



UNIVERSIDAD NACIONAL DE LOJA

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

CARRERA DE INGENIERÍA EN SISTEMAS

**“CONSTRUCCIÓN DE UN SISTEMA INTELIGENTE
PARA LA ASIGNACIÓN DINÁMICA DE TURNOS EN EL
ÁREA DE CONSULTA EXTERNA DEL HOSPITAL
REGIONAL ISIDRO AYORA, MEDIANTE EL ANÁLISIS
HEURÍSTICO UTILIZANDO HARDWARE Y SOFTWARE
LIBRE”.**

TESIS PREVIA A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS.

AUTORAS:

Alba del Rocío Pineda Torres

Carmen Elizabeth Becerra González

DIRECTOR DE TESIS:

Ing. Edwin René Guamán Quinche, Mg.Sc.

LOJA- ECUADOR

2014

1859



CERTIFICACIÓN

Sr. Ing.

Edwin René Guamán Quinche

**DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE LA
UNIVERSIDAD NACIONAL DE LOJA**

CERTIFICA:

Haber dirigido, corregido y revisado en todas sus partes, el desarrollo del proyecto de titulación de Ingeniería en Sistemas, denominado **“Construcción de un sistema inteligente para la asignación dinámica de turnos en el Área de Consulta Externa del Hospital Regional Isidro Ayora, mediante el análisis heurístico utilizando hardware y software libre”** realizado por las egresadas Alba del Rocío Pineda Torres y Carmen Elizabeth Becerra González, el cual cumple con los requisitos establecidos por las normas generales para la graduación en la Universidad Nacional de Loja, tanto en aspectos de forma como de contenido; por lo cual me permito autorizar su presentación para los fines pertinentes.

Loja Julio de 2014



.....
Ing. Edwin René Guamán Quinche, Mg.Sc
DIRECTOR DE TESIS



AUTORÍA

Nosotras: Alba del Rocío Pineda Torres y Carmen Elizabeth Becerra González 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: Alba del Rocío pineda Torres

Autor: Carmen Elizabeth Becerra González

Firma:

Firma:

Cédula: 1104125255

Cédula: 1103780407

Fecha: 28 de Noviembre del 2014

Fecha: 28 de Noviembre del 2014



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

Nosotras Alba del Rocío Pineda Torres y Carmen Elizabeth Becerra González, declaramos ser autoras de la tesis titulada: **“CONSTRUCCIÓN DE UN SISTEMA INTELIGENTE PARA LA ASIGNACIÓN DINÁMICA DE TURNOS EN EL ÁREA DE CONSULTA EXTERNA DEL HOSPITAL REGIONAL ISIDRO AYORA, MEDIANTE EL ANÁLISIS HEURÍSTICO UTILIZANDO HARDWARE Y SOFTWARE LIBRE”**, como requisito para optar al grado de: **Ingeniero en Sistemas**; autorizamos al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestren 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 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 a los 28 días del mes de Noviembre del dos mil catorce.

Firma:

Autor: Alba del Rocío Pineda Torres

Cédula: 1104125255

Dirección: Loja (Av. Diego de Troya y
Pedro de León)

Correo Electrónico: albipin_t@yahoo.es

Teléfono: (07) 2542157

Celular: 0990110147

Firma:

Autor: Carmen Elizabeth Becerra González

Cédula: 1103780407

Dirección: Quito (Av. Alonso de Angulo y
Lauro Guerreo)

Correo Electrónico: careli_b@hotmail.com

Teléfono: (02) 6006928

Celular: 0995010823

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Edwin René Guamán Quinche, Mg.Sc.

Tribunal de Grado: Ing. Luis Roberto Jácome Galarza, Mg.Sc.

Ing. Waldemar Vitorino Espinoza Tituana, Mg.Sc.

Ing. Iván Siguencia Larreategui Tene, Mg.Sc.



AGRADECIMIENTO

Queremos dejar constancia de nuestros sentimientos de reconocimiento a las autoridades y catedráticos de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, quienes con su significativa labor educativa nos ayudan a descubrir y abrir las puertas del conocimiento, contribuyendo así con la formación profesional en el campo de la informática.

Nuestro agradecimiento muy sincero y respetuoso a los directivos y personal administrativo del Hospital Regional Isidro Ayora quienes nos permitieron realizar el proyecto de tesis para culminar nuestros estudios universitarios.

Gratitud a nuestro Director de tesis Ing. Edwin René Guamán Quinche por compartir sus conocimientos científicos-pedagógicos y darnos el asesoramiento necesario para desarrollar y concluir con éxito este proyecto.

Finalmente, a todas y cada una de las personas que con su desinteresado apoyo lograron motivarnos para obtener el resultado final de nuestro esfuerzo, la graduación.



DEDICATORIA

Está dedicado principalmente a DIOS, por permitirme culminar con éxito el esfuerzo de todos estos años de estudio. Para Él mi agradecimiento infinito.

A mi madre María Torres, por ser una persona excepcional. Quien me ha brindado su apoyo incondicional y ha hecho suyos mis preocupaciones y problemas. Gracias por tu amor, paciencia y comprensión.

Y a mi hija María José, por ser lo más grande y valioso que Dios me ha regalado, quien es la fuente de inspiración y la razón que me impulsa a salir adelante.

Alba del Rocío Pineda Torres

El presente proyecto de tesis lo dedico a Dios que me ha dado capacidad y voluntad para cumplir con los objetivos que me he planteado, a mis padres Manuel Becerra y Enma González por su abnegación y apoyo incondicional.

A mi esposo Wladimir que me ha apoyado incondicionalmente en todo momento y a mis hijos Adrián e Itzayana Silva.

Carmen Elizabeth Becerra González



CESIÓN DE DERECHOS

Nosotras Alba del Rocío Pineda Torres con cédula de identidad número 1104125255 y Carmen Elizabeth Becerra González con cédula de identidad número 1103780407, autoras intelectuales del presente trabajo de titulación, autorizamos a la Universidad Nacional de Loja, al Área de la Energía, las Industrias y los Recursos Naturales no Renovables y por ende a la Carrera de Ingeniería en Sistemas para hacer uso del mismo en lo que estime sea conveniente.

.....
Alba del Rocío Pineda Torres

.....
Carmen Elizabeth Becerra González



a. Título

“CONSTRUCCIÓN DE UN SISTEMA INTELIGENTE PARA LA ASIGNACIÓN DINÁMICA DE TURNOS EN EL ÁREA DE CONSULTA EXTERNA DEL HOSPITAL REGIONAL ISIDRO AYORA, MEDIANTE EL ANÁLISIS HEURÍSTICO UTILIZANDO HARDWARE Y SOFTWARE LIBRE”.



b. Resumen

El presente proyecto de titulación denominado ARDAC comprende el análisis, diseño e implementación de un sistema inteligente de asignación de turnos en el Área de Consulta Externa del Hospital Regional Isidro Ayora, el cual permite centralizar y mejorar el proceso de la administración y distribución de turnos, administrar los horarios de atención de cada médico tratante, presentar reportes de los turnos generados y presentar el código del turno en un display el mismo que fue construido utilizando hardware libre. En el caso de los médicos tratantes el sistema les permite visualizar los turnos que se les ha asignado y registrar la atención de un paciente.

Es por ello que para el proceso de desarrollo que inicia con la fase de análisis e interpretación del sistema se utilizaron las técnicas de recolección de información y métodos científicos idóneos que fueron aplicados a todos los actores que intervienen en el desarrollo de nuestro proyecto, también se utilizó ICONIX como metodología de desarrollo de software, la misma que permitió un desarrollo ágil, simplificado, iterativo e incremental sin desviarse de las necesidades del usuario. Este sistema se implementó en el lenguaje de programación C que es un software libre, se utilizó la biblioteca WxWidget para el desarrollo de interfaces gráficas y como gestor de la base de datos se utilizó MySql.

Además para la sustentación teórica descrita en la sección de revisión de literatura se recolectó información respecto a conceptos sobre el sistema y se definieron las herramientas de desarrollo utilizadas.

En la sección de resultados se detalla todo el proceso para el desarrollo de ARDAC siguiendo la metodología de desarrollo utilizada, de esta forma se describen los requerimientos del sistema, diseño de la aplicación; diagramas como el de clases, de casos de uso entre otros para lograr los resultados esperados; así como también las pruebas y análisis de resultados. En la discusión se indica cómo se alcanzaron cada uno de los objetivos planteados al inicio del proyecto.



Finalmente se exponen las conclusiones obtenidas luego de finalizar el proyecto, así como se incluyen las recomendaciones derivadas en base a la experiencia alcanzada durante todo el proceso de desarrollo del presente proyecto denominado ARDAC.



Summary

The present titling project called ARDAC includes the analysis, design and implementation of an intelligent system of allocation of shifts in the Area of consultation outside of the Hospital Regional Isidro Ayora, which allows to centralize and improve the management and distribution of shift process, manage the schedules of each physician care, present generated shift reports and present the code of the turn on a display which was built using free hardware. In the case of the treating physicians system allows to monitor the shifts that have been assigned and record a patient's attention.

Therefore, that for the development process that begins with the analysis phase and interpretation of the system information gathering techniques were used and appropriate scientific methods that were applied to all the actors involved in the development of our project, also used ICONIX as a methodology of software development, which allowed an agile development simplified, iterative and incremental without deviating from the needs of the user. This system was implemented in the C programming language which is a free software, used the WxWidget library for the development of graphical interfaces and MySql was used as the database manager.

Also for the theoretical support described in the literature review section was collected information with respect to concepts about the system and the development tools used were defined. In addition to the theoretical support described in the literature review section was collected information with respect to concepts about the system and the development tools used were defined.

The results section details the process for the development of ARDAC following the methodology of development, thus describes the requirements of the system, design of the application; diagrams as classes, use cases among others to achieve the expected results; as well as the testing and analysis of results. The discussion shows how each of the objectives established at the beginning of the project were achieved.

Finally the conclusions are exhibited obtained after finishing the project, as well as there are included the recommendations derived based on the experience reached during the whole process of development of the present project named ARDAC.



Índice de Contenidos

CERTIFICACIÓN	II
AUDITORÍA	III
CARTA DE AUTORIZACIÓN DE TESIS	IV
AGRADECIMIENTO	V
DEDICATORIA	VI
CESIÓN DE DERECHOS	VII
a. Título	VIII
b. Resumen	IX
Summary.....	XI
Índice de Contenidos.....	XII
Índice de Figuras.....	XVII
Índice de Tablas.....	XXI
c. Introducción	23
d. Revisión de Literatura	25
1. Sistema Inteligente de Asignación de Turnos.....	25
1.1. ¿Qué es un Sistema Inteligente de Asignación de Turnos?.....	25
1.2. Elementos de un Sistema Inteligente de Asignación de Turnos.....	25
1.3. Heurística del sistema.....	25
1.3.1. Búsquedas eficientes.....	25
1.3.2. Optimización.....	26
2. Herramientas de Desarrollo.....	28
2.1. Hardware Libre (open-hardware).....	28
2.1.1. Placa Arduino.....	28
2.1.1.1. Introducción.....	28
2.1.1.2. Uso.....	28
2.1.1.3. Versiones de la Placa Arduino.....	29



2.1.1.4. Características físicas de la Placa Duemilanove:	30
2.1.1.5. Elementos necesarios para trabajar con Arduino Duemilanove.	31
2.1.2. Display de Siete Segmentos	33
2.1.2.1. Introducción:	33
2.1.2.2. Concepto:	33
2.1.2.3. Funcionamiento:	33
2.1.2.4. Esquema eléctrico del visualizador:	35
2.1.2.5. Características generales de un display de 7 segmentos.....	36
2.1.2.6. Protección.....	37
2.1.2.7. Construcción del display de 7 segmentos	37
2.2. Software Libre.....	39
2.2.1. Entorno de desarrollo de la Placa Arduino Duemilanove.....	40
2.2.2. Wxwidgets	46
2.2.2.1. Introducción	46
2.2.2.2. Bibliotecas de Wxwidgets	46
2.2.3.. g++	48
2.2.4. Lenguaje de programación C.....	51
2.2.4.1. Introducción	51
2.2.4.2. Propiedades.....	51
2.2.5. Teoría de colas	53
2.2.5.1. Introducción	53
2.2.5.2. Conceptos básicos.....	53
2.2.5.3. Descripción de un problema de colas	54
2.2.5.4. Elementos existentes en la teoría de colas	54
2.2.5.5. Tipos de colas.....	56
2.2.5.6. Objetivos de la teoria de colas	57
2.2.5.7. Características de las colas	57
2.2.5.8. Terminología	58



e. Materiales y Métodos	59
1. Técnicas de recolección de Información.....	59
2. Métodos de Investigación.....	60
3. Metodología para el desarrollo del software.....	62
4. Metodología para el desarrollo del hardware.....	64
f. Resultados	66
1. Desarrollo de las metodologías empleadas.....	66
1.1. Presentación de resultados con la metodología Iconix.....	66
1.1.1. Especificación de requerimientos.....	66
1. Introducción.....	66
1.1. Propósito.....	66
1.2. Ámbito del Sistema.....	66
1.3. Definiciones y Abreviaturas.....	67
1.4. Referencias.....	69
2. Descripción General.....	69
2.1. Perspectiva del sistema.....	69
2.2. Funciones del sistema.....	69
2.3. Características de los usuarios.....	70
2.4. Restricciones.....	72
2.5. Suposiciones y Dependencias.....	73
2.6. Requerimientos Específicos.....	74
2.6.1. Interfaces Externas.....	74
2.6.2. Requisitos Funcionales.....	74
2.6.3. Requerimientos no Funcionales.....	76
2.7. Atributos del Sistema.....	77
2.7.1. Fiabilidad.....	77
2.7.2. Disponibilidad.....	77
2.7.3. Seguridad.....	78



2.7.4. Mantenimiento	78
2.7.5. Restricciones de Diseño	78
2.8. Modelado de la aplicación.....	79
2.8.1. Modelo del Dominio	79
2.8.2. Diagrama de Casos de Uso	80
2.8.2.1. Descripción de los casos de uso	81
2.8.2.1.1. Ingreso al Sistema	81
2.8.2.1.2. Solicitar Turno.....	84
2.8.2.1.3. Administración de Turnos.....	94
2.8.2.1.4. Administrar Horarios	100
2.8.2.1.5. Administrar Doctor-Especialidad	107
2.8.2.1.6. Ingreso al Sistema	122
2.8.2.1.7. Turnos Doctor	125
2.8.2.1.8. Generar Reportes	128
2.8.2.1.9. Administrar Usuario.....	136
2.9. Diagrama de Paquetes	140
2.10. Diagrama de Clases	141
2.11. Diagrama de la Base de datos.....	142
2.12. Presentación de Resultados , metodología de desarrollo de hardware CAD.....	143
2.12.1. Diagrama de Componentes del hardware	144
2.13. Pruebas de la aplicación.....	149
2.13.1. Tipos de pruebas	149
2.13.1.1. Pruebas de funcionalidad.....	149
2.13.1.2. Pruebas de aceptación	150
2.13.2. Validación de los resultados de la Aplicación.....	151
2.13.2.1. Validación del desenvolvimiento de la Aplicación	151
2.13.2.2. Ejecución del plan de Pruebas.....	152
2.13.2.3. Informe del Resultado de las Pruebas de Validación	153



g. Discusión	155
1. Desarrollo de la propuesta alternativa.....	155
2. Valoración técnica económica ambiental	157
h. Conclusiones	161
i.Recomendaciones	162
J. Bibliografía	162
k. Anexos	166
ANEXO I	166
ANEXO II	167
ANEXO III	168
ANEXO IV	170
ANEXO V	172
ANEXO VI.....	182
ANEXO VII.....	183
ANEXO VIII.....	185
ANEXO IX.....	186
ANEXO X.....	187



ÍNDICE DE FIGURAS

Figura 1: Heurística del Sistema	27
Figura 2: Placa Duemilanove	29
Figura 3: Polaridad	31
Figura 4: Fuente de Alimentación	31
Figura 5: Cable USB.....	32
Figura 6: Denominación de cada segmento del display	34
Figura 7: Esquema del display de tipo ánodo	34
Figura 8: Esquema eléctrico de las conexiones del interior de un indicador luminoso de 7 segmentos.	35
Figura 9: Esquema eléctrico del visualizador	36
Figura 10: Cajas en forma de 8.....	37
Figura 11: Cajas con soportes para leds.....	38
Figura 12: Cajas con 7 segmentos de tres leds en cada soporte	38
Figura 13: Conexión entre leds	38
Figura 14: Displays terminados de 7 segmentos	39
Figura 15: Gestor de Paquetes Synaptic	40
Figura 16: Orígenes del Software	41
Figura 17: Mensaje de Aviso “Los repositorios han cambiado”.	41
Figura 18: Descargando información de paquetes.....	42
Figura 19: Entorno de Arduino.....	43
Figura 20: Archivo sieteseg.pde desarrollado en Entorno de Arduino.....	44
Figura 21: Archivo sieteseg.cpp desarrollado en Entorno de Arduino	44
Figura 22: Archivo sietesegdriver.h desarrollado en Entorno de Arduino	45
Figura 23: Bibliotecas de wxWidgets	47
Figura 24: Pantalla Menú Principal	49
Figura 25: Código para crear la interfaz gráfica Menú Principal.....	50



Figura 26: Conexión a la base de datos.....	52
Figura 27: Proceso de un sistema de colas	53
Figura 28: Un sistema de colas típico	54
Figura 29: Tipos de colas.....	57
Figura 30: Modelo del Dominio	79
Figura 31: Diagrama de casos de Uso.....	80
Figura 32: Diagrama de Secuencia: Ingreso al Sistema.....	83
Figura 33: Diagrama de Secuencia: Solicitar Turno	89
Figura 34: Diagrama de Secuencia: Curso Alterno A: Campos Vacíos	90
Figura 35: Diagrama de Secuencia: Curso Alterno B: Opciones no Seleccionadas. ...	90
Figura 36: Diagrama de Secuencia: Curso Alterno C: Opciones no Seleccionadas. ...	91
Figura 37: Diagrama de Secuencia: Curso Alterno D: Elección de la opción Buscar. .	91
Figura 38: Diagrama de Secuencia: Curso Alterno del Alterno D.A.: Mensaje de Aviso.....	92
Figura 39: Diagrama de Secuencia: Curso Alterno del Alterno D.B: Elige opción Cancelar.....	92
Figura 40: Diagrama de Secuencia: Curso Alterno E: Mensaje de Aviso.....	93
Figura 41: Diagrama de Secuencia: Curso Alterno F: Elige opción Cancelar.....	93
Figura 42: Diagrama de Secuencia: Administración de Turnos.....	97
Figura 43: Diagrama de Secuencia: Curso Alterno A: Elige opción Activar Turno.....	98
Figura 44: Diagrama de Secuencia: Curso Alterno B: Selección de Fecha.....	98
Figura 45: Diagrama de Secuencia: Curso Alterno C: Elige la opción Salir.....	99
Figura 46: Diagrama de Secuencia: Administrar Horario	104
Figura 47: Diagrama de Secuencia: Curso Alterno A: Elige la opción Agregar.....	105
Figura 48: Diagrama de Secuencia: Curso Alterno del Alterno A.A: Mensaje de Error.....	105
Figura 49: Diagrama de Secuencia: Curso Alterno del Alterno A.B: Elección de la opción Salir.....	106
Figura 50: Diagrama de Secuencia: Curso Alterno B: Elección de la opción Borrar..	106



Figura 51: Diagrama de Secuencia: Administrar Doctor-Especialidad	114
Figura 52: Diagrama de Secuencia: Curso Alterno A: Elección de opción Eliminar Doctor.....	115
Figura 53: Diagrama de Secuencia: Curso Alterno del Alterno A.A: Elige opción No.....	115
Figura 54: Diagrama de Secuencia: Curso Alterno B: Elige opción Editar Doctor.....	116
Figura 55: Diagrama de Secuencia: Curso Alterno del Alterno B.A: Campos Vacíos.....	117
Figura 56: Diagrama de Secuencia: Curso Alterno del Alterno B.B: Elige opción Cancelar.....	117
Figura 57: Diagrama de Secuencia: Curso Alterno C: Elige opción Agregar Especialidad.....	118
Figura 58: Diagrama de Secuencia: Curso Alterno del Alterno C.A: Elige opción Cancelar.....	119
Figura 59: Diagrama de Secuencia: Curso Alterno D: Eliminar Especialidad.....	119
Figura 60: Diagrama de Secuencia: Curso Alterno E: Elige opción Editar Especialidad.....	120
Figura 61: Diagrama de Secuencia: Curso Alterno del Alterno E.A: Campos Vacíos.....	120
Figura 62: Diagrama de Secuencia: Curso Alterno del Alterno E.B: Elige opción Cancelar.....	121
Figura 63: Diagrama de Secuencia: Curso Alterno F: Elige opción Salir.....	121
Figura 64: Diagrama de Secuencia: Ingreso al Módulo Doctor.....	124
Figura 65: Diagrama de Secuencia: Turnos Doctor	127
Figura 66: Diagrama de Secuencia:Curso Alterno A: Escoge la opción Salir.....	127
Figura 67: Diagrama de Secuencia: Generar Reportes	132
Figura 68: Diagrama de Secuencia: Curso Alterno A: Escoge la opción Reporte de turnos por Doctor	133
Figura 69: Diagrama de Secuencia: Curso Alterno B. Escoge la opción Reporte de Turnos por Especialidad.....	134
Figura 70: Diagrama de Secuencia: Curso Alterno C: Mensaje de Aviso.....	134



Figura 71: Diagrama de Secuencia: Curso Alterno D: Selección de la opción Salir.....	135
Figura 72: Diagrama de Secuencia: Curso Normal de Eventos: Administrar Usuarios.....	138
Figura 73: Diagrama de Secuencia: Curso Alterno de Eventos A: Elección de la opción Cancelar.....	139
Figura 74: Diagrama de Secuencia: Curso Alterno de Eventos B: Elige opción Inhabilitada.....	139
Figura 75: Diagrama de Paquetes	140
Figura 76: Diagrama de Clases	141
Figura 77: Diagrama de la Base de Datos	142
Figura 78: Interfaz electrónica del hardware	143
Figura 79: Diagrama de componentes del hardware.....	144
Figura 80: Diagrama esquemático de partes del display de 7 segmentos.....	145
Figura 81: Diagrama esquemático de partes del controlador	146
Figura 82: Conexión de las partes del Hardware.....	148



ÍNDICE DE TABLAS:

TABLA I. CARACTERISTICAS FISICAS DE LA PLACA DUEMILANOVE	28
TABLA II. CARACTERISTICAS GENERALES DE UN DISPLAY DE 7 SEGMENTOS	34
TABLA III: DEFINICIONES Y ABREVIATURAS DEL SISTEMA	69
TABLA IV. CARACTERISTICAS USUARIO: ADMINISTRADOR	72
TABLA V. CARACTERISTICAS USUARIO: ASISTENTE ADMINISTRATIVO	73
TABLA VI. CARACTERISTICAS USURIO: DOCTOR.....	74
TABLA VII: REQUERIMIENTOS FUNCIONALES DEL SISTEMA	76
TABLA VIII: REQUERIMIENTOS FUNCIONALES DEL ASISTENTE ADMINISTRATIVO	77
TABLA IX: REQUERIMIENTOS FUNCIONALES DEL USUARIO PACIENTE	77
TABLA X: REQUERIMIENTOS FUNCIONALES DEL USUARIO DOCTOR	78
TABLA XI: REQUERIMIENTOS FUNCIONALES DEL USUARIO DIRECTOR	78
TABLA XII: REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA	78
TABLA XIII: CASO DE USO: INGRESO AL SISTEMA	83
TABLA XIV: CASO DE USO: SOLICITAR TURNO.....	86
TABLA XV. CASO DE USO: ADMINISTRACION DE TURNOS.....	96
TABLA XVI. CASO DE USO: ADMINISTRAR HORARIO	102
TABLA XVII. CASO DE USO: ADMINISTRAR DOCTOR-ESPECIALIDAD.....	109
TABLA XVIII. CASO DE USO: INGRESO AL MÓDULO DOCTOR.....	124
TABLA XIX. CASO DE USO: TURNOS DOCTOR.....	127
TABLA XX. CASO DE USO: GENERAR REPORTES	130
TABLA XXI. CASO DE USO: ADMINISTRAR USUARIOS	138
TABLA XXII: ESPECIFICACIONES TECNICAS DEL CONTROLADOR.....	149
TABLA XXIII. EJECUCIÓN DEL PLAN DE PRUEBAS APLICADO A USUARIOS DEL SISTEMA.....	154
TABLA XXIV. INFORME DE RESULTADOS DE LAS PRUEBAS DE	



VALIDACIÓN.....	155
TABLA XXV. RECURSOS HUMANOS.....	159
TABLA XXVI. RECURSOS TENICOS Y TECNOLOGICOS.....	160
TABLA XXVII. RECURSOS MATERIALES.....	161
TABLA XXVIII. RESUMEN DE COSTOS.....	162



c. Introducción

El Hospital Regional "Isidro Ayora" de Loja, es una Entidad del Sistema de Servicios de Salud del Ministerio de Salud Pública del Ecuador, implementado para prestar servicios de diagnóstico, atención de salud ambulatoria y de hospitalización, con calidad, calidez, equidad y solidaridad a usuarios que demanden sus servicios. Dispone de personal capacitado e infraestructura física básica. Además contribuye al incremento del desarrollo profesional en procura de mejorar la calidad de vida de la población de la región sur y del país. Sin embargo se debe mencionar que con la aceleración de las comunicaciones y la rapidez con que la innovación tecnológica se presenta en la vida pública y privada, especialmente en el área médica, los procesos de atención al paciente en lo que se refiere a la obtención de turnos de atención médica se han vuelto caducos y obsoletos es así que en la mayoría de Instituciones de salud, los usuarios deben formar colas de atención para poder acceder a este servicio, ocasionando así muchas veces una excesiva aglomeración de usuarios y también una distribución no equitativa de los turnos de atención a los diferentes médicos que en dichos centros laboran.

Por lo tanto, como entes investigadores del presente proyecto de titulación con miras a dar solución a los problemas antes mencionados se ha desarrollado un sistema denominado ARDAC con el objetivo de mejorar el manejo de los procesos de administración de turnos a través de un hardware y software diseñado expresamente para el efecto.

En este sentido, este proyecto está estructurado por la sección d. Revisión de Literatura en la que se describe conceptos, definiciones y teorías pertinentes con el objeto de investigación, además se detallan las herramientas de desarrollo utilizadas para el desarrollo del sistema ARDAC.

En la sección e. Materiales y Métodos se describen las Técnicas de recolección de Información que se aplicaron para la obtención de requerimientos, también se describen los Métodos Científicos utilizados para realizar el proyecto. Por último se describe la



metodología de desarrollo de software y hardware utilizada, haciendo una breve explicación de las fases que la integran.

En la sección f. Resultados se detalla el proceso metodológico de desarrollo utilizado para Software y Hardware respectivamente, permitiendo presentar claramente las actividades desarrolladas en cada fase.

En la sección g. Discusión se explica el desarrollo de la propuesta alternativa, describiendo como se ha dado cumplimiento a cada uno de los objetivos y en la valoración técnico económica ambiental se expone los recursos: humanos, materiales, técnicos/tecnológicos utilizados durante el desarrollo del presente sistema, así como la valoración ambiental.

Las conclusiones se obtuvieron en base al cumplimiento de los objetivos planteados y de los resultados conseguidos. Las recomendaciones describen las sugerencias para que en un futuro se mejore el presente sistema así como también se emiten consideraciones que se deben tomar en cuenta para el desarrollo de proyectos similares.

Finalmente la Bibliografía consta de una lista de fuentes bibliográficas que han servido como fuentes para el desarrollo y ejecución del presente proyecto de titulación. Además se agregan los anexos que también es una parte complementaria y necesaria para la ejecución del sistema.



d. Revisión de Literatura

1. Sistema Inteligente de Asignación de Turnos

1.1. ¿Qué es un Sistema Inteligente de Asignación de Turnos?

Un Sistema Inteligente de Asignación de Turnos es un programa de computación que reúne características y comportamientos asimilables a la inteligencia humana, diseñado para hacer más fácil el control de administración de colas físicas que forman los usuarios de un establecimiento de atención al público, proporcionando eficiencia en la gestión de los mismos en espera de ser atendidos.[2]

Es así que el desarrollo de la aplicación ARDAC basada en este concepto mejorará la calidad de servicio en el Área de Consulta Externa del Hospital Regional Isidro Ayora, perfeccionando el flujo de pacientes y presentando reportes en los cuales se detalla los turnos que han sido emitidos ya sea en forma general, de acuerdo a la especialidad o médico tratante.

1.2. Elementos de un Sistema Inteligente de Asignación de Turnos

- Base de datos de pacientes
- Base de datos de médicos
- Base de datos de especialidades
- Datos de horarios de labor de cada medico
- Reportes de turnos entregados

1.3. Heurística del sistema.

1.3.1. Búsquedas eficientes.

Una búsqueda eficiente la definiremos como una instrucción o instrucciones que por sobre todo reduce las búsquedas de grandes cantidades de datos a grupos pequeños que se ajustan a las características de la necesidad en presencia. Es así que para el desarrollo de ARDAC se aplicó el análisis heurístico basado en este tipo de búsqueda con la cual se



encontró una solución al problema de asignación de turnos, usando para ello una cantidad de recursos razonables

1.3.2. Optimización.

La mayoría de problemas que se presentan en la gestión de datos de nuestro algoritmo se resuelven con búsquedas anidadas, con el propósito específico de lograr procesos óptimos y eficientes. Es así que hemos creado a cada objeto (Doctor, horario, Paciente, etc.) atributos que nos permitan acelerar búsquedas por medio de condiciones que reducen drásticamente los grupos de datos a evaluarse.

Un Doctor tiene una o varias especialidades y un horario para cada uno de ser el caso. Entonces, el algoritmo de asignación de turnos, primero recoge la información o los parámetros que nos permiten saber quién o quienes pueden atender a aquel paciente. Como parámetro escondido que manipula el sistema tenemos el de disponibilidad, que consiste en verificar si el doctor no tiene previamente un turno asignado y de la misma manera si el tiempo está dentro del horario del doctor.

Estos parámetros que recoge el algoritmo son los que posteriormente nos ayudaran a simplificar las búsquedas. Pues para determinar el rango de búsquedas, primero debemos hacer varias búsquedas simples con el objeto de establecer las condiciones que se aplicaran a los grupos de datos que cumplan con los parámetros solicitados. Por Ejemplo:

- 1.- Determinar horarios.
- 2.- Determinar especialidad
- 3.- Determinar disponibilidad
- 4.- Determinar duración.
- 5.- Determinar exigencia.

Todas las condicionantes se marcan sobre atributos creados lógicamente para satisfacer la eficiencia de las búsquedas y no aplicar búsquedas sobre datos que no cumplen con los



requerimientos solicitados, puesto que esto ralentizaría los procesos del sistema y la integridad de la información.

A continuación se indica la fracción de código en donde se aplica la heurística para poder realizar el proceso de asignación de turnos del sistema.

```
FrmSolicitarTurno::onSelectEspecialidad(wxCommandEvent& event){  
  
    ListaEspecialidades::Node* ref;  
    lista_doctores=NULL;  
  
    if (cbxEspecialidad->GetSelection()==0){  
        lista_doctores.Clear();  
        return;  
    }  
    ref = especialidades.Item(cbxEspecialidad->GetSelection() -1);  
  
    Especialidad* e = ref->GetData();  
    e->debug=1;  
    wxString hora_sel = this->choiceHoraTurno->GetString(choiceHoraTurno->GetCurrentSelection());  
  
    this->fecha = wxString::Format(wxT("%d-%02d-%02d %s:00"),  
    this->m_datePicker1->GetValue().GetYear(),  
    this->m_datePicker1->GetValue().GetMonth() + 1, // subiendo uno al mes  
    this->m_datePicker1->GetValue().GetDay(),  
    hora_sel.c_str()  
    );  
    ok=0;  
  
    doctores.DeleteContents(false);  
    doctores = e->lista_doctores_disponibles(this->fecha,wxAtoi(e->id_especialidad));  
    ListaDoctores::iterator iter;  
    for (iter = doctores.begin(); iter != doctores.end(); ++iter){  
        Doctor *d = *iter;  
        if (d->Disponibilidad(d->id_doctor,this->fecha)){  
            lista_doctores.Add(d->nombre);  
        }  
        //cout << " - " << d->nombre.mb_str() << endl;  
        ok=1;  
        btnSolicitar->Enable();  
    }  
    this->listDoctores->Set(lista_doctores); .....
```

Figura 1: Heurística del Sistema



2. Herramientas de Desarrollo

2.1. Hardware Libre (open-hardware)

Se llama hardware libre a los dispositivos hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita. La filosofía, las ideas sobre libertad del conocimiento de Software Libre es aplicable a la de Hardware Libre. Se debe recordar en todo momento que Libre No significa Gratis. [3]

2.1.1. Placa Arduino

2.1.1.1. Introducción

Arduino es una plataforma open-hardware basada en una sencilla placa con entradas y salidas (E/S), analógicas y digitales, y en un entorno de desarrollo para crear software (programas) para la placa. Su corazón es el chip Atmega8¹, un chip sencillo y de bajo costo que permite el desarrollo de múltiples diseños. [4]

Al ser open-hardware tanto su diseño como su distribución es libre. Es decir, puede utilizarse libremente para desarrollar cualquier tipo de proyecto sin tener que adquirir ningún tipo de licencia. [5]

2.1.1.2. Uso

Arduino puede utilizarse en el desarrollo de objetos interactivos autónomos o puede conectarse a un PC a través del puerto serie utilizando lenguajes como Flash, Processing², MaxMSP, etc. Las posibilidades de realizar desarrollos basados en Arduino tienen como límite la imaginación.

Así mismo, su sencillez y su bajo costo, recomiendan su uso como elemento de aprendizaje e iniciación en el mundo de la electrónica digital.[6]

¹ Atmega8- Circuito integrado de muy alta escala de integración que contiene las partes funcionales de un computador

² Más información en <http://www.processing.org>



2.1.1.3. Versiones de la Placa Arduino

Existen varias versiones de la placa Arduino. La mayoría usan el ATmega168 o ATmega328 de Atmel, mientras que las placas más antiguas usan ATmega8, entre las más populares podemos mencionar las siguientes:[7]

- Placa Serie
- Placa Serie de una Capa
- Placa USB
- Placa de Prototipos
- Placa Bluetooth
- Placa Stand-alone (Autónoma)
- Placa Duemilanove

En función de nuestro interés para el desarrollo de ARDAC se ha elegido la Placa Duemilanove que es la que más se adapta al desarrollo del proyecto de titulación, cuyo nombre significa 2009 en italiano que fue el año cuando salió al mercado.

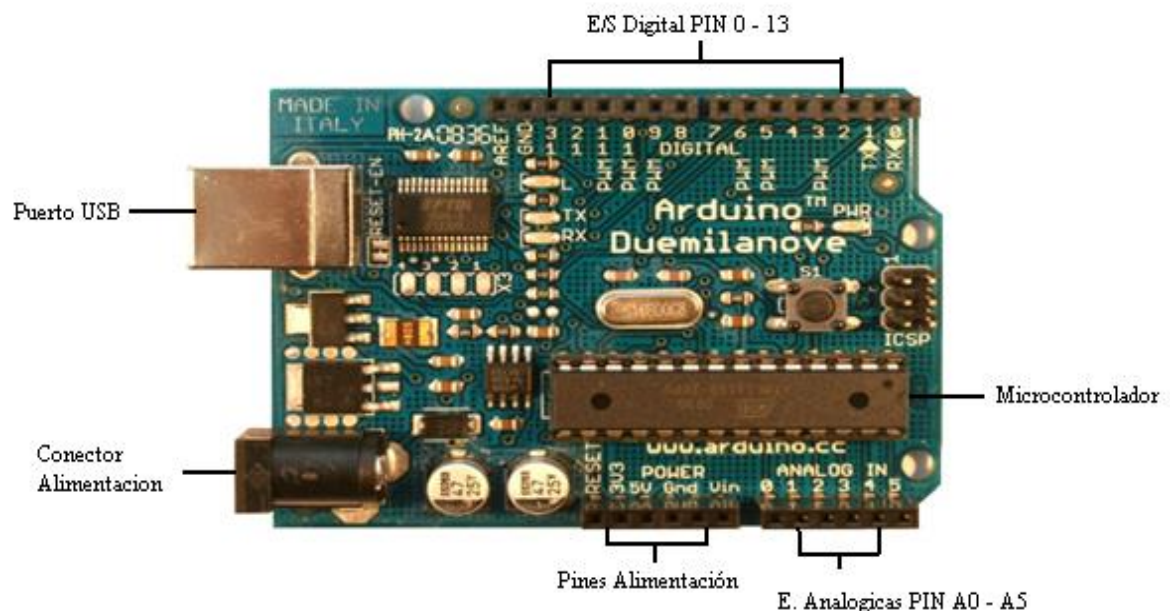


Figura 2: Placa Duemilanove



Esta placa constituye la manera más rápida y económica de desarrollar proyectos con microcontroladores ya que viene totalmente montada y con un bootloader³ listo para conectar al USB del ordenador. Es un producto Open-Source⁴ que lleva una placa con procesador y un sencillo entorno de desarrollo que implementa el lenguaje Processing/Wiring⁵. Al ser un proyecto abierto hay una gran cantidad de ejemplos y librerías listas para utilizar con Arduino.

2.1.1.4. Características físicas de la Placa Duemilanove:

La longitud y amplitud máxima de la placa Duemilanove es de 2.7 y 2.1 pulgadas respectivamente, con el conector USB y la conexión de alimentación sobresaliendo de estas dimensiones. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas. A continuación describimos más detalladamente sus características: [8]

TABLA I. CARACTERÍSTICAS FÍSICAS DE LA PLACA DUEMILANOVE

Características físicas de la Placa Duemilanove	
Microcontrolador	ATmega368 (ATmega168 en versiones anteriores)
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Pines E/S digitales	14 (6 proporcionan salida PWM ⁶)
Pines de entrada analógica	6
Intensidad por pin	40 Ma
Intensidad en pin 3.3V	50 mA
Memoria Flash	16 KB (ATmega168) o 32 KB (ATmega328) de las cuales 2 KB las usa el gestor de arranque(bootloader)
SRAM	1 KB (ATmega168) o 2 KB (ATmega328)

³Bootloader (gestor de arranque) programa sencillo que no tiene la totalidad de las funcionalidades de un sistema operativo, y que está diseñado exclusivamente para preparar todo lo que necesita el sistema operativo para funcionar.

⁴ Más información en <http://www.libertya.org/comunidad/open-source>

⁵ Más información en <http://projectbot.blogspot.com/2011/01/sobre-arduino-y-su-lenguaje.html>

⁶PWM (Pulse Width Modulation), señal modulada por anchura de pulso.



EEPROM	512 bytes (ATmega168) o 1 KB (ATmega328)
Velocidad de reloj	16 MHz

2.1.1.5. Elementos necesarios para trabajar con Arduino Duemilanove.

Hardware

En lo que ha hardware se refiere podemos mencionar los siguientes elementos:

- **Alimentación**

El Arduino Duemilanove puede ser alimentado vía conexión USB o con una fuente de alimentación externa. Las fuentes de alimentación externas (no-USB) pueden ser tanto un transformador o una batería.

La fuente de alimentación que se ha elegido para la placa Duemilanove es continua ya que de esta manera no se tendrá que estar pendiente de sustituir las pilas en caso de que se queden sin carga. Es así que el voltaje de la fuente puede ser de entre 6 y 25 voltios, y la polaridad del conector debe ser como se indica en la imagen.



Figura 3: Polaridad



Figura 4: Fuente de Alimentación



- **Cable de Comunicaciones (Serie/USB)**

En función del modelo de nuestra placa el cable de comunicaciones es USB tal y como se muestra en la imagen inferior. Con un conector tipo A (para conectar al PC) y otro tipo B (para conectar a la placa) en sus extremos. No hay que equivocarlo con el cable mini-USB que habitualmente se utiliza con dispositivos más pequeños como cámaras de fotos y lectores de tarjetas.



Figura 5: Cable USB

- **Memoria**

El ATmega328 tiene 32KB (el ATmega168 tiene 16 KB) de memoria flash para almacenar código (2KB son usados para el arranque del sistema (bootloader). El ATmega328 tiene 2 KB (Atmega168 1 KB) de memoria SRAM . El ATmega328 tiene 1KB (ATmega168 512 bytes) de EEPROM, que puede a la cual se puede acceder para leer o escribir con la [Reference/EEPROM |librería EEPROM]].

- **Entradas y Salidas**

Cada uno de los 14 pines digitales en el Duemilanove pueden utilizarse como entradas o como salidas usando las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()` . Las E/S operan a 5 voltios. Cada pin puede proporcionar o recibir una intensidad máxima de 40mA y tiene una resistencia interna (desconectada por defecto)de 20-50kOhms.

El Duemilanove tiene 6 entradas analógicas, y cada una de ellas proporciona una resolución de 10bits (1024 valores). [8], [9]



2.1.2. Display de Siete Segmentos

2.1.2.1. Introducción

En muchos lugares públicos se ha visto unos indicadores luminosos (display) que indican el turno. Normalmente son de dos dígitos, lo que les permite contar hasta 99.

Es por ello que para el desarrollo del proyecto de titulación que trata sobre la asignación de turnos se ha creído conveniente construir este tipo de display utilizando hardware libre; el cual facilitará la visualización del turno que se ha asignado al paciente.

2.1.2.2. Concepto:

El visualizador de siete segmentos (llamado también display) es una forma de representar números en equipos electrónicos. Está compuesto de siete segmentos que se pueden encender o apagar individualmente. Cada segmento tiene la forma de una pequeña línea.[10]

2.1.2.3. Funcionamiento:

El display de 7 segmentos o visualizador de 7 segmentos es un componente que se utiliza para la representación de números en muchos dispositivos electrónicos debido en gran medida a su simplicidad. Aunque externamente su forma difiere considerablemente de un diodo LED⁷ (diodos emisores de luz) típico, internamente están constituidos por una serie de diodos LED con unas determinadas conexiones internas, estratégicamente ubicados de tal forma que forme un número 8.

A cada uno de los segmentos que forman el display se les denomina a, b, c, d, e, f y g y están ensamblados de forma que se permita activar cada segmento por separado consiguiendo formar cualquier dígito numérico. A continuación se muestra la denominación de cada segmento del display.[11]

⁷LED: dispositivo semiconductor que emite radiación visible, infrarroja o ultravioleta cuando se hace pasar un flujo de corriente eléctrica a través de este en sentido directo.

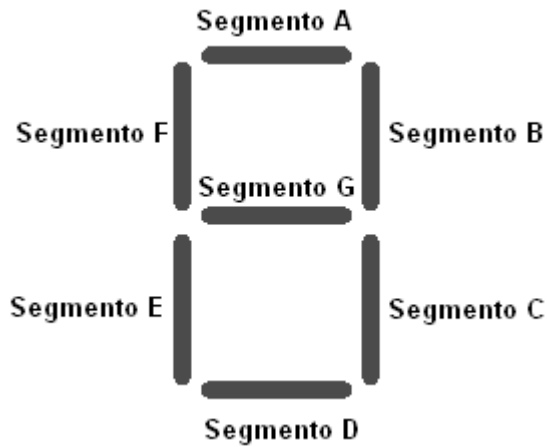


Figura 6: Denominación de cada segmento del display

En la siguiente figura se indica el esquema del display de tipo ánodo común el mismo que fue empleado para el desarrollo de nuestro sistema.

