



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE PACIENTES Y
PERSONAL MÉDICO DE LA CLÍNICA ODONTOLÓGICA DE LA UNIVERSIDAD DE
LAS AMÉRICAS.

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de
Ingenieros en Electrónica y Redes de Información

Profesor Guía

MIS. Eduardo Mauricio Campaña Ortega

Autores

Carlos Andrés Izurieta Dueñas

Edwin Patricio Vargas Bohórquez

Año

2017

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con los estudiantes, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

Eduardo Mauricio Campaña Ortega

Máster en Ingeniería de Software

CI: 1708856701

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

Carlos Andrés Muñoz Cueva
Magister en Gerencia de Sistemas
CI: 1712981511

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaramos que este trabajo es original, de nuestra autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Carlos Andrés Izurieta Dueñas

1720794617

Edwin Patricio Vargas Bohórquez

1722335203

AGRADECIMIENTOS

A mis padres y hermana quienes,
siempre me apoyaron y supieron
darme ánimos para lograr el
objetivo de graduarme de la
carrera.

A mi profesor guía Eduardo
Campaña, por su ayuda en el
diseño de este proyecto

Carlos Izurieta

AGRADECIMIENTOS

A mis padres y mis hermanas al igual que todos los amigos, compañeros y conocidos en todos estos años de estudio, que fueron parte importante en mi formación

A mi coordinador Ángel Jaramillo y a mi profesor guía por toda su colaboración en el desarrollo de este proyecto

Edwin Vargas

DEDICATORIA

Quisiera dedicar el presente proyecto a mis padres que con mucho sacrificio me ayudaron a seguir mis estudios.

A mi hermana por su apoyo incondicional para lograr este objetivo de vida

Carlos Izurieta

DEDICATORIA

Dedico el presente proyecto a todas las personas que me conocen, en especial a mis padres y hermanas que permitieron que todo esto fuera posible y por creer siempre en mi

Edwin Vargas

RESUMEN

El presente proyecto nace de la necesidad de la clínica odontológica de la Universidad de las Américas de tener un sistema que brinde la facilidad a sus estudiantes para agendar citas. Así como también automatizar el proceso de gestión de turnos por parte de los docentes. Mediante la implementación de este sistema se mejorará el proceso administrativo interno de atención al cliente. Los módulos que se contemplaron en el desarrollo de este software son: administración de estudiantes, administración de docentes, gestión de citas, registro de pacientes, gestión de horarios para los docentes.

Para el diseño y construcción del sistema web se utilizaron herramientas open source. Para realizar el sitio web se utilizó el lenguaje de programación php, javascript y jquery. La base de datos se encuentra implementada en MySQL.

Además para este proyecto se utilizó la metodología de desarrollo PUD (Proceso Unificado de Desarrollo). La cual permite desarrollar el proyecto de forma iterativa e incremental. PUD está hecho para adaptarse a la gran variedad de sistema de software. Por lo que se consideró adecuado implantarlo en nuestro proyecto para cumplir con los requisitos de tiempo que contábamos para desarrollar el sistema.

ABSTRACT

The present project arises from the need of the odontological clinic of the University of the Americas to have a system that provides the facility to their students to schedule medical appointments. As well as automate the process of management of shifts by the teachers. The implementation of this system will improve the internal administrative process of customer service. The modules that were contemplated in the development of this software are: administration of students, administration of teachers, management of medical appointments, registration of patients, management of schedules for teachers.

For the design and construction of the web system we used open source tools. To make the website we used the programming language php, java script and jquery. The database is implemented in MySQL.

Also for this project we used the methodology of development PUD (Unified Process of Development). This allows to develop the project in an iterative and incremental way. PUD is made to fit the wide variety of software system. So it was considered appropriate to implement it in our project to meet the time requirements we had to develop the system.

ÍNDICE

1. INTRODUCCIÓN	1
1.1 Antecedentes.....	1
1.2 Alcance	1
1.3 Justificación	2
1.4 Objetivos.....	3
1.4.1 Objetivo General.....	3
1.4.2 Objetivos Específicos	3
2. MARCO TEÓRICO.....	4
2.1 Front End.....	4
2.1.1 PHP	4
2.1.1.1 Ventajas de PHP	4
2.1.2 Codeigniter	4
2.1.3 Sublime text.....	5
2.1.4 JavaScript.....	5
2.1.5 JQuery.....	6
2.1.6 AJAX (JavaScript asíncrono y XML).....	6
2.1.7 JSON (JavaScript Object Notation)	6
2.1.8 Bootstrap	7
2.2 Back end (SQLyog).....	8
2.3 Herramienta case	9
2.3.1 Visio.....	9
2.3.2 MySQL Workbench.....	9
2.4 Servidor web.....	10
2.4.1 WampServer.....	10
2.4.1.1 Funcionalidad	11
2.5 Proceso Unificado de Desarrollo	11
2.5.1 Dirigido por Casos de Uso.....	12
2.5.2 Centrado en la Arquitectura.....	12
2.5.3 Iterativo e Incremental	12

2.5.4	La vida del proceso unificado	13
2.5.5	El producto	13
2.5.6	Fases de un ciclo.....	14
2.6	Persistencia de Objetos	15
3.	ANÁLISIS Y DESARROLLO.....	15
3.1	Análisis de la situación actual (Fase de inicio)	15
3.2	Fase de elaboración	16
3.2.1	Requerimientos Funcionales	16
3.2.2	Requerimientos no funcionales.....	18
3.2.2.1	Requisitos de Interfaz	18
3.2.2.2	Requisitos de Navegación	19
3.2.2.3	Requisitos de Usabilidad.....	19
3.2.2.4	Requisitos de Escalabilidad	19
3.2.2.5	Requisitos Operacionales	19
3.2.2.6	Requisitos de Seguridad	19
3.3	Diagramas UML.....	20
3.3.1	Diagrama de Casos de Uso.....	20
3.3.1.1	Caso de Uso: Agendar Cita.....	22
3.3.1.2	Caso de Uso: Registrar Estudiante	22
3.3.2	Diagrama de Secuencia	23
3.3.3	Diagrama de Clases	25
3.3.4	Diagrama de Base de Datos.....	26
3.3.5	Diagrama de Paquetes	27
3.3.6	Diagrama de Distribución	28
3.4	Arquitectura.....	29
3.5	Diseño de Interfaces (Fase de construcción)	29
3.5.1	Interfaces “Perfil Administrador”	29
3.5.1.1	Interfaz de Gestión de docentes	29
3.5.1.2	Interfaz Gestionar Cita	31
3.5.1.3	Interfaz Gestionar Turno	32
3.5.1.4	Modificar Tratamientos.....	32
3.5.1.5	Registrar Horarios Docente.....	33
3.5.2	Interfaces Perfil Docente	33

3.5.2.1	Registrar Estudiante.....	33
3.5.2.2	Aprobar Requisitos.....	34
3.5.3	Interfaces “Perfil Estudiante”	34
3.5.3.1	Registrar Paciente.....	34
3.5.3.2	Agendar Cita	35
3.5.3.3	Visualizar Horarios Docente.....	35
4.	PRUEBAS (Fase de transición)	36
4.1	Introducción	36
4.2	Pruebas de Caja Blanca.....	36
4.3	Pruebas de Caja Negra	37
4.4	Casos de Prueba.....	38
4.4.1	Ingreso al Sistema	38
4.4.2	Registrar Información Estudiante.....	38
4.4.3	Registrar Información Paciente.....	39
4.4.4	Actualizar Información	40
4.4.5	Agendar Cita.....	40
5.	CONCLUSIONES Y RECOMENDACIONES	42
5.1	Conclusiones.....	42
5.2	Recomendaciones.....	43
	REFERENCIAS	44
	ANEXOS	49

ÍNDICE DE FIGURAS

Figura 1. Ejemplo de archivo de texto en "Sublime Text"	5
Figura 2. Logo JavaScript	5
Figura 3. Ejemplo tema de Bootstrap	7
Figura 4. Página principal de Bootstrap	8
Figura 5. Logo SQLyog	8
Figura 6. Logo Visio	9
Figura 7. Página de descarga WampServer.....	11
Figura 8. El ciclo de vida del Proceso Unificado.....	13
Figura 9. Ciclo con sus fases e iteraciones	13
Figura 10. Ejemplo de Diagrama de Caso de Uso	20
Figura 11. Diagrama de Caso de Uso General del Sistema.....	22
Figura 12. Diagrama de Caso de Uso: Agendar Cita	22
Figura 13. Diagrama de Caso de Uso: Registrar Estudiante.....	23
Figura 14. Ejemplo de Diagrama de Secuencia	23
Figura 15. Diagrama de Clases.....	25
Figura 16. Diagrama de Base de Datos	26
Figura 17. Diagrama de Distribución	28
Figura 18. Arquitectura de la aplicación	29
Figura 19. Interfaz Gestión Docente.....	30
Figura 20. Formulario Ingreso de Docente.....	31
Figura 21. Interfaz gestionar cita	31
Figura 22. Gestión de Turnos.....	32
Figura 23. Formulario para modificar los tratamientos de la cita	32
Figura 24. Interfaz para el ingreso de horarios.....	33
Figura 25. Formulario Ingreso de Estudiante	33
Figura 26. Formulario para gestionar la aprobación de requisitos.....	34
Figura 27. Registrar Paciente.....	34
Figura 28. Formulario para agendar citas	35
Figura 29. Horarios Docente	35
Figura 30. Código php para la función guardar estudiante.....	37
Figura 31. Formulario de Prueba Docente	37

Figura 32. Flujo para Agendar Cita	40
--	----

ÍNDICE DE TABLAS

Tabla 1. Requerimiento Gestionar Alumno	16
Tabla 2. Requerimiento Gestionar Paciente.....	16
Tabla 3. Requerimiento Gestionar Docente	17
Tabla 4. Requerimiento Gestionar Pre-requisitos Turno	17
Tabla 5. Requerimiento Gestión Turno	18
Tabla 6. Requerimiento Gestionar Tratamiento.....	18
Tabla 7. Descripción de los elementos de un caso de uso	20
Tabla 8. Descripción de los elementos de un diagrama de Secuencia	23
Tabla 9. Prueba: Acceso a la aplicación	38
Tabla 10. Prueba: Ingreso de información Estudiante.....	38
Tabla 11. Caso de Prueba: Registrar Información Paciente	39
Tabla 12. Caso de Prueba: Actualizar Información	40
Tabla 13. Caso de Prueba: Agendar Cita.....	41

1. INTRODUCCIÓN

1.1 Antecedentes

En los últimos años la informática ha logrado convertirse en una herramienta indispensable que facilita la automatización de los procesos que se llevan a cabo en las actuales empresas, también lo hizo en el campo de la medicina donde la implementación de sistemas informáticos ha facilitado el proceso de gestión de citas médicas, permitiéndoles conocer de manera rápida y oportuna la información de sus pacientes.

En la Clínica de Odontología de la UDLA, actualmente, el proceso de registro de pacientes se lo hace manualmente, los estudiantes agendan una cita dependiendo de los horarios disponibles de los docentes y tomando en cuenta los requisitos previos que deben cumplir para atender a sus pacientes. Los alumnos de odontología se encuentran asignados a diferentes clínicas según el semestre en el que se encuentran.

El centro de atención se encuentra dividido en cubículos de acuerdo al tipo de tratamiento. Existen 4 tipos de atención y según el caso se designa a los estudiantes de odontología un cubículo para que atiendan a los pacientes.

Todo este proceso se encuentra registrado en papel y en un archivo de Excel en donde se registran los turnos, el horario y el cubículo en el cual atenderán al paciente.

1.2 Alcance

El alcance de este trabajo de titulación es desarrollar un sistema que estará disponible en la web para que se pueda acceder a la misma a través de cualquier equipo que disponga de conexión a internet.

De esta forma la clínica odontológica podrá tener una mejor administración de sus pacientes y de su personal.

Los usuarios podrán agendar citas, confirmarlas, anularlas, consultar diagnósticos y seleccionar al especialista, dependiendo de los horarios establecidos por la clínica.

El personal que brinda atención podrá acceder a un historial de los pacientes que han atendido y de los tratamientos que han realizado y también será el encargado de registrar o modificar la información de los pacientes.

En el sistema se definirán perfiles de acceso a la información con el fin de brindar seguridad a los datos de los usuarios y que solo el personal autorizado sea el que administre la información.

Este aplicativo no se encargará de administrar la facturación, transacciones bancarias o el seguro médico de los pacientes que acudan a ser atendidos en la clínica.

1.3 Justificación

Debido a la gran demanda que existe actualmente, se decidió implementar una solución que permita gestionar adecuadamente los procesos de atención a sus clientes, así como también la gestión del personal que brinda el servicio. Esta aplicación impulsará el crecimiento de la clínica para que de esta manera pueda cumplir con los objetivos propuestos en su planificación.

Otro motivo por el cual se desarrolla este proyecto es debido a que los registros médicos se los lleva aún en papel y generan problemas tales como pérdida de información confidencial de los pacientes, pérdida de tiempo al buscar la documentación, falta de control sobre su personal, lo cual ocasiona insatisfacción al cliente.

El desarrollo de este trabajo de titulación permitirá:

- Facilitar al paciente la obtención de una cita médica de una manera ágil.
- Disponer de información actualizada.

- Proveer a la Clínica de una herramienta tecnológica que le permita mejorar el proceso administrativo interno de atención al cliente.
- Acceso a los estudiantes de la carrera de odontología a la página web para que interactúen con la aplicación y la información de sus pacientes

1.4 Objetivos

1.4.1 Objetivo General

- Implementar un sistema Web orientado a la gestión de pacientes de odontología y del personal médico del área de Quirófanos en la Clínica Odontológica de la Universidad de las Américas.

1.4.2 Objetivos Específicos

- Analizar los requerimientos para elaborar el sistema de manera que permita a los estudiantes del área de quirófano agendar una cita para sus pacientes.
- Desarrollar un sistema Web con el uso de herramientas específicas de desarrollo de software que permita incorporar todos los parámetros antes analizados.
- Verificar el correcto funcionamiento del sistema y realizar las capacitaciones necesarias para su óptimo funcionamiento.

2. MARCO TEÓRICO

2.1 Front End

2.1.1 PHP

PHP es un tipo de código abierto que se ejecuta en el lado del servidor que permite dinámicamente desarrollar contenido de una página o aplicación web. Una de las mayores ventajas de php consiste en que se puede incorporar código directamente en el diseño HTML lo que hace prescindir de un procesamiento externo. El código resultante simplemente debe ser interpretado por un servidor de php para que la página web final sea generada y codificada.

2.1.1.1 Ventajas de PHP

- Lenguaje totalmente libre y abierto: El código se encuentra disponible para todos a través de Internet.
- Los entornos de desarrollo son de rápida y fácil configuración: Se puede programar a través de cualquier editor de texto.
- Fácil despliegue: Existe código disponible que puede ser ejecutado inmediatamente.
- Fácil acceso a bases de datos: Se puede acceder a un motor de BD con solo unas pocas líneas de código, sin necesidad de instalar paquetes adicionales.
- Comunidad muy grande: “PHP ofrece la mayor ventaja de todas: tiene, probablemente, una de las comunidades en Internet más grandes con respecto a otros lenguajes” (LanceTalent, 2016).

2.1.2 Codeigniter

Es un entorno de desarrollo web orientado específicamente para el trabajo con PHP en conjunto con HTML, este marco permite de manera fácil y organizada la implementación de aplicaciones y sitios web. La principal idea de codeigniter es brindar al usuario una base para el desarrollo de su proyecto, de esta manera no se requiere la creación de archivos que son básicos para cualquier tipo de sitio web, simplemente se requiere cambiar ciertos valores o atributos para activar las diferentes funcionalidades de codeigniter.

