



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE
LA INFORMACIÓN DE UN CENTRO ODONTOLÓGICO USANDO
EL FRAMEWORK ZEND.**

TRABAJO DE TITULACIÓN: PROYECTO TÉCNICO
Para optar al Grado Académico de:
INGENIERA EN SISTEMAS INFORMÁTICOS

AUTORA: EVELYN CAROLINA VALENCIA ESPINOZA
TUTOR: DR. DANILO PÁSTOR RAMÍREZ

Riobamba – Ecuador

2018

©2018, Evelyn Carolina Valencia Espinoza

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Evelyn Carolina Valencia Espinoza soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados.

Evelyn Carolina Valencia Espinoza

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal del Trabajo de Titulación certifica que: El trabajo de Titulación: “DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE UN CENTRO ODONTOLÓGICO USANDO EL FRAMEWORK ZEND”, de responsabilidad de la señorita Evelyn Carolina Valencia Espinoza, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

NOMBRES	FIRMAS	FECHA
Dr. Julio Santillán Castillo VICEDECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
M.Sc. Patricio Moreno Costales DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS	_____	_____
Dr. Danilo Pástor Ramírez DIRECTORA DEL TRABAJO DE TITULACIÓN	_____	_____
Dra. Leticia Vaca Cárdenas MIEMBRO DEL TRIBUNAL DEL TRABAJO DE TITULACIÓN	_____	_____

AGRADECIMIENTO

Agradezco infinitamente a Dios por brindarme la oportunidad de vivir para culminar una etapa importante de mi vida, a mis padres por brindarme su apoyo y haberme depositado toda su confianza en mí para cumplir mi meta tan anhelada.

Quiero extender mi agradecimiento a todos los profesores por los conocimientos impartidos para convertirme en una profesional y la cual está plasmado en el presente trabajo y con el mismo quiero expresar mi agradecimiento a la “ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO” por darme la oportunidad de forma parte de esta prestigiosa institución.

Al Ing. Danilo Pastor y a la Dra. Leticia Vaca presidente y miembro de mi trabajo de titulación quienes con su paciencia, apoyo y sabios conocimientos me ayudaron en la realización de este proyecto. Y de manera especial al Centro Odontológico del Dr. Frank Feijoo por permitirme realizar el sistema en su empresa.

Evelyn

DEDICATORIA

A mis padres Fausto Valencia y Lastenia Espinoza por el apoyo y sacrificio diario que me brindaron para seguir adelante. A mi hermano Andy Valencia y mi querida sobrina Ashly Valencia, ustedes son mi principal inspiración y de superación diaria para darles el mejor ejemplo cuando uno se propone una meta lo debe cumplir hasta alcanzarlo y superar obstáculos que la vida se presente.

Evelyn

TABLA DE CONTENIDO

CERTIFICACIÓN	iii
AGRADECIMIENTO	v
DEDICATORIA	vi
INDICE DE TABLAS	x
ÍNDICE DE FIGURAS	xi
ÍNDICE DE GRÁFICOS	xiv
LISTADO DE ANEXOS	xv
RESUMEN	xvii
SUMMARY	xviii
INTRODUCCIÓN	1
CAPÍTULO I	
1. MARCO TEÓRICO	7
1.1. Herramientas para el desarrollo del software	7
1.1.1. Lenguaje de programación PHP	7
1.1.2. Funcionamiento de PHP	8
1.1.3. Características de PHP.....	8
1.1.4. Ventajas	9
1.2. Patrón de Diseño MVC.....	9
1.3. Framework de desarrollo	10
1.3.1. Ventajas	10
1.3.2. Desventajas.....	10
1.3.3. Tipos de Framework PHP.....	11
1.4. Zend framework	11
1.4.1. Características del framework Zend	12
1.4.2. Requerimientos	13
1.4.3. Estructura de una aplicación en ZF	14
1.5. Composer.....	15
1.6. Metodologías Ágiles.....	16
1.6.1. Metodología de desarrollo Scrum.....	16
1.6.2. ¿Por qué usar Scrum?	16
1.6.3. Beneficios	17
1.6.4. Ciclo de vida Scrum	17

1.7.	PostgreSQL.....	19
1.7.1.	Ventajas	19
1.8.	Bootstrap.....	19
1.8.1.	Características.....	20
1.8.2.	Ventajas	20
1.8.3.	Desventajas.....	21
1.8.4.	Estructura del framework	22
1.9.	CSS	23
1.9.1.	Ventajas	23
1.9.2.	Desventajas.....	23
1.10.	jQuery	23
1.10.1.	Ventajas	24
CAPÍTULO II		
2.	MARCO METODOLÓGICO	25
2.1.	Tipo de Estudio.....	25
2.2.	Métodos	25
2.2.1.	Revisión de Documentos	25
2.2.2.	Método de Observación.....	26
2.2.3.	Método Sintético.....	26
2.2.4.	Práctico	26
2.2.5.	Metodología Scrum	26
2.3.	Terminología metodología Scrum	26
2.4.	Técnicas	27
2.4.1.	Entrevista.....	27
2.4.2.	Encuesta.....	27
2.5.	Desarrollo del proyecto mediante Scrum.....	28
2.5.1.	Personas y roles del proyecto	28
2.5.2.	Tipo de Usuario y Roles del proyecto	28
2.5.3.	Planificación	29
2.5.4.	Requerimientos.....	29
2.4.3.1.	Historia de Usuario.....	31
2.4.3.2.	Plan de entrega.....	32
2.6.	Desarrollo	35
2.6.1.	Diseño de la Base de datos	35

2.6.2.	Diseño físico de la Base de Datos.....	37
2.6.3.	Arquitectura del Sistema.....	39
2.6.4.	Diccionario de Datos	39
2.6.5.	Instalación, configuración y codificación.....	40
2.6.6.	Reuniones y entregables	44
2.6.7.	Documentación.....	44
2.6.8.	Manual de Usuario.....	45
CAPÍTULO III		
3.	RESULTADOS Y DISCUSIÓN	46
3.1.	Calidad de Software.....	46
3.2.	Estándar ISO/IEC 9126	46
3.3.	Eficiencia.....	46
3.4.	Definición de Criterios	47
3.4.1.	Población	47
3.4.2.	Muestra	48
3.5.	Tiempo de Respuesta.....	48
3.5.1.	Comparación de Resultados	50
3.6.	Utilización de Recursos	51
3.7.	Resultados Obtenidos	71
CONCLUSIONES.....		76
RECOMENDACIONES.....		78
BIBLIOGRAFIA.....		79
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-2 Términos de Scrum.....	27
Tabla 2-2: Personas y Roles del Proyecto.....	28
Tabla 3-2: Tipo de Usuario y Roles del Proyecto.....	28
Tabla 4-2: Estimaciones del Sistema.....	29
Tabla 5-2: Product Backlog.....	29
Tabla 6-2: Modelo de Historia de Usuario.....	32
Tabla 7-2: Plan de Entrega.....	32
Tabla 8-2: Diccionario de datos Tabla hca_historia_clinica.....	40
Tabla 1-3: Variables e Indicadores para medir la eficiencia.....	47
Tabla 2-3: Tareas para el análisis.....	48
Tabla 3-3: Tiempo que realiza sus funciones el Centro Odontológico.....	49
Tabla 4-3: Tiempo en realizar los procesos utilizando la aplicación Web.....	50
Tabla 5-3: Comparación de Resultados.....	50
Tabla 6-3: Eficiencia en Utilización de Recursos.....	51
Tabla 7-3: Densidad del mensaje de uso entrada/salida.....	52
Tabla 8-3: Resultados de la métrica Densidad del mensaje I/O.....	61
Tabla 9-3: Métrica de uso de memoria RAM.....	62
Tabla 10-3: Resultados de la métrica utilización de memoria.....	66
Tabla 11-3: Descripción de la métrica uso de Procesador.....	66
Tabla 12-3: Resultados de la métrica utilización de procesador.....	70
Tabla 13-3: Indicador de evaluación, tiempo de respuesta.....	71
Tabla 14-3: Resultados obtenidos tiempo de respuesta.....	71
Tabla 15-3: Indicador de evaluación.....	72
Tabla 16-3: Resultados de la métrica Densidad del mensaje de uso I/O.....	72
Tabla 17-3: Indicador de evaluación.....	73
Tabla 18-3: Resultados obtenidos memoria RAM.....	73
Tabla 19-3: Indicadores de evaluación.....	74
Tabla 20-3: Resultados obtenidos de la métrica utilización de procesador.....	74
Tabla 21-3: Resumen de resultados obtenidos.....	75

ÍNDICE DE FIGURAS

Figura 1-1: Logotipo PHP	7
Figura 2-1: Esquema del funcionamiento de PHP	8
Figura 3-1: Patrón de Diseño Modelo-Vista-Controlador	9
Figura 4-1: Componentes de Zend Framework PHP	13
Figura 5-1: Estructura de los directorios framework Zend	15
Figura 6-1: Logo de composer.....	15
Figura 7-1: Esquema del Ciclo de vida Scrum	18
Figura 8-1: Los tipos de Encabezados	20
Figura 9-1: Estructura de Bootstrap.....	22
Figura 1-2: Diagrama Entidad-Relación.....	36
Figura 2-2: Diseño Físico de la Base de Datos.....	38
Figura 3-2: Arquitectura del Sistema	39
Figura 4-2: Descargar el programa Composer.....	40
Figura 5-2: Descargar el esqueleto del Framework Zend	41
Figura 6-2: Página de Bienvenida del Framework Zend	41
Figura 7-2: Estructura de las carpetas del Sistema	42
Figura 8-2: Estructura MVC módulo Signos Vitales.....	42
Figura 9-2: Controlador de Signos vitales	43
Figura 10-2: Modelo del módulo Signos vitales.....	43
Figura 11-2: Vista del Módulo de Signos vitales.....	44
Figura 1-3: Error controlado, la cédula sea valida	52
Figura 2-3: Error controlado, la cédula no esté duplicada	53
Figura 3-3: Error controlado, el correo no esté duplicado	53
Figura 4-3: Error controlado, no ingresar datos vacíos.....	54
Figura 5-3: Error controlado enviar datos vacíos	54
Figura 6-3: Error controlado, cuando no se encuentra los datos.....	55
Figura 7-3: Error controlado, enviar datos vacíos	55
Figura 8-3: Error controlado, enviar datos vacíos	56
Figura 9-3: Error controlado, enviar datos vacíos	56
Figura 10-3: Error controlado, enviar datos vacíos.....	57
Figura 11-3: Error controlado, no encontrar datos de búsqueda.....	57

Figura 12-3: Error controlado, enviar datos vacíos	58
Figura 13-3: Error controlado, enviar datos vacíos	58
Figura 14-3: Error controlado, enviar datos vacíos	59
Figura 15-3: Error controlado, enviar datos vacíos	59
Figura 16-3: Error controlado, enviar datos vacíos	59
Figura 17-3: Error controlado, enviar datos vacíos	60
Figura 18-3: Error controlado, establecer permiso para el reporte	60
Figura 19-3: Error controlado, establecer permisos para el reporte.....	61
Figura 20-3: Utilización de memoria, registrar paciente	62
Figura 21-3: Utilización de memoria, modificar paciente	62
Figura 22-3: Utilización de memoria, buscar paciente	62
Figura 23-3: Utilización de memoria, ingresar tratamiento.....	63
Figura 24-3: Utilización de memoria, modificar tratamiento	63
Figura 25-3: Utilización de memoria, buscar tratamiento	63
Figura 26-3: Utilización de memoria, registrar tratamiento	63
Figura 27-3: Utilización de memoria, modificar tratamiento detalle.....	64
Figura 28-3: Utilización de memoria, registrar diagnóstico	64
Figura 29-3: Utilización de memoria, modificar diagnóstico	64
Figura 30-3: Utilización de memoria, registrar signos vitales	64
Figura 31-3: Utilización de memoria, modificar signos vitales.....	65
Figura 32-3: Utilización de memoria, ingresar odontograma	65
Figura 33-3: Utilización de memoria, modificar odontograma	65
Figura 34-3: Utilización de memoria, reporte de pacientes	65
Figura 35-3: Utilización de memoria, reporte de materiales desbastados	65
Figura 36-3: Utilización de procesador, registrar paciente	67
Figura 37-3: Utilización de procesador, modificar paciente.....	67
Figura 38-3: Utilización de procesador, buscar paciente.....	67
Figura 39-3: Utilización de procesador, ingresar paciente	67
Figura 40-3: Utilización de procesador, modificar tratamiento.....	68
Figura 41-3: Utilización de procesador, buscar tratamiento	68
Figura 42-3: Utilización de procesador, registrar tratamiento detalle	68
Figura 43-3: Utilización de procesador, modificar tratamiento detalle	68
Figura 44-3: Utilización de procesador, registrar diagnóstico.....	68
Figura 45-3: Utilización de procesador, modificar diagnóstico.....	69

Figura 46-3: Utilización de procesador, registrar signos vitales.....	69
Figura 47-3: Utilización de procesador, modificar signos vitales	69
Figura 48-3: Utilización de procesador, ingresar odontograma.....	69
Figura 49-3: Utilización de procesador, modificar odontograma	69
Figura 50-3: Utilización de procesador, reportes de pacientes	70
Figura 51-3: Utilización de procesador, reporte de materiales desbastecidos	70

ÍNDICE DE GRÁFICOS

Gráfico 1-3: Tiempo de Respuesta.....	51
Gráfico 2-3: Resultados de la eficiencia del sistema	75

LISTADO DE ANEXOS

Anexo A Diseño Físico de la Base de Datos

Anexo B. Historias de Usuario

Anexo C. Diccionario de Datos

Anexo D. Tiempo de Respuesta

Anexo E. Manual de Usuario

ÍNDICE DE ABREVIATURAS

SW	Software
HW	Hardware
BD	Base de datos
IO	Entrada/Salida
PHP	Lenguaje de Programación Interpretado (Hypertext Pre-Processor)
JSON	Notación de Objetos de JavaScript
CSS	Hojas de Estilo en Cascada
SQL	Lenguaje de consulta estructurada
PC	Computadora personal (personal computer)
XML	Lenguaje de Marcado Extensible
MVC	Modelo Vista Controlador
HTTP	Protocolo de transferencia de hipertexto
HTML	Lenguaje de marcas de hipertexto
IDE	Entorno de desarrollo integrado
PDF	Formato de documento portátil
ISO	Organización Internacional de Normalización

RESUMEN

En este trabajo de titulación se desarrolló una aplicación web para la gestión de la información de un centro odontológico, el cual automatizó todos los procesos que se realizan en el mismo desde la parte de los pacientes (usuarios) así como los tratamientos con sus respectivas historias clínicas (administrador), con el fin de facilitar el manejo de la información. Se recolectaron los datos con las técnicas como entrevista, observación, revisión de documentos para el desarrollo del sistema, también se hizo uso de las herramientas como el editor de texto Sublime Text en el cual se codificó en base al lenguaje de programación PHP, utilizando el Framework Zend para tener una codificación estructurada y ordenada, como Base de datos se usó PostgreSQL, con el servidor web Apache, para los reportes se empleó TCPDF. Cabe recalcar que se utilizó la metodología ágil Scrum para el desarrollo del sistema cumpliendo con cada uno de los requisitos, para validar la eficiencia se adoptó el estándar ISO/IEC 9126 del cual se tomó las subcaracterísticas: tiempo de respuesta y utilización de recursos, para la eficiencia interna se usó la métrica de utilización de recurso y se evaluó: Utilización I/O Densidad del mensaje, para la eficiencia externa se analizó el uso de memoria y procesador. Los resultados obtenidos de la evaluación cuentan con un 89% de eficiencia basados en las métricas ya establecidas de la norma, se analizó el tiempo de respuesta tanto en la gestión automatizada con un 5,5% como en la gestión tradicional con un 100%, esto quiere decir que se redujo el tiempo de respuesta un 94,5% en realizar los mismos procesos en el centro odontológico. Se recomienda para la utilización del framework Zend descargar el propio IDE que es el Zend Studio esto permite obtener el esqueleto o la estructura del Zend actualizado y también facilita el reconocimiento de la sintaxis o librerías.

Palabras clave: <TECNOLOGÍAS Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA DE SISTEMAS>, <APLICACIÓN WEB>, <ZEND FRAMEWORK>, <LENGUAJE PHP>, <HISTORIAS CLÍNICAS>, <TRATAMIENTOS>, <BASE DE DATOS (POSTGRESQL)>, <SERVIDOR (APACHE)>, <METODOLOGIA SCRUM>, <EFICIENCIA>, <NORMA ISO/IEC 9126>.

SUMMARY

In this present degree work, a web application was developed to manage the information of a dental center, which automated all the processes carried out in the same work from patients' side as well as the treatment with their respective medical records, in order to facilitate the handling of information. The data was collected with some techniques such as interview, observation, review of documents development system, also it was used tools such as the text editor "Sublime Text" which it was codified on PHP programming language, using the Framework Zend to have a structured and ordered coding, like PostgreSQL data base, with Apache web server, and reports TCPDF was applied. It should be highlighted that Agile Scrum methodology was applied in order to develop the system accomplishing all the requirements and validate the efficiency, the ISO/IEC 9126 standard was adopted where the sub-characteristics such as response time and usage of resources were taken, for internal efficiency the metric resource was evaluated: the application I/O Message density and for external efficiency the memory and processor usage was analyzed.

The results obtained from the evaluation were 89% of efficiency based on the efficiency established metrics standard, the response time was analyzed such as automated management with 5.5% and the traditional management with 100% and it means that the response time was reduced by 94.5% in carrying out the processes in the dental center. It is recommended that the application of Zend framework to the "IDE" itself, which represents the Zend Studio, this allows to obtain the skeleton or structure of the updated Zend and also facilitates the recognition of the syntax or libraries.

Keywords: <TECHNOLOGIES AND ENGINEERING SCIENCES>, <SYSTEM ENGINEERING>, <WEB APPLICATION>, <ZEND FRAMEWORK>, <PHP LANGUAGE>, <CLINICAL STORIES>, <TREATMENTS>, <DATABASE (POSTGRESQL)>, <SERVER (APACHE)>, <METHODOLOGY SCRUM>, <EFFICIENCY>, <ISO/IEC 9126 NORM>

INTRODUCCIÓN

Antecedentes

En los sistemas de software que son desarrollados con lenguajes puros se deben manejar los diferentes problemas que se pueden presentar como, por ejemplo: el cambio de requisitos, mientras está en desarrollo de una aplicación, lo cual es una situación común. Unos de los factores que hacen que el desarrollo de Software sea una tarea compleja, se produce cuando se trata de introducir nuevas tecnologías o cambios de ámbito de las aplicaciones esto pueden causar serios problemas de desarrollo ya que se tendría que comenzar desde cero. Es por eso que el desarrollo de aplicaciones reutilizables se ha vuelto una costumbre, tanto para las empresas como para los programadores independientes. Por otro lado, el diseño e implementación de Framework, ha ido creciendo de una forma acelerada ya que, al ser reutilizable, reducen drásticamente los costos asociados al mantenimiento y la complejidad de los proyectos de Software y también fomentan el uso de estándares (MORENO y ÁLVAREZ, 2005).

Existen muchos frameworks en el mercado basado en PHP los más populares son Laravel, Symfony, CodeIgniter y Zend los cuales proporcionan grandes beneficios para las empresas como para los desarrolladores, pero específicamente el Framework Zend está fortalecido por su soporte y su estabilidad en el mercado, sobre todo porque se ha convertido en uno de los más populares en el desarrollo de aplicaciones web empresariales en grandes corporaciones (Sánchez, 2014).

El centro Odontológico del Dr. Frank Feijoo se encuentra ubicado en la ciudad de Huaquillas provincia de El Oro, se dedica a realizar curaciones, tratamientos dentales y tiene el control de sus pacientes. El proceso normal para realizar un tratamiento o curación es: la auxiliar de odontología lleva un registro de los pacientes de forma tradicional, anota la información del paciente como el tratamiento que se va a realizar y las próximas citas del paciente con sus respectivos abonos. En cuanto al control de inventarios no llevaban un estricto control, esto generaba inconvenientes al momento de realizar un tratamiento ya que se dan cuenta de forma tardía de la falta de algún material. Además, se realizan informes del número de pacientes son atendidos en el centro Odontológico lo cual es tedioso realizarlo manualmente. La realización de todos estos procesos ha llevado al centro odontológico a presentar problemas de falta de eficiencia en el servicio que presta, uno de los motivos es el tiempo que requiere el cliente, cuando se va a realizar el tratamiento, ya que el cliente debe esperar su turno, de ahí le toman los datos y comienzan con el tratamiento, esto genera clientes

insatisfechos debido al tiempo de espera que llevan estos procesos y posteriormente ocasiona pérdidas de clientes para el centro odontológico.

Con la finalidad de resolver estos problemas y para optimizar el tiempo invertido en los procesos que conllevan a realizar el tratamiento o consultas, se ha considerado la necesidad de implementar un sistema web para automatizar estos procesos desde que el paciente reserva la cita hasta que termine con el tratamiento. Es por eso, que el uso del framework Zend brinda muchas posibilidades de tener un sistema potente, interactivo y sobre todo seguro.

Se ha revisado la literatura científica para el desarrollo del presente trabajo de titulación y se encontraron algunas tesis relacionadas. Una de estas es la de (Persad and Patricia, 2009) en la cual se desarrolló un software para la gestión de control que sirva a todas las oficinas odontológicas para mejorar el servicio que prestan a los usuarios finales, pero no se fundamentó en ningún Framework y utilizó el lenguaje Python y la base de datos SQLite. Otro trabajo es el de (Changotaxi and Patricio, 2014) en la que se plantea un sistema para registrar la información de las Historias Clínicas para una Clínica Odontológica para así mejorar la atención de sus pacientes desarrollado en .NET y MICROSOFT SQL para el almacenamiento de información.

También se revisó el trabajo de (Sánchez et al., 2015) que hace un análisis comparativo entre Zend y YUPP, después de hacer un conjunto de pruebas mediante prototipos determinaron que Zend tiene mayor productividad y se implementó un sitio web con el Framework y la base de datos MySQL.

En el presente trabajo de titulación se propuso utilizar el Framework Zend junto a PostgreSQL ya que posee beneficios como la integridad de los datos, la fiabilidad y lo más importante se está comparando con los gestores de base de datos de alto nivel como es ORACLE.

Por lo tanto, se plantea el desarrollo del sistema para gestionar la información de un centro odontológico usando el framework Zend, para así obtener un sistema de calidad en el que se mida de manera cuantitativa la eficiencia de la aplicación.

Formulación del Problema

¿Se puede desarrollar un Sistema para la automatización de información de un Centro Odontológico usando el Framework Zend para mejorar la eficiencia de los procesos?

Sistematización del Problema

¿Cómo es la situación actual de los procesos que realiza en el Centro Odontológico?

¿Cómo se beneficiaría el Centro Odontológico con la implementación del Sistema para su automatización?

¿Cómo se puede desarrollar e implementar de forma ágil y productiva un sistema para automatizar los procesos en el Centro Odontológico?

¿Cómo se podría validar la eficiencia de la aplicación?

Justificación

Justificación Teórica

En la actualidad se ha creado varios frameworks para ayudar a incrementar la productividad de aplicaciones web y así promover buenas prácticas de desarrollo como el uso de patrones. De allí que las ventajas de usar un framework a la hora de realizar un proyecto son diversas, entre otras, se disminuye el tiempo de creación de las aplicaciones, facilita el mantenimiento del código y hace uso de patrones. El patrón más utilizado por casi todos los frameworks es el conocido como Modelo Vista Controlador (MVC) (Pantoja y Pardo, 2016).

El patrón MVC puede implementarse sin la necesidad de utilizar un framework; no obstante, y a diferencia de aplicarlo de forma manual, el framework obliga al desarrollador a utilizarlo, creando de esta forma un código mucho más robusto. De esta manera, se evita el —código spaghetti, el cual se caracteriza por agregar funcionalidades en capas que no corresponden, lo que incide negativamente sobre el código fuente y el cumplimiento de las características de calidad del producto en relación con la mantenibilidad, especialmente sobre algunas de sus sub-características como: analizable, modificable y testeable (Pantoja y Pardo, 2016).

Es por ello que se propuso utilizar el framework Zend ya que es un framework híbrido tiene componentes usables de modo stand- alone, es fácil empezar a utilizar Zend al estilo PEAR (PHP Extension and Application Repository), sin cambiar el paradigma de la app existente. Se cuenta con la implementación completa del Modelo-Vista-Controlador y también tiene un amplio soporte de la comunidad ya que utiliza de PHPUnit para realizar Testing (calidad), existen más colaboradores libres, que trabajan para Zend (Iparraguirre Villanueva, 2011).

Tiene ventajas para el desarrollador como: Arquitectura adecuada, por ejemplo: Facilidad para realizar cambios (Módulos y Look&feel), velocidad de desarrollo, fácil depuración, también en la parte de rendimiento del sistema como, por ejemplo: bajo tiempo de ejecución y baja cantidad de memoria utilizada.

Por todos estos beneficios se ha planteado desarrollar el sistema utilizando el Framework Zend ya que cuenta con componentes que permiten reducir drásticamente el tiempo de desarrollo de una aplicación web, se puede adaptar a gran cantidad de tipo de base de datos, mejora la seguridad y otros beneficios más.

Justificación Aplicativa

En cuanto en la resolución del problema antes mencionado del Centro Odontológico se busca realizar el sistema propuesto, ya que el sistema facilitará a la gestión de la información respecto a los clientes ingresados en el sistema, permitirá al cliente reservar citas, visualizar los diagnósticos como sus tratamientos. En la parte de los empleados de la Odontología les permitirá el manejar eficiente a la información del cliente, los historiales clínicos y los tratamientos que se han realizado los clientes.

El dueño de la Odontología podrá revisar el número de clientes que ha ingresado mensualmente, para tener una visión clara cómo va el progreso que va teniendo en su local, también el control de los inventarios de sus materiales que utiliza para realizar los tratamientos o curaciones dentales.

Para mejorar los procesos mencionados anteriormente el sistema contará con una base de datos en la cual se podrá mantener la información perdurable en el tiempo de tanto empleados como clientes de la odontología.

El sistema contendrá los siguientes módulos:

Gestión de Usuarios. - el sistema permitirá en registro de usuarios tanto como el Administrador como sus pacientes.

Gestión de Reservaciones de Citas. - esta modulo le permitirá consultar la lista de reservaciones de citas que realizan los clientes para ser atendidos.

Gestión de Historial Clínico. - esto permitirá el ingreso del Historial clínico de los clientes.

Gestión de Reportes. - el administrador del sistema tendrá el acceso a todos los reportes que se genere en el sistema para la toma de decisiones por parte del área administrativa de la Odontología.

Gestión de tipos de Tratamientos. - este módulo permitirá registrar todos los tratamientos que se realicen en la odontología con opción en agregar en un futuro más tipos de tratamientos.

Respecto a las líneas de investigación de la EIS – ESPOCH –PNBV se encuentra: Definido como un Programa para el desarrollo de aplicaciones de software, hardware y telecomunicaciones.

En la línea de investigación de la EIS está clasificado como Proceso de Desarrollo de Software.

Objetivos

Objetivos Generales

Desarrollar una aplicación Web para la gestión de la información de un Centro Odontológico usando el Framework Zend.

Objetivos Específicos

- Estudiar las definiciones, elementos, características y el funcionamiento del Framework Zend
- Aplicar el patrón de diseño modelo-vista-controlador
- Desarrollar el sistema de información que permita gestionar las actividades del centro odontológico basado en el Framework Zend
- Validar la eficiencia de la aplicación web utilizando el Framework Zend.

CAPÍTULO I

1. MARCO TEÓRICO

El siguiente capítulo corresponde al marco teórico, que permite tener aspectos claros como las definiciones y características de las herramientas utilizadas para el desarrollo de la aplicación.

1.1. Herramientas para el desarrollo del software

Las herramientas que se usaron durante el desarrollo de este sistema se las explica a continuación.

1.1.1. *Lenguaje de programación PHP*

PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML (Cobo et al., 2005).

PHP es un acrónimo que significa PHP Hypertext Pre-processor (Lenguaje de programación Interpretado), lo creó Rasmus Lerdorf ya que inicialmente era PHP Tools, después con el pasar del tiempo se incorporaron dos personajes más y lo mejoraron. Es importante saber que fue el primer lenguaje de programación que incrustó código HTML mediante comandos.

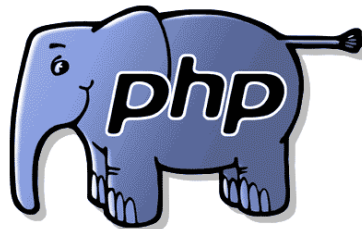


Figura 1-1: Logotipo PHP

Fuente: (Sysadmin Unix, 2012, p1)

1.1.2. Funcionamiento de PHP

En la figura 2-1, se muestra el esquema de funcionamiento de este lenguaje. Es decir, el cliente realiza una petición por medio de la PC utilizando el navegador web, se envía al servidor web, busca la petición procesa el código PHP, si es una consulta va a la base de datos PostgreSQL, y le envía los resultados que solicitó el cliente y se muestra en el navegador. Es importante resaltar que el código PHP no se muestra en el navegador porque es solo interpretado por el servidor web.

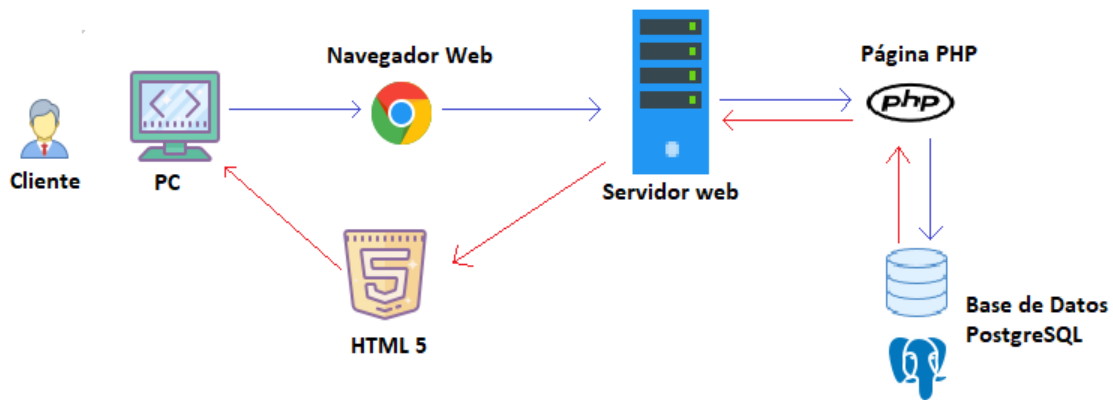


Figura 2-1: Esquema del funcionamiento de PHP

Realizado por: Evelyn Valencia, 2018

1.1.3. Características de PHP

PHP es conocido por sus características avanzadas, las fundamentales son:

- Es fácil de aprender
- Tiene una amplia documentación y tiene una enorme comunidad que programa en PHP.
- Casi el 85% de las páginas web están realizadas en PHP usando frameworks o gestor de contenidos, por lo tanto, es el más usado.
- Tiene más soportado por los servidores ya que es código abierto.
- Posee múltiples librerías que se puede descargar a través de composer, tiene varios frameworks que está basado en PHP, es por ello que tiene capacidad de expansión.
- Se pueden utilizar para crear imágenes y ficheros de lectura/escritura, así como para enviar mensajes de correo electrónico. Para proporcionar estos servicios, PHP sirve de bastantes protocolos, como HTTP, POP3, LDAP e IMAP (De la Cruz Villar, 2008).

1.1.4. Ventajas

Las ventajas que tiene PHP son las siguientes:

- Todo los frameworks PHP trabajan programación orientada a objetos como, por ejemplo: herencia, abstracción, encapsulamiento, etc.
- Utiliza el patrón MVC (Modelo-Vista-Controlador)
- Tiene independencia de SGBD (Sistemas de gestión de bases de datos), esto quiere decir, que se puede usar diferentes gestores de base de datos, instalando los drivers para que funcione, utiliza Query Builders esto permite construir una consulta mediante métodos y objetos o también ORMs que es el mapeo objeto racional.

1.2. Patrón de Diseño MVC (Modelo-Vista-Controlador)

El patrón de diseño MVC se utiliza como guía para las arquitecturas de aplicaciones ya que ofrecen una mayor interactividad con el usuario. Este patrón proporciona soluciones arquitectónicas al problema de separar los aspectos relacionados al código de la aplicación hacia a la interfaz de usuario, permitiendo a los diseñadores incorporar características deseables en dicha interfaz en una etapa temprana del diseño (Bustos Torres et al., 2008).

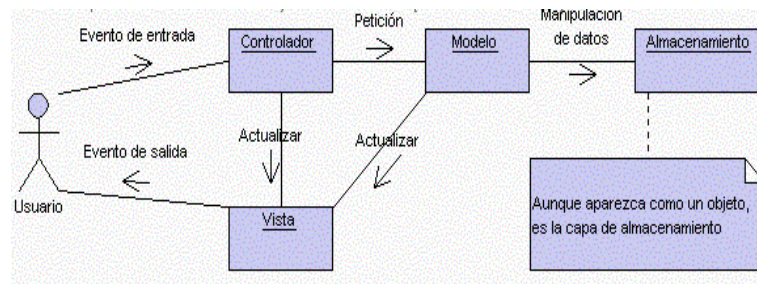


Figura 3-1: Patrón de Diseño Modelo-Vista-Controlador

Fuente: (Sánchez et al., 2015)

Las capas del patrón diseño MVC son las siguientes:

- **Modelo:** es donde se encapsula las funcionalidades y datos.
- **Vista:** es la parte de la interfaz de usuario.

- **Controlador:** es donde responde a eventos, esto quiere decir que son las acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

1.3. Framework de desarrollo

Framework es el esqueleto de una aplicación que debe ser adaptado por el programador para desarrollar una aplicación específica (Acosta et al., 2012).

Framework es un marco de trabajo cuenta con un conjunto de estándares o paradigma, buenas prácticas, algunas veces funciones costosas que ya han sido desarrolladas, que se puede reutilizar. El objetivo de un framework es: aumentar la productividad, fortalecer el equipo de trabajo e infundir buenas prácticas.

1.3.1. Ventajas

Las ventajas de utilizar un framework PHP para el desarrollo de software son las siguientes (Pablo Suarez, 2012):

- El framework se basa en los estándares, esto permite que sea compatible con la mayoría de los navegadores.
- Permite crear flujos de trabajo estandarizados, esto quiere decir, que todo el equipo de desarrolladores pueda entender el código sin ningún problema ya que se obtiene un código limpio y sencillo y fácil de mantener.
- Permite la expansión de librerías, helpers y plugins, ya que tiene una comunidad enorme de desarrolladores, esto permite que se pueda reutilizar código, como, por ejemplo: paginación, correos, tratamientos de imágenes, etc.
- La seguridad, es uno de los temas importantes, cuenta con la protección de sesiones, urls y formularios y también las consultas en la base de datos.

1.3.2. Desventajas

También se tiene desventajas al utilizar un framework PHP para el desarrollo de software, entre ellas están (Pablo Suarez, 2012):

- **Tiempo de aprendizaje:** programar con un framework al principio cuesta trabajo aprender, pero existen manuales o cursos que ayudan a aprender y entender sobre el framework que se está usando, también es importante dominar el lenguaje de programación PHP.
- **Desconocimiento del núcleo:** esto quiere decir, que no se sabe cómo es el funcionamiento interno del framework, pero es rara vez que se modifique el núcleo del framework.

1.3.3. *Tipos de Framework PHP*

Existen variedad de frameworks para PHP, pero los que más se destacan actualmente son:

- Zend Framework
- Symfony
- Laravel
- CakePHP
- CodeIgniter
- YiiFramework y muchos más.

1.4. **Zend framework**

Es un framework de código abierto para desarrollar aplicaciones y servicios web utilizando el lenguaje de programación PHP. Este framework utiliza el 100% código orientado a objeto y también usa las nuevas características PHP 5.3 (Montoya, 2013).

De acuerdo con el artículo del autor (Sierra et al., 2017) se describe que el punto importante del framework Zend es el estándar de codificación que se debe seguir en el desarrollo del proyecto. A su vez, cuenta con soporte para internacionalización y localización de aplicaciones como, por ejemplo: construir sitios multi idiomas, convertir formatos de fechas, moneda, etc. Según la región, esto es importante ya que la aplicación puede tener un enfoque global y llegar de la mejor manera a la mayor cantidad de personas. Posee adaptadores para los diferentes tipos de base de datos; brinda componentes para la autenticación y autorización de usuarios, envíos de mails, creación de web services, etc.

1.4.1. Características del framework Zend

Entre las principales características que se pueden mencionar tenemos:

- Basado en PHP
- Está orientado a objetos
- Usa el paradigma MVC (aunque no al 100%)
- Es open source
- Ofrece un gran rendimiento
- Ofrece una capa de abstracción a bases de datos fácil de usar
- Facilita el acceso a los servicios web de distintas compañías como Google o Microsoft
- Cuenta con una gran comunidad de desarrolladores que contribuyen con el mantenimiento y mejora del proyecto (Balu, 2011).

Entre sus metas se encuentran:

- Proveer un repositorio de componentes de alta calidad y que cuenten con soporte activo.
- Proveer un sistema completo para el desarrollo de aplicaciones web elaboradas en PHP5.
- Facilitar el aprendizaje en el uso del framework sin tener que aprender un nuevo lenguaje de programación.
- Organizar la colaboración de la comunidad para una programación avanzada en PHP5 (Rodríguez González, 2008).

Los componentes con los que cuenta este framework pueden ser agrupados en las siguientes categorías:

- **Infraestructura del núcleo del framework:** componentes requeridos por otros bloques del framework, como memoria caché, configuración del ambiente de trabajo, operación por la línea de comandos, registro de actividades y gestión de memoria, entre otros.
- **Autenticación y autorización de acceso:** responsables de la configuración de listas de control de acceso, autenticación de usuarios y manejo de sesiones. Base de datos: clases de acceso, modificación de tablas, obtención de datos mediante consultas SQL y lectura en formato XML.

- **Internacionalización y Localización:** para configurar la fecha y hora, la ubicación geográfica que tomará como base la aplicación web, las unidades de medida a usar y la posibilidad de traducir la información a otros idiomas.
- **Emails, formatos y búsquedas:** generación de archivos PDF, mensajes de correo electrónico en formato de texto y MIME (Multi-Purpose Internet Mail Extensions, Extensiones de correo Internet multipropósito)
- **Modelo-Vista-Controlador:** centra su atención en el desarrollo de controladores genéricos y específicos: de acciones, de atención de peticiones, de generación de URLs; además de una clase para las vistas.
- **Web y Servicios Web:** permite que la aplicación web pueda hacer uso de servicios web ofrecidos por aplicaciones externas y exponga servicios web propios.
- **Documentación:** intenta facilitar la lectura del código fuente, brindar ejemplos de su uso, promover equipos de traducción de la documentación y elaborar un tutorial del desarrollo de una aplicación web.

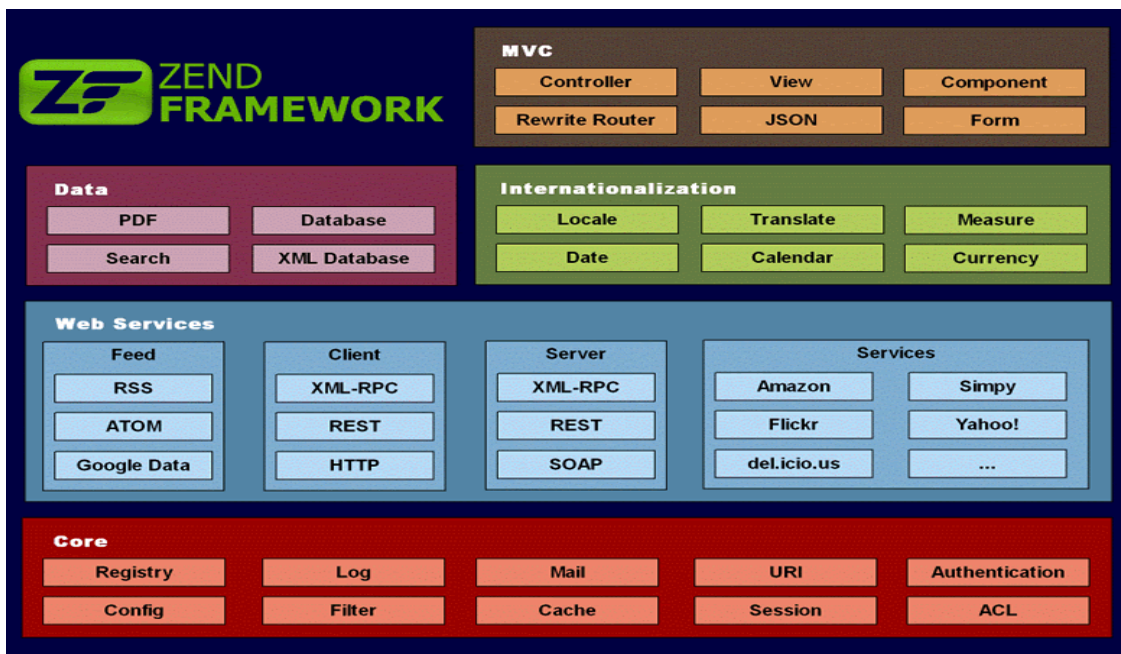


Figura 4-1: Componentes de Zend Framework PHP

Fuente: Juan Carlos Conde, 2012, p1

1.4.2. *Requerimientos*

Zend Framework tiene los siguientes requerimientos:

- PHP 5.1.4 (o superior)
- Un servidor de Web que tenga soporte con la funcionalidad `mod_rewrite`.

