

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:  
Ingeniero de Sistemas**

**TEMA:  
ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN  
SISTEMA WEB DE CONTROL DE HISTORIAS Y CITAS MÉDICAS PARA  
EL DEPARTAMENTO MÉDICO DE LA UPS CAMPUS SUR.**

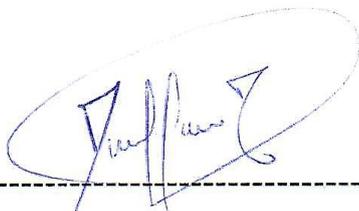
**AUTOR:  
ALEJANDRO ANDRÉS LLANO JIMÉNEZ**

**TUTOR:  
ROBINSON DIMITRI LLERENA PAZ**

**Quito, marzo del 2018**

## CESIÓN DE DERECHOS DE AUTOR

Yo ALEJANDRO ANDRÉS LLANO JIMÉNEZ, con documento de identificación N° 1715522536, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: “ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE CONTROL DE HISTORIAS Y CITAS MÉDICAS PARA EL DEPARTAMENTO MÉDICO DE LA UPS CAMPUS SUR”, mismo que ha sido desarrollado para optar por el título de: INGENIERO DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



---

ALEJANDRO ANDRÉS LLANO JIMÉNEZ  
CI: 1715522536

Quito, marzo del 2018

## **DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR**

Yo declaro que bajo mi dirección y asesoría fue desarrollado el PROYECTO TÉCNICO, ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE CONTROL DE HISTORIAS Y CITAS MÉDICAS PARA EL DEPARTAMENTO MÉDICO DE LA UPS CAMPUS SUR realizado por ALEJANDRO ANDRÉS LLANO JIMÉNEZ, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, marzo del 2018

A handwritten signature in blue ink, consisting of several overlapping loops and strokes, positioned above a horizontal dashed line.

ROBINSON DIMITRI LLERENA PAZ

CI: 1707106942

## **AGRADECIMIENTO**

Agradezco a la Universidad Politécnica Salesiana que ha contribuido en mi formación profesional y personal, a mis padres que han dado todo el esfuerzo para que yo ahora este culminando esta etapa de mi vida, a mi tutor de proyecto de titulación Robinson Llerena por haberme orientado y motivado para poder realizar mi trabajo.

Alejandro Andrés Llano Jiménez

# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
Antecedentes .....	2
Problema .....	2
Objetivo General .....	3
Objetivos Específicos.....	3
<b>Capítulo 1.....</b>	<b>4</b>
1.1 Entorno del sistema.....	4
1.2 Principales teorías que sustentan el desarrollo de la aplicación.....	5
1.2.3 Programación orientada a objetos. ....	5
1.2.4 Estructura de bases de datos relacionales.....	6
1.2.5 Arquitectura de la aplicación web.....	7
1.2.6 Aspectos de seguridad en un sitio web. ....	8
1.2.7 Atención médica y tecnología.....	9
<b>Capítulo 2.....</b>	<b>11</b>
2.1 Acta de constitución del proyecto .....	11
2.1.1 Información del proyecto. ....	11
2.1.2 Propósito del proyecto.....	11
2.1.3 Alcance.....	12
2.1.4 Descripción del proyecto. ....	12
2.1.5 Módulos del sistema a ser desarrollados.....	13
2.1.6 Reglas.....	15
2.1.7 Riesgos. ....	16
2.1.8 Costo estimado.....	16
2.1.9 Involucrados.....	17
2.1.10 Recursos. ....	17
2.2 Requerimientos funcionales .....	17
2.2.1 Tipos y Características del usuario. ....	18
2.2.2 Ambiente Operativo.....	19
2.2.3 Requerimientos de Interfaces Externas.....	19
2.4 Especificación ágil de requerimientos .....	20
2.5 Análisis y Diseño del Sistema.....	28
2.5.1 Priorización. ....	28

2.5.2 Especificación de la carga de trabajo para el desarrollo. ....	28
2.5.3 Product Backlog. ....	28
2.5.4 Priorización. ....	30
2.5.5 Plan de Entregas. ....	32
2.5.6 BPMS (Suite de gestión de procesos empresariales). ....	33
2.5.7 WBS. ....	35
2.5.8 Diagramas UML aplicados al sistema. ....	36
2.5.8.2 Modelo Relacional. ....	38
2.5.8.3 Interfaces del sistema. ....	39
2.6 Herramientas de desarrollo. ....	44
2.6.1 PHP. ....	44
2.6.2 CodeIgniter. ....	45
2.6.3 Base de Datos. ....	46
2.6.4 Bootstrap. ....	47
2.6.5 CSS. ....	47
2.6.6 JavaScript. ....	48
2.6.7 Herramienta de entorno de desarrollo XAMPP. ....	48
2.6.8 Metodología de desarrollo SCRUM. ....	49
2.6.8.1 Características de Scrum. ....	49
2.6.8.2 Fases Scrum. ....	50
2.6.8.3 Proceso. ....	50
2.7 Implementación. ....	51
2.7.1 Codificación relevante. ....	52
2.7.1.1 Conexión a base de datos. ....	52
2.7.1.2 Estructura. ....	53
2.8 Pruebas de aceptación. ....	65
2.9 Capacitaciones. ....	69
<b>Capítulo 3. ....</b>	<b>70</b>
3.1 Análisis de resultados con GTmetrix. ....	70
3.2 Pruebas de satisfacción de usuario. ....	71
3.2.1 Formato Encuesta. ....	72
3.2.2 Análisis de respuestas obtenidas en las encuestas. ....	73
<b>CONCLUSIONES. ....</b>	<b>77</b>
<b>RECOMENDACIONES. ....</b>	<b>78</b>
<b>LISTA DE REFERENCIA. ....</b>	<b>79</b>

## ÍNDICE DE TABLAS

Tabla 1. Costo estimado.....	16
Tabla 2. Involucrados.....	17
Tabla 3. Requerimiento de gestión de usuarios .....	20
Tabla 4. Requerimientos Registro de Pacientes.....	21
Tabla 5. Requerimientos de Agendamiento de Citas .....	22
Tabla 6. Requerimientos de Registro de Atención Médica.....	23
Tabla 7. Requerimientos de Parte Diario .....	24
Tabla 8. Requerimientos de Informe de inspecciones .....	24
Tabla 9. Requerimientos de Indicadores morbilidad .....	25
Tabla 10. Requerimientos de Informes Epidemiológicos .....	26
Tabla 11. Requerimientos de Plan de Vigilancia Médico.....	27
Tabla 12. Product Backlog .....	29
Tabla 13. Priorización .....	31
Tabla 14. Plan de entregas .....	32
Tabla 15. Prueba de aceptación para la gestión de usuarios. ....	65
Tabla 16. Prueba de aceptación Registro de Pacientes .....	65
Tabla 17. Prueba de aceptación agendamiento de citas .....	66
Tabla 18. Prueba de aceptación de registro de atención medica.....	66
Tabla 19. Prueba de aceptación del parte diario .....	67
Tabla 20. Prueba de aceptación de informe de inspecciones .....	67
Tabla 21. Prueba de aceptación de Indicadores de morbilidad.....	68
Tabla 22. Prueba de aceptación de informes epidemiológicos .....	68
Tabla 23. Prueba de aceptación de plan de vigilancia médica.....	69

## ÍNDICE DE FIGURAS

Figura 1. Diagrama de procesos BPMN .....	34
Figura 2. WBS Estructura de desglose de trabajo.....	35
Figura 3. Diagrama de Casos de Uso para la Atención Médica.....	36
Figura 4. Diagrama de Casos de Uso para las Inspecciones Preventivas .....	37
Figura 5. Diagrama de Casos de Uso para el Plan de vigilancia.....	37
Figura 6. Diagrama relacional.....	38
Figura 7. Interfaz para el registro y visualización de pacientes .....	39
Figura 8. Interfaz del listado de pacientes registrados .....	39
Figura 9. Interfaz que visualizará el usuario para agendar citas .....	40
Figura 10. Interfaz del listado de citas .....	40
Figura 11. Interfaz del perfil de paciente .....	41
Figura 12. Interfaz de parte diario.....	41
Figura 13. Interfaz de indicadores de morbilidad .....	42
Figura 14. Interfaz de Indicadores Epidemiológicos .....	42
Figura 15. Interfaz para visualizar la lista de Inspecciones BPMS.....	43
Figura 16. Interfaz del registro de Inspecciones BPMS.....	43
Figura 17. Interfaz de registro del Informe de BPMS.....	44
Figura 18. Configuración para conexión a base de datos.....	53
Figura 19. Codificación de la clase Modelo .....	54
Figura 20. Codificación de la clase vista .....	55
Figura 21. Codificación de formulario de ingreso de datos .....	56
Figura 22. Método JavaScript para visualizar información de la base de datos .....	57
Figura 23. Método JavaScript para eliminar registros de la base de datos .....	58
Figura 24. Codificación de la vista controlador .....	59
Figura 25. Codificación de la clase vista para el método de autenticación.....	60
Figura 26. Codificación del controlador para el método de autenticación .....	61
Figura 27. Codificación del controlador para subir imágenes .....	62
Figura 28. Codificación de la clase vista para subir imágenes .....	63
Figura 29. Codificación del método controlador para subir imágenes .....	64
Figura 30. Puntajes de rendimiento.....	70
Figura 31. Tiempos de carga.....	70
Figura 32. Detalles del sistema .....	71
Figura 33. Encuesta de satisfacción .....	72
Figura 34. Análisis - Pregunta 1.....	73
Figura 35. Análisis - Pregunta 2.....	74
Figura 36. Análisis - Pregunta 3.....	74
Figura 37. Análisis - Pregunta 4.....	75
Figura 38. Análisis - Pregunta 5.....	75
Figura 39. Análisis - Pregunta 6.....	76

## **Resumen**

Este proyecto pone en práctica todos los conocimientos que se han adquirido a lo largo la carrera universitaria y el mismo consiste en un sistema web para la automatización de procesos de atención en el centro médico de la universidad politécnica salesiana campus sur, debido a que en la actualidad se maneja la información de los pacientes de la forma tradicional es decir mediante documentos físicos, por lo cual el proceso es lento e impreciso provocando que exista el riesgo de pérdida de información o que se registre información equivocada, lo cual será controlado por el sistema evidenciando una atención más rápida y eficiente. Los usuarios serán parte integral del sistema por lo tanto el éxito dependerá de su correcto uso, de esta manera se logrará que el sistema de control se integre eficientemente y contribuya a mejorar la calidad de la atención reflejando una satisfacción notable por parte de médicos y pacientes.

## **Abstract**

This project will use all knowledge that has been acquired throughout the university career and it consists of a web system for automation of care processes in the medical center of the university's south campus, because it handles information of patients in a traditional way by physical documents, so the process is slow, there is also the risk of information loss or that inaccurate information is recorded, which will be controlled by the system showing a faster and efficient attention for patients. The users will be an integral part of the system therefore the success will depend on its correct use in this way will be achieved that the control system be integrated efficiently and contribute to improve the quality of care reflecting a remarkable satisfaction on the part of doctors and patients.

## INTRODUCCIÓN

El presente proyecto dará una solución al manejo de los expedientes médicos mediante un sistema web que satisfaga la problemática actual del centro médico de la universidad politécnica salesiana campus sur proponiendo un sistema que automatice el proceso de atención médica y evite realizar procesos repetitivos e innecesarios lo cual es posible ya que el sistema puede reutilizar la información. También se pretende generar una organización eficiente mediante agendas programadas que incentiven tanto a pacientes y médicos a cumplir con los horarios establecidos. Además el sistema proporcionará información más detallada y precisa por lo cual ya no será necesario registrar información repetitiva, tratado de que el proceso de atención medica pueda ser llevado a cabo solamente con unos cuantos pasos que permitirán que le medico concentre toda su atención en dar un diagnóstico médico de calidad sin distracciones que podrían darse al generar documentación innecesaria lo que conllevará a una reducción de carga de trabajo para el usuario, mismo que al final del día toda la documentación que hasta ahora debe realizar manualmente podrá obtenerla de forma automática.

Este documento consta de III capítulos: El capítulo I se enfoca en el entorno de desarrollo del sistema, detallando las herramientas que serán utilizadas y la metodología que permitirá estructurar el trabajo para lograr satisfacer los requerimientos del proyecto; En el capítulo II se detallarán las características del sistema, la organización de trabajo, las actividades a realizarse la estimación de recursos, su implementación entre otros temas referentes a lo mencionado; En el capítulo III se realizarán pruebas de carga al sistema para evaluar su rendimiento y encontrar posibles errores que a simple vista no podrían ser percibidos y finalmente se evaluará la satisfacción del usuario.

## **Antecedentes**

En la actualidad el personal del centro médico invierte grandes cantidades de tiempo en pasar a limpio toda la información que se obtiene en las actividades diarias, además se realizan otras actividades de las que está a cargo este departamento como cuadros estadísticos, inspecciones, informes, planificaciones entre otras actividades las cuales al no estar automatizadas han ocasionado pérdidas de información y también conllevan una inversión de tiempo innecesario. Sin dejar de lado el punto más importante como es la atención a los pacientes los cuales al no contar con una adecuada coordinación muchas de las veces no logran ser atendidos.

## **Problema**

El centro médico de la UPS maneja la información de los pacientes de la forma tradicional es decir mediante documentos físicos, por lo cual el proceso de buscar la información de cada paciente hace más lenta la atención, también existe riesgo de pérdida de información, o que los mismos datos de cada paciente no sean registrados con precisión en cada cita médica lo cual puede provocar que el médico no tenga una visión clara del estado del paciente. Además, existe descoordinación en la asistencia a las citas, olvido de la dosificación de medicamentos o de las indicaciones del médico. Otro problema importante aparece en el momento que el doctor realiza los informes estadísticos en base a la atención, mismos que en la actualidad no reflejan porcentajes reales ya que es muy complicado procesar toda la información manualmente. Los médicos invierten grandes cantidades de tiempo en pasar a limpio toda la información que al final no muestra el resultado deseado y al no estar automatizadas conllevan a una pérdida de tiempo y esfuerzo.

