWebServices es:

- Importante ( )  
Normal ( )  
Regular ( )

#### Resultados:

OPCIONES	RESULTADOS
Importante	15
Normal	0
Regular	0
<b>Total</b>	<b>15</b>

Tabla. 38: Resultados Pregunta#1 para Prueba de Usabilidad

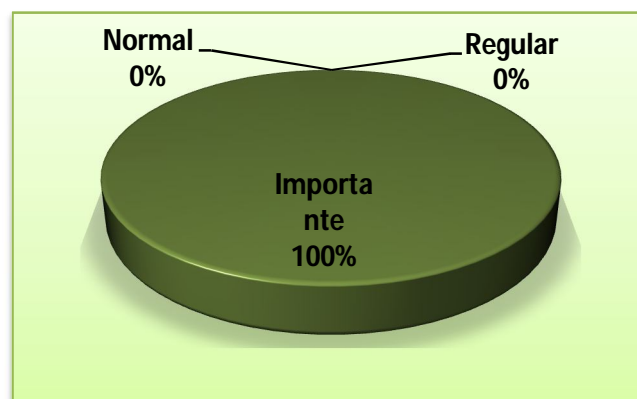


Fig.148: Resultados Pregunta#1 para Prueba de Usabilidad

**Interpretación de Resultados:**

El 100% de los encuestados consideran que la información obtenida del Sistema de Gestión Académica mediante el SGAWebServices es importante, ya que esta aplicación retorna: información académica, institucional, estadística y personal, además también permite realizar la validación de usuarios del SGA, datos que son de mucha importancia para los usuarios que desarrollan aplicaciones que requieren información de la Universidad evitando con esto redundar los datos ya registrados.

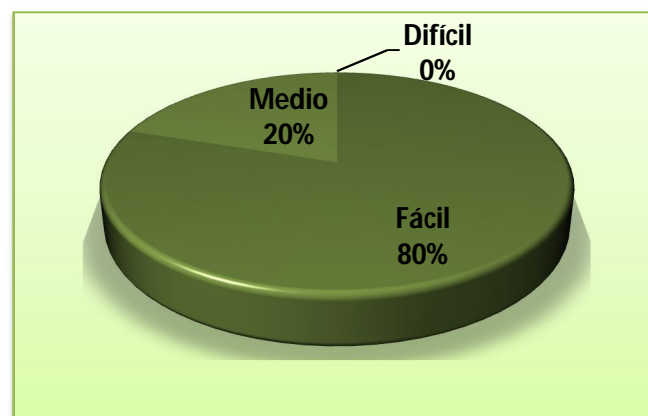
**2. Considera usted que el uso del SGAWebServices es:**

- Fácil ( )  
Medio ( )  
Difícil ( )

**Resultados:**

OPCIONES	RESULTADOS
Fácil	13
Medio	2
Regular	0
<b>Total</b>	<b>15</b>

**Tabla. 39: Resultados Pregunta#2 para Prueba de Usabilidad**



**Fig.149: Resultados Pregunta#2 para Prueba de Usabilidad**



**Interpretación de Resultados:**

El 80% de los encuestados consideran que el uso del SGAWebServices es fácil, mientras que el 20% de estos manifiestan que el uso del software es medio, revelando de esta forma que la mayoría de los usuarios establecieron que usar la aplicación es sencillo, debido a que es factible y se acopla para poder utilizar con cualquier lenguaje de programación en el cual desarrollen su proyecto.

**3.Cuál de los método del SGAWebServices utilizaría usted?:**

sgaws_lista_areas	( )	sgaws_carreras_estudiante	( )
sgaws_datos_area	( )	sgaws_reporte_matricula	( )
sgaws_carreras_area	( )	sgaws_nmatriculados_oferta	( )
sgaws_datos_carreras	( )	sgaws_nmatriculados_area	( )
sgaws_modulos_carrera	( )	sgaws_nmatriculados_carrera	( )
sgaws_paralelos_carrera	( )	sgaws_nmatriculados_modulo	( )
sgaws_validar_docente	( )	sgaws_nmatriculados_paralelo	( )
sgaws_validar_estudiante	( )	sgaws_naprobados_oferta	( )
sgaws_datos_docente	( )	sgaws_naprobados_area	( )
sgaws_datos_estudiante	( )	sgaws_naprobados_carrera	( )
sgaws_datos_usuario	( )	sgaws_naprobados_modulo	( )
sgaws_periodos_lectivos	( )	sgaws_naprobados_paralelo	( )
sgaws_ofertas_academicas	( )	sgaws_nreprobados_oferta	( )
sgaws_fechas_matriculasa	( )	sgaws_nreprobados_area	( )
sgaws_estadoestudiantes_paralelo	( )	sgaws_nreprobados_carrera	( )
sgaws_plan_estudio	( )	sgaws_nreprobados_modulo	( )
sgaws_carga_horaria_docente	( )	sgaws_nreprobados_paralelo	( )
sgaws_notas_estudiante	( )		

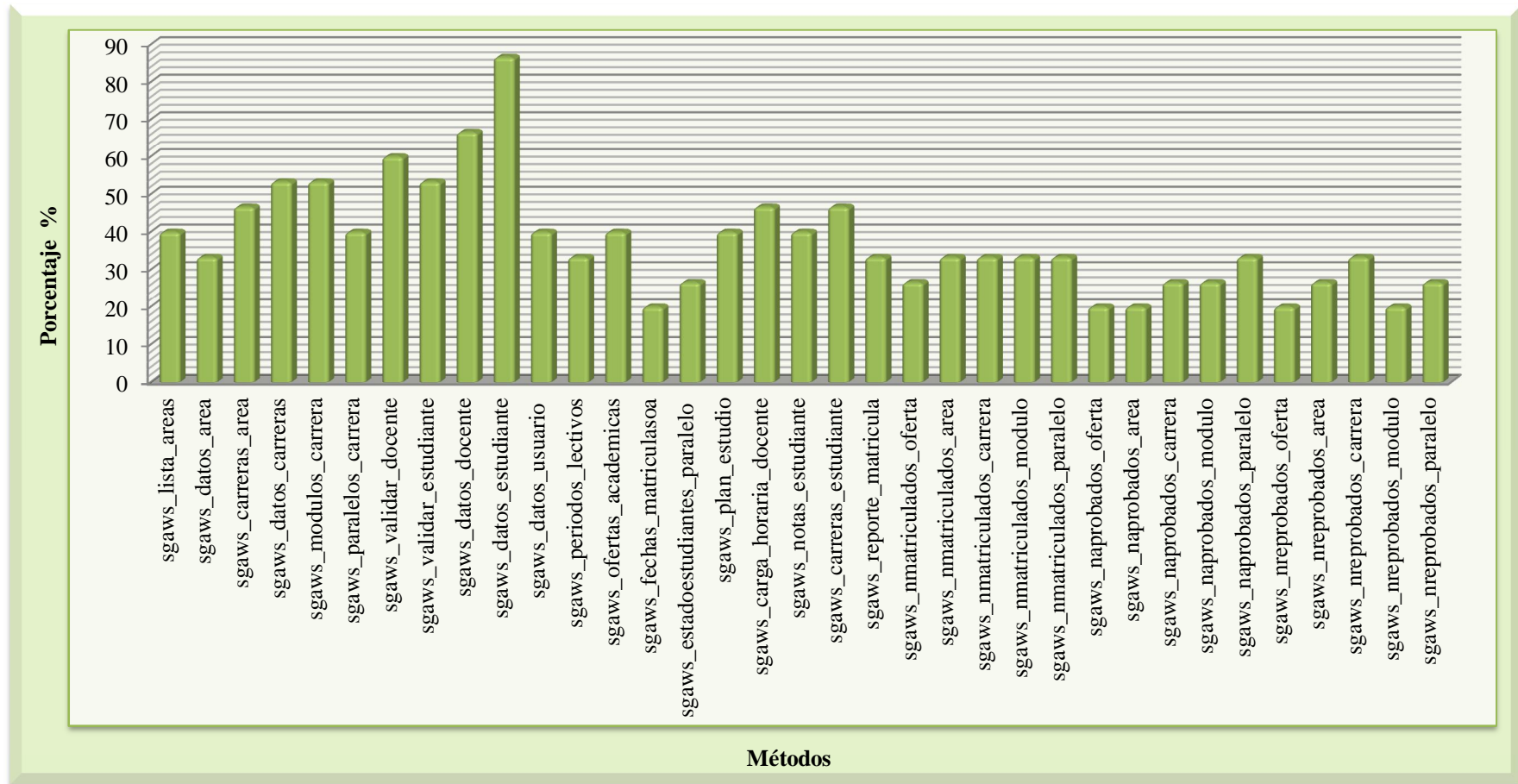


Fig.150: Resultados Pregunta#3 para Prueba de Usabilidad

### Interpretación de Resultados:

El 40% de los encuestados utilizarían el método `sgaws_lista_areas`, el cual retorna las áreas de la universidad.

El 33.3% de los encuestados utilizarían el método `sgaws_datos_area`, el cual retorna los datos de un área de la universidad.

El 46.7% de los encuestados utilizarían el método `sgaws_carreras_area`, el cual retorna los datos de las carreras de un área de la universidad.

El 55.3% de los encuestados utilizarían el método `sgaws_datos_carreras`, el cual retorna los datos todas las carreras de la universidad.

El 55.3% de los encuestados utilizarían el método `sgaws_modulos_carrera`, el cual retorna los módulos de una carrera.

El 40% de los encuestados utilizarían el método `sgaws_paralelos_carrera`, el cual retorna los paralelos de una carrera.

El 60% de los encuestados utilizarían el método `sgaws_validar_docente`, el cual permite verificar si la clave y la cedula de un docente se encuentra registrado en el SGA.

El 55.3% de los encuestados utilizarían el método `sgaws_validar_estudiante`, el cual permite verificar si la clave y la cedula de un estudiante se encuentra registrado en el SGA.

El 66.7% de los encuestados utilizarían el método `sgaws_datos_docente`, el cual retorna los datos personales de un docente.

El 86.7% de los encuestados utilizarían el método `sgaws_datos_estudiante`, el cual retorna los datos personales de un estudiante.

El 40% de los encuestados utilizarían el método `sgaws_datos_usuario`, el cual verifica si el usuario es un docente o estudiante y retorna los datos personales de este.

El 33.3% de los encuestados utilizarían el método `sgaws_periodos_lectivos`, el cual retorna los periodos lectivos de la universidad.

El 40% de los encuestados utilizarían el método `sgaws_ofertas_academicas`, el cual retorna las ofertas académicas que se han ofertado en un periodo lectivo.

El 20% de los encuestados utilizarían el método `sgaws_fecha_matriculacion`, el cual retorna las fechas de matrículas de una oferta académica.

El 26.7% de los encuestados utilizarían el método `sgaws_estadoestudiantes_paralelo`, el cual retorna el estado de los estudiantes de un paralelo.



El 40% de los encuestados utilizarían el método `sgaws_plan_estudio`, el cual retorna el plan de estudio de un paralelo.

El 46.7% de los encuestados utilizarían el método `sgaws_carga_horaria_docente`, el cual retorna la carga horaria de un docente en una oferta académica.

El 40% de los encuestados utilizarían el método `sgaws_notas_estudiante`, el cual retorna las notas de un estudiante en una oferta académica.

El 46.7% de los encuestados utilizarían el método `sgaws_carreras_estudiante`, el cual retorna las carreras que cursa el estudiante.

El 33.3% de los encuestados utilizarían el método `sgaws_reporte_matricula`, el cual retorna el reporte de la matrícula de un estudiante en una determinada oferta.

El 26.7% de los encuestados utilizarían el método `sgaws_nmatriculados_oferta`, el cual retorna el número de estudiante matriculados en una oferta académica.

El 33.3% de los encuestados utilizarían el método `sgaws_nmatriculados_area`, el cual retorna el número de estudiante matriculados en un área durante una oferta académica.

El 33.3% de los encuestados utilizarían el método `sgaws_nmatriculados_carrera`, el cual retorna el número de estudiante matriculados en una carrera durante una oferta académica.

El 33.3% de los encuestados utilizarían el método `sgaws_nmatriculados_modulo`, el cual retorna el número de estudiante matriculados en un módulo durante una oferta académica.

El 33.3% de los encuestados utilizarían el método `sgaws_nmatriculados_paralelo`, el cual retorna el número de estudiante matriculados en un paralelo durante una oferta académica.

El 20% de los encuestados utilizarían el método `sgaws_naprobados_oferta`, el cual retorna el número de estudiante aprobados en una oferta académica.

El 20% de los encuestados utilizarían el método `sgaws_naprobados_area`, el cual retorna el número de estudiante aprobados en un área durante una oferta académica.

El 26.7% de los encuestados utilizarían el método `sgaws_naprobados_carrera`, el cual retorna el número de estudiante aprobados en una carrera durante una oferta académica.

El 26.7% de los encuestados utilizarían el método `sgaws_naprobados_modulo`, el cual retorna el número de estudiante aprobados en un módulo durante una oferta académica.

El 33.3% de los encuestados utilizarían el método `sgaws_naprobados_paralelo`, el cual retorna el número de estudiante aprobados en un paralelo durante una oferta académica.

El 20% de los encuestados utilizarían el método `sgaws_nreprobados_oferta`, el cual retorna el número de estudiante reprobados en una oferta académica.

El 26.7% de los encuestados utilizarían el método `sgaws_nreprobados_area`, el cual retorna el número de estudiante reprobados en un área durante una oferta académica.

El 33.3% de los encuestados utilizarían el método `sgaws_nreprobados_carrera`, el cual retorna el número de estudiante reprobados en una carrera durante una oferta académica.

El 20% de los encuestados utilizarían el método `sgaws_nreprobados_modulo`, el cual retorna el número de estudiante reprobados en un módulo durante una oferta académica.

El 26.7% de los encuestados utilizarían el método `sgaws_nreprobados_paralelo`, el cual retorna el número de estudiante reprobados en un paralelo durante una oferta académica.

Este análisis permitió concluir que el método más utilizado por parte de los usuarios sería el `sgaws_datos_estudiantes`, esto por la razón de que la mayoría de los desarrolladores necesitan trabajar con datos de estudiantes cuando implementan sus propios sistemas.

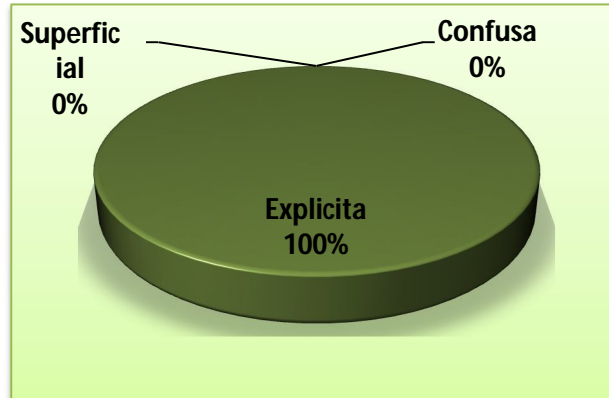
#### 4. La documentación que indica la utilización de los métodos del SGASWebServices es:

- Explicita ( )
- Superficial ( )
- Confusa ( )

#### Resultados:

OPCIONES	RESULTADOS
Explicita	15
Superficial	0
Confusa	0
<b>Total</b>	<b>15</b>

**Tabla. 40: Resultados Pregunta#4 para Prueba de Usabilidad**



**Fig.151: Resultados Pregunta#4 para Prueba de Usabilidad**

#### **Interpretación de Resultados:**

El 100% de los encuestados consideran que la documentación que indica la utilización de los métodos del SGAWebServices es explícita, ya que se ha proporcionado páginas web en donde se detallan minuciosamente cada uno de los métodos que contiene el sistema, así como también la forma de utilizarlos a estos.

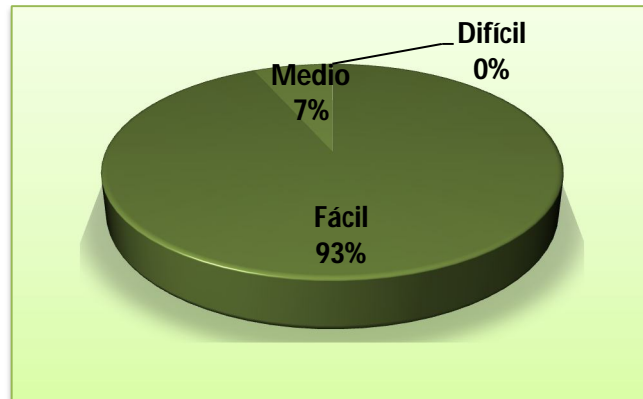
#### **5. Considera usted que el acceso al SGAWebServices es:**

- Fácil ( )  
Medio ( )  
Difícil ( )

#### **Resultados:**

OPCIONES	RESULTADOS
Fácil	13
Medio	2
Regular	0
<b>Total</b>	<b>15</b>

**Tabla. 41: Resultados Pregunta#5 para Prueba de Usabilidad**



**Fig.152: Resultados Pregunta#5 para Prueba de Usabilidad**

#### **Interpretación de los Resultados:**

El 93% de los encuestados consideran que el acceso al SGAWebServices es fácil, mientras que el 7% de estos expresan que acceder a este servicio tiene un nivel medio, lo cual manifiesta que el módulo de autenticación básica de apache utilizado en este sistema es relativamente simple, debido a que solo se requiere de un password y clave para poder ingresar a esta aplicación.