2.1.3 Sublime text

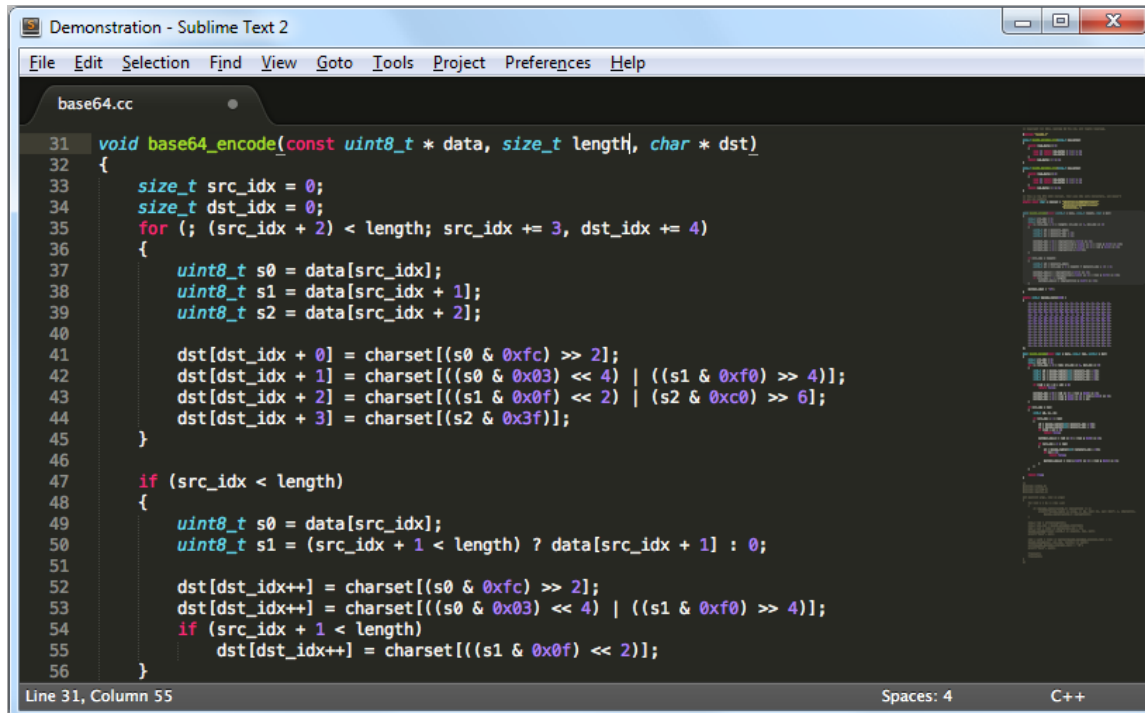


Figura 1. Ejemplo de archivo de texto en "Sublime Text"

Tomado de (Sublime Text, 2016)

“Sublime Text es un editor de texto sofisticado para el código, el marcado y la prosa. Te encantará el elegante interfaz de usuario, características extraordinarias y un rendimiento increíble” (Tomado de sublime text, 2016).

2.1.4 JavaScript



Figura 2. Logo JavaScript

Tomado de (Uneweb, 2017)

Más conocido como código o lenguaje interpretado, está basado en el estándar ECMAScript, se interpreta en el lado del cliente lo cual permite la navegación dinámica y mejoras en la interfaz de usuario en tiempo real sin la necesidad de recargar la página web una y otra vez si no es necesario.

2.1.5 JQuery

Una librería basada en JavaScript, este archivo permite facilitar la interacción con distintos tipos de documentos HTML, es actualmente una de las librerías más utilizadas de JS, permite diferentes acciones como la manipulación del árbol de DOM, agregar animaciones, controlar eventos y permite interactuar por medio de AJAX a aplicativos webs.

2.1.6 AJAX (JavaScript asíncrono y XML)

Es una funcionalidad de desarrollo que permite la creación de aplicaciones interactivas. La ventaja que ofrece AJAX es su ejecución en el navegador del cliente mientras se establece una comunicación en segundo plano en el lado del servidor de manera asíncrona. Esto permite que se realicen cambios en las páginas sin la necesidad de enviar una nueva petición al proveedor de servicios, aumentando así la velocidad de respuesta en el lado del usuario.

2.1.7 JSON (JavaScript Object Notation)

Es un formato de texto usado para el intercambio de datos, debido a que es un muy ligero ha sido adoptado como alternativa a XML y hoy en día es considerado como un lenguaje independiente.

“Se basa en un subconjunto del lenguaje de programación JavaScript (...) es un formato de texto que es completamente independiente del lenguaje, pero utiliza las convenciones que son familiares para los programadores

del C-familia de lenguajes, incluyendo C, C ++, C #, Java, JavaScript, Perl, Python, y muchos otros” (Introducing JSON, 2016).

2.1.8 Bootstrap

Es un entorno de trabajo o grupo de herramientas muy útil para el diseño de sitios y aplicaciones web, puesto que cuenta con gran variedad de formatos predefinidos para diferentes componentes como botones, cuadros de texto, formularios, menús, entre otros, estos elementos cuentan con un muy buen aspecto, lo cual mejora de gran manera la apariencia del sitio web, estas plantillas están basadas en HTML y CSS además de archivos JS que son opcionales. Otra de las grandes ventajas que ofrece bootstrap es la adaptabilidad de la página a diferentes dispositivos mediante una configuración muy simple basada en columnas toda esta información se puede obtener directamente de su página oficial.

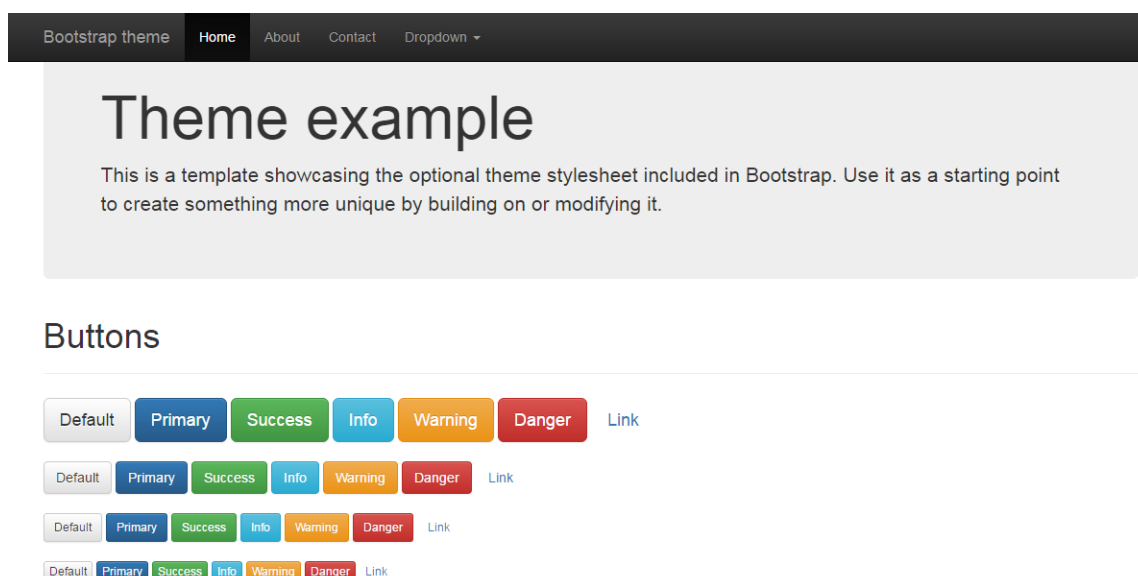


Figura 3. Ejemplo tema de Bootstrap

Tomado de (Bootstrap, s.f)



Figura 4. Página principal de Bootstrap

Tomado de (Bootstrap, s.f)

Bootstrap ofrece una interfaz amigable, moderna y dinámica que permite al usuario sentirse cómodo al usarla y de igual manera le permite acceder a una aplicación desde cualquier dispositivo que posea, eso le facilita la lectura y el manejo de los distintos controles de la página e incluso puede volver más agradable el trabajo o una transacción.

2.2 Back end (SQLyog)

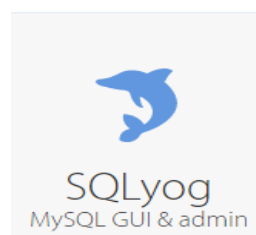


Figura 5. Logo SQLyog

Tomado de (Webyog, s.f)

SQLyog es una de las herramientas de administración de bases de datos más robustas y de fácil manejo, este elemento es preferido por muchos desarrolladores y diseñadores debido a sus múltiples utilidades como la

importación y exportación de datos en diferentes tipos de archivos y además la simplicidad de creación de tablas con diferentes atributos.

“SQLyog es la mejor herramienta de gestión de base de datos que he utilizado siempre. Es fácil de usar, las copias de seguridad son fáciles y rápidas. El generador de consultas hace la vida mucho más fácil” (Hubbard, 2016).

2.3 Herramienta case

2.3.1 Visio



Figura 6. Logo Visio

Tomado de (Microsoft Visio, 2017)

Es un software utilizado para la creación de distintos diseños gráficos como diagramas UML, diagramas de flujo, mapas conceptuales, cuadros sinópticos, planos a baja escala, entre otros. Es usado principalmente para la representación de diferentes modelos que son usados principalmente por programadores, desarrolladores, ingenieros, según la necesidad que estos tengan para diseñar un proyecto.

2.3.2 MySQL Workbench

MySQL Workbench es una herramienta que permite realizar modelado de datos, obtener el script de SQL que facilita la configuración del servidor, administración de usuarios, copia de seguridad, migración de datos. MySQL Workbench proporciona los DBA y desarrolladores un entorno integrado de herramientas:

- Diseño y Modelado de base de datos
- Desarrollo de SQL
- Administración de base de datos
- La migración de bases de datos

Herramientas que aceleran el manejo de Bases de datos

- Esquema y datos de sincronización: Aplica cambios visuales de esquema y cambios de datos en tiempo real. Replica datos de manera periódica a intervalos programados utilizando un potente algoritmo de revisión.
- Tunneling SSH y HTTP: Gracias a su amplia compatibilidad con múltiples entornos se puede suplir de manera óptima bases de datos como MySQL y phpMyAdmin.
- Copias de seguridad programadas: Cuenta con un asistente para crear copias de seguridad y programación de las mismas de una forma fácil y rápida mientras se trabaja en la base.
- Importación de datos externos: La migración de diferentes fuentes de datos es menos compleja con respecto a otros lenguajes, se puede importar datos de distinto ODBC compatibles como access, excel, etc.

2.4 Servidor web

2.4.1 WampServer

WampServer es un entorno completo para el desarrollo de aplicaciones web, puesto que incluye un servidor web, lenguaje php y además permite la administración de base de datos en MySql por medio de PhpMyAdmin

Se puede descargar el programa directamente de la página del servidor <http://www.wampserver.com/en/>



Figura 7. Página de descarga WampServer

Tomado de (Wampserver, s.f)

2.4.1.1 Funcionalidad

- Permite la gestión de servicios MySql y Apache
- La opción de servidor local o global
- Comunicación entre Apache, PHP y MySQL
- Administración de la configuración de servidores
- Acceso a registros
- Acceso a archivos para la configuración
- Creación de alias
- Múltiples idiomas

2.5 Proceso Unificado de Desarrollo

El proceso unificado constituye un marco de trabajo genérico que se especializa para adaptarse a la gran variedad de sistema de software, diferentes áreas de aplicación, tipos de organizaciones y diferentes tamaños de proyecto.

Esta metodología utiliza el lenguaje unificado de modelado para elaborar los esquemas del sistema.

Las principales características que definen al proceso unificado son:

- Dirigido por casos de uso
- Arquitectura
- Iterativo e incremental

2.5.1 Dirigido por Casos de Uso

Un Caso de Uso se refiere a una parte de la funcionalidad del sistema que provee al usuario un resultado después de este interactuar con el mismo. Los casos de uso reflejan los requisitos funcionales del sistema. Los cuales guían el diseño, implementación y pruebas: es decir, orientan el proceso de desarrollo.

Dirigido por casos de uso significa que la construcción del sistema sigue una serie de flujos a partir de los casos de uso. Los mismos que se especifican, diseñan y son la fuente mediante la cual los desarrolladores construyen sus casos de prueba.

2.5.2 Centrado en la Arquitectura

El concepto de arquitectura de software engloba los aspectos estáticos y dinámicos del sistema, así como también la plataforma en la que va a funcionar el software, arquitectura de hardware, sistema operativo, sistema de gestión de base de datos, protocolos para comunicarse en red. Constituye una vista del diseño conjuntamente con las principales características dejando los detalles de lado. Este proceso facilita a los desarrolladores a focalizarse en los objetivos adecuados, como la capacidad de adaptación al cambio y reutilización.

2.5.3 Iterativo e Incremental

Los desarrolladores seleccionan lo que diseñarán en una iteración basándose en dos factores. Primero, la iteración que agrupa un conjunto de casos de uso que amplían la utilidad del producto desarrollado hasta el momento. Segundo, la iteración que se encarga de los riesgos más significativos.

En cada iteración los desarrolladores determinan y especifican los casos de uso más importantes, crean un diseño utilizando la arquitectura establecida como guía, implementan el diseño a través de componentes y prueban que los mismos cumplen con los casos de uso.

2.5.4 La vida del proceso unificado

El proceso unificado se realiza mediante una serie de ciclos que representan la vida del sistema. Cada ciclo finaliza con una versión del software. Se compone de 4 fases: inicio, elaboración, construcción y transición. Y cada fase a su vez se divide en iteraciones como se explicó previamente.

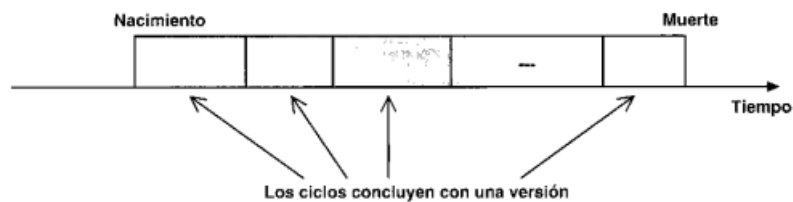


Figura 8. El ciclo de vida del Proceso Unificado

Tomado de (Booch, Jacobson y Rumbaugh (2000, pp. 8))

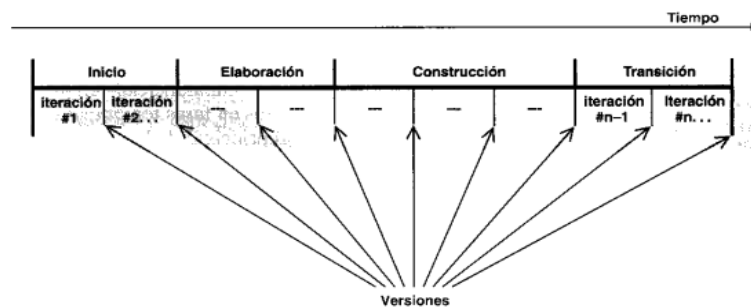


Figura 9. Ciclo con sus fases e iteraciones

Tomado de (Booch, Jacobson y Rumbaugh (2000, pp. 8))

2.5.5 El producto

En cada ciclo se obtiene una nueva versión del sistema, la misma que consta de un cuerpo de código fuente preparado para su entrega incluida en componentes que puede compilarse y ejecutarse.

El producto terminado incluye los requisitos, casos de uso, especificaciones no funcionales y casos de prueba. Así como también abarca el modelo de

arquitectura y el modelo visual que se elaboran mediante el lenguaje unificado de modelado.

- Modelo de casos de uso y su relación con los usuarios
- Modelo de despliegue definiendo los nodos físicos (ordenadores) y la relación de los componentes con esos nodos
- Modelo de prueba, que detalla los casos de prueba que se utilizaran para comprobar los casos de uso
- Representación de la arquitectura

2.5.6 Fases de un ciclo

- Fase de Inicio: se elabora una descripción del producto final a partir de una idea, se presenta un análisis del producto. Se detallan cuáles son las principales funciones del sistema y su arquitectura.
- Fase de elaboración: se especifican a detalle los casos de uso del sistema y se diseña la arquitectura. Se planifican las actividades y estiman recursos para terminar el proyecto.

En esta fase se utilizan los diagramas UML para realizar la especificación de requerimientos.

- Diagrama de Casos de Uso
 - Diagrama de Secuencia
 - Diagrama de Clases
 - Diagrama de Base de Datos
 - Diagrama de Paquetes
 - Diagrama de Distribución
-
- Fase de construcción: Se crea el producto. En esta fase el sistema evoluciona hasta estar preparado para ser entregado a los usuarios.

- Fase de transición: Se realizan las pruebas conjuntamente con los usuarios del sistema. Se corrigen problemas y realizan mejoras.

2.6 Persistencia de Objetos

Es un proceso que consiste en la codificación de un objeto o programa dentro de un medio de almacenamiento, el cual puede ser conocido como; archivo o buffer de memoria con la finalidad de transmitirlo a través de una conexión en red como una serie de bytes.

La serialización es un mecanismo usado para transportar objetos a través de una red y lograr hacer persistente un objeto en un archivo o base de datos.

3. ANÁLISIS Y DESARROLLO

3.1 Análisis de la situación actual (Fase de inicio)

Actualmente los estudiantes para agendar una cita en la clínica, así como también para averiguar los horarios de disponibilidad de los docentes que supervisan la atención a los pacientes deben acudir presencialmente. La asignación de turnos; así como la presentación de requisitos previos a la atención del cliente, que deben ser aprobados por el docente, se encuentra registrado en la herramienta de Excel, lo cual puede ocasionar pérdida de información importante.

Es por eso que se decidió implementar una solución que permita gestionar adecuadamente los procesos de atención a sus clientes, así como también la gestión del personal que brinda el servicio.

Mediante el desarrollo de esta aplicación los usuarios tendrán las siguientes facilidades:

- Tener un control sobre los horarios de los docentes para que de esta manera los estudiantes a través de la página web del sistema consulten la disponibilidad de los docentes para solicitar su cita. Con esto se logra tener un mejor tiempo de respuesta por parte de la clínica ya que existen

alumnos que vienen desde otras provincias y ya no deberán asistir personalmente.