## **Objetivo General**

Desarrollar un sistema de control web que automatice eficientemente los procesos del centro médico contribuyendo a aumentar la calidad en la atención mejorando así de forma notable la atención médica y por lo tanto la satisfacción de los pacientes.

## **Objetivos Específicos**

Aplicar los instrumentos de SCRUM para el levantamiento de la información.

Diseñar y analizar todos los procesos del sistema con el fin de identificar los requerimientos reales.

Analizar la información recopilada determinando así el alcance del sistema.

Identificar buenas prácticas de construcción de aplicaciones web referentes al campo de la salud.

Crear un software estructurado utilizando un entorno de desarrollo integrado orientado a la producción de un sistema web.

Incentivar a los usuarios al uso de un sistema de información para obtener beneficios y mejorar la calidad de atención.

# Capítulo 1

## 1.1 Entorno del sistema

El desafío de un sistema de información es dar respuesta a la necesidad de gestionar distintos niveles de información en este caso referentes a registros personales de salud con acceso a fuentes de información médica de interés.

Este proceso comienza con la necesidad de entender que se está produciendo un cambio para luego trabajar sobre la necesidad de que las instituciones se adapten a la nueva situación y rediseñen sus procesos organizativos, para lo cual es necesaria la participación de todos los involucrados, conformando grupos capacitados para enfrentar un entorno diferente.

Una vez aprobada la implantación del sistema de información se deberá superar diferentes desafíos en el desarrollo. Algunos de estos factores son propios del desarrollo de cualquier tecnología y otros se originarán a lo largo del desarrollo de este proyecto y serán superados acordes a la voluntad de cambio.

“Cuando se invierte en sistemas de información, la finalidad esencial de la organización debe identificarse claramente. En el caso de los servicios de salud, dicha finalidad es mejorar el estado de salud de los pacientes, mediante el mejoramiento de los sistemas de atención” (Rodrigues, 1999). Contar con información real es esencial para tomar decisiones fundamentadas, las cuales aumentan la probabilidad de realizar acciones correctas ya que un sistema de información contiene procesos que se ejecutan de acuerdo con la información con la que cuenta.

## **1.2 Principales teorías que sustentan el desarrollo de la aplicación**

### **1.2.3 Programación orientada a objetos.**

Es una metodología de programación que permite modelar sistemas mediante la creación de clases, mismas que se pueden definir como un conjunto de datos y funcionalidades a partir de las cuales se crean los objetos los cuales contienen los datos y funciones definidos en la clase.

Este tipo de programación permite crear aplicaciones de una manera intuitiva. A continuación definiremos los fundamentos principales del lenguaje de programación a ser utilizado para desarrollar este proyecto (PHP) para utilizar objetos.

**Clases:** Se define como un conjunto de atributos y funciones que forman métodos mismos que son una especificación de propiedades y funcionalidades de elementos que van a ser utilizados en el desarrollo.

**Herencia:** Permite heredar atributos y operaciones entre distintas clases como si esos atributos y operaciones hubiesen sido declarados por la clase que haga uso de los mismos.

**Objeto:** Se puede definir como una entidad que posee propiedades o atributos que definen su comportamiento y reaccionan a eventos. Como si se tratara de objetos reales estos interactuarán con objetos internos del sistema.

**Método:** Es un algoritmo asociado a un objeto que puede ser ejecutado al recibir datos pudiendo producir cambios en las propiedades del objeto.

Una vez analizados estos fundamentos se reconoce que trabajar con este tipo de programación es similar a como se lo haría en la vida real. La programación

orientada a objetos, simula el mundo real definiendo objetos que contienen características y funciones. Se puede decir que se basa en la idea de un objeto que es una combinación de variables y procedimientos que forman métodos.

#### **1.2.4 Estructura de bases de datos relacionales.**

Modelo relacional consistente en datos almacenados en tablas relacionadas unas entre otras mismas que se componen por filas y columnas, resultando como un modelo más comprensible por ejemplo que el modelo jerárquico ya que se basa en la lógica de predicados para establecer relaciones entre distintas tablas haciendo referencia a los datos que contienen. La relación se representa mediante una tabla entidad/relación cuyas principales características son:

Atributo: corresponden a las columnas de la tabla se definen por un nombre y contienen valores.

Tupla: corresponden a las filas que componen la tabla.

Claves: Cada tupla debe tener un identificador el cual será único dentro de cada tabla

Clave primaria: Es un valor que identifica una fila dentro de una tabla y siempre deberá contener un valor.

Clave foránea: Es un valor dentro de una tabla que hacer referencia a una clave primaria en otra tabla. Formando así las relaciones

Estructurar una base de datos con el modelo relacional garantiza la no duplicidad de registros, así como una integridad referencial que como una ventaja destacada sería que al eliminar un registro se eliminarán todos los registros relacionados. Sin dejar de lado que este modelo facilita la normalización por ser más comprensible.

## **1.2.5 Arquitectura de la aplicación web.**

### ***1.2.5.1 Cliente Servidor.***

Al tratarse del desarrollo de un sistema web la arquitectura cliente-servidor es la adecuada ya que es un modelo de aplicación distribuida en el que las tareas se reparten entre el servidor y los usuarios que harán solicitudes con el sistema. En esta arquitectura los procesos son repartidos entre los clientes y los servidores siendo los clientes quienes soportan la parte visual y el servidor los procesos de lógica de negocio, teniendo como principal ventaja la separación de responsabilidades.

La arquitectura cliente/servidor consiste en la comunicación de los clientes hacia un servidor en el que se centralizan los recursos y aplicaciones mismas que estarán disponibles para los clientes que tengan acceso cada vez que los soliciten.

La arquitectura cliente servidor permite centralizar el control como son accesos, recursos y la integridad de los datos que ahora serán controlados por el servidor teniendo así un impacto sumamente positivo ya que si existiera algún cliente defectuoso o que no posea autorización el mismo no podrá afectar el funcionamiento del sistema, además de evitar la tarea de actualización de datos ya que los clientes accederán a una base de datos común. Una característica importante de esta arquitectura es que permite la escalabilidad ya que permite aumentar la capacidad de clientes y servidores en cualquier momento.

Existe la posibilidad de una congestión cuando una gran cantidad de clientes realizan peticiones concurrentemente esto depende también de las capacidades del servidor y la red que se utilizará.

### **1.2.6 Aspectos de seguridad en un sitio web.**

Subir una página web al internet conlleva riesgos, mismos que pueden ser controlados teniendo en cuenta que la seguridad en una página debe asegurar la privacidad de la información que esté reservada a usuarios autorizados. Las condiciones de seguridad de una página web tienen riesgos dentro o fuera del sistema por lo cual para establecer medidas de seguridad que principalmente son:

La encriptación de datos que consiste en una técnica para ocultar datos de manera que sólo puedan ser vistos por los usuarios autorizados, este proceso consiste en transformar un mensaje de manera que no pueda ser entendido o sea difícil descifrar por otros.

Existen elementos informáticos como Firewalls, Wrappers y Proxies que proveen de una línea de defensa para los servidores web ya que protegen los puertos y evita que los atacantes accedan al sistema, permitiendo que sólo ciertos nombres de dominio confiables tengan acceso al sistema. Por ejemplo el proxy es un método que permite ocultar datos por medio de re enrutamiento de solicitudes. Los usuarios establecen una dirección proxy de su navegador para que apunte hacia el servidor, mismo que dirige los datos hacia el cliente permitiendo al usuario enrutar los datos a través de sus propias firewalls.

#### ***1.2.6.1 Prácticas de seguridad web.***

Creación de un login que mediante credenciales otorgue acceso al sistema de forma transparente a los usuarios sin complicar la usabilidad del sistema.

Las credenciales de acceso almacenadas en la base de datos como usuario y contraseña son consideradas sensibles por lo que deben ser manejados con discreción haciendo uso de encriptación por ejemplo para la contraseña.

Se debe manejar con prudencia las variables de sesión o las cookies por ejemplo cuando un usuario accede al sistema mediante un login se debe establecer tiempos de inactividad en los cuales estas variables expiren y no se conviertan en brechas de seguridad.

### **1.2.7 Atención médica y tecnología.**

La tecnología se ha convertido en un elemento clave en el manejo de los procesos médicos ya que es capaz de procesar bastas cantidades de información inteligentemente dando un notable valor agregado a la atención médica y no solamente a este ámbito sino que también modifican la propia estructura de todos los procesos que se manejan en el área de la salud redefiniendo el uso de tecnologías médicas.

“En los Estados Unidos un 80% de los centros de salud utilizan registros electrónicos para gestionar la información, aunque solo entre el 10% y el 15% alcanzaron el nivel más alto de madurez digital, según la escala elaborada por KPMG. En lo que respecta a servicios de cloud computing cerca de un 15% de los hospitales ya utilizaban la nube para almacenar sus imágenes desde 2013, y se espera que para el final de 2016 ese número aumente a más del 50%” (DonWeb, 2014).

Según el estudio realizado por KPMG (DonWeb, 2014). “En pocos años alrededor de 500 millones de personas de todo el mundo utilizarán alguna aplicación relacionada con la salud y el 26% de los hospitales están presentes en las redes sociales”.

Los sistemas informáticos de salud reciben millones de registros a sus bases de datos diariamente, sin embargo a medida que se implementa la tecnología aumentan los riesgos de seguridad informática. Por ejemplo, según una encuesta realizada por el Instituto Ponemon el 40% de las instituciones de salud a nivel mundial reportaron ataques en 2013.

En la actualidad los sistemas informáticos de medicina ya se encuentran implementados en varios países siendo bastante común y después de lo analizado incluso es algo lógico que los sistemas informáticos manejen la información y los procesos creando una verdadera red logrando que las personas tengan a disposición toda su historial clínico en cualquier momento y lugar. En Ecuador este tipo de aplicaciones se encuentran en etapa inicial pero seguramente la globalización obligará a adoptar rápidamente la tecnología que hasta ahora ha demostrado ser necesaria para el mundo moderno.

## **Capítulo 2**

### **2.1 Acta de constitución del proyecto**

#### **2.1.1 Información del proyecto.**

Empresa: Proyecto de titulación Alejandro Llano

Proyecto: ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE CONTROL DE HISTORIAS Y CITAS MÉDICAS PARA EL DEPARTAMENTO MÉDICO DE LA UPS CAMPUS SUR.

Cliente: Centro Médico UPS Campus Sur

Involucrado principal: Byron Gangotena.

Gerente del proyecto: Robinson Llerena.

Desarrollador: Alejandro Llano

#### **2.1.2 Propósito del proyecto.**

Después de haber analizado el estado actual del centro médico se comprende claramente la necesidad de desarrollar un proyecto que automatice los procesos que se manejan dentro de este departamento, siendo esto posible ya que el sistema puede reutilizar la información automatizando así los procesos de tal forma que se pueda acceder a la información con una búsqueda rápida y de la misma manera se pueda gestionar la información de forma precisa, logrando también una organización eficiente de actividades a realizarse que agilizarán los procesos permitiendo una mayor acogida de pacientes y mejorarán la imagen del centro médico.

### **2.1.3 Alcance.**

La solución de este proyecto para la problemática actual del centro médico antes descrita contempla la automatización de actividades tales como la realización de informes, reportes estadísticos, planificaciones, inspecciones. La atención médica será el módulo principal ya que aborda varios procesos los mismos que deberán ser mejorados partiendo del estado actual. El proyecto no creará procesos adicionales como nuevos reportes estadísticos, emisión de pedidos de exámenes, facturación u otros que podrían aparentar ser necesarios. Debido al planteamiento del este proyecto son funcionalidades innecesarias que no aportarían a mejorar los procesos que maneja el centro médico.

### **2.1.4 Descripción del proyecto.**

Para poder alcanzar la solución más óptima se desarrollará un sistema médico de control, con acceso a una base de datos en la que se almacenarán los datos generados en las interacciones con los pacientes y otros datos obtenidos con actividades que realiza este departamento aparte de la atención médica como se detalla en la justificación.

Será un sistema web desarrollado con el framework CodeIgniter el cual está basado en PHP, se hará uso del gestor de base de datos MySQL, para el diseño se utilizará BOOTSTRAP.

El sistema será implementado en un hosting gratuito, todos los usuarios podrán acceder a este a través de internet con sus respectivas credenciales, esto es factible ya que en un acercamiento previo con los beneficiarios de este proyecto se pudo constatar que disponen de dos computadores con acceso a internet.

Será implementado de esta manera para que pueda ser probado y valorado por los usuarios del centro médico. Si la universidad considera pertinente la implementación del sistema de forma oficial es responsabilidad suya realizar este proceso aclarando que el desarrollador de este proyecto tendrá toda la predisposición para colaborar de darse el caso.

El sistema requerirá información personal de los pacientes necesarios para cumplir con los procesos principalmente para los módulos de atención médica e informes estadísticos mismos que reutilizarán toda la información de los pacientes. Con toda la información registrada en las interacciones el sistema se encargará de procesar y mostrar reportes en formatos requeridos y aprobados por los usuarios.

### **2.1.5 Módulos del sistema a ser desarrollados.**

#### **Reportes Estadísticos**

##### **Indicadores morbilidad**

Consiste en certificados que proporcionan permisos médicos mismos que el doctor emitirá una vez realizada la atención, esta evaluará si el paciente tiene alguna patología que le impida realizar sus actividades con normalidad. Además contiene informes estadísticos de acuerdo a los tipos de patologías, edad, tipo de paciente, entre otros.

##### **Indicadores epidemiológicos**

Consiste en una serie de informes y reportes estadísticos acerca de las incidencias de ciertas patologías en pacientes.