1.4.3. Estructura de una aplicación en ZF

Zend framework no requiere de una estructura específica de carpetas, pero en el manual recomienda una estructura común es que se va a utilizar. Para la utilización de esta estructura o del framework se debe tener configurado el servidor Apache (rolfipy, 2012).

Por tanto, los autores (Miranda Chinlli et al., 2015) explican los directorios relevantes que se debe conocer para la utilización del framework como son:

- **Directorio config:** En este archivo de configuración del proyecto, se cuenta con la configuración de la conexión de la base de datos, configuración a los módulos específicos, módulos de los layouts, etc.
- **Directorio public:** es la parte primordial o es la puerta de entrada para la invocación del proyecto. En este directorio cuenta con los archivos js, css, imágenes, librerías externas por parte del navegador.
- **Archivo composer:** está incluido en este archivo `composer.phar` y `composer.json`, estos dos archivos sirven para la configuración de las librerías y/o modules que necesita el proyecto.
- **Directorio module:** En este directorio contará con los módulos que se desarrollará en la aplicación, la cual estarán separados por carpetas y por defecto está incorporador el módulo `Application`.
- **Directorio vendor:** en este directorio está las librerías del framework Zend y otros módulos externos para los recursos que se usan en el proyecto.

En la Figura 5-1, muestra la estructura del framework Zend

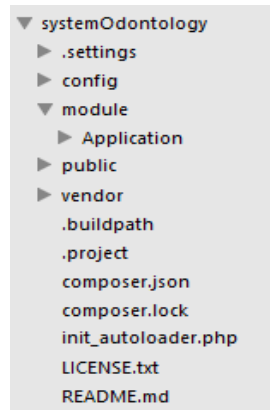


Figura 5-1: Estructura de los directorios framework Zend

Realizado por: Evelyn Valencia, 2018

1.5. Composer

Composer es un programa que permite gestionar las dependencias para PHP, nos facilita descargar componentes que estén bien estructurados para utilizarlos en el proyecto, mediante comandos a través de una consola.

Composer es un programa que permite instalar y desinstalar paquetes desde un repositorio, las clases que contiene funcionalidades básicas para resolver problemas como, por ejemplo: sesiones, administración de usuarios, abstracción de base de datos, etc. Ya que esto da la facilidad de reutilizarlo en el proyecto.



Dependency Manager for PHP

Latest: v1.6.5

Figura 6-1: Logo de composer

Fuente: (Adermann y Boggiano, 2018)

Composer es multiplataforma, es decir que se puede instalar en Windows, Linux y OSX. Es importante saber antes de la instalación se debe tener instalado el servidor que se va a usar.

Al momento ya tener instalado se genera dos ficheros que son: composer.json y composer.lock. y se crea por defecto la carpeta /vendor.

La carpeta ventor es donde se almacena los paquetes o archivos necesarios de las dependencias que son descargadas.

1.6. Metodologías Ágiles

Las metodologías ágiles son una evolución de las metodologías iterativas, ya que ambas se basan en objetivos a corto plazo. La metodología ágil propone una forma de trabajo flexible cuya planificación se actualiza continuamente. Esto contrasta con las metodologías tradicionales, las cuales siguen una planificación precisa desde el principio. Para esto el rol del cliente tiene mucha importancia. Ahora al no haber contratos cerrados, el cliente es uno más del equipo, quien indica cuál es su idea continuamente en el proceso, y a cada versión que se le muestra, éste va modificando, mejorando y puliendo los detalles más importantes para él, al ser software todo se puede modificar sin un costo demasiado elevado (García, 2015).

1.6.1. Metodología de desarrollo Scrum

Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán Sprints.(Trigás Gallego, 2012). Los Sprints son ciclos iterativos cortos que van desde una semana hasta un mes, periodo generalmente llamado iteración o sprint. El objetivo y beneficio en trabajar por sprint es entregar un producto listo para entregar. Por lo tanto tiene un enfoque que facilita la solución de los problemas, ya que ahí mismo se va creando la documentación y da mayor facilidad y rapidez al equipo en el proceso de codificación (Dimes, 2015).

1.6.2. ¿Por qué usar Scrum?

Las nuevas empresas de desarrollo de software, frecuentemente se enfrentan a dificultades para entregar prototipos a tiempo, sus productos no son entregados satisfactoriamente, por eso que aquellas compañías que han comenzado a usar Scrum han experimentado cambios significativos en la calidad de los productos y su entrega oportuna. Los programadores son más productivos ya que las tareas son divididas en partes más pequeñas y sea manejable. El software puede ser actualizado fácilmente para acomodarse a los cambios constantes del cliente o del usuario. También reciben su retroalimentación y las nuevas versiones sean liberadas y el software sea más estable (Dimes, 2015).

1.6.3. Beneficios

Los beneficios que ofrece la metodología Scrum son los siguientes (Alonso Baldonado, 2017):

- **Flexibilidad:** Scrum se adapta muy bien a cambios en los requisitos. El marco de trabajo está diseñado para adecuarse a las nuevas exigencias que implican proyectos complejos.
- **Reducción del Time o Market:** esta característica le hace ser muy atractivo en el mundo startup donde el tiempo de desarrollo es “tiempo perdido” y lo que prime es disponer de un producto viable para ser comercializado.
- **Buena calidad de software:** el trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad.
- **Productividad:** las reuniones cortas, la auto organización del equipo y la falta de burocracia consigue que el equipo de desarrollo emplee la mayor parte del tiempo en desarrollar.
- **Maximización de la inversión:** Se tiende a producir aquello que se necesita y por tanto se evita crear componentes inútiles para el negocio.
- **Predicciones de tiempos:** a través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil cuando podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.
- **Reducción de riesgos:** el hecho de desarrollar, en primer lugar, las funcionalidades de mayor valor y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada.

1.6.4. Ciclo de vida Scrum

En la figura 6-1, se muestra una visión global de un ciclo iterativo dentro de un proyecto de desarrollo utilizando la metodología Scrum. Los artefactos que se destaca es el Product Backlog ya que se definen los requerimientos del sistema tanto funcionales como no funcionales para realizar en cada Sprint por lo que es gestionado por el Scrum Master, la persona que lidera en el proyecto y al final de cada sprint se realiza Sprint Review para evaluar el avance del proyecto.

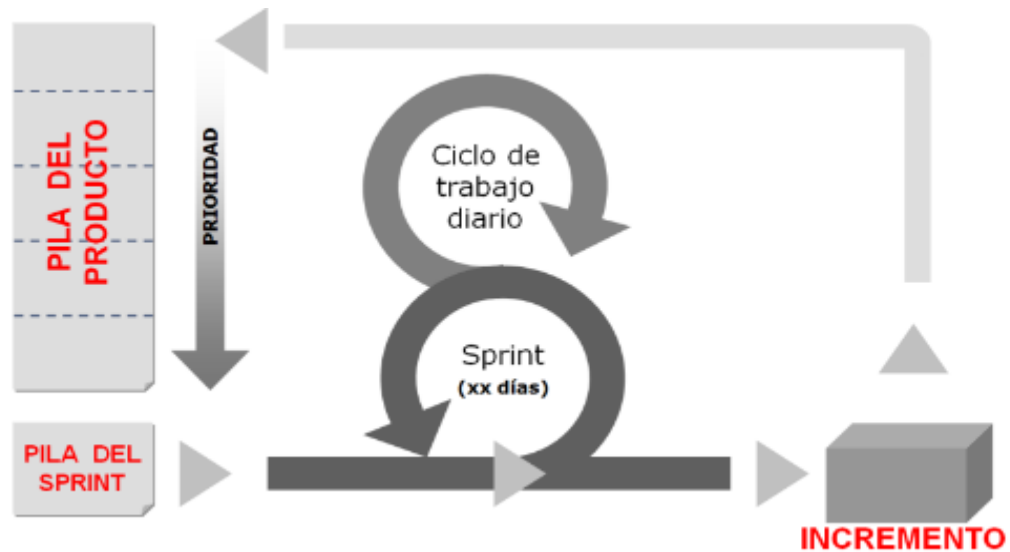


Figura 7-1: Esquema del Ciclo de vida Scrum

Fuente:(Menzinsky et al., 2016)

Los eventos de ciclo de vida Scrum se detallan así:

- **Sprint Planning (Planificación Sprint):** se realiza una reunión de trabajo para la planificación de Sprint, que está integrado el Product Owner para poder realizar las estimaciones en esfuerzo, el tiempo que deben desarrollar, es por eso que se compromete todo el equipo para cumplir con cada una de las tareas. Llevan un registro del tiempo que se desarrolló para obtener la velocidad del proyecto.
- **Scrum Diario o Daily (Reuniones diarias):** se debe realizar reuniones diarias que tienen una duración de 15 minutos, en la que cada miembro de equipo expresa como la va en el desarrollo, es importante saber que aquí no se resuelven los problemas, es decir, en caso de existir se convoca a una reunión para resolverlo. También da a conocer lo que se va a realizar en las próximas 24 horas y se actualiza el Product Backlog el tiempo o esfuerzo pendientes de las tareas y para poder monitorear el avance del sprint.
- **Sprint Review (Revisión de Sprint):** Se realiza la reunión de equipo conjuntamente con el cliente. Se muestra un demo del producto para que el cliente compruebe y valide para así satisfacer sus necesidades operativas o de negocio, en el caso de que haya algún problema con una tarea, regresa al Product backlog. Si todo va bien con el desarrollo se finaliza ese Sprint.
- **Scrum retrospective (retrospectiva del Sprint):** es una reunión del final de cada ciclo de desarrollo en que normalmente participa solamente los miembros del equipo de

desarrollo, el Scrum Master y el Product Owner. El objetivo de esta reunión es proponer un plan de mejoras para aplicar en el próximo Sprint.

1.7. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales (BGEO, 2014).

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (BGEO, 2014).

1.7.1. Ventajas

Basándose a un estudio comparativo de los gestores de base de datos del autor (López Herrera, 2016), presenta las siguientes ventajas:

- Fácil de administrar
- Multiplataforma
- La sintaxis SQL es estándar y fácil de aprender
- Soporte empresarial disponible
- Capacidades de replicación de datos
- Licencia gratuita
- Se puede realizar consultas complejas, disparadores.

1.8. Bootstrap

Es un framework originalmente creado por Twitter, que permite crear interfaces Web con CSS (Cascading Style Sheets - Hojas de Estilo en Cascada) y JavaScript, cuya particularidad es la de adaptar la interfaz del Sitio Web al tamaño del dispositivo en que se visualice, como al tamaño de una PC, una Tableta u otro dispositivo (Ramos Torres, 2015).

Bootstrap es un framework CSS desarrollado inicialmente (en el año 2011) por Twitter que permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web (González González, 2016).

EXAMPLE	
h1. Bootstrap heading	Semibold 36px
h2. Bootstrap heading	Semibold 30px
h3. Bootstrap heading	Semibold 24px
h4. Bootstrap heading	Semibold 18px
h5. Bootstrap heading	Semibold 14px
h6. Bootstrap heading	Semibold 12px

```
<h1>h1. Bootstrap heading</h1>  
<h2>h2. Bootstrap heading</h2>  
<h3>h3. Bootstrap heading</h3>  
<h4>h4. Bootstrap heading</h4>  
<h5>h5. Bootstrap heading</h5>  
<h6>h6. Bootstrap heading</h6>
```

Figura 8-1: Los tipos de Encabezados

Fuente: <https://getbootstrap.com/docs/3.3/css/>

1.8.1. Características

Las características que se pueden encontrar en Bootstrap que ha integrado una increíble combinación entre CSS y LESS para el diseño y estilo que se le puede dar a la página web como resultado se obtendrá una interfaz elegante e interactiva, contiene JavaScript que se puede usar efectos de decoración o animación la cual se añade componentes, diseño responsive para que se adaptable a cualquier dispositivo y cuenta con una documentación extensa y detalla, en la cual se encontrara con ejemplos fáciles para aprender el uso de componentes y el diseño web con Bootstrap (González, 2014).

1.8.2. Ventajas

Las ventajas más destacables de Bootstrap con los siguientes, en la que menciona la autora (Teresa Alba, 2014):

1. **Cuenta con un mantenimiento y actualización realizados por Twitter:** Esto no quiere decir que esta herramienta sea perfecta, pero gran parte del trabajo interno ya está llevado a cabo por sus creadores.

2. **Ofrece un paquete de elementos web personalizables:** Con Bootstrap puedes diseñar una web jugando con sus elementos compuestos por diferentes combinaciones de HTML, CSS y Javascript, de manera que las piezas siempre encajan.
3. **Utiliza componentes vitales para los desarrolladores:** Como HTML5, CSS3, jQuery o GitHub, entre otros.
4. **Sus plantillas son de sencilla adaptación responsive:** Se desarrolló con la idea de facilitar el proceso de adaptación web a todo tipo de dispositivos.
5. **Incluye Grid system:** Muy útil para maquetar por columnas.
6. **Se integra con librerías JavaScript.**
7. **Usa Less:** Un lenguaje de las hojas de estilo CSS preparado para enriquecer los estilos de la web.
8. **Es una herramienta de uso ágil y sencillo:** Facilita enormemente el diseño de interfaces y además incluye por defecto una plantilla bastante optimizada.
9. **Contiene tutoriales:** Este framework facilita mucha documentación para resolver dudas, tanto a principiantes como a desarrolladores expertos.
10. **Cada vez ofrece más plugins:** Cada vez incluye más características gracias a la aparición de nuevos plugins de terceros.

1.8.3. Desventajas

En las desventajas que se puede encontrar en la utilización Bootstrap son las siguientes (Lessin Jorge, 2014):

- Es necesario adaptarse a su forma de trabajo, si bien su curva de aprendizaje es liviana, deberás comprender y familiarizarte con su estructura y nomenclatura.
- Debes adaptar tu diseño a un grid de 12 columnas, que se modifican según el dispositivo.
- Aquí empiezan los problemas, Bootstrap por defecto te trae anchos, márgenes y altos de línea, y realizar cambios específicos es por decir, un poco tedioso.
- Es complicado, cambiar de versión si has realizado modificaciones profundas sobre el core.
- Si necesitas añadir componentes que no existen, debes hacerlos tú mismo en CSS y cuidar de que mantenga coherencia con tu diseño y cuidando el responsive.
- Un problema que he visto a menudo en muchos Themes es que el responsive funciona bien, pero a nivel de diseño en ocasiones se le escapan ciertos detalles que le quitan estética al sitio.

Si eres un perfeccionista del diseño, ten por seguro que Bootstrap te va a dar algunos problemas.

1.8.4. Estructura del framework

El código fuente descargado está estructurado en tres directorios con una pequeña cantidad de ficheros fácilmente reutilizables e integrables para el proyecto. Concretamente nos encontraremos con los siguientes directorios (Alonso Vega, 2013):

- **CSS:** Esta carpeta contiene dos ficheros css más sus versiones minimizadas. Los ficheros son `bootstrap.css` y `bootstrap-responsive.css`. Estos ficheros se emplean para estimular los elementos de la web. La versión responsive incluye todos los componentes necesarios para incluirlos en el proyecto.
- **JS:** Esta carpeta incluye el fichero `bootstrap.js` además de su versión minimizada donde se encuentra todo el código JavaScript necesario para el correcto funcionamiento de los widgets de Bootstrap.
- **IMG:** Esta carpeta incluye los sprites empleados para emplear los iconos de Bootstrap cedidos por Glyphicons.



Figura 9-1: Estructura de Bootstrap

Fuente: <https://getbootstrap.com/docs/4.1/getting-started/contents/>

1.9. CSS

CSS es un lenguaje que trabaja junto HTML para promover estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes etc. Cabe recalcar en la actualidad han incorporado la nueva versión de CSS3 para los navegadores más populares (Juan Diego Gauchat, 2012).

1.9.1. Ventajas

Estas son las ventajas de usar CSS según el autor (Xalambri, 2014) :

- Código más organizado
- Proyectos más fáciles de mantenerlos
- Reutilización de código

1.9.2. Desventajas

Estas son algunas desventajas que tiene CSS según al autor (Xalambri, 2014) :

- Deben aprender a usarse
- Puede llevar a algunas malas prácticas, como, por ejemplo: no nombrar bien las clases, no agrupar bien las clases, aplicar estilos a los elementos HTML, etc.
- Repetir código usando Mixins

1.10. jQuery

jQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Hace cosas como el recorrido y manipulación de documentos HTML, manejo de eventos, animación, y Ajax mucho más simple con una API fácil de usar que funciona en una multitud de navegadores (jquery.org, 2018).

Es una librería JavaScript ligera, rápida y concisa que simplifica el tratamiento de documentos HTML, el manejo de eventos, la creación de animaciones y las interacciones vía Ajax, para agilizar el desarrollo de aplicaciones web y sus puntos fuertes son los siguientes (Sánchez Suárez, 2018):

- mejora la interacción entre JavaScript y HTML: esto es, el manejo del árbol DOM ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos) en el cliente

- Existen gran número de plugins que extienden su funcionalidad y cualquiera puede crear sus propios plugins,
- tiene un módulo de widgets que proporcionan componentes predefinidos y efectos visuales a la interfaz de usuario: jQuery UI, y
- es Cross browser, de modo que está testada en los siguientes navegadores: I.E. 6.0+, FF2+, Safari 2.0+, Opera 9.0+, Chrome.

1.10.1. Ventajas

Las ventajas para utilizar jQuery tenemos lo siguiente (Eugenio Duarte, 2013):

- jQuery es flexible y rápido para el desarrollo web
- Viene con licencia MIT y es Open Source
- Tiene una excelente comunidad de soporte
- Tiene Plugins
- Bugs son resueltos rápidamente
- Excelente integración con AJAX

CAPÍTULO II

2. MARCO METODOLÓGICO

El presente capítulo, muestra el desarrollo del sistema web, seleccionando la metodología de desarrollo ágil SCRUM es por ello que se hace necesario seguir con el ciclo de vida de software que cuenta con 3 fases que son: fase de planificación, desarrollo y pruebas. A continuación, se detalla cada una de las etapas para obtener un software de calidad.

2.1. Tipo de Estudio

Es una investigación aplicada, esto permite solucionar un problema utilizando herramientas tecnológicas y finalmente beneficia a las personas involucradas con su implementación.

Además, también se aplican metodologías ágiles de desarrollo de software, se utiliza los métodos sintético y analítico. Por otra parte, se usa las técnicas como la entrevista para obtener los datos y los requerimientos para el desarrollo del sistema y la encuesta para analizar la métrica de eficiencia usando la norma de estándar ISO/IEC 9126.

2.2. Métodos

Los métodos seleccionados para el desarrollo del proyecto se describen a continuación, además se presenta una justificación del método y como se emplea.

2.2.1. *Revisión de Documentos*

Este método se emplea para tener una información que sea útil para nuestro objeto de estudio, el cual permite extraer y recopilar la información relevante y necesaria que sea importante para resolver el problema de investigación.

2.2.2. Método de Observación

Este método permite observar de forma directa la manera como llevan los procesos dentro del centro odontológico, desde el momento de realizar el historial Clínico hasta que se procede hacerse el tratamiento por parte del cliente.

2.2.3. Método Sintético

Este método permite tener un conjunto sistemático de los elementos consecutivos y relacionados existentes para obtener una solución adecuada para sí tener un producto de software completo el manejo correcto de los diferentes procesos.

2.2.4. Práctico

Con este método se ejecuta el desarrollo específico de la aplicación web usando el framework más utilizado Zend y con la metodología Scrum que están establecidas para este proyecto.

2.2.5. Metodología Scrum

Se selecciona la metodología Scrum para realizar las entregas de las tareas y el software pueda ser actualizado fácilmente para acomodarse a los cambios constante de acuerdo a las necesidades del cliente o corregir errores que se pueden presentar en el desarrollo es por ello que es flexible y tiene una estrecha comunicación con el cliente.

2.3. Terminología metodología Scrum

En la metodología Scrum se cuenta con varios términos con base de funciones, artefactos y encuentros, continuación se muestre en la tabla 1-2 los términos con sus respectivos conceptos para mayor comprensión.

Tabla 1-2: Términos de Scrum

TERMINOS SCRUM	CONCEPTOS
Product Owner	Es el que representa a todos los involucrados con el producto final, es el responsable del Product Backlog, porque ahí es donde se priorizan los requerimientos, conoce el producto, es el responsable y es capaz de tomar decisiones.
Scrum Master	Es una persona que conoce totalmente de la metodología es por ello que es un guía, porque sabe ¿qué se debe hacer?, ¿cómo se debe hacer? y ¿cuándo se debe hacer?, es decir, cuando el grupo de trabajo son nuevos, el verifica que todos los eventos del Scrum se realicen, también realiza un reporte del avance del proyecto. Él es el soporte para el equipo como el propietario del producto.
Scrum Team	Es el grupo de trabajo en la cual cuenta con conocimientos técnicos seleccionados para desarrollar o ejecutar cada sprint.
Customer	El cliente también es parte del equipo, ya que participa en las tareas que involucra en el Product Backlog.
Product Backlog	Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto.
Development Team	Está conformado por 3-9 personas, no solo son desarrolladores sino cuenta con profesionales de todas las áreas, en scrum no hay jefe porque todos son un grupo de trabajo es por eso que son auto gestionados. Este grupo de trabajo están comprometidos para cumplir con los objetivos del proyecto.
Sprint	Conjunto de ciclos iterativos realizado por el grupo de trabajo de la cual se desarrolla o mejora una funcionalidad para producir nuevas versiones.
Sprint Planning	el equipo se reúne con el propietario del producto para elegir un conjunto de características del producto a realizarse durante un sprint

Realizado por: Evelyn Valencia, 2018

2.4. Técnicas

2.4.1. Entrevista

Esta técnica permitió la recolección de la información y de los requerimientos de los procesos que se realiza en la Odontología.

2.4.2. Encuesta

Esta técnica permitió realizar un cuestionario previamente elaborado para adquirir información sobre el tiempo que lleva a realizar las actividades del centro odontológico desde que el paciente se registra,

se llena los datos de la historia clínica como lo tratamientos del mismo. Dependiendo de las respuestas obtenidas determinar conclusiones en cuanto a la eficiencia de dichos procesos.

2.5. Desarrollo del proyecto mediante Scrum

Scrum es una metodología ágil y flexible para el desarrollo de Software, la cual se realiza entregas parciales y regulares del producto final, para los requerimientos del sistema son priorizadas por parte del cliente para así realizar las respectivas entregas.

2.5.1. Personas y roles del proyecto

Para el desarrollo de la aplicación Web” System Odontology”, se identifican dos tipos de usuarios, que cuenta con su propio rol dentro del sistema, a continuación, se detalla en la Tabla 2-2.

Tabla 2-2: Personas y Roles del Proyecto

Persona	Rol
Ing. Danilo Pastor	Scrum Master
Dr. Frank Feijoo	Product Owner
Evelyn Valencia	Development Team

Realizado por: Evelyn Valencia,2018

2.5.2. Tipo de Usuario y Roles del proyecto

Para los tipos de usuarios y roles del proyecto se realiza mediante las reuniones con los miembros del centro odontológico, se establece dos tipos de usuarios para el requerimiento, tal como se muestra en la tabla 3-2.

Tabla 3-2: Tipo de Usuario y Roles del Proyecto

Tipo de Usuario	Rol
Administrador	Es el usuario principal en la cual tiene acceso a todas las funcionalidades del sistema
Paciente	Es el usuario que podrá realizar citas y también puede observar su información personal.

Realizado por: Evelyn Valencia,2018

2.5.3. Planificación

De acuerdo con la metodología ágil Scrum, se realiza la planificación para poder cumplir con cada uno de los requerimientos del sistema, se obtiene un listado de los requerimientos tanto técnicos y de usuarios para elaborar los Sprint de acuerdo a la prioridad que tenga en el sistema. Por lo tanto, se realiza las estimaciones de cada una de las tareas utilizando el método de la talla de la camiseta o t-shirt, esto permite darle un valor a las estimaciones que son S, M, XM, y L como se presenta en la tabla 4-2.

Tabla 4-2: Estimaciones del Sistema

Talla	Puntos estimados	Horas de trabajo
S	8	8
M	16	16
XM	32	32
L	80	80

Realizado por: Evelyn Valencia,2018

Para asignarle a cada tarea, cabe recalcar que un punto estimado es igual a una hora de trabajo, un día de trabajo es de 8 horas realizado por una sola persona por consecuente una semana de trabajo es 5 días, la cual equivale 40 puntos estimados, dos semanas de trabajo son 10 días que equivalen a 80 puntos siendo el valor máximo ya que cada Sprint es de 2 semanas.

2.5.4. Requerimientos

Los requerimientos de la aplicación web se establecen y se priorizan conjuntamente con el dueño del centro odontológico mediante varias reuniones. Se usa la herramienta de Product Backlog en donde se detalla las estimaciones y las priorizaciones de los requerimientos del sistema, a continuación, se muestra en la Tabla 5-2.

Tabla 5-2: Product Backlog

ID	TAREA	PRIORIDAD	ESTIMACION
HT-01	Diseño e Implementación de la BD	ALTA	40
HT-02	Diseño Físico de la BD	MEDIA	16
HT-03	Conexión de la Base de datos	BAJA	8
HT-04	Diseño de la Arquitectura del sistema	BAJA	8
HT-05	Diseño de la Interfaz de Usuario	MEDIA	16

HT-06	Investigación del Framework Zend	ALTA	40
HU-01	Ingresar Usuario	MEDIA	16
HU-02	Modificar Usuario	MEDIA	16
HU-03	Eliminar Usuario	MEDIA	16
HU-04	Ingresar Catalogo Maestro	MEDIA	16
HU-05	Modificar Catalogo Maestro	MEDIA	16
HU-06	Eliminar Catalogo Maestro	MEDIA	8
HU-07	Ingresar Catalogo Detalle	MEDIA	16
HU-08	Modificar Catalogo Detalle	MEDIA	16
HU-09	Eliminar Catalogo Detalle	BAJA	8
HU-10	Ingresar Persona	MEDIA	16
HU-11	Modificar Persona	MEDIA	16
HU-12	Eliminar Persona	BAJA	8
HU-13	Ingresar Signos vitales	MEDIA	16
HU-14	Modificar Signos Vitales	MEDIA	16
HU-15	Eliminar Signos Vitales	BAJA	8
HU-16	Ingresar examen Estomatognático	MEDIA	16
HU-17	Modificar examen Estomatognático	MEDIA	16
HU-18	Eliminar Examen estomatognático	BAJA	8
HU-19	Ingresar Diagnostico	MEDIA	16
HU-20	Modificar Diagnostico	MEDIA	16
HU-21	Eliminar Diagnostico	BAJA	8
HU-22	Ingresar Diagnostico Terapéutico	MEDIA	16
HU-23	Modificar Diagnostico Terapéutico	MEDIA	16
HU-24	Eliminar Diagnostico Terapéutico	MEDIA	16
HU-25	Ingresar Tratamiento	MEDIA	16
HU-26	Modificar Tratamiento	MEDIA	16
HU-27	Eliminar Tratamiento	MEDIA	16
HU-28	Ingresar Tratamiento Detalle	ALTA	40
HU-29	Modificar Tratamiento Detalle	MEDIA	16
HU-30	Eliminar Tratamiento Detalle	BAJA	8
HU-31	Realizar la Autenticación	ALTA	40
HU-32	Ingresar Historia Clínica	MEDIA	16
HU-33	Modificar Historia Clínica	MEDIA	16
HU-34	Eliminar Historial Clínica	BAJA	8
HU-35	Ingresar Material	MEDIA	16
HU-36	Modificar Material	MEDIA	16
HU-37	Eliminar Material	BAJA	8
HU-38	Ingresar Indicador Enfermedad	MEDIA	16
HU-39	Modificar Indicador Enfermedad	MEDIA	16
HU-40	Eliminar Indicador Enfermedad	BAJA	8
HU-41	Ingresar indicador higiene oral	ALTA	40
HU-42	Modificar indicador higiene oral	MEDIA	16
HU-43	Eliminar indicador higiene oral	BAJA	8

HU-44	Ingresar Cita	BAJA	8
HU-45	Modificar Cita	BAJA	8
HU-46	Eliminar Cita	BAJA	8
HU-47	Ingresar Cita detalle	BAJA	8
HU-48	Modificar Cita Detalle	BAJA	8
HU-49	Eliminar Cita Detalle	BAJA	8
HU-50	Ingresar Odontograma patología	ALTA	40
HU-51	modificar Odontograma patología	ALTA	40
HU-52	Eliminar Odontograma patología	MEDIA	16
HU-53	Ingresar Motivo consulta	ALTA	40
HU-54	Modificar Motivo consulta	MEDIA	16
HU-55	Eliminar Motivo Consulta	BAJA	8
HU-56	Ingresar Antecedentes personales	ALTA	40
HU-57	Modificar Antecedentes personales	MEDIA	16
HU-58	Eliminar Antecedentes personales	BAJA	8
HU-59	Reporte de Materiales desbastecidos	MEDIA	16
HU-60	Reporte de Materiales	MEDIA	16
HU-61	Reporte de Pacientes	MEDIA	16
HU-62	Manual de Usuario	ALTA	40

Realizado por: Evelyn Valencia,2018

2.4.3.1. Historia de Usuario

Se realizan las historias de Usuario con sus respectivos criterios para la prueba de aceptación, deben estar bien diseñadas para asegurar que los requisitos del cliente estén claramente detallados. La cual está compuesta por:

- ID o Número: es el identificador único para cada elemento o tarea, el formato que se seguirá es **HU** (historia de Usuario) o **HT** (Historia Técnica) después puesto dos dígitos.
- Nombre: es el nombre de la tarea o del requerimiento que se va a realizar.
- Modificación de historia de Usuario: esto permite saber si la HU o HT fue modificada en la cual se ubica el número de modificación.
- Usuario: es el usuario la que va a acceder a la funcionalidad.
- Sprint Asignado: se coloca el número de Sprint al que pertenece la tarea.
- Prioridad de Negocio: se le asigna si es: (Alta, Media o Baja) de acuerdo a las necesidades del cliente.
- Riesgo en el desarrollo: se asigna si es: (Alta, Media o Baja) lo denominada de acuerdo que le dé al grupo de trabajo.
- Puntos estimados: es el tiempo que requiere para su realización.

- Puntos Reales: es el tiempo real que se tomó en realizar dicha tarea.
- Descripción: es lo que el usuario requiere del sistema y el beneficio que obtiene.
- Observación: se describe datos importantes para el desarrollo.
- En la parte reversa de la Historia de Usuario va las pruebas de aceptación: son las condiciones que debe cumplir la tarea para su buen funcionamiento y cada prueba de aceptación se debe documentar. Como se muestra en la Tabla un ejemplo de pruebas de aceptación.

En la Tabla 6-2 se puede observar un ejemplo de historia de usuario.

Tabla 6-2: Modelo de Historia de Usuario

HISTORIA DE USUARIO	
Número: HU-01	Nombre: Ingreso usuario
Modificación historia de usuario:	
Usuario: Paciente	Sprint Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 16
Riesgo en desarrollo: Alta	Puntos Reales: 16
Descripción: Como administrador me gustaría que se puedan registrar los datos del usuario para poder tener un registro con información real de los mismos.	
Observaciones:	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Si se registrar todos los campos del formulario y se procede a guardar, se mostrará un mensaje de éxito. • Si se registra datos de manera duplicada nos aparecerá un mensaje usuario ya existe. • Si los datos se ingresan de manera errónea como el correo nos mostrara un mensaje de error de correo para que siga el formato. • Si existen campos vacíos nos mostrara un mensaje de llene todos los campos. 	

Realizado por: Evelyn Valencia, 2018.

2.4.3.2. Plan de entrega

En esta sección se detalla cada uno de los Sprint, en la Metodología Scrum es la característica principal que marca la diferencia, debido a que cada Sprint se asigna historias de usuarios de acuerdo a la planificación realizada. A continuación, se detalla en la Tabla 7-2.

Tabla 7-2: Plan de Entrega

Nombre Tare	Duración	Comienzo	Final	Estimación
Fase de Desarrollo	140 días	lun 04/12/17	vie 15/06/18	
Sprint 1	10 días	lun 04/12/17	vie 15/12/17	Puntos

Diseño e implementación de la base de datos	5 días	lun 04/12/17	vie 08/12/17	80
Diseño Físico de la BD	2 días	lun 11/12/17	mar 12/12/17	
Diseño de la Arquitectura del Sistema	1 día	mié 13/12/17	mié 13/12/17	
Conexión con la Base de datos	2 días	jue 14/12/17	vie 15/12/17	
Sprint 2	10 días	lun 18/12/17	vie 29/12/17	
Investigación del Framework Zend	5 días	lun 18/12/17	vie 22/12/17	80
Diseño de la Interfaz de Usuario	2 días	lun 25/12/17	mar 26/12/17	
Ingresar Catalogo Maestro	2 días	mié 27/12/17	jue 28/12/17	
Eliminar Catalogo Maestro	1 día	vie 29/12/17	vie 29/12/17	
Sprint 3	10 días	lun 01/01/18	vie 12/01/18	
Ingresar Usuario	2 días	lun 01/01/18	mar 02/01/18	80
Modificar Usuario	2 días	mié 03/01/18	jue 04/01/18	
Eliminar Usuario	2 días	vie 05/01/18	lun 08/01/18	
Modificar Catalogo Maestro	2 días	Mar 09/01/18	mié 10/01/18	
Ingresar Catalogo Detalle	2 días	jue 11/01/18	vie 12/01/18	
Sprint 4	10 días	lun 15/01/18	vie 26/01/18	
Modificar Catalogo Detalle	2 días	lun 15/01/18	mar 16/01/18	80
Eliminar Catalogo Detalle	1 día	mié 17/01/18	mié 17/01/18	
Ingresar Persona	2 días	jue 18/01/18	vie 19/01/18	
Modificar Persona	2 días	lun 22/01/18	mar 23/01/18	
Eliminar Persona	1 día	mié 24/01/18	mié 24/01/18	
Ingresar Signos vitales	2 días	jue 25/01/18	vie 26/01/18	
Sprint 5	10 días	lun 29/01/18	vie 09/02/18	
Modificar Signos Vitales	2 días	lun 29/01/18	mar 30/01/18	80
Eliminar Signos Vitales	1 día	mié 31/01/18	mié 31/01/18	
Ingresar examen Estomatognático	2 días	jue 01/02/18	vie 02/02/18	
Modificar examen Estomatognático	2 días	lun 05/02/18	mar 06/02/18	
Eliminar Examen estomatognático	1 día	mié 07/02/18	mié 07/02/18	
Ingresar Diagnostico	2 días	jue 08/02/18	vie 09/02/18	
Sprint 6	10 días	lun 12/02/18	vie 23/02/18	
Modificar Diagnostico	2 días	lun 12/02/18	mar 13/02/18	80
Eliminar Diagnostico	1 día	mié 14/02/18	mié 14/02/18	
Ingresar Diagnostico Terapéutico	2 días	jue 15/02/18	vie 16/02/18	
Modificar Diagnostico Terapéutico	2 días	lun 19/02/18	mar 20/02/18	
Eliminar Diagnostico Terapéutico	1 día	mié 21/02/18	mié 21/02/18	
Ingresar Tratamiento	2 días	jue 22/02/18	vie 23/02/18	
Sprint 7	10 días	lun 26/02/18	vie 09/03/18	
Modificar Tratamiento	2 días	lun 26/02/18	mar 27/02/18	80
Eliminar Tratamiento	2 días	mié 28/02/18	jue 01/03/18	
Ingresar Tratamiento Detalle	5 días	vie 02/03/18	jue 08/03/18	
Eliminar Tratamiento Detalle	1 día	vie 09/03/18	vie 09/03/18	

Sprint 8	10 días	lun 12/03/18	vie 23/03/18	
Realizar la Autenticación	5 días	lun 12/03/18	vie 16/03/18	80
Modificar Tratamiento Detalle	2 días	lun 19/03/18	mar 20/03/18	
Ingresar Historia Clínica	2 días	mié 21/03/18	jue 22/03/18	
Eliminar Historial Clínica	1 día	vie 23/03/18	vie 23/03/18	
Sprint 9	10 días	lun 26/03/18	vie 06/04/18	
Modificar Historia Clínica	2 días	lun 26/03/18	mar 27/03/18	80
Ingresar Material	2 días	mié 28/03/18	jue 29/03/18	
Modificar Material	2 días	vie 30/03/18	lun 02/04/18	
Ingresar Indicador Enfermedad	2 días	mar 03/04/18	mié 04/04/18	
Modificar Indicador Enfermedad	2 días	jue 05/04/18	vie 06/04/18	
Sprint 10	10 días	lun 09/04/18	vie 20/04/18	
Eliminar Material	1 día	lun 09/04/18	lun 09/04/18	80
Eliminar Indicador Enfermedad	1 día	mar 10/04/18	mar 10/04/18	
Ingresar indicador higiene oral	5 días	mié 11/04/18	mar 17/04/18	
Modificar indicador higiene oral	2 días	mié 18/04/18	jue 19/04/18	
Eliminar indicador higiene oral	1 día	vie 20/04/18	vie 20/04/18	
Sprint 11	10 días	lun 23/04/18	vie 04/05/18	
Ingresar Cita	1 día	lun 23/04/18	lun 23/04/18	80
Modificar Cita	1 día	mar 24/04/18	mar 24/04/18	
Eliminar Cita	1 día	mié 25/04/18	mié 25/04/18	
Ingresar Cita detalle	2 días	jue 26/04/18	vie 27/04/18	
Modificar Cita Detalle	2 días	lun 30/04/18	mar 01/05/18	
Eliminar Cita Detalle	1 día	mié 02/05/18	mié 02/05/18	
Reporte de Materiales desbastecidos	2 días	jue 03/05/18	vie 04/05/18	
Sprint 12	10 días	lun 07/05/18	vie 18/05/18	
Ingresar Odontograma patología	5 días	lun 07/05/18	vie 11/05/18	80
modificar Odontograma patología	5 días	lun 14/05/18	vie 18/05/18	
Sprint 13	10 días	lun 21/05/18	vie 01/06/18	
Ingresar Motivo Consulta	2 días	lun 21/05/18	mar 22/05/18	80
Modificar Motivo Consulta	2 días	mié 23/05/18	jue 24/05/18	
Eliminar Motivo Consulta	1 día	vie 25/05/18	vie 25/05/18	
Ingresar Antecedentes Personales	2 días	lun 28/05/18	mar 29/05/18	
Modificar Antecedentes personales	2 días	mié 30/05/18	jue 31/05/18	
Eliminar Antecedentes personales	1 día	vie 01/06/18	vie 01/06/18	
Sprint 14	10 días	lun 04/06/18	vie 15/06/18	
Eliminar Odontograma patologia	1 día	lun 04/06/18	lun 04/06/18	80
Reporte de Materiales	2 días	mar 05/06/18	mié 06/06/18	
Reporte de Pacientes	2 días	jue 07/06/18	vie 08/06/18	
Manual de Usuario	5 días	lun 11/06/18	vie 15/06/18	
TOTAL				1120

Realizado por: Evelyn Valencia, 2018.

Se obtiene un total de 14 Sprints con fecha de inicio viernes 04 de diciembre de 2017 y fecha finalización el jueves 15 de junio de 2018 y un total de 1120 puntos de estimados o 1120 horas.

2.6. Desarrollo

2.6.1. Diseño de la Base de datos

El diseño de la Base de datos es uno de los principales componentes del sistema informático para garantizar su correcta implementación y funcionamiento, permite almacenar y obtener la información exacta y actualizada del centro odontológico. Se usa la herramienta Lucid chart 2018 para realizar el diseño de la base de datos mediante el diagrama Entidad- Relación (DER), la cual permite identificar las entidades del negocio con sus respectivos atributos y relaciones para así poder cumplir con los requerimientos obtenidos como se muestra en la Figura 1-2.

