UNIVERSIDAD NACIONAL DE LOJA

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

CARRERA DE INGENIERÍA EN SISTEMAS

"DISEÑO Y CONSTRUCCIÓN DE UN KIOSKO ELECTRÓNICO PROTOTIPADO PARA LA IMPRESIÓN AUTOMÁTICA DE CERTIFICADOS ACADÉMICOS PARA LOS ESTUDIANTES DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES".

Tesis previa a la obtención del Título de Ingeniero en Sistemas.

AUTORAS:

Alexandra Esperanza Jadán Patiño

Mayra Vanessa Loja Sandoya

DIRECTOR:

Ing. Juan Manuel Galindo Vera, Mg. Sc.

LOJA - ECUADOR

2015

CERTIFICACIÓN

Ing. Juan Manuel Galindo Vera, Mg. Sc.

A petición de la parte interesada,

CERTIFICA:

Que la presente tesis titulada **"Diseño Y Construcción de un Kiosko Electrónico Prototipado para la Impresión Automática de Certificados Académicos para los Estudiantes del Área de la Energía, las Industrias y los Recursos Naturales No Renovables"**, realizada por las egresadas Alexandra Esperanza Jadán Patiño y Mayra Vanessa Loja Sandoya, ha sido orientada, revisada, corregida y aprobada a cabalidad, cumpliendo así con todos los requisitos y reglamentos establecidos por la carrera de Ingeniería en Sistemas; por lo que autorizo su presentación.

Loja, julio de 2014

Ing. Juan Manuel Galindo Vera, Mg. Sc.
DIRECTOR DE TESIS

AUTORIA

Nosotros **ALEXANDRA ESPERANZA JADÁN PATIÑO** y **MAYRA VANESSA LOJA SANDOYA,** declaramos ser autoras del presente trabajo de tesis y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación de nuestra tesis en el Repositorio Institucional – Biblioteca Virtual.

Autor: Alexandra Esperanza Jadán Patiño

Firma:

Cédula: 1103954556

Fecha: 13 febrero de 2015

Autor: Mayra Vanessa Loja Sandoya

Firma:

Cédula: 1103741672

Fecha: 13 febrero de 2015

CARTA DE AUTORIZACION DE TESIS POR PARTE DE LOS AUTORAS, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Nosotros, **ALEXANDRA ESPERANZA JADÁN PATIÑO** y **MAYRA VANESSA LOJA SANDOYA** declaramos ser autores de la tesis titulada: **"DISEÑO Y CONSTRUCCIÓN DE UN KIOSKO ELECTRÓNICO PROTOTIPADO PARA LA IMPRESIÓN AUTOMÁTICA DE CERTIFICADOS ACADÉMICOS PARA LOS ESTUDIANTES DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES",** como requisito para optar al grado de: **INGENIEROS EN SISTEMAS;** autorizamos al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga el convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, 02 días del mes de febrero del dos mil quince.

Firma:

Autor: Alexandra Esperanza Jadán Patiño
Cédula: 1103954556
Dirección: Loja, Paraguay y Paltas
Correo Elect.: alexaejp2080@hotmail.com
Teléf.: 2107795 Celular: 0993194182

Firma:

Autor: Mayra Vanessa Loja Sandoya
Cédula: 1103741672
Dirección: Loja, Amaluza y Sucre
Correo Elect.: vanel26@hotmail.com
Teléf.:2107795 **Celular:** 0994323129

DATOS COMPLEMENTARIOS:

Director de Tesis: Ing. Juan Manuel Galindo Vera, Mg. Sc.
Tribunal de Grado: Ing. Ana Lucía Colala Troya, Mg.Sc.
Ing. Alex Vinicio Padilla Encalada, Mg. Sc.
Ing. Marco Augusto Ocampo Carpio.

AGRADECIMIENTO

A la Universidad Nacional de Loja, a través de la Carrera de Ingeniería en Sistemas a sus autoridades y al personal administrativo. Así mismo presentamos nuestro más sincero agradecimiento al Ing. Milton Palacios y a todas las autoridades de la Universidad que de una u otra manera nos permitieron tener acceso a la información necesaria para el desarrollo de esta tesis.

De igual manera un eterno agradecimiento a todos los docentes que nos brindaron sus sabias enseñanzas durante los años de estudio de la carrera universitaria, a quienes debemos gran parte de nuestros conocimientos.

Finalmente este proyecto de tesis ha sido posible gracias al esfuerzo conjunto realizado con nuestro Director de Tesis el Ing. Juan Manuel Galindo, de manera especial dejamos constancia de nuestra gratitud hacia él por su valiosa ayuda durante el desarrollo de este trabajo investigativo.

Las Autoras

DEDICATORIA

A mis PADRES, por su apoyo incondicional en toda mi vida, en especial a mi MADRE que con su abnegación y sacrificio ha sabido ayudarme haciendo posible la culminación de mis estudios universitarios.

A mi ESPOSO y a MIS HIJOS David, Aylin Victoria y Aylin del Cisne, por ser el motor que me impulsa a seguir adelante con las metas que a veces parecen inalcanzables, pero que hoy una de ellas se hace realidad.

A mis hermanos por su apoyo y sus palabras de impulso y motivación.

Finalmente a mis profesores y autoridades de la AEIRNNR que han hecho posible mi formación y término de estudios superiores.

Alexandra

La concepción de este proyecto de tesis está dedicada a la tenacidad y lucha insaciable de mis padres, pilares fundamentales en mi vida.

A mi esposo y a mis hijos, Ariel, Matías, Valeria y Gabriel que han sido mis compañeros en cada jornada. Ellos representan el esfuerzo y tesón en momentos de decline y cansancio.

A mis maestros de la Universidad por sus enseñanzas y ejemplo de profesionalidad que nunca olvidaré.

A ellos este proyecto, que sin ellos, no hubiese podido ser.

Vanessa

A. TÍTULO:

“DISEÑO Y CONSTRUCCIÓN DE UN KIOSKO ELECTRÓNICO PROTOTIPADO PARA LA IMPRESIÓN AUTOMÁTICA DE CERTIFICADOS ACADÉMICOS PARA LOS ESTUDIANTES DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y DE LOS RECURSOS NATURALES NO RENOVABLES”

B. RESUMEN

El presente trabajo de titulación se centra en el Diseño y Construcción de un Kiosko Electrónico Prototipado para la Impresión Automática de Certificados Académicos para los Estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables, la cual tiene como objetivo principal aportar en el desarrollo de sistemas automatizados para la gestión académica de la Universidad Nacional de Loja (U.N.L.) mediante la generación del software y las especificaciones de hardware de un Kiosko Electrónico Prototipo, que ofrezca a los estudiantes un servicio de generación e impresión de certificados académicos de forma automática y basado en tecnologías Web 2.0, permitiendo a esta Institución Educativa ofrecer un servicio ágil, eficiente y de calidad.

Iniciando con un proceso de determinación de la demanda del servicio de certificaciones Académicas de los estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables de la U.N.L. Para lo cual se debió seleccionar el toolkits javascript de generación de tecnologías Web 2.0, que mejor se adapte a las necesidades y alternativas técnicas más convenientes, estableciendo las especificaciones de Hardware y Sistema Operativo para la operación brindada por el Kiosko.

El sistema operativo del Kiosko interactuará de manera segura y eficiente con el sistema de gestión Académica de la Universidad Nacional de Loja ya que cuenta un sistema gráfico seleccionado y colocado específicamente para la interacción y vinculación de esta información.

Mediante las pruebas de funcionalidad y usabilidad se pudo verificar el adecuado y eficiente servicio del Kiosko y sus componentes; coincidentes con los requerimientos planteados inicialmente, mejorando notablemente la eficiencia, eficacia y calidad de este servicio en nuestra institución Universitaria.

SUMMARY

This thesis deals with the theme : "**Design and Construction of an Electronic Kiosk Prototyping for Automatic Printing Certificates for Students Academic Area Energy Industries and Non- Renewable Natural Resources**", whose main objective is to create a tool information that will improve services broadcast transcripts capable of meeting the requirements of students AEIRNNR .

For the design of the tool were performed: diagram use cases, class diagrams and sequence diagrams with the help of Enterprise Architect, a platform for modeling software to analyze, design and document applications, and offering a detailed way to generate the physical data model.

As NetBeans 6.5 programming environment which supports the surprising and prototype javascript library jQuery with which developed the solution plus some snippets of php and latex is used also allows the use of pattern MVC (Model-View -Driver) needed for web design.

The implementation of the Jquery library containing common functionalities of DOM, events, effects and AJAX is a fundamental tool in building application allowing you to create effects and custom animations.

With all these tools is able to build the application and its operation has been duly confirmed by the users of the Universidad Nacional de Loja.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN II

AUTORIA III

CARTA DE AUTORIZACION DE TESIS IV

AGRADECIMIENTO V

DEDICATORIA vi

A. TÍTULO: 7

B. RESUMEN 8

ÍNDICE DE CONTENIDOS 10

C. INTRODUCCIÓN 18

D. REVISIÓN DE LITERATURA 20

1. Metodología y Técnicas 20

1.1. Metodología de desarrollo: Iconix 20

Fase 1: Análisis de requisitos 21

Fase 2: Análisis y diseño preliminar 21

Fase 3: Diseño 21

Fase 4: Implementación 22

1.2. Método Deductivo 22

1.3. Método Inductivo 22

1.4. Técnicas 23

2. Kiosko Electrónico 25

2.1 Ventajas de los kioskos electrónicos 25

2.2 Aplicaciones de kiosko: Proyecto kiosko Linux 26

2.3. kiosko Browser 26

2.4. Creación Kiosko en Linux 27

3. SGA – Sistema de Gestión Académica 33

3.1. Servicios del Sistema de Gestión Académico 33

3.2. Descripción del servicio WSDL 38

4. Web 2.0 39

4.1. Orígenes del Término 40

4.2. Tecnologías que dan vida a un proyecto Web 2.0 40

4.3. ¿En qué nos sirve la Web 2.0? 41

5. Toolkits de javascript 41

5.1. DojoToolkits 43

5.2. Script.aculo.us 46

6. Herramientas para el desarrollo 48

6.1. Ajax 48

6.2. Javascript 50

6.3. Librería jquery 51

7. Apache 2.0 53

7.1 Características 54

8. LaTex 54

8.1 Caracteres reservados de LaTeX 56

9. Sistema integrado, empotrado o embebido 56

9.1. Componentes de un Sistema Integrado 57

9.2. Aplicaciones de un Sistema Integrado 58

10. Circuito Impreso o PCB 59

10.1 Soldermask o máscara de soldado 60

10.2. Serigrafía o silkscreen 61

10.3. Stackup o pila de capas 62

10.4. Encapsulado de componentes 63

10.5. Pads 64

10.6. Caminos de cobres (pistas o tracks) 65

10.7. Perforaciones Metalizadas a través de orificio 66

10.8. Perforaciones Metalizadas ciegas (Blind Vías) 67

10.9. Perforaciones Metalizadas enterradas u ocultas 69

10.10. Construcción de una PCB (Fotolitografía) 69

11. Uml 76

11.1. Clases y Objetos 76

11.2. Diagramas de Clases 77

11.3. Diagramas de Objetos. 78

11.4. Modelo de Casos de Usos 78

11.5. Diagramas de Paquetes 81

11.6. Diagramas de Interacción 83

11.7. Diagrama de Secuencia. 83

E. MATERIALES Y MÉTODOS 84

1. Métodos 84

2. Materiales 85

2.1 Componentes hardware 85

2.2. Software 85

2.3. Materiales de Oficina 86

F. RESULTADOS 87

1. Antecedentes 88

G. DISCUSIÓN 91

1. Desarrollo de la propuesta alternativa 91

1.1 Objetivo General 91

1.2 Objetivos Específicos 91

1.3. Glosario de términos 93

1.4. Documento de requerimientos 94

1.5. Modelo de la aplicación 96

1.6. Modelo dinámico 98

1.7. Modelo y Arquitectura 114

1.8 Construcción del Teclado 116

2. Valoración técnica económica ambiental 119

2.1 Valoración técnica económica 119

2.2. Valoración Ambiental 121

H. CONCLUSIONES 123

I. RECOMENDACIONES 124

J. BIBLIOGRAFÍA 125

k. ANEXOS 127

INDICE DE FIGURAS

Figura 1. Tecnología Web 2.0 41
Figura 2. Widgets del toolkit..... 42
Figura 3. Compilación de documentos en Látex..... 55
Figura 4. PCB de 2 o más layers 60
Figura 5. Expansión de la máscara de soldado 61
Figura 6. Ejemplos de Stackups 63
Figura 7. Pads SMD y Thru-hole 65
Figura 8. Tracks que interconectan 2 circuitos impresos 65
Figura 9. Dos chips en caras opuestas se conectan 66
Figura 10. Track de la capa superior atravesando la PCB 67
Figura 11. Comparación de una Thru-hole con una Blind 68
Figura 12. Comparación de una Thru-hole, Blind y Burried..... 69
Figura 13. Diseño de circuito en positivo y en negativo 70
Figura 14. Emisión UV..... 71
Figura 15. Reacción provocada en la resina..... 71
Figura 16. Reacción frente a la insolación..... 71
Figura 17. Contacto de la placa con revelador 72
Figura 18. Placa sin pistas conductoras 72
Figura 19. Recorte de bordes sobrantes..... 74
Figura 20. Soldadura con cautín y estaño 75
Figura 21. Material sobrante recortado con alicates..... 75
Figura 22. Modelo de Diagrama de Clase..... 76
Figura 23. Objetos 77
Figura 24. Modelo de Diagrama de Caso de Uso……..79
Figura 25. Representación de actor..... 79
Figura 26. Representación de Caso de Uso..... 80

Figura 27. Representación de una Asociación 80
Figura 28. Representación de una Dependencia..... 80
Figura 29. Representación de una Generalización 80
Figura 30. Modelo de Diagrama de Paquetes 82
Figura 31. Modelo del dominio 96
Figura 32. Modelo del dominio de la Aplicación 97
Figura 33. Diagrama de Casos de Uso 98
Figura 34. Menú principal para emisión de certificados..... 99
Figura 35. Menú principal para emisión de certificados..... 100
Figura 36. Diagrama de secuencia, Caso de Uso: Obtener un certificado (Flujo normal) 103
Figura 37. Diagrama de secuencia Caso de Uso: Obtener un certificado (Flujo alterno A)..... 104
Figura 38. Ventana “Autenticar un usuario”..... 105
Figura 39. Diagrama de secuencia Caso de Uso: Autenticar un usuario (Flujo normal)..... 107
Figura 40. Diagrama de secuencia Caso de Uso: Autenticar un usuario (Flujo alterno A) 107
Figura 41. Ventana Principal de Administración de parámetros del sistema. 108
Figura 42. Diagrama de secuencia Caso de Uso: Administrar parámetros del sistema (Flujo normal) 110
Figura 43. Diagrama de secuencia Caso de Uso: Administrar paráme0tros del sistema (Flujo alterno A)..... 111
Figura 44. Diagrama de secuencia Caso de Uso: Administrar parámetros del sistema (Flujo alterno B)..... 111
Figura 45. Pantalla para administración del kiosko 112

Figura 46. Modelo de Kiosko 115
Figura 47. Construcción Teclado 116
Figura 48. Pinaje del Teclado 117
Figura 49. Pinaje Específico 118
Figura 50. Esquema de coordenadas del teclado, generación de números 118

INDICE DE TABLAS

TABLA I: MÉTODOS QUE RETORNAN INFORMACIÓN ACADÉMICA 33
TABLA II: MÉTODOS QUE RETORNAN INFORMACIÓN DE LA INSTITUCIÓN.....35
TABLA III: METODOS QUE RETORNAN INFORMACIÓN PERSONAL 35
TABLA IV. MÉTODOS PARA VALIDACIÓN 36
TABLA V: MÉTODOS QUE RETORNAN INFORMACIÓN ESTADÍSTICA.....36
TABLA VI: ORDEN DE PILAS DE CAPAS CAD.....62
TABLA VII: ENCAPSULADOS ELECTRÓNICOS 63
TABLA VIII: SIGNIFICADO DE COLORES DE UN CIRUITO IMPRESO.....66
TABLA IX: REQUERIMIENTOS FUNCIONALES 94
TABLA X: REQUERIMIENTOS NO FUNCIONALES.....95
TABLA XI: CASO DE USO OBTENER CERTIFICADO.....100
TABLA XII:CASO DE USO AUTENTICAR UN USUARIO.....105
TABLA XIII: CASO DE USO ADMINISTRAR PARÁMETROS....108
TABLA XIV: CASO DE USO OBTENER REPORTES DE USO ..112
TABLA XV: RECURSOS HUMANOS 119
TABLA XVI: RECURSOS MATERIALES.....119
TABLA XVII: RECURSOS TÉCNICOS Y TECNOLÓGICOS.....120
TABLA XVII. PRESUPUESTO DEL PROYECTO 121

C. INTRODUCCIÓN

A través de los tiempos el ser humano se encuentra en constante evolución y desarrollo, basado especialmente a la solución de necesidad y requerimientos de manera ágil y eficiente, es por esto que se han dado grandes avances de la tecnología y estos han hecho que los recursos en sus diferentes tipos sean manejados de mejor manera, garantizando con ello el éxito de los estamentos que utilizan estas herramienta0s tecnológicas innovadoras.

Por ello se ha considerado pertinente realizar el proyecto de Diseño y Construcción de un Kiosko Electrónico Prototipado para la Impresión Automática de Certificados Académicos para los Estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables (A.E.I.R.N.N.R), de la Universidad Nacional de Loja (U.N.L.). El mismo que consiste en un equipo electrónico que contiene una aplicación que interactúa con los usuarios a través de una pantalla y un teclado tipo push, siendo una solución similar a los servicios un cajero automático.

De acuerdo a los problemas encontrados en la emisión de certificados, se ha determinado que el problema principal y considerado para ser solucionado en este proyecto es “La falta de un prototipo tecnológico que brinde un servicio de generación e impresión de certificados académicos de forma automática, eficiente y de calidad, y que use tecnología tipo Web 2.0”.

Para dar cumplimiento al presente trabajo de titulación, en este documento se desarrolla la revisión de la literatura que sirvió como base para obtener la información necesaria tanto de software como de hardware, que nos permita determinar las herramientas adecuadas para el diseño y construcción de nuestra aplicación. Seguidamente se describen los materiales y métodos que se aplicaron, así como las técnicas que facilitaron la obtención de los datos y procedimientos, que se utilizan y llevan a cabo para la emisión de los certificados académicos.

Después se detallan los resultados obtenidos de acuerdo a los objetivos específicos que son el camino para cumplir con el objetivo general del presente proyecto. Empezando con el primer objetivo que consiste en disponer de la información necesaria de la

demanda de las certificaciones en el AEIRNNR se determinan los certificados más solicitados por los estudiantes y que serán generados por nuestra herramienta. El segundo objetivo trata de seleccionar el toolkits javascript de generación de tecnologías Web 2.0, que mejor corresponda a las necesidades y alternativas técnicas más convenientes, para ellos se ha revisado las diferentes librerías de JavaScript para elegir la más conveniente para la realización del proyecto. Luego se establecen las especificaciones de Hardware y Sistema Operativo para la operación dedicada del Kiosko, cumpliendo con el tercer objetivo y finalmente concluimos con la generación de un Software Gráfico derivado del toolkit javascript elegido, embebido como proceso dedicado del Sistema Operativo del Kiosko y que interactué con el Sistema de Gestión Académica de la Universidad Nacional de Loja.

Posteriormente se realiza la discusión en la cual se plasmó de forma general los objetivos de acuerdo a los resultados obtenidos y finalmente se encuentran las conclusiones y recomendaciones que son expuestas de acuerdo a la culminación del presente trabajo de titulación.

Con este proyecto, el Área de la Energía, las Industrias y de los Recursos Naturales no Renovables dará un servicio de vanguardia, que permitirá mejorar los tiempos, disminuir los recursos económicos, humanos y a la vez permitirá convertir a la Universidad en una institución mucho más eficiente, eficaz y dadora de servicios de calidad.

D. REVISIÓN DE LITERATURA

En el presente proyecto de titulación se fundamenta en los siguientes ítems teóricos:

1. Metodología y Técnicas

Los métodos se pueden considerar como el camino más adecuado para lograr un fin, se puede decir también que el método es el conjunto de procedimientos lógicos a través de los cuales se plantean los problemas científicos y se ponen a prueba las hipótesis y los instrumentos de trabajo a investigar.

1.1. Metodología de desarrollo: Iconix

ICONIX, es una metodología que consiste en un lenguaje de modelamiento y un proceso de desarrollo de software práctico. Es un proceso dirigido, como RUP (Rational Unified Process), relativamente pequeño y ligero, como XP(eXtreme Programming). Es un proceso simplificado, que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto. Da soporte y conocimiento a una metodología del mismo nombre (1993) que presenta de forma clara las actividades de cada fase y la secuencia que debe ser seguida, está adaptado a los patrones y ofrece el soporte de UML[1].

1.1.1. Características de la Metodología Iconix

- Iterativo e incremental: Suceden iteraciones entre el desarrollo de modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos [1].

- Trazabilidad: Cada paso está referenciado por algún requisito. Se debe considerar a la trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos [1].

- Dinámica del UML: Uso dinámico de UML en los diagramas de caso de uso, diagramas de secuencia y de colaboración [1].

1.1.2. Fases principales de la Metodología Iconix

Fase 1: Análisis de requisitos

En esta fase se realizan las siguientes tareas [3]:

- Modelo del dominio
- Elaboración rápida de prototipos
- Modelo de casos de usos

Fase 2: Análisis y diseño preliminar

En esta fase se realizan las siguientes tareas [3]:

- Descripción de los casos de uso
- Diagramas de robustez

En base a la recolección de información en esta etapa nos centramos a la redacción y descripción de los casos de uso de nuestro sistema como son: Obtener certificado, Autenticar usuario, Administrar parámetros del sistema y Obtener reportes de uso.

Fase 3: Diseño

Dentro de esta fase se realiza la siguiente tarea:

- Diagramas de secuencia

Para realizar los diagramas de secuencia de nuestro sistema, nos basamos en cómo funciona la obtención de certificados en el kiosko, para poder establecer la interacción del sistema con el estudiante y administrador.

Fase 4: Implementación

Dentro de esta fase se realiza la siguiente tarea:

- Escribir y generar código

En esta nos dedicamos a generar el código en base al análisis realizado en la fase anterior, con los componentes seleccionados para el desarrollo de nuestra aplicación, en esta fase podemos señalar que la conexión con el webservice se la realiza mediante una clave de usuario y contraseña de acceso al SGA del Web service de la Universidad, empleando los métodos creados en el mismo.

1.2. Método Deductivo

El método deductivo nos permite conocer las posibilidades de solucionar el problema abordado en el anteproyecto; el cual se desarrolla en las siguientes fases en una forma secuencial e interactiva:

1. Metodología Iconix
2. Metodología de la Investigación
3. Gestión Académica de la Universidad Nacional de Loja
4. Gestión Académica del Área de la Energía, las Industrias y los Recursos Naturales no Renovables.
5. Uso de la Tecnología RAD
6. Utilización de Web 2.0
7. Utilización de Toolkits JavaScript, Apache y Latex
8. Utilización de UML
9. Circuitos Impresos PCB

1.3. Método Inductivo

Este método de investigación se lo utilizó en la presente investigación partiendo de una causa específica como es la impresión automática de certificados académicos basados en la utilización de UML para el análisis y diseño de la herramienta que permita generar

dichos documentos, en donde para la construcción se recurrió a conocer y comprender la construcción de circuitos impresos en PCB para generar el hardware necesario para que el estudiante pueda obtener los certificados en forma rápida, confiable y fácil, descartando los largos procesos que en la actualidad se llevan para obtener un certificado académico.

El presente proyecto toma real importancia para solucionar la problemática de emisión de certificados debido a que el proyecto se ubica en el tiempo, espacio social académico y tecnológico del Área de la Energía, las Industrias y los Recursos Naturales no Renovables, además de ser un proyecto factible e interactuante en su funcionamiento con el usuario que está apoyado en objetivos determinantes.

1.4. Técnicas

Las técnicas que se han empleado en el presente proyecto de investigación son:

1.4.1. Entrevistas

La entrevista siempre tiene un objetivo, que debe ser conocido al menos por el entrevistador y que será tenido en cuenta antes y durante su desarrollo. Por lo tanto es un proceso de interacción con un fin determinado [3].

En el presente proyecto de investigación se empleó la técnica de la entrevista con el fin de obtener información sobre el uso del Sistema de Gestión Académica, las mismas que fueron aplicadas a los Administradores del SGA y a las Sras. Secretarias de las carreras del Área de Energía, las Industrias y los Recursos naturales no renovables, lo que nos ha dado una orientación de la necesidad de investigar la construcción de herramientas que generen soluciones viables.

Para mayor referencia de utilización de esta técnica de búsqueda de información de acuerdo a los requerimientos de usuario, se indican en el Anexo 2.

1.4.2. Cuestionario

Otra de las técnicas de investigación utilizadas en el presente proyecto de titulación es el cuestionario que se define como un documento formado por un conjunto de preguntas que deben estar redactadas de forma coherente, y organizadas, secuenciadas y estructuradas de acuerdo con una determinada planificación, con el fin de que sus respuestas nos puedan ofrecer toda la información que se precisa [2].

En el presente proyecto de investigación se ha empleado la técnica del cuestionario, formado por un conjunto de preguntas puntuales la misma que se realizó a los estudiantes de la Carrera de Electromecánica del Área de Energía, las Industrias y los Recursos Naturales no Renovables. Para constatación de ésta técnica ver el Anexo 3.

1.4.3. Observación

Finalmente se utilizó la técnica de la observación para indagar la información requerida en el presente trabajo, la cual se fundamenta en la siguiente definición: procedimiento de recopilación de datos e información consistente en utilizar los sentidos para observar hechos y realidades presentes, y a actores sociales en el contexto real (físico, social, cultural, laboral, etc.) en donde desarrollan normalmente sus actividades [2].

En esta fase se aplica encuestas a estudiantes y entrevistas a los administradores del SGA, para conocer la forma de obtención de los certificados, y así priorizar los certificados más solicitados por parte del estudiantado del AEIRNNR, con esto se determinan los requerimientos funcionales y no funcionales, para luego establecer el modelo del dominio y de acuerdo al proceso de obtención del certificado generar los prototipos necesarios en el despacho de los mismos y finalmente modelar los casos de uso del sistema.

Mediante esta técnica se evidenció de forma activa como se desarrollan las actividades de los usuarios del SGA, además nos permitió afianzar la información recabada a través de las técnicas de la entrevista y el cuestionario.

2. Kiosko Electrónico

Un kiosko electrónico (kiosko computarizado o kiosko interactivo) contiene una terminal de cómputo que a menudo utiliza software específico para kiosko diseñado para trabajar sin fallas y evitar que el usuario acceda a funciones del sistema. De hecho, el modo kiosko del software define a esta forma de uso del software. Los kioskos electrónicos pueden almacenar la información de manera local, o traerla de una red de cómputo [16].

Algunos kioskos electrónicos proveen acceso gratuito a servicios públicos, mientras que otros proveen acceso a información de empresas privadas. Touchscreens, trackballs, teclados, y botones son dispositivos típicos de interacción con un kiosko electrónico [16].

Para poder obtener información de opinión de la ciudadanía dentro de estos kioskos se necesita un sistema que sea flexible y se pueda adaptar a cambios, la tecnología que se ha detectado como la más adecuada es la de aplicaciones web [16].

Los Kioskos Electrónicos son módulos de autoservicio que funcionan como una ventanilla única automatizada donde se ofrecen múltiples servicios y que están situados en lugares estratégicos con el fin de acercarlos al usuario [4].

Esta herramienta digital ayuda al usuario en el ahorro de tiempo, y otros recursos al momento de solicitar alguna transacción.

Sus componentes corresponden a la integración de sistemas, dispositivos electrónicos y base de datos, el cual nos permite tener ingreso por el número de cédula generando cantidad de transacciones mayor a los generados con la emisión de transacciones de forma manual [4]-[16].

2.1 Ventajas de los kioskos electrónicos

Las principales ventajas de los kioskos electrónicos son:

- **Coste Reducido:** Porque el tiempo necesario para obtener el servicio suele ser reducido y esto ahorra costes de personal.

- **Comodidad:** Para el usuario, porque no tiene que esperar, ni hacer colas, ni ajustarse a un horario.
- **Facilidad de distribución:** Por que las máquinas se puede instalar en muchos lugares, como ocurre con los cajeros automáticos.

2.2 Aplicaciones de kiosko: Proyecto kiosko Linux

El Proyecto Kiosko es un sistema de kiosko basado en GNU/Linux que es simple y robusto. Hay temas especiales que deben abordarse en el establecimiento de un sistema de kiosco. Estos son la interfaz de usuario, los gestores de ventanas. En este apartado se tratará acerca de las herramientas y la información que se pueden utilizar para configurar un sistema de kiosko basado en la tecnología de montaje de Mozilla y GNU/Linux [15].

2.3. Kiosko Browser

Kiosko Browser es una aplicación XUL que proporciona un conjunto de imágenes para firefox y en la distro de kioskBrowser personalizado (basado en Mozilla Firefox). El concepto aquí es tener un navegador que no hace más que navegar no hay características de lujo y bloqueado.

Diseñado para un sistema de estilo kiosco. Hay una serie de ilustraciones de kiosko creado previamente en la distro. Es posible crear nuevas interfaces (imágenes) para el kioskbrowser utilizando los módulos incluidos (como superposiciones dinámicas).

El navegador kiosko funciona en todas las versiones de mozilla y obras (probados) en versiones de Firefox hasta 2.0.0.6. También hay una distribución navegador Kiosko completo basado en Firefox que tiene características adicionales que no son posibles con un XPI solo [5].

2.4. Creación Kiosko en Linux

Las páginas siguientes describen el proceso necesario para crear una instalación de kiosco web segura de Linux. El kiosko se ejecuta una instalación mínima de Linux e inmediatamente inicia un navegador web.

Este kiosko es esencialmente un aparato de navegación web. La belleza de Linux es que el sistema operativo puede ser personalizado para realizar una sola tarea, dejando de lado todos los servicios de software y red innecesarios. El sistema también se puede bloquear muy firmemente para evitar que un usuario habitual pueda correr cualquier otro programa o hacer cualquier modificación. La seguridad física (contraseña del BIOS y ambiente sellado) es la defensa final para proteger la configuración del software [15].

El booteo del kiosko se introduce en un navegador de pantalla completa (Mozilla Firefox) iniciado por la cuenta sin privilegios "invitado”. La cuenta de invitado no tiene acceso al Shell por lo que no tiene usuario, ni contraseña de inicio de sesión. No existe ninguna otra cuenta, excepto la cuenta de root, que está protegido por contraseña, incluso en el modo de un solo usuario [15].

Firefox está configurado para no permitir el acceso de usuarios invitados al sistema de archivos, ni para almacenar contraseñas, ni para almacenar en caché cualquier cosa, etc. Si el usuario cierra el navegador, X se reinicia y se vuelve a crear un nuevo usuario en el home directory. El kiosko también se reinicia utilizando el mismo procedimiento usando el protector de pantalla, después de 5 minutos de inactividad [15].

Los detalles del proceso de arranque son estos. El registro del sistema y un firewall bidireccional (netfilter) se activan en el arranque. El sistema de auto arranque está en el nivel de ejecución 5, entonces inittab ejecuta un script que limpia la cuenta de invitado y empieza X como el usuario invitado.

La xsession de Invitado lanza el protector de pantalla y el script de monitoreo de inactividad, seguido por el gestor de ventanas Metacity (necesario para permitir que los menús de Firefox para que funcione correctamente), y finalmente el propio Firefox. Si X es abortado, se regenerado por inittab como se acaba de describir. El usuario no puede

iniciar un reinicio, pero puede reiniciar manualmente X al cerrar el navegador o la secuencia de teclas Ctrl -Alt- Retroceso [15].

El kiosko se puede construir y ejecutar en varios modos. Se puede ejecutar como un "LIVECD", es decir, inicia y se ejecuta desde un CD, de modo que el sistema operativo no puede ser alterado en absoluto. En ese caso, los cambios y las actualizaciones deben ser realizados por la imagen del LiveCD remasterizado. También puede ser instalado en el disco duro, luego administrado desde allí mediante la herramienta de actualización de Debian (apt-get) [15]-[4].

En este caso el kiosko se puede configurar según sea necesario con una página de inicio por defecto y la lista blanca de sitios accesibles. Por último, puede ser PXE a arrancar y ejecutar diskless (una opción si muchos de estos equipos deben ser manejados convenientemente). El servicio SSH está ejecutando en la máquina en su configuración actual (el único servicio expuesto) para permitir la supervisión y el mantenimiento remoto.

Actualmente los scripts de booteo se han incorporado en un LiveCD basado en Knoppix. Esto permite a los usuarios ejecutar directamente desde el CD o instalar en el disco duro mediante el arranque en modo monousuario. El livecd tiene una raíz por defecto establecer una contraseña, boothbox, el ssh está desactivado, y la página de inicio es about:blank . Se recomienda que el livecd sea remasterizado a por lo menos establecer un passwd para root. Sin passwd de root sería trivial para alguien para arrancar en modo monousuario.

2.4.1. Personalización con Firefox

La mayoría de las configuraciones de Firefox están incorporadas dentro de la extensión de booteo ubicado en la raíz del livecd. A pesar de que termina en una extensión .dsl en realidad es sólo un gzip comprimido. Esto puede ser descomprimido en un directorio aparte y los archivos editados. Dentro de la extensión está un archivo llamado guest.tar.gz (usr / share / booth / guest.tar.gz) que contiene el contenido del directorio principal del usuario invitado que se regenera a partir del archvie cada vez que Firefox se inicia. Si la página de inicio necesitaba ser cambiado el archivo de configuración

user.js en el directorio de perfil de Firefox situado en guest.tar.gz necesitaría ser editada [15]-[16].

Example:

```
tar -xzf guest.tar.gz
cd guest/.mozilla/firefox/c2d9ocbh.default/
vi user.js
```

Las siguientes líneas pueden ser cambiadas para ser modificadas a cualquier url que desee cambiar.

```
user_pref("browser.startup.homepage", "about:blank");
user_pref("browser.throbber.url", "about:blank");
```

Hay otras opciones de configuración en el archivo que pueda ser de su interés, tales como la configuración del proxy. Después de realizar los cambios que el archivo debe ser embalados de nuevo.

```
tar -cf - guest | gzip - > guest.tar.gz
rm -rf guest
```

En el directorio de perfil de Firefox hay un llamado cromo sub-directorio que contiene el archivo userChrome.css que puede ser editado para manipular muchos de los elementos de la interfaz.

Lista Blanca

Tenga en cuenta que cualquier versión 0.3.2 después del Live CD basado en DSL ya no utiliza el programa 'lista blanca'. En cambio, todos los cambios se deben hacer a través tinyproxy. En el archivo de configuración tinyproxy.conf hay una línea [15]

FilterDefaultDeny: que permite la creación de cualquiera de una lista blanca o una lista negra junto con el /etc/tinyproxy/filter

Por ejemplo, sólo para permitir a los usuarios acceder a sitios web desde dentro de su dominio que podría utilizar la siguiente configuración.

/etc/tinyproxy/tinyproxy.conf:FilterDefaultDeny Yes
/etc/tinyproxy/filter:mydomain.com

Con estos ajustes sólo el tráfico que va a los sitios en mydomain.com se permitirá a través. Tenga en cuenta que todo el tráfico está configurado para pasar por tinyproxy el fin de generar estadísticas de uso para el kiosko [15].

Para activar proxy en el navegador firefox necesita cambiar guest/.mozilla/firefox/ c2d9ocbh.default/user.js user_pref ("network.proxy.type", 1); Tinyconf debe ejecutarse como usuario "invitado " o como grupo de "invitado" [15].

Cuando se utiliza el xserver Xvesa en Damn Small Linux puede especificar la resolución y otros ajustes editando el .xserverrc archivo. Este archivo normalmente se genera automáticamente en el arranque, pero se puede especificar su propia si desea ejecutar una resolución menor. Desempaquete la extensión stand - 0.xxdsl y extraiga el archivo guest.tar.gz en usr/share/stand. Después de haber hecho de este lugar un archivo llamado .xserverrc en el directorio de invitados y re empaqueta los archivos [15].

Una vez que el CD se ha vuelto a mezclar con los cambios que ha hecho que se iniciará en la resolución especificada. He aquí un ejemplo .xserverrc:exec /usr/bin/X11/Xvesa -mouse "/dev/psaux",5 -screen 800x600x32 -shadow -nolisten tcp -I &>/dev/null [4].

2.4.2. Impresión

El soporte para la impresión actualmente es en forma de cupsys-cliente. La impresión puede ser configurado mediante el establecimiento del **ServerName,** variable de configuración en /etc/cups/client.conf al servidor CUPS que tiene configuración. Todos los trabajos de impresión serán dirigidos al servidor CUPS especificado. Este archivo se puede encontrar en la extensión stand - 0.xxdsl en la raíz de la imagen livecd [15].

2.4.3. Acceso Terminal

Por defecto, el livecd arrancará directamente en X y empezar firefox junto con algunos scripts de ayuda que se ejecutan en segundo plano. Sin embargo, es útil ser capaz de

tener acceso a la terminal para fines de depuración y remastering. El acceso a los ordenadores está disponible a través del modo de usuario único. En el arranque del sistema, escriba 'single dsl', después de que se complete la detección automática de hardware que se le solicitará una contraseña de root. La contraseña de root por defecto para el livecd es 'boothbox'.

2.4.4. Modo de presentación

Firefox se puede configurar para actuar como una pantalla estática o dinámica. Cuando se combina con un poco de javascript una diapositiva se puede crear, por ejemplo enginfo. Algunas cosas deben cambiarse desde la configuración cabina estándar. En primer lugar de todos los elementos emergentes es probable que necesite que se le permita. De acuerdo con el apartado anterior en la personalización de Firefox el entorno [15]:

user_pref("dom.disable_open_during_load", false)

Para permitir elementos emergentes. El archivo .xinitrc en guest.tar.gz debe ser modificada y la inactividad, xscreensaver, y líneas icewm debe ser comentada. Para la elaboración de un cursor x transparente para que no se muestra el puntero. Sus pasos fueron los siguientes:

Crear un directorio llamado /home/guest/.icons/default/cursors/
Copiar un archivo PNG completamente transparente para el directorio.
Crear un archivo: 'left_ptr.conf' que contiene la siguiente línea:
<literal> 32 0 0 blankCursor.png </ literal>
(Donde blankCursor.png es el nombre del archivo PNG transparente)
- Ejecutar: 'xcursorgen left_ptr.conf left_ptr' y debe crear un cursor presentar desde el archivo PNG.

Sólo el cursor puntero será invisible. Todos los otros cursores voluntad siguen apareciendo (por ejemplo, cuando las páginas se cargan el cursor reloj aparecerá). Para quitar el cursor, simplemente se borra el archivo.