- Disponer de una aplicación que agilite el proceso asignación de turnos. El docente encargado solo deberá ingresar al sistema y fijar el horario de atención al estudiante y también al docente encargado de supervisar la atención al cliente, con lo cual ya no tendrán que registrar información sobre sus pacientes en la herramienta Excel.
- Tanto los estudiantes como docentes de Odontología podrán tener a disposición un historial clínico sobre los tratamientos que se han realizado a los pacientes, de esta manera, se podrá mejorar el proceso administrativo interno de atención al cliente.
- Los docentes dispondrán de una herramienta que les permitirá tener una mejor gestión sobre los requisitos que necesitan aprobar a los estudiantes de Odontología, ya que únicamente deberán acceder al sistema y cambiar su estado a pendiente o aprobado, según sea el caso.

3.2 Fase de elaboración

3.2.1 Requerimientos Funcionales

Tabla 1.
Requerimiento Gestionar Alumno

RF-0001	Gestionar Alumnos
Descripción	Crear, Buscar, Actualizar y Eliminar los alumnos. Los alumnos contarán con los siguientes atributos: código, cédula, nombre, apellido, clave, correo
RF-0001	Gestionar Alumnos
Importancia	Alta
Estado	Disponible
Estabilidad	Alta
Comentarios	

Tabla 2.
Requerimiento Gestionar Paciente

RF-002	Gestionar Pacientes
Descripción	<p>Crear, Buscar, Actualizar y Eliminar los pacientes</p> <p>Los pacientes contarán con los siguientes atributos: código, apellido, nombre, numero Historia Clínica</p>
Importancia	Alta
Estado	Disponible
Estabilidad	Alta

Tabla 3.

Requerimiento Gestionar Docente

RF-003	Gestionar Docentes
Descripción	<p>Crear, Buscar, Actualizar y Eliminar los docentes</p> <p>Los docentes contarán con los siguientes atributos: código, cédula, nombre, apellido, clave, correo</p>
Importancia	Alta
Estado	Disponible
Estabilidad	Alta
Comentarios	

Tabla 4.

Requerimiento Gestionar Pre-requisitos Turno

RF-004	Gestionar Prerrequisitos Turno
Descripción	<p>Crear, Buscar, Actualizar y Eliminar los prerrequisitos</p> <p>El prerrequisito contará con los siguientes atributos: código, nombre, descripción, estado</p>

Importancia	Alta
Estado	Disponible
Estabilidad	Alta

Tabla 5.

Requerimiento Gestión Turno

RF-005	Gestionar Turnos
Descripción	<p>Crear, Buscar, Actualizar y Eliminar los turnos</p> <p>El turno contará con los siguientes atributos: código, fecha, hora, numero de sala, asistencia</p>
Importancia	Alta
Estado	Disponible
Estabilidad	Alta

Tabla 6.

Requerimiento Gestionar Tratamiento

RF-006	Gestionar Tratamiento de la cita
Descripción	<p>Crear, Buscar, Actualizar y Eliminar los tratamientos</p> <p>El tratamiento contará con los siguientes atributos: código, nombre, descripción</p>
Importancia	Alta
Estado	Disponible

3.2.2 Requerimientos no funcionales**3.2.2.1 Requisitos de Interfaz**

El sistema será amigable con el usuario y se adaptará a los distintos dispositivos con los que interactúe como tablets, smartphones y otros dispositivos.

El dispositivo manejará el idioma español con sus distintos caracteres como tildes y otros caracteres especiales.

3.2.2.2 Requisitos de Navegación

- La interfaz se compondrá por un menú principal que permitirá el acceso al usuario a diferentes páginas para cada opción.
- El sistema contará con un acceso para los distintos tipos de usuarios y además permitirá un cifrado de la contraseña para una mayor seguridad.
- El sistema contará con distintas funcionalidades para facilitar su manipulación como botones para el ingreso, modificación, y eliminación de datos según el caso.

3.2.2.3 Requisitos de Usabilidad

- El sistema tendrá una interfaz fácil de manipular y entendible para todos los usuarios.
- El sistema contará con distintos mensajes de advertencia y de error para las distintas situaciones en las que se requiera.
- El sistema no permitirá el acceso de datos ingresados en un formato erróneo.

3.2.2.4 Requisitos de Escalabilidad

- El sistema permitirá el acceso a más usuarios según su perfil y además el ingreso a una base de datos según se requiera.

3.2.2.5 Requisitos Operacionales

- El sistema tendrá campos obligatorios para el ingreso.
- El sistema no permitirá el ingreso de datos incorrectos en ciertos campos.
- El sistema exigirá el ingreso de un usuario y contraseña para el ingreso.
- El sistema verificará la existencia de los usuarios y permitirá registrarse.

3.2.2.6 Requisitos de Seguridad

- El sistema contará con confidencialidad de los datos del usuario.
- El sistema mantendrá de manera segura toda la información ingresada.
- El sistema contará con un perfil de usuario para el acceso.
- El sistema no permitirá el acceso a información personal de otros usuarios al público en general.

3.3 Diagramas UML

UML son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha creado para crear esquemas, diagramas y documentación relativa a los desarrollos de software.

Estos diagramas son esenciales en el desarrollo de software ya que sirven para:

- Describir la estructura de un sistema complejo
- Especificar el comportamiento deseado del sistema
- Simplificación y reutilización

3.3.1 Diagrama de Casos de Uso

Un diagrama de casos de uso resume quién usa la aplicación o el sistema, y qué se puede hacer con ellos. Facilita la descripción de los requerimientos del usuario. Muestra la relación entre los actores, requisitos y los componentes principales de la aplicación.

A continuación se describen los elementos de un caso de uso, utilizando como referencia el caso de uso agendar cita de nuestro sistema.

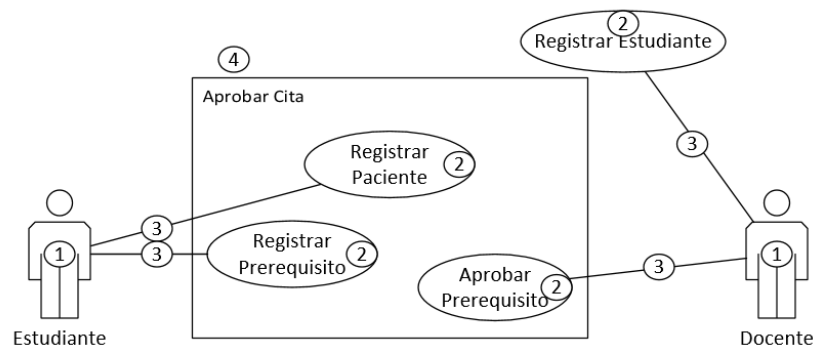


Figura 10. Ejemplo de Diagrama de Caso de Uso

Tabla 7.

Descripción de los elementos de un caso de uso

Forma	Elemento	Descripción y propiedades principales
-------	----------	---------------------------------------

1	Actor	Identifica a los usuarios, entidades que realizan una acción en el sistema.
2	Caso de uso	Son las actividades llevadas a cabo por los actores
3	Asociación	Indica que un actor forma parte de un caso de uso.
4	Subsistema o componente	El sistema o la aplicación en los que se está trabajando (o parte de él).

Para poder representar de forma adecuada los requisitos del usuario, todo proyecto de software necesitar detallar las actividades de cada actor en el sistema, la finalidad de los casos de uso es describir el comportamiento del sistema.

A continuación se presenta la división lógica del proyecto:

- Gestión de usuarios , asignación de permisos
- Gestión de docentes
- Gestión de estudiantes
- Gestionar citas
- Gestión de pacientes
- Gestión para los horarios de docentes

En la siguiente figura se observa el diagrama de caso de uso general del sistema.

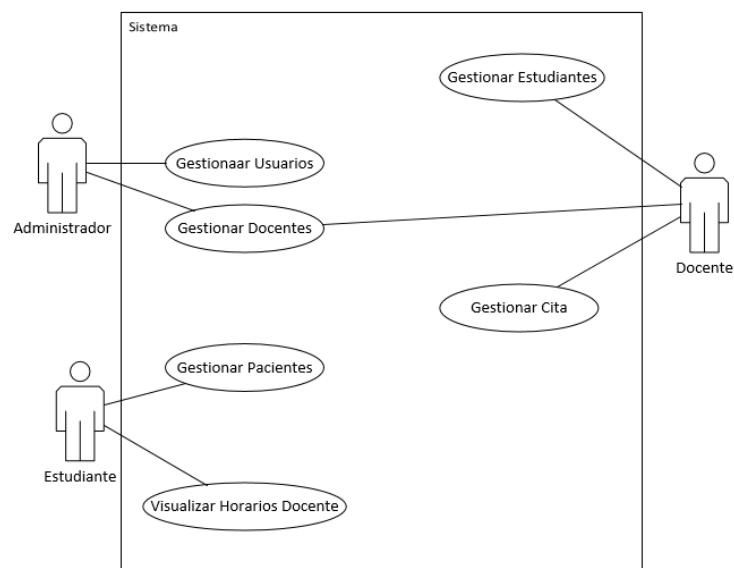


Figura 11. Diagrama de Caso de Uso General del Sistema

3.3.1.1 Caso de Uso: Agendar Cita

En la siguiente figura se observa el proceso mediante el cual el estudiante agenda una cita para atender a sus pacientes. Los alumnos deben primero registrar los datos de sus pacientes, luego registrar cada uno de los requisitos solicitados para que el docente encargado los apruebe y de esta manera asignar el cubículo y el horario para que el alumno realice la atención al cliente.

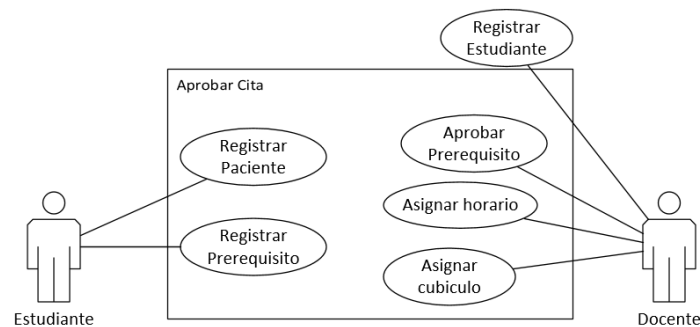


Figura 12. Diagrama de Caso de Uso: Agendar Cita

3.3.1.2 Caso de Uso: Registrar Estudiante

En la siguiente ilustración podemos observar el proceso mediante el cual los docentes realizan el registro de información sobre sus estudiantes. Primero el administrador del sistema debe crear el usuario y asignar el perfil mediante el

cual los estudiantes podrán ingresar al sistema. Después el docente realiza el ingreso al sistema mediante su usuario y clave previamente creados y realizar la inscripción de la información de los alumnos asignándoles el usuario generado previamente.

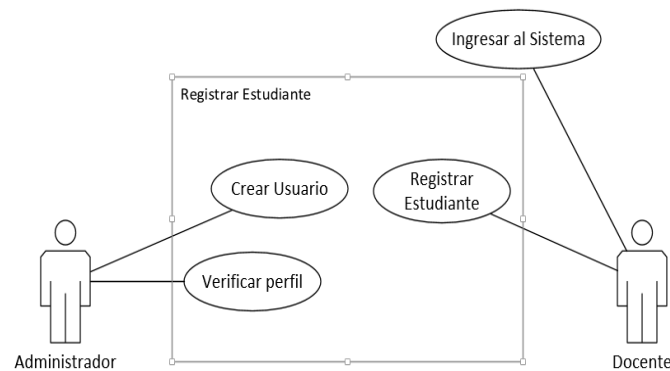


Figura 13. Diagrama de Caso de Uso: Registrar Estudiante

3.3.2 Diagrama de Secuencia

Un diagrama de secuencia muestra la interrelación, es decir la secuencia de mensajes entre instancias de clases, componentes, subsistemas o actores.

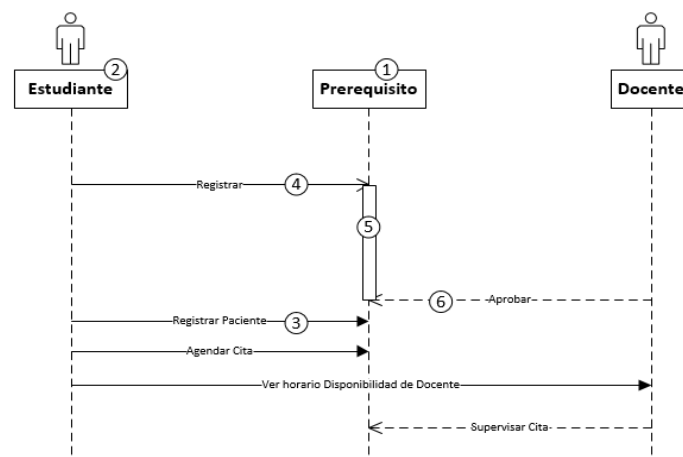


Figura 14. Ejemplo de Diagrama de Secuencia

Tabla 8.

Descripción de los elementos de un diagrama de Secuencia

Forma	Elemento	Descripción
-------	----------	-------------

1	Lifeline	Secuencia de eventos de un participante durante una interacción
2	Actor	Identifica a una entidad, usuario del sistema
3	Mensaje sincrónico	Se utilizan para hacer referencia a una función dentro del programa.
4	Mensaje asincrónico	Mensaje que no requiere una respuesta para que el remitente continúe
5	Ocurrencia de ejecución	Periodo de tiempo en el que el actor realiza la operación
6	devolución de llamada	Mensaje que se devuelve hacia el actor