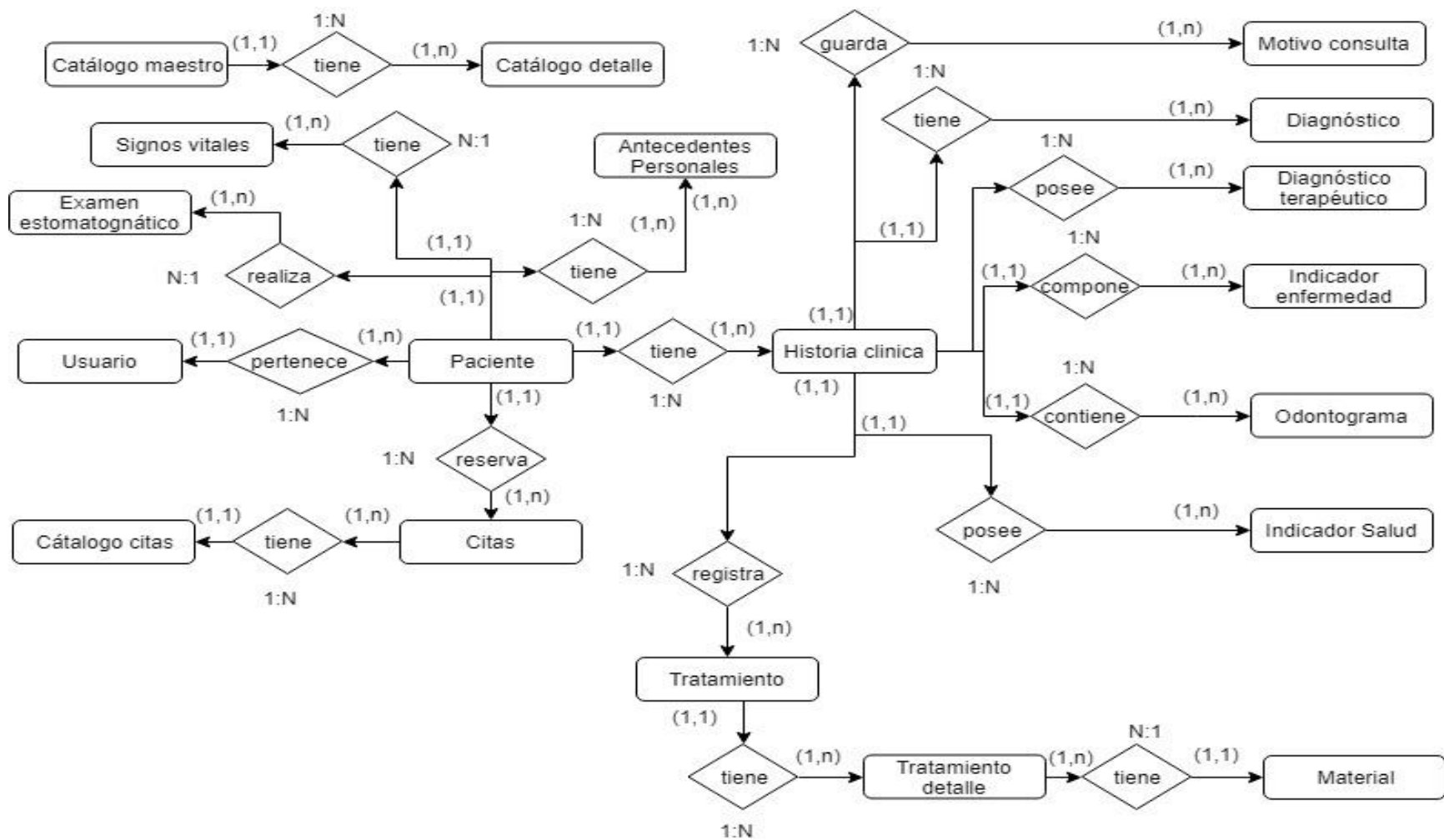


Figura 1-2: Diagrama Entidad-Relación

Realizado por: Evelyn Valencia, 2018

Se obtiene un número de 19 entidades las cuales son: catálogo maestro, catálogo detalle, catálogo citas, cita, usuario, paciente, examen estomatognático, antecedentes personales, signos vitales, historia clínica, indicador enfermedad, tratamiento, odontograma, tratamiento detalle, material, diagnóstico, indicador salud, diagnóstico terapéutico, motivo consulta.

2.6.2. Diseño físico de la Base de Datos

El diseño físico permite tener un mejor entendimiento como va a estar estructurado los requerimientos del negocio para poder realizar la implementación de la Base de datos se utiliza PostgreSQL por ser uno de los motores más potentes y para este diseño se usa la herramienta Power Designer ya que facilita la generación porque anteriormente se contó con el diagrama entidad- relación y para este diseño cuenta con un total de 19 tablas con sus respectivas relaciones y llaves foráneas como se muestra en la Figura 2-2 .

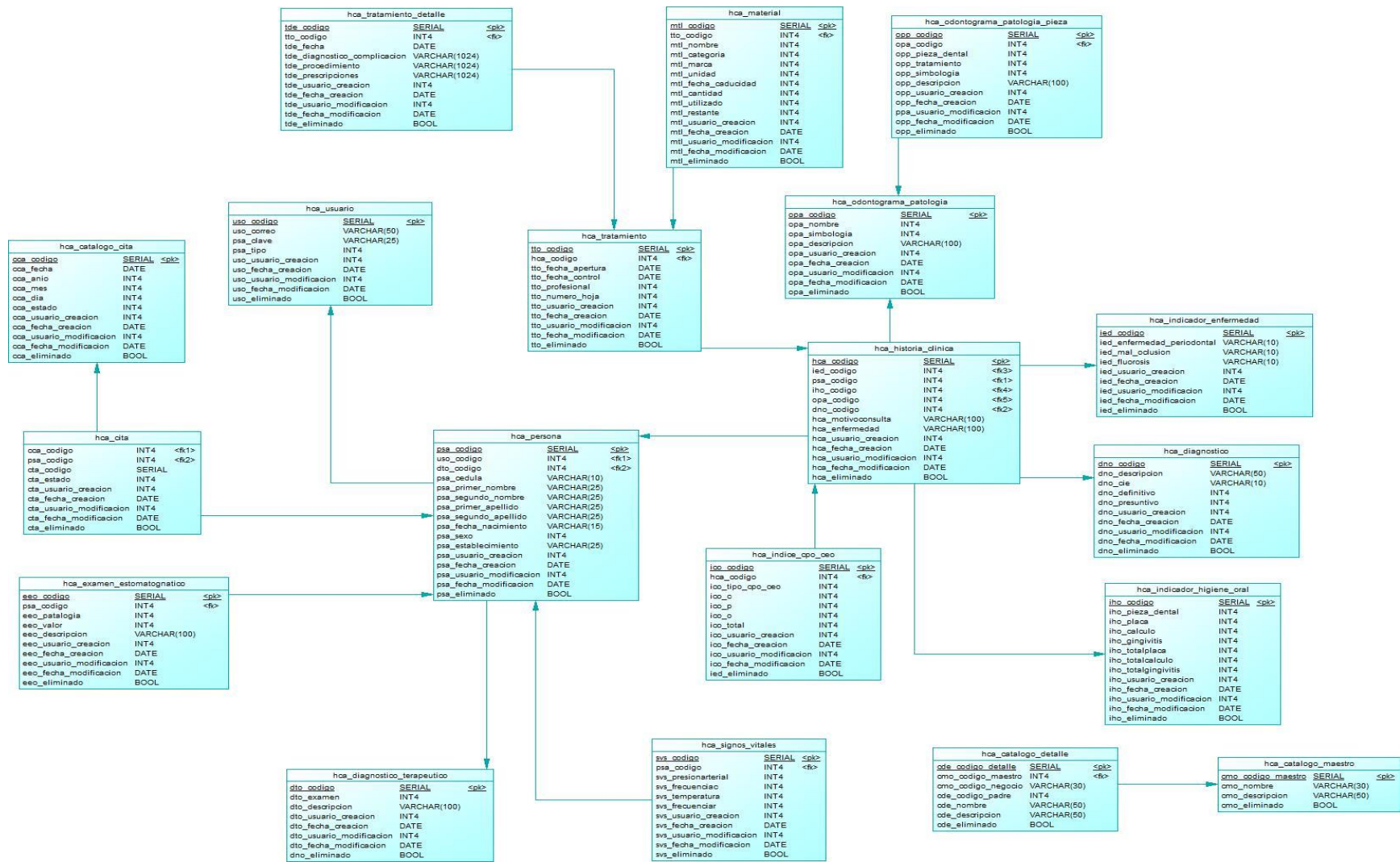


Figura 2-2: Diseño Físico de la Base de Datos

Realizado por: Evelyn Valencia, 2018

2.6.3. Arquitectura del Sistema

La arquitectura del Software es importante ya que permite mostrar cómo está estructurado el sistema y la forma en la que cada uno de los componentes trabaja entre sí.

La arquitectura del software es Cliente/Servidor, en la que se usa el patrón de diseño MVC, está compuesta por la parte de la vista donde se encuentra las interfaces de usuarios, Modelo es la parte del acceso de datos, permite consumir las funciones para acceder a la base de datos, Controlador es donde se aplica las reglas de negocio y el servidor de base de datos se encuentra el motor de base de datos PostgreSQL donde se va almacenar y se realiza las diferentes operaciones de cada uno de los requerimientos implicadas para el funcionamiento del sistema.

Como se muestra en la Figura 3-2.

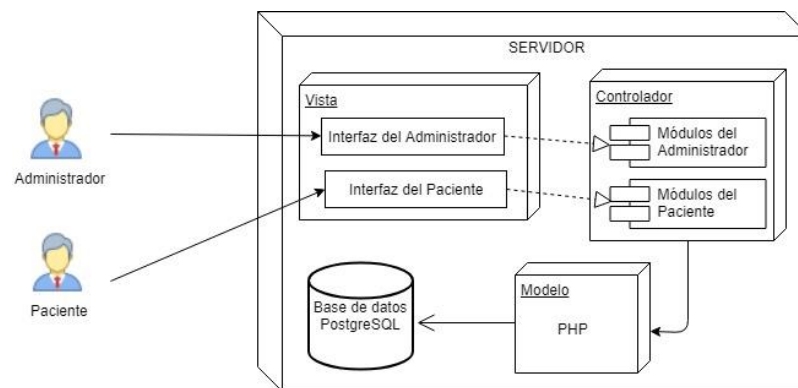


Figura 3-2: Arquitectura del Sistema

Realizado por: Evelyn Valencia, 2018

2.6.4. Diccionario de Datos

Un diccionario de datos es el conjunto de metadatos que describen las características lógicas y puntuales de los datos que se utilizarán en el sistema, también se detalla las entidades y sus atributos con su respectiva descripción.

A continuación, en la Tabla 8-2, se detalla el diccionario de datos de la Tabla hca_historia_clinica de la base de datos Historia_Clinica en el **ANEXO C** se detallará las demás tablas.

Tabla 8-2: Diccionario de datos Tabla hca_historia_clinica

Columna	Tipo	Nulo	Comentario
hca_codigo	Serial	No	Clave única. Identificador de la Historia_Clinica
hca_motivoconsulta	Varchar (100)	No	Motivo de la consulta del paciente
hca_enfermedad	Varchar (100)	No	Enfermedad que tiene el paciente
hca_usuario_creacion	Integer		Identificador del Usuario que lo creo
hca_fecha_creacion	Date		Fecha que se creo
hca_usuario_modificacion	Integer		Identificador del usuario que modifiko
hca_fecha_modificacion	Date		Fecha de modificación
hca_eliminado	Boolean	No	Identificador para saber si esta eliminado

Realizado por: Evelyn Valencia,2018

2.6.5. *Instalación, configuración y codificación*

Para la codificación se usa el editor de texto Sublime Text, Composer y PostgreSQL. Es importante mencionar que en el desarrollo se utiliza el framework Zend y el servidor apache para que funcione el código PHP.

Para la realización de los respectivos reportes se utiliza TCPDF es la librería libre en PHP para la creación de documentos pdf.

Instalación del Framework Zend

1. Instalar el programa Composer desde el link oficial de la página <https://getcomposer.org/download/>
2. Descargar el programa para Windows, como se muestra en la Figura 4-2.

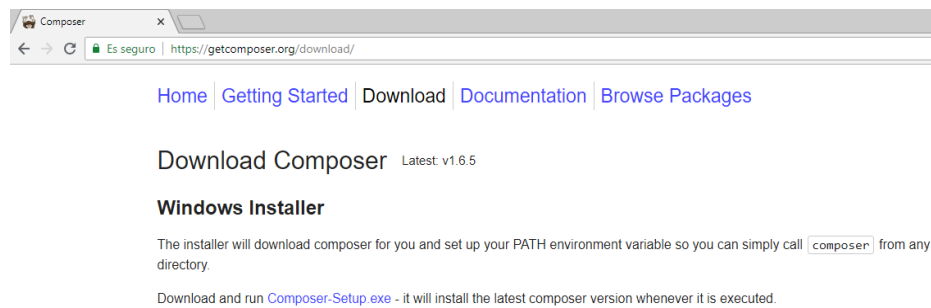
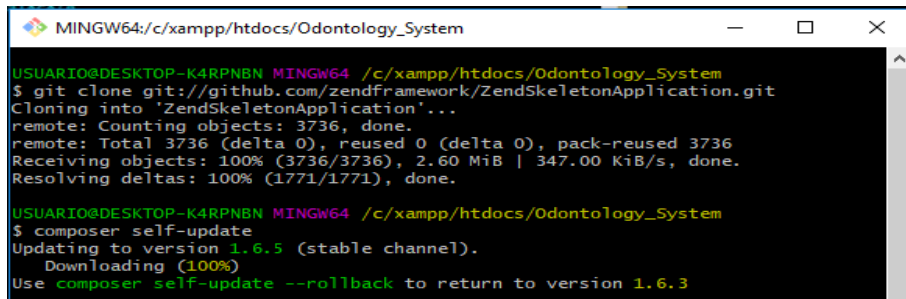


Figura 4-2: Descargar el programa Composer

Realizado por: Evelyn Valencia,2018

3. Se crea una carpeta del Xampp/htdocs y se pone el nombre del proyecto

4. Abrir la consola y poner los comandos para descargar el esqueleto del framework Zend, como se muestra en la Figura 5-2



```
MINGW64:/c:/xampp/htdocs/Odontology_System
USUARIO@DESKTOP-K4RPNBN MINGW64 /c:/xampp/htdocs/Odontology_System
$ git clone git://github.com/zendframework/ZendSkeletonApplication.git
Cloning into 'ZendSkeletonApplication'...
remote: Counting objects: 3736, done.
remote: Total 3736 (delta 0), reused 0 (delta 0), pack-reused 3736
Receiving objects: 100% (3736/3736), 2.60 MiB | 347.00 KiB/s, done.
Resolving deltas: 100% (1771/1771), done.
USUARIO@DESKTOP-K4RPNBN MINGW64 /c:/xampp/htdocs/Odontology_System
$ composer self-update
Updating to version 1.6.5 (stable channel).
  Downloading (100%)
Use composer self-update --rollback to return to version 1.6.3
```

Figura 5-2: Descargar el esqueleto del Framework Zend

Realizado por: Evelyn Valencia,2018

5. Ingresar al navegador y poner el nombre de la carpeta y debe mostrar la página de bienvenida del Framework Zend, como se muestra en la Figura 6-2, esto significa que se instaló correctamente.

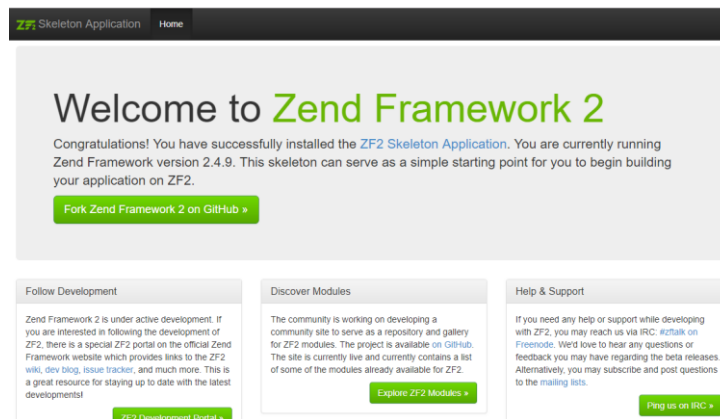


Figura 6-2: Página de Bienvenida del Framework Zend

Realizado por: Evelyn Valencia,2018

Esta es la estructura del sistema como se muestra en la Figura 7-2, los módulos que se desarrollan para cumplir con cada uno de los requisitos del sistema.

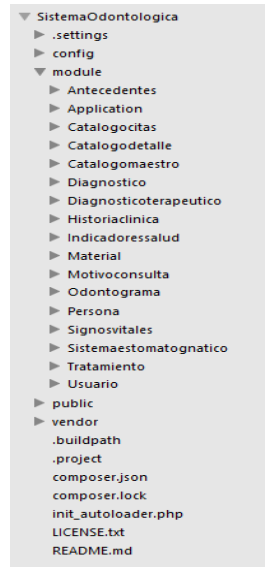


Figura 7-2: Estructura de las carpetas del Sistema

Realizado por: Evelyn Valencia,2018

A continuación, se muestra la estructura del Módulo Signos Vitales con su respectiva estructura Modelo, Vista y Controlador como se muestra en la Figura 8-2.

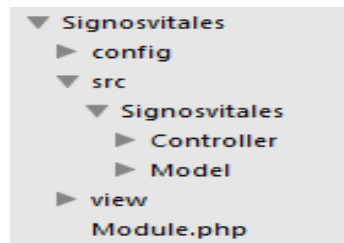


Figura 8-2: Estructura MVC módulo Signos Vitales

Realizado por: Evelyn Valencia,2018

En la carpeta Controller, contiene el archivo SignosvitalController, ahí cuenta cada una de las acciones que va a tener el módulo Signos vitales, como se muestra en la Figura 9-2.

```

10 namespace Signosvtales\Controller;
11
12 use Zend\Mvc\Controller\AbstractActionController;
13 use Zend\View\Model\ViewModel;
14 use Usuario\Form\Formulario;
15 use Persona\Model\Entity\Persona;
16 use Zend\Db\Adapter\Adapter;
17 use Signosvtales\Model\Entity\Signosvtales;
18 use Zend\Validator;
19 use Zend\I18n\Validator as I18nValidator;
20 //use Zend\Http\Header\Date;
21 use Zend\Stdlib;
22 use Zend\Crypt\Password\Bcrypt;
23 use Zend\Authentication\Adapter\DbTable as AuthAdapter;
24
25 //Componentes de autenticación
26 use Zend\Authentication\AuthenticationService;
27 use Zend\Authentication\Storage\Session as SessionStorage;
28 use Zend\Session\Container;
29
30 class SignosvtalesController extends AbstractActionController
31 {
32     public function __construct() {
33         //Creamos el servicio de autenticación en el constructor
34         $this->auth = new AuthenticationService();
35     }
36
37     public function gestionsignosvtalesAction()
38     {
39         $this->dbAdapter=$this->getServiceLocator()->get('Zend\Db\Adapter');
40         $u=new Signosvtales($this->dbAdapter);
41         $valores=array
42         (
43             "titulo" => "Gestion Signos vitales",
44             'datos' => $u->getSignosvtales()
45         );
46         return new ViewModel($valores);
47     }
48
49     public function registrarAction()
50     {
51         $identi=$this->auth->getStorage()->read();
52     }

```

Figura 9-2: Controlador de Signos vitales

Realizado por: Evelyn Valencia,2018

En la Figura 10-2 se muestra la parte del Modelo del módulo de Signos vitales están las consultas para la base de datos como ingresar, modificar y eliminar.

```

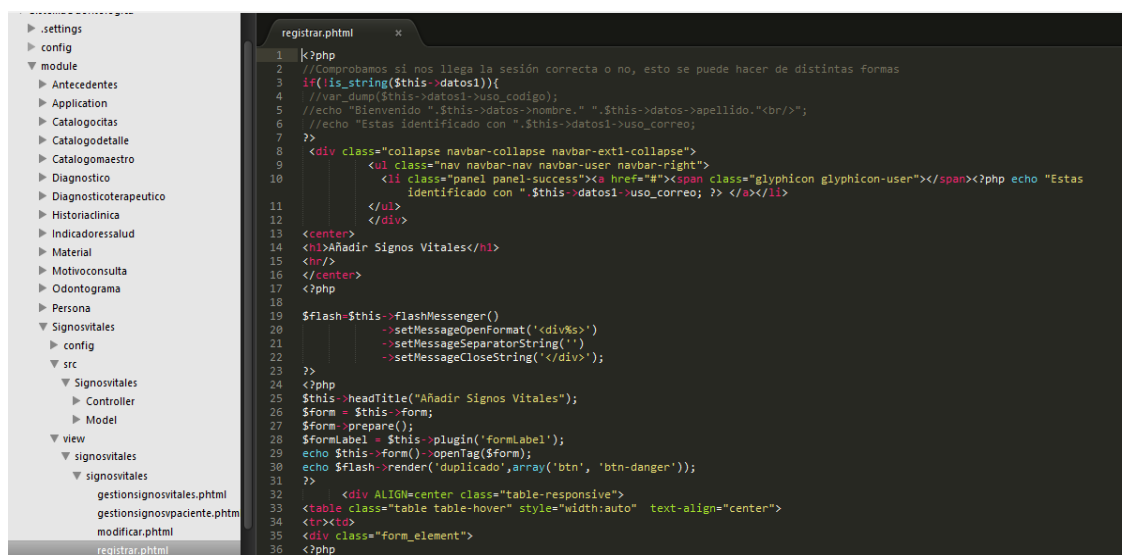
1 |<?php
2
3 namespace Signosvtales\Model\Entity;
4
5 use Zend\Db\TableGateway\TableGateway;
6 use Zend\Db\Adapter\Adapter;
7 use Zend\Db\Sql\Sql;
8 use Zend\Db\Sql>Select;
9
10 class Signosvtales extends TableGateway
11 {
12     private $pass;
13     private $correo;
14     private $dbAdapter;
15     private $sqlInsert;
16
17     public function __construct(Adapter $adapter = null, $databaseSchema = null, ResultSet $selectResultPrototype = null)
18     {
19         return parent::__construct('hca_signos_vitales', $adapter, $databaseSchema,$selectResultPrototype);
20     }
21
22     public function getSignosvtales()
23     {
24         $datos = $this->select();
25         return $datos->toArray();
26     }
27
28     private function cargaAtributos($datos=array())
29     {
30         $this->correo=$datos["email"];
31         $this->pass=$datos["pass"];
32     }
33
34     public function getSignosvtalesPorId($id)
35     {
36         $id = (int) $id;
37         $rowset = $this->select(array('svs_codigo' => $id));
38         $row = $rowset->current();
39
40         if (!$row) {
41             throw new \Exception("No hay registros asociados al valor $id");
42         }
43     }
44
45

```

Figura 10-2: Modelo del módulo Signos vitales

Realizado por: Evelyn Valencia,2018

En la carpeta view, consta los archivos de la vista del módulo de signos vitales como se muestra en la Figura 11-2.



```
1 <?php
2 //Comprobamos si nos llega la sesión correcta o no, esto se puede hacer de distintas formas
3 if(!is_string($this->datos1)){
4 //var_dump($this->datos1->uso_codigo);
5 //echo "Bienvenido ". $this->datos->nombre." ". $this->datos->apellido."<br/>";
6 //echo "Estas identificado con ". $this->datos1->uso_correo;
7 }
8 <div class="collapse navbar-collapse navbar-ext1-collapse">
9 <ul class="nav navbar-nav navbar-user navbar-right">
10 <li class="panel panel-success"><a href="#"><span class="glyphicon glyphicon-user"><?php echo "Estas
11 identificado con " . $this->datos1->uso_correo; ?> </a></li>
12 </ul>
13 </div>
14 <center>
15 <h1>Añadir Signos Vitales</h1>
16 </center>
17 <?php
18
19 $flash=$this->FlashMessenger()
20 <tr><td->setMessageOpenFormat('<div>')
21 <td->setMessageSeparatorString('')
22 <td->setMessageCloseString('</div>');
23 ?>
24 <?php
25 $this->headTitle("Añadir Signos Vitales");
26 $form = $this->form;
27 $form->prepare();
28 $formLabel = $this->plugin('formLabel');
29 echo $this->form()->openTag($form);
30 echo $flash->render('duplicado',array('btn', 'btn-danger'));
31 ?>
32 <div ALIGN=center class="table-responsive">
33 <table class="table table-hover" style="width:auto" text-align="center">
34 <tr><td>
35 <div class="form_element">
36 <?php
```

Figura 11-2: Vista del Módulo de Signos vitales

Realizado por: Evelyn Valencia,2018

Cabe recalcar para cada uno de los módulos se realizaron de la misma manera para cumplir con el patrón de diseño MVC usando con el Framework Zend.

2.6.6. Reuniones y entregables

La metodología Scrum nos permite una comunicación con el cliente también denominado Product Owner, para así desarrollar las funcionalidades del proyecto en base a las necesidades del cliente y también pueda estar al tanto con el desarrollo del sistema.

Según la metodología se obtiene un entregable cada dos semanas cada Sprint en la cual cuenta aproximadamente con 5 requerimientos entre historias de Usuarios e Historias técnicas.

2.6.7. Documentación

Se documenta cada una de las historias de Usuarios e historias técnicas esto permite evidenciar su realización y funcionamiento, de la cual cuenta con 265 tareas de Ingenierías y 659 pruebas de

Aceptación, las mismas se distribuyen en las 62 historias de Usuarios (HU) y 6 historias técnicas (HT).

2.6.8. *Manual de Usuario*

El manual de Usuario es un elemento importante ya que se explica en detalle el funcionamiento del sistema, en este manual se incluye desde el modo de acceso, ingreso, modificación, consulta, reportes o eliminación de información de acuerdo a su requerimiento. Se encuentra en el **ANEXO E**.

CAPITULO III

3. RESULTADOS Y DISCUSIÓN

En este capítulo se realiza el análisis de la eficiencia de la aplicación web para la gestión de la información del centro odontológico estableciendo indicadores en base de la situación actual del mismo con el fin de ejecutar cada una de las actividades requeridas en el cumplimiento de los objetivos planteados en el trabajo de titulación.

3.1. Calidad de Software

En la norma UNE-EN ISO 8402 define la calidad de software como “la calidad es el conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas”. (ISO-8402:1994,2016), en la norma ISO 9126 describe que la calidad de un proceso contribuye a mejorar la calidad del producto contribuye a mejorar la calidad en su uso (Oktaba et al., 2005).

3.2. Estándar ISO/IEC 9126

Esta parte de la ISO 9126 describe el modelo de calidad del producto de software. La primera parte del modelo especifica 6 características de calidad interna y externa, las cuales están divididas en subcaracterísticas, son manifestadas externamente cuando el software es utilizado como parte de un sistema, y son un resultado de atributos internos del software (Scalone, 2006).

3.3. Eficiencia

Es la capacidad de un software de proporcionar un rendimiento apropiado, de acuerdo a la cantidad de recursos usados bajo condiciones establecidos (Rojas, 2009).

Es la cantidad de recursos de computadoras y de código requeridos por un programa para llevar a cabo sus funciones (Scalone, 2006).

Sub características

La Eficiencia cuenta con dos sub-características:

- **Comportamiento con respecto al Tiempo.** Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos. ·
- **Comportamiento con respecto a Recursos.** Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

3.4. Definición de Criterios

Para evaluar la eficiencia del sistema para la gestión del centro odontológico, se define indicadores a base a la norma ISO/IEC 9126 estándar para evaluar la calidad de software.

En la Tabla 1-3, se muestra las variables e indicadores establecidos que se empleara en este trabajo de titulación.

Tabla 1-3: Variables e Indicadores para medir la eficiencia

Parámetro	Indicador	Descripción	Interpretación
Eficiencia	Tiempo de respuesta	¿Qué tan rápido responde el sistema?	Entre menos tiempo es mejor
	Utilización de recursos	¿El sistema utiliza los recursos de manera eficiente?	Cuanto más bajo, mejor

Realizado por: Evelyn Valencia,2018

3.4.1. Población

Para medir la eficiencia de la aplicación se tomó una población de 62 requerimientos funciones del sistema, los mismos que están definidos como historias de Usuarios para el sistema denominado “System Odontology”.

3.4.2. Muestra

Para la muestra se seleccionó una población de 7 categorías que se incluyen en los requerimientos funcionales del sistema. En la tabla 2-3, se muestra los requerimientos se tomaron en cuenta ya que tiene mayor uso en la aplicación.

Tabla 2-3: Tareas para el análisis

CATEGORÍA	SUBCATEGORÍA
GESTIÓN PACIENTE	<ul style="list-style-type: none">• Registrar Paciente• Modificar Paciente• Buscar Paciente
GESTIÓN SIGNOS VITALES	<ul style="list-style-type: none">• Registra Signos vitales• Modificar Signos vitales
GESTIÓN TRATAMIENTO	<ul style="list-style-type: none">• Registrar Tratamiento• Modificar Tratamiento• Buscar Tratamiento
GESTIÓN TRATAMIENTO DETALLE	<ul style="list-style-type: none">• Registrar tratamiento detalle• Modificar tratamiento detalle
GESTIÓN DIAGNÓSTICO	<ul style="list-style-type: none">• Registrar Diagnóstico• Modificar Diagnostico
GESTIÓN ODONTOGRAMA	<ul style="list-style-type: none">• Registrar Odontograma• Modificar Odontograma
EMITIR REPORTE	<ul style="list-style-type: none">• Reporte de Pacientes• Reporte de Materiales desbastecido

Realizado por: Evelyn Valencia, 2018

3.5. Tiempo de Respuesta

Para evaluar este parámetro de la eficiencia se tomó la variable X= tiempo de respuesta de una operación o actividad. Para esto se tomó la técnica de observación y con el uso de un cronometro.

Las tareas que están descritos en el *ítem 3.4.2.* son las que se miden en el indicador tiempo de respuesta.

Análisis actual del proceso de gestión de información del Centro Odontológico

En el centro Odontológico Frank Feijoo se maneja toda la información de sus pacientes de manera manual, ha ocasionado una seria de inconvenientes los mismo ya fueron descritos en el capítulo I en la parte de justificación aplicativa. Se realizó una entrevista a la auxiliar con el fin de conocer los tiempos aproximados en la realización de las tareas desde que registra el paciente, signos vitales,

tratamiento y generación de reportes, donde la auxiliar manifestó estos datos como un promedio general de los pacientes que son atendidos diariamente.

Para este estudio solo se especifican las tareas más importantes que utilizan comúnmente para la gestión de pacientes, gestión de diagnósticos, gestión de historia clínica, gestión de tratamientos y generación de reportes, en la Tabla 3-3, se detalla los tiempos en realizar dichas funciones.

Tabla 3-3: Tiempo que realiza sus funciones el Centro Odontológico

TAREAS	GESTIÓN TRADICIONAL (MIN)
Gestión datos del Paciente	14
Gestión de tratamiento del paciente	6
Gestión de tratamiento detalle paciente	14
Gestión de diagnóstico del paciente	14
Gestión Signos vitales	10
Odontograma	7
Reporte Materiales	6
Reporte de Pacientes	5
TOTAL	76
PROMEDIO	4,47

Realizado por: Evelyn Valencia, 2018

Fuente: (Horta and Mesías, 2016)

Tiempo de respuesta con la aplicación web para la gestión de la información del Centro Odontológico

Con la implementación de la aplicación se pretende reducir los tiempos empleados en la realización de las tareas en el centro odontológico para la gestión de la información, estos tiempos fueron cronometrados conjuntamente con el Odontólogo y la auxiliar. En la Tabla 4-3 se detalla los tiempos obtenidos para esta prueba.

Tabla 4-3: Tiempo en realizar los procesos utilizando la aplicación Web

Nro.	Tarea	Tiempo(MIN)
1	Gestión datos del Paciente	0,02325
3	Gestión de tratamiento del paciente	0,0327
4	Gestión de tratamiento detalle paciente	0,03388333
5	Gestión de diagnóstico del paciente	0,03525
6	Gestión de Signos vitales	0,03011667
7	Odontograma	0,02111667
8	Reporte Materiales	1,91666667
9	Reporte de Pacientes	2,11666667
TOTAL		4,20
PROMEDIO		0,46

Realizado por: Evelyn Valencia,2018

Fuente:(Horta y Mesías, 2016)

A través del análisis de tiempo de carga en la aplicación web, tiene como resultado de las tareas que se procesaron de un total 4,20 min en la que se detalla en el **ANEXO D**.

3.5.1. Comparación de Resultados

Con la finalidad de presentar los resultados de las tareas que se seleccionó para medir el tiempo tanto la gestión manual o tradicional como la gestión automatizada se usaron los métodos y técnicas antemencionadas.

Tabla 5-3: Comparación de Resultados

TAREAS	GESTIÓN TRADICIONAL (MIN)	GESTIÓN AUTOMATIZADA (MIN)
Gestión datos del Paciente	14	0,02325
Gestión de tratamiento del paciente	6	0,0327
Gestión de tratamiento detalle paciente	14	0,03388333
Gestión de diagnóstico del paciente	14	0,03525
Gestión Signos vitales	10	0,03011667
Odontograma	7	0,02111667
Reporte Materiales	6	1,91666667
Reporte de Pacientes	5	2,11666667
TOTAL	76	4,18

Realizado por: Evelyn Valencia,2018

Fuente:(Horta y Mesías, 2016)

Obteniendo los totales en minutos para la gestión tradicional con un resultado del 76 min tomando cuenta que es el 100% y para la gestión automatizada con el 4,18 min sale una diferencia del 5,5%. Por lo tanto, se puede destacar que con la implementación de la aplicación Web para la gestión del centro odontológico se redujo un 94,5% en realizar los procesos relevantes del mismo como se muestra en el gráfico 1-3.

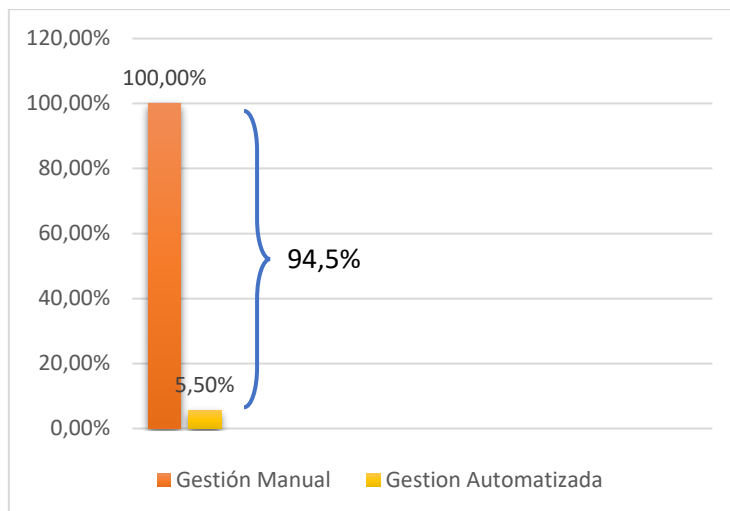


Gráfico 1-3: Tiempo de Respuesta
Realizado por: Evelyn Valencia, 2018

3.6. Utilización de Recursos

Para la característica de utilización, cuenta con dos sub-características y sus métricas, para este caso de estudio cabe recalcar que se aplicó de las tareas del *ítem 3.4.2.* anteriormente mencionadas, estas métricas son externa e interna, como se muestra en la siguiente tabla 6-3.

Tabla 6-3: Eficiencia en Utilización de Recursos

SUBCARACTERÍSTICA	MÉTRICAS
Utilización de Recurso	Utilización I/O Densidad de Mensaje
	Utilización de Memoria

Realizado por: Evelyn Valencia, 2018

Fuente: (Vivanco Villamar y Palán Tamayo, 2011)

Densidad del mensaje de uso de entrada/salida

Para el control de error, se realizó en la parte del controlador como también en la vista de la aplicación.