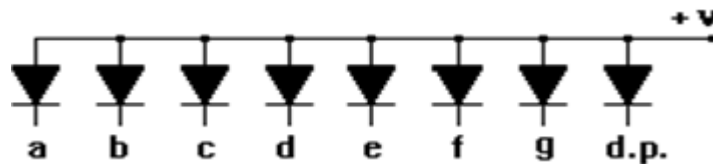


Figura 7: Esquema del display de tipo ánodo

En este tipo de display todos los ánodos de los diodos LED están unidos y conectados a la fuente de alimentación.

En este caso para activar cualquier elemento hay que poner el cátodo⁸ del elemento a tierra a través de una resistencia para limitar la corriente que pasa por el elemento.

Los segmentos pueden ser de diversos colores, aunque el display más comúnmente utilizado es el de color rojo, por su facilidad de visualización.

En la figura siguiente se muestra un indicador de siete segmentos que contiene siete LED rectangulares (a - g), en el que cada uno recibe el nombre de segmento porque forma

⁸Cátodo: Electrodo o polo negativo de un generador eléctrico; es el electrodo de menor potencial y por el que sale la energía eléctrica. En los conductores eléctricos, la corriente entra por el ánodo y sale por el cátodo.



parte del símbolo que está mostrando. Con un indicador de siete segmentos se pueden formar los dígitos del 0 al 9, también las letras a, c, e y f y las letras minúsculas b y d. Polarizando los diferentes diodos, se iluminaran los segmentos correspondientes. De esta manera podemos señalar todos los números en base 10. Por ejemplo, si queremos representar el número de 1 en el display deberemos mandar señal a los diodos b y b, y los otros diodos deben de tener tensión cero. El primer dígito representa al diodo a, el segundo al b, el tercero al c, y así sucesivamente. Un cero representa que no polarizamos el diodo, es decir no le aplicamos tensión. Un uno representa que el diodo esta polarizado, y por lo tanto, emite luz.

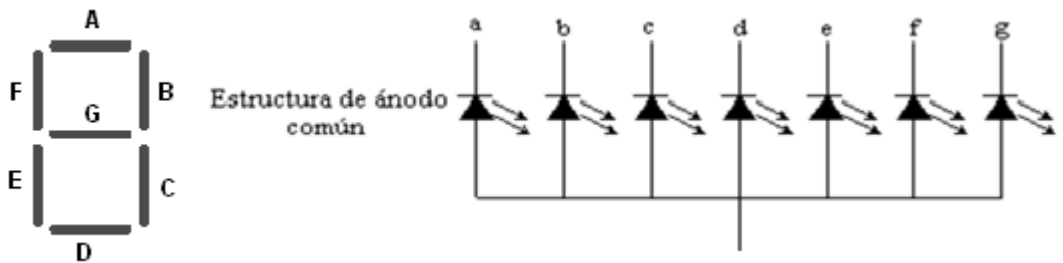


Figura 8: Esquema eléctrico de las conexiones del interior de un indicador luminoso de 7 segmentos.

2.1.2.4. Esquema eléctrico del visualizador:

Se ha realizado el esquema de tres segmentos de los 7 que lo componen ya que el resto es idéntico.

Como se puede deducir del esquema, el número de cables entre la placa controladora y el visualizador digital es de 8. (7 para los segmentos y 1 para el negativo la alimentación).[11]

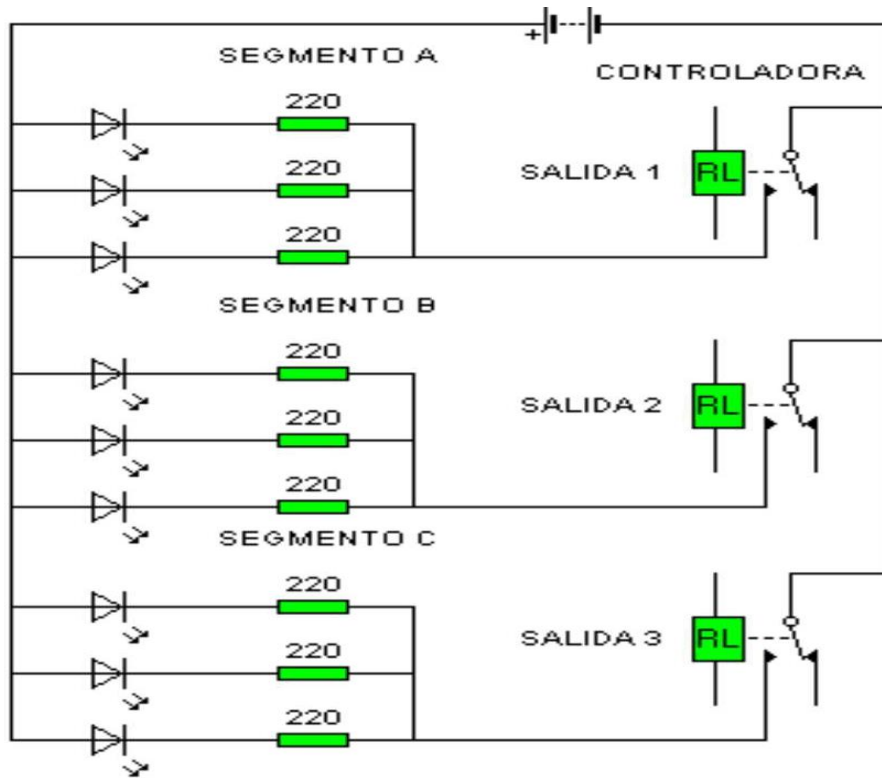


Figura 9: Esquema eléctrico del visualizador

2.1.2.5. Características generales de un display de 7 segmentos

TABLA II. CARACTERISTICAS GENERALES DE UN DISPLAY DE 7 SEGMENTOS

Solidez:	excelente
Angulo de visibilidad:	150 grados
Consumo por dígito:	50 Mw
Vida media en horas:	100000
Luminosidad:	Buena
Facilidad de montaje:	excelente
Vcc general:	1.5 volt.
La Vcc depende del color del LED. Para un color rojo: Vcc=	1.7volt.
Vcc más =	2 volt.
Vcc (máx):	2 V



Dependiendo de la tensión aplicada obtendremos una intensidad. Es aconsejable no sobrepasar la V_{cc} recomendada. Si se alcanza la V_{cc} máxima se puede destruir el segmento.

2.1.2.6. Protección

Cada segmento (y el punto) es un led como cualquier otro. Debido a esto la corriente media que se debe aplicar es de 15mA. Dependiendo de la lógica que estemos empleando debemos utilizar una resistencia por cada entrada y así no forzar el dispositivo:

Lógica TTL (5V): 220 ohmios.[11]

2.1.2.7. Construcción del display de 7 segmentos

Para efecto del proyecto se construyó 2 cajas en forma de 8 que crean el display de siete segmentos, utilizando para ello espuma flex, láminas de acrílico, pegamento y otros componentes reciclables.



Figura 10: Cajas en forma de 8

En cada extremo de las cajas se colocaron soportes para luego ubicar ahí segmentos de 3 leds. El número de soportes colocados son siete uno por cada segmento del display.



Figura 11: Cajas con soportes para leds

Se ubicó 7 segmentos de tres leds conectados en serie en cada uno de los soportes de las cajas, por consiguiente para formar cada número necesitamos de (7 segmentos * 3 Diodos Leds) utilizando en su totalidad 21 diodos (leds) por cada display, los mismos que para funcionar necesitan de un voltaje de 11 a 15v con 1 Amperio de fuerza.



Figura 12: Cajas con 7 segmentos de tres leds en cada soporte

Se realizó la conexión entre los leds utilizando para ello la estructura del ánodo común donde todos los **diodos LED** están unidos y conectados a la fuente de alimentación.

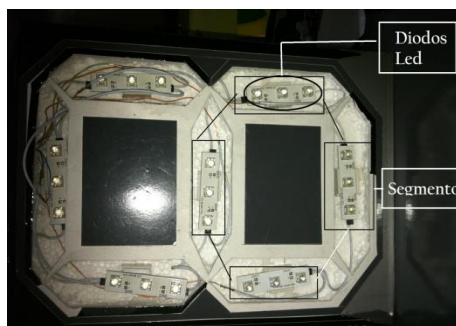


Figura 13: Conexión entre leds



Una vez que se conectó todos los componentes, se procedió a sellar las cajas quedando listo los displays de siete segmentos.



Figura 14: Displays terminados de 7 segmentos

2.2. Software Libre

Software libre (en inglés free software) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al código fuente es un requisito previo). [12]

En base a lo anteriormente expuesto cabe indicar que para el desarrollo del proyecto se ha hecho uso del sistema operativo Linux. A continuación se detalla cómo se utilizaron las diferentes herramientas de desarrollo.



2.2.1. Entorno de desarrollo de la Placa ArduinoDuemilanove

Para instalar Arduino en un sistema Linux se necesita los siguientes programas para resolver las dependencias:

- Sun java runtime, jre.
- avr-gcc, compilador para la familia de microcontroladores avr de atmel.
- avr-libc, libc del compilador avr-gcc.

Una vez que se obtenga estos programas procedemos de la siguiente manera:

1. Se ejecuta el Gestor de Paquetes Synaptic (en Sistema » Administración).

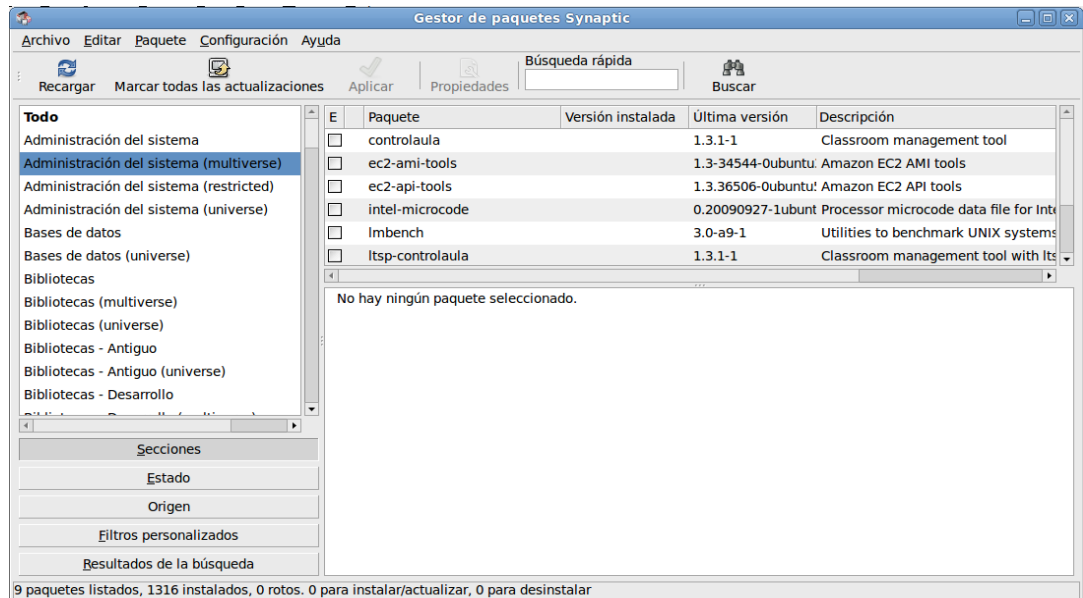


Figura 15: Gestor de Paquetes Synaptic

2. Es necesario habilitar los repositorios “Universe” y “Multiverse” para que se pueda acceder a todos los paquetes que se necesitan, para ello seguimos los siguientes pasos:

- Ir a Configuración » Repositorios.
- Hacer click en Añadir.
- Marcar la opción (Software restringido por copyright o cuestiones legales (multiverse)) y (Software libre mantenido por la comunidad (universe)) y hacer click en Añadir.

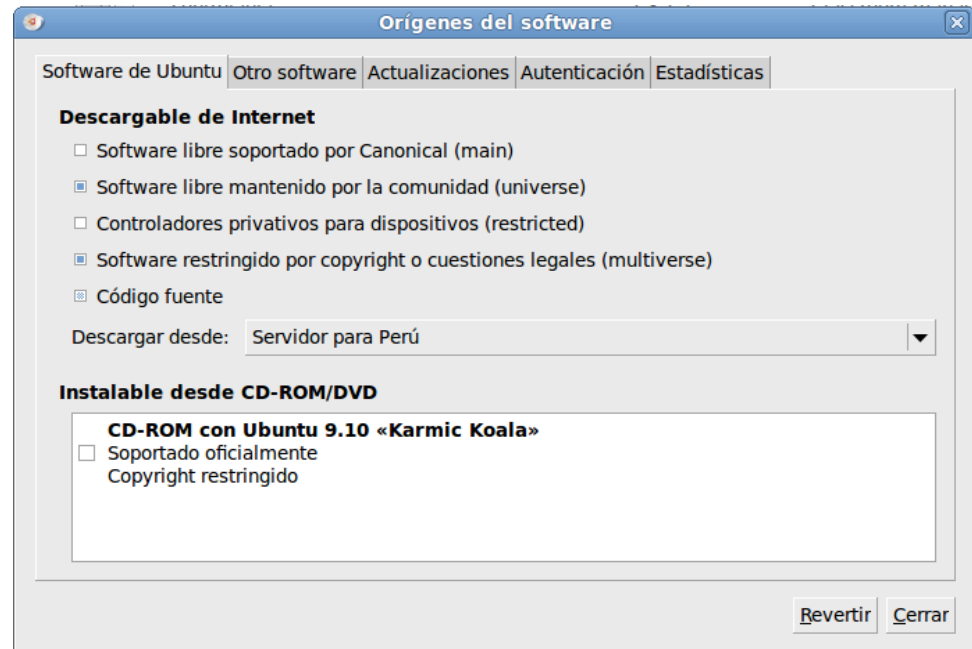


Figura 16: Orígenes del Software

- Hacer click en Cerrar y click en Cerrar de él diálogo “Los repositorios han cambiado”.

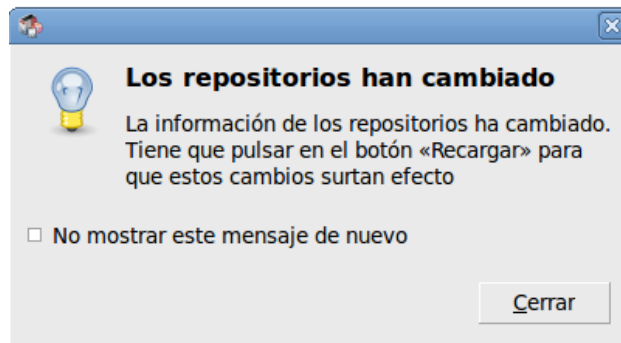


Figura 17: Mensaje de Aviso “Los repositorios han cambiado”.

- Hacer click en el botón Recargar de la barra de herramientas, y esperar que termine la descarga.

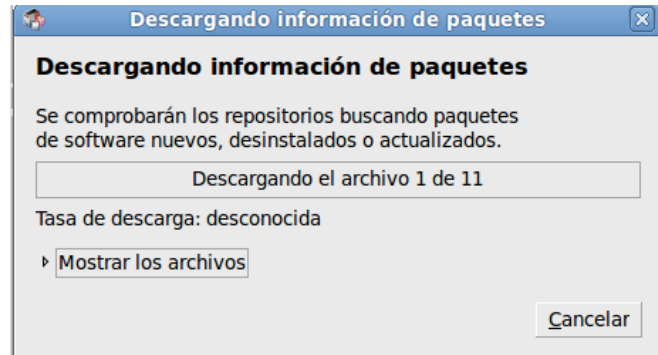


Figura 18: Descargando información de paquetes

- Marcar las siguientes opciones para instalar: “sun-java5-jre”, “gcc-avr”, “avr-libc”.
 - Hacer click en Aplicar de la barra de herramientas.
 - Hacer click en Aplicar del cuadro de diálogo. Esto instalará los paquetes seleccionados.
 - Aceptar la licencia de Java.
 - Esperar hasta completar la instalación: el cuadro de diálogo dirá “Cambios aplicados”. Luego hacer click en Cerrar.
 - Cerrar Synaptic.
3. Descargar la distribución de GNU/Linux de Arduino desde la siguiente página <http://www.arduino.cc/en/Main/Software>.
 4. Hacer doble click en el archivo .zip y arrastrar la carpeta que contiene a algún lugar (por ejemplo el Escritorio).
 5. Ejecutar el Terminal (en Aplicaciones » Accesorios).
 6. Escribir lo siguiente “_sudo update-alternatives -config java” y presionar Enter.
 7. Teclear el número de opción que tiene “java-1.5.0-sun” en él y presiona Enter. Esto hará de la versión de Java de Sun la predeterminada del sistema.
 8. Hacer doble click en “Arduino” en el directorio de aplicación de Arduino. Esto debería emitir un diálogo preguntando dónde guardamos los archivo directorio “Arduino” en la carpeta home es la ubicación típica.
 9. Hacer click en OK. El entorno Arduino debería abrirse tal como se muestra en la figura siguiente.[13]

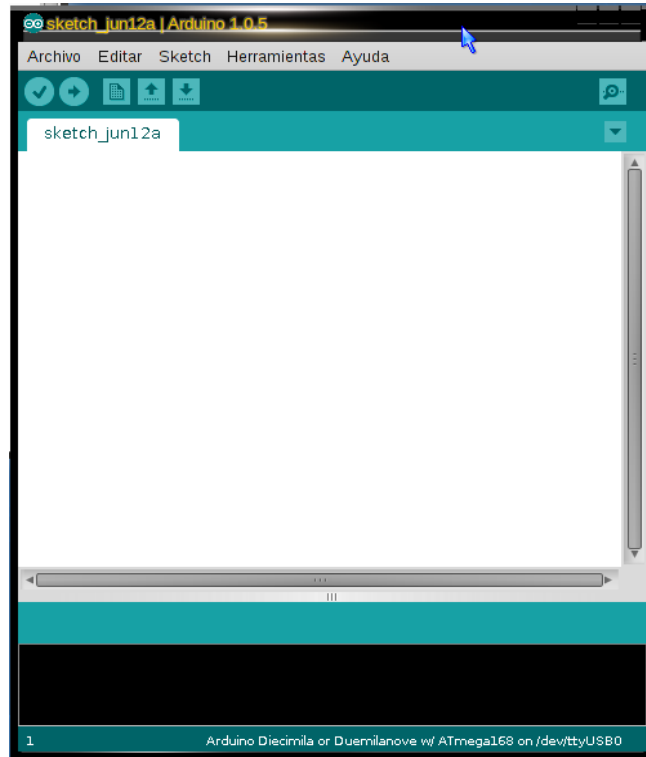


Figura 19: Entorno de Arduino

Una vez que se tiene el entorno de Arduino listo para operar se desarrolla el algoritmo que será el encargado de realizar la comunicación entre el software y el hardware; este algoritmo alojado en el microprocesador de la placa Arduino Duemilanove tendrá un estado tipo escucha, es decir que se estará ejecutando permanentemente a fin de transmitir los datos que se envíen del módulo Doctor del sistema Ardac hacia el comparador para luego presentar gráficamente dicho dato en los display de siete segmentos.

El algoritmo descrito anteriormente está desarrollado en lenguaje C puro y consta de tres partes:

- De un archivo denominado sieteseg.pde que se encuentra el directorio "ardac/7segplexed/" que es la parte del algoritmo que se estará ejecutando permanentemente en espera de algún dato que se envíe desde el software.



```
sieteseg | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda
sieteseg sietesegdriver.cpp sietesegdriver.h
/*
 * 7seg
 * Primer programa para el manejo del 74ls48
 */
#include "sietesegdriver.h"
Sietesegdriver Sietesd(40); //40ms

void setup()
{
  Serial.begin(9600);
  Sietesd.initialize();
  Serial.println("Iniciando debug en tty");
}

void loop()
{
  Sietesd.autotest(500);
  Serial.println("fin autotest");
}

'byte' does not name a type
In file included from sieteseg.ino:6:
sietesegdriver.h:17: error: 'byte' does not name a type
1 Arduino Diecimila or Duemilanove w/ ATmega168 on /dev/ttyUSB0
```

Figura 20: Archivo sieteseg.pde desarrollado en Entorno de Arduino

- La segunda parte consta de un archivo denominado sietesegdriver.cpp que se encuentra el directorio “ardac/7segplexed/”, esta parte del algoritmo está desarrollada para determinar cuál es el número de turno que fue enviado desde el módulo Doctor del sistema Ardac, para así asociar cada segmento del display con los pines de salida de la placa Arduino y presentar el número de turno gráficamente.

```
sieteseg | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda
sietesegdriver.cpp sietesegdriver.h
/*
Driver para el control del display doble de 7 segmentos
released into the public domain under Creative Commons
Esta libreria usa 7 lineas para cada segmento para poder represent
caracteres que no se podrian representar simplemente usando el cod
de BCD a 7 segmentos.
*/
#include "sietesegdriver.h"
#include "WConstants.h"

byte Sietesegdriver::caracteres[16][7] = {
  //{a,b,c,d,e,f,g}
  {1,1,1,1,1,0,0}, // 0
  {0,1,1,0,0,0,0}, // 1
  {1,1,0,1,1,0,1}, // 2
  {1,1,1,1,0,0,1}, // 3
  {0,1,1,0,0,1,1}, // 4
  {1,0,1,1,0,1,1}, // 5
  {1,0,1,1,1,1,1}, // 6
  {1,1,1,0,0,0,0}, // 7
  {1,1,1,1,1,1,1}, // 8
  {1,1,1,1,0,1,1}, // 9
};

'byte' does not name a type
In file included from sieteseg.ino:6:
sietesegdriver.h:17: error: 'byte' does not name a type
1 Arduino Diecimila or Duemilanove w/ ATmega168 on /dev/ttyUSB0
```

Figura 21: Archivo sietesegdriver.cpp desarrollado en Entorno de Arduino



- Por último la tercera parte consta de un archivo denominado `sietesegdriver.h` que se encuentra el directorio “`ardac/7segplexed/`”, en este archivo se definen las variables y procedimientos que deberán ejecutarse además se determina la frecuencia en que se presenta el número de turno en el display.

```
sieteseg | Arduino 1.0.5
Archivo  Editar  Sketch  Herramientas  Ayuda
sieteseg  sietesegdriver.cpp  sietesegdriver.h 5
#define MSB0 8
#define MSB1 4
#define MSB2 2
#define MSB3 1

class Sietesegdriver{
public:
ewgtfrfnhmfndbfvd
kjtjykurtr

    Sietesegdriver(int frecuencia); // auto constructor
    //Sietesegdriver(int userPines[]); // constructor con un array
    void initialize(); // inicializar lo que haya que inicializar
    void autotest(int delay_milli_second);
    void print(int number);
    void writepins(int pin, int data);
private:
    int freq; // la frecuencia de retardo antes de presentar el si
    int pines[NUMPINES]; // los pines de control y bus de datos
};

#endif

'byte' does not name a type
```

Figura 22: Archivo `sietesegdriver.h` desarrollado en Entorno de Arduino



2.2.2. Wxwidgets

2.2.2.1. Introducción

Las **wxWidgets** son unas bibliotecas multiplataforma y libres, para el desarrollo de interfaces gráficas programadas en lenguaje C++. Están publicadas bajo una licencia LGPL, similar a la GPL con la excepción de que el código binario producido por el usuario a partir de ellas, puede ser propietario, permitiendo desarrollar aplicaciones empresariales sin costo. [14]

Las wxWidgets proporcionan una interfaz gráfica basada en las bibliotecas ya existentes en el sistema (nativas), con lo que se integran de forma óptima y resultan muy portables entre distintos sistemas operativos. Están disponibles para Windows, MacOS, GTK+, Motif, OpenVMS y OS/2.

Las wxWidgets también pueden ser utilizadas desde otros lenguajes de programación, aparte del C++: Java, Javascript, Perl, Python, Smalltalk, Ruby. [15]

2.2.2.2. Bibliotecas de Wxwidgets

Empezando desde la versión 2.5.0 wxWidgets puede construirse como una sola bibliotecas grande (*en inglés monolithicbuild*) o como varias bibliotecas más pequeñas (*en inglés multilibbuild*). La estructura (*multilibbuild*) es la predeterminada por defecto.

La biblioteca wxWidgets es dividida en sub-bibliotecas tal como se muestra en el siguiente diagrama donde se indica las dependencias entre ellas:

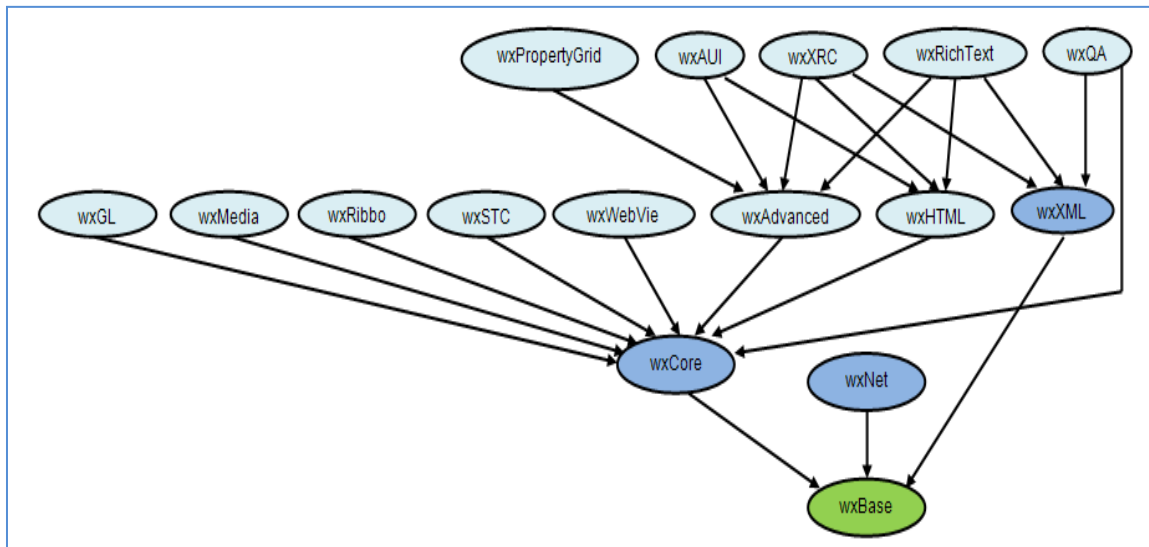


Figura 23: Bibliotecas de wxWidgets

A continuación se describe brevemente cada una de las bibliotecas principales de wxWidgets:[16]

Biblioteca wxBase

Esta biblioteca incluye clases como wxString, clases para el manejo de archivos y directorios de manera independiente del sistema, funcionalidades como: gráficos 2D, 3D con OpenGL, Bases de Datos (ODBC), Redes, Impresión, Hilos, visión e impresión del HTML, un sistema de archivos virtual y cuenta con algunos IDEs.

wxBase se puede utilizar para desarrollar aplicaciones en modo de consola, que no requiere de ninguna biblioteca de GUI o ejecutar el sistema X Window en Unix.

Biblioteca wxCore

Es una librería de enlace dinámico en donde las clases básicas GUI tales como clases de GDI⁹ o los controles se encuentran en esta biblioteca. Todas las aplicaciones wxWidgets

⁹(GraphicsDevice Interface) Interfaz de dispositivos para gráficos. Lenguaje de gráficos Windows utilizado para proveer salida a la pantalla e impresora. Las aplicaciones llaman a las funciones GDI en Windows para presentación e impresión.



GUI tienen que enlazar con la biblioteca, sólo las aplicaciones en modo consola no lo hacen.

Biblioteca wxXML

Esta biblioteca contiene clases sencillas para analizar documentos XML.

Biblioteca WxNet

Contiene clases para el acceso de red tales como:

- clases wxSocket (wxSocketClient , wxSocketServer y clases relacionadas)
- wxSocketOutputStream y wxSocketInputStream
- clases de la CIP basadas sockets (wxTCPClient , wxTCPConnection)
- wxURL
- wxInternetFSHandler (a wxFileSystem handler)

En la Figura siguiente se observa la línea de código ingresada en (Ubuntu: desde el terminal de comandos) mediante la cual se instalan los paquetes que son necesarios para poder desarrollar aplicaciones con wxWidgets.

```
sudo aptitude install libwxgtk2.8-0 libwxgtk2.8-dev wx2.8-headers wx-common
```

La instrucción indicada creara un directorio dentro de /usr/include/, en nuestro caso es el directorio /usr/include/wx-2.8/, ubicación que contendrá todas las librerías WX de desarrollo.

2.2.3. g++

g++ es el alias tradicional de *GNU C++*, un conjunto gratuito de compiladores de C++. Forma parte del GCC, *GNU CompilerCollection* (del inglés, colección de compiladores GNU).

En sistemas operativos GNU, *gcc* es el comando usado para ejecutar el compilador de C, mientras que *g++* ejecuta el compilador de C++¹⁰.

Los programas se compilan con el comando:

```
$ g++ nombre_clase.cpp -o nombre_clase `wx-config --cppflags --libs`
```

¹⁰<http://www.taringa.net/posts/linux/3840507/Compilar-y-Ejecutar-programas-en-C-C-sobre-Ubuntu.html>



Y lo ejecutamos con:

```
$. /nombre_clase
```

Una vez instalados todos los paquetes de wxWidgets se le da un mejor estilo a las interfaces diseñadas para el sistema ARDAC para ello se utilizó algunas de las bibliotecas descritas anteriormente separando así el contenido de la vista de la forma en cómo las verá el usuario. Los archivos referentes a la interfaz gráfica se encuentran ubicados en el directorio "ardac/ardacwx/" de la aplicación.

A continuación se indica una fracción de código del archivo frmprincipal.h en donde se implementan algunas de las clases de la biblioteca wxBase para crear la siguiente interfaz gráfica de menú de opciones del sistema Ardac.[17]



Figura 24: Pantalla Menú Principal



```
#include<wx/wx.h> //contiene la definicion de la mayoria de tipos de datos
#include<wx/string.h> //para manejo de cadenas
#include <wx/statbox.h>
#include <wx/dialog.h>
#include <wx/settings.h>
#include <wx/frame.h>
#include <wx/textctrl.h>
#include <wx/stattext.h>
#include <wx/fileconf.h>

classFrmPrincipal : public wxFrame{
wxMenuBar *menu;
wxMenu *archivo;
wxMenu *editar;
wxMenu *ayuda;
wxToolBar *toolBar;
wxPanel *panel;
wxButton *btnPausar;
wxButton *btnReiniciar;
wxButton *btnAgregar;
wxButton *btnBorrar;
wxButton *btnAgregar2;
wxButton *btnAgregar3;
wxButton *btnBorrar3;
wxButton *btnSalir;
wxStatusBar* statusBar;

public:
FrmPrincipal(constwxString& title);
voidonSalir(wxCommandEvent& event);
voidwakeUp(); // levantarse y configurarsuscosas
voidonConfig(wxCommandEvent& event);.....
```

Figura 25: Código para crear la interfaz gráfica Menú Principal



2.2.4. Lenguaje de programación C

2.2.4.1. Introducción

C es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores. No es un lenguaje de muy alto nivel y más bien un lenguaje pequeño, sencillo y no está especializado en ningún tipo de aplicación. Esto lo hace un lenguaje potente, con un campo de aplicación ilimitado. [18]

2.2.4.2. Propiedades

- Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de archivos, proporcionadas por bibliotecas.
- Es un lenguaje muy flexible que permite programar con múltiples estilos. Uno de los más empleados es el estructurado "no llevado al extremo" (permitiendo ciertas licencias de ruptura).
- Un sistema de tipos que impide operaciones sin sentido.
- Usa un lenguaje de preprocesado para tareas como definir macros e incluir múltiples archivos de código fuente.
- Acceso a memoria de bajo nivel mediante el uso de punteros.
- Interrupciones al procesador con uniones.
- Un conjunto reducido de palabras clave.
- Por defecto, el paso de parámetros a una función se realiza por valor. El paso por referencia se consigue pasando explícitamente a las funciones las direcciones de memoria de dichos parámetros.
- Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.
- Tipos de datos agregados (struct) que permiten que datos relacionados (como un empleado, que tiene un id, un nombre y un salario) se combinen y se manipulen como un todo (en una única variable "empleado").[19]



A continuación se detalla parte del código que se utilizó para la conexión a la base de datos utilizando este lenguaje cuyo archivo dbcliente.cpp se encuentra en el directorio “ardac/ardaclub”.

```
#include "dbcliente.h"
#include <iostream>
using namespace std;
DBCliente::DBCliente(){
// especificar el sql de la variable
//sql = wxT("SELECT * FROM doctores");
debug=0;
}
boolDBCliente::conectar(){
// conectarse a la base de datos y retornar una conexion,
// deberÃa recibir un configurador sino, estos son los parÃmetros por defecto
// Aqui tiene que buscar un .conf y de equiparsearlo y configurarlo
//if (!this->host && !this->dbname){
wxFileInputStream input(wxT("ardac.ini"));
wxTextInputStream text(input);
this->config = new wxFileConfig(input);
this->config->Read(wxT("general/host"), &host);
this->config->Read(wxT("general/dbname"), &dbname);
this->config->Read(wxT("general/dbuser"), &dbuser);
this->config->Read(wxT("general/dbpasswd"), &dbpasswd);
//cout<< "DEBUG DBCliente::conectar : conectando a: host "<<this->host.mb_str()<<"
dbname:"<<this->dbname.mb_str()<<" dbuser:"<<this->dbuser.mb_str()<<" dbpasswd:"<<this-
>dbpasswd.mb_str()<<endl;
//}
mysql_init(&mysql);
connection =
mysql_real_connect(&mysql,host.Trim().mb_str(),dbuser.Trim().mb_str(),dbpasswd.Trim().mb_str(),db
name.Trim().mb_str(),0,0,0);
if (connection == NULL){
printf("%s",mysql_error(&mysql));
this->conectado=false;
return 0;
}else{
this->conectado=true;
return 1;
}
}
/*
* close: cerrar la conexiÃ3n con la BD, mandatorio en cada conexiÃ3n
*/
voidDBCliente::close(){
//cout<<"DEBUG DBCliente::close: cerrando la conexion"<<endl;
mysql_close(connection);
/*
if(connection!=NULL || result){
cout<<"DEBUG DBCliente::close cerrando la conexion"<<endl;
if (result) mysql_free_result(result);
if (connection) mysql_close(connection);
}
*/
}
```

Figura 26: Conexión a la base de datos



2.2.5. Teoría de colas

2.2.5.1. Introducción

La teoría de colas es el estudio matemático de las líneas de espera (o colas) permitiendo el análisis de varios procesos relacionados como: la llegada al final de la cola, la espera en la cola, etc. [22]

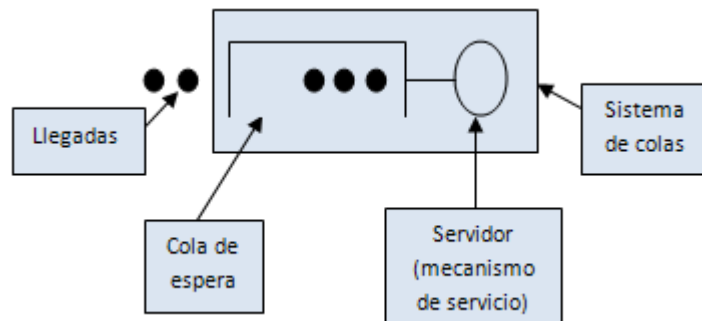


Figura 27: Proceso de un sistema de colas

2.2.5.2. Conceptos básicos

Clientes: Término usado en un sistema de colas para referirse a:

- Gente esperando líneas telefónicas desocupadas.
- Máquinas que esperan ser reparadas.
- Aviones esperando aterrizar, etc.

Instalaciones de Servicio: Este término se usa para referirse a:

- Líneas telefónicas.
- Talleres de reparación.
- Pistas de aeropuerto.

Llegadas: Es el número de clientes que llegan a las instalaciones de servicio.

Tasa de Servicio: Este término se usa para designar la capacidad de servicio, por ejemplo:

- Un sistema telefónico entre dos ciudades puede manejar 90 llamadas por minuto.



- Una instalación de reparación puede de media, reparar máquinas a razón una cada 8 horas.
- Una pista de aeropuerto en la que aterrizan dos aviones por minuto.

Número de servidores de servicio: Es la cantidad de servidores de que disponemos:

- Número de conmutadores telefónicos.
- Número de puestos de reparación.
- Número de pistas de aterrizaje de un aeropuerto.

El número de servidores no tiene porqué ser siempre en paralelo, es decir, puede que un sistema de colas tenga varias fases.

2.2.5.3. Descripción de un problema de colas

Un sistema de colas se puede describir como: “clientes” que llegan buscando un servicio, esperan si este no es inmediato, y abandonan el sistema una vez han sido atendidos. En algunos casos se puede admitir que los clientes abandonan el sistema si se cansan de esperar.

El término “cliente” se usa con un sentido general y no implica que sea un ser humano, puede significar piezas esperando su turno para ser procesadas o una lista de trabajo esperando para imprimir en una impresora en red, etc. [23]

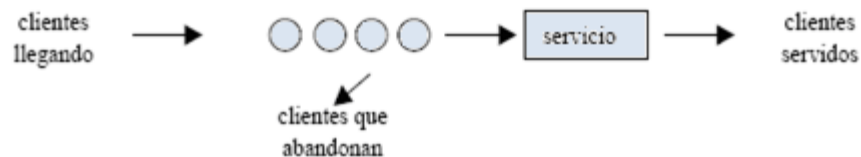


Figura 28: Un sistema de colas típico

2.2.5.4. Elementos existentes en la teoría de colas

- **Proceso básico de colas:** Los clientes que requieren un servicio se generan en una fase de entrada. Estos clientes entran al sistema y se unen a una cola. En determinado momento se selecciona un miembro de la cola, para proporcionarle el servicio,



mediante alguna regla conocida como disciplina de servicio. Luego, se lleva a cabo el servicio requerido por el cliente en un mecanismo de servicio, después de lo cual el cliente sale del sistema de colas.

- **Fuente de entrada o población potencial:** Una característica de la fuente de entrada es su tamaño. El tamaño es el número total de clientes que pueden requerir servicio en determinado momento. Puede suponerse que el tamaño es infinito o finito.
- **Cliente:** Es todo individuo de la población potencial que solicita servicio como por ejemplo una lista de trabajo esperando para imprimirse.
- **Capacidad de la cola:** Es el máximo número de clientes que pueden estar haciendo cola (antes de comenzar a ser servidos). De nuevo, puede suponerse finita o infinita.
- **Disciplina de la cola:** La disciplina de la cola se refiere al orden en el que se seleccionan sus miembros para recibir el servicio. Por ejemplo, puede ser:
 - FIFO (first in firstout) primero en entrar, primero en salir, según la cual se atiende primero al cliente que antes haya llegado.
 - LIFO (last in firstout) también conocida como pila que consiste en atender primero al cliente que ha llegado el último.
 - RSS (random selection of service) que selecciona los clientes de manera aleatoria, de acuerdo a algún procedimiento de prioridad o a algún otro orden.
 - Processor Sharing – sirve a los clientes igualmente. La capacidad de la red se comparte entre los clientes y todos experimentan con eficacia el mismo retraso.
- **Mecanismo de servicio:** El mecanismo de servicio consiste en una o más instalaciones de servicio, cada una de ellas con uno o más canales paralelos de servicio, llamados servidores.



- **Redes de colas.** Sistema donde existen varias colas y los trabajos fluyen de una a otra. Por ejemplo: las redes de comunicaciones o los sistemas operativos multitarea.
- **Cola:** Una cola se caracteriza por el número máximo de clientes que puede admitir. Las colas pueden ser finitas o infinitas.
- **El proceso de servicio:** Define cómo son atendidos los clientes. [23] ,[24]

2.2.5.5. Tipos de colas

Según el tipo de sistema de colas, tenemos varios tipos de éstas, las cuales son:

a) Una línea, un servidor

El primer sistema que se muestra se llama un sistema de un servidor y una cola o puede describir una consulta de un médico.

b) Una línea, múltiples servidores

El segundo, una línea con múltiples servidores, es típico de una peluquería o una panadería en donde los clientes toman un número al entrar y se les sirve cuando les llega el turno.

c) Varias líneas, múltiples servidores

El tercer sistema, en que cada servidor tiene una línea separada, es característico de los bancos y las tiendas de autoservicio. Para este tipo de servicio pueden separarse los servidores y tratarlos como sistemas independientes de un servidor y una cola. Esto sería válido sólo si hubiera muy pocos intercambios entre las colas. Cuando el intercambio es sencillo y ocurre con frecuencia, como dentro de un banco, la separación no sería válida.

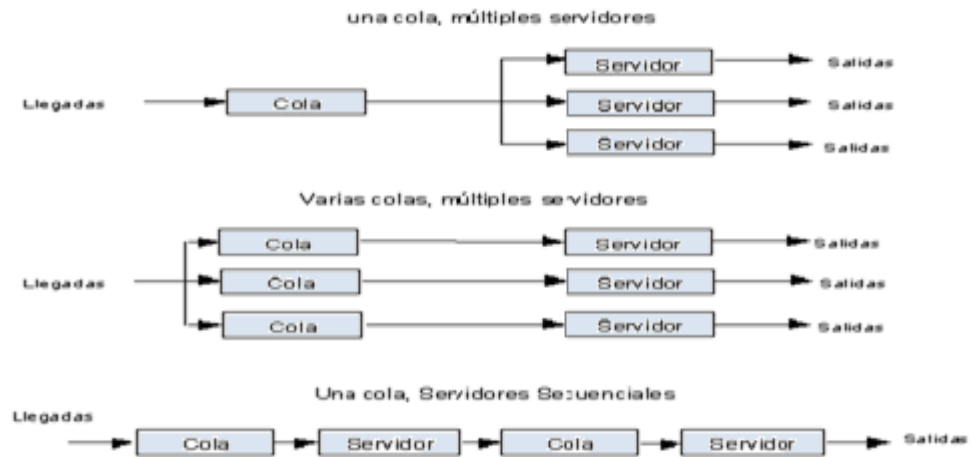


Figura 29: Tipos de colas

2.2.5.6. Objetivos de la teoría de colas

- Identificar el nivel óptimo de capacidad del sistema que minimiza el coste del mismo.
- Evaluar el impacto que las posibles alternativas de modificación de la capacidad del sistema tendrían en el coste total del mismo.
- Establecer un balance equilibrado (“óptimo”) entre las consideraciones cuantitativas de costes y las cualitativas de servicio.

Hay que prestar atención al tiempo de permanencia en el sistema o en la Cola: la “paciencia” de los clientes depende del tipo de servicio específico considerado y eso puede hacer que un cliente “abandone” el sistema.

2.2.5.7. Características de las colas

Seis son las características básicas que se deben utilizar para describir adecuadamente un sistema de colas:

- a) Patrón de llegada de los clientes
- b) Patrón de servicio de los servidores



- c) Disciplina de cola
- d) Capacidad del sistema
- e) Número de canales de servicio
- f) Número de etapas de servicio

Algunos autores incluyen una séptima característica que es la población de posibles clientes.