#### **Informes**

##### **Parte diario**

Se basa en un formulario en el cual el médico podrá apreciar el resumen de las atenciones médicas al final de cada jornada de trabajo, este formulario mostrará los siguientes datos como son la información del paciente, anamnesis, patología, tratamiento, medicación entre otros.

### **Directrices estratégicas médicas**

Es una directriz de seguridad y salud que se entregará al paciente con indicaciones acerca de los cuidados que debe tener en su área de trabajo según la patología que se le haya diagnosticado en la consulta.

## **Inspecciones**

### **Registro de control BPM (buenas prácticas médicas)**

Consiste en una evaluación del personal, instalaciones, procesos, etc., que el inspector irá calificando según la apreciación que obtenga al momento de realizar esta inspección la cual calculará de manera instantánea la calificación total. El usuario podrá hacer uso de este módulo desde cualquier dispositivo móvil lo cual le permitirá obtener los resultados en tiempo real.

### **Informe de observaciones preventivas del bar campus sur**

Consiste en un informe que evalúa y detalla toda la información obtenida en la inspección BPM, está basado en el formato que se utiliza actualmente, el mismo tendrá una adaptabilidad responsive (diseño web adaptativo) para que le medico pueda realizar este informe desde su dispositivo móvil.

## **Plan de vigilancia médico**

Consiste en un cronograma digital que establece el tiempo que se destinará a realizar actividades tales como: programas de salud ocupacional, campañas de salud, capacitaciones, entre otras. También se permitirá agregar descripciones, alertas y más datos informativos para lograr que estas actividades se realicen de forma

oportuna y eficiente. Este módulo será visible para todos los usuarios que tengan acceso al sistema ya que son actividades generales.

### **Módulo de Autenticación**

Para acceder a la información el usuario debe identificarse, otorgándole esa identificación diferentes niveles de permisos. La contraseña es personal y renovable.

### **Gestión de pacientes**

Para obtener la información de un determinado paciente se dispone de un sistema de consultas que admite datos de identificación como parámetros de búsqueda, o mediante un listado de todos los pacientes registrados. Una vez seleccionado un paciente se dispondrá de toda su información misma que podrá ser actualizada o simplemente visualizada.

#### **2.1.6 Reglas.**

**Reuniones:** Se programarán reuniones cada dos semanas aproximadamente para evaluar el avance de las tareas planteadas y programar nuevas tareas, esta programación podrá variar según las condiciones en las que se encuentre el proyecto durante su desarrollo.

#### Documentación en Anexo 6 – Reuniones con el Cliente

**Comunicación:** Los actores involucrados en este proyecto tienen la predisposición de atender inquietudes que se darán a lo largo del desarrollo en cualquier momento.

**Toma de decisiones:** Las decisiones dependerán en gran medida de la aceptación del cliente (Doctor del centro médico). Cualquier cambio no planificado previamente deberá esperar a la evaluación y aprobación por parte del actor mencionado y el

tutor. Para este proceso el estudiante encargado de desarrollar el sistema será el intermediario.

**Reporte de tareas:** Cada tres semanas aproximadamente se hará el seguimiento. Una vez culminadas las tareas asignadas se realizará el registro de cumplimiento y será revisado por el tutor.

### 2.1.7 Riesgos.

Resistencia al cambio por parte de los usuarios ya que en la actualidad no utilizan ningún tipo de sistema informático para la atención médica.

Inexperiencia con ciertas herramientas de desarrollo que se emplearan para la realización de este proyecto.

Cambios en las prioridades planteadas en el inicio del proyecto por parte del cliente.

### 2.1.8 Costo estimado.

Tabla 1. Costo estimado

N°	RUBRO DE GASTOS	CANTIDAD	VALOR UNITARIO (dólares)	VALOR TOTAL (dólares)
	RECURSOS HUMANOS			
	Investigador (Autor del trabajo de grado)	984 horas	3,00	2952.00
	Subtotal			2952.00
	RECURSOS MATERIALES			
	Material de Escritorio	1	50.00	50.00
2	Material Bibliográfico (internet)	6 meses	25	150.00
	Empastado de trabajo de grado	1	50.00	50.00
	Subtotal		125.00	250.00
3	OTROS			

Logística (Transporte, alimentación, varios)	6 meses	25.00	150.00
Subtotal			3352.00
Imprevistos (5%)			167.60
<b>TOTAL, GASTOS</b>			<b>3519.60</b>

Nota: Esta tabla contiene la estimación de costos.

### 2.1.9 Involucrados.

Tabla 2. Involucrados

Nombre	Cargo
Robinson Llerena	Tutor
Byron Gangotena	Doctor Centro Médico UPS Campus SUR
Ligia Arellano	Enfermera Centro Médico UPS Campus SUR
Alejandro Llano	Estudiante Desarrollador del Proyecto de Titulación

Nota: Tabla de lista de involucrados

### 2.1.10 Recursos.

#### Recursos entregados por el cliente previo al desarrollo

Formatos en Word y Excel que los actores del centro médico utilizan actualmente para registrar todos los procesos que realizan.

## 2.2 Requerimientos funcionales

El sistema deberá contar con los siguientes módulos:

Gestión de Pacientes

Gestión de Autenticación Usuarios

Agendamiento de citas

Registro de Atención médica

Informes

Parte diario

Directrices Estratégicas Médicas

Reportes estadísticos

Indicadores morbilidad

Indicadores epidemiológicos

Inspecciones

Registro de control BPM (buenas prácticas médicas)

Informe de inspección observaciones preventivas, bar campus sur

Plan de vigilancia médico

### **2.2.1 Tipos y Características del usuario.**

Se han identificado cuatro tipos de usuarios:

**Doctor(a):** Es el encargado de realizar la atención médica, también tiene privilegios para agendar y re agendar citas, registrar y modificar pacientes, tomar datos previos a una cita por ejemplo signos vitales o exámenes, emitir certificados médicos, directrices estratégicas médicas, generar informes e ingresar información general al sistema como nuevas patologías y demás.

**Enfermero(a):** Es el (la) encargado(a) de registrar pacientes, agendar citas, tomar datos de los pacientes previo a la atención médica, generar informes. También tendrá privilegios para modificar información. El sistema no le permitirá realizar atención médica a este usuario.

**Paciente:** Deberá proporcionar información precisa para el correcto funcionamiento del sistema. No tendrá ningún acceso al sistema, únicamente recibirá notificaciones por correo electrónico cada vez que se agende o modifique una cita médica.

**Administrador del Sistema:** Usuario con conocimiento en el manejo del sistema con una previa capacitación. Encargado de manejar el sistema con responsabilidad sobre los criterios de permisos sobre los usuarios.

### **2.2.2 Ambiente Operativo.**

El sistema deberá correr en un navegador web con acceso a internet. El tipo de sistema operativo que se utilice no representa ningún limitante para correcto funcionamiento del producto.

Pese a que el producto contará con un diseño responsive (diseño adaptativo) que le permita adaptarse a diferentes tamaños de pantalla como dispositivos móviles, este sistema será desarrollado y probado principalmente para ser utilizado en un computador por lo tanto no se garantiza que el sistema funcione con total eficiencia en otros dispositivos.

### **2.2.3 Requerimientos de Interfaces Externas.**

**Computador:** Es donde se ejecutará el sistema y para su óptimo funcionamiento se requiere que las características básicas que deberá poseer este equipo sean:

Procesador Core 2 Duo 2.9 GHZ en adelante.

Windows 7 en adelante.

2 GB RAM.

Monitor 17"

**Internet:** Es importante que se asegure el acceso a internet ya que el sistema estará alojado en un servidor web y sin conexión no se podrá acceder al mismo. Se recomienda que el internet supere los 10mbps.

**Cámara Fotográfica:** El sistema soportará el ingreso de imágenes como información en varios módulos que así lo requieran. Para esto se podrá utilizar una cámara de características normales, pudiendo ser utilizada la cámara de un Smartphone de manera exitosa.

**Impresora:** El sistema permitirá generar informes en formato PDF y el usuario podría requerir imprimirlos.

## 2.4 Especificación ágil de requerimientos

Tabla 3. Requerimiento de gestión de usuarios

Especificación de Requerimientos Funcionales			
Código	Nombre	Fecha	Grado Necesidad
REQF-01	Gestión de usuarios	05/05/2017	Esencial
Descripción	El sistema debe permitir registrar y gestionar usuarios.		
Entradas	Proceso	Salidas	
Datos personales del usuario.	Se mostrará un formulario para ingresar la información necesaria del usuario. Se deberá definir un rol	Las salidas van dirigidas a: Administrador del sistema, Doctor/a, Enfermero/a. Mensaje de aviso: usuario registrado	

	específico para cada usuario según sus funciones.	satisfactoriamente.  Mensaje de error: en caso de no haber llenado algún campo obligatorio o haberlo llenado con información errónea.  Con las credenciales creadas el usuario podrá acceder al sistema y hacer uso del mismo según los privilegios que se le hayan asignado.
--	---	---

Nota: Tabla de requerimientos gestión de usuarios.

Tabla 4. Requerimientos Registro de Pacientes

Especificación de Requerimientos Funcionales			
Código	Nombre	Fecha	Grado Necesidad
REQF-02	Registrar Pacientes	15/05/2017	Esencial
Descripción	El sistema requiere el registro de todos los pacientes que vayan a ser atendidos.		
Entradas	Proceso	Salidas	
Datos personales del paciente.	Se mostrará un formulario para ingresar la información necesaria del paciente	Las salidas van dirigidas a: Doctor/a, Enfermero/a.  Mensaje de aviso: paciente registrado satisfactoriamente.  Mensaje de error: en caso de no haber llenado algún campo	

		obligatorio o haberlo llenado con información errónea.
--	--	--

Nota: Tabla con los requerimientos para el registro de pacientes.

Tabla 5. Requerimientos de Agendamiento de Citas

Especificación de Requerimientos Funcionales			
Código	Nombre	Fecha	Grado Necesidad
REQF-03	Agendamiento de citas	22/05/2017	Esencial
Descripción	Uno de los objetivos del sistema es lograr una organización de la agenda del doctor, para lo cual se podrá agendar citas mediante un calendario.		
Entradas	Proceso	Salidas	
Requerimiento de cita por parte del paciente o cita de seguimiento Especificación de datos para la cita por parte del paciente	Se validará la disponibilidad del horario requerido y se procederá a registrar la cita	Las salidas van dirigidas a: Doctor/a, Enfermero/a, Paciente. Mensaje de aviso: cita registrada satisfactoriamente. Mensaje de error: en caso de no haber llenado algún campo obligatorio o haberlo llenado con información errónea. Correo Electrónico: el paciente recibirá un correo de confirmación con los datos de la cita agendada o si hubiese alguna modificación en una cita previamente registrada.	

		Visualización de citas con filtro por fechas.
--	--	---

Nota: Tabla con los requerimientos para el agendamiento de citas.

Tabla 6. Requerimientos de Registro de Atención Médica

Especificación de Requerimientos Funcionales			
Código	Nombre	Fecha	Grado Necesidad
REQF-04	Registro de Atención médica	01/06/2017	Esencial
Descripción	Previo al agendamiento de una cita el Doctor podrá atender las citas según el cronograma para lo cual se le mostrará una vista que contendrá la historia clínica del paciente. En la misma podrá realizar la atención, registrar exámenes, emitir certificados y directrices médicas.		
Entradas	Proceso	Salida	
Evaluación médica del paciente, historiales clínicos. Especificación de datos para la atención por parte del paciente	Se validará la disponibilidad del horario requerido y se procederá a registrar la cita	Las salidas van dirigidas a: Doctor/a, Enfermero/a, Paciente. Mensaje de aviso: cita registrada satisfactoriamente. Mensaje de error: en caso de no haber llenado algún campo obligatorio o haberlo llenado con información errónea. Correo Electrónico: el paciente recibirá un correo de confirmación con los datos	

		de la cita agendada o si hubiese alguna modificación en una cita previamente registrada.
--	--	--

Nota: Tabla con los requerimientos para la atención médica.

Tabla 7. Requerimientos de Parte Diario

Especificación de Requerimientos Funcionales			
Código	Nombre	Fecha	Grado Necesidad
REQF-05	Parte diario	18/07/2017	Esencial
Descripción	Se mostrará un listado con los mismos datos del formato físico para este informe en base a las atenciones médicas que el Doctor haya realizado en cada día.		
Entradas	Proceso	Salida	
Registro de Atención Médica.	Se realizarán consultas a la base de datos para extraer la información necesaria en base a los registros que el usuario realice diariamente	<p>Las salidas van dirigidas a: Doctor/a, Enfermero/a.</p> <p>En caso de no haber realizado ninguna atención la lista se mostrará vacía.</p> <p>El usuario podrá exportar este informe en formato PDF.</p> <p>El sistema permitirá revisar los partes de todos los días que el usuario requiera.</p>	

Nota: Tabla con los requerimientos para el parte diario.

Tabla 8. Requerimientos de Informe de inspecciones

Especificación de Requerimientos Funcionales			
Código	Nombre	Fecha	Grado Necesidad
REQF-06	Informe de inspección observaciones preventivas.	01/09/2017	Esencial
Descripción	Se podrá realizar una inspección BPM en tiempo real haciendo uso del sistema web y generar un informe que reciba información detallada por el usuario y la procese en un formato establecido.		
Entradas	Proceso	Salida	
Ingreso de información obtenida momento de inspeccionar.	El sistema recibirá los datos que el usuario vaya ingresado mientras inspecciona y a la vez irá calculando la calificación según los parámetros indicados. Una vez finalizada esta inspección el usuario podrá realizar un informe mismo que el sistema le dará un formato	Las salidas van dirigidas a: Doctor/a, Enfermero/a. Resultado de la evaluación El usuario podrá exportar este informe en formato PDF.	