Tabla 7-3: Densidad del mensaje de uso entrada/salida

Característica:	Eficiencia interna
Sub-característica:	Utilización de recursos.
Métrica:	Densidad del mensaje de uso entrada/salida
Propósito de la Métrica:	esto permite contar el número de errores que pertenece de I/O, la advertencia y comparar el número estimado de líneas de código.
Tipo de Medida:	x=contable/contable A= contable B=contable
Interpretación:	El mayor es mejor
Tipo de Escala:	Absoluto
Fuente de Medición:	Código Fuente

Realizado por: Evelyn Valencia,2018

Fuente: (Vivanco Villamar y Palán Tamayo, 2011)

1. Registrar Paciente

Total, de errores controlados=4

1.1 La Cédula sea valida

```
35 var cedula = form.cedula.value;
36 array = cedula.split("");
37 num = array.length;
38 if ( num == 10 )
39 {
40     total = 0;
41     digito = (array[9]*1);
42     for( i=0; i < (num-1); i++)
43     {
44         mult = 0;
45         if ( ( i%2 ) != 0 ) {
46             total = total + ( array[i] * 1 );
47         }
48         else
49         {
50             mult = array[i] * 2;
51             if ( mult > 9 )
52                 total = total + ( mult - 9 );
53             else
54                 total = total + mult;
55         }
56     }
57     decena = total / 10;
58     decena = Math.floor( decena );
59     decena = ( decena + 1 ) * 10;
60     final = ( decena - total );
61     if ( ( final == 10 && digito == 0 ) || ( final == digito ) ) {
62         return true;
63     }
64     else
65     {
66         $('#ModalError').modal('show');
67         form.cedula.value="";
68         return false;
69     }
70 }
71 else
72 {
```

Figura 1-3: Error controlado, la cédula sea valida

Realizado por: Evelyn Valencia,2018

1.2 La cédula no esté duplicada

```
$persona2=$persona1->getPersonaPorCed($cedula);  
$codigo = $persona2["psa_codigo"];  
$idusuario=$datos1->uso_codigo;  
$historiac =new HistoriaClinica($this->dbAdapter);  
$insert1=$historiac->addHistoriaClinica($codigo,$idusuario);  
if($insert1==false){  
    $this->flashMessenger()->setNamespace("duplicado")->addMessage("Datos vacios para registrar");  
    return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'./persona/registrar');  
}else{  
    $this->flashMessenger()->setNamespace("add_correcto")->addMessage("Datos de la persona añadido correctame");  
    return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'./persona/gestionpersona');  
}  
}else{  
    $this->flashMessenger()->setNamespace("duplicado")->addMessage("Datos de la persona ya existe");  
    return $this->redirect()->refresh();  
}
```

Figura 2-3: Error controlado, la cédula no esté duplicada

Realizado por: Evelyn Valencia,2018

1.3 El correo no esté duplicado

```
if($datos[0]["count"]==0 && $datosr[0]["count"]){  
    //Insertamos en la bd  
    // $sqlInsert = "INSERT INTO hca_usuario (uso_codigo, uso_correo, psa_clave,psa_tipo) "  
    //."VALUES (14,$correo, $pass, $tipoU)";  
    //$this->dbAdapter->query("INSERT INTO hca_usuario (uso_codigo, uso_correo, psa_clave,psa_tipo,uso_fecha_modi  
    $insert=$persona->addPersona($cedula,$idcorreo,$primernombre,$segundonombre,$primerapellido,$segundoapellido,$fec  
    //Mensajes flash $this->flashMessenger()->addMessage("mensaje")  
    $persona =new Persona($this->dbAdapter);  
    $persona2=$persona1->getPersonaPorCed($cedula);  
    $codigo = $persona2["psa_codigo"];  
    $idusuario=$datos1->uso_codigo;  
    $historiac =new HistoriaClinica($this->dbAdapter);  
    $insert1=$historiac->addHistoriaClinica($codigo,$idusuario);  
    if($insert1==false){  
        $this->flashMessenger()->setNamespace("duplicado")->addMessage("Datos vacios para registrar");  
        return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'./persona/registrar');  
    }else{  
        $this->flashMessenger()->setNamespace("add_correcto")->addMessage("Datos de la persona añadido correctame");  
        return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'./persona/gestionpersona');  
    }  
}else{  
    $this->flashMessenger()->setNamespace("duplicado")->addMessage("Datos de la persona ya existe");  
    return $this->redirect()->refresh();  
}
```

Figura 3-3: Error controlado, el correo no esté duplicado

Realizado por: Evelyn Valencia,2018

1.4 No ingresar datos vacíos

```
$insert=$persona->addPersona($cedula,$idcorreo,$primernombre,$segundonombre,$primerapellido,$segundoapellido,$fec  
//Mensajes flash $this->flashMessenger()->addMessage("mensaje")  
$persona1 =new Persona($this->dbAdapter);  
$persona2=$persona1->getPersonaPorCed($cedula);  
$codigo =$persona2["psa_codigo"];  
$idusuario=$datos1->uso_codigo;  
$historiac =new HistoriaClinica($this->dbAdapter);  
$insert1=$historiac->addHistoriaClinica($codigo,$idusuario);  
if($insert1==false){  
    $this->flashMessenger()->setNamespace("duplicado")->addMessage("Datos vacios para registrar");  
    return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/persona/registrar');  
}else{  
    $this->flashMessenger()->setNamespace("add_correcto")->addMessage("Datos de la persona añadido correctame  
    return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/persona/gestionpersona');  
}  
}else{  
    $this->flashMessenger()->setNamespace("duplicado")->addMessage("Datos de la persona ya existe");  
    return $this->redirect()->refresh();  
}  
}
```

Figura 4-3: Error controlado, no ingresar datos vacíos

Realizado por: Evelyn Valencia,2018

2. Modificar Paciente

Total, de errores controlados=1

2.1. Enviar datos vacíos

```
// $date = new DateTime();  
// $date->getDateFormat('YYYY-MM-DD');  
$update=$persona->updatePersona($id, $idcorreo,$primernombre,$segundonombre,$primerapellido,$segundoapellido,$fechan  
//Mensajes flash $this->flashMessenger()->addMessage("mensaje")  
if($update==FALSE)  
{  
    $this->flashMessenger()->setNamespace("duplicado")->addMessage("Datos vacios por favor ingrese todos los datos");  
    return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/persona/modificar/'.$id);  
}else{  
    $this->flashMessenger()->setNamespace("add_correcto")->addMessage("Datos de la Persona se ha modificado correctamen  
    return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/persona/gestionpersona');  
}  
}
```

Figura 5-3: Error controlado enviar datos vacíos

Realizado por: Evelyn Valencia,2018

3. Buscar Paciente

Total, de errores controlados=1

3.1. Cuando no encuentra los datos


```

})
if($target.find('tbody tr:visible').size() === 0) {
  var col_count = $target.find('tr').first().find('td').size();
  var no_results = $('<tr class="filterTable_no_results"><td colspan="'+col_count+'>Datos no
    encontrados</td></tr>');
  $target.find('tbody').append(no_results);
}

```

Figura 6-3: Error controlado, cuando no se encuentra los datos

Realizado por: Evelyn Valencia,2018

4. Registrar Signos Vitales

Total, de errores controlados=1

4.1. Enviar datos vacíos

```

$sv =new Signosvital($this->dbAdapter);

$presionA=$this->request->getPost("presionArterial");
$FrecuenciaC=$this->request->getPost("FrecuenciaC");
$Temperatura=$this->request->getPost("Temperatura");
$Frespiratorio=$this->request->getPost("Frespiratorio");
$idusuario=$datos1->uso_codigo;

//Insertamos en la bd
// $sqlInsert = "INSERT INTO hca_usuario (uso_codigo, uso_correo, psa_clave,psa_tipo) "
//."VALUES (14,$correo, $pass, $tipoU)";
// $this->dbAdapter->query("INSERT INTO hca_usuario (uso_codigo, uso_correo, psa_clave,psa_tipo,uso_fecha_modi
$insert=$sv->addSignosvital($id,$presionA,$FrecuenciaC,$Temperatura,$Frespiratorio,$idusuario);
//Mensajes flash $this->flashMessenger()->addMessage("mensaje")
if($insert==FALSE){
  $this->flashMessenger()->setNamespace("duplicado")->addMessage("Por favor ingrese los datos completos");
  return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/signosvital/registrar/'.$id);
}
else{
  $this->flashMessenger()->setNamespace("add_correcto")->addMessage("Signos Vitales añadido correctamente");
  return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/signosvital/gestionsignosvtpaciente/'.$id);
}
}

```

Figura 7-3: Error controlado, enviar datos vacíos

Realizado por: Evelyn Valencia,2018

5. Modificar Signos Vitales

Total, de errores controlados=1

5.1. Enviar Datos vacíos

```

$presionA=$this->request->getPost("presionArterial");
$FrecuenciaC=$this->request->getPost("FrecuenciaC");
$Temperatura=$this->request->getPost("Temperatura");
$Frespiratorio=$this->request->getPost("Frespiratorio");
$date=$this->request->getPost("fecha_modificacion");
$idusuario=$datos1->uso_codigo;
//Insertamos en la bd
// $date = new Zend_Date();
//echo($date);

// $date = new DateTime();
// $date->getDateFormat('YYYY-MM-DD');

$update=$sv1->updateSignosvital($idsvs,$presionA,$FrecuenciaC,$Temperatura,$Frespiratorio,$idusuario,$date
//Mensajes flash $this->flashMessenger()->addMessage("mensaje")
if($update==FALSE){
..... $this->flashMessenger()->setNamespace("duplicado")->addMessage("Por favor ingrese todos los datos");
..... return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/signosvital/modificar/'.$id);
}

```

Figura 8-3: Error controlado, enviar datos vacíos

Realizado por: Evelyn Valencia,2018

6. Registrar Tratamiento

Total, de errores controlados=1

6.1. Enviar datos vacíos

```

$this->dbAdapter=$this->getServiceLocator()->get('Zend\Db\Adapter');
$tratamiento=new Tratamiento($this->dbAdapter);

$hca_codigo=$id;
$tto_profesional=$this->request->getPost("idprofesional");
$tto_fecha_apertura=$this->request->getPost("fecha_apertura");
$tto_fecha_control=$this->request->getPost("fecha_control");
$tto_numero_hoja=$this->request->getPost("numerot");
$tto_usuario_creacion=$datos1->uso_codigo;

$insert=$tratamiento->addTratamiento($hca_codigo, $tto_fecha_apertura,$tto_fecha_control,$tto_profesional,$tto_numero
// $insert=$tratamiento->addTratamiento($tto_codigo, $tto_fecha,$tto_fecha_control,$tto_profesional,$tto_numero
//Mensajes flash $this->flashMessenger()->addMessage("mensaje")
if($insert==false){
..... $this->flashMessenger()->setNamespace("duplicado")->addMessage("Por favor ingresar todos los datos");
..... return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/tratamiento/registro/'.$id);
}
else{
$this->flashMessenger()->setNamespace("add_correcto")->addMessage("Tratamiento añadido correctamente");
return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/tratamiento/gestiontratamientopaciente/'.$id);
}
}

```

Figura 9-3: Error controlado, enviar datos vacíos

Realizado por: Evelyn Valencia,2018

7. Modificar tratamiento

Total, de errores controlados=1

7.1. Enviar datos vacíos

```

$tto_profesional=$this->request->getPost("idprofesional");
$tto_fecha_apertura=$this->request->getPost("fecha_apertura");
$tto_fecha_control=$this->request->getPost("fecha_control");
$tto_numero_hoja=$this->request->getPost("numeroot");
$tto_usuario_modificacion=$datos1->uso_codigo;
$date=$this->request->getPost("fecha_modificacion");

$update=$tratamiento->updateTratamiento($id, $tto_fecha_apertura,$tto_fecha_control,$tto_profesional,$tto_numero_hoja,$
// $insert=$tratamientod->addTratamientodetalle($tto_codigo, $tde_fecha,$tde_diagnostico_complicacion,$tde_procedimiento
//Mensajes flash $this->flashMessenger()->addMessage("mensaje")
if($update==false){
    $this->flashMessenger()->setNamespace("duplicado")->addMessage("Por favor ingrese todos los datos");
    return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/tratamiento/modificar/'.$id);
}else{
    $this->flashMessenger()->setNamespace("add_correcto")->addMessage("Se ha modificado correctamente");
    return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'/tratamiento/gestiontratamientopaciente/'.$historia
}

```

Figura 10-3: Error controlado, enviar datos vacíos
Realizado por: Evelyn Valencia,2018

8. Buscar Tratamiento

Total, de errores controlados=1

8.1. No encontrar datos de búsqueda

```

$rows->each(function(){
    var $this = $(this);
    $this.text().toLowerCase().indexOf(search) === -1 ? $this.hide() : $this.show();
})
if($target.find('tbody tr:visible').size() === 0){
    var col_count = $target.find('tr').first().find('td').size();
    var no_results = '<tr class="filterTable_no_results"><td colspan="'+col_count+'>Datos no encontrados</td></tr>';
    $target.find('tbody').append(no_results);
}

```

Figura 11-3: Error controlado, no encontrar datos de búsqueda
Realizado por: Evelyn Valencia,2018

9. Registrar tratamiento detalle

Total, de errores controlados=1

9.1. Enviar datos vacíos

```

$idMat=$valor;
$cantidadnuevo=$this->request->getPost("numcantidad").$valor;
$insert=$tratamientod->addTratamientodetalle($tto_codigo,$idMat, $cantidadnuevo,$tde_sesion,$tde_fecha,$tde_diagnostico);

$cantidadM=$material1->getUnMaterialxId($idMat);
$cantidadAnterior=$cantidadM['mtl_cantidad'];
$mtl_utilizado=$cantidadM['mtl_utilizado'];
$restante=$cantidadAnterior-$cantidadnuevo;
$actualizadoS=$cantidadnuevo+$mtl_utilizado;
$update=$material->updateMaterialDetalle($idMat,$restante,$actualizadoS,$restante);
}
}
}
if($insert==false)
{
$this->flashMessenger()->setNamespace("duplicado")->addMessage("Por favor ingrese todos los datos");
return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'./tratamiento/agregar/'.$id);//Retornar Paciente
}
else{
$this->flashMessenger()->setNamespace("add_correcto")->addMessage("Detalle Tratamiento a adido correctamente");
return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'./tratamiento/gestiondetalle/'.$id);//Retornar Paciente
}
}

```

Figura 12-3: Error controlado, enviar datos vac os

Realizado por: Evelyn Valencia,2018

10. Modificar tratamiento detalle

Total, de errores controlados=1

10.1. Enviar datos vac os

```

$cantidadAnterior=$cantidadM['mtl_cantidad'];
$mtl_utilizado=$cantidadM['mtl_utilizado'];
$restante=$cantidadAnterior-$cantidadnuevo;
$actualizadoS=$cantidadnuevo+$mtl_utilizado;
$update=$material->updateMaterialDetalle($idMat,$restante,$actualizadoS,$restante);
$update1=$tratamiento2->updateTratamientoDetalle($sesion,$idTratamiento, $cantidadnuevo,$tde_fecha,$tde_diagnostico);
}
}
}
if($update1==false){
$this->flashMessenger()->setNamespace("duplicado")->addMessage("Por favor ingrese todos los datos");
return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'./tratamiento/modificartratamientod/'.$id);//Retornar Paciente
}
else{
}
}
}

```

Figura 13-3: Error controlado, enviar datos vac os

Realizado por: Evelyn Valencia,2018

11. Registrar diagn stico

Total, de errores controlados=1

11.1. Enviar datos vac os

```

// $this->dbAdapter->query("INSERT INTO hca_usuario (uso_codigo, uso_correo, psa_clave, psa_tipo, uso_fecha_modi
$insert=$sv->addDiagnosticoterapeutico($id,$value, $dto_descripcion,$idusuario);
// Mensajes flash $this->flashMessenger()->addMessage("mensaje")
}
}
}
}
}
}
if($insert==false){
..... $this->flashMessenger()->setNamespace("duplicado")->addMessage("No hay datos para Ingresar");
..... return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'diagnosticoterapeutico/registrar/'. $id);
} else{
$this->flashMessenger()->setNamespace("add_correcto")->addMessage("Diagnostico Terapeutico añadido correctamente");
return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'diagnosticoterapeutico/gestiondiagnostico/
}
}

```

Figura 14-3: Error controlado, enviar datos vacíos

Realizado por: Evelyn Valencia, 2018

12. Modificar diagnóstico

Total, de errores controlados=1

12.1. Enviar datos vacíos

```

$vista=array("form"=>$form, 'resultado'=>$resultado, 'datos1'=>$datos1, 'datos2'=>$datos2, 'datos3'=>$datos3, $view);
if($this->getRequest()->isPost()) {
    $form->setData($this->getRequest()->getPost());
    //Cargamos el modelo
    $this->dbAdapter=$this->getServiceLocator()->get('Zend\Db\Adapter');
    $sv =new Diagnosticoterapeutico($this->dbAdapter);
    $diagnosticoterapeutico=new Diagnosticoterapeutico($this->dbAdapter);
    if($_POST["list_diagnostico"]!=""){
..... $this->flashMessenger()->setNamespace("duplicado")->addMessage("No hay datos para Ingresar");
..... return $this->redirect()->toUrl($this->getRequest()->getBaseUrl().'diagnosticoterapeutico/modificar/'. $id);
..... }
}
}

```

Figura 15-3: Error controlado, enviar datos vacíos

Realizado por: Evelyn Valencia, 2018

13. Registrar Odontograma

Total, de errores controlados=1

13.1. Enviar datos vacíos

```

var respuesta= document.getElementById('resp_total');
respuesta.value=cadena_resp;

if (cadena_resp=="")
..... $('#ModalError').modal('show');
else
{
    //document.getElementById('total').value=cadena_resp;
    document.getElementById('form_resultados').submit();
}
}
}

```

Figura 16-3: Error controlado, enviar datos vacíos

Realizado por: Evelyn Valencia, 2018

14. Modificar Odontograma

Total, de errores controlados=1

14.1 Enviar datos vacíos

```
var respuesta= document.getElementById('resp_total');
respuesta.value=cadena_resp;

if (cadena_resp=="")
    .....$('#ModalError').modal('show');
else
{
    //document.getElementById('total').value=cadena_resp;
    document.getElementById('form_resultados').submit();
}
}
```

Figura 17-3: Error controlado, enviar datos vacíos

Realizado por: Evelyn Valencia,2018

15. Reporte de Materiales desbastecidos

Total, de errores controlados=1

15.1 establecer permisos para la generación de reporte

```
$identi=$this->auth->getStorage()->read();

if($identi!=false && $identi!=null){
    $datos1= (object) $identi; /// esto se incremento object
    $view=new ViewModel();
    // $t=count($this->datos1);
    .....
    .....if($datos1->psa_tipo=='2'){/// esta parte del codigo tuvo problemas
    .....$this->layout('layout/layout2');

    $this->dbAdapter=$this->getServiceLocator()->get('Zend\Db\Adapter');
```

Figura 18-3: Error controlado, establecer permiso para el reporte

Realizado por: Evelyn Valencia,2018

16. Reporte Paciente

Total, de errores controlados=1

16.1 establecer permisos para la generación de reporte

```

if($identi!=false && $identi!=null){
    $datos1= (object) $identi; // esto se incremento object
    $view=new ViewModel();
    //$t=count($this->datos1);
    .....
    if($datos1->psa_tipo == 2){ // esta parte del codigo tuvo problemas
    $this->layout('layout/layout2');
    .....
    $this->dbAdapter=$this->getServiceLocator()->get('Zend\Db\Adapter');
    $u=new Persona($this->dbAdapter);
    $vista=array
}

```

Figura 19-3: Error controlado, establecer permisos para el reporte
Realizado por: Evelyn Valencia,2018

En la tabla 8-3, se muestra los resultados obtenidos en esta métrica de la eficiencia interna

Tabla 8-3: Resultados de la métrica Densidad del mensaje I/O

Tarea	Medición, fórmula y cálculo de datos		Resultado
Registrar Paciente	A=4	B=87	X=0,046
Modificar Paciente	A=1	B=84	X=0,011
Buscar Paciente	A=1	B=49	X=0,020
Registrar Signos vitales	A=1	B=60	X=0,017
Modificar Signos vitales	A=1	B=62	X=0,016
Registrar Tratamiento	A=1	B=59	X=0,017
Modificar Tratamiento	A=1	B=63	X=0,016
Buscar Tratamiento	A=1	B=49	X=0,020
Registrar Tratamientos detalle	A=1	B=81	X=0,012
Modificar Tratamiento detalle	A=1	B=110	X=0,009
Registrar Diagnostico	A=1	B=76	X=0,013
Modificar Diagnostico	A=1	B=100	X=0,01
Registrar Odontograma	A=1	B=171	X=0,006
Modificar Odontograma	A=1	B=268	X=0,004
Reporte Paciente	A=1	B=33	X=0,030
Reporte de Materiales desbastecidos	A=1	B=44	X=0,023
TOTAL			0,27

Realizado por: Evelyn Valencia,2018

Métrica Uso de memoria RAM

Para evaluar este parámetro de la eficiencia en uso de memoria RAM se tomó la variable que es la medida en realizar una operación o actividad. Para esto se realizó con el administrador de tareas que nos proporciona el computador.

El tamaño que ocupa la memoria RAM con la aplicación es de MB al momento de realizar los procesos ya establecidos.

A continuación, en la tabla 9-3, se detalla la información acerca de la métrica uso de memoria RAM.

Tabla 9-3: Métrica de uso de memoria RAM

Característica:	Eficiencia externa
Sub-característica:	Utilización de recursos.
Métrica:	Uso de memoria RAM
Propósito de la Métrica:	conocer cuánto de memoria utiliza en cada tarea específica.
Tipo de Medida:	X = MB
Interpretación:	El menor es mejor
Tipo de Escala:	Ratio
Fuente de Medición:	Código Fuente

Realizado por: Evelyn Valencia,2018

Fuente: (Vivanco Villamar y Palán Tamayo, 2011)

17. Registrar Paciente

El promedio total que utiliza de memoria es: 127,5MB

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		1,8%	127,5 MB	0,1 MB/s	0 Mbps

Figura 20-3: Utilización de memoria, registrar paciente

Realizado por: Evelyn Valencia,2018

18. Modificar Paciente

El promedio total que utiliza de memoria es: 121,7MB

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		1,8%	121,7 MB	0,1 MB/s	0 Mbps

Figura 21-3: Utilización de memoria, modificar paciente

Realizado por: Evelyn Valencia,2018

19. Buscar paciente

El promedio total que utiliza de memoria es: 119,7MB

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0,6%	119,7 MB	0,1 MB/s	0 Mbps

Figura 22-3: Utilización de memoria, buscar paciente

Realizado por: Evelyn Valencia,2018

20. Ingresar Tratamiento

El promedio total que utiliza de memoria es: 122,0 MB


Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0,7%	122,0 MB	0,1 MB/s	0 Mbps

Figura 23-3: Utilización de memoria, ingresar tratamiento

Realizado por: Evelyn Valencia,2018

21. Modificar Tratamiento

El promedio total que utiliza de memoria es: 122,1MB

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0,3%	122,1 MB	0,1 MB/s	0 Mbps

Figura 24-3: Utilización de memoria, modificar tratamiento

Realizado por: Evelyn Valencia,2018

22. Buscar Tratamiento

El promedio total que utiliza de memoria es: 120,1 MB

Nombre	Estado	CPU	Memoria	Disco	Red
 System		0,5%	0,1 MB	0,1 MB/s	0 Mbps
>  Host de servicio: Servicio de usu...		0,6%	5,6 MB	0,1 MB/s	0 Mbps
>  Google Chrome (6)		0%	120,1 MB	0,1 MB/s	0 Mbps

Figura 25-3: Utilización de memoria, buscar tratamiento

Realizado por: Evelyn Valencia,2018

23. Registrar Tratamiento Detalle

El promedio total que utilizar de memoria es: 119,9 MB

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0%	119,9 MB	0,1 MB/s	0 Mbps

Figura 26-3: Utilización de memoria, registrar tratamiento

Realizado por: Evelyn Valencia,2018

24. Modificar Tratamiento Detalle

El promedio total que utiliza de memoria es: 120,2 MB

Nombre	Estado	8% CPU	34% Memoria	9% Disco	0% Red
>  Google Chrome (6)		0,5%	120,2 MB	0,1 MB/s	0 Mbps

Figura 27-3: Utilización de memoria, modificar tratamiento detalle

Realizado por: Evelyn Valencia,2018

25. Registra Diagnóstico

El promedio total que utiliza de memoria es: 119,8 MB

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		4,0%	119,8 MB	0,1 MB/s	0 Mbps

Figura 28-3: Utilización de memoria, registrar diagnóstico

Realizado por: Evelyn Valencia,2018

26. Modificar Diagnostico

El promedio total que utiliza de memoria es: 118,9 MB

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0%	118,9 MB	0,1 MB/s	0 Mbps

Figura 29-3: Utilización de memoria, modificar diagnóstico

Realizado por: Evelyn Valencia,2018

27. Registrar Signos Vitales

El promedio total que utiliza de memoria es: 119,8 MB

Nombre	Estado	6% CPU	34% Memoria	4% Disco	0% Red
>  Google Chrome (6)		0,6%	119,8 MB	0,1 MB/s	0 Mbps

Figura 30-3: Utilización de memoria, registrar signos vitales

Realizado por: Evelyn Valencia,2018

28. Modificar Signos Vitales

El promedio total que utiliza de memoria es: 121,1 MB

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (6)		0,6%	121,1 MB	0,1 MB/s	0 Mbps

Figura 31-3: Utilización de memoria, modificar signos vitales

Realizado por: Evelyn Valencia,2018

29. Ingresar Odontograma

El promedio total que utiliza de memoria es: 138,2 MB

Nombre	Estado	CPU	Memoria	Disco	Red
Google Chrome (7)		8,8%	138,2 MB	0,1 MB/s	0 Mbps

Figura 32-3: Utilización de memoria, ingresar odontograma

Realizado por: Evelyn Valencia,2018

30. Modificar Odontograma

El promedio total de utilización de memoria es: 113,1 MB

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (6)		0%	113,1 MB	0,1 MB/s	0 Mbps

Figura 33-3: Utilización de memoria, modificar odontograma

Realizado por: Evelyn Valencia,2018

31. Reporte de Pacientes

El promedio total que utiliza de memoria es: 164,0 MB

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (8)		0%	164,0 MB	0,1 MB/s	0 Mbps

Figura 34-3: Utilización de memoria, reporte de pacientes

Realizado por: Evelyn Valencia,2018

32. Reporte de materiales desbastecidos

El promedio total que utiliza de memoria es: 164,5 MB

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (8)		0,3%	164,5 MB	0,1 MB/s	0 Mbps

Figura 35-3: Utilización de memoria, reporte de materiales desbastecidos

Realizado por: Evelyn Valencia,2018

A continuación, se presenta la siguiente tabla 10-3 de manera más sintetizado los resultados de cada tarea que se midió bajo a esta métrica.

Tabla 10-3: Resultados de la métrica utilización de memoria

TAREAS	UTILIZACIÓN DE MEMORIA RAM (MB)
Gestión datos del Paciente	368,9
Gestión de tratamiento del paciente	364,2
Gestión de tratamiento detalle paciente	240,1
Gestión de diagnóstico del paciente	238,7
Gestión Signos vitales	240,9
Odontograma	251,2
Reporte Materiales	164,0
Reporte de Pacientes	164,5
TOTAL	2032,5
PROMEDIO	254,06 MB

Realizado por: Evelyn Valencia,2018

Fuente:(Vivanco Villamar y Palán Tamayo, 2011)

Métrica uso de procesador

Para evaluar este parámetro de la eficiencia en uso de procesador se tomó la variable X = porcentaje que es la medida en realizar una operación o actividad. Para esto se realizó con el administrador de tareas que nos proporciona el computador.

Tabla 11-3: Descripción de la métrica uso de Procesador

Característica:	Eficiencia externa
Sub-característica:	Utilización de recursos.
Métrica:	Uso de procesador
Propósito de la Métrica:	conocer cuánto de procesador utiliza en cada tarea.
Tipo de Medida:	x= Porcentaje.
Interpretación:	El menor es mejor
Tipo de Escala:	Ratio

Realizado por: Evelyn Valencia,2018

Fuente:(Vivanco Villamar y Palán Tamayo, 2011)

1. Ingresar Registrar Paciente

El porcentaje promedio de utilización de procesador es: 1,8%

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		1,8%	127,5 MB	0,1 MB/s	0 Mbps

Figura 36-3: Utilización de procesador, registrar paciente

Realizado por: Evelyn Valencia,2018

2. Modificar Paciente

El porcentaje promedio de utilización de procesador es: 1,8%

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		1,8%	121,7 MB	0,1 MB/s	0 Mbps

Figura 37-3: Utilización de procesador, modificar paciente

Realizado por: Evelyn Valencia,2018

3. Buscar paciente

El porcentaje promedio de utilización de procesador es: 0,6%

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0,6%	119,7 MB	0,1 MB/s	0 Mbps

Figura 38-3: Utilización de procesador, buscar paciente

Realizado por: Evelyn Valencia,2018

4. Ingresar Tratamiento

El porcentaje promedio de utilización de procesador es: 0,7%

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0,7%	122,0 MB	0,1 MB/s	0 Mbps

Figura 39-3: Utilización de procesador, ingresar paciente

Realizado por: Evelyn Valencia,2018

5. Modificar Tratamiento

El porcentaje promedio de utilización de procesador es: 0,3%

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0,3%	122,1 MB	0,1 MB/s	0 Mbps

Figura 40-3: Utilización de procesador, modificar tratamiento

Realizado por: Evelyn Valencia,2018

6. Buscar Tratamiento

El porcentaje promedio de utilización de procesador es: 0%

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0%	120,1 MB	0,1 MB/s	0 Mbps

Figura 41-3: Utilización de procesador, buscar tratamiento

Realizado por: Evelyn Valencia,2018

7. Registrar Tratamiento Detalle

El porcentaje promedio de utilización de procesador es: 0%

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0%	119,9 MB	0,1 MB/s	0 Mbps

Figura 42-3: Utilización de procesador, registrar tratamiento detalle

Realizado por: Evelyn Valencia,2018

8. Modificar Tratamiento Detalle

El porcentaje promedio de utilización de procesador es: 0,5%

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		0,5%	120,2 MB	0,1 MB/s	0 Mbps

Figura 43-3: Utilización de procesador, modificar tratamiento detalle

Realizado por: Evelyn Valencia,2018

9. Registra Diagnostico

El porcentaje promedio de utilización de procesador es: 4%

Nombre	Estado	CPU	Memoria	Disco	Red
>  Google Chrome (6)		4,0%	119,8 MB	0,1 MB/s	0 Mbps

Figura 44-3: Utilización de procesador, registrar diagnóstico

Realizado por: Evelyn Valencia,2018

10. Modificar Diagnostico

El porcentaje promedio de utilización de procesador es: 0%

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (6)		0%	118,9 MB	0,1 MB/s	0 Mbps

Figura 45-3: Utilización de procesador, modificar diagnóstico

Realizado por: Evelyn Valencia,2018

11. Registrar Signos Vitales

El porcentaje promedio de utilización de procesador es: 0,6%

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (6)		0,6%	119,8 MB	0,1 MB/s	0 Mbps

Figura 46-3: Utilización de procesador, registrar signos vitales

Realizado por: Evelyn Valencia,2018

12. Modificar Signos Vitales

El porcentaje promedio de utilización de procesador es: 0,6%

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (6)		0,6%	121,1 MB	0,1 MB/s	0 Mbps

Figura 47-3: Utilización de procesador, modificar signos vitales

Realizado por: Evelyn Valencia,2018

13. Ingresar Odontograma

El porcentaje promedio de utilización de procesador es: 8,8%

Nombre	Estado	CPU	Memoria	Disco	Red
Google Chrome (7)		8,8%	138,2 MB	0,1 MB/s	0 Mbps

Figura 48-3: Utilización de procesador, ingresar odontograma

Realizado por: Evelyn Valencia,2018

14. Modificar Odontograma

El porcentaje promedio de utilización de procesador es: 0 %

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (6)		0%	113,1 MB	0,1 MB/s	0 Mbps

Figura 49-3: Utilización de procesador, modificar odontograma

Realizado por: Evelyn Valencia,2018

15. Reporte de Pacientes

El porcentaje promedio de utilización de procesador es: 0%

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (8)		0%	164,0 MB	0,1 MB/s	0 Mbps

Figura 50-3: Utilización de procesador, reportes de pacientes

Realizado por: Evelyn Valencia,2018

16. Reporte de materiales desbastecidos

El porcentaje promedio de utilización de procesador es: 0,3%

Nombre	Estado	CPU	Memoria	Disco	Red
> Google Chrome (8)		0,3%	164,5 MB	0,1 MB/s	0 Mbps

Figura 51-3: Utilización de procesador, reporte de materiales desbastecidos

Realizado por: Evelyn Valencia,2018

Para tener una información más sintetizado de los resultados obtenidos de cada tarea bajo a esta métrica se muestra en la tabla 12-3.

Tabla 12-3: Resultados de la métrica utilización de procesador

TAREAS	UTILIZACION DE PROCESADOR (%)
Gestión datos del Paciente	4,2
Gestión de tratamiento del paciente	1
Gestión de tratamiento detalle paciente	0,5
Gestión de diagnóstico del paciente	4
Gestión Signos vitales	1,2
Odontograma	8,8
Reporte Materiales	0,3
Reporte de Pacientes	0
TOTAL	20
PROMEDIO	2,5

Realizado por: Evelyn Valencia,2018

3.7. Resultados Obtenidos

Para cada sub-característica de la eficiencia tiene su tabla de valoración para así ubicar los datos obtenidos en las mediciones y establecer si cumple con el grado de eficiencia optado por la norma ISO 9126.

Tiempo de Respuesta

El indicador de evaluación aceptable entre más peticiones que procesa la aplicación debe responder en menos tiempo para que su calificación sea alta.

Tabla 13-3: Indicador de evaluación, tiempo de respuesta

Calificación %	Tiempo	Valor Cualitativo
100%	[0 – 4.2] minutos	Excelente
90%	[4.3 – 7.5] minutos	Muy Bueno
75%	[7.6 – 10.9] minutos	Bueno
50%	[11.0 – 14.2] minutos	Aceptable
20%	[14.3 – 17.5] minutos	Malo
0%	[18.0 – ∞] minutos	Regular

Realizado por: Evelyn Valencia,2018

Con la finalidad de presentar los resultados de las tareas que están previamente establecidas para medir la eficiencia se usaron el método de observación y con la ayuda con un cronometro.

Tabla 14-3: Resultados obtenidos tiempo de respuesta

Nro.	Tarea	Tiempo (min)
1	Gestión datos del Paciente	0,02325
3	Gestión de tratamiento del paciente	0,0327
4	Gestión de tratamiento detalle paciente	0,03388333
5	Gestión de diagnóstico del paciente	0,03525
6	Gestión de Signos vitales	0,03011667
7	Odontograma	0,02111667
8	Reporte Materiales	1,91666667
9	Reporte de Pacientes	2,11666667
TOTAL		4,20
PROMEDIO		0,46

Realizado por: Evelyn Valencia,2018

Al analizar la sub característica Tiempo de respuesta, tuvo un promedio de 0,46 min, de acuerdo con la tabla o escala de valoración de 0 a 4,3 min determina que se encuentra en el rango de Excelente con un porcentaje de 100%.

Utilización de Recursos

Densidad del mensaje de uso de entrada/salida

El indicador de evaluación aceptable entre más errores controlados cada tarea su calificación va a ser alta.

Tabla 15-3: Indicador de evaluación

Calificación %	Contable	Valor Cualitativo
0%	[0 – 0.05]	Regular
20%	[0.010 – 0.015]	Malo
50%	[0.020 – 0.025]	Aceptable
75%	[0.030 – 0.035]	Bueno
90%	[0.040 – 0.045]	Muy bueno
100%	[0.050– ∞]	Excelente

Realizado por: Evelyn Valencia,2018

Con la finalidad de presentar los resultados de las tareas específicas para medir la densidad del mensaje de uso de entrada/salida.

A continuación, se muestra la tabla 16-3 de las tareas con sus respectivos valores.

Tabla 16-3: Resultados de la métrica Densidad del mensaje de uso I/O

TAREA	RESULTADO
Gestión Paciente	0,077
Gestión Signos vitales	0,033
Gestión tratamiento	0,053
Gestión tratamiento detalle	0,021
Gestión Diagnostico	0,023
Gestión Odontograma	0,01
Reporte Paciente	0,03
Reporte Materiales desbastecidos	0,023
TOTAL	0,27
PROMEDIO	0,03375

Realizado por: Evelyn Valencia,2018

Al analizar la métrica de la densidad de mensaje de uso de entrada/salida que pertenece a la sub característica Utilización de Recursos, tuvo un promedio de 0,34, de acuerdo con las mediciones que se realizó con cada una de las tareas específicas, de acuerdo con la tabla o la escala de valoración de 0,030 a 0,035% determina que se encuentra en el rango de Bueno con un porcentaje 75%/

1. Uso de Memoria RAM

El indicador de evaluación aceptable entre menos memoria RAM utiliza en cada tarea su calificación va a ser alta.

Tabla 17-3: Indicador de evaluación

Calificación %	Tiempo	Valor Cualitativo
100%	[0 – 150] MB	Excelente
90%	[200 – 250] MB	Muy Bueno
75%	[300 – 350] MB	Bueno
50%	[400 – 450] MB	Aceptable
20%	[500 – 550] MB	Malo
0%	[600 – ∞] MB	Regular

Realizado por: Evelyn Valencia,2018

Con la finalidad de presentar los resultados de las tareas específicas para medir la utilización de memoria RAM para la gestión automatizada.

A continuación, se muestra la Tabla 18-3 de las tareas con sus respectivos valores.

Tabla 18-3: Resultados obtenidos memoria RAM

TAREAS	UTILIZACION DE MEMORIA RAM (MB)
Gestión datos del Paciente	368,9
Gestión de tratamiento del paciente	364,2
Gestión de tratamiento detalle paciente	240,1
Gestión de diagnóstico del paciente	238,7
Gestión Signos vitales	240,9
Odontograma	251,2
Reporte Materiales	164,0
Reporte de Pacientes	164,5
TOTAL	2032,5
PROMEDIO	254,06 MB

Realizado por: Evelyn Valencia,2018

Al analizar la métrica utilización de memoria que pertenece a la sub característica utilización de recursos, tuvo un promedio de los datos obtenidos de 254,06 MB, pertinentes a las tareas específicas, de acuerdo con la tabla o escala de 200 a 250 MB nos determina que se encuentra en el rango de muy bueno con un porcentaje de 90%.

2. Uso de Procesador

El indicador de evaluación aceptable entre menos utiliza el procesador en cada tarea su calificación va a ser alta.

Tabla 19-3: Indicadores de evaluación

Calificación %	Porcentaje	Valor Cualitativo
100%	[0 – 0,5]	Excelente
90%	[1 – 1,5]	Muy Bueno
75%	[2 – 2,5]	Bueno
50%	[3 – 3,5]	Aceptable
20%	[4 – 4,5]	Malo
0%	[5 – ∞]	Regular

Realizado por: Evelyn Valencia,2018

Con la finalidad de presentar los resultados de las tareas específicas para medir la utilización de procesador del sistema informático para conocer qué tan eficiente es del mismo.

A continuación, se muestra la Tabla 20-3 los resultados obtenidos de este parámetro.

Tabla 20-3: Resultados obtenidos de la métrica utilización de procesador

TAREAS	UTILIZACION DE PROCESADOR (%)
Gestión datos del Paciente	4,2
Gestión de tratamiento del paciente	1
Gestión de tratamiento detalle paciente	0,5
Gestión de diagnóstico del paciente	4
Gestión Signos vitales	1,2
Odontograma	8,8
Reporte Materiales	0,3
Reporte de Pacientes	0
TOTAL	20
PROMEDIO	2,5

Realizado por: Evelyn Valencia,2018

Al analizar la sub característica utilización de procesador pertenece a la eficiencia externa, tuvo un promedio de los datos obtenidos de 2,5%, pertinentes a las tareas específicas, de acuerdo con la tabla o escala de 2 a 2,5 nos determina que se encuentra en el rango de bueno con un porcentaje de 75%

Resumen de los valores obtenidos en la métrica de eficiencia de cada una de las sub-características como son: tiempo de respuesta y utilización de recursos, con sus respectivas ponderaciones, como se muestra en la Tabla 21-3.

Tabla 21-3: Resumen de resultados obtenidos

EFICIENCIA DE LA APLICACIÓN		
Sub característica	Métrica	Porcentaje
Tiempo de respuesta	Tiempo de respuesta	45%
Utilización de Recurso	Densidad del mensaje de uso de entrada/salida	5%
	Uso de memoria RAM	30%
	Uso de procesador	5%

Realizado por: Evelyn Valencia,2018

En el siguiente Grafico se puede observar la eficiencia total obtenida con los resultados con cada uno de los parámetros basados con la norma ISO/IEC 9621, también se muestra el faltante de la eficiencia que se obtuvo en la métrica uso de procesador y densidad del mensaje de uso de entrada/salida que pertenecen a la sub característica utilización de recursos, debido que este valor se encuentra en el rango de Bueno esto no permitió que el sistema sea 100% eficiente.

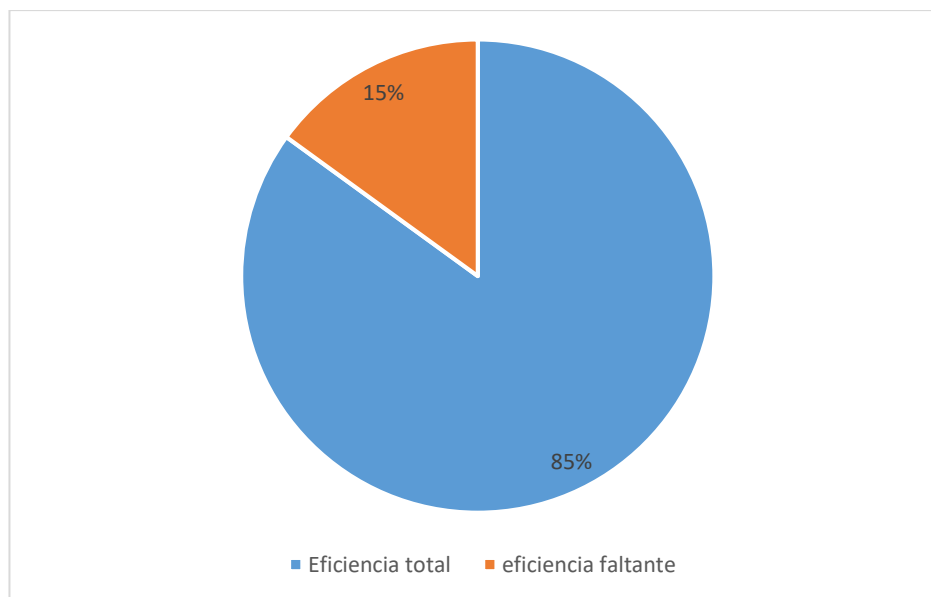


Gráfico 2-3: Resultados de la eficiencia del sistema

Realizado por: Evelyn Valencia,2018

CONCLUSIONES

- Para el desarrollo de la aplicación web se ha estudiado el framework Zend como el funcionamiento, la estructura y su estándar de codificación. Una de las características principales permite realizar abstracción en las consultas de base de datos, son fáciles de usar y viene incluido el framework Bootstrap, esto permite que la aplicación sea adaptable para los distintos dispositivos.
- Como se puede evidenciar, el sistema se desarrolló bajo el patrón MVC para cada módulo, por lo que se obtuvo un sistema dinámico y flexible que contiene una estructura modular, facilitando de este modo, la actualización y posibles cambios que se puede tener en el centro odontológico.
- Se aplicó una solución innovadora para la parte del módulo Odontograma, ya que cuenta con cada una de las piezas dentales esto permite al usuario final tener una interfaz dinámica, amigable y fácil de utilizar.
- Se realizó un estudio comparativo sobre los tiempos de respuesta tanto en la gestión Tradicional como la gestión Automatizada, obteniendo como resultados 100% en la gestión tradicional que equivale a 76 min y un 5,5% en la gestión automatizada que equivale el 4,18 min en realizar las tareas relevantes del sistema. Esto permitió saber que se redujo un 94,5% de tiempo en realizar sus actividades usando la aplicación web.
- Para validar la eficiencia se evaluó de acuerdo con la norma ISO/IEC 9126 en la cual determinó un 45% en tiempo de respuesta y en utilización de recursos un 40% dando como resultado el 85% en eficiencia de la aplicación web para la gestión de la información del centro odontológico.
- Para poder saber los procesos que gestionan tradicionalmente en el centro odontológico se usó el método de observación, esto permitió obtener y establecer los requerimientos para el desarrollo del sistema.
- Mediante el desarrollo de la aplicación se pudo mejorar los procesos de gestión de la información que se maneja en el centro odontológico desde la parte de los pacientes hasta los

tratamientos que se realizan, con el fin de mantener la integridad de los datos y reducir el tiempo de los procesos.

- Con la utilización de la metodología de desarrollo ágil Scrum elevó la productividad en cada una de las fases del ciclo de vida del Software permitiendo una estrecha comunicación con el cliente y se involucra en todo el proceso de desarrollo de la aplicación, de tal modo obtener un software de calidad y cumplir con cada uno de los requerimientos.

RECOMENDACIONES

- Se recomienda especificar claramente los requerimientos y el alcance que tendrá la aplicación web para así no tener inconvenientes con el cliente.
- Para la utilización del framework Zend se recomienda descargar el propio IDE que es el Zend Studio esto permite obtener el esqueleto o la estructura del Zend actualizado y también facilita en el reconocimiento de la sintaxis o de las librerías, ya que este framework cada año actualiza las librerías o ya les dan de baja.
- Se recomienda revisar el manual de usuario para aprender cómo se utiliza la aplicación web de tal manera haya un buen uso de la herramienta para mejorar el desempeño de la empresa.
- El framework Zend cuenta con una variedad de patrones como MVC, Observer, Doctryne, entre otros. También en componentes de las cuales se utilizan de acuerdo a las necesidades que requiera.
- Es necesario promover el uso de estándares de calidad de software, para así lograr resultados óptimos que satisfagan las necesidades del cliente y obtener un software de calidad.

BIBLIOGRAFÍA

ACOSTA, J.C., GREINER, C.L., DAPOZO, G.N. y ESTAYNO, M.G., 2012. Medición de atributos POO en frameworks de desarrollo PHP. *XVIII Congreso Argentino de Ciencias de la Computación* [en línea]. S.l.: s.n., [Consulta: 9 julio 2018]. Disponible en: <http://hdl.handle.net/10915/23734>.