3.3.3 Diagrama de Clases

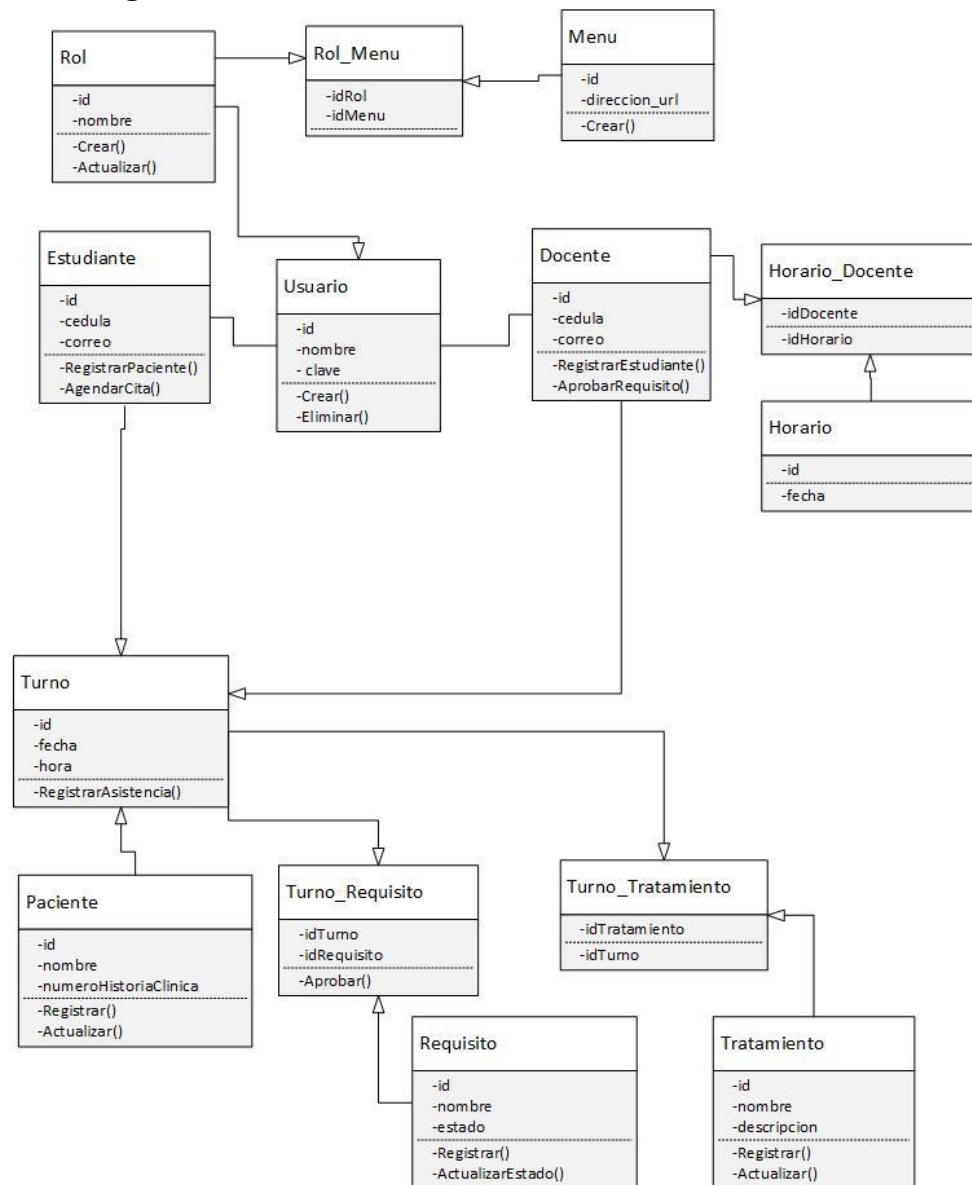


Figura 15. Diagrama de Clases

3.3.4 Diagrama de Base de Datos

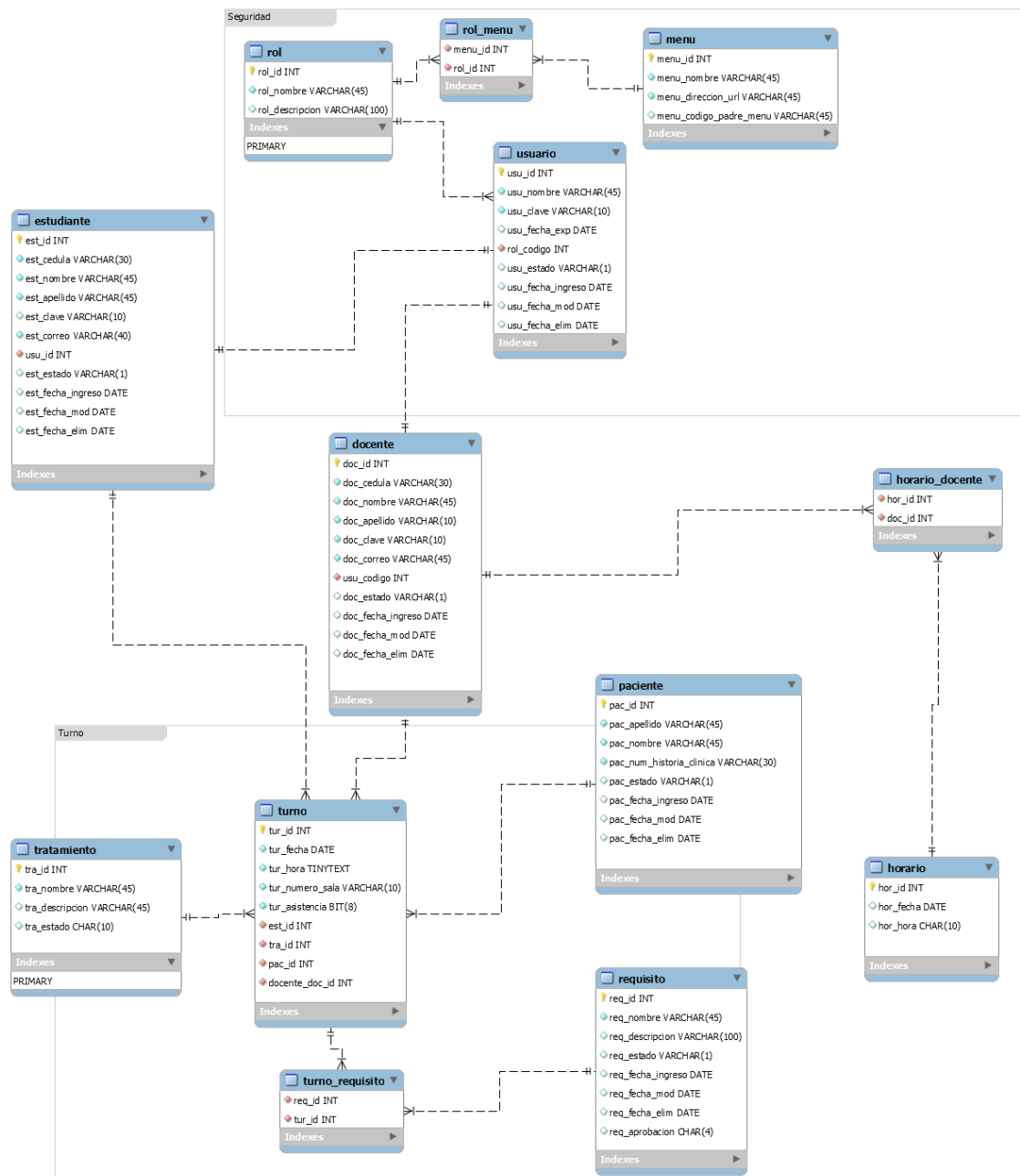


Figura 16. Diagrama de Base de Datos

3.3.5 Diagrama de Paquetes

Constituyen un mecanismo para organizar las clases del aplicativo. Se utilizan para definir la arquitectura del sistema y de esta manera entender mejor la estructura lógica del software.

Para la implementación del sistema se utilizó el patrón de arquitectura Modelo-Vista-Controlador (MVC). El mismo que se encarga de separar la lógica de negocio de la interfaz del usuario y es el más utilizado en aplicaciones Web, ya que facilita la funcionalidad, mantenibilidad y escalabilidad del sistema.

Mediante esta arquitectura las aplicaciones se dividen en:

- **Modelo:** sirve para representar la información con la cual el sistema opera, gestiona la forma en que se accede a los datos, esto es consulta, actualizaciones, ayudando de administrar los privilegios de acceso descritos en la aplicación
- **Vista:** Es la parte del sistema que se muestra gráficamente al usuario
- **Controlador:** recibe las peticiones de los usuarios. Realiza las llamadas al modelo para obtener la información solicitada y enviársela a la capa de vista donde se presenta los datos al usuario.

3.3.6 Diagrama de Distribución

El diagrama de distribución UML muestra la arquitectura física de un sistema informático. Puede representar a los equipos y a los dispositivos, y también mostrar sus interconexiones y el software que se encontrará en cada máquina.

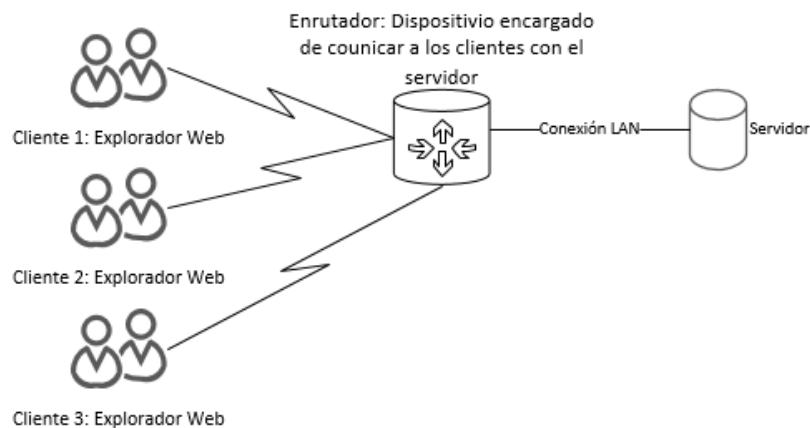


Figura 17. Diagrama de Distribución

Nodo Servidor.- Formado por el aplicativo web y la base de datos que se encuentra implementada en My Sql y php

Nodo Dispositivo router.- Conjunto de dispositivos encargados establecer el canal de comunicación entre los clientes y el servidor

Nodo Cliente.- Corresponde a los usuarios del sistema que mediante el explorador web y la red LAN realizarán la conexión al servidor

3.4 Arquitectura

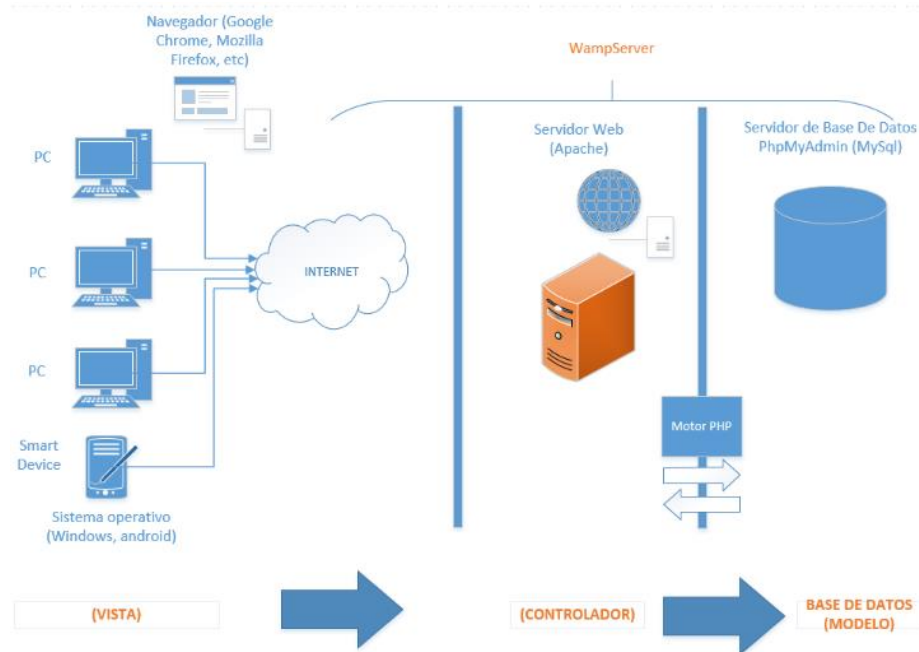


Figura 18. Arquitectura de la aplicación

3.5 Diseño de Interfaces (Fase de construcción)

Una interfaz es el medio mediante el cual el usuario interactúa con el sistema. También conocido como el conjunto de elementos gráficos que permite la presentación y la navegación del sitio. Haciendo que los usuarios tengan completo control sobre las funcionalidades del sistema desde el momento mismo de entrar a él sin que para ello deba tener amplios conocimientos ni preparación anterior alguna.

3.5.1 Interfaces “Perfil Administrador”

3.5.1.1 Interfaz de Gestión de docentes

Para la gestión de docentes se usa una tabla con la opción de editar, y eliminar, así como de un botón para el ingreso de un nuevo docente por medio de una ventana modal.

Administrador: admin Log out

Inicio Citas Usuarios Horarios Otros

Nuevo docente

Mostrar 10 resultados

APELLIDOS	NOMBRES	C.I.	Correo electrónico	Fecha de nacimiento	EDITAR	ELIMINAR
Padilla	Andres	1718093298	apadilla@udanet.ec	2016-11-25		
Izurieta	Carlos	1710510783	czurieta@udanet.ec	1990-10-01		
Rosero	Edgar	1722335203	erosero@udia.edu.ec	1990-10-01		
Zeas	Elizabeth	1708356561	ezeas@udia.edu.ec	1990-10-01		

Mostrando 1 a 4 de 4 resultados

Anterior 1 Siguiente

Tratamientos

- Exodoncia de terceros molares
- Crugia regenerativa
- Restauraciones
- Endodoncias
- Crugia periodontal
- Crugia estética periodontal

Horarios

De lunes a viernes de 10:05 a 18:20

Contacto

Myriam Gordon
Correo: mgordon@udia.edu.ec
Teléfono: 397000 ext. 677

Figura 19. Interfaz Gestión Docente

La siguiente figura muestra el formulario junto con los datos necesarios para registrar un docente. En este formulario los campos con asterisco (*) son obligatorios de llenar. Los campos de fecha de ingreso y de salida manejan un calendario cuyo formato es yyyy-mm-dd, donde “yyyy” hace referencia al año, “mm” al mes y “dd” al día. El campo de código corresponde a la cédula del docente que se encuentra validado para que se verifique si se ingresan solo números con una longitud de 10 dígitos, además se ha realizado una función para comprobar el ultimo dígito y así revisar si los dígitos introducidos son de una cédula válida



DATOS DEL USUARIO

Perfil de usuario: Docente

Fecha creación: 23/01/2017

Fecha exp.: 2100-01-01
Ej: 23/01/2017 - 01/01/2100

*Código: Número de cédula
(10 dígitos)

*Usuario: Ingrese el nombre de usuario
(Max 20 caracteres)

*Contraseña: Ingrese la contraseña
(Max 20 caracteres)

Ingresar

Figura 20. Formulario Ingreso de Docente

3.5.1.2 Interfaz Gestionar Cita

A continuación se puede observar el formulario requerido para que el docente realice la designación de las citas a los diferentes estudiantes. Se debe asignar al docente y al estudiante junto con el paciente que será atendido, seleccionar el tipo de tratamiento así como también aprobar los requisitos que el estudiante debe presentar



DATOS DE LA CITA MÉDICA

Turno N°: 0036

Fecha agenda: 2016-12-01

* Docente: --- Seleccione una opción ---

* Estudiante: --- Seleccione una opción ---

* Paciente: --- Seleccione una opción ---

* Tratamientos:

<input type="checkbox"/>	Alargamiento
<input type="checkbox"/>	Raspado
<input type="checkbox"/>	Insición

Requisitos entregados:

<input type="checkbox"/>	historia clínica llena
<input type="checkbox"/>	hoja de remisión
<input type="checkbox"/>	hoja de interconsulta
<input type="checkbox"/>	radiografía panorámica
<input type="checkbox"/>	factura de cancelación previa
<input type="checkbox"/>	exámenes de laboratorio
<input type="checkbox"/>	hoja de vida

Confirmar cita

Figura 21. Interfaz gestionar cita

3.5.1.3 Interfaz Gestionar Turno

A través de la siguiente interfaz el administrador podrá llevar un control sobre cuales turnos se encuentran aprobados todos sus requisitos así como también podrá notificar a los alumnos que requisitos faltan por entregar para poder confirmar su cita.

Nº Turno	Fecha agenda	Nombre docente	Nombre estudiante	Nombre paciente	Fecha cita	Hora	REQUISITOS	ESTADO	Aprobar	Finalizar	Anular
0037	2016-12-02	Zeas Elizabeth	Vargas Edwin	López Alvaro	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0036	2016-12-02	Zeas Elizabeth	Sampertegui Gabriela	Sánchez Oscar	2016-12-05	10:30:00/11:30:00	APROBADOS	ACTIVO			
0035	2016-12-01	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0034	2016-12-01	Zeas Elizabeth	Sampertegui Gabriela	Sánchez Oscar	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0033	2016-12-01	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
			Sampertegui								

Figura 22. Gestión de Turnos

3.5.1.4 Modificar Tratamientos

GESTIÓN DE TRATAMIENTOS DEL TURNO

Turno N°:

0024

Buscar

Fecha agenda

2016-11-27

Ej: 23/01/2017 - 01/01/2100

*Docente:

Zeas Elizabeth

*Estudiante:

Sampertegui Gabriela

*Paciente:

Alvarez Luis

Tratamientos actuales:

Alargamiento

Raspado

Insición

(Aquí seleccione todos los tratamientos del paciente)

*Nuevos tratamientos:

☐

Alargamiento

☐

Raspado

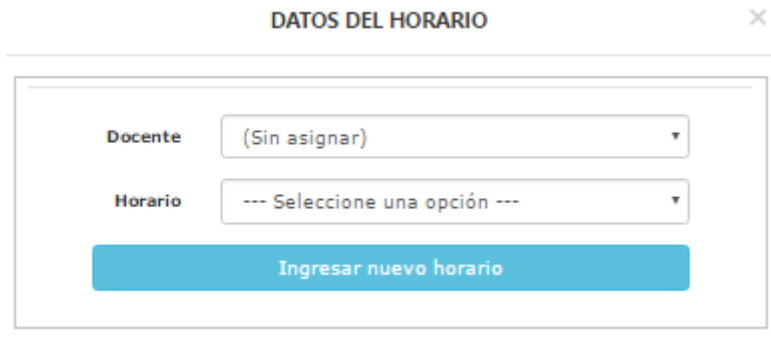
☐

Insición

Guardar cambios

Figura 23. Formulario para modificar los tratamientos de la cita

3.5.1.5 Registrar Horarios Docente



Formulario de registro de horarios docentes. El título es "DATOS DEL HORARIO" con un botón de cerrar (X) a la derecha. El formulario contiene:

- Un campo "Docente" con un menú desplegable que muestra "(Sin asignar)".
- Un campo "Horario" con un menú desplegable que muestra "--- Seleccione una opción ---".
- Un botón azul que dice "Ingresar nuevo horario".

Figura 24. Interfaz para el ingreso de horarios

3.5.2 Interfaces Perfil Docente

3.5.2.1 Registrar Estudiante

Para registrar la información tomar en cuenta que el campo código corresponde a la matrícula del estudiante por lo que solo se permite 6 dígitos, los campos con (*) son obligatorios



Formulario de registro de usuario. El título es "DATOS DEL USUARIO" con un botón de cerrar (X) a la derecha. El formulario contiene:

- Un campo "Perfil de usuario" con un menú desplegable que muestra "Estudiante".
- Un campo "Fecha creación" con un menú desplegable que muestra "23/01/2017".
- Un campo "Fecha exp." con un menú desplegable que muestra "2100-01-01". Debajo de este campo hay un ejemplo: "Ej: 23/01/2017 - 01/01/2100".
- Un campo "Código" (obligatorio) con un menú desplegable que muestra "Número de matrícula (6 dígitos)".
- Un campo "Usuario" (obligatorio) con un menú desplegable que muestra "Ingrese el nombre de usuario (Max 20 caracteres)".
- Un campo "Contraseña" (obligatorio) con un menú desplegable que muestra "Ingrese la contraseña (Max 20 caracteres)".
- Un botón azul que dice "Ingresar".