Nota: tabla con los requerimientos para el informe de inspecciones.

Tabla 9. Requerimientos de Indicadores morbilidad

Especificación de Requerimientos Funcionales			
Código	Nombre	Fecha	Grado Necesidad
REQF-07	Indicadores morbilidad	26/07/2017	Esencial
Descripción	El sistema mostrará informes estadísticos acerca de la morbilidad dividido en tres categorías: General, Áreas, CIE 10.		

Entradas	Proceso	Salida
Certificados médicos emitidos por el Doctor y con petición del paciente después de realizar una atención médica.	El sistema irá generando estos informes a medida que se vaya registrando la información, para que el usuario pueda visualizar y/o exportar a PDF estos informes en cualquier momento.	Las salidas van dirigidas a: Doctor/a, Enfermero/a. Informes actualizados en tiempo real. El usuario podrá exportar estos informes en formato PDF. Se podrá filtrar por fechas según el usuario requiera.

Nota: Tabla con los requerimientos para los reportes de morbilidad.

Tabla 10. Requerimientos de Informes Epidemiológicos

Especificación de Requerimientos Funcionales			
Código	Nombre	Fecha	Grado Necesidad
REQF-08	Informes epidemiológicos	07/08/2017	Esencial
Descripción	El sistema mostrará informes estadísticos epidemiológicos divididos en cuatro categorías: Género, Edad, Patologías más comunes, CIE 10.		
Entradas	Proceso	Salida	
Información ingresada en cada atención médica.	El sistema irá generando estos informes a medida que se vaya registrando la información, para que el usuario pueda visualizar y/o exportar a PDF estos informes en cualquier momento.	Las salidas van dirigidas a: Doctor/a, Enfermero/a. Informes actualizados en tiempo real. El usuario podrá exportar estos informes en formato PDF.	

		Se podrá filtrar por fechas según el usuario requiera.
--	--	--

Nota: Tabla con los requerimientos para los reportes epidemiológicos.

Tabla 11. Requerimientos de Plan de Vigilancia Médico

Especificación de Requerimientos Funcionales			
Código	Nombre	Fecha	Grado Necesidad
REQF-09	Plan de vigilancia médico	08/09/2017	Esencial
Descripción	El sistema pretende lograr una organización de la agenda de planificaciones para vigilancia mediante un calendario.		
Entradas	Proceso	Salidas	
Generación de un área de planificación.	Se validará la disponibilidad del horario requerido y se procederá a registrar la planificación	<p>Las salidas van dirigidas a: Doctor/a, Enfermero/a.</p> <p>Mensaje de aviso: planificación registrada satisfactoriamente.</p> <p>Mensaje de error: en caso de no haber llenado algún campo obligatorio o haberlo llenado con información errónea.</p>	

Nota: Tabla con los requerimientos para el plan de vigilancia médico.

## **2.5 Análisis y Diseño del Sistema**

Se utilizarán herramientas típicas de Scrum como son Product Backlog, Priorización, Plan de entregas los cuales nos permiten equilibrar la carga de trabajo y tener una estimación clara de cuánto tiempo y esfuerzo serán requeridos para cumplir las actividades propuestas en cada sprint mismo que deberá ser culminado y presentado según las fechas establecidas previo a los análisis realizados.

### **2.5.1 Priorización.**

La herramienta para realizar esta estimación son las historias de usuario las mismas que también detallan los requerimientos funcionales del sistema agrupándolas según los módulos que van a ser desarrollados.

### **2.5.2 Especificación de la carga de trabajo para el desarrollo.**

Esta especificación se hace tomando en cuenta la disponibilidad de tiempo de la persona encargada de desarrollar el sistema. La jornada de trabajo constara de 6 horas diarias, por 5 días a la semana (lunes a viernes).

### **2.5.3 Product Backlog.**

Es básicamente un listado de características del producto a construir priorizado por el Product Owner. “Es importante que exista una clara priorización, ya que es esta priorización la que determinará el orden en el que el equipo de desarrollo transformará las características (ítems) en un producto funcional acabado” (Alaimo, 2013).

Este artefacto nos permite especificar la carga de trabajo según las actividades a realizarse.

Tabla 12. Product Backlog

Nombre de tarea	Días	Prioridad	Riesgo
<b>SPRINT 1</b>			
Gestión de usuarios	6 días	Alta	Alto
Gestión de pacientes	5 días	Alta	Alto
Agendamiento de citas	8 días	Alta	Alto
Atención médica	33 días	Alta	Alto
Parte diario	6 días	Alta	Alto
<b>SPRINT 2</b>			
<b>Reportes estadísticos</b>			
Indicadores morbilidad			
General	3 días	Alta	Alto
Áreas	3 días	Alta	Alto
CIE 10	2 días	Alta	Alto
Informes epidemiológicos			
Género	2 días	Alta	Alto
Edad	3 días	Alta	Alto
Patologías más comunes	2 días	Alta	Alto
CIE 10	3 días	Alta	Alto
<b>SPRINT 3</b>			
<b>Inspecciones</b>			
Registro de control BPM	9 días	Media	Alto
Informe de inspección observaciones preventivas	5 días	Media	Alto
<b>SPRINT 4</b>			

Plan de vigilancia médico	6 días	Media	Medio
Formularios para ingreso de información general	6 días	Alta	Alto
Editor de imágenes	3 días	Media	Bajo
<b>SPRINT 5</b>			
Implementación			
Proporcionar Demo	1 día	Alta	Bajo
Reunir Herramientas Necesarias	1 día	Media	Medio
Puesta en marcha de implementación	1 día	Alta	Alto
Manuales de Usuario	1 día	Alta	Bajo
Pruebas			
Pruebas Integrales	2 días	Alta	Alto
Pruebas de Aceptación de usuario	5 días	Alta	Alto
<b>SPRINT 6</b>			
Entrega	2 días	Alta	Alto
Informe de Puesta en Producción	3 días	Media	Bajo

Nota: Esta tabla contiene el artefacto Product Backlog.

#### **2.5.4 Priorización.**

Es importante definir la prioridad que tienen las actividades a desarrollarse ya que esto permite plantear alternativas que cumplan los propósitos en el caso de que el costo de la funcionalidad solicitada sea alto o su construcción no sea viable.

Tabla 13. Priorización

<b>Módulos</b>	<b>N°</b>	<b>Nombre Historia</b>	<b>Iteración</b>
<b>Gestión de usuarios, pacientes y atención Médica</b>	1	Gestión de usuarios	1
	2	Gestión de pacientes	1
	3	Agendamiento de citas	1
	4	Atención médica	1
	5	Parte diario	1
<b>Reportes Estadísticos</b>	6	Indicadores morbilidad	2
		General	2
		Áreas	2
		CIE 10	2
	7	Informes epidemiológicos	2
		Género	2
		Edad	2
		Patologías más comunes	2
	CIE 10	2	
<b>Inspecciones</b>	8	Registro de control BPM	3
	9	Informe de inspección observaciones preventivas	3
<b>Gestión de información general</b>	10	Plan de vigilancia médico	4
	11	Formularios para ingreso de información general	4
	12	Editor de imágenes	4

Nota: esta tabla contiene la priorización.

### 2.5.5 Plan de Entregas.

Tabla 14. Plan de entregas

Actividad	Días	Comienzo	Fin
<b>DESARROLLO</b>			
Preparar ambiente de desarrollo	5 días	vie 14/04/17	jue 20/04/17
Estructurar Base de Datos	10 días	vie 21/04/17	jue 04/05/17
<b>SPRINT 1</b>			
Gestión de usuarios	6 días	vie 05/05/17	vie 12/05/17
Gestión de pacientes	5 días	lun 15/05/17	vie 19/05/17
Agendamiento de citas	8 días	lun 22/05/17	mié 31/05/17
Atención médica	33 días	jue 01/06/17	lun 17/07/17
Parte diario	6 días	mar 18/07/17	mar 25/07/17
<b>SPRINT 2</b>			
<b>Reportes estadísticos</b>			
Indicadores morbilidad			
General	3 días	mié 26/07/17	vie 28/07/17
Áreas	3 días	lun 31/07/17	mié 02/08/17
CIE 10	2 días	jue 03/08/17	vie 04/08/17
Informes epidemiológicos			
Género	2 días	lun 07/08/17	mar 08/08/17
Edad	3 días	mié 09/08/17	vie 11/08/17
Patologías más comunes	2 días	lun 14/08/17	mar 15/08/17
CIE 10	3 días	mié 16/08/17	vie 18/08/17
<b>SPRINT 3</b>			
<b>Inspecciones</b>			

Registro de control BPM	9 días	lun 21/08/17	jue 31/08/17
Informe de inspección observaciones preventivas	5 días	vie 01/09/17	jue 07/09/17
SPRINT 4			
Plan de vigilancia médico	6 días	vie 08/09/17	vie 15/09/17
Formularios para ingreso de información general	10 días	lun 18/09/17	vie 29/09/17
Editor de imágenes	3 días	lun 02/10/17	mié 04/10/17
SPRINT 5			
Implementación			
Proporcionar Demo	5 días	jue 05/10/17	mié 11/10/17
Reunir Herramientas Necesarias	3 días	jue 12/10/17	lun 16/10/17
Puesta en marcha de implementación	7 días	mar 17/10/17	mié 25/10/17
Manuales de Usuario	3 días	jue 26/10/17	lun 30/10/17
Pruebas			
Pruebas Integrales	10 días	mar 31/10/17	lun 13/11/17
Pruebas de Aceptación de usuario	10 días	mar 14/11/17	lun 27/11/17
SPRINT 6			
Entrega	2 días	mar 28/11/17	mié 29/11/17
Informe de Puesta en Producción	15 días	jue 30/11/17	mié 20/12/17

Nota: Esta tabla contiene el plan de entregas.

### **2.5.6 BPMS (Suite de gestión de procesos empresariales).**

Permite definir el ciclo de vida de los procesos de negocio aumentando la eficiencia entre las personas y procesos. Es una herramienta de automatización de procesos.

“Le ayuda a planificar sus procesos cotidianos para identificar y eliminar los cuellos de botella, controlar los costos de su empresa, hacer sus procesos de día a día lo más eficiente posible, y garantizar la eficacia de las personas que participan en sus procesos” (ultimus, 2017).

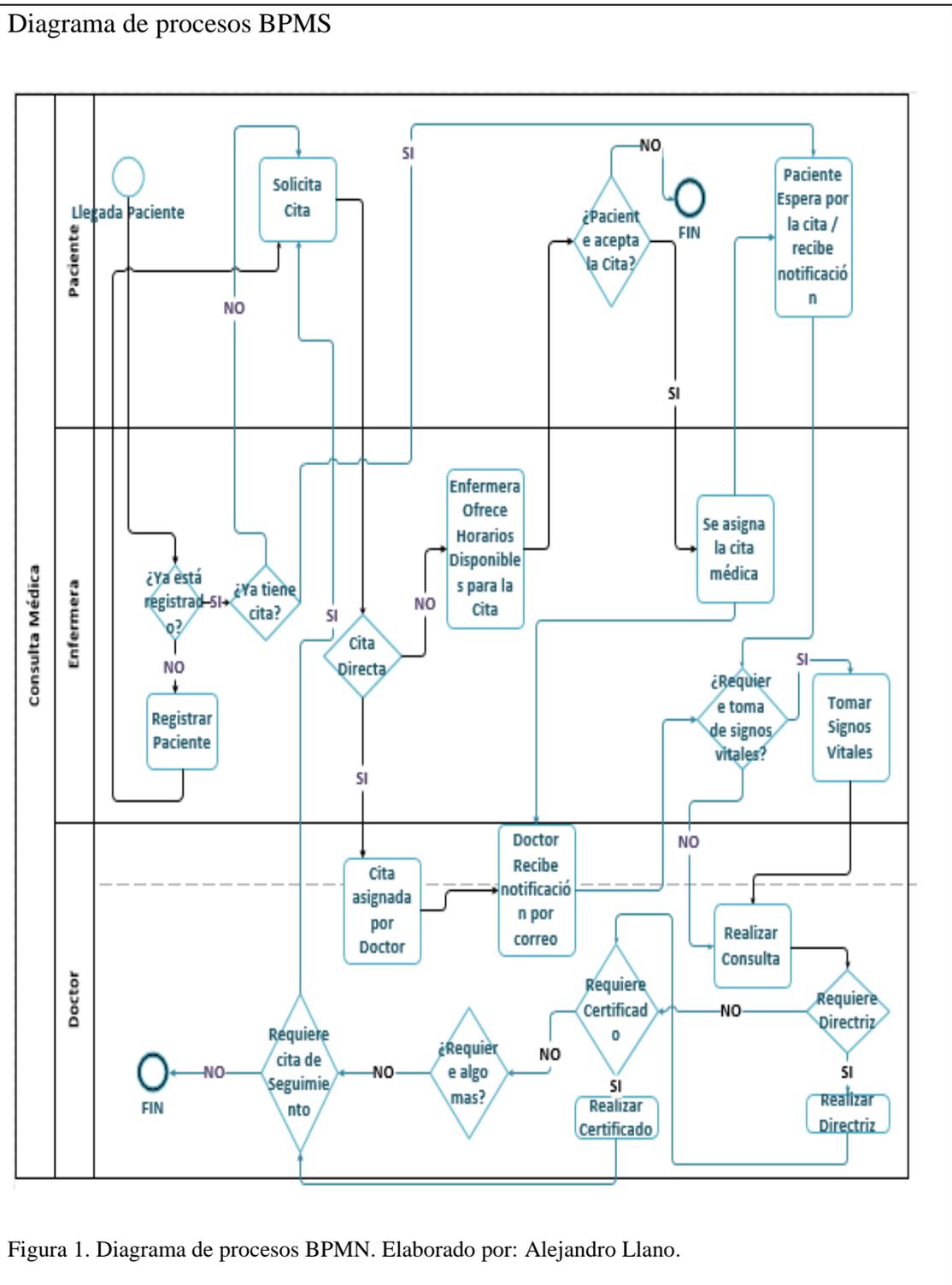


Figura 1. Diagrama de procesos BPMN. Elaborado por: Alejandro Llano.