ADERMANN, N. y BOGGIANO, J., 2018. Composer. [en línea]. [Consulta: 16 julio 2018]. Disponible en: <https://getcomposer.org/download/>.

ALONSO BALDONEDO, J., 2017. Modelo CMMI y métodos ágiles en la gestión de proyectos software. [en línea], [Consulta: 12 abril 2018]. Disponible en: <http://digibuo.uniovi.es/dspace/handle/10651/43638>.

ALONSO VEGA, A., 2013. Responsive Web Design interfaces web adaptables al dispositivo empleando HTML5 y CSS3. [en línea], [Consulta: 17 marzo 2018]. Disponible en: <https://ebuah.uah.es/dspace/handle/10017/19972>.

BALU, 2011. Zend Framework desde cero | Baluart.NET. [en línea]. [Consulta: 8 julio 2018]. Disponible en: <http://www.baluart.net/articulo/zend-framework-desde-cero>.

BGEO, 2014. POSTGRESQL. *BGEO - OPEN GIS & WATER SOLUTIONS* [en línea]. [Consulta: 8 julio 2018]. Disponible en: <http://www.bgeo.es/postgresql-2/?lang=es>.

BUSTOS TORRES, B.V., MARTÍN, A.E. y CECHICH, A., 2008. Extendiendo MVC para diseñar interfaces de usuario accesibles. *XIV Congreso Argentino de Ciencias de la Computación* [en línea]. S.l.: s.n., [Consulta: 1 junio 2017]. Disponible en: <http://hdl.handle.net/10915/21969>.

CHANGOTAXI, G. y PATRICIO, N., 2014. GESTIÓN DE HISTORIAS CLÍNICAS Y CONTROL DE CITAS MEDICAS PARA ADENTAL CLÍNICA ODONTOLÓGICA DE LA CIUDAD DE QUITO. [en línea], [Consulta: 10 julio 2017]. Disponible en: <http://www.dspace.cordillera.edu.ec/xmlui/handle/123456789/905>.

COBO, Á., GÓMEZ, P., PERÉZ, D. y ROCHA, R., 2005. *PHP y MySQL: tecnologías para el desarrollo de aplicaciones web*. Madrid: Díaz de Santos. ISBN 978-84-7978-706-6.

DE LA CRUZ VILLAR, J., 2008. *PHP 5 y MYSQL 5*. segunda. Lima: Megabyte. ISBN 9972-821-55-2.

DIMES, T., 2015. *Conceptos Básicos De Scrum: Desarrollo De Software Agile Y Manejo De Proyectos Agile*. S.l.: Babelcube Inc. ISBN 978-1-5071-0273-2.

EUGENIO DUARTE, 2013. jQuery: Qué es, Orígenes, Ventajas y Desventajas. *capacity academy* [en línea]. [Consulta: 17 marzo 2018]. Disponible en: <http://blog.capacityacademy.com/2013/03/16/jquery-que-es-origenes-ventajas-desventajas/>.

GARCÍA, I.M., 2015. Metodologías de desarrollo software. ¿tradicional o ágil? *MoleQla: revista de Ciencias de la Universidad Pablo de Olavide*, no. 19, pp. 7. ISSN 2173-0903.

GONZALEZ, 2014. 6 Características Que Hacen Tan Popular A Bootstrap. *JaGonzalez* [en línea]. [Consulta: 13 abril 2018]. Disponible en: <https://jagonzalez.org/6-caracteristicas-que-hacen-tan-popular-a-bootstrap/>.

GONZÁLEZ GONZÁLEZ, E., 2016. Desarrollo de una página web infantil en HTML5 Y Bootstrap. [en línea], [Consulta: 17 marzo 2018]. Disponible en: <http://academica-e.unavarra.es/xmlui/handle/2454/22726>.

HORTA, P. y MESÍAS, J., 2016. Diseño e implementación de un sistema web de gestión odontológico para la Empresa Eléctrica Riobamba S.A. [en línea], [Consulta: 17 julio 2018]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/6252>.

IPARRAGUIRRE VILLANUEVA, O., 2011. Zend Framework como marco de desarrollo de software. [en línea]. Tecnología. S.l. [Consulta: 9 julio 2018]. Disponible en: <https://es.slideshare.net/oiparraguirrev/zend-framework-como-marco-de-desarrollo-de-software-10378591>.

JQUERY.ORG, jQuery F., 2018. jQuery API Documentation. [en línea]. [Consulta: 17 marzo 2018]. Disponible en: <http://api.jquery.com/>.

JUAN DIEGO GAUCHAT, 2012. *El gran libro de HTML5, CSS3 y Javascript* [en línea]. S.l.: Marcombo. [Consulta: 17 marzo 2018]. ISBN 978-84-267-1782-5. Disponible en: https://books.google.com/books/about/El_gran_libro_de_HTML5_CSS3_y_Javascript.html?hl=es&id=szDMIRzwzuUC.

LESSIN JORGE, 2014. Ventajas y desventajas de usar Bootstrap. *jorgelessin.com* [en línea]. [Consulta: 17 marzo 2018]. Disponible en: <http://jorgelessin.com/ventajas-y-desventajas-de-usar-bootstrap/>.

LÓPEZ HERRERA, P., 2016. Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL. [en línea], [Consulta: 11 julio 2018]. Disponible en: <http://ri.uaemex.mx/handle/20.500.11799/62548>.

MENZINSKY, A., LÓPEZ, G. y PALACIO, J., 2016. Guía de Scrum Manager. [en línea]. [Consulta: 14 julio 2018]. Disponible en: http://scrummanager.net/files/scrum_manager.pdf.

MIRANDA CHINLLI, M., ROCHINA, R. y GONZALO, C., 2015. Análisis de los Frameworks Yii y Zend para mejorar la Usabilidad y la Compatibilidad del Sitio Web de la Cooperativa de Ahorros y Crédito Minga Ltda. [en línea], [Consulta: 11 julio 2018]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/4581>.

MONTOYA, J.E.H., 2013. Desarrollo de modelos de comunicación entre aplicaciones móviles y Web. , pp. 52.

MORENO, D.A.G. y ÁLVAREZ, J.L.P., 2005. DESARROLLO ADAPTABLE DE SOFTWARE, UNA SOLUCIÓN ÁGIL PARA APLICACIONES E-COMMERCE. , pp. 74.

OKTABA, H., ALQUICIRA, C., ANGÉLICA, E., RAMOS, S., MARTÍNEZ, A., QUINTANILLA, G., MARA, O., LÓPEZ, R., LÓPEZ, F., HINOJO MARÍA, L., RIVERA, E., MARÍA, L., OROZCO, J., YOLANDA, M., ORDÓÑEZ, F., ÁNGEL, M., LEMUS VERSIÓN, F. y ÍNDICE, A., 2005. *Modelo de Procesos para la Industria de Software: MoProSoft*. S.l.: s.n.

PABLO SUAREZ, 2012. Usar o no un Framework. *Kabytes* [en línea]. [Consulta: 12 abril 2018]. Disponible en: <https://www.kabytes.com/programacion/usar-o-no-un-framework/>.

PANTOJA, L. y PARDO, C., 2016. Evaluando la Facilidad de Aprendizaje de Frameworks mvc en el Desarrollo de Aplicaciones Web. *Publicaciones e Investigación*, vol. 10, no. 0, pp. 129-142. ISSN 2539-4088. DOI 10.22490/25394088.1592.

PERSAD, B.D. y PATRICIA, K., 2009. Trabajo Especial de Grado para optar al Título de Ingeniero en Computación. , pp. 83.

RAMOS TORRES, D., 2015. *Sitio Web para el Monitoreo Ambulatorio de PA* [en línea]. Thesis. S.l.: Universidad Central «Marta Abreu» de Las Villas. [Consulta: 17 marzo 2018]. Disponible en: <http://dspace.uclv.edu.cu:8089/xmlui/handle/123456789/3119>.

RODRIGUEZ GONZALEZ, P., 2008. *Estudio de la Aplicación de Metodologías Ágiles para la Evolución de Productos Software* [en línea]. masters. S.l.: Facultad de Informática (UPM). [Consulta: 9 julio 2018]. Disponible en: <http://oa.upm.es/1939/>.

ROJAS, A.D., 2009. Calidad del software, el camino al éxito. *Décima Primera Reunión de Otoño de Potencia, Electrónica y Computación del IEEE, XI ROPEC*, DOI 10.13140/2.1.2684.0322.

ROLFIPY, 2012. Estructura de una aplicación en zend framework. *Aprendiendo el zend framework* [en línea]. [Consulta: 8 julio 2018]. Disponible en: <https://myzendframework.wordpress.com/2012/06/23/estructura-de-una-aplicacion-en-zend-framework/>.

SANCHEZ, J., 2014. Los frameworks PHP más populares: ventajas de uso y elección | Freelancer Blog. [en línea]. [Consulta: 9 julio 2018]. Disponible en: <https://www.freelancer.es/community/articles/los-frameworks-php-mas-populares-ventajas-uso-eleccion>.

SÁNCHEZ SUÁREZ, J.M., 2018. Introducción a jQuery. | adictosaltrabajo. *Adictos al Trabajo* [en línea]. [Consulta: 17 marzo 2018]. Disponible en: <https://www.adictosaltrabajo.com/tutoriales/jquery-intro/>.

SÁNCHEZ, U., MARILÚ, E., IGUASNIA, A. y TERESA, L., 2015. Análisis Comparativo de los Frameworks PHP:ZEND y YUPP, para mejorar la Productividad en el Desarrollo del Sistema de Control y Gestión de la Empresa Regulada S.A. [en línea], [Consulta: 11 julio 2018]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/4595>.

SCALONE, F., 2006. *UNIVERSIDAD TECNOLOGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES MAESTRIA EN INGENIERIA EN CALIDAD ESTUDIO COMPARATIVO DE LOS MODELOS Y - PDF* [en línea]. Buenos Aires: Universidad Tecnológica Nacional Facultad Regional Buenos Aires. [Consulta: 8 julio 2018]. Disponible en: <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf>.

SIERRA, F., ACOSTA, J., ARIZA, J. y SALAS, M., 2017. Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web. *Revista Investigación y Desarrollo en TIC* [en línea], vol. 4, no. 2. [Consulta: 11 julio 2018]. ISSN 2216-1570. Disponible en:

<http://publicaciones.unisimonbolivar.edu.co/rdigital/ojs/index.php/identific/article/view/1517>.

TERESA ALBA, 2014. Desarrollo web en Bootstrap: Definición, ventajas y consejos. *Diseño Creativo: Marketing de Contenidos para Negocio* [en línea]. [Consulta: 13 abril 2018]. Disponible en: <https://xn--diseocreativo-lkb.com/desarrollar-web-bootstrap/>.

TRIGÁS GALLEGO, M., 2012. Metodología Scrum. [en línea], [Consulta: 8 julio 2018]. Disponible en: <http://openaccess.uoc.edu/webapps/o2/handle/10609/17885>.

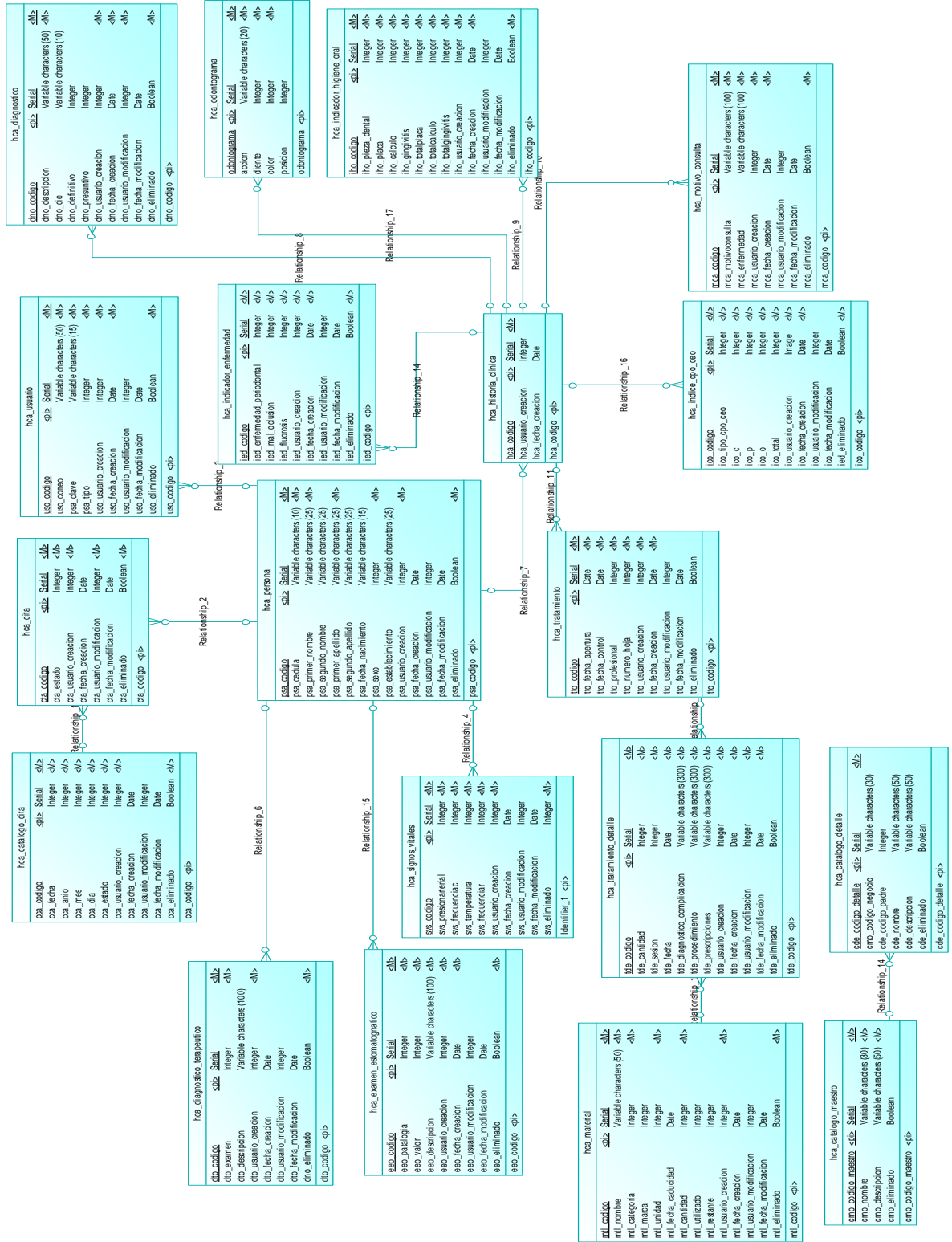
VIVANCO VILLAMAR, A.A. y PALÁN TAMAYO, B.O., 2011. *Evaluación de calidad del sistema Integrado para casas de valores SICAV de la bolsa de valores de Quito utilizando la norma ISO/IEC 14598.* QUITO: EPN. T-IS 1213 /CD 3948

XALAMBRÍ, S., 2014. Ventajas y desventajas de los pre-procesadores de CSS. *Medium* [en línea]. [Consulta: 13 abril 2018]. Disponible en: <https://medium.com/@sergiodxa/ventas-y-desventajas-de-los-pre-procesadores-de-css-6528fdac9926>.

ZEND. Zend Framework. [En línea] <https://framework.zend.com/about>.

ANEXOS

Anexo A Diseño Físico de la Base de Datos



Anexo B. Historias de Usuario

HT-01 Diseño e implementación de la base de datos

HISTORIA TÉCNICA	
Numero: HT-01	Nombre: Diseño e implementación de la base de datos
Modificación Historia técnica:	
Usuario: Desarrollador	Sprint Asignado: 1
Prioridad en Negocio: Media	Puntos Estimados: 40
Riesgo en desarrollo: Alta	Puntos Reales: 40
Descripción: Como desarrollador necesito que se diseñe y desarrolle la base de datos para el sistema	
Observaciones:	
Pruebas de Aceptación: <ul style="list-style-type: none">• El diseño creado debe soportar la cantidad de datos que manejará el negocio• La base de datos creada deberá contar con los nombres de los campos y tablas del diseño de la base de datos	

Prueba de Aceptación 1, HT-01 Diseño e implementación de la base de datos

Prueba de aceptación	
Código: PA-01	Historia Técnica: HT-01 Diseño e implementación de la base de datos
Nombre: La base de datos creada deberá contar con los nombres de los campos y tablas	
Responsable: Carolina Valencia	Fecha: 04-12-2017
Descripción: La base de datos creada deberá contar con los nombres de los campos y tablas	
Condiciones de Ejecución: <ul style="list-style-type: none">• Base de datos implementada	
Pasos de ejecución: <ul style="list-style-type: none">• Verificar que los nombres están acorde al estándar de programación.	
Resultado esperado: Los nombres de los campos de la base de datos están acorde al estándar de programación	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación 2, HT-01 Diseño e implementación de la base de datos

Prueba de aceptación	
Código: PA-02	Historia de Usuario: HT-01 Diseño e implementación de la base de datos
Nombre: El diseño creado debe soportar la cantidad de datos que manejará el negocio	
Responsable: Carolina Valencia	Fecha: 05-12-2012
Descripción: El diseño creado debe soportar la cantidad de datos que manejará el negocio	
Condiciones de Ejecución: <ul style="list-style-type: none">• Base de datos implementada	
Pasos de ejecución: <ul style="list-style-type: none">• Verificar que el gestor de base de datos es adecuado para la cantidad de carga que soportará.	

Resultado esperado: El gestor es adecuado.
Evaluación de la prueba: Exitosa.

HT-05 Diseño del estándar de interfaz de usuario

HISTORIA TÉCNICA	
Numero: HT-05	Nombre: Diseño del estándar de interfaz de usuario
Modificación Historia técnica:	
Usuario: Desarrollador	Sprint Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 16
Riesgo en desarrollo: Baja	Puntos Reales: 16
Descripción: Como desarrolladores se necesita establecer los colores, tipo de letra, posición de los colores de esta forma establecer un estándar en el diseño de la interfaz para que todos estén iguales.	
Observaciones:	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Los colores no deben forzar la vista de los usuarios y deben ser agradables. • La letra debe ser legible para los usuarios. • Las interfaces deben ser de fácil manejo por parte del usuario. • Las interfaces no deben estar recargadas con excesivas decoraciones. 	

Prueba de Aceptación 1, HT-05 Diseño del estándar de interfaz de usuario

Prueba de aceptación	
Código: PA-01	Historia Técnica: HT-05 Diseño del estándar de interfaz de usuario
Nombre: Los colores no deben forzar la vista de los usuarios y deben ser agradables.	
Responsable: Carolina Valencia	Fecha: 26-12-2017
Descripción: La interfaz deberá contener colores suaves pero llamativos para los usuarios.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Los colores deben estar definidos en el estándar. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Verificar que los colores son apropiados 	
Resultado esperado: Los colores elegidos son apropiados.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación 2, HT-05 Diseño del estándar de interfaz de usuario

Prueba de aceptación	
Código: PA-02	Historia Técnica: HT-05 Diseño del estándar de interfaz de usuario
Nombre: La letra debe ser legible para los usuarios.	
Responsable: Carolina Valencia	Fecha: 26-12-2017
Descripción: La interfaz deberá contener una fuente apropiada para el contenido en modo texto en el sistema.	
Condiciones de Ejecución:	

<ul style="list-style-type: none"> • Haber seleccionado el estándar de fuente que se sea legible dentro de las páginas web.
Pasos de ejecución: <ul style="list-style-type: none"> • Verificar que se pueda leer el contenido sin mayor esfuerzo.
Resultado esperado: La fuente consultada es aceptada.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación 3, HT-05 Diseño del estándar de interfaz de usuario

Prueba de aceptación	
Código: PA-03	Historia Técnica: HT-05 Diseño del estándar de interfaz de usuario
Nombre: Las interfaces deben ser de fácil manejo por parte del usuario.	
Responsable: Carolina Valencia	Fecha: 26-12-2017
Descripción: El contenido de las interfaces debe ser amigables y fáciles de entender para los usuarios.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener ya definido el estándar de interfaz de usuario. 	
Pasos de ejecución: <ul style="list-style-type: none"> • Comprobar que la interfaz de usuario es intuitiva. 	
Resultado esperado: El diseño escogido es aceptado.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación 4, HT-05 Diseño del estándar de interfaz de usuario

Prueba de aceptación	
Código: PA-04	Historia Técnica: HT-05 Diseño del estándar de interfaz de usuario
Nombre: Las interfaces no deben estar recargadas con excesivas decoraciones.	
Responsable: Carolina Valencia	Fecha: 26-12-2017
Descripción: El contenido de las interfaces debe ser el necesario sin abusar de la decoración apartando al usuario del mensaje de la página web.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener ya definido el estándar de interfaz de usuario. 	
Pasos de ejecución: <ul style="list-style-type: none"> • Comprobar que no haya elementos innecesarios en la interfaz. 	
Resultado esperado: El diseño interpretado en bosquejos por los desarrolladores cumple con las expectativas del Cliente y las recomendaciones adquiridas.	
Evaluación de la prueba: Exitosa.	

Tarea de Ingeniería 1, HT-05 Diseño del estándar de interfaz de usuario

TAREA DE INGENIERÍA	
Historia Técnica: HT-05 Diseño del estándar de interfaz de usuario	
Número de Tarea: TI-01	Nombre de Tarea: Consultar estándares para diagramar en HTML5.
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 25-12-2017	Fecha Fin: 25-12-2017
Programador Responsable: Carolina Valencia	
Descripción: Se investigará los métodos para diagramar y estructurar los contenidos de una interfaz web mediante HTML5.	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que el estándar consultado cumple con los estándares actuales en el diagramado de páginas web. 	

Prueba de Aceptación 6, HT-05 Diseño del estándar de interfaz de usuario

Prueba de aceptación	
Código: PA-05	Historia Técnica: HT-05 Diseño del estándar de interfaz de usuario
Nombre: Verificar que el estándar consultado cumple con los estándares actuales en el diagramado de páginas web.	
Responsable: Carolina Valencia	Fecha: 25/12/2017
Descripción: El estándar debe estar actualizado acorde con los últimos estándares para diagramar páginas web.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener ya definido el estándar de interfaz de usuario. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Comprobar que el estándar sea actualizado. 	
Resultado esperado: El estándar debe estar actual.	
Evaluación de la prueba: Exitosa.	

TAREAS DE INGENIERIA

Tarea de Ingeniería 2, HT-05 Diseño del estándar de interfaz de usuario

TAREA DE INGENIERÍA	
Historia Técnica: HT-05 Diseño del estándar de interfaz de usuario	
Número de Tarea: TI-02	Nombre de Tarea: Diseñar bosquejos de pantalla acerca de las interfaces con su respectivo estándar de diseño.
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 26-12-2017	Fecha Fin: 26-12-2017
Programador Responsable: Carolina Valencia	

Descripción: Se establece los bosquejos bajo el estándar establecido
Pruebas de Aceptación
<ul style="list-style-type: none"> • Verificar que los bosquejos y estándares sean aceptados.

Prueba de Aceptación 6, HT-05 Diseño del estándar de interfaz de usuario

Prueba de aceptación	
Código: PA-06	Historia Técnica: HT-05 Diseño del estándar de interfaz de usuario
Nombre: Verificar que los bosquejos y estándares sean aceptados	
Responsable: Carolina Valencia	Fecha: 26-12-2017
Descripción: El estándar que se ha definido debe ser aceptado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener ya definido el estándar de interfaz de usuario. 	
Pasos de ejecución:	
Comprobar que el estándar cumple con ciertas expectativas.	
Resultado esperado: El estándar debe ser actual.	
Evaluación de la prueba: Exitosa.	

HT-03 Conexión con la base de datos

HISTORIA TÉCNICA	
Número: HT-03	Nombre: Conexión con la base de datos
Modificación Historia técnica:	
Usuario: Desarrollador	Iteración Asignada: 1
Prioridad en Negocio: Baja	Puntos Estimados: 16
Riesgo en desarrollo: Baja	Puntos Reales: 16
Descripción:	
Como desarrollador quiero que la aplicación tenga conexión a la base de datos para poder manejar los datos desde la interfaz de usuario.	
Observaciones:	
Los datos no deberán ser manejados por el usuario directamente en el motor de base de datos.	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Cuando se realice la conexión se deberá poder acceder a la información almacenada en la base de datos. • Se podrán realizar las operaciones de ingreso, modificación y borrar registros. 	

Prueba de Aceptación 1, HT-03 Conexión con la Base de datos

Prueba de aceptación	
Código: PA-01	Historia Técnica: HT-03 Conexión con la base de datos
Nombre: Cuando se realice la conexión se deberá poder acceder a la información almacenada en la base de datos.	
Responsable: Carolina Valencia	Fecha: 15-12-2017
Descripción: Se prueba la conexión a una base de datos de prueba mediante la recuperación de datos.	

Condiciones de Ejecución: Conexión a la base de datos de prueba, base de datos de prueba implementada.
Pasos de ejecución: <ul style="list-style-type: none"> • Probar el acceso a la base de datos de prueba mediante una conexión remota en NetBeans. • Visualización de datos de la base de datos de prueba.
Resultado esperado: Los datos de la base de datos de prueba son accesibles.
Evaluación de la prueba: Exitosa

Prueba de Aceptación 2, HT-03 Conexión con la Base de datos

Prueba de aceptación	
Código: PA-02	Historia Técnica: HT-03 Conexión con la base de datos
Nombre: Se podrán realizar las operaciones de ingreso, modificación y borrar registros.	
Responsable: Carolina Valencia	Fecha: 15-12-2017
Descripción: Se prueba la conexión a una base de datos de prueba mediante la modificación, eliminación e ingreso de datos.	
Condiciones de Ejecución: Conexión a la base de datos de prueba, base de datos de prueba implementada.	
Pasos de ejecución: <ul style="list-style-type: none"> • Probar que la ruta de conexión a la base de datos este bien definida. • Ingreso de datos en la base de datos de prueba. • Modificación de datos en la base de datos de prueba. • Eliminación de datos en la base de datos de prueba. 	
Resultado esperado: Todas las funciones de ingreso, eliminación y modificación sean realizadas con éxito.	
Evaluación de la prueba: Exitosa	

TAREAS DE INGENIERIA

Tarea de Ingeniería 1, HT-03 Conexión con la Base de datos

TAREA DE INGENIERÍA	
Historia Técnica: HT-03 Conexión con la base de datos	
Número de Tarea: TI_01	Nombre de Tarea: Configurar los drivers en la carpeta local.php
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 14-12-2017	Fecha Fin: 14-12-2017
Programador Responsable: Carolina Valencia.	
Descripción: Verificar que haya conexión con la base de datos	
Pruebas de Aceptación: Verificar la configuración del driver y la ruta de la BD de prueba este bien escrito.	

Prueba de Aceptación 3, HT-03 Conexión con la Base de datos

Prueba de aceptación	
Código: PA-03	Historia Técnica: HT-06 Conexión con la base de datos
Nombre: Verificar la configuración del driver y la ruta de la BD este bien escrito.	
Responsable: Carolina Valencia	Fecha: 14-12-2017
Descripción: Se requiere conectarse a la base de datos Postgresql desde el Framework Zend	

Condiciones de Ejecución: Base de datos de prueba este implementada.
Pasos de ejecución <ul style="list-style-type: none"> • Verificar que este el driver este bien escrito • Verificar el nombre de la base de datos este bien escrito y también el usuario con su respectiva contraseña
Resultado esperado Drivers y nombre de la base de datos sean correctos
Evaluación de la prueba: Exitosa

TAREAS DE INGENIERIA

Tarea de Ingeniería 2, HT-03 Conexión con la Base de datos

TAREA DE INGENIERÍA	
Historia Técnica: HT-03 Conexión con la base de datos	
Número de Tarea: TI-02	Nombre de Tarea: Realizar pruebas de consultas para verificar si hay conexión con la base de datos prueba
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 15-12-2017	Fecha Fin: 15-12-2017
Programador Responsable: Carolina Valencia	
Descripción: Realizar pruebas de consultas para saber si existe conexión con la base de datos de prueba	
Pruebas de Aceptación	
Verificar que se muestre en la vista la consulta de prueba	

Prueba de Aceptación 4, HT-03 Conexión con la Base de datos

Prueba de aceptación	
Código: PA-04	Historia Técnica: HT-03 Conexión con la base de datos
Nombre: Verificar que se muestre en la vista la consulta de prueba	
Responsable: Carolina Valencia	Fecha: 15-12-2017
Descripción: se requiere realizar un archivo para extraer datos de una base de datos de prueba para saber si existe conexión	
Condiciones de Ejecución: Base de datos de prueba este implementada	
Pasos de ejecución <ul style="list-style-type: none"> • Verificar que en el controlador Application este las clases para la conexión a la Base datos • Este creado el atributo del Adapter • Este creado la función para la vista • Este creado la consulta de Base datos prueba 	
Resultado esperado: Se muestre en la Vista prueba los datos de la consulta de la base de datos prueba.	
Evaluación de la prueba: Exitosa	

Prueba de Aceptación 1, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA_01	Historia de Usuario: HU-01 Ingresar Usuario

Nombre: Si se registrar todos los campos del formulario y se procede a guardar se mostrará un mensaje de éxito	
Responsable: Carolina Valencia	Fecha: 18-01-2018
Descripción: Comprobar que se guarde los datos del usuario	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Usuario • Da clic en el icono de registrar Usuario • Se presenta los campos: Correo, contraseña • Se presiona en Guardar. 	
Resultado esperado: Debe registrar el usuario y mostrar el mensaje de éxito.	
Evaluación: Exitosa.	

Prueba de Aceptación 2, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA_02	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Si se registra datos de manera duplicada nos aparecerá un mensaje usuario ya existe.	
Responsable: Carolina Valencia	Fecha: 18-01-2018
Descripción: Comprobar que no se guarde información duplicado	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Usuario • Da clic en el icono de registrar Usuario • Se presenta los campos: Correo, contraseña • Se presiona en Guardar. 	
Resultado esperado: No debe guardar la información y debe salir un mensaje de error.	
Evaluación: Exitosa.	

Prueba de Aceptación 3, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA_03	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Si los datos se ingresan de manera errónea como el correo nos mostrara un mensaje de error de correo para que siga el formato.	
Responsable: Carolina Valencia	Fecha: 18-01-2018
Descripción: Comprobar de que se cumpla el formato del correo.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	

<ul style="list-style-type: none"> • Ingresar al menú Usuario • Da clic en el icono de registrar Usuario • Se presenta los campos: Correo, contraseña • Se presiona en Guardar.
Resultado esperado: Debe mostrar un mensaje que debe seguir el formato que tiene el correo
Evaluación: Exitosa.

Prueba de Aceptación 4, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA_04	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Si existen campos vacíos nos mostrará un mensaje de llene todos los campos.	
Responsable: Carolina Valencia	Fecha: 18-01-2018
Descripción: Comprobar que no se guarde campos vacíos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Usuario • Da clic en el icono de registrar Usuario • Se presenta los campos: Correo y contraseña con los campos vacíos • Se presiona en Guardar. 	
Resultado esperado: Debe mostrar un mensaje que se debe llenar todos los campos.	
Evaluación: Exitosa.	

TAREAS DE INGENIERIA

Tarea de Ingeniería 1, HU-01 Ingresar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-01 Ingresar Usuario	
Número de Tarea: TI_01	Nombre de Tarea: crear el módulo Usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 01-01-2018	Fecha Fin: 01-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe realizar el módulo usuario con su respectiva estructura	
(Reverso)Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Verificar que la estructura este bien establecida • Verificar que el módulo este incorporado en la configuración de la aplicación • Verificar que el archivo modulo.php y la configuración del módulo estén bien estructurados para que funcione el controlador. 	

Tarea de Ingeniería 1- PA 1, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-01-01	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que la estructura este bien establecida	

Responsable: Carolina Valencia	Fecha: 01-01-2018
Descripción: Comprobar que el módulo de usuario este bien estructurada para su buen funcionamiento.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo de usuario ya debe existir. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la carpeta Modulo de Usuario • Verificar si están las carpetas: src, config y view • Verificar si están los archivos: module.config.php y module.php 	
Resultado esperado:	
Las carpetas y archivos del módulo existan.	
Evaluación: Exitosa.	

Tarea de Ingeniería1- PA 2, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-02-01	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que el módulo este incorporado en la configuración de la aplicación.	
Responsable: Carolina Valencia	Fecha: 01-01-2018
Descripción: Comprobar que el módulo de usuario este en la lista de la configuración de la aplicación	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo de usuario ya debe existir. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar la carpeta autoloadd de la aplicación • Ingresar a la carpeta application.config.php 	
Resultado esperado:	
Debe estar el nombre del módulo en la lista de la aplicación.	
Evaluación: Exitosa.	

Tarea de Ingeniería 1- PA 3, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-03-01	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que el archivo modulo.php y la configuración del módulo estén bien estructurado para que funcione el controlador.	
Responsable: Carolina Valencia	Fecha: 01-01-2018
Descripción: Comprobar que los archivos del módulo Usuario estén bien estructurados	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo de usuario ya debe existir. • Los archivos modulo.php y module.config.php ya debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al archivo modulo.php • Que contenga las librerías, creada la clase del módulo con sus respectivas funciones como getConfig y getAutoloaderConfi. • Ingresar a la carpeta config • Abrir el archivo module.config.php • debe invocar el controlador Usuario y que este el view manager. 	

Resultado esperado: Debe estar ambos archivos configurados correctamente
Evaluación: Exitosa.

Tarea de Ingeniería 2, HU-01 Ingresar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-01 Ingresar Usuario	
Número de Tarea: TI_02	Nombre de Tarea: Crear el controlador de Usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 17-01-2018	Fecha Fin: 01-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe realizar el módulo usuario con su respectiva estructura	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que el controlador este bien estructurada • Verificar que se muestre en la aplicación que existe el controlador 	

Tarea de Ingeniería 2- PA 1, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-01-02	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que el controlador Usuario este bien estructurada	
Responsable: Carolina Valencia	Fecha: 01-01-2018
Descripción: Comprobar que este la carpeta del controlador y el archivo controlador usuario	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo de usuario ya debe existir. • Debe estar la carpeta del controlador y su respectivo archivo 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al módulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario y después abrir la carpeta Controller • Abrir el archivo UsuarioController • Debe contener el namespace del archivo con su ruta, deben estar las clases adapter, AbstractActionController, ViewModel, etc. • Debe coincidir el nombre del archivo con la clase y debe estar extendido con la clase AbstractActionController. 	
Resultado esperado: Debe estar el archivo UsuarioController estructurado correctamente	
Evaluación: Exitosa.	

Tarea de Ingeniería 2- PA 2, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-02-02	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que se muestre en la aplicación que existe el controlador	
Responsable: Carolina Valencia	Fecha: 01-01-2018
Descripción: Comprobar que la aplicación invoque al controlador	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El controlador usuario debe existir 	

Pasos de ejecución: <ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Usuario
Resultado esperado: Debe mostrar un mensaje que no existe la vista, eso significa que está funcionando el controlador.
Evaluación: Exitosa.

Tarea de Ingeniería 3, HU-01 Ingresar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-01 Ingresar Usuario	
Número de Tarea: TI_03	Nombre de Tarea: crear la vista del módulo Usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 17-01-2018	Fecha Fin: 17-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe realizar la vista del módulo Usuario para poder saber si la estructura en general funciona.	
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que el controlador invoque a la vista • Verificar que se muestre en la aplicación la vista de prueba 	

Tarea de Ingeniería 3- PA 1, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-01-03	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que el controlador invoque a la vista	
Responsable: Carolina Valencia	Fecha: 01-01-2018
Descripción: Comprobar que el controlador llame a la vista ya creada.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Debe existir una vista 	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir la carpeta modulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario y luego Controller • Abrir el archivo UsuarioController • Observar que este la función que viene creado por defecto 	
Resultado esperado: Debe estar invocado la vista	
Evaluación: Exitosa.	

Tarea de Ingeniería 3- PA 2, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-02-03	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que se muestre en la aplicación la vista de prueba	
Responsable: Carolina Valencia	Fecha: 01-01-2018
Descripción: Comprobar que la aplicación muestre la vista de prueba	

Condiciones de Ejecución: <ul style="list-style-type: none"> • Debe estar el nombre del módulo en la lista de módulos de la aplicación • Debe existir tanto el módulo usuario como el controlador y la vista.
Pasos de ejecución: <ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Usuario
Resultado esperado: Debe mostrar la vista de prueba con el mensaje “Este es la vista de prueba”
Evaluación: Exitosa.

Tarea de Ingeniería 4, HU-01 Ingresar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-01 Ingresar Usuario	
Número de Tarea: TI_04	Nombre de Tarea: crear la clase usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 17-01-2018	Fecha Fin: 01-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la clase usuario para poder realizar las diferentes funcionalidades que va a tener el usuario.	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que la clase Usuario este bien estructurada • Verificar que el controlador invoque a la clase 	

Tarea de Ingeniería 4- PA 1, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-01-04	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que la clase Usuario este bien estructurada	
Responsable: Carolina Valencia	Fecha: 01-01-2018
Descripción: Comprobar que la clase Usuario este bien estructurada para su buen funcionamiento	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe estar creado la clase Usuario en la carpeta Model 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la carpeta Modulo Usuario • Debe tener su namespace y las respectivas clases para su funcionamiento • La clase debe estar extendido Gateway • Debe tener su adaptador para llamar a la tabla Usuario de la Base de datos 	
Resultado esperado: Exitosa Debe estar configurado la clase Usuario.	
Evaluación: Exitosa.	

Tarea de Ingeniería 4- PA 2, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-02-04	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que el controlador invoque a la clase	
Responsable: Carolina Valencia	Fecha: 01-01-2018

Descripción: Comprobar que la aplicación muestre la vista de prueba
Condiciones de Ejecución: <ul style="list-style-type: none"> • Debe existir la clase Usuario en la clase Model
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el módulo Usuario • Abrir la carpeta src • Abrir el archivo UsuarioController
Resultado esperado: Debe mostrar que se invoque la clase Usuario
Evaluación: Exitosa.

Tarea de Ingeniería 5, HU-01 Ingresar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-01 Ingresar Usuario	
Número de Tarea: TI_05	Nombre de Tarea: crear la clase y la vista del formulario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 17-01-2018	Fecha Fin: 02-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la clase formulario para poder realizar el registro de usuario	
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que la estructura de la clase formulario este bien • Verificar que en el controlador Usuario llame la clase formulario 	

Tarea de Ingeniería 5- PA 1, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-01-05	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que la estructura de la clase formulario este bien	
Responsable: Carolina Valencia	Fecha: 02-01-2018
Descripción: Comprobar que la clase formulario este bien estructurado para su buen funcionamiento.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Debe existir la clase formulario en la carpeta Form 	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el módulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario • Abrir la carpeta Form • Abrir el archivo Formulario 	
Resultado esperado: Debe tener el namespace, llamar a las clases AdapterInterface, Element, Form, Captcha, Factory y por último debe estar extendido Form.	
Evaluación: Exitosa.	

Tarea de Ingeniería 5- PA 2, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-02-05	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que en el controlador Usuario llame la clase formulario	
Responsable: Carolina Valencia	Fecha: 02-01-2018
Descripción: Comprobar que la clase UsuarioController llame a la clase formulario para que la vista pueda utilizar el formulario.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir la clase formulario en la carpeta Form 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el módulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario • Abrir la carpeta Controller • Abrir el archivo UsuarioController 	
Resultado esperado:	
Debe llamar la clase Formulario.	
Evaluación: Exitosa.	

Tarea de Ingeniería 6, HU-01 Ingresar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-01 Ingresar Usuario	
Número de Tarea: TI_06	Nombre de Tarea: crear la función registroAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 17-01-2018	Fecha Fin: 02-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la registroAction para poder realizar el registro del usuario.	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que cumpla con la estructura • Verificar si la aplicación puede llamar a esa función 	

Tarea de Ingeniería 6- PA 1, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-01-06	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 02-01-2018
Descripción: Comprobar que la función que está en la clase UsuarioController está bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase UsuarioController 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario 	

<ul style="list-style-type: none"> • Abrir la carpeta Controller • Abrir la carpeta UsuarioController
Resultado esperado: Debe estar la función registroAction y que retorne la nueva vista.
Evaluación: Exitosa.

Tarea de Ingeniería 6- PA 2, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-02-06	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 02-01-2018
Descripción: Comprobar que la función incorporada en la clase UsuarioController se pueda llamar en la aplicación	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase UsuarioController • La vista debe llamar registro 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la aplicación • Abrir Usuario • Dar clic en el icono registrar 	
Resultado esperado: No debe salir ningún error.	
Evaluación: Exitosa.	

Tarea de Ingeniería 7, HU-01 Ingresar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-01 Ingresar Usuario	
Número de Tarea: TI_07	Nombre de Tarea: crear la función de registrar usuario en la clase usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18-01-2018	Fecha Fin: 02-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la función de registrar usuario.	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se llame a la tabla Usuario en la clase Usuario • Verificar si la consulta funciona. 	