# VALORACIÓN TÉCNICA ECONÓMICA AMBIENTAL



## 8. VALORACIÓN TÉCNICO-ECONÓMICA-AMBIENTAL

Culminado el proyecto y puesto en producción de manera satisfactoria, acatando todos los requerimientos planteados inicialmente se dio solución a los problemas presentados, se procedió a realizar la valoración del mismo, dentro de lo cual cabe destacar principalmente las herramientas que se utilizaron para el desarrollo del software son de libre distribución, razón por la cual fueron fáciles de adquirirlas e instalarlas. En cuanto a los costos de movilización, comunicación, materiales de oficina, entre otros, es apropiado señalar que estos fueron cubiertos por las autoras del proyecto. El costo final del Sistema SGAWebServices se detalla a continuación mediante las siguientes tablas:

### 8.1. RECURSOS HUMANOS

Recursos Humanos	Cantidad	Horas c/u	Costo/ Hora	Costo Total
Director de Tesis	----	-----	-----	----
Desarrolladores • <i>Daysi Ordóñez</i> • <i>Silvana Pacheco</i>	2	1000	\$4.00	\$8000.00
<b>TOTAL</b>				<b>\$8000.00</b>

Tabla. 42: Recursos Humanos

### 8.2. RECURSOS MATERIALES

Materiales	Cantidad	Costo Unitario	Costo Total
Resma de Papel	5	\$3.00	\$15.00
Cartuchos de tinta.	5	\$5.00	\$25.00
Empastados	4	\$8.00	\$32.00
Copias	2000	\$0.02	\$40.00
Suministros de Oficina	---	---	\$30.00
Anillados	8	\$1.00	\$8.00
CD	5	\$1.00	\$5.00

Flash Memory (1GB)	2	\$7.00	\$14.00
<b>TOTAL</b>			<b>\$169.00</b>

Tabla. 43: Recursos Materiales

### 8.3. RECURSOS TÉCNICOS

Comunicaciones	Horas	Costo/Hora	Total
Internet	100	\$1.00	\$100.00
Llamadas telefónicas	25	---	\$50.00
<b>TOTAL</b>			<b>\$150.00</b>

Tabla. 44: Recursos Técnicos

### 8.4. RECURSOS TECNOLÓGICOS

Recursos Tecnológicos	Costo Unitario		Costo Total			
<b>SOFTWARE</b>						
Python 2.5	Gratuito		\$0.00			
PHP 5.2.11	Gratuito		\$0.00			
XML	Gratuito		\$0.00			
Open Office V.3.0	Gratuito		\$0.00			
Linux Ubuntu 9.0.1	Gratuito		\$0.00			
Apache Tomcat	Gratuito		\$0.00			
<b>TOTAL</b>			<b>\$0.00</b>			
Recursos Tecnológicos: <i>Comunicaciones</i>	Cantidad	Horas	Hojas	Costo		Costo Total
				Hora	Hoja	
Computadores	2	924		0.50		\$462.00
Impresora	1		2500		0.05	\$125.00
<b>TOTAL</b>						<b>\$587.00</b>

Tabla. 45: Recursos Tecnológicos

**8.5. RESUMEN DEL PRESUPUESTO**

Recursos Humanos	\$8000.00
Recursos Materiales	\$169.00
Recursos Técnicos	\$150.00
Recursos Tecnológicos	\$587.00
<b>SUBTOTAL</b>	<b>\$8906.00</b>
Imprevistos 10 %	\$890.60
<b>TOTAL PROYECTO</b>	<b>\$9796.60</b>

**Tabla. 46: Resumen Presupuesto**



# CONCLUSIONES

## 9. CONCLUSIONES

Las conclusiones que se derivan del presente trabajo desarrollado, se detallan a continuación:

- La metodología, técnicas, herramientas y estudio minucioso del Sistema de Gestión Académica de la Universidad Nacional de Loja y de servicios web contribuye de manera firme a la definición de los requerimientos del sistema, debido a que se puede aprovechar las funcionalidades e información contenida en el mismo de forma fiable, funcional y acoplada a las necesidades del usuario.
- El análisis diseño e implementación de la biblioteca de métodos SGASWebServices sirve de apoyo para la de recuperación de información contenida en el Sistema de Gestión Académica, para el desarrollo de aplicaciones con fines académicos.
- La configuración e implementación de la plataforma open source LimeSurvey, contribuye con la verificación del funcionamiento adecuado del SGASWebServices, comprobando que la aplicación está funcionando de manera eficiente, eficaz, y confiable cumpliendo con los requerimientos planteados.
- El SGASWebServices proporciona servicios a múltiples aplicaciones con fines académicos puesto que están disponible vía internet y usa protocolos web estándares y evita la duplicación de la información.
- La correcta y detallada documentación del SGASWebServices permite a los usuarios obtener una visión clara sobre el manejo y funcionamiento de cada uno de los servicios proporcionados en la biblioteca de métodos.



# RECOMENDACIONES

## 10. RECOMENDACIONES

Las recomendaciones que se derivan del presente trabajo, se detallan a continuación:

- Seleccionar la información más idónea y relevante acerca del tema central de investigación, herramientas a utilizar y requerimientos de los usuarios, para que la etapa de planificación garantice el correcto diseño y codificación del proyecto.
- Utilizar herramientas open source, como el framework de desarrollo web Turbogears ya que es muy potente, versátil y documentado, brinda la posibilidad de integrar diversos componentes y utilizarlos durante las fases de desarrollo de la aplicación.
- Analizar y diseñar cuidadosamente el software a desarrollar, con la definición adecuada del modelo del dominio, diagrama de secuencia, diagrama de clases final, modelo entidad relación, para evitar problemas en la etapa de codificación del proyecto.
- Realizar las pruebas de validación y usabilidad necesarias para asegurar que el software desarrollado funcione correctamente y depurar cualquier tipo de error.
- Revisar la documentación acerca del funcionamiento de cada uno de los métodos o servicios que ofrece el SGAWebServices para aprovechar todas las funcionalidades del mismo.
- Utilizar la biblioteca de métodos SGAWebServices para obtener información contenida en el Sistema de Gestión Académica, para el desarrollo de nuevas aplicaciones que estén orientados a fines académicos.



# BIBLIOGRAFÍA



## 11. BIBLIOGRAFÍA Y REFERENCIAS

### 11.1. LIBRO

RICHARDSON, LEONARD; RUBY, SAM.2007.RESTful Web Services.United States of America, O'Reilly.23-44pág. y 299-314pág.

### 11.2. RECURSOS DE INTERNET

- ALEGSA , [<http://www.alegsa.com.ar/Dic/soap.php>], [Diciembre - 2009]
- APACHE SOFTWARE FOUNDATION , [[http://http.apache.org/docs/2.0/es/mod/mod\\_proxy.html](http://http.apache.org/docs/2.0/es/mod/mod_proxy.html)], [Enero - 2010 ]
- APLICACIONESEMPRESARIALES.COM , [<http://www.aplicacionesempresariales.com/apache-el-servidor-web-mas-reconocido.html>], [Junio - 2010]
- APLICACIONESEMPRESARIALES.COM, [<http://www.aplicacionesempresariales.com/sqlite-el-motor-de-base-de-datos-agil-y-robusto.html>], [Junio – 2010]
- EL MUNDO ES OPEN SOURCE, [<http://blogs.antartec.com/opensource/2010/03/limesurvey-todo-lo-que-necesitas-para-encuestas-en-linea/>], [Enero-2011]
- GRUPO DE INVESTIGACIÓN FIEC – ESPOL, [ <http://blog.espol.edu.ec/taws/category/web-services/> ] , [Enero - 2010]
- CIBERNETIA,[[http://www.cibernetia.com/manuales/servicios\\_web/4\\_wsdl.php](http://www.cibernetia.com/manuales/servicios_web/4_wsdl.php)], [Febrero-2010]
- CSIRC, [[http://csirc.ugr.es/manualLimeSurvey\\_CSIRC\\_v5.pdf](http://csirc.ugr.es/manualLimeSurvey_CSIRC_v5.pdf)], [Enero-2011]
- DESARROLLOWEB.COM, [<http://www.desarrolloweb.com/articulos/450.php>], [Consulta: Noviembre-2009]
- DESARROLLOWEB.COM, [<http://www.desarrolloweb.com/articulos/460.php>], [Consulta: Diciembre-2009]
- DESARROLLOWEB.COM , [<http://www.desarrolloweb.com/articulos/1325.php>], [Consulta: Marzo-2010]
- DESARROLLOWEB.COM , [<http://www.desarrolloweb.com/articulos/1557.php>], [Consulta: Marzo-2010]
- DESARROLLOWEB.COM , [<http://www.desarrolloweb.com/articulos/1581.php>], [Consulta: Marzo-2010]

- DESARROLLOWEB.COM , [<http://www.desarrolloweb.com/articulos/1883.php>], [Consulta: Marzo-2010]
- DESARROLLOWEB.COM , [<http://www.desarrolloweb.com/articulos/2511.php>], [Consulta: Marzo-2010]
- LIMESURVEY.ORG, [<http://docs.limesurvey.org/tikiindex.php?page=Administraci%C3%B3n+de+Encuestas>], [Enero-2011]
- DOC.UBUNTU-ES.ORG, [[http://doc.ubuntu-es.org/Sobre\\_Ubuntu](http://doc.ubuntu-es.org/Sobre_Ubuntu)], [Diciembre - 2009]
- TURBOGears.ORG, [[http://docs.turbogears.org/1.0/WebServices\\_with\\_TGWebServices](http://docs.turbogears.org/1.0/WebServices_with_TGWebServices)], [Noviembre-2009]
- KIOSKEA.NET, [<http://es.kioskea.net/contents/linux/linintro.php3>], [Consulta: Octubre-2010]
- ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/LimeSurvey>], [Consulta: Febrero-2011]
- ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/PostgreSQL>], [Consulta: Julio-2010]
- ENCICLOPEDIA LIBRE WIKIPEDIA, [[http://es.wikipedia.org/wiki/Programacion%C3%B3n\\_Extrema](http://es.wikipedia.org/wiki/Programacion%C3%B3n_Extrema)], [Consulta: Julio-2010]
- ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/SOAP>], [Consulta: Julio-2010]
- ENCICLOPEDIA LIBRE WIKIPEDIA, [[http://es.wikipedia.org/wiki/Ubuntu#El\\_software\\_incluido](http://es.wikipedia.org/wiki/Ubuntu#El_software_incluido)], [Consulta: Julio-2010]
- ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/UDDI>], [Consulta: Agosto-2010]
- ENCICLOPEDIA LIBRE WIKIPEDIA, [[http://es.wikipedia.org/wiki/Web\\_Services\\_Protocol\\_Stack](http://es.wikipedia.org/wiki/Web_Services_Protocol_Stack)], [Consulta: Julio-2010]
- ENCICLOPEDIA LIBRE WIKIPEDIA, [[http://es.wikipedia.org/wiki/Web\\_Services\\_Protocol\\_Stack](http://es.wikipedia.org/wiki/Web_Services_Protocol_Stack)], [Consulta: Agosto-2010]
- ENCICLOPEDIA LIBRE WIKIPEDIA, [<http://es.wikipedia.org/wiki/XML-RPC>], [Consulta: Julio-2010]
- GXTECHNICAL, [[http://www.gxtechnical.com/gxdlsp/pub/GeneXus/Internet/TechnicalPapers/Web\\_Services.htm](http://www.gxtechnical.com/gxdlsp/pub/GeneXus/Internet/TechnicalPapers/Web_Services.htm)], [Noviembre-2009]

- IMAPAX, [http://www.imapax.com/soluciones/joomla/que-es-subversion-svn.html], [Mayo-2010]
- YUKEI.NET, [http://jesaavedra.opentelematics.org/?p=35], [Mayo-2010]
- KAMUS BLOG, [http://www.korporation.cl/blog/2009/03/acceso-filtrado-con-apache-y-mod\_proxy/], [Julio-2010]
- MONOGRAFIAS, [http://www.monografias.com/trabajos29/protocolo-acceso/protocolo-acceso.shtml], [Abril-2010]
- MSAFFIRIO, [http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/], , [Abril-2010]
- OSMOSISLATINA, [http://www.osmosislatina.com/apache/modulos.htm], [Enero - 2010]
- DIT.UPM.ES, [http://polaris.dit.upm.es/~rubentb/docs/subversion/Tutorial Subversion/index.html], [Enero - 2010]
- PROGRAMACIONEXTREMA, [http://www.programacionextrema.org], [Enero - 2010]
- PYTHON.ORG, [http://pypi.python.org/pypi/TGWebServices/2.0.0a2], [Noviembre-2009]
- RKBLOG, [http://www.rkblog.rk.edu.pl/w/p/mod\_wsgi/], [Julio-2010]
- RM-RF.ES, [http://rm-rf.es/configurar-apache-como-reverse-proxy-proxy-inverso/], [Febrero-2010]
- SCRIBD, [http://www.scribd.com/doc/7253619/Sqlite-Introduccion], [Abril-2010]
- SOLODISENIO, [http://www.solodisenio.com/lenguaje-de-programacion-python/], [Abril-2010]
- SQLALCHEMY.ORG, [http://www.sqlalchemy.org/], [Abril-2010]
- TURBOGARS.ORG, [ http://www.turbogears.org], [Abril-2010]
- W3C, [http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML], [Enero-2010]
- XOLUCIONESINFORMATICAS, [http://www.xolucionesinformaticas.com/index.php?option=com\_content&view=article&id=77&Itemid], [Enero-2010]



# ANEXOS

## 12. ANEXOS

### 12.1. ANEXO #1: Documentación de la Tecnología Web Services

#### WEB SERVICES

##### 1. HISTORIA DE LOS SERVICIOS WEB

En los primeros computadores corría un solo programa a la vez, pero en la medida que en un mismo computador podían correr varios programas al mismo tiempo, surgió la necesidad de contar con mecanismo de comunicación entre ellos, esto se llamó comunicación Task to Task y, este mecanismo a evolucionado debido que los computadores conforman redes. Por tanto, esta comunicación debe poder efectuarse entre un programa X, que corre en el computador Alfa, y otro programa Y, que corre en el computador Beta.

Para que esta comunicación funcione, primero debe existir un medio de comunicación entre el computador Alfa y el computador Beta; esto lo ha resuelto la Internet. Y segundo, el programa X debe saber conversar con el programa X. Para que esto ocurra el programador a cargo de X debe conocer de Y. A su vez el programador a cargo de Y debe conocer de X, por lo menos en los que se refiere al intercambio de datos. Esto hace que si no hay acuerdo entre el programador de X y el programador de Y, no hay comunicación posible.

La magia de los Web Services está en que el programador de X puede crear un Web Service para transferir datos sin necesidad de conocer al programador Y, ni a los programas que éste tiene a cargo. De modo que quien quiera recibir los datos solo necesita usar el Web Service y punto. Esto significa que pueden existir transferencias de datos entre distintas aplicaciones que funcionan en varios computadores, con distintos sistemas operativos, lenguajes de programación (PHP, C#, Java.), y que pertenezcan a diferentes empresas o instituciones.

Anteriormente se habían realizado intentos de crear estándares pero fracasaron o no tuvieron el suficiente éxito, algunos de ellos son DCOM y CORBA, por ser

dependientes de la implementación del vendedor DCOM - Microsoft, y CORBA - ORB. Otro gran problema es que se hacía uso de RPC (Remote Procedure Call) para realizar la comunicación entre diferentes nodos.

Es por esto que en 1999 se comenzó a plantear un nuevo estándar, al cual se lo terminó utilizando XML, SOAP, WSDL, y UDDI.

## 2. WEB SERVICES

Web Services describe una forma estandarizada de integrar aplicaciones WEB mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet, permiten a las organizaciones intercambiar datos sin necesidad de conocer los detalles de sus respectivos Sistemas de Información.

A diferencia de los modelos Cliente/Servidor, tales como un servidor de páginas Web, los Web Services no proveen al usuario una interfaz gráfica (GUI). En vez de ello, los Web Services comparten la lógica del negocio, los datos y los procesos, por medio de una interfaz de programas a través de la red. Es decir conectan programas, por tanto son programas que no interactúan directamente con los usuarios. Los desarrolladores pueden por consiguiente agregar a los Web Services la interfaz para usuarios, por ejemplo mediante una página Web o un programa ejecutable, tal de entregarle a los usuarios un funcionalidad específica que provee un determinado Web Service.

Los Web Services permiten a distintas aplicaciones, de diferentes orígenes, comunicarse entre ellos sin necesidad de escribir programas costosos, esto porque la comunicación se hace con XML. Los Web Services no están ligados a ningún Sistema Operativo o Lenguaje de Programación. Por ejemplo, un programa escrito en Java puede conversar con otro escrito en Pearl; Aplicaciones Windows puede conversar con aplicaciones Unix. Por otra parte los Web Services no necesitan usar browsers (Explorer) ni el lenguaje de especificación HTML.

El modelo de computación distribuida de los Web Services permite la comunicación de aplicación a aplicación. Por ejemplo, la aplicación que procesa las órdenes de compra se

puede comunicar con el sistema de inventarios, tal que este último le puede informar a la aplicación de compras cuales ítems deben comprarse por estar bajo su nivel mínimo. Dado el nivel integración que proveen para las aplicaciones.

### 3. CARACTERISTICAS DE LOS WEB SERVICES

La tecnología de web services posee varias características entre las cuales se pueden citar:

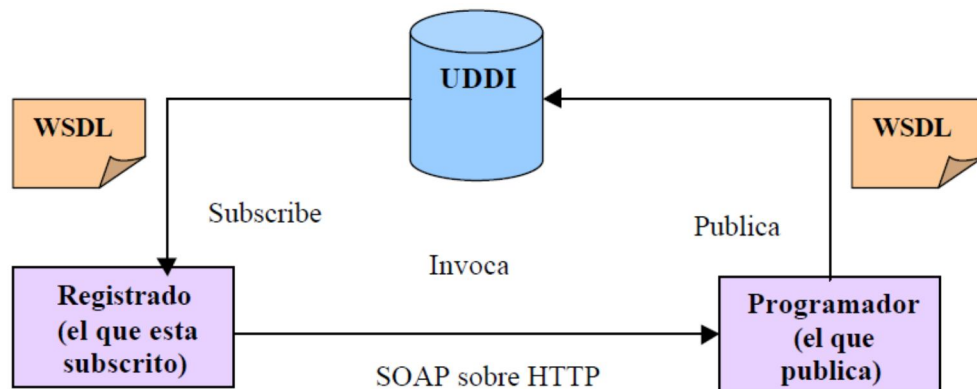
- **Interoperabilidad:** Los Servicios Web se pueden ser consumir por clientes de otras plataformas.
- **Acceso externo desde Internet:** realizan una buena gestión para
  - los accesos que provienen de clientes de Internet.
- **Tipos de datos de las Interfaces:** Los tipo de datos definidos para los Servicios Web se corresponde con los tipos de datos definidos por la mayoría de lenguajes de programación.
- **Uso de los estándares de Internet:** Los servicios Web utilizan los estándares de Internet y evitan, en la medida de lo posible, reinventar soluciones a problemas que ya están resueltas.
- **Soporte de cualquier lenguaje:** La implementación de un Servicio Web no está ligada a un particular lenguaje de programación. Esta es una gran ventaja frente a otras tecnologías como Java RMI, que está completamente ligada al uso de lenguaje Java, haciendo realmente difícil hacer una llamada a un objeto Java desde un objeto Visual Basic o Perl. De este modo, un cliente puede implementar o usar un Servicio Web independientemente del lenguaje de programación en el que fue implementado.
- **Soporte para cualquier infraestructura de componentes distribuidas:** Los Servicios Web no están ligados a una arquitectura de componentes en particular. Los protocolos facilitan a nivel base la comunicación entre las distintas

infraestructuras de objetos distribuidos. Por este motivo, únicamente es necesario preocuparse del desarrollo y utilización de Servicios Web.

#### 4. ARQUITECTURA DE LOS SERVICIOS WEB

En una arquitectura de Web Services hay dos partes claramente diferenciadas, el modo de utilizar un Web Services y cómo desarrollarlo. Existen varias partes implicadas y pasos necesarios para publicar un Web Service ya desarrollado y cómo puede ser utilizado, los cuales se detallan a continuación.

- El programador desarrolla el Web Service
- El programador describe el Web Service en un fichero WSDL
- El programador publica el Web Service en un directorio como UDDI
- La persona suscrita al directorio busca el Web Service
- La persona suscrita al directorio invoca el servicio con SOAP
- La persona suscrita al directorio recibe la respuesta mediante SOAP.



La arquitectura necesaria para el desarrollo de Web Services es la de un servidor que contenga las herramientas adecuadas para el soporte al desarrollo de este tipo de tecnología. Estas herramientas proporcionan el entorno de desarrollo de Web Services y la gestión de invocaciones de los servicios Web.



## 5. FUNCIONES DE LOS WEB SERVICES

Existen tres principales funciones dentro de la arquitectura de un web services:

- **Proveedor de Servicio:**

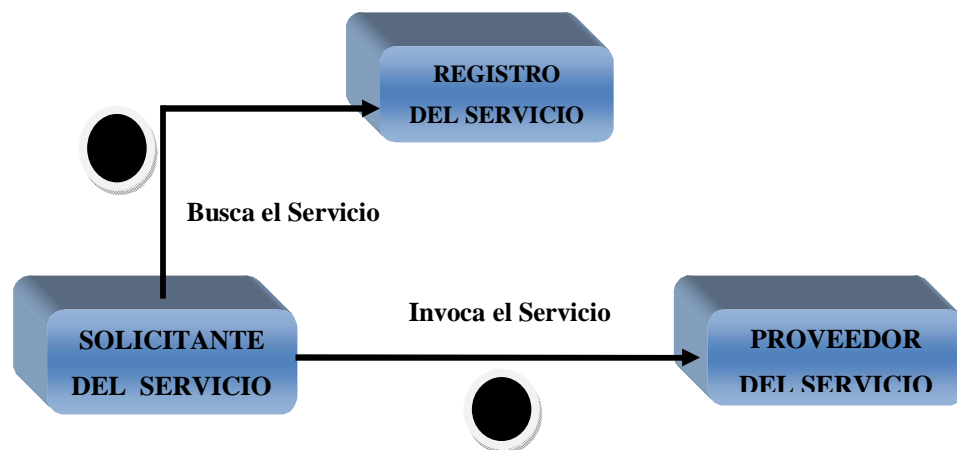
Este es el proveedor del web services. El proveedor del servicio está encargado de implementar el servicio, así hacerlo disponible en Internet.