Figura 25. Formulario Ingreso de Estudiante

3.5.2.2 Aprobar Requisitos

Turno N°: 0025

Buscar

Fecha agenda: 2016-11-28

Paciente: Alvarez Luis

Estudiante: Sampertegui Gabriela

Docente: Zeas Elizabeth

Tratamientos: Alargamiento
Raspado

REQUISITO	ESTADO	* APROBAR	* DESAPROBAR
historia clínica llena	APROBADO		
hoja de remisión	APROBADO		
hoja de interconsulta	APROBADO		
radiografía panorámica	APROBADO		
factura de cancelación previa	ENTREGADO		
exámenes de laboratorio	ENTREGADO		

Figura 26. Formulario para gestionar la aprobación de requisitos

3.5.3 Interfaces “Perfil Estudiante”

3.5.3.1 Registrar Paciente

A continuación, se muestra el formulario para el registro de información de los pacientes. Los campos con el asterisco (*) son obligatorios. La fecha se maneja con el formato yyyy-mm-dd, donde “yyyy” es el año, “mm” el mes y “dd” el día. El número de historia clínica permite únicamente el ingreso de 10 dígitos

DATOS DEL PACIENTE

Fecha de ingreso: 2017-01-23

*N° historia clínica: Número de historia clínica
(Max 10 caracteres)

*Nombre/s: Ingrese el nombre
(Max 20 caracteres)

*Apellido/s: Ingrese el apellido
(Max 20 caracteres)

Ingresar

Figura 27. Registrar Paciente

3.5.3.2 Agendar Cita

Turno N°: 0039

Fecha agenda: 2017-01-19

* Docente: --- Seleccione una opción ---

* Estudiante: --- Seleccione una opción ---

* Paciente: --- Seleccione una opción ---

* Tratamientos:

<input type="checkbox"/>	Alargamiento
<input type="checkbox"/>	Raspado
<input type="checkbox"/>	insición

Requisitos entregados:

<input type="checkbox"/>	historia clinica llena
<input type="checkbox"/>	hoja de remisión
<input type="checkbox"/>	hoja de interconsulta
<input type="checkbox"/>	radiografia panorámica
<input type="checkbox"/>	factura de cancelación previa
<input type="checkbox"/>	exámenes de laboratorio
<input type="checkbox"/>	hoja de vida

Confirmar cita

Figura 28. Formulario para agendar citas

3.5.3.3 Visualizar Horarios Docente

HORARIO DOCENTES					
Hora/ Día	Lunes	Martes	Miércoles	Jueves	Viernes
7:00/8:00	(No asignado)	Rosero Edgar	Zeas Elizabeth	Rosero Edgar	Rosero Edgar
8:05/9:05	Rosero Edgar	Izurieta Carlos	Izurieta Carlos	Izurieta Carlos	Padilla Andres
9:10/10:10	Zeas Elizabeth	(No asignado)	(No asignado)	Padilla Andres	(No asignado)
10:15/11:15	Rosero Edgar	(No asignado)	(No asignado)	(No asignado)	(No asignado)
11:20/12:20	Zeas Elizabeth	(No asignado)	(No asignado)	(No asignado)	(No asignado)
12:25/13:25	(No asignado)	(No asignado)	(No asignado)	(No asignado)	(No asignado)
13:30/14:35	(No asignado)	(No asignado)	(No asignado)	(No asignado)	(No asignado)
14:35/15:35	(No asignado)	(No asignado)	(No asignado)	(No asignado)	(No asignado)
15:40/16:40	Padilla Andres	(No asignado)	(No asignado)	(No asignado)	(No asignado)
16:45/17:45	(No asignado)	(No asignado)	(No asignado)	(No asignado)	(No asignado)

Figura 29. Horarios Docente

4. PRUEBAS (Fase de transición)

4.1 Introducción

En esta etapa se realiza un análisis para determinar el grado de satisfacción del usuario con el producto final implementado. La finalidad de este proceso es revisar los errores y defectos de la aplicación, para realizar las correcciones respectivas y de esta manera dar un total cumplimiento a los requerimientos funcionales.

Debido a que en la fase de diseño se aplicó la metodología PUD, este proceso ya fue llevado a cabo en cada una de las iteraciones. Ya que esta metodología se basa en un proceso iterativo e incremental donde los desarrolladores implementan el diseño a través de componentes y prueban que los mismos cumplen con los casos de uso y por lo tanto los requerimientos funcionales y no funcionales.

A continuación se describen las técnicas que se utilizaron para realizar las pruebas con la participación de los usuarios.

4.2 Pruebas de Caja Blanca

Se refieren a la implementación de pruebas estructurales. Su finalidad es analizar detalladamente los procedimientos del sistema. Se centran en buscar los errores en el código fuente.

Estas pruebas fueron implementadas por los desarrolladores del sistema conjuntamente como la supervisión del docente guía. Mediante este proceso se pudo realizar las validaciones respectivas a los datos de entrada del módulo estudiante. Donde se verificó que la información de entrada cumpla con las reglas que se encuentran en la base de datos y de esta forma no exista inconvenientes para guardar información. Para realizar esta prueba se utilizó la técnica del camino básico donde se comprobó que cada sentencia de código se ejecutó al menos una vez.

```

public function ingresar()
{
    $data_estudiante=array(
        'usu_id'=>$this->input->post('txt_id_usuario_estudiante'),
        'est_fecha_ingreso'=>$this->input->post('txt_fecha_ing_estudiante'),
        'est_fecha_nac'=>$this->input->post('txt_fecha_nac_estudiante'),
        'est_cedula'=>$this->input->post('txt_cedula_estudiante'),
        'est_nombre'=>$this->input->post('txt_nombre_estudiante'),
        'est_apellido'=>$this->input->post('txt_apellido_estudiante'),
        'est_correo'=>$this->input->post('txt_correo_estudiante'),
        'est_estado'=>'A',
        'doc_id'=>$this->input->post('txt_docente_estudiante'),
        'est_fecha_ingreso'=>date("Y") . "-" . date("m") . "-" . date("d"),
        'est_celular'=>$this->input->post('txt_celular_estudiante'))
    );

    //$data2=$this->input->post('id');
    $datos=$this->modelo_estudiante->guardar($data_estudiante);

    $this->modelo_usuario->asignado($data_estudiante['usu_id']);

    if($datos)
    {
        redirect("estudiante");
    }
}

```

Figura 30. Código php para la función guardar estudiante

4.3 Pruebas de Caja Negra

Se refieren a las pruebas funcionales. Que se realizan a partir de los requerimientos funcionales y no funcionales, a muy alto nivel, para diseñar pruebas que se aplican sobre el sistema sin necesidad de conocer como está construido por dentro.

Estas pruebas se realizaron, agendando reuniones con los usuarios del sistema. Los mismos que comprobaron el cumplimiento de los requisitos funcionales del sistema, realizando el proceso de agendamiento de citas, registro tanto de pacientes, docentes y estudiantes. A continuación se muestra el formulario con el cual se desarrolló este proceso.



Formulario de Prueba Docente (DATOS DEL DOCENTE):

- Fecha de ingreso:
- Fecha de salida:
- Fecha de nacimiento:
- Nombre de usuario:
- *Nombres:
- *Apellidos:
- Nº cédula:
- Correo electrónico:

Botón: Ingresar

Figura 31. Formulario de Prueba Docente

4.4 Casos de Prueba

4.4.1 Ingreso al Sistema

Tabla 9.

Prueba: Acceso a la aplicación

Identificador:	1
Usuario:	Administrador
Descripción:	Acceso al sistema de un usuario previamente creado
Variables de entrada:	usuario, clave
Proceso	En el explorador ingresar la dirección de la página web del aplicativo. En el formulario de login digitar el usuario y clave. Hacer click sobre el botón Ingresar.
Comportamiento esperado	La aplicación verifica la información ingresada. Después se muestra el menú de opciones que corresponde según el perfil del usuario que ha ingresado.
Ejecutado por:	Elizabeth Zeas
Resultado	El usuario accede a la aplicación sin ningún problema.
Observaciones:	Se realiza correctamente la validación para el usuario que ha ingresado.
Observaciones del programador	No olvidar sus credenciales de acceso

4.4.2 Registrar Información Estudiante

En la siguiente tabla se puede observar la prueba realizada siguiendo el proceso antes descrito.

Tabla 10.

Prueba: Ingreso de información Estudiante

Identificador:	2
Caso de Uso:	Ingresar Estudiante
Usuario:	Docente
Descripción:	Proceso mediante el cual los docente realizan el registro de los estudiantes

Identificador:	2
Datos entrada:	usuario , nombre , apellido ,celular, email
Flujo normal del evento:	Una vez creado el usuario y contraseña por parte del administrador. El docente ingresa a la aplicación con el debido perfil. Dirigirse a la opción Registrar Estudiante. Pulsar en el botón Nuevo Estudiante. Registrar los datos del estudiante. Presionar el botón Guardar.
Resultado	El sistema valida la información ingresada y guarda los datos del nuevo estudiante
Ejecutado por:	Elizabeth Zeas
Resultados Obtenidos:	El usuario realiza el registro de sus estudiantes de manera correcta

4.4.3 Registrar Información Paciente

A continuación, se observa la prueba realizada por parte de un estudiante de la clínica.

Tabla 11.

Caso de Prueba: Registrar Información Paciente

Identificador:	3
Usuario:	Estudiante
Descripción:	Proceso mediante el cual los estudiantes realizan el registro de sus pacientes
Variables de entrada:	Datos de entrada: número historia clínica, nombres, apellidos
Flujo normal del evento:	Ingresa a la dirección de la página web.Ir a la opción paciente. Hacer click en el botón nuevo paciente y llenar todos los campos
Resultado	La aplicación verifica los datos ingresados y los guarda en la base de datos.
Ejecutado por:	Daniela Del Salto
Resultados Obtenidos:	El usuario realiza el registro de sus pacientes de forma correcta

4.4.4 Actualizar Información

Tabla 12.

Caso de Prueba: Actualizar Información

Identificador:	4
Usuario:	Docente
Descripción:	Proceso mediante el cual los docente realizan la actualización de datos a través del sistema
Valores de entrada:	Nombres , Apellidos , Cedula, Correo Electrónico
Flujo normal del evento:	El administrador ingresa al sistema. Ir a la opción Usuarios >> Docentes. Pulsar en el botón editar. Actualizar los datos necesarios. Presionar el botón Guardar.
Resultado	El sistema comprueba la información registrada y actualiza los campos que se

4.4.5 Agendar Cita

En la siguiente imagen se puede ilustra el flujo actual del Sistema para agenda una cita.

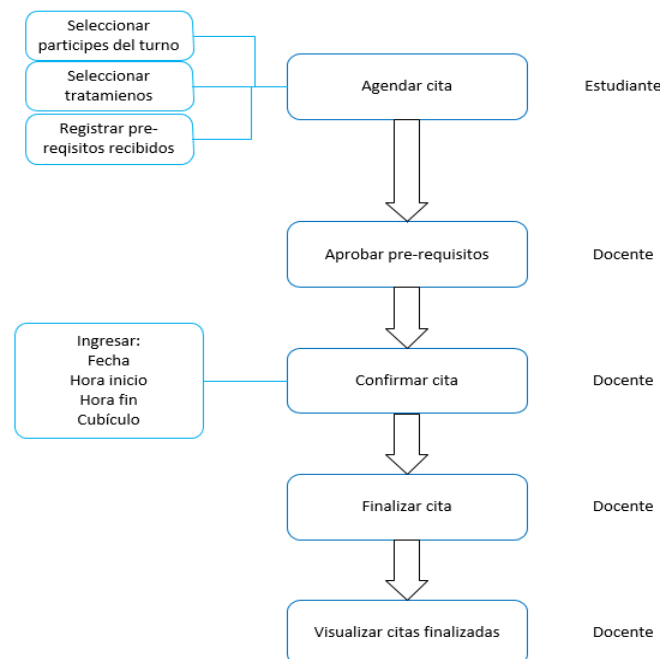


Figura 32. Flujo para Agendar Cita

A continuación, se describe el proceso de prueba realizado por un estudiante habiendo seguido el flujo descrito en la anterior imagen.

Tabla 13.

Caso de Prueba: Agendar Cita

Identificador:	5
Caso de Uso	Agendar Cita
Usuario:	Estudiante
Descripción:	Proceso mediante el cual los estudiantes agendan una cita
Variables de entrada:	Datos de entrada: fecha , estudiante encargado, paciente y tratamiento a realizarse
Proceso	Acceder al aplicativo con el perfil de estudiante. En el menú ir la opción citas. Hacer click en el botón agendar cita .
Resultado	Se realiza la validación de los datos ingresados y se guarda los datos

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La metodología empleada para el diseño de esta aplicación, ha permitido realizar un óptimo análisis de la situación actual de la clínica para la gestión de turnos de sus estudiantes. Con lo cual se definió claramente los módulos a desarrollar así como también los requisitos funcionales y no funcionales que son reflejados en el producto final obtenido. Además debido a las iteraciones que se utiliza en la metodología RUP se facilitó a los interesados en el proyecto la entrega de productos finales de una manera rápida y continua, para que de esta manera puedan percibir los avances que se iban implementando.

El desarrollo de la aplicación utilizando el lenguaje de programación php y como base de datos MySQL, permitieron que el proceso de implementación de la aplicación sea más sencillo e intuitivo, lo cual posteriormente ahorrara costos durante los mantenimientos.

El desarrollo de este proyecto permitirá tener a la clínica una herramienta mediante la cual podrán llevar una mejor administración tanto de sus clientes como del personal que se encarga de realizar la atención a los mismos. Ya que actualmente están expuestos a la pérdida de información, debido a que el proceso de gestión de citas se lo realiza manualmente.

Con este sistema se logra automatizar el proceso de toma de turnos por parte de los estudiantes, así como también facilitar a los docentes la aprobación de los requisitos a sus alumnos y con ello brindar una mejor atención.

Para el desarrollo de una aplicación de gestión de citas médicas es muy importante tomar en cuenta todos los escenarios posibles, así como obtener toda la información por parte de los usuarios, de esta manera se obtiene un mejor resultado y se evita la pérdida de tiempo innecesaria en el proceso de programación.

Un sistema puede realizar innumerables funciones y puede ser lo más amigable e intuitivo, pero si los usuarios no tienen la adecuada capacitación no se le sacará el máximo provecho al mismo. Una aplicación web de este tipo conlleva

mucho tiempo y en ocasiones cuesta mucho dinero dependiendo de las herramientas, licencias, etc., que se utilicen, por lo cual se debe especificar desde un inicio todos los requerimientos por parte de cada uno de los usuarios, de esta manera se obtendrá al final una herramienta útil, rápida y efectiva que facilitará el desempeño de los usuarios y podrán trabajar sin inconvenientes en el futuro.

5.2 Recomendaciones

Al momento de crear su usuario y contraseña utilizar caracteres especiales, alfanuméricos, incluir letras mayúsculas para que de esta manera no existan problemas de acceso no autorizados y alteración de información.

Debido a que esta aplicación constituye únicamente en un prototipo, se decidió implementar su base de datos en MySQL, por lo cual la capacidad para guardar información es reducida, a futuro se recomienda realizar una migración de base de datos para de esta manera cumplir con la demanda que existirá por parte de los usuarios.

Se recomienda al administrador del sistema monitorear continuamente el proceso de agendamiento de citas por parte de los estudiantes para de esta manera evitar problemas en cuanto a la asignación de cubículo y horario de atención al cliente.

Se sugiere la debida capacitación a cada uno de los usuarios del sistema para sacarle el máximo provecho al mismo y con ello no colmar al administrador con todo el trabajo de ingresos de usuarios, pacientes, etc.

Es recomendable mantener actualizado el sistema de manera continua con los datos físicos obtenidos por la clínica, para de esta que de esta manera no se presenten inconvenientes al momento de que los usuarios requieran consultar información acerca de horarios, tratamientos, pacientes.

REFERENCIAS

- Bahit, E.(2008). El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC.Recuperado el 20 de Noviembre de 2016 de: <http://www1.herrera.unt.edu.ar/biblcet/wp-content/uploads/2014/12/eugeniabahitpooymvcenphp.pdf>
- Barnes, D. J., & Kölling, M. (2007). Programación orientada a objetos con Java. MANEJO EFICIENTE DEL TIEMPO.
- Berzal, F., Cubero, J. C., & Cortijo, F. J. (2004). Desarrollo Profesional de Aplicaciones Web con Asp. net. iKor Consulting. México: Addison Wesley Longman.
- Booch, G. (1996). Análisis y diseño orientado a objetos. México: Addison Wesley Longman.
- Booch, G., Rumbaugh, J., Jacobson, I., & Molina, J. J. G. (1999). El lenguaje unificado de modelado (Vol. 1). Madrid, Español: Addison-Wesley.
- Bootstrap. (2015.). Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. Recuperado el 24 de septiembre del 2016 de: <http://getbootstrap.com/>
- Bourdon, R. (2016). Wampserver. Recuperado el 2 de marzo del 2017 de: <http://www.wampserver.com/en/>

British Columbia Institute of Technology. (2016). Welcome to CodeIgniter.