Tarea de Ingeniería 7- PA 1, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-01-07	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que se llame a la tabla Usuario en la clase Usuario	
Responsable: Carolina Valencia	Fecha: 02-01-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla Usuario para poder realizar el registro de Usuario.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase Usuario ya debe existir 	

Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el Módulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario • Abrir la carpeta Model • Abrir la carpeta Entity • Abrir la clase Usuario
Resultado esperado: Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla Usuario
Evaluación: Exitosa.

Tarea de Ingeniería 7- PA 2, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-02-07	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar si la consulta funciona.	
Responsable: Carolina Valencia	Fecha: 02-01-2018
Descripción: Comprobar que la consulta permita registrar la información del Usuario.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La clase Usuario debe existir • La vista debe existir 	
Pasos de ejecución: <ul style="list-style-type: none"> • En el UsuarioControlador debe 	
Resultado esperado: La interfaz gráfica de la funcionalidad cumple con los estándares preestablecidos.	
Evaluación: Exitosa.	

Tarea de Ingeniería 8, HU-01 Ingresar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-01 Ingresar Usuario	
Número de Tarea: TI_08	Nombre de Tarea: Implementar la interfaz para el registro de usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 02-01-2018	Fecha Fin: 02-01-2018
Programador Responsable: Carolina Valencia	
Descripción: implementar la interfaz gráfica para la historia de usuario ingresar usuario	
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado. • Verificar que se haya ingresado correctamente el usuario. 	

Tarea de Ingeniería 8- PA 1, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-01-08	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Responsable: Carolina Valencia	Fecha: 02-01-2018

Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.
Condiciones de Ejecución: <ul style="list-style-type: none"> • La interfaz gráfica de la funcionalidad está implementada.
Pasos de ejecución: <ul style="list-style-type: none"> • Ejecución de las interfaces. • Comprobar el estándar.
Resultado esperado: La interfaz gráfica de la funcionalidad cumple con los estándares preestablecidos.
Evaluación: Exitosa.

Tarea de Ingeniería 8- PA 2, HU-01 Ingresar Usuario

Prueba de aceptación	
Código: PA-02-08	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que se haya ingresado correctamente el usuario.	
Responsable: Carolina Valencia	Fecha: 02-01-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad debe estar implementada con su interfaz gráfica. 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ejecución de la funcionalidad. • Se ingresa a la pantalla ingresar usuario • Se presenta la pantalla de registrar usuario • Se ingresa los datos: correo, contraseña. • Se presiona en Guardar. • Debe mostrar un mensaje de éxito o de error 	
Resultado esperado: Los datos ingresados estén registrados en la Base de datos	
Evaluación: Exitosa.	

HU-02 Modificar Usuario

HISTORIA DE USUARIO	
Número: HU-02	Nombre: Modificar Usuario
Modificación historia de usuario:	
Usuario: Paciente	Sprint Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 16
Riesgo en desarrollo: Media	Puntos Reales: 16
Descripción: Como administrador me gustaría que puedan modificar los datos del usuario para poder tener un registro con la información actualizada.	
Observaciones: No podrá modificar el correo	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Si se modifica los datos del usuario se mostrará un mensaje se Modificó con éxito. • Si se modifica y deja campos en blanco el sistema mostrará un mensaje de error indicando que se llene todos los campos. 	

- Si no modifica los datos del usuario y presiona cancelar se mostrará los datos del usuario sin modificar.

Prueba de Aceptación 1, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA_01	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Si se modifica los datos del usuario se mostrará un mensaje se Modificó con éxito.	
Responsable: Carolina Valencia	Fecha: 04-01-2018
Descripción: Comprobar que se haya modificado los datos del usuario	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Usuario • Muestra la lista de los usuarios • Dar clic modificar • Se presenta el campo contraseña: cambiamos la contraseña. • Se presiona en Guardar. 	
Resultado esperado:	
Debe modificarse los datos del usuario y mostrar el mensaje de éxito.	
Evaluación: Exitosa.	

Prueba de Aceptación 2, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA_02	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Si se modifica y deja campos en blanco el sistema mostrará un mensaje de error indicando que se llene todos los campos.	
Responsable: Carolina Valencia	Fecha: 04-01-2018
Descripción: Comprobar que no se guarde información en blanco	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Usuario • Muestra la lista de los usuarios • Dar clic modificar al cliente que se ingreso • Se presenta el campo contraseña y dejamos en blanco • Se presiona en Guardar. 	
Resultado esperado:	
Debe mostrar un mensaje de error	
Evaluación: Exitosa.	

Prueba de Aceptación 3, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA_03	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Si no modifica los datos del usuario y presiona cancelar se mostrará los datos del usuario sin modificar.	

Responsable: Carolina Valencia	Fecha: 04-01-2018
Descripción: Comprobar que si cancela la modificación no debe cambiar nada de la información.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Usuario • Muestra la lista de los usuarios • Dar clic modificar al cliente que se ingreso • Se presenta el campo contraseña y dejamos en blanco • Se presiona en Cancelar 	
Resultado esperado: No se debe alterar la información del usuario.	
Evaluación: Exitosa.	

Tarea de Ingeniería 1, HU-02 Modificar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-02 Modificar Usuario	
Número de Tarea: TI_01	Nombre de Tarea: crear la función gestionusuarioAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 03-01-2018	Fecha Fin: 03-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la gestionusuarioAction para que liste todos los Usuarios y poder modificar sus datos	
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que cumpla con la estructura • Verificar si la aplicación puede llamar a esa función 	

Tarea de Ingeniería 1- PA 1, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-01-01	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 03-01-2018
Descripción: Comprobar que la función que está en la clase UsuarioController está bien estructurado	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La clase UsuarioController 	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el Módulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario • Abrir la carpeta Controller • Abrir la carpeta UsuarioController 	
Resultado esperado: Debe estar la función gestionusuarioAction y que retorne la nueva vista.	

Evaluación: Exitosa.

Tarea de Ingeniería 1- PA 2, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-02-01	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 03-01-2018
Descripción: Comprobar que la función incorporada en la clase UsuarioController se pueda llamar en la aplicación	
Condiciones de Ejecución: <ul style="list-style-type: none">• La clase UsuarioController• La vista debe llamar registro	
Pasos de ejecución: <ul style="list-style-type: none">• Abrir la aplicación• Abrir Usuario	
Resultado esperado: No debe salir ningún error.	
Evaluación: Exitosa.	

Tarea de Ingeniería 2, HU-02 Modificar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-02 Modificar Usuario	
Número de Tarea: TI_02	Nombre de Tarea: crear la función getUsers en la clase usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 19-01-2018	Fecha Fin: 03-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la función para que liste todos los usuarios	
Pruebas de Aceptación <ul style="list-style-type: none">• Verificar que se llame a la tabla Usuario en la clase Usuario• Verificar si la consulta funciona.	

Tarea de Ingeniería 2- PA 1, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-01-02	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar que se llame a la tabla Usuario en la clase Usuario	
Responsable: Carolina Valencia	Fecha: 03-01-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla Usuario para poder realizar el registro de Usuario.	
Condiciones de Ejecución: <ul style="list-style-type: none">• La clase Usuario ya debe existir	
Pasos de ejecución: <ul style="list-style-type: none">• Abrir el Módulo Usuario• Abrir la carpeta src• Abrir la carpeta Usuario• Abrir la carpeta Model	

<ul style="list-style-type: none"> • Abrir la carpeta Entity • Abrir la clase Usuario
Resultado esperado: Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla Usuario
Evaluación: Exitosa.

Tarea de Ingeniería 2- PA 2, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-02-02	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar si la consulta funciona.	
Responsable: Carolina Valencia	Fecha: 03-01-2018
Descripción: Comprobar que la consulta permita mostrar todos los usuarios registrados	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener conectado con la base de datos • Tener la consulta de modificar 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la aplicación • Ingresar la página de registrar usuario 	
Resultado: Mostrar los datos de los usuarios registrados	
Evaluación: Exitosa.	

Tarea de Ingeniería 3, HU-02 Modificar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-02 Modificar Usuario	
Número de Tarea: TI_03	Nombre de Tarea: Implementar la interfaz para listar todos usuarios
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 19-01-2018	Fecha Fin: 03-01-2018
Programador Responsable: Carolina Valencia	
Descripción: implementar la interfaz gráfica para la historia de usuario modificar usuario	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado. • Verificar que se muestre los usuarios. 	

Tarea de Ingeniería 3- PA 1, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-01-03	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Responsable: Carolina Valencia	Fecha: 03-01-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La interfaz gráfica de la funcionalidad está implementada. 	
Pasos de ejecución:	

<ul style="list-style-type: none"> • Ejecución de las interfaces. • Comprobar el estándar.
Resultado esperado: La interfaz gráfica de la funcionalidad cumple con los estándares preestablecidos.
Evaluación: Exitosa.

Tarea de Ingeniería 3- PA 2, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-02-03	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar que se muestre los usuarios	
Responsable: Carolina Valencia	Fecha: 03-01-2018
Descripción: Comprobar que en la interfaz se muestre todos los usuarios	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad debe estar implementada con su interfaz gráfica. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecución de la funcionalidad. • Se ingresar al Menú Persona • Dar clic en Gestión Persona 	
Resultado esperado: Se debe mostrar los usuarios que están registrados	
Evaluación: Exitosa.	

Tarea de Ingeniería 4, HU-02 Modificar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-02 Modificar Usuario	
Número de Tarea: TI_04	Nombre de Tarea: crear la función modificarAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 04-01-2018	Fecha Fin: 04-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la modificarAction para poder realizar la modificación del Usuario	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que cumpla con la estructura • Verificar si la aplicación puede llamar a esa función 	

Tarea de Ingeniería 4- PA 1, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-01-04	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 04-01-2018
Descripción: Comprobar que la función que está en la clase UsuarioController está bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase UsuarioController 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Usuario 	

<ul style="list-style-type: none"> • Abrir la carpeta src • Abrir la carpeta Usuario • Abrir la carpeta Controller • Abrir la carpeta UsuarioController
Resultado esperado: Debe estar la función modificarAction y que retorne la nueva vista.
Evaluación: Exitosa.

Tarea de Ingeniería 4- PA 2, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-02-04	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 04-01-2018
Descripción: Comprobar que la función incorporada en la clase UsuarioController se pueda llamar en la aplicación	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La clase UsuarioController • La vista debe llamar registro 	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir la aplicación • Abrir Usuario • Dar clic registrar Usuario 	
Resultado esperado: No debe salir ningún error.	
Evaluación: Exitosa.	

Tarea de Ingeniería 5, HU-02 Modificar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-02 Modificar Usuario	
Número de Tarea: TI_05	Nombre de Tarea: crear la función updateUsuario en la clase usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22-01-2018	Fecha Fin: 04-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la función para realizar la modificación del Usuario	
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que se llame a la tabla Usuario en la clase Usuario • Verificar si la consulta funciona. 	

Tarea de Ingeniería 5-PA 1, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-01-05	Historia de Usuario: HU-01 Ingresar Usuario
Nombre: Verificar que se llame a la tabla Usuario en la clase Usuario	
Responsable: Carolina Valencia	Fecha: 04-01-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla Usuario para poder realizar el registro de Usuario.	
Condiciones de Ejecución:	

<ul style="list-style-type: none"> • La clase Usuario ya debe existir
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el Módulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario • Abrir la carpeta Model • Abrir la carpeta Entity • Abrir la clase Usuario
Resultado esperado: Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla Usuario
Evaluación: Exitosa.

Tarea de Ingeniería 5-PA 2, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-02-05	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar si la consulta funciona.	
Responsable: Carolina Valencia	Fecha: 04-01-2018
Descripción: Comprobar que la consulta permita modificar la información del usuario	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener la conexión de la base de datos • Tener la función para modificar 	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir la aplicación • Dar clic en Gestión Usuario • se ingresa en la función de modificar los datos del usuario • Dar clic en Modificar 	
Resultado esperado: Debe modificarse los datos del usuario	
Evaluación: Exitosa.	

Tarea de Ingeniería 6, HU-02 Modificar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-02 Modificar Usuario	
Número de Tarea: TI_06	Nombre de Tarea: Implementar la interfaz para modificar usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22-01-2018	Fecha Fin: 04-01-2018
Programador Responsable: Carolina Valencia	
Descripción: implementar la interfaz gráfica para la historia de usuario modificar usuario	
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado. • Verificar que se haya modificado correctamente el usuario. 	

Tarea de Ingeniería 6-PA 1, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-01-06	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Responsable: Carolina Valencia	Fecha: 04-01-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La interfaz gráfica de la funcionalidad está implementada. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecución de las interfaces. • Comprobar el estándar. 	
Resultado esperado:	
La interfaz gráfica de la funcionalidad cumple con los estándares preestablecidos.	
Evaluación: Exitosa.	

Tarea de Ingeniería 6-PA 2, HU-02 Modificar Usuario

Prueba de aceptación	
Código: PA-02-06	Historia de Usuario: HU-02 Modificar Usuario
Nombre: Verificar que se haya modificado correctamente el usuario.	
Responsable: Carolina Valencia	Fecha: 04-01-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad debe estar implementada con su interfaz gráfica. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecución de la funcionalidad. • Se ingresa a la pantalla ingresar usuario • Se presenta la pantalla de registrar usuario • Se ingresa los datos: correo, contraseña. • Se presiona en Guardar. • Debe mostrar un mensaje de éxito o de error 	
Resultado esperado:	
Los datos ingresados estén registrados en la Base de datos	
Evaluación: Exitosa.	

HU-03 Eliminar Usuario

HISTORIA DE USUARIO	
Número: HU-03	Nombre: Eliminar Usuario
Modificación historia de usuario:	
Usuario: Paciente	Sprint Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 16
Riesgo en desarrollo: Media	Puntos Reales: 16
Descripción:	
Como administrador me gustaría poder desactivar a los usuarios que ya se dejen de atender en la odontología	
Observaciones:	
Pruebas de Aceptación:	

- Si se eliminó el usuario se mostrará un mensaje se Eliminó con éxito.
- No debe aparecer en gestión Usuario

Prueba de Aceptación 1, HU-03 Eliminar Usuario

Prueba de aceptación	
Código: PA_01	Historia de Usuario: HU-03 Eliminar Usuario
Nombre: Si se eliminó el usuario se mostrará un mensaje se Eliminó con éxito.	
Responsable: Carolina Valencia	Fecha: 08-01-2018
Descripción: Comprobar que se haya eliminado correctamente el usuario	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Usuario • Muestra la lista de los usuarios • Dar clic eliminar 	
Resultado esperado:	
Verificar que se muestre el mensaje de éxito	
Evaluación: Exitosa.	

Prueba de Aceptación 2, HU-03 Eliminar Usuario

Prueba de aceptación	
Código: PA_02	Historia de Usuario: HU-03 Eliminar Usuario
Nombre: No debe aparecer en gestión Usuario	
Responsable: Carolina Valencia	Fecha: 08-01-2018
Descripción: Comprobar que no se liste el usuario que se elimino	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Usuario • Muestra la lista de los usuarios • Dar clic eliminar 	
Resultado esperado:	
No debe aparecer en la lista de gestión Usuario	
Evaluación: Exitosa.	

TAREAS DE INGENIERIA

Tarea de Ingeniería 1, HU-03 Eliminar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-03 Eliminar Usuario	
Número de Tarea: TI_01	Nombre de Tarea: crear la función eliminarAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 05-01-2018	Fecha Fin: 05-01-2018
Programador Responsable: Carolina Valencia	

Descripción: se debe crear la eliminarAction para eliminar usuario
Pruebas de Aceptación
<ul style="list-style-type: none"> • Verificar que cumpla con la estructura • Verificar si la aplicación puede llamar a esa función

Tarea de Ingeniería 1-PA 1, HU-03 Eliminar Usuario

Prueba de aceptación	
Código: PA-01-01	Historia de Usuario: HU-03 Eliminar Usuario
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 05-01-2018
Descripción: Comprobar que la función que está en la clase UsuarioController está bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase UsuarioController 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario • Abrir la carpeta Controller • Abrir la carpeta UsuarioController 	
Resultado esperado:	
Debe estar la función eliminarAction y que retorne la nueva vista.	
Evaluación: Exitosa.	

Tarea de Ingeniería 1-PA 2, HU-03 Eliminar Usuario

Prueba de aceptación	
Código: PA-02-01	Historia de Usuario: HU-03 Eliminar Usuario
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 05-01-2018
Descripción: Comprobar que la función incorporada en la clase UsuarioController se pueda llamar en la aplicación	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase UsuarioController • La vista debe llamar registrar 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la aplicación • Abrir Usuario 	
Resultado esperado:	
No debe salir ningún error.	
Evaluación: Exitosa.	

Tarea de Ingeniería 2, HU-03 Eliminar Usuario

TAREA DE INGENIERÍA	
Historia de Usuario: HU-03 Eliminar Usuario	
Número de Tarea: TI_02	Nombre de Tarea: crear la función eliminacionlogicaUsuario en la clase usuario

Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 05-01-2018	Fecha Fin: 05-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la función para poder eliminar el usuario	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se llame a la tabla Usuario en la clase Usuario • Verificar si la consulta funciona. 	

Tarea de Ingeniería 2-PA 1, HU-03 Eliminar Usuario

Prueba de aceptación	
Código: PA-01-02	Historia de Usuario: HU-03 Eliminar Usuario
Nombre: Verificar que se llame a la tabla Usuario en la clase Usuario	
Responsable: Carolina Valencia	Fecha: 05-01-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla Usuario para poder realizar el registro de Usuario.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase Usuario ya debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Usuario • Abrir la carpeta src • Abrir la carpeta Usuario • Abrir la carpeta Model • Abrir la carpeta Entity • Abrir la clase Usuario 	
Resultado esperado:	
Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla Usuario	
Evaluación: Exitosa.	

Tarea de Ingeniería 2-PA 2, HU-03 Eliminar Usuario

Prueba de aceptación	
Código: PA-02-02	Historia de Usuario: HU-03 Eliminar Usuario
Nombre: Verificar si la consulta funciona.	
Responsable: Carolina Valencia	Fecha: 05-01-2018
Descripción: Comprobar que se elimine el usuario	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase Usuario debe existir • La vista debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Usuario • Muestra la lista de los usuarios • Dar clic eliminar 	
Resultado esperado:	
No debe aparecer el usuario eliminado.	
Evaluación: Exitosa.	

HU-10 Ingresar Persona

HISTORIA DE USUARIO	
Número: HU-10	Nombre: Ingresar Persona
Modificación historia de usuario:	
Usuario: Administrador	Sprint Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 16
Riesgo en desarrollo: Alta	Puntos Reales: 16
Descripción: Como administrador quiera tener el control de las personas que se van haciendo los tratamientos respectivos.	
Observaciones:	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Si se ingresa una persona debe mostrar un mensaje de éxito • No se debe ingresar datos en blancos • Si se llena el formulario y da clic en cancelar no debe ingresar la información 	

Prueba de Aceptación 1, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA_01	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Si se registrar todos los campos del formulario y se procede a guardar se mostrará un mensaje de éxito	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que se guarde los datos de la persona	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Persona debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Persona • Dar clic en gestión de Paciente • Da clic en el icono de registrar Persona • Se presenta los campos: Selecciona el correo, Cedula, Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Fecha de Nacimiento, Selecciona el Sexo y Establecimiento. • Se presiona en Guardar. 	
Resultado esperado: Debe registrar el usuario y mostrar el mensaje de éxito.	
Evaluación: Exitosa.	

Prueba de Aceptación 2, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA_02	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Si se registra datos de manera duplicada nos aparecerá un mensaje usuario ya existe.	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que no se guarde información duplicado	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Persona debe estar implementada. 	

<ul style="list-style-type: none"> • Debe estar conectada con la base de datos
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Persona • Da clic en el icono de registrar Persona • Se presenta los campos: Correo, contraseña • Se presiona en Guardar.
Resultado esperado: No debe guardar la información y debe salir un mensaje de error.
Evaluación: Exitosa.

Prueba de Aceptación 3, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA_03	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Si los datos se ingresan de manera errónea como el correo nos mostrara un mensaje de error de correo para que siga el formato.	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar de que se cumpla el formato del correo.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Persona debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Persona • Dar clic en gestión de Paciente • Da clic en el icono de registrar Persona • Se presenta los campos: Selecciona el correo, Cedula, Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Fecha de Nacimiento, Seleccione el Sexo y Establecimiento. • Se presiona en Guardar. 	
Resultado esperado: Debe mostrar un mensaje que debe seguir el formato que tiene el correo	
Evaluación: Exitosa.	

Prueba de Aceptación 4, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA_04	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Si existen campos vacíos nos mostrará un mensaje de llene todos los campos.	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que no se guarde campos vacíos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Usuario debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Persona • Dar clic en gestión de Paciente • Da clic en el icono de registrar Persona 	

<ul style="list-style-type: none"> • Se presenta los campos: Selecciona el correo, Cedula, Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Fecha de Nacimiento, Seleccione el Sexo y Establecimiento. • Se presiona en Guardar.
Resultado esperado: Debe mostrar un mensaje que se debe llenar todos los campos.
Evaluación: Exitosa.

TAREAS DE INGENIERIA

Tarea de Ingeniería 1, HU-10 Ingresar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-10 Ingresar Persona	
Número de Tarea: TI_01	Nombre de Tarea: crear el módulo Persona
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18-01-2018	Fecha Fin: 18-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe realizar el módulo persona con su respectiva estructura	
(Reverso)Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Verificar que la estructura este bien establecida • Verificar que el módulo este incorporado en la configuración de la aplicación • Verificar que el archivo modulo.php y la configuración del módulo estén bien estructurado para que funcione el controlador. 	

Tarea de Ingeniería 1 PA 1, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-01-01	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que la estructura este bien establecida	
Responsable: Carolina Valencia	Fecha: 18-01-2018
Descripción: Comprobar que el módulo de persona esté bien estructurado para su buen funcionamiento.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo de persona ya debe existir. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la carpeta Modulo de Persona • Verificar si están las carpetas: src, config y view • Verificar si están los archivos: module.config.php y module.php 	
Resultado esperado: Las carpetas y archivos del módulo existan.	
Evaluación: Exitosa.	

Tarea de Ingeniería 1 PA 2, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-02-01	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que el módulo este incorporado en la configuración de la aplicación.	
Responsable: Carolina Valencia	Fecha: 18-01-2018

Descripción: Comprobar que el módulo de usuario este en la lista de la configuración de la aplicación
Condiciones de Ejecución: <ul style="list-style-type: none"> El módulo de persona ya debe existir.
Pasos de ejecución: <ul style="list-style-type: none"> Ingresar la carpeta autoload de la aplicación Ingresar a la carpeta application.config.php
Resultado esperado: Debe estar el nombre del módulo en la lista de la aplicación.
Evaluación: Exitosa.

Tarea de Ingeniería 1 PA 3, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-03-01	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que el archivo modulo.php y la configuración del módulo estén bien estructurado para que funcione el controlador.	
Responsable: Carolina Valencia	Fecha: 18-01-2018
Descripción: Comprobar que los archivos del módulo persona estén bien estructurado	
Condiciones de Ejecución: <ul style="list-style-type: none"> El módulo de persona ya debe existir. Los archivos modulo.php y module.config.php ya debe existir 	
Pasos de ejecución: <ul style="list-style-type: none"> Ingresar al archivo modulo.php Que contenga las librerías, creada la clase del módulo con sus respectivas funciones como getConfig y getAutoloaderConfi. Ingresar a la carpeta config Abrir el archivo module.config.php debe invocar el controlador Usuario y que este el view manager. 	
Resultado esperado: Debe estar ambos archivos configurados correctamente	
Evaluación: Exitosa.	

Tarea de Ingeniería 2, HU-10 Ingresar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-10 Ingresar Persona	
Número de Tarea: TI_02	Nombre de Tarea: Crear el controlador de Persona
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18-01-2018	Fecha Fin: 18-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe realizar el módulo persona con su respectiva estructura	
Pruebas de Aceptación <ul style="list-style-type: none"> Verificar que el controlador este bien estructurada Verificar que se muestre en la aplicación que existe el controlador 	

Tarea de Ingeniería 2 PA 1, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-01-02	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que el controlador Persona esté bien estructurada	
Responsable: Carolina Valencia	Fecha: 18-01-2018
Descripción: Comprobar que este la carpeta del controlador y el archivo controlador persona	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo de persona ya debe existir. • Debe estar la carpeta del controlador y su respectivo archivo 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al módulo Persona • Abrir la carpeta src • Abrir la carpeta Persona y después abrir la carpeta Controller • Abrir el archivo PersonaController • Debe contener el namespace del archivo con su ruta, deben estar las clases adapter, AbstractActionController, ViewModel, etc. • Debe coincidir el nombre del archivo con la clase y debe estar extendido con la clase AbstractActionController. 	
Resultado esperado:	
Debe estar el archivo PersonaController estructurado correctamente	
Evaluación: Exitosa.	

Tarea de Ingeniería 2 PA 2, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-02-02	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que se muestre en la aplicación que existe el controlador	
Responsable: Carolina Valencia	Fecha: 18-01-2018
Descripción: Comprobar que la aplicación invoque al controlador	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El controlador persona debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Persona 	
Resultado esperado:	
Debe mostrar un mensaje que no existe la vista, eso significa que está funcionando el controlador.	
Evaluación: Exitosa.	

Tarea de Ingeniería 3, HU-10 Ingresar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-10 Ingresar Persona	
Número de Tarea: TI_03	Nombre de Tarea: crear la vista del módulo Persona
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 19-01-2018	Fecha Fin: 19-01-2018
Programador Responsable: Carolina Valencia	

Descripción: se debe realizar la vista del módulo Persona para poder saber si la estructura en general funciona.
Pruebas de Aceptación
<ul style="list-style-type: none"> • Verificar que el controlador invoque a la vista • Verificar que se muestre en la aplicación la vista de prueba

Tarea de Ingeniería 3 PA 1, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-01-03	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que el controlador invoque a la vista	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que el controlador llame a la vista ya creada.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir una vista 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la carpeta modulo Persona • Abrir la carpeta src • Abrir la carpeta Persona y luego Controller • Abrir el archivo PersonaController • Observar que este la función que viene creado por defecto 	
Resultado esperado:	
Debe estar invocado la vista	
Evaluación: Exitosa.	

Tarea de Ingeniería 3 PA 2, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-02-03	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que se muestre en la aplicación la vista de prueba	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la aplicación muestre la vista de prueba	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe estar el nombre del módulo en la lista de módulos de la aplicación • Debe existir tanto el módulo persona como el controlador y la vista. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Persona 	
Resultado esperado:	
Debe mostrar la vista de prueba con el mensaje “Este es la vista de prueba”	
Evaluación: Exitosa.	

Tarea de Ingeniería 4, HU-10 Ingresar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-10 Ingresar Persona	
Número de Tarea: TI_04	Nombre de Tarea: crear la clase persona
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 19-01-2018	Fecha Fin: 19-01-2018

Programador Responsable: Carolina Valencia
Descripción: se debe crear la clase usuario para poder realizar las diferentes funcionalidades que va a tener la persona.
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que la clase Persona esté bien estructurada • Verificar que el controlador invoque a la clase

Tarea de Ingeniería 4 PA 1, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-01-04	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que la clase Persona esté bien estructurada	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la clase Usuario este bien estructurada para su buen funcionamiento	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Debe estar creado la clase Persona en la carpeta Model 	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir la carpeta Modulo Persona • Debe tener su namespace y las respectivas clases para su funcionamiento • La clase debe estar extendido Gateway • Debe tener su adaptador para llamar a la tabla Persona de la Base de datos 	
Resultado esperado: Debe estar configurado la clase Persona.	
Evaluación: Exitosa.	

Tarea de Ingeniería 4 PA 2, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-02-04	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que el controlador invoque a la clase	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la aplicación muestre la vista de prueba	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Debe existir la clase Persona en la clase Model 	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el módulo Persona • Abrir la carpeta src • Abrir el archivo PersonaController 	
Resultado esperado: Debe mostrar que se invoque la clase Persona	
Evaluación: Exitosa.	

Tarea de Ingeniería 5, HU-10 Ingresar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-10 Ingresar Persona	
Número de Tarea: TI_05	Nombre de Tarea: crear la clase y la vista del formulario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1

Fecha Inicio: 19-01-2018	Fecha Fin: 19-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la clase formulario para poder realizar el registro de persona	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que la estructura de la clase formulario este bien • Verificar que en el controlador Persona llame la clase formulario 	

Tarea de Ingeniería 5 PA 1, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-01-05	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que la estructura de la clase formulario este bien	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la clase formulario este bien estructurado para su buen funcionamiento.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir la clase formulario en la carpeta Form 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el módulo Persona • Abrir la carpeta src • Abrir la carpeta Persona • Abrir la carpeta Form • Abrir el archivo Formulario 	
Resultado esperado:	
Debe tener el namespace, llamar a las clases AdapterInterface, Element, Form, Captcha, Factory y por último debe estar extendido Form.	
Evaluación: Exitosa.	

Tarea de Ingeniería 5 PA 2, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-02-05	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que en el controlador Usuario llame la clase formulario	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la clase PersonaController llame a la clase formulario para que la vista pueda utilizar el formulario.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir la clase formulario en la carpeta Form 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el módulo Persona • Abrir la carpeta src • Abrir la carpeta Persona • Abrir la carpeta Controller • Abrir el archivo PersonaController 	
Resultado esperado:	
Debe llamar la clase Formulario.	
Evaluación: Exitosa.	

Tarea de Ingeniería 6, HU-10 Ingresar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-10 Ingresar Persona	
Número de Tarea: TI_06	Nombre de Tarea: crear la función registroAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 19-01-2018	Fecha Fin: 19-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la registroAction para poder realizar el registro de la persona.	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que cumpla con la estructura • Verificar si la aplicación puede llamar a esa función 	

Tarea de Ingeniería 6 PA 1, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-01-06	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la función que está en la clase PersonaController está bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase PersonaController 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Persona • Abrir la carpeta src • Abrir la carpeta Persona • Abrir la carpeta Controller • Abrir la carpeta PersonaController 	
Resultado esperado:	
Debe estar la función registroAction y que retorne la nueva vista.	
Evaluación: Exitosa.	

Tarea de Ingeniería 6 PA 2, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-02-06	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la función incorporada en la clase PersonaController se pueda llamar en la aplicación	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase PersonaController • La vista debe llamar registro 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la aplicación • Abrir Persona • Dar clic en el icono registrar 	

Resultado esperado: No debe salir ningún error.
Evaluación: Exitosa.

Tarea de Ingeniería 7, HU-10 Ingresar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-10 Ingresar Persona	
Número de Tarea: TI_07	Nombre de Tarea: crear la función de registrar persona en la clase persona
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 19-01-2018	Fecha Fin: 19-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la función de registrar persona.	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se llame a la tabla Persona en la clase Persona • Verificar si la consulta funciona. 	

Tarea de Ingeniería 7 PA 1, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-01-07	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que se llame a la tabla Persona en la clase Persona	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla Persona para poder realizar el registro de Persona.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase Persona ya debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Persona • Abrir la carpeta src • Abrir la carpeta Persona • Abrir la carpeta Model • Abrir la carpeta Entity • Abrir la clase Persona 	
Resultado esperado:	
Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla Persona	
Evaluación: Exitosa.	

Tarea de Ingeniería 7 PA 2, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-02-07	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar si la consulta funciona.	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la consulta permita registrar la información de la Persona.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase Persona debe existir 	

<ul style="list-style-type: none"> • La vista debe existir
Pasos de ejecución: <ul style="list-style-type: none"> • En el PersonaControlador debe llamar la función addpersona • Enviar parámetros de la persona
Resultado esperado: Debe estar los datos ingresados en la base de datos
Evaluación: Exitosa.

Tarea de Ingeniería 8, HU-10 Ingresar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-10 Ingresar Persona	
Número de Tarea: TI_08	Nombre de Tarea: Implementar la interfaz para el registro de usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 09-02-2018	Fecha Fin: 19-01-2018
Programador Responsable: Carolina Valencia	
Descripción: implementar la interfaz gráfica para la historia de usuario ingresar persona	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado. • Verificar que se haya ingresado correctamente la persona. 	

Tarea de Ingeniería 8 PA 1, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-01-08	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Responsable: Carolina Valencia	Fecha: 09-02-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La interfaz gráfica de la funcionalidad está implementada. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecución de las interfaces. • Comprobar el estándar. 	
Resultado esperado: La interfaz gráfica de la funcionalidad cumple con los estándares preestablecidos.	
Evaluación: Exitosa.	

Tarea de Ingeniería 8 PA 2, HU-10 Ingresar Persona

Prueba de aceptación	
Código: PA-02-08	Historia de Usuario: HU-10 Ingresar Persona
Nombre: Verificar que se haya ingresado correctamente la persona.	
Responsable: Carolina Valencia	Fecha: 19-01-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad debe estar implementada con su interfaz gráfica. 	

Pasos de ejecución: <ul style="list-style-type: none"> • Ejecución de la funcionalidad. • Se ingresa a la pantalla ingresar Persona • Se presenta la pantalla de registrar persona • Se ingresa los datos: Seleccionar el correo, cédula, Nombre, segundo Nombre, primer apellido, segundo apellido, fecha de nacimiento, seleccione el sexo y establecimiento. • Se presiona en Guardar. • Debe mostrar un mensaje de éxito o de error
Resultado esperado: Los datos ingresados estén registrados en la Base de datos
Evaluación: Exitosa.

HU-11 Modificar Persona

HISTORIA DE USUARIO	
Número: HU-11	Nombre: Modificar Persona
Modificación historia de usuario:	
Usuario: Paciente	Sprint Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 16
Riesgo en desarrollo: Alta	Puntos Reales: 16
Descripción: Como administrador me gustaría que puedan modificar los datos de la persona para poder tener un registro con la información actualizada.	
Observaciones: No podrá modificar el correo	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Si se modifica los datos de la persona se mostrará un mensaje se Modificó con éxito. • Si se modifica y deja campos en blanco el sistema mostrará un mensaje de error indicando que se llene todos los campos. • Si no modifica los datos de la persona y presiona cancelar se mostrará los datos del usuario sin modificar. 	

Prueba de Aceptación 1, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA_01	Historia de Usuario: HU-11 Modificar Persona
Nombre: Si se modifica los datos de la persona se mostrará un mensaje se Modificó con éxito.	
Responsable: Carolina Valencia	Fecha: 23-01-2018
Descripción: Comprobar que se haya modificado los datos de la persona	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Persona debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Persona • Seleccionar Gestión Persona • Muestra la lista de los pacientes registrados • Dar clic modificar 	

<ul style="list-style-type: none"> • Se presenta el campo contraseña: cambiamos el campo de Sexo y el primer apellido. • Se presiona en Guardar.
Resultado esperado: Debe modificarse los datos del usuario y mostrar el mensaje de éxito.
Evaluación: Exitosa.

Prueba de Aceptación 2, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA_02	Historia de Usuario: HU-11 Modificar Persona
Nombre: Si se modifica y deja campos en blanco el sistema mostrará un mensaje de error indicando que se llene todos los campos.	
Responsable: Carolina Valencia	Fecha: 23-01-2018
Descripción: Comprobar que no se guarde información en blanco	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Persona debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Persona • Seleccionar Gestión Persona • Muestra la lista de los pacientes • Dar clic modificar al paciente que se ingreso • Se presenta todos los campos del paciente y dejamos en blanco la cédula • Se presiona en Guardar. 	
Resultado esperado: Debe mostrar un mensaje de error	
Evaluación: Exitosa.	

Prueba de Aceptación 3, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA_03	Historia de Usuario: HU-11 Modificar Persona
Nombre: Si no modifica los datos de la persona y presiona cancelar se mostrará los datos de la persona sin modificar.	
Responsable: Carolina Valencia	Fecha: 23-01-2018
Descripción: Comprobar que si cancela la modificación no debe cambiar nada de la información.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Modificar Persona debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Persona • Seleccionar Gestión de Persona • Muestra la lista de los pacientes • Dar clic modificar al cliente que se ingreso • Se presenta los campos y dejamos en blanco el nombre del paciente • Se presiona en Cancelar 	
Resultado esperado: No se debe alterar la información del paciente.	

Evaluación: Exitosa.

Tarea de Ingeniería 1, HU-11 Modificar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-11 Modificar Persona	
Número de Tarea: TI_01	Nombre de Tarea: crear la función gestionpersonaAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22-01-2018	Fecha Fin: 22-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la gestionpersonaAction para que liste todos los Pacientes y poder modificar sus datos	
Pruebas de Aceptación	
<ul style="list-style-type: none">• Verificar que cumpla con la estructura• Verificar si la aplicación puede llamar a esa función	

Tarea de Ingeniería 1 PA 1, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-01-01	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 22-01-2018
Descripción: Comprobar que la función que está en la clase PersonaController está bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none">• La clase PersonaController	
Pasos de ejecución:	
<ul style="list-style-type: none">• Abrir el Módulo Persona• Abrir la carpeta src• Abrir la carpeta Persona• Abrir la carpeta Controller• Abrir la carpeta PersonaController	
Resultado esperado: Debe estar la función gestionpersonaAction y que retorne la nueva vista.	
Evaluación: Exitosa.	

Tarea de Ingeniería 1 PA 2, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-02-01	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 22-01-2018
Descripción: Comprobar que la función incorporada en la clase PersonaController se pueda llamar en la aplicación	
Condiciones de Ejecución:	
<ul style="list-style-type: none">• La clase PersonaController• La vista debe llamar registro	
Pasos de ejecución:	

<ul style="list-style-type: none"> • Abrir la aplicación • Abrir Persona
Resultado esperado: No debe salir ningún error.
Evaluación: Exitosa.

Tarea de Ingeniería 2, HU-11 Modificar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-11 Modificar Persona	
Número de Tarea: TI_02	Nombre de Tarea: crear la función getPersona en la clase persona
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22-01-2018	Fecha Fin: 22-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la función para que liste todos las personas o pacientes	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se llame a la tabla Persona en la clase Persona • Verificar si la consulta funciona. 	

Tarea de Ingeniería 2 PA 1, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-01-02	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar que se llame a la tabla Persona en la clase Persona	
Responsable: Carolina Valencia	Fecha: 12-02-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla Persona para poder realizar el registro de Persona.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase Persona ya debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Persona • Abrir la carpeta src • Abrir la carpeta Persona • Abrir la carpeta Model • Abrir la carpeta Entity • Abrir la clase Persona 	
Resultado esperado: Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla Persona	
Evaluación: Exitosa.	

Tarea de Ingeniería 2 PA 2, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-02-02	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar si la consulta funciona.	
Responsable: Carolina Valencia	Fecha: 22-01-2018
Descripción: Comprobar que la consulta permita mostrar todas las personas registradas	

Condiciones de Ejecución: <ul style="list-style-type: none"> Tener conectado con la base de datos Tener la consulta de modificar
Pasos de ejecución: <ul style="list-style-type: none"> Abrir la aplicación Ingresar la página de registrar personas
Resultado: Mostrar los datos de las personas registrados
Evaluación: Exitosa.

Tarea de Ingeniería 3, HU-11 Modificar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-11 Modificar Persona	
Número de Tarea: TI_03	Nombre de Tarea: Implementar la interfaz para listar todas personas
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22-01-2018	Fecha Fin: 22-01-2018
Programador Responsable: Carolina Valencia	
Descripción: implementar la interfaz gráfica para la historia de usuario modificar Persona	
Pruebas de Aceptación <ul style="list-style-type: none"> Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado. Verificar que se haya modificado correctamente la persona. 	

Tarea de Ingeniería 3 PA 1, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-01-03	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Responsable: Carolina Valencia	Fecha: 12-02-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Condiciones de Ejecución: <ul style="list-style-type: none"> La interfaz gráfica de la funcionalidad está implementada. 	
Pasos de ejecución: <ul style="list-style-type: none"> Ejecución de las interfaces. Comprobar el estándar. 	
Resultado esperado: La interfaz gráfica de la funcionalidad cumple con los estándares preestablecidos.	
Evaluación: Exitosa.	

Tarea de Ingeniería 3 PA 2, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-02-03	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar que se haya modificado correctamente la persona.	
Responsable: Carolina Valencia	Fecha: 22-01-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	

Condiciones de Ejecución: <ul style="list-style-type: none"> La funcionalidad debe estar implementada con su interfaz gráfica.
Pasos de ejecución: <ul style="list-style-type: none"> Dar clic en Menú de Persona Seleccionar Gestión de Persona Se lista los pacientes registrados Dar clic en modificar uno de los registros Modificar el Nombre del Paciente Se presiona en Guardar. Debe mostrar un mensaje de éxito o de error
Resultado esperado: Que se haya modificado el nombre del paciente.
Evaluación: Exitosa.

Tarea de Ingeniería 4, HU-11 Modificar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-11 Modificar Persona	
Número de Tarea: TI_04	Nombre de Tarea: crear la función modificarAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23-01-2018	Fecha Fin: 23-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la modificarAction para poder realizar la modificación de la Persona	
Pruebas de Aceptación	
<ul style="list-style-type: none"> Verificar que cumpla con la estructura Verificar si la aplicación puede llamar a esa función 	

Tarea de Ingeniería 4 PA 1, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-01-04	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 23-01-2018
Descripción: Comprobar que la función que está en la clase PersonaController está bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> La clase PersonaController 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Abrir el Módulo Persona Abrir la carpeta src Abrir la carpeta Persona Abrir la carpeta Controller Abrir la carpeta PersonaController 	
Resultado esperado: Debe estar la función modificarAction y que retorne la nueva vista.	
Evaluación: Exitosa.	