- **Solicitante del Servicio:**

Este es cualquier consumidor del web services. El solicitante utiliza un servicio existente en el web al abrir una conexión de red y enviar una petición con XML.

- **Registro del Servicio:**

Este es un directorio lógicamente centralizado de servicios. El registro provee un repositorio central donde los desarrolladores pueden publicar nuevos servicios, así como encontrar los existentes. Él, por lo tanto, sirve como una cámara de compensación centralizada para compañías y sus servicios.



## 6. BLOQUES CONSTRUCTIVOS DE SERVICIOS WEB

En el siguiente grafico se muestran los bloques constructivos principales necesarios para facilitar las comunicaciones remotas entre aplicaciones.

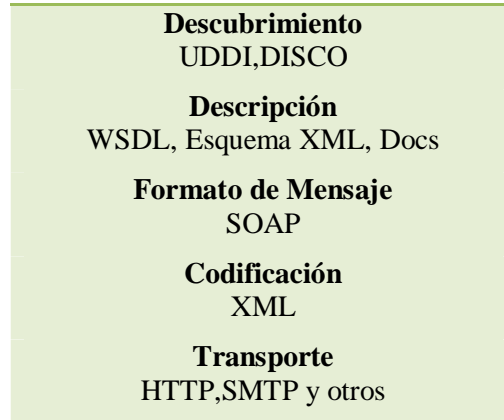


Figura: "Bloques constructivos de Servicios Web"

- **Descubrimiento:** La aplicación cliente que necesita acceder a la funcionalidad que expone un Servicio Web necesita una forma de resolver la ubicación de servicio remoto. Se logra mediante un proceso llamado, normalmente descubrimiento (discovery). El descubrimiento se puede proporcionar mediante un directorio centralizado así como por otros métodos.
- **Descripción:** Una vez que se ha resuelto el extremo de un servicio Web dado, el cliente necesita suficiente información para interactuar adecuadamente con el mismo. La descripción de un servicio Web implica meta datos estructurados sobre la interfaz que intenta utilizar la aplicación cliente así como documentación escrita sobre el servicio Web incluyendo ejemplo de uso.
- **Formato del mensaje:** Para el intercambio de datos, el cliente y el servidor tienen que estar de acuerdo en un mecanismo común de codificación y formato de mensaje. El uso de un mecanismo estándar de codificar los datos asegura que los datos que codifica el cliente los interpretará correctamente el servidor.
- **Codificación:** Los datos que se transmiten entre el cliente y el servidor necesitan codificarse en un cuerpo de mensaje.
- **Transporte:** Una vez se ha dado formato al mensaje y se han serializado los datos en el cuerpo del mensaje se debe transferir entre el cliente y el servidor utilizando algún protocolo de transporte.

## 7. PILA DE PROTOCOLOS DE LOS WEB SERVICES

La Pila de protocolos para Servicios Web es una colección de protocolos para redes de Computadores que son utilizados para definir, localizar, implementar y hacer que un Servicio Web interactúe con otro. La Pila de Protocolos para servicios está comprendida principalmente por cuatro áreas:

7.1. **Servicio de Transporte:** responsable del transporte de mensajes entre las Aplicaciones de red y los protocolos en los cuales se incluyen protocolos tales como:

- **“HTTP.-** El protocolo de transferencia de hipertexto es el protocolo usado en cada transacción de la Web (www). Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición se lo conoce como agente del usuario. A la información transmitida se la llama recurso y se la identifica mediante un URL. Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, etc.”
- **SMTP.-** Protocolo Simple de Transferencia de Correo, es un protocolo de la capa de aplicación. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (teléfonos móviles, etc.). Está definido en el RFC 2821 y es un estándar oficial de Internet.
- **FTP.-** Protocolo de Transferencia de Archivos, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

El Servicio FTP es ofrecido por la capa de Aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier

archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder al servidor, o apropiarse de los archivos transferidos.

**7.2. Mensajería XML:** responsable por la codificación de mensajes en un formato común XML así que ellos puedan ser entendidos en cualquier extremo de una conexión de red. Actualmente, esta área incluye protocolos tales como:

- **XML-RPC.-** es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. Es un protocolo muy simple ya que sólo define unos cuantos tipos de datos y comandos útiles, además de una descripción completa de corta extensión. La simplicidad del XML-RPC está en contraste con la mayoría de protocolos RPC que tiene una documentación extensa y requiere considerable soporte de software para su uso.
- **SOAP.-** es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP está diseñado para llevar a cabo intercambios de información en XML para sistemas altamente distribuidos, en específico: Internet.

**7.3. Descripción del Servicio:** usado para describir la interfaz pública de un Servicio Web específico.

- **WSDL.-** Es un formato XML que se utiliza para describir servicios Web, escribe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligán después al protocolo concreto de red y al formato del mensaje.

**7.4. Descubrimiento de servicios:** centraliza servicios en un registro común tal que los

servicios Web de la red puedan publicar su localización y descripción, y hace que sea fácil descubrir que servicios están disponibles en la red.

- **UDDI** son las siglas del catálogo de negocios de Internet denominado Universal Description, Discovery and Integration. UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

## 8. MENSAJERIA XML

XML es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Vamos a ver a lo largo de varios capítulos una introducción al mundo XML, es decir, al lenguaje así como a las tecnologías que trabajan con él, sus usos, ventajas y modos de llevar a cabo las tareas.

XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. El XML juega un papel importantísimo en este mundo actual, que tiende a la globalización y la compatibilidad entre los sistemas, ya que es la tecnología que permitirá compartir la información de una manera segura, fiable, fácil. Además, XML permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

XML no está sólo, sino que hay un mundo de tecnologías alrededor de él, de posibilidades, maneras más fáciles e interesantes de trabajar con los datos y, en definitiva, un avance a la hora de tratar la información, que es en realidad el objetivo de la informática en general. XML, o mejor dicho, el mundo XML no es un lenguaje, sino

varios lenguajes, no es una sintaxis, sino varias y no es una manera totalmente nueva de trabajar, sino una manera más refinada que permitirá que todas las anteriores se puedan comunicar entre sí sin problemas, ya que los datos cobran sentido.

XML es interesante en el mundo de Internet y el e-bussiness, ya que existen muchos sistemas distintos que tienen que comunicarse entre sí, pero como se ha podido imaginar, interesa por igual a todas las ramas de la informática y el tratamiento de datos, ya que permite muchos avances a la hora de trabajar con ellos.

### 8.1. Sintaxis de XML

Tiene una sintaxis extremadamente sencilla, el XML es un 10% del SGML (Lenguaje de Marcado Generalizado), porque las normas que tiene son muy simples. Se escribe en un documento de texto ASCII, igual que el HTML y en la cabecera del documento se tiene que poner el texto

```
<?xml version="1.0"?>
```

En el resto del documento se deben escribir etiquetas como las de HTML, las etiquetas que nosotros queramos, por eso el lenguaje se llama XML, lenguaje de etiquetas extendido. Las etiquetas se escriben anidadas, unas dentro de otras.

```
<ETIQ1>...<ETIQ2>...</ETIQ2>...</ETIQ1>
```

Cualquier etiqueta puede tener atributos. Le podemos poner los atributos que queramos.

```
<ETIQ atributo1="valor1" atributo2="valor2"...>
```

Los comentarios de XML se escriben igual que los de HTML.

```
<!-- Comentario -->
```

Y esto es todo lo que es el lenguaje XML en sí, aunque tenemos que tener en cuenta que el XML tiene muchos otros lenguajes y tecnologías trabajando alrededor de él.

Para definir qué etiquetas y atributos debemos utilizar al escribir en XML tenemos que fijarnos en la manera de guardar la información de una forma estructurada y ordenada. Por ejemplo, si deseamos guardar la información relacionada con una película en un documento XML podríamos utilizar un esquema con las siguientes etiquetas.

```
<?xml version="1.0"?>
<PELICULA nombre="El Padrino" año=1985>
<PERSONAL>
</DIRECTOR nombre="Georgie Lucar">
</INTERPRETE nombre="Marlon Brando" interpreta-a="Don Corleone">
</INTERPRETE nombre="Al Pacino" interpreta-a="Michael Corleone">
</PERSONAL>
</ARGUMENTO descripción="Película de mafias sicilianas en Estados Unidos">
</PELICULA>
```

Se han puesto las etiquetas que se han creído conveniente para poner este ejemplo y se lo anidado de manera que la etiqueta más grande es la PELICULA y dentro de ella tenemos el PERSONAL y el ARGUMENTO. A su vez, dentro de PERSONAL se ha puesto al DIRECTOR como a los actores (INTERPRETE).

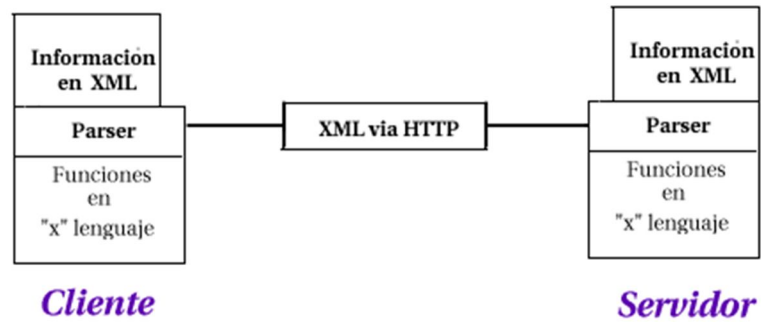
## 9. XML-RCP

XML-RPC es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.

Es un protocolo muy simple ya que sólo define unos cuantos tipos de datos y comandos útiles, además de una descripción completa de corta extensión. La simplicidad del XML-RPC está en contraste con la mayoría de protocolos RPC que tiene una documentación extensa y requiere considerable soporte de software para su uso.

### 9.1. Arquitectura de XMLRPC

En XMLRPC siempre se habla en términos de cliente/servidor, existe un sistema que realiza la solicitud ("el cliente") y otro que la atiende ("el servidor"), y como es de imaginarse un elemento clave en ambos puntos es el parser XML.



### 9.2. Implementaciones de XML-RPC.

Aunque es posible diseñar desde cero cualquier tipo de cliente y/o servidor para emplear XML, hoy en día ya existen diversas implementaciones para diversos ambientes y lenguajes, es a través de estas implementaciones que se logran ahorrar diversas labores como configuraciones de parser, cuestiones de seguridad, integración a servidores de páginas y otros detalles secundarios, utilizando estas estructuras ("Frameworks") se concentra en los procedimientos específicos y no en detalles comunes o secundarios que siempre son utilizados en XMLRPC.

### 9.3. Tipos de Datos

Según la especificación de XML-RPC, los principales tipos de datos son:

Nombre	Tags de ejemplo	Descripción
array	<pre>&lt;array&gt;   &lt;data&gt;     &lt;value&gt;&lt;i4&gt;1404&lt;/i4&gt;&lt;/value&gt;     &lt;value&gt;&lt;string&gt;Algo acá&lt;/string&gt;&lt;/value&gt;     &lt;value&gt;&lt;i4&gt;1&lt;/i4&gt;&lt;/value&gt;   &lt;/data&gt; &lt;/array&gt;</pre>	Array de valores, sin almacenar claves
base64	<pre>&lt;base64&gt;eW91IGNhbid0IHJlYWQgdGhpcyE=&lt;/base64&gt;</pre>	Datos binarios codificados en base 64
boolean	<pre>&lt;boolean&gt;1&lt;/boolean&gt;</pre>	Valor lógico (0 o 1)
dateTime	<pre>&lt;dateTime.iso8601&gt;19980717T14:08:55&lt;/dateTime.iso8601&gt;</pre>	Día y hora
double	<pre>&lt;double&gt;-12.53&lt;/double&gt;</pre>	Número de coma flotante de doble precisión
integer	<pre>&lt;i4&gt;42&lt;/i4&gt; &lt;int&gt;42&lt;/int&gt;</pre>	Número entero
string	<pre>&lt;string&gt;Hola mundo&lt;/string&gt;</pre>	String (cadena) de caracteres. Debe seguir la codificación XML.
struct	<pre>&lt;struct&gt;   &lt;member&gt;     &lt;name&gt;foo&lt;/name&gt;     &lt;value&gt;&lt;i4&gt;1&lt;/i4&gt;&lt;/value&gt;   &lt;/member&gt;   &lt;member&gt;     &lt;name&gt;bar&lt;/name&gt;     &lt;value&gt;&lt;i4&gt;2&lt;/i4&gt;&lt;/value&gt;   &lt;/member&gt; &lt;/struct&gt;</pre>	Array de valores, almacenando claves
nil	<pre>&lt;nil/&gt;</pre>	Valor nulo; una extensión XML-RPC



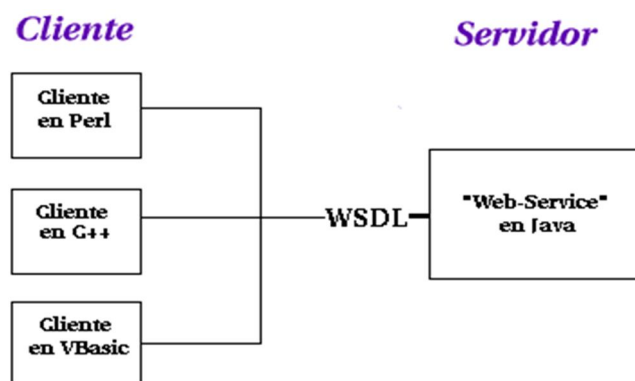
## 10. SOAP

SOAP es un protocolo para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP. Está basado en XML, a diferencia de DCOM y CORBA que son binarios; esto facilita la lectura por parte de los humanos, pero también los mensajes resultan más largos y, por lo tanto, considerablemente más lentos de transferir. Existen múltiples tipos de modelos de mensajes en SOAP pero, por lejos, el más común es el RPC, en donde un nodo de red (el cliente) envía un mensaje de solicitud a otro nodo (el servidor) y el servidor inmediatamente responde el mensaje al cliente. Los mensajes SOAP, son independientes del sistema operativo, y pueden transportarse en varios protocolos de internet como SMTP, MIME y HTTP.

### 10.1. Arquitectura de SOAP.

SOAP está diseñado para llevar a cabo intercambios de información en XML para sistemas altamente distribuidos, en específico: "Internet". Uno de los conceptos integrados a SOAP que no existe en XML-RPC es el uso de un lenguaje neutro (descendiente de XML) para describir las funciones/métodos residentes en "el servidor", esto tiene una ventaja muy evidente.

Al utilizar un lenguaje neutro (WSDL/"Web-Services Description Language") para describir las funcionalidades de "el servidor", éste funciona como un contrato al que se deben apegar los distintos "clientes"; lo anterior facilita que puedan ser escritos "clientes" en diversos lenguajes a partir de este contrato.



A partir de WSDL ("Web-Services Description Language") se describen las funcionalidades del web service, en XMLRPC es necesario conocer de antemano que funciones/métodos residen en "el servidor" y no solo esto, sino que además se requiere conocer el lenguaje en el que está escrito "el servidor", a través de WSDL se logra aislar el lenguaje específico del web service.

## 10.2. Ventajas de SOAP

Algunas de las Ventajas de SOAP son:

- **No está asociado con ningún lenguaje:** los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación, pero los desarrolladores responsables de mantener antiguas aficciones heredadas podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación.
- **No se encuentra fuertemente asociado a ningún protocolo de transporte:** La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- **No está atado a ninguna infraestructura de objeto distribuido** La mayoría de los sistemas de objetos distribuidos se pueden extender, y ya lo están alguno de ellos para que admitan SOAP.
- **Aprovecha los estándares existentes en la industria:** Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML., no define un medio de transporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.

- **Permite la interoperabilidad entre múltiples entornos:** SOAP se desarrolló sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dicho estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas.

### 10.3. Anatomía de un mensaje de SOAP

SOAP proporciona un mecanismo estándar de empaquetar un mensaje. Un mensaje SOAP se compone de un sobre que contiene el cuerpo del mensaje y cualquier información de cabecera que se utiliza para describir el mensaje. A continuación tiene un ejemplo:

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <!--Optional header information goes here. -->
    <To>Scott</To>
    <From>Suzanne</From>
  </soap:Header>
  <soap:Body>
    <!--Message goes here. -->
    Please pick up some milk on your way home from work.
  </soap:Body>
</soap:Envelope>
```

El elemento raíz del documento es el elemento Envelope. El ejemplo contiene dos subelementos, Body y Header. Un ejemplo de SOAP válido también puede contener otros elementos hijo en el sobre.

El sobre puede contener un elemento Header opcional que contiene información sobre el mensaje. En el ejemplo anterior, la cabecera contiene dos elementos que describen a quien compuso el mensaje, y posible receptor del mismo.

El sobre debe contener un elemento body, el cual contiene la carga de datos del mensaje. En el ejemplo el cuerpo contiene una simple cadena de caracteres.

Un mensaje debe estar dentro de sobre de SOAP bien construido. Un sobre se compone de un único elemento envelope el sobre puede contener un elemento Header y puede contener un elemento body. Si existe, la cabecera debe ser el elemento hijo inmediato del sobre, con el cuerpo siguiendo inmediatamente a la cabecera.

El cuerpo contiene la carga de datos del mensaje y la cabecera contiene los datos adicionales que no pertenecen necesariamente al cuerpo del mensaje. Además de definir un sobre de SOAP, la especificación de SOAP define una forma de codificar los datos contenidos en un mensaje. La codificación de SOAP proporciona un mecanismo estándar para serializar tipos de datos no definidos en la parte 1 de la especificación del esquema de XML.

La especificación de SOAP también proporciona un patrón de mensaje estándar para facilitar el comportamiento de tipo RPC. Se emparejan dos mensajes de SOAP para facilitar la asociación de un mensaje de petición con un mensaje de respuesta.

La llamada a un método y sus parámetros se serializan en el cuerpo del mensaje de petición en forma de una estructura. El elemento raíz tiene el mismo nombre que el método objetivo, con cada uno de los parámetros codificado como un subelemento.

El mensaje de respuesta puede contener los resultados de la llamada al método o una estructura de fallo bien definida. Por convenio, el elemento raíz tiene el mismo nombre que el método original al que se añade result. Los parámetros de retorno se serializan como elementos hijo, con el parámetro de retorno en primer lugar. Si se encuentra un error el cuerpo del mensaje de respuesta contendrá una estructura de fallo bien definida.

## 11. DESCRIPCIÓN DEL SERVICIO WSDL

WSDL son las siglas de *Web Services Description Language*, es un formato XML que se utiliza para describir servicios Web. Describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar

qué funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

Mediante esta descripción del WSDL se podrá distribuir un "Web-Service" para que este sea implementado en otros lenguajes como C++,Perl, .Net, Java, Python etc, de la misma manera, es este mismo archivo el que es utilizado para publicar "Web-Services" a un directorio distribuido como UDDI.

Podríamos decir que el WSDL es el manual de operación del web service, porque nos indica cuales son las interfaces que proveen el Servicio web y los tipos de datos necesarios para la utilización del mismo. Los documentos WSDL definen los servicios como colecciones de puntos finales de red o puertos. En WSDL, la definición abstracta de puntos finales y de mensajes se separa de la instalación concreta de red o de los enlaces del formato de datos. Esto permite la reutilización de definiciones abstractas: mensajes, que son descripciones abstractas de los datos que se están intercambiando y tipos de puertos, que son colecciones abstractas de operaciones. Las especificaciones concretas del protocolo y del formato de datos para un tipo de puerto determinado constituyen un enlace reutilizable. Un puerto se define por la asociación de una dirección de red y un enlace reutilizable; una colección de puertos define un servicio.

A continuación se detalla un poco más en profundidad cada uno de estos elementos:

- **Elemento types**

El elemento Types contiene información de esquema referenciado en el documento WSDL. El sistema de tipos predeterminado que admite WSDL es de esquema de XML.

- **Elemento message**

El elemento Message proporciona una abstracción común para el paso de mensajes entre el cliente y el servidor. Como puede utilizar múltiples formatos de definición de esquema en documento WSDL es necesario de disponer de un mecanismo común de identificar los mensajes. El elemento Message proporciona este nivel común de abstracción al que se hará referencia en otras partes del documento WSDL.

- **Elemento portType**

El elemento portType contiene un conjunto de operaciones abstractas que representan los tipos de correspondencia que pueden producirse entre el cliente y el servidor. Para los Servicios Web de estilo RPC se puede pensar en un portType como una definición de internas en donde cada método se puede definir como una operación.

- **Elemento binding**

El elemento binding contiene las definiciones de la asociación de un protocolo como SOAP a un determinado bindingType. Las definiciones binding especifican detalles de formatos del mensaje y el protocolo. Por ejemplo, la información de asociación especifica si se puede acceder a una instancia de un portType de forma RPC. Las definiciones binding también indican el número de comunicaciones de red que se requieren para realizar una determinada acción. Por ejemplo, una llamada RPC de SOAP sobre HTTP podría involucrar un intercambio de comunicación HTTP, pero esa misma llamada sobre SMTP podría involucrar dos intercambios de comunicaciones de SMTP discretas.

La asociación se logra utilizando elementos de extensión. Cada protocolo tiene su propio conjunto de elementos de extensión para especificar los detalles del protocolo y el formato de los mensajes. Para un determinado protocolo los elementos de extensión se suelen utilizar para decorar las acciones individuales de una operación y la propia operación con la información de asociación del protocolo. A veces los elementos de extensión se utilizan en el propio nivel portType.

- **Elemento service**

Un servicio es un grupo de puertos relacionados y se definen en el elemento service. Un puerto es un extremo concreto de un Servicio Web al que se hace referencia por una dirección única. Los puertos que se definen en determinado servicio son independientes. Por ejemplo, la salida de un puerto que no puede utilizarse como una entrada de otro.

- **Elementos de Extensibilidad**

Los elementos de extensibilidad se utilizan para representar determinadas tecnologías. Por ejemplo, se puede utilizar los elementos de extensibilidad para especificar el idioma en que se utiliza en el esquema de los elementos types.

El esquema para un determinado conjunto de elementos de extensibilidad se debe definir dentro de distintos espacios de nombres que WSDL. La definición de los propios elementos puede contener un atributo `wsdl:required` que indique un valor boolean si el atributo `required` se establece a `true` en una definición de elementos una asociación que haga referencia a ese conjunto concreto de electos de extensibilidad tiene que incluir dicho elemento.

La especificación WSDL define conjunto de elementos de extensibilidad para la asociación SOAP, HTTP GET, HTTP POS, MIME. Sin embargo, la especificación sólo define las asociaciones para dos de los cuatro tipos de operaciones. Un sentido y petición repuesta.

### **Ejemplo de un documento WSDL:**

`<?xml version="1.0">` - Este es otro documento XML, es por esto que debe comenzar con el tag `<?xml .. ?>`

`<definitions>` - Comienzo del documento, este tag agrupa a todos los demás.

`<types>` - Se definen los tipos de datos utilizados en el Web Service.

`</types>` - Fin de la definición de tipos.

`<message>` - Se definen los métodos y parámetros para realizar la operación. Cada message puede consistir en una o más partes (parámetros).

`</message>` - Fin de la definición de los parámetros.

`<portType>` - Esta sección es la más importante, ya que se definen las operaciones que pueden ser realizadas, y los mensajes que involucran (por ejemplo el mensaje de petición y el de respuesta).

`</portType>` - Fin de la definición de las operaciones y mensajes.

`<binding>` - Se definen el formato del mensaje y detalles del protocolo para cada portType.

`</binding>` - Fin de la definición del formato del mensaje y detalles del protocolo para cada PortType.

`</definitions>` - Fin del documento WSDL

### 11.1. Tipos de Operación

Existen varios tipos de operación en WSDL. El tipo más frecuente es el denominado "de petición-respuesta". Disponemos, además, de:

- Unidireccional: la operación recibe mensajes, sin retornar respuestas.
- Petición-respuesta: la operación recibe una petición y devuelve una respuesta.
- Solicitud-respuesta: la operación puede enviar una petición y permanecerá a la espera de una respuesta.
- Notificación: la operación puede enviar un mensaje sin esperar respuesta.

## 12. DESCRIPCIÓN DEL UDDI

UDDI es un registro público diseñado para almacenar de forma estructurada información sobre empresas y los servicios que éstas ofrecen. A través de UDDI, se puede publicar y descubrir información de una empresa y de sus servicios.

Lo más importante es que UDDI contiene información sobre las interfaces técnicas de los servicios de una empresa. A través de un conjunto de llamadas a API XML basadas en SOAP, se puede interactuar con UDDI tanto en tiempo de diseño como de ejecución para descubrir datos técnicos de los servicios que permitan invocarlos y utilizarlos. De este modo, UDDI sirve como infraestructura para una colección de software basado en servicios Web.

A partir de la creación de esta infraestructura para servicios Web, los datos sobre estos servicios se pueden encontrar de forma sistemática y confiable en una capacidad universal totalmente independiente de proveedores. Se pueden llevar a cabo búsquedas categóricas precisas utilizando sistemas de identificación y taxonómicos extensibles.

La integración de UDDI en tiempo de ejecución se puede incorporar a las aplicaciones. Como resultado, se fomenta el desarrollo de un entorno de software de servicios Web.

El UDDI es un modelo de directorios para Web Services. Es una especificación para mantener directorios estandarizados de información acerca de los Web Services, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente. UDDI utiliza WSDL para describir las interfaces de los Web Services. Es un lugar en el cual podemos buscar cuales son los Servicios web disponibles, una especie de directorio



en el cual podemos encontrar los Web Services publicados y publicar los Web Services que se desarrollan.

UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

Antes no había ningún estándar que permitiese localizar o dar a conocer servicios basados en el tratamiento de la información, ni existía ningún método que permitiera integrar los diferentes sistemas de información de las organizaciones.

Algunas de las aportaciones de UDDI son:

- Permite publicar y descubrir información acerca de quién provee que servicios
- Especificación distribuida de registro de web servicios
- Montado sobre Http, XML y SOAP
- Permite localizar el servicio apropiado entre los miles que están registrados en Internet.
- Define la interacción con el servicio elegido.
- Extiende el mercado potencial de los servicios.
- Describe los servicios y los métodos de negocio de forma automática, en un entorno seguro, sencillo y abierto.

El UDDI está compuesto de dos partes, la primera es una especificación técnica para construir un directorio distribuido de negocios y web services. La información es almacenada dentro de un formato específico en XML. La especificación de UDDI incluye detalles del API para búsqueda de información existente y de nueva información publicada. Segundo, el registro de negocios UDDI es una implementación completamente funcional de la especificación UDDI.

El registro de un negocio en UDDI tiene tres partes:

- Páginas blancas - dirección, contacto y otros identificadores conocidos.
- Páginas amarillas - categorización industrial basada en taxonomías.

- Páginas verdes - información técnica sobre los servicios que aportan las propias empresas.

### 12.1. WSDL y UDDI

WSDL se ha convertido en una pieza clave de la pila de protocolos de los servicios Web. Por eso, es importante saber cómo colaboran UDDI y WSDL y por qué la idea de interfaces frente implementaciones forma parte de cada protocolo. WSDL y UDDI se diseñaron para diferenciar claramente los metadatos abstractos y las implementaciones concretas. Para entender cómo funcionan WSDL y UDDI resulta esencial comprender las consecuencias de esta división. **Por ejemplo**, WSDL distingue claramente los mensajes de los puertos: los mensajes (la sintaxis y semántica que necesita un servicio Web) son siempre abstractos, mientras que los puertos (las direcciones de red en las que se invoca al servicio Web) son siempre concretos. No es necesario que un archivo WSDL incluya información sobre el puerto. Un archivo WSDL puede contener simplemente información abstracta de interfaz, sin facilitar datos de implementación concretos, y ser válido. De este modo, los archivos WSDL se separan de las implementaciones.

Una de las consecuencias más interesantes de esto es que pueden existir varias implementaciones de una única interfaz WSDL. Este diseño permite que sistemas dispares escriban implementaciones de la misma interfaz, para garantizar así la comunicación entre ellos. Si tres empresas diferentes implementan el mismo archivo WSDL y una parte del software de cliente crea el código auxiliar/proxy a partir de esa interfaz, dicho software se podrá comunicar con las tres implementaciones con el mismo código de base, cambiando simplemente el punto de acceso.

La publicación en UDDI es un proceso relativamente sencillo. El primer paso consiste en determinar información básica sobre cómo definir la empresa y los servicios en UDDI. El siguiente paso, una vez determinada esta información, consiste en llevar a cabo el registro, ya sea mediante programación o a través de una interfaz de usuario basada en el Web. Por último, se debe probar la entrada para asegurar que se registró correctamente y que aparece tal y como se esperaba en diferentes tipos de búsquedas y

herramientas.

UDDI y WSDL funcionan como especificaciones gratuitas que facilitan el desarrollo de una colección de software basado en servicios Web. WSDL ofrece un modo formal de definir servicios Web, independientemente del proveedor, que permitirá realizar llamadas a procedimientos remotos de próxima generación, mientras que UDDI proporciona una amplia infraestructura estandarizada que permite al usuario describir y descubrir servicios Web. Mediante la combinación de estos dos estándares, se podrá desarrollar todo un universo de servicios Web.

### 13. CONSIDERACIONES DE SEGURIDAD

La seguridad es un factor crítico de los web services. Sin embargo, ni las especificaciones de XML-RCP así como de Soap tienen algún requerimiento explícito de seguridad o autenticación. Es más, la comunidad de web services ha propuesto numerosos marcos de trabajo de seguridad y protocolos, pero todavía tiene que hacer un consenso para ponerse de acuerdo en determinar un paquete de seguridad comprensivo. En general existen tres problemas de seguridad: confidencialidad, autenticación y seguridad de red.

#### 13.1. Seguridad en servicios web

La seguridad en Web Service puede configurarse de las siguientes formas:

- Nivel de transporte
- Nivel de mensajes

##### 13.1.1. Seguridad a nivel de transporte

La conexión entre los clientes y el servidor se mantiene segura con SSL que proporciona servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4 o IDEA, y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública, típicamente el RSA.; si hay intermediarios, deben configurarse para que todos funcionen bajo SSL.

### 13.1.2. Seguridad a nivel de mensajes

La seguridad consiste en la protección encriptando los XML intercambiados y del uso de firmas digitales, los mensajes intercambiados son firmados digitalmente y encriptados. Por defecto, esta funcionalidad está deshabilitada; para habilitarla, se debe ir a Nodo (o instancia) / Configuration / Security / Message Security y seleccionar SOAP.

## 14. CONFIDENCIALIDAD

XML-RCP y SOAP corren principalmente sobre HTTP, la comunicación XML puede por lo tanto ser encriptada vía Secure Sockets Layer. SSL es una tecnología madura, ampliamente utilizada y es, por lo tanto, una opción muy viable de encriptamiento de mensajes.

### 14.1. WS-Security (Seguridad en Servicios Web)

Es un protocolo de comunicaciones que suministra un medio para aplicar seguridad a los Servicios Web. El protocolo contiene especificaciones sobre cómo debe garantizarse la integridad y seguridad en mensajería de Servicios Web.

La Integridad de datos y confidencialidad pueden garantizarse sobre Servicios Web a través del uso de la Transport Layer Security (TLS), por ejemplo enviando mensajes sobre HTTPS. Esto puede reducir significativamente la sobrecarga, por ejemplo eliminando la necesidad de codificar claves y firmas de mensaje en ASCII antes de enviar. La parte negativa de usar TLS sería si los mensajes necesitaran pasar a través de un servidor proxy, como si fuera necesario ver la petición para enrutado. En tal caso, el servidor vería la petición que llega del proxy, no del cliente; esto podría ser solventado si el proxy tiene una copia de la clave y certificado del cliente, o teniendo un certificado de firmado de confianza para el servidor, con el cual podría generar un par clave/certificado que coincida con aquellos del cliente. Sin embargo, el hecho de que el proxy está operando el mensaje significa que no asegura la seguridad extremo a extremo, sino que solo asegura la seguridad punto a punto.

WS-Security incorpora características de seguridad en el encabezado de un mensaje SOAP, trabajando en la capa aplicación. Así asegura seguridad extremo a extremo.

## 15. AUTENTICACIÓN Y AUTORIZACIÓN

### 15.1. AUTENTICACIÓN:

La autenticación es el proceso por el que se comprueba la identidad de alguien o algo, para ver si es lo que dice ser. Ese "alguien" o "algo" se denomina principal. La autenticación requiere pruebas de identidad, denominadas credenciales. Por ejemplo, una aplicación cliente puede presentar una contraseña como sus credenciales. Si la aplicación cliente presenta las credenciales correctas, se asume que es quien dice ser.

Existen varios modos de proveer autenticación para un web service. Vamos a identificar alguno de ellos y luego veremos en mayor detalle la utilización de headers soap.

1. Se puede proveer de seguridad integrada, agregando entradas al archivo web.config
2. Pueden utilizarse las extensiones para Web Services WS-E que permiten desarrollar web services seguros utilizando los últimos estándares de seguridad. Es la forma más completa y aconsejable a nivel de arquitectura.
3. Utilizar headers soap para pasarle información del usuario al web service.

### 15.2. AUTORIZACIÓN

Una vez que se ha autenticado la identidad de un principal, deben tomarse decisiones sobre la autorización. El acceso se determina comparando la información del principal con información de control de acceso, como listas de control de acceso (ACL). Es posible que los clientes tengan distintos grados de acceso. Por ejemplo, algunos clientes pueden tener acceso total a los servicios Web XML, mientras que otros estarían autorizados sólo a ciertas operaciones. A algunos clientes se les permitirá un acceso total a todos los datos, mientras que a otros sólo se les permitirá acceso a un subconjunto de los datos y otros tendrán acceso de sólo lectura.

Un modo sencillo de implementar servicios Web XML, consiste en aprovechar las características de autenticación del protocolo que se utilice para intercambiar mensajes. Para la mayoría de los servicios Web XML, esto significa aprovechar las características de autenticación disponibles en HTTP. Microsoft Internet Información Server (IIS) e ISA Server funcionan en conjunción con Windows 2000 Server y ofrecen soporte para varios mecanismos de autenticación en HTTP.

- **Básica:** utilizada para identificación no segura o poco segura de clientes, ya que el nombre de usuario y la contraseña se envían como texto codificado en base 64, que puede ser fácilmente descodificado. IIS autorizará el acceso a los servicios Web XML si las credenciales coinciden con las de una cuenta de usuario válida.
- **Básica sobre SSL:** igual que la autenticación básica, excepto que el canal de comunicación está cifrado y protege de ese modo el nombre de usuario y la contraseña. Una buena opción para entornos en Internet; sin embargo, el uso de SSL influye negativamente en el rendimiento.
- **Implícita:** utiliza algoritmos hash para transmitir las credenciales del cliente de forma segura. Sin embargo, es posible que no sea compatible con todas las herramientas de desarrollo para generar clientes de servicios Web XML. IIS autorizará el acceso a los servicios Web XML si las credenciales coinciden con las de una cuenta de usuario válida.