2.2.5.8. Terminología

Usualmente siempre es común utilizar la siguiente terminología estándar:

- **Estado del sistema:** Número de clientes en el sistema.
- **Longitud de la cola:** Número de clientes que esperan servicio.
- **$N(t)$:** Número de clientes en el sistema de colas en el tiempo t ($t \geq 0$).
- **$P_n(t)$:** Probabilidad de que exactamente n clientes estén en el sistema en el tiempo t , dado el número en el tiempo cero.
- **s :** Número de servidores en el sistema de colas.
- **λ_n :** Tasa media de llegadas (número esperado de llegadas por unidad de tiempo) de nuevos clientes cuando hay n clientes en el sistema.
- **μ_n :** Tasa media de servicio para todo el sistema (número esperado clientes que completan su servicio por unidad de tiempo) cuando hay n clientes en el sistema.

Nota: λ representa la tasa combinada a la que todos los servidores ocupados logran terminar sus servicios

- **λ :** Cuando λ es constante para toda n
- **λ_n :** Cuando λ es constante para toda $n \geq 1$ [23]



e. Materiales y Métodos

1. Técnicas de recolección de Información.

Para una correcta investigación en el desarrollo del proyecto de titulación, se utilizaron las siguientes técnicas; las mismas fueron aplicadas a todos los actores que intervienen en el proyecto ARDAC (Administrador del Hospital, Asistentes administrativas, Doctores).

- **La observación directa.**- Consistió en la observación de los procesos referentes a la asignación de turnos que se desarrollan en el área de consulta externa del Hospital Isidro Ayora y así palpar la situación actual en la que se encuentra, permitiendo de esta manera conocer, ¿Cuál es la principal dificultad que tienen para realizar la administración de turnos?, ¿Qué necesidades tiene el Área de Consulta Externa para realizar este proceso?; además analizar e interpretar los datos obtenidos a través de ella con el fin de plantear una solución que mejore la administración de turnos en el Hospital.
- **Técnica de la entrevista.**- La técnica utilizada fue la entrevista no estructurada o libre, la misma que plantea preguntas abiertas que surgen de acuerdo a las respuestas del entrevistado. Esta técnica permitió de forma específica la interacción social con las personas (asistentes administrativas, administrador y doctores) para recolectar los requerimientos necesarios de manera directa.
- **Técnica de la encuesta.**- Esta técnica fue aplicada a los usuarios que intervienen en el desarrollo del proyecto ARDAC, tales como asistentes administrativos, administrador y doctores del Hospital Regional Isidro Ayora durante las pruebas de funcionalidad y validación del sistema, con el fin de conocer los diferentes puntos de vista que tienen así como también la aceptación del mismo (Ver sección k. Anexos apartados Anexo III, Anexo IV).



2. Métodos de Investigación

Los métodos investigativos que se utilizaron para el desarrollo del proyecto se detallan a continuación:

- **Método Exploratorio.-** Este método va de la mano de la técnica de observación directa el cual nos permitió familiarizarnos de los problemas existentes en el Hospital Regional Isidro Ayora y en especial en el Área de Consulta Externa como lo son:
 - Excesiva aglomeración de usuarios al momento de solicitar un turno de atención.
 - El proceso de asignación de turnos es lento y un poco obsoleto puesto que se lo realiza de una forma manual.
 - No existe una notificación sobre el estado y disponibilidad de turnos.
 - No existe una visualización previa del turno a atenderse.
- **Método inductivo.-** Este método, se lo utilizó en la recolección de la información de manera independiente de varias fuentes como son libros, direcciones de Internet, asesorías, entre otros, por cuanto se investigó textos relacionados con el tema global, para luego realizar un estudio e investigación exhaustiva de los puntos que estén relacionados.
- **Método Descriptivo.-** Este método se utilizó para la descripción de las características de la aplicación conjuntamente con sus interfaces gráficas. Las características son las siguientes:
 - Es una aplicación que utiliza tanto hardware como software libre para su desarrollo.
 - Esta aplicación está compuesta de tres módulos uno para la administración de turnos, otro de notificación de estados de turnos hacia los terminales de cada médico tratante y el tercero de notificación del turno a ser atendido en el display.



- **Método Cualitativo y Cuantitativo.-** El *cualitativo* se lo uso en las observaciones que se realizó de los elementos de observación como:
 - Asignación de turnos en colas de atención a pacientes
 - Gestión de registro de las especialidades que conforman el Área de Consulta Externa
 - Metodología para el desarrollo del software (Metodología ICONIX)
 - Organización de la información

Y el método *cuantitativo* fue utilizado en las tabulaciones de las pruebas de validación del sistema (ver sección f. Resultados apartado 2.3.2.3. Análisis de los resultados de las pruebas de validación), y en procesos donde se trabaja con cantidades, cifras, etc.

- **Método Científico.-** Mediante el uso de este método se realizó una serie de pasos sistemáticos que permitieron alcanzar y cumplir con los objetivos planteados al inicio del proyecto. Los objetivos cumplidos se demuestran a través de los resultados de las encuestas aplicadas a los usuarios del sistema y en las pruebas de aceptación elaboradas durante el desarrollo del sistema. (Ver sección f. Resultados apartado 2.13.1.2. Pruebas de aceptación).
- **Método Bibliográfico.-** Uno de los métodos fundamentales para el correcto desarrollo del presente proyecto de titulación fue el bibliográfico, gracias al cual se adquirieron las bases teóricas necesarias que permitieron lograr los objetivos trazados. Estas bases teóricas sirvieron para conocer a fondo el problema de investigación y las herramientas útiles para darle solución. (Ver sección d. Revisión de literatura).



3. Metodología para el desarrollo del software

A continuación se describen las actividades desarrolladas en cada una de las fases.

- **Análisis de requisitos.-** En esta etapa el trabajo es iniciado con un relevamiento informal de todos los requisitos que en principio deberían ser parte del sistema para posteriormente realizar la clasificación de estos datos (Ver sección f. Resultados apartado 2.6.2. Requerimientos funcionales del sistema), una vez ordenados fueron aprobados por parte de los usuarios, para la realización de los primeros diagramas que representa las agrupaciones funcionales con que se estructura el sistema desarrollado.

A partir de dichos requerimientos se elaboró un prototipado rápido de las interfaces del sistema para dar a los usuarios una mayor comprensión del sistema propuesto. Con esto intentamos establecer las especificaciones iniciales que no se hayan completado y hacer que los usuarios puedan comenzar a evaluar la aplicación para percatarse de lo que les gusta o no y así poder hacer los cambios necesarios. Este proceso lo repetimos en algunas ocasiones hasta llegar a concretar todas las características y funcionalidades del sistema.

Una vez que se definieron bien las características del sistema con el usuario, se realizó el prototipado final, la identificación de los casos de uso del sistema y sus actores, mostrando el conjunto de funcionalidades de cada uno (Ver sección f. Resultados apartado 2.8.2. Diagrama de Casos de Uso).

- **Análisis y diseño preliminar.-** En esta etapa se describió cada caso de uso con su curso normal y alterno de eventos (Ver sección f. Resultados apartado 2.8.2.1. Descripción de los casos de uso); luego se realizaron los diagramas de robustez en donde se ilustran gráficamente las iteraciones existentes entre los objetos participantes en cada caso de uso; esto nos permitió comprobar que las especificaciones del sistema son razonables.



Por último en esta etapa se actualizó el diagrama de clases que se lo definió previamente en el modelo del dominio con las clases y atributos encontrados en los diagramas de robustez.

- **Diseño detallado.**- En esta etapa se creó un diseño detallado de los casos de uso y se asignó mensajes y métodos para cada uno; para lo cual se elaboraron los diagramas de secuencia, con el fin de mostrar las interacciones entre objetos, así como para observar que métodos llevan las clases del sistema.

Para la realización de los diagramas de secuencia en la parte superior se copió: actores, objetos de tipo interfaz y entidad, identificados en el diagrama de robustez.

Los objetos controladores (verbos) del diagrama de robustez se convierten en los mensajes, se asignó las operaciones a las clases mientras se dibujaban los mensajes, verificando si el diseño satisfacía todos los requerimientos identificados desde el inicio.

- **Implementación.**- Después de realizar un buen análisis y diseño se procedió a generar el código basado en los diagramas ya elaborados y especificar la relación entre ellos, tomando en cuenta algunos factores como: **reusabilidad, extensibilidad y confiabilidad.**

Dentro de esta etapa también se realizaron test de pruebas de unidad e interacción al sistema (Ver sección f. Resultados apartado 2.13.1.2. Pruebas de aceptación), las mismas que nos permitieron validar la accesibilidad, navegabilidad y usabilidad del sistema las cuales fueron aplicadas en el Área de Consulta Externa del Hospital Regional Isidro Ayora, con la colaboración de las asistentes administrativas, administrador y doctores del mismo.

Finalmente con los resultados de las pruebas se determinó que los usuarios se sienten cómodos y seguros al hacer uso de la aplicación; este proceso además permitió depurar y corregir los errores e implementar los nuevos requerimientos.



4. Metodología para el desarrollo del hardware

Para la construcción del hardware que es parte del sistema ARDAC se ha utilizado la metodología de diseño asistida por computadora (**Computer Aided Design, CAD**), la misma que emplea técnicas gráficas para soportar el proceso de diseño. La introducción de dichas técnicas en el proceso de diseño de circuitos electrónicos es fundamental, ya que más allá de proveer interfaces gráficas para asistir el proceso, brinda la posibilidad de simular y verificar la descripción antes de llevar a cabo su implementación, minimizando el costo de elaborar circuitos potencialmente defectuosos y acelerando el diseño global.

A continuación se enumeran y explican brevemente algunas de las herramientas que ofrece esta metodología para utilizarlas durante el diseño e implementación del hardware y de las cuales se han utilizado para el presente proyecto:

Grafos y diagramas de flujo: mediante esta herramienta se elaboró el diagrama de la interfaz electrónica del hardware permitiendo de esta manera tener una mejor comprensión y descripción funcional del comportamiento de éste con respecto al software.

Descripción mediante esquemas: Para la construcción del hardware primero se elaboró el diagrama esquemático donde se muestra la ubicación y conexión de los componentes del display de siete segmentos, los mismos que facilitarán la visualización del turno que se ha asignado al paciente; así mismo se realizó el esquema del circuito para el controlador del hardware en donde se indica cada una de las partes que lo integran, el cual actúa como un traductor entre el dispositivo y el software del sistema.

Lenguajes de descripción: Para obtener el diseño del controlador del hardware y el diagrama esquemático de los displays de siete segmentos se utilizó el programa EAGLE versión 5.4.0. Además se hizo uso del lenguaje C tradicional permitiendo generar un software encargado de realizar tareas de automatización y asignación de turnos a través de la placa Arduino y finalmente se utilizó **Ubuntu** que es un sistema operativo basado en Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity.



Programación de dispositivos: En esta parte se procedió a instalar en el controlador de la placa Arduino el algoritmo para la presentación del número de turno en el display de 7 segmentos.



f. Resultados

1. Desarrollo de las metodologías empleadas

1.1. Presentación de resultados con la metodología Iconix

Mediante el uso de la metodología Iconix se desarrolló el software **“CONSTRUCCIÓN DE UN SISTEMA INTELIGENTE PARA LA ASIGNACIÓN DINÁMICA DE TURNOS EN EL ÁREA DE CONSULTA EXTERNA DEL HOSPITAL REGIONAL ISIDRO AYORA, MEDIANTE EL ANÁLISIS HEURÍSTICO UTILIZANDO HARDWARE Y SOFTWARE LIBRE”**, abarcando así todo el proceso de desarrollo de Software que se describe a continuación:

1.1.1. Especificación de Requerimientos

La especificación de requerimientos ha sido creada para definir los requerimientos funcionales del presente sistema utilizando la norma IEEE 830 1998.

1. Introducción

1.1. Propósito

El propósito del desarrollo del presente documento es el siguiente:

- Determinar los requisitos específicos del sistema.
- Determinar la funcionalidad del sistema.
- Determinar la operatividad del sistema.

Es así que una vez cumplidos estos propósitos el sistema será dirigido a las asistentes administrativas, administrador y doctores del Área de Consulta Externa del Hospital Regional Isidro Ayora.

1.2. Ámbito del Sistema

El sistema a desarrollar se denomina ARDAC el cual comprende el análisis, diseño e implementación de un sistema inteligente de asignación de turnos para el Área de Consulta Externa del Hospital Regional Isidro Ayora, mismo que permite centralizar y



mejorar el proceso de la administración y distribución de turnos, administrar los horarios de atención de cada médico tratante, presentar reportes de los turnos generados y presentar el código del turno en un display. En el caso de los médicos tratantes les permite visualizar los turnos que se les ha asignado y registrar la atención de un paciente. Cabe recalcar que el sistema no integrará módulos como el historial clínico de cada paciente.

Con todas las características detalladas anteriormente el sistema será capaz de

- Reducir o eliminar el tiempo de espera de un paciente para conseguir un turno.
- Agilizar la búsqueda de datos de un paciente.
- Reservar un turno para una fecha posterior.
- Indicar visualmente el número de turno que será atendido.

1.3. Definiciones y Abreviaturas

TABLA III: DEFINICIONES Y ABREVIATURAS DEL SISTEMA

Nombre	Descripción
Administrador	Usuario que posee privilegios para crear, modificar y eliminar datos, además puede obtener reportes del estado de turnos.
Reporte	Informe detallado sobre el estado de la asignación de turnos
Usuario	Persona que hace uso del sistema.
Sección	Cada una de las partes en que se divide o considera dividida una empresa, una organización, etc.
Paciente	Persona que padece física y corporalmente, y especialmente quien se halla bajo atención médica



Doctor	Persona legalmente autorizada para profesar y ejercer la medicina.
Archivo	Lugar donde se custodian uno o varios archivos
Turno	Dicho de una persona o de una cosa: A la que corresponde actuar en cierto momento, según la alternativa previamente acordada.
Horario	Tiempo durante el cual se desarrolla habitual o regularmente una acción o se realiza una actividad.
Reservar	Conservar algo para lo futuro.
Código	Combinación de signos que tiene un determinado valor dentro de un sistema establecido.
Especialidad	Rama de una ciencia, arte o actividad, cuyo objeto es una parte limitada de ellas, sobre la cual poseen saberes o habilidades muy precisos quienes la cultivan.
Ventanilla	Abertura pequeña que hay en la pared o tabique de los despachos de billetes, bancos y otras oficinas para que los empleados de estas comuniquen desde dentro con el público que está en la parte de fuera.
Ticket	Boleto: También se le llama así a pases de entrada para espectáculos, boletos para pasar en filas como las del banco, o para acceder a algún servicio en especial.
Dispositivo	Aparato electrónico.
Asistente Administrativo	Persona que hace uso del sistema para la asignación de turnos a los usuarios.



1.4. Referencias

[1] ANSI/IEEE Std. 830-1984 Guía del IEEE para la Especificación de Requerimientos Software.

2. Descripción General

2.1. Perspectiva del sistema

La aplicación ARDAC, es un producto independiente y no forma parte de un sistema mayor, sin embargo es importante mencionar que el sistema solicitará datos como el número de historia clínica de los pacientes de la base de datos existente actualmente en el Hospital, para de esta manera poder realizar tareas como la asignación y reservación de turnos y búsqueda de pacientes.

2.2. Funciones del sistema

El sistema ARDAC consta de las siguientes funciones:

- Asignar de turnos
- Imprimir comprobante del turno asignado
- Reservar turnos
- Liberar (cancelar) turnos.
- Buscar pacientes
- Administrar horario de doctores
- Administrar usuarios
- Administrar doctores
- Administrar especialidades
- Notificar estado de turnos
- Registrar atención de turnos asignados
- Visualizar turnos asignados en un display
- Generar reportes



2.3. Características de los usuarios

Los usuarios que van a utilizar el sistema son:

TABLA IV. CARACTERÍSTICAS USUARIO: ADMINISTRADOR

Usuario	Características
Tipo de usuario	Administrador
Formación	Administrador
Habilidades	<ul style="list-style-type: none">• Capacidad de integración de conocimientos.• Dirigir, controlar y organizar cualquier situación que se pueda presentar.• Juzgar el valor de la información en función del razonamiento.• Trabajar con precisión dentro de los límites, tolerancias o normas establecidas.
Actividades	<ul style="list-style-type: none">• Administrar Usuarios• Generar Reportes• Visualizar turnos asignados y reservados• Administrar Especialidades• Administrar Doctores



TABLA V. CARACTERÍSTICAS USUARIO: ASISTENTE ADMINISTRATIVO

Usuario	Características
Tipo de usuario	Asistentes Administrativo
Formación	Asistente Administrativo
Habilidades	<ul style="list-style-type: none">• Capacidad de adaptación a los cambios• Habilidades comunicativas y escucha activa• Criterio propio para actuar oportunamente y distinguir prioridades• Capacidad de observación, concentración y amplitud de memoria• Tacto y prudencia para manejar situaciones diversas
Actividades	<ul style="list-style-type: none">• Asignar turnos.• Generar reportes de los turnos asignados por el sistema.• Buscar turnos.• Liberar turnos.• Imprimir comprobante del turno.• Administrar un horario doctor-especialidad.



TABLA VI. CARÁCTERÍSTICAS USUARIO: DOCTOR

Usuario	Características
Tipo de usuario	Doctor
Formación	Doctor en Medicina Humana
Habilidades	<ul style="list-style-type: none">• Comunicarse y relacionarse con los demás.• Analizar y sintetizar información a fin de emitir juicios.• Ser creativo, cordial y tener un sentido de responsabilidad absoluta, además de un sentimiento humanitario altamente desarrollado.• Laboriosidad: Optimizar el tiempo disponible para las tareas encomendadas.• Trabajar con precisión dentro de los límites, tolerancias o normas establecidas.
Actividades	<ul style="list-style-type: none">• Visualizar turnos asignados por el sistema.• Registrar la atención del paciente.

2.4. Restricciones

El Hospital Regional Isidro Ayora cuenta con una infraestructura muy adecuada para la atención de sus pacientes, al nivel de instrumentos médicos, la capacitación de su personal es muy buena y eficiente; pero no cuenta con una adecuada distribución de cableado y puntos de red, ocasionando con esto una dificultosa tarea de ubicación del servidor donde se instalará el sistema. A su vez no todos los consultorios de los médicos tratantes del Área de consulta Externa cuenta con un computador en donde se instale el módulo de atención de cada médico, que es parte del sistema ARDAC.



2.5. Suposiciones y Dependencias

El sistema inteligente para la asignación dinámica de turnos en el Área de Consulta Externa del Hospital Regional Isidro Ayora, estará disponible para plataformas implementadas con un sistema operativo Linux.

2.6. Requerimientos Específicos

2.6.1. Interfaces Externas

Interfaces de Usuario

En las interfaces de usuario es donde se toman en cuenta los procesos en los cuales el usuario interactúa con el sistema. Es así que la aplicación se presentará al usuario en función de:

- Sistema de barra de menús
- Sistema de barra de iconos estandarizados
- Sistema de menús contextuales

Todas estas características proporcionan una interfaz amigable al usuario, el cual podrá realizar sus entradas de datos a través del teclado o ratón.

Interfaces de Hardware

El sistema ARDAC cuenta con un dispositivo electrónico (display) para visualizar el turno que será atendido, este a su vez consta de:

- Placa Arduino Duemilanove
- 2 Displays de 7 segmentos
- Controlador

Interfaces de Software

Para la administración, mantenimiento y desarrollo de base de datos se utilizará la herramienta MySQL.



Interfaces de Comunicación

Protocolo TCP/IP: La aplicación establecerá una conexión mediante red con la base de datos.

2.6.2. Requisitos Funcionales

Los requisitos funcionales del sistema de administración de turnos se describirán así:

TABLA VII: REQUERIMIENTOS FUNCIONALES DEL SISTEMA

Referencia	Descripción	Categoría
RF 01	Pedir una clave de acceso al administrador para activarse.	Visible
RF 02	Ordenar los turnos de acuerdo a los doctores designados.	Oculto
RF 03	Mostrar en el display el código del turno.	Visible
RF 04	Administrar el control del dispositivo para definir la lógica del funcionamiento.	Visible
RF 05	Debe permitir visualizar los turnos disponibles de cada doctor.	Visible
RF 06	Administrar usuarios.	Visible



TABLA VIII: REQUERIMIENTOS FUNCIONALES DELASISTENTE ADMINISTRATIVO

Referencia	Descripción	Categoría
RF 07	Asignar un turno a un usuario Paciente.	Visible
RF 08	Presentar un reporte de los turnos generados por el sistema.	Visible
RF 09	Buscar un turno de acuerdo al criterio de búsqueda (nombres del paciente).	Visible
RF 10	Liberar un turno	Visible
RF 11	Imprimir el turno solicitado por el paciente.	Visible
RF 12	Administrar un horario doctor-especialidad (por la especialidad y por el médico tratante)	Visible

TABLA IX: REQUERIMIENTOS FUNCIONALES DEL USUARIO PACIENTE

Referencia	Descripción	Categoría
RF 13	Reservar un turno.	Visible
RF 14	Solicitar turno	Visible



TABLA X: REQUERIMIENTOS FUNCIONALES DEL USUARIO DOCTOR

Referencia	Descripción	Categoría
RF 15	Visualizar turnos asignados por el sistema.	Visible
RF 16	Registrar la atención de un paciente.	Visible

TABLA XI: REQUERIMIENTOS FUNCIONALES DEL USUARIO DIRECTOR

Referencia	Descripción	Categoría
RF 15	Visualizar los turnos solicitados por los usuarios Pacientes.	Visible

2.6.3. Requisitos no Funcionales

TABLA XII: REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA

Código	Atributo	Descripción
RNF 01	Control de acceso	Para poder hacer uso del sistema se tendrá que disponer de una cuenta independiente para cada usuario.
RNF 02	Manejo del sistema	El sistema será amigable, interactivo y sencillo.
RNF 03	Integridad de datos	La información será relacionada entre sí utilizando para esto la integridad referencial en la base de datos.
RNF 04	Interfaz	El sistema estará orientado al manejo de



	gráfica	formularios y cuadros de mensajes. El sistema maximizará la navegación a través del teclado y el ratón.
RNF 05	Rendimiento	El sistema podrá soportar cualquier número de usuarios interactuando con éste en el mismo instante.
RNF 06	Plataforma	El sistema podrá ser implementado en los sistemas operativos de Windows XP, Windows Vista y Linux en sus diferentes versiones
RNF 07	Tiempo de Respuesta	El tiempo de respuesta será máximo de 15 segundos

2.7. Atributos del Sistema

2.7.1. Fiabilidad

Los datos contenidos en el sistema deben ser reales, y la asignación de turnos a los pacientes debe cumplirse sin restricciones.

Además verificar que la información que se ingresa en la base de datos es real.

2.7.2. Disponibilidad

El sistema podrá ser usado únicamente dentro de la jornada laborable del Área de Consulta Externa del Hospital Regional Isidro Ayora, de igual manera la base de datos debe poder utilizarse siempre que se esté en horas laborables y cuando se requiera de ella.



2.7.3. Seguridad

El sistema deberá utilizar usuario y contraseña para acceso al mismo, así mismo la base de datos debe tener una gran seguridad de su información, ya que esta es para uso exclusivo de los administradores y lo que se ingrese en la misma son datos personales del paciente, es por ello que se utilizarán métodos de seguridad como el control de usuarios

2.7.4. Mantenimiento

La instalación del sistema será realizada por los desarrolladores, los mismos que entregarán al administrador del sistema, los manuales de usuario e instalación para que el a futuro por algún motivo o razón ajena deba eliminar o reinstalar el sistema. También se deberá dar un mantenimiento preventivo tanto al sistema como a la base de datos para evitar posibles fallos.

2.7.5. Restricciones de Diseño

El sistema será desarrollado cumpliendo con los estándares establecidos para este tipo de aplicación. Además cumplirá con los estándares y políticas establecidos por el Área de Consulta Externa del Hospital Regional Isidro Ayora.



2.8. Modelado de la aplicación

2.8.1. Modelo del Dominio

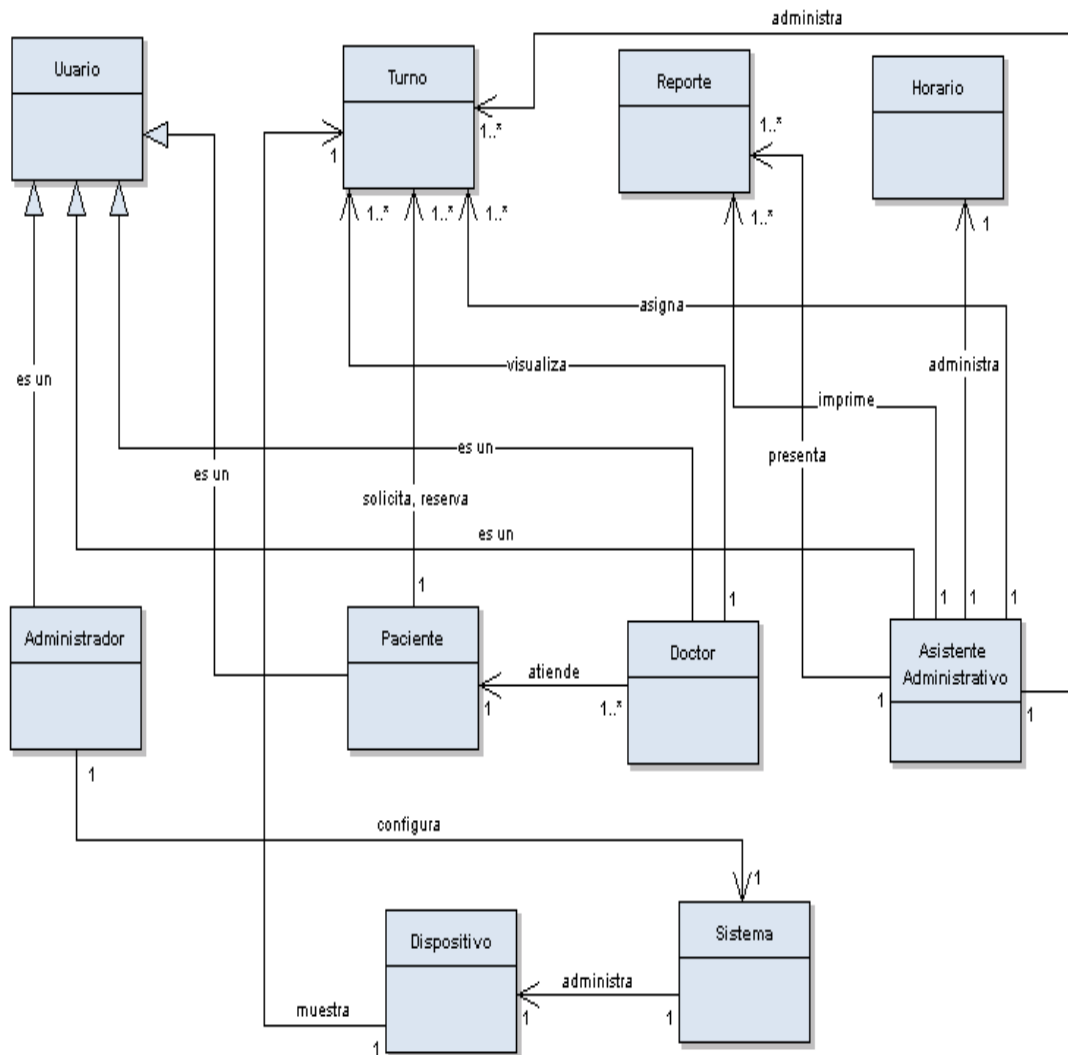


Figura 30: Modelo del Dominio

2.8.2. Diagrama de Casos de Uso

Los diagramas de casos de uso se utilizan para representar las actividades que realizan los usuarios con el sistema.

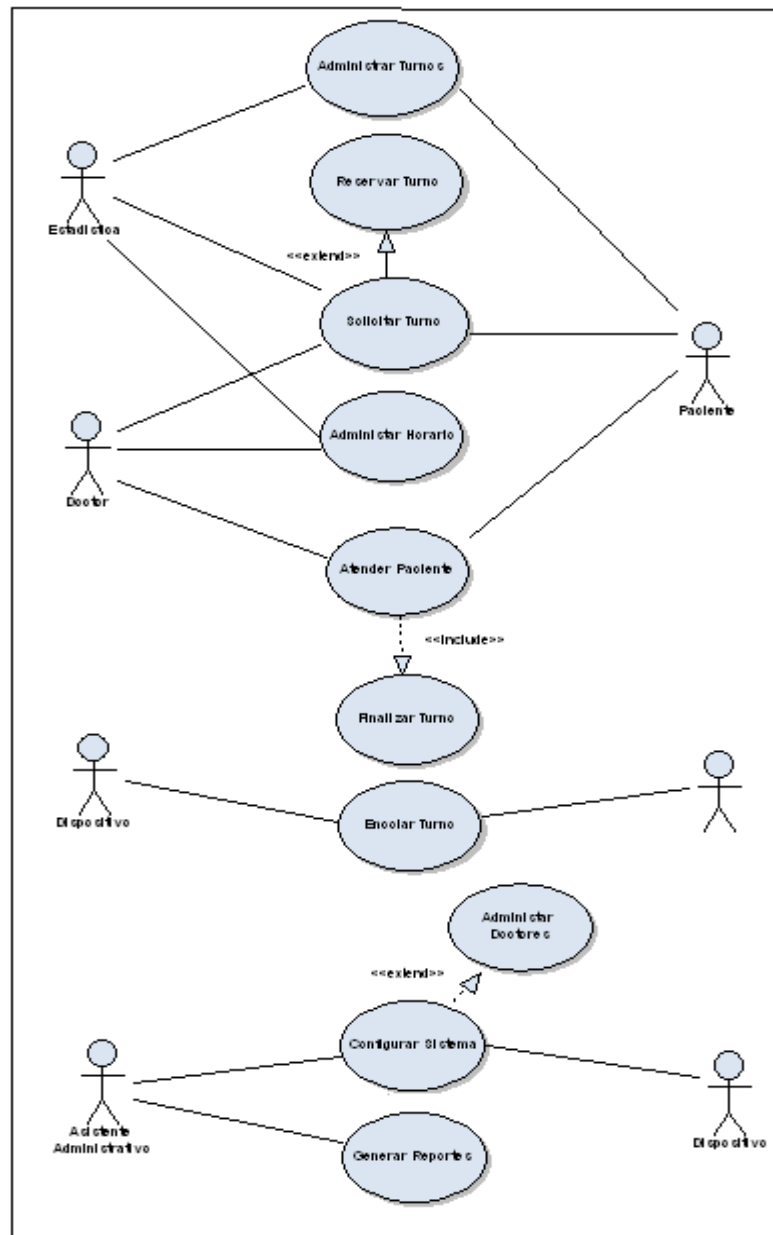


Figura 31: Diagrama de casos de Uso



2.8.2.1. Descripción de los casos de uso

2.8.2.1.1. Ingreso al Sistema

TABLAXIII.
CASO DE USO: INGRESO AL SISTEMA

Nombre de la Pantalla:	Ingreso al sistema
Referencia de requerimientos:	RQF 01
	
Realizado por:	Alba Pineda, Carmen Becerra
Fecha:	

Nombre:	Ingreso al Sistema
Actor(es):	Usuario(iniciador)
Propósito:	Confirmar que el usuario está autorizado para ingresar al sistema.
Descripción Breve:	El usuario deberá ingresar el nombre de Usuario y Password para acceder al sistema. El sistema informa si está autorizado o no.
Tipo:	Primario, esencial
Precondición(es)	El usuario haya iniciado la ejecución del sistema.
Post-condiciones(es)	El usuario accede al sistema con sus respectivas funcionalidades de su cuenta de usuario.



CURSO NORMAL DE EVENTOS	
Usuario (iniciador)	Sistema
<p>1.- Digita los datos en los campos de texto Usuario y Password de la ventana Ingreso al Sistema.</p> <p>2.- Elige la opción Aceptar de la ventana Ingreso al Sistema.</p>	<p>3.- Valida campos obligatorios</p> <p>4.- Autentifica usuario a través de la cuenta.</p> <p>5.- Muestra la ventana Menú Principal.</p> <p>6.- El use case finaliza.</p>
CURSO ALTERNO DE EVENTOS	
Curso Alterno A: Elige opción <i>Cancelar</i>	
<p>A2.- Elige la opción Cancelar de la ventana Ingreso al Sistema.</p>	<p>A3.- Cierra la ventana Ingreso al Sistema.</p> <p>A4.- El sistema se cierra.</p> <p>A5.- El use case finaliza.</p>



DIAGRAMA DE SECUENCIA: INGRESO AL SISTEMA

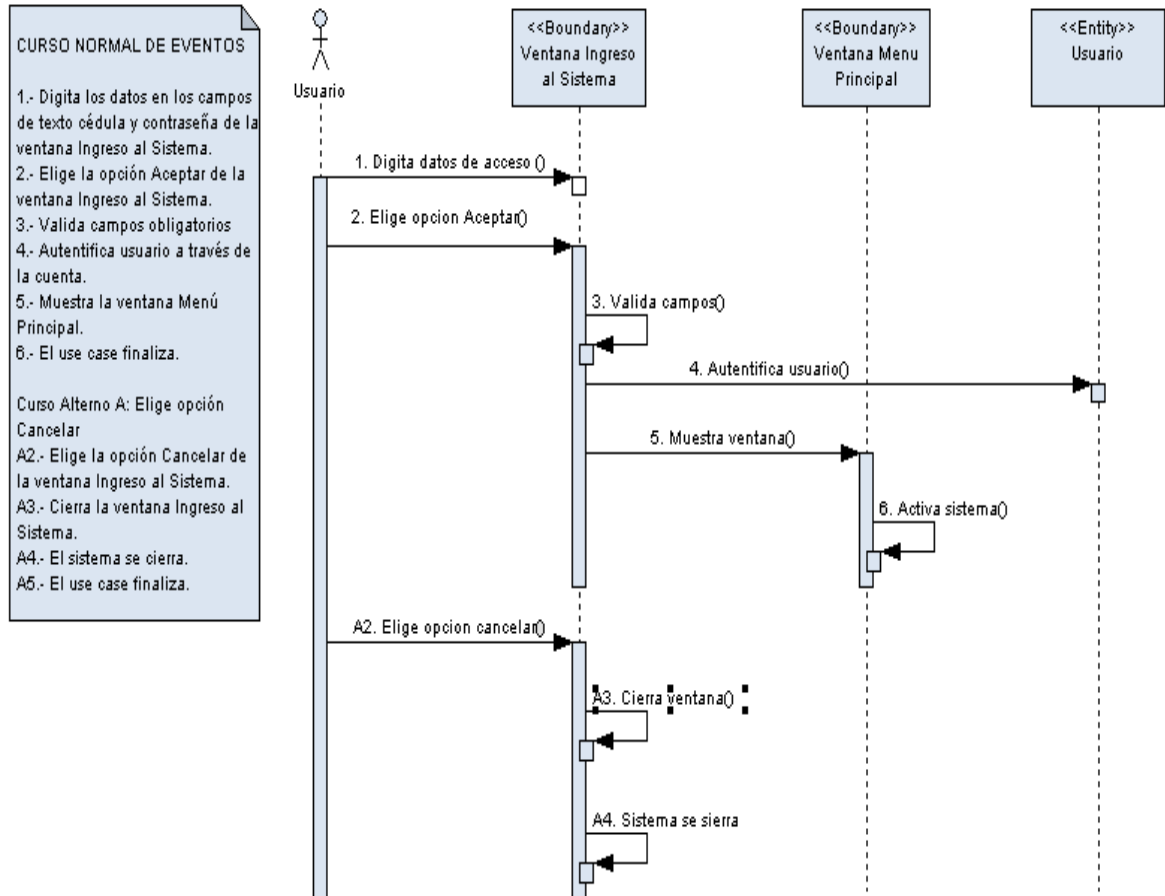



Figura 32: Diagrama de Secuencia: Ingreso al Sistema.



2.8.2.1.2. Solicitar Turno

TABLA XIV.
CASO DE USO: SOLICITAR TURNO

Nombre de la Pantalla:	Solicitar Turno
Referencia de requerimientos:	RQF 05, RQF7, RQF9, RQF11, RQF12
	
Realizado por:	Alba Pineda, Carmen Becerra
Fecha:	

Nombre:	Solicitar Turno
Actor(es):	Asistente Administrativo
Propósito:	Proporcionar un turno de atención médica al paciente.
Descripción Breve:	El sistema permite al asistente administrativo reservar un turno de atención médica para el paciente ingresado.
Tipo:	Primario, esencial



Precondición(es)	<p>-El Asistente Administrativo debe haber ingresado al sistema por medio de su usuario y contraseña personal digitada y validada en la pantalla de ingreso.</p> <p>-El Asistente Administrativo debe haber elegido la opción Solicitar Turno de la pantalla Menú Principal.</p>
Post-condiciones(es)	El sistema debe generar un turno de atención médica y notificar al Doctor al que se le ha asignado dicho turno.
CURSO NORMAL DE EVENTOS	
Asistente Administrativo	Acciones del Sistema
<p>5. Escoge la fecha para la cual se asignará el turno en el calendario Fecha</p> <p>6.- Ingresar el nombre y apellido del paciente en el campo Paciente de la ventana Solicitar Turno.</p> <p>7.- Selecciona la especialidad de la lista desplegable Especialidad de la ventana Solicitar Turno.</p>	<p>1.- Carga la fecha actual en el calendario Fecha.</p> <p>2.- Busca las especialidades en el repositorio de datos especialidad.</p> <p>3.- Carga las especialidades existentes en la lista desplegable Especialidad de la ventana Solicitar Turno.</p> <p>4.- Carga la hora de inicio de la atención en los campos hora de la ventana Solicitar Turno.</p> <p>8.- Busca en el repositorio de datos doctor los doctores correspondientes a la especialidad seleccionada.</p> <p>9.- Carga el nombre de los doctores en la lista Doctores de la ventana Solicitar</p>



<p>10.- Selecciona un doctor de la lista Doctores de la ventana Solicitar Turno.</p> <p>11.- Escoge la hora disponible para la cual se asignará el turno en la lista Hora.</p> <p>16.- Elige la opción Solicitar de la ventana Solicitar Turno.</p>	<p>Turno.</p> <p>12.- Valida que se haya seleccionada una hora disponible para la asignación del turno.</p> <p>13.- Valida que el campo Paciente de la ventana Solicitar Turno este lleno.</p> <p>14.- Valida que se haya seleccionado una opción de las lista Especialidad de la ventana Solicitar Turno.</p> <p>15.- Valida que se haya seleccionado una opción de las lista Doctores de la ventana Solicitar Turno.</p> <p>17.- Guarda la transacción en el repositorio de datos turnos.</p> <p>18.- Presenta un mensaje de aviso "Turno Asignado".</p> <p>19.- El use case finaliza.</p>
<p>CURSO ALTERNO DE EVENTOS</p>	
<p>Curso Alterno A: Campos Vacíos</p>	
<p>A17.- El use case continúa en el paso 6 del curso normal de eventos.</p> <p>A18.- El use case continúa en el paso 16 del curso normal de eventos.</p>	<p>A16.- Muestra un mensaje de error "Existen campos vacíos"</p>
<p>Curso Alterno B: Opciones no seleccionadas</p>	
	<p>B16.- Muestra un mensaje de error</p>



B17.- El use case continúa en el paso 7 del curso normal de eventos.	"Debe seleccionar una Especialidad de la lista".
Curso Alterno C: Opciones no seleccionadas	
C17.- El use case continúa en el paso 10 del curso normal de eventos.	C16.- Muestra un mensaje de error "Debe seleccionar un Doctor de la lista Doctores".
Curso Alterno D: Elección de la opción Buscar	
D6.- Elige la opción Buscar de la ventana Solicitar Turno . D8.- Ingresa el nombre y apellido en el campo Paciente de la ventana Buscar Paciente . D11.- Selecciona el paciente deseado de la tabla Pacientes de la ventana Buscar Paciente . D12.- Elige la opción Aceptar de la ventana Buscar Paciente .	D7.- Muestra la ventana Buscar Paciente . D9.- Busca datos en el repositorio de datos paciente . D10.- Filtra y presenta la información de los pacientes la tabla Pacientes de la ventana Buscar Paciente . D13.- Cierra la ventana Buscar Paciente . D14.- Carga el nombre y apellido del paciente seleccionado en el campo Paciente de la ventana Solicitar Turno .



D15.-El use continúa en el paso 7 del curso normal de eventos.	
Curso Alterno del Alterno D.A: Mensaje de aviso	
D.A.11.- El use case continua en el paso 6 del curso normal de eventos.	D.A.10.- Muestra mensaje de aviso “el paciente ingresado no existe”
Curso Alterno del Alterno D.B: Elige opción Cancelar de la ventana Buscar Paciente .	
D.B.12.- Elige la opción Cancelar de la ventana Buscar Paciente .	D.B.13.- Cierra la ventana Buscar Paciente .
D.B.14. El continúa en el paso 6 del curso normal de eventos.	
Curso Alterno E: Mensaje de aviso	
	E13.- Muestra mensaje de aviso “ La hora seleccionada no está disponible” E14.- El use case continua en el paso 11 del curso normal de eventos.
Curso Alterno F: Elige opción Cancelar de la ventana Solicitar Turno .	
F16.- Elige la opción CancelarTurno de la ventana Solicitar Turno .	F17.- Limpia los campos de la ventana Solicitar Turno . F18.- El use case continua en el paso 6 del curso normal de eventos.
Curso Alterno G: Elige opción Salir de la ventana Solicitar Turno .	
G16.- El asistente administrativo elige la opción Salir de la ventana Solicitar Turno .	G17.- Cierra la ventana Solicitar Turno



	G18.- Muestra la ventana Menú Principal .
	G19.- El use case Finaliza.