Tarea de Ingeniería 4 PA 2, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-02-04	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 23-01-2018
Descripción: Comprobar que la función incorporada en la clase PersonaController se pueda llamar en la aplicación	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase PersonaController • La vista debe llamar modificar 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la aplicación • Dar clic en menú de Persona • Seleccionar Gestión de Persona • Se lista los pacientes • Dar clic en modificar 	
Resultado esperado:	
No debe salir ningún error.	
Evaluación: Exitosa.	

Tarea de Ingeniería 5, HU-11 Modificar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-11 Modificar Persona	
Número de Tarea: TI_05	Nombre de Tarea: crear la función updatePersona en la clase usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23-01-2018	Fecha Fin: 23-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la función para realizar la modificación del paciente	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se llame a la tabla Persona en la clase Persona 	

Tarea de Ingeniería 5 PA 1, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-01-05	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar que se llame a la tabla persona en la clase Persona	
Responsable: Carolina Valencia	Fecha: 23-01-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla Persona para poder realizar la modificación del paciente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase Persona ya debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Persona • Abrir la carpeta src • Abrir la carpeta Persona • Abrir la carpeta Model 	

<ul style="list-style-type: none"> • Abrir la carpeta Entity • Abrir la clase Persona
Resultado esperado: Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla Persona
Evaluación: Exitosa.

Tarea de Ingeniería 6, HU-11 Modificar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-11 Modificar Persona	
Número de Tarea: TI_06	Nombre de Tarea: Implementar la interfaz para modificar persona
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23-01-2018	Fecha Fin: 23-01-2018
Programador Responsable: Carolina Valencia	
Descripción: implementar la interfaz gráfica para la historia de usuario modificar persona	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado. • Verificar que se haya modificado los datos del paciente 	

Tarea de Ingeniería 6 PA 1, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-01-06	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Responsable: Carolina Valencia	Fecha: 23-01-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La interfaz gráfica de la funcionalidad está implementada. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecución de las interfaces. • Comprobar el estándar. 	
Resultado esperado:	
La interfaz gráfica de la funcionalidad cumple con los estándares preestablecidos.	
Evaluación: Exitosa.	

Tarea de Ingeniería 6 PA 2, HU-11 Modificar Persona

Prueba de aceptación	
Código: PA-02-06	Historia de Usuario: HU-11 Modificar Persona
Nombre: Verificar que se haya modificado los datos del paciente	
Responsable: Carolina Valencia	Fecha: 23-01-2018
Descripción: comprobar que los datos se hayan modificado correctamente	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir la función • Debe existir la conexión de la base de datos 	
Pasos de ejecución:	

<ul style="list-style-type: none"> • Ejecución de la aplicación • Dar clic al menú de Persona • Seleccionar Gestión de Persona • Muestra la lista de los pacientes • Seleccionar un paciente registrado • Se modifica el establecimiento y Nombres del paciente • Se presiona en Guardar • Debe mostrar un mensaje de éxito o de error
Resultado esperado: Verificar los datos de la persona se haya modificado
Evaluación: Exitosa.

HU-12 Eliminar Persona

HISTORIA DE USUARIO	
Número: HU-12	Nombre: Eliminar Persona
Modificación historia de usuario:	
Usuario: Paciente	Sprint Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 8
Riesgo en desarrollo: Alta	Puntos Reales: 8
Descripción: Como administrador me gustaría poder desactivar a las que ya se dejen de atender en la odontología	
Observaciones:	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Si se eliminó la persona se mostrará un mensaje se Eliminó con éxito. • No debe aparecer en gestión Persona 	

Prueba de Aceptación 1, HU-12 Eliminar Persona

Prueba de aceptación	
Código: PA_01	Historia de Usuario: HU-12 Eliminar Persona
Nombre: Si se eliminó la persona se mostrará un mensaje se Eliminó con éxito.	
Responsable: Carolina Valencia	Fecha: 24-01-2018
Descripción: Comprobar que se haya eliminado correctamente la persona	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar Paciente debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Paciente • Dar clic en Gestión Paciente • Muestra la lista de los pacientes • Dar clic eliminar 	
Resultado esperado: Verificar que se muestre el mensaje de éxito	
Evaluación: Exitosa.	

Prueba de Aceptación 2, HU-12 Eliminar Persona

Prueba de aceptación	
Código: PA_02	Historia de Usuario: HU-12 Eliminar Persona
Nombre: No debe aparecer en gestión Persona	
Responsable: Carolina Valencia	Fecha: 24-01-2018
Descripción: Comprobar que no se liste el paciente que se elimino	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Eliminar Paciente debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Paciente • Dar clic en Gestión Paciente • Muestra la lista de los pacientes • Dar clic eliminar 	
Resultado esperado:	
No debe aparecer en la lista de gestión Paciente	
Evaluación: Exitosa.	

TAREAS DE INGENIERÍA

Tarea de Ingeniería 1, HU-12 Eliminar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-12 Eliminar Persona	
Número de Tarea: TI_01	Nombre de Tarea: crear la función eliminarAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24-01-2018	Fecha Fin: 24-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la eliminarAction para eliminar paciente	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que cumpla con la estructura • Verificar si la aplicación puede llamar a esa función 	

Tarea de Ingeniería 1 PA 1, HU-12 Eliminar Persona

Prueba de aceptación	
Código: PA-01-01	Historia de Usuario: HU-12 Eliminar Persona
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 24-01-2018
Descripción: Comprobar que la función que está en la clase PersonaController está bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase PersonaController 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo persona • Abrir la carpeta src • Abrir la carpeta Persona • Abrir la carpeta Controller • Abrir la carpeta PersonaController 	

Resultado esperado: Debe estar la función eliminarAction y que retorne la nueva vista.
Evaluación: Exitosa.

Tarea de Ingeniería 1 PA 2, HU-12 Eliminar Persona

Prueba de aceptación	
Código: PA-02-01	Historia de Usuario: HU-12 Eliminar Persona
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 24-01-2018
Descripción: Comprobar que la función incorporada en la clase PersonaController se pueda llamar en la aplicación	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase PersonaController • La vista debe llamar registrar 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la aplicación • Abrir Persona 	
Resultado esperado: No debe salir ningún error.	
Evaluación: Exitosa.	

Tarea de Ingeniería 2, HU-12 Eliminar Persona

TAREA DE INGENIERÍA	
Historia de Usuario: HU-12 Eliminar Persona	
Número de Tarea: TI_02	Nombre de Tarea: crear la función eliminacionlogicaPersona en la clase usuario
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24-01-2018	Fecha Fin: 24-01-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la función para poder eliminar la persona	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se llame a la tabla Persona en la clase Persona 	

Tarea de Ingeniería 2 PA 1, HU-12 Eliminar Persona

Prueba de aceptación	
Código: PA-01-02	Historia de Usuario: HU-12 Eliminar Persona
Nombre: Verificar que se llame a la tabla Persona en la clase Persona	
Responsable: Carolina Valencia	Fecha: 24-01-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla persona para poder eliminar persona.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase Persona ya debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Persona • Abrir la carpeta src • Abrir la carpeta Persona 	

<ul style="list-style-type: none"> • Abrir la carpeta Model • Abrir la carpeta Entity • Abrir la clase Persona
Resultado esperado: Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla Persona
Evaluación: Exitosa.

HU-25 Ingresar Tratamiento

HISTORIA DE USUARIO	
Número: HU-25	Nombre: Ingresar Tratamiento
Modificación historia de usuario:	
Usuario: Administrador	Sprint Asignada: 6
Prioridad en Negocio: Media	Puntos Estimados: 16
Riesgo en desarrollo: Alta	Puntos Reales: 16
Descripción: Como Administrador quiero tener registrado cada tratamiento que se va realizando los pacientes	
Observaciones:	
Pruebas de Aceptación: <ul style="list-style-type: none"> • Si se ingresa un nuevo tratamiento debe mostrar un mensaje de éxito • No se debe ingresar datos en blancos • Si se llena el formulario y da clic en cancelar no debe ingresar la información 	

Prueba de Aceptación 1, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA_01	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Si se ingresa un nuevo tratamiento debe mostrar un mensaje de éxito	
Responsable: Carolina Valencia	Fecha: 23-02-2018
Descripción: Comprobar que la información se ingrese correctamente	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La funcionalidad Ingresar tratamiento debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al menú Paciente • Clic en Gestión Paciente • Seleccionar al paciente que desea añadir información • Dar clic en el icono de ingresar tratamiento • Se presenta los campos: • Se presiona Guardar 	
Resultado esperado: Debe guardarse el tratamiento del paciente seleccionado y Salir el mensaje de éxito.	
Evaluación: Exitosa.	

Prueba de Aceptación 2, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA_02	Historia de Usuario: HU-25 Ingresa Tratamiento
Nombre: No se debe ingresar datos en blancos	
Responsable: Carolina Valencia	Fecha: 23-02-2018
Descripción: Comprobar que no se guarde información en blanco	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar tratamiento debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Paciente • Clic en Gestión Paciente • Seleccionar al paciente que desea añadir información • Dar clic en el icono de ingresar tratamiento • Se presenta los campos: • Se presiona Guardar 	
Resultado esperado:	
Mostrar mensaje de campos obligatorio	
Evaluación: Exitosa.	

Prueba de Aceptación 3, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA_03	Historia de Usuario: HU-25 Ingresa Tratamiento
Nombre: Si se llena el formulario y da clic en cancelar no debe ingresar la información	
Responsable: Carolina Valencia	Fecha: 23-02-2018
Descripción: Comprobar que el botón cancelar funcione y no se guarde los datos	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar tratamiento debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Paciente • Clic en Gestión Paciente • Seleccionar al paciente que desea añadir información • Dar clic en el icono de ingresar tratamiento • Se presenta los campos: • Se presiona en Cancelar 	
Resultado esperado:	
Debe regresar a la página de datos de la información del paciente	
Evaluación: Exitosa.	

Tarea de Ingeniería 1, HU-25 Ingresar Tratamiento

TAREA DE INGENIERÍA	
Historia de Usuario: HU-25 Ingresa Tratamiento	
Número de Tarea: TI_01	Nombre de Tarea: crear el módulo tratamiento
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22-02-2018	Fecha Fin: 22-02-2018

Programador Responsable: Carolina Valencia
Descripción: se debe realizar el módulo tratamiento con su respectiva estructura
(Reverso)Pruebas de Aceptación: <ul style="list-style-type: none"> • Verificar que la estructura este bien establecida • Verificar que el módulo este incorporado en la configuración de la aplicación • Verificar que el archivo modulo.php y la configuración del módulo estén bien estructurado para que funcione el controlador.

Tarea de Ingeniería 1 PA 1, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-01-01	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que la estructura este bien establecida	
Responsable: Carolina Valencia	Fecha: 22-02-2018
Descripción: Comprobar que el módulo tratamiento este bien estructurada para su buen funcionamiento.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • El módulo tratamiento ya debe existir. 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar a la carpeta Módulo Tratamiento • Verificar si están las carpetas: src, config y view • Verificar si están los archivos: module.config.php y module.php 	
Resultado esperado: Las carpetas y archivos del módulo existan.	
Evaluación: Exitosa.	

Tarea de Ingeniería 1 PA 2, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-02-01	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que el módulo este incorporado en la configuración de la aplicación.	
Responsable: Carolina Valencia	Fecha: 22-02-2018
Descripción: Comprobar que el módulo tratamiento este en la lista de la configuración de la aplicación	
Condiciones de Ejecución: <ul style="list-style-type: none"> • El módulo tratamiento ya debe existir. 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar la carpeta autoload de la aplicación • Ingresar a la carpeta application.config.php 	
Resultado esperado: Debe estar el nombre del módulo en la lista de la aplicación.	
Evaluación: Exitosa.	

Tarea de Ingeniería 1 PA 3, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-03-01	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que el archivo modulo.php y la configuración del módulo estén bien estructurado para que funcione el controlador.	

Responsable: Carolina Valencia	Fecha: 22-02-2018
Descripción: Comprobar que los archivos del módulo tratamiento estén bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo tratamiento ya debe existir. • Los archivos modulo.php y module.config.php ya debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al archivo modulo.php • Que contenga las librerías, creada la clase del módulo con sus respectivas funciones como getConfig y getAutoloaderConfi. • Ingresar a la carpeta config • Abrir el archivo module.config.php • debe invocar el controlador CatalogoMestro y que este el view manager. 	
Resultado esperado:	
Debe estar ambos archivos configurados correctamente	
Evaluación: Exitosa.	

Tarea de Ingeniería 2, HU-25 Ingresar Tratamiento

TAREA DE INGENIERÍA	
Historia de Usuario: HU-25 Ingresar Tratamiento	
Número de Tarea: TI_02	Nombre de Tarea: Crear el controlador de TratamientoController
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22-02-2018	Fecha Fin: 22-02-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe realizar el módulo tratamiento con su respectiva estructura	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que el controlador este bien estructurada • Verificar que se muestre en la aplicación que existe el controlador 	

Tarea de Ingeniería 2 PA 1, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-01-02	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que el controlador tratamiento este bien estructurada	
Responsable: Carolina Valencia	Fecha: 16-03-2018
Descripción: Comprobar que este la carpeta del controlador y el archivo controlador tratamiento	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo Tratamiento ya debe existir. • Debe estar la carpeta del controlador y su respectivo archivo 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al módulo Tratamiento • Abrir la carpeta src • Abrir la carpeta Tratamiento y después abrir la carpeta Controller • Abrir el archivo Tratamientocontroller • Debe contener el namespace del archivo con su ruta, deben estar las clases adapter, AbstractActionController, ViewModel, etc. 	

<ul style="list-style-type: none"> • Debe coincidir el nombre del archivo con la clase y debe estar extendido con la clase ActionController.
Resultado esperado: Debe estar el archivo Tratamientocontroller estructurado correctamente
Evaluación: Exitosa.

Tarea de Ingeniería 2 PA 2, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-02-02	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que se muestre en la aplicación que existe el controlador	
Responsable: Carolina Valencia	Fecha: 16-03-2018
Descripción: Comprobar que la aplicación invoque al controlador	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El controlador Tratamiento debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Catalogo Paciente • Dar clic en gestión Paciente • Seleccionar el paciente • Dar clic en Añadir datos informativos • Dar clic en el icono de ingresar tratamiento 	
Resultado esperado: Debe mostrar un mensaje que no existe la vista, eso significa que está funcionando el controlador.	
Evaluación: Exitosa.	

Tarea de Ingeniería 3, HU-25 Ingresar Tratamiento

TAREA DE INGENIERÍA	
Historia de Usuario: HU-25 Ingresar Tratamiento	
Número de Tarea: TI_03	Nombre de Tarea: crear la vista del módulo Tratamiento
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 16-03-2018	Fecha Fin: 16-03-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe realizar la vista del módulo Tratamiento para poder saber si la estructura en general funciona.	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que el controlador invoque a la vista • Verificar que se muestre en la aplicación la vista de prueba 	

Tarea de Ingeniería 3 PA 1, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-01-03	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que el controlador invoque a la vista	
Responsable: Carolina Valencia	Fecha: 22-02-2018
Descripción: Comprobar que el controlador llame a la vista ya creada.	
Condiciones de Ejecución:	

<ul style="list-style-type: none"> • Debe existir una vista
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir la carpeta módulo Tratamiento • Abrir la carpeta src • Abrir la carpeta Tratamiento y luego Controller • Abrir el archivo TratamientoController • Observar que este la función que viene creado por defecto
Resultado esperado: Debe estar invocado la vista
Evaluación: Exitosa.

Tarea de Ingeniería 3 PA 2, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-02-03	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que se muestre en la aplicación la vista de prueba	
Responsable: Carolina Valencia	Fecha: 16-03-2018
Descripción: Comprobar que la aplicación muestre la vista de prueba	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Debe estar el nombre del módulo en la lista de módulos de la aplicación • Debe existir tanto el módulo tratamiento como el controlador y la vista. 	
Pasos de ejecución: <ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Paciente • Seleccionar Gestión paciente • Se lista los pacientes • Dar clic en añadir información de uno de los pacientes que se lista • Dar clic en el icono de Tratamiento 	
Resultado esperado: Debe mostrar la vista de prueba con el mensaje “Este es la vista de prueba”	
Evaluación: Exitosa.	

Tarea de Ingeniería 4, HU-25 Ingresar Tratamiento

TAREA DE INGENIERÍA	
Historia de Usuario: HU-25 Ingresar Tratamiento	
Número de Tarea: TI_04	Nombre de Tarea: crear la clase Diagnostico
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22-02-2018	Fecha Fin: 22-02-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la clase tratamiento para poder realizar las diferentes funcionalidades.	
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que la clase diagnostico este bien estructurada • Verificar que el controlador invoque a la clase 	

Tarea de Ingeniería 4 PA 1, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-01-04	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que la clase tratamiento este bien estructurada	
Responsable: Carolina Valencia	Fecha: 22-02-2018
Descripción: Comprobar que la clase tratamiento este bien estructurada para su buen funcionamiento	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe estar creado la clase tratamiento en la carpeta Model 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la carpeta Módulo Tratamiento • Debe tener su namespace y las respectivas clases para su funcionamiento • La clase debe estar extendido Gateway • Debe tener su adaptador para llamar a la tabla tratamiento de la Base de datos 	
Resultado esperado:	
Debe estar configurado la clase Tratamiento	
Evaluación: Exitosa.	

Tarea de Ingeniería 4 PA 2, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-02-04	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que el controlador invoque a la clase	
Responsable: Carolina Valencia	Fecha: 22-02-2018
Descripción: Comprobar que la aplicación muestre la vista de prueba	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir la clase tratamiento en la clase Model 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el módulo Tratamiento • Abrir la carpeta src • Abrir el archivo TratamientoController 	
Resultado esperado:	
Debe mostrar que se invoque la clase Tratamiento	
Evaluación: Exitosa.	

Tarea de Ingeniería 5, HU-25 Ingresar Tratamiento

TAREA DE INGENIERÍA	
Historia de Usuario: HU-25 Ingresar Tratamiento	
Número de Tarea: TI_05	Nombre de Tarea: crear la función registroAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23-02-2018	Fecha Fin: 23-02-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la registroAction para poder realizar el registro tratamiento	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que cumpla con la estructura • Verificar si la aplicación puede llamar a esa función 	

Tarea de Ingeniería 5 PA 1, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-01-05	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 23-02-2018
Descripción: Comprobar que la función que está en la clase TratamientoController está bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase tratamientoController 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Tratamiento • Abrir la carpeta src • Abrir la carpeta Tratamiento • Abrir la carpeta Controller • Abrir la carpeta TratamientoController 	
Resultado esperado:	
Debe estar la función registroAction y que retorne la nueva vista.	
Evaluación: Exitosa.	

Tarea de Ingeniería 5 PA 2, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-02-05	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 23-02-2018
Descripción: Comprobar que la función incorporada en la clase TratamientoController se pueda llamar en la aplicación	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase TratamientoController • La vista debe llamar registro 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Paciente • Seleccionar Gestión paciente • Se lista los pacientes • Dar clic en añadir información de uno de los pacientes que se lista • Dar clic en el icono de Tratamiento 	
Resultado esperado:	
No debe salir ningún error.	
Evaluación: Exitosa.	

Tarea de Ingeniería 6, HU-25 Ingresar Tratamiento

TAREA DE INGENIERÍA	
Historia de Usuario: HU-25 Ingresar Tratamiento	
Número de Tarea: TI_06	Nombre de Tarea: crear la función de registrar diagnostico en la clase Diagnostico
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23-02-2018	Fecha Fin: 23-02-2018

Programador Responsable: Carolina Valencia
Descripción: se debe crear la función de registrar Tratamiento
Pruebas de Aceptación
<ul style="list-style-type: none"> • Verificar que se llame a la tabla hca_tratamiento en la clase Tratamiento

Tarea de Ingeniería 6 PA 1, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-01-06	Historia de Usuario: HU-25 Ingresa Tratamiento
Nombre: Verificar que se llame a la tabla hca_Tratamiento en la clase Tratamiento	
Responsable: Carolina Valencia	Fecha: 23-02-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla hca_tratamiento para poder ingresar la información de Tratamiento	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La clase Tratamiento ya debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir el Módulo Tratamiento • Abrir la carpeta src • Abrir la carpeta Tratamiento • Abrir la carpeta Model • Abrir la carpeta Entity • Abrir la clase Tratamiento 	
Resultado esperado:	
Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla Tratamiento	
Evaluación: Exitosa.	

Tarea de Ingeniería 7, HU-25 Ingresar Tratamiento

TAREA DE INGENIERÍA	
Historia de Usuario: HU-25 Ingresa Tratamiento	
Número de Tarea: TI_07	Nombre de Tarea: Implementar la interfaz para el registro de Tratamiento
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23-02-2018	Fecha Fin: 23-02-2018
Programador Responsable: Carolina Valencia	
Descripción: implementar la interfaz gráfica para el ingreso de Tratamiento	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado. • Verificar que se haya ingresado correctamente el tratamiento 	

Tarea de Ingeniería 7 PA 1, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-01-07	Historia de Usuario: HU-25 Ingresa Tratamiento
Nombre: Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Responsable: Carolina Valencia	Fecha: 23-02-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	

Condiciones de Ejecución:
<ul style="list-style-type: none"> • La interfaz gráfica de la funcionalidad está implementada.
Pasos de ejecución:
<ul style="list-style-type: none"> • Ejecución de las interfaces. • Comprobar el estándar.
Resultado esperado:
La interfaz gráfica de la funcionalidad cumple con los estándares pre-establecidos.
Evaluación: Exitosa.

Tarea de Ingeniería 7 PA 2, HU-25 Ingresar Tratamiento

Prueba de aceptación	
Código: PA-02-07	Historia de Usuario: HU-25 Ingresar Tratamiento
Nombre: Verificar que se haya ingresado correctamente el tratamiento	
Responsable: Carolina Valencia	Fecha: 23-02-2018
Descripción: Comprobar que se ingrese la información del tratamiento del paciente	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad debe estar implementada con su interfaz gráfica. • Conexión a la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Paciente • Seleccionar Gestión paciente • Se lista los pacientes • Dar clic en añadir información de uno de los pacientes que se lista • Dar clic en el icono de Tratamiento • Ingresar datos • Se presiona en Guardar. • Debe mostrar un mensaje de éxito o de error 	
Resultado esperado:	
Los datos ingresados estén registrados en la Base de datos	
Evaluación: Exitosa.	

HU-39 Ingresar Indicador Enfermedad

HISTORIA DE USUARIO	
Número: HU-39	Nombre: Ingresar Indicador Enfermedad
Modificación historia de usuario:	
Usuario: Administrador	Sprint Asignada: 9
Prioridad en Negocio: Alta	Puntos Estimados: 16
Riesgo en desarrollo: Alta	Puntos Reales: 16
Descripción:	
Como Administrador quiero que cada paciente tenga el Indicador Enfermedad de cada diente	
Observaciones:	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Si se ingresa el indicador de enfermedad del Paciente debe mostrar un mensaje de éxito • No se debe ingresar datos en blancos • Si se llena el formulario y da clic en cancelar no debe ingresar la información 	

Prueba de aceptación 1, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA_01	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Si se ingresa el indicador de enfermedad del Paciente debe mostrar un mensaje de éxito	
Responsable: Carolina Valencia	Fecha: 24-04-2018
Descripción: Comprobar que la información se ingrese correctamente	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar Indicador Enfermedad debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Paciente • Clic en Gestión Paciente • Seleccionar al paciente que desea añadir información • Dar clic en el icono de ingresar indicador enfermedad • Se presenta los campos: • Se presiona Guardar 	
Resultado esperado:	
Debe guardarse el indicador Enfermedad del paciente seleccionado y Salir el mensaje de éxito.	
Evaluación: Exitosa.	

Prueba de aceptación 2, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA_02	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: No se debe ingresar datos en blancos	
Responsable: Carolina Valencia	Fecha: 24-04-2018
Descripción: Comprobar que no se guarde información en blanco	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar indicador enfermedad debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Paciente • Clic en Gestión Paciente • Seleccionar al paciente que desea añadir información • Dar clic en el icono de ingresar indicador enfermedad • Se presenta los campos: • Se presiona Guardar 	
Resultado esperado:	
Mostrar mensaje de campos obligatorio	
Evaluación: Exitosa.	

Prueba de aceptación 3, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA_03	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Si se llena el formulario y da clic en cancelar no debe ingresar la información	

Responsable: Carolina Valencia	Fecha: 24-04-2018
Descripción: Comprobar que el botón cancelar funcione y no se guarde los datos	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad Ingresar indicador enfermedad debe estar implementada. • Debe estar conectada con la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al menú Paciente • Clic en Gestión Paciente • Seleccionar al paciente que desea añadir información • Dar clic en el icono de ingresar indicador enfermedad • Se presenta los campos: • Se presiona en Cancelar 	
Resultado esperado:	
Debe regresar a la página de datos de la información del paciente	
Evaluación: Exitosa.	

Tarea de Ingeniería 1, HU-39 Ingresar Indicador Enfermedad

TAREA DE INGENIERÍA	
Historia de Usuario: HU-39 Ingresar Indicador Enfermedad	
Número de Tarea: TI_01	Nombre de Tarea: crear el módulo Indicadorenerfermedad
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23-04-2018	Fecha Fin: 23-04-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe realizar el módulo indicador enfermedad con su respectiva estructura	
(Reverso)Pruebas de Aceptación:	
<ul style="list-style-type: none"> • Verificar que la estructura este bien establecida • Verificar que el módulo este incorporado en la configuración de la aplicación • Verificar que el archivo modulo.php y la configuración del módulo estén bien estructurado para que funcione el controlador. 	

Tarea de Ingeniería 1 PA 1, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-01-01	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que la estructura este bien establecida	
Responsable: Carolina Valencia	Fecha: 23-04-2018
Descripción: Comprobar que el módulo indicador enfermedad este bien estructurada para su buen funcionamiento.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo de indicador enfermedad ya debe existir. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la carpeta Modulo de Indicador enfermedad • Verificar si están las carpetas: src, config y view • Verificar si están los archivos: module.config.php y module.php 	
Resultado esperado:	
Las carpetas y archivos del módulo existan.	
Evaluación: Exitosa.	

Tarea de Ingeniería 1 PA 2, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-02-01	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que el módulo este incorporado en la configuración de la aplicación.	
Responsable: Carolina Valencia	Fecha: 23-04-2018
Descripción: Comprobar que el módulo de indicador enfermedad este en la lista de la configuración de la aplicación	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> El módulo de indicador enfermedad ya debe existir. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar la carpeta autoload de la aplicación Ingresar a la carpeta application.config.php 	
Resultado esperado:	
Debe estar el nombre del módulo en la lista de la aplicación.	
Evaluación: Exitosa.	

Tarea de Ingeniería 1 PA 3, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-03-01	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que el archivo modulo.php y la configuración del módulo estén bien estructurado para que funcione el controlador.	
Responsable: Carolina Valencia	Fecha: 23-04-2018
Descripción: Comprobar que los archivos del módulo indicador enfermedad estén bien estructurado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> El módulo de indicador enfermedad ya debe existir. Los archivos modulo.php y module.config.php ya debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar al archivo modulo.php Que contenga las librerías, creada la clase del módulo con sus respectivas funciones como getConfig y getAutoloaderConfi. Ingresar a la carpeta config Abrir el archivo module.config.php debe invocar el controlador Indicadorenermedad y que este el view manager. 	
Resultado esperado:	
Debe estar ambos archivos configurados correctamente	
Evaluación: Exitosa.	

Tarea de Ingeniería 2, HU-39 Ingresar Indicador Enfermedad

TAREA DE INGENIERÍA	
Historia de Usuario: HU-39 Ingresar Indicador Enfermedad	
Número de Tarea: TI_02	Nombre de Tarea: Crear el controlador de IndicadorenermedadController
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23-04-2018	Fecha Fin: 23-04-2018
Programador Responsable: Carolina Valencia	

Descripción: se debe realizar el módulo Indicador Enfermedad con su respectiva estructura
Pruebas de Aceptación
<ul style="list-style-type: none"> • Verificar que el controlador este bien estructurada • Verificar que se muestre en la aplicación que existe el controlador

Tarea de Ingeniería 2 PA 1, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-01-02	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que el controlador Indicador enfermedad este bien estructurada	
Responsable: Carolina Valencia	Fecha: 23-04-2018
Descripción: Comprobar que este la carpeta del controlador y el archivo controlador Indicador Enfermedad	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El módulo de diagnóstico ya debe existir. • Debe estar la carpeta del controlador y su respectivo archivo 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al módulo Indicador Enfermedad • Abrir la carpeta src • Abrir la carpeta IndicadorEnfermedad y después abrir la carpeta Controller • Abrir el archivo IndicadorenermedadController • Debe contener el namespace del archivo con su ruta, deben estar las clases adapter, AbstractActionController, ViewModel, etc. • Debe coincidir el nombre del archivo con la clase y debe estar extendido con la clase AbstractActionController. 	
Resultado esperado:	
Debe estar el archivo IndicadorenermedadController estructurado correctamente	
Evaluación: Exitosa.	

Tarea de Ingeniería 2 PA 2, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-02-02	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que se muestre en la aplicación que existe el controlador	
Responsable: Carolina Valencia	Fecha: 23-04-2018
Descripción: Comprobar que la aplicación invoque al controlador	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • El controlador Indicador Enfermedad debe existir 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Catalogo Paciente • Dar clic en gestión Paciente • Seleccionar el paciente • Dar clic en Añadir datos informativos • Dar clic en el icono de ingresar Indicador Enfermedad 	
Resultado esperado:	
Debe mostrar un mensaje que no existe la vista, eso significa que está funcionando el controlador.	
Evaluación: Exitosa.	

Tarea de Ingeniería 3, HU-39 Ingresar Indicador Enfermedad

TAREA DE INGENIERÍA	
Historia de Usuario: HU-39 Ingresar Indicador Enfermedad	
Número de Tarea: TI_03	Nombre de Tarea: crear la vista del módulo Indicador Enfermedad
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23-04-2018	Fecha Fin: 23-04-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe realizar la vista del módulo Indicador Enfermedad para poder saber si la estructura en general funciona.	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que el controlador invoque a la vista • Verificar que se muestre en la aplicación la vista de prueba 	

Tarea de Ingeniería 3 PA 1, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-01-03	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que el controlador invoque a la vista	
Responsable: Carolina Valencia	Fecha: 23-04-2018
Descripción: Comprobar que el controlador llame a la vista ya creada.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir una vista 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la carpeta módulo Indicador Enfermedad • Abrir la carpeta src • Abrir la carpeta Indicadorenenfermedad y luego Controller • Abrir el archivo IndicadorenenfermedadController • Observar que este la función que viene creado por defecto 	
Resultado esperado:	
Debe estar invocado la vista	
Evaluación: Exitosa.	

Tarea de Ingeniería 3 PA 2, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-02-03	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que se muestre en la aplicación la vista de prueba	
Responsable: Carolina Valencia	Fecha: 23-04-2018
Descripción: Comprobar que la aplicación muestre la vista de prueba	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe estar el nombre del módulo en la lista de módulos de la aplicación • Debe existir tanto el módulo Indicador Enfermedad como el controlador y la vista. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Paciente • Seleccionar Gestión paciente • Se lista los pacientes 	

<ul style="list-style-type: none"> • Dar clic en añadir información de uno de los pacientes que se lista • Dar clic en el icono de Indicador Enfermedad
Resultado esperado: Debe mostrar la vista de prueba con el mensaje “Este es la vista de prueba”
Evaluación: Exitosa.

Tarea de Ingeniería 4, HU-39 Ingresar Indicador Enfermedad

TAREA DE INGENIERÍA	
Historia de Usuario: HU-39 Ingresar Indicador Enfermedad	
Número de Tarea: TI_04	Nombre de Tarea: crear la clase Indicador Enfermedad
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24-04-2018	Fecha Fin: 24-04-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la clase Indicador Enfermedad para poder realizar las diferentes funcionalidades.	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que la clase Indicador Enfermedad este bien estructurada • Verificar que el controlador invoque a la clase 	

Tarea de Ingeniería 4 PA 1, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-01-04	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que la clase Indicador Enfermedad este bien estructurada	
Responsable: Carolina Valencia	Fecha: 24-04-2018
Descripción: Comprobar que la clase Indicador Enfermedad este bien estructurada para su buen funcionamiento	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe estar creado la clase Indicador Enfermedad en la carpeta Model 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Abrir la carpeta Módulo Indicador Enfermedad • Debe tener su namespace y las respectivas clases para su funcionamiento • La clase debe estar extendido Gateway • Debe tener su adaptador para llamar a la tabla Indicador Enfermedad de la Base de datos 	
Resultado esperado: Debe estar configurado la clase Indicador Enfermedad	
Evaluación: Exitosa.	

Tarea de Ingeniería 4 PA 2, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-02-04	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que el controlador invoque a la clase	
Responsable: Carolina Valencia	Fecha: 24-04-2018
Descripción: Comprobar que la aplicación muestre la vista de prueba	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir la clase Indicador Enfermedad en la clase Model 	

Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el módulo Indicadorenerfermedad • Abrir la carpeta src • Abrir el archivo IndicadorenerfermedadController
Resultado esperado: Debe mostrar que se invoque la clase Indicador Enfermedad
Evaluación: Exitosa.

Tarea de Ingeniería 5, HU-39 Ingresar Indicador Enfermedad

TAREA DE INGENIERÍA	
Historia de Usuario: HU-39 Ingresar Indicador Enfermedad	
Número de Tarea: TI_05	Nombre de Tarea: crear la función registroAction en el controlador
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24-04-2018	Fecha Fin: 24-04-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la registroAction para poder realizar el registro Indicador Enfermedad	
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que cumpla con la estructura • Verificar si la aplicación puede llamar a esa función 	

Tarea de Ingeniería 5 PA 1, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-01-05	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que cumpla con la estructura	
Responsable: Carolina Valencia	Fecha: 24-04-2018
Descripción: Comprobar que la función que está en la clase IndicadorEnfermedadController está bien estructurado	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La clase IndicadorEnfermedadController 	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el Módulo Indicador Enfermedad • Abrir la carpeta src • Abrir la carpeta Indicador Enfermedad • Abrir la carpeta Controller • Abrir la carpeta IndicadorEnfermedadController 	
Resultado esperado: Debe estar la función registroAction y que retorne la nueva vista.	
Evaluación: Exitosa.	

Tarea de Ingeniería 5 PA 2, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-02-05	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar si la aplicación puede llamar a esa función	
Responsable: Carolina Valencia	Fecha: 24-04-2018

Descripción: Comprobar que la función incorporada en la clase IndicadorenenfermedadController se pueda llamar en la aplicación
Condiciones de Ejecución: <ul style="list-style-type: none"> • La clase IndicadorenenfermedadController • La vista debe llamar registro
Pasos de ejecución: <ul style="list-style-type: none"> • Ejecutar la aplicación • Ingresar al menú Paciente • Seleccionar Gestión paciente • Se lista los pacientes • Dar clic en añadir información de uno de los pacientes que se lista • Dar clic en el icono de Indicador Enfermedad
Resultado esperado: No debe salir ningún error.
Evaluación: Exitosa.

Tarea de Ingeniería 6, HU-39 Ingresar Indicador Enfermedad

TAREA DE INGENIERÍA	
Historia de Usuario: HU-39 Ingresar Indicador Enfermedad	
Número de Tarea: TI_06	Nombre de Tarea: crear la función de registrar Indicador Enfermedad en la clase
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24-04-2018	Fecha Fin: 24-04-2018
Programador Responsable: Carolina Valencia	
Descripción: se debe crear la función de registrar Indicador Enfermedad	
Pruebas de Aceptación <ul style="list-style-type: none"> • Verificar que se llame a la tabla hca_indicador_enfermedad en la clase Indicador Enfermedad • Verificar si la consulta funciona. 	

Tarea de Ingeniería 6 PA 1, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-01-06	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que se llame a la tabla hca_indicador_enfermedad en la clase Indicador Enfermedad	
Responsable: Carolina Valencia	Fecha: 24-04-2018
Descripción: Comprobar que la clase este llamando correctamente la tabla hca_indicador_enfermedad para poder ingresar la información de Indicador Enfermedad	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La clase Diagnostico ya debe existir 	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el Módulo Indicador Enfermedad • Abrir la carpeta src • Abrir la carpeta Indicadorenenfermedad • Abrir la carpeta Model • Abrir la carpeta Entity 	

<ul style="list-style-type: none"> • Abrir la clase Indicador enfermedad
Resultado esperado: Debe tener el constructor con el adaptador para que exista la conexión con la base de datos y también el nombre de la tabla hca_Indicador_enfermedad
Evaluación: Exitosa.

Tarea de Ingeniería 7, HU-39 Ingresar Indicador Enfermedad

TAREA DE INGENIERÍA	
Historia de Usuario: HU-39 Ingresar Indicador Enfermedad	
Número de Tarea: TI_07	Nombre de Tarea: Implementar la interfaz para el registro de Indicador Enfermedad
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24-04-2018	Fecha Fin: 24-04-2018
Programador Responsable: Carolina Valencia	
Descripción: implementar la interfaz gráfica para el ingreso de Indicador Enfermedad	
Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado. • Verificar que se haya ingresado correctamente el indicador enfermedad 	

Tarea de Ingeniería 7 PA 1, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-01-07	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Responsable: Carolina Valencia	Fecha: 24-04-2018
Descripción: Comprobar que la interfaz gráfica cumpla con el estándar de interfaces utilizado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La interfaz gráfica de la funcionalidad está implementada. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ejecución de las interfaces. • Comprobar el estándar. 	
Resultado esperado:	
La interfaz gráfica de la funcionalidad cumple con los estándares preestablecidos.	
Evaluación: Exitosa.	

Tarea de Ingeniería 7 PA 2, HU-39 Ingresar Indicador Enfermedad

Prueba de aceptación	
Código: PA-02-07	Historia de Usuario: HU-39 Ingresar Indicador Enfermedad
Nombre: Verificar que se haya ingresado correctamente el indicador enfermedad	
Responsable: Carolina Valencia	Fecha: 24-04-2018
Descripción: Comprobar que se ingrese la información del indicador enfermedad del paciente	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La funcionalidad debe estar implementada con su interfaz gráfica. • Conexión a la base de datos 	
Pasos de ejecución:	

- Ejecutar la aplicación
- Ingresar al menú Paciente
- Seleccionar Gestión paciente
- Se lista los pacientes
- Dar clic en añadir información de uno de los pacientes que se lista
- Dar clic en el icono de Indicador Enfermedad
- Ingresar datos
- Se presiona en Guardar.
- Debe mostrar un mensaje de éxito o de error

Resultado esperado:

Los datos ingresados estén registrados en la Base de datos

Evaluación: Exitosa.