- **Autenticación de Windows integrada:** resulta útil sobre todo en entornos en Intranet. Utiliza NTLM o Kerberos. El cliente debe pertenecer al mismo dominio que el servidor o a un dominio en el que el dominio del servidor confía. IIS autorizará el acceso a los servicios Web XML si las credenciales coinciden con las de una cuenta de usuario válida.
- **Certificados de cliente a través de SSL:** requiere que cada cliente obtenga un certificado. Los certificados se asignan a las cuentas de usuario, que son utilizadas por IIS para autorizar el acceso a los servicios Web XML. Se trata de una solución viable para entornos en Internet, aunque el uso de certificados



digitales no está muy extendido actualmente. Es posible que no sea compatible con todas las herramientas de desarrollo para generar clientes de servicios Web XML. Sólo está disponible en conexiones SSL, de modo que el rendimiento puede verse afectado.

Desde la perspectiva de alguien que intenta implementar servicios Web XML, una de las ventajas de utilizar estos mecanismos de autenticación es que no se necesita escribir nuevo código. IIS/ISA Server completa todo el proceso de autenticación y autorización ACL antes de llamar a los servicios Web XML. Sin embargo, al implementar el cliente será necesario realizar algunas tareas adicionales. La aplicación cliente necesitará responder a las peticiones de autenticación y credenciales del servidor.



12.2. ANEXO #2: Formato de Encuesta para Recolección de Requerimientos

**UNIVERSIDAD NACIONAL DE LOJA**

**CARRERA DE INGENIERÍA EN SISTEMAS**

**Recolección de Requerimientos**

Los egresados de la carrera de Ingeniería en Sistemas con la finalidad de obtener los requerimientos necesarios para el desarrollo del proyecto de tesis titulado “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SUB SISTEMA DE SERVICIOS WEB (WEB SERVICES) BASADO EN UNA BIBLIOTECA DE MÉTODOS DE RECUPERACIÓN DE INFORMACIÓN DEL SISTEMA DE GESTIÓN ACADÉMICA DE LA UNIVERSIDAD NACIONAL DE LOJA, QUE PERMITA LA COMUNICACIÓN E INTEROPERABILIDAD CON APLICACIONES RELACIONADAS AFINES ACADÉMICOS”, ponemos a consideración la siguiente entrevista:

El Sistema de Gestión Académico actualmente se encuentra en fase de implementación en la Universidad Nacional de Loja, y surge de la necesidad de automatizar los procesos académicos que se venían desarrollando de forma manual.

**¿Conoce usted acerca de la tecnología de servicios web (web services)?**

Si( )

No( )

En qué consiste.....  
.....

**¿Qué información del Sistema de Gestión Académica le gustaría obtener?**

1. Información Docentes( )
2. Información Estudiantes( )
3. Información de Acreditaciones( )
4. Información de Asistencias( )
5. Información Matriculación( )

Otros.....





**Del siguiente listado ¿Qué información considera que sería utilizada en el desarrollo de aplicaciones de software a fines académico?**

- Ofertas Académicas por modalidad de estudio ( )
- Planes de Estudio por carrera ( )
- Número Estudiantes Matriculados por oferta académica( )
- Número Estudiantes Matriculados por área ( )
- Número Estudiantes Matriculados por carrera ( )
- Número Estudiantes Aprobados por oferta académica ( )
- Número Estudiantes Aprobados por área ( )
- Número Estudiantes Aprobados por carrera ( )
- Número Estudiantes Reprobados por oferta académica ( )
- Número Estudiantes Reprobados por área ( )
- Número Estudiantes Reprobados por carrera ( )
- Reporte de estudiantes aprobados por paralelo ( )
- Reporte de estudiantes reprobados por paralelo ( )
- Reporte de estudiantes matriculados por paralelo ( )
- Listado de Paralelos por carrera ( )
- Listado de Estudiantes por paralelo ( )
- Listado de Estudiantes por carrera( )
- Reporte de asistencia por alumno ( )
- Reporte de notas por alumno ( )
- Datos personales de estudiantes ( )
- Datos de estudiantes con una o más matriculas reprobadas ( )
- Datos personales docentes ( )
- Periodos de matrícula ( )
- Fechas en que se ofertaron los módulos de cada carrera ( )
- Carga horaria de docentes ( )
- Tipo del plan de estudio vigente ( )
- Estudiantes por carrera ( )

Otros.....  
.....

### 12.3. ANEXO # 3: Formato de Encuesta Socioeconómica proporcionada por el IDICE



## *ENCUESTA SOCIOECONÓMICA A LOS ESTUDIANTE DE LA UNIVERSIDAD NACIONAL DE LOJA*

Mensaje al estudiante:

La información que entregará a través de este cuestionario es totalmente confidencial, y será utilizado exclusivamente para mejorar los sistemas de información universitaria y mejorar nuestros servicios a la comunidad estudiantil.

**¡AGRADECEMOS POR SU VALIOSA COLABORACION!**

#### **Notas para el departamento informático:**

- 1. Texto resaltado en rojo es para uso del departamento de sistemas.**
- 2. Si no se especifica, las preguntas de opción múltiple implican la elección de una sola alternativa; cuando alguna alternativa es señalada por el estudiante deberán desactivarse el resto de opciones.**

#### **1. INFORMACIÓN GENERAL**

##### **1.1. ¿De dónde es usted?**

**(Ingresar opciones en informática)**

País	
Provincia/Departamento	
Cantón/Municipio	

##### **1.2. Teléfono**



<b>Domicilio:</b>	
<b>Celular:</b>	

**1.3. E-mail**

--

**1.4. Lugar donde vive**

Ciudad:	
Dirección:	

**2. ANTECEDENTES PERSONALES Y SOCIOECONÓMICOS****2.1. Fecha de Nacimiento**

Día (número)	Mes (número)	Año (número)

**2.2. Sexo**

1.	Masculino	( )
2.	Femenino	( )

**2.3. Estado Civil**

1.	Soltero (a)	( )	4.	Divorciado(a)	( )
2.	Casado (a)	( )	5.	Separado(a)	( )
3.	Viudo(a)	( )	6.	Unión Libre	( )

**2.4. ¿Tiene usted hijos?**

1.	Sí	( )	Cuántos:	
2.	No	( )		

**2.5. ¿Cuántas personas componen su hogar?**

*(Contar todos aquellos que comparten la alimentación diaria en casa)*

<b>Número:</b>	
----------------	--

**2.6. ¿Quién es el jefe de Hogar?**

1.	Padre	( )	4.	Hermano	( )
----	-------	-----	----	---------	-----

2.	Madre	( )	5.	Otro	( )
3.	Usted	( )	6.	Ninguno	( )

**2.7. ¿Viven sus padres?**

1.	Ambos viven	( )	3.	Sólo el padre	( )
2.	Sólo la madre	( )	4.	Ninguno Vive	( )

**2.8. ¿Con quién vivió los últimos seis meses?**

1.	Madre y Padre	( )	5.	Amigos	( )
2.	Con el padre	( )	6.	Sólo	( )
3.	Con la Madre	( )	7.	Otros Parientes	( )
4.	Pareja	( )			

**2.9. La vivienda o lugar en el que habita usted es:**

1.	Arrendada	( )	5.	Cedida	( )
2.	Dada por anticresis	( )	6.	Recibida por servicios	( )
3.	Propia y la está pagando	( )	7.	Otra.	( )
4.	Propia y totalmente pagada	( )			

**2.10. ¿Quién financia sus estudios (marcar una o más alternativas)?**

1.	Padres	( )	5.	Préstamo	( )
2.	Cónyuge	( )	6.	Trabajo personal	( )
3.	Otros familiares	( )	7.	Otras fuentes	( )
4.	Beca	( )			

**2.11. ¿Además de usted algún integrante de su hogar estudia?**

1.	Sí	( )	Cuántos:	
2.	No	( )	<b>Desactivar 2.12</b>	

**2.12. Especifique el nivel de educación más alto de los miembros de su hogar que actualmente estudian: (Se puede anotar varias opciones)**

	Nivel educativo	Marcar	¿Cuántos?
1.	Pre-Básica	( )	
2.	Educación Básica	( )	



3.	Educación Media o Bachillerato	( )	
4.	Beca	( )	
5.	Post Bachillerato	( )	
6.	Universidad	( )	
7.	Post-grado	( )	

**2.13. Usted está afiliado y/o cubierto, respecto de su salud, por:**

*(Se puede anotar varias opciones)*

1.	Seguro de Salud privado	( )	4.	Seguro Comunitario	( )
2.	IESS, seguro general	( )	5.	Ninguno	( )
3.	IESS, seguro campesino	( )	6.	Otro	( )
4.	Seguro ISFFA o ISSPOL (policía o fuerzas armadas)	( )			

**2.14. Por favor, digite el número de años aprobados en el nivel de educación más alto que alcanzaron sus padres:**

	<i>Nivel de educación más avanzado que alcanzó el padre o la madre</i>	<i>Años aprobados</i>	
		<i>Padre</i>	<i>Madre</i>
1.	Ninguno		
2.	Centro de alfabetización		
3.	Educación Básica		
4.	Primaria		
5.	Educación media a bachillerato		
6.	Secundaria		
7.	Post Bachillerato		
8.	Centro de formación técnica		
9.	Superior		
10.	Posgrado		

**2.15. ¿Cuántos miembros de su hogar trabajan?**

Número:	<input type="text"/>
---------	----------------------

**2.16. ¿Cuál es la situación ocupacional de sus padres?**

(Desactivar esta pregunta si en pregunta 2.7 se contestó código 1; ó desactivar columna padre si en pregunta 2.7 se contestó código 2; ó desactivar columna madre si en pregunta 2.7 se contestó código 3) (Se marca sólo una alternativa sea por el padre o bien sea por la madre)

	<i>Situación ocupacional</i>	<i>Padre</i>	<i>Madre</i>	
1.	Activo, tiene trabajo remunerado a tiempo completo.			
2.	Activo, tiene trabajo remunerado a tiempo parcial.			
3.	Trabaja sólo ocasionalmente			
4.	Cesante			→ Saltar a 2.19
5.	Jubilado o pensionado			→ Saltar a 2.19
6.	No trabaja por (enfermedad, edad avanzada, etc)			→ Saltar a 2.19

**2.17. ¿Cuál es la principal categoría laboral de sus padres (a la que dedican más tiempo)?**

(Desactivar esta pregunta si en pregunta 2.7 se contestó código 1; ó desactivar columna padre si en pregunta 2.7 se contestó código 2; ó desactivar columna madre si en pregunta 2.7 se contestó código 3)

(Se marca sólo una alternativa sea por el padre o bien sea por la madre)

	<i>Situación ocupacional</i>	<i>Padre</i>	<i>Madre</i>
1.	Empleado/ Obrero de Gobierno		
2.	Empleado/ Obrero Privado		
3.	Jornalero o peón		
4.	Patrono (tiene negocio, taller, vehículo de trabajo)		
5.	Cuenta Propia (profesional independiente por ejemplo:		



	medico, abogado, otro)		
6.	Empleado doméstico(a)		
7.	Trabajador del Hogar sin pago		
8.	Trabajador no del Hogar sin pago		

**2.18. Aparte de estudiar, ¿también usted trabaja?**

1.	Si	( )	
2.	No	( )	<b>Desactivar 2.19</b>

**2.19. ¿Cuál es la principal categoría laboral a la que usted dedica más tiempo?**

	<i>Situación ocupacional</i>	<i>Marcar</i>
1.	Empleado/ Obrero de Gobierno	
2.	Empleado/ Obrero Privado	
3.	Jornalero o peón	
4.	Patrono (tiene negocio, taller, vehículo de trabajo)	
5.	Cuenta Propia (profesional independiente por ejemplo: medico, abogado, otro)	
6.	Empleado doméstico(a)	
7.	Trabajador del Hogar sin pago	
8.	Trabajador fuera del Hogar sin pago	

**2.20. Mencione por favor el ingreso total aproximado del Hogar.**

Cifra en dólares:	
-------------------	--

**3. ANTECEDENTES EDUCACIONALES****3.1. ¿De qué establecimiento de educación secundaria procede usted?**

**(Ingresar opciones en informática: si es extranjero activar opción texto de preguntas 3.2 y 3.3)**

--

3.2. ¿En qué provincia (o similar si es usted extranjero) se encuentra este establecimiento?

**(Ingresar opciones en informática)**

3.3. ¿Cuál es la localidad (cantón o similar si es usted extranjero) del establecimiento?

**(Ingresar opciones en informática)**

3.4. ¿De qué tipo es el establecimiento de educación secundaria donde estudio?

1.	Fiscal o del Estado	( )
2.	Privado o Particular	( )
3.	Fiscomisional	( )

3.5. ¿En qué año egresó de la secundaria?

Año:

3.6. ¿Cuál es su área de formación?

1.	Física-matemática	( )	5.	Contabilidad	( )
2.	Químico-Biológicas	( )	6.	Ciencias Básicas	( )
3.	Informática	( )	7.	Otra.	( )
4.	Ciencias Sociales	( )			

3.7. Marque el área en que obtuvo mayor y menor promedio durante el último año de educación secundaria-media

**(Permitir sólo una alternativa, sea en mayor o bien en menor)**

	Área	Mayor	Menor
1.	Lenguaje y comunicación		
2.	Matemáticas		
3.	Científica (Biología, química, física)		
4.	Deporte y recreación		
5.	Artístico-Cultural		





6.	Área técnica		
7.	Otra		

**3.8. ¿Está conforme con la carrera en que se matriculó?**

1.	Sí	( )	<i>Desactivar 3.9</i>
2.	No	( )	

**3.9. Si su respuesta en la pregunta anterior es NO, señale el motivo:**

1.	No era la carrera que deseaba	( )
2.	No llena sus expectativas	( )
3.	La infraestructura no es la más idónea	( )
4.	Profesores	( )
5.	Reputación de la carrera	( )
6.	Falta de enseñanza Práctica	( )
7.	Otra razón	( )

**3.10. ¿Por qué eligió esta Carrera?**

1.	Por la planta de profesores	( )
3.	Infraestructura física	( )
4.	Reputación de la carrera	( )
5.	Enseñanza Práctica	( )
6.	Referencia de Terceros	( )
7.	Por gusto	( )
8.	Cercanía con el hogar	( )
9.	Tema familiar	( )
10.	Otra razón	( )

**3.11. ¿Por qué eligió esta Universidad?**

1.	Por ser Pública	( )
2.	Profesores	( )
3.	Infraestructura física	( )
4.	Por su prestigio académico	( )

5.	Enseñanza Práctica	( )
6.	Por estar acreditada	( )
7.	Referencia de Terceros	( )
8.	Por la calidad de sus egresados	( )
9.	Por su otorgamiento de beneficios	( )
10.	Por gusto	( )
11.	Cercanía con el hogar	( )
12.	Tema familiar	( )
13.	Otra razón	( )

**3.12. ¿Por qué otra alternativa de estudio habría optado de no haberse matriculado en la UNL?**

<b>CARRERA</b>	<b>UNIVERSIDAD</b>	<b>PROVINCIA</b>	<b>NINGUNA</b>
<b>Permitir texto</b>	<b>Permitir texto</b>	<b>Permitir texto</b>	<b>Marca</b>

**3.13. ¿Ha realizado o está actualmente cursando estudios superiores en otra carrera de la UNL o de otro centro de estudios?**

1.	Estuve cursando alguna vez otra carrera	( )	<b>Activar alternativas cuadro siguiente</b>
2.	Estoy cursando actualmente otra carrera	( )	<b>Activar alternativas cuadro siguiente</b>
3.	Ninguna de las anteriores respuestas	( )	<b>Desactivar 3.14</b>
	<b>Nombre de la institución</b>	<b>Carrera</b>	<b>Número de semestres o módulos aprobados</b>
	Universidad	<b>Permitir texto</b>	<b>Permitir número</b>
	Instituto	<b>Permitir texto</b>	<b>Permitir número</b>
	Centro de Formación Técnica	<b>Permitir texto</b>	<b>Permitir número</b>

**3.14. Si ya no está cursando esa otra carrera indique la razón:**

1.	Ya la terminó	( )
2.	Enfermedad	( )
3.	Labores Domésticas	( )
4.	Falta de Dinero	( )
5.	Trabajo	( )
6.	No le interesó	( )
7.	Malas calificaciones	( )
8.	Problemas Familiares	( )
9.	Embarazo	( )
10.	Otra razón	( )

**3.15. ¿Pertenece usted a algún pueblo originario?**

1.	Otavaló	( )	4.	Montuvio	( )
2.	Saraguro	( )	5.	No se siente identificado	( )
3.	Shuar	( )	6.	Otro.	( )

**3.16. ¿Se siente usted identificado con algún pueblo originario?**

1.	Sí	( )
2.	No	( )

**4. RECURSOS DE APOYO****4.1. Señale el lugar en donde dispone más frecuentemente de los siguientes recursos tecnológicos:**

		<i>Computador</i>	<i>Internet</i>
1.	En el Hogar		
2.	En la Universidad		
3.	En algún cibercafé		
4.	Dónde un pariente o amigo		
5.	Otros lugares		



4.2. ¿Para qué utiliza frecuentemente Internet? (Puede marcar más de una alternativa)

1.	Para obtener información académica o de estudios	( )
2.	Para comunicarse (correo electrónico, chat)	( )
3.	Para trámites	( )
4.	Para actividades con fines de entretenimiento	( )
5.	No lo utiliza	( )

4.3. De las siguientes alternativas marque el idioma y/o lengua (aparte del español) de la que usted tenga mayor conocimiento:

1.	Inglés	( )	4.	Lengua Originaria, (Quechua, etc.)	( )
2.	Francés	( )	5.	Otra	( )
3.	Alemán	( )	6.	Ninguna ( <i>desactivar 4.4</i> )	( )

4.4. Acerca del idioma o lengua que posee mayor conocimiento, señale el nivel alcanzado:

1.	Básico (poca comprensión lectora y auditiva. No mantiene conversación)	( )
2.	Intermedio (mayor grado de comprensión y se comunica con dificultades)	( )
3.	Avanzado (comprende y se comunica en forma apropiada)	( )

4.5. ¿Cuántas horas diarias dedica al estudio de su módulo fuera del horario de clase?

Número de horas:	<input type="text"/>
------------------	----------------------

4.6. ¿Qué material de apoyo utiliza habitualmente durante el tiempo que estudia? (Puede marcar una o más alternativas)

1.	Apuntes de clases propios	( )	4.	Apuntes de Internet	( )
2.	Libros	( )	5.	Otros	( )
3.	Fotocopias	( )			



4.7. ¿Cómo estudia con mayor frecuencia?

1.	Sólo	( )	3.	Con tutor	( )
2.	Con compañeros	( )	4.	Otros	( )

5. PARTICIPACION

5.1. ¿Participa o ha participado en alguna(s) de las siguientes organizaciones?

(Puede marcar una o más alternativas)

1.	Junta de vecinos u organización barrial	( )
2.	Representación estudiantil en la UNL	( )
3.	ONG o Similar	( )
4.	Beneficencia y/o voluntariado (cruz roja, defensa civil, bomberos, acilos)	( )
5.	Club Deportivo/ recreativo	( )
6.	Partido Político	( )
7.	Congregación religiosa	( )
8.	Organización artístico-cultural	( )
9.	No participa, ni ha participado en ninguna	( )

6. SALUD MENTAL

6.1. ¿Tiene algún familiar que sufra una o más enfermedades graves que requiera tratamiento permanente?

1.	Sí	( )	
2.	No	( )	<b>Desactivar 6.2</b>
3.	No sabe	( )	<b>Desactivar 6.2</b>

6.2. ¿Qué enfermedad (es)?

(Permitir ingreso de texto)

--

6.3. ¿Tiene usted alguna enfermedad que requiera tratamiento médico permanente?

1.	Sí	( )	
2.	No	( )	<b>Desactivar 6.4</b>
3.	No sabe	( )	<b>Desactivar 6.4</b>

6.4. ¿Cuál enfermedad?

**(Permitir ingreso de texto)**

6.5. ¿Práctica usted en forma permanente (al menos una vez por semana) algún deporte?

1.	Sí	( )	
2.	No	( )	<b>Desactivar 6.6</b>

6.6. ¿Cuál deporte practica generalmente?

**(Permitir ingreso de texto)**

6.7. ¿Ha recibido ó recibe actualmente atención psicológica y/o psiquiátrica?

1.	Sí	( )
2.	No	( )

6.8. ¿Ha presentado o presenta usted alguno de los siguientes problemas?

**(Puede marcar una o más alternativas)**

1.	Depresión	( )	5.	Violencia intrafamiliar	( )
2.	Fobias o aversiones	( )	6.	Otro (mencione)	( )
3.	Dificultades para adaptarse al cambio	( )	7.	No ha presentado	( )
4.	Trastornos de ansiedad	( )			



6.9. Considera usted el consumo de alguna sustancia como:

**(Permitir marcar solo una alternativa por fila)**

		<i>CIGARRILLO</i>	<i>DROGA</i>	<i>ALCOHOL</i>
1.	Beneficioso			
2.	Perjudicial			
3.	Le es indiferente			



12.4. ANEXO #4: Formato de Encuesta de Prueba de Validación

**UNIVERSIDAD NACIONAL DE LOJA**

**ENCUESTA AL ADMINISTRADOR DEL SISTEMA DE GESTIÓN  
ACADEMICA**

**Encuesta para la validación y comprobación del subsistema de servicios web basado en una biblioteca de recuperación de información del Sistema de Gestión Académica de la Universidad Nacional de Loja que permite la comunicación e interoperabilidad con aplicaciones relacionadas a fines académicos.**

Mediante la presente encuestas se pretende realizar la comprobación y validación de los métodos para obtención de información del SGA.

**Orientado**

**a:.....**

**1. Considera usted que la tecnología que se ha utilizado para el desarrollo de la biblioteca de métodos(TurboGears2.0, TGWebServices) es:**

- Excelente ( )
- Muy Bueno ( )
- Bueno ( )
- Regular ( )

**2. Considera usted que el proceso de administración del SGAWebServices (ejemplo: creación de nuevos métodos, modificación de los métodos, etc ) es:**

- Fácil ( )
- Medio ( )
- Difícil ( )





3. La creación y manipulación de los usuarios del SGAWebServices con mysql es:

Fácil ( )

Medio ( )

Difícil ( )

4. La información obtenida del SGAWebServices para los usuarios del Sistema consideras usted que es:

Importante ( )

Normal ( )

Regular ( )

5. Considera usted que el sistema cumple con la finalidad con que fue planteado inicialmente?:

Si ( )

Parcialmente ( )

No ( )

6. Considera usted que la seguridad aplicada para el acceso al WebServices es:

Excelente ( )

Muy Bueno ( )

Bueno ( )

Regular ( )

GRACIAS POR SU COLABORACIÓN

**12.5. ANEXO #5: Formato de Encuesta de Prueba Usabilidad de la Aplicación****UNIVERSIDAD NACIONAL DE LOJA  
ENCUESTA A LOS USUARIOS DEL SGAWEBSERVICES**

**Encuesta para la validación y comprobación del subsistema de servicios web basado en una biblioteca de recuperación de información del Sistema de Gestión Académica de la Universidad Nacional de Loja que permite la comunicación e interoperabilidad con aplicaciones relacionadas a fines académicos.**

Mediante la presente encuestas se pretende realizar la comprobación y validación de los métodos para obtención de información del SGA.

- **Considera usted que la información obtenida del Sistema de Gestión Académica mediante el SGAWebServices es:**

Importante ( )

Normal ( )

Regular ( )

- **Considera usted que el uso del SGAWebServices es:**

Fácil ( )

Medio ( )

Difícil ( )

- **Cuál de los método del SGAWebServices utilizaría usted?:**

sgaws\_lista\_areas ( )

sgaws\_datos\_area ( )

sgaws\_carreras\_area ( )

sgaws\_datos\_carreras ( )

sgaws\_modulos\_carrera ( )

sgaws\_paralelos\_carrera ( )

sgaws\_validar\_docente ( )



- sgaws\_validar\_estudiante ( )
- sgaws\_datos\_docente ( )
- sgaws\_datos\_estudiante ( )
- sgaws\_datos\_usuario ( )
- sgaws\_periodos\_lectivos ( )
- sgaws\_ofertas\_academicas ( )
- sgaws\_fechas\_matriculaoa ( )
- sgaws\_estadoestudiantes\_paralelo ( )
- sgaws\_plan\_estudio ( )
- sgaws\_carga\_horaria\_docente ( )
- sgaws\_notas\_estudiante ( )
- sgaws\_carreras\_estudiante ( )
- sgaws\_reporte\_matricula ( )
- sgaws\_nmatriculados\_oferta ( )
- sgaws\_nmatriculados\_area ( )
- sgaws\_nmatriculados\_carrera ( )
- sgaws\_nmatriculados\_modulo ( )
- sgaws\_nmatriculados\_paralelo ( )
- sgaws\_naprobados\_oferta ( )
- sgaws\_naprobados\_area ( )
- sgaws\_naprobados\_carrera ( )
- sgaws\_naprobados\_modulo ( )
- sgaws\_naprobados\_paralelo ( )
- sgaws\_nreprobados\_oferta ( )
- sgaws\_nreprobados\_area ( )
- sgaws\_nreprobados\_carrera ( )
- sgaws\_nreprobados\_modulo ( )
- sgaws\_nreprobados\_paralelo ( )



- **La documentación que indica la utilización de los métodos del SGAWebServices es:**

Explicita ( )

Superficial ( )

Confusa ( )

- **Considera usted que el acceso al SGAWebServices es:**

Fácil ( )

Medio ( )

Difícil ( )

GRACIAS POR SU COLABORACIÓN

## 12.6. ANEXO #6: Contenido de Script de creación de Usuarios del SGAWebservices.