Para desactivar el borrado de pantalla se añaden las siguientes líneas al archivo .xinitrc en guest.tar.gz

setterm -blank 0 -powersave off -powerdown 0 xset s off

Para desactivar la barra de estado del archivo userChrome.css debe editarse, la única línea que necesita ser añadido así:

#status-bar {display:none !important;}

Ahora los elementos emergentes están habilitados, el cursor por defecto es transparente, el protector de pantalla se desactiva y la decoración de la ventana ya no está presente. Todo lo que hay que hacer es establecer la página principal para el sitio configurado para mostrar la presentación.

Firefox fue elegido como el navegador kiosco debido a las extensiones disponibles para ello y la facilidad con que su interfaz de usuario puede ser manipulada. Tres extensiones se instalaron ocultación automática, restablecimiento de inactividad, y la ventanilla única.

Estas extensiones permiten abrir mozilla en modo de pantalla completa, la información de usuario clara, y utilizar sólo una ventana. Todavía es necesario bloquear la interfaz del navegador en sí. Firefox expone el sistema de ficheros a través del archivo: // URI, la configuración a través del about: config URI, y finalmente expone el sistema de archivos a través de recursos://. Otros han experimentado problemas semejantes de bloqueo hacia abajo firefox. Afortunadamente, la solución a esto es un poco de javascript que utiliza expresiones regulares para bloquear tipos de URI que no son deseados [15]-[16].

Esto se debe colocar en el archivo browser.js que se puede extraer de archivo browser.jar de firefox.

```
if (location.match(/^file:/) ||
        location.match(/^\/) ||
        location.match(/^chrome:/) ||
        location.match(/^resource:/) ||
```

```
(!location.match(/^about:blank/) &&
location.match(/^about:/))) {
loadURI("about:blank");
 }
```

3. SGA – Sistema de Gestión Académica de la Universidad Nacional De Loja.

El proyecto Sistema de Gestión Académica, es un conjunto de soluciones de software orientadas a la gestión académica de la Universidad Nacional de Loja. Este sistema consta de una gran variedad de módulos para tareas cotidianas tomando en cuenta roles y escenarios. Ha sido construido utilizando tecnología web, usando el framework de desarrollo RAD Turbogears, y una variedad de herramientas Open Source.

3.1. Servicios del Sistema de Gestión Académico

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 [16]. Las categorías son:

3.1.1. Información Académica

En esta categoría, se exponen todos los métodos o servicios relacionados a la información académica, de estudiantes así como de docentes [16].

Listado de métodos de esta categoría:

TABLA I:

MÉTODOS QUE RETORNAN INFORMACIÓN ACADÉMICA

sgaws_periodos_lectivos:	Retorna una lista de períodos 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

3.1.2. Información Institucional

En esta categoría se dispone de todos los métodos o servicios relacionados con la información institucional, como son las Áreas, Carreras, Módulos y Paralelos.

Listado de métodos de esta categoría [16]:

TABLA II:

MÉTODOS QUE RETORNAN INFORMACIÓN DE LA INSTITUCIÓN

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 un 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

3.1.3. Información Personal

En esta categoría se exponen todos los métodos o servicios relacionados con información personal de docentes y estudiantes.

Listado de métodos de esta categoría [16]:

TABLA III:

MÉTODOS QUE RETORNAN INFORMACIÓN PERSONAL

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

3.1.4. Validación

En esta categoría se exponen todos los métodos o servicios relacionados con la validación de docentes y estudiantes.

Listado de métodos de esta categoría [16]:

TABLA IV.
MÉTODOS PARA VALIDACIÓN

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

3.1.5. Información Estadística

En esta categoría se detallan los métodos o servicios relacionados con información estadística, como número de estudiantes matriculados, número de estudiantes aprobados y número de estudiantes reprobados

Listado de métodos de esta categoría [16]:

TABLA V:
MÉTODOS QUE RETORNAN INFORMACIÓN ESTADÍSTICA

sgaws_nmatriculados_oferta:	Devuelve el número de estudiantes matriculados en una oferta académica
sgaws_nmatriculados_area:	Devuelve el número de estudiantes matriculados en una área 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 área 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 área 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

3.2. 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 [6].

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.

3.2.1. 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 [16].

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 [16].

4. Web 2.0

La Web 2.0 es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones web enfocadas al usuario final. La Web 2.0 es una actitud y no precisamente una tecnología.

La Web 2.0 es la transición que se ha dado de aplicaciones tradicionales hacia aplicaciones que funcionan a través de los web enfocados al usuario final. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio [7].

Es una etapa que ha definido nuevos proyectos en Internet y está preocupándose por brindar mejores soluciones para el usuario final. Muchos aseguran que hemos reinventado lo que era el Internet, otros hablan de burbujas e inversiones, pero la realidad es que la evolución natural del medio realmente ha propuesto cosas más interesantes como lo analizamos diariamente en las notas de Actualidad [7].

Y es que cuando el web inició, nos encontrábamos en un entorno estático, con páginas en HTML que sufrían pocas actualizaciones y no tenían interacción con el usuario.

Pero para entender de donde viene el término de Web 2.0 tenemos que remontarnos al momento en que Dale Dougherty de O'Reilly Media utilizó este término en una conferencia en la que compartió una lluvia de ideas junto a Craig Cline de MediaLive. En dicho evento se hablaba del renacimiento y evolución de la web.

4.1. Orígenes del Término

El término fue acuñado por Dale Dougherty de O'Reilly Media en una tormenta de ideas con Craig Cline de MediaLive para desarrollar ideas para una conferencia. Dougherty sugirió que la web estaba en un renacimiento, con reglas que cambiaban y modelos de negocio que evolucionaban. Dougherty puso ejemplos "DoubleClick era la Web 1.0; Google AdSense es la Web 2.0 [7].

En 2005, Tim O'Reilly definió el concepto de Web 2.0. El mapa meme mostrado resume el meme de Web 2.0, con algunos ejemplos de servicios.

En su conferencia, O'Reilly y Battelle resumieron los principios clave que creen que caracterizan a las aplicaciones web 2.0: la web como plataforma; datos como el "Intel Inside"; efectos de red conducidos por una "arquitectura de participación"; innovación y desarrolladores independientes; pequeños modelos de negocio capaces de redifundir servicios y contenidos; el perpetuo beta; software por encima de un solo aparato.

En general, cuando mencionamos el término Web 2.0 nos referimos a una serie de aplicaciones y páginas de Internet que utilizan la inteligencia colectiva para proporcionar servicios interactivos en red dando al usuario el control de sus datos [7].

Así, podemos entender por Web 2.0, como propuso Xavier Ribes en 2007, todas aquellas utilidades y servicios de Internet que se sustentan en una base de datos, la cual puede ser modificada por los usuarios del servicio, ya sea en su contenido (añadiendo, cambiando o borrando información o asociando datos a la información existente), bien en la forma de presentarlos o en contenido y forma simultáneamente. [7].

4.2. Tecnologías que dan vida a un proyecto Web 2.0

Las tecnologías que dan vida a un proyecto Web 2.0 son:

- Transformar software de escritorio hacia la plataforma del web.
- Respeto a los estándares como el XHTML.

- Separación de contenido del diseño con uso de hojas de estilo.
- Sindicación de contenidos.
- Ajax (javascript ascincrónico y XML).
- Uso de Flash, Flex o Lazlo.
- Uso de Ruby onRails para programar páginas dinámicas.
- Utilización de redes sociales al manejar usuarios y comunidades.
- Dar control total a los usuarios en el manejo de su información.
- Proveer APis o XML para que las aplicaciones puedan ser manipuladas por otros.
- Facilitar el posicionamiento con URL sencillos.

4.3. ¿En qué nos sirve la Web 2.0?

El uso de él término de Web 2.0 está de moda, dándole mucho peso a una tendencia que ha estado presente desde hace algún tiempo. En Internet las especulaciones han sido causantes de grandes burbujas tecnológicas y han hecho fracasar a muchos proyectos.

Además, nuestros proyectos tienen que renovarse y evolucionar. El Web 2.0 no es precisamente una tecnología, sino es la actitud con la que debemos trabajar para desarrollar en Internet. Tal vez allí está la reflexión más importante del Web 2.0 [7].

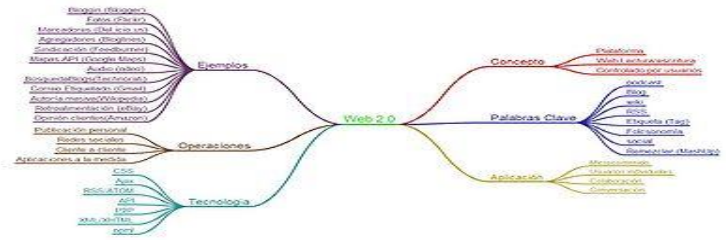

Figura 1. Tecnología Web 2.0

5. Toolkits de javascript

Existen herramientas que facilitan la creación de interfaces gráficas de usuario. Estas son conocidas como *toolkits*. Un toolkit consiste en una biblioteca de widgets que pueden ser llamados por los programas de aplicación. Típicamente los widgets usados

en los toolkits son objetos como barras de scroll, botones, cuadros de texto, etc. En la figura de abajo se muestran algunos widgets del toolkit [10].

Figura 2. Widgets del toolkit

Los toolkits facilitan enormemente la creación de Interfaz de Usuario (IUs); sin embargo, aun utilizando el toolkit, es necesario conocimientos de programación para la elaboración de la interfaz puesto que los widgets sólo tienen una interfaz procedural [10].

El uso de toolkits tiene grandes beneficios, como los que se mencionan a continuación:

- **Independencia de la Interfaz de Usuario**. *Separa la parte corr*espondiente al diseño de la Interfaz de Usuario de las complejidades de la programación. Esta separación le permite a los diseñadores el realizar prototipos rápida y fácilmente, hacer revisiones en minutos, y de esta manera agilizar los procedimientos de prueba. La programación necesaria para el sistema se empieza a escribir una vez que el diseño de la Interfaz de Usuario se ha estabilizado [10].

- **Código más confiable**. El código generado por los toolkits es frecuentemente más confiable que el código creado por un implementador humano, por lo que requiere una menor cantidad de *debugging*. Esto se debe a que el código generado automáticamente tiene un alto nivel de especificación [10].

- **Elaboración rápida de prototipos**. Con un toolkit las Interfaces de Usuario pueden ser construidas y modificadas rápidamente. Gran parte del trabajo de diseño y construcción de un programa se debe a la creación de la IU. Al utilizar un toolkit para su elaboración, el tiempo de fabricación se reduce de una manera considerable. El tiempo ganado puede ser utilizado para hacer una mayor cantidad de pruebas de evaluación

(probar, revisar, probar, revisar...), y tener al final un producto apropiado para el usuario final [10].

- **Interfaces de Usuario más fáciles de crear y de mantener.** Esto es porque las especificaciones de la interfaz pueden ser representadas, validadas y evaluadas más fácilmente. gracias al empleo de los widgets incluidos en el toolkit, la cantidad de código a escribir es mucho menor [10].

- **Facilidad para hacer modificaciones.** El empleo de un toolkit permite que la incorporación de cambios a la interfaz después de haber hecho las pruebas correspondientes sea más sencilla [10].

Un aspecto muy importante a considerar, y fundamental en este trabajo de tesis, es, como ya se ha mencionado, la utilización de herramientas que produzcan programas multiplataforma, lo que posibilita el que la interfaz pueda ser usada en distintos ambientes operativos como Windows o Unix, sin hacer modificación alguna a la Interfaz de Usuario.

Así pues, podemos concluir que el uso de un toolkit provee al diseñador de la IU de gran control y flexibilidad. Dentro de los toolkits de javascript tenemos:

5.1. DojoToolkits

Dojo es un framework que contiene APIs y widgets (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Contiene un sistema de empaquetado inteligente, los efectos de Interfaz de Usuario, drag and dropAPIs, widgetAPIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX [8].

Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas (bookmarking), y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Es conocido como "la navaja suiza del ejército de las bibliotecas Javascript". Proporciona una gama más amplia de opciones en una sola biblioteca JavaScript y es compatible con navegadores antiguos [8].

5.1.1. Complementos

Los complementos de Dojo son componentes pre empaquetados de código JavaScript, HTML y CSS que pueden ser usados para enriquecer aplicaciones web:

- Menús, pestañas y tooltips.
- Tablas ordenables, gráficos dinámicos y dibujado de vectores 2D.
- Efectos de animación y la posibilidad de crear animaciones personalizables.
- Soporte para arrastrar y soltar.
- Formularios y rutinas de validación para los parámetros.
- Calendario, selector de tiempo y reloj.
- Editor online de texto enriquecido.
- Núcleo de componentes (*dijit*) accesible desde versiones anteriores y lector de pantalla.

5.1.2. Comunicación Asíncrona

Una característica importante de las aplicaciones AJAX es la comunicación asíncrona entre el navegador y el servidor. Tradicionalmente, se realizaba con el comando JavaScript XMLHttpRequest. *Dojo* provee de una capa de abstracción (*dojo.io.bind*) para varios navegadores web con la que se pueden usar otros transportes (como *IFrames* ocultos) y diferentes formatos de datos. De esta forma podemos obtener los campos que se van a enviar como parámetros del formulario de una manera sencilla [8].

5.1.3. Sistema de paquetes

Dojo provee de un sistema de paquetes para facilitar el desarrollo modular. El script de inicio inicializa una serie de jerarquías de paquetes de espacios de nombre (*io*, *event*, etc.) bajo el paquete raíz *dojo*. Después de la inicialización del paquete *dojo*, cualquier otro paquete puede ser cargado (vía XMLHttpRequest o cualquier otro transporte similar) usando las utilidades ofrecidas en el arranque. También es posible inicializar paquetes adicionales dentro o al mismo nivel que el paquete *Dojo*, permitiendo extensiones o librerías de terceros [8].

Los paquetes de *Dojo* pueden contener múltiples archivos. Cualquier paquete o archivo puede depender de otro. En este caso, cuando el paquete es cargado, cualquier dependencia será también cargada [8].

Dojo también brinda una manera de crear perfiles; el sistema ofrece una lista de paquetes y usa Apache Ant para crear un archivo JavaScript comprimido que contiene dichos paquetes y dependencias. De esta manera se tiene todo el código necesario para ser cargado y es inicializado de una sola vez, permitiendo así el cacheado (la mayoría de los navegadores web no permiten el cacheado de archivos vía XMLHttpRequest) [8].

5.1.4. Almacenamiento en el cliente de datos

Adicionalmente, ofrece funciones para leer y escribir cookies, proporcionando en el lado cliente una abstracción llamada *Dojo Storage*. *Dojo Storage* permite a la aplicación web almacenar datos en el lado cliente, persistencia y seguridad. Cuando se incluye en una página web, determina cual es el mejor método para almacenar la información. Cuando la aplicación web ha sido cargada desde el sistema de archivos (por ejemplo desde file://URL), *Dojo Storage* usa de manera transparente XPCOM en Firefox y ActiveX en Internet Explorer para mantener la persistencia de la información. El desarrollador que use *Dojo Storage* no se tiene que preocupar de esto, ya que *Dojo* tiene una capa de abstracción con métodos *put()* y *get()* [8].

5.1.5. Almacenamiento en el servidor

Desde enero de 2007, *Dojo* incluye las siguientes implementaciones de almacenamiento de datos en el paquete *dojo.data*:

- CsvStore: almacenamiento de sólo lectura y acceso CSV.
- OpmlStore: almacenamiento de sólo lectura y lectura jerárquica desde archivos en formato OPML.
- YahooStore: almacenamiento de sólo lectura que obtiene los resultado del servicio web del buscador de Yahoo Search!.
- DeliciousStore: almacenamiento de sólo lectura que obtiene los marcadores del servicio web que ofrece Del.icio.us.
- RdfStore: almacenamiento de solo lectura que usa SPARQL para comunicarse con el servidor de datos RDF [8].

5.1.6. Soporte para Adobe Integrated Runtime (AIR)

Dojo permite usar aplicaciones Adobe AIR basadas en JavaScript. Ha sido modificada para satisfacer los requisitos de seguridad de Adobe.

5.1.7. Desarrollo con Dojo

El desarrollo con *Dojo* puede hacerse bien con los IDE *Aptana* o *Komodo*. Existe una versión para la comunidad de *Aptana* y una versión de 21 días de prueba de *Komodo*. Los plugins de *Dojo* están disponibles para ambos (actualmente *Dojo 0.4.3* y *0.9.0*). Para el desarrollo WYSIWYG usando Dojo, se puede utilizar la aplicación de código abierto Wave Maker Visual Ajax Studio [8].

5.2. Script.aculo.us

Otra de las librerías que han surgido para facilitar el desarrollo de aplicaciones JavaScript y que están basadas en Prototype. Su autor original de la librería es **Thomas**

Fuchs, y se distribuye de forma completamente gratuita y dispone de una buena documentación que se puede encontrar en la siguiente dirección: http://wiki.script.aculo.us/scriptaculous/

La librería está dividida en varios módulos:

- Efectos: permite añadir de forma muy sencilla efectos especiales a cualquier elemento de la página. La librería incluye una serie de efectos básicos y otros efectos complejos construidos con la combinación de esos efectos básicos. Entre los efectos prediseñados se encuentran el parpadeo, movimiento rápido, aparecer/desaparecer, aumentar/disminuir de tamaño, desplegarse, etc.[10].

- Controles: define varios *controles* que se pueden añadir directamente a cualquier aplicación web. Los tres controles que forman este módulo son: "arrastrar y soltar", que permite definir los elementos que se pueden arrastrar y las zonas en las que se pueden soltar elementos; "autocompletar", que permite definir un cuadro de texto en el que los valores que se escriben se autocompletan con ayuda del servidor; editor de contenidos, que permite modificar los contenidos de cualquier página web añadiendo un sencillo editor AJAX en cada elemento [10].

- Utilidades: la utilidad principal que incluye se llama builder, que se utiliza para crear fácilmente nodos y fragmentos complejos de DOM [10].

A continuación se muestra un ejemplo del uso de uno de sus componentes más populares.

```
window.onload = function() {
  // Crear elemento de tipo <div> para mostrar las sugerencias del servidor
  var elDiv = Builder.node('div', {id:'sugerencias'});
  document.body.appendChild(elDiv);

  new Ajax.Autocompleter('municipio', 'sugerencias',
    'http://localhost/autocompletaMunicipios.php?modo=ul',
    {paramName: 'municipio'}
  );
}
```

La sintaxis del control Ajax.Autocompleter() es la siguiente:
new Ajax.Autocompleter(idCuadroTexto, idDivResultados, url, opciones);

El primer parámetro (idCuadroTexto) es el identificador del cuadro de texto en el que el usuario escribe las letras que se van a autocompletar. El segundo parámetro (idDivResultados) indica el identificador del elemento en el que se va a mostrar la respuesta del servidor. En el ejemplo anterior, este se crea dinámicamente cuando se carga la página. El tercer parámetro (url) indica la URL del script de servidor que recibe las letras escritas por el usuario y devuelve la lista de sugerencias que se muestran. El último parámetro (opciones) permite modificar algunas de las opciones por defecto de este control [10].

6. Herramientas para el desarrollo

6.1. Ajax

Es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones [9]-[11].

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XML Http Request, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML [9].

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dados que está basado en estándares abiertos como JavaScript y DocumentObjectModel (DOM).

AJAX es una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XML Http Request para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.
Los navegadores que soportan las aplicaciones AJAX dependerán de las características que el navegador permita [9].

- Navegadores basados en Gecko como Mozilla, Mozilla Firefox, SeaMonkey, Camino, K-Meleon, IceWeasel, Flock, Epiphany, Galeon y Netscape versión 7.1 y superiores.
- Google Chrome
- Microsoft Internet Explorer para Windows versión 5.0 y superiores, y los navegadores basados en él.
- Navegadores con el API KHTML versión 3.2 y superiores implementado, incluyendo Konqueror versión 3.2 y superiores, Apple Safari versión 1.2 y superiores, y el Web Browser for S60 de Nokia tercera generación y posteriores.
- Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores.

6.2. Javascript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario [10].

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios [10].

6.2.1. Definir JavaScript en un archivo externo

Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos XHTML enlazan mediante la etiqueta <script>. Se pueden crear todos los archivos JavaScript que sean necesarios y cada documento XHTML puede enlazar tantos archivos JavaScript como necesite [10].

Ejemplo: Archivo codigo.js

```
alert("Un mensaje de prueba");
```

Documento XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/
  xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Ejemplo de código JavaScript en el propio documento</title>
         <script type="text/javascript" src="/js/codigo.js"></script>
        </head>
        <body>
        <p>Un párrafo de texto.</p>
```

```
        </body>
</html>
```

Además del atributo type, este método requiere definir el atributo src, que es el que indica la URL correspondiente al archivo JavaScript que se quiere enlazar. Cada etiqueta <script> solamente puede enlazar un único archivo, pero en una misma página se pueden incluir tantas etiquetas <script> como sean necesarias.

Los archivos de tipo JavaScript son documentos normales de texto con la extensión .js, que se pueden crear con cualquier editor de texto como Notepad, Wordpad, EmEditor, UltraEdit, Vi, etc. La principal ventaja de enlazar un archivo JavaScript externo es que se simplifica el código XHTML de la página, que se puede reutilizar el mismo código JavaScript en todas las páginas del sitio web y que cualquier modificación realizada en el archivo JavaScript se ve reflejada inmediatamente en todas las páginas XHTML que lo enlazan [9].

6.3. Librería jquery

jQuery es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio [11].

6.3.1. Características

Entre las principales características de Jquery tenemos:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+ [11].

jQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Para ello utiliza las funciones $() o jQuery() [11].

Función $()

La forma de interactuar con la página es mediante la función $(), un alias de jQuery(), que recibe como parámetro una expresión CSS o el nombre de una etiqueta HTML y devuelve todos los nodos (elementos) que concuerden con la expresión. Esta expresión es denominada **selector** en la terminología de jQuery

```
$("#tablaAlumnos"); // Devolverá el elemento con id="tablaAlumnos"
$(".activo");       // Devolverá una matriz de elementos con class="activo"
```

Una vez obtenidos los nodos, se les puede aplicar cualquiera de las funciones que facilita la biblioteca.

// Se elimina el estilo (con removeClass()) y se aplica uno nuevo (con addClass()) a todos los nodos con class="activo"

```
$(".activo").removeClass("activo").addClass("inactivo");
```

O por ejemplo, efectos gráficos:

// Anima todos los componentes con class="activo"

```
$(".activo").slideToggle("slow");
```

6.3.2. Inicio de jQuery

Comúnmente antes de realizar cualquier acción en el documento con jQuery(), debemos percatarnos de que el documento esté listo. Para ello usamos $(document).ready(); de esta forma:

```
$(document).ready(function() {
  //Aquí van todas las acciones del documento.
});
```

7. Apache 2.0

Apache es un servidor web HTTP de código abierto para plataformas Unix-like (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual [12]-[22].

En sus inicios se basaba en el código de NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Actualmente es el servidor web más usado en todo el mundo**,** superando en 2009 los 100 millones de sitios web, el 70% del total. Está desarrollado y mantenido por una comunidad de usuarios en torno a la Apache Software Foundation [12].

Otro de los detalles importantes en esta nueva versión es el soporte para el sistema operativo BeOS, que se une a la saga de Unix, con GNU/Linux a la cabeza, y a la plataforma Microsoft Windows [12].

La versión 2.0 de Apache es más rápida y más estable en sistemas que no son tipo Unix, tales como BeOS, OS/2 y Windows, que la versión antigua. Con la introducción de módulos de multiprocesamiento (MPMs) específicos para cada plataforma y del Apache Portable Runtime (APR), estas plataformas tienen ahora implementada su propia API nativa, evitando las capas de emulación POSIX que provocan problemas y un bajo rendimiento, la ordenación de módulos se hace ahora por hooks, lo que ofrece una mayor flexibilidad [12]-[22].

También se han añadido nuevas llamadas que ofrecen capacidades adicionales sin tener que parchear el núcleo del servidor Apache.

Los módulos de Apache pueden ahora escribirse para que se comporten como filtros que actúan sobre el flujo de contenidos tal y como salen del servidor o tal y como son recibidos por el servidor. Esto permite, por ejemplo, que el resultado de un script CGI sea analizado por las directivas Server Side Include usando el filtro INCLUDES del módulo mod_include. El módulo mod_ext_filter permite que programas externos actúen como filtros casi del mismo modo que los CGIs pueden actuar como handlers [22].

7.1 Características

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características destacan: [12]

- Multiplataforma
- Es un servidor de web conforme al protocolo HTTP/1.1
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hebras en la versión 2.0
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

8. Latex

Latex es un procesador de texto, de gran potencialidad en el manejo de fórmulas matemáticas, cuadros y tablas. LaTeX está formado por un gran conjunto de macros de TeX, con la intención de facilitar el uso del lenguaje de composición tipográfica, TEX, creado por Donald Knuth.

Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados con LaTeX es comparable a la de una editorial científica de primera línea. LaTeX es software libre bajo licencia LPPL [13].

Los archivos de LaTeX tienen la terminación *.tex* es decir un documento escrito en Latex debe estar contenido en un archivo cuya extensión sea *.tex*

Con LaTeX, la elaboración del documento requiere normalmente de dos etapas: en la primera hay que crear mediante cualquier editor de texto llano un *fichero fuente* que, con las órdenes y comandos adecuados, contenga el texto que queramos imprimir [13].

La segunda consiste en procesar este fichero; el *procesador de textos* interpreta las órdenes escritas en él y *compila* el documento, dejándolo preparado para que pueda ser enviado a la salida correspondiente, ya sea la pantalla o la impresora. Ahora bien, si se quiere añadir o cambiar algo en el documento, se deberá hacer los cambios en el fichero fuente y procesarlo de nuevo.

Esta idea, que puede parecer poco práctica *a priori*, es conocida a los que están familiarizados con el proceso de compilación que se realiza con los lenguajes de programación de alto nivel (C, C++, etc.), ya que es completamente análogo.

Figura 3. Compilación de documentos en Látex

Una de las ventajas de LaTeX es que la salida que ofrece es siempre la misma, con independencia del dispositivo (impresora, pantalla, etc.) o el sistema operativo (MS Windows, MacOS, Unix, GNU/Linux, etc.) y puede ser exportado a partir de una misma fuente a numerosos formatos tales como Postscript, PDF, SGML, HTML, RTF, etc. El modo en que LaTeX interpreta la «forma» que debe tener el documento es mediante *etiquetas* [13].

8.1 Caracteres reservados de LaTeX

Para evitar un tipo común de errores en Latex, es necesario conocer cuales caracteres son usados como comandos o delimitadores por el compilador de Latex, estos caracteres son:

```
# $ % & _ { } ~ ^ \
```

Para poder incluir estos caracteres dentro de tu documento, como parte del texto debes anteponerle un *backslash*, por ejemplo

Para obtener # , se escribe \#

Para obtener $, se escribe \$

Para obtener % , se escribe \%

9. Sistema integrado, empotrado o embebido

Un sistema integrado, empotrado o embebido es un sistema informático de uso específico construido dentro de un dispositivo mayor. Los sistemas integrados se utilizan para usos muy diferentes a los usos generales a los que se suelen someter a las computadoras personales. En un sistema integrado la mayoría de los componentes se encuentran incluidos en la placa base [21].

Dos de las diferencias principales son el precio y el consumo. Puesto que los sistemas integrados se pueden fabricar por decenas de millares o por millones de unidades, una de las principales preocupaciones es reducir los costes. Los sistemas integrados suelen usar un procesador relativamente pequeño y una memoria pequeña para reducir los costes. Se enfrentan, sobre todo, al problema de que un fallo en un elemento implica la necesidad de reparar la placa íntegra [21].

Lentitud no significa que vayan a la velocidad del reloj. En general, se suele simplificar toda la arquitectura del ordenador o computadora para reducir los costes. Por ejemplo, los sistemas integrados emplean a menudo periféricos controlados por interfaces síncronos en serie, que son de diez a cientos de veces más lentos que los periféricos de un ordenador o computadora personal normal. Los primeros equipos integrados que

se desarrollaron fueron elaborados por IBM en los años 1980. Los programas de sistemas integrados se enfrentan normalmente a problemas de tiempo real.

9.1. Componentes de un Sistema Integrado

En la parte central se encuentra el microprocesador, micro controlador, DSP, etc. Es decir, la CPU o unidad que aporta capacidad de cómputo al sistema, pudiendo incluir memoria interna o externa, un micro con arquitectura específica según requisitos [21].

La comunicación adquiere gran importancia en los sistemas integrados. Lo normal es que el sistema pueda comunicarse mediante interfaces estándar de cable o inalámbricas. Así un Sistema Integrado normalmente incorporará puertos de comunicaciones del tipo RS-232, RS-485, SPI, I^2C, CAN, USB, IP, Wi-Fi, GSM, GPRS, DSRC, etc. [21].

El subsistema de presentación tipo suele ser una pantalla gráfica, táctil, LCD, alfanumérico, etc.

Denominamos actuadores a los posibles elementos electrónicos que el sistema se encarga de controlar. Puede ser un motor eléctrico, un conmutador tipo relé etc. El más habitual puede ser una salida de señal PWM para control de la velocidad en motores de corriente continua.

El módulo de E/S analógicas y digitales suele emplearse para digitalizar señales analógicas procedentes de sensores, activar diodos LED, reconocer el estado abierto cerrado de un conmutador o pulsador, etc.

El módulo de reloj es el encargado de generar las diferentes señales de reloj a partir de un único oscilador principal. El tipo de oscilador es importante por varios aspectos: por la frecuencia necesaria, por la estabilidad necesaria y por el consumo de corriente requerido. El oscilador con mejores características en cuanto a estabilidad y coste son los basados en resonador de cristal de cuarzo, mientras que los que requieren menor consumo son los RC. Mediante sistemas PLL se obtienen otras frecuencias con la misma estabilidad que el oscilador patrón [21].

El módulo de energía (power) se encarga de generar las diferentes tensiones y corrientes necesarias para alimentar los diferentes circuitos del SE. Usualmente se trabaja con un rango de posibles tensiones de entrada que mediante conversores ac/dc o dc/dc se obtienen las diferentes tensiones necesarias para alimentar los diversos componentes activos del circuito [21].

Además de los conversores AC/DC y DC/DC, otros módulos típicos, filtros, circuitos integrados supervisores de alimentación, etc. El consumo de energía puede ser determinante en el desarrollo de algunos sistemas integrados que necesariamente se alimentan con baterías, con lo que el tiempo de uso del SE suele ser la duración de la carga de las baterías.

9.2. Aplicaciones de un Sistema Integrado

Los lugares donde se pueden encontrar los sistemas integrados son numerosos y de varias naturalezas. A continuación se exponen varios ejemplos para ilustrar las posibilidades de los mismos [21]:

- En una fábrica, para controlar un proceso de montaje o producción. Una máquina que se encargue de una determinada tarea hoy en día contiene numerosos circuitos electrónicos y eléctricos para el control de motores, hornos, etc. que deben ser gobernados por un procesador, el cual ofrece un interfaz persona – máquina para ser dirigido por un operario e informarle al mismo de la marcha del proceso.

- Puntos de servicio o venta (POS, Point Of Service). Las cajas donde se paga la compra en un supermercado son cada vez más completas, integrando teclados numéricos, lectores de códigos de barras mediante láser, lectores de tarjetas bancarias de banda magnética o chip, pantalla alfanumérica de cristal líquido, etc. El sistema empotrado en este caso requiere numerosos conectores de entrada y salida y unas características robustas para la operación continuada.

- Puntos de información al ciudadano. En oficinas de turismo, grandes almacenes, bibliotecas, etc. existen equipos con una pantalla táctil donde se puede pulsar sobre la

misma y elegir la consulta a realizar, obteniendo una respuesta personalizada en un entorno gráfico amigable.

- Decodificadores y set-top boxes para la recepción de televisión. Cada vez existe un mayor número de operadores de televisión que aprovechando las tecnologías vía satélite y de red de cable ofrecen un servicio de televisión de pago diferenciado del convencional. En primer lugar envían la señal en formato digital MPEG-2 con lo que es necesario un procesado para decodificarla y mandarla al televisor. Además viaja cifrada para evitar que la reciban en claro usuarios sin contrato, lo que requiere descifrarla en casa del abonado. También ofrecen un servicio de televisión interactiva o web-TV que necesita de un software específico para mostrar páginas web y con ello un sistema basado en procesador con salida de señal de televisión.

- Sistemas radar de aviones. El procesado de la señal recibida o reflejada del sistema radar embarcado en un avión requiere alta potencia de cálculo además de ocupar poco espacio, pesar poco y soportar condiciones extremas de funcionamiento (temperatura, presión atmosférica, vibraciones, etc.).

- Equipos de medicina en hospitales y ambulancias UVI – móvil.

- Máquinas de revelado automático de fotos.

- Cajeros automáticos.

- Pasarelas (Gateway) Internet-LAN.

Y un sin fin de posibilidades aún por descubrir o en estado embrionario como son las neveras inteligentes que controlen su suministro vía Internet, PC de bolsillo, etc.

10. Circuito Impreso o PCB

Un circuito impreso o PCB en inglés, es una tarjeta o placa utilizada para realizar el emplazamiento de los distintos elementos que conforman el circuito y las interconexiones eléctricas entre ellos.

Antiguamente era habitual la fabricación de circuitos impresos para el diseño de sistemas mediante técnicas caseras, sin embargo esta práctica ha ido disminuyendo con el tiempo. En los últimos años el tamaño de las componentes electrónicas se ha reducido en forma considerable, lo que implica menor separación entre pines para circuitos integrados de alta densidad. Teniendo también en consideración las actuales frecuencias de operación de los dispositivos, es necesaria una muy buena precisión en el proceso de impresión de la placa con la finalidad de garantizar tolerancias mínimas [19].

Los circuitos impresos más sencillos corresponden a los que contienen caminos de cobre (tracks) solamente por una de las superficies de la placa. A estas placas se les conoce como circuitos impresos de una capa, o en inglés, 1 Layer PCB [19].

Los circuitos impresos más comunes de hoy en día son los de 2 capas o 2 Layer PCB. Sin embargo, dependiendo de la complejidad del diseño del físico del circuito (o PCB layout), pueden llegar a fabricarse hasta de 8 o más layers [19].

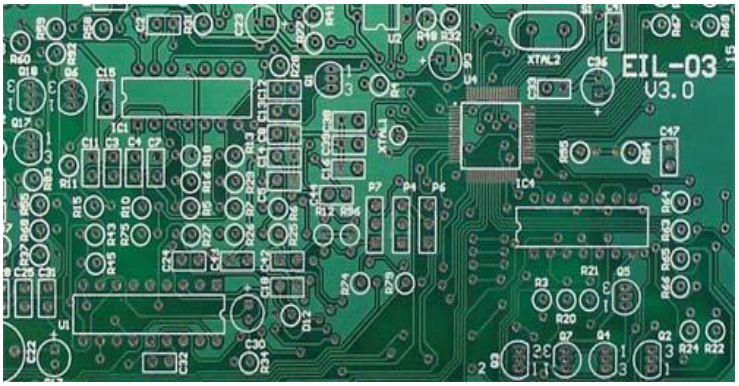

Figura 4. PCB de 2 o más layers

10.1. Soldermask o máscara de soldado

Para montar los componentes electrónicos en los circuitos impresos se requiere de un proceso de ensamblado, que puede ser manual o mediante maquinaria especializada.

Los procesos de ensamblado requieren la utilización de soldadura para poder fijar los componentes a la placa. Para evitar que la soldadura pueda cortocircuitar accidentalmente dos tracks pertenecientes a nodos distintos se utiliza una máscara de

soldado, o soldermask en inglés. Esta máscara de soldado es un barniz que se aplica a los circuitos impresos en la etapa de fabricación y puede ser de variados colores. El color que se utiliza más frecuente es el verde seguidos del rojo y azul [19].

En los programas CAD EDA, existe generalmente una regla asociada a la expansión de máscara de soldado. Esta regla especifica la distancia que existe entre el borde de los pads y el borde de la máscara de soldado. Este concepto se puede apreciar en la figura 5 (a).

10.2. Serigrafía o silkscreen

La serigrafía es el proceso en donde se imprime sobre la máscara de soldado información conducente a facilitar la labor del ensamblado y de posterior verificación. Generalmente se imprime para indicar puntos de prueba como también la posición, orientación y referencia de las componentes que conforman el circuito. También puede utilizarse para cualquier propósito que el diseñador requiera, como por ejemplo para el nombre del producto, compañía, instrucciones de configuración, etc. La serigrafía puede ir en ambas capas externas o caras del circuito impreso. En inglés se conoce como silkscreen u overlay. En la figura 5 se puede apreciar la serigrafía, que corresponde a todo lo impreso en color blanco [19].

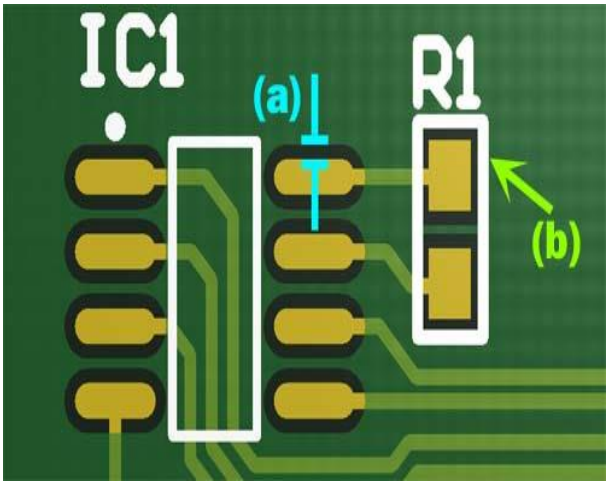

Figura 5. Expansión de la máscara de soldado (a) y serigrafía (b)

10.3. Stackup o pila de capas

Tal como se dijo en al principio de este artículo, los circuitos impresos o PCBs pueden fabricarse de varias layers o capas. Cuando una PCB es diseñada mediante un software CAD EDA, se suelen especificar varias capas en el diseño que no necesariamente son conductoras, como por ejemplo la serigrafía, máscara de soldado, etc. Esto puede prestarse para confusión ya que generalmente uno se refiere a layers o capas solamente cuando se está hablando de las que son conductoras. De ahora en adelante cuando nos refiramos al concepto de capa o layer sin el sufijo CAD, nos estamos refiriendo solamente a las capas que son conductoras [19].

El orden de la pila de capas CAD es la siguiente:

TABLA VI:
ORDEN DE PILAS DE CAPAS CAD

CAD Layer	Descripción Capa CAD
1	Top silkscreen/overlay (serigrafía superior)
2	Top soldermask (máscara de soldado superior)
3	Top paste mask
4	Layer 1 (Capa 1)
5	Sustrato
6	Layer 2 (Capa 2)
...	...
n-1	Sustrato
N	Layer n (Capa n) (conductora)
n+1	Bottom paste mask
n+2	Bottom solder mask (máscara de soldado inferior)
n+3	Bottom silkscreen/overlay (serigrafía inferior)

En la figura 6 podemos apreciar 3 stackups o pilas de capas. Lo que se destaca de color naranja corresponden a las layers de cada uno de los stackups. La altura de los stackup (espesor del circuitos impresos) puede variar dependiendo de la aplicación, sin embargo la más utilizada es 1.6 [mm].

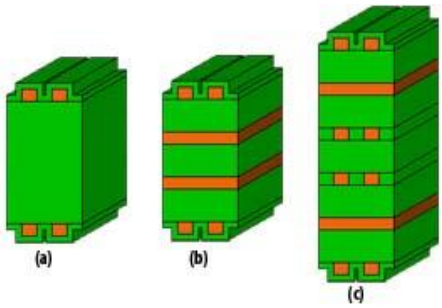

Figura 6. Ejemplos de Stackups de 2 capas (a), 4 capas (b) y 6 capas (c)

10.4. Encapsulado de componentes

En el mercado se encuentran diversos tipos de encapsulados de componentes electrónicos. Es común encontrar varios tipos de encapsulado para un mismo dispositivo. Existen básicamente 3 grandes familias de encapsulados [19]:

Básicamente existen 3 grandes familias de encapsulados electrónicos:

TABLA VII:
ENCAPSULADOS ELECTRÓNICOS

PAQUETE	DESCRIPCION	IMÁGENES EJEMPLO
Thru-Hole (A través de orificio)	Son todos aquellos componentes que poseen pines para ser instalados en perforaciones metalizadas (llamadas thru-hole pads). Este tipo de componentes se suelda por la capa opuesta. Generalmente son montados por un solo lado de la placa.	

SMD/SMT (montaje superficial)	Son todos aquellos componentes que se montan en forma superficial. Tienen la ventaja de que pueden montarse por ambos lados además de ser más pequeños que los thru-hole, lo que permite hacer circuitos más pequeños y densos. Son interesantes para diseños en alta frecuencia debido a su tamaño reducido.	
BGA (Arreglo de bolas en grilla)	Este tipo de encapsulado es utilizado para chips que contienen una cantidad elevada de pines (i.e 300 a 1000). Se requiere de maquinaria muy especializada para su instalación ya que los pines son bolas de soldadura que deben ser fundidas para conectarse con los pads, por lo que la alineación es fundamental. Son ideales para circuitos integrados de alta frecuencia ya que la inductancia parásita entre el "ball" y el "pad" es mínima. Son muy comunes en placas madres de computadores, tarjetas de video, etc.	

10.5. Pads

Un pad es una superficie de cobre en un circuito impreso o PCB que permite soldar o fijar la componente a la placa. Existen dos tipos de pads; los thru-hole y los smd (montaje de superficie) [19].

Los pads thru-hole están pensados para introducir el pin de la componente para luego soldarla por el lado opuesto al cual se introdujo. Este tipo de pads es muy similar a una via thru-hole [19].

Los pads smd están pensados para montaje superficial, es decir, soldar la componente por el mismo lado de la placa en donde se emplazó [19].

A continuación se muestran en la figura 7, cuatro componentes. La componente IC1 y R1 tienen 8 y 2 pads SMD respectivamente, mientras que ambas componentes Q1 y PW tienen 3 pads thru-hole.

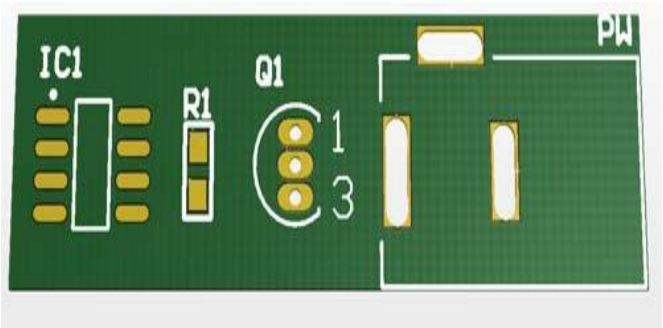

Figura 7. Pads SMD y Thru-hole

10.6. Caminos de cobres (pistas o tracks)

Un track es un camino conductor de cobre que sirve para conectar un pad (donde descansa el pin o terminal de un componente) a otro. Los tracks pueden ser de distinto ancho dependiendo de las corrientes que fluyen a través de ellos.

Cabe destacar, que en altas frecuencias es necesario calcular el ancho del track de forma que exista una adaptación de impedancias durante todo su recorrido.[19]

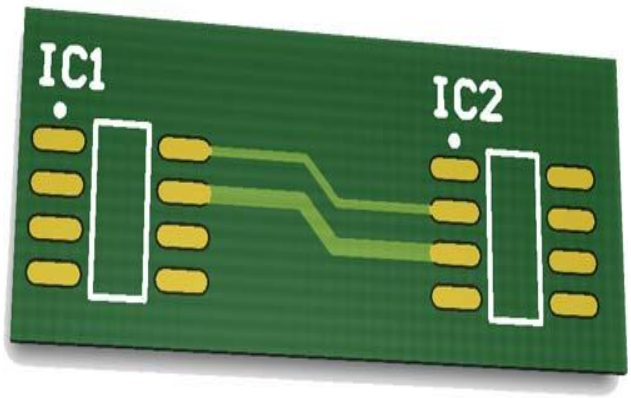

Figura 8. Tracks que interconectan 2 circuitos impresos (chips)

10.7. Perforaciones Metalizadas a través de orificio (Thru-hole Vias o Full Stack Vías)

Cuando se debe realizar una conexión de un componente que se encuentra en la capa superior de la PCB con otro de la capa inferior, se utiliza una vía. Una vía es una perforación metalizada (en inglés, plated vía) que permite que la conducción eléctrica no se interrumpa cuando se pasa de una superficie a otra. En la figura 9 puede apreciarse como salen 2 tracks desde los pads de un chip que se encuentra en la capa superior de la PCB, que luego de pasar por 2 vías, se conectan a los pads del chip que se encuentra en la capa inferior [19].

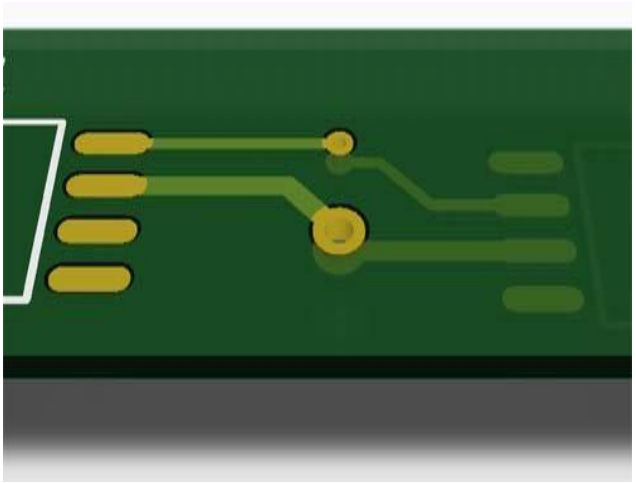

Figura 9. Dos chips en caras opuestas se conectan, atravesando la placa con vías

En la TABLA VIII, se muestra una vista más detallada de una sección de circuito impreso. Los colores significan**:**

TABLA VIII:
SIGNIFICADO DE COLORES DE UN CIRUITO IMPRESO

COLOR	LEYENDA
Verde	Máscaras de soldado de las capas superior (top) e inferior (bottom)
Rojo	corresponde a la capa superior (top)
Violeta	corresponde a la segunda layer que en este caso es un plano de energía (i.e. Vcc o Gnd)

Amarillo	corresponde a la tercera layer que en este caso es un plano de energía (i.e. Vcc o Gnd)
Azul	corresponde a la capa inferior (bottom)

Este circuito impreso PCB es de 4 capas, y se puede apreciar como un track de la capa superior (top) atraviesa la placa donde sale un track en la capa inferior (bottom).

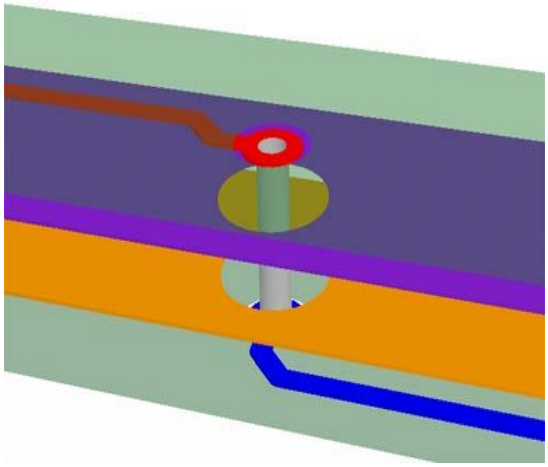

Figura 10. Track de la capa superior atravesando la PCB

10.8. Perforaciones Metalizadas ciegas (Blind Vías)

En diseños complejos de alta densidad es necesario utilizar más de 2 layers como hemos mostrado en la figura 7. Generalmente en los diseños de sistemas multicapas donde hay muchos integrados, se utilizan planos de energía (de Vcc's o de Gnd) para evitar tener que routear muchos tracks de alimentación. Dicho en otras palabras es más fácil y seguro conectarse a la alimentación directamente bajo del chip (capa contigua a la superior) que con tracks largos hacia el PDS (Power Delivery System) [20].

También hay veces que se debe routear un track de señal desde una capa externa a una interna con el mínimo largo posible debido a problemas que se generan cuando se trabaja en frecuencias altas que afectan la integridad de las señales (más de esto en otra publicación futura). Para esos tipos de conexionado se utilizan vías ciegas, las cuales permiten conectar una capa externa a una capa interna. La vía ciega comienza

en una capa externa y termina en un capa interna, es por eso que se llama ciega. Para darse cuenta si una vía es ciega, puede ponerse el circuito impreso contra una fuente de luz y ver si la luz pasa a través de ella. En el caso de que la luz no traspase la PCB estamos hablando de una vía ciega, en caso contrario, de una vía thru-hole [19]-[20].

Es muy útil utilizar este tipo de vías en diseño de circuitos impresos cuando no se posee de mucho espacio para emplazar componentes, por lo que se debe poblar la placa por componentes en ambas caras. Si las vías fuesen del tipo thru-hole utilizaría espacio extra el hecho de que una vía atravesara la placa completa.

En la figura 11, podemos apreciar 3 vías en este circuito impreso. Si vemos de izquierda a derecha, la primera vía que veremos es una thru-hole o fullstack. La segunda es una vía que nace en la capa superior y termina en la segunda capa, por lo que decimos que es una 1-2 blind vía. Por último la tercera es una vía que nace en la capa inferior y termina en la tercera, por lo tanto es una 3-4 blind vía [19].

Es importante destacar que las blind vías se suelen manufacturar en layers consecutivas, en otras palabras L1 L2, L3 L4, Ln-1 Ln.

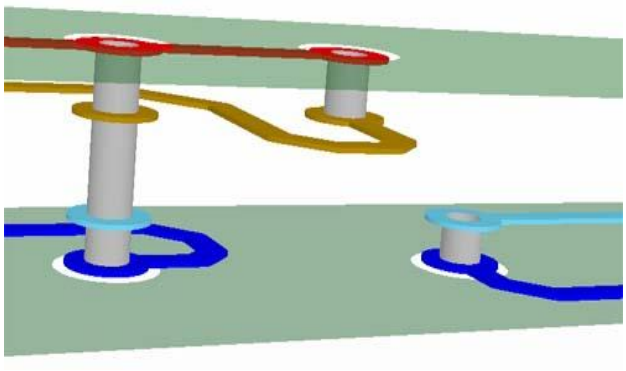

Figura 11. Comparación de una Thru-hole vía con una Blind vía

La desventaja que tiene el utilizar este tipo de vías es su alto costo en comparación a la alternativa de vías thruhole [20].

10.9. Perforaciones Metalizadas enterradas u ocultas (Buried Vías)

Estas vías son similares a las ciegas, la diferencia es que no nacen ni terminan en ninguna capa externa o de superficie. Si vemos de izquierda a derecha, la primera es una thru-hole o fullstack y la segunda es una 1-2 blind, al igual que en el ejemplo anterior. La tercera en este caso es una 2-3 buried vía que nace en la segunda capa y termina en la tercera [20].

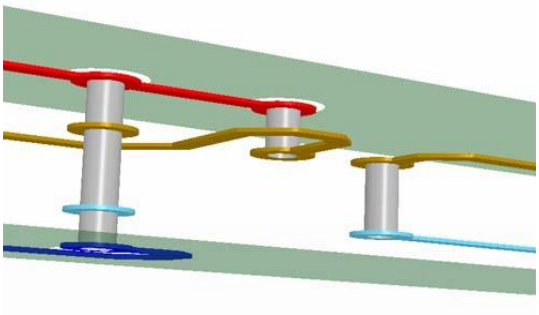

Figura 12. Comparación de una Thru-hole vía, Blind vía y Burried vía

Es importante destacar que las buried vías se suelen manufacturar en layers consecutivas (i.e. L1 L2)

La desventaja que tiene el utilizar este tipo de vías es su alto costo en comparación a la alternativa de vías thruhole.

10.10. Construcción de una PCB (Fotolitografía)

Para la construcción de un circuito PCB es necesario disponer del circuito a realizar impreso en papel de transparencia o en papel cebolla o albanene. Se recomienda utilizar papel cebolla y utilizar una fotocopiadora láser, puesto que esta máquina resalta mucho más el negro que una fotocopiadora convencional. La construcción del PCB se desarrolla en cinco fases bien diferenciadas [20].

10.10.1. Insolación

Para esta primera fase en la construcción de la PCB se han utilizado los siguientes materiales:

- Placa positiva de fibra de vidrio con cobre.
- Lámpara de rayos ultravioleta.
- Impresión en Negativo del circuito.

Primeramente se debe asegurar que la placa de cobre esté libre de grasa y perfectamente seca. La placa de fibra de vidrio debe tener una resina fotosensible en su superficie. Si la luz ultravioleta incide sobre ella, la resina reacciona químicamente y se endurecerá. Este es el principio en que se basa la insolación: se somete a la placa a una emisión constante de luz ultravioleta. Una vez que se ha untado la placa de cobre con la resina, se coloca el negativo del circuito [20]

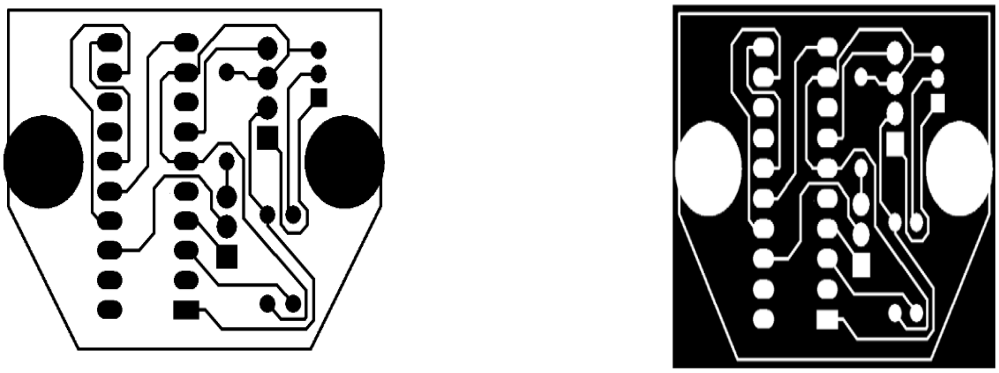

Figura 13. Diseño de circuito en positivo y en negativo

sobre la placa cuidando de que la luz no incida sobre ella de tal manera que la resina reaccione sólo en las zonas donde no existe tinta ya que la luz pasa a través del papel transparente. En el caso de circuitos de doble cara, la mayor dificultad de este proceso estriba en que ambas caras deben coincidir perfectamente para que a la hora de hacer os agujeros de los componentes no haya ningún problema. En la siguiente figura se observa la emisión UV (figura 14) [20].

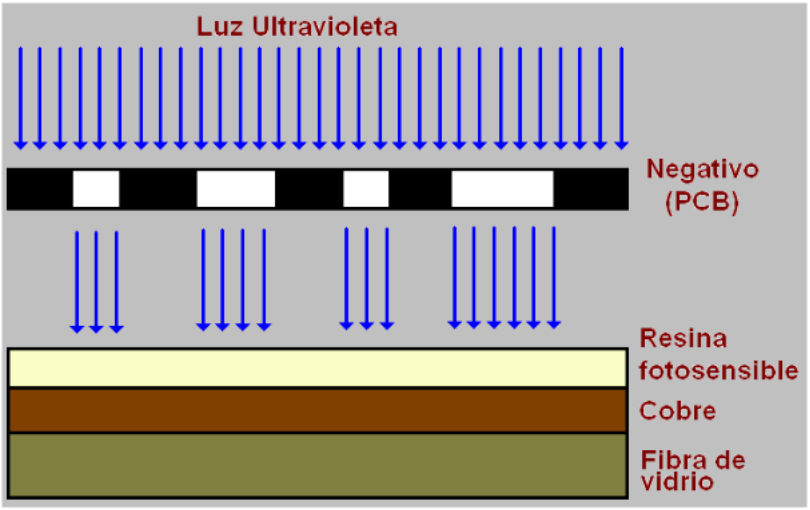

Figura 14. Emisión UV

La reacción que provoca en la resina se muestra en la siguiente imagen (figura 15).

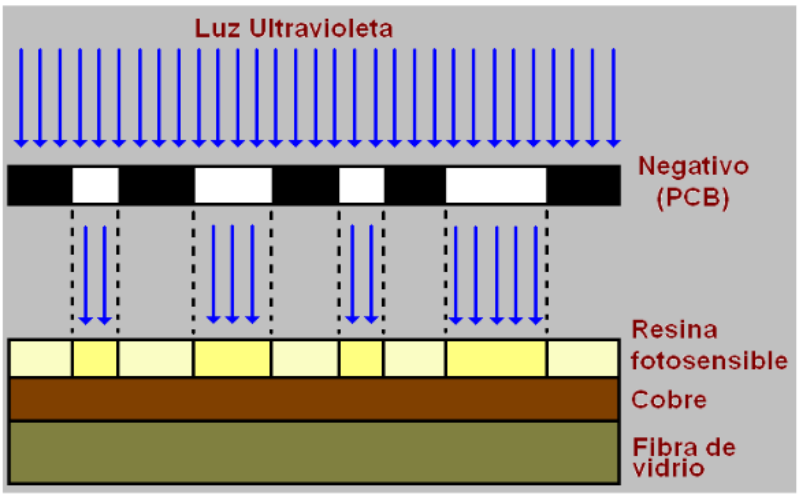

Figura 15. Reacción provocada en la resina

El resultado tras la insolación se observa en la siguiente imagen (Figura 16).

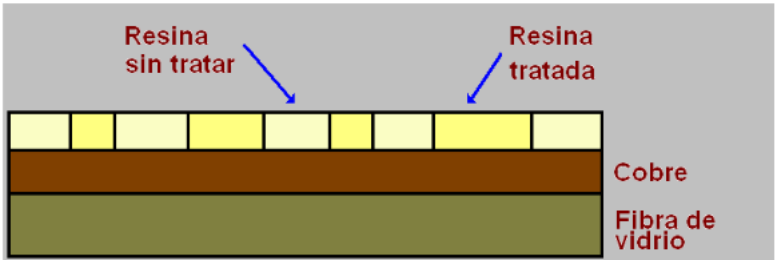

Figura 16. Reacción frente a la insolación

10.10.2. Revelado

El segundo paso en la construcción de la placa, es el revelado. Consiste en eliminar de la placa la resina tratada químicamente. Para ello, la debemos poner en contacto la placa con una sustancia química llamada revelador como se observa en la figura siguiente (figura 17) [20].

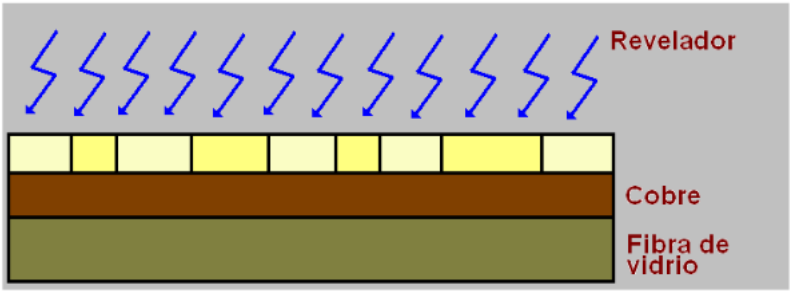

Figura 17. Contacto de la placa con revelador

El revelador elimina de la placa de fibra de vidrio la resina tratada químicamente y deja al descubierto el cobre únicamente en aquellas zonas donde no incidió la luz ultravioleta (donde no debe existir pista conductora). En la figura siguiente se presenta el resultado de este proceso (figura 18) [20].

Figura 18. Placa sin pistas conductoras

Por lo tanto, los materiales utilizados para revelar la placa ya insolada han sido los siguientes:

- Placa insolada.
- Líquido de revelado para placa con cobre.
- Equipo de revelado y aclarado.

Al sacar la placa de la insoladora, hay que tener cuidado para que la luz no incida sobre ella. Se introduce la placa en un recipiente de revelado y empieza el proceso explicado

anteriormente. Transcurridos unos 15 segundos más o menos, se pasa la placa al recipiente de agua para aclararla antes de comenzar la siguiente fase. [20]

10.10.3. Atacado

Lo que se ha obtenido hasta ahora es una placa donde las zonas en las que se desea que haya pista (blanco) poseen el cobre protegido por la resina fotosensible. En aquellas regiones donde queremos que no halla pista (negro) el cobre está directamente expuesto al exterior [20].

El objetivo del atacado es eliminar todo el cobre expuesto - no protegido por la resina. El proceso se resuelve químicamente. Se utiliza una solución de Cloruro férrico calentada a unos 40 ºC. Dicha solución, en contacto con el cobre reacciona con él de tal forma que éste se desprende depositándose en el fondo de la cubeta o recipiente que se utilice [19]-[20].

Al terminar la fase de revelado, se pasa la placa al recipiente que contiene la solución de Cloruro Férrico. Se mantiene en la cubeta durante veinte minutos aproximadamente o hasta que se vea que el proceso ha terminado. Por último, se introduce la placa en la última cubeta de agua para aclararla. Se extrae y se seca la placa, y ya la tenemos preparada para hacer los agujeros y soldar los componentes.

Una vez que la placa se ha secado del todo y antes de pasar a la siguiente fase de la construcción, es necesario recortar los bordes sobrantes para disminuir las medidas. Se puede utilizar una guillotina. En la siguiente imagen (figura 19) se muestra el efecto del atacado en la placa [19].

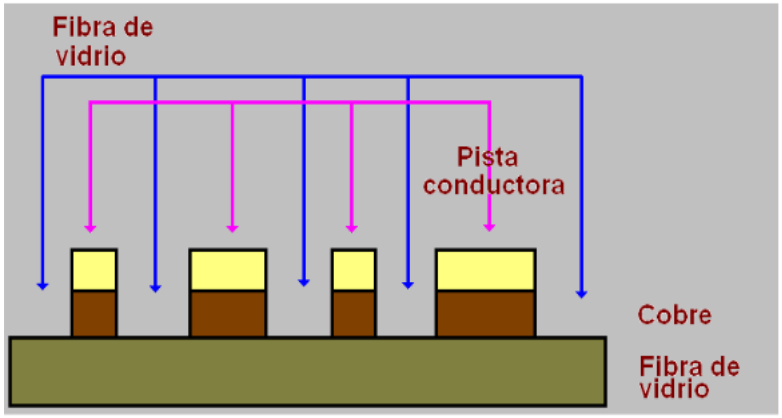

Figura 19. Recorte de bordes sobrantes

10.10.4. Taladrado de la Placa

Cuando se ha terminado por completo la fase de atacado, ha llegado el momento de dar los últimos retoques a nuestra placa. Teniendo todas las pistas ya trazadas, es necesario hacer los agujeros para que podamos introducir los componentes en los lugares indicados. Para esta fase, se utiliza un taladro de broca fina [20].

En esta fase, es importante realizar los agujeros con mucho cuidado para que todos los componentes hagan contacto con las pistas. Además, se ha de tener en cuenta el grosor de las terminales de los distintos componentes a situar para lo cual se habrá de utilizar diferentes brocas [20].

También merece especial atención el mango (taladro de banco) que posee este taladro para su uso. Tan sólo hay que situar la broca sobre el agujero que se desea realizar y bajar el taladro con este mango. Si el taladro no poseyera esta estructura es mucho más complicado realizar los agujeros y se corre un gran riesgo de que la punta de la broca patine y se dañe alguna pista cercana al agujero que se va a taladrar [20].

10.10.5. Soldadura

Cuando todos los agujeros se han terminado, se puede comenzar a unir los componentes a la placa con la ayuda de un cautín y un poco de soldadura de estaño. En el caso de que la placa sea de dos caras, se ha de tener que soldar la mayoría de los componentes por los dos lados para que hagan contacto, con la dificultad que eso conlleva. Como se muestra en la siguiente figura [20]:

Figura 20. Soldadura con cautín y estaño

Cada vez que se suelda un nuevo componente, es posible que haya material sobrante en la cara inferior de la placa, por lo que hay que cortarlo con unos alicates tal y como se muestra en la Figura 21.

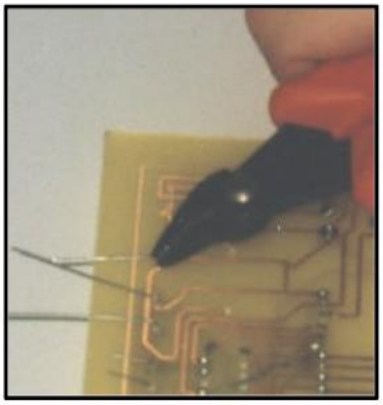

Figura 21. Material sobrante recortado con alicates

11. UML

11.1. Clases y Objetos

A continuación se describen los principales conceptos de uml.

Clases.

Una Clase, desde una perspectiva conceptual, representa un conjunto de Objetos, el mismo que comparte las mismas propiedades (Atributos), el mismo comportamiento (Métodos), las mismas relaciones con otros Objetos (Mensajes) y la misma semántica dentro del sistema [18].

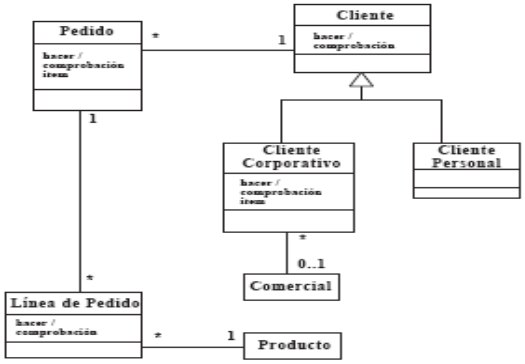

Figura 22. Modelo de Diagrama de Clase

Una clase puede representarse de forma esquemática (plegada), con los detalles como atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase [18].

Mientras que desde una perspectiva física, una Clase es una pieza de software que actúa como un molde para fabricar tipos particulares de objetos que disponen de los mismos atributos y métodos [18].

Objetos.

Un objeto se representa de la misma forma que una clase, por lo que representa una entidad del mundo real o inventado [18].

Los Objetos se estructuran por:

- La Identidad ¿Quién soy? = Atributos
- Propósito ¿Cuál es mi misión? = Justificación
- Responsabilidades ¿Qué debo hacer? = Métodos
- Procedencia ¿De qué Clase provengo? = Origen
- Relaciones ¿Qué mensajes entiendo? = Comportamiento

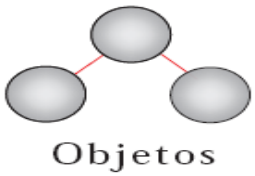

Figura 23. Objetos

Los objetos que se han creado a partir de una Clase concreta, se llaman instancias de esta Clase y se diferencian entre ellos únicamente por los valores de sus atributos.

11.2. Diagramas de Clases

Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Los diagramas de Clases son estáticos, esto es, representan que partes interactúan entre sí, es decir muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc.) [18].

Las clases se documentan con una descripción de lo que hacen, sus métodos y sus atributos. Las relaciones entre clases se documentan con una descripción de su propósito, su cardinalidad (cuantos objetos intervienen en la relación) y su opcionalidad (cuando un objeto es opcional el que intervenga en una relación). La descripción de clases complejas se puede documentar con diagramas de estados 18].

11.3. Diagramas de Objetos

Un diagrama de objetos representa un conjunto de objetos y sus relaciones en un momento concreto, estos son utilizados durante el proceso de Análisis y Diseño de los sistemas informáticos en la metodología UML.

Se utilizan para describir estructuras de datos, instantáneas de las instancias de los elementos encontrados en un diagrama de clases. Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase. Los diagramas de objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase. Una diferencia con los diagramas de clase es que el compartimiento de arriba va en la forma, Nombre de objeto: Nombre de clase [18].

Los diagramas de objetos modelan las instancias de elementos contenidos en los diagramas de clases. Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento concreto. En UML, los diagramas de clase se utilizan para visualizar los aspectos estáticos del sistema y los diagramas de interacción se utilizan para ver los aspectos dinámicos de los sistemas, y constan de instancias de los elementos del diagrama de clases y mensajes enviados entre ellos [18].

En un punto intermedio podemos situar los diagramas de objetos, que contiene un conjunto de instancias de los elementos encontrados en el diagrama de clases, representando sólo la parte estática de un interacción, consistiendo en los objetos que colaborar pero sin ninguno de los mensajes intercambiados entre ellos [18].

11.4. Modelo de Casos de Usos

Los Casos de Uso son requerimientos funcionales que describen de una manera detallada el comportamiento del sistema con los distintos Actores que interactúan con él independientes de la implementación [18].

11.4.1. Diagrama de Casos de Uso

El Diagrama de Casos de Usos, es la representación de la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Es aquí donde se muestra la relación entre los actores y los casos de uso del sistema [18].

Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

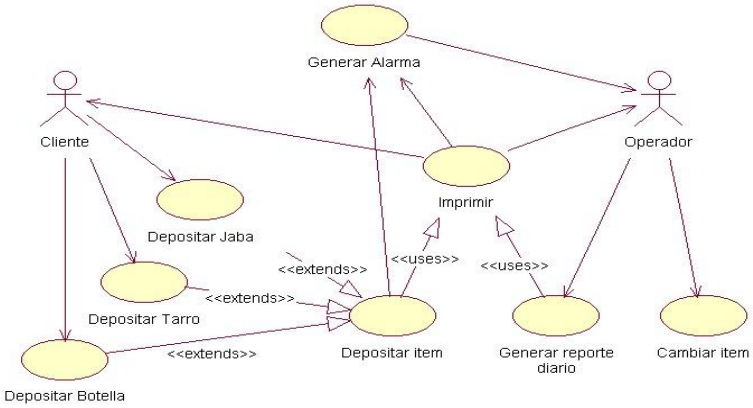

Figura 24. Modelo de Diagrama de Caso de Uso

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: Actores, casos de uso y relaciones entre casos de uso [18].

Actores

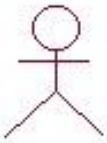

Figura 25. Representación de actor

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes. Un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema [18].

Casos de Uso

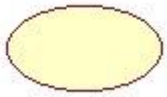

Figura 26. Representación de Caso de Uso

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica, es decir, es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso [18].

Relaciones entre Casos de Uso.

Asociación.-Es el tipo de relación básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso) [18].

Figura 27. Representación de una Asociación

Dependencia o Instanciación.- Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia [18].

Figura 28. Representación de una Dependencia

Generalización.- Cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>) [18].

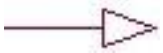

Figura 29. Representación de una Generalización

Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

Extends*:* Se utilizar cuando un caso de uso es similar a otro (características).

Uses: Se utiliza cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica. [18]

11.4.2. Descripción de Casos de Uso

Es la descripción detallada del Caso de Uso; es decir es la descripción detallada de la funcionalidad del sistema en lo que se refiere a su interacción externa [18].

La descripción de los casos de uso comprende:

- **El inicio:** ¿Cuándo y qué actor lo produce?
- **El fin:** ¿Cuándo se produce y qué valor devuelve?
- **La interacción actor-caso de uso:** ¿Qué mensajes intercambian ambos?
- **Objetivo del caso de uso:** ¿Qué lleva a cabo o intenta?
- **Cronología y origen de las interacciones**
- **Repeticiones de comportamiento:** ¿Qué operaciones son iteradas?
- **Situaciones opcionales:** ¿Qué ejecuciones alternativas se presentan en el caso de uso?

11.5. Diagramas de Paquetes

Un paquete es el elemento de organización básica de un modelo de sistema UML. Los diagramas de Paquetes se usan para reflejar la organización de paquetes y sus elementos [18].

Los usos más comunes de los diagramas de paquete son para organizar diagramas de casos de uso y diagramas de clases, estos paquetes son como grandes contenedores

de clases, es decir muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones.

Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. Un paquete puede contener paquetes subordinados, diagramas o elementos únicos, y se puede establecer su visibilidad así como la de los elementos que contiene.

Los elementos contenidos en un paquete comparten el mismo espacio de nombres, esto significa que los elementos contenidos en un mismo espacio de nombres específico deben tener nombres únicos.

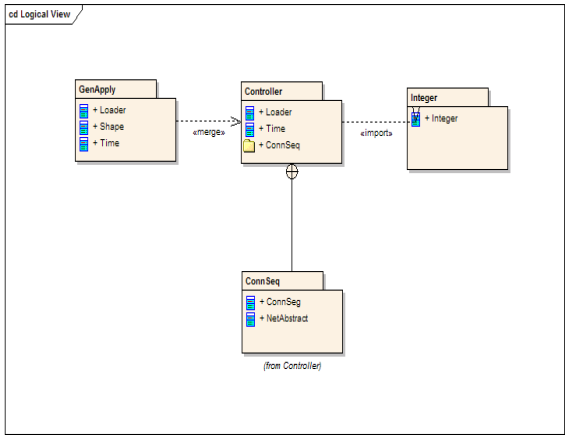

Figura 30. Modelo de Diagrama de Paquetes

Los Paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Con estas líneas maestras sobre la mesa, los paquetes son buenos elementos de gestión. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido [18].

A medida que aumenten sus conocimientos acerca de las características del problema, el modelo conceptual se hará más complicado y podrá alcanzar un tamaño difícil de controlar. Puede utilizar paquetes para dividir el modelo en subconjuntos más pequeños y manejables.

Los paquetes se pueden construir para representar relaciones tanto físicas como lógicas. Cuando se elige incluir las clases a los paquetes específicos, es útil asignar las

clases con la misma jerarquía de herencia a los paquetes, las clases que están relacionadas a través de la composición y las clases que colaboran que también tienen un fuerte argumento para ser incluidas en el mismo paquete.

Los paquetes se representan en UML como carpetas y contienen los elementos que comparten un espacio de nombre; todos los elementos dentro de un paquete deben tener un identificador único. El paquete debe mostrar el nombre del paquete y puede opcionalmente mostrar los elementos dentro del paquete en compartimientos extras.

Combinación de Paquetes: Cuando un conector «merge» se usa en un paquete, la fuente de la combinación importa los contenidos importados y anidados del destino. Si existe un elemento dentro del origen y el destino, las definiciones del elemento origen se expandirán para incluir las definiciones del elemento contenidas en el destino. Todos los elementos agregados o actualizados por una combinación se notan por una relación de generalización desde el origen hasta el destino [18].