### 2.5.7 WBS.

Estructura de desglose del trabajo (WBS) es un documento que descompone el producto resultante del proyecto en paquetes de trabajo individuales, de tal forma que cada nivel muestra los paquetes de trabajo que forman parte de cada módulo.

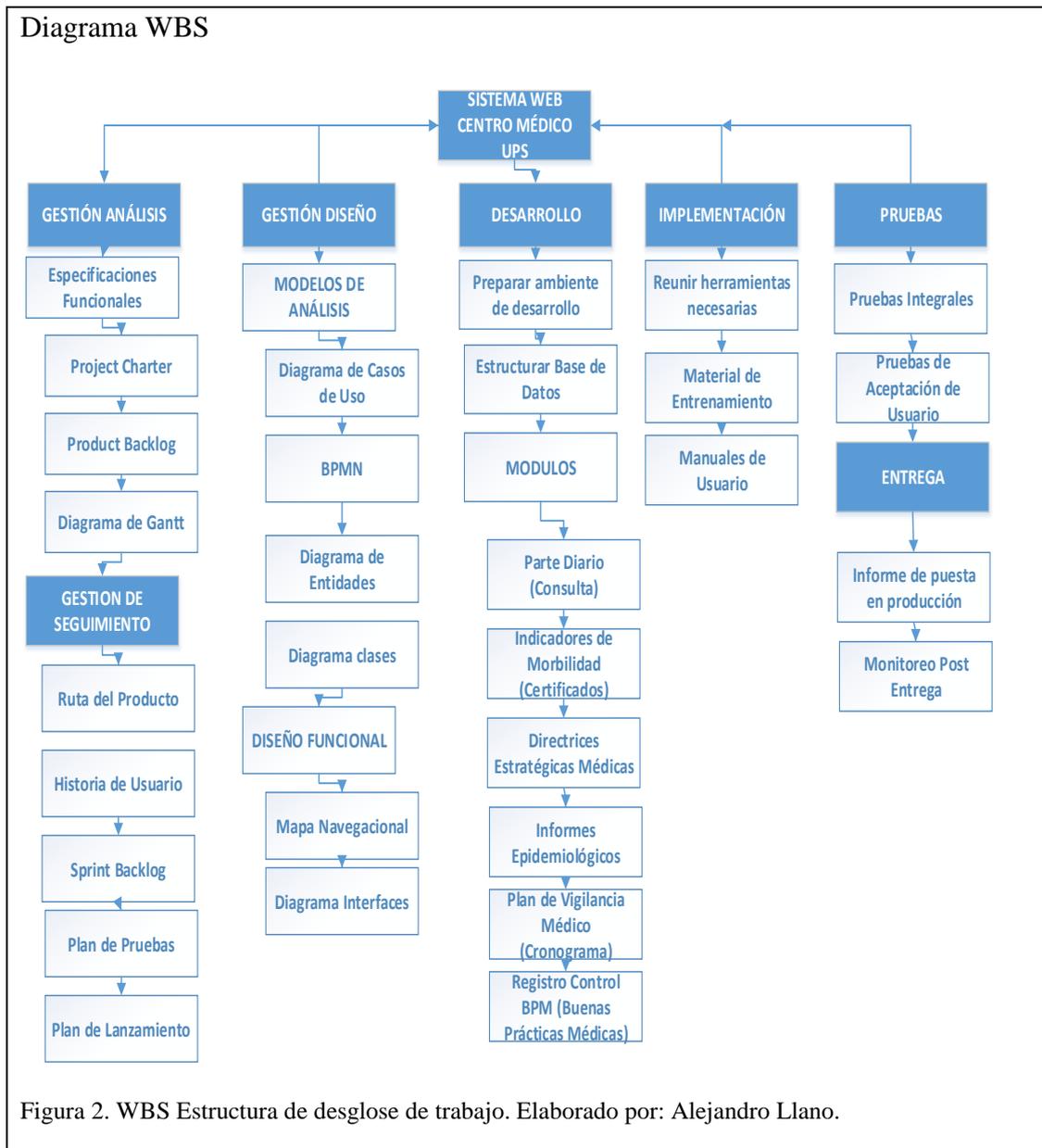


Figura 2. WBS Estructura de desglose de trabajo. Elaborado por: Alejandro Llano.

### 2.5.8 Diagramas UML aplicados al sistema.

El Lenguaje de Modelado Unificado son métodos de análisis y diseño orientados a objetos, es llamado un lenguaje de modelado. El lenguaje de modelado es una notación principalmente gráfica que se usa para expresar un proceso que indica los pasos que se deben seguir para llegar a un diseño.

#### 2.5.8.1 Casos de Uso.

Representa la forma en como un Actor opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan.

Las siguientes ilustraciones se tratan de representar con que procesos interactúan los actores que participan en este sistema.

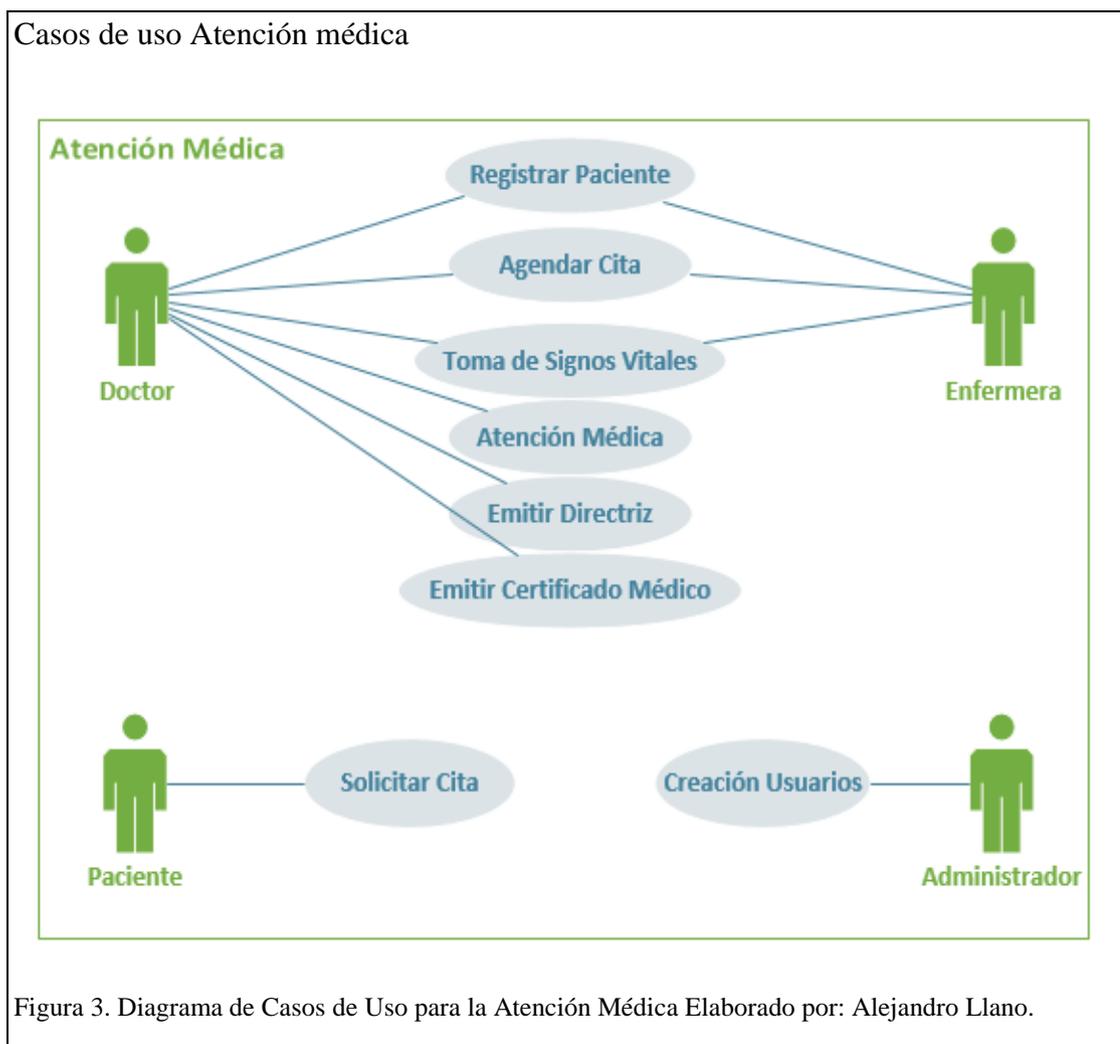


Figura 3. Diagrama de Casos de Uso para la Atención Médica Elaborado por: Alejandro Llano.

### Casos de uso inspecciones preventivas

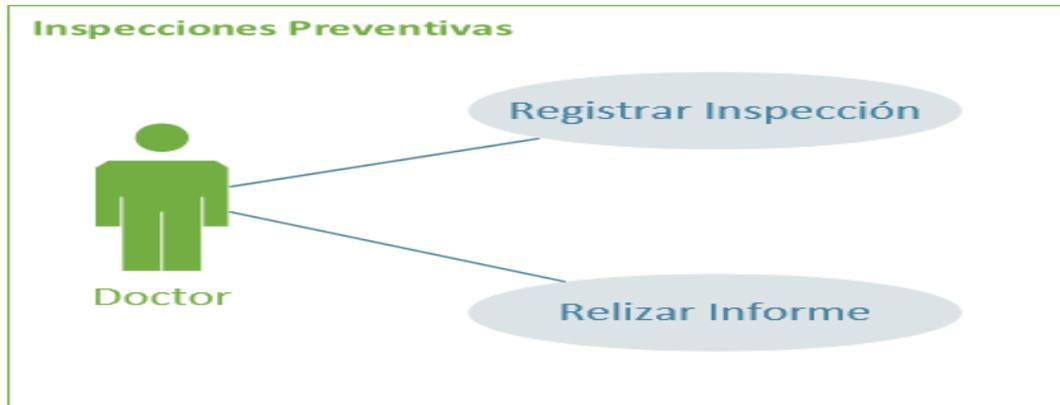


Figura 4. Diagrama de Casos de Uso para las Inspecciones Preventivas Elaborado por: Alejandro Llano.