```
# -*- coding: utf-8 -*-
#!/usr/bin/python

import MySQLdb

def comprobarUsuario(user):
    lista = []

db=MySQLdb.connect(host='localhost',user='sgaws',passwd='sgawebsservices',db='usuario')
    cursor = db.cursor()
    cursor.execute('select username from user')
    resultado = cursor.fetchall()
    for r in resultado:
        lista.append(r[0])
    ver = lista.count(user)
    if ver == 0:
        pass
    if ver ==1:
        print '.....El nombre del Usuario ya ha sido registrado
!!INTENTE CON OTRO NOMBRE!!.....'

def ingresarUsuarioBD():

    try:
        user = raw_input("Ingrese el nombre del usuario: ")
        user1 = raw_input("Ingrese nuevamente su usuario para
verificar: ")

        usuario = None
        clave = None

        if user == user1:
            verificar = "verdadero"
        else:
            verificar = "falso"

        if verificar == "verdadero":
            usuario = user
            password = raw_input("Ingrese la clave de su usuario
")
            password1 = raw_input("Ingrese la clave de su
usuario para verificar ")
            if password == password1:
                verificar1 = "verdadero"
            else:
                verificar1 = "falso"

            if verificar1 == "verdadero":
```

```
        clave = password
    else:
        print "Ha ingresado claves diferentes"
    else:
        print "Ha ingresado usuarios diferentes"

    if (not usuario or not clave):
        print ".....No ha ingresado correctamente el
nombre o la clave del usuario....."
    else:

db=MySQLdb.connect(host='localhost',user='sgaws',passwd='sgawebs
ervices',db='usuario')
        cursor=db.cursor()
        sql= 'INSERT INTO user
VALUES ("%s", "%s")' %(usuario, clave)

        comprobarUsuario(usuario)

        insertado = cursor.execute(sql)

        if insertado == 1:
            print ".....USUARIO INSERTADO
CORRECTAMENTE....."

        except:
            print "Intente Nuevamente"