Recuperado el 26 de septiembre del 2016, de codeigniter.com Sitio web: http://www.codeigniter.com/user_guide/general/welcome.html

Canós, J. H., Letelier, P., & Penadés, M. C. (2003). Metodologías ágiles en el desarrollo de software. Universidad Politécnica de Valencia, Valencia, España: Addison Wesley Longman.

Carabalí, M. 2013. Pruebas de Caja Negra y Caja Blanca. Recuperado el 2 de Diciembre de 2016 de <https://prezi.com/sjwfwmix7slk/pruebas-de-caja-negra-y-caja-blanca/#>

Efron, B., & Efron, B. (1982). The jackknife, the bootstrap and other resampling plans (Vol. 38). Philadelphia, Estados Unidos: Society for industrial and applied mathematics.

Elmasri, R., Navathe, S. B., Castillo, V. C., Espiga, B. G., & Pérez, G. Z. (2002). Fundamentos de sistemas de bases de datos. México: Addison-Wesley.

Falaki, H., Lymberopoulos, D., Mahajan, R., Kandula, S., & Estrin, D. (2010, November). A first look at traffic on smartphones. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (pp. 281-287). ACM.

Fernandez A. 2010. El diseño de la interfaz. Recuperado el 9 de Noviembre de 2016 de <http://www.lawebera.es/disenio-web/el-diseno-de-la-interfaz.php>

Flanagan, D. (2006). JavaScript: the definitive guide. New York, Estados Unidos: O'Reilly Media, Inc.

Gutiérrez D. (2011). UML Diagramas de Clases. Recuperado el 15 de Octubre de 2011 de http://www.codecompiling.net/files/slides/UML_clase_04_UML_clases.pdf

Hubbard, J. (2017). SQLyog is the most complete and easy to use MySQL GUI. Recuperado el 16 de Octubre del 2016 de: <https://www.webyog.com/product/sqllyog>

Jacobson, I., Booch, G y Rumbaugh, J. (2000). El proceso Unificado de desarrollo de software. Madrid, España: Pearson Educación S.A

JavaScript Object Notation. (2016). Introducing JSON. Recuperado el 27 de septiembre del 2016 de: <http://www.json.org/>

Joyanes, L. (2001). Fundamentos de programación, algoritmos y estructuras de datos. México : McGraw-Hill

LanceTalent. 6 Buenos Motivos Para Trabajar Con PHP. (2016). Recuperado el 19 de Octubre del 2016 de: <https://www.lancetalent.com/blog/6-buenos-motivos-para-trabajar-con-php/>

Microsoft Developer Network. (2016). Diagramas de casos de uso de UML: Referencia. Recuperado el 15 de Octubre de 2017 de: <https://msdn.microsoft.com/es-es/library/dd409427.aspx>

Microsoft Developer Network. (2016). Diagramas de secuencia UML: Referencia.

Recuperado el 15 de Octubre de 2017 de:
<https://msdn.microsoft.com/es-es/library/dd409427.aspx>

Microsoft Visio. (2017). Recuperado el 2 de Marzo del 2017 de:

https://es.wikipedia.org/wiki/Microsoft_Visio

Microsoft. (2016). Características de Microsoft SQL Server. Recuperado el 28 de

septiembre del 2016 de <https://msdn.microsoft.com/es-es/library/ms174219.aspx>

Pressman, R. S., & Troya, J. M. (1988). Ingeniería del software (No. 001.64

P74s.). México: McGraw Hill.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). Fundamentos de bases

de datos (No. 001.64 S719f 2003.). Nueva York, Estados Unidos: McGraw-Hill.

Softeng. (2016). Metodología Ágil para el desarrollo de software. Recuperado el

11 de junio de 2016 de <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/softeng-agile.html>

Stallings, W. (1997). Sistemas operativos (Vol. 732). Madrid, España: Prentice

Hall.

Sublime text. (2016). Sublime Text. Recuperado el 26 de septiembre del 2016

de: <https://www.sublimetext.com/>

Tuya, J., Román, I. R., & Cosín, J. D. (Eds.). (2007). Técnicas cuantitativas para

la gestión en la ingeniería del software. Madrid, España: NetBiblo.

Uneweb. (2017). Recuperado el 2 de Marzo del 2017 de:
<http://tecnologiaenvivo.com/metodo-de-javascript-para-animaciones-de-alta-calidad/>

Universidad Interamericana para el Desarrollo. (s.f). Programación Orientada a Objetos. Recuperado el 13 de Octubre de 2016 de:
http://moodle2.unid.edu.mx/dts_cursos_md/iel/POO/S05/POO05_Lectura.pdf

Webyog. (2016.). Tools to Manage and Monitor MySQL Servers. Recuperado el 24 de septiembre del 2016 de: <https://www.webyog.com/>

ANEXOS

Anexo a)

Manual de instalación

Paso 1: Dirección Web wampserver

Ir a la dirección web <http://www.wampserver.es/> para descargar wampserver.

Paso 2: Descarga Wamp Server

Elegir el instalador según la versión del sistema operativo (32 o 4 bits)

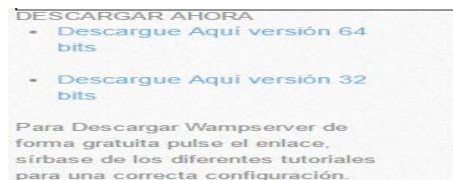


Figura 33. Version Wamp Server

Comprobar si el servicio de wampserver está ejecutándose.

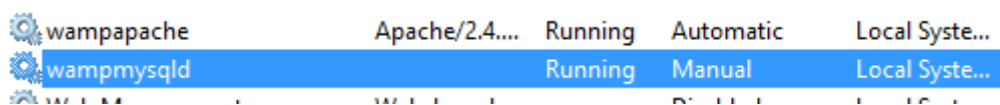


Figura 34. Servicios Wamp Server Ejecutándose

Paso 3: Copiar la carpeta del sistema

Dentro del CD ingresar a la carpeta “PROGRAMA” y copiar la carpeta “sistema_clinica”

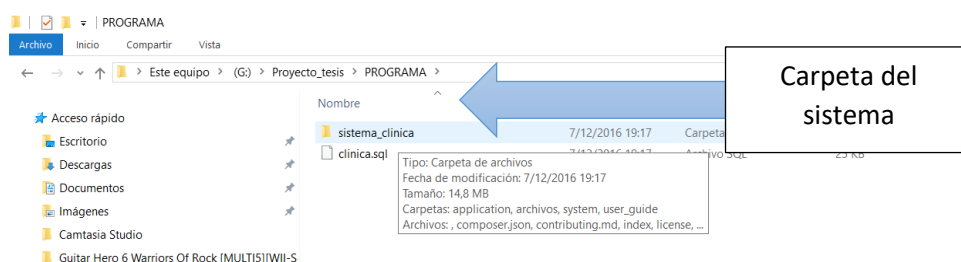


Figura 35. Carpeta del sistema

Paso 4: Insertar la carpeta del sistema en el servidor

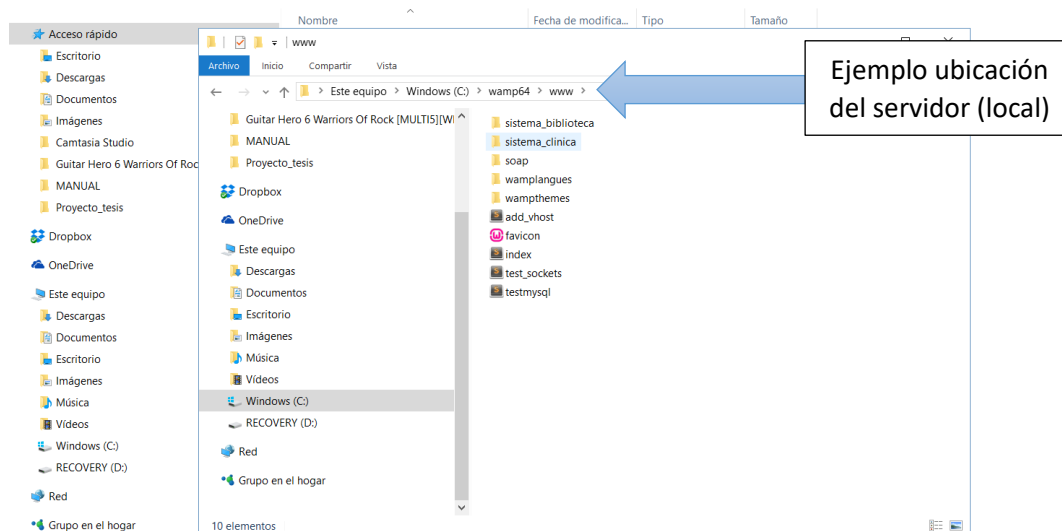


Figura 36. Ubicación del servidor localmente

Paso 5: Configurar la dirección de dominio

Dentro de la carpeta del programa pegado en el servidor, abrir el archivo “config.php” siguiendo la siguiente extensión: sistema_clinica\application\config y modificar “localhost” por la dirección de dominio deseada como se muestra a continuación:

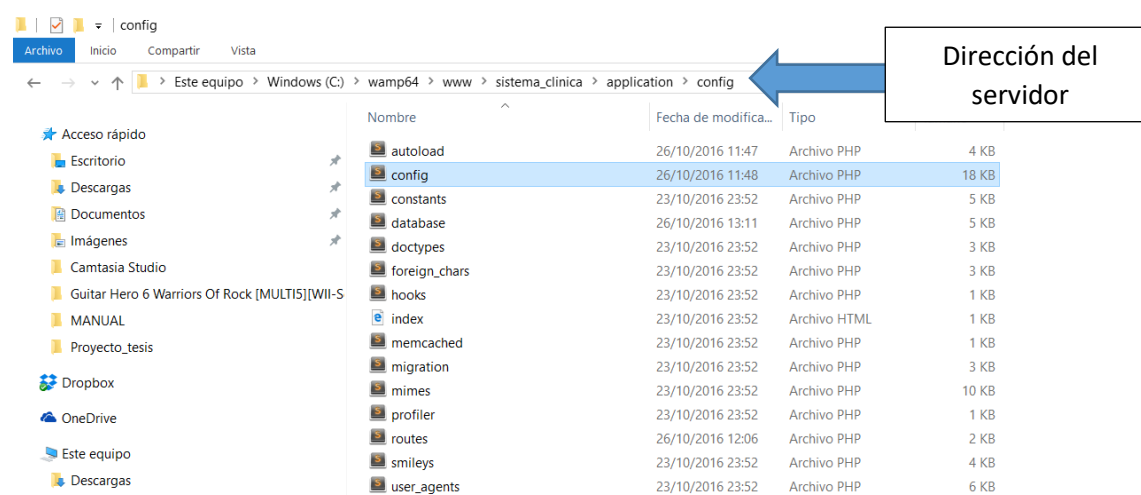


Figura 37. Archivos de configuración

```

| If it is not set, then CodeIgniter will try guess the protocol and path
| your installation, but due to security concerns the hostname will be set
| to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
| The auto-detection mechanism exists only for convenience during
| development and MUST NOT be used in production!
|
| If you need to allow multiple domains, remember that this file is still
| a PHP script and you can easily do that on your own.
|
| */
$config['base_url'] = 'http://localhost/sistema_clinica/';
|
| /*
| -----
| Index File
| -----
|
| Typically this will be your index.php file, unless you've renamed it to
| something else. If you are using mod_rewrite to remove the page set this
| variable so that it is blank.
|
| */
$config['index_page'] = 'index.php';

```

Figura 38. Configuración de la dirección de dominio

Paso 6: Configurar conexión a la base de datos

Dentro de la carpeta anterior abrir el archivo “database.php” y cambiar los datos mostrados con los valores de conexión del servidor de BD.

```

73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'clinica',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );

```

Datos de conexión a la base de datos del servidor

Figura 39. Datos de la conexión a la base de datos

Paso 7: Importar base de datos

En el navegador ir a la dirección <http://localhost/phpmyadmin/> elegir la opción base de datos. Donde se observará la siguiente imagen. Ingresar el nombre de la base de datos “clinica” y dar click en crear.

Bases de datos



Crear base de datos ?

Cotejamiento ▼ Crear

Figura 40. Direccion URL para importar la base de datos

Verificar si se creó la base de datos

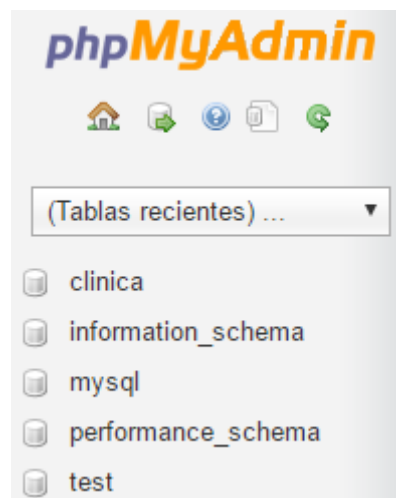


Figura 41. Verificar la creación de la base de datos

Después ir a la opción importar base de datos

localhost

Bases de datos SQL Estado actual Exportar Importar Configuración Sincronizar

Importando al servidor actual

Archivo a importar:

El archivo puede ser comprimido (gzip, zip) o descomprimido.
Un archivo comprimido tiene que terminar en **[formato].[compresión]**. Por ejemplo: **.sql.zip**

Buscar en su ordenador: **Choose File** No file chosen (Máximo: 2,048KB)

Conjunto de caracteres del archivo: utf-8

Importación parcial:

☒ Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría sin embargo, puede dañar las transacciones.)

Número de filas a omitir, iniciando de la primer fila: 0

Formato:

SQL

Opciones específicas al formato:

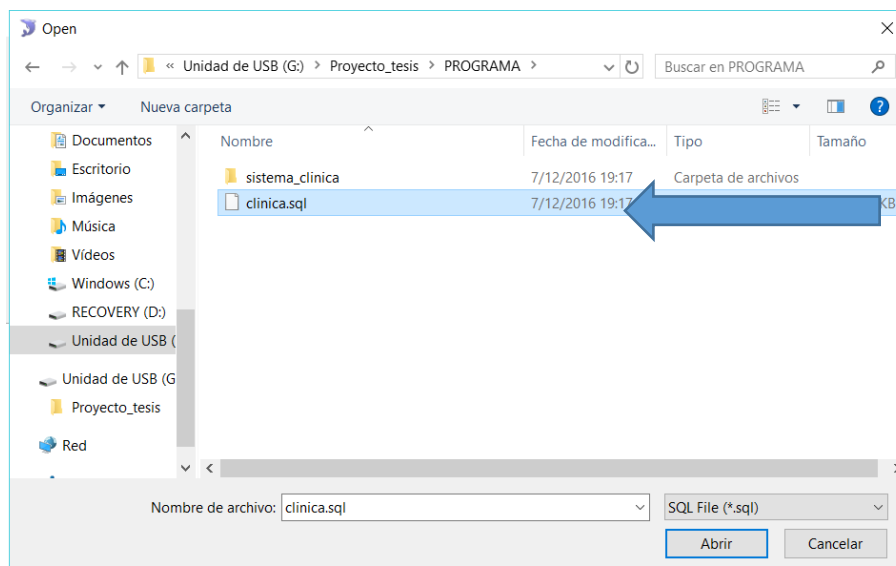
Modalidad SQL compatible: NONE

☒ No utilizar AUTO_INCREMENT con el valor 0

Continuar

Figura 42. Menú para importar la base de datos

Ir a la opción Elegir archivo, donde importaremos el script sql



Script de la
base de
datos de la
clínica

Figura 43. Importar el script de la base de datos

Damos click en continuar y aparecerá el siguiente mensaje

✓ La importación se ejecutó exitosamente, se ejecutaron 87 consultas. (clinica.sql)

Figura 44. Notificación de importación de la base de datos

Paso 8: Ingreso al sistema

Desde un navegador ingresar la dirección de dominio del servidor más “/sistema_clinica”, a continuación, se mostrará la pantalla de inicio al sistema.



Figura 45. Ingreso al sistema

Credenciales de acceso

Perfil Administrador

Usuario : admin

Clave: cUd!@-2016

Para el perfil docente:

Usuario: carlosiz

Clave: 123

Para perfil estudiante:

Usuario: epvargas

Clave: epvargas

Anexo b)

Manual de usuario

1. Elementos principales del Sistema

Formulario

En un formulario se deben ingresar varios parámetros en los campos de texto mostrados (al menos los marcados con “*”) y presionar en el botón de confirmación para realizar alguna acción dependiendo de las opciones mostradas en pantalla.

Interfaz de Login

En la siguiente figura se muestra la pantalla inicial para el acceso al sistema, mediante la cual todos los usuarios del sistema se autenticarán haciendo el uso de su usuario y clave previamente creados.