### Casos de uso Plan de vigilancia médico

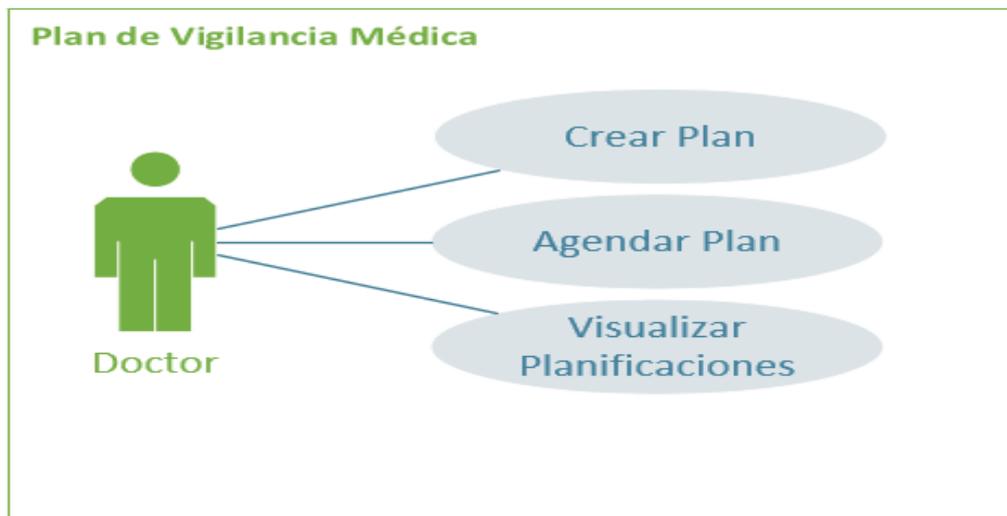
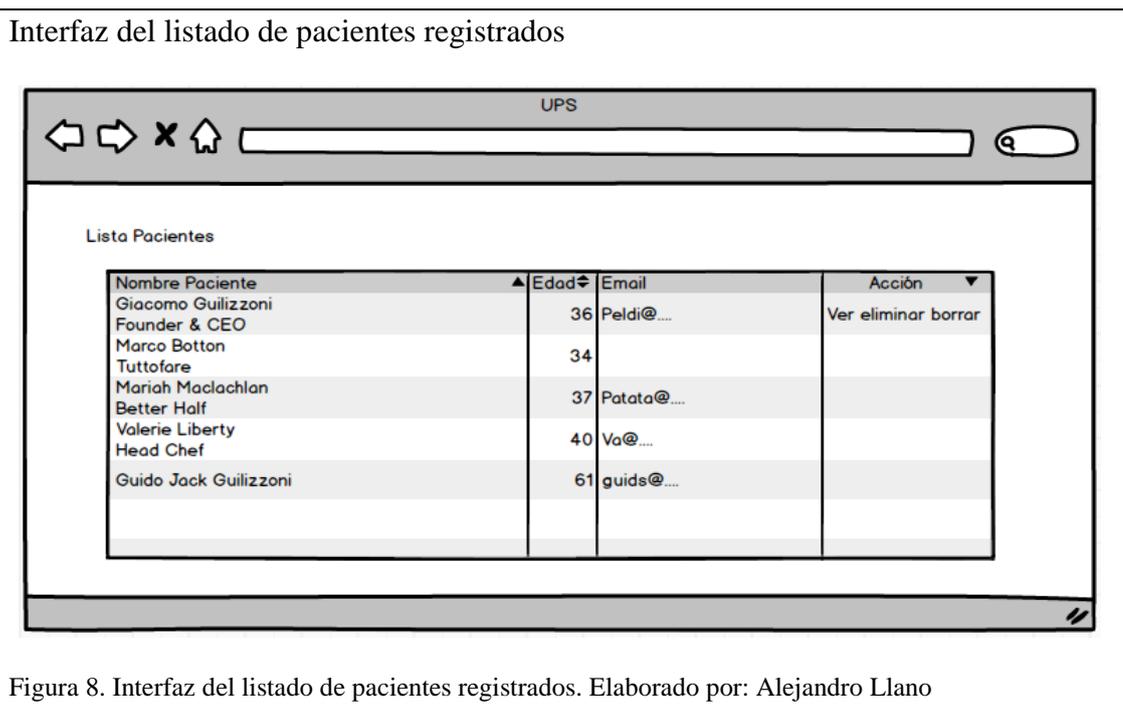
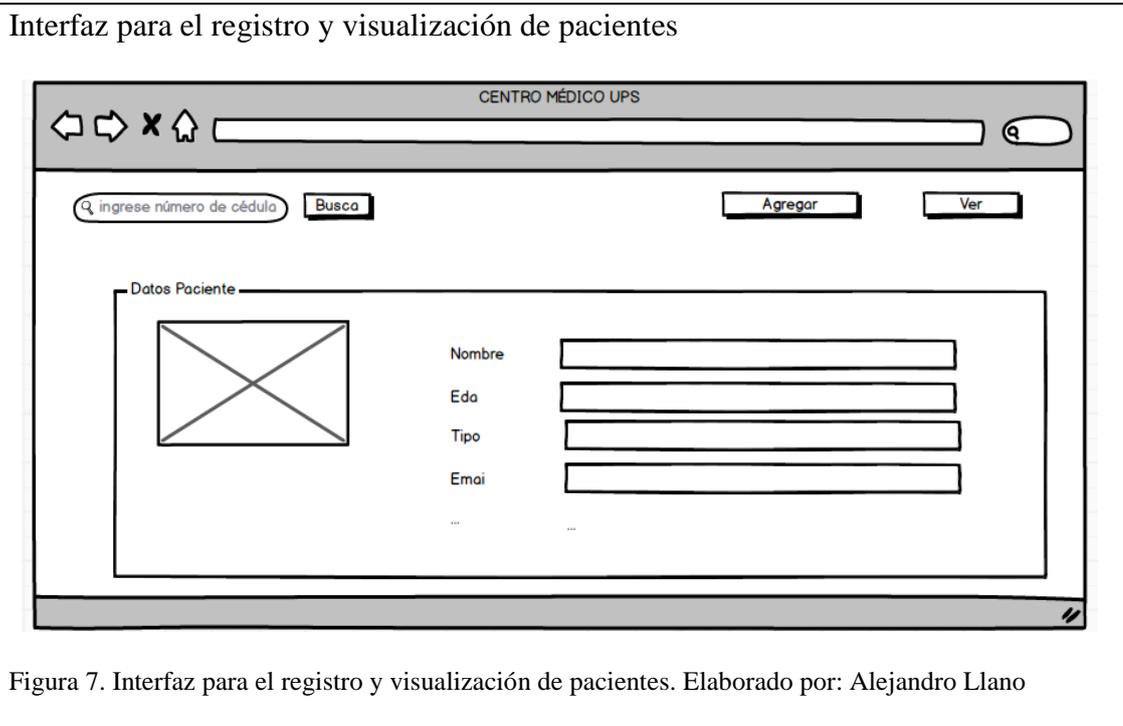


Figura 5. Diagrama de Casos de Uso para el Plan de vigilancia Elaborado por: Alejandro Llano.



### 2.5.8.3 Interfaces del sistema.

A través de ilustraciones que permiten apreciar cómo serán las interfaces del sistema una vez desarrollado podemos tener una idea más clara de cómo se efectuarán los procesos referentes a cada módulo antes detallados.



## Interfaz para agendar citas

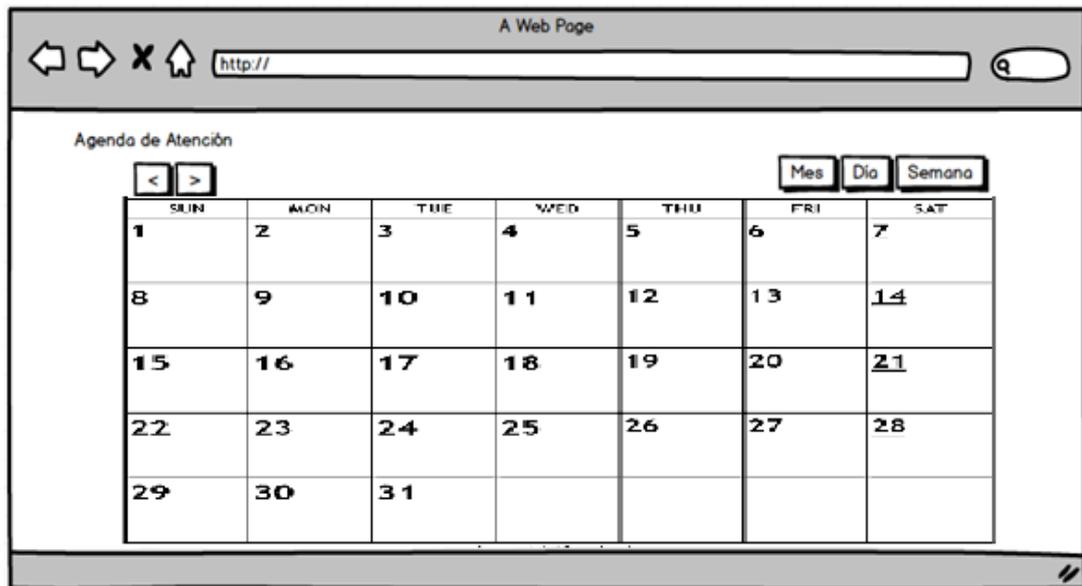


Figura 9. Interfaz que visualizará el usuario para agendar citas. Elaborado por: Alejandro Llano

## Interfaz del listado de citas

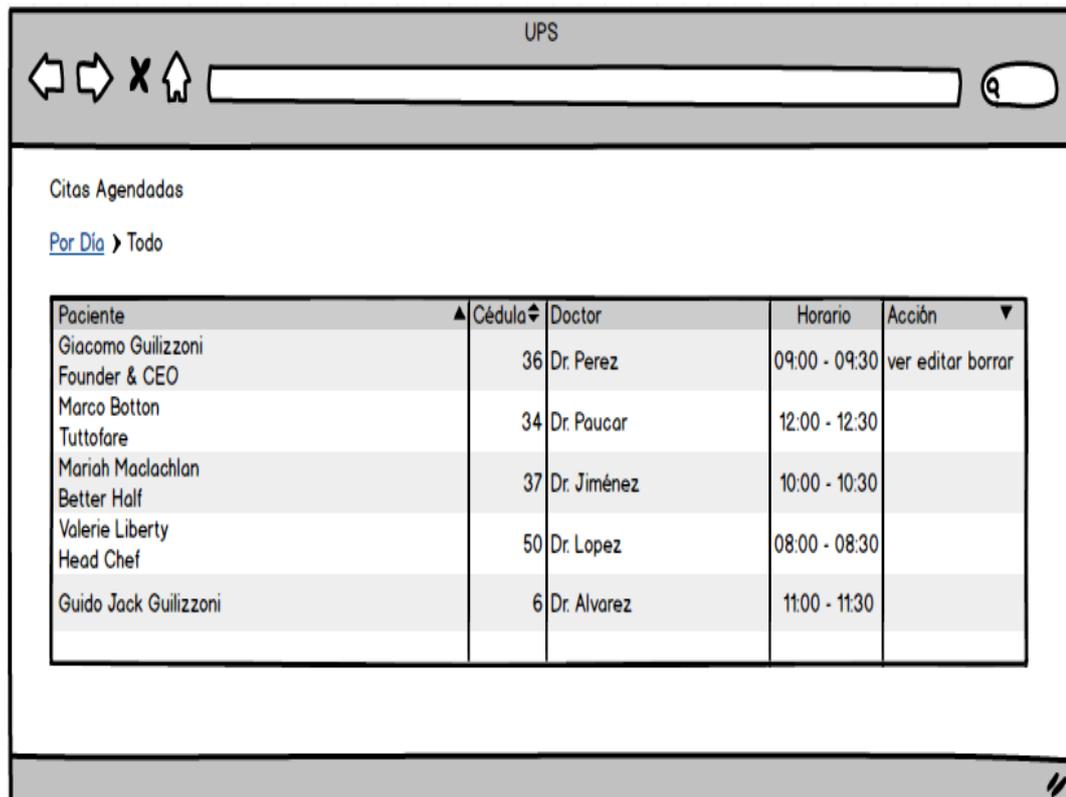


Figura 10. Interfaz del listado de citas. Elaborado por: Alejandro Llano

## Interfaz del perfil de paciente

UPS

Perfil Paciente: -----

Cédula .....

Fecha Nacimiento .....

Edad .....

Tipo Sangre .....

Email .....

Historiales

Consultas

Medicaciones

Signos Vitales

Patologías

Medicamento ▲	Cantidad ⇅	Observaciones ▼
Paracetamol	36	Tomar despues de almuerzo

Patología Seleccione Patología

Detalle Ingrese un detalle de la consulta

Emitir certificado Emitir Directriz Médica Agendar cita de seguimiento

Figura 11. Interfaz del perfil de paciente. Elaborado por: Alejandro Llano.

## Interfaz de parte diario

UPS

Parte Diario de Atención

Nombre: Doctor

Fecha: -/-/

Nombre ▲	Edad ⇅	Área	Cargo ▲	Patología ▲	Certificado
Giacomo Guilizzoni Founder & CEO	36		Docente		
Marco Botton Tuttofare	34		Docente		
Mariah Maclachlan Better Half	37		Docente		
Valerie Liberty Head Chef	20		Docente		

Figura 12. Interfaz de parte diario. Elaborado por: Alejandro Llano.

### Interfaz de indicadores de morbilidad

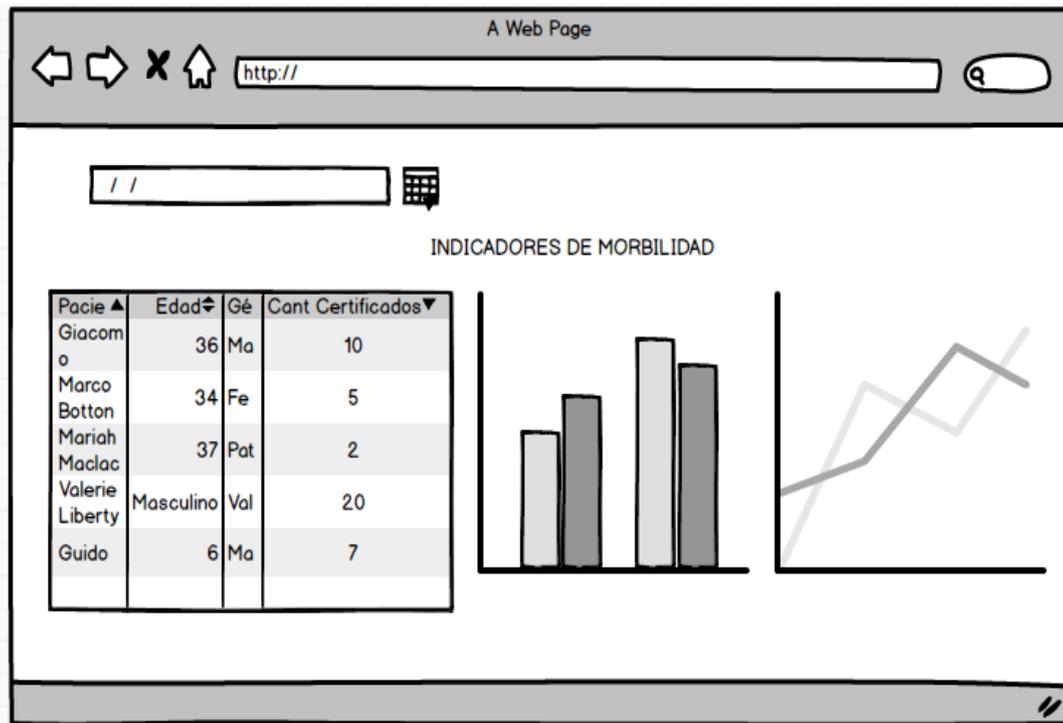


Figura 13. Interfaz de indicadores de morbilidad. Elaborado por: Alejandro Llano.