11.6. Diagramas de Interacción

En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración [18].

11.7. Diagrama de Secuencia.

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo [18].

E. MATERIALES Y MÉTODOS

1. Métodos

Para el desarrollo del presente proyecto de investigación se emplea la metodología ICONIX, debido a su facilidad de uso en desarrollo de software ya que considera la participación de los usuarios y la documentación de todo el proceso.

En la fase de recolección de requerimientos se utiliza la investigación de campo, para ello se recurre al lugar donde se encuentra el objeto de investigación es decir a la dirección de Informática y telecomunicaciones de la Universidad para obtener toda la información relacionada con los archivos log almacenados en el servidor web de la Universidad Nacional de Loja, así como a la AEIRNNR a las secretarias y estudiantes quienes despachan y solicitan respectivamente los certificados académicos.

En las secretarías se aplican entrevistas a las Secretarias con el fin de determinar los certificados más solicitados así como el proceso para el despacho de los mismos. Y a los estudiantes se aplican encuestas para conocer su valoración y acción en la adquisición de los certificados.

Luego se establecen los requerimientos funcionales, no funcionales, el glosario de términos y el modelo del dominio de la aplicación, aplicando los métodos inductivo y deductivo.

Seguidamente y en base a los requerimientos se diseñan los diagramas de caso de uso para el usuario y el administrador en base a los cuales se redactaron los mismos indicando el flujo normal y alterno de los eventos entre el sistema y los actores.

Además se realizan los prototipos de pantallas, diagramas de clases y diagramas de secuencia; todos estos elementos sirven para comprobar los requerimientos de los casos de uso.

Para la codificación del diseño de la aplicación se utiliza NetBeans 6.5 con el Framework **JSF** el cual está basado en el patrón MVC (Modelo-Vista-Controlador) que sirve para

crear aplicaciones basadas en la web.

2. Materiales

En la construcción del kiosko se usa:

2.1 Componentes hardware

Para la construcción del kiosko electrónico se ha usado los siguientes dispositivos:

- Computador i686, Max speed 300Mhz, Current speed 566Mhz
- Memoria SIMM 256 MB
- Puertos USB, para uso de Impresora
- Puerto Serial, para conexión del teclado
- Disco Duro 40Gb Disco Duro.
- Impresora HP Deskjet 1000 Printer series - J110
- Teclado tipo keypad, con reemplazo de membranas táctiles por botones tipo push.
- Placa de circuitos para encendido de equipos, impresora, monitor y mainboard.
- Montaje de madera tropan y hierro
- Monitor LG 14``
- Supresor de picos
- Switch D-link 8 puertos

2.2. Software

Para la construcción del sistema embebido se usa:

- Sistema Operativo Debian GNU/Linux
- Lenguaje de programación: Javascript
- Entorno de programación: Netbeans 6.5
- Servidor web: Apache 2
- Herramienta de modelado físico, lógico y dinámico: Enterprise 8
- Librería para construir conexiones con el webservice: jQuery
- Lenguaje de programación PHP5.

2.3. Materiales de Oficina

- Resmas de papel bond
- Cartuchos de tinta
- Plumas
- Carpetas
- Servicio de internet

F. RESULTADOS

Para obtener resultados se debe enfocar la problemática existente para plantear la solución.

En la Universidad Nacional de Loja se posee una potente herramienta donde reposa la información académica de los estudiantes que forman parte de la misma y brinda servicios informáticos académicos a toda hora sin importar el día, a pesar de esto la emisión de certificados académicos no se la realiza de forma ininterrumpida ocasionando que los estudiantes al momento de solicitar estos documentos se rijan al horario administrativo de los usuarios del sistema puesto que cada una de las Áreas Académicas Administrativas tiene un horario establecido para la emisión de certificados dentro del Sistema.

A esto se suma que los estudiantes para realizar estos tipos de trámites deben obligadamente solicitar con un oficio físico dicho documento académico de manera personal en las secretarías de cada una de las carreras.

Actualmente para presentar, entregar, mostrar o difundir la información se la realiza utilizando los reportes impresos, publicaciones, medio audio- visuales, y en el Internet. Es por ello que dentro de las universidades se están desarrollando nuevas alternativas para que la información fluya de manera más rápida y el trámite de entrega de documento se lo lleve de manera más eficiente.

Es por ello que el desarrollo de nuevas tecnologías informáticas orientadas en la automatización de certificados académicos, ayudarán a eliminar los requisitos o trámites innecesarios con el objetivo de mejorar el servicio que se presta a los estudiantes que soliciten dichos documentos académicos legales.

Enmarcados en éste contexto es que proponemos como tema de proyecto de tesis “Diseño y Construcción de un Kiosko Electrónico Prototipado para la Impresión Automática de Certificados Académicos para los Estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables”, con el cual pretendemos crear

una herramienta informática que permita mejorar los servicios de emisión de certificados académicos capaz de satisfacer los requerimientos de los estudiantes.

En el presente trabajo de titulación está desarrollado bajo la filosofía analista vs usuario, es decir de acuerdo a los requerimientos del usuario. Para ello se utilizarán los métodos: Iconix, deductivo, inductivo así como técnicas como entrevista, cuestionario y observación cuyos resultados fueron los siguientes:

1. Antecedentes

Para conocer la forma actual en que se lleva el proceso de emisión de certificados se procede a realizar encuestas a los estudiantes de la AEIRNNR de la Universidad Nacional de Loja, obteniéndose los siguientes resultados, mismos que han sido tabulados metódicamente con el fin de obtener conclusiones específicas.

1. ¿Qué tipo de certificado ha solicitado Ud.?

De acuerdo a los resultados del cuestionario aplicado se concluye que la mayoría de estudiantes necesitan y solicitan en algún momento algún certificado de cualquier índole, siendo los más relevantes los certificados asistencias, notas, aprobación y matrícula, por lo que el sistema está diseñado para generar dichos certificados que son de adquisición continúa por parte de los estudiantes.

2. ¿Cuáles son los requisitos para obtener un certificado?

En base a la tabulación del cuestionario se puede denotar que para la adquisición de cualquier certificado que solicite el estudiantado es necesario contar bien con la aprobación del director de área y la adquisición de un derecho, por lo conlleva a que la entrega de su certificado demore.

3. ¿Qué procedimientos se siguen para obtener un certificado?

Al realizar la valoración del cuestionario encontramos que varios son los estudiantes que consideran que para la adquisición de certificados son demasiados los trámites que

se deben llevar a cabo, por lo que aconsejan que se entreguen de forma inmediata y sus trámites sean más breves.

4. ¿En qué tiempo le entregan el certificado?

Los estudiantes manifiestan que los certificados no se entregan de forma inmediata y si hay retraso para su realización y entrega, por lo que a ellos les gustaría que el trámite para su adquisición sea ágil e inmediato.

5. ¿Cómo califica Usted la emisión de certificados en las secretarías?

Algunos de los estudiantes opinan que la atención es buena al momento de emitir los certificados, otros que es rápida, existiendo una igualdad si sumamos regular con malo por lo que podemos determinar que una gran población no están conformes con la atención brindada al momento de solicitar certificados.

6. ¿En qué horario puede Ud. solicitar un certificado?

Al realizar el análisis nos encontramos que los estudiantes deben realizar el proceso de obtención de certificados en horario laborable, por lo que obliga al estudiante depender de ellos, aun cuando no deban asistir a sus respectivas clases.

7. ¿Qué opina del uso de los cajeros automáticos?

Realizando el respectivo análisis podemos visualizar la afirmación del 58% de los estudiantes de la Universidad testifican que el uso del cajero automático es cómodo y fácil reduciendo así el tiempo y el hacer colas en bancos, además que su uso es fácil y agradable para el usuario.

8. ¿Cree usted que un kiosco electrónico es necesario y facilitaría la adquisición de certificados?

El 80% de los estudiantes de la población consideran que con la creación de un kiosco electrónico la emisión de certificados será más rápida y eficiente, en donde el alumno

se encargará personalmente de su propio proceso de adquisición, además se evitarán realizar algunos trámites, y se apoyarán en una herramienta tecnológica logrando cambios significativos.

G. DISCUSIÓN

1. DESARROLLO DE LA PROPUESTA ALTERNATIVA

El presente trabajo de titulación se basa en los 4 objetivos específicos planteados, los cuales permitieron dar cumplimiento con el objetivo general, a continuación se detallan los objetivos de la presente investigación:

1.1. Objetivo General

- Aportar en el desarrollo de sistemas automatizados para la gestión académica de la Universidad Nacional de Loja, mediante la generación del software y las especificaciones de hardware de un Kiosko Electrónico Prototipado, que ofrezca a los estudiantes un servicio de generación e impresión de certificados académicos de forma automática y basada en tecnologías Web 2.0.

Este objetivo se logró gracias al análisis minucioso de nuevas tecnologías de información y comunicación que reduzcan al usuario en este caso al estudiante, la tediosa concurrencia hacia una secretaría de área en particular para realizar trámites sencillos; como la obtención de certificados académicos. A más de lo mencionado se construyó la estructura del kiosko para el montaje de los equipos electrónicos y la aplicación que interactúa con los usuarios a través de una pantalla y un teclado tipo push.

1.2. Objetivos Específicos

- Disponer de la información sobre las demandas de certificaciones de los estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.

Para la obtención de la demanda se realizó a cabalidad el análisis preliminar dentro de la ingeniería de requerimientos, y análisis metodológico (encuestas) el mismo que

permitió establecer los requerimientos funcionales y no funcionales de la herramienta informática.

- Seleccionar el toolkits javascript de generación de tecnologías Web 2.0, que mejor corresponda a las necesidades y alternativas técnicas más convenientes.

Este objetivo se cumplió haciendo una investigación de toolkits para JavaScript con las que se logre grandes resultados en menos tiempo y espacio. Es así que finalmente se concluyó con la elección de JQuery ya que ofrece una serie de funcionalidades que simplifican código y hacen más fácil la tarea al momento de la codificación obteniendo los resultados deseados en la emisión de certificados académicos para los estudiantes del AEIRNNR.

- Establecer las especificaciones de Hardware y Sistema Operativo para la operación dedicada del Kiosko.

Para cumplir con este objetivo cabe mencionar que se especificó lo siguiente: un computador de prestaciones básicas en el que se instaló Debian GNU/Linux, para la impresión de los certificados igualmente se montó al kiosko una impresora y para la interacción con el usuario se utilizó un teclado tipo push para controlar las opciones presentadas en pantalla tras el despliegue de la aplicación desarrollada en JavaScript y utilizando el enfoque sistemático de la ingeniería de software donde se realizaron los procesos arquitectónicos como los diagramas de casos de uso, de robustez y de secuencia.

- Contar con un Software Gráfico derivado del toolkit javascript elegido, embebido como proceso dedicado del Sistema Operativo del Kiosko y que interactué con el Sistema de Gestión Académica de la Universidad Nacional de Loja.

La parte concluyente de este proyecto es el producto fundamental al momento se sacar adelante el proyecto fue establecer las especificaciones de hardware y software empleados para el montaje del kiosko.

1.3. Glosario de términos

- **Aplicación web:** Es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet.

- **Script:** cada uno de los programas, aplicaciones o trozos de código creados con el lenguaje de programación JavaScript. Unas pocas líneas de código forman un script y un archivo de miles de líneas de JavaScript también se considera un script. A veces se traduce al español directamente como "guion", aunque script es una palabra más adecuada y comúnmente aceptada.

- **Sentencia:** cada una de las instrucciones que forman un script.

- **Palabras reservadas:** son las palabras (en inglés) que se utilizan para construir las sentencias de JavaScript y que por tanto no pueden ser utilizadas libremente

- **Sintaxis:** La sintaxis de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.

- **HTML** (Hyper Text MarkupLanguage): Es un "lenguaje de descripción de página" independiente de la plataforma en que se utilice. Define la estructura y contenido de los documentos insertando en el texto de los mismos una serie de marcas (tags o etiquetas) que delimitan elementos del documento como cabeceras, párrafos, etc., y se usan para dar un tratamiento diferente al texto que se encuentre entre ellas.

- **MVC:** Patrón de arquitectura Modelo-Vista-Controlador.

- **UML** Unified Modeling Language. Lenguaje estandarizado para especificar, visualizar, construir y documentar los artefactos de un sistema software.

- **XML** Extensible Markup Language. Es un metalenguaje extensible de etiquetas. Permite definir la gramática de lenguajes específicos (de la misma manera que HTML). Por lo tanto XML no es realmente un lenguaje en particular sino una manera de definir lenguajes para diferentes necesidades.

1.4. Documento de requerimientos

Previo al diseño del presente proyecto se pueden definir los siguientes requerimientos

1.4.1. Requerimientos Funcionales

El sistema permitirá:

TABLA VIII:
REQUERIMIENTOS FUNCIONALES

CÓDIGO	DESCRIPCIÓN	CATEGORÍA
RF 001	Ingreso de la cédula y contraseña de un estudiante para su autenticación.	Evidente
RF 002	Autenticar un estudiante mediante una petición de validación al web Service del SGA.	Oculto
RF 003	Obtener los datos de un certificado académico: asistencia, notas o matrícula.	Evidente
RF 004	Imprimir un certificado académico.	Evidente
RF 005	Obtener datos del estudiante desde el SGA.	Evidente
RF 006	Autenticación al sistema como usuario administrador	Evidente
RF 007	Acceso al panel del kiosko de forma remota con el usuario administrador.	Evidente
RF 008	Obtener reportes de uso del sistema por parte del usuario administrador del kiosko.	Evidente
RF 009	Administración de los parámetros principales del kiosko.	Evidente

1.4.2. Requerimientos No Funcionales

TABLA IX:
REQUERIMIENTOS NO FUNCIONALES

CÓDIGO	DESCRIPCIÓN
RNF 001	El sistema se podrá ejecutar bajo plataformas Unix y Windows 9x o superior.
RNF 002	La aplicación será accedida en un entorno web, como también en un entorno remoto para su administración.
RNF 003	La aplicación fue desarrollada en JavaScript utilizando NetBeans con el toolkitJQuery y basada en el patrón de arquitectura Modelo-Vista-Controlador (MVC).
RNF 004	La aplicación se desplegará en el Servidor Web Apache 2 que permite la presentación de JavaScript y la generación de documentos usando Php.
RNF 005	La aplicación empleará la arquitectura cliente-servidor, con interfaz web, utilizando el protocolo HTTP, XML y SOAP para el intercambio de datos con el web servicie del SGA.
RNF 006	La aplicación emplea interfaces gráficas simples y amigables con efectos visuales, de modo que usuarios puedan realizar los procesos fácilmente.
RNF 007	El acceso a la administración de la aplicación estará restringido por el uso de claves asignadas a los usuarios.
RNF 008	La aplicación responderá al acceso de los usuarios y a sus procesos dependiendo de la velocidad de conexión.
RNF 009	La aplicación se conectará al Web Service del Sistema de Gestión Académica de la UNL a través de una capa de acceso a datos simple donde se tendrá un singleton para la configuración del mismo.
RNF 010	La aplicación estará completamente documentada tanto en el código fuente como en los manuales del programador y del usuario.
RNF 011	Ergonómicamente accesible con comodidad y facilidad.

1.5. MODELO DE LA APLICACIÓN

Para el diseño de la aplicación se ha realizado el modelo estático y el modelo dinámico.

1.5.1. MODELO ESTÁTICO

1.5.1.1. Modelo del dominio

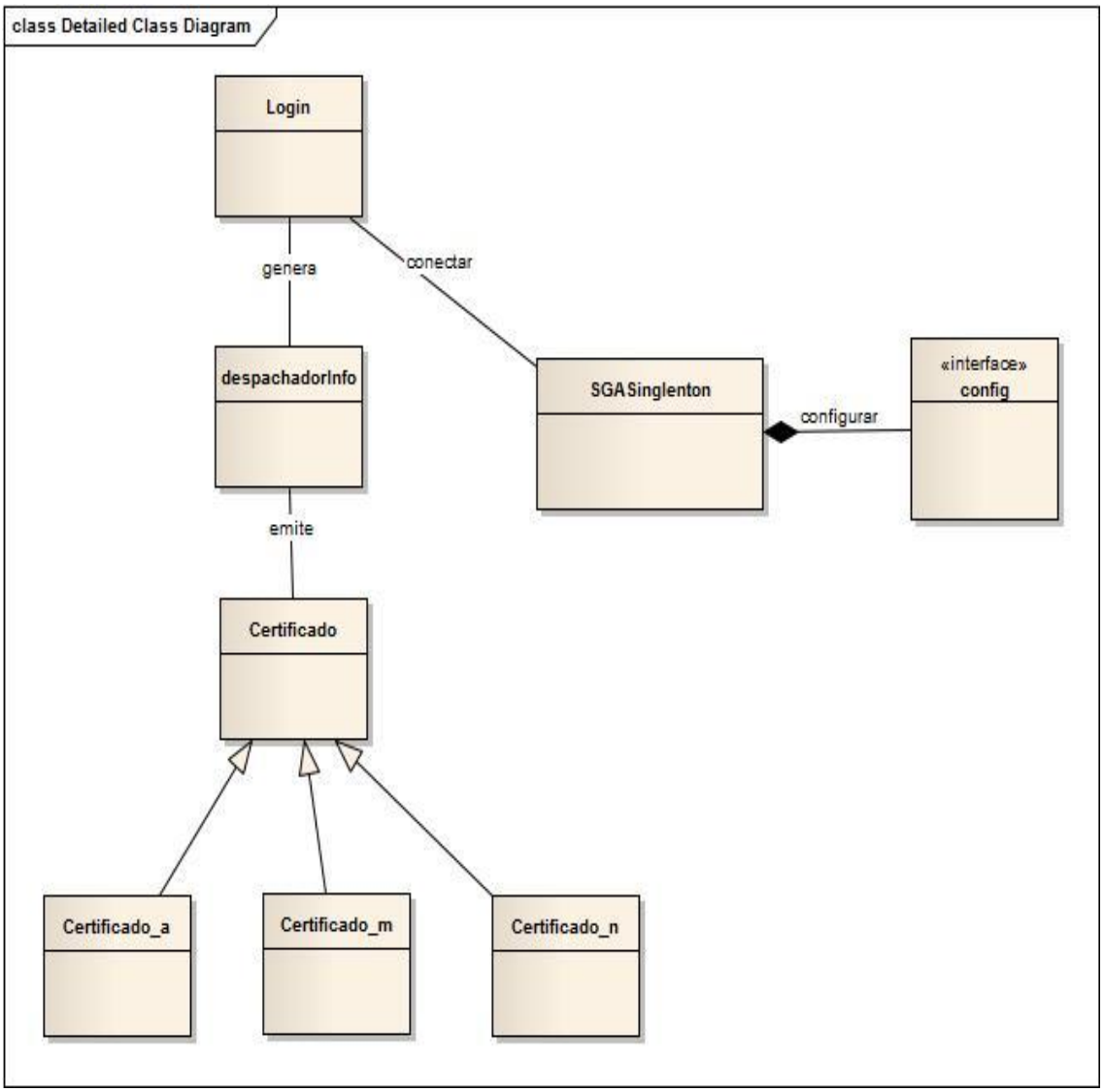

Figura 31. Modelo del dominio

1.5.1.2. Modelo de la aplicación-componentes

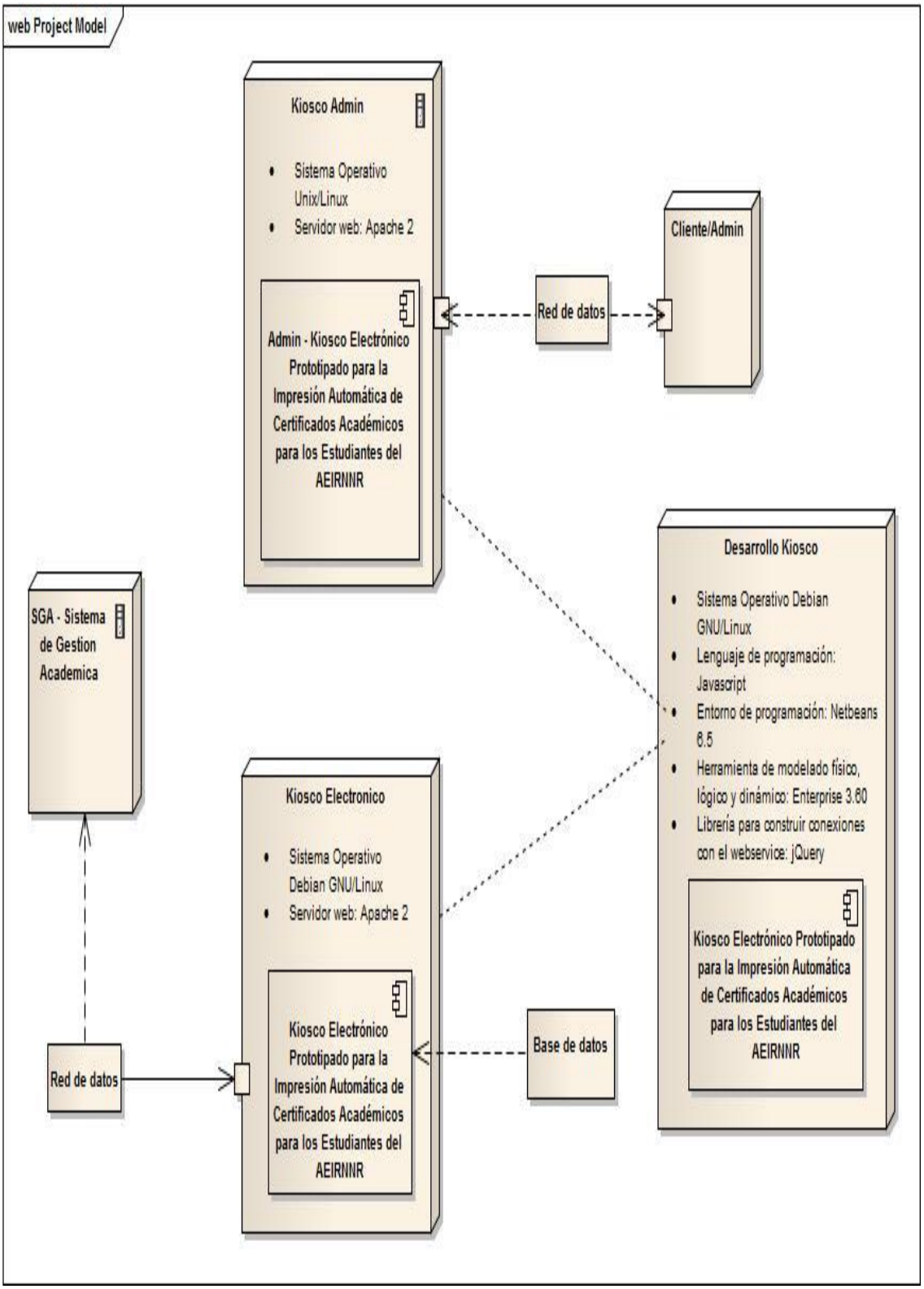

Figura 32. Modelo del dominio de la Aplicación- Componentes

1.6. MODELO DINÁMICO

A continuación se describen los casos de uso para obtener el certificado así como el caso de uso, obtener reportes del sistema.

1.6.1. Diagrama de Casos de Uso

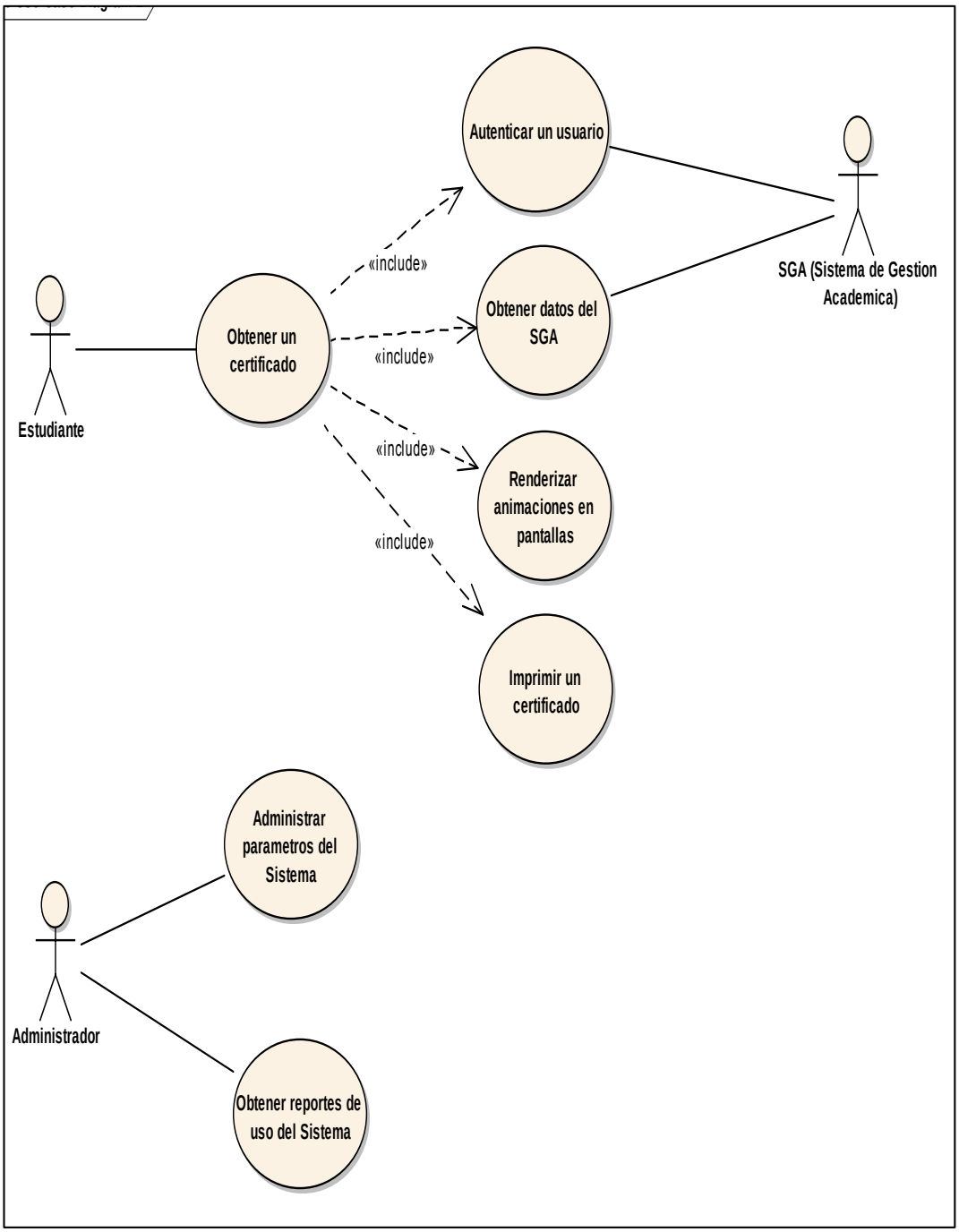

Figura 33. Diagrama de Casos de Uso

1.6.2. Descripción de Casos de Uso

A continuación la descripción de los casos de uso del sistema kiosko

1.6.2.1. Caso de Uso: Obtener un Certificado

Pantallas

Figura 34. Menú principal para emisión de certificados

Figura 35. Menú principal para emisión de certificados

TABLA X:

CASO DE USO: OBTENER CERTIFICADO

NOMBRE DEL CASO DE USO: OBTENER UN CERTIFICACO		**CÓDIGO:** 001CU
REFERENCIA DE REQUISITOS	**ACTORES**	**TIPO DE C.U.**
RF 001 RF 002 RF 003	- Estudiante - SGA	- Sistema
OBJETIVO (S)	**DESCRIPCIÓN**	
Obtener un certificado académico.	El estudiante debe autenticarse previamente para acceder a las opciones de emisión de certificados que presenta la aplicación del kiosko, las mismas que seleccionara a través de un keypad que	

	corresponden con un menú de botones numerados en la aplicación del kiosko.
PRE-CONDICIONES	**POST-CONDICIONES**
El sistema debe permitir el ingreso de los datos de autenticación y posteriormente la validación de los mismos. El estudiante debe visualizar la animación que explica el proceso de obtención de un certificado. El estudiante debe visualizar el menú para la selección de las opciones.	Se ha obtenido el certificado académico.
FLUJO NORMAL	
ACCIÓN DEL ACTOR	**RESPUESTA DEL SISTEMA**
1. El estudiante elige una de las opciones presentadas en el menú de opciones para la emisión de un certificado a través de un keypad. Pudiendo seleccionar: certificado de asistencia, certificado de notas, certificado de matrícula, otros certificados o terminar.	
	2. El sistema recoge la petición y se conecta con el SGA para obtener la o las carreras del estudiante y presenta una pantalla de selección de carrera.
3. El estudiante selecciona a través del keypad la carrera de la que necesita emitir el certificado.	
	4. El sistema recoge la petición y se conecta con el SGA para obtener los módulos que el

	estudiante ha cursado y presenta una pantalla de selección de módulos.
5. El estudiante selecciona a través del keypad el módulo del que necesita obtener el certificado con las teclas ← y →, se desplaza por la pantalla de módulos.	**6.** El sistema emite un mensaje confirmando los datos seleccionados hasta el momento por el estudiante y pide que confirme los mismos.
7. El estudiante a través del keypad confirma los datos.	**8.** El sistema presenta el certificado respectivo en la pantalla de impresión previa.
9. El estudiante selecciona la opción de imprimir o cancelar con la tecla correspondiente del keypad.	**10.** El sistema imprime el certificado. **11.** El sistema agradece por usar el servicio del kiosko y sale a la pantalla principal.
12. El estudiante presiona el botón correspondiente del keypad dependiendo si desea o no realizar otra emisión. El caso de uso continúa en el paso 1.	
FLUJO ALTERNO	
A. EL ESTUDIANTE NO DESEA IMPRIMIR EL CERTIFICADO **A9.** El estudiante selecciona la opción cancelar con la tecla correspondiente del keypad **A10.** El Sistema muestra el menú de opciones y continúa en el paso 1 del curso normal.	

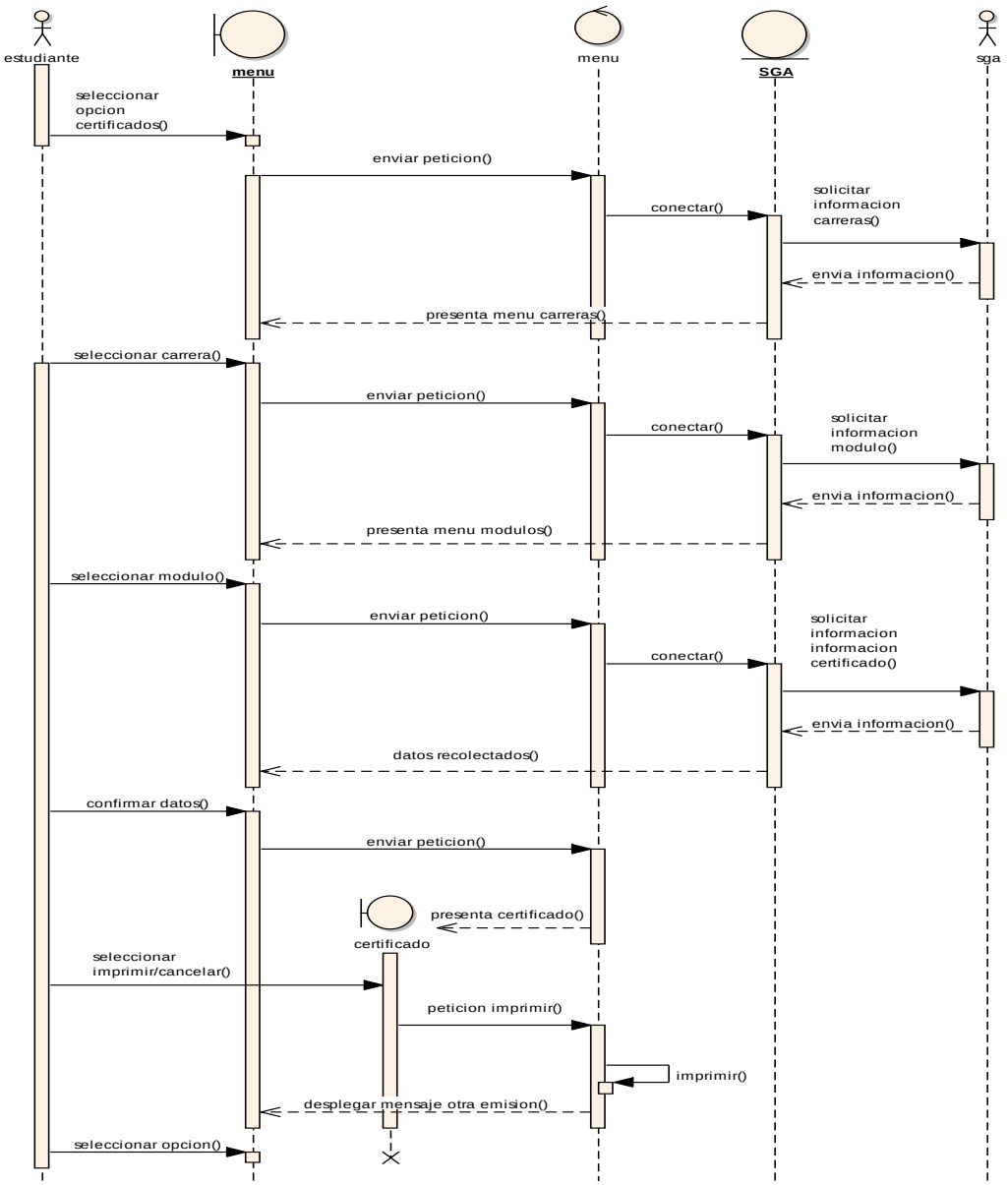

Figura 36. Diagrama de secuencia Caso de Uso: Obtener un certificado (Flujo normal)

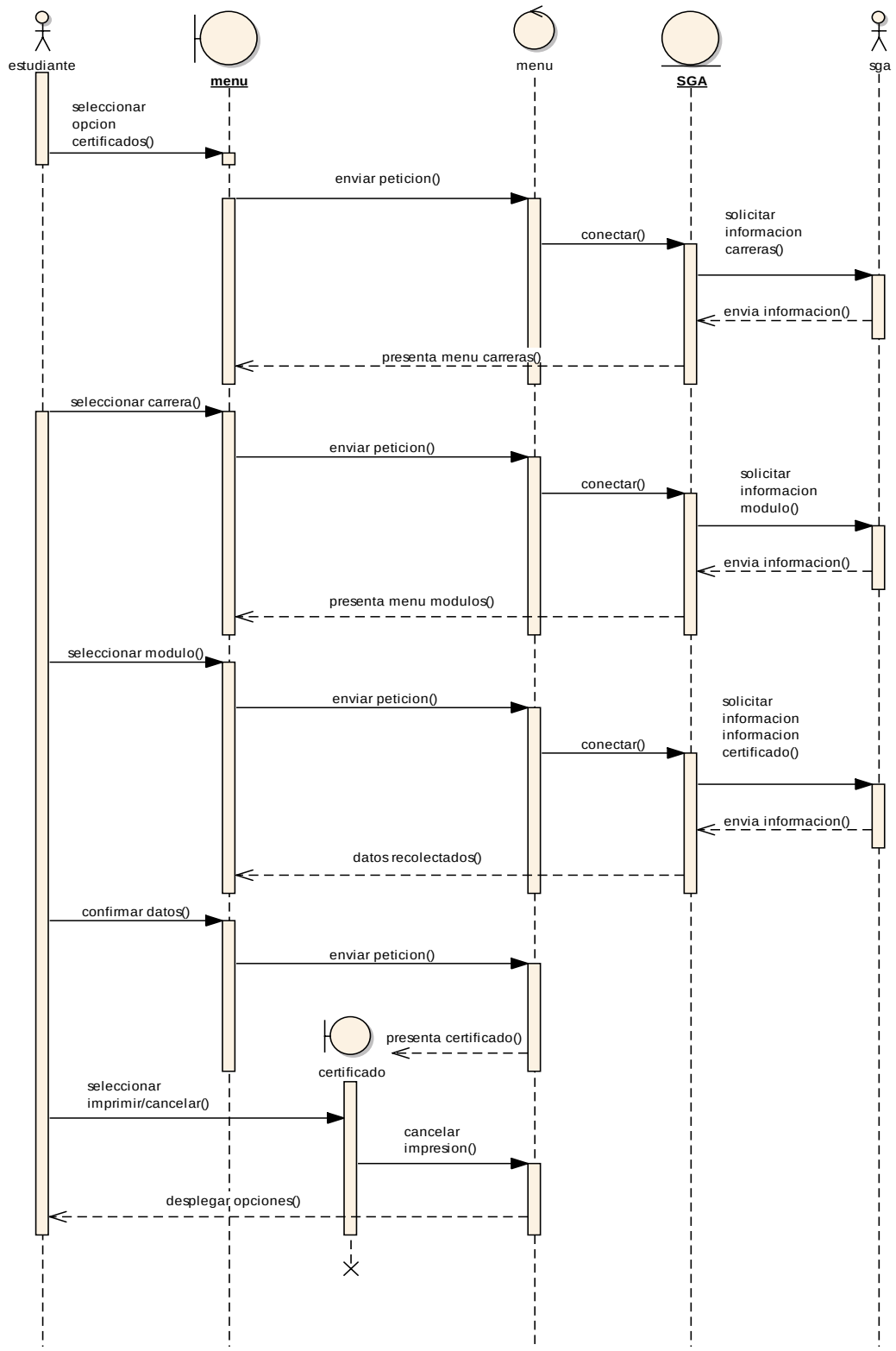

Figura 37. Diagrama de secuencia Caso de Uso: Obtener un certificado (Flujo alterno A)

1.6.2.2. Caso de Uso: Autenticar un usuario

Pantalla:

Figura 38. **Ventana "Autenticar un usuario"**

TABLA XI:

CASO DE USO AUTENTICAR UN USUARIO

NOMBRE DEL CASO DE USO: AUTENTICAR UN USUARIO		**CÓDIGO:** 002CU
REFERENCIA DE REQUISITOS	**ACTORES**	**TIPO DE C.U.**
RF 001 RF 002	- Estudiante - SGA	- Sistema
OBJETIVO (S)	**DESCRIPCIÓN**	
Autenticar o validar un estudiante en el sistema.	El estudiante debe ingresar los datos para su debida autenticación en el sistema una vez que el mismo haya obtenido una autenticación exitosa desde el SGA.	

PRE-CONDICIONES	POST-CONDICIONES
El estudiante visualiza la pantalla de ingreso de datos para la autenticación.	Se ha autenticado al estudiante.
FLUJO NORMAL	
ACCIÓN DEL ACTOR	**RESPUESTA DEL SISTEMA**
1. El estudiante ingresa la cédula y la clave a través del keypad del kiosko. **3.** El usuario presiona el botón “SI” o el botón 1 del keypad. . **5.** El usuario visualiza la pantalla con un menú de opciones donde puede realizar la emisión de certificados.	**2.** El sistema envía la petición de autenticación al SGA **4.** El SGA verifica los datos de autenticación y envía la respuesta al Sistema. De ser la respuesta correcta y una vez verificado que los datos del estudiante son correctos el sistema pone al usuario como validado.
FLUJO ALTERNO	
A. LOS DATOS DEL ESTUDIANTE SON INCORRECTOS **A6.**El SGA envía una respuesta de datos incorrectos del estudiante, el sistema no valida al estudiante. **A5.** El Sistema muestra la pantalla de ingreso de datos para autenticación, el caso de uso continua en el paso 1 del curso normal.	

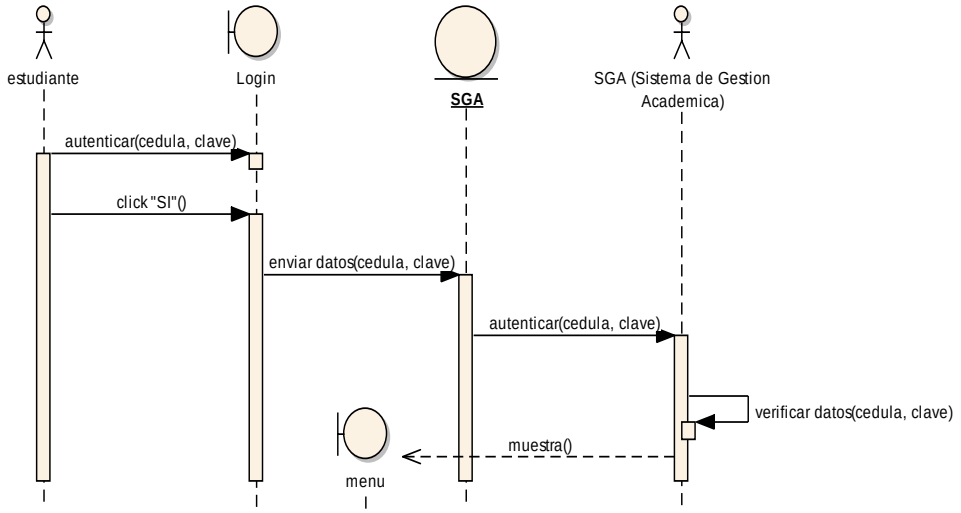

Figura 39. Diagrama de secuencia Caso de Uso: Autenticar un usuario (Flujo normal)

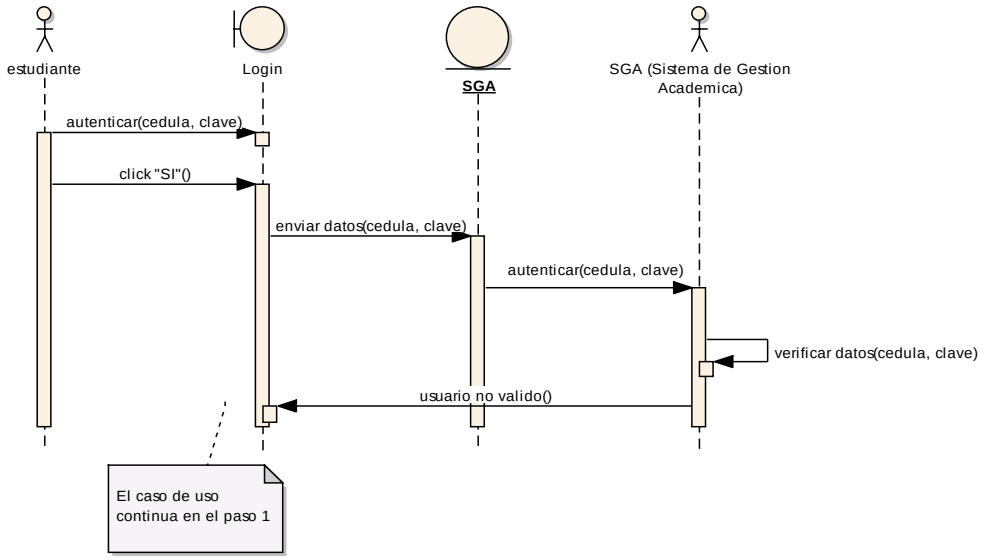

Figura 40. Diagrama de secuencia Caso de Uso: Autenticar un usuario (Flujo alterno A)

1.6.2.3. Caso de Uso: Administrar parámetros del sistema.

Pantalla

Figura 41. Ventana Principal de Administración de parámetros del sistema.

TABLA XII:

CASO DE USO: ADMINISTRAR PARÁMETROS

NOMBRE DEL CASO DE USO: ADMINISTRAR PARAMETROS DEL SISTEMA		**CÓDIGO:** 003CU
REFERENCIA DE REQUISITOS	**ACTORES**	**TIPO DE C.U.**
RF 007 RF 008 RF 010	- Administrador	- Sistema
OBJETIVO (S)	**DESCRIPCIÓN**	
Administrar la tabla de parámetros básicos de la aplicación.	El administrador del kiosko al entrar a la opción de administración de la aplicación deberá autenticarse para poder actualizar la tabla de los parámetros básicos de la misma como: Datos de red local y la dirección IP del sistema.	

PRE-CONDICIONES	POST-CONDICIONES
El administrador deberá acceder de forma remota al panel de administración del kiosko. El administrador debe visualizar las opciones de administración para la selección de una de ellas.	Se ha obtenido el certificado académico.
FLUJO NORMAL	
ACCIÓN DEL ACTOR	**RESPUESTA DEL SISTEMA**
1. El usuario ingresar opción de administración del kiosko.	
	2. El sistema presenta una pantalla de ingreso de la clave para autenticarse como administrador.
3. El usuario administrador debe proporcionar su clave para ingresar al sistema. **4.** El usuario presiona el botón “aceptar” para enviar los datos que se autentifiquen.	
	5. El sistema verifica que los datos ingresados sean correctos. **6.** El sistema presenta la pantalla de administración de parámetros.
7. El usuario tiene la opción de ingresar los parámetros de datos de red local y la dirección IP del sistema. **8.**El usuario presiona el botón “aceptar” para mandar a actualizar la tabla parámetros.	
	9. El sistema actualiza la tabla de parámetros y pide al usuario que reinicie

10. El usuario reinicia el sistema. El caso de uso continúa en el paso 1.	el sistema para que los cambios tengan efecto.
FLUJO ALTERNO	
A. EL USUARIO CANCELA LA AUTENTICACION COMO ADMINISTRADOR **A4.** El usuario presiona el botón ”CANCELAR” **A5.** El Sistema muestra el panel de administración del kiosko el caso de uso continúa en el paso 1 del curso normal. **B.DATOS INCORRECTOS DEL USUARIO** **A5.** El sistema verifica que los datos del usuario son incorrectos. **A6.** El caso de uso continúa en el paso 2 del curso normal.	

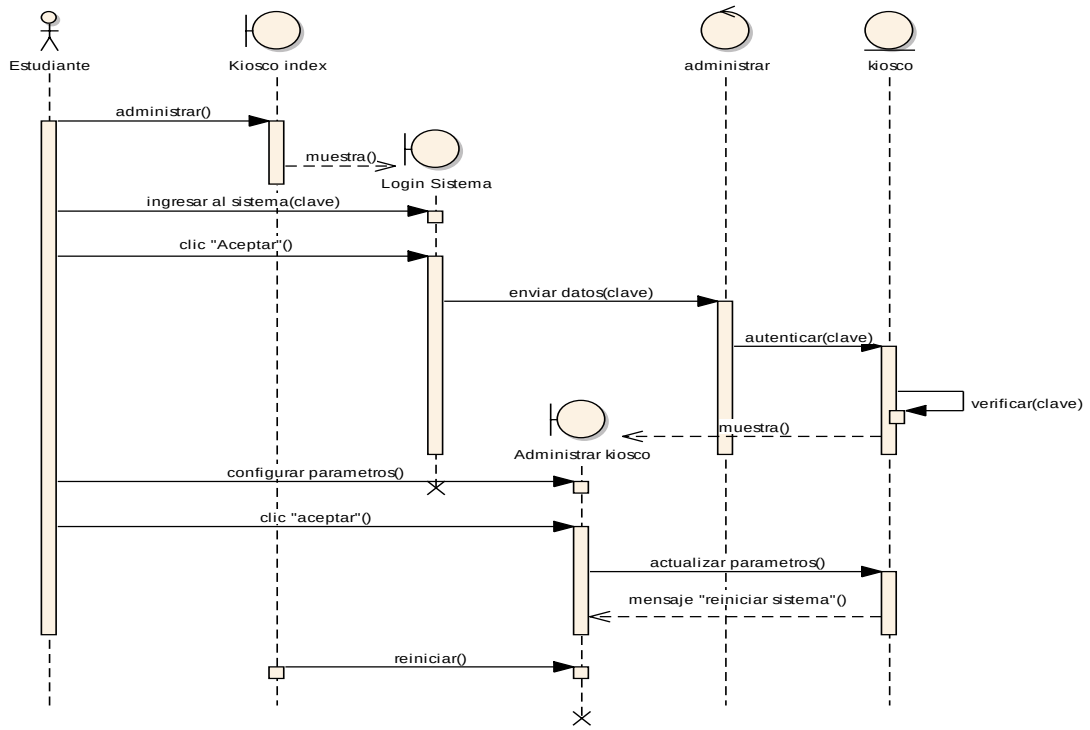

Figura 42. Diagrama de secuencia Caso de Uso: Administrar parámetros del sistema (Flujo normal)

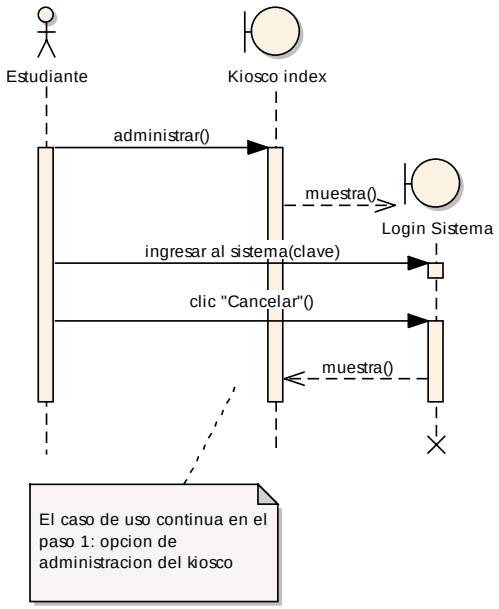

Figura 43. Diagrama de secuencia Caso de Uso: Administrar parámetros del sistema (Flujo alterno A)

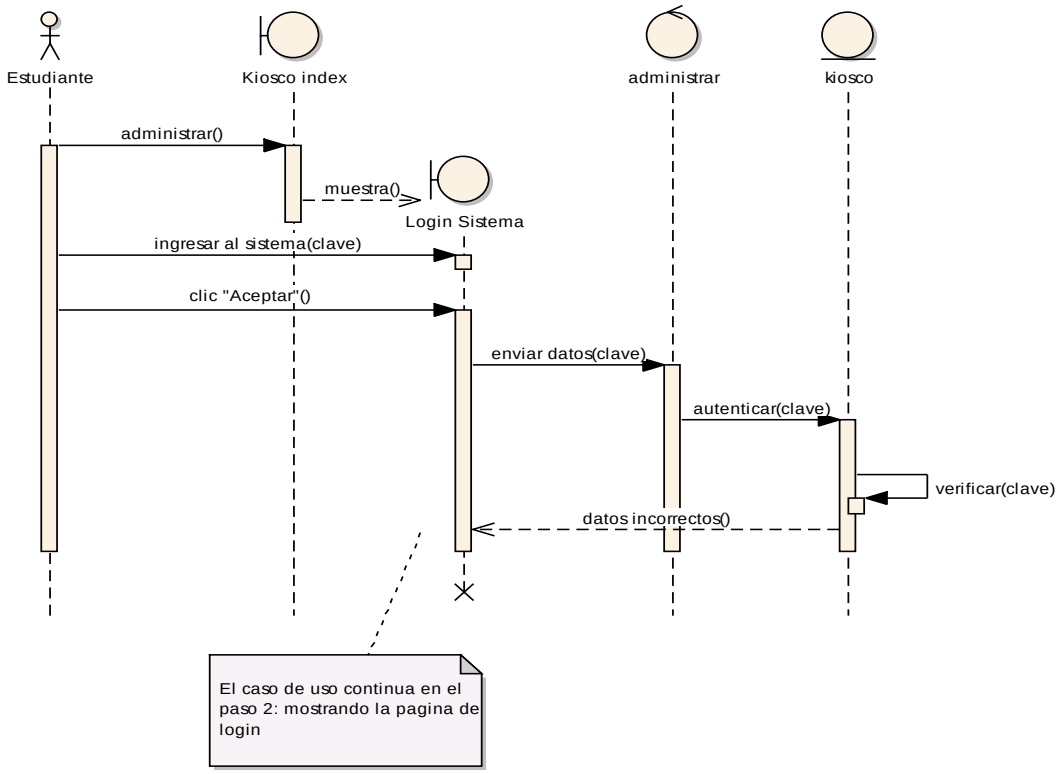

Figura 44. Diagrama de secuencia Caso de Uso: Administrar parámetros del sistema (Flujo alterno B)

1.6.2.4. Caso de Uso: Obtener reportes de uso

Pantalla

Figura 45. Pantalla para administración del kiosko

TABLA XIII:

CASO DE USO OBTENER REPORTES DE USO

NOMBRE DEL CASO DE USO: OBTENER REPORTES DE USO.		**CÓDIGO:** 004CU
REFERENCIA DE REQUISITOS	**ACTORES**	**TIPO DE C.U.**
RF 008 RF 009	- Administrador	- Sistema
OBJETIVO (S)	**DESCRIPCIÓN**	
Obtener reportes de uso del sistema accediendo de manera remota al panel de control del mismo.	El administrador del sistema del kiosko deberá acceder de forma remota al panel de administración del kiosko para obtener reportes de uso del mismo.	
PRE-CONDICIONES	**POST-CONDICIONES**	

El administrador deberá acceder de forma remota al panel de administración del kiosko. El administrador debe visualizar las opciones de administración para la selección de una de ellas.	Se ha obtenido los reportes de uso del sistema.
FLUJO NORMAL	
ACCIÓN DEL ACTOR	**RESPUESTA DEL SISTEMA**
1. El usuario debe ingresar a la opción de administración.	
	2. El sistema presenta una pantalla de ingreso de la clave para autenticarse como administrador.
3. El usuario administrador debe proporcionar su clave para ingresar al sistema. **4.** El usuario presiona el botón “aceptar” para enviar los datos que se autentifiquen.	
	5. El sistema verifica que los datos ingresados sean correctos. **6.** El sistema presenta la pantalla de administración de parámetros.
7. El usuario tiene la opción de ingresar los parámetros de datos de ip del sistema, base de datos, títulos, mensajes y otros parámetros **8.** El usuario presiona el botón “aceptar” para mandar a actualizar la tabla parámetros.	
	9. El sistema actualiza la tabla de parámetros y presenta la pantalla actualizada.
FLUJO ALTERNO	

A. EL USUARIO CANCELA LA AUTENTICACION COMO ADMINISTRADOR
A4. El usuario presiona el botón ”CANCELAR”
A5. El Sistema muestra el panel de administración del kiosko el caso de uso continuo en el paso 1 del curso normal.
B. DATOS INCORRECTOS DEL USUARIO
A5. El sistema verifica que los datos del usuario son incorrectos.
A6. El caso de uso continúa en el paso 2 del curso normal.

1.7. Modelo y Arquitectura

Para el desarrollo del presente proyecto se usó los siguientes componentes hardware y software:

Arquitectura

- Computador i686, Max speed 300Mhz, Current speed 566Mhz
- Memoria SIMM 256 MB
- Puertos USB, para uso de Impresora
- Puerto Serial, para conexión del teclado
- Disco Duro 40Gb Disco Duro.
- Impresora HP Deskjet 1000 Printer series - J110
- Teclado tipo keypad, con reemplazo de membranas táctiles por botones tipo push.
- Placa de circuitos para encendido de equipos, impresora, monitor y mainboard.
- Montaje de madera tropan y hierro de las siguientes medidas:

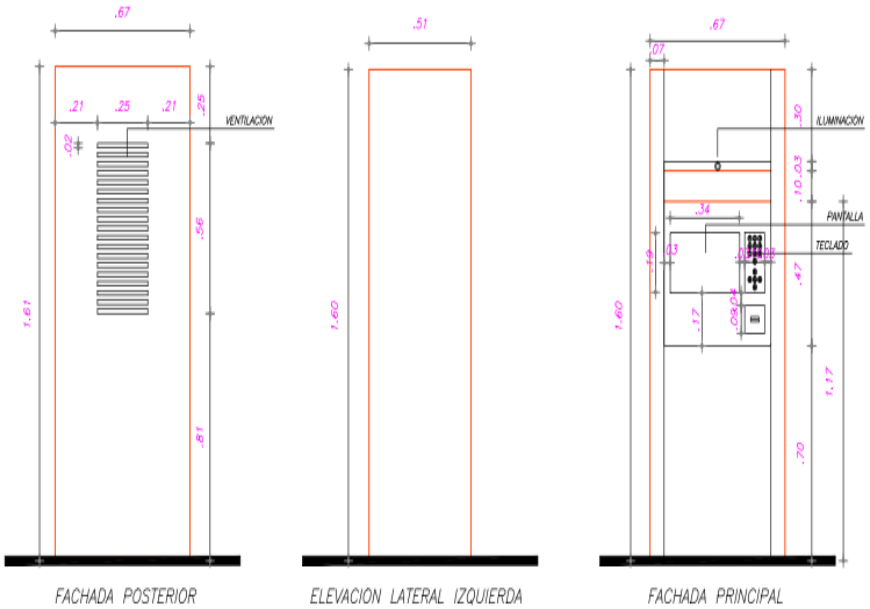

Figura 46. Modelo de Kiosko

Software

- Sistema Operativo Debian GNU/Linux
- Lenguaje de programación: Javascript
- Entorno de programación: Netbeans 6.5
- Servidor web: Apache 2
- Herramienta de modelado físico, lógico y dinámico: Enterprise 8
- Librería para construir conexiones con el webservice: jQuery
- Lenguaje de programación PHP5.

1.8 Construcción del Teclado

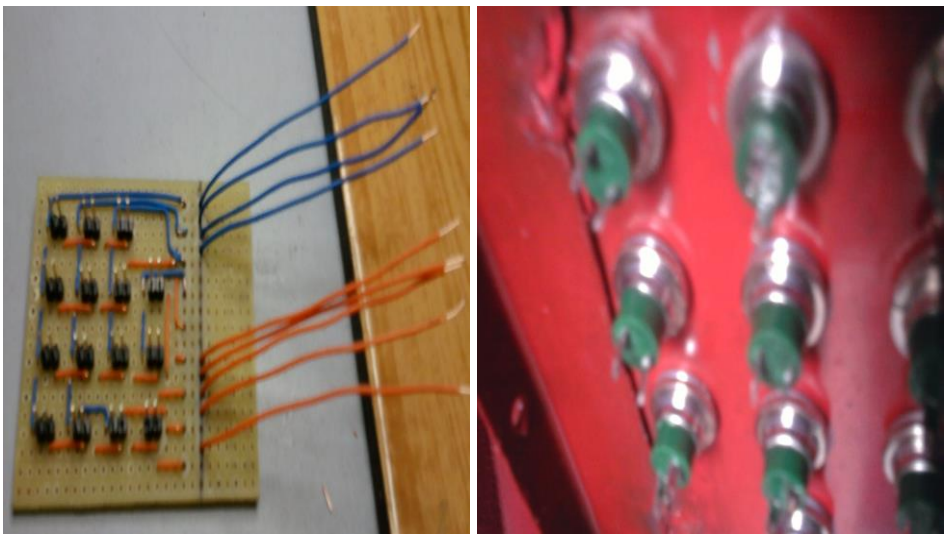

Figura 47. Construcción Teclado

Con la finalidad de ahorrar costos y a la vez aplicar los conocimientos adquiridos en la Universidad, hemos decidido construir el teclado del kiosko, el cual está basado en un Teclado estándar de pc de escritorio.

Para ellos acudimos al estudio de la parte interna del teclado, el mismo que tiene dos membranas que al hacer contacto entre sí, generan los caracteres, signos y números en la pantalla, se envían al procesador en nuestro caso para la elección del certificado que el estudiante necesite.

Una vez observado y comprobado los pulsantes decidimos presentar los que nos servirán para el uso de nuestro proyecto como son las flechas arriba, abajo, izquierda, derecha, y el símbolo más que representará la tecla enter, los diez dígitos, que se ubican en el panel numérico, ya que la elección de certificados se realizará pidiendo la digitación de un número según la necesidad del estudiante.

Se ha usado una tarjeta perforada PCB para la circuitería de los pulsantes y la tarjeta madre del teclado de mesa, de la cual hemos utilizado los pines 15, 16, 17 y 18 correspondientes a la membrana superior, de los correspondientes a la membrana inferior utilizaremos los pines 1, 2, 3, 4, 5 y 6 para la generación de los números que haciendo contacto entre pines generan un número, tal es el caso del número nueve que se lo obtiene al cerrar el circuito entre los pines 17 y 1 de la placa madre del teclado.

Para más detalle presentamos el esquema de los pulsantes, de los circuitos, las combinaciones de pines para la generación de los números del panel numérico, flechas y el signo más.

PINAJE DE ENTRADA DEL CONTROLADOR DE TECLADO DE PC

	MEMBRANA 1																		MEMBRANA 2							
CARACTER	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	1	2	3	4	5	6	7	8
q	1																		1							
a	1																				1					
z	1																						1			
1	1																									1
w		1																	1							
s		1																			1					
x		1																					1			
2		1																								1
e			1																1							
d			1																		1					
c			1																				1			
3			1																							1
r				1															1							
t				1																1						
f				1																	1					
g				1																		1				
v				1																			1			
b				1																				1		
5				1																					1	
4				1																						1
u					1														1							
y					1															1						
j					1																1					
h					1																	1				
m					1																		1			
n					1																			1		
6					1																				1	
7					1																					1
i						1													1							
+						1														1						
k						1															1					
8						1																				1
o							1												1							
9							1																			1
p										1									1							
ñ										1											1					
0										1																1
7															1				1							
4															1					1						
1															1						1					
↓															1									1		
8																1			1							
5																1				1						
2																1					1					
0																1						1				
→																1								1		
9																	1		1							
6																	1			1						
3																	1				1					
+																		1	1							
↑																		1				1				
←																		1						1		

Figura 48. Pinaje del Teclado

El teclado común consta de dos membranas que al hacer contacto en un punto determinado, el controlador detecta su coordenada y envía el dato por el puerto PS2.

Para desarrollar el teclado del kiosko fue necesario identificar cada uno de los caracteres del teclado

Y para nuestro caso específico trabajaremos con el pad numérico del y los cursores, también la tecla (+) la cual nos servirá como botón de confirmación.

PINAJE ESPECIFICO																											
	MEMBRANA 1																		MEMBRANA 2							
CARACTER	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	1	2	3	4	5	6	7	8
7															1				1							
4															1					1						
1															1						1					
↓															1									1		
8																1			1							
5																1				1						
2																1					1					
0																1						1				
→																1								1		
9																	1		1							
6																	1			1						
3																	1				1					
+																		1	1							
↑																		1				1				
←																		1						1		

Figura 49. Pinaje Específico

Habiendo identificado nuestros caracteres y mapeado las coordenadas, se procede a la conexión de los pulsantes.

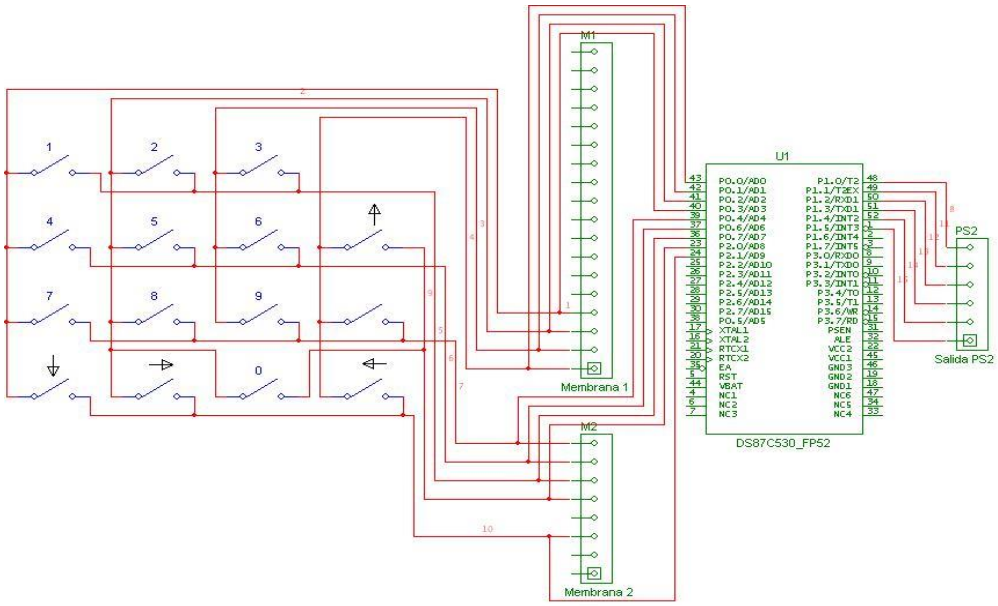

Figura 50. Esquema de coordenadas del teclado, generación de números

2. Valoración técnica económica ambiental

2.1. Valoración técnica económica

El presente trabajo de titulación se ha desarrollado con la utilización de diferentes recursos como son: humanos, materiales y técnicos-tecnológicos. A continuación en la tabla XV, se indica detalladamente los recursos humanos que hace referencia al personal que aportó en el desarrollo del proyecto, además del asesoramiento técnico del tutor.

TABLA XIV:
RECURSOS HUMANOS

Recursos	Número de horas	Costo de horas	Costo total
Alexandra Jadán	1100	$ 5.00	$ 5.500
Mayra Loja	1100	$ 5.00	$ 5.500
Juan M. Galindo	50	-	-
		Subtotal	$ 11.000

En la tabla XVI, se detallan los recursos materiales utilizados y que han sido necesarios para representar los diferentes avances, entrevistas y el informe final con respectivos manuales del trabajo de titulación.

TABLA XV:
RECURSOS MATERIALES

Recursos	Cantidad	Valor Unitario	Valor Total
Cartuchos de tinta	5	$ 23.00	$ 115.00
Resmas de papel	2	$ 3.75	$ 7.50
Movilización	2	$ 0.25	$ 200.00
Internet	600	$ 0.60	$ 360.00

Copias	500	$ 0.02	$ 10.00
Anillados	5	$ 2.00	$ 10.00
Empastados	4	$ 4.00	$ 16.00
Suministros de oficina	-	$ 20.00	$ 20.00
		Subtotal	$ 738.50

En la tabla XVII, se indica detalladamente cada uno de los recursos técnicos y tecnológicos que se emplearon en el desarrollo del presente trabajo de titulación, también se detallan recursos adquiridos de forma gratuita.

TABLA XVI:

RECURSOS TÉCNICOS Y TECNOLÓGICOS

RECURSOS	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Computadoras	2	$ 1200.00	$ 2400.00
Impresoras	1	$ 65.00	$ 65.00
Disco USB	2	$ 8.00	$ 16.00
CONSTRUCCIÓN FÍSICA DEL KIOSKO			