El diagrama ilustra la interfaz de login de la Universidad de las Américas. En la parte superior, se encuentra el logo de la universidad, que consiste en un símbolo rojo estilizado y el texto "UNIVERSIDAD DE LAS AMÉRICAS" y "Laureate International Universities". Debajo del logo, hay un formulario con los siguientes elementos:

- Un campo de texto etiquetado como "* Usuario" con el placeholder "Nombre de usuario para el ingreso".
- Un campo de texto etiquetado como "* Contraseña" con el placeholder "Contraseña del usuario".
- Un botón azul con el texto "Ingresar".
- Una barra horizontal que sirve como menú.

Señales de diseño:

- Una llave de corchete azul agrupa los dos campos de texto, con una etiqueta "Campos de texto" a su derecha.
- Una flecha azul apunta desde la barra de menú hacia el botón "Ingresar", con una etiqueta "Botón de Opciones del menú" a su derecha.

Figura 46. Interfaz Ingreso Sistema

Interfaz de bienvenida al Sistema

En esta sección se muestra el menú principal de bienvenida al Sistema el cual muestra el logo de la institución, información del usuario, además de una breve información acerca de la clínica sus horarios de atención y contacto.



Figura 47. Bienvenida al Sistema

Interfaces de gestión de entidades

El sistema cuenta con tablas dinámicas para la gestión de los distintos tipos de usuarios y demás entidades como tratamiento, requisito, etc. Estas tablas permiten de manera rápida la gestión de los datos (creación, modificación, eliminación). Además se ha agregado un botón en la parte superior de la tabla para el ingreso de nuevos registros.

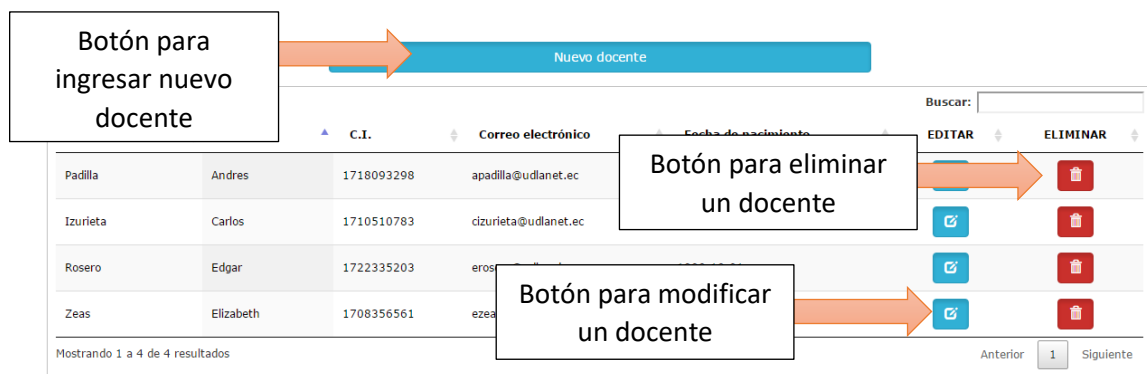


Figura 48. Gestión de Entidades

Ventanas modales

Se ha implementado el uso de pantallas modales con el fin de que la gestión de la información sea mucho más rápida y más eficaz, y se muestre de mejor manera al usuario puesto que estas ventanas aparecerán solo en el momento adecuado, sin necesidad de que se recargue la página ya que se cargará en la misma página, pero contará con las mismas funcionalidades de una ventana ordinaria de html.

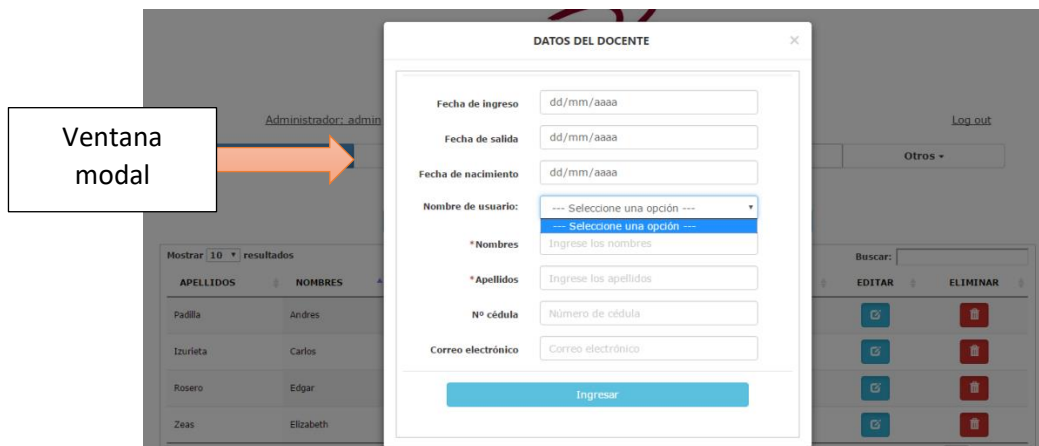


Figura 49. Ventana Modal Docente

Menú principal

El menú principal muestra todas las opciones disponibles para el usuario que acceda al sistema.



Figura 50. Menú principal del sistema

Submenús

En las opciones principales del menu se muestran algunas opciones adicionales en listas desplegables como se muestra en la siguiente imagen.



Figura 51. Menú desplegable para Citas

Tablas

Las tablas muestran una lista de datos según las distintas entidades a las que acceda el usuario, y además muestran botones adicionales para la gestión de datos como: editar, eliminar, etc., como se muestra a continuación:



Figura 52. Lista de datos para Entidad Docente.

Además de los datos de la tabla y los botones de acción, las tablas tienen varias opciones de navegación como las que se muestran a continuación.

Opción de búsqueda (Disponible para cualquier tabla con el campo de búsqueda)

Para esta opción se ingresa el nombre del atributo (nombre de usuario, clave, rol, estado o fecha de creación), en el campo de búsqueda de la tabla de gestión de usuarios:

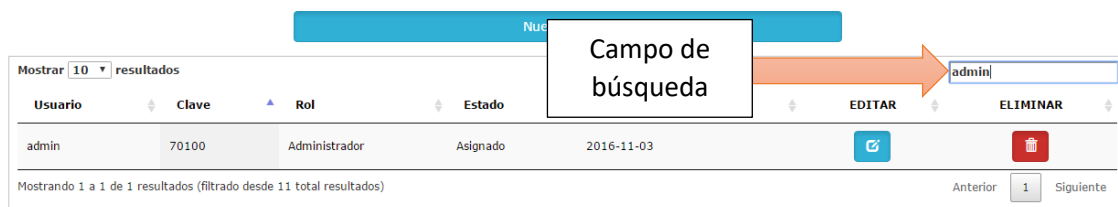


Figura 53. Campo de Búsqueda

Opción resultados

Muestra la cantidad de filas que se aparecerán en la tabla

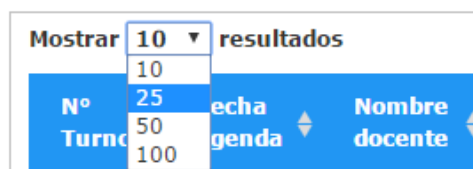


Figura 54. Opción de resultados a mostrarse en la tabla

Opción de página

Permite seleccionar la página que se visualizará dentro de la tabla según el número de página seleccionado.

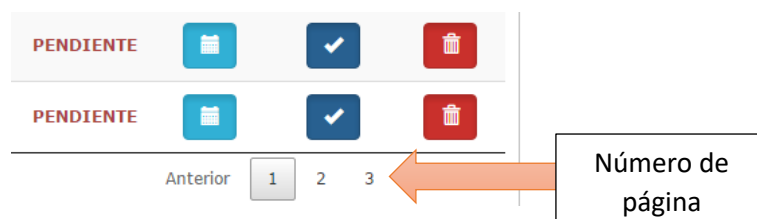


Figura 55. Opción para visualizar página

Campos

Mensajes de alerta

Los mensajes de alerta se muestran antes de realizar una acción decisiva en el sistema, como eliminar a un estudiante o paciente de una tabla de datos.

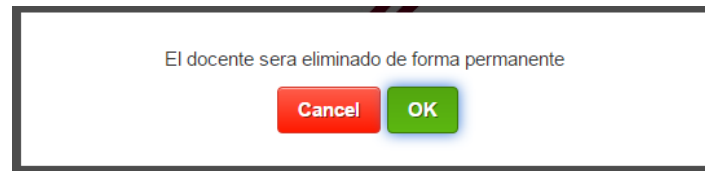


Figura 56. Mensajes de alerta

Cajas de confirmación (Checkbox)

Estos campos permitirán seleccionar una opción de una lista del sistema, aquellos marcados con visto serán seleccionados mientras los demás serán descartados.

Requisitos entregados:

<input checked="" type="checkbox"/>	historia clinica llena
<input type="checkbox"/>	hoja de remisión
<input checked="" type="checkbox"/>	he
<input checked="" type="checkbox"/>	radiografía panorámica
<input checked="" type="checkbox"/>	factura de cancelación previa
<input type="checkbox"/>	exámenes de laboratorio
<input type="checkbox"/>	hoja de vida

Confirmar cita

Campo de confirmación (seleccionado)

Figura 57. Cajas de Confirmación

2. Ingreso al sistema

Paso 1: Acceder a la página del Sistema

Para acceder al sistema primer se debe ingresar a un navegador e ingresar el nombre de la página otorgada por el administrador, localmente se ingresa con: localhost/sistema_clinica

Se mostrará la siguiente pantalla:



The screenshot shows a login interface for a system titled "Sistema de gestión de citas médicas". Below the title is the subtitle "Acceso para los usuarios del sistema". The interface features the logo of the Universidad de las Américas (UNDA) at the top center. Below the logo, there are two input fields: one for the username labeled "*Usuario" and another for the password labeled "*Contraseña". The username field contains the placeholder text "Nombre de usuario para el ingreso". The password field contains the placeholder text "Contraseña del usuario". A blue button labeled "Ingresar" is positioned below the password field.

Figura 58. Ingreso al Sistema

Paso 2: Ingreso de datos de usuario

Para ingresar al sistema se debe acceder con el usuario y contraseña obtenidos por parte del administrador del sistema.



This screenshot shows the same login interface as Figure 58, but with user data entered. The username field now contains the text "admin". The password field contains a series of dots, indicating that the password has been masked. The blue "Ingresar" button remains at the bottom.

Figura 59. Datos de usuario

A continuación, se observa la pantalla de bienvenida, la cual mostrará un menú dependiendo del perfil del usuario (Administrador, docente, estudiante):

Menú administrador



Figura 60. Menú Administrador

Menú docente



Figura 61. Menú Docente

Menú estudiante

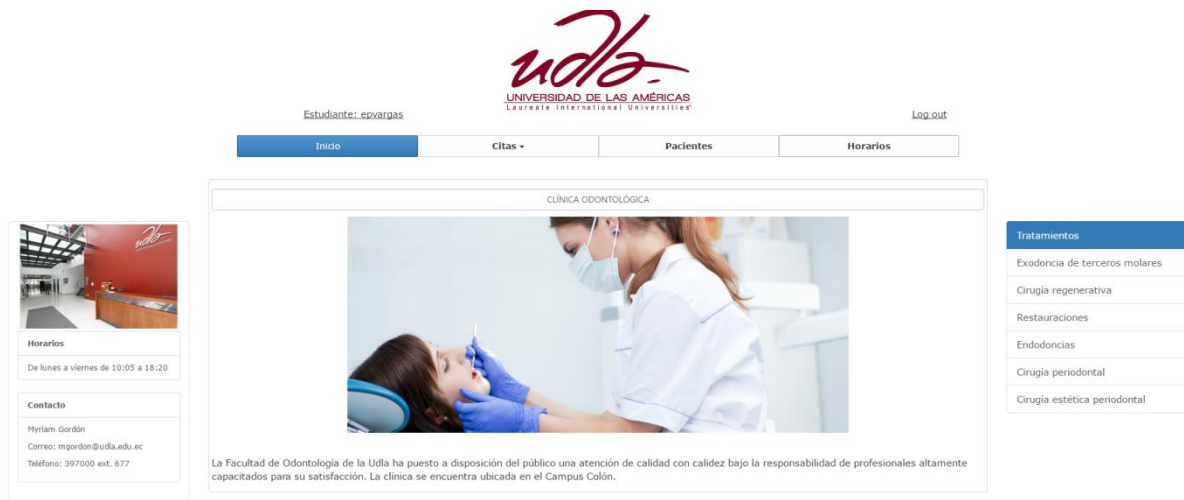


Figura 62. Menú Estudiante

En el menú principal se muestran varias opciones e información acerca de la clínica como se muestra en la siguiente imagen:



Figura 63. Opciones menú principal

3. Ingreso de usuarios

Paso 1: Ingresar a la opción “Gestión usuarios”

Para el ingreso de un nuevo usuario, el administrador del sistema deberá acceder e ingresar a la opción gestionar usuarios del menu principal como se muestra a continuación:



Figura 64. Opción para el ingreso de usuarios

Paso 2: Seleccionar la opción “Nuevo Usuario”

A continuación se debe seleccionar la opción para ingresar un nuevo usuario dependiendo del perfil de usuario que se requiera crear existen dos opciones: docente y estudiante.



Figura 65. Botón Nuevo Usuario

Luego de ingresar los datos del usuario, presionar la opción “Ingresar”, luego de este proceso se podrá observar los datos del usuario en tabla usuarios como se muestra a continuación:

Mostrar 10 resultados Buscar:

Usuario	Clave	Rol	Estado	Fecha creación	EDITAR	ELIMINAR
admin	70100	Administrador	Asignado	2016-11-03		
gabs	70101	Estudiante	Asignado	2016-11-03		
elzeas	70102	Docente	Asignado	2016-11-03		
carlosiz	70103	Docente	Asignado	2016-11-03		
epvargas	70104	Estudiante	Asignado	2016-11-03		
cizurieta	70105	Docente	Asignado	2016-11-03		
pcastro	70106	Estudiante	Asignado	2016-11-16		
apadilla	70107	Docente	Asignado	2016-11-16		
pledasma	70200	Estudiante	Asignado	2016-11-16		
dvalencia	70600	Estudiante	Asignado	2016-12-01		

Figura 66. Lista de usuarios ingresados

Para registrar un usuario como docente o estudiante se procede dentro del menú usuarios como se muestra a continuación:

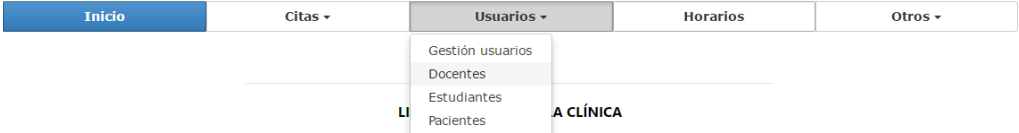


Figura 67. Opción para registrar un Docente

Administrador: admin Log out

Inicio Citas Usuarios Horarios Otros

Nuevo docente

Mostrar 10 resultados					Buscar: <input type="text"/>	
APELLIDOS	NOMBRES	C.I.	Correo electrónico	Fecha de nacimiento	EDITAR	ELIMINAR
Padilla	Andres	1718093298	apadilla@udianet.ec	2016-11-25		
Izurietta	Carlos	1710510783	cizurieta@udianet.ec	1990-10-01		
Rosero	Edgar	1722335203	erosero@udia.edu.ec	1990-10-01		
Zeas	Elizabeth	1708356561	ezeas@udia.edu.ec	1990-10-01		

Mostrando 1 a 4 de 4 resultados Anterior 1 Siguiente

Figura 68. Botón nuevo Docente

A continuación, se selecciona la opción Nuevo (docente/estudiante) para ingresar al formulario de ingreso como el que se muestra en la siguiente imagen.



Formulario de registro de un nuevo docente. El formulario se titula "DATOS DEL USUARIO" y contiene los siguientes campos:

- Perfil de usuario:** Seleccionado "Docente".
- Fecha creación:** 23/01/2017.
- Fecha exp.:** 2100-01-01. Ejemplo: 23/01/2017 - 01/01/2100.
- *Código:** Número de cédula (10 dígitos).
- *Usuario:** Ingrese el nombre de usuario (Max 20 caracteres).
- *Contraseña:** Ingrese la contraseña (Max 20 caracteres).

Botón de acción: Ingresar.

Figura 69. Formulario para el registro de un nuevo docente

4. Modificar un usuario

La interfaz para modificar un docente se maneja por medio de una ventana modal parecida a la de ingreso con la única diferencia que esta cargará automáticamente los datos del docente seleccionado en la tabla de gestión de docente.



Interfaz para modificar la información del docente. El formulario se titula "DATOS DEL DOCENTE" y contiene los siguientes campos:

- Fecha de ingreso:** yyyy-mm-dd.
- Fecha de salida:** 2016-11-12. (De: 23/01/2017 en adelante).
- *Nombres:** Andres (Max 20 caracteres).
- *Apellidos:** Padilla (Max 20 caracteres).
- Correo electrónico:** apadilla@udlanet.ec (Max 50 caracteres).
- Fecha de nacimiento:** 2016-11-25. (Ej: 01/01/1990).