if __name__ == "__main__":
    ingresarUsuarioBD()
```

## 12.7. ANEXO #7: Anteproyecto de Tesis



**UNIVERSIDAD NACIONAL DE LOJA**

**ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS  
RECURSOS NATURALES NO RENOVABLES**

**INGENIERÍA EN SISTEMAS**

***Anteproyecto de Tesis***

**RESPONSABLES:**

Ordóñez González Daysi Carolina

Pacheco Ludeña Silvana de los Ángeles

**DIRECTOR:**

Ing. René Rolando Elizalde Solano

**LOJA – ECUADOR**

**2009**



## 1. TITULO

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SUB SISTEMA DE SERVICIOS WEB (WEB SERVICES) BASADO EN UNA BIBLIOTECA DE MÉTODOS DE RECUPERACIÓN DE INFORMACIÓN DEL SISTEMA DE GESTIÓN ACADÉMICA DE LA UNIVERSIDAD NACIONAL DE LOJA, QUE PERMITA LA COMUNICACIÓN E INTEROPERABILIDAD CON APLICACIONES RELACIONADAS AFINES ACADÉMICOS.



## 2. PROBLEMÁTICA

### 2.1. Situación Problemática

Hoy en día el uso de aplicaciones web se ha constituido en el nuevo recurso para la prestación de servicios y soporte informático a las instituciones, empresas y sociedad en general, es por ello que es evidente la importancia del uso de tecnologías web para el desarrollo de aplicaciones informáticas.

Además podemos considerar que para el desarrollo de aplicaciones basadas en web, podemos encontrar una gran cantidad de tecnologías, muchas de ellas incompatibles entre sí, dificultando esto la integración de aplicaciones o sistemas informáticos, debido a esto nace la necesidad de hacer uso de servicios web (Web Services) que faciliten la intercomunicación e interoperabilidad entre diferentes arquitecturas, las cuales son la base principal dentro del desarrollo Web.

Los Web Services permiten la comunicación entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes y de manera independiente al lenguaje de programación, plataforma de implantación, formato de presentación o sistema operativo. Un Web Services es un contenedor que encapsula métodos específicos y hace que estos métodos puedan ser utilizados en otros servidores. Están contruidos con varias tecnologías que trabajan conjuntamente con estándares para asegurar la seguridad e interoperabilidad, de modo de hacer realidad, el uso combinado de varios servicios web.

La perspectiva de la sociedad universitaria actual demanda que las instituciones de educación superior posean soluciones y mecanismos de administración y control de los procesos académicos, aprovechado los avances recientes de la tecnología informática, ya que dentro del proceso administrativo y gestión universitaria existen diferentes actividades que deben ser realizadas mediante soluciones informáticas adaptadas a las características de la institución y al sistema académico vigente.

En la actualidad la Universidad Nacional de Loja posee un Sistema de Gestión Académica, automatizado, integrado y adaptado a las especificaciones del Sistema Académico Modular por Objetos de Transformación SAMOT, el mismo que está desarrollado bajo el paradigma del software libre sobre la plataforma Linux, con el propósito de llevar una mejor organización en las actividades que realiza diariamente, con el objetivo de permitir el control adecuado de información, en cuanto a lo referente a: planificación académica, matriculación, gestión de docentes, gestión de estudiantes, control académico, entre otros, son algunas de las múltiples prestaciones que brinda este sistema. Es importante destacar que el Sistema de Gestión Académica de la Universidad Nacional de Loja carece de un web services que le permita recuperar información del mismo así como también integrar aplicaciones que no estén contempladas o desarrolladas bajo sus lineamientos técnicos y de planificación.

Razón por la cual surge la necesidad de desarrollar una biblioteca de métodos de recuperación de información basada en la tecnología web services, que constara de: Seguridades: autenticación de usuarios, validación de usuarios; Consultas: Recuperación de datos personales de estudiantes y docentes, Recuperación de datos académicos como: recuperación de periodos de matrícula, recuperación de ofertas académicas de un módulo en las diferentes modalidades, recuperación de fechas en que se ofertaron los módulos de cada carrera, recuperación de notas, recuperación de asistencias, recuperación de planes de estudio por carrera, obtención de horarios por paralelos, recuperación de carga horaria de docentes, recuperación del tipo del plan de estudio vigente en un módulo con asistencia y aprobación obligatoria, recuperación de costos de matrículas en casos especiales; Datos Estadísticos: número y lista de estudiantes matriculados (áreas, carreras) modalidad presencial, semipresencial y a distancia, número de estudiantes reprobados (áreas, carreras) en las diferentes modalidades, número de estudiantes aprobados (áreas, carreras) en las diferentes modalidades, número y lista de egresados por áreas y carreras, listado de estudiantes por carrera y paralelos, listado de estudiantes por cursos.

Facilitando así obtención de información para aquellos usuarios que deseen hacer uso de la ésta, sin que haya la necesidad de duplicar la información, sin importar que las aplicaciones utilizadas estén desarrolladas en un lenguaje de programación o plataforma diferente al del Sistema de Gestión Académica.

## 2.2. Problema General de Investigación

Luego de realizado el análisis de los inconvenientes que se presentan por la falta del uso de servicios web se ha determinado el problema investigativo que se enuncia a continuación:

*“La recuperación de información desde nuevas aplicaciones que están desarrolladas en lenguajes de programación o sistemas operativos diferentes al del Sistema de Gestión Académica (SGA) de la Universidad Nacional de Loja, se ve afectada, debido a la inexistencia de servicios web (web services) que permita la comunicación e interoperabilidad entre ellas”*

## 2.3. Delimitación

El presente proyecto será implementado en el Sistema de Gestión Académica de la Universidad Nacional de Loja, dedicado a la organización y automatización de los procesos académicos de la misma. El problema de investigación se encuentra delimitado en base a los siguientes puntos.

### 2.3.1. Problemas específicos de investigación

Dentro de los problemas encontrados una vez realizada la situación problemática de nuestra investigación, hemos identificados los siguientes problemas específicos:

- Falta de una biblioteca de métodos que permitan la recuperación de información e integración de nuevas características al SGA-UNL usando otros lenguajes de programación.
- Inconvenientes al intercambiar datos debido a la incompatibilidad existente entre aplicaciones por estar desarrolladas en distintos lenguajes de programación.
- Problemas al establecer comunicación con otras aplicaciones.
- Dificultad para la reutilización de información existente para nuevas aplicaciones ya sea para uso particular en distintos proyectos.

### 2.3.2. Espacio

El presente proyecto de tesis será desarrollado en la ciudad de Loja en una de las instituciones de educación superior de gran prestigio, como lo es la Universidad Nacional de Loja en la Jefatura Informática de la misma y específicamente en el proyecto del Sistema de Gestión Académica.

### 2.3.3. Tiempo

Este proyecto se estima será realizado durante el transcurso de un año, cuyo inicio se proyecta desde el mes de octubre del 2009 a octubre del 2010, tiempo estimado para el desarrollo nuestra aplicación.

### 2.3.4. Unidades de Observación

Las unidades de observación que se tomarán en cuenta para el análisis diseño e implementación de nuestro trabajo tenemos:

- Sistema de Gestión Académica de la Universidad Nacional de Loja
- Tecnología de Servicios Web (Web services)
- Metodología de Programación Extrema (Xp)
- Python, WSDL, SOAP, XML, PHP.
- Pruebas de software.

### 3. JUSTIFICACIÓN Y VIABILIDAD

#### 3.1. JUSTIFICACIÓN

##### **Justificación Académica**

Previo a la obtención del grado de Ingeniero en Sistemas se realiza el presente trabajo investigativo que hará posible que se pongan en práctica los conocimientos adquiridos durante el transcurso de la vida estudiantil y debido a que la Universidad Nacional de Loja posee un Sistema Académico Modular por Objeto de Transformación (SAMOT) que propone a la investigación como uno de los pilares fundamentales en el proceso académico permitiéndonos estar en contacto con la realidad en la que vivimos y formándonos como personas críticas, creativas y propositivas.

##### **Justificación Técnica**

Promover una alternativa de solución que permitan establecer la comunicación e intercambio de información entre las diferentes aplicaciones que se encuentren desarrolladas en diferentes lenguajes o diferentes plataformas de desarrollo, realizando el análisis, diseño e implementación de una biblioteca de métodos basada la tecnología de Servicios web.

##### **Justificación Operativa**

El presente proyecto investigativo se justifica operativamente debido a que se dispone de los recursos humanos como: asesores e investigadores, así como también recursos financieros, técnicos y materiales los cuales nos permitirán realizar la investigación que servirá como base para el desarrollo de nuestro proyecto.

##### **Justificación Económica**

Desde el punto de vista económico el presente proyecto de tesis demanda costos moderados, en relación al beneficio que se obtendrá al implementar los componentes de software en base a la tecnología de Servicios Web. Así mismo cabe señalar que el costo que implica su elaboración va por cuenta de los desarrolladores.

### 3.2. VIABILIDAD

El presente trabajo se considera viable ya que luego del estudio realizado se cuenta con conocimientos teóricos sobre lo que son los servicios web, los mismos que nos servirán para el desarrollo de nuestro proyecto.

Además se cuenta con la apertura y receptividad en la zona a investigar brindada por el director de la Jefatura Informática de la Universidad Nacional de Loja y se posee la facilidad de acceder a la información gracias a la colaboración del director del Sistema de Gestión Académica de la UNL, también gracias a la disponibilidad de recursos económicos, técnicos, materiales, y tiempo suficiente para realizar el proyecto. Es necesario recalcar que se cuenta con la experiencia de los ingenieros en sistemas que laboran como docentes en la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja.

Y finalmente se considera viable ya que la realización del presente proyecto nos permitirá desarrollarnos como profesionales y consolidar los conocimientos y habilidades adquiridas durante el transcurso de nuestra formación académica.

## 4. OBJETIVOS

### 4.1. GENERAL

Analizar, diseñar e implementar un sub sistema de servicios web (web services) basado en una biblioteca de métodos de recuperación de información del sistema de gestión académica de la universidad nacional de Loja, que permita la comunicación e interoperabilidad con aplicaciones relacionadas afines académicos.

### 4.2. ESPECÍFICOS

- Realizar el estudio y documentación acerca de servicios web y el Sistema de Gestión Académica, que nos permitirá adquirir conocimientos sobre el funcionamiento, requerimientos, conceptos y herramientas que se utilizan para el desarrollo del mismo.
- Analizar, diseñar e implementar la biblioteca de métodos de recuperación de información que formaran parte del sub sistema de servicios web.
- Realizar una aplicación en un lenguaje de programación diferente al utilizado por el Sistema de Gestión Académica, con la finalidad de realizar las pruebas necesarias al producto desarrollado y depurar cualquier tipo de error.
- Documentar el funcionamiento y manejo de los métodos desarrollados para la recuperación de información, orientada a los usuarios que harán uso del SubSistema de Servicios Web.

## 5. MARCO TEÓRICO

### CAPÍTULO 1

#### 1. SISTEMA DE GESTIÓN ACADÉMICA DE LA UNIVERSIDAD NACIONAL DE LOJA

##### 1.1. QUE ES EL SGA?

La Universidad Nacional de Loja con el propósito de llevar una mejor organización en las actividades que realiza diariamente, tuvo la necesidad de crear el Sistema de Gestión Académica; cuyo propósito es permitir el control adecuado de información, en el momento del ingreso de planes de estudio, docentes, estudiantes, administrar calificaciones, asistencias, matriculación, etc., que son algunas de las múltiples prestaciones a las cuales personal administrativo, docentes, estudiantes, usuarios del SGA pueden acceder.

##### 1.2. CARACTERÍSTICAS TÉCNICAS

El Sistema de Gestión Académica fue desarrollado mediante el uso de la plataforma Linux ya que es un sistema operativo libre y cuenta con software de código abierto, por esta razón en el proceso de desarrollo de software se ha empleado:

**Phyton:** Lenguaje de programación de propósito general, orientado a objetos, que es usado para el desarrollo web, también es considerado un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

**Komodo:** Es una interesante solución para edición de código fuente multiplataforma y con soporte para diferentes lenguajes de programación. Soporta múltiples lenguajes como son Perl, PHP, Python, Ruby, Tcl, HTML, CSS, XML, JavaScript, RHTML, Template-Toolkit, HTML-Smarty y Django.



**PHP (acrónimo de “PHP: Hypertext Preprocessor”):** Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor, además ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día que el código comparables en otros lenguajes.

**Firefox:** Firefox es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL/MPL.

**Metodología xp:** Con la Programación Extrema, el enfoque utilizado para llevar a cabo la planificación es eminentemente pragmático. Gran parte de la eficacia de este modelo de planificación deriva de una división clara de responsabilidades, que tiene en cuenta las necesidades del negocio en todo momento.

**Arquitectura de 64 bits:** En arquitectura de ordenadores, 64 bits es un adjetivo usado para describir enteros, direcciones de memoria u otras unidades de datos que comprenden hasta 64 bits (8 octetos) de ancho, o para referirse a una arquitectura de CPU y ALU basadas en registros, bus de direcciones o bus de datos de ese ancho.

**Software Libre:** Es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre es muy importante.

## CAPÍTULO 2

### 2. WEB SERVICES<sup>47</sup>

El término Web Services (Servicios Web) describe una forma estandarizada de integrar aplicaciones web mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet, estos no proveen al usuario una interfaz gráfica en vez de ello comparten la lógica del negocio, los datos y los procesos, por medio de una interfaz de programas a través de la red. Conectan programas, por tanto son programas que no interactúan directamente con los usuarios.

Los servicios web permiten a distintas aplicaciones, de diferentes orígenes, comunicarse entre ellos sin necesidad de escribir programas costosos, esto porque la comunicación se hace con XML. No están ligados a ningún sistema operativo o lenguaje de programación. El modelo de computación distribuida de estos permite la comunicación de aplicación a aplicación.

#### 2.1. REQUISITOS DE UN WEB SERVICE<sup>48</sup>

- **Interoperabilidad:** Un servicio remoto debe permitir su utilización por clientes de otras plataformas.
- **Amigabilidad con Internet:** La solución debe poder funcionar para soportar clientes que accedan a los servicios remotos desde internet.
- **Posibilidad de aprovechar los estándares de Internet existentes:** La implementación del servicio remoto debería aprovechar estándares de Internet existentes tanto como sea posible.
- **Soporte para cualquier lenguaje:** La solución no debería ligarse a un lenguaje de programación particular Java RMI, por ejemplo, está ligada completamente a lenguaje Java. Sería muy difícil invocar funcionalidad de un objeto Java remoto desde Visual Basic o PERL.
- **Soporte para cualquier infraestructura de componente distribuida:** La solución no debe estar fuertemente ligada a una infraestructura de componentes en particular. De hecho, no se debería requerir el comprar, instalar o mantener

<sup>47</sup> <http://blog.espol.edu.ec/taws/category/web-services/>

<sup>48</sup> [http://www.gxtechnical.com/gxdlsp/pub/GeneXus/Internet/TechnicalPapers/Web\\_Services.htm](http://www.gxtechnical.com/gxdlsp/pub/GeneXus/Internet/TechnicalPapers/Web_Services.htm)

una infraestructura de objetos distribuidos, solo construir un nuevo servicio remoto utilizar un servicio existente.

## 2.2. BLOQUES CONSTRUCTIVOS DE SERVICIOS WEB

<b>Descubrimiento</b> UDDI,DISCO
<b>Descripción</b> WSDL, Esquema XML, Docs
<b>Formato de Mensaje</b> SOAP
<b>Codificación</b> XML
<b>Transporte</b> HTTP,SMTP y otros

- **Descubrimiento:** La aplicación cliente que necesita acceder a la funcionalidad que expone un Servicio Web necesita una forma de resolver la ubicación de servicio remoto. Se logra mediante un proceso llamado, normalmente descubrimiento.
- **Descripción:** Una vez que se ha resuelto el extremo de un servicio Web dado, el cliente necesita suficiente información para interactuar adecuadamente con el mismo. La descripción de un servicio web implica meta datos estructurados sobre la interfaz que intenta utilizar la aplicación cliente así como documentación escrita sobre el servicio web incluyendo ejemplo de uso.
- **Formato del mensaje:** Para el intercambio de datos, el cliente y el servidor tienen que estar de acuerdo en un mecanismo común de codificación y formato de mensaje.
- **Codificación:** Los datos que se transmiten entre el cliente y el servidor necesitan codificarse en un cuerpo de mensaje.

- **Transporte:** Una vez se ha dado formato al mensaje y se han serializado los datos en el cuerpo del mensaje se debe transferir entre el cliente y el servidor utilizando algún protocolo de transporte.

### 2.3. VENTAJAS DE UN WEB SERVICES<sup>49</sup>

Los servicios web apuntan a ser la piedra fundamental de la nueva generación de sistemas distribuidos. Estos son algunos puntos para fundamentar esta afirmación:

- **Interoperabilidad.-** Como los servicios web pueden ser implementados en cualquier lenguaje, los desarrolladores no necesitan cambiar sus ambientes de desarrollo para producir o consumir web services.
- **Fácil de utilizar.-** El concepto detrás de los web services es fácil de entender, incluso existen toolkits de vendedores como IBM o Microsoft que permiten a los desarrolladores crear web services en forma rápida y fácil.
- **Soporte de la Industria.-** Todas las empresas de software importantes soportan SOAP, e incluso están impulsando el desarrollo de web services.

---

<sup>49</sup> <http://sistemas3.wordpress.com/2007/06/14/web-services/>

## CAPÍTULO 3

### 3. TECNOLOGÍA WEB SERVICES<sup>50</sup>

Los Web Services están contruidos con varias tecnologías que trabajan conjuntamente con los estándares que están emergiendo para asegurar la seguridad y operatibilidad, de modo de hacer realidad que el uso combinado de varios Web Services, independiente de la o las empresas que los proveen, esté garantizado. Los estándares utilizados en el web services son:

#### 3.1. SOAP<sup>51</sup>

SOAP es un protocolo para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP. Está basado en XML, esto facilita la lectura por parte de los humanos, pero también los mensajes resultan más largos y, por lo tanto, considerablemente más lentos de transferir.

Existen múltiples tipos de modelos de mensajes en SOAP pero, el más común es el RPC, ya que proporciona un mecanismo estándar de empaquetar mensajes. SOAP ha recibido gran atención debido a que facilita una comunicación del estilo RPC entre un cliente y un servidor remoto. Los mensajes SOAP, son independientes del sistema operativo, y pueden transportarse en varios protocolos de internet como SMTP, MIME y HTTP.

##### 3.1.1. OBJETIVOS DE SOAP<sup>52</sup>

- Establecer un protocolo estándar de invocación de servicios remotos, basado en protocolos estándares de Internet: HTTP para la transmisión y XML para la codificación de datos.
- Independencia de plataforma, lenguaje de desarrollo e implementación (modelo de objetos).

##### 3.1.2. FUNCIONAMIENTO DE SOAP

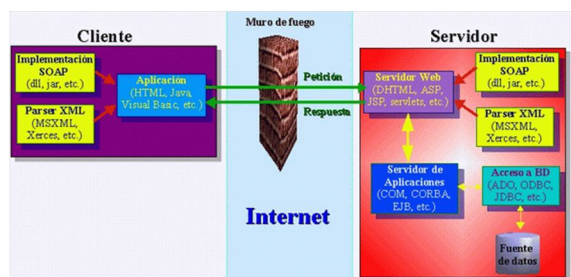
La especificación SOAP consiste en que las aplicaciones deben ser independientes del lenguaje de desarrollo, por lo que las aplicaciones cliente y servidor pueden estar

<sup>50</sup> <http://www.alegsa.com.ar/Dic/servicio%20web.php>

<sup>51</sup> <http://www.probandocodigo.com/2009/05/que-es-soap.html>

<sup>52</sup> <http://www.gxtechnical.com/gxdsp/pub/GeneXus/Internet/Docum/ReleaseNotes/7.5/CallSOAP.htm>

escritas con HTML, DHTML, Java, Visual Basic u otras herramientas y lenguajes disponibles. Las peticiones con el uso del protocolo HTTP emplean el comando POST para transmitir información entre el cliente y el servidor.



SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí. Estos mensajes constan de 3 secciones: envelope, header y body.



Dónde:

- **envelope (envoltura):** Es el elemento raíz del mensaje para describir su contenido y la forma de procesarlo.
- **header (encabezado):** Es la información de identificación del contenido. Un grupo de reglas de codificación para expresar las instancias de tipos de datos definidos por la aplicación.
- **body (cuerpo):** Es el contenido del mensaje. Una convención para representar las llamadas y las respuestas a procedimientos remotos.

### 3.2. XML<sup>53</sup>

<sup>53</sup>

<http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml>

XML es una tecnología muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. XML, es el estandar de Extensible Markup Language. XML no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

### 3.2.1. Principales características

- Es una arquitectura más abierta y extensible. No se necesita versiones para que puedan funcionar en futuros navegadores.
- Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML, en comparación a los atributos de la etiqueta <META> del HTML.
- Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.
- Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje permite agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- Gestión y manipulación de los datos desde el propio cliente web.
- Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible.
- Se desarrollarán de manera extensible las búsquedas personalizables y subjetivas para robots y agentes inteligentes.
- Permite un comportamiento más estable y actualizable de las aplicaciones web, incluyendo enlaces bidireccionales.

### 3.3. Estructura del XML

El metalenguaje XML consta de cuatro especificaciones (el propio XML sienta las bases sintácticas y el alcance de su implementación):

- **DTD.-** Es, en general, un archivo/s que encierra una definición formal de un tipo de documento y, a la vez, especifica la estructura lógica de cada documento. Define tanto los elementos de una página como sus atributos.
- **XSL.-** Define o implementa el lenguaje de estilo de los documentos escritos para XML. Permite modificar el aspecto de un documento. Se puede lograr múltiple columnas, texto girado, orden de visualización de los datos de una tabla.

### 3.3.1. WSDL: Web Services Description Language<sup>54</sup>

WSDL es un lenguaje basado en XML que se utiliza para describir un Web Services, ha sido suministrado por la W3C por estandarización.

Un archivo con formato WSDL provee información de los distintos métodos que el Web Services brinda, muestra cómo accederlos y que formatos deben de tener los mensajes que se envían y se reciben. Es como un contrato entre el proveedor del servicio y el cliente, en el cual el proveedor se compromete a brindar ciertos servicios solo si el cliente envía un requerimiento con determinado formato.

En la mayoría de los casos es conveniente que este acompañado por un documento escrito en lenguaje natural que brinde información de que es lo que hace cada uno de los métodos brindados por el Web Services así como también ejemplos, por ejemplo, de los mensajes SOAP que espera y responde el servicio. Un archivo WSDL describe lo siguiente:

- Mensajes que el servicio espera y mensajes que el servicio responde.
- Protocolos que el servicio soporta.
- A donde mandar los mensajes.

### 3.3.2. Formato de un archivo WSDL

El formato básico de un archivo WSDL consta de:

- **Type:** Describe los tipos no estándar usados por los mensajes.
- **Message:** Define los datos que contienen los mensajes pasados de un punto a otro.

---

<sup>54</sup> <http://www.w3.org/TR/wsdl.html>



- **PortType:** Define una colección de operaciones brindadas por el servicio. Cada operación tiene un mensaje de entrada y uno de salida que se corresponde con algún Message antes definido.
- **Binding:** Describe los protocolos que se utilizan para llevar a cabo la comunicación en un determinado PortType; actualmente los protocolos soportados son SOAP, HTTP GET, HTTP POST y MIME, siendo SOAP el más utilizado.
- **Port:** Define una dirección (URL) para un determinado Binding

### 3.3.3. UDDI<sup>55</sup>

UDDI (Universal Description, Discovery and Integration) es un proyecto inicialmente propuesto por Ariba, Microsoft e IBM; es un estándar para registrar y descubrir web services. La idea es que las distintas empresas registran su información acerca de los web services que proveen para que puedan ser descubiertas y utilizadas por potenciales usuarios.