Computador i686	1	$ 150.00	$ 150.00
Impresora HP J110	1	$ 45.00	$ 45.00
Teclado tipo keypad	1	$ 80.00	$ 80.00
Piloto automático	1	$ 120.00	$ 120.00
Montaje tropan y hierro	1	$ 250.00	$ 250.00
Monitor LG 14``	1	$ 100.00	$ 100.00
Supresor de picos	2	$ 5.00	$ 10.00

Cables de red	3	$ 4.00	$ 12.00
SOFTWARE			
Sistema Operativo Debian GNU/Linux		Gratuito	$ 0.00
NetBeans 6.5		Gratuito	$ 0.00
Servidor Apache 2		Gratuito	$ 0.00
Enterprise 8		Gratuito	$ 0.00
Librería JQuery		Gratuito	$ 0.00
		Subtotal	$ 3248.00

En la tabla XVII, se indica el presupuesto del proyecto de trabajo de titulación, el cual se realizó en función de la estimación de los recursos humanos, materiales, técnicos y tecnológicos.

TABLA XVII.

PRESUPUESTO DEL PROYECTO

Recursos	Subtotal
Recursos Humanos	$ 11000.00
Recursos Materiales	$ 738.50
Recursos técnicos y tecnológicos	$ 3248.00
Total	**$ 14138.00**

2.2. Valoración Ambiental

Para la construcción del kiosko electrónico se ha recurrido al análisis exhaustivo de los materiales más convenientes para el desarrollo de la aplicación, en consecuencia de ellos, se determinó reutilizar tecnología que ha sido reemplazada por nuevas

tecnologías, sin embargo que tengan los requerimientos de hardware necesarios para funcionar como un sistema embebido, rápido y eficiente.

H. CONCLUSIONES

Al finalizar el presente proyecto se ha llegado a las siguientes conclusiones:

- Gracias a la aplicación de técnicas y métodos de investigación se logró disponer de la información sobre las demandas de certificaciones de los estudiantes del Área de la Energía, las Industrias y los Recursos Naturales no Renovables, lo que permitió obtener los requerimientos del sistema necesarios con el fin de obtener una herramienta eficiente y acorde a las necesidades de los estudiantes.

- Se eligió el toolkits jQuery, por ser una librería muy flexible, rápida y efectiva, la misma que permitió el desarrollo de la herramienta con excelente integración de Ajax, obteniendo de esta concatenación un alto nivel de renderización del sistema.

- Se logró el diseño de una aplicación dedicada en Línux con la distro Debian, para interactuar con los usuarios mediante un procedimiento gráfico de selección e interacción con el Sistema de Gestión Académica de la Universidad Nacional de Loja, a través de un teclado tipo push y una pantalla, brindando seguridad y eficiencia en la obtención de los diferentes certificados académicos.

- Con la implementación de este proyecto de investigación se logró demostrar que el prototipo de Kiosko diseñado y construido, funciona de manera eficaz, eficiente, entrega un servicio de calidad, aporta al proceso de mejora, calificación y engrandecimiento de nuestra Universidad.

- Se logró comprobar que el diseño y construcción del kiosko se constituye en una herramienta sólida y eficiente para la emisión de certificados académicos en la Universidad Nacional de Loja en todas sus áreas.

I. RECOMENDACIONES

Luego de haber realizado el presente trabajo de titulación se concluye con las siguientes recomendaciones:

- Impulsar el uso de kioskos electrónicos para la emisión de certificados académicos de las diferentes Áreas de la Universidad, que permita brindar a los estudiantes un servicio de forma eficiente, eficaz y de calidad.

- Incrementar las opciones de generación de documentos legales como certificaciones para egresados, plan de estudios, oferta académica, con el fin de apoyar a las tareas administrativas de la Universidad Nacional de Loja.

- Promover el desarrollo de software con tecnologías web y toolkits javascript, valuado por su facilidad en el momento de desarrollo y efectos visuales llamativos en su interacción con el usuario.

- Incentivar la generación de proyectos de investigación, que incluyan diseño de software, construcción y ensamblaje de equipos, con el fin de nivelar el desarrollo institucional con los avances tecnológicos de la actualidad, en el Área de la Energía, las Industrias y los Recursos Naturales no Renovales.

- Reutilizar tecnologías de hardware en la creación de proyectos, como computadores de prestaciones básicas que han sido reemplazados y que puedan funcionar para procesos empotrados o dedicados con la finalidad de reducir costos, y aportar a la conservación de la ecología.

- Aportar con desarrollo de aplicaciones que utilicen la información contenida por el Sistema de Gestión Académica de la Universidad Nacional de Loja.

J. BIBLIOGRAFÍA

Referencias:

[1] GUTIERREZ, M.A. Métodos y Técnicas de Investigación, Texto guía para el estudiante, Quito, Editorial Época, 1984.

[2] CABRERA, F.A. y ESPIN, J.V. (1986): Medición y evaluación Educativa. Barcelona: PPU.

[3] Aquino Carrillo Corazón Lusleth, García Carrillo Emmanuel Antonio, Luna Montiel Liliana, Padilla Álvarez Gustavo. (2011, Agosto 12). Informática V – ICONIX. [En línea] Disponible en http://informatica-v-iconix.blogspot.com/

[4] Tecnología y los kioskos electrónicos, Wikipedia (2011, enero 08), [En línea]. Disponible en: http://es.wikipedia.org/wiki/kiosco

[5] Modo Kiosko en Firefox, Chrome, Opera, Safari e IE, 2009. [En línea]. Disponible en: http://gespadas.com/modo-kiosko

[6] Web Services Description Language, 2009. [En línea]. Disponible en http://es.wikipedia.org/wiki/WSDL

[7] VANDER HENST, Cristian 2005. ¿Qué es la Web 2.0? [En Línea]. Disponible en http://www.maestrosdelweb.com/editorial/web2/

[8] MENEZ Julia, 2006. DojoToolkits [En línea].Disponible en <http://es.wikipedia.org/wiki/Dojo_toolkit>.

[9] DAMIAN, Alexi, 2009. AJAX.[En Línea]. Disponible en [http://es.wikipedia.org/wiki/AJAX]

[10] PEREZ, Javier. 2002. Introducción a JavaScript. España [En Línea][Consulta: 23 de Noviembre 2009]. Disponible en [http://www.librosweb.es/javascript/]

[11] PEREZ, Javier. 2009. La librería jQuery. [En línea]. Disponible en http://librosweb.es/ajax/capitulo_10/la_libreria_jquery.html

[12] Apache, Diccionario de Informática y Tecnología. [En línea]Disponible en http://www.alegsa.com.ar/Dic/apache.php

[13] Pablo Luis De Nápoli, Mini Tutorial de LaTeX - versión 0.2.1, sept 2008. [En línea]. Disponible en: http://mate.dm.uba.ar/~pdenapo/tutorial-latex/node2.html

[14]Página oficial de Debían. “Configuración-interfaces-red-bajo-Debían”. [En línea]. Disponible en http://www.esdebian.org/wiki/configuracion-interfaces-red-bajo-debian

[15] DavidCollie -10 Sep 2005, Linux│kioskoSetup, [En línea]. Disponible en: http://www.eng.uwaterloo.ca/twiki/bin/view/Linux/KioskSetup

[16] Daysi Ordóñez, Silvana Pacheco. Tesis-SGAWebserviceUNL, [En línea]. Disponible en http://ws.unl.edu.ec/sgaws/académica.

[17]Laura M. Castro Souto. A Coruña, 8 – 17 de Noviembre de 2004, [En línea] Disponible en: http://webdelprofesor.ula.ve/ciencias/ccova/Latex/manual.pdf

[18]Extraído de: UML y Patrones. 2ª Edición. Craig Larman. Prentice Hall. 2003 http://jupiter.utm.mx/~tesis_dig/11004.pdf

[19] Iván Trueva, 2011. Construcción de una placa PCB [En línea]. Disponible en http://ocw.bib.upct.es/pluginfile.php/7990/mod_resource/content/1/Construccion_pcb.pdf

[20]Ing. Ricardo Godínez Bravo, [En línea]. Disponible en: http://www.azc.uam.mx/cbi/electronica/rgodinez/Docs_PROTEL/Fotolitografia.pdf

[21] Carlos Iglesias. SISTEMA INTEGRADO, 2011 [En línea]. Disponible en http://diccionario.sensagent.com/SISTEMA%20INTEGRADO/es-es/

[22] Digital learning, Qué hace un servidor web como Apache? Configuración (marzo, 2012). [En línea]. Disponible en: http://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/

[23] BAEZA-YATES, Ricardo. RIBEIRO-NETO, Berthier. (2000), Modern Information Retieval, Addison-Wesley.

[24] BROWN, C.M. DANZING, P.B. HARDY, D. MAMBER, U. y SCHWARTZ, F. (1994), The Harvest information discovery and access system, in Proceedings of the Second International World Wide Web Conference, pp. 763-771.

[25] COOLEY, Robert. TAN, Pang-Ning. SRIVASTAVA, Jaideep. “Discovery of interesting usage patterns from web data”.

k. ANEXOS

Anexo 1. Anteproyecto del trabajo de titulación

UNIVERSIDAD NACIONAL DE LOJA

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

Proyecto de Desarrollo

"DISEÑO Y CONSTRUCCIÓN DE UN KIOSKO ELECTRÓNICO PROTOTIPADO PARA LA IMPRESIÓN AUTOMÁTICA DE CERTIFICADOS ACADÉMICOS PARA LOS ESTUDIANTES DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES".

Proyecto de Desarrollo Previa a la obtención del Grado de Ingeniero en Sistemas.

Autoras:

Alexandra Esperanza Jadán Patiño

Mayra Vanessa Loja Sandoya

LOJA- ECUADOR

2010

1. TÍTULO:

“DISEÑO Y CONSTRUCCIÓN DE UN KIOSKO ELECTRÓNICO PROTOTIPADO PARA LA IMPRESIÓN AUTOMÁTICA DE CERTIFICADOS ACADÉMICOS PARA LOS ESTUDIANTES DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y DE LOS RECURSOS NATURALES NO RENOVABLES”.

2. Problemática:

2.1. Situación Problemática

Desde el inicio de la sociedad, las personas siempre han desarrollado nuevos métodos y herramientas para mejorar las actividades diarias, con ello se ha creado diferentes tipos de comunicación entre las personas que va desde la comunicación a través de cartas hasta el procesamiento de la información de manera electrónica. Pero con el transcurso del tiempo las personas empezaron a organizarse en diferentes áreas y se implementaron niveles de organización tantos en las instituciones privadas como públicas. Dentro de estos nuevos tipos de organización la información juega un papel muy importante ya que el manejo se lo realizaba de diferentes maneras como entrega de cartas, oficios, reportes.

Actualmente para presentar, entregar, mostrar o difundir la información se la realiza utilizando los reportes impresos, publicaciones, medios audios visuales, y en el Internet. Es por ello que dentro de las universidades se están desarrollando nuevas alternativas para que la información fluya de manera más rápida y el trámite de entrega de documento se lo lleve de manera más eficiente.

Dentro de la Universidad Nacional de Loja se ha hecho necesario mejorar estos tipos de procesos mediante la implementación del Sistema de Gestión Académica que ha permitido automatizar los procesos que se llevaban de manera completamente manual en nuestra universidad. Pero a pesar de contar con esta herramienta donde reposa la información académica de los estudiantes se hace necesario mejorar la recepción y emisión de certificados académicos a los estudiantes.

En la actualidad el Sistema de Gestión Académica brinda los servicios informáticos académicos a toda hora sin importar el día, pero a la vez todo sistema informático es utilizado por los usuarios del sistema, dichos usuarios

dentro de nuestra Universidad cumplen con un horario administrativo establecido por cada una de las Áreas Académicas Administrativas, lo que impide que la emisión de certificados académicos se la realice de forma ininterrumpida ocasionando que los estudiantes al momento de solicitar esta clase de documentos académicos se rijan al horario administrativo de los usuarios del sistema.

A esto se suma que los estudiantes para realizar estos tipos de trámites deben obligadamente solicitar con un oficio físico dicho documento académico de manera personal en las secretarías de cada una de las carreras, ocasionando que los estudiantes empleen demasiado tiempo para la realización de dicha actividad.

Es por ello que el desarrollo de nuevas tecnologías informáticas orientadas en la automatización de certificados académicos, ayudarán a eliminar los requisitos o trámites innecesarios con el objetivo de mejorar el servicio que se presta a los estudiantes que soliciten dichos documentos académicos legales.

Enmarcados en éste contexto es que proponemos como tema de proyecto de tesis **“Diseño y Construcción de un Kiosko Electrónico Prototipado para la Impresión Automática de Certificados Académicos para los Estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables”**, con el cual pretendemos crear una herramienta informática que permita mejorar los servicios de emisión de certificados académicos capaz de satisfacer los requerimientos de los estudiantes.

2.2. Problema General de Investigación

Dificultad de obtener certificados académicos debido a los trámites manuales requeridos para su emisión, y a la carga de solicitudes de despacho por parte de los secretarios.

2.3. Delimitación

2.3.1. Problemas específicos de Investigación

Una vez realizado un análisis minucioso en la problemática planteada, se ha detectado algunos inconvenientes, los mismos que a nuestro criterio, han sido definidos como problemas específicos de investigación:

- ✓ Escaso análisis de datos para la interpretación de la información y determinar las necesidades del Área de la Energía, las Industrias y los Recursos Naturales no Renovables.
- ✓ Desconocimiento en la utilización y aplicación de técnicas para identificar los requerimientos funcionales y el atributo necesario para el desarrollo del kiosko electrónico.
- ✓ Selección incorrecta de la metodología para el desarrollo de herramientas informáticas que vinculen la construcción del hardware y el desarrollo del software apropiado a los requerimientos técnicos del hardware.
- ✓ La falta de un medio o solución tecnológica que permita automatizar el proceso de obtención de certificados académicos.

2.3.2. Espacio

El presente tema de investigación denominado **“Diseño y Construcción de un Kiosko Electrónico Prototipado para la Impresión Automática de Certificados Académicos para los Estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables”**, será realizado e

implementado en el Área de la Energía, las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja.

2.3.3. Tiempo

El proceso de desarrollo de la presente investigación estará comprendido durante el período de un año a partir de la aprobación del tema.

2.3.4. Unidades de Observación

- ✓ Formatos de certificados académicos.
- ✓ Equipo de desarrollo del Sistema de Gestión Académica.
- ✓ Secretaría del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.

3. Justificación

3.1. Justificación

La Universidad Nacional de Loja a través del Sistema Académico Modular por Objetos de Transformación (SAMOT), propicia el logro de aprendizajes significativos, en un proceso de vinculación permanente de la docencia, la investigación y la extensión, en base de las problemáticas y demandas de la sociedad.

Es así que, tomando como pilar fundamental la extensión, nos orientamos a vincular la teoría con la práctica, aplicando todos los conocimientos adquiridos durante nuestra vida universitaria. Todo este proceso lo ejecutaremos mediante la construcción de un Kiosko Electrónico y la realización de un Sistema Informático que permita la emisión de certificados académicos, y con ello eliminar los trámites burocráticos para la obtención de dichos certificados

El tema de investigación denominada “Diseño y Construcción de un Kiosko Electrónico Prototipado para la Impresión Automática de Certificados para los Estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables”, es una propuesta fundamentada por el grupo de desarrollado para mejorar los servicios de emisión de certificados académicos que prestan las secretarias del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.

Para cumplir con los objetivos planteados, nos apoyaremos de los conocimientos adquiridos en nuestra formación académica, así como también en los que se han adquirido en el desarrollo de proyectos investigativos de carácter científico – técnico realizado en los diferentes módulos como parte del cumplimiento del Sistema Académico Modular por Objetos de Transformación.

3.2. Viabilidad

3.2.1. Viabilidad Económica

El presente proyecto de desarrollo es viable económicamente porque el grupo de investigación está en la capacidad de cubrir todos los costos que se generarán dentro de todas las fases que comprenda este proyecto.

3.2.2 Viabilidad Técnica

Para el presente proyecto se puede manifestar que existe factibilidad técnica debido a que cuenta con el apoyo suficiente del grupo de desarrolladores del Sistema de Gestión Académica, quienes nos asesorarán y permitirán el acceso a la información mediante la utilización de los servicios web.

3.2.3 Viabilidad Operativa

De igual manera el presente proyecto de desarrollo es viable porque el grupo de investigación cuenta con la autorización y la colaboración por parte de la autoridades del Área de la Energía, las Industrias y los Recursos Naturales no Renovables. Dicho proyecto pretende ayudar a mejorar los servicios informáticos para la emisión de certificados académicos.

4. Objetivos

4.1. General

Aportar en el desarrollo de sistemas automatizados para la gestión académica de la Universidad Nacional de Loja mediante la generación del software y las especificaciones de hardware de un Kiosko Electrónico Prototipo, que ofrezca a los estudiantes un servicio de generación e impresión de certificados académicos de forma automática y basada en tecnologías Web 2.0.

4.2 Objetivos Específicos

- ✓ Disponer de la información sobre las demandas de certificaciones de los estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.
- ✓ Seleccionar el toolkits javascript de generación de tecnologías Web 2.0, que mejor corresponda a las necesidades y alternativas técnicas más convenientes.
- ✓ Establecer las especificaciones de Hardware y Sistema Operativo para la operación dedicada del Kiosko.
- ✓ Contar con un Software Gráfico derivado del toolkit javascript elegido, embebido como proceso dedicado del Sistema Operativo del Kiosko y que interactué con el Sistema de Gestión Académica de la Universidad Nacional de Loja.

5. Marco Teórico

5.1. SISTEMA DE GESTIÓN ACADÉMICA

El proyecto Sistema de Gestión Académica, es un conjunto de soluciones de software orientadas a la gestión académica de la Universidad Nacional de Loja. Este sistema consta de una gran variedad de módulos para tareas cotidianas tomando en cuenta roles y escenarios. Ha sido construido utilizando tecnología web, usando el framework de desarrollo RAD Turbogears, y una variedad de herramientas Open Source.

Se diseñó una arquitectura propia de software basándose en los siguientes estándares:

- ✓ Arquitectura de software multicapas, principalmente MVC (Modelo, Vista Controlador).
- ✓ Patrones de diseño de clases para la incorporación de extensiones y plugins, que soporten la adhesión de cambios en el camino (on the fly) con el menor impacto sobre los sistemas que ya estén desarrollados.
- ✓ Un marco de desarrollo (framework) que permite el rápido prototipado de programas y que soporte el desarrollo de aplicaciones en ambiente web.
- ✓ Un plan de iteraciones en el que se determinó el alcance de cada módulo a desarrollar, puntos de control, test unitarios, requerimientos específicos, cronograma de desarrollo así como el plan de versionamiento de cada módulo.

5.1.1. Lineamientos Generales.

- ✓ Usar tecnologías con larga expectativa de vida.
- ✓ Usar Ingeniería de Software como principal recursos para la resolución de problemas.
- ✓ Obtener productos mesurables a corto plazo basado en productos de renombrada calidad.

5.1.2. La Solución Tecnológica.

- ✓ Usar Software Libre
- ✓ Usar tecnologías RAD (Turbogears)
- ✓ Usar una metodología de desarrollo ágil (XP)
- ✓ Usar Python un lenguaje de programación multiparadigma
- ✓ Usar una base de datos libre y robusta.

5.1.3. Uso de tecnologías RAD

- ✓ Desarrollo de aplicaciones funcionales en corto tiempo (<90 días)
- ✓ Satisfacción de Requerimientos Cambiantes
- ✓ Usar patrones de diseño para desarrollar soluciones simples y elegantes a problemas específicos y comunes
- ✓ Hacerle frente a los cambios inesperados
- ✓ Resultados casi inmediatos

5.2 Web 2.0

5.2.1. Concepto

"La Web 2.0 es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones web enfocadas al usuario final. La Web 2.0 es una actitud y no precisamente una tecnología.

La Web 2.0 es la transición que se ha dado de aplicaciones tradicionales hacia aplicaciones que funcionan a través de los web enfocados al usuario final. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio”[1].

Es una etapa que ha definido nuevos proyectos en Internet y está preocupándose por brindar mejores soluciones para el usuario final. Muchos aseguran que hemos reinventado lo que era el Internet, otros hablan de burbujas e inversiones, pero la realidad es que la evolución natural del medio realmente ha propuesto cosas más interesantes como lo analizamos diariamente en las notas de Actualidad.

Y es que cuando el web inició, nos encontrábamos en un entorno estático, con páginas en HTML que sufrían pocas actualizaciones y no tenían interacción con el usuario.

Pero para entender de donde viene el término de Web 2.0 tenemos que remontarnos al momento en que Dale Dougherty de O’Reilly Media utilizó este término en una conferencia en la que compartió una lluvia de ideas junto a Craig Cline de MediaLive. En dicho evento se hablaba del renacimiento y evolución de la web.

5.2.2. Orígenes del Término.

El término fue acuñado por Dale Dougherty de O'Reilly Media en una tormenta de ideas con Craig Cline de MediaLive para desarrollar ideas para una conferencia. Dougherty sugirió que la web estaba en un renacimiento, con reglas que cambiaban y modelos de negocio que evolucionaban. Dougherty puso ejemplos "DoubleClick era la Web 1.0; Google AdSense es la Web 2.0.

En 2005, Tim O'Reilly definió el concepto de Web 2.0. El mapa meme mostrado resume el meme de Web 2.0, con algunos ejemplos de servicios.

[1] VANDER HENST, Cristian 2005. ¿Qué es la Web 2.0? [En Línea], [Consulta: 29 de septiembre 2009]. Disponible en < http://www.maestrosdelweb.com/editorial/web2/>

En su conferencia, O'Reilly y Battelle resumieron los principios clave que creen que caracterizan a las aplicaciones web 2.0: la web como plataforma; datos como el "Intel Inside"; efectos de red conducidos por una "arquitectura de participación"; innovación y desarrolladores independientes; pequeños modelos de negocio capaces de redifundir servicios y contenidos; el perpetuo beta; software por encima de un solo aparato.

En general, cuando mencionamos el término Web 2.0 nos referimos a una serie de aplicaciones y páginas de Internet que utilizan la inteligencia colectiva para proporcionar servicios interactivos en red dando al usuario el control de sus datos.

Así, podemos entender por Web 2.0, como propuso Xavier Ribes en 2007, "todas aquellas utilidades y servicios de Internet que se sustentan en una base de datos, la cual puede ser modificada por los usuarios del servicio, ya sea en su contenido (añadiendo, cambiando o borrando información o asociando datos a la información existente), bien en la forma de presentarlos o en contenido y forma simultáneamente".[2]

5.2.3. Tecnologías que dan vida a un proyecto Web 2.0:

Transformar software de escritorio hacia la plataforma del web.

- ✓Respeto a los estándares como el XHTML.
- ✓Separación de contenido del diseño con uso de hojas de estilo.
- ✓Sindicación de contenidos.
- ✓Ajax (javascript ascincrónico y XML).
- ✓Uso de Flash, Flex o Lazlo.
- ✓Uso de Ruby on Rails para programar páginas dinámicas.
- ✓Utilización de redes sociales al manejar usuarios y comunidades.
- ✓Dar control total a los usuarios en el manejo de su información.

[2] PEREZ, Javier. 2002. Introducción a JavaScript. España [En Línea][Consulta: 23 de Noviembre 2009]. Disponible en [http://www.librosweb.es/javascript/],

✓Proveer APis o XML para que las aplicaciones puedan ser manipuladas por otros.

✓Facilitar el posicionamiento con URL sencillos.

5.2.4. ¿En qué nos sirve la Web 2.0?

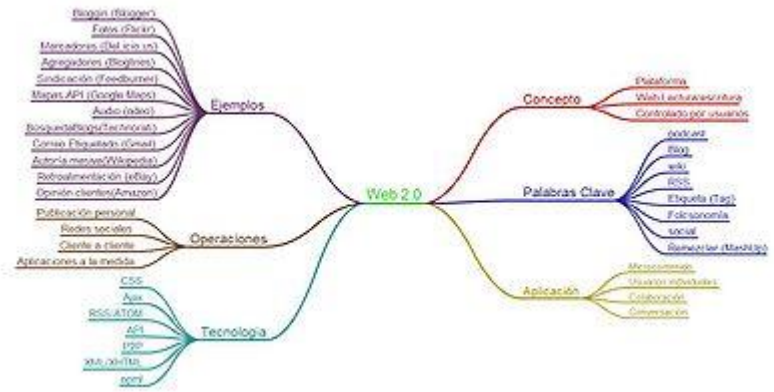

Figura 51. Servicios de la Web 2.0

El uso de él término de Web 2.0 está de moda, dándole mucho peso a una tendencia que ha estado presente desde hace algún tiempo. En Internet las especulaciones han sido causantes de grandes burbujas tecnológicas y han hecho fracasar a muchos proyectos.

Además, nuestros proyectos tienen que renovarse y evolucionar. El Web 2.0 no es precisamente una tecnología, sino es la actitud con la que debemos trabajar para desarrollar en Internet. Tal vez allí está la reflexión más importante del Web 2.0.

5.3. TOOLKITS

Existen herramientas que facilitan la creación de interfaces gráficas de usuario. Estas son conocidas como *toolkits*. “Un toolkit consiste en una biblioteca de widgets que pueden ser llamados por los programas de aplicación. Típicamente los widgets usados en los toolkits son objetos como barras de scroll, botones,

cuadros de texto, etc. En la figura de abajo se muestran algunos widgets del toolkit". [3]

Widgets del toolkit

Los toolkits facilitan enormemente la creación de Interfaz de Usuario (IUs); sin embargo, aun utilizando el toolkit, es necesario conocimientos de programación para la elaboración de la interfaz puesto que los widgets sólo tienen una interfaz procedural.

El uso de toolkits tiene grandes beneficios, como los que se mencionan a continuación:

- ✓ **Independencia de la Interfaz de Usuario**. Separa la parte correspondiente al diseño de la Interfaz de Usuario de las complejidades de la programación. Esta separación le permite a los diseñadores el realizar prototipos rápida y fácilmente, hacer revisiones en minutos, y de esta

[3] CRIVERA, John, 2000. Toolkits, [En línea]. [Consulta: 09 de Diciembre 2009]. Disponible en <http://www.fismat.umich.mx/~crivera/tesis/node20.html>

manera agilizar los procedimientos de prueba. La programación necesaria para el sistema se empieza a escribir una vez que el diseño de la Interfaz de Usuario se ha estabilizado.

- ✓ **Código más confiable***.* El código generado por los toolkits es frecuentemente más confiable que el código creado por un implementador humano, por lo que requiere una menor cantidad de debugging. Esto se debe a que el código generado automáticamente tiene un alto nivel de especificación.

- ✓ **Elaboración rápida de prototipos***.* Con un toolkit las Interfaces de Usuario pueden ser construidas y modificadas rápidamente. Gran parte del trabajo de diseño y construcción de un programa se debe a la creación de la IU. Al utilizar un toolkit para su elaboración, el tiempo de fabricación se reduce de una manera considerable. El tiempo ganado puede ser utilizado para hacer una mayor cantidad de pruebas de evaluación (probar, revisar, probar, revisar...), y tener al final un producto apropiado para el usuario final.

- ✓ **Interfaces de Usuario más fáciles de crear y de mantener.** Esto es porque las especificaciones de la interfaz pueden ser representadas, validadas y evaluadas más fácilmente. gracias al empleo de los widgets incluidos en el toolkit, la cantidad de código a escribir es mucho menor.

- ✓ **Facilidad para hacer modificaciones.** El empleo de un toolkit permite que la incorporación de cambios a la interfaz después de haber hecho las pruebas correspondientes sea más sencilla.

Un aspecto muy importante a considerar, y fundamental en este trabajo de tesis, es, como ya se ha mencionado, la utilización de herramientas que produzcan programas multiplataforma, lo que posibilita el que la interfaz pueda ser usada en distintos ambientes operativos como Windows o Unix, sin hacer modificación alguna a la Interfaz de Usuario.

Así pues, podemos concluir que el uso de un toolkit proveé al diseñador de la IU de gran control y flexibilidad.

5.3.1. Dojo Toolkits

"Dojo es un framework que contiene APIs y widgets (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Contiene un sistema de empaquetado inteligente, los efectos de Interfaz de Usuario, drag and drop APIs, widget APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX".[4]

Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas (bookmarking), y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Es conocido como "la navaja suiza del ejército de las bibliotecas Javascript". Proporciona una gama más amplia de opciones en una sola biblioteca JavaScript y es compatible con navegadores antiguos.

5.3.1.1 Complementos

Los complementos de Dojo son componentes pre empaquetados de código JavaScript, HTML y CSS que pueden ser usados para enriquecer aplicaciones web:

- ✓ Menús, pestañas y tooltips.
- ✓ Tablas ordenables, gráficos dinámicos y dibujado de vectores 2D.

[4] MENEZ, Julia, 2006. Dojo Toolkits [En línea] [Consulta: 13 de Obtubre 2009] Disponible en <http://es.wikipedia.org/wiki/Dojo_toolkit>.

- ✓ Efectos de animación y la posibilidad de crear animaciones personalizables.
- ✓ Soporte para arrastrar y soltar.
- ✓ Formularios y rutinas de validación para los parámetros.
- ✓ Calendario, selector de tiempo y reloj.
- ✓ Editor online de texto enriquecido.
- ✓ Núcleo de componentes (*dijit*) accesible desde versiones anteriores y lector de pantalla.

5.3.1.2. Comunicación Asíncrona

Una característica importante de las aplicaciones AJAX es la comunicación asíncrona entre el navegador y el servidor. Tradicionalmente, se realizaba con el comando JavaScript XMLHttpRequest. *Dojo* provee de una capa de abstracción (*dojo.io.bind*) para varios navegadores web con la que se pueden usar otros transportes (como *IFrames* ocultos) y diferentes formatos de datos. De esta forma podemos obtener los campos que se van a enviar como parámetros del formulario de una manera sencilla.

5.3.1.3. Sistema de paquetes

Dojo provee de un sistema de paquetes para facilitar el desarrollo modular. El script de inicio inicializa una serie de jerarquías de paquetes de espacios de nombre (*io*, *event*, etc.) bajo el paquete raíz *dojo*. Después de la inicialización del paquete *dojo*, cualquier otro paquete puede ser cargado (vía XMLHttpRequest o cualquier otro transporte similar) usando las utilidades ofrecidas en el arranque. También es posible inicializar paquetes adicionales dentro o al mismo nivel que el paquete *Dojo*, permitiendo extensiones o librerías de terceros.

Los paquetes de *Dojo* pueden contener múltiples archivos. Cualquier paquete o archivo puede depender de otro. En este caso, cuando el paquete es cargado, cualquier dependencia será también cargada.

Dojo también brinda una manera de crear perfiles; el sistema ofrece una lista de paquetes y usa Apache Ant para crear un archivo JavaScript comprimido que contiene dichos paquetes y dependencias. De esta manera se tiene todo el código necesario para ser cargado y es inicializado de una sola vez, permitiendo así el cacheado (la mayoría de los navegadores web no permiten el cacheado de archivos vía XMLHttpRequest).

5.3.1.4 Almacenamiento en el cliente de datos

Adicionalmente, ofrece funciones para leer y escribir cookies, proporcionando en el lado cliente una abstracción llamada *Dojo Storage*. *Dojo Storage* permite a la aplicación web almacenar datos en el lado cliente, persistencia y seguridad. Cuando se incluye en una página web, determina cual es el mejor método para almacenar la información. Cuando la aplicación web ha sido cargada desde el sistema de archivos (por ejemplo desde file://URL), *Dojo Storage* usa de manera transparente XPCOM en Firefox y ActiveX en Internet Explorer para mantener la persistencia de la información. El desarrollador que use *Dojo Storage* no se tiene que preocupar de esto, ya que *Dojo* tiene una capa de abstracción con métodos *put()* y *get()*.

5.3.1.5. Almacenamiento en el servidor

Desde enero de 2007, *Dojo* incluye las siguientes implementaciones de almacenamiento de datos en el paquete *dojo.data*:

- ✓ CsvStore: almacenamiento de sólo lectura y acceso CSV.
- ✓ OpmlStore: almacenamiento de sólo lectura y lectura jerárquica desde archivos en formato OPML.
- ✓ YahooStore: almacenamiento de sólo lectura que obtiene los resultado del servicio web del buscador de Yahoo Search!.
- ✓ DeliciousStore: almacenamiento de sólo lectura que obtiene los marcadores del servicio web que ofrece Del.icio.us.
- ✓ RdfStore: almacenamiento de solo lectura que usa SPARQL para comunicarse con el servidor de datos RDF.

5.3.1.6. Soporte para Adobe Integrated Runtime (AIR)

Dojo permite usar aplicaciones Adobe AIR basadas en JavaScript. Ha sido modificada para satisfacer los requisitos de seguridad de Adobe.

5.3.1.7. Desarrollo con Dojo

El desarrollo con *Dojo* puede hacerse bien con los IDE *Aptana* o *Komodo*. Existe una versión para la comunidad de *Aptana* y una versión de 21 días de prueba de *Komodo*. Los plugins de *Dojo* están disponibles para ambos (actualmente *Dojo 0.4.3* y *0.9.0*). Para el desarrollo WYSIWYG usando Dojo, se puede utilizar la aplicación de código abierto WaveMaker Visual Ajax Studio.

5.3.2 Script.aculo.us

script.aculo.us es una biblioteca JavaScript que permite el uso de controles AJAX, drag 'n drop, y otros efectos visuales en una página web.

script.aculo.us se distribuye mediante descargas en varios formatos de archivo, y también está incluido en Ruby on Rails y otros frameworks de desarrollo web.

5.4. Herramientas para el desarrollo

5.4.1. Ajax

Es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

"Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML"[5].

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dados que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

[5] DAMIAN, Alexi, 2009. AJAX. [En Línea]. [Consulta: 14 de Noviembre 2009]. Disponible en <[http://es.wikipedia.org/wiki/AJAX],>

AJAX es una combinación de cuatro tecnologías ya existentes:

- ✓XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- ✓Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- ✓El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- ✓XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Los navegadores que soportan las aplicaciones AJAX dependerán de las características que el navegador permita.

- ✓Navegadores basados en Gecko como Mozilla, Mozilla Firefox, SeaMonkey, Camino, K-Meleon, IceWeasel, Flock, Epiphany, Galeon y Netscape versión 7.1 y superiores.
- ✓Google Chrome
- ✓Microsoft Internet Explorer para Windows versión 5.0 y superiores, y los navegadores basados en él.
- ✓Navegadores con el API KHTML versión 3.2 y superiores implementado, incluyendo Konqueror versión 3.2 y superiores, Apple Safari versión 1.2 y superiores, y el Web Browser for S60 de Nokia tercera generación y posteriores.
- ✓Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores.

5.4.2. Javascript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

5.4.2.1. Definir JavaScript en un archivo externo

Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos XHTML enlazan mediante la etiqueta <script>. Se pueden crear todos los archivos JavaScript que sean necesarios y cada documento XHTML puede enlazar tantos archivos JavaScript como necesite.

Ejemplo: Archivo codigo.js