Botón de acción: Guardar.

Figura 70. Interfaz para modificar la información del docente

Nota: Para la gestión de docente, estudiante, requisito y tratamiento se manejarán los mismos modelos de interfaces con la única diferencia de los distintos campos que maneja cada entidad.

5. Agendar cita

Para agendar citas a sus pacientes los estudiantes deben seguir los siguientes pasos:

1. Ingresar al sistema utilizando su usuario y contraseña. Ir a la opción de citas médicas >> agendar cita. Donde aparecerá el siguiente formulario. El estudiante deberá escoger su nombre de la lista en el campo estudiante y el paciente previamente registrado, además elegir el(los) tratamiento(s) y dar click en confirmar cita.



Formulario de agendar cita médica:

- Turno N°: 0034
- Fecha agenda: 2016-12-01
- Estudiante: Sampertegui Gabriela
- Paciente: Jairala Alex
- * Tratamientos:
 - ☒ Alargamiento
 - ☐ Raspado
 - ☒ insición
- Botón: Confirmar cita

Figura 71. Datos de la cita medica

2. Entregar al docente respectivo los requisitos necesarios para que se pueda realizar la confirmación de la cita. El docente encargado deberá ingresar al sistema y buscar el turno para aprobar los requisitos.

Para realizar la búsqueda de turnos se debe empezar por "00" y digitar el número de turno correspondiente como se muestra a continuación



Formulario de búsqueda de turnos:

- Turno N°: 0034
- Botón: Buscar

Figura 72. Búsqueda de prerrequisitos

REQUISITO	ESTADO	* APROBAR	* DESAPROBAR
historia clínica llena	APROBADO		
hoja de remisión	APROBADO		
hoja de interconsulta	APROBADO		
radiografía panorámica	APROBADO		
factura de cancelación previa	APROBADO		
exámenes de laboratorio	APROBADO		
hoja de vida	APROBADO		

Figura 73. Aprobación de requisitos

3. Ir a la opción Citas >> Lista de turnos pendientes, para verificar el estado de la cita médica.

N° Turno	Fecha agenda	Nombre Estudiante	Paciente	Fecha cita	Hora	PRE-REQUISITOS	ESTADO	CONFIRMAR	FINALIZAR	ANULAR
0034	2016-12-01	Sampertegui Gabriela	Jairala Alex	PENDIENTE	PENDIENTE	APROBADOS	PENDIENTE			

Figura 74. Estado de la Cita Médica

4. Una vez aprobados todos los requisitos, el docente debe realizar la confirmación de la cita. Asignando la fecha, la hora de inicio, la hora de finalización y el cubículo donde el estudiante atenderá al paciente

DATOS DEL TURNO

Turno N°:

0034

Fecha agenda:

2016-12-01

Estudiante:

Gabriela

Paciente:

Alex

*Fecha cita:

2016-12-08

*Hora inicio:

08:30 AM

*Hora fin:

09:30 AM

*Cubículo:

2

Guardar

Figura 75. Datos del turno

5. Una vez confirmada la cita, el docente deberá realizar la finalización de la misma, donde puede registrar la asistencia del estudiante , así como también realizar las respectivas observaciones

DATOS DEL TURNO ✕

Turno N°:	<input type="text" value="0034"/>
Fecha agenda:	<input type="text" value="2016-12-01"/>
Estudiante:	<input type="text" value="Gabriela"/>
Paciente:	<input type="text" value="Alex"/>
Fecha cita:	<input type="text" value="2016-12-08"/>
Hora inicio:	<input type="text" value="08:30 AM"/>
Hora fin:	<input type="text" value="09:30 AM"/>
Cubiculo:	<input type="text" value="2"/>
*Asistencia:	<input type="text" value="Con asistencia"/>
*Observación:	<div style="border: 1px solid #ccc; padding: 5px; min-height: 30px;">sin observaciones</div>

Figura 76. Finalización de turno

Turnos/Pre-requisitos

En esta ventana se dispone en un inicio solo un campo de texto en el cual se debe ingresar el número de turno, presionar el botón “Buscar”, posteriormente se mostrarán los datos del turno, como se muestra en las siguientes imágenes.



UNIVERSIDAD DE LAS AMÉRICAS
Laureate International Universities

Administrador: admin
Log out

Inicio

Citas ▾

Usuarios ▾

Horarios

Otros ▾

PRE-REQUISITOS DEL TURNO ✕

Turno N°:

Figura 77. Interfaz para la búsqueda de turnos

En la siguiente figura se muestran los detalles del turno ingresado así como la lista de tratamientos y requisitos con su estado actual, de la misma se podrá realizar la búsqueda de un nuevo turno o a su vez se podrá aprobar los requisitos del turno mostrado.

The interface displays the following details for a specific turn:

- Turno N°:** 0030
- Fecha agenda:** 29/11/2016
- Paciente:** Alvarez Luis
- Estudiante:** Ledesma Patricio

Tratamientos:

- Alargamiento
- Raspado
- insición

Requisitos:

Requisitos	Estado	*RECIBIR & APROBAR
historia clínica llena	APROBADO	
hoja de remisión	APROBADO	
hoja de interconsulta	APROBADO	
radiografía panorámica	APROBADO	
factura de cancelación previa	ENTREGADO	
exámenes de laboratorio	ENTREGADO	

Annotations:

- Detalles del turno:** Points to the fields for Turno N°, Fecha agenda, Paciente, and Estudiante.
- Lista de tratamientos:** Points to the list of treatments.
- Botón para la aprobación de un requisito:** Points to the thumbs up icon in the 'Requisitos' table.
- Estado actual de un requisito:** Points to the 'Estado' column in the 'Requisitos' table.

Figura 78. Interfaz detalles turno

El estado mostrará una de 3 opciones posibles:

- **Pendiente:** Significa que el requisito no ha sido entregado
- **Entregado:** El requisito ha sido entregado, pero no ha sido aprobado
- **Aprobado:** El requisito ha sido entregado y aprobado

Lista turnos pendientes

En la siguiente interfaz se muestra una lista de todos los turnos pendientes en la clínica, es decir todos los turnos que aún no hayan sido tratados, la lista incluirá aquellos turnos que tengan requisitos aprobados o aun sin aprobar.

Esta tabla también permitirá al usuario aprobar, finalizar y anular un turno, como se muestra a continuación.

Inicio

Citas

Usuarios

Horarios

Otros

LISTA DE TURNOS DE LA CLÍNICA

Mostrar 10 resultados

Buscar:

Nº Turno	Fecha agenda	Nombre docente	Nombre estudiante	Nombre paciente	Fecha cita	Hora	REQUISITOS	ESTADO	Aprobar	Finalizar	Anular
0035	2016-12-01	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0034	2016-12-01	Zeas Elizabeth	Sampertegui Gabriela	Sánchez Oscar	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0033	2016-12-01	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0032	2016-12-01	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0030	2016-11-29	Zeas Elizabeth	Ledesma Patricio	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0026	2016-11-28	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0025	2016-11-28	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0024	2016-11-27	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0023	2016-11-27	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			
0022	2016-11-27	Zeas Elizabeth	Sampertegui Gabriela	Alvarez Luis	PENDIENTE	PENDIENTE	PENDIENTES	PENDIENTE			

Mostrando 1 a 10 de 22 resultados

Anterior

1

2

3

Siguiente

Figura 79. Lista de turnos pendientes

En esta tabla se mostrarán los detalles del turno, así como su estado los cuales se detallan a continuación.

- Nº turno: Indica el número de la cita médica dentro de la clínica

- Fecha agenda: Representa la fecha en la que se ingresó por primera vez la petición para la cita.
- Nombre docente: Muestra el nombre del doctor a cargo del turno.
- Nombre del estudiante: Muestra el nombre del estudiante que agendo la cita para un paciente.
- Nombre del paciente: Nombre del paciente que será atendido.
- Fecha cita: Es la fecha en la que el paciente será atendido por el médico.
- Hora: Muestra la hora exacta en la que el paciente debe ser atendido.
- Requisitos: Muestra si todos los requisitos del turno han sido entregados (PENDIENTES/APROBADOS)
- Estado: Muestra el estado en el que se encuentra actualmente la cita, puede ser “PENDIENTE” si aún o ha sido confirmada o “ACTIVO” si el turno ya ha sido confirmado es decir ya tiene una hora, fecha y cubículo asignado.

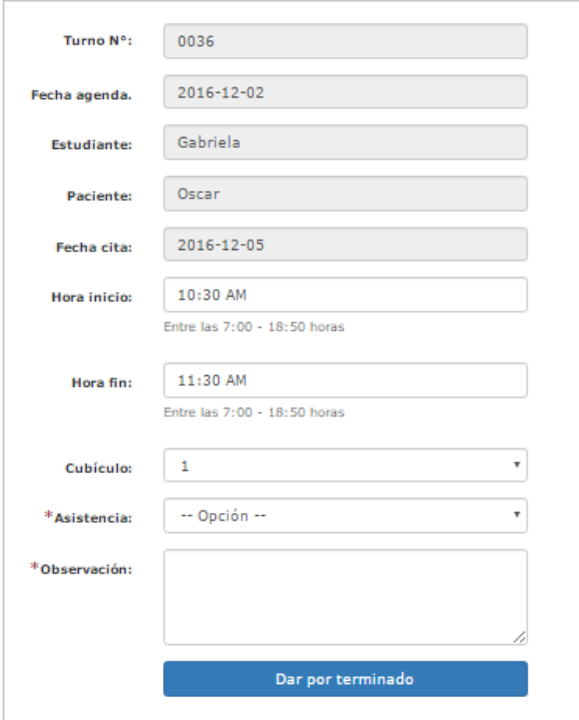
Para aprobar una cita se debe seleccionar la opción aprobar del turno y a continuación ingresar la hora de inicio, hora de fin y el número de cubículo para la cita médica.

DATOS DEL TURNO ✕

Turno N°:	<input type="text" value="0035"/>
Fecha agenda:	<input type="text" value="2016-12-01"/>
Estudiante:	<input type="text" value="Sampertegui Gabriela"/>
Paciente:	<input type="text" value="Alvarez Luis"/>
*Fecha cita:	<input type="text" value="2017-01-23"/> <small>(Entre: 23/01/2017 y 01/01/2100)</small>
*Hora inicio:	<input type="text" value="--:-- --"/> <small>Entre las 7:00 - 18:50 horas</small>
*Hora fin:	<input type="text" value="--:-- --"/> <small>Entre las 7:00 - 18:50 horas</small>
Cubículo:	<input type="text" value="--- Seleccione una opción ---"/>

Figura 80. Interfaz Confirmar Cita

Para finalizar un turno solo se debe seleccionar la opción finalizar de la tabla de turnos y a continuación ingresar la asistencia y de manera opcional una observación, luego dar por terminado el turno.



Formulario de finalización de turno con los siguientes campos:

- Turno N°: 0036
- Fecha agenda: 2016-12-02
- Estudiante: Gabriela
- Paciente: Oscar
- Fecha cita: 2016-12-05
- Hora inicio: 10:30 AM (Rango: Entre las 7:00 - 18:50 horas)
- Hora fin: 11:30 AM (Rango: Entre las 7:00 - 18:50 horas)
- Cubículo: 1
- *Asistencia: -- Opción --
- *Observación: (Campo de texto libre)
- Botón: Dar por terminado

Figura 81. Interfaz Finalizar Cita

Para anular una cita médica “pendiente” se debe seleccionar la opción “Anular” en la tabla de turnos pendientes y a continuación confirmar la eliminación del mismo.

Lista turnos finalizados

En esta interfaz se muestra la lista de todos los turnos que ya han sido tratados, e ella se muestra la información básica del mismo como los nombres y fechas de gestión y su estado actual “FINALIZADO”, como se muestra en la siguiente imagen.

LISTA DE TURNOS REALIZADOS						
Mostrar	10	resultados	Buscar:			
Nº Turno	Fecha agenda	Nombre Estudiante	Paciente	Fecha cita	Hora	ESTADO
0031	2016-12-01	Ledesma Patricio	Martinez Carlos	2016-12-30	08:30:00/08:30:00	FINALIZADO
0027	2016-11-28	Sampertegui Gabriela	Sánchez Oscar	2016-11-30	10:30:00/10:30:00	FINALIZADO
Mostrando 1 a 2 de 2 resultados						Anterior 1 Siguiente

Figura 82. Lista de turnos finalizados

6. Horarios

En esta interfaz se podrá visualizar los horarios de atención de los docentes dentro de la clínica, y además se podrá asignar un docente a cualquiera de las horas de atención de la clínica.

HORARIO DOCENTES					
Hora/Día	Lunes	Martes	Miércoles	Jueves	Viernes
7:00/8:00	Zeas Elizabeth	Zeas Elizabeth	Zeas Elizabeth	Rosero Edgar	Rosero Edgar
8:05/9:05	Rosero Edgar	Izurieta Carlos	Izurieta Carlos	Izurieta Carlos	Padilla Andres
9:10/10:10	Zeas Elizabeth	(No asignado)	(No asignado)	Padilla Andres	(No asignado)
10:15/11:15	Rosero Edgar	(No asignado)	(No asignado)	(No asignado)	(No asignado)
11:20/12:20	(No asignado)	(No asignado)	(No asignado)	(No asignado)	(No asignado)
12:25/13:25	(No asignado)	(No asignado)	(No asignado)	(No asignado)	(No asignado)
13:30/14:35	(No asignado)	(No asignado)	(No asignado)	(No asignado)	(No asignado)
14:35/15:35	(No asignado)	(No asignado)	(No asignado)	(No asignado)	(No asignado)
15:40/16:40	Padilla Andres	(No asignado)	(No asignado)	(No asignado)	(No asignado)
16:45/17:45	(No asignado)	(No asignado)	(No asignado)	(No asignado)	(No asignado)

Botón para la asignación de un docente a una hora específica

Figura 83. Interfaz Visualizar Horarios Docente

Para asignar un docente a una hora específica solo se debe seleccionar el nombre del docente y la hora y día como se muestra a continuación:

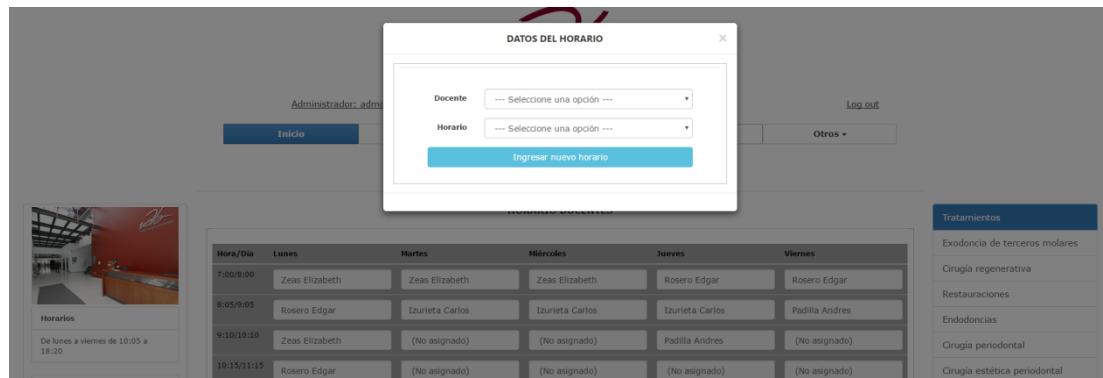


Figura 84. Interfaz Ingresar Horarios

En la siguiente opción se debe seleccionar el nombre del docente y la hora y día para el horario que se desea ingresar y a continuación dar click en la opción “Ingresar nuevo horario” para que el horario sea agregado, posteriormente se mostrará el horario de docentes actualizado.

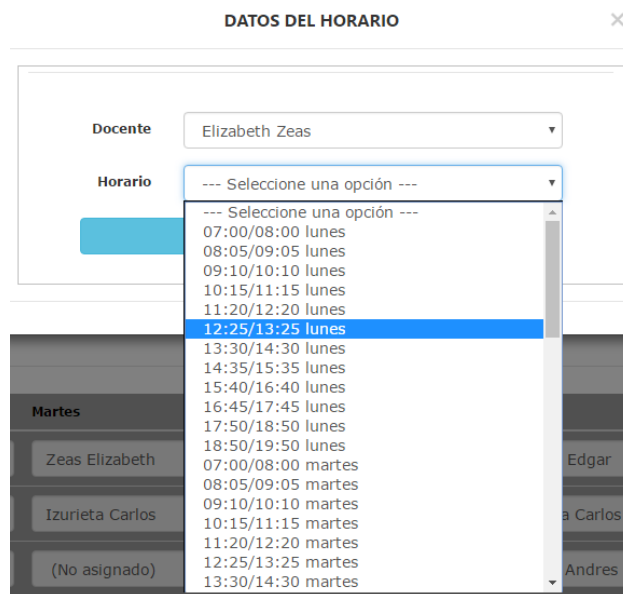


Figura 85. Interfaz para asignar horarios al docente