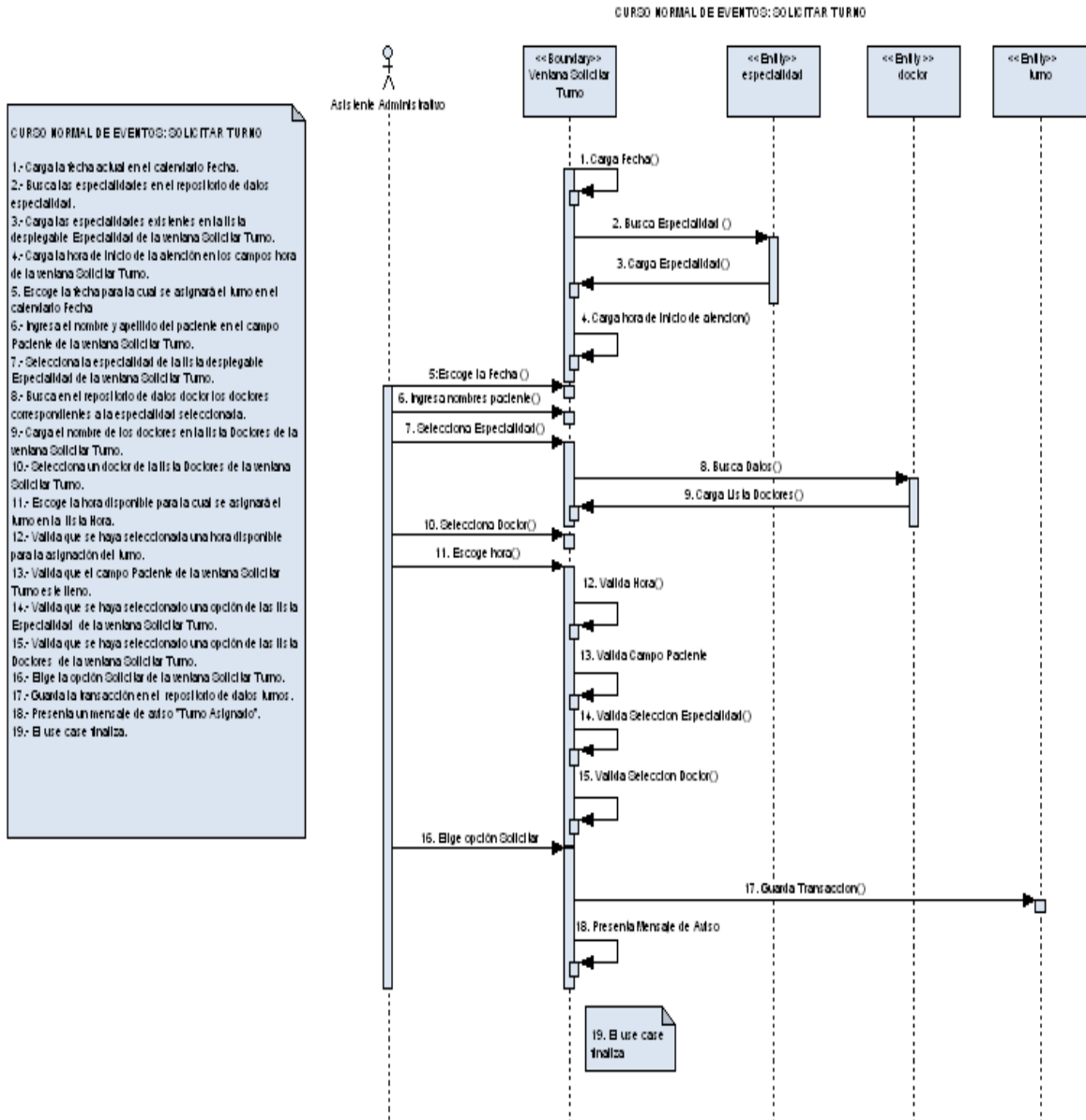


Figura 33: Diagrama de Secuencia: Solicitar Turno

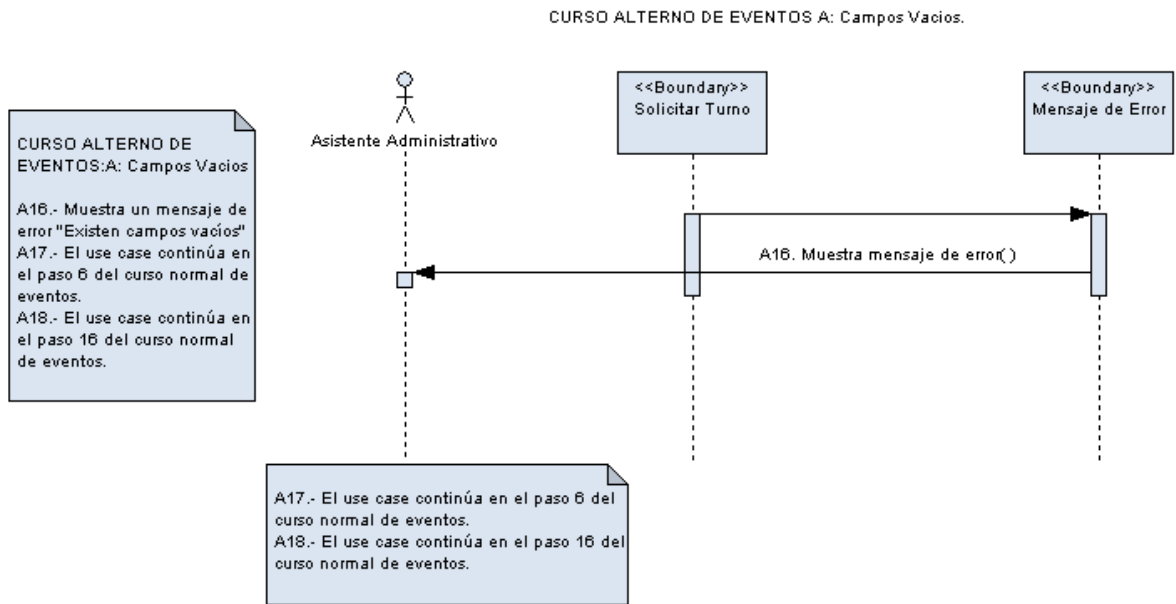


Figura 34: Diagrama de Secuencia: Curso Alterno A: Campos Vacíos.

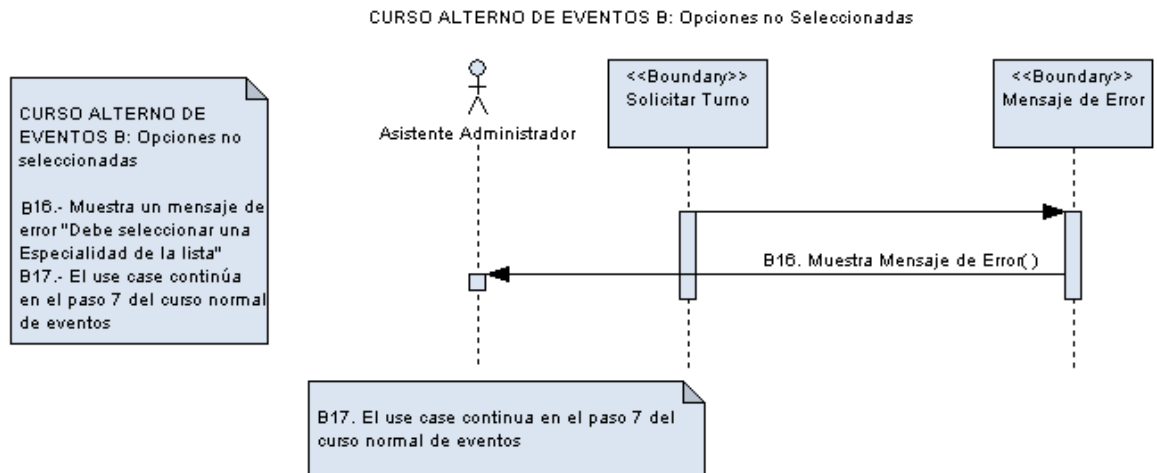


Figura 35: Diagrama de Secuencia: Curso Alterno B: Opciones no Seleccionadas.



CURSO ALTERNO DE EVENTOS C: Opciones no Seleccionadas

CURSO ALTERNO DE EVENTOS C: Opciones no seleccionadas

C16.- Muestra un mensaje de error "Debe seleccionar un Doctor de la lista Doctores".

C17.- El use case continúa en el paso 10 del curso normal de eventos.

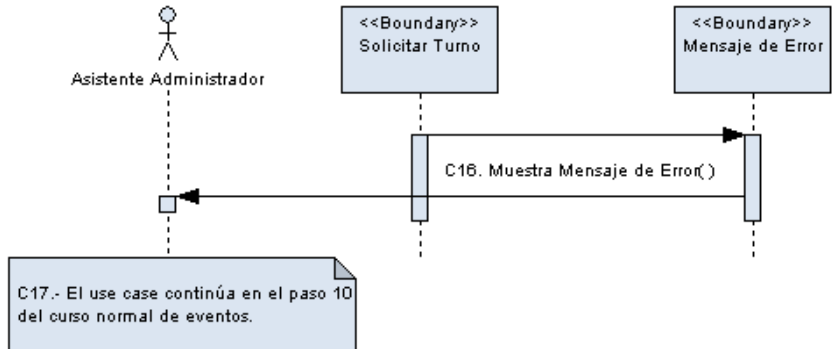


Figura 36: Diagrama de Secuencia: Curso Alterno C: Opciones no Seleccionadas.

CURSO ALTERNO DE EVENTOS D: Elección de la opción Buscar

CURSO ALTERNO DE EVENTOS D: Elección de la opción Buscar

D6.- Elige la opción Buscar de la ventana Solicitar Turno.

D7.- Muestra la ventana Buscar Paciente.

D8.- Ingresar el nombre y apellido en el campo Paciente de la ventana Buscar Paciente..

D9.- Busca datos en el repositorio de datos paciente.

D10.- Filtra y Presenta la información de los pacientes presentándola en una tabla Pacientes de la ventana Buscar Paciente.

D11.- Selecciona el paciente deseado de la tabla Pacientes de la ventana Buscar Paciente.

D12.- Elige la opción Aceptar de la ventana Buscar Paciente.

D13.- Cierra la ventana Buscar Paciente.

D14.- Carga el nombre y apellido del paciente seleccionado en el campo Paciente de la ventana Solicitar Turno.

D15.- El use case continúa en el paso 7 del curso normal de eventos.

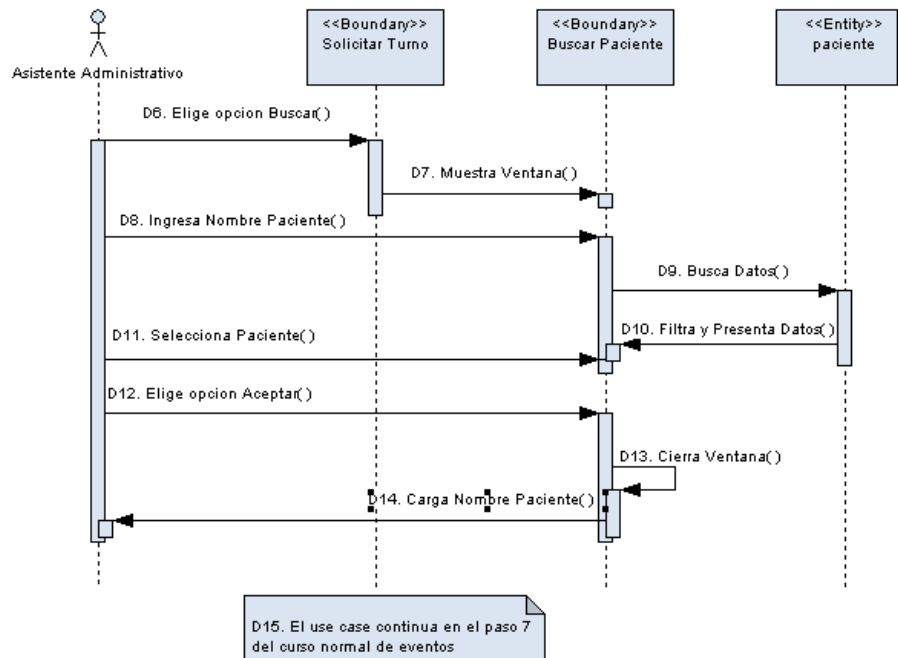


Figura 37: Diagrama de Secuencia: Curso Alterno D: Elección de la opción Buscar.

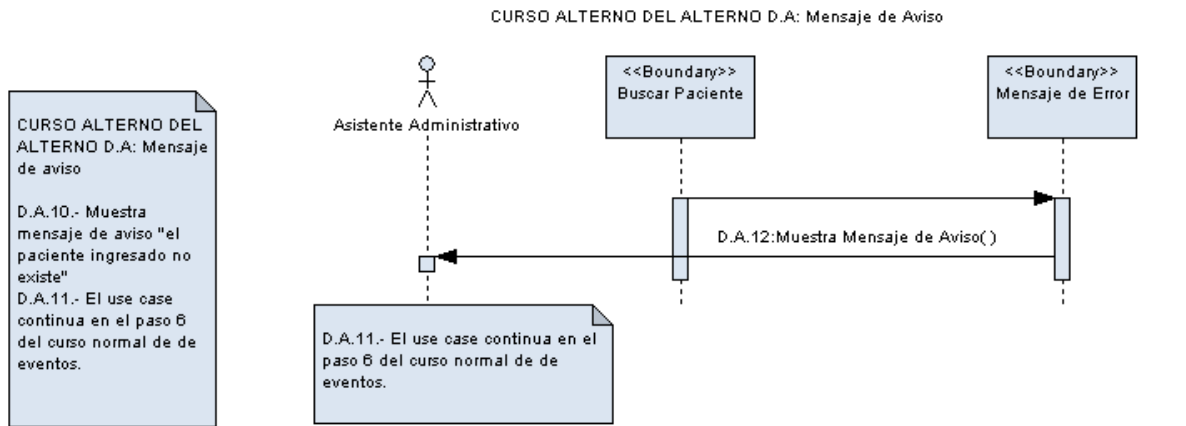


Figura 38: Diagrama de Secuencia: Curso Alterno del Alterno D.A: Mensaje de Aviso.

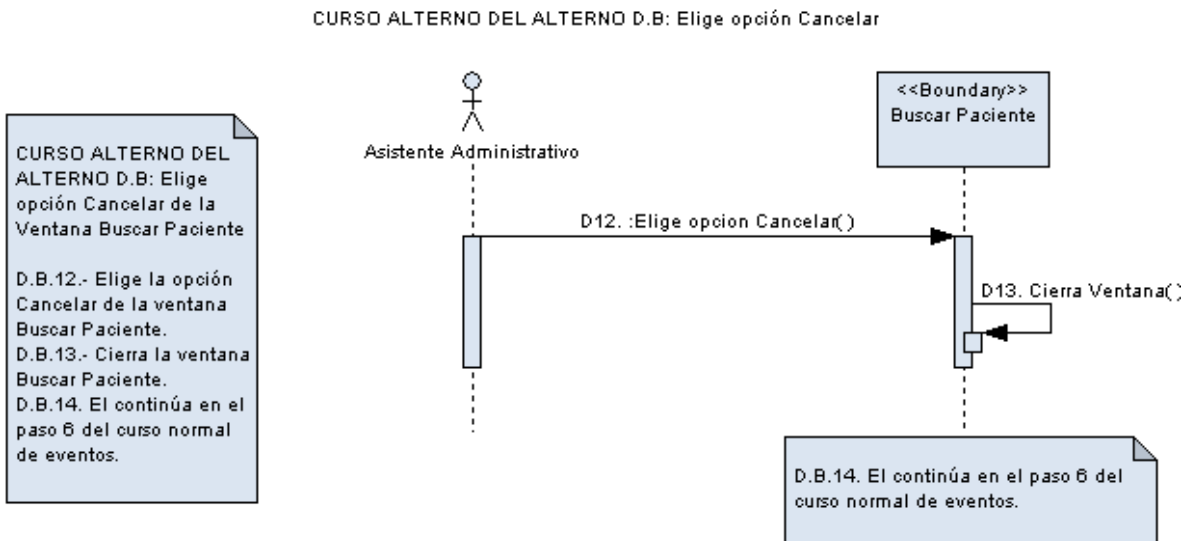


Figura 39: Diagrama de Secuencia: Curso Alterno del Alterno D.B: Elige opción Cancelar.

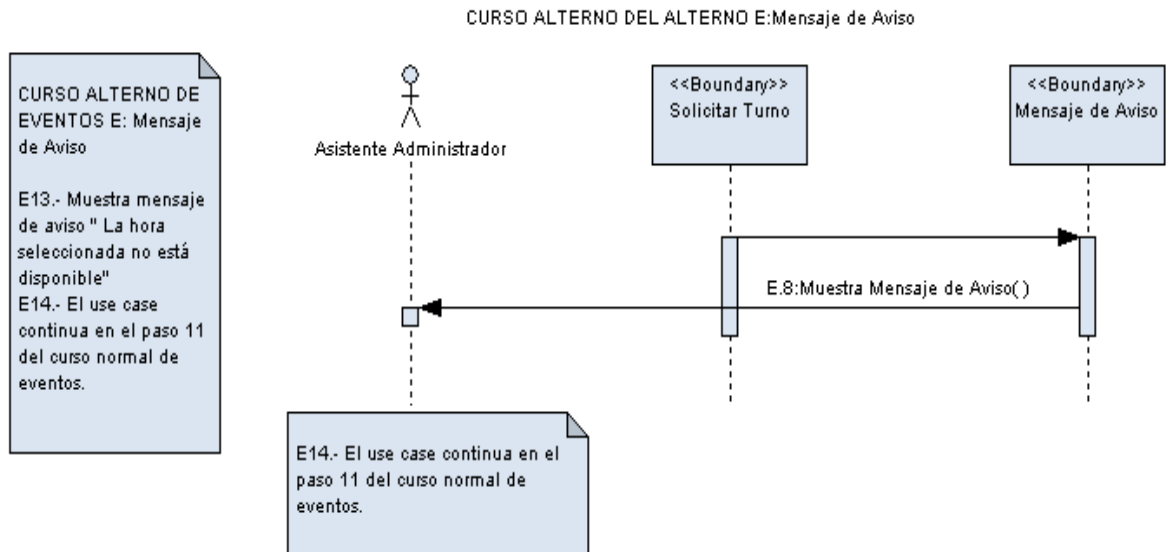


Figura 40: Diagrama de Secuencia: Curso Alternativo E: Mensaje de Aviso.

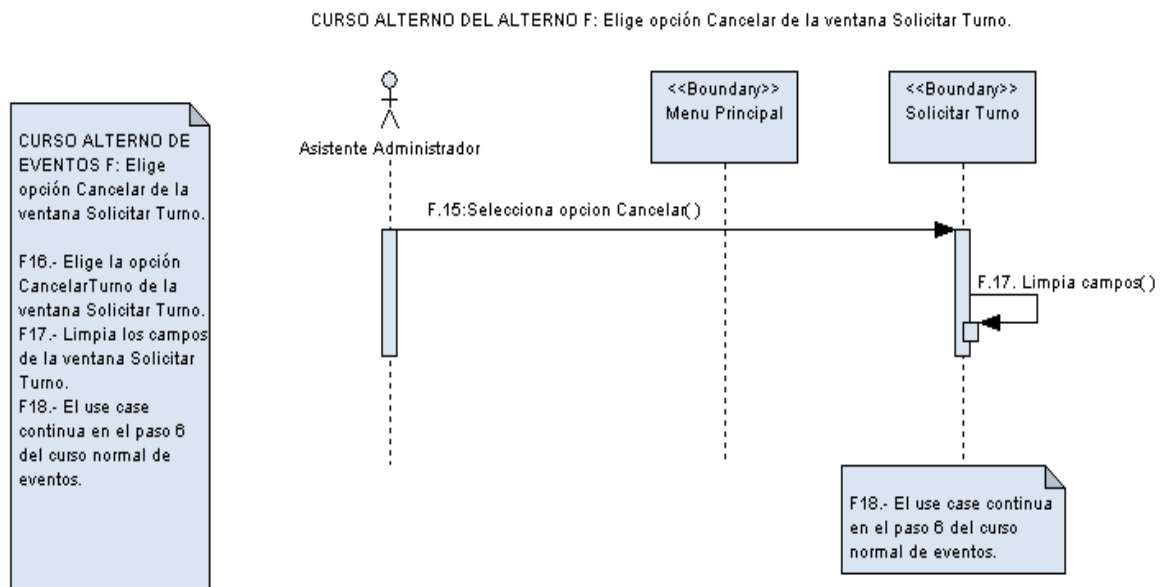


Figura 41: Diagrama de Secuencia: Curso Alternativo F: Elige opción Cancelar.



2.8.2.1.3. Administración de Turnos

TABLA XV.
CASO DE USO: ADMINISTRACIÓN DE TURNOS

Nombre de la Pantalla:	Administración de Turnos
Referencia de requerimientos:	RQF 08
Realizado por:	Alba Pineda, Carmen Becerra
Fecha:	

Nombre:	Administración de turnos
Actor(es):	Asistente Administrativo
Propósito:	Visualizar la lista diaria y posfecha de turnos de atención que se han asignado, y permitir la cancelación y activación de los mismos.
Descripción Breve:	El sistema permite al asistente administrativo cancelar y activar turnos.



Tipo:	Primario, esencial
Precondición(es)	El asistente administrativo debe haber ingresado al sistema por medio de su usuario y contraseña personal digitada y validada en la pantalla de ingreso, y elegido la opción Administración de Turnos de la pantalla Menú Principal .
Post-condiciones(es)	El sistema debe generar la lista de turnos que se han asignado en el día y aquellos que han sido reservados.
CURSO NORMAL DE EVENTOS	
Asistente Administrativo	Acciones del Sistema
4.- Selecciona una fila de la tabla Listado de turnos correspondiente al turno que desea cancelar de la ventana Administración de Turnos .	1.- Carga la fecha actual en el calendario Fecha .
5.- Elige la opción Cancelar Turno de la ventana Administración de Turnos .	2.- Busca en el repositorio de datos turno los turnos asignados en la fecha elegida.
	3.- Carga la tabla Listado de turnos con los turnos que se han asignado en la fecha elegida en la ventana Administración de Turnos .
	6- Actualiza datos en el repositorio de datos turno .
	7.- Notifica al doctor en la ventana Turnos Doctor sobre el estado actual del turno.
	8.- Notifica al asistente administrativo sobre el estado actual del turno en la tabla Listado de Turnos de la ventana Administración de



9.- Elige la opción Salir de la ventana Administración de Turnos .	Turnos. 10.- Cierra la ventana Administración de Turnos 11.- El use case finaliza.
CURSO ALTERNO DE EVENTOS	
Curso alternativo A: Elige la opción Activar Turno	
A.5. Elige la opción Activar Turno de la ventana Administración de Turnos	A.6.- El use case continua en el paso 6 del curso normal de eventos.
Curso alternativo B: Selección de Fecha	
B4.- Selecciona una fecha posterior a la fecha actual en el calendario Fecha de la ventana Administración de Turnos .	B.5.- El use case continua en el paso 2 del curso normal de eventos.
Curso alternativo C: Elige la opción Salir	
C5.- Elige la opción Salir de la ventana Administración de Turnos .	C6.- Cierra la venta Administración de Turnos . C7.- El use case finaliza.

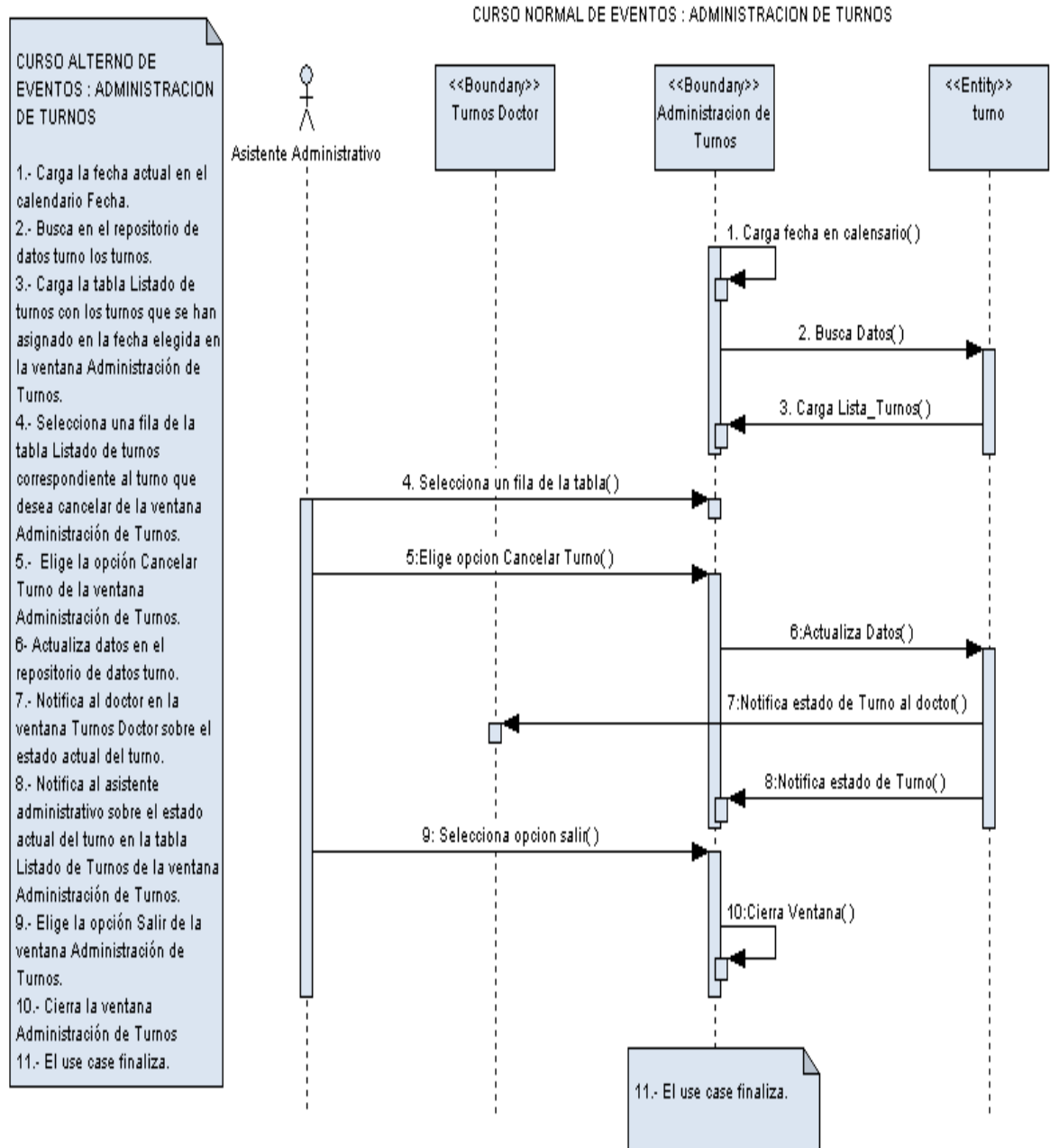


Figura 42: Diagrama de Secuencia: Administración de Turnos



CURSO ALTERNO DE EVENTOS A: Elige la opción Activar Turno

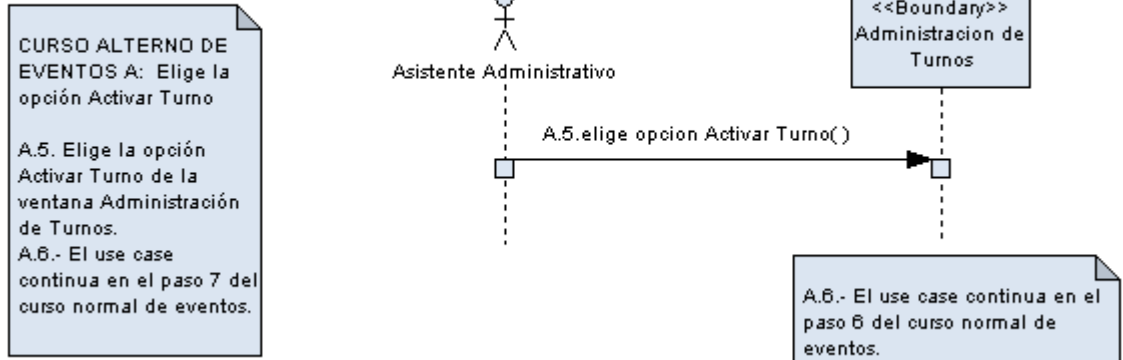


Figura 43: Diagrama de Secuencia: Curso Alternativo A: Elige opción Activar Turno.

CURSO ALTERNO DE EVENTOS B: Selección de Fecha

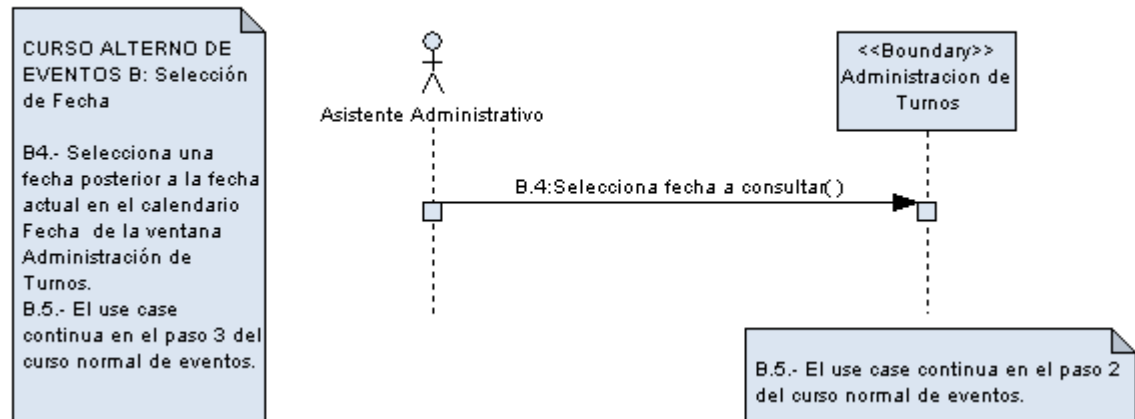


Figura 44: Diagrama de Secuencia: Curso Alternativo B: Selección de Fecha.

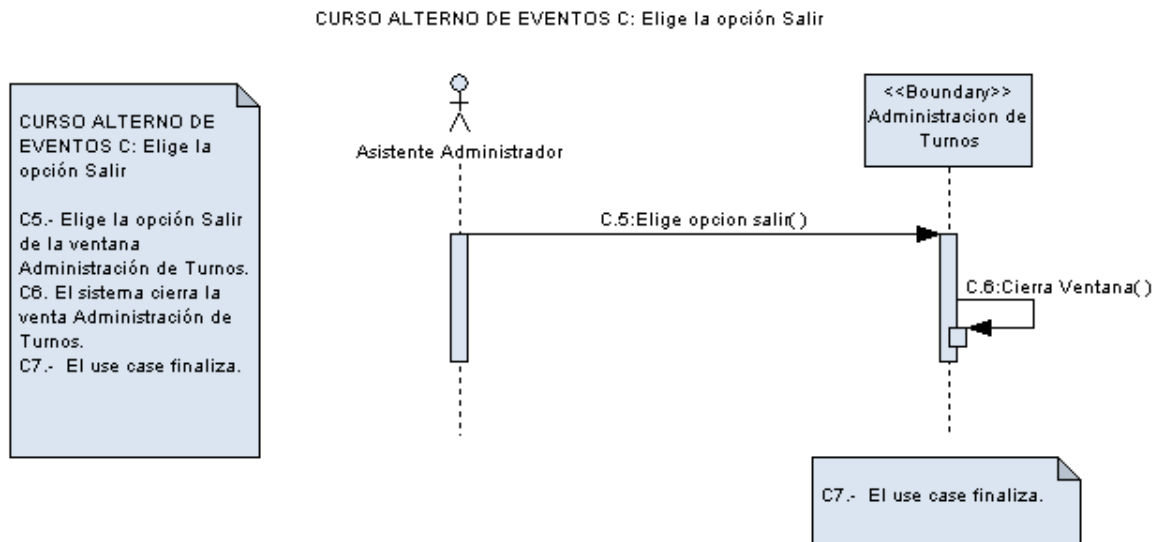


Figura 45: Diagrama de Secuencia: Curso Alterno C: Elige la opción Salir.



2.8.2.1.4. Administrar Horarios

TABLA XVI.
CASO DE USO: ADMINISTRAR HORARIO

Nombre de la Pantalla:	Administrar Horario
Referencia de requerimientos:	RQF 10
Realizado por:	Alba Pineda, Carmen Becerra
Fecha:	

Nombre:	Administrar Horarios
Actor(es):	Asistente Administrativo
Propósito:	Administrar un horario de atención para cada médico tratante.
Descripción Breve:	El sistema permite al asistente administrativo visualizar e ingresar el horario de atención de cada médico.
Tipo:	Primario, esencial
Precondición(es)	- El Asistente Administrativo debe haber ingresado al sistema por medio de su contraseña personal digitada y validada en la pantalla de ingreso.



Post-condiciones(es)	El sistema debe generar una tabla indicando la fecha y hora de atención (hora de inicio, hora de culminación) existentes para cada médico.
CURSO NORMAL DE EVENTOS	
Asistente Administrativo	Acciones del Sistema
<p>2.- Escoge el nombre del doctor del cual desea consultar el horario de atención de la lista desplegable Nombre del Doctoren la ventana Administrar Horarios.</p> <p>5. Escoge la opción Salir de la ventana Administrar Horarios.</p>	<p>1.- Carga los nombres de los doctores en una lista desplegable Nombre del Doctoren la ventana Administrar Horarios.</p> <p>3.- Busca en el repositorio de datos horarios, los datos correspondientes al horario del doctor seleccionado.</p> <p>4.- Carga en la tabla Horario_ Doctor, la fecha y hora (inicio-fin) de trabajo del doctor seleccionado.</p> <p>6.- Cierra la ventana Administrar Horarios.</p> <p>7.- El use case finaliza.</p>
CURSO ALTERNO DE EVENTOS	
Curso Alterno A: Escoge la opción Agregar	
A5. Escoge la opción Agregar en la ventana Administrar Horarios .	<p>A6.- Muestra la ventana Agregar Horario.</p> <p>A7.- Carga los datos correspondientes en las listas desplegables (fecha, hora de inicio y hora de finalización).</p>



<p>A8.- Escoge la fecha, hora de inicio y hora de finalización de trabajo laboral del médico que desea agregar en la ventana Agregar Horario.</p> <p>A9.- Escoge la opción Agregar en la ventana Agregar Horario.</p>	<p>A10.- Cierra automáticamente la ventana Agregar Horario.</p> <p>A11.- Actualiza los datos en el repositorio doctor.</p> <p>A12.- Carga la tabla Horario_ Doctor con los datos actualizados en la ventana Administrar Horarios.</p> <p>A13.- El use case continua en el paso 5 del curso normal de eventos.</p>
<p>Curso alternativo del alternativo A.A.: Mensaje de error.</p>	
	<p>A.A.5.- Muestra un mensaje de error “Debe seleccionar un Doctor para agregar el horario de atención.</p> <p>A.A.6.- El use case continua en el paso 2 del curso normal de eventos.</p>
<p>Curso alternativo del alternativo A.B.: Elección de la opción Salir</p>	
<p>A.B.5.- Elige la opción Salir de la ventana Agregar Horario.</p>	<p>A.B.6.- Cierra la venta Agregar Horario.</p> <p>A.B.7.- El use case continua en el paso 5 del curso normal de eventos.</p>
<p>Curso Alternativo B: Elección de la opción Borrar</p>	



<p>B.5.- Escoge la fila de la tabla horario-doctor que desea borrar de la ventana Agregar Horario.</p> <p>B.6.- Escoge la opción Borrar de la ventana Agregar Horario.</p> <p>B.9.- Escoge la opción si del mensaje de aviso.</p>	<p>B.7.- Comprueba si en el horario a eliminar no existen turnos asignados</p> <p>B.8.- Presenta un mensaje de aviso indicando si está seguro de eliminar el dato seleccionado.</p> <p>B10.- Elimina los datos en el repositorio de datos horarios.</p> <p>B.10.- Actualiza los datos en el repositorio horarios.</p> <p>B.11.- El use case continua en el paso 5 del curso normal de eventos.</p>
<p>Curso Alterno BA: Mensaje de Aviso</p>	
<p>B.A.9.- El use case continua en el paso 5 del curso normal de eventos.</p>	<p>B.A.8.- Presenta un mensaje de aviso indicando "El horario no se puede eliminar, tiene turnos asignados"</p>



CURSO NORMAL DE EVENTOS: ADMINISTRAR HORARIOS

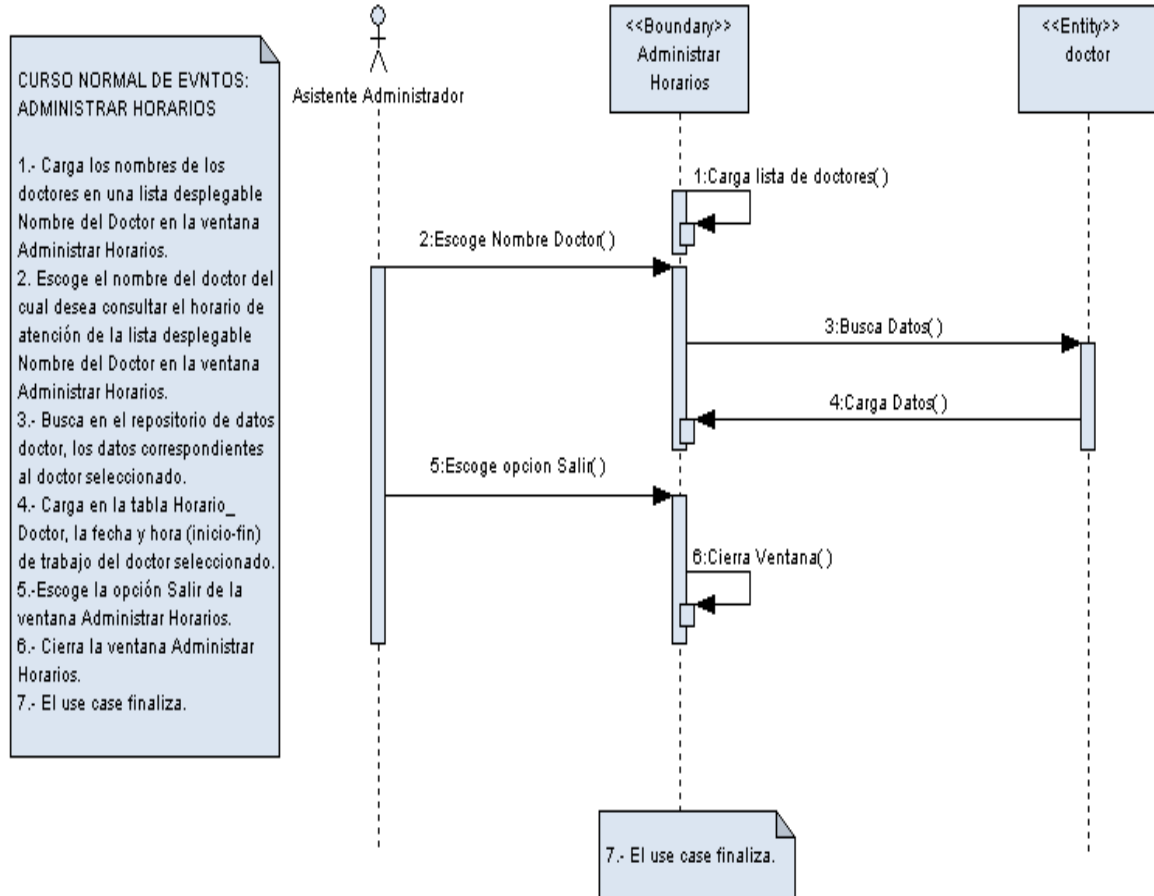


Figura 46: Diagrama de Secuencia: Administrar Horario

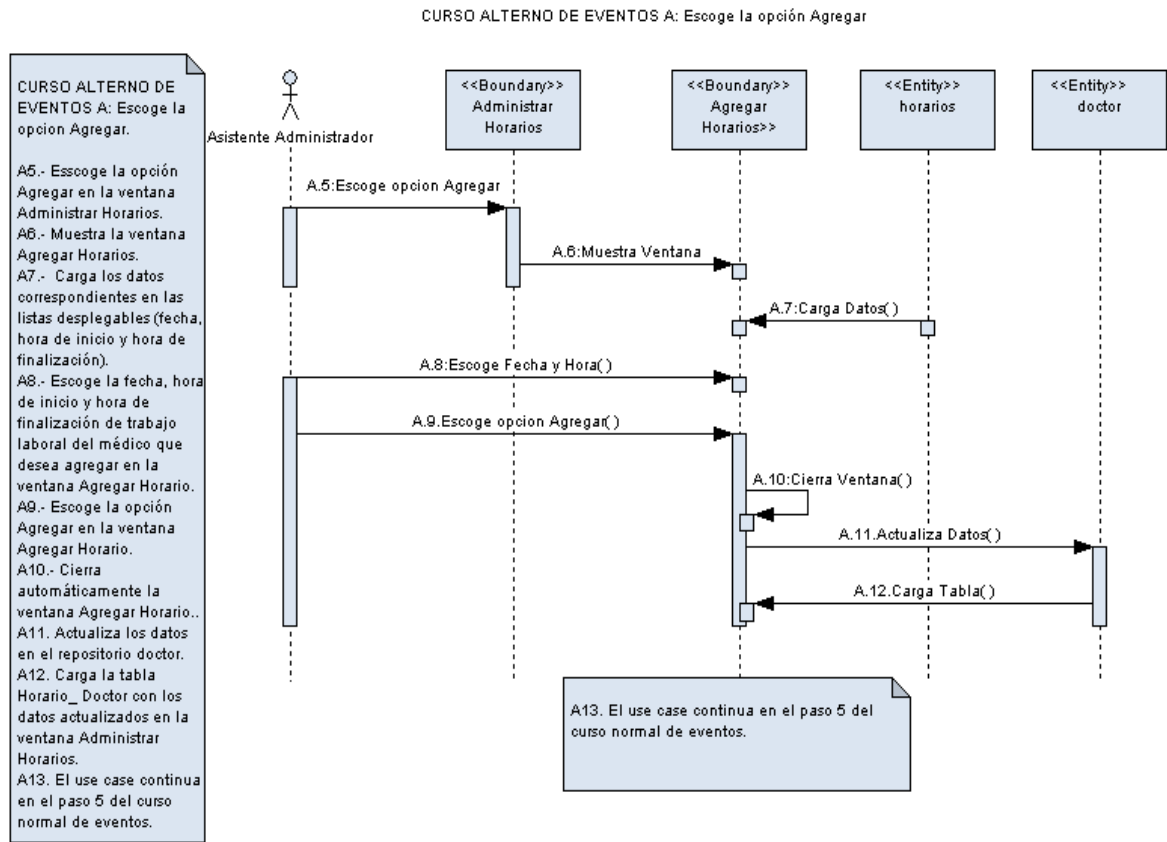


Figura 47: Diagrama de Secuencia: Curso Alterno A: Elige la opción Agregar.

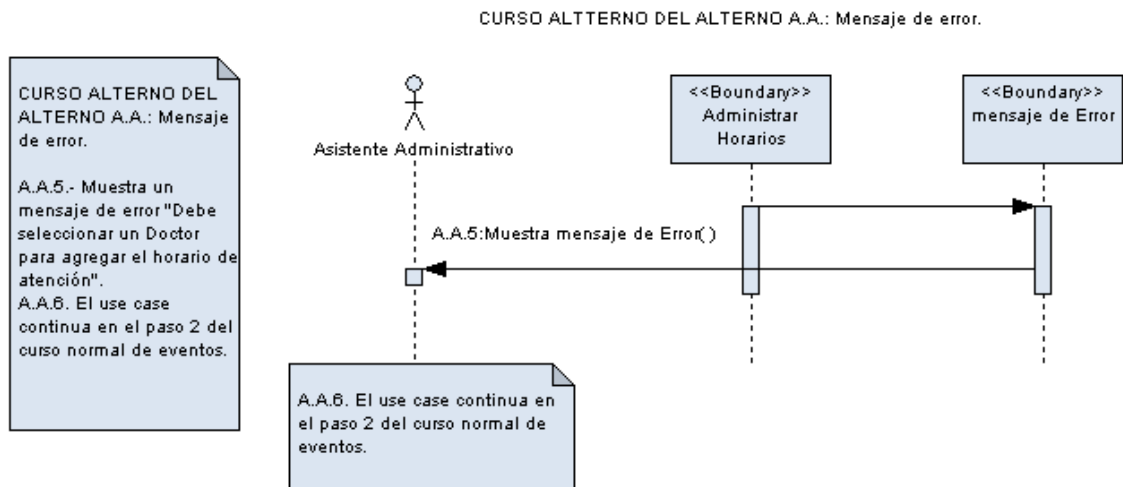


Figura 48: Diagrama de Secuencia: Curso Alterno del Alterno A.A: Mensaje de Error.