```
alert("Un mensaje de prueba");
```

Documento XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Ejemplo de código JavaScript en el propio documento</title>
        <script type="text/javascript" src="/js/codigo.js"></script>
    </head>
    <body>
```

```
            <p>Un párrafo de texto.</p>
        </body>
    </html>
```

Además del atributo type, este método requiere definir el atributo src, que es el que indica la URL correspondiente al archivo JavaScript que se quiere enlazar. Cada etiqueta <script> solamente puede enlazar un único archivo, pero en una misma página se pueden incluir tantas etiquetas <script> como sean necesarias.

Los archivos de tipo JavaScript son documentos normales de texto con la extensión .js, que se pueden crear con cualquier editor de texto como Notepad, Wordpad, EmEditor, UltraEdit, Vi, etc. La principal ventaja de enlazar un archivo JavaScript externo es que se simplifica el código XHTML de la página, que se puede reutilizar el mismo código JavaScript en todas las páginas del sitio web y que cualquier modificación realizada en el archivo JavaScript se ve reflejada inmediatamente en todas las páginas XHTML que lo enlazan.

5.5. Glosario básico

Script: cada uno de los programas, aplicaciones o trozos de código creados con el lenguaje de programación JavaScript. Unas pocas líneas de código forman un script y un archivo de miles de líneas de JavaScript también se considera un script. A veces se traduce al español directamente como "guión", aunque script es una palabra más adecuada y comúnmente aceptada.

Sentencia: cada una de las instrucciones que forman un script.

Palabras reservadas: son las palabras (en inglés) que se utilizan para construir las sentencias de JavaScript y que por tanto no pueden ser utilizadas libremente. Las palabras actualmente reservadas por JavaScript son: break, case, catch, continue, default, delete, do, else, finally, for, function, if, in, instanceof, new, return, switch, this, throw, try, typeof, var, void, while, with.

Sintaxis: La sintaxis de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.

La sintaxis de JavaScript es muy similar a la de otros lenguajes de programación como Java y C.

Las normas básicas que definen la sintaxis de JavaScript son las siguientes:

✓**No se tienen en cuenta los espacios en blanco y las nuevas líneas:** como sucede con XHTML, el intérprete de JavaScript ignora cualquier espacio en blanco sobrante, por lo que el código se puede ordenar de forma adecuada para entenderlo mejor (tabulando las líneas, añadiendo espacios, creando nuevas líneas, etc.)

✓**Se distinguen las mayúsculas y minúsculas:** al igual que sucede con la sintaxis de las etiquetas y elementos XHTML. Sin embargo, si en una página XHTML se utilizan indistintamente mayúsculas y minúsculas, la página se visualiza correctamente, siendo el único problema la no validación de la página. En cambio, si en JavaScript se intercambian mayúsculas y minúsculas el script no funciona.

✓**No se define el tipo de las variables:** al crear una variable, no es necesario indicar el tipo de dato que almacenará. De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.

✓**No es necesario terminar cada sentencia con el carácter de punto y coma (;)**: en la mayoría de lenguajes de programación, es obligatorio terminar cada sentencia con el carácter ;. Aunque JavaScript no obliga a hacerlo, es conveniente seguir la tradición de terminar cada sentencia con el carácter del punto y coma (;).

✓**Se pueden incluir comentarios:** los comentarios se utilizan para añadir información en el código fuente del programa. Aunque el contenido de los comentarios no se visualiza por pantalla, si que se envía al navegador del usuario

junto con el resto del script, por lo que es necesario extremar las precauciones sobre la información incluida en los comentarios.

✓**JavaScript define dos tipos de comentarios:** los de una sola línea y los que ocupan varias líneas.

5.6. Sistema Integrado, Empotrado o Embebido

Un sistema integrado, empotrado o embebido es un sistema informático de uso específico construido dentro de un dispositivo mayor. Los sistemas integrados se utilizan para usos muy diferentes a los usos generales a los que se suelen someter a las computadoras personales. En un sistema integrado la mayoría de los componentes se encuentran incluidos en la placa base.

Dos de las diferencias principales son el precio y el consumo. Puesto que los sistemas integrados se pueden fabricar por decenas de millares o por millones de unidades, una de las principales preocupaciones es reducir los costes. Los sistemas integrados suelen usar un procesador relativamente pequeño y una memoria pequeña para reducir los costes. Se enfrentan, sobre todo, al problema de que un fallo en un elemento implica la necesidad de reparar la placa íntegra.

Lentitud no significa que vayan a la velocidad del reloj. En general, se suele simplificar toda la arquitectura del ordenador o computadora para reducir los costes. Por ejemplo, los sistemas integrados emplean a menudo periféricos controlados por interfaces síncronos en serie, que son de diez a cientos de veces más lentos que los periféricos de un ordenador o computadora personal normal. Los primeros equipos integrados que se desarrollaron fueron elaborados por IBM en los años 1980.

Los programas de sistemas integrados se enfrentan normalmente a problemas de tiempo real.

5.6.1. Componentes de un Sistema Integrado

En la parte central se encuentra el microprocesador, microcontrolador, DSP, etc. Es decir, la CPU o unidad que aporta capacidad de cómputo al sistema, pudiendo incluir memoria interna o externa, un micro con arquitectura específica según requisitos.

La comunicación adquiere gran importancia en los sistemas integrados. Lo normal es que el sistema pueda comunicarse mediante interfaces estándar de cable o inalámbricas. Así un Sistema Integrado normalmente incorporará puertos de comunicaciones del tipo RS-232, RS-485, SPI, I²C, CAN, USB, IP, Wi-Fi, GSM, GPRS, DSRC, etc.

El subsistema de presentación tipo suele ser una pantalla gráfica, táctil, LCD, alfanumérico, etc.

Denominamos actuadores a los posibles elementos electrónicos que el sistema se encarga de controlar. Puede ser un motor eléctrico, un conmutador tipo relé etc. El más habitual puede ser una salida de señal PWM para control de la velocidad en motores de corriente continua.

El módulo de E/S analógicas y digitales suele emplearse para digitalizar señales analógicas procedentes de sensores, activar diodos LED, reconocer el estado abierto cerrado de un conmutador o pulsador, etc.

El módulo de reloj es el encargado de generar las diferentes señales de reloj a partir de un único oscilador principal. El tipo de oscilador es importante por varios aspectos: por la frecuencia necesaria, por la estabilidad necesaria y por el consumo de corriente requerido. El oscilador con mejores características en cuanto a estabilidad y coste son los basados en resonador de cristal de cuarzo, mientras que los que requieren menor consumo son los RC. Mediante sistemas PLL se obtienen otras frecuencias con la misma estabilidad que el oscilador patrón.

El módulo de energía (power) se encarga de generar las diferentes tensiones y corrientes necesarias para alimentar los diferentes circuitos del SE. Usualmente se trabaja con un rango de posibles tensiones de entrada que mediante conversores ac/dc

o dc/dc se obtienen las diferentes tensiones necesarias para alimentar los diversos componentes activos del circuito.

Además de los conversores ac/dc y dc/dc, otros módulos típicos, filtros, circuitos integrados supervisores de alimentación, etc. El consumo de energía puede ser determinante en el desarrollo de algunos sistemas integrados que necesariamente se alimentan con baterías, con lo que el tiempo de uso del SE suele ser la duración de la carga de las baterías.

5.6.2. Aplicaciones de un Sistema Integrado

Los lugares donde se pueden encontrar los sistemas integrados son numerosos y de varias naturalezas. A continuación se exponen varios ejemplos para ilustrar las posibilidades de los mismos:

- ✓ En una fábrica, para controlar un proceso de montaje o producción. Una máquina que se encargue de una determinada tarea hoy en día contiene numerosos circuitos electrónicos y eléctricos para el control de motores, hornos, etc. que deben ser gobernados por un procesador, el cual ofrece un interfaz persona – máquina para ser dirigido por un operario e informarle al mismo de la marcha del proceso.
- ✓ Puntos de servicio o venta (POS, Point Of Service). Las cajas donde se paga la compra en un supermercado son cada vez más completas, integrando teclados numéricos, lectores de códigos de barras mediante láser, lectores de tarjetas bancarias de banda magnética o chip, pantalla alfanumérica de cristal líquido, etc. El sistema empotrado en este caso requiere numerosos conectores de entrada y salida y unas características robustas para la operación continuada.
- ✓ Puntos de información al ciudadano. En oficinas de turismo, grandes almacenes, bibliotecas, etc. existen equipos con una pantalla táctil donde se puede pulsar sobre la misma y elegir la consulta a realizar, obteniendo una respuesta personalizada en un entorno gráfico amigable.
- ✓ Decodificadores y set-top boxes para la recepción de televisión. Cada vez existe un mayor número de operadores de televisión que aprovechando las tecnologías vía satélite y de red de cable ofrecen un servicio de televisión de pago diferenciado del convencional. En primer lugar envían la señal en formato digital MPEG-2 con

lo que es necesario un procesado para decodificarla y mandarla al televisor. Además viaja cifrada para evitar que la reciban en claro usuarios sin contrato, lo que requiere descifrarla en casa del abonado. También ofrecen un servicio de televisión interactiva o web-TV que necesita de un software específico para mostrar páginas web y con ello un sistema basado en procesador con salida de señal de televisión.

✓Sistemas radar de aviones. El procesado de la señal recibida o reflejada del sistema radar embarcado en un avión requiere alta potencia de cálculo además de ocupar poco espacio, pesar poco y soportar condiciones extremas de funcionamiento (temperatura, presión atmosférica, vibraciones, etc.).

- ✓ Equipos de medicina en hospitales y ambulancias UVI – móvil.
- ✓ Máquinas de revelado automático de fotos.
- ✓ Cajeros automáticos.
- ✓ Pasarelas (Gateways) Internet-LAN.
- ✓ Y un sin fin de posibilidades aún por descubrir o en estado embrionario como son las neveras inteligentes que controlen su suministro vía Internet, PC de bolsillo, etc.

5.7. Herramientas UML

5.7.1. Clases y Objetos

Clases.

Una Clase, desde una perspectiva conceptual, representa un conjunto de Objetos, el mismo que comparte las mismas propiedades (Atributos), el mismo comportamiento (Métodos), las mismas relaciones con otros Objetos (Mensajes) y la misma semántica dentro del sistema.

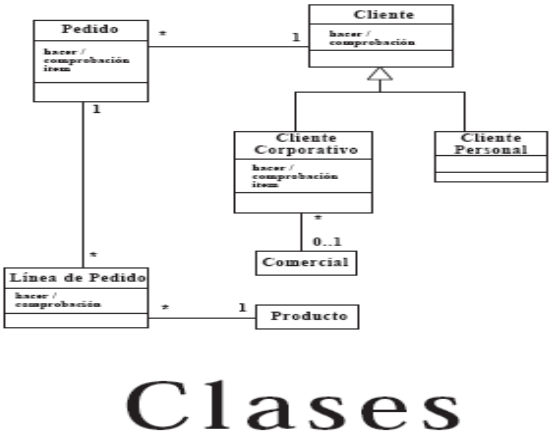

Figura 52: Diagrama de clases

Una clase puede representarse de forma esquemática (plegada), con los detalles como atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase.

Mientras que desde una perspectiva física, una Clase es una pieza de software que actúa como un molde para fabricar tipos particulares de objetos que disponen de los mismos atributos y métodos.

Objetos.

Un objeto se representa de la misma forma que una clase, por lo que representa una entidad del mundo real o inventado.

Los Objetos se estructuran por:

- ✓ La Identidad ¿Quién soy? = Atributos
- ✓ Propósito ¿Cuál es mi misión? = Justificación
- ✓ Responsabilidades ¿Qué debo hacer? = Métodos
- ✓ Procedencia ¿De qué Clase provengo? = Origen
- ✓ Relaciones ¿Qué mensajes entiendo? = Comportamiento

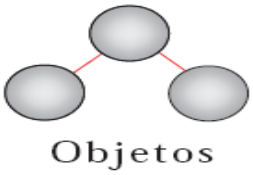

Figura 53: Objetos

Los objetos que se han creado a partir de una Clase concreta, se llaman instancias de esta Clase y se diferencian entre ellos únicamente por los valores de sus atributos (variables).

5.7.2. Diagramas de Clases

Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Los diagramas de Clases son estáticos, esto es, representan que partes interactúan entre sí, es decir muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc).

Las clases se documentan con una descripción de lo que hacen, sus métodos y sus atributos. Las relaciones entre clases se documentan con una descripción de su propósito, su cardinalidad (cuantos objetos intervienen en la relación) y su opcionalidad (cuando un objeto es opcional el que intervenga en una relación). La descripción de clases complejas se puede documentar con diagramas de estados.

5.7.3. Diagramas de Objetos.

Un diagrama de objetos representa un conjunto de objetos y sus relaciones en un momento concreto, estos son utilizados durante el proceso de Análisis y Diseño de los sistemas informáticos en la metodología UML.

Se utilizan para describir estructuras de datos, instantáneas de las instancias de los elementos encontrados en un diagrama de clases. Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase. Los diagramas de objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de

clase. Una diferencia con los diagramas de clase es que el compartimiento de arriba va en la forma, Nombre de objeto: Nombre de clase.

Los diagramas de objetos modelan las instancias de elementos contenidos en los diagramas de clases. Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento concreto. En UML, los diagramas de clase se utilizan para visualizar los aspectos estáticos del sistema y los diagramas de interacción se utilizan para ver los aspectos dinámicos de los sistemas, y constan de instancias de los elementos del diagrama de clases y mensajes enviados entre ellos. En un punto intermedio podemos situar los diagramas de objetos, que contiene un conjunto de instancias de los elementos encontrados en el diagrama de clases, representando sólo la parte estática de un interacción, consistiendo en los objetos que colaborar pero sin ninguno de los mensajes intercambiados entre ellos.

5.7.4. Modelo de Casos de Usos

Los Casos de Uso son requerimientos funcionales que describen de una manera detallada el comportamiento del sistema con los distintos Actores que interactúan con él independientes de la implementación.

5.7.4.1 Diagrama de Casos de Uso

El Diagrama de Casos de Usos, es la representación de la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Es aquí donde se muestra la relación entre los actores y los casos de uso del sistema.

Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

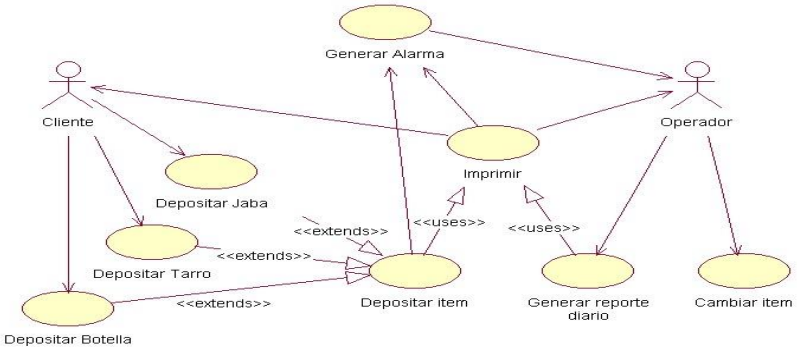

Figura 54: Diagrama de casos de uso

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: Actores, casos de uso y relaciones entre casos de uso.

Actores.

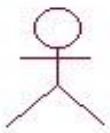

Figura 55: Actores

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes. Un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Casos de Uso.

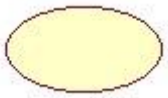

Figura 56: Casos de uso

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica, es decir, es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

Relaciones entre Casos de Uso.

Asociación.- Es el tipo de relación básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso).

Figura 57: Representación de asociación

Dependencia o Instanciación.- Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia.

Figura 58: Representación de dependencia

Generalización.- Cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>).

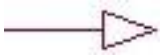

Figura 59: Representación de generalización

Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

Extends*:* Se utilizar cuando un caso de uso es similar a otro (características).

Uses: Se utiliza cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

5.7.4.2. Descripción de Casos de Uso

Es la descripción detallada del Caso de Uso; es decir es la descripción detallada de la funcionalidad del sistema en lo que se refiere a su interacción externa.

La descripción de los casos de uso comprende:

- ✓ **El inicio:** ¿Cuándo y qué actor lo produce?
- ✓ **El fin:** ¿Cuándo se produce y qué valor devuelve?
- ✓ **La interacción actor-caso de uso:** ¿Qué mensajes intercambian ambos?
- ✓ **Objetivo del caso de uso:** qué lleva a cabo o intenta?
- ✓ **Cronología y origen de las interacciones**
- ✓ **Repeticiones de comportamiento:** qué operaciones son iteradas?
- ✓ **Situaciones opcionales:** qué ejecuciones alternativas se presentan en el caso de uso.

5.7.5. Diagramas de Paquetes

Un paquete es el elemento de organización básica de un modelo de sistema UML. Los diagramas de Paquetes se usan para reflejar la organización de paquetes y sus elementos. Los usos más comunes de para los diagrama de paquete son para organizar diagramas de casos de uso y diagramas de clases, estos paquetes son como grandes contenedores de clases, es decir muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Un paquete puede contener paquetes subordinados, diagramas o elementos únicos, y se puede establecer su visibilidad así como la de los elementos que contiene.

Los elementos contenidos en un paquete comparten el mismo espacio de nombres, esto significa que los elementos contenidos en un mismo espacio de nombres específico deben tener nombres únicos.

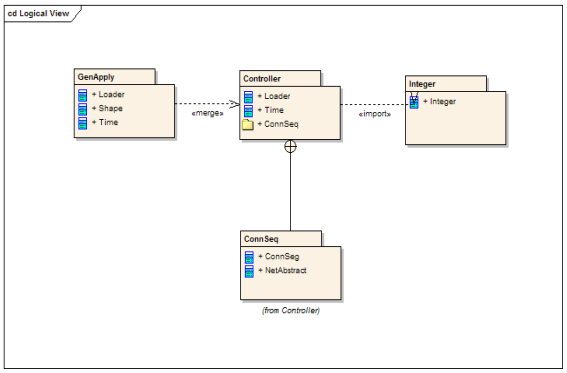

Figura 60: Diagrama de paquetes

Los Paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Con estas líneas maestras sobre la mesa, los paquetes son buenos elementos de gestión. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido.

A medida que aumenten sus conocimientos acerca de las características del problema, el modelo conceptual se hará más complicado y podrá alcanzar un tamaño difícil de controlar. Puede utilizar paquetes para dividir el modelo en subconjuntos más pequeños y manejables.

Los paquetes se pueden construir para representar relaciones tanto físicas como lógicas. Cuando se elige incluir las clases a los paquetes específicos, es útil asignar las clases con la misma jerarquía de herencia a los paquetes, las clases

que están relacionadas a través de la composición y las clases que colaboran que también tienen un fuerte argumento para ser incluidas en el mismo paquete.

Los paquetes se representan en UML como carpetas y contienen los elementos que comparten un espacio de nombre; todos los elementos dentro de un paquete deben tener un identificador único. El paquete debe mostrar el nombre del paquete y puede opcionalmente mostrar los elementos dentro del paquete en compartimientos extras.

Combinación de Paquetes: Cuando un conector «merge» se usa en un paquete, la fuente de la combinación importa los contenidos importados y anidados del destino. Si existe un elemento dentro del origen y el destino, las definiciones del elemento origen se expandirán para incluir las definiciones del elemento contenidas en el destino. Todos los elementos agregados o actualizados por una combinación se notan por una relación de generalización desde el origen hasta el destino.