Anexo C. Diccionario de Datos

Tabla hca_usuario

Columna	Tipo	Nulo	Comentario
uso_codigo	Serial	No	Clave única. Identificador de usuario
uso_correo	varchar (50)	No	Correo del usuario
psa_clave	varchar (25)	No	Clave del usuario
psa_tipo	Int	No	Tipo de usuario
uso_usuario_creacion	Int	No	Identificador del usuario de creación
uso_fecha_creacion	Date		Fecha de creación del registro
uso_usuario_modificacion	Int		Identificador del usuario de modificación
uso_fecha_modificacion	Date		Fecha de modificación del registro
uso_eliminado	Boolean	No	Verificar si esta eliminado el registro

Tabla hca_persona

Columna	Tipo	Nulo	Comentario
psa_codigo	Serial	No	Clave única. Identificador de persona
uso_codigo	int	No	Clave foránea. Identificador del usuario
psa_cedula	varchar (10)	No	Cédula de la persona
psa_primer_nombre	varchar (25)	No	Primer nombre de la persona
psa_segundo_nombre	varchar (25)	No	Segundo nombre de la persona
psa_primer_apellido	varchar (25)	No	Primer apellido de la persona
psa_segundo_apellido	varchar (25)	No	Segundo apellido de la persona
psa_fecha_nacimiento	varchar (15)		Fecha de nacimiento de la persona
psasexo	Integer		Sexo de la persona
psa_establecimiento	varchar(25)		Establecimiento que pertenece la persona
psa_usuario_creacion	Integer		Identificador del usuario de creación
psa_fecha_creacion	Date		Fecha de creación del registro
psa_usuario_modificacion	Integer		Identificador del usuario de modificación
psa_fecha_modificacion	Date		Fecha de modificación del registro
psa_eliminado	Boolean	No	Verificar si esta eliminado el registro

Tabla hca_diagnostico_terapeutico

Columna	Tipo	Nulo	Comentario
dto_codigo	Serial	No	Clave única. Identificador del diagnostico
psa_codigo	Int	No	Clave foránea. Identificador de la persona
dto_examen	Integer	No	Identificador de los exámenes
dto_descripcion	varchar (100)		Descripción del diagnóstico terapéutico
dto_usuario_creacion	Integer		Identificador del usuario de creación
dto_fecha_creacion	Date		Fecha de creación del registro

dto_usuario_modificacion	Integer		Identificador del usuario de modificación
dto_fecha_modificacion	Date		Fecha de modificación del registro
dno_eliminado	Boolean	No	Verificar si esta eliminado el registro

Tabla hca_signos_vitales

Columna	Tipo	Nul o	Comentario
svs_codigo	Serial	No	Clave única. Identificador del diagnostico
psa_codigo	Integer	No	Clave foránea. Identificador de la persona
svs_presionarterial	Integer	No	Presión arterial
svs_frecuenciad	Integer	No	Frecuencia cardiaca
svs_temperatura	Integer	No	temperatura
svs_frecuenciar	Integer	No	Frecuencia respiratoria
svs_usuario_creacion	Integer		Identificador del usuario de creación
svs_fecha_creacion	Date		Fecha de creación del registro
svs_usuario_modificacion	Integer		Identificador del usuario de modificación
svs_fecha_modificacion	Date		Fecha de modificación del registro
svs_eliminado	Boolean	No	Verificar si esta eliminado el registro

Tabla hca_examen_estomatognatico

Columna	Tipo	Nulo	Comentario
eeo_codigo	Serial	No	Clave única. Identificador del diagnostico
psa_codigo	Integer	No	Clave foránea. Identificador de la persona
eeo_patalogia	Integer	No	Patología
eeo_valor	Integer	No	Valor de la patología
eeo_descripcion	varchar (100)	No	Descripción
eeo_usuario_creacion	Integer		Identificador del usuario de creación
eeo_fecha_creacion	Date		Fecha de creación del registro
eeo_usuario_modificacion	Integer		Identificador del usuario de modificación
eeo_fecha_modificacion	Date		Fecha de modificación del registro
eeo_eliminado	Boolean	No	Verificar si esta eliminado el registro

Tabla hca_catalogo_maestro

Columna	Tipo	Nul o	Comentario
cmo_codigo_maestro	Serial	No	Clave única. Identificador del diagnostico
cmo_nombre	Varchar (30)	No	Nombre del catálogo maestro
cmo_descripcion	Varchar (50)	No	Descripción del catálogo maestro
cmo_eliminado	Boolean	No	Verificar si esta eliminado el registro

Tabla hca_catalogo_detalle

Columna	Tipo	Nulo	Comentario
cde_codigo_detalle	Serial	No	Clave única. Identificador del diagnostico
cmo_codigo_maestro	Integer	No	Clave foránea. Identificador de la persona
cmo_codigo_negocio	Varchar (30)		Codigo de Negocio
cde_codigo_padre	Integer		Codigo padre
cde_nombre	Varchar (50)	No	Nombre del catalogo detalle
cde_descripcion	Varchar (50)	No	Descripción del catalogo detalle
cde_eliminado	Boolean	No	Verificar si esta eliminado el registro

Tabla hca_catalogo_cita

Columna	Tipo	Nulo	Comentario
cca_codigo	Serial	No	Clave única. Identificador del diagnostico
cca_fecha	Date	No	Fecha
cca_anio	Integer	No	Año del catálogo cita
cca_mes	Integer	No	mes del catálogo cita
cca_dia	Integer	No	día del catálogo cita
cca_estado	Integer	No	Estado del catálogo cita
cca_usuario_creacion	Integer		Identificador del usuario de creación
cca_fecha_creacion	Date		Fecha de creación del registro
cca_usuario_modificacion	Integer		Identificador del usuario de modificación
cca_fecha_modificacion	Date		Fecha de modificación del registro
cca_eliminado	Boolean	No	Verificar si esta eliminado el registro

Anexo D. Tiempo de Respuesta

Tabla para la gestión tradicional, cada tarea se tomó en minutos.

Nro.	Tareas	Tiempo (min)
1	Registrar datos del paciente	4
2	Modificar datos del paciente	4
3	Buscar Paciente	6
4	Registrar Signos vitales	2
5	Modificar Signos vitales	3
6	Buscar Signos vitales	5
7	Registrar el tratamiento del paciente	4
8	Modificar el tratamiento del paciente	2
9	Registrar tratamiento detalle del paciente	4
10	Modificar el tratamiento detalle del paciente	4
11	Buscar Tratamiento detalle del Paciente	6
12	Registrar el Diagnostico del Paciente	4
13	Modificar el Diagnostico del Paciente	4
14	Buscar el diagnóstico del Paciente	6
15	Registrar el Odontograma	4
16	Modificar el Odontograma	3
17	Emitir reporte	11
TOTAL		76
PROMEDIO		4,47

Gestión Automatizada, con la ayuda de la herramienta de Chrome los tiempos que se mostraba es en milisegundos en cada tarea, por lo tanto, el resultado general se le convirtió en minutos.

No	Tarea	Tiempo en Milisegundos
1	Ingresar Paciente	724
2	Modificar Paciente	671
3	Ingresar Signos Vitales	612
4	Modificar Signos Vitales	627
5	Eliminar Signos vitales	568
6	Ingresar Tratamientos	704
7	Modificar Tratamientos	622
8	Eliminar Tratamiento	636
9	Ingresar Tratamiento Detalle	622
10	Modificar Tratamiento Detalle	606
11	Eliminar Tratamiento Detalle	805
12	Ingresar Indicador de Salud	783
13	Modificar Indicador Salud	704
14	Eliminar Indicador Salud	601
15	Ingresar Diagnostico	604
16	Modificar Diagnostico	703
17	Eliminar Diagnostico	808
18	Ingresar Odontograma	637
19	Mostrar Odontograma	630
20	Reporte Materiales	115000

21	Reporte de Pacientes	127000
TOTAL, EN MILISEGUNDOS		254667
PROMEDIO EN MILISEGUNDOS		12127
TOTAL, EN MINUTOS		4,18

Anexo E. Manual de Usuario



CENTRO ODONTOLÓGICO

Dr. Frank Feijoo Apolo

MANUAL DE USUARIO

SYSTEM ODONTOLOGY

HUAQUILLAS-EL ORO-ECUADOR

INTRODUCCIÓN

En este manual de usuario, se explicará paso a paso de los procesos para el funcionamiento de la Aplicación web, que está dedicado para la gestión de la información del centro Odontológico. Es importante tomar en cuenta de este manual para el uso del sistema

OBJETIVO

Guiar al usuario en el uso de la aplicación web del centro odontológico de todas las tareas como: la autenticación, los ingresos de la información, modificaciones, eliminaciones y las realizaciones de reportes.

Dirigido a:

Este manual está dirigido para los usuarios finales como:





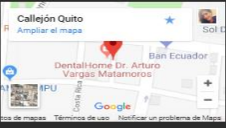
- Administrador
- Pacientes

FUNCIONALIDADES DE LA APLICACIÓN WEB

Página de Inicio

A continuación, se muestra la pantalla de inicio de la aplicación web, se detalla los elementos que contiene como: el encabezado, el menú principal, la información y el pie de página.



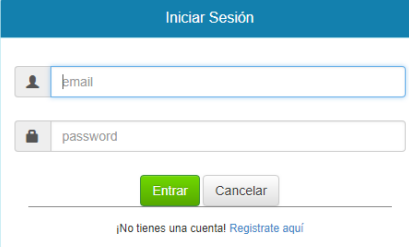
<ul style="list-style-type: none"> • ODONTOLÓGIA • ENDODONCIA • ODONTOLÓGIA PEDIÁTRICA • ESTOMATOLOGÍA • ORTODONCIA • ENDODONCIA 	<p>Hemos diseñado nuestras instalaciones para su máximo confort y periódicamente renovamos nuestros equipos con la tecnología más avanzada.</p>	<p>Dirección: Pasaje Quito E/ Av República y Machala Teléfono: 07-2995786 Ubicación: ECUADOR, EL ORO, HUAQUILLAS</p>
 <p>Diseño de Sonrisa</p>	 <p>Ortodoncia</p>	 <p>Blanqueamiento</p>
<p>Contactos Dirección: Pasaje Quito E/ Av República y Machala Teléfono: 07-2995786 Ubicación: ECUADOR, EL ORO, HUAQUILLAS</p>		
<p>Síguenos:</p> 		
		
<p>© 2005 - 2018 by Zend Technologies Ltd. All rights reserved.</p>		

Formulario de la Autenticación

Para acceder al sistema se debe poner lo siguiente:

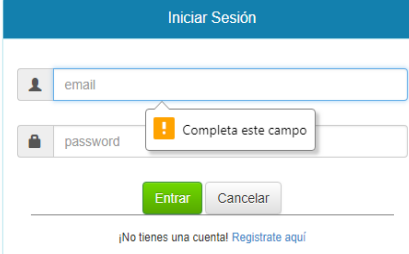
El correo electrónico y su respectiva contraseña

Luego dar clic en Acceder.

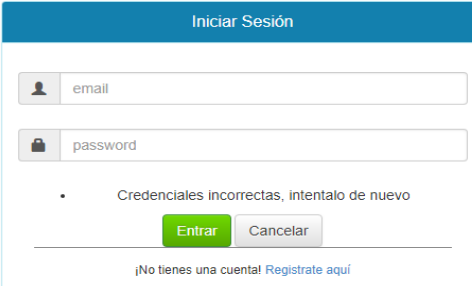


Mensajes

Cuando no se llena los campos sale un mensaje que los campos son obligatorios.



Cuando se ingresa correo electrónico o contraseña sale un mensaje



Para registrar en la aplicación da clic en [¡No tienes una cuenta! Regístrate aquí](#) y se muestra el siguiente formulario.



Registrar

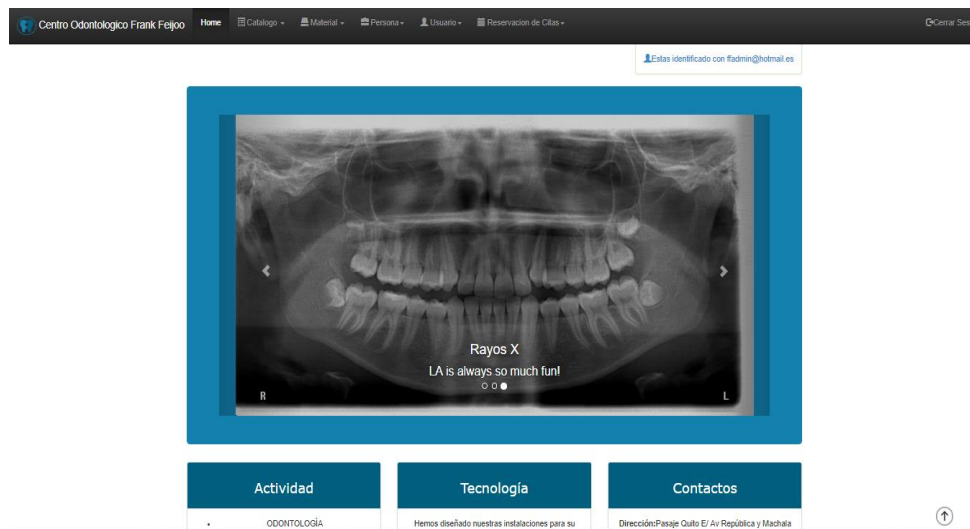
Email: Contraseña:

Cédula: Apellido:

Nombre: Establecimiento:

PANTALLA PRINCIPAL PARA EL ADMINISTRADOR


Cuando se autentifica el Administrador se le presenta esta pantalla de inicio con su respectivo menú



En este menú incluye lo siguiente:

- **Catálogo.** - El catálogo permite administrar todo acerca de las configuraciones que pueden existir ahora o en un futuro como por ejemplo ingreso de profesionales, marcas y categorías de Materiales,
- **Material.** - corresponde a la gestión de la información de los materiales
- **Persona.** - corresponde a la gestión de la información de los pacientes, como sus historias clínicas, diagnósticos, indicadores de Salud, etc.
- **Usuario.** - corresponde a los usuarios que se han registrados.
- **Reservación de Citas.** - corresponde al ingreso de las fechas y horas para que puedan reservar como también observar las reservaciones que realizan los pacientes.

Ingreso de Persona

Se presenta la siguiente interfaz, la cual cuenta los datos de los pacientes que ya están ingresados y cuenta con un buscador, para ingresar nuevo paciente es con el siguiente icono .

Gestión Persona



Buscador							
Filtrar Datos							
ID	CEDULA	PRIMER NOMBRE	PRIMER APELLIDO	ESTABLECIMIENTO	ACCIONES		
1	0704424560	Frank	Fejoo	Odontologia	Añadir Datos Informativos		
3	0704424282	Jonathan	Valencia	OCEAN PRODUCT	Añadir Datos Informativos		
2	0704424290	Evelyn	Valencia	ESPOCH	Añadir Datos Informativos		
7	1707515472	Fausto	Valencia	OCEAN PRODUCT	Añadir Datos Informativos		

Para el ingreso del paciente se muestra la siguiente interfaz.

Añadir Datos de la Persona

Seleccione el Correo:	Cedula
chiquis14@gmail.com	0803713551
Nombre	Segundo Nombre
Giannella	Katherine
Primer Apellido	Segundo Apellido
Montaño	Ortiz
Fecha de Nacimiento:	Seleccione el Sexo:
15/05/1991	Femenino
Establecimiento	
ESPOCH	
Enviar	

Esta interfaz cuenta con los siguientes campos como son: se ingresa la cédula, Nombre, primer apellido, fecha de nacimiento, sexo y establecimiento que son obligatorios mientras los dos campos no son necesarios que se ingrese.


Después de Ingresar los datos del paciente se lista con los pacientes ingresados.

Gestión Persona

Datos de la persona añadido correctamente



Buscador							
Filtrar Datos							
ID	CEDULA	PRIMER NOMBRE	PRIMER APELLIDO	ESTABLECIMIENTO	ACCIONES		
1	0704424560	Frank	Fejoo	Odontologia	Añadir Datos Informativos		
3	0704424282	Jonathan	Valencia	OCEAN PRODUCT	Añadir Datos Informativos		
2	0704424290	Evelyn	Valencia	ESPOCH	Añadir Datos Informativos		
7	1707515472	Fausto	Valencia	OCEAN PRODUCT	Añadir Datos Informativos		
9	0803713551	Giannella	Montaño	ESPOCH	Añadir Datos Informativos		

Para modificar los datos del paciente da clic en el icono  y se presenta la siguiente interfaz, cabe recalcar que la cedula no se modifica.

Modificar Datos de la Persona

Seleccione el Correo: chiquis14@gmail.com		Cedula: 0803713551
Primer Nombre: Giannella	Segundo Nombre: Katherine	
Primer Apellido: Montaño	Segundo Apellido: Ortiz	
Fecha de Nacimiento: 15/05/1991	Seleccione el Sexo: Femenino	
Establecimiento: ESPOCH		
<input type="button" value="Enviar"/>		<input type="button" value="Cancelar"/>



















Para eliminar el paciente, debe dar clic en el icono  y se muestra el mensaje para confirmar si desea eliminar.

¿Estás seguro?

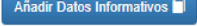
¿Seguro que quieres borrar este elemento?
Si lo borras, nunca podrás recuperarlo.

Se da clic en eliminar, en la interfaz de Gestión de paciente no se debe listar el paciente que se eliminó como se muestra en la siguiente pantalla y se muestra un mensaje de éxito.

Gestión Paciente

ID	CEDULA	PRIMER NOMBRE	PRIMER APELLIDO	ESTABLECIMIENTO	ACCIONES
1	1890548981	Frank	Fejoo	ODONTOLOGIA	<input type="button" value="Añadir Datos Informativos"/>  
4	070442482	Andy	Valencia	OCEAN PRODUCT	<input type="button" value="Añadir Datos Informativos"/>  
3	1809768954	Evelyn	Guzman	ESPOCH	<input type="button" value="Añadir Datos Informativos"/>  
8	0803713551	Giannella	Montaño	ESPOCH	<input type="button" value="Añadir Datos Informativos"/>  
2	0704424290	Evelyn	Valencia	ESPOCH	<input type="button" value="Añadir Datos Informativos"/>  
11	0201809076	Alexander	Betancourt	ESPOCH	<input type="button" value="Añadir Datos Informativos"/>  
12	0604375626	Jordy	Roldan	ESPOCH	<input type="button" value="Añadir Datos Informativos"/>  
10	1804625158	Natalia	Plico	ESPOCH	<input type="button" value="Añadir Datos Informativos"/>  
9	0604532218	Liseth	Aucancela	ESPOCH	<input type="button" value="Añadir Datos Informativos"/>  

Añadir Datos importantes del paciente

Para añadir los datos relevantes del paciente da clic en el botón  , se muestra la siguiente interfaz con su respectiva información, también se tiene le botón para regresar a la pantalla anterior y opciones que se debe ingresar.





Datos del Paciente

DATOS PERSONALES	
Cédula: 070442482	N° Historia Clínica: 4
Nombre: Andy Jonathan Valencia Espinoza	Edad: 27
Establecimiento: OCEAN PRODUCT	Sexo: Masculino




Ingresar Antecedentes personales y familiares


Para ingresar, da clic en el icono  , se presenta la siguiente interfaz, como se puede observar no tiene ingresado datos y se presenta el mensaje que no hay datos ingresados.



Antecedentes Personales y Familiares

No hay Datos ingresados.!!



Para ingresar datos da clic en  , la cual contiene la siguiente información que debe ser ingresada, selecciona las opciones dependiendo la información que el paciente le brinda y da clic en Guardar, si desea para notas importantes se pone una descripción.

Añadir Antecedentes Personales y Familiares

Alergia Antibiótico Alergia Anestesia Hemorragias VIH/Sida Tuberculosis Asma Diabetes Hipertensión Enf. Cardíaca

Descripción:

Enviar Cancelar

Al momento de ingresar la información, muestra el mensaje de éxito y los datos que se ingresó, también contiene las opciones de modificar y eliminar.

Antecedentes Personales y Familiares

Los datos se ha ingresado correctamente


Opciones seleccionados


Asma

descripcion:


 


Ingresar Signos Vitales

Para el ingreso de los signos vitales para el paciente, debe dar clic en el icono  , se presenta la siguiente interfaz, si no contiene información se muestra el mensaje no hay datos ingresados.

 **Gestión de Examen del Sistema Estomatognático**

No hay Datos Ingresados..!!



Para ingresar los datos, da clic en el icono  ,se muestra la siguiente interfaz, la cual contiene los siguientes exámenes que realiza el Doctor, selecciona y en la parte de descripción sale el nombre de la opción que seleccionó y se pone la información correspondiente y da clic en Guardar.

Añadir Examen del Sistema Estomatognático

Labios Mejillas Maxilar Superior Maxilar Inferior Lengua Paladar Piso Carrillos Glándulas Salivales Oro Faringe A.T.M. Ganglios

Descripción:

Al momento de guardar se muestra el mensaje de éxito y también la información ingresada. A continuación, se muestra la siguiente interfaz, con sus respectivas opciones de modificar y eliminar.

Gestión de Examen del Sistema Estomatognático



Los datos se ha ingresado correctamente

Opciones seleccionados

Maxilar Inferior Lengua Paladar

descripcion:

34. Maxilar Inferior
35. Lengua
36. Paladar


Ingresar Exámenes Realizados

Para el ingreso los exámenes que el Doctor le envió realizarse, si el paciente no tiene datos y se muestra la siguiente interfaz con su respectivo mensaje.



Gestion de Diagnostico Terapeutico

No hay Datos ingresados.!!



Para ingresar el diagnostico, debe dar clic en el icono  y se muestra la siguiente interfaz, se selecciona las opciones que se muestra, se pone una descripción y da clic en Enviar.

Añadir Diagnosticos

Biometria Química Sanguínea Rayos- x Otros

Descripcion

se le envio a realizar estos e:

Al momento de guardar la información se muestra un mensaje de éxito y se muestra la información que se ingresó.



Gestion de Diagnostico Terapeutico


Diagnostico Terapeutico añadido correctamente

Opciones seleccionados

Biometria Química Sanguínea

descripcion: se le envio a realizar estos exámenes

Para modificar el diagnostico Terapéutico, da clic en el icono  y se muestra la siguiente interfaz.

Modificar Diagnostico Terapeutico

Opciones seleccionados

Biometria Química Sanguínea Rayos- x Otros

Descripcion:

se le envio a realizar estos

Se muestra un mensaje de éxito y se muestra los cambios que se le hizo.



Gestion de Diagnostico Terapeutico


Diagnostico Terapeutico se ha modificado correctamente

Opciones seleccionados

Biometria Química Sanguínea Otros

descripcion: se le envio a realizar estos exámenes

Para Eliminar el Diagnostico, da clic en el icono , se muestra una ventana de confirmación si desea eliminar la información ingresada, como se muestra a continuación.

¿Estás seguro?

¿Seguro que quieres borrar este elemento?

Si lo borras, nunca podrás recuperarlo.


Al momento de dar clic en eliminar se muestra la siguiente interfaz con un mensaje notificando que no hay datos ingresados y la notificación que se ha eliminado los datos.

Gestion de Diagnostico Terapeutico


Diagnostico Terapeutico se ha eliminado correctamente

No hay Datos ingresados.!!

Ingresar Motivo de la Consulta

Para ingresar el motivo de la consulta se tiene el siguiente icono  , se muestra la siguiente interfaz, si el paciente no cuenta con datos ingresados se muestra un mensaje, cuenta con el botón de regresar y el icono de ingresar la información.

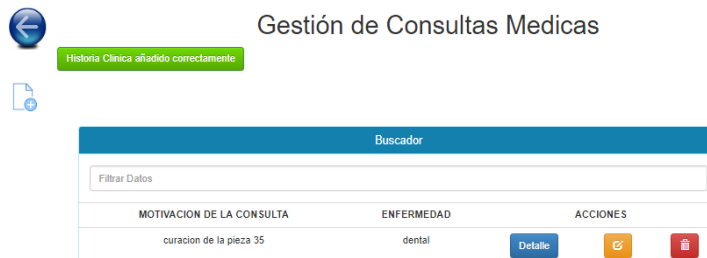


Para ingresa la información de consultas médicas, da clic en el icono  y se presenta la siguiente interfaz, la cual cuenta con los campos motivo de consulta y enfermedad y clic en enviar.

Añadir Motivo de la Consulta del Paciente

Motivo de la Consulta :	Enfermedad :
curacion de la pieza 35	dental
<input type="button" value="Enviar"/>	<input type="button" value="Cancelar"/>

A continuación, se muestra el mensaje de éxito y se muestra la información ingresada, con sus respectivas opciones.



Para ver el detalle de la información, da clic en el botón , en la que se muestra en una ventana.


Para editar la información, da clic en el botón y se muestra los siguientes campos.

Modificar Historia Clinica

Motivo de la Consulta:	Enfermedad:
<input type="text" value="curación de la pieza 36"/>	<input type="text" value="dental"/>
<input type="button" value="Enviar"/>	<input type="button" value="Cancelar"/>

Se muestra la información modificada con su respectivo mensaje de éxito.

 **Gestión de Consultas Medicas**




Buscador

Filtrar Datos

MOTIVACION DE LA CONSULTA	ENFERMEDAD	ACCIONES
curación de la pieza 36	dental	<input type="button" value="Detalle"/> <input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Para eliminar la consulta Médica, da clic en el botón , se muestra una ventana de confirmación si desea eliminar la información.

Ingresar el Odontograma

Para el ingreso del odontograma, da clic en la opción  ya que es la parte principal del paciente porque se le diagnostica los dientes para poder realizar algún tratamiento. A continuación, se muestra la siguiente interfaz con sus respectivas opciones.

Añadir Odontograma Paciente



Ingresar Sellante

Para ingresar sellante, da clic en la opción  y se muestra la siguiente interfaz.

Sellante


Necesario Realizado

11	12	13	14	15	16	17	18	21	22	23	24	25	26	27	28
31	32	33	34	35	36	37	38	41	42	43	44	45	46	47	48

Se muestra dos opciones sellante necesario o realizado, al momento de ingresar la información muestra la siguiente interfaz, con las opciones de modificar, eliminar y cancelar.



Ingresar Calce

Para ingresar calce, da clic en el icono  debe tomar en cuenta que existe las dos opciones obturado o caries, primero se selecciona la parte del diente afectado, y nos despliega una mini opción y se escoge el color que desea y finalmente clic en Guardar.




Cuando ya se ingresa los datos nos muestra el mensaje de éxito.

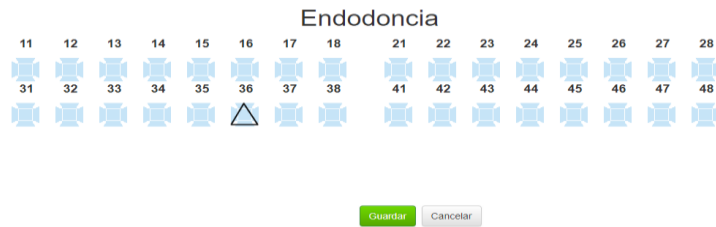


Para observar los datos que se ingresa nuevamente al icono que se indicó al inicio y se muestra la información ingresada con las opciones de modificar, eliminar y cancelar.



Ingresar Endodoncia

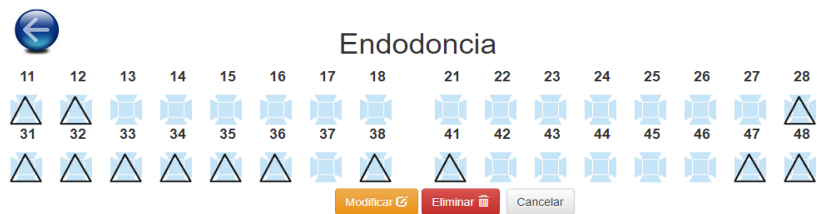
Para ingresar endodoncia, da clic en la opción , muestra a continuación la siguiente interfaz, para guardar la información, da clic sobre el diente y da clic en Guardar.



Cuando se guarda la información se muestra el mensaje de éxito.



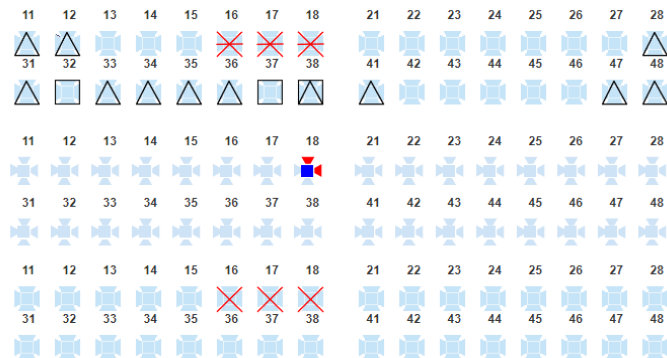
Para verificar la información ingresada, da clic sobre la opción anteriormente presentada y se presenta la siguiente interfaz la cual contiene las opciones de modificar, eliminar y cancelar.



Así es para cada una de las opciones presentadas, para obtener una información completa de la parte de odontograma da clic en Ver y nos muestra la siguiente interfaz.




Odontograma



SIMBOLOGIA DEL ODONTOGRAMA					
△ SELLANTE NECESARIO	× PERDIDA(OTRA CAUSA)	■ PROTESIS TOTAL			
△ SELLANTE REALIZADO	△ ENDODONCIA	□ CORONA			
× EXTRACCIÓN INDICADA	■ PROTESIS FIJA	● OBSTURADO			
× PERDIDA POR CARIES	■ PROTESIS REMOVIBLE	● CARIES			

Ingresar Indicadores de Salud


Para ingresar los indicadores de Salud del paciente, da clic en la opción  , se presenta la siguiente pantalla con sus opciones, tiene el botón de regresar, el botón de ingresar los indicadores y el mensaje indicando que no hay datos ingresados.



Añadir Indicadores de Salud Bucal

No hay Datos ingresados..!!



Para ingresar los indicadores de salud, da clic en el icono  , se presenta la siguiente pantalla, con su respectiva información.

Añadir Indicadores de Salud Bucal

ENFERMEDAD PERIODONTAL		MAL OCLUSION		FLUOROSIS	
LEVE	<input type="radio"/>	ANGLE I	<input type="radio"/>	LEVE	<input type="radio"/>
MODERADA	<input type="radio"/>	ANGLE II	<input type="radio"/>	MODERADA	<input type="radio"/>
SEVERA	<input type="radio"/>	ANGLE III	<input type="radio"/>	SEVERA	<input type="radio"/>

HIGIENE ORAL SIMPLICADA

PIEZAS DENTALES	PLACA (0-1-2-3)	CALCULO (0-1-2-3)	(0-1)
16 <input type="checkbox"/> 17 <input type="checkbox"/> 55 <input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
11 <input type="checkbox"/> 21 <input type="checkbox"/> 51 <input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
26 <input type="checkbox"/> 27 <input type="checkbox"/> 65 <input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
36 <input type="checkbox"/> 37 <input type="checkbox"/> 75 <input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
31 <input type="checkbox"/> 41 <input type="checkbox"/> 71 <input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
46 <input type="checkbox"/> 47 <input type="checkbox"/> 85 <input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
TOTALES	<input type="text"/>	<input type="text"/>	<input type="text"/>

Al momento de ingresar los datos se presenta la información con sus opciones de modificar y eliminar.

Añadir Indicadores de Salud Bucal

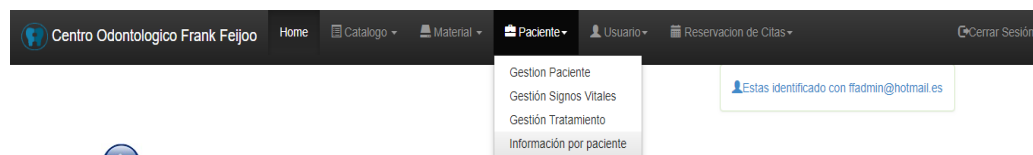
ENFERMEDAD PERIODONTAL		MAL OCLUSION		FLUOROSIS	
LEVE	<input type="radio"/>	ANGLE I	<input type="radio"/>	LEVE	<input type="radio"/>
MODERADA	<input type="radio"/>	ANGLE II	<input type="radio"/>	MODERADA	<input type="radio"/>
SEVERA	<input type="radio"/>	ANGLE III	<input type="radio"/>	SEVERA	<input type="radio"/>

HIGIENE ORAL SIMPLICADA

PIEZAS DENTALES	PLACA (0-1-2-3)	CALCULO (0-1-2-3)	(0-1)
<input type="checkbox"/> 16 <input checked="" type="checkbox"/> 17 <input checked="" type="checkbox"/> 55	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="1"/>
<input type="checkbox"/> 11 <input type="checkbox"/> 21 <input type="checkbox"/> 51	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="checkbox"/> 26 <input type="checkbox"/> 27 <input type="checkbox"/> 65	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="checkbox"/> 36 <input type="checkbox"/> 37 <input type="checkbox"/> 75	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="checkbox"/> 31 <input type="checkbox"/> 41 <input type="checkbox"/> 71	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="checkbox"/> 46 <input type="checkbox"/> 47 <input type="checkbox"/> 85	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
TOTALES	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="1"/>

Reporte de información por paciente

Para realizar observar la información relevante del paciente, debe ir al menú Paciente y da clic en información por paciente como se muestra en la figura.




Se presenta la interfaz, la cual consta el número de cédula y sus nombres y apellidos y el establecimiento.

Información Completa por Paciente

Buscador					
Filtrar Datos					
ID	CEDULA	PRIMER NOMBRE	PRIMER APELLIDO	ESTABLECIMIENTO	ACCIONES
1	1890548981	Frank	Fejoo	ODONTOLOGIA	Mostrar información Completa
4	070442482	Andy	Valencia	OCEAN PRODUCT	Mostrar información Completa
3	1809768954	Evelyn	Guzman	ESPOCH	Mostrar información Completa
8	0803713551	Giannella	Montaño	ESPOCH	Mostrar información Completa
2	0704424290	Evelyn	Valencia	ESPOCH	Mostrar información Completa
11	0201809076	Alexander	Betancourt	ESPOCH	Mostrar información Completa
12	0604375626	Jordy	Roldan	ESPOCH	Mostrar información Completa
10	1804625158	Natalia	Pilco	ESPOCH	Mostrar información Completa
9	0604532218	Liseth	Aucancela	ESPOCH	Mostrar información Completa

Para ver la información completa del paciente da clic en el botón [Mostrar información Completa](#) y se muestra la siguiente pantalla.



CENTRO ODONTOLÓGICO FRANK FEJOO

Dirección: Paseo Quito E/ Av. República y Machala
 Ubicación: Mosquitas-El Oro-Ecuador
 Teléfono: 07-2595788

Generar reporte por Paciente

Cedula: 1809768954

N° Historia Clínica: 3

Nombre: Evelyn Katherine

Apellidos: Guzman Bone

Establecimiento: ESPOCH

ANTECEDENTES PERSONALES Y FAMILIARES

Alergia Antibiótica
 Alergia Anestésica
 Hemorragias
 VIH/Sida
 Tuberculosis

descripcion:

SIGNOS VITALES [-]

Presion Arterial	Frecuencia Cardíaca	temperatura	Frecuencia Respiratoria	Fecha
25	25	12	34	2018-05-30
10	12	38	12	2018-07-04

MOTIVOS CONSULTAS [-]


Motivo Consulta	Enfermedad	Fecha
le duele el diente	sea por caries	2018-07-04

TRATAMIENTOS [-]


Fecha de Apertura	Fecha de Control	Profesores	Número de Hoja
2018-07-04	2018-07-04	Frank Fejoo	1

Número de Sesión	Fecha	Diagnóstico complicación	Procedimiento	Prescripción
1	2018-05-12	se realizó el blanqueamiento	se aplicó la pasta profiláctica	no debe tomar gaseosas o cafe ya que esto puede manchar los dientes.
2	2018-05-16	se realizó la segunda sesión para el blanqueamiento	se aplicó el pH para el blanqueamiento	se emiten prescripciones seriamente escritas.

Ingresar Tratamientos

Para el ingreso de los tratamientos con sus respectivos detalles, da clic en la opción , la cual se muestra la siguiente pantalla

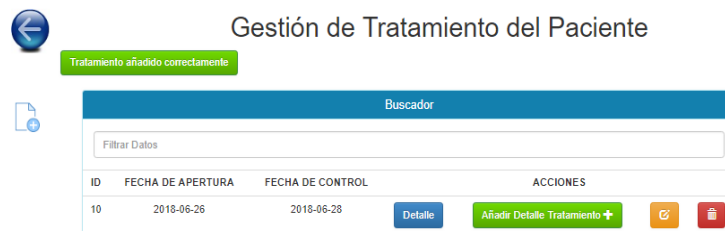


Para registrar el tratamiento , se muestra el formulario se selecciona el profesional que existen el centro odontológico, se selecciona la fecha de apertura, se selecciona la fecha de control y el número de apertura.

Registro del Tratamiento

Seleccione el profesional:	Fecha de Apertura:	Fecha de Control:	Numero de Apertura
Frank Fejoo ▼	26/06/2018	28/06/2018	1
<input type="button" value="Enviar"/>	<input type="button" value="Cancelar"/>		


Al momento que se ingresó la información se muestra en la pantalla con sus respectivas opciones como detalla, añadir tratamiento, modificar, eliminar tratamiento y el botón de regreso. Como se puede ver se muestra el mensaje de éxito.



Cuando da clic en detalle se nos muestra una ventana con los detalles de tratamiento.



Para modificar se da clic en el botón  y para eliminar el botón .

Para registrar el detalle tratamiento se da clic en el botón **Añadir Detalle Tratamiento +**, se muestra la siguiente pantalla, cuenta con el botón de regresar, para ingresar detalle tratamiento, da clic en el icono .



Luego de dar clic se muestra la interfaz para llenar los campos la fecha de la primera sesión, diagnóstico y complicaciones, procedimientos, prescripciones y se selecciona los materiales que se utilizó para ese tratamiento.

Registro del Tratamiento Detalle

SESION	FECHA	DIAGNOSTICOS Y COMPLICACIONES	PROCEDIMIENTOS	PRESCRIPCIONES
1	28/06/2018	tuvo una complicacion en su	se le puso una anestesia en	que tome paracetamol

Añadir Materiales

ID	NOMBRE	CANTIDAD
<input type="checkbox"/>	Limas	1
<input checked="" type="checkbox"/>	PG0	2



Enviar **Cancelar**

Cuando se haya llenado la información se lista el detalle tratamiento que se ingresó, también se muestra el mensaje de éxito.



Para ver el detalle de tratamiento, da clic en el botón detalle, la cual se muestra la ventana con la información que se ingresó.



Para modificar el detalle tratamiento da clic en el botón  y para eliminar da clic en el botón .

Ingresar Material


Esta pantalla es para la gestión de Materiales tiene las opciones de registrar, modificar, ver detalle y eliminar de cada material.

Gestión de Materiales

Buscador

Filtrar Datos

ID	NOMBRE	CATEGORIA	MARCA	CANTIDAD	UTILIZADO	RESTANTE	ACCIONES
2	Resinas	Trabajos	Zhermack	0	2	0	Detalle  
3	Limas	Trabajos	Mallefer	12	1	12	Detalle  
1	P60	Blanqueamiento	BioSystem	7	13	7	Detalle  

Para registrar el material da clic en el icono , se muestra la siguiente pantalla con sus respectivos campos que se debe llenar.

Añadir Material

Nombre

Seleccione el Categoría:

Seleccione el Marca:

Seleccione el unidad:

Fecha de Caducidad:

Cantidad:

Al momento de ingresar los datos se muestra el mensaje de éxito y se lista el material que se ingresó. Y también se muestra las acciones de cada material que se puede hacer como detalle, modificar y eliminar.

Gestión de Materiales

Material añadido correctamente



The interface shows a search bar and a table of materials. The table has columns for ID, Nombre, Categoría, Marca, Cantidad, Utilizado, Restante, and Acciones. There are five rows of material data, each with 'Detalle', 'Modificar', and 'Eliminar' buttons.

ID	NOMBRE	CATEGORIA	MARCA	CANTIDAD	UTILIZADO	RESTANTE	ACCIONES
5	POLA OFFICE + PLUS 37.5% - BLANQUEADOR DENTAL	Blanqueamiento	SDI	10	0	0	Detalle, Modificar, Eliminar
4	POLA DAY CP 35% MINI KIT BLANQUEAMIENTO DENTAL	Blanqueamiento	SDI	20	0	0	Detalle, Modificar, Eliminar
6	ZARIS WHITE & BRITE - Gel Blanqueador	Blanqueamiento	3M	7	0	0	Detalle, Modificar, Eliminar
7	Opalescende GO 6% PATIENT KIT BLANQUEAMIENTO MENTA	Blanqueamiento	Ultradent	7	0	0	Detalle, Modificar, Eliminar
8	Dentífico blanqueador OPALESCENCE	Blanqueamiento	Ultradent	9	0	0	Detalle, Modificar, Eliminar

PANTALLA PRINCIPAL PARA EL PACIENTE

Cuando se autentifica el Paciente se le presenta esta pantalla de inicio con su respectivo menú



En el menú Editar datos se presenta la información del paciente, contiene el botón Modificar datos permite al usuario o paciente actualizar la información.

Estas identificado con carolinavalencia@hotmail.es

Datos Personales

DATOS PERSONALES	
Cédula: 0704424290	
Nombre: Evelyn Carolina Valencia Espinoza	Edad: 25
Establecimiento: ESPOCH	Sexo: Femenino
Modificar Datos	

Al momento de dar clic en el botón, se presenta el siguiente formulario, cabe resaltar que la información que no se puede modificar es la cédula y el correo.

Modificar Información

Correo:	Cedula:
<input type="text" value="carolinavalencia@hotmail.e"/>	<input type="text" value="0704424290"/>
Primer Nombre:	Segundo Nombre:
<input type="text" value="Evelyn"/>	<input type="text" value="Carolina"/>
Primer Apellido:	Segundo Apellido:
<input type="text" value="Valencia"/>	<input type="text" value="Espinoza"/>
Fecha de Nacimiento:	Seleccione el Sexo:
<input type="text" value="07/04/1993"/>	<input type="text" value="Femenino"/>
Establecimiento:	
<input type="text" value="ESPOCH"/>	
Enviar	Cancelar

También se cuenta con el menú de Reservación de citas, el paciente puede reservar citas la fecha que desea con hora deseada. Como se muestra a continuación.

Reservación de Citas

Seleccione la fecha:	
<input type="text" value="Jueves, 19/ Julio/ 2018"/>	
<input type="radio"/> 8:00 am	<input checked="" type="radio"/> 9:00 am
<input type="radio"/> 10:00 am	<input type="radio"/> 11:00 am
<input type="radio"/> 12:00 am	<input type="radio"/> 3:00 pm
<input type="radio"/> 2:00 pm	<input type="radio"/> 4:00 pm
<input type="radio"/> 5:00 pm	<input type="radio"/> 6:00 pm
Enviar	Cancelar

en el menú mostrar reservaciones de citas médicas se muestra la pantalla de la siguiente manera.

Reservaciones de Citas Médicas

ID	FECHA	ACCIONES
3	2018-07-19	Eliminar