CURSO ALTERNO DEL ALTERNO A.B.: Elección de la opción Salir

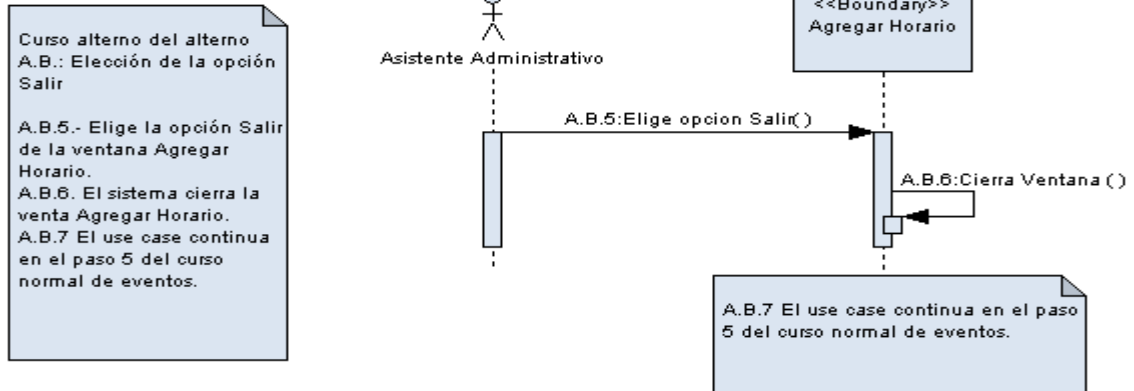


Figura 49: Diagrama de Secuencia: Curso Alternativo del Alternativo A.B: Elección de la opción Salir.

CURSO ALTERNO DE EVENTOS B: Elección de la opción Borrar

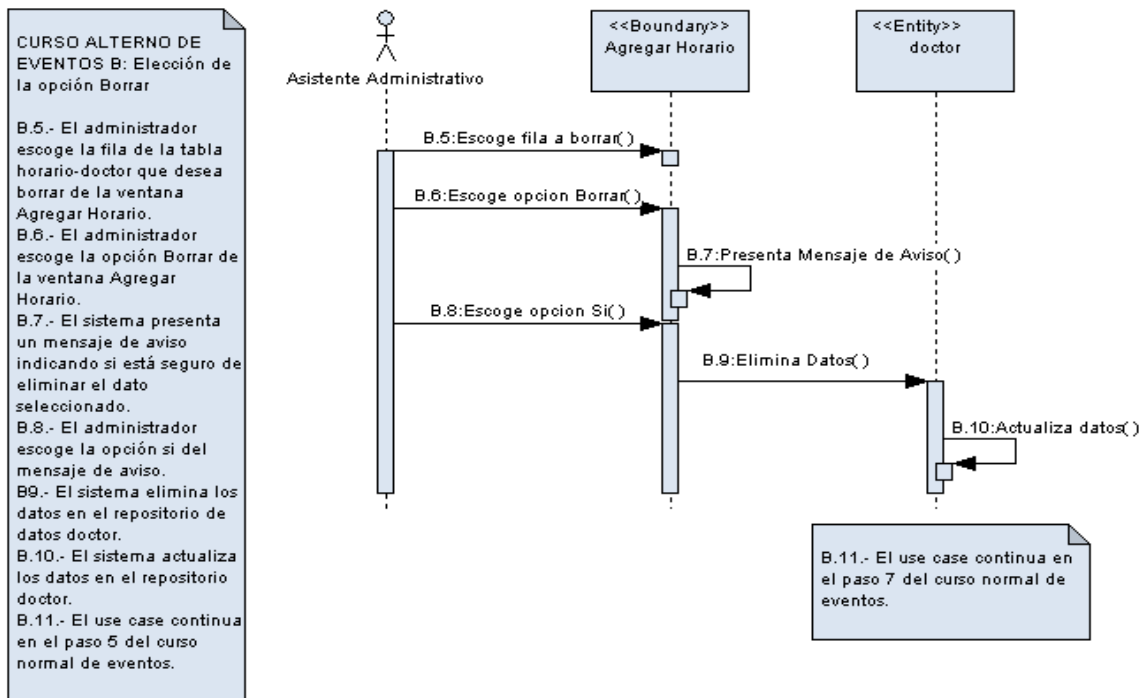


Figura 50: Diagrama de Secuencia: Curso Alternativo B: Elección de la opción Borrar.



2.8.2.1.5. Administrar Doctor-Especialidad

TABLA XVII.

CASO DE USO: ADMINISTRAR DOCTOR-ESPECIALIDAD

Nombre de la Pantalla:	Administrar Doctor-Especialidad
Referencia de requerimientos:	RQF 10
Realizado por:	Alba Pineda, Carmen Becerra
Fecha:	

Nombre:	Administrar Doctor-Especialidad
Actor(es):	Asistente Administrativo
Propósito:	Ingresar datos de nuevos doctores y especialidades, permitiendo además crearlos y eliminarlos.
Descripción Breve:	El sistema permite al asistente administrativo ingresar datos de nuevos doctores y especialidades así como modificarlos y eliminarlos.
Tipo:	Primario, esencial
Precondición(es)	El asistente administrativo debe haber ingresado al sistema por medio de su contraseña personal digitada y validada en la pantalla de ingreso.



Post-condiciones(es)	Creación, modificación y eliminación de datos de doctores y especialidades.
CURSO NORMAL DE EVENTOS	
Asistente Administrativo	Acciones del Sistema
1.- Elige la opción Administrar Doctor-Especialidad de la ventana Menú Principal .	2.- Muestra la ventana Administrar Doctor-Especialidad
3.- Elige la opción Agregar Doctor de la ventana Administrar Doctor-Especialidad .	4.- Muestra la ventana Agregar Doctor .
5.- Digita los datos en los campos correspondientes (cedula, apellidos, nombres)de la ventana Agregar Doctor .	7.- Valida los datos ingresados (cedula, apellidos, nombres)
6.- Elige la opción Aceptar de la ventana Agregar Doctor .	8.- Guarda los datos en el repositorio de datos doctor .
11.- Selecciona el nombre del doctor ingresado en la lista Doctores ventana Administrar Doctor-Especialidad .	9.- Cierra automáticamente la ventana Agregar Doctor .
12.- Selecciona una especialidad de la lista especialidades de la ventana Administrar Doctor-Especialidad .	10.- Muestra la ventana Administrar Doctor-Especialidad .
13.- Escoge la opción Incluir	



<p>Especialidad de la ventana Administrar Doctor-Especialidad.</p> <p>15.- Escoge la opción Salir de la ventana Administrar Doctor-Especialidad.</p>	<p>14. Carga la especialidad selecciona en la lista Especialidades del doctor de la ventana Administrar Doctor-Especialidad.</p> <p>16.- El use case finaliza.</p>
<p>CURSO ALTERNO DE EVENTOS</p>	
<p>Curso Alterno A: Elección de la opción Eliminar Doctor</p>	
<p>A3.- Selecciona un doctor de la lista Doctor de la ventana Administrar Doctor-Especialidad.</p> <p>A4.- Selecciona la opción Eliminar Doctor de la ventana Administrar Doctor-Especialidad.</p> <p>A7.- Elige la opción Si del mensaje de confirmación</p>	<p>A5.- Comprueba que el doctor a eliminar no tenga un historial de turnos asignados.</p> <p>A6- Muestra un mensaje de confirmación de que si está seguro de querer eliminar el dato del sistema.</p> <p>A8.- Elimina los datos del doctor correspondiente en el repositorio de datos de datos doctor.</p> <p>A9- Muestra la ventana Administrar Doctor-Especialidad.</p> <p>A10.- El use case finaliza.</p>
<p>Curso Alterno del Alterno A.A.: Elige opción No</p>	
<p>A.A7.-Elige la opción No del mensaje de confirmación</p>	<p>A.A8.- El use case continua en el curso</p>



	alterno A8 del curso normal de eventos.
Curso Alterno B: Elige opción <i>Editar Doctor</i>	
<p>B3.- Selecciona el nombre del doctor de la lista Doctores de ventana Administrar Doctor-Especialidad.</p> <p>B4.- Elige la opción Editar Doctor de la ventana Administrar Doctor-Especialidad.</p> <p>B6.- Modifica los datos requeridos en la ventana Modificar Doctor.</p>	<p>B5.- Muestra la ventana Modificar Doctor.</p> <p>B7.- Valida los datos ingresados (cedula, apellidos, nombres).</p> <p>B8.- Elige la opción Guardar de la ventana Modificar Doctor.</p> <p>B9.- Almacena los datos en el repositorio de datos doctor.</p> <p>B10- Cierra automáticamente la ventana Modificar Doctor.</p> <p>B11.- Muestra la ventana Administrar Doctor-Especialidad.</p> <p>B12.- El use case finaliza.</p>
Curso Alterno del Alterno B.A.: Campos vacíos	
	<p>B.A7.- Presenta mensaje de aviso "Existen campos vacíos"</p> <p>B.A8.- Cierra automáticamente la ventana Modificar Doctor.</p> <p>B.A9.- El use case continua en el curso alterno B3 del curso normal de eventos.</p>



Curso Alterno del Alterno B.B.: Elige opción <i>Cancelar</i>	
B.B8.- Elige la opción Cancelar de la ventana <i>Modificar Doctor.</i>	B.A9.- El use case continua paso B10 del curso normal de eventos.
Curso Alterno C: Elige opción <i>Agregar Especialidad</i>	
C3.- Elige la opción <i>Agregar Especialidad</i> de la ventana <i>Administrar Doctor-Especialidad.</i>	C4.- Muestra la ventana <i>Agregar Especialidad.</i>
C5.- Digita los datos en los campos correspondientes de la ventana <i>Agregar Especialidad.</i>	C7.- Valida los datos ingresados (especialidad).
C6.- Elige la opción <i>Aceptar</i> de la pantalla <i>Agregar Especialidad.</i>	C8.- Almacena los datos en el repositorio de datos <i>especialidad.</i>
	C9.- Cierra automáticamente la ventana <i>Agregar Especialidad.</i>
	C10.- Regresa a la pantalla <i>Administrar Doctor-Especialidad.</i>
	C11.- El use case finaliza.
Curso Alterno del Alterno C.A.: Elige opción <i>Cancelar</i>	
C.A6.- Elige la opción Cancelar de la ventana <i>Agregar Especialidad.</i>	C.A7.- El use case continua paso C9 del curso alternativo de eventos.
Alterno D: Elige opción <i>Eliminar Especialidad</i>	
D3.- Selecciona un doctor de la lista Doctoresde la ventana <i>Administrar</i>	



<p>Doctor-Especialidad.</p> <p>D4.- Selecciona una especialidad de la lista Especialidades del doctor de la ventana Administrar Doctor-Especialidad.</p> <p>D5.- Selecciona la opción Eliminar Especialidad de la ventana Administrar Doctor-Especialidad.</p> <p>D7.- Elige la opción Aceptar del mensaje de confirmación</p>	<p>D6.- Muestra un mensaje de confirmación de que si está seguro de querer eliminar el dato del sistema.</p> <p>D8.- Elimina los datos de la especialidad en el repositorio de datos doctor_especialidad.</p> <p>D9- Muestra la ventana Administrar Doctor-Especialidad.</p> <p>D10.- El use case finaliza</p>
<p>Curso Alternativo E: Elige opción <i>Editar Especialidad</i></p>	
<p>E3.- Selecciona el nombre de la especialidad de la lista Especialidades de ventana Administrar Doctor-Especialidad.</p> <p>E4.- Elige la opción Editar Especialidad de la ventana Administrar Doctor-Especialidad.</p> <p>E6.- Modifica los datos requeridos en la ventana Editar Especialidad.</p>	<p>E5- Muestra la ventana Editar especialidad.</p> <p>E7.- Valida el dato ingresados.</p> <p>E8.- Elige la opción Guardar de la</p>



	<p>ventana Editar Especialidad.</p> <p>E9.- Almacena el dato en el repositorio de datos especialidad.</p> <p>E10- Cierra automáticamente la ventana Editar Especialidad..</p> <p>E11.- Regresa a la pantalla Administrar Doctor-Especialidad.</p> <p>E12.- El use case finaliza.</p>
Curso Alterno del Alterno E.A.: Campos vacíos	
	<p>E.A7.- Presenta mensaje de aviso “Existen campos vacíos”</p> <p>E.A8.- Cierra automáticamente la ventana Editar Especialidad..</p> <p>E.A9.- El use case continua en el curso alternativo E3 del curso alternativo de eventos.</p>
Curso Alterno del Alterno E.B.: Elige opción <i>Cancelar</i>	
E.B8.- Elige la opción <i>Cancelar</i> de la ventana Editar Especialidad.	<p>E.B9.- El use case continua paso E10 del curso alternativo de eventos.</p>
Alterno F: Elige opción <i>Salir</i>	
F3.- Elige la opción Salir de la ventana Administrar Doctor-Especialidad.	<p>F4.- Cierra la ventana Administrar Doctor-Especialidad.</p> <p>F5.- Muestra la ventana Menú Principal</p> <p>F6.- El use case finaliza</p>



CURSO NORMAL DE EVENTOS

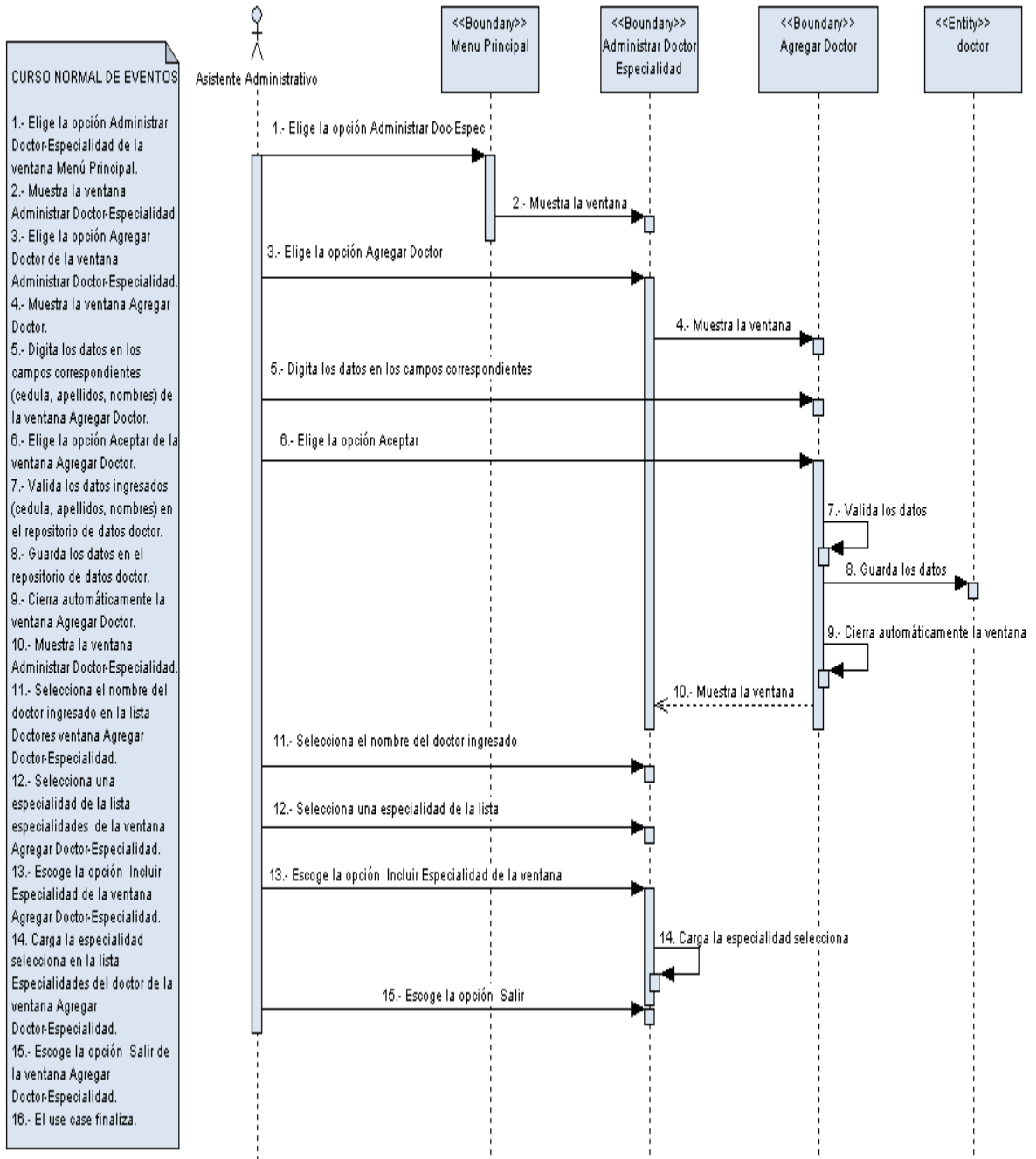


Figura 51: Diagrama de Secuencia: Administrar Doctor-Especialidad



CURSO ALTERNO A: ELECCION DE OPCION ELIMINAR DOCTOR

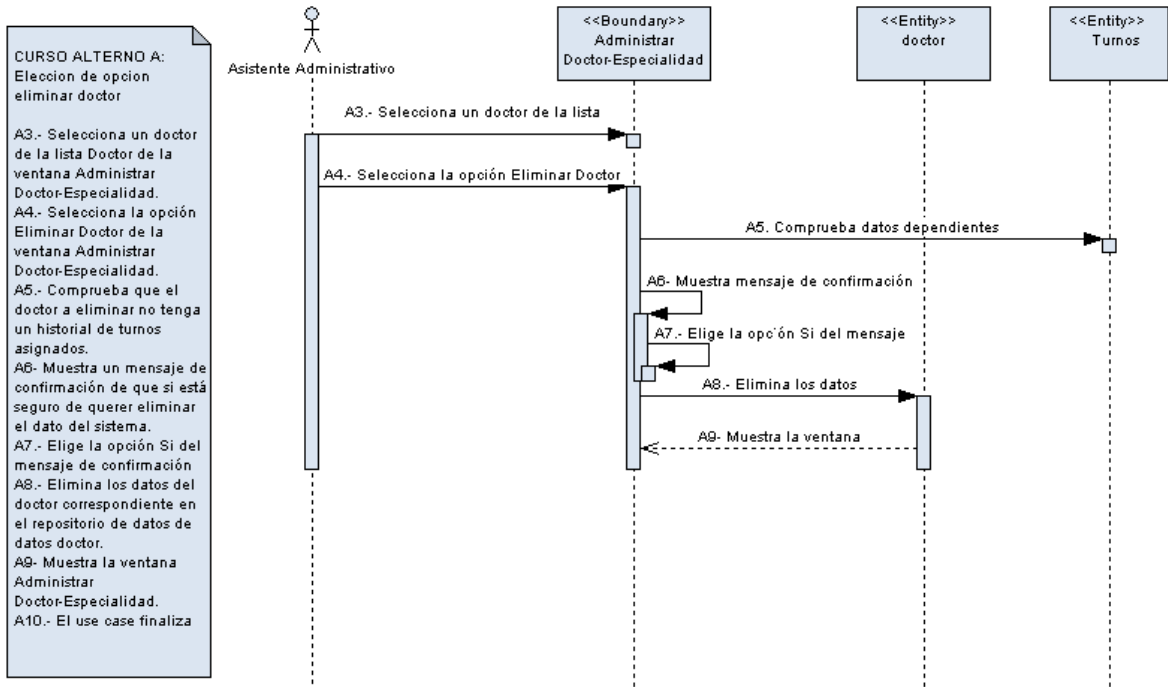


Figura 52: Diagrama de Secuencia: Curso Alternativo A: Elección de opción Eliminar Doctor

CURSO ALTERNATIVO DEL ALTERNATIVO A.A: Elige opción No

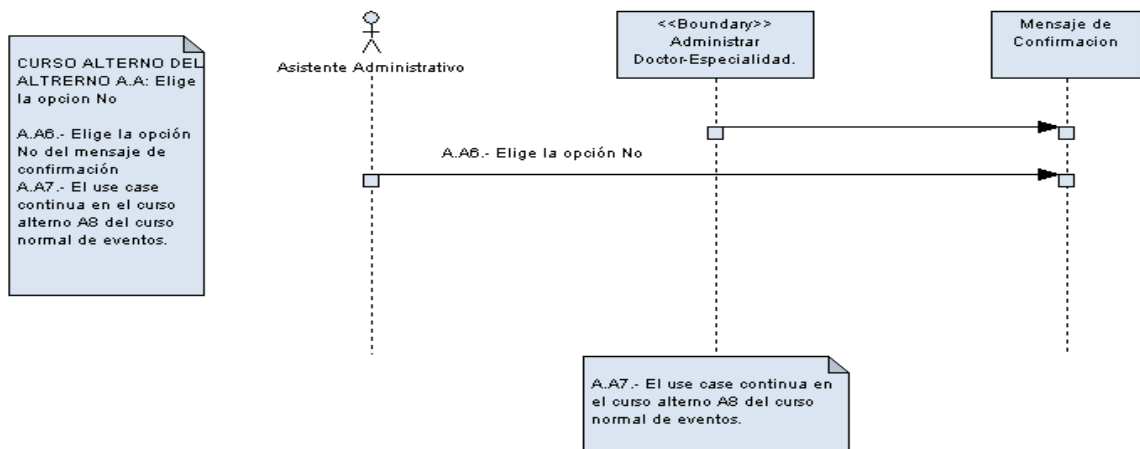


Figura 53: Diagrama de Secuencia: Curso Alternativo del Alternativo A.A: Elige la opción No.

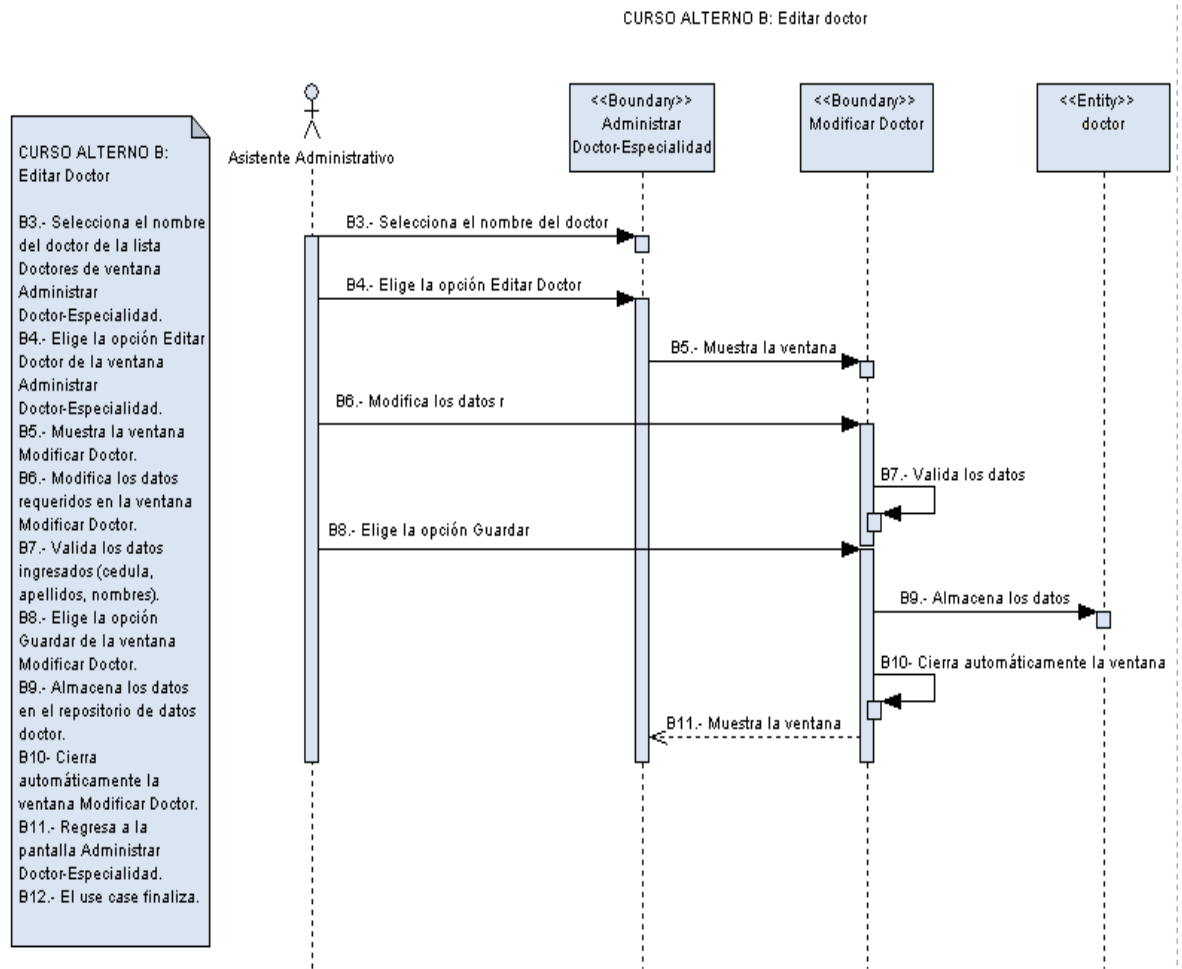


Figura 54: Diagrama de Secuencia: Curso Alternativo B: Elige opción Editar Doctor.

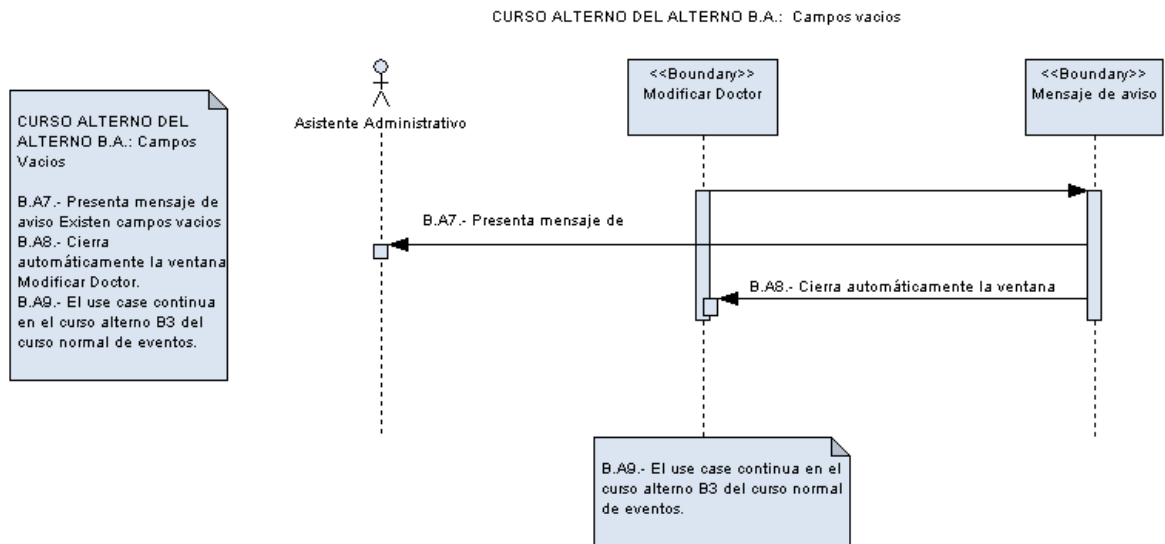


Figura 55: Diagrama de Secuencia: Curso Alterno del Alterno B.A: Campos Vacíos.

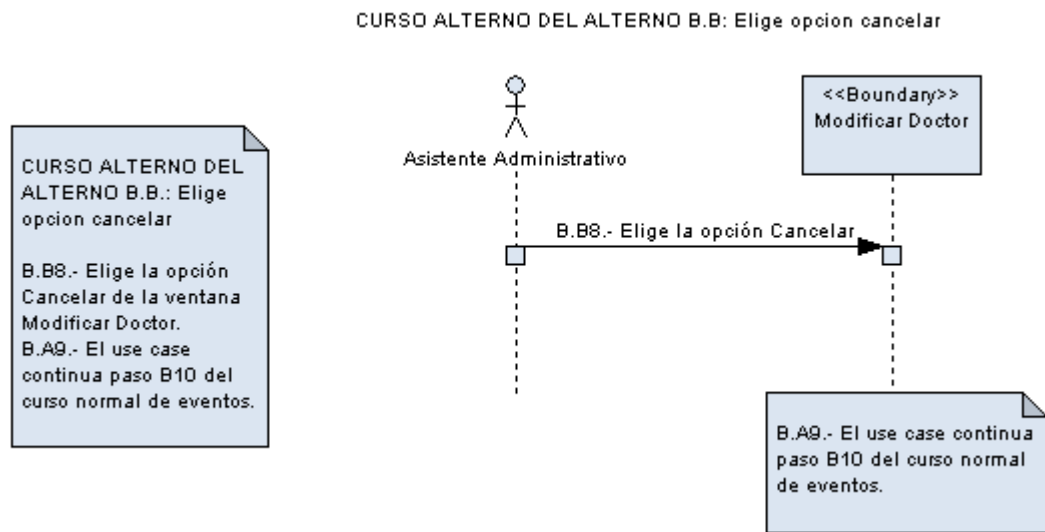


Figura 56: Diagrama de Secuencia: Curso Alterno del Alterno B.B: Elige opción Cancelar.

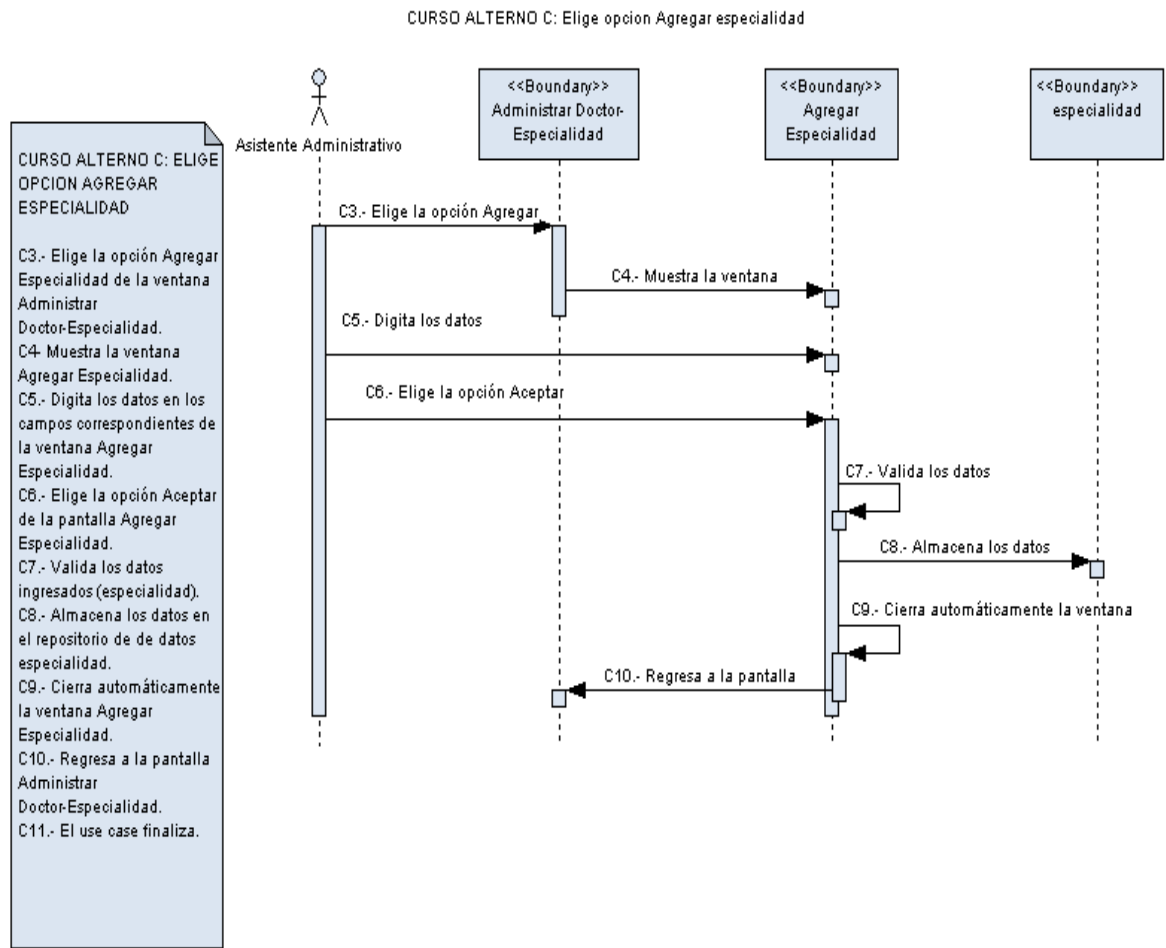


Figura 57: Diagrama de Secuencia: Curso Alterno C: Elige opción Agregar Especialidad



CURSO ALTERNO DEL ALTERNO C.A: Elige opción cancelar

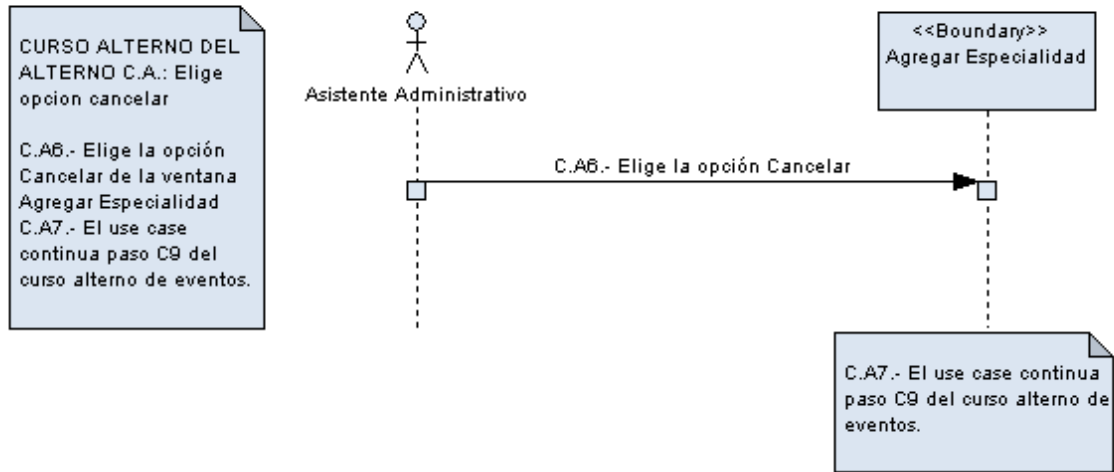


Figura 58: Diagrama de Secuencia: Curso Alterno del Alterno C.A: Elige opción Cancelar.

CURSO ALTERNO D: Eliminar Especialidad

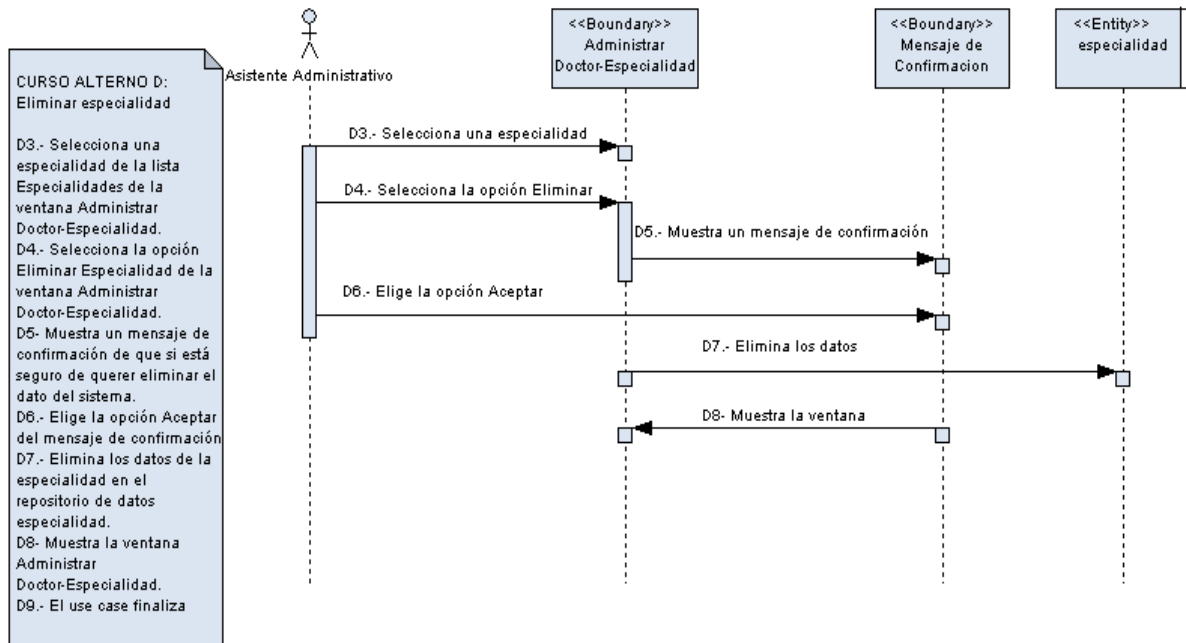


Figura 59: Diagrama de Secuencia: Curso Alterno D: Eliminar Especialidad.

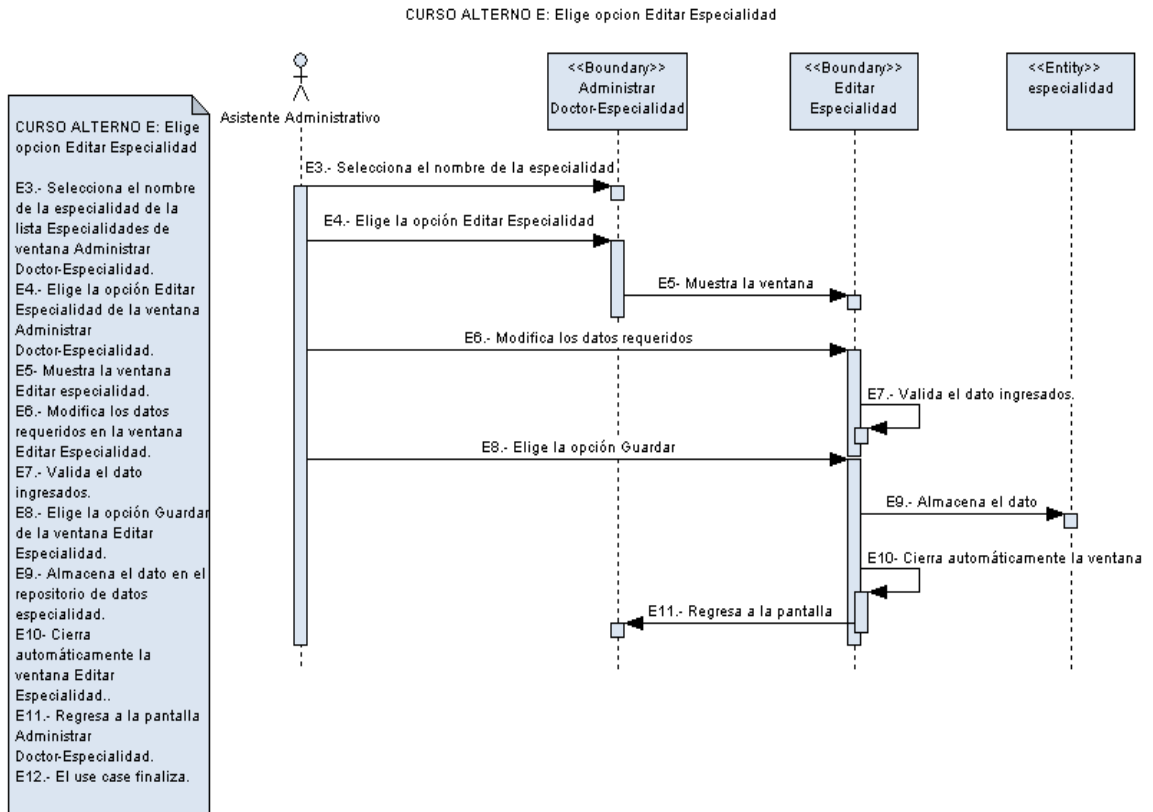


Figura 60: Diagrama de Secuencia: Curso Alterno E: Elige opción Editar Especialidad.

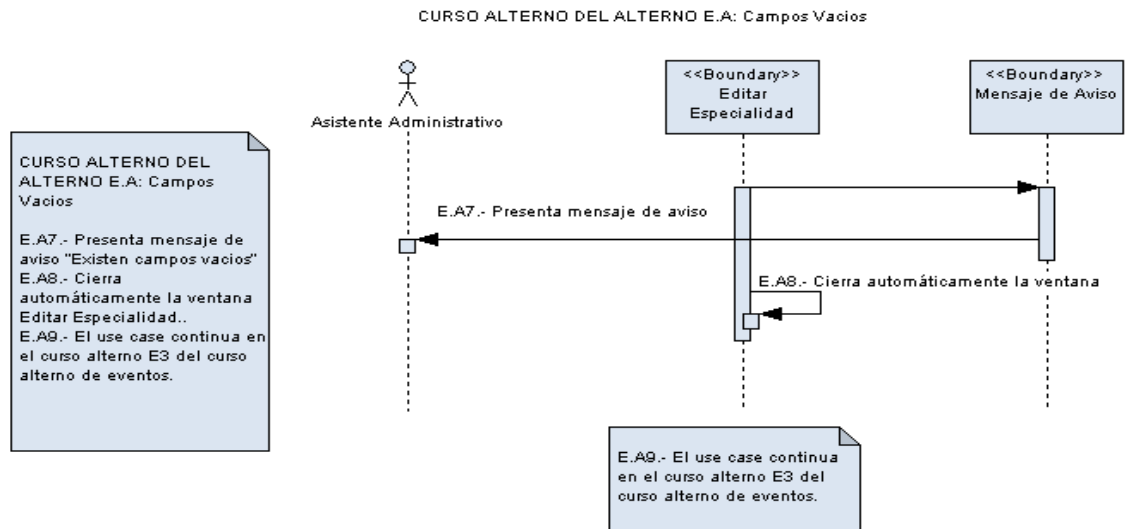


Figura 61: Diagrama de Secuencia: Curso Alterno del Alterno E.A: campos Vacíos.