5.7.6. Diagramas de Interacción

En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración

5.7.7. Diagrama de Secuencia.

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

6. Metodología

6.1. Matriz de Consistencia General.

Problema General de Investigación. Dificultad de obtener certificados académicos debido a los trámites manuales requeridos para su emisión, y a la carga de solicitudes de despacho por parte de los secretarios.			
Tema	**Objeto de Investigación**	**Objetivo de la Investigación**	**Hipótesis de Investigación**
Diseño y Construcción de un Kiosko Electrónico Prototipado para la Impresión Automática de Certificados para los Estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.	Automatizar los procesos de emisión de certificado académicos del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.	Aportar en el desarrollo de sistemas automatizados para la gestión académica del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables mediante la generación del software y las especificaciones de hardware de un Kiosko Electrónico Prototipo, que ofrezca a los estudiantes un servicio de generación e impresión de certificados académicos de forma automática y basado en tecnologías Web 2.0.	La construcción de un Kiosko y el desarrollo de una aplicación que permita a los estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables registrados en el Sistema de Gestión Académica, obtener sus certificados académicos de forma automática ayudarán a mejorar los servicios informáticos.

6.2. Materiales, Métodos y Técnicas de trabajo.

6.2.1. Materiales

Aquí detallaremos los materiales para el desarrollo de la presente investigación

TABLA VIII:

RECURSOS HARDWARE

Nº	Detalle	Características
1	Computador Portátil	Vostro XPS 13
1	Impresora Multifunción	Hp Deskjet F4480
1	Memoria Flash KINGSTON	4GB

TABLA XIX:

RECURSOS SOFTWARE

Detalle
Sistema Operativo Ubunto 9.10
Web 2.0

6.2.2. Métodos

Para la realización de la presente investigación el grupo de desarrollo hará uso de métodos y técnicas que ayudarán a cada uno de los pasos que debe seguir una investigación de carácter práctico científica, el cual sigua un orden secuencial, sistemático y lógico.

Esta investigación es de tipo Práctico-Científico ya que sigue los pasos que requiere la investigación científica y al final del proceso de investigación se procederá a su implementación, ejecución y corrección de errores.

Adicionalmente el grupo de investigación usará el Método Científico puesto que el objeto de estudio parte de los hechos dados en la realidad actual y que requieren de una fundamentación científica para poder aportar con una solución. También permitirá plantear los problemas específicos del problema general.

El Método Inductivo permitirá analizar los problemas particulares con un enfoque descriptivo de los procesos que se generan en la emisión de certificados académicos. Con la utilización de este método nosotros estaremos en capacidad de identificar los problemas y deducir en un problema general de investigación. La utilización del método deductivo permitirá analizar las causas de las problemáticas identificadas en proceso de inducción y así establecer las posibles soluciones al problema planteado por el equipo de investigación.

Finalmente cabe resaltar el uso del método Empírico que nos permitirá estudiar las características fundamentales y las relaciones esenciales con el problema de estudio y se la aplicará en la acumulación empírica de datos, para esta recopilación se hará uso de las técnicas de observación directa, encuesta, y la entrevista. Y por último el Método Teórico nos permitirá la explicación de manera científica de los datos e información recopilada.

6.2.3. Técnicas

En el proceso de recopilación de la información, toda investigación de carácter científico necesita el uso de técnicas capaces de obtener la información desde diferentes puntos de vista en el objeto de estudio.

La observación directa nos permitirá observar hechos, sucesos y procesos que se realizan en el espacio mismo donde se generará el proceso investigativo. Con la utilización de esta técnica nuestro grupo de desarrollo podrá identificar los problemas existentes y tomar la información para su posterior análisis. La importancia de la observación directa es que permitirá a los investigadores estar en contacto con el objeto de estudio, conocer sus fortalezas y debilidades.

La técnica del cuestionario es fundamental en el desarrollo de nuestro proyecto ya que podremos obtener la información de manera cuantitativa o una descripción estadística de los procesos que se desarrollan. Esta técnica está dirigida a los actores que están interactuando con el proceso de emisión de certificado como los estudiantes del Área de la Energía, las Industrias y los Recursos Naturales no Renovables. Luego de la obtención de la información se procederá a la tabulación de información haciendo uso de las tablas estadísticas que nos ayudará saber con exactitud cuáles son los requerimientos y necesidades de esta para de la población.

La entrevista es una técnica que establecerá un diálogo directo entre el grupo de investigación y las autoridades del Área de la Energía, las Industrias y los Recursos Naturales no Renovables y el Director del Proyecto Sistema de Gestión Académica.

7. Cronograma

Nombre de tarea	Duración	Comienzo	Fin
Cronograma de actividades	**354 días**	**jue 10/12/09**	**mié 02/02/11**
Analisis Preliminar e Ingeniería de Requerimientos	**52 días**	**jue 10/12/09**	**sáb 13/02/10**
Investigación preliminar	12 días	jue 10/12/09	mié 23/12/09
Clasificación de Documentos	11 días	lun 04/01/10	vie 15/01/10
Organización de la información.	5 días	sáb 16/01/10	jue 21/01/10
Documento de Requerimientos	15 días	vie 22/01/10	lun 08/02/10
Glosario de Términos	5 días	mar 09/02/10	sáb 13/02/10
Diseño	**157 días**	**lun 15/02/10**	**mar 17/08/10**
Modelo del dominio	12 días	lun 15/02/10	sáb 27/02/10
Elaboración del diagrama de casos de uso	27 días	lun 01/03/10	mié 31/03/10
Elaboracion del prototipado	13 días	jue 01/04/10	jue 15/04/10
Narración de los casos de uso	30 días	vie 16/04/10	vie 21/05/10
Elaboración de los diagramas de robustez	16 días	lun 24/05/10	jue 10/06/10
Elaboración de los diagramas de secuencia	43 días	vie 11/06/10	vie 30/07/10
Elaboración del diseño de la Base de Datos	14 días	lun 02/08/10	mar 17/08/10
Codificación de Software y Construcción de Hardwa	**105 días**	**mié 18/08/10**	**vie 17/12/10**
Construcción del Hardware	6 días	mié 18/08/10	mar 24/08/10
Codificación de sistema	57 días	mié 25/08/10	vie 29/10/10
Realizar pruebas de integración de hardware y softwa	21 días	lun 01/11/10	mié 24/11/10
Corrección de errores	20 días	jue 25/11/10	vie 17/12/10
Implementación y documentación	**40 días**	**sáb 18/12/10**	**mié 02/02/11**
Documentacion del Hardware	4 días	sáb 18/12/10	mié 22/12/10
Documentación del proyecto de titulación	12 días	lun 03/01/11	sáb 15/01/11
Prueba final	15 días	lun 17/01/11	mié 02/02/11

8. Presupuesto y Financiamiento

8.1. Recursos.

Todo proyecto de carácter científico requiere de recursos tanto económico, humanos, tecnológicos. Aquí detallaremos los recursos que se utilizarán para el desarrollo del proyecto de desarrollo.

TABLA XX:

RECURSOS HUMANOS

Detalle	Cantidad	# horas	V. U.	V. T.
Director del Proyecto	1	0	$ 0,00	0,00
Grupo de Investigación	2	250	$ 5,00	2.500,00
Capacitación de Cursos	1	150	$ 5,00	750,00
			Subtotal	3250,00

- ✓ **Técnicos**

TABLA XXI:

RECURSOS HARDWARE

Detalle	Cantidad	V. U.	V. T
Computador Portátil Vostro XPS 13	2	1549,00	3098,00
Impresora Multifunción Hp Deskjet F4480	1	102,00	102,00
Flash Memory 4 GB Kington	2	25,00	50,00
Componentes de montaje	10	50,00	500,00
Teclado multifunciones	2	100,00	200,00

Monitor dell	1	150,00	150,00
		Subtotal	3892,00

TABLA XXII:

RECURSOS SOFTWARE

Detalle	V. Total
Software de oficina Open Office 3.0	00,00
Web 2.0	00,00
Sistema Operativo Ubuntu 9.10	00,00
Subtotal	00,00

TABLA XXIII:

RECURSOS MATERIALES

Detalle	Cantidad	V. Unitario	V. Total
Resmas papel bon	5	4,00	20,00
Impresiones	500	0,10	50,00
Plumas	4	1,00	4,00
Materiales de Oficina	1	30,00	30,00
Libros	4	60	240,00
Tinta de Cartucho	4	10,00	40,00
Carpetas	4	0,25	1,00
		Subtotal	479,00

TABLA XIV:

DETALLE DE COSTOS

Detalle	V. Total
Recursos Humanos	3250,00

Recursos Materiales	479,00
Recursos Técnicos Hardware	3892,00
Recursos Técnicos Software	0,00
Internet	200,00
Servicios Básicos	250,00
Imprevistos	1000,00
Total	**9071,00**

8.2. Financiamiento.

El total de los gastos que se generarán en el desarrollo del presente proyecto de investigación asciende a nueve mil setenta y un dólares, con cero centavos, será financiado en su totalidad por el grupo de investigación, sin la necesidad de requerir algún tipo de crédito para el efecto.

9. Bibliografía

- ✓ PEREZ, Javier. 2002. Introducción a JavaScript. España [http://www.librosweb.es/javascript/], [Consulta: 23 de Noviembre 2009]
- ✓ CRIVERA, John, 2000. Toolkits [http://www.fismat.umich.mx/~crivera/tesis/node20.html], [Consulta: 09 de Diciembre 2009]
- ✓ MENEZ, Julia, 2006. Dojo Toolkits [http://es.wikipedia.org/wiki/Dojo_toolkit], [Consulta: 13 de Obtubre 2009]
- ✓ DAMIAN, Alexi, 2009. AJAX.

  [http://es.wikipedia.org/wiki/AJAX], [Consulta: 14 de Noviembre 2009]
- ✓ CABRERA Ana Margarita, 2005. Los kioskos electronicos. [http://www.wikilearning.com/monografia/las_kioscos_electrónicos /772] [Consulta: 29 de septiembre 2009]
- ✓ VAN DER HENST, Christian. 2005. ¿Qué es la Web 2.0?. http://www.maestrosdelweb.com/editorial/web2/ Consulta: 29 de septiembre 2009

10. Anexos

10.1. Matriz de Consistencia Específica

10.1.1. Objetivo 1

Problema Específico 1: Escaso análisis de datos para la interpretación de la información y determinar las necesidades en la emisión de certificados académicos del Área de la Energía, las Industrias y los Recursos Naturales no Renovables.			
Objetivo Específico	**Hipótesis Específico**	**Unidad de Observación**	**Sistema Categorial**
Disponer de la información sobre las demandas de certificaciones de los estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.	La realización del análisis preliminar dentro de la ingeniería de requerimientos permitirá establecer los requerimientos funcionales y no funcionales de la herramienta informática.	• Procesos de la Ingeniería de Requerimiento. • Técnicas de Recolección de la información • Formatos de certificados Académicos.	• Partes, definiciones y características de los sistemas expertos • Tabulación de la información • Documento de requerimientos. • Glosario de Términos

10.1.2. Objetivo 2

Problema Específico 2: Desconocimiento en la utilización y aplicación de técnicas para identificar los requerimientos funcionales y el atributo necesario para el desarrollo del kiosko electrónico.			
Objetivo Específico	**Hipótesis Específico**	**Unidad de Observación**	**Sistema Categorial**
Seleccionar el toolkit javascript de generación de tecnologías Web 2.0, que mejor corresponda a las necesidades y alternativas técnicas más convenientes.	Al utilizar un toolkits javascript para la generación de tecnologias Web 2.0 permitirá mejorar los procesos de emisión de certificados académicos para los estudiantes del Área de la Energía, las Industrias y los Recursos Naturales no Renovables.	• Procesos de la Ingeniería de Requerimiento. • Técnicas de Recolección de la información • Toolkits javascript • Formatos de certificados Académicos.	• Tabulación de la información • Documento de requerimientos.

10.1.3. Objetivo 3

Problema Específico 3: Selección incorrecta de la metodología para el desarrollo de herramientas informáticas que vinculen la construcción del hardware y el desarrollo del software apropiado a los requerimientos técnicos del hardware.			
Objetivo Específico	**Hipótesis Específico**	**Unidad de Observación**	**Sistema Categorial**
Establecer las especificaciones de Hardware y Sistema Operativo para la operación dedicada del Kiosko.	La construcción del hardware del Kiosko electrónico y el desarrollo del software ayudarán a eliminar los trámites burocráticos para la emisión de certificados académicos	✓ Hardware ✓ Software empotrado ✓ Formatos de certificados Académicos.	✓ Diagramas de Casos de Uso. ✓ Diagramas de Robustez. ✓ Diagramas de Secuencia. .

10.2. Matriz de Operatividad de Objetivos Específicos

10.2.1. Objetivo 1

Objetivo 1: Disponer de la información sobre las demandas de certificaciones de los estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.						
Actividad o Tarea	Metodología	Fecha		Responsable	Presupuesto	Resultado Esperados
		Inicio	Final			
Investigación Preliminar	Recopilar información acerca de los diferentes toolkits y determinar el más apropiado para el desarrollo de la aplicación. Así mismo realizar Encuestas, entrevistas y observación directa dirigidas a las secretarías de la carreras del AEIRNNR, director del proyecto de	2009-12-10	2009-12-23	Alexandra Jadán Vanessa Loja	$100,00	Recolección y análisis de la información necesaria para determinar necesidades

	SGA, estudiantes del AEIRNNR					
Documentos de los Resultados obtenidos en el análisis preliminar	Clasificación de documentos	2010-01-04	2010-01-15	Alexandra Jadán Vanessa Loja	$50,00	Documentación de la información en formatos establecidos por el equipo de desarrollo
Organización de la información	Tabulación y utilización de los instrumentos de análisis de información	2010-01-16	2010-01-21	Alexandra Jadán Vanessa Loja	$70,00	Identificación de las necesidades en el espacio de estudio
Documento de Requerimientos	Identificar necesidades del sistema	2010-01-22	2010-02-08	Alexandra Jadán Vanessa Loja	$100,00	Realizar el documento de requerimiento funcional
Glosario de Términos	Identificar el vocabulario del sistema	2010-02-09	2010-02-13	Alexandra Jadán Vanessa Loja	$25,00	Dar definición a los conceptos desconocidos en el sistema

10.2.2. Objetivo 2

Objetivo Específico 2: Seleccionar el toolkit Javascript de generación de tecnologías Web 2.0, que mejor corresponda a las necesidades y alternativas técnicas más convenientes						
Actividad o Tarea	**Metodología**	**Fecha**		**Responsable**	**Presupuesto**	**Resultado Esperados**
		Inicio	Final			
Modelo del dominio	Identificar las frases sustantivas	2010-02-15	2010-02-26	Alexandra Jadán Vanessa Loja	$100,00	Realizar el modelo del dominio
Elaboración del diagrama de casos de usos	Identificar Casos de Uso.	2010-03-01	2010-03-31	Alexandra Jadán Vanessa Loja	$50,00	Diagrama de casos d uso
Realización de los prototipados.	Realizar una maqueta de eventos del sistema	2010-04-01	2010-04-15	Alexandra Jadán Vanessa Loja	$1000,00	Establecer las interfaces del sistemas
Narración de los casos de Uso	Diagramar de casos de uso	2010-04-16	2010-05-21	Alexandra Jadán Vanessa Loja	$200,00	Conocer la lógica de negocios.

Elaboración de diagramas de, robustez	Diagramar los diagramas de robustez	2010-05-24	2010-06-10	Alexandra Jadán Vanessa Loja	$100,00	Identificar los objetos del sistemas
Elaboración de diagramas de, secuencia	Diagramar los diagramas de secuencia	2010-06-11	2010-07-30	Alexandra Jadán Vanessa Loja	$100,00	Identificar las acciones de sistemas.
Elaboración del diseño de la Base de Datos	Identificar los entidades relacionales	2010-08-02	2010-08-17	Alexandra Jadán Vanessa Loja	$100,00	Realizar el diagrama y codificación de la base de datos

10.2.3. Objetivo 3

Objetivo Específico 3: Establecer las especificaciones de Hardware y Sistema Operativo para la operación dedicada del Kiosko.						
Actividad o Tarea	**Metodología**	**Fecha**		**Responsable**	**Presupuesto**	**Resultado Esperados**
		Inicio	Final			
Construcción de Hardware	Ensamblar los componentes físicos	2010-08-18	2010-08-24	Alexandra Jadán Vanessa Loja	$1500,00	Construir el kiosco electrónico
Codificación de Clases	Programación delos diagramas	2010-08-25	2010-10-29	Alexandra Jadán Vanessa Loja	$2000,00	Generar código fuente del software
Realizar pruebas de integración	Establecer formatos de identificación de la calidad del Software	2010-11-01	2010-11-24	Alexandra Jadán Vanessa Loja	$100,00	Efectuar integración del software con el hardware
Corrección de errores	Programación de errores	2010-11-25	2010-12-10	Alexandra Jadán Vanessa Loja	$50,00	Depuración de clases

10.2.4. Objetivo 4

Objetivo Específico 4: Contar con un Software Gráfico derivado del toolkit javascript elegido, embebido como proceso dedicado del Sistema Operativo del Kiosko y que interactué con el Sistema de Gestión Académica de la Universidad Nacional de Loja.						
Actividad o Tarea	**Metodología**	**Fecha**		**Responsable**	**Presupuesto**	**Resultado Esperados**
		Inicio	Final			
Documentación de construcción del kiosko	Utilizar un mapas y diagramas de componentes	2010-12-13	2010-12-20	Alexandra Jadán Vanessa Loja	$100,00	Documento de especificaciones técnicas del hardware
Documentación de las clases	Utilizar un API de clases y métodos	2011-01-03	2011-01-15	Alexandra Jadán Vanessa Loja	$100,00	Documentación de clases
lación de la amienta mática	Ejecutar la aplicación	2011-01-16	2011-02-02	Alexandra Jadán Vanessa Loja	$100,00	Utilización de la herramienta Informática.

Anexo 2. Entrevista a Secretarías del Área

ENTREVISTA A SECRETARIAS DEL AREA

1. ¿Qué certificados se emiten en las secretarías a los estudiantes?

Matrículas y Asistencia, Certificado del Mod. I, Certificados de Egresados, Certificados de Pagos

2. ¿Cuántas secretarías existen en el área?

6 Secretarias

2. ¿Qué procedimiento debe realizar el alumno para obtener un certificado?

Hacer cabilas

3. ¿Cuál es el procedimiento de la secretaria para emitir un certificado?

Basarse en los archivos

4. ¿De dónde extrae la información, para la elaboración de los certificados?

Basarse en los archivos

6. ¿Quién autoriza el certificado?

La secretaria Encargada

7. ¿En que tiempo se entregan los certificados a los alumnos?

En el mismo día

8. ¿Con que frecuencia se emiten los certificados?

En forma regular

9. ¿Que certificados son los mas solicitados por los estudiantes?

Matricula y Asistencia, Certificados Mod. I (Pensum)

10. ¿En qué horario, los alumnos pueden solicitar los certificados?

8h a 12H y de 15H00 a 19 H00

Anexo 3. Cuestionario para estudiantes para determinar las demandas de certificados

CUESTIONARIO PARA ESTUDIANTES

1. ¿Qué tipo de certificados ha solicitado Ud.?

Certificado de asistencia
Certificado de calificaciones
Certificado de aprobación de materias

2. ¿Cuáles son los requisitos para obtener un certificado?

Una solicitud

3. ¿Qué procedimientos se siguen para obtener un certificado?

Presentar la solicitud en la ventanilla general

4. ¿En qué tiempo le entregan el certificado?

Entre 12 o 24 horas (solo en la tarde)

5. ¿Qué criterio le merece la emisión de certificados en las secretarias?

Una atención buena

6. ¿En qué horario puede Ud. Solicitar un certificado?

en cualquier momento pero atienden solo por las tardes

7. ¿A utilizado Ud. Un cajero automático, que le parece la experiencia?

Si muy cómodo y rápido

8. ¿Existe algún inconveniente en la emisión de los certificados solicitados por parte de las secretarias? Explique cual(es).

No

Anexo 4. Resultados de cuestionarios a estudiantes para determinar las demandas de certificados.

1. ¿Qué tipo de certificado ha solicitado Ud.?

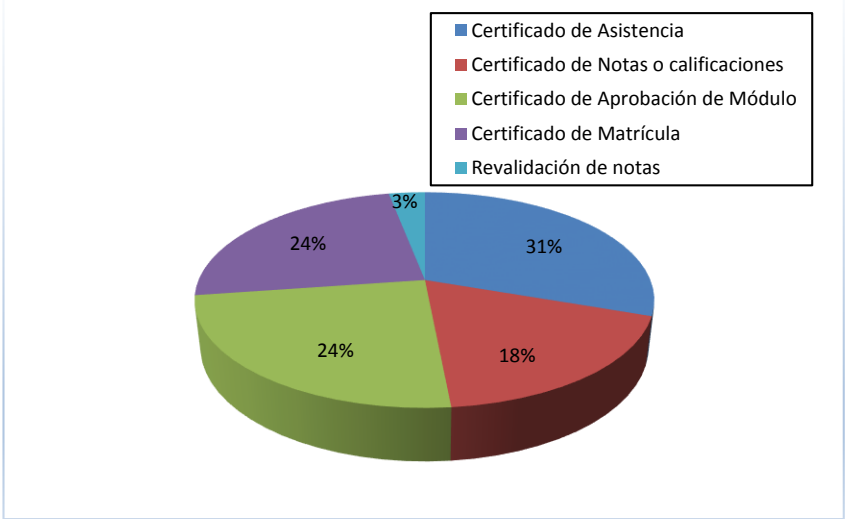

Figura 61. Representación gráfica de resultados de pregunta 1.

TABLA XVIII:

TABULACIÓN RESULTADOS PREGUNTA 1

CERTIFICADOS			
No.	**VARIABLES**	**No Encuestados**	**%**
1	Certificado de Asistencia	10	31
2	Certificado de Notas o calificaciones	6	18
3	Certificado de Aprobación de Módulo	8	24
4	Certificado de Matrícula	8	24
5	Revalidación de notas	1	3
		TOTAL	**100%**

2. ¿Cuáles son los requisitos para obtener un certificado?

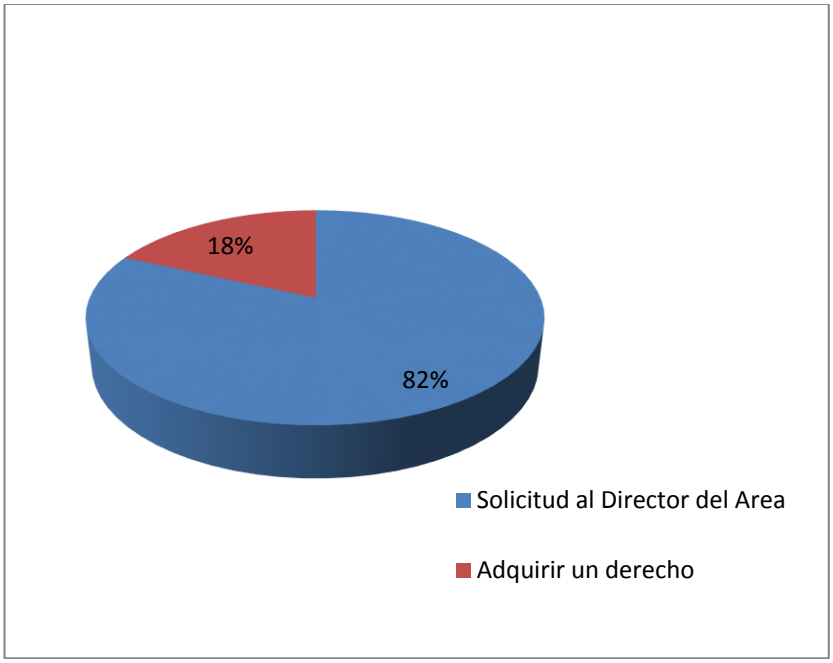

Figura 62. Representación pregunta 2.

TABLA XIX:

TABULACIÓN PREGUNTA 2

No.	VARIABLES	No Encuestados	%
1	Realizar una solicitud al Director del Área	18	82
2	Adquirir un derecho	4	18
		TOTAL	**100%**

3. ¿Qué procedimientos se siguen para obtener un certificado?

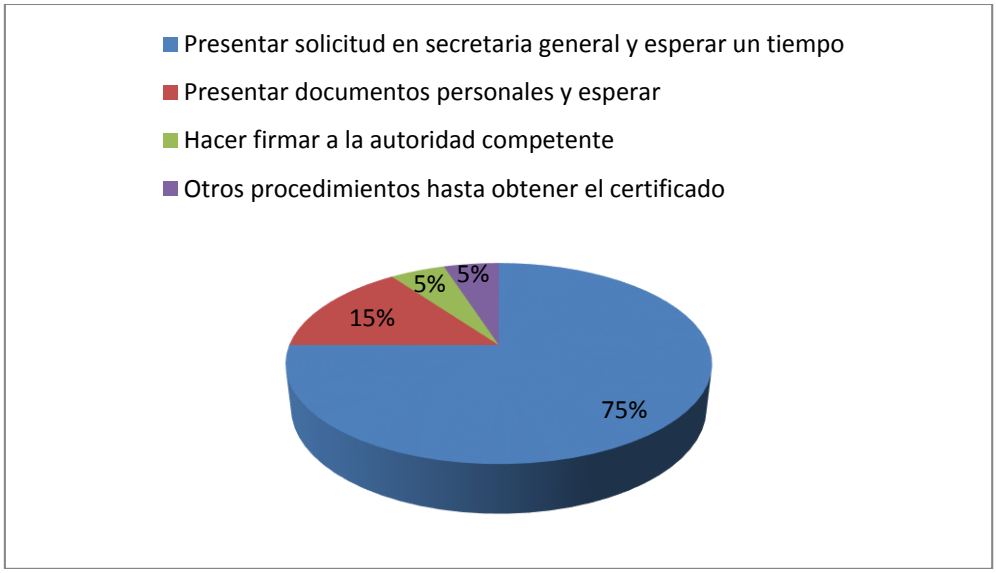

Figura 63. Representación gráfica de resultados de pregunta 3

TABLA XX:

TABULACIÓN RESULTADOS PREGUNTA 3

No.	**VARIABLES**	**No Encuestados**	**%**
1	Presentar solicitud en secretaria general y esperar un tiempo	15	75
2	Presentar documentos personales y esperar	3	15
3	Hacer firmar a la autoridad competente	1	5
4	Otros procedimientos hasta obtener el certificado	1	5
		TOTAL	**100%**

4. ¿En qué tiempo le entregan el certificado?

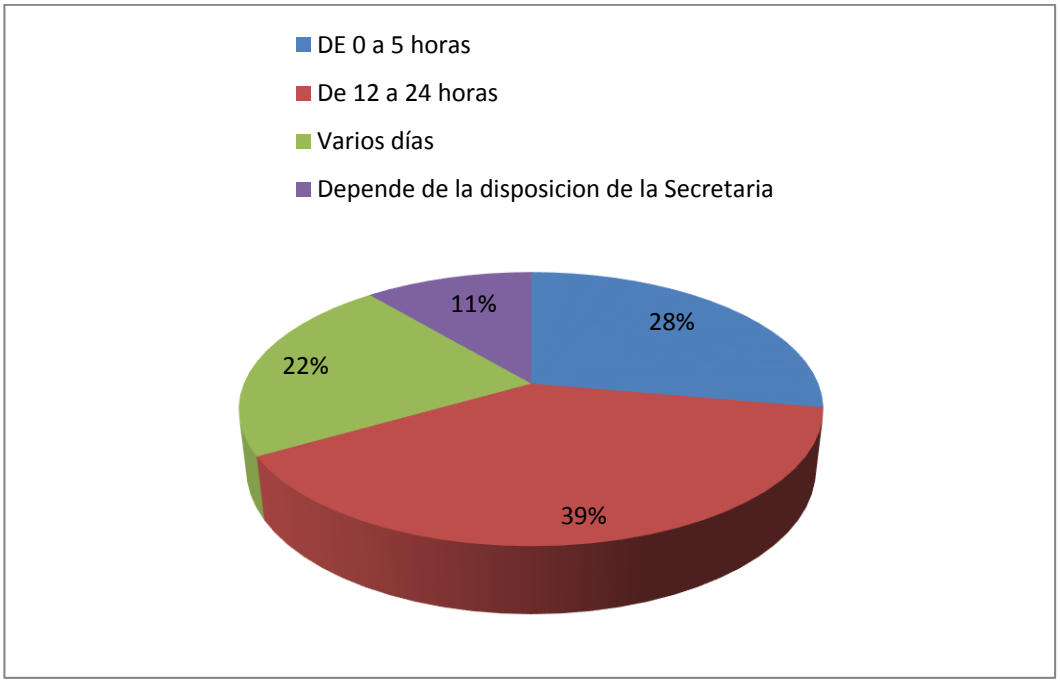

Figura 64. Representación gráfica de resultados de pregunta 4.

TABLA XXI:

TABULACIÓN RESULTADOS PREGUNTA 4

No.	**VARIABLES**	**No Encuestados**	**%**
1	DE 0 a 5 horas	5	28
2	De 12 a 24 horas	7	39
3	Varios días	4	22
4	Depende de la disposición de la Secretaria	2	11
		TOTAL	**100%**

5. ¿Qué criterio le merece la emisión de certificados en las secretarías?

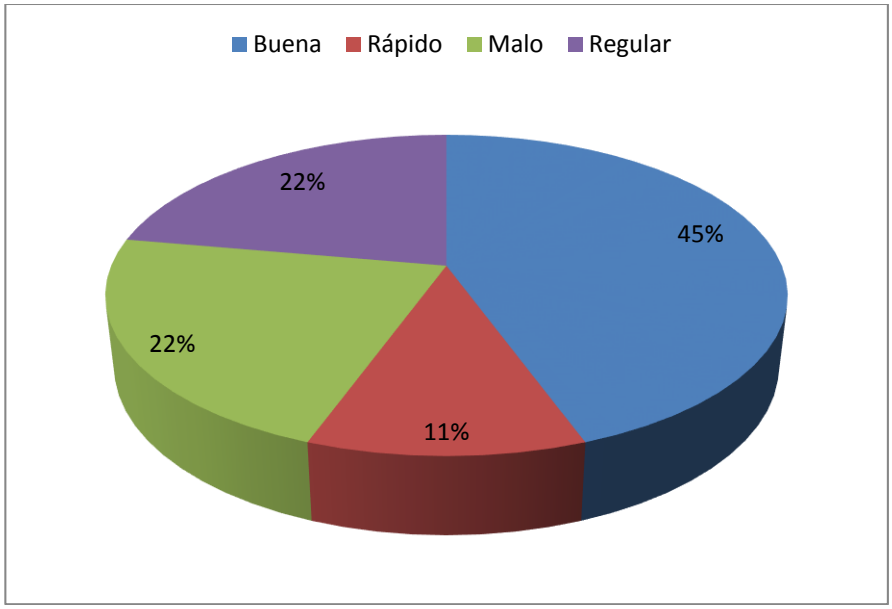

Figura 65. Representación gráfica de resultados de pregunta 5.

TABLA XXII: TABULACIÓN RESULTADOS PREGUNTA 5

No.	VARIABLES	No Encuestados	%
1	Buena	8	45
2	Rápido	2	11
3	Malo	4	22
4	Regular	4	22
		TOTAL	**100 %**

6. ¿En qué horario puede Ud. solicitar un certificado?

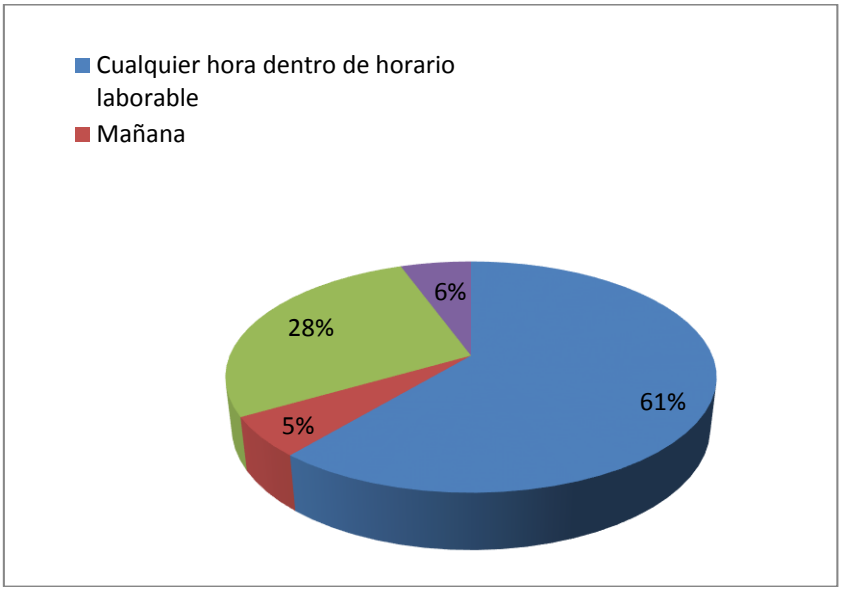

Figura 66. Representación gráfica de resultados de pregunta 6.

TABLA XXIII:

TABULACIÓN RESULTADOS PREGUNTA 6

No.	**VARIABLES**	**No Encuestados**	**%**
1	Cualquier hora dentro de horario laborable	11	61
2	Mañana	1	5
3	Tarde	5	28
4	Cuando la secretaria este libre	1	6
		TOTAL	**100%**

7. ¿Ha utilizado Ud. un cajero automático, que le parece la experiencia?

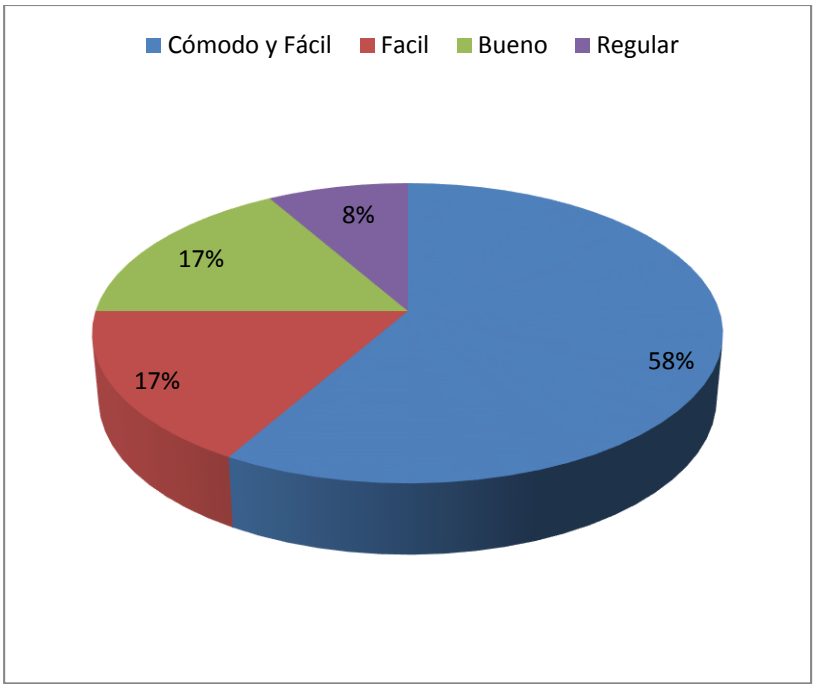

Figura 67. Representación gráfica de resultados de pregunta 7

TABLA XXIV:

TABULACIÓN RESULTADOS PREGUNTA 7

No.	**VARIABLES**	**No Encuestados**	**%**
1	Cómodo y Fácil	7	58
2	Fácil	2	17
3	Bueno	2	17
4	Regular	1	8
		TOTAL	**100%**

8. ¿Existe algún inconveniente en la emisión de los certificados solicitados por parte de las secretarías?

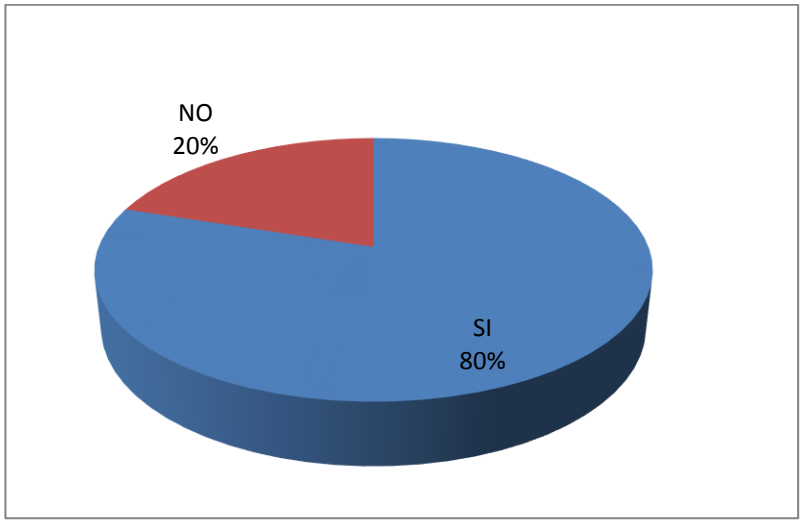

Figura 68. Representación gráfica de resultados de pregunta 8.