La información es ingresada al registro de empresas UDDI, un servicio lógicamente centralizado, y físicamente distribuido a través de múltiples nodos los cuales replican su información en forma regular. Una vez que una empresa se registra en un determinado nodo del registro de empresas UDDI la información es replicada a los otros nodos y queda disponible para ser descubierta por otras empresas.

#### 3.3.3.1. Esquema UDDI

El modelo de información base utilizado por los registros UDDI es definido en un esquema XML. Este esquema define cuatro tipos básicos de información, cada uno de los cuales proveen la clase de información que un usuario necesita saber para utilizar un web service de otra empresa.

Los cuatro tipos de información son:

- **Información del negocio.-** Este tipo de información está definido en el elemento businessEntity. Contiene información de la empresa, como ser su nombre, los contactos, el tipo de empresa, etc.
- **Información del servicio.-** Dentro del elemento businessEntity se encuentran los elementos businessServices, estos elementos contienen información sobre

---

<sup>55</sup> [http://www.uddi.org/pubs/ProgrammersAPI\\_v2.pdf](http://www.uddi.org/pubs/ProgrammersAPI_v2.pdf)

web services generalmente agrupados por procesos de negocio o categorías de servicios.

- **Información del enlace.-** Dentro de cada elemento `businessServices` se encuentran los elementos `bindingTemplate`. Cada uno de ellos brinda una dirección física para hacer contacto con los servicios descritos anteriormente.
- **Información sobre las especificaciones del servicio.-** Cada `bindingTemplate` tiene asociado un `tModel`, el cual brinda información sobre las especificaciones del servicio.

### 3.4. XML-RCP<sup>56</sup>

Es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. Es muy simple ya que sólo define unos cuantos tipos de datos y comandos útiles, además de una descripción completa de corta extensión. La simplicidad del XML-RPC está en contraste con la mayoría de protocolos RPC que tiene una documentación extensa y requiere considerable soporte de software para su uso.

Una diferencia fundamental es que en los procedimientos en *SOAP* los parámetros tienen nombre y no interesan su orden, no siendo así en XML-RPC.

#### 3.4.1. Diferencia entre XMLRPC o SOAP

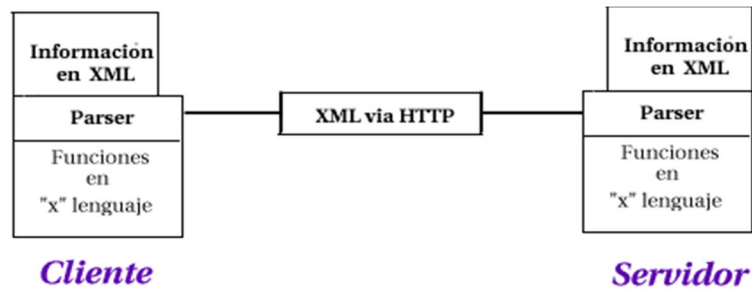
XMLRPC fue el primer mecanismo que surgió para invocar procedimientos remotos vía XML, ofrece una manera muy sencilla de invocar operaciones en sistemas heterogéneos a través de una estructura simple; SOAP es una implementación más robusta para llevar a cabo una intercomunicación en XML, a diferencia de XMLRPC a SOAP se le han integrado diversos mecanismos que le permiten operar en ambientes distribuidos más complejos tales como: un lenguaje neutro para su descripción WSDL y directorios distribuidos para su ubicación UDDI.

---

<sup>56</sup><http://es.wikipedia.org/wiki/XML-RPC>

### 3.4.1.1. Arquitectura de XMLRPC.<sup>57</sup>

En XMLRPC siempre se habla en términos de cliente/servidor, existe un sistema que realiza la solicitud ("el cliente") y otro que la atiende ("el servidor"), y como es de imaginarse un elemento clave en ambos puntos es: el parser XML.



<sup>57</sup> [http://xml.osmosislatina.com/cursos/XMLRPC y SOAP - Web Services.htm](http://xml.osmosislatina.com/cursos/XMLRPC%20y%20SOAP%20-%20Web%20Services.htm)

## CAPÍTULO 4

### 4. PRUEBAS DE SOFTWARE<sup>58</sup>

Las pruebas de software son elementos críticos para determinar la calidad del software. La importancia de los costos asociados a los errores promueven la definición y aplicación de un proceso de pruebas minuciosas y bien planificadas. Las pruebas permiten validar y verificar el software, entendiendo como validación del software el proceso que determina si el software satisface los requisitos, y verificación como el proceso que determina si los productos de una fase satisfacen las condiciones de dicha fase. Las estrategias de pruebas permiten enfocar el plan de pruebas; éste comprende la visión global del proceso de pruebas, y la definición de actividades y roles involucrados en cada una de ellas.

La prueba de software es el proceso de ejecutar el software de manera controlada. Y usualmente se usa en conjunto con los términos verificación y validación. Verificación es la revisión o las pruebas de los elementos, incluyendo el software para obtener conformidad y consistencia con una especificación asociada. La validación es el proceso de revisión que lo que se haya especificado es lo que el usuario realmente quiere.

#### 4.1. Estrategias de verificación

Las estrategias de verificación incluyen:

- Revisión de requerimientos
- Revisión del diseño
- Revisión general del código
- Inspección del código

#### 4.2. Los tipos de pruebas

Los tipos de pruebas que se pueden realizar son:

- Estrés
- Ejecución
- Recuperación
- Operación
- Cumplimiento
- Seguridad

---

<sup>58</sup> <http://www.citip.org.mx/servicios/software/pruebaslab/Pages/PruebasSoftware.aspx>

## CAPÍTULO 5

### 5. DOCUMENTACIÓN DE SISTEMAS

La documentación de sistemas es el conjunto de información que nos dice qué hacen los sistemas, cómo lo hacen y para quién lo hacen.

La documentación consiste en material que explica las características técnicas y la operación de un sistema. Es esencial para proporcionar entendimiento de un sistema a quien lo vaya a usar para mantenerlo, para permitir auditoría del sistema y para enseñar a los usuarios cómo interactuar con el sistema y a los operarios cómo hacerlo funcionar.

#### 5.1. Importancia De La Documentación De Sistemas

La importancia de la documentación bien podría ser comparada con la importancia de la existencia de una Póliza de Seguro; mientras todo va bien no existe la precaución de confirmar si nuestra Póliza de Seguros está o no vigente.

La documentación adecuada y completa, de una aplicación que se desea implantar, mantener y actualizar en forma satisfactoria, es esencial en cualquier Sistema de Información, sin embargo, frecuentemente es la parte a la cual se dedica el menor tiempo y se le presta menos atención.

#### 5.2. Manual De Usuario

Expone los procesos que el usuario puede realizar con el sistema implantado. Para lograr esto, es necesario que se detallen todas y cada una de las características que tienen los programas y la forma de acceder e introducir información. Permite a los usuarios conocer el detalle de qué actividades ellos deberán desarrollar para la consecución de los objetivos del sistema. Reúne la información, normas y documentación necesaria para que el usuario conozca y utilice adecuadamente la aplicación desarrollada.

#### Objetivos

- Que el usuario conozca cómo preparar los datos de entrada.
- Que el usuario aprenda a obtener los resultados y los datos de salida.
- Servir como manual de aprendizaje.



- Servir como manual de referencia.
- Definir las funciones que debe realizar el usuario.
- Informar al usuario de la respuesta a cada mensaje de error.



## 6. METODOLOGÍA

### 6.1. MATRIZ DE CONSISTENCIA GENERAL

**PROBLEMA GENERAL DE INVESTIGACIÓN (ENUNCIADO):** “La recuperación de información desde nuevas aplicaciones que están desarrolladas en lenguajes de programación o sistemas operativos diferentes al del Sistema de Gestión Académica (SGA) de la Universidad Nacional de Loja, se ve afectada, debido a la inexistencia de servicios web (web services) que permita la comunicación e interoperabilidad entre ellas”

TEMA	OBJETO DE INVESTIGACIÓN	OBJETIVO DE LA INVESTIGACIÓN	HIPÓTESIS DE INVESTIGACIÓN
Análisis, diseño e implementación de un sub sistema de servicios web (web services) basado en una biblioteca de métodos de recuperación de información del Sistema de Gestión Académica de la Universidad Nacional de Loja, que permita la comunicación e interoperabilidad con aplicaciones relacionadas afines académicos.	Tecnologías y metodologías aplicables en el análisis, diseño e implementación de un Sub Sistema de Servicios Web para recuperación de información del SGA.	Analizar, diseñar e implementar un sub sistema de servicios web (web services) basado en una biblioteca de métodos de recuperación de información del Sistema de Gestión Académica de la Universidad Nacional de Loja, que permita la comunicación e interoperabilidad con aplicaciones relacionadas afines académicos	El desarrollo de un web services dará mayor soporte al Sistema de Gestión Académica facilitando la comunicación, recuperación de información e interoperabilidad entre distintas aplicaciones.

## 6.2. MATERIALES, MÉTODOS Y TÉCNICAS DE TRABAJO

### 6.2.1. MATERIALES

#### Recursos Humanos

##### *Desarrolladores:*

- Daysi Carolina Ordóñez González
- Silvana de los Ángeles Pacheco Ludeña

#### Recursos Materiales

- CD'S
- Flash Memory
- Resmas papel
- Cartuchos de Tinta
- Suministros de Oficina
- Copias
- Anillados
- Empastados

#### Recursos Técnicos

##### *Comunicaciones:*

- Internet
- Llamadas telefónicas

#### Recursos Tecnológicos

##### *Hardware:*

- Computadores
- Impresoras

##### *Software:*

- Open Office V.3.0
- Python
- PHP
- XML
- Apache Tomcat
- Linux Ubuntu 9.0.1
- Internet



### 6.2.2. MÉTODOS

Para el desarrollo del presente trabajo se utilizará varios métodos entre los cuales tenemos:

**Método Inductivo - Deductivo.-** Este método se utilizará para establecer las diferentes definiciones que servirán para efectuar la documentación teórica, además nos permitirá realizar la recopilación y organización de la información que nos será útil para el desarrollo del presente proyecto.

**Método Analítico.-** Este método nos servirá para realizar un análisis del objeto en estudio. Se utiliza para efectuar un estudio a fondo de los problemas, causas y consecuencias que se están presentando por la falta de un Sub Sistema de Servicios Web (Web Services) que permita la comunicación entre las diferentes aplicaciones.

**Método Científico:** Se utilizará este método para realizar un estudio detallado del funcionamiento de un web services a través de las técnicas de observación de la forma de trabajar de esta tecnología.

**Método de Desarrollo de Software XP:** Utilizaremos esta metodología de desarrollo de software ya que es muy ágil y nos permitirá hacer adaptabilidades de cualquier cambio que se realice durante el progreso de la aplicación, lo cual nos permitirá tener éxito en el desarrollo del mismo, para lo cual nos basaremos en sus distintas fases que tiene esta metodología como: Planificación del proyecto, Diseño, Codificación. Pruebas. Promueve los valores de comunicación y simplicidad. Los individuos e interacciones son más importantes que los procesos y herramientas, el software que funciona es más importante que la documentación exhaustiva.

### 6.2.3. TÉCNICAS

Para la correcta realización del desarrollo del proyecto, se utilizarán técnicas que ayudaran a la recolección de datos e información acerca del tema en cuestión.

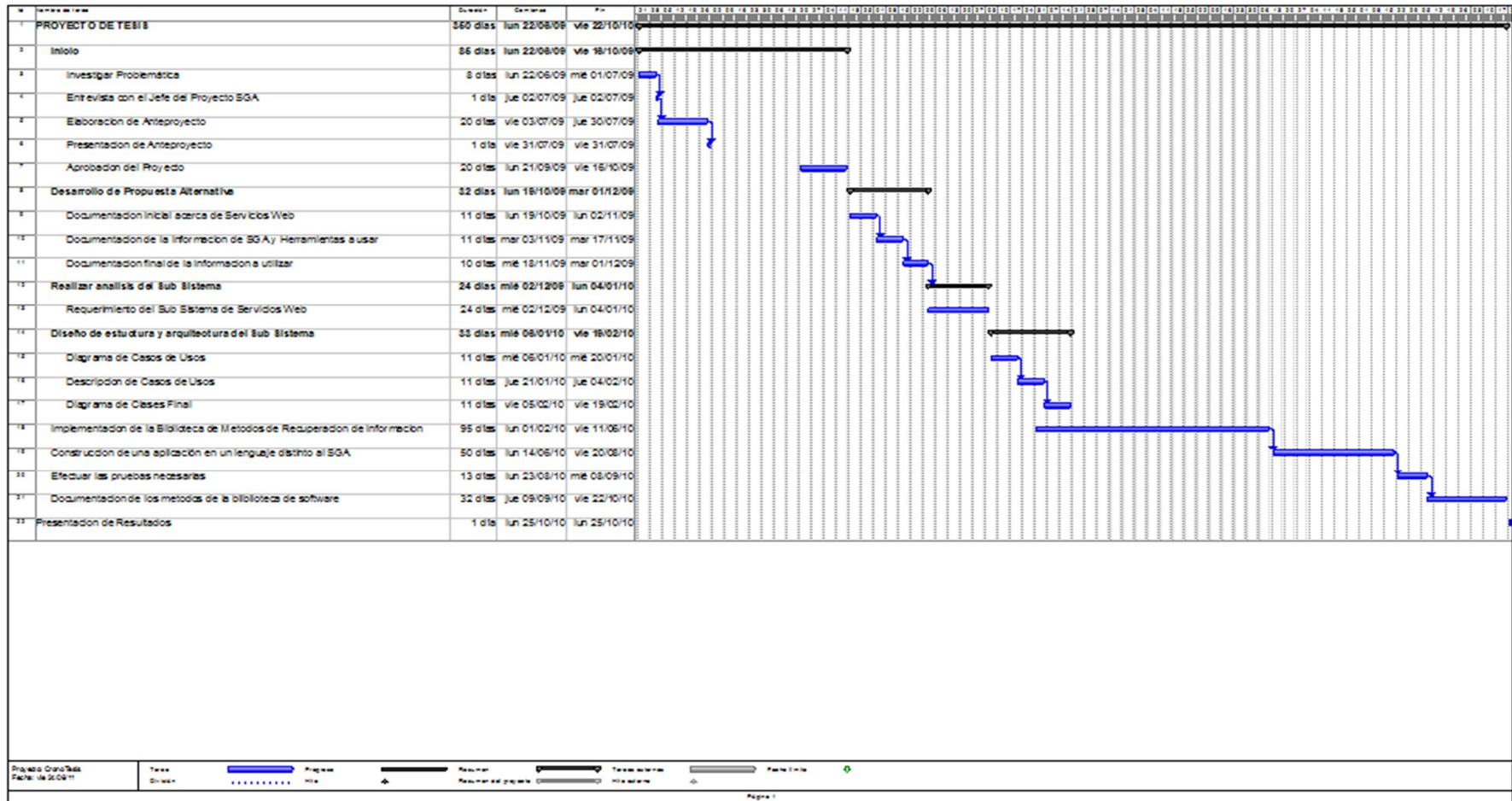
**Entrevista.-** Esta técnica será de gran utilidad ya que permitirá obtener los requerimientos de manera directa de las personas encargadas del Sistema de Gestión Académica lo cual nos permitirá orientarnos mejor para que nuestra aplicación cubra con las necesidades que aquí se presentan.



**Lectura comprensiva.**- Nos permitirá obtener un conocimiento ordenado y sistemático de los hechos o ideas relacionadas con el tema específico, además nos servirá para comprender correctamente como debemos efectuar la implementación de nuestro proyecto.



### 7. CRONOGRAMA



## 8. PRESUPUESTO Y FINANCIAMIENTO

### 8.1. PRESUPUESTO

#### Recursos Humanos

Recursos Humanos	Cantidad	Horas c/u	Costo/ Hora	Costo Total
Director de Tesis	----	-----	-----	----
Desarrolladores • <i>Daysi Ordóñez</i> • <i>Silvana Pacheco</i>	2	1000	\$4.00	\$8000.00
<b>TOTAL</b>				\$8000.00

#### Recursos Materiales

Materiales	Cantidad	Costo Unitario	Costo Total
Resma de Papel	5	\$3.00	\$15.00
Cartuchos de tinta.	5	\$5.00	\$25.00
Empastados	4	\$8.00	\$32.00
Copias	2000	\$0.02	\$40.00
Suministros de Oficina	---	---	\$30.00
Anillados	8	\$1.00	\$8.00
CD	5	\$1.00	\$5.00
Flash Memory (1GB)	2	\$7.00	\$14.00
<b>TOTAL</b>			\$169.00

#### Recursos Técnicos

Comunicaciones	Horas	Costo/Hora	Total
Internet	100	\$1.00	\$100.00
Llamadas telefónicas	25	---	\$50.00
<b>TOTAL</b>			\$150.00

**Recursos Tecnológicos**

Recursos Tecnológicos		Costo Unitario	Costo Total			
<b>SOFTWARE</b>						
Python 2.5		Gratuito	\$0.00			
PHP 5.2.11		Gratuito	\$0.00			
XML		Gratuito	\$0.00			
Open Office V.3.0		Gratuito	\$0.00			
Linux Ubuntu 9.0.1		Gratuito	\$0.00			
Apache Tomcat		Gratuito	\$0.00			
<b>TOTAL</b>			\$0.00			
Recursos Técnicos:	Cantidad	Horas	Hojas	Costo		Costo
<i>Comunicaciones</i>				hora	Hoja	Total
Computadores	2	924		0.50		\$462.00
Impresora	1		2500		0.05	\$125.00
<b>TOTAL</b>						\$587.00

**Resumen del Presupuesto**

Recursos Humanos	\$8000.00
Recursos Materiales	\$169.00
Recursos Técnicos	\$150.00
Recursos Tecnológicos	\$587.00
<b>SUBTOTAL</b>	<b>\$8906.00</b>
Imprevistos 10 %	\$890.60
<b>TOTAL PROYECTO</b>	<b>\$9796.60</b>



## 8.2. FINANCIAMIENTO

Luego de haber realizado el análisis de recursos a utilizar para la puesta en marcha de nuestro proyecto, se estima un gasto de **9796.60** dólares, el cual puede variar de acuerdo.

De la misma manera cabe recalcar que el financiamiento del proyecto lo asumirán las integrantes del mismo.

## 9. BIBLIOGRAFÍA

### Fuentes Terciarias

- Centro de Innovación de TI, [http://www.citip.org.mx/servicios/software/pruebaslab/Pages/PruebasSoftware.aspx], [Consulta:02-07-2009]
- Universidad de Alicante, 1996, Web Services, [http://cv1.cpd.ua.es/ws/default.asp], [Consulta: 25-06-2009]
- Guiarte Multimedia S.L., 2001, Qué es XML, [http://www.desarrolloweb.com/articulos/449.php], [Consulta: 25-06-2009]
- Licencia Creative Commons, Wikipedia, 2000, SOAP, [http://es.wikipedia.org/wiki/SOAP], [Consulta: 25-06-2009]
- Licencia Creative Commons, Wikipedia, modificado 30 jun 2009, XML-RPC, [http://es.wikipedia.org/wiki/XML-RPC], [Consulta:02-07-2009]
- Gxtechnical, Web Services, 2005, [http://www.gxtechnical.com/gxdlsp/pub/GeneXus/Internet/TechnicalPapers/Web\_Services.htm], [Consulta: 02-07-2009]
- Rodolfo A. Diéguez H, Monografías.com, Documentación de Sistemas, [http://www.monografias.com/trabajos6/dosi/dosi.shtml], [Consulta:02-07-2009]
- ProbandoCodigo.com, 2009, SOAP = Services Object Access Protocol, [http://www.probandocodigo.com/2009/05/que-es-soap.html], [Consulta: 02-07-2009]
- Castillo Botello, Alejandro, 2007, Construcción de Servicios Web con SOAP, [http://www.revista.unam.mx/vol.3/num1/art3/], [Consulta: 25-06-2009]
- Valdivieso Daniel, 2000, Ingeniería Web, Web Services, [http://sistemas3.wordpress.com/2007/06/14/web-services/], [ 25-06-2009]



## 10. ANEXOS

### 10.1. Matriz de consistencia específica

<b>PROBLEMA ESPECÍFICO:</b> La falta de servicios web(web services) dificulta la comunicación y obtención de información, debido a la incompatibilidad existente entre aplicaciones, por estar desarrolladas en distintos lenguajes de programación, al del Sistema de Gestión Académica (SGA) de la Universidad Nacional de Loja			
<b>OBJETIVO ESPECÍFICO</b>	<b>HIPÓTESIS ESPECÍFICA</b>	<b>UNIDAD DE OBSERVACIÓN</b>	<b>SISTEMA CATEGORIAL</b>
Realizar el estudio y documentación acerca de servicios web y el Sistema de Gestión Académica, que nos permitirá adquirir conocimientos sobre el funcionamiento, requerimientos, conceptos y herramientas que se utilizan para el desarrollo del mismo.	Con la implementación del Sub Sistema de Servicios web se facilitará y dará soporte para la comunicación y recuperación de información	Sistema de Gestión Académica (SGA) de la Universidad Nacional de Loja Servicios web	<ul style="list-style-type: none"><li>• Que es el SGA?</li><li>• Características Técnicas</li><li>• Web services</li><li>• Ventajas de los web services</li><li>• Requisitos de los web services</li><li>• Bloques constructivos de servicios web</li></ul>





**PROBLEMA ESPECÍFICO:** Falta de documentación necesaria para realizar el análisis, diseño e implementación del Sub Sistema de servicios web para la recuperación de los datos contenidos en el SGA

<b>OBJETIVO ESPECÍFICO</b>	<b>HIPÓTESIS ESPECÍFICA</b>	<b>UNIDAD DE OBSERVACIÓN</b>	<b>SISTEMA CATEGORIAL</b>
Analizar, diseñar e implementar la biblioteca de métodos de recuperación de información que formaran parte del sub sistema de servicios web.	El análisis nos permitirá obtener los requerimientos necesarios para la realización de nuestra aplicación, así mismo el correcto diseño nos ayudará a crear los métodos que formarán parte de nuestro Sub Sistema de servicios web para la recuperación de información del SGA y la implementación donde se darán las instrucciones o procedimientos necesarios para obtener el software deseado en el lenguaje de programación a utilizar.	Sistema de Gestión Académica (SGA) de la Universidad Nacional de Loja Metodología para desarrollo de sistema XP (Programación Extrema) Tecnología de Web services Herramientas para el desarrollo del Software	<ul style="list-style-type: none"><li>• Tecnología web services</li><li>• SOAP</li><li>• Objetivos del SOAP</li><li>• Funcionamiento del SOAP</li><li>• Estructura del XML</li><li>• WSDL: Web Services Description Language</li><li>• Esquema UDDI</li><li>• XML-RCP</li></ul>



**PROBLEMA ESPECÍFICO:** Falta de una biblioteca de métodos que permitan obtener información contenida en el SGA-UNL y el uso inadecuado de la aplicación al no conocer claramente el funcionamiento de la misma.

<b>OBJETIVO ESPECÍFICO</b>	<b>HIPÓTESIS ESPECÍFICA</b>	<b>UNIDAD DE OBSERVACIÓN</b>	<b>SISTEMA CATEGORIAL</b>
Documentar el funcionamiento y manejo de los métodos desarrollados para la recuperación de información, orientada a los usuarios que harán uso del SubSistema de Servicios Web.	<ul style="list-style-type: none"><li>• La implantación del software dará mayor soporte para la comunicación e interoperabilidad entre aplicaciones</li><li>• La documentación del funcionamiento y manejo de los métodos desarrollados permitirá a los usuarios su uso adecuado</li></ul>	<ul style="list-style-type: none"><li>• Sistema de Gestión Académica de la Universidad Nacional de Loja</li><li>• Usuarios del Sub Sistema de servicios web</li></ul>	<ul style="list-style-type: none"><li>• Documentación de sistemas</li><li>• Importancia de la documentación de sistemas</li><li>• Manual de usuario</li></ul>

**10.2. Matriz de operatividad de objetivos específicos**

ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
Buscar Información acerca de la tecnología de servicios web	<ul style="list-style-type: none"><li>• Consultas bibliográficas</li><li>• Lectura comprensiva</li></ul>	19-10-2009	02-11-2009	Daysi Ordoñez Silvana Pacheco	\$ 40.00	Documentación inicial acerca de servicios web
Recolectar la información necesaria del SGA y herramientas a utilizar en el desarrollo de nuestra aplicación	<ul style="list-style-type: none"><li>• Consultas bibliográficas</li><li>• Técnica de entrevista</li><li>• Lectura comprensiva</li></ul>	03-11-2009	17-11-2009	Daysi Ordoñez Silvana Pacheco	\$40.00	Documentación de la información necesaria acerca del SGA Documentación de la información necesaria las herramientas a utilizar



Determinar la información más relevante	<ul style="list-style-type: none"> <li>Lectura comprensiva</li> </ul>	17-11-2009	01-12-2009	Daysi Ordoñez Silvana Pacheco	\$20.00	Documento final de la información a utilizar
---	---	------------	------------	----------------------------------	---------	--

**OBJETIVO ESPECÍFICO:** Analizar, diseñar e implementar la biblioteca de métodos de recuperación de información que formaran parte del sub sistema de servicios web.

ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
Realizar el análisis	<ul style="list-style-type: none"> <li>Técnica de entrevista</li> <li>XP (Programación extrema)</li> </ul>	02-12-2009	04-01-2010	Daysi Ordoñez Silvana Pacheco	\$500	Requerimientos del Sub sistema de servicios web



Elaborar el diseño	<ul style="list-style-type: none"><li>• XP (Programación extrema)</li></ul>	05-01-2010	19-02-2010	Daysi Ordoñez Silvana Pacheco	\$1000	Diseño de la estructura y arquitectura de la aplicación plasmado en el diagrama de clases
Construcción de la biblioteca de métodos de recuperación de información	<ul style="list-style-type: none"><li>• XP (Programación extrema)</li></ul>	19-02-2010	11-06-2010	Daysi Ordoñez Silvana Pacheco	\$1000	Transformación del diagrama de clases final en la aplicación deseada



**OBJETIVO ESPECÍFICO:** Realizar una aplicación en un lenguaje de programación diferente al utilizado por el Sistema de Gestión Académica, con la finalidad de realizar las pruebas necesarias al producto desarrollado y depurar cualquier tipo de error.

ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
Construir una aplicación en un lenguaje de programación distinto al del SGA	<ul style="list-style-type: none"><li>• Método Deductivo</li><li>• XP(Programación Extrema)</li><li>• Método Analítico</li></ul>	15-06-2010	20-08-2010	Daysi Ordoñez Silvana Pacheco	\$ 500.00	Aplicación que compruebe el funcionamiento del Web Services
Efectuar las pruebas necesarias que nos ayuden a comprobar la efectividad de nuestra aplicación	<ul style="list-style-type: none"><li>• Método Deductivo</li><li>• XP(Programación Extrema)</li><li>• Método Analítico</li></ul>	20-08-2010	08-09-2010	Daysi Ordoñez Silvana Pacheco	200.00	Funcionamiento de la Aplicación que permitirá comprobar el funcionamiento del Web Services.



**OBJETIVO ESPECÍFICO:** Documentar el funcionamiento y manejo de los métodos desarrollados para la recuperación de información, orientada a los usuarios que harán uso del SubSistema de Servicios Web.

ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
Documentar los métodos de la bibliotecas de software	4. Método Analítico	13-09-2010	22-10-2010	Daysi Ordoñez Silvana Pacheco	\$ 500.00	Documentación del Sub sistema de servicios web con sus respectivos métodos desarrollados.

**10.3. Matriz de control de resultados**

No.	RESULTADOS	FECHA	FIRMA DEL DOCENTE
1	Documentación inicial acerca de servicios web	02-11-2009	
2	Documentación de la información necesaria acerca del SGA	17-11-2009	
3	Documentación de la información necesaria las herramientas a utilizar	17-11-2009	
4	Documento final de la información a utilizar	01-12-2009	
5	Requerimientos del Sub sistema de servicios web	04-01-2010	
6	Diseño de la estructura y arquitectura de la aplicación plasmado en el diagrama de clases	19-02-2010	
7	Transformación del diagrama de clases final en la aplicación deseada	11-06-2010	
8	Aplicación que compruebe el funcionamiento del Web Services	20-08-2010	
9	Funcionamiento de la Aplicación que permitirá comprobar el funcionamiento del Web Services.	08-09-2010	
10	Documentación del Sub sistema de servicios web con sus respectivos métodos desarrollados.	22-10-2010	






**12.8. ANEXO #8: Listado de métodos migrados del sgaservices del Sistema de Gestión Académica al SGAWebservices**

- sga\_acad\_arancelinfo
- sga\_acad\_carrerainfo
- sga\_acad\_modulo
- sga\_acad\_paralelo\_hs
- sga\_acad\_paraleloinfo
- sga\_acad\_userinfo
- sga\_auth\_docente\_login
- sga\_auth\_user\_exists
- sga\_auth\_user\_login
- sga\_auth\_userlist
- sga\_docente\_carreras
- sga\_docente\_info



## 12.9. ANEXO # 9 Certificación de la sección de software de UNL

  
**UNIVERSIDAD NACIONAL DE LOJA**  
UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN  
SECCIÓN DESARROLLO DE SOFTWARE

---

Memorando No. UNL-SDS-2011-MEM-0050

**PARA:** Lic. MgSc. Edgar Jamil Ramón Carrión  
**DIRECTOR DE TELECOMUNICACIONES E INFORMACIÓN**

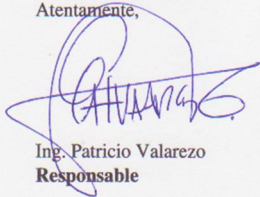
**ASUNTO:** PProyecto WebService SGA

**FECHA:** 15 de Junio de 2011

De conformidad con la solicitud del Ing. René Elizalde, director del proyecto de Tesis cuya problemática versa sobre: "Análisis , Diseño e Implementación de un Subsistema de Servicios Web (WebService) basado en una biblioteca de métodos de recuperación de información del Sistema de Gestión Académica de la Universidad Nacional de Loja, que permita la comunicación e interoperatividad con aplicaciones relacionadas a fines Académicos." de revisar el software desarrollado por los responsables de éste módulo del proyecto, srts Silvana de los Angeles Pacheco y Daysi Carolina Ordoñez, se ha procedido a la revisión del producto, tomando en cuenta: aspectos técnicos, lenguaje de programación, plataforma de desarrollo y lógica de algoritmos, así como se ha procedido a su implementación en uno de los servidores del SGA, tanto en su producto final como con los productos derivados demostrativos, y se ha determinado que el producto cumple con los lineamientos técnicos definidos por la Sección de Desarrollo de Sistemas, recibiendo a cabalidad los productos derivados del presente trabajo..

Particular que pongo a su conocimiento para los fines pertinentes.

Atentamente,



Ing. Patricio Valarezo  
**Responsable**

PV

C.C.: Ingeniero en Sistemas  
René Elizalde  
**DESARROLLADOR**  
UNIVERSIDAD NACIONAL DE LOJA

1/1

\* Documento generado por Qispur SGD - UNL