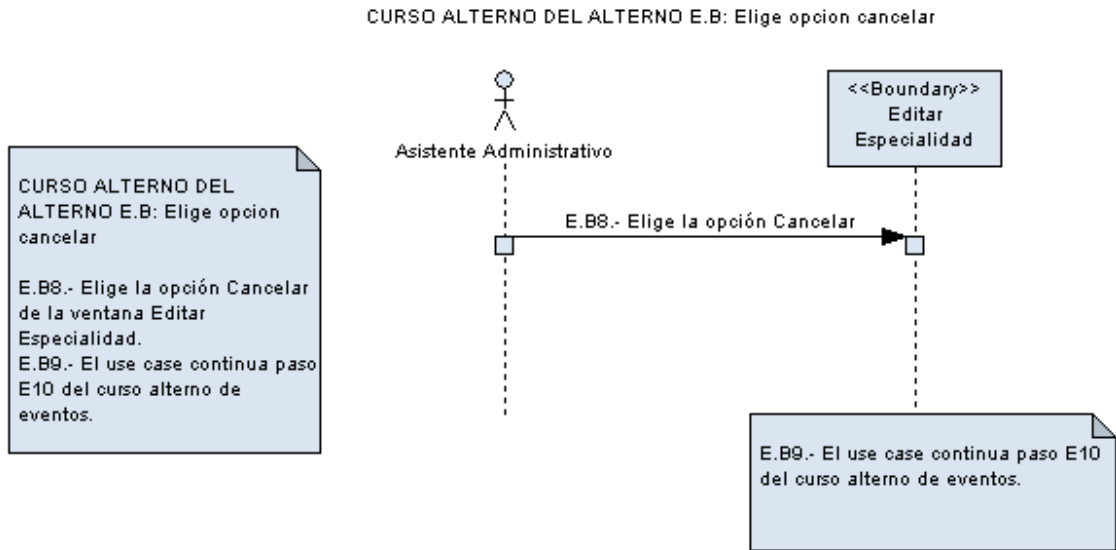


Figura 62: Diagrama de Secuencia: Curso Alterno del Alterno E.B: Elige opción Cancelar.

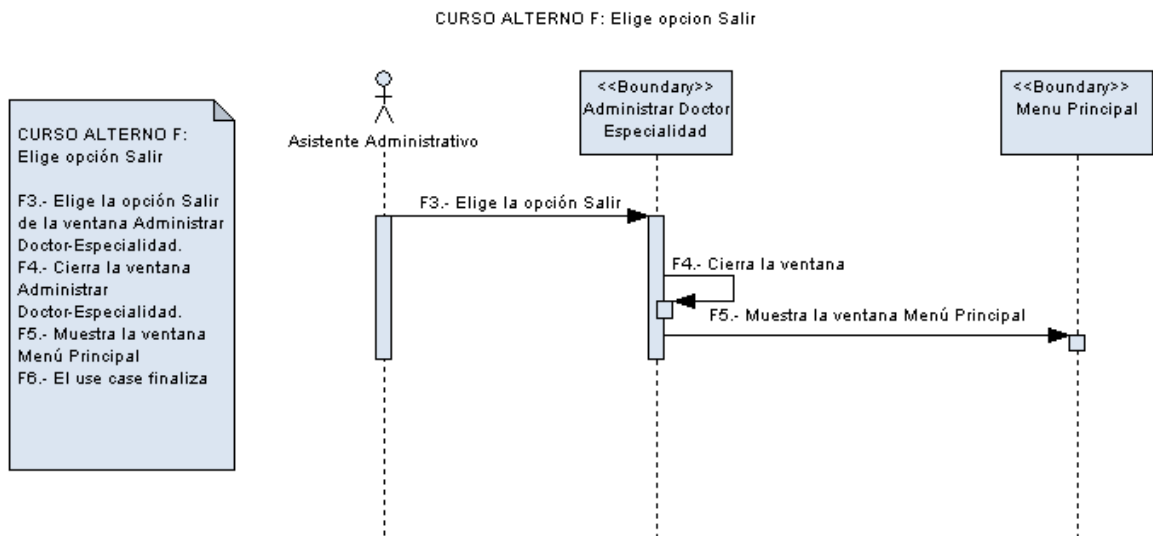


Figura 63: Diagrama de Secuencia: Curso Alterno F: Elige opción Salir.



2.8.2.1.6. Ingreso al Sistema

TABLAXVIII.
CASO DE USO: INGRESO AL MODULO DOCTOR

Nombre de la Pantalla:	Ingreso al
Referencia de requerimientos:	RQF 15, RQF 16
	
Realizado por:	Alba Pineda, Carmen Becerra
Fecha:	

Nombre:	Ingreso al Modulo Doctor
Actor(es):	Doctor
Propósito:	Confirmar que el usuario está autorizado para ingresar al sistema.
Descripción Breve:	El usuario deberá ingresar el número de cédula y password para acceder al sistema. El sistema informa si está autorizado o no.
Tipo:	Primario, esencial
Precondición(es)	El usuario haya iniciado la ejecución del sistema.
Post-condiciones(es)	El usuario accede al sistema con sus respectivas funcionalidades de su cuenta de usuario.



CURSO NORMAL DE EVENTOS	
Doctor	Sistema
<p>1.- Digita los datos en los campos de texto Cédula y Password de la ventana Ingreso al Sistema.</p> <p>2.- Elige la opción Aceptar de la ventana Ingreso al Sistema.</p>	<p>3.- Valida campos obligatorios</p> <p>4.- Autentifica usuario a través de la cuenta.</p> <p>5.- Muestra la ventana Turnos Doctor.</p> <p>6.- El use case finaliza.</p>
CURSO ALTERNO DE EVENTOS	
Curso Alterno A: Elige opción Cancelar	
<p>A2.- Elige la opción Cancelar de la ventana Ingreso al Sistema.</p>	<p>A3.- Cierra la ventana Ingreso al Sistema.</p> <p>A4.- El sistema se cierra.</p> <p>A5.- El use case finaliza.</p>



DIAGRAMA DE SECUENCIA: INGRESO AL MÓDULO DOCTOR

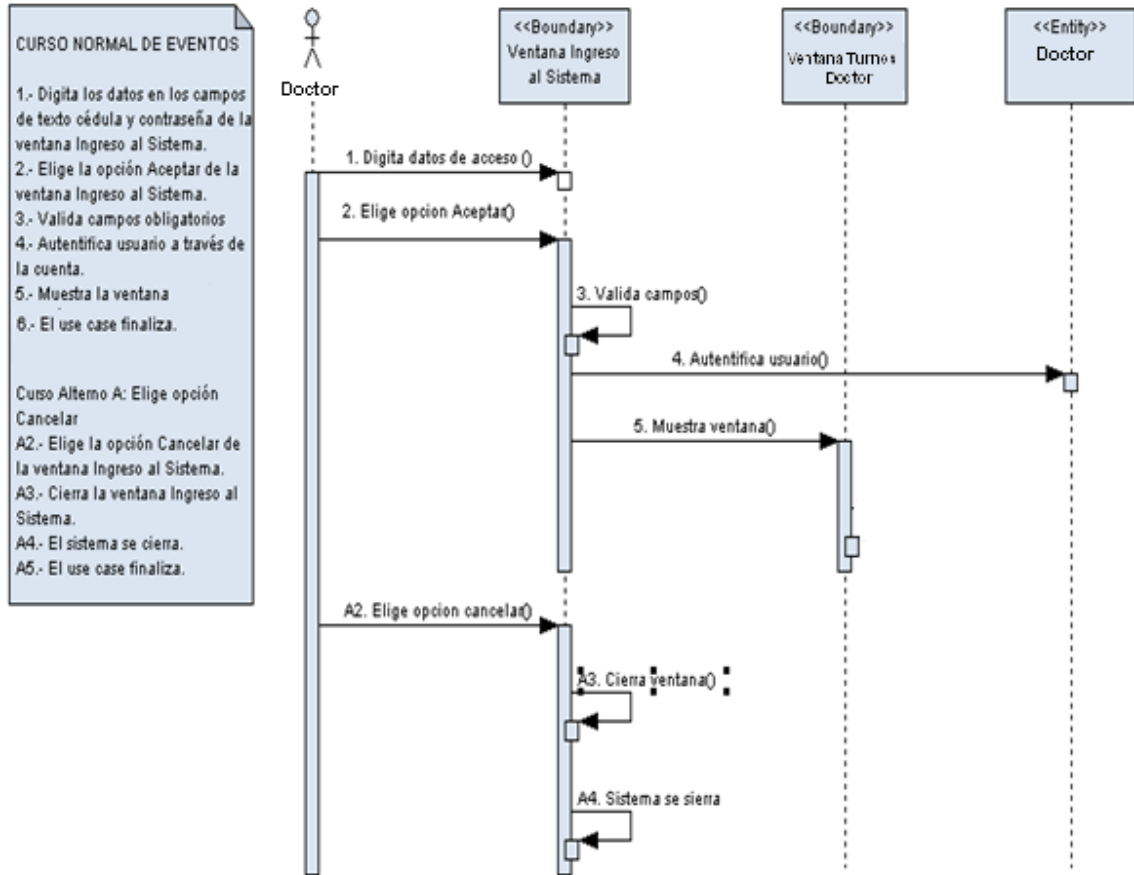
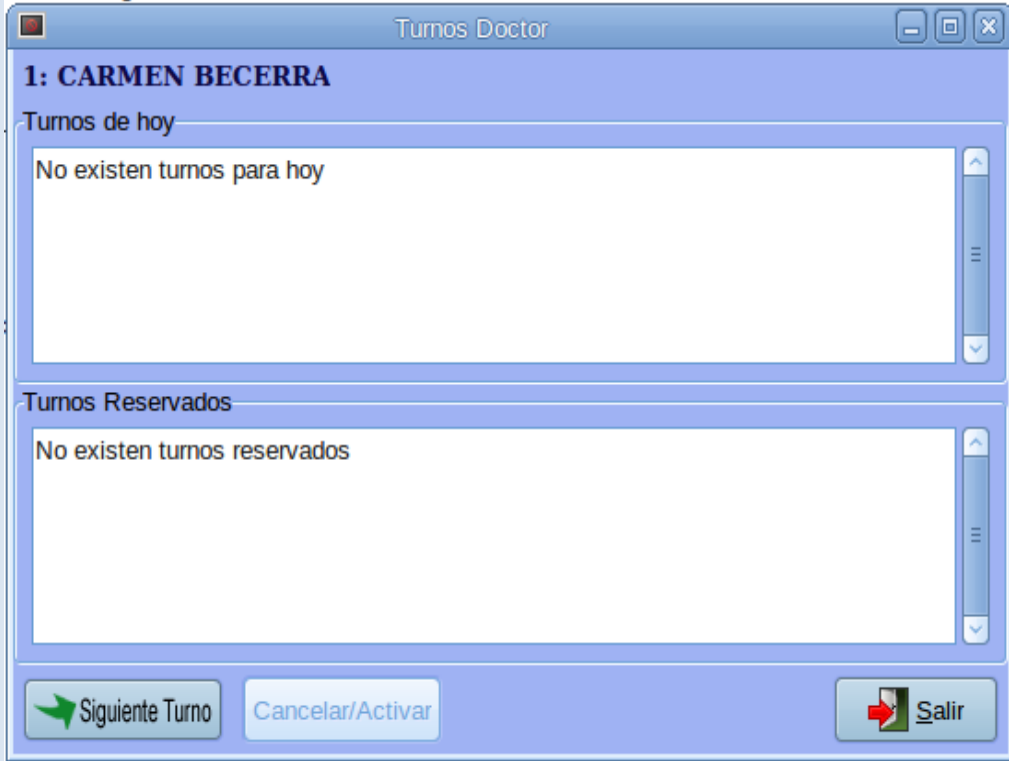


Figura 64: Diagrama de Secuencia: Ingreso al Módulo Doctor.



2.8.2.1.7. Turnos Doctor

TABLA XIX.
CASO DE USO: TURNOS DOCTOR

Nombre de la Pantalla:	Turnos Doctor
Referencia de requerimientos:	RQF 13, RQF 14
	
Realizado por:	Alba Pineda, Carmen Becerra
Fecha:	

Nombre:	Turnos Doctor
Actor(es):	Doctor
Propósito:	Visualizar la lista de turnos de atención que se ha asignado, y registrar la atención de los mismos.
Descripción Breve:	El sistema permite al doctor visualizar los turnos y



	registrar su atención.
Tipo:	Primario, esencial
Precondición(es)	El doctor debe haber ingresado al sistema por medio de su contraseña personal digitada y validada en la pantalla de ingreso, y elegido la opción Turnos Doctor de la pantalla Menú Principal .
Post-condiciones(es)	El sistema debe generar la lista de turnos de atención para el doctor.
CURSO NORMAL DE EVENTOS	
Doctor	Acciones del Sistema
2.- Escoge la opción Siguiente Turno de la ventana Turnos Doctor .	1.- Carga la lista de los Turnos de Hoy y Turnos reservados en la ventana Turnos Doctor 3.- Presenta en el display el número de turno que será atendido. 4.- El use case finaliza.
CURSO ALTERNO DE EVENTOS	
Curso alternativo A: Elección de la opción <i>Salir</i>	
A2.- Elige la opción Salir de la ventana Turno Doctor .	A3. - El sistema cierra la venta Turnos Doctor . A4.- El use case finaliza.

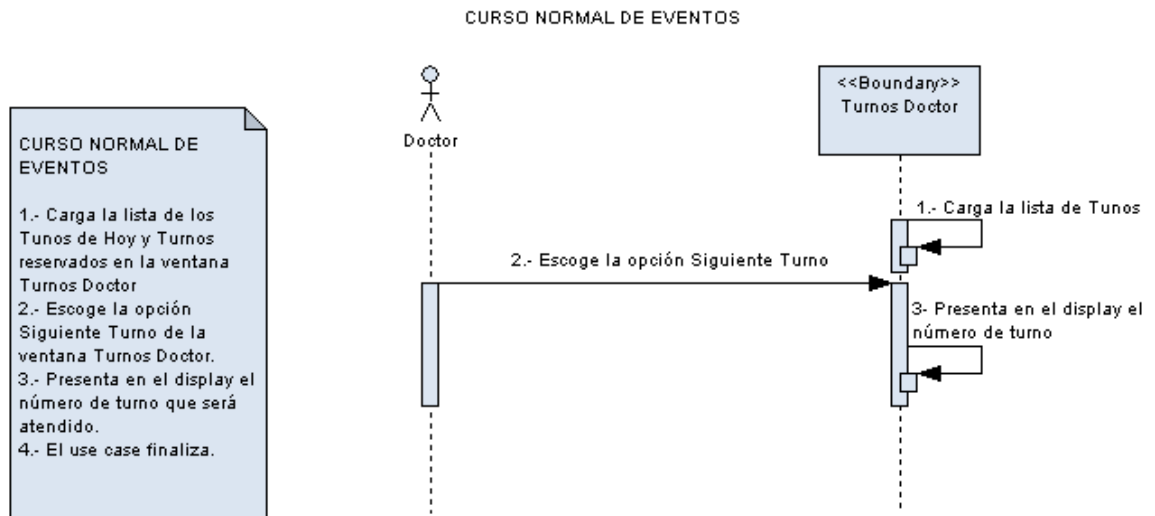


Figura 65: Diagrama de Secuencia: Turnos Doctor.

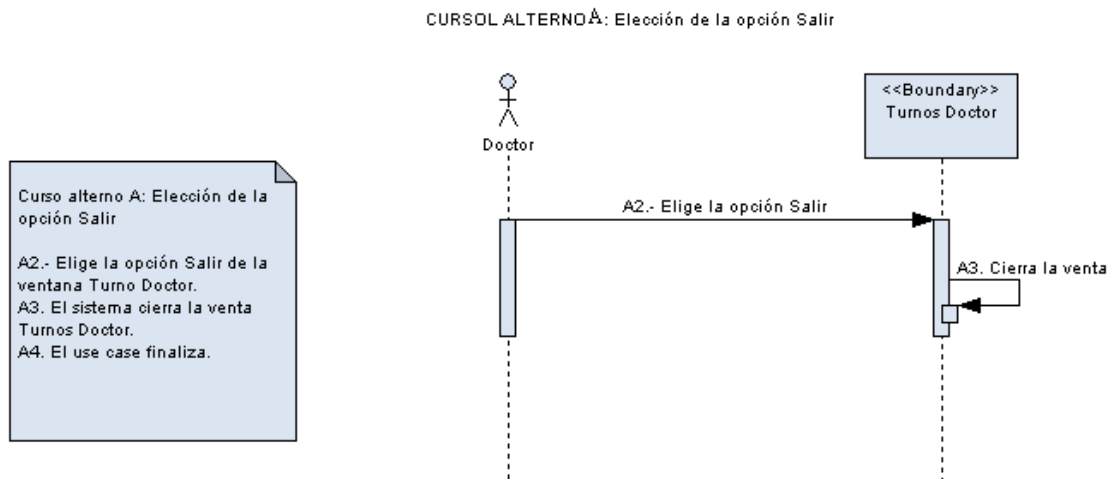


Figura 66: Diagrama de Secuencia: Curso Alternativo A: Escoge la opción Salir.



2.8.2.1.8. Generar Reportes

TABLA XX.
CASO DE USO: GENERAR REPORTES

Nombre de la Pantalla:	Generar Reportes
Referencia de requerimientos:	RQF 06

Reportes

Generar Reporte

Tipo de Reporte

Num Total Turnos
 Turnos por Doctor
 Turnos por Especialidad

Tipo de Búsqueda

Por Doctor
 Por Especialidad

Fecha Inicio
 Fecha Fin

Turno	Paciente	Especialidad	Doctor	Fecha/Hora
1	4	Patrici Carrion	GINECOLOGIA	ANDRADE CARLOS
2				
3	Total	1		
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Realizado por:	Alba Pineda, Carmen Becerra
Fecha:	

Nombre:	Generar Reportes
Actor(es):	Asistente Administrativo (iniciador), Usuario.
Propósito:	Generar un reporte de los turnos asignados a los pacientes.



Descripción Breve:	El sistema permite al usuario imprimir un reporte sobre el número total de turnos asignados a cada doctor y especialidad. El sistema utiliza la base de datos la cual proporciona los datos para generar los diferentes tipos de reportes.
Tipo:	Primario, esencial
Precondición(es)	El asistente administrativo debe haber ingresado al sistema por medio de contraseña personal digitada y validada en la pantalla de ingreso; y, haber seleccionado la opción Generar Reportes del Menú Principal .
Post-condiciones(es)	Que exista por lo menos un turno de atención al cliente asignado.

CURSO NORMAL DE EVENTOS

Asistente Administrativo	Acciones del Sistema
<p>1.- Elige el ítem Num. Total Turnos de la ventana Generar Reportes.</p> <p>3. Selecciona la Fecha Inicial y la Fecha Final en que desea generar el reporte de la ventana Generar Reportes</p> <p>6. Elige la opción Generar Reporte de la ventana Generar Reportes.</p> <p>10.- Elige la opción Imprimir</p>	<p>2. Inhabilita los ítems Turno por Doctor y Turnos por Especialidad de la ventana Generar Reportes</p> <p>4.- Activa la opción Generar Reporte de la ventana Generar Reportes.</p> <p>5.- Valida la fecha ingresada</p> <p>7.- Busca los datos correspondientes en el repositorio de datos reporte.</p> <p>8.- Carga los datos del reporte en la tabla de la ventana Generar Reportes.</p> <p>9.-. Activa la opción Imprimir Reporte de la Ventana Generar Reportes.</p>



<p>Reporte de la ventana Generar Reportes.</p> <p>12.- Elige la opción Imprimir de la ventana Imprimir Reporte.</p>	<p>11.- Muestra la ventana Imprimir Reporte.</p> <p>13.- Imprime el Reporte.</p> <p>14.- Cierra la ventana Imprimir Reporte.</p> <p>15.- Regresa a la ventana Generar Reportes.</p> <p>16. El use case finaliza.</p>
CURSO ALTERNO DE EVENTOS	
Curso Alterno A.- Reporte de turnos por Doctor	
<p>A1.- Elige el ítem Turnos por Doctor de la ventana Generar Reportes.</p> <p>A5.- Selecciona un doctor de la lista Por Doctor de la venta Generar Reportes.</p>	<p>A2.- Inhabilita los ítems Num. Total Turnos y Turnos por Especialidad de la ventana Generar Reportes</p> <p>A3.- Activa la lista Por Doctor de la ventana Generar Reportes.</p> <p>A4.- Carga la lista Por Doctor de la ventana Generar Reportes.</p> <p>A6.- El use case continua en el paso 3 del curso normal de eventos.</p>
Curso Alterno B.- Reporte de turnos por Especialidad	
<p>B1.- Elige el ítem Turnos por ..Especialidad de la ventana Generar Reportes.</p>	<p>B2.- Inhabilita los ítems Num Total Turnos</p>



<p>B5.- Selecciona una especialidad de la lista Por Especialidad de la venta Generar Reportes.</p>	<p>y Turnos por Doctor de la ventana Generar Reportes</p> <p>B3.- Activa la lista Por Especialidad de la ventana Generar Reportes.</p> <p>B4.- Carga la lista Por Especialidad de la ventana Generar Reportes.</p> <p>B6.- El use case continua en el paso 5 del curso normal de eventos.</p>
<p>Curso Alterno C.- Muestra mensaje de aviso</p>	
	<p>C5.-Muestra mensaje “La fecha indicada no es válida”</p> <p>C7.- El use case continua en el paso 3 del curso normal de eventos.</p>
<p>Curso Alterno D.- Selección de la opción Salir.</p>	
<p>D1.- Elige la opción Salir de la ventana Generar Reportes.</p>	<p>D2.- Cierra la ventana Generar Reportes.</p> <p>D3.- Regresa a la ventana Menú Principal.</p> <p>D4. El use case finaliza.</p>

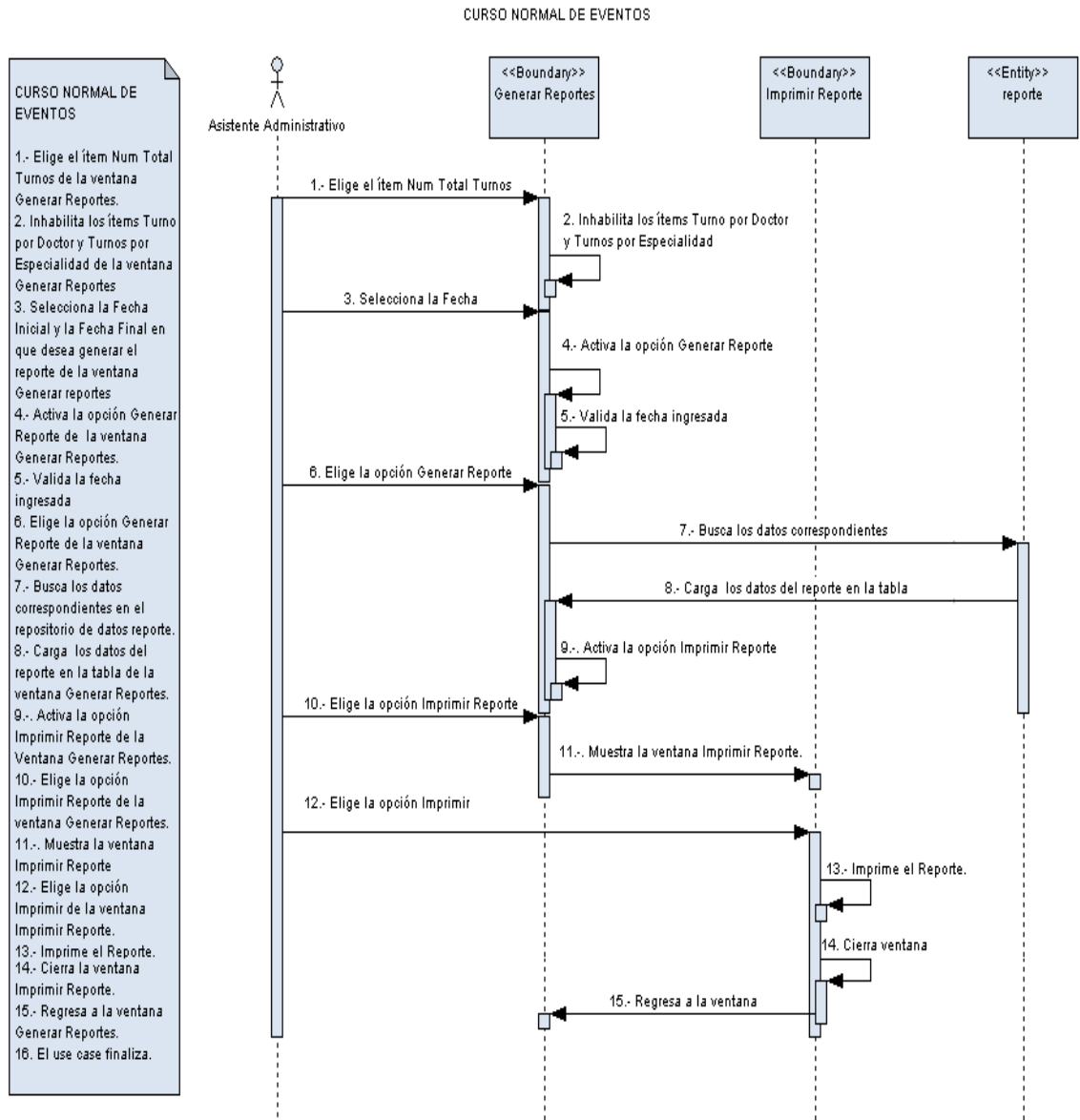


Figura 67: Diagrama de Secuencia: Generar Reportes.

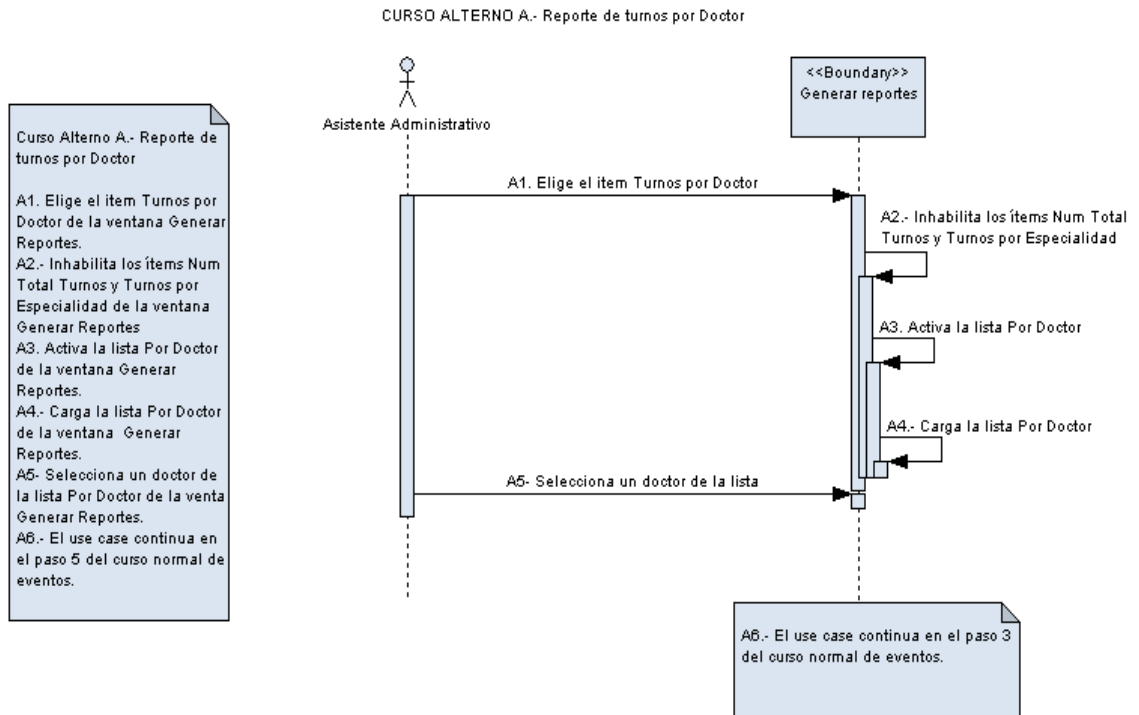


Figura 68: Diagrama de Secuencia: Curso Alterno A. Escoge opción Reporte de Turnos por Doctor.

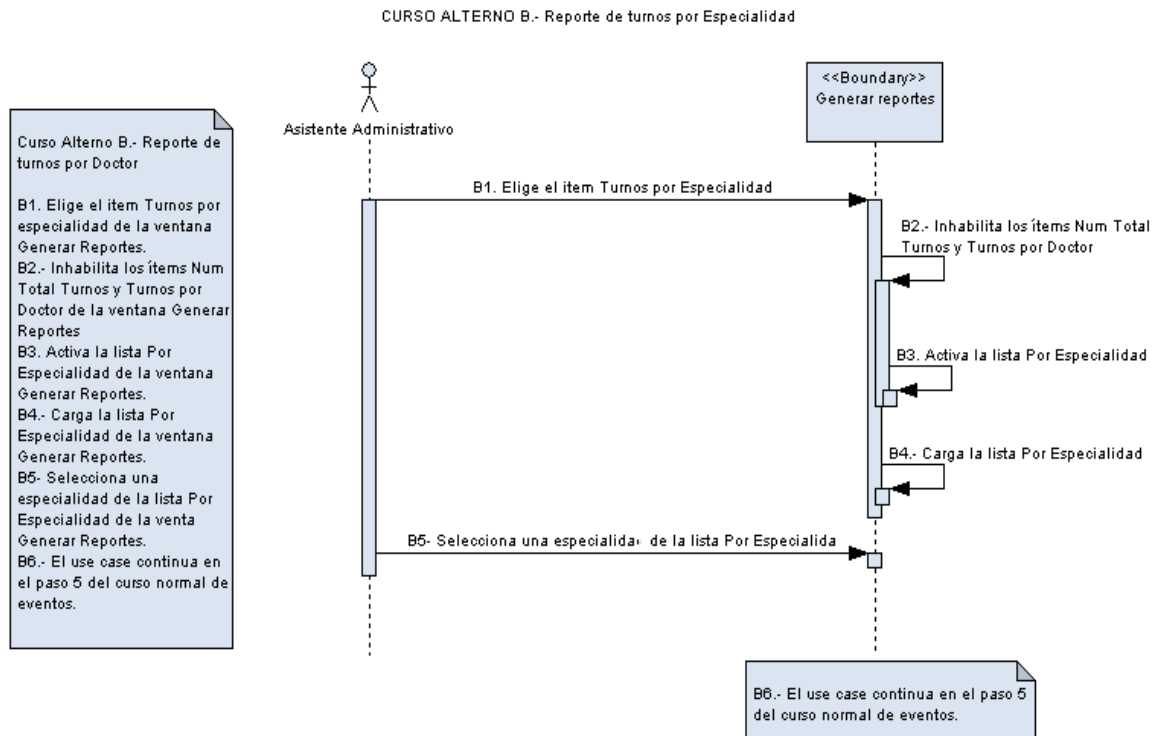


Figura 69: Diagrama de Secuencia: Curso Alternativo B. Escoge opción Reporte de Turnos por Especialidad.

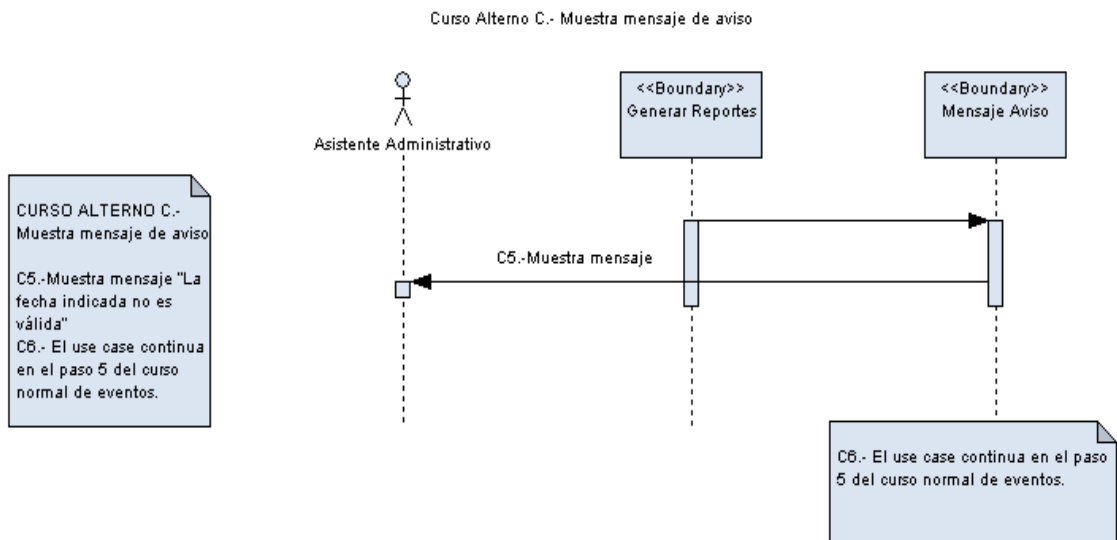


Figura 70: Diagrama de Secuencia: Curso Alternativo C. Mensaje de Aviso.

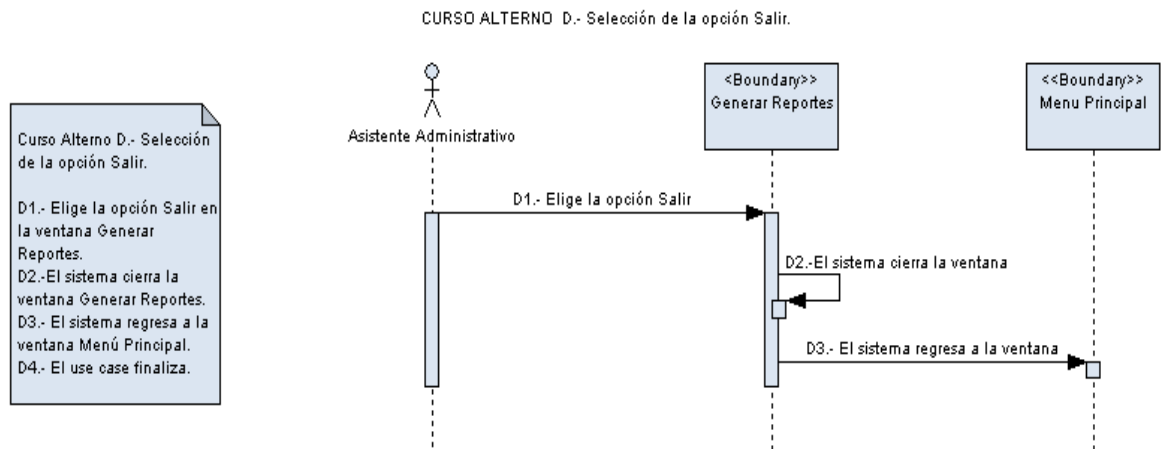


Figura 71: Diagrama de Secuencia: Curso Alternativo D: Selección de la opción Salir.



2.8.2.1.9. Administrar Usuario

TABLA XXI.
CASO DE USO: ADMINISTRAR USUARIOS

Nombre de la Pantalla:	Administrar Usuarios
Referencia de requerimientos:	RQF 06
Realizado por:	Alba Pineda, Carmen Becerra
Fecha:	

Nombre:	Administrar Usuarios
Actor(es):	Usuario(iniciador)
Propósito:	Crear nuevos usuarios que hagan uso del sistema
Descripción Breve:	El sistema le permite al administrador crear nuevos usuarios y habilitarlos para que hagan uso del mismo, así como también inhabilitarlos en caso de que sea necesario
Tipo:	Primario, esencial
Precondición(es)	El administrador debe haber ingresado al sistema por medio de su usuario y contraseña personal digitada y validada en la pantalla de ingreso; y, haber seleccionado la opción Administrar Usuarios del Menú Principal e ingresado nuevamente la clave de administrador.



Post- condiciones(es)	Crear, habilitar o inhabilitar usuarios.	
CURSO NORMAL DE EVENTOS		
Administrador	Sistema	
<p>1.- Digita los datos en los campos de texto Nombre, a Password y volver a ingresar Password de la ventana Administración de Usuarios.</p> <p>2.- Elige la opción Crear de la ventana Administración de Usuarios.</p>	<p>3.- Valida que los campos obligatorios estén llenos.</p> <p>4.- Habilita al usuario para que haga uso del sistema.</p> <p>5.- Guarda datos en el repositorio de datos usuarios.</p> <p>6.- Muestra el nombre del usuario creado en la lista Usuarios de la ventana Administración de Usuarios.</p> <p>7.- El use case finaliza.</p>	
CURSO ALTERNO DE EVENTOS		
Curso Alterno A: Elige opción <i>Cancelar</i>		
<p>A2.- Elige la opción Cancelar de la ventana Administración de Usuarios.</p>	<p>A3.- Cierra la ventana Administración de Usuarios.</p> <p>A4.- Muestra la ventana Menú Principal.</p> <p>A5.- El use case finaliza.</p>	
Curso Alterno B: Elige opción <i>Inhabilitar</i>		
<p>B1.- Selecciona un usuario de las lista usuarios de la ventana</p>		



Administración de Usuarios.

B2.- Elige la opción **Inhabilitar** de la ventana **Administración de Usuarios**.

B3.- Verifica estado del usuario.

B4.- Inhabilita al usuario seleccionado.

B5.- Guarda el estado del usuario en el repositorio de datos **usuarios**

B6.- El use case finaliza.

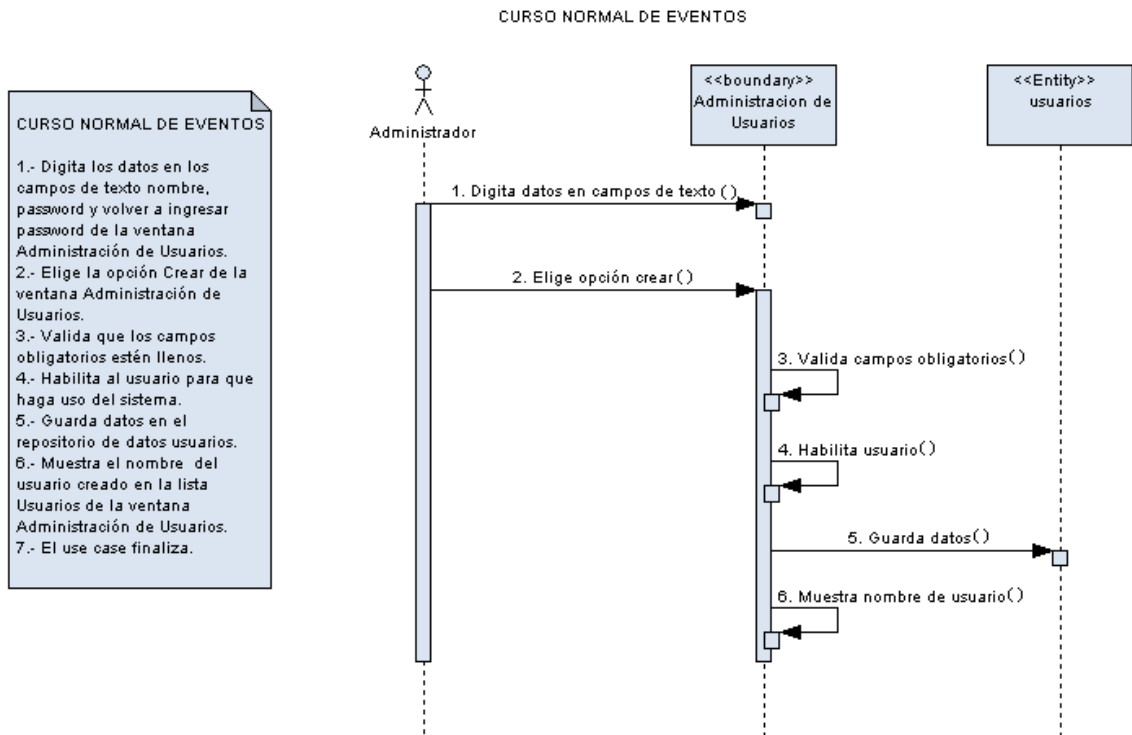


Figura 72: Diagrama de Secuencia: Curso Normal de Eventos: Administrar Usuarios.



CURSO ALTERNO A: Elige opción cancelar

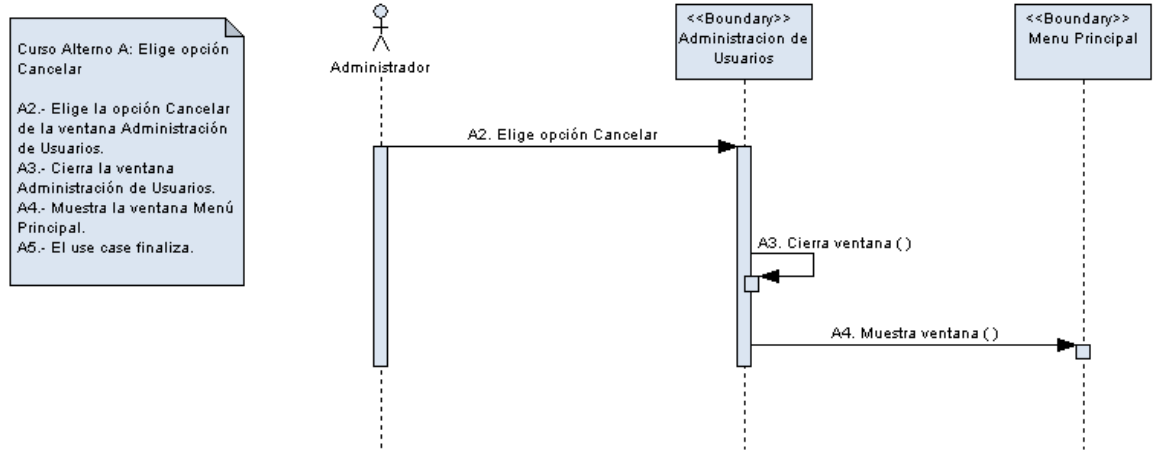


Figura 73: Diagrama de Secuencia: Curso Alternativo de Eventos A: Elige opción Cancelar.

CURSO ALTERNATIVO DE EVENTOS: Elige opción Inhabilitar

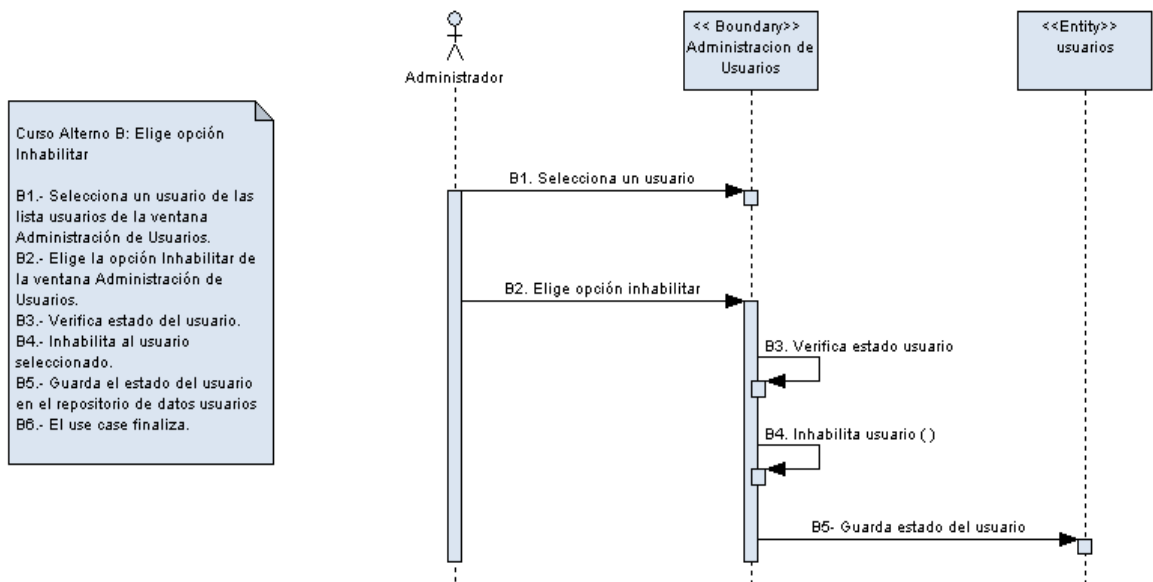


Figura 74: Diagrama de Secuencia: Curso Alternativo de Eventos B: Elige opción Inhabilita.