### Interfaz de Indicadores Epidemiológicos

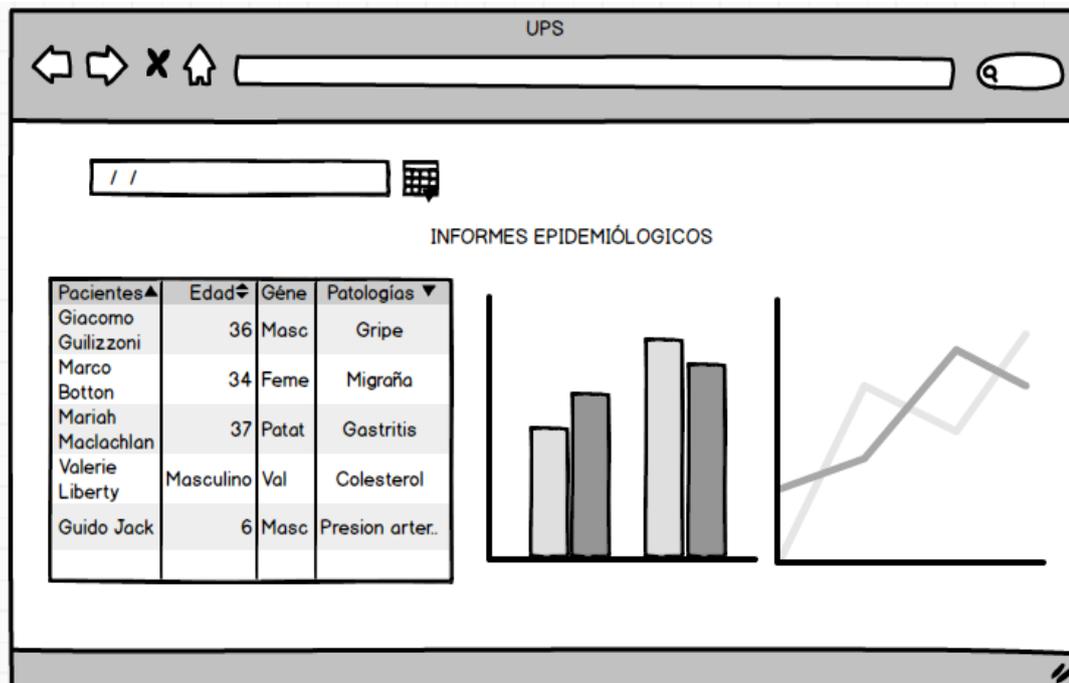


Figura 14. Interfaz de Indicadores Epidemiológicos. Elaborado por: Alejandro Llano.

### Interfaz lista de Inspecciones BPMS

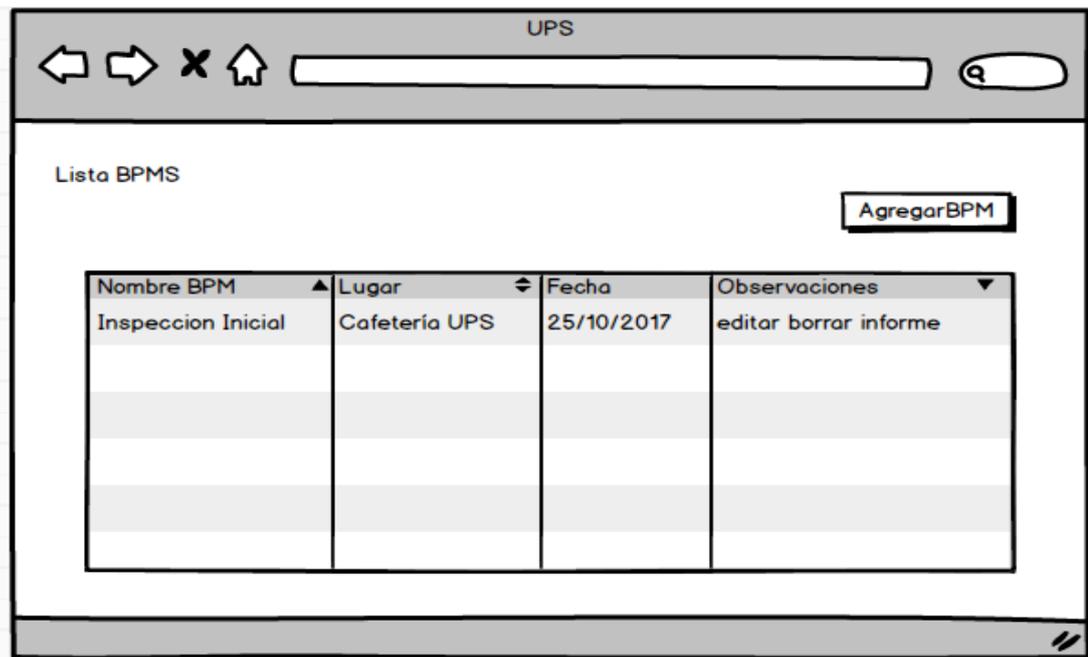


Figura 15. Interfaz para visualizar la lista de Inspecciones BPMS. Elaborado por: Alejandro Llano.

### Interfaz del registro de Inspecciones BPMS.

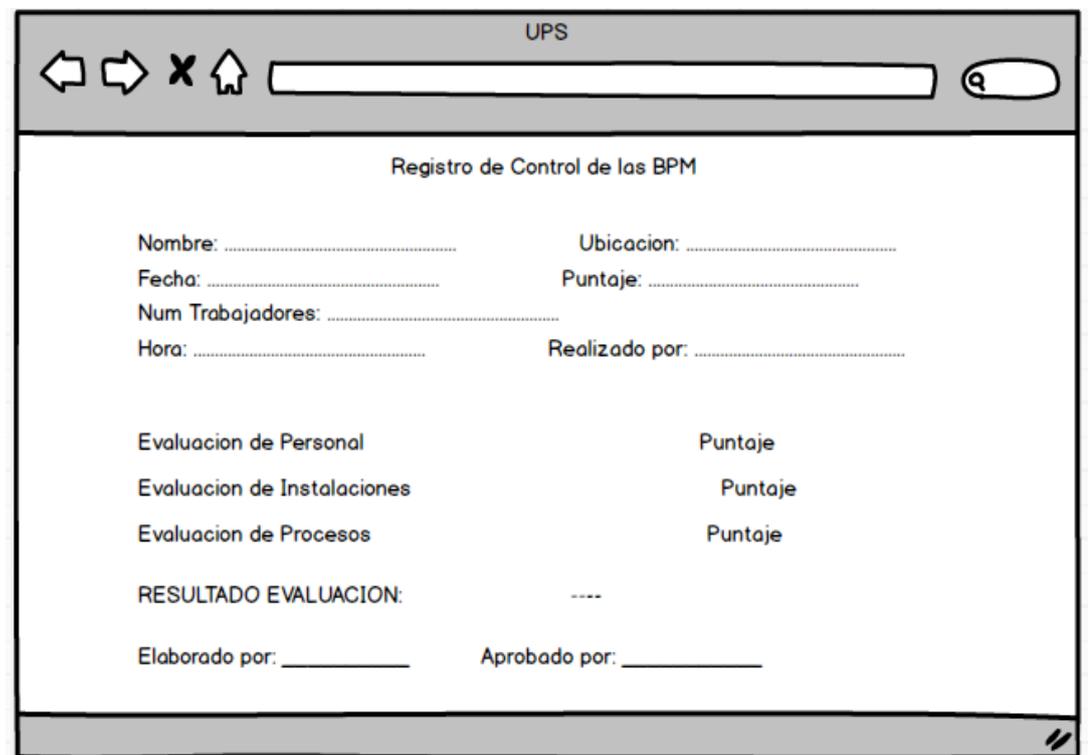


Figura 16. Interfaz del registro de Inspecciones BPMS. Elaborado por: Alejandro Llano.

## Interfaz de registro del Informe de BPMS

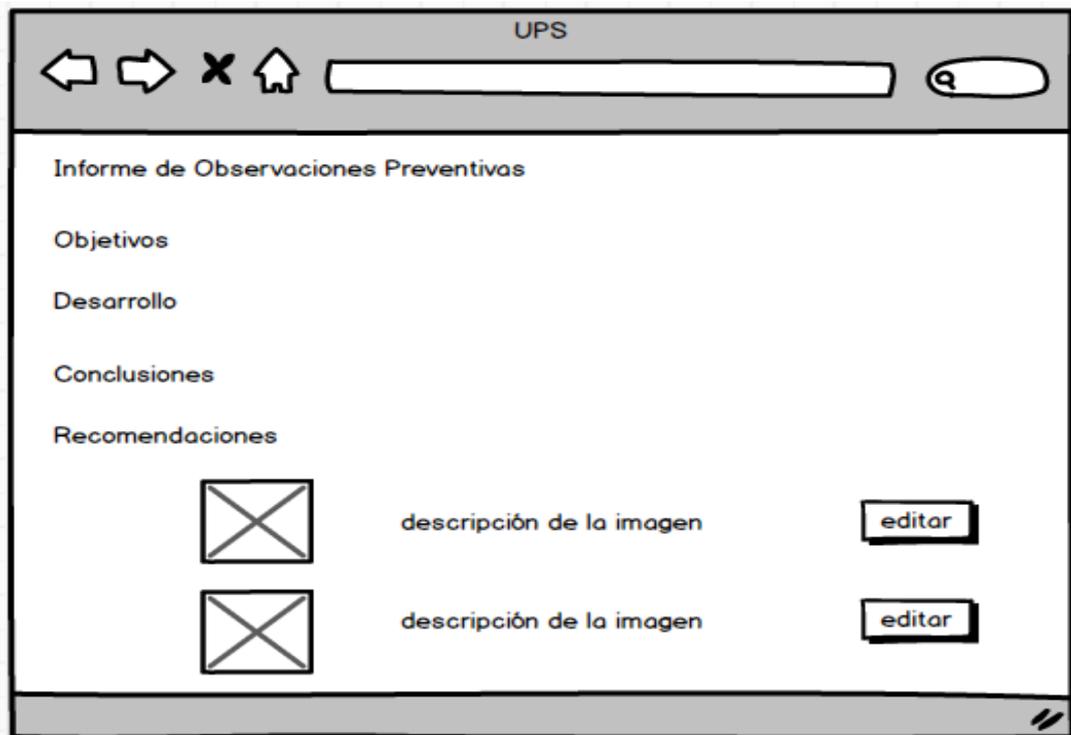


Figura 17. Interfaz de registro del Informe de BPMS. Elaborado por: Alejandro Llano.

## 2.6 Herramientas de desarrollo

### 2.6.1 PHP.

Es un lenguaje de programación utilizado para la creación de páginas web dinámicas. Es de código abierto, por lo cual utilizarlo no contempla costes. “La función básica de PHP es leer datos de formularios y convertirlos en variables PHP. Posteriormente, las variables pueden, por ejemplo, ser introducidas en una base de datos o enviadas por correo electrónico” (1and1, 2016).

Las ventajas de PHP incluyen su integración con el protocolo de Internet y un amplio soporte de diferentes modelos de bases de datos. Este lenguaje de programación es usado por más de un 82% de las páginas web del mercado y

suele ser, por lo tanto, pre instalado por la mayoría de proveedores de alojamiento web (1and1, 2016).

La codificación creada con este lenguaje se procesa únicamente en el lado del servidor mediante el intérprete de PHP que procesa los comandos y los envía al navegador, a diferencia de otros lenguajes de programación como JavaScript que ejecuta su codificación del lado del cliente o de quien realiza las peticiones.

Los tiempos de carga de una página web tienen gran importancia ya que al ser reducidos provocan un efecto positivo que será evidenciado por parte de los usuarios y no solo eso sino que además según (1and1, 2016) “será recompensado con un mejor posicionamiento en buscadores como Google” y con la utilización de este lenguaje de programación tendremos en buena parte asegurado este beneficio. Es por esto y lo mencionado anteriormente que se ha optado por usar esta herramienta para desarrollar este proyecto ya que se considera la mejor alternativa para lograr los objetivos planteados.

### **2.6.2 CodeIgniter.**

Es un entorno de desarrollo web escrito en PHP que como principal característica tiene a su capacidad de acelerar y optimizar el desarrollo de aplicaciones gracias a un diseño de software bastante ligero y que no dista demasiado de la programación tradicional. “El código fuente de CodeIgniter es distribuido con una licencia MIT (Massachusetts Institute of Technology) y puede descargarse desde la plataforma GitHub. La última versión estable se ofrece para su descarga gratuita en la página oficial del proyecto” (1and1, 2017).

Al haber elegido el lenguaje de programación PHP para el desarrollo de este proyecto se considera también que lo más idóneo es utilizar un framework que

potencie el desarrollo como lo es CodeIgniter el cual tiene varias ventajas como la versatilidad, ligereza y su característica más relevante que es su capacidad de trabajar la con la mayoría de los entornos o servidores, según (desarrolloweb, 2009) “Este framework puede trabajar incluso en sistemas de alojamiento compartido, donde sólo tenemos acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración”. Podemos resaltar su capacidad de adaptación ya que en la actualidad Codeigniter es compatible con la versión PHP 7 y posteriores, lo que hace que se pueda utilizar en cualquier servidor, siendo soportado incluso por servidores con versiones desactualizadas, lo cual es muy beneficioso ya que al momento de poner el sistema en un ambiente de producción la compatibilidad no será un problema. Todo esto sumado a su notable ligereza al poseer un núcleo sumamente ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes cantidades de código. “La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente”. (desarrolloweb, 2009)

### **2.6.3 Base de Datos.**

Se utilizará el gestor de base de datos PhpMyAdmin el cual está basado en MySQL que es un sistema de administración de bases de datos para bases de datos relacionales.

Es un programa escrito en PHP destinado a la administración de MySQL a través de la web. Resultando una aplicación muy completa que permite realizar todas las operaciones que se manejan en bases de datos SQL.

“El administrador de bases de datos por defecto en muchos paneles de control comerciales pueden ser cPanel, Plesk o DirectAdmin” (vozidea, 2013), Esto facilitará

la transición de un ambiente de desarrollo a un ambiente de producción en un hosting que será plenamente compatible en este aspecto.

Luego de haber analizado las características de esta herramienta se puede concluir que será de gran ayuda al momento de desarrollar el sistema ya que nos da una idea clara de que el sistema se adaptará de manera adecuada una vez implementado en un hosting.

#### **2.6.4 Bootstrap.**

La parte del diseño será desarrollado con Bootstrap, es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Fue desarrollado por Twitter y una de las mayores ventajas que posee es su capacidad de adaptación a los distintos navegadores y a visualizaciones responsive que podrían ser requeridas al acceder desde un dispositivo móvil.

Bootstrap posee plantillas CSS y librerías JavaScript que nos permiten integrar el framework en nuestro proyecto, además posee otras características esenciales como son:

Permite crear interfaces que se adapten a los diferentes navegadores tradicionales y también en dispositivos móviles.