TABLA XXV:

TABULACIÓN RESULTADOS PREGUNTA 8

No.	**VARIABLES**	**No Encuestados**	**%**
1	SI	32	80
2	NO	8	20
	TOTAL	**40**	**100%**

Anexo 5. Fase de Construcción física del kiosko

Figura 69. Montaje de madera y hierro parte frontal

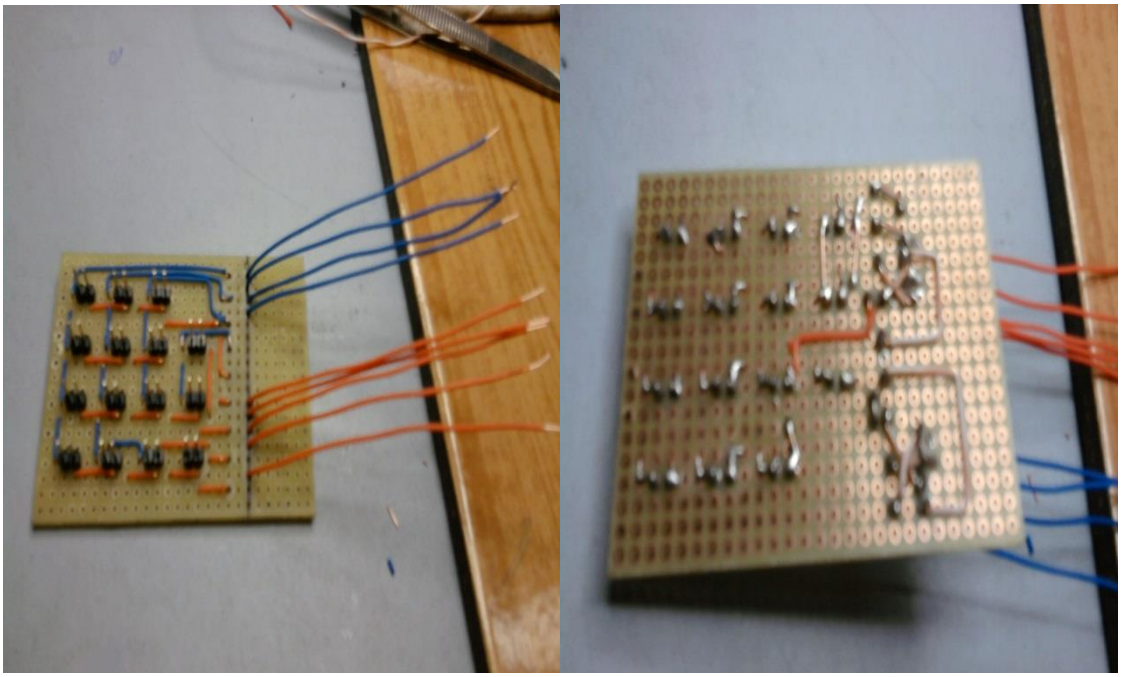

Figura 70. Construcción del circuito del teclado en la PCB

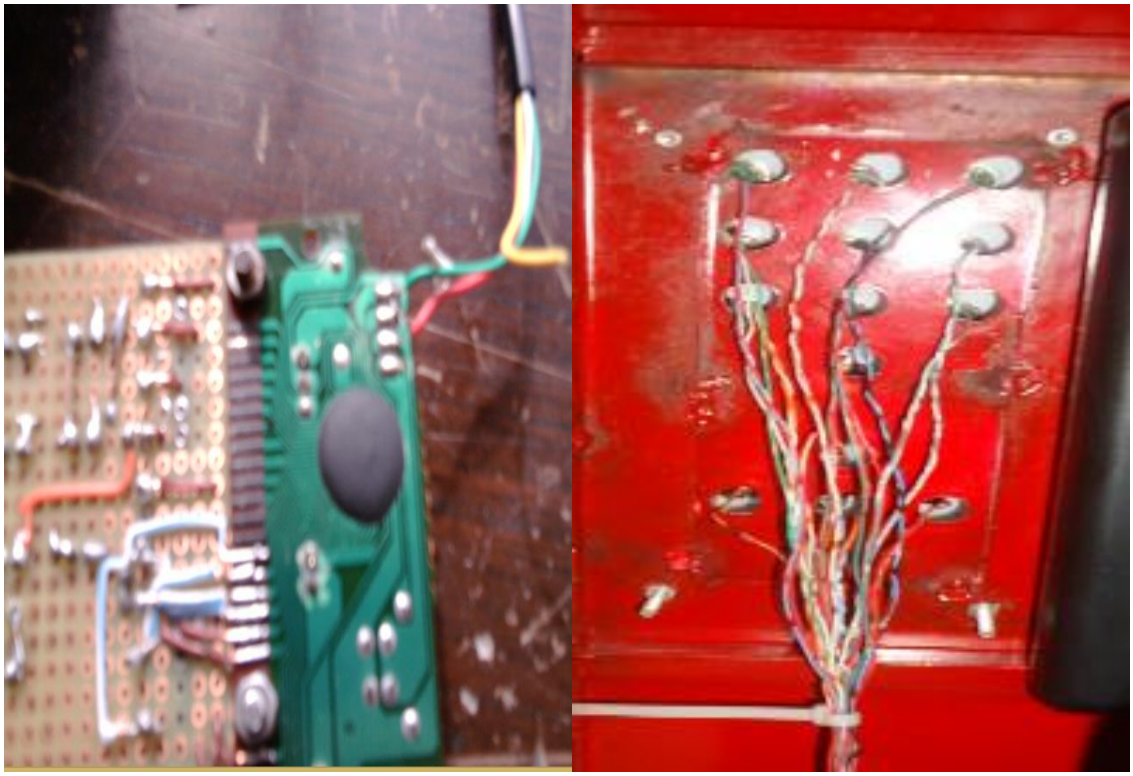

Figura 71. Conexión del teclado a los pulsantes

Figura 72. Montaje del teclado en la parte frontal

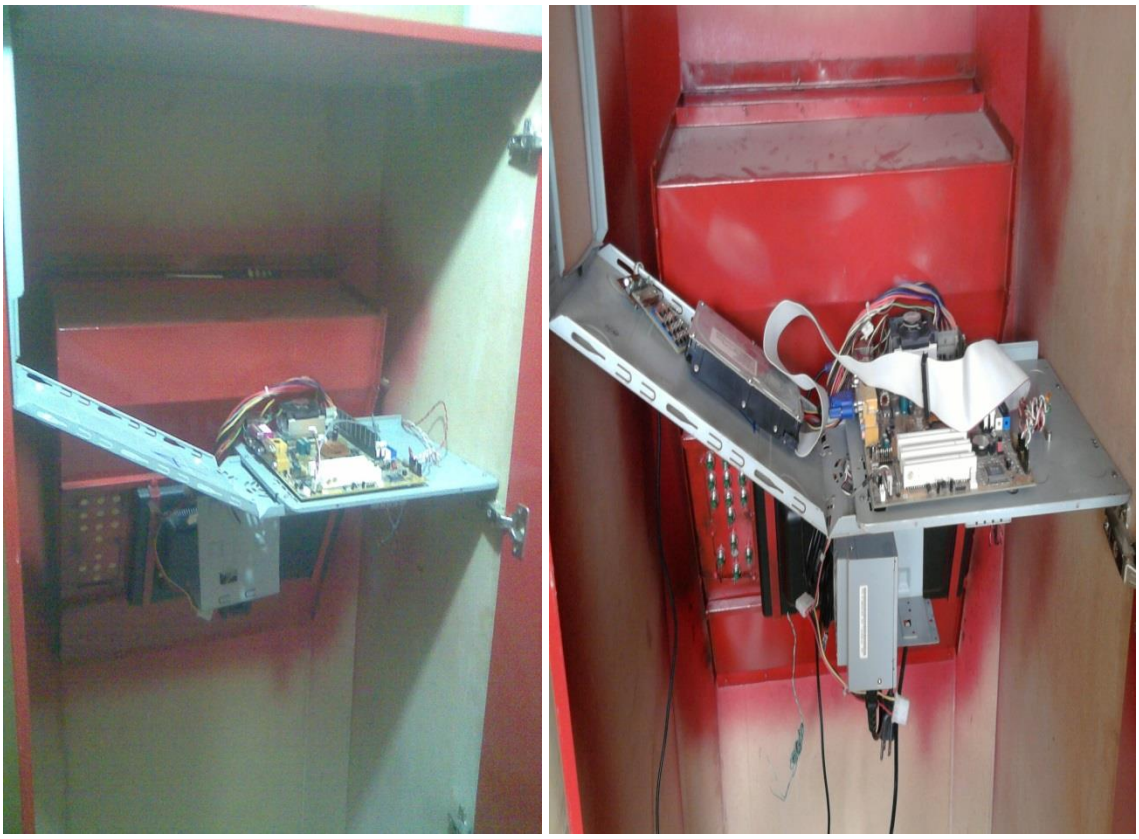

Figura 73. Montaje del hardware interno del kiosko

Figura 74. Arrancando con Linux Debian

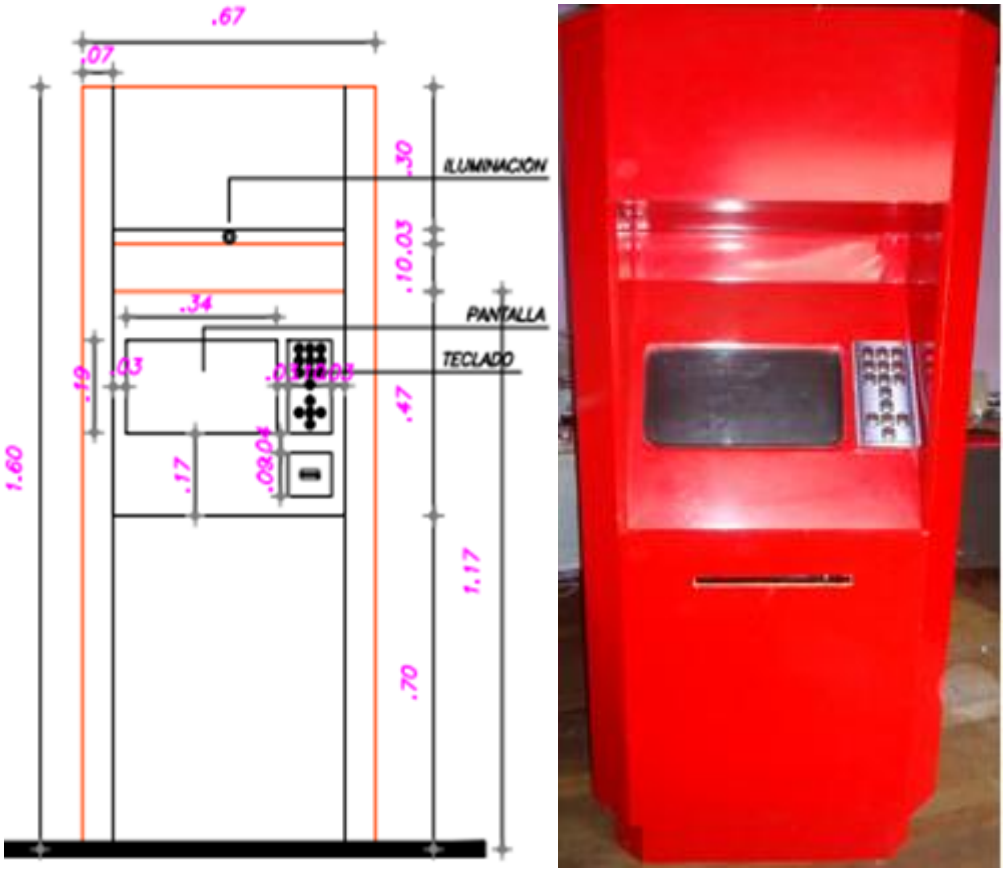

Figura 75. Proyección Inicial VS Proyección final

Anexo 6. Software gráfico del kiosko

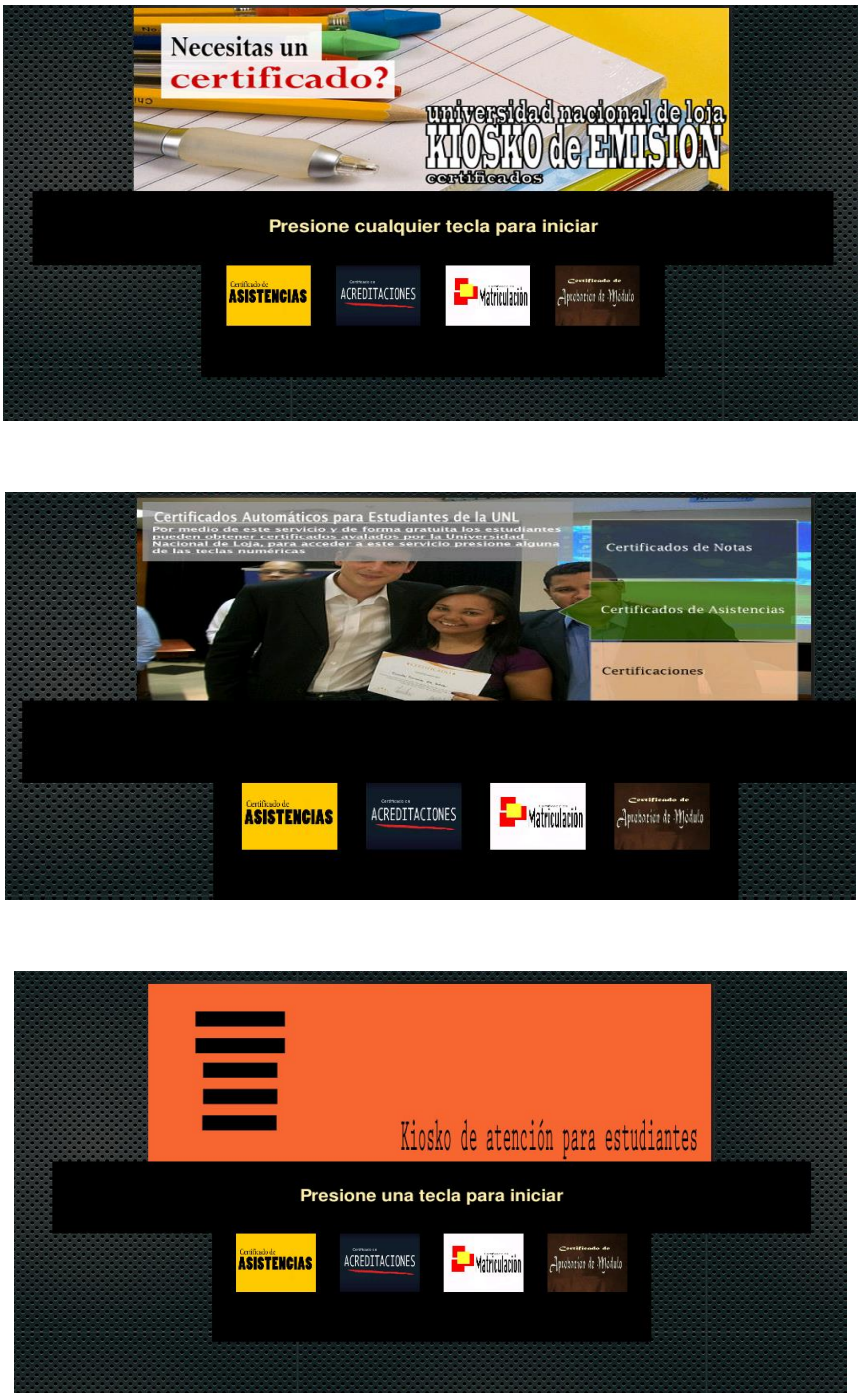

Figura 76. Pantallas del splash de bienvenida

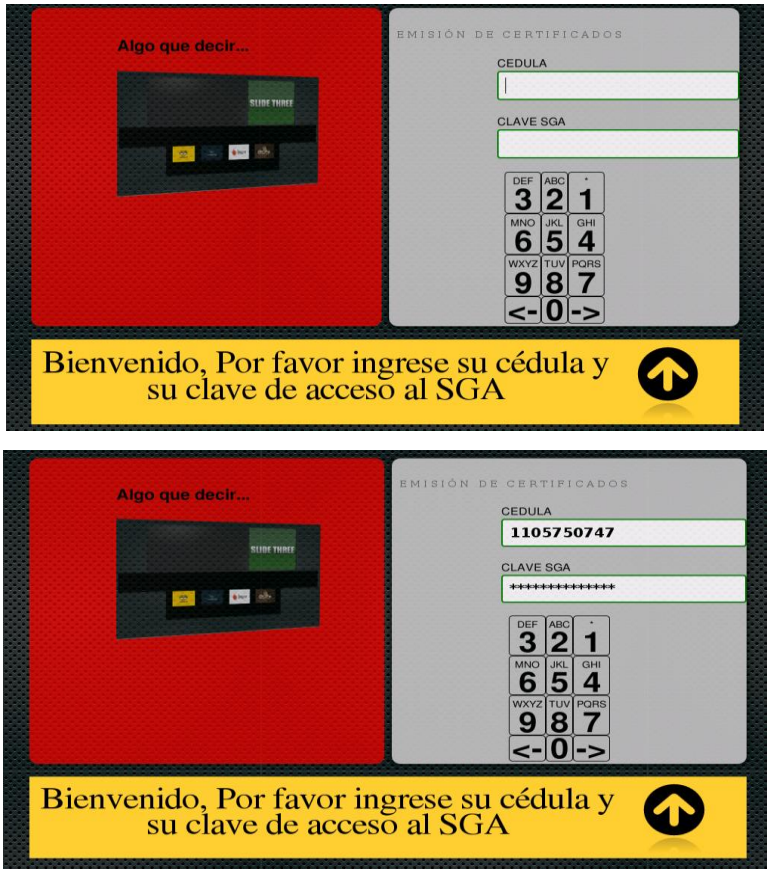

Figura 77. Pantalla Ingreso de datos del estudiante

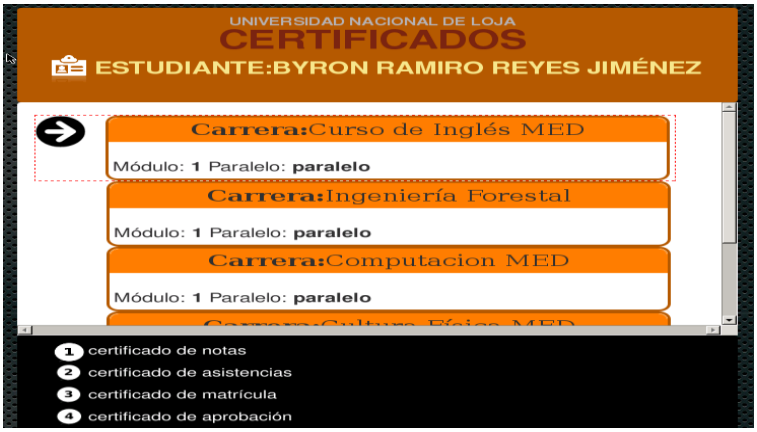

Figura 78. Pantalla de selección de carreras y selección de certificados

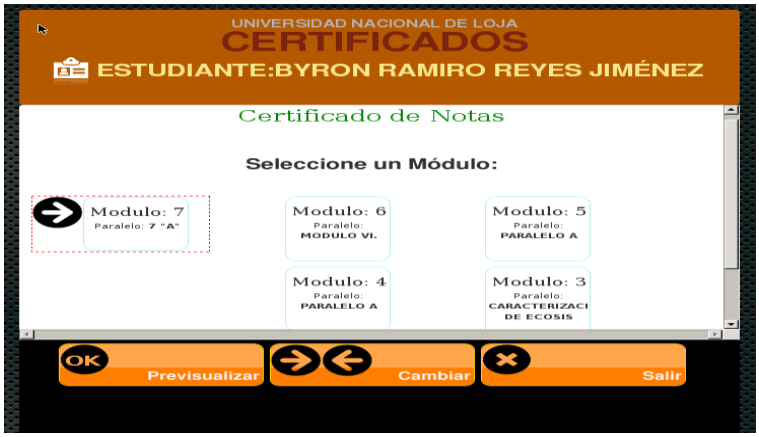

Figura 79. Pantalla selección del Módulo (notas)

Figura 80. Pantalla Vista previa - notas

Figura 81. **Pantalla selección del módulo (asistencias)**

Figura 82. Pantalla vista previa- Asistencias

Figura 83. Pantalla selección del módulo (Matrícula)

Figura 84. Pantalla vista previa- Matrícula

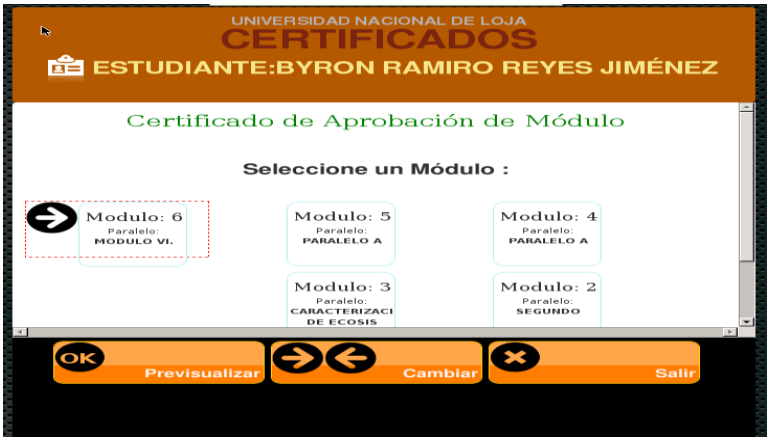

Figura 85. Pantalla selección del Módulo Aprobación del Módulo

Figura 86. Pantalla vista previa- Aprobación de Módulo

Figura 87. Pantalla final de emisión de certificados

Anexo 7. Pruebas de usabilidad

Pruebas de usabilidad del kiosko

Estimado estudiante, la finalidad del presente cuestionario es medir la iteración que tiene con el sistema kiosko de emisión de certificados académicos y de esta manera comprobar la usabilidad y funcionalidad del software. Por lo cual, solicitamos se nos brinde un poco de su tiempo para usar el kiosko y luego responder a las siguientes preguntas:

1. ¿Cómo considera usted el nivel de usabilidad del kiosko electrónico para la emisión de certificados?

Fácil (✓)

Media ( )

Difícil ( )

2. La iteración que tuvo con la interfaz del software. ¿Fue?

Agradable (✓)

Poco Agradable ( )

Nada agradable ( )

3. ¿Cómo considera Usted el nivel de rendimiento del kiosko?

Rápido (✓)

Normal ( )

Lento ( )

4. ¿Los menús para la emisión de los certificados indican claramente las opciones deseadas?

Si (✓) No ( )

5. ¿Los formatos de los certificados generados por el kiosko, cumplen sus expectativas?

Certificado de Notas	Si (✓)	No ( )
Certificado de Matrícula	Si (✓)	No ( )
Certificado de Asistencia	Si (✓)	No ( )
Certificado de Aprobación de módulo	Si (✓)	No ( )

6. ¿La aplicación contiene colores coherentes y agradables en todos sus menús?

Si (✓) No ( )

7. ¿Considera usted que el kiosko electrónico para la emisión de certificados es una herramienta innovadora en éste tipo de procesos?

Si (✓) No ( )

8. ¿Cree usted que se debería implementar el uso de kioskos en los diferentes trámites que realizan los estudiantes en el Área?

Si (✓) No ( )

Firma:

Gracias por su colaboración

Anexo 8. Formato de certificados generados en pruebas de usabilidad

a. Certificado de notas

Universidad Nacional de Loja
AREA DE ENERGÍA, INDUSTRIA Y RECURSOS NATURALES NO RENOVABLES

Universidad Nacional de Loja, AREA DE ENERGÍA, INDUSTRIA Y RECURSOS NATURALES NO RENOVABLES

CERTIFICA

29 de enero de 2015

Luego de haber revisado los registros del SGA,

Dr. Estuardo Napoleón Rodríguez Ludeña
SECRETARIO ABOGADO DE AREA DE ENERGÍA, INDUSTRIA Y RECURSOS NATURALES NO RENOVABLES

Que el señor bachiller: **JOSE LUIS AGILA HIDALGO** de nacionalidad *ecuatoriana*, con cédula Nro. **1104909443**, en la matrícula del **1 Módulo** denominado: "PROBLEMÁTICAS GLOBALES DE LA REALIDAD SOCIAL", **Modalidad Presencial**, que corresponde al **I AÑO** de la carrera de: **Ingeniería en Electromecánica**, con la matrícula **Nro. 213102** Folio **Nro.115** del libro de matrículas, período académico **Pregrado Septiembre 2011 Febrero 2012** ha obtenido las siguientes calificaciones:

- MATEMATICAS 8.3
- FISICA 8.2
- PROCESO INVESTIGATIVO 7.9
- QUIMICA 7.8
- GEOMETRIA PLANA 8.6

Dr. Estuardo Napoleón Rodríguez
Secretario Abogado

b. Certificado de asistencias

AREA AGROPECUARIA Y DE RECURSOS NATURALES RENOVABLES

Universidad Nacional de Loja - Burrologia, AREA AGROPECUARIA Y DE RECURSOS NATURALES RENOVABLES

CERTIFICA

30 de enero de 2015

Luego de haber revisado los registros del SGA,

Dr. Vicente Analuisa
SECRETARIO ABOGADO DE AREA AGROPECUARIA Y DE RECURSOS NATURALES RENOVABLES

Que el señor bachiller: **BYRON RAMIRO REYES JIMéNEZ** de nacionalidad ecuatoriana, con cédula Nro. **1105750747**, en la matrícula del **7 Módulo** denominado: **'APROVECHAMIENTO DE LA MADERA.'**, **Modalidad Presencial**, que corresponde al **IV AÑO** de la carrera de: **Ingeniería Forestal**, con la matrícula **Nro. 351393** Folio **Nro.172** del libro de matrículas, período académico **Pregrado Septiembre 2014 - Febrero 2015** ha registrado un total de **123** horas asistidas.

Dr. Vicente Analuisa
Secretario Abogado

c. Certificado de Matrícula

Universidad Nacional de Loja
AREA DE ENERGÍA, INDUSTRIA Y RECURSOS NATURALES NO RENOVABLES

Universidad Nacional de Loja, AREA DE ENERGÍA, INDUSTRIA Y RECURSOS NATURALES NO RENOVABLES

CERTIFICA

29 de enero de 2015

Luego de haber revisado los registros del SGA,

Dr. Estuardo Napoleón Rodríguez Ludeña
SECRETARIO ABOGADO DE AREA DE ENERGÍA, INDUSTRIA Y RECURSOS NATURALES NO RENOVABLES

Que el señor bachiller: **JOSE LUIS AGILA HIDALGO** de nacionalidad ecuatoriana, con cédula Nro. **1104909443**, se encuentra matriculado en el **2 Módulo** denominado: **'PROBLEMÁTICAS DE LA REALIDAD DEL ÁREA, SUS REFERENTES TEÓRICOS Y SU INCIDENCIA EN LA FORMACIÓN PROFESIONAL'**, **Modalidad Presencial**, que corresponde al **1 AÑO** de la carrera de: **Ingeniería en Electromecánica**, con el **Nro. 354771** Folio **Nro.308** del libro de matrículas, período académico **Pregrado Septiembre 2014 - Febrero 2015**.

Dr. Estuardo Napoleón Rodríguez Ludeña
Secretario Abogado

d. Certificado de Aprobación de Módulo

Universidad Nacional de Loja
AREA DE ENERGÍA, INDUSTRIA Y RECURSOS NATURALES NO RENOVABLES

Universidad Nacional de Loja, AREA DE ENERGÍA, INDUSTRIA Y RECURSOS NATURALES NO RENOVABLES

CERTIFICA

29 de enero de 2015

Luego de haber revisado los registros del SGA,

Dr. Estuardo Napoleón Rodríguez Ludeña
SECRETARIO ABOGADO DE AREA DE ENERGÍA, INDUSTRIA Y RECURSOS NATURALES NO RENOVABLES

Que el señor bachiller: **JOSE LUIS AGILA HIDALGO** de nacionalidad ecuatoriana, con cédula Nro. **1104909443**, APROBÓ el **1 Módulo** denominado: **'PROBLEMÁTIC GLOBALES DE LA REALIDAD SOCIAL'**, **Modalidad Presencial**, que corresponde al **I AÑO** de la carrera de: **Ingeniería en Electromecánica**

Dr. Estuardo Napoleón Rodríguez Ludeña
Secretario Abogado

Anexo 9. Artículo Científico

"Diseño y Construcción de un Kiosko Electrónico Prototipado para la Impresión Automática de Certificados Académicos para los Estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables",

Alexandra Esperanza Jadán Patiño, Mayra Vanessa Loja Sandoya

Carrera de Ingeniería en Sistemas, Universidad Nacional de Loja

Loja, Ecuador

vanel26@hotmail.com

alexaejp2080@hotmail.com

RESUMEN: El presente trabajo de titulación, tiene como objetivo principal crear una herramienta informática que permita mejorar los servicios de emisión de certificados académicos capaz de satisfacer los requerimientos de los estudiantes del AEIRNNR.

Para el desarrollo de la herramienta hemos empleado la metodología ICONIX, en la fase de recolección de requerimientos se utiliza la investigación de campo, acudiendo a la dirección de Informática y telecomunicaciones de la Universidad para obtener toda la información relacionada con los archivos log almacenados en el servidor web de la Universidad Nacional de Loja, así como a la AEIRNNR a las 4 secretarías(entrevistas) y una muestra de 20 estudiantes(encuestas) quienes despachan y solicitan respectivamente los certificados académicos. Con esta información se establecen los requerimientos funcionales, no funcionales, el glosario de términos y el modelo del dominio de la aplicación, aplicando los métodos inductivo y deductivo para luego diseñar los diagramas de caso de uso para el usuario y el administrador en base a los cuales se redactaron los mismos indicando el flujo normal y alterno de los eventos entre el sistema y los actores. Para la codificación del diseño de la aplicación se utiliza NetBeans 6.5 con el Framework **JSF** el cual está basado en el patrón MVC (Modelo-Vista-Controlador) que sirve para crear aplicaciones basadas en la web.

Con la implementación del **Kiosko Electrónico Prototipado para la Impresión Automática de Certificados Académicos** se mejorará los servicios de emisión de certificados académicos capaz de

satisfacer los requerimientos de los estudiantes del Área.

PALABRAS CLAVES: ***Kiosko electrónico, Sistema de Gestión Académico.***

Abstract

This paper titration has as main objective to create a software tool to improve services issuance of transcripts capable of meeting the requirements of students AEIRNNR.

For the development of the tool we can used the ICONIX methodology at the stage of requirements gathering field research is used, going to the direction of Informatics and Telecommunications of the University for all information related to the log files stored in the web server of the National University of Loja, as well as the AEIRNNR at 4 secretariats (interviews) and a sample of 20 students (surveys) who dispense and apply academic certificates respectively. This information establishes the functional requirements, non-functional, glossary of terms and model of the application domain, using inductive and deductive methods then design the use case diagrams for the user and administrator based on the indicating which like normal and alternate events between the system and the actors were drafted flow. For coding the application design using NetBeans 6.5 with JSF Framework which is based on the MVC pattern (Model-View-Controller) used to create web-based applications.

With the implementation of the Electronic Kiosk Prototyping for Automatic Printing services Academic Certificates issuance of transcripts able to meet the needs of students in the area will be improved.

KEYWORDS: Electronic Kiosk, Academic Management System.

I. INTRODUCCIÓN

La propuesta de un kiosko de información e impresión de documentos que es un aparato electrónico conjuntamente con un aplicativo que interactúa con los usuarios a través de una pantalla y teclado tipo push, es una solución que ha dejado de ser solamente para el sector transaccional bancario y se ha expandido a todos los sectores del mercado, ya sea para consultas de información o bien automatizando procesos de pago mediante el autoservicio.

Por ello hemos considerado conveniente el diseño y construcción de un kiosko electrónico prototipado, en este caso para la emisión de certificados académicos de los estudiantes de la Universidad Nacional de Loja a través del Web Service del Sistema de Gestión.

II. DESARROLLO DE CONTENIDOS

Para el desarrollo de nuestro proyecto se ha recurrido a la investigación científica

destacando los siguientes conceptos como principales.

A. kiosko Electrónico: Los Kioskos Electrónicos son módulos de autoservicio que funcionan como una ventanilla única automatizada donde se ofrecen múltiples servicios y que están situados en lugares estratégicos con el fin de acercarlos al usuario.

B. ***Sistema de Gestión Académico:*** Es un conjunto de soluciones de software orientadas a la gestión académica de la Universidad Nacional de Loja. Este sistema consta de una gran variedad de módulos para tareas cotidianas tomando en cuenta roles y escenarios.

C. Certificado académico: Son documentos académicos legalizados por las autoridades universitarias que muestran la información requerida por el estudiante y que son elaborados de acuerdo al expediente académico del mismo. El Kiosko emitirá cuatro tipos de certificado que son:

- Certificado de notas
- Certificado de asistencia
- Certificado de matrícula
- Certificado de aprobación de módulo

III. RESULTADOS

Realizado el análisis minucioso de las tecnologías más apropiadas para la construcción del kiosko, hemos optado por la siguiente estructura donde se incluye el montaje de los equipos electrónicos y la aplicación que interactúa con los usuarios a través de una pantalla y un teclado tipo push que se detalla en la siguiente figura:

Figura 1. Estructura del kiosko

En el análisis preliminar dentro de la ingeniería de requerimientos, y análisis metodológico (encuestas) nos ha permitido establecer los certificados más solicitados por los estudiantes:

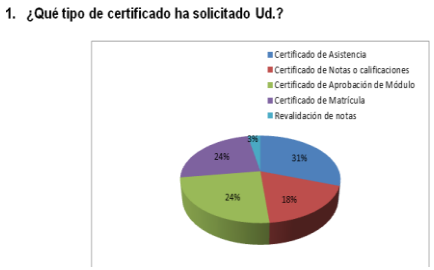

Figura2. Certificados académicos más solicitados.

De acuerdo a la información anterior se construye la aplicación que permita al estudiante consultar los datos en el kiosko para posteriormente imprimir. A

continuación se detalla el proceso para la obtención del certificado de notas:

1) El kiosko muestra el splash de bienvenida al estudiante y sugiere presionar cualquier tecla para iniciar el proceso.

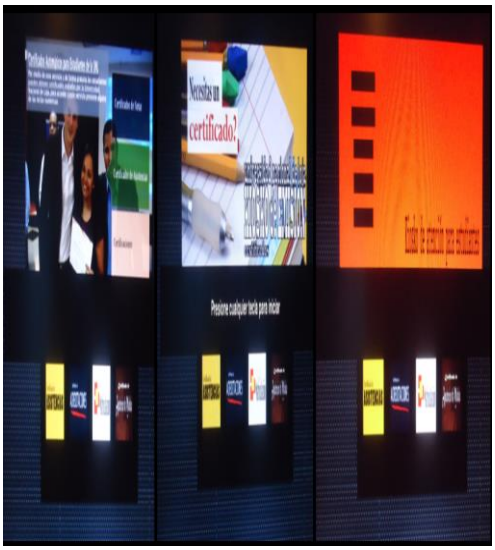

Figura 3. Splash de bienvenida

2) El estudiante tiene que ingresar los datos para pasar al menú de elección del certificados de la carrera elegida y posteriormente elegir el módulo como se muestra a continuación:

Figura 4. Ingreso de datos (logueo)

Figura5. Elección de la carrera

Figura 6. Elección del módulo

Una vez seleccionada la carrera y el módulo, el sistema muestra la vista previa del certificado, como se muestra a continuación:

Figura 7. Vista previa del documento.

Si el estudiante desea imprimir, entonces muestra la pantalla Imprimiendo, caso contrario el sistema vuelve a la pantalla de bienvenida, como se muestra en la siguiente figura:

Figura 8. Vista de impresión del documento

Una vez impreso el documento, se agradece al usuario y vuelve al splash de bienvenida.

Cabe mencionar que el proceso para el certificado de matrícula, certificado de aprobación de módulo y asistencias es el mismo.

De acuerdo al procedimiento descrito anteriormente podemos concluir que los requisitos y tiempo para la obtencion por parte del estudiante son minimos ya que se descarta el paso de realizar una solicitud, y espera para el despacho del certificado por parte de la secretaria de carrera, debido a que el estudiante obtiene el documento en el kiosko en cuestion de segundos. Además la poblacion de encuestados se muestra optimista de hacer uso del kiosko en la obtención de sus certificados pues consideran el uso de los cajeros automáticos como un proceso cómodo y fácil.

IV. CONCLUSIONES

Al finalizar el presente proyecto se ha llegado a las siguientes conclusiones:

- Gracias a la aplicación de técnicas y métodos de investigación se logró disponer de la información sobre las demandas de certificaciones de los estudiantes del Área de la Energía, las Industrias y los Recursos Naturales no Renovables, lo que permitió obtener los requerimientos del sistema necesarios con el fin de obtener una herramienta eficiente y acorde a las necesidades de los estudiantes.

- Se eligió el toolkits JQuery, por ser una librería muy flexible, rápida y efectiva, la misma que permitió el desarrollo de la herramienta con excelente integración de Ajax, obteniendo de esta concatenación un alto nivel de renderización del sistema.

- Se logró el diseño de una aplicación dedicada en Línux con la distro Debian, para interactuar con los usuarios mediante un procedimiento gráfico de selección e interacción con el Sistema de Gestión Académica de la Universidad Nacional de Loja, a través de un teclado tipo push y una pantalla, brindando seguridad y eficiencia en la obtención de los diferentes certificados académicos.
- Con la implementación de este proyecto de investigación se logró demostrar que el prototipo de Kiosko diseñado y construido, funciona de manera eficaz, eficiente, entrega un servicio de calidad, aporta al proceso de

mejora, calificación y engrandecimiento de nuestra Universidad.

- Se logró comprobar que el diseño y construcción del kiosko se constituye en una herramienta sólida y eficiente para la emisión de certificados académicos en la Universidad Nacional de Loja en todas sus áreas.

IV. BIBLIOGRAFÍA

[1] BAEZA-YATES, Ricardo. RIBEIRO-NETO, Berthier. (2000), Modern Information Retieval, Addison-Wesley.

[2] COOLEY, Robert. TAN, Pang-Ning. SRIVASTAVA, Jaideep. "Discovery of interesting usage patterns from web data".

[3] GUTIERREZ, M.A. Métodos y Técnicas de Investigación, Texto guía para el estudiante, Quito, Editorial Época, 1984.

[4] Lineamientos para el desarrollo de las investigaciones de tesis de grado en la Carrera de Ingeniería en Sistemas, Universidad Nacional de Loja, 2014.

Sitios web:

[5]PEREZ, Javier. 2002. Introducción a JavaScript. España [En Línea][Consulta: 23 de Noviembre 2013]. Disponible en [http://www.librosweb.es/javascript/]

[6] DAMIAN, Alexi, 2009. AJAX.[En Línea]. [Consulta: 14 de Noviembre 2013]. Disponible en: <[http://es.wikipedia.org/wiki/AJAX]

[7]Iván Tueva,[En Línea]. [Consulta: febrero 2013].Disponible en: http://ocw.bib.upct.es/pluginfile.php/7990/mod_resource/content/1/Construccion_pcb.pdf

Alexandra Jadán Patiño**,** egresada de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, conocedora en análisis y desarrollo de software con Java, mantenimiento y reparación de computadoras, redes y telecomunicaciones. Provincia de Loja, Ciudad Loja, 2015

Vanessa Loja Sandoya, egresada de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, conocedora en análisis y desarrollo de software en Java, mantenimiento e instalación de redes de computadores. Provincia de Loja, Ciudad Loja, 2015.

Juan Manuel Galindo Vera, Ingeniero en Redes de comunicación y Sistemas de conmutación, en la Universidad Belgord State University, Docente especialista. Analista en Telecomunicaciones en la Universidad Nacional de Loja, Provincia de Loja, Ciudad Loja, Ecuador, 2015.

Anexo 9. Certificado de traducción del resumen de memoria y artículo

LOJA

CIUDAD.

Loja, 10 de diciembre de 2014

CERTIFICADO

Yo **JOBITA LILIA SANDOYA JUMBO**, Licenciada en Ciencias de la Educación, Mención Idioma Inglés con cédula N° 1101932141, certifico que se ha realizado la traducción del resumen del Trabajo de Titulación cuya problemática versa sobre el tema: **"DISEÑO Y CONSTRUCCION DE UN KIOSKO ELECTRONICO PROTOTIPADO PARA LA IMPRESIÓN AUTOMATICA DE CERTIFICADOS ACADEMICOS PARA LOS ESTUDIANTES DEL AREA DE LA ENERGIA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES."**, realizado por las egresadas **ALEXANDRA ESPERANZA JADAN PATIÑO** y **MAYRA VANESSA LOJA SANDOYA**, previa a la obtención del título de Ingeniero en Sistemas.

Es todo cuanto puedo informar sobre el particular y autorizo a las interesadas hacer uso del presente certificado de la manera que estimen conveniente.

Atentamente,

JOBITA LILIA SANDOYA JUMBO

Anexo 10. Licencia

Kiosko Electrónico Prototipado para la impresión automática de certificados académicos para los estudiantes del Área de la Energía, las Industrias y los Recursos Naturales no Renobles por Alexandra Esperanza Jadán Patiño, Mayra Vanessa Loja Sandoya se distribuye bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.