2.10. Diagrama de Clases

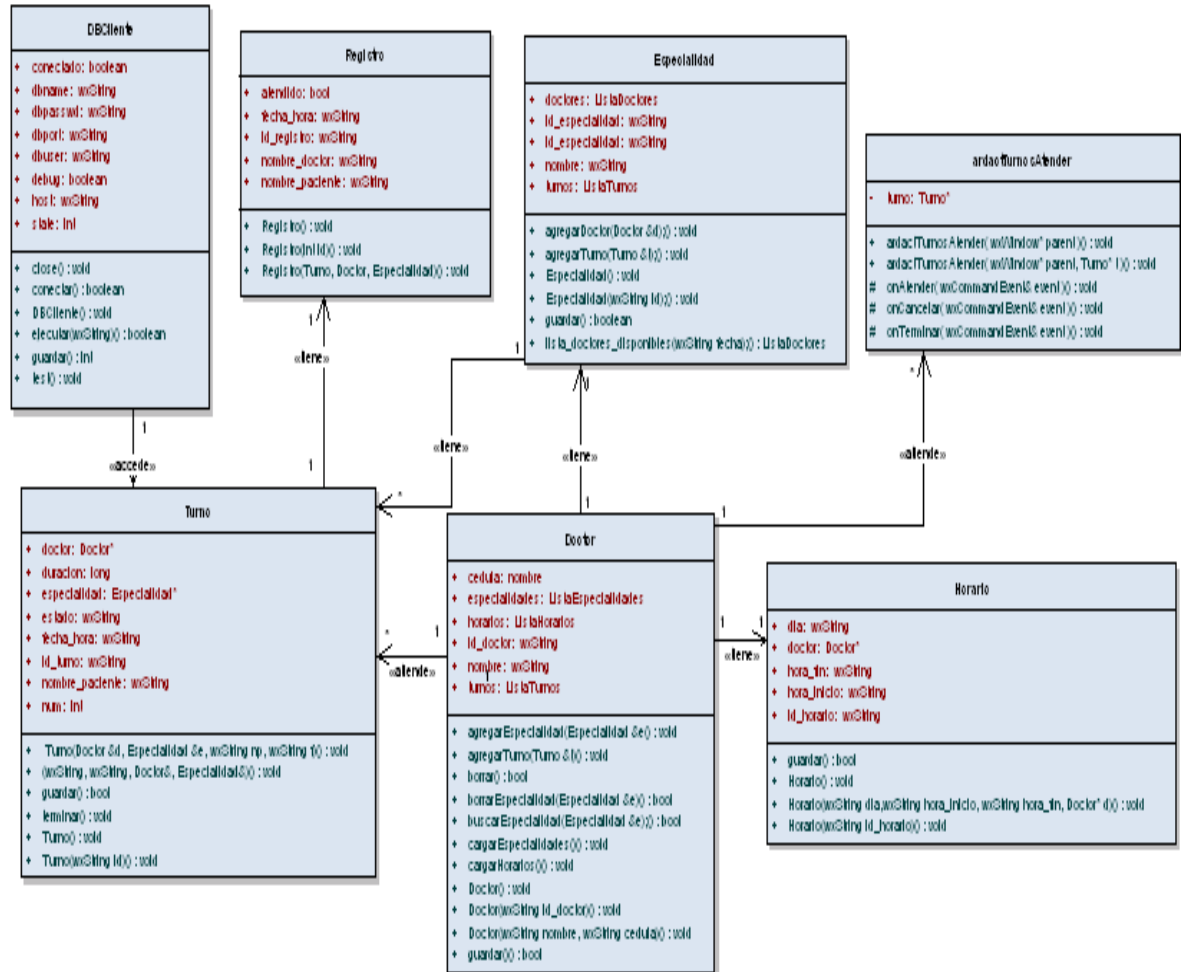


Figura 76: Diagrama de Clases



2.11. Diagrama de la Base de Datos

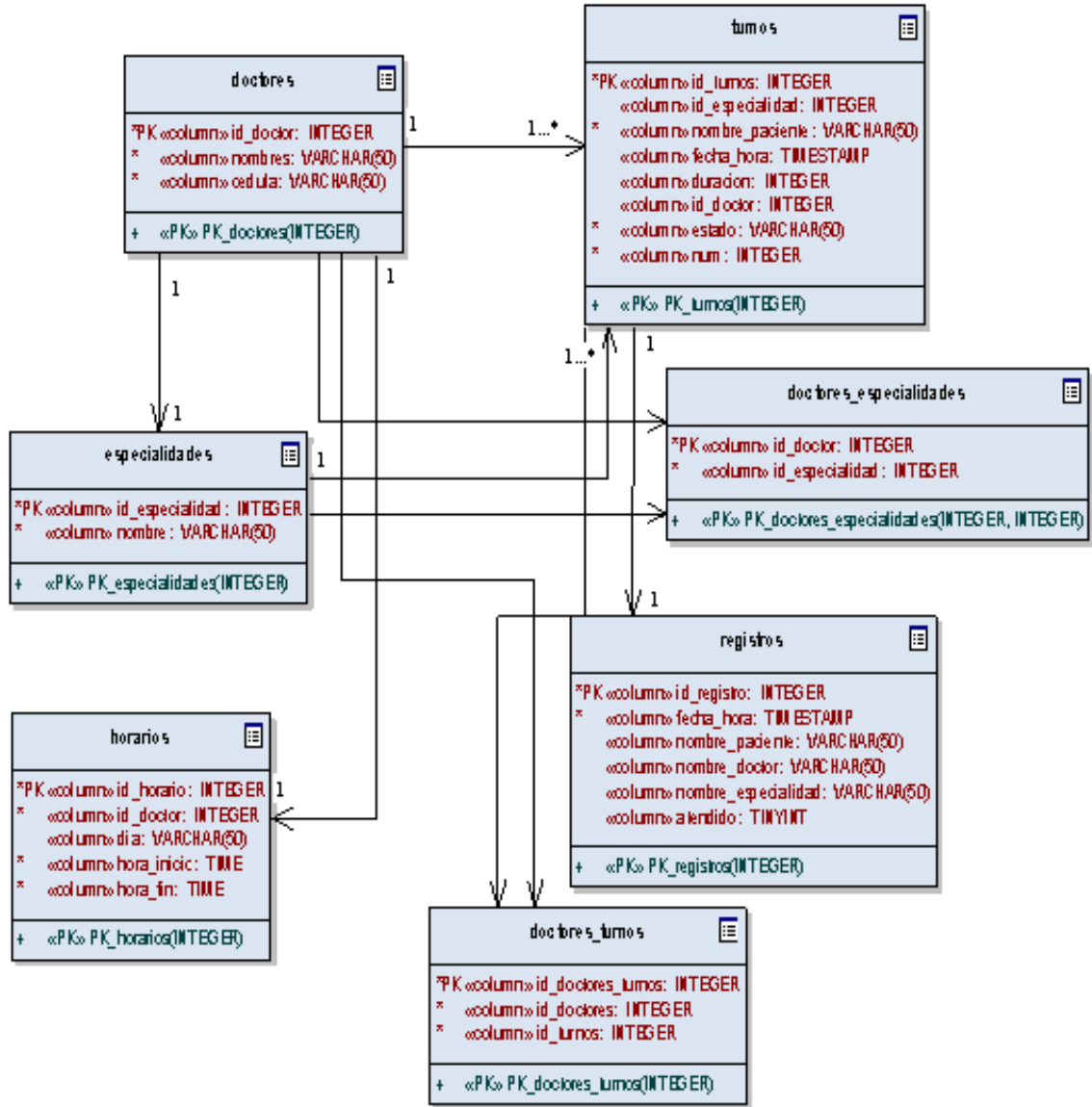


Figura 77: Diagrama de Base de Datos



2.12. Presentación de resultados con la metodología de desarrollo de hardware CAD.

Mediante el uso de metodología CAD se construyó el hardware para el sistema ARDAC, abarcando así todo el proceso de desarrollo que se indica a continuación.

Descripción comportamental del hardware: la parte de hardware de este sistema, la componen tres partes físicas que son: 1) placa Arduino Duemilanove en donde se encuentra instalado el algoritmo encargado de enviar los datos al display de 7 segmentos, 2) Un controlador que será el encargado de determinar si el número de turno consta de uno o dos dígitos y hacer la distribución de los mismos a los display, 3) Display de 7 segmentos que es el dispositivo en donde se graficará el número de turno que se presenta al usuario.

Grafos y diagramas de flujo: Para tener una mejor comprensión y descripción funcional del comportamiento global del sistema se realizaron los siguientes esquemas:

Grafo de la interfaz electrónica

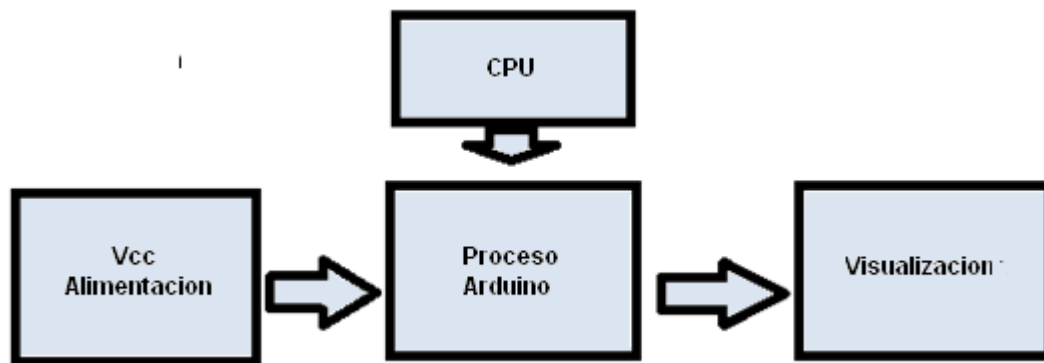


Figura 78: Interfaz electrónica del hardware

Con este esquema se determina el flujo de conexión del software con el hardware en donde la placa Arduino necesita 5v para funcionar, tales 5v, puede recibirlos a través del puerto USB, directamente de la computadora, o con un adaptador de 5v (como los de los teléfonos). Ahora bien, cada segmento del display lleva 3 diodos LED, dichos diodos

trabajan con un voltaje DC de 15v. Entonces para que todo el diseño funcione, se necesita una fuente de alimentación que de los voltajes mencionados. Para esto se utilizó un adaptador genérico de impresora HP, por cuanto son fuentes con dos tipos de voltajes (15, 32), más el voltaje que realmente interesa es el de 15v, debido a que los 5v que faltarían puede conseguirlos de una pc.

2.12.1. Diagrama de Componentes del hardware

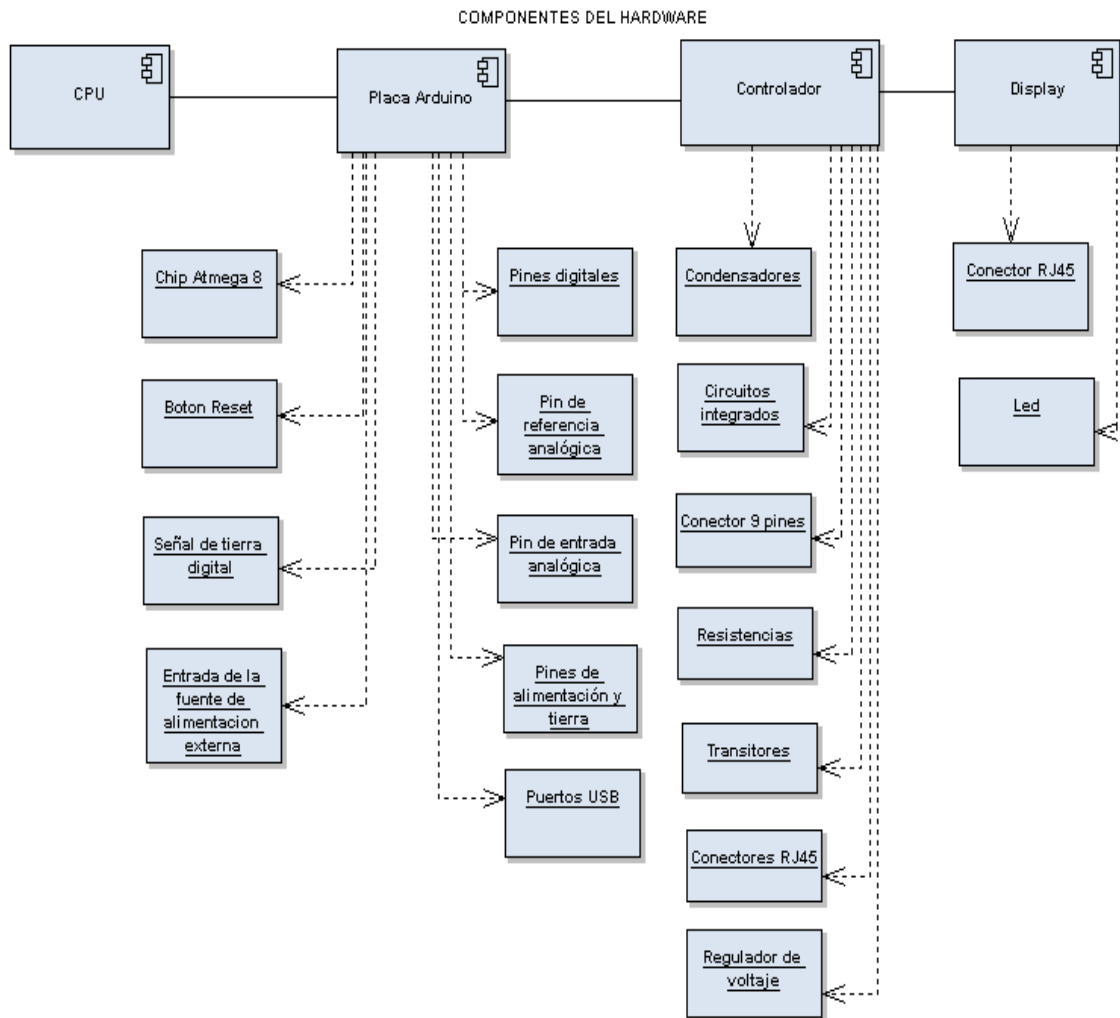


Figura 79: Diagrama de componentes del hardware

En este diagrama se puede evidenciar como quedará construido el hardware en su totalidad así como las partes que integran cada segmento del hardware.

Descripción mediante esquemas: Para facilitar la construcción el hardware se procedió a realizar el esquema del display de 7 segmentos en donde se muestra la ubicación y conexión de sus componentes tal como se indica en figura siguiente:

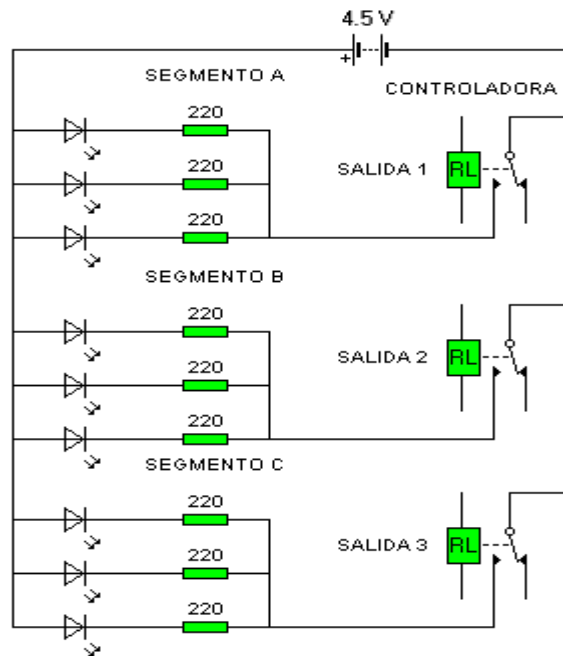


Figura 80: Diagrama esquemático de partes del display de 7 segmentos

Aquí únicamente se ha realizado el esquema de tres segmentos de los 7 que lo componen ya que el resto es idéntico, como se puede deducir del esquema, el número de cables entre la placa controladora y el visualizador digital es de 8 (7 para los segmentos y 1 para el negativo de la alimentación).

También se elaboró el esquema del circuito del controlador del hardware que actuará como un traductor entre el dispositivo y el software del sistema.

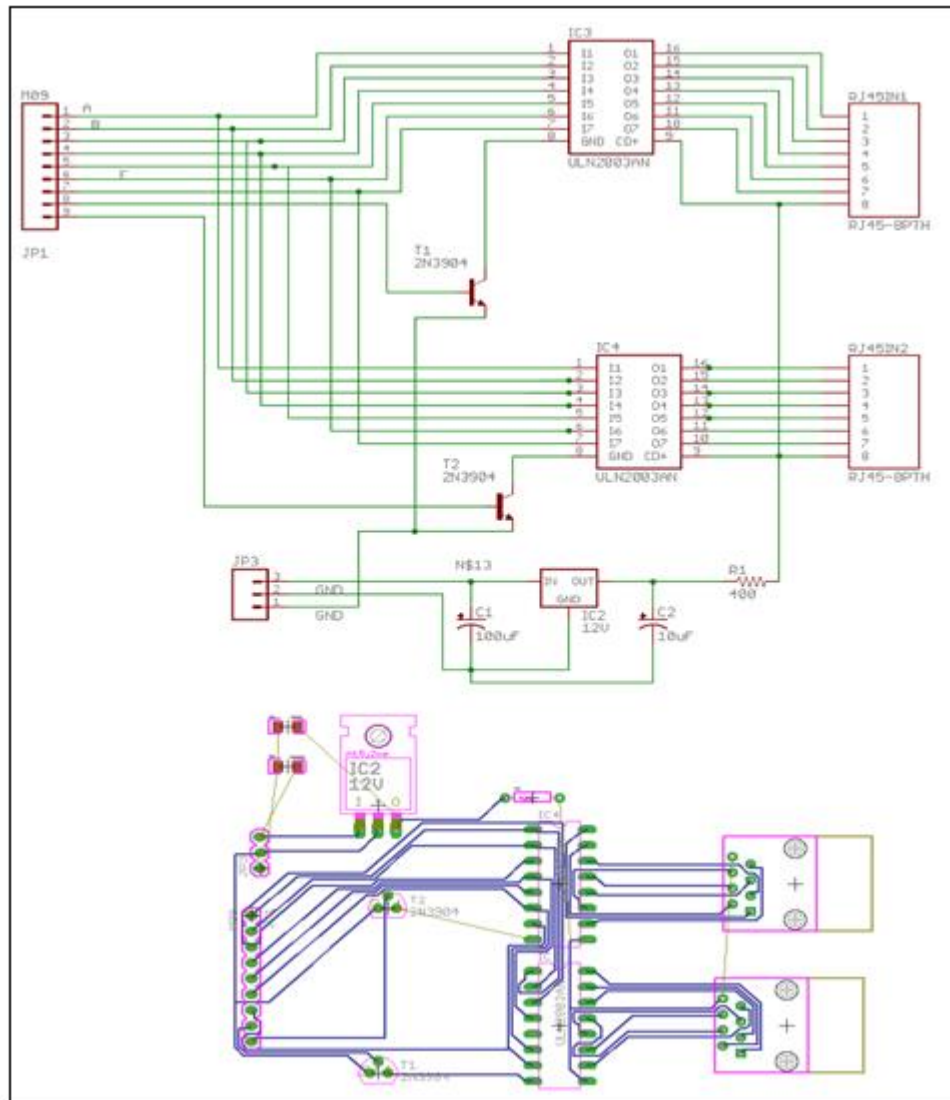


Figura 81: Diagrama esquemático de partes del controlador

Esquema de partes:

- C1: condensador 1
- C2: condensador 2
- IC2: Regulador de voltaje a 12V
- IC3: circuito integrado UNL2003AN (Arreglo de transistores darlington)
- IC4: circuito integrado UNL2003AN (Arreglo de transistores darlington)
- JP1: conector de 9 pines



- JP3: conector
- R1: resistencia 400ohm
- RJ45IN1: jack rj45
- RJ45IN2: jack rj45
- T1: transistor
- T2: transistor

EspecificacionesTécnicas:

TABLA XXII: ESPECIFICACIONES TÉCNICAS DEL CONTROLADOR

PARTE	VALOR	DISPOSITIVO	PAQUETE
C1	100uF	CAP_POL1206	EIA3216
C2	10uF	CAP_POL1206	EIA3216
IC2	12V	V_REG_78XXSIDE	78XXL
IC3	UNL2003AN	UNL2003AN	DIL16
IC4	UNL2003AN	UNL2003AN	DIL16
JP1	MO9	MO9	1X09
JP3		MO3LOCK	1X03_LOCK
R1	400	RESISTORPTH1	AXIAL-0.3
RJ45IN1	RJ45-8PTH	RJ45-8PTH	RJ45-8
RJ45IN2	RJ45-8PTH	RJ45-8PTH	RJ45-8
T1	2N3904	2N3904	TO92
T2	2N3904	2N3904	TO92

Por último se elaboró el esquema de conexión de las partes del hardware: Placa Arduino, Controlador, Display de siete segmentos y un timbre con entrada de corriente de 110V. para que genere un pitido de aviso cuando se presente el número de turno.

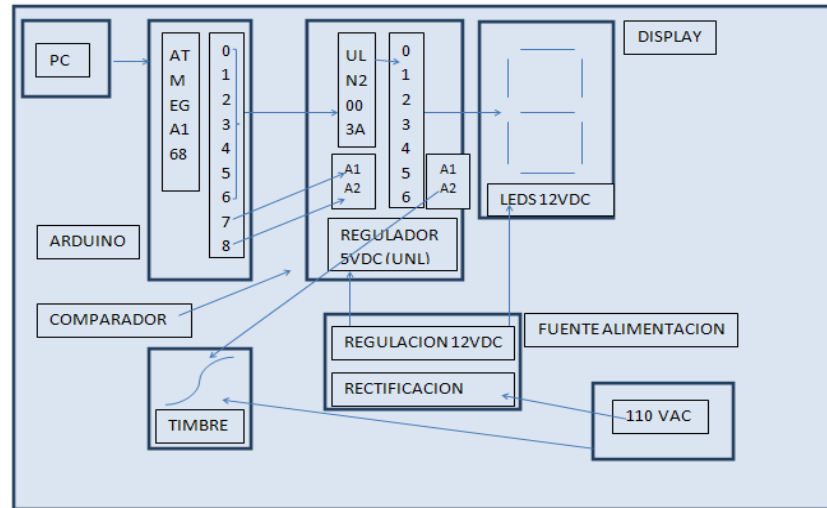


Figura 82: Conexión de las partes del Hardware

La placa Arduino se comunica con el sistema ARDAC por el puerto USB. El algoritmo pregrabado, viene a ser el firmware del PIC ATMEGA168-20PU, que produce resultados a través de los pines 0 al 6, que vienen a ser los segmentos a pintarse, los cuales se organizan a través de comparador ULN2003AN, pines 1 al 6. Así mismo de la interfaz inteligente se obtienen A1 y A2, que vienen a representar la bandera de número correcto (A2->TIMBRE) y cuando se produzca acarreo para números superiores a 9 (A1->DISPLAY 2) números del 10 al 19.

Como ya se mencionó, la alimentación para el comparador es 5 y 12 voltios corriente directa (VDC), mientras el timbre funciona con 110 voltios de corriente alterna (VAC).

Lenguajes de descripción: Para obtener el diseño del controlador del hardware y el diagrama esquemático de los displays de siete segmentos se utilizó el programa EAGLE versión 5.4.0. Además se hizo uso del lenguaje C tradicional permitiendo generar un software encargado de realizar tareas de automatización y asignación de turnos a través de la placa Arduino y finalmente se utilizó **Ubuntu** que es un sistema operativo basado en Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity.



Lenguajes de descripción: Para obtener el diseño del controlador del hardware y el diagrama esquemático de los displays de siete segmentos se utilizó el programa EAGLE versión 5.4.0. Además se hizo uso del lenguaje C tradicional permitiendo generar un software encargado de realizar tareas de automatización y asignación de turnos a través de la placa Arduino y finalmente se utilizó **Ubuntu** que es un sistema operativo basado en Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity.

Programación de dispositivos: La placa Arduino es el dispositivo que fue programado para que funcione en base a los requerimientos del sistema ARDAC, para ello se instaló el software que proporciona el fabricante de la placa para que ésta este habilitada; una vez instalado el software se programa el algoritmo de transmisión de datos en el entorno de Arduino para luego ser instalado en el microprocesador de la misma.

2.13. Pruebas de la Aplicación

El proceso de pruebas y validación permitió evaluar el sistema durante su desarrollo para determinar que los requerimientos del análisis se cumplieron satisfactoriamente; en otras palabras el objetivo de realizar estas pruebas y su validación respectiva es comprobar que lo establecido anteriormente es lo que el usuario quería.

Estas pruebas fueron realizadas a cada uno de los usuarios que interactúan con el sistema; los mismos que primeramente utilizaron el sistema y luego, mediante la aplicación de encuestas se pudo obtener una valoración de su funcionamiento.

2.13.1. Tipos de pruebas

2.13.1.1. Pruebas de funcionalidad

Estas pruebas fueron realizadas directamente por el programador, esto asegura que los módulos que se hagan, tengan la funcionalidad que "se cree" que deben tener, garantizando que el sistema funcione correctamente, como ingreso de datos o manipulación de información por parte del usuario, cumpliendo de esta manera con los requerimientos solicitados por el mismo. Estas pruebas se aplicaron en la finalización de cada caso de uso, alcanzando los siguientes objetivos



- Verificar las funciones incorrectas o ausentes del sistema
- Detectar errores de la interfaz de usuario
- Detectar errores en la base de datos

2.13.1.2. Pruebas de aceptación

Estas pruebas las realizaron los posibles usuarios que harán uso del sistema. Para lo cual se preparó un documento con la funcionalidad esperada del módulo y se lo usó como guía para realizar estas pruebas, lo que permitió observar las apreciaciones finales de los usuarios, verificar la funcionalidad total de la aplicación para poder pulir lo que quede del sistema y definir la finalización del proyecto y el paso a producción.

Los usuarios que intervinieron en estas pruebas fueron:

- Administrador
- Los Asistentes administrativos que en este caso son 4.
- Los Doctores que laboran en el Área de Consulta externa del Hospital

Para realizar estas pruebas, se consideró como población para el cálculo de la muestra al número de Doctores que laboran en el Área de Consulta Externa del Hospital, en este caso estos son un total de 25.

Al ser la población finita, es decir se conoce con exactitud el total de la población, se decidió aplicar la Fórmula para cálculo de la muestra de poblaciones finitas. La fórmula es la siguiente:

$$n = \frac{N * Z^2 * p * q}{d^2 * (N-1) + Z^2 * p * q}$$

Donde:

n = tamaño de la muestra obtenido

N = tamaño de la población (en este caso 25)

Z = nivel de confianza elegido (1.64 tomando en cuenta una seguridad del 90%)



p = proporción esperada (5% que serían 0.05)

q = es igual a 1-p (serían 0.95)

d = precisión o margen de error permitido (se estima un 5% que serían 0.05)

Para obtener el tamaño de la muestra de doctores encuestados cuya población es de 25, se calcula:

$$n = \frac{25 * 1.64^2 * 0.05 * 0.95}{0.05^2 * (25-1) + 1.64^2 * 0.05 * 0.95}$$

$$n = 17$$

Para representar las respuestas de los encuestados se establecieron los siguientes rangos de evaluación.

MS = Muy satisfactorio **S** = Satisfactorio **PS** = Poco satisfactorio

2.13.2. Validación de los resultados de la Aplicación

2.13.2.1. Validación del desenvolvimiento de la aplicación.

Cada usuario que prueba el sistema, llena una encuesta en la cual expresa su grado de aceptación/rechazo del mismo.

Formato de encuesta de aceptación/rechazo de la aplicación dirigida al Administrador, Asistentes administrativos y doctores del Hospital Regional Isidro Ayora (Ver k Anexo apartado Anexo VIII).



2.13.2.2. Ejecución del Plan de Pruebas

Versión: 1.0

TABLA XXIII

EJECUCIÓN DEL PLAN DE PRUEBAS APLICADO A USUARIOS DEL SISTEMA

Identificador	Administrador Asistente Administrativo y Doctores del Hospital Regional Isidro Ayora.
Alcance	Se probará la seguridad, controles de ingreso de datos, confiabilidad, requerimientos de usuarios, integración de módulos.
Ítems a probar	Todos los módulos de la aplicación Sistema Inteligente para la administración de turnos tanto del Asistente Administrativo, Administrador y Doctores
Estrategia	Análisis de entradas y salidas.
Criterios de suspensión y reanudación	<p>Se suspenderá el proceso de pruebas en caso de que no existan las condiciones necesarias; tales como: disponibilidad de tiempo del usuario, detección de errores que no permitan culminar las pruebas del sistema.</p> <p>Se reanudará el proceso de pruebas cuando los criterios de suspensión sean superados.</p> <p>Se culminarán las pruebas de validación una vez que se ha verificado el cumplimiento de los requerimientos impuestos por el usuario y los errores encontrados no requieran de una nueva revisión por parte del usuario.</p>
Documentación	Se realizará un informe de pruebas y las correcciones realizadas, adjuntando como respaldo las encuestas entregadas al usuario.
	<ul style="list-style-type: none">✓ Display de 7 segmentos✓ Lenguaje de programación C✓ Base de datos MySQL✓ Display de 7 segmentos✓ Placa Arduino✓ Personal capacitado para manejo del sistema



Responsable	Alba Pineda y Carmen Becerra
--------------------	------------------------------

2.13.2.3. Informe de Resultados de las Pruebas de Validación

Informe de: Funcionamiento del Sistema Inteligente ARDAC

TABLA XXIV
INFORME DE RESULTADOS DE LAS PRUEBAS DE VALIDACIÓN

Identificador	Ficha FV01. Y FV02. Fichas de validación para el “Sistema de Asignación de Turnos”.
Resumen	Las pruebas se realizaron con la participación del administrador y los asistentes administrativos que laboran en el departamento de archivo, y los doctores del área de consulta externa del Hospital Isidro Ayora.
Variaciones	Se entregó los manuales del funcionamiento de la aplicación a cada usuario correspondientemente.
Resumen de resultados	En la sección de análisis de resultados de la validación, se presentan las estadísticas de los resultados de pruebas de la aplicación.
Resumen de actividades	Luego de la entrega de manuales e indicaciones correspondientes a cada usuario, el administrador y asistentes administrativos procedieron a la manipulación de la aplicación, de igual manera los doctores del área de consulta externa procedieron a manipular el módulo de aplicación Turnos Doctor en donde se registran todos los turnos que deben ser atendidos y aquellos que están reservados, finalizando con la obtención de reportes por medio del sistema.
Aprobación	Los asistentes administrativos del área aprobaron la aplicación, y sugirieron realizar cambios menores.



Según los resultados obtenidos al aplicar las pruebas de validación, no se tuvo mayores sugerencias, ni críticas sobre fallos o incumplimiento de requerimientos, por lo que la fase de pruebas queda concluida y se da por aceptada la aplicación desarrollada.

Según las gráficos estadísticos se puede observar que el usuario da una calificación Muy Buena y Buena a la aplicación con lo que podemos concluir que el sistema si cumple el propósito para el cual fue realizado.



g. Discusión

1. Desarrollo de la propuesta alternativa

Los objetivos específicos que a continuación se detallan así como el objetivo general fueron cumplidos en su totalidad gracias al correcto análisis de requerimientos y diseño basado en la metodología de desarrollo Iconix. A continuación se enuncian cada uno de los objetivos planteados así como también se describe el proceso que se realizó para dar cumplimiento a los mismos.

Objetivo Específico 1: Ensamblar un dispositivo electrónico basado en diseños y Esquemas abiertos, usando componentes reciclados.

Para el cumplimiento de este objetivo en su totalidad se utilizó la información necesaria acerca de la construcción y montaje de una placa Arduino mediante consultas bibliográficas [13] [14], recolectadas durante la etapa del análisis, se elaboró el diagrama esquemático del controlador utilizando la herramienta EAGLE versión 5.4.0 el mismo que fue diseñado en la etapa del diseño, como también se adquirió el tipo de placa Arduino Duemilanove puesto que fue la que mejor se ajustó a los requerimientos y necesidades del proyecto de titulación. Además se construyó dos paneles de 7 segmentos que conectados a la placa Arduino mediante un controlador permitirán evidenciar el número de turno que será atendido.

Objetivo Específico 2: Diseñar e implementar un software de administración para definir la lógica del funcionamiento autónomo del dispositivo electrónico.

Para dar cumplimiento a este objetivo se hizo primeramente una interacción lógica entre el Arduino y displays mediante el controlador digital que sirvió de interfaz entre el computador y los displays, siendo en esta parte donde el Arduino interviene, ofreciendo por un lado, una interfaz USB serial para la comunicación con el software del sistema y por otro lado 15 salidas digitales para el control del dispositivo, se ha utilizado 9 líneas digitales para el control del display, las 7 primeras controlan el carácter a representar en el display y la línea 8 y 9 controlan en que display se va a presentar dicho carácter, permitiendo unamultiplexación de salidas para la representación de dígitos.



Para el desarrollo del software del Arduino se tomó en cuenta los siguientes objetivos:

- ✓ Definir e inicializar las salidas del Arduino (9)
- ✓ Crear un espacio de memoria para el almacenamiento de las definiciones de cada carácter, del 0 al F
- ✓ Leer la entrada del USB para capturar eventos del software
- ✓ Decodificar la entrada capturada en el USB y
- ✓ Representar la entrada en los displays

Además se requiere de lenguaje C tradicional permitiendo generar un software encargado de realizar tareas de automatización; esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie que es la principal característica que lo identifica, se utilizó **Ubuntu** que es un sistema operativo basado en Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity.

Al cumplir con este objetivo se puede visualizar el comportamiento del dispositivo genérico en forma real y verificar su respuesta o posibles fallas, proporcionando una idea del performance y rendimiento del hardware.

Objetivo Específico 3: Diseñar y desarrollar el software de funcionamiento inteligente del proceso de gestión de colas, para la asignación de turnos.

Este objetivo se cumplió en su totalidad debido a que en base a la información obtenida en la etapa del análisis y tomando en cuenta algunos elementos básicos necesarios para diseñar un modelo de colas como:

- Forma en que los clientes llegan al punto de servicio:
- Forma como se realiza el servicio:
- Modo de elegir los clientes de la fila que espera el servicio:

Se pudo optar por el tipo de disciplina de cola FIFO (first in first out) primero en entrar, primero en salir, según la cual se atiende primero al cliente que antes haya llegado, el mismo que se ajustó a los requerimientos del proyecto de titulación, permitiendo establecer un balance equilibrado (“óptimo”) entre las consideraciones cuantitativas de



costes y las cualitativas de servicio, mejorando así el proceso de asignación de turnos en el área de consulta externa del hospital.

Objetivo Específico 4: Diseñar y construir un sistema de notificación de estados de turnos hacia los terminales remotos registrados en el dispositivo.

Para cumplir con este objetivo primeramente se definieron bien las características del sistema con el usuario, aplicando la técnica de la entrevista, con el uso de la metodología Iconix, se realizó el prototipado final, la identificación de los casos de uso del sistema y sus actores, mostrando el conjunto de funcionalidades de cada uno, facilitando de esta manera a través del software desarrollado una mejor visualización del turno en el display y consecuentemente modificó el modelo de trabajo y la disponibilidad de las personas que lo utilicen con respecto a sus tiempos de ejecución.

2. Valoración técnica económica ambiental

El software final cumple con todas las necesidades y requerimientos expuestos al inicio del proyecto, debido a que se utilizaron métodos y herramientas para su desarrollo de muy buena calidad.

Los costos para el desarrollo del sistemas son medianamente bajos debidos a que se utilizó software y hardware libre como es lenguaje C, base de datos MySQL, así como la librería wxWidgets, en lo que a hardware se refiere se utilizó la Placa Arduino.

Finalmente los costos de fabricación en su totalidad fueron asumidos por los desarrolladores, estos son:

TABLA XXV
RECURSOS HUMANOS

RECURSOS HUMANOS				
DESCRIPCIÓN	CANTIDAD	Nº HORAS	V. UNITARIO	V. TOTAL
<u>Tesistas:</u>				
✓ Alba Pineda	2	3240	2.00	12960
✓ Carmen Becerra				



<u>Expertos:</u>				
✓ Personal administrativo del Hospital Isidro ayora	-----	-----	-----	-----
✓ Médicos del Hospital Isidro ayora	-----	-----	-----	-----
Asesor Profesional	2	120	8.50	\$ 2040.00
Total:				\$ 15000.00

TABLA XXVI
RECURSOS TÉCNICOS Y TECNOLÓGICOS

RECURSOS TÉCNICOS Y TECNOLÓGICOS				
DESCRIPCIÓN	CANTIDAD	Nº HORAS	V. UNITARIO	V. TOTAL
<u>Software:</u>				
✓ Lenguaje de programación: C++	1	-----	-----	Gratuito
✓ Gestor de Base de Datos: MySQL	1	-----	-----	Gratuito
✓ WXWidgets	1	-----	-----	Gratuito
✓ UML para el desarrollo de diagramas de clases del sistema	1	-----	-----	Gratuito
<u>Hardware:</u>				
✓ Memoria Flash	1	-----	\$ 19	\$ 19



16 GB. ✓ Memoria Flash	1	-----	\$ 12	\$ 12
8 GB. ✓ Placa Arduino	1	-----	\$ 650	\$ 650
✓ Computador portátil AMD Sempron (disco duro 160GB., memoria 2GB.)	1	-----	\$ 1300	\$ 1300
✓ Computador de escritorio Intel Pentium 4 (disco duro 120GB., memoria 510MB.)	1	-----	\$ 850	\$ 850
✓ Impresora Canon ip1000	1	-----	\$ 80	\$ 80
<u>Comunicaciones:</u>				
✓ Internet	-----	600	0.80	\$ 480
Total:				\$ 3391

TABLA XXVII
RECURSOS MATERIALES

RECURSOS MATERIALES				
DESCRIPCIÓN	CANTIDAD	Nº HORAS	V. UNITARIO	V. TOTAL
<u> Materiales de Oficina:</u>				
✓ Cartuchos para la impresora	13	-----	\$ 3.50	\$ 45.50
✓ Perforadora	1	-----	\$ 6.00	\$ 6.00



✓ Grapadora	1	-----	\$ 4.50	\$ 4.50
✓ Caja de grapas	1	-----	\$ 1.25	\$ 1.25
✓ Caja de clips	1	-----	\$ 0.30	\$ 0.30
✓ Caja de Cds.	1	-----	\$ 3.00	\$ 3.00
✓ Esferográficos	5	-----	\$ 0.25	\$ 1.25
✓ Anillados	6	-----	\$ 1.75	\$ 10.50
✓ Carpetas	5	-----	\$ 0.50	\$ 2.50
✓ Copias	300	-----	\$ 0.02	\$ 6.00
✓ Resmas de papel bond	8	-----	\$ 3.50	\$ 28.00
<u>Servicios Básicos:</u>				
✓ Transporte		-----	-----	150
✓ Electricidad		-----	-----	80
✓ Agua		-----	-----	10
✓ Teléfono		-----	-----	30
Total:				\$ 378.80

TABLA XXVIII
RESUMEN DE COSTOS

RÉSUMEN DE COSTOS	
RECURSO	VALOR TOTAL
Recursos Humanos	\$ 15000.00
Recursos Técnicos y Tecnológicos	\$ 3391.00
Recursos Materiales	\$ 378.80
SUBTOTAL:	\$ 18769.80
IMPREVISTOS 10%:	\$ 1876.98
TOTAL:	\$ 20646.78



h. Conclusiones

- Con el desarrollo de este sistema se ha contribuido a mejorar el nivel de satisfacción de los pacientes y usuarios gracias a la automatización, organización y optimización del proceso de entrega de turnos, permitiendo de esta manera obtener información accesible, rápida y pertinente lo que se pudo comprobar en las pruebas de validación logrando que el usuario ahorre tiempo en buscar en otros medios los recursos que necesita.
- La Teoría de Colas nos permitió conocer información útil sobre el funcionamiento de los sistemas de espera basándose en los tiempos entre llegadas y en los tiempos de servicio, ofrece ayuda para ajustar el tiempo de espera medio y el número de clientes medio tanto en cola como en el sistema a valores asumibles para una institución.
- De la unión de plataformas de software y hardware abierto, nacen proyectos que suponen una ventaja para usuarios y desarrolladores al disponer estos de mayor libertad para crear e instalar aplicaciones al mismo tiempo que adaptar algunas funciones del hardware.
- Es importante realizar diseños flexibles del sistema de tal forma que éste tenga la posibilidad de crecer a futuro, acoplando nuevos módulos para las distintas Áreas del Hospital.



i. Recomendaciones

- Se recomienda adaptar un módulo para la digitalización de las Historias Clínicas y su sistematización informática, a fin de que los médicos tengan acceso a ellas a través de una red interna; de esta manera se elimina la pérdida del tiempo y en ocasiones de las Historias Clínicas.
- La administración del hospital y su personal deberán tener una participación más activa en la elaboración de su plan estratégico y operativo, con la finalidad de que se involucren en la formulación y consecución de sus objetivos.
- Se recomienda establecer un procedimiento para el aviso previo al cumplimiento de una cita, es decir, se alerte al paciente mediante una central telefónica indicando la hora, fecha y doctor de la cita.
- Se recomienda tener en cuenta dos cuestiones principales que hacen la gran diferencia a la hora de elegir el tipo de modelo de una placa arduino: la primera es el tipo de microcontrolador a utilizar, y la segunda es el modo de comunicación que poseerá la placa Arduino con el ordenador, las mismas que permitirá elegir la mejor que se adapte al desarrollo del sistema



J. Bibliografía

- [1] HERBERT SCHILDT. Guía de Auto enseñanza. México. OSEOBRENE/MSGRAN_WILL. Aplique Turbo C++ para Windows. México, [Accedido: 10 de Octubre del 2011].
- [2] Alberto Pacheco. **Sistema Inteligente**, http://www.depi.itch.edu.mx/apacheco/ai/s_intel.htm (1999), [Accedido: 08 de Octubre, 2009].
- [3] **BrunoProg 64. Hardware libre**, http://www.ecured.cu/index.php/Hardware_libre, (agosto 2006), [Accedido: 08 de Octubre, 2009].
- [4] Nacho Morato. **Definición de placa Arduino**, <http://www.ikkaro.com/definicion-arduino/>, (2009), [Accedido: 23 de Octubre, 2012].
- [5] Jonathan Amangandi. **Definición de placa Arduino**, <http://jamangandi2012.blogspot.com/2012/10/que-es-arduino-te-lo-mostramos-en-un.html#uds-search-results>, (2012), [Accedido: 23 de Octubre, 2012].
- [6] Zoe Romano. **Uso de placa Arduino**, <http://arduino.cc/es/Guide/Introduction>, (jun 2012), [Accedido: 23 de Octubre, 2012].
- [7] Zoe Romano. **Versiones de la placa Arduino**, <http://arduino.cc/es/Main/Boards>, (2012), [Accedido: 08 de Octubre, 2012].
- [8] Zoe Romano. **Características físicas de la Placa Duemilanove**, <http://arduino.cc/es/Main/arduinoBoardDuemilanove>, (2012), [Accedido: 08 de Octubre, 2012].
- [9] Zoe Romano. **Elementos necesarios para trabajar con Arduino Duemilanove**, <http://arduino.cc/es/Main/arduinoBoardDuemilanove>, (2012), [Accedido: 08 de Octubre, 2012].
- [10] Percy_Ivan. **Definición de un display de 7 segmentos**,



- http://www.unicrom.com/Tut_display-7-segmentos.asp, (2006), [Accedido: 08 de Octubre, 2009].
- [11] Instituto de Ingeniería UNAM. **Funcionamiento de un display de 7 segmentos**,http://www.ingenieria.unam.mx/crofi/wp-content/uploads/Descargas/Datasheet/Displays/display_de_7_segmentos.pdf[Accedido: 02 de Octubre, 2009].
- [12]Joel Burton. **Definición de Software Libre**, <http://www.libre.org/es/libre/software-libre/definicion-de-software-libre>, (2009) [Accedido: 06 de Octubre, 2009].
- [13]Isaac P.E. **Entorno de desarrollo de la Placa ArduinoDuemilanove**,
<http://www.linuxadictos.com/arduino-ide-y-ardublock-como-instalarlos-en-linux.html>,
(2013),[Accedido: 10 de Octubre, 2013].
- [14] Rodrigo Loyola.**Definición de WxWidgest**,
<http://www.google.com.ec/url?url=http://profesores.elo.utfsm.cl/~agv/elo329/1s05/projects/Rodrigo.Loyola/presentacion.ppt&rct=j&frm=1&q=&esrc=s&sa=U&ei=eLuYU7yYCcXmsATzmYC4BQ&ved=0CEIQFjAJ&usg=AFQjCNFCYN4BjsbPjQIGGHFx0btz4KijXg>, (2007), [Accedido: 26 de Octubre, 2009.]
- [15] Pueril. **Trabajando con WxWidgest**,
<http://wxwidgets.wikispaces.com/Cap%C3%ADtulo+2+-+Primeros+Pasos>, (Feb2012),
[Accedido: 10 de Jun, 2012].
- [16] Jhon James Quintero Osorio. **Introducción a la programación con WxWidgest**,<http://www.freewebs.com/jojaqui/wxWidgetsCB.pdf>, (2008) [Accedido: 26 de Octubre, 2009].
- [17] Alex Allain.**Introduccion a g++**, <http://www.cprogramming.com/g++.html>, (1998),
[Accedido: 12 de Octubre, 2009].
- [18] Kervin Vergara. **Definición del lenguaje C**,<http://www.bloginformatico.com/lenguaje-de-programacion-c.php>, (2007), [Accedido: 12 de Octubre, 2009].



- [19] ZatorSystems. **Propiedades de C**,
http://www.zator.com/Cpp/E1_2.htm, (2007), [Accedido: 15 de Octubre, 2009].
- [20] *ThemesVermilion Christmas*. **Definición de Algoritmo Voraz**,
<http://danubuntu.wordpress.com/2008/11/11/algoritmos-voraces>, (2008), [Accedido: 12 de Noviembre, 2009].
- [21] María Teresa Abad Soriano. **Esquema y elementos de un algoritmo voraz**,
<http://www.lsi.upc.edu/~ada/apunts/MTA/greedy.pdf>, 2007), [Accedido: 12 de Noviembre, 2009].
- [22] Alberto Zaragoza Heredia. **Teoría de Colas**,
http://exa.unne.edu.ar/informatica/evalua/Sitio%20Oficial%20ESPD-Temas%20Adicionales/teoria_de_colas.pdf, (2007), [Accedido: 17 de Noviembre, 2009].
- [23] Martínez Ferreira. **Introducción a la Teoría de Colas**,
<http://www.monografias.com/trabajos18/teoria-colas/teoria-colas.shtml>, (2006), [Accedido: 17 de Noviembre, 2009].
- [24] Lucila Stancato. **Definición de Teoría de Colas**,
http://exa.unne.edu.ar/depar/areas/informatica/evalua/teoria_de_colas.pdf, (2006), [Accedido: 18 de Noviembre, 2009].
- [25] Molina Segovia Jaime Ernesto, “**Desarrollo e implantación de un Sistema de Gestión de Turnos para Consulta Externa del Hospital de Especialidades Eugenio Espejo.**”
[http://eelalnx01.epn.edu.ec/bitstream/15000/924/1/CD-1826\(2009-01-21-11-40-25\).pdf](http://eelalnx01.epn.edu.ec/bitstream/15000/924/1/CD-1826(2009-01-21-11-40-25).pdf)
(2008), [Accedido: 18 de Diciembre, 2013].
- [26] Richard Vinicio Siza, **Diseño, desarrollo e implementación del sistema de gestión de turnos programados para el Sub-centro de salud Carapungo**,



k. Anexos

Anexo I

Certificado de autorización para realizar el proyecto en el Hospital Isidro Ayora



HOSPITAL PROVINCIAL GENERAL "ISIDRO AYORA"

Loja, 06 de Marzo de 2009 LOJA - ECUADOR

Doctor
Edgar Armijos
DIRECTOR DEL HOSPITAL PROVINCIAL GENERAL
ISIDRO AYORA

CERTIFICO:

Que la Srtas. Alba del Roció Pineda Torres y Carmen Elizabeth Becerra González egresadas de la carrera de Ingeniería en sistemas de la Universidad Nacional de Loja, ha presentado su petición en esta Casa de Salud para facilitar la información requerida para la construcción e Implementación de un sistema inteligente de Asignación Dinámica de turnos en el Área de Consulta Externa en el HOSPITAL PROVINCIAL GENERAL ISIDRO AYORA, la cual fue aprobada.

Particular que certifico para los fines consiguientes.

Dr. Edgar Armijos
DIRECTOR DEL HOSPITAL PROVINCIAL GENERAL
ISIDRO AYORA



Anexo II

Certificado de traducción del resumen

Loja 26 de Febrero del 2014

Lcda. Yesenia Ruiz L.
DOCENTE EN IDIOMA INGLES

Certifico que:

Las egresadas Alba del Rocío Pineda con número 1104125255 y Carmen Elizabeth Becerra González con número de cedula 1103780407 de la carrera de Ingeniería en Sistema de la Universidad Nacional de Loja , autoras del proyecto de fin de carrera denominado: **"CONSTRUCCIÓN DE UN SISTEMA INTELIGENTE PARA LA ASIGNACIÓN DINÁMICA DE TURNOS EN EL ÁREA DE CONSULTA EXTERNA DEL HOSPITAL REGIONAL ISIDRO AYORA, MEDIANTE EL ANÁLISIS HEURÍSTICO UTILIZANDO HARDWARE Y SOFTWARE LIBRE"**, han realizado la traducción del resumen, cumpliendo con todas las normas y reglas gramaticales al idioma inglés, las cuales han sido revisadas minuciosamente para dar cumplimiento con la sección **Summary**.

Es todo cuanto puedo decir en honor a la verdad, pudiendo los interesados hacer uso del presente en lo que se estime conveniente.

Atentamente



Lcda. Yesenia RUIZ L.



Anexo III

Encuesta al Administrador y Asistentes Administrativos del Departamento de Archivo del Hospital Isidro Ayora

UNIVERSIDAD NACIONAL DE LOJA

SISTEMA INTELIGENTE PARA LA ASIGNACIÓN DINÁMICA DE TURNOS EN EL ÁREA DE CONSULTA EXTERNA DEL HOSPITAL REGIONAL ISIDRO AYORA, MEDIANTE EL ANÁLISIS HEURÍSTICO UTILIZANDO HARDWARE Y SOFTWARE LIBRE

FICHA DE VALIDACIÓN DE LA APLICACIÓN

Nº	FV 01
Usuarios	Administrador y Asistentes Administrativos

Instrucciones:

Conteste a las siguientes preguntas señalando con una X la respuesta de acuerdo al grado de medición de aceptación del sistema, en caso de no ser aceptable escribir las razones por las cuales este sistema deba mejorar.

Grado de medición

- ✓ Muy Bueno
- ✓ Bueno
- ✓ Regular
- ✓ Malo
- ✓ Muy Malo

1. ¿Considera Ud., que la aplicación es amigable con el usuario?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()



¿Por qué?:.....

2. ¿Considera Ud., que el diseño de la aplicación es apropiado para realizar la ejecución de sus actividades?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....

3. ¿Considera Ud., que el tiempo de ingreso a las opciones del sistema es razonable?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....

4. ¿El sistema posee las funcionalidades que fueron requeridas al inicio del proyecto?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....

5. ¿Tuvo alguna dificultad para utilizar las opciones de la aplicación?

Solicitar Turno	Muy Bueno ()	Bueno ()	Regular ()	Malo ()	Muy Malo ()
Administrar Turno	Muy Bueno ()	Bueno ()	Regular ()	Malo ()	Muy Malo ()
Administrar Horario-Doctores	Muy Bueno ()	Bueno ()	Regular ()	Malo ()	Muy Malo ()
Administrar Doc-Esp	Muy Bueno ()	Bueno ()	Regular ()	Malo ()	Muy Malo ()
Generar Reportes	Muy Bueno ()	Bueno ()	Regular ()	Malo ()	Muy Malo ()
Buscar paciente	Muy Bueno ()	Bueno ()	Regular ()	Malo ()	Muy Malo ()
Administrar Usuarios	Muy Bueno ()	Bueno ()	Regular ()	Malo ()	Muy Malo ()

¿Por qué?:.....

6. ¿Los mensajes que devuelve el sistema son lo suficientemente descriptivos?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....

7. ¿Los reportes que se emiten reflejan resultados reales?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....



Anexo IV

Encuesta a Doctores del Hospital Isidro Ayora

UNIVERSIDAD NACIONAL DE LOJA

SISTEMA INTELIGENTE PARA LA ASIGNACIÓN DINÁMICA DE TURNOS EN EL ÁREA DE CONSULTA EXTERNA DEL HOSPITAL REGIONAL ISIDRO AYORA, MEDIANTE EL ANÁLISIS HEURÍSTICO UTILIZANDO HARDWARE Y SOFTWARE LIBRE

FICHA DE VALIDACIÓN DE LA APLICACIÓN

Nº	FV 02
Usuarios	Doctores del Hospital Regional Isidro Ayora

Instrucciones:

Conteste a las siguientes preguntas señalando con una X la respuesta de acuerdo al grado de medición de aceptación del sistema, en caso de no ser aceptable escribir las razones por las cuales este sistema deba mejorar.

Grado de medición

- ✓ Muy Bueno
- ✓ Bueno
- ✓ Regular
- ✓ Malo
- ✓ Muy Malo

1. ¿Considera Ud., que la aplicación es amigable con el usuario?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....



2. ¿Considera Ud., que el diseño de la aplicación es apropiado para realizar la ejecución de sus actividades?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....

3. ¿Considera Ud., que el tiempo de ingreso al módulo (Turnos Doctor) es razonable?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....

4. ¿El sistema posee las funcionalidades que fueron requeridas al inicio del proyecto?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....

5. ¿Tuvo alguna dificultad para utilizar el módulo (Turnos Doctor) del sistema?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....

6. ¿El tiempo que demora en presentarse el número del turno en el display es razonable?

Muy Bueno () Bueno () Regular () Malo () Muy Malo ()

¿Por qué?:.....



Anexo V

Análisis de los resultados de las pruebas de validación

Ficha de validación para el Administrador y Asistentes Administrativos.

1. ¿Considera Ud., que la aplicación es amigable con el usuario?

TABLA I
PRESENTACION
DE LA APLICACION

MUY BUENO	4
BUENO	0
REGULAR	1
MALO	0
MUY MALO	0
TOTAL	5

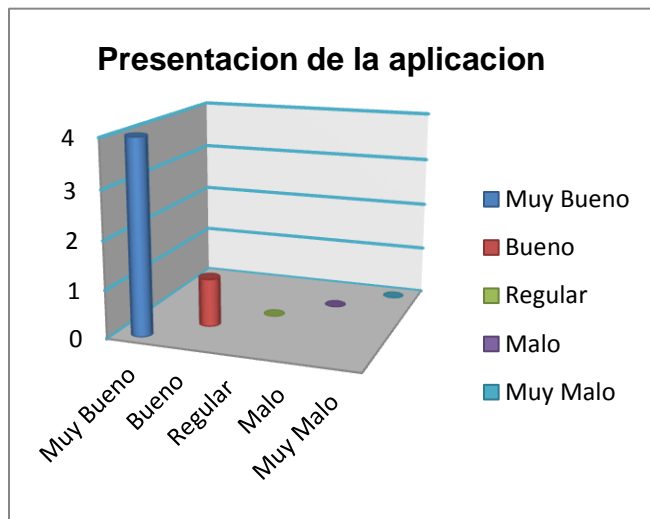


Figura 1: Presentación de la aplicación

2. ¿Considera Ud., que el diseño de la aplicación es apropiado para realizar la ejecución de sus actividades?

TABLA II
DISEÑO DE LA APLICACIÓN
APROPIADO PARA LA
EJECUCIÓN DE ACTIVIDADES

MUY BUENO	2
BUENO	3
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

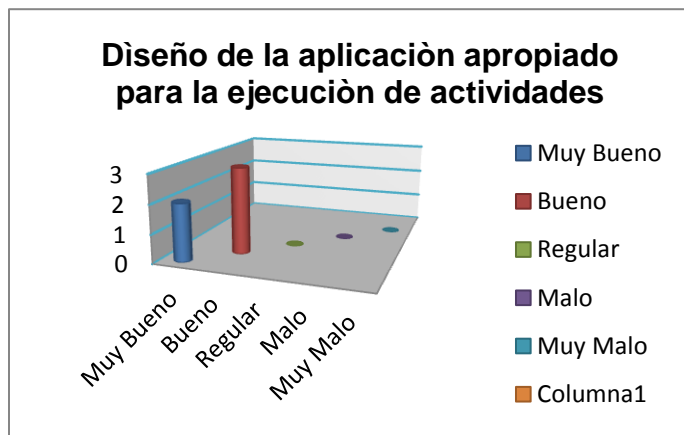




Figura 2: Diseño de la aplicación apropiado para la ejecución de actividades.

3. ¿Considera Ud., que el tiempo de ingreso a las opciones del sistema es razonable?

TABLA III
TIEMPO DE INGRESO A OPCIONES DEL SISTEMA

MUY BUENO	3
BUENO	2
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

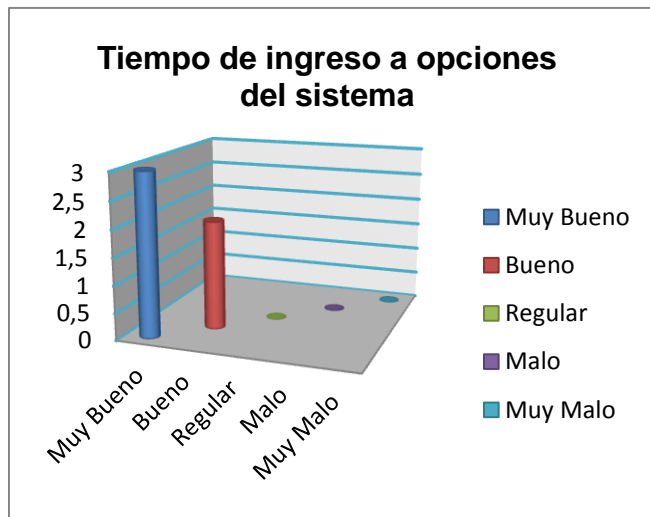


Figura 3: Tiempo de ingreso a opciones del sistema

4. ¿El sistema posee las funcionalidades que fueron requeridas al inicio del proyecto?

TABLA IV
POSEER LAS FUNCIONALIDADES QUE FUERON REQUERIDAS

MUY BUENO	1
BUENO	4
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

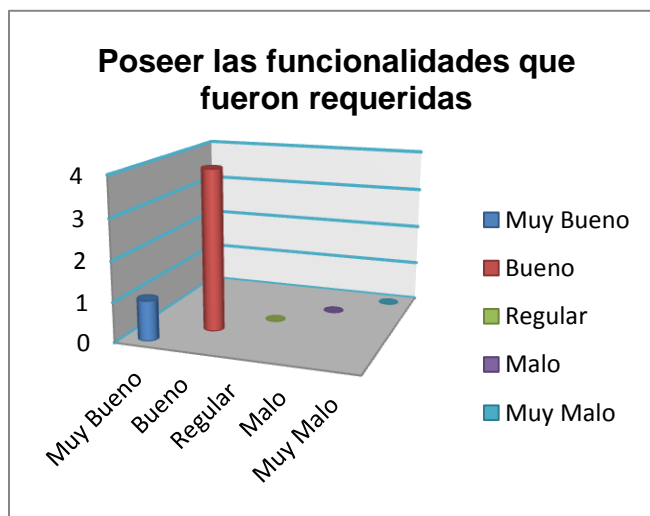


Figura 4: Poseer las funcionalidades que fueron requeridas.



5. ¿Tuvo alguna dificultad para utilizar los módulos de la aplicación?

TABLA V
DIFICULTAD AL SOLICITAR
TURNO

Solicitar Turno	
MUY BUENO	2
BUENO	3
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

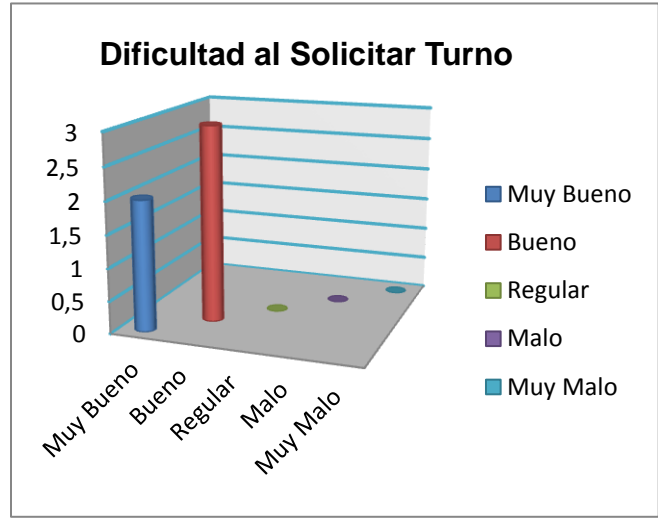


Figura 5: Dificultad al Solicitar Turno.

TABLA VI
DIFICULTAD AL ADMINISTRAR
TURNO

Administrar Turno	
MUY BUENO	3
BUENO	2
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

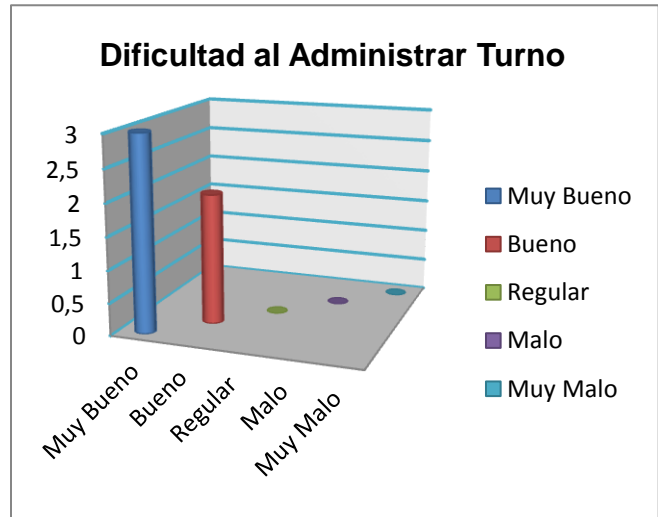


Figura 6: Dificultad al Administrar Turno.



TABLA VII
DIFICULTAD AL ADMINISTRAR
HORARIO-DOCTORES

Administrar Horario-Doctores	
MUY BUENO	1
BUENO	3
REGULAR	1
MALO	0
MUY MALO	0
TOTAL	5

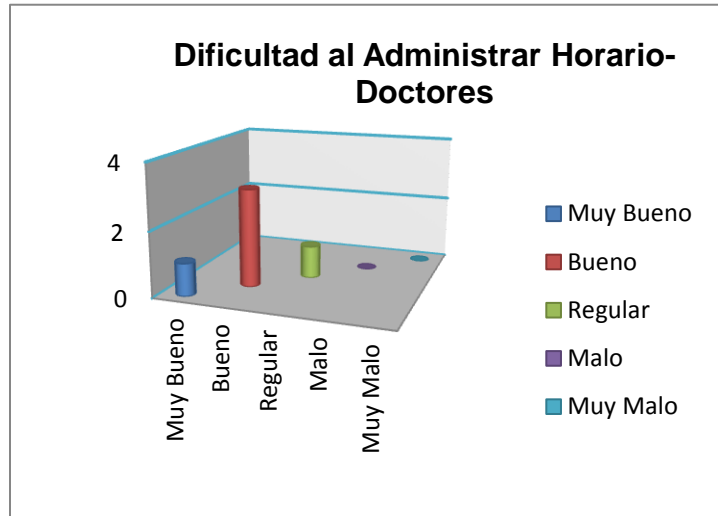


Figura 7: Dificultad al Administrar Horario-Doctores

TABLA VIII
DIFICULTAD AL ADMINISTRAR
DOC-ESP

Administrar Doc-Esp	
MUY BUENO	0
BUENO	4
REGULAR	1
MALO	0
MUY MALO	0
TOTAL	5

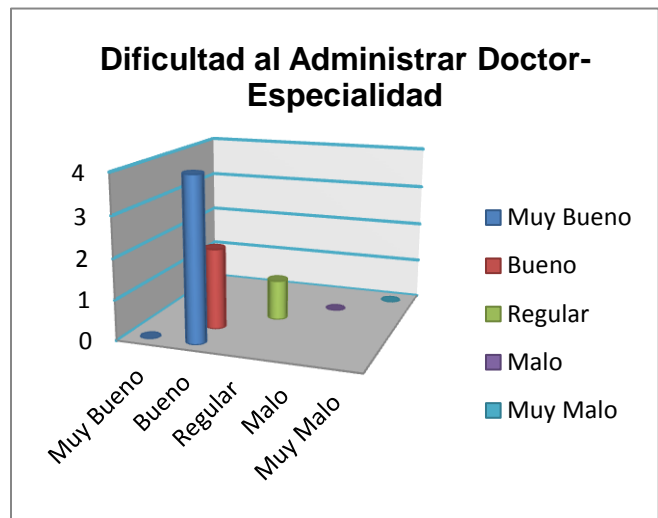


Figura 8: Dificultad al Administrar Doctor- Especialidad



TABLA IX
DIFICULTAD AL GENERAR
REPORTES

Generar Reportes	
MUY BUENO	5
BUENO	0
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

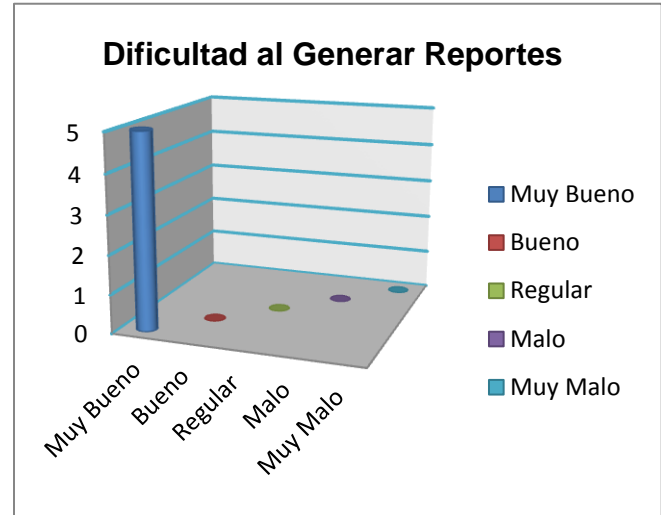


Figura 9: Dificultad al Generar Reportes

TABLA X
DIFICULTAD AL BUSCAR
PACIENTE

Buscar Paciente	
MUY BUENO	4
BUENO	1
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

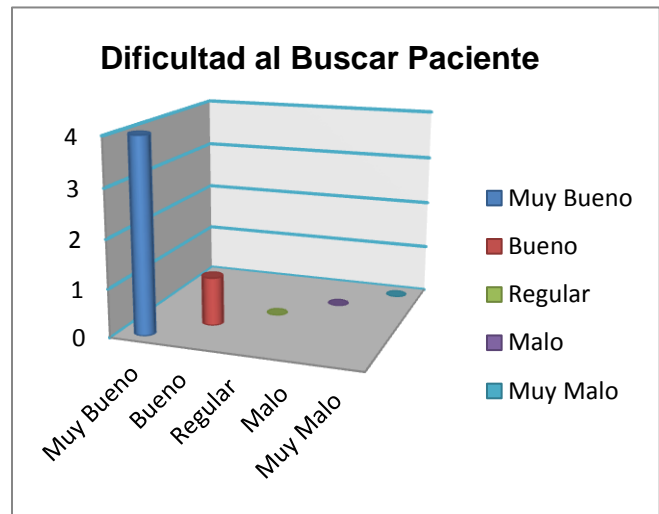


Figura 10: Dificultad al Buscar Paciente



TABLA XI
DIFICULTAD AL ADMINISTRAR
USUARIOS

Administrar Usuarios	
MUY BUENO	1
BUENO	0
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

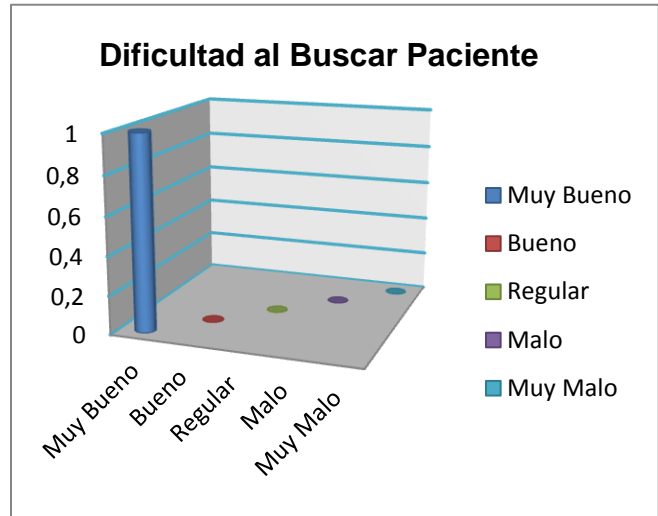


Figura 11: Dificultad al Administrar Usuarios

6. ¿Los mensajes que devuelve el sistema son lo suficientemente descriptivos?

TABLA XII
MENSAJES DESCRIPTIVOS

MUY BUENO	4
BUENO	1
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

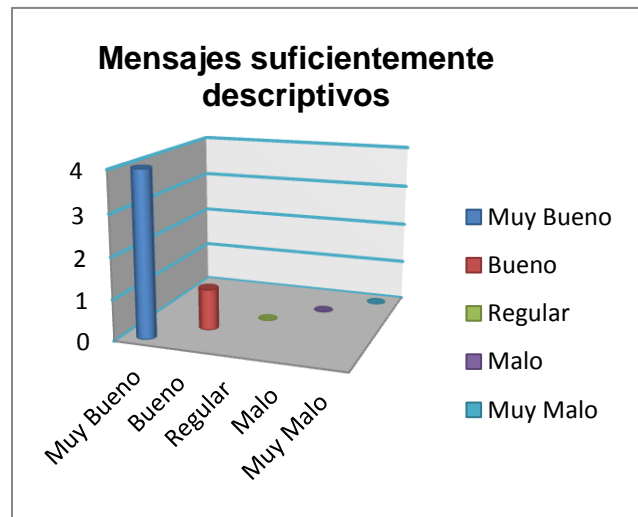


Figura 12: Mensajes suficientemente descriptivos



7. ¿Los reportes que se emiten reflejan resultados reales?

TABLA XIII
REPORTES REFLEJAN
RESULTADOS REALES

MUY BUENO	4
BUENO	1
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	5

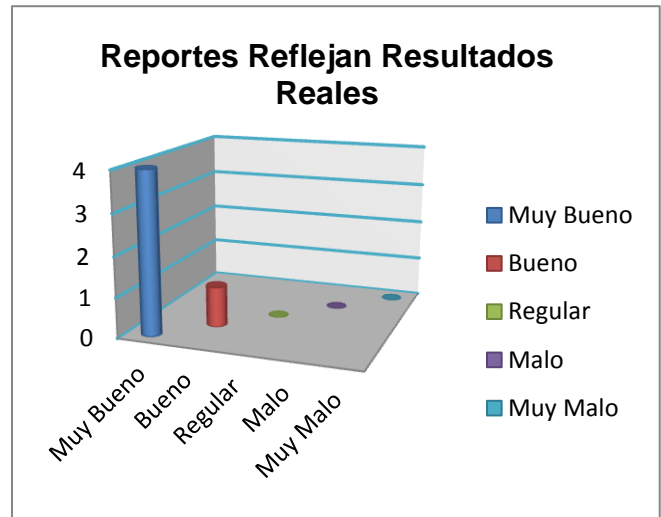


Figura 13: Reportes reflejan resultados reales



Ficha de validación para los Doctores del Área de Consulta Externa.

1. ¿Considera Ud., que la aplicación es amigable con el usuario?

TABLA XIV
PRESENTACION DE LA APLICACIÓN
DOCTOR

MUY BUENO	0
BUENO	10
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	10

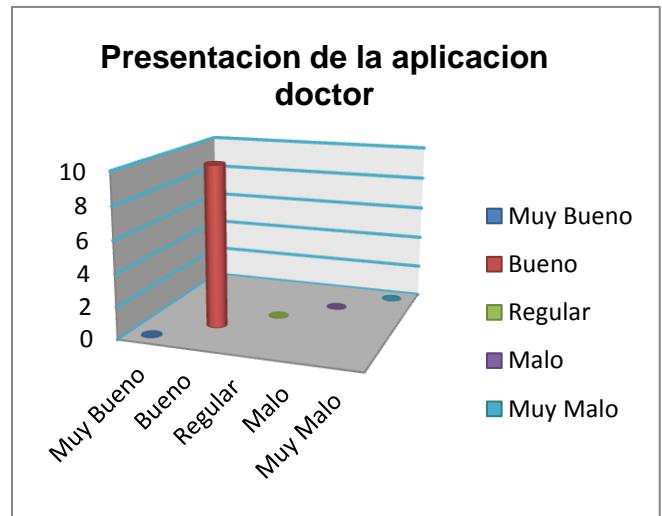


Figura 14: Presentación de la aplicación doctor

2. ¿Considera Ud., que el diseño de la aplicación es apropiado para realizar la ejecución de sus actividades?

TABLA XV
DISEÑO DE LA APLICACIÓN
DOCTOR

MUY BUENO	0
BUENO	4
REGULAR	6
MALO	0
MUY MALO	0
TOTAL	10

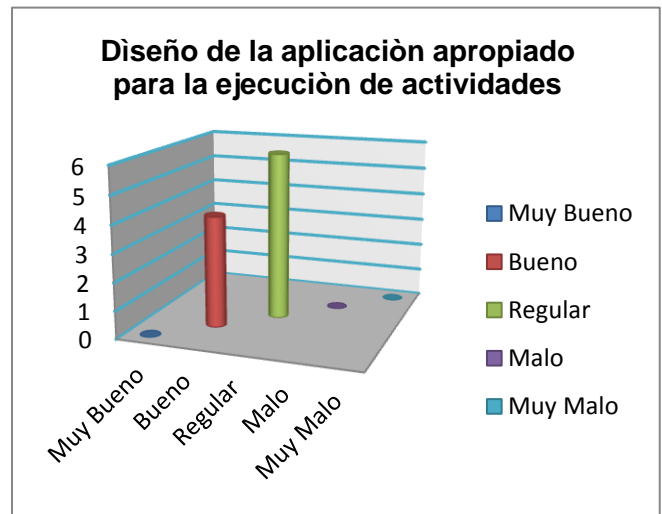




Figura 15: Diseño de la aplicación doctor

3. ¿Considera Ud., que el tiempo de ingreso al módulo (Turnos Doctor) es razonable?

TABLA XVI
TIEMPO DE INGRESO MODULO
TURNOS DOCTOR

MUY BUENO	0
BUENO	10
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	10

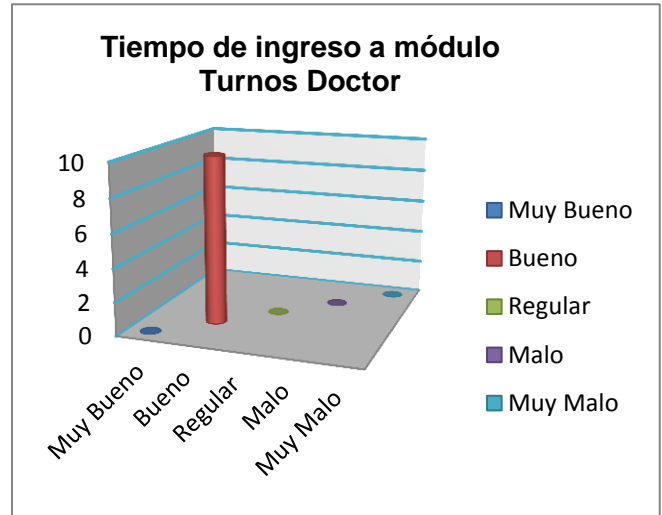


Figura 16: Tiempo de ingreso modulo turnos doctor

4. ¿El sistema posee las funcionalidades que fueron requeridas al inicio del proyecto?

TABLA XVII
FUNCIONALIDADES REQUERIDAS
MODULO TURNOS DOCTOR

MUY BUENO	0
BUENO	6
REGULAR	4
MALO	0
MUY MALO	0
TOTAL	10

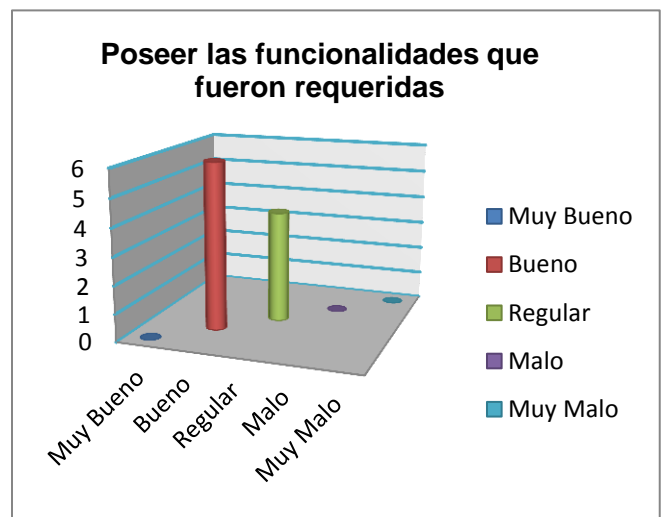


Figura 17: Funcionalidades requeridas modulo Turnos Doctor



5. ¿Tuvo alguna dificultad para utilizar el módulo (Turnos Doctor) del sistema?

TABLA XVIII
DIFICULTAD USO MODULO
TURNOS DOCTOR

MUY BUENO	0
BUENO	10
REGULAR	0
MALO	0
MUY MALO	0
TOTAL	10

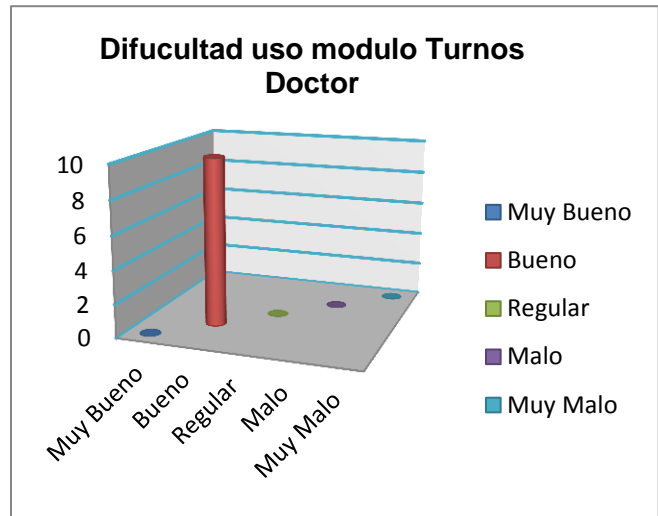


Figura 18: Dificultad uso modulo Turnos Doctor

6. ¿El tiempo que demora en presentarse el número del turno en el display es razonable?

TABLA XIX
TIEMPO EN PRESENTAR
NUMERO DE TURNO EN
DISPLAY

MUY BUENO	0
BUENO	9
REGULAR	1
MALO	0
MUY MALO	0
TOTAL	10

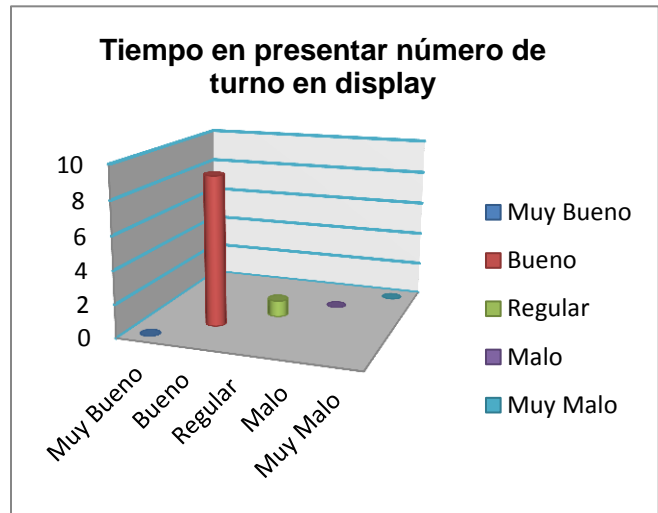


Figura 19: Tiempo en presentar número de turno en display



Anexo VI

Certificado de presentación y aprobación del sistema en el Hospital Isidro Ayora

 Ministerio de Salud Pública
HOSPITAL GENERAL ISIDRO AYORA
Gestión Asistencial



Loja, 16 de diciembre de 2013

Doctor JORGE GUAPULEMA OCAMPO
Director Asistencial del Hospital Isidro Ayora Loja

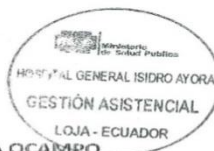
A petición de parte interesada:

CERTIFICA:

Que la Srtas. Alba del Rocío Pineda Torres y Carmen Elizabeth Becerra González egresadas de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, han presentado un sistema informático denominado Sistema inteligente de Asignación Dinámica de turnos en el Área de Consulta Externa del HOSPITAL PROVINCIAL GENERAL ISIDRO AYORA el cual luego de realizarle las pruebas pertinentes se ha demostrado que cumple con los requerimientos establecidos por la institución y los objetivos planteados por las Srtas.

Particular que certifico para los fines consiguientes.


Doctor JORGE GUAPULEMA OCAMPO
Director Asistencial del Hospital Isidro Ayora Loja



Av. Manuel Agustín Aguirre y Juan José Samaniego
Teléfono: 2570540 ext. 7279
<http://instituciones.msp.gob.ec/dps/loja/>



Anexo VII

Declaración de confidencialidad.

Alba del Rocío Pineda Torres con cédula de identidad 1104125255 y Carmen Elizabeth Becerra González con cedula de identidad 1103780407(en adelante: las declarantes).

DECLARAN lo siguiente:

PRIMERO: Antecedentes

1. Las declarantes van a participar y han participado en el trabajo de titulación “Construcción de un Sistema Inteligente para la Asignación Dinámica de turnos en el Área de Consulta Externa del Hospital Regional Isidro Ayora, mediante el Análisis Heurístico utilizando Hardware y Software Libre” dirigido por el Ing. Edwin René Guamán Quinche, en calidad de director de trabajo.
2. Por el presente documento se regula el tratamiento que las declarantes han de dar la información a la que pueda tener acceso en el desarrollo de las tareas de investigación que se realicen en dicho trabajo, en el cual se regulará por las disposiciones contenidas en las cláusulas siguientes.

SEGUNDO: Información Confidencial

La información referida a materiales, métodos y resultados científicos, técnicos y comerciales utilizados u obtenidos durante la realización del trabajo de investigación o una vez realizado el mismo, se considerará siempre Información Confidencial.

Tercero: Excepciones

No será considerada como información confidencial:

- a) La información que las declarantes puedan probar que tenían en su legítima posesión con anterioridad al conocimiento de la Información Confidencial.
- b) La información que las declarantes puedan probar que era de dominio público en la fecha de la divulgación o pase a serlo, con posterioridad, por haberse publicado o por otro medio, sin intervención ni negligencia de la declarante.



- c) La información que las declarantes puedan probar que corresponde en esencia a información facilitada por terceros, sin restricción alguna sobre su divulgación, en virtud de un derecho de las declarantes a recibirla.

CUARTO: Secreto de la Información Confidencial

Las declarantes se comprometen a mantener totalmente en secreto la información Confidencial recibida en relación con el trabajo referido anteriormente y no divulgarla a terceros durante la vigencia de esta Declaración de Confidencialidad.

Así mismo las declarantes se comprometen a emplear la Información Confidencial, exclusivamente, en el desempeño de las tareas que tengan encomendadas en dicho trabajo.

QUINTO: Declaración

La obligación de las declarantes respecto al mantenimiento del compromiso de secreto de la Información Confidencial, será indefinido para fines de investigación a partir de la fecha de la recepción de la Información Confidencial.

Loja, Noviembre 2014

Alba del Rocío Pineda Torres

Carmen Elizabeth Becerra González



Anexo VIII

Licencia Creative Commons



Construcción de un Sistema Inteligente para la Asignación Dinámica de turnos en el Área de Consulta Externa del Hospital Regional Isidro Ayora, mediante el Análisis Heurístico utilizando Hardware y Software Libre, desarrollado por Alba del Roció Pineda Torres y Carmen Elizabeth Becerra González se distribuye bajo una **Licencia Creative Commons Atribución-NonComercial-CompartirIgual 4.0 Internacional**



Anexo IX

Certificado de cumplimiento de normas y reglas gramaticales

Loja, 25 de Julio del 2014

Lcda. Julia Mariela León Rodríguez
DOCENTE DE LA ESCUELA "JULIO SERVIO ORDOÑEZ ESPINOSA"
DEL CANTÓN LOJA, PROVINCIA DE LOJA

CERTIFICA:

Que las Egresadas Alba del Rocío Pineda Torres con cédula Nro. 1104125255 y Carmen Elizabeth Becerra González con cédula Nro. 1103780407, autoras del presente proyecto de titulación "CONSTRUCCIÓN DE UN SISTEMA INTELIGENTE PARA LA ASIGNACIÓN DINÁMICA DE TURNOS EN EL ÁREA DE CONSULTA EXTERNA DEL HOSPITAL REGIONAL ISIDRO AYORA, MEDIANTE EL ANÁLISIS HEURÍSTICO UTILIZANDO HARDWARE Y SOFTWARE LIBRE", en la elaboración del mismo han cumplido con todas las normas y reglas gramaticales y ortográficas de la lengua Española.

Es todo cuanto puedo decir en honor a la verdad, pudiendo las interesadas hacer uso del presente en lo que estimen conveniente.



LICDA. Julia Mariela León Rodríguez

C. I. 1900208842

Nro. Reg. CENESCYT 1029-02-166662

LICENCIADA EN CIENCIAS DE LA EDUCACIÓN
ESPECIALIDAD LENGUA Y LITERATURA