Ha demostrado compatibilidad con las principales librerías JavaScript, como JQuery.

Ofrece un diseño iterativo y adaptable usando estándares como CSS3 y HTML5.

#### **2.6.5 CSS.**

Esta tecnología nos permite tener un mayor control sobre nuestra página web, potenciando el diseño el cual es muy limitado si únicamente se hiciera uso de

HTML. CSS son las siglas de Cascading Style Sheets, en español hojas de estilo en cascada.

### **2.6.6 JavaScript.**

El desarrollo aplicaciones web evidenció las limitadas prestaciones del lenguaje básico que se tiene con HTML ya solamente provee de elementos que actúan sobre el texto y su estilo, lo cual es un limitante al momento de desarrollar un sistema que requiera implementar características más avanzadas. “La temprana aparición de este lenguaje, es posiblemente la causa de que se haya convertido en un estándar soportado por todos los navegadores actuales, a diferencia de otros, que solo funcionan en los navegadores de sus firmas creadoras” (uv, 2017).

Según (uv, 2017) “Netscape fue quien inició la implementación de JavaScript y posteriormente, una alianza entre Netscape y Sun, creadora del lenguaje Java, permitió que JavaScript tomase la debida consistencia, definiéndose como una variante de Java, pero mucho más simple de usar. Esto no significa que JavaScript sea Java simplificado o reducido. Salvo el nombre y la sintaxis, JavaScript no tiene mucho en común con Java, pero cumple su propósito de lenguaje auxiliar del HTML en los navegadores, y sólo en ellos ya que no es posible utilizarlo fuera del entorno de las páginas”.

### **2.6.7 Herramienta de entorno de desarrollo XAMPP.**

Este programa está liberado bajo la licencia GNU y actúa como un servidor web capaz de interpretar páginas dinámicas. Xampp no es para uso en producción, sólo es un entorno de desarrollo local. Está configurado lo más abierto posible, permitiendo que el desarrollador trabaje sin restricciones. Se recomienda su uso para un entorno de desarrollo ya que emula cómo se comportará el sistema desarrollado al subirlo a

un servidor web, esta transición será significativamente similar ya que el servidor web donde se buscará alojar el sistema poseerá las mismas características vistas con Xampp.

### **2.6.8 Metodología de desarrollo SCRUM.**

Al tratarse de un proyecto de software con varios módulos los cuales deben satisfacer los requerimientos del cliente, cumplir con un cronograma y sobre todo debe asegurar un desarrollo de ingeniería que cumpla estándares de calidad se optará por una metodología ágil como Scrum que es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para obtener el mejor resultado posible de un proyecto.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Resultando una metodología muy apropiada para proyectos donde se necesita obtener resultados pronto y donde los requisitos son cambiantes o imprecisos. “Al estar orientado a resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto esta metodología es la indicada” Palacio, J. (2014).

#### **2.6.8.1 Características de Scrum.**

Ayuda a llevar a cabo las actividades diarias, proporcionando las actualizaciones que el equipo de desarrollo necesita para trabajar en los proyectos.

“Si los desarrolladores no saben cómo abordar las tareas, el Scrum Master los juntará a todos para explicarles en qué consisten y qué tarea abordará cada uno” (iebschool, 2015).

El Scrum Master facilita el sprint final cuando los miembros del equipo presenten a los stakeholders el proyecto, actúa como guía y moderador de las presentaciones en muchos casos.

“La figura del Scrum Master refuerza la idea de que esta metodología mejora el trabajo en equipo, centrando todos los esfuerzos en conseguir un mismo objetivo para satisfacer las necesidades de los stakeholders. Al igual que el Product Owner, el Scrum Master trabaja con otras figuras como son los stakeholders, y representa al equipo de trabajo en las reuniones de Scrum of Scrums. En estas reuniones es donde el Scrum Master presenta los temas de diseño técnico y las acciones llevadas a cabo por el equipo” (iebschool, 2015).

#### ***2.6.8.2 Fases Scrum.***

El desarrollo de proyectos con Scrum requiere de un procedimiento organizado cuya primera actividad consiste en la planificación, la cual define una versión de la pila del producto junto con una estimación de costos y una agenda. Si se trata de un nuevo sistema, esta fase abarcará la visión y el análisis. Si se tratara de la mejora de un sistema existente se realizaría un análisis de alcance más limitado. Esta fase también define la arquitectura diseñando la implementación de las funcionalidades.

Una vez culminada esta fase empieza el desarrollo de sprints de funcionalidad de las versiones con respeto continuo a las variables de tiempo, requisitos, costo y competencia. El sistema va evolucionando a través de múltiples iteraciones de desarrollo para preparar la presentación de las versiones, incluyendo la documentación y pruebas.

#### ***2.6.8.3 Proceso.***

El proyecto a realizarse inicia con la elaboración del Product Backlog que es un archivo que recoge el conjunto de tareas, requerimientos y las funcionalidades

requeridas por el proyecto. “Cualquier miembro del equipo puede modificar este documento, pero el único con autoridad para agregar prioridades es el Product Owner, responsable del documento” (iebschool, 2015).

En la segunda etapa se define el Sprint Backlog, documento que recoge las tareas a realizar y se asigna un responsable para finalmente asignar las horas de trabajo que va a suponer realizar cada una de ellas. El Sprint es el periodo en el que se realizan y entregan todas las tareas planificadas hasta finalizar con todas las entregas planificadas.

## **2.7 Implementación**

La herramienta que soportará al sistema y que deberá ser instalada en primer lugar es Xampp 7.1.1 que fue la versión utilizada misma que puede ser obtenida de su página oficial <https://www.apachefriends.org/es/index.html>. Esta herramienta funcionará como servidor web para el entorno de desarrollo del proyecto ya que es una distribución de Apache fácil de instalar que contiene MariaDB, PHP y Perl. No requiere de configuraciones especiales.

Como segundo paso se requiere de un IDE (Entorno de Desarrollo Integrado) <https://netbeans.org/> para codificar el sistema el cual debe soportar el lenguaje de programación PHP 7. En este proyecto se ha optado por NetBeans 8.2 sin embargo no habría ningún inconveniente si se usa otro IDE.

El proyecto estará basado en el framework CodeIgniter 3 mismo que se puede obtener de la página oficial <https://codeigniter.com/> y se debe agregar a nuestro proyecto PHP para que al momento de programar sean identificadas sus librerías y su arquitectura de desarrollo.

Otra herramienta utilizada es Bootstrap que también la podemos obtener de la página oficial <https://getbootstrap.com/> y consiste en un conjunto de librerías que permiten dar un diseño a nuestro sistema web. Esta librería también debe ser agregada al proyecto y referenciada desde las clases que requieran diseño.

### **2.7.1 Codificación relevante.**

#### ***2.7.1.1 Conexión a base de datos.***

La base de datos será creada con la utilización de la herramienta phpMyAdmin antes detallada cuyo funcionamiento es muy similar a los gestores de bases de datos tradicionales.

La utilización del Framework mencionado nos da notables beneficios ya que nos permite usar la base de datos para generar un CRUD de las vistas necesarias. La conexión a la base de datos es sumamente sencilla ya que CodeIgniter cuenta con las clases necesarias mismas que solo requieren ser modificadas con los identificadores de nuestra base de datos.

## Configuración para conectar a la base de datos

application/config/database.php

```
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'centro_medico',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

Figura 18. Configuración para conexión a base de datos. Elaborado por: Alejandro Llano.

### 2.7.1.2 Estructura.

CodeIgniter tiene una estructura Modelo-Vista-Controlador (MVC) cuya idea básica es separar nuestro sistema en tres capas lo cual permite dividir la lógica de negocio del diseño, dando como resultado un proyecto más escalable y seguro.

A continuación, se presenta la codificación de estas tres capas:

### 2.7.1.3 Modelo.

La clase modelo es la encargada de ejecutar las solicitudes del controlador mismas que obtienen información de la base de datos como podemos ver a continuación.

## Clase modelo

```
class Areas_model extends CI_Model {  
  
    var $table = 'areas';  
  
    public function __construct() {  
        parent::__construct();  
        $this->load->database();  
    }  
  
    public function get_all_areas() {  
        $this->db->from('areas');  
        $query = $this->db->get();  
        return $query->result();  
    }  
  
    public function get_by_id($id) {  
        $this->db->from($this->table);  
        $this->db->where('id_area', $id);  
        $query = $this->db->get();  
        return $query->row();  
    }  
  
    public function areas_add($data) {  
        $this->db->insert($this->table, $data);  
        return $this->db->insert_id();  
    }  
  
    public function areas_update($where, $data) {  
        $this->db->update($this->table, $data, $where);  
        return $this->db->affected_rows();  
    }  
  
    public function delete_by_id($id) {  
        $this->db->where('id_area', $id);  
        $this->db->delete($this->table);  
    }  
}
```

Figura 19. Codificación de la clase Modelo. Elaborado por Alejandro Llano

La codificación vista en la Figura 19 muestra los métodos típicos para visualizar, agregar, modificar y borrar registros de la base de datos, con la información obtenida de las vistas como vemos a continuación.

### 2.7.1.4 Vista.

Para mostrar información obtenida de la base de datos la vista únicamente debe imprimir en pantalla las variables generadas en la clase controlador logrando así separar la lógica de negocio como se ve a continuación.

## Clase Vista

```
<tbody>
  <?php foreach ($areas as $area) { ?>
    <tr>
      <td><?php echo $area->id_area; ?></td>
      <td><?php echo $area->nombre_area; ?></td>
      <td>
        <button class="btn editar"
          onclick="edit_areas(<?php echo $area->id_area; ?>)">
          <i class="glyphicon glyphicon-pencil"></i></button>
        <button class="btn cerrar"
          onclick="delete_areas(<?php echo $area->id_area; ?>)">
          <i class="glyphicon glyphicon-remove"></i></button>
      </td>
    </tr>
  <?php } ?>
</tbody>
```

Figura 20. Codificación de la clase vista. Elaborado por Alejandro Llano

La vista únicamente crea formularios los mismos que deberán ser relacionado con los métodos de la clase controlador que poseen la lógica de negocio.

La información se ingresa por medio de la vista, pero el método que permite guardar la información se encuentra en la clase controlador.

Formulario presentado al usuario.

```
<div class="modal-body form">
  <div id="the-message"></div>
  <?php echo form_open("areas/save_areas",
    array("id" => "form-areas", "class" => "form-horizontal")) ?>
  <input type="hidden" name="id_area" id="id_area" >
  <div class="form-group">
    <label for="nombre_area" class="col-md-3 col-sm-4 control-label">
      Nombre Área</label>
    <div class="col-md-9 col-sm-8">
      <input type="text" name="nombre_area"
        id="nombre_area" class="form-control">
    </div>
  </div>
</div>
<div class="modal-footer">
  <button type="submit" class="btn guardar">Guardar</button>
  <button type="reset" class="btn cerrar" data-dismiss="modal">
    Cerrar</button>
</div>
</form>
```

Figura 21. Codificación de formulario de ingreso de datos. Elaborado por Alejandro Llano

Una vez ingresada la información se la deberá enviar a ser procesada por los métodos controlador y modelo y para poder enviar la información sin recargos de página es necesario utilizar JavaScript el cual debe estar relacionado a la vista y se encarga de obtener los datos ingresados en los campos del formulario y enviarlos a través de Ajax.

## Método JavaScript para visualizar datos.

```
function ver_areas(id)
{
    save_method = 'update';
    $('#form-areas')[0].reset();
    $.ajax({
        url: "<?php echo site_url('index.php/areas/ajax_edit/') ?>/" + id,
        type: "GET",
        dataType: "JSON",
        success: function (data)
        {
            $('[name="id_area"]').val(data.id_area);
            $('[name="nombre_area"]').val(data.nombre_area);
            $('#modal_form').modal('show');
            $('.modal-title').text('Editar Área');
        },
        error: function (jqXHR, textStatus, errorThrown)
        {
            alert('Error al obtener información');
        }
    });
}
```

Figura 22. Método JavaScript para visualizar información obtenida de la base de datos. Elaborado por: Alejandro Llano

La utilización de JavaScript también es necesaria para poder enviar desde la vista un identificador que permita al controlador eliminar el registro deseado de la base de datos.

## Método JavaScript para eliminar registros

```
function eliminar_areas(id)
{
    if (confirm('¿Está seguro de que desea eliminar este registro?'))
    {
        $.ajax({
            url: "<?php echo site_url('index.php/areas/areas_delete') ?>/" + id,
            type: "POST",
            dataType: "JSON",
            success: function (data)
            {
                location.reload();
            },
            error: function (jqXHR, textStatus, errorThrown)
            {
                alert('Error al eliminar registro');
            }
        });
    }
}
```

Figura 23. Método JavaScript para eliminar registros de la base de datos. Elaborado por: Alejandro Llano

### ***2.7.1.5 Controlador.***

Es la clase que contiene la lógica de negocio. Se encarga de gestionar las peticiones de los usuarios. Es el encargado de responder la información solicitada dependiendo también del modelo y de la vista.

## Clase Controlador

```
class Areas extends CI_Controller {

    public function __construct() {
        parent::__construct();

        $this->load->helper('url');
        $this->load->model('areas_model');
    }

    public function areas_view() {
        $data = array();
        $data['nombre_usuario'] = $this->session->userdata('nombre_usuario');
        $data['areas'] = $this->areas_model->get_all_areas();
        $this->load->view("header");
        $this->load->view('areas/areas_view', $data);
        $this->load->view("footer");
    }

    public function areas_add() {
        $data = array(
            'id_area' => $this->input->post('id_area'),
            'nombre_area' => $this->input->post('nombre_area'),
        );
        $insert = $this->areas_model->areas_add($data);
        echo json_encode(array("status" => TRUE));
    }

    public function ajax_edit($id) {
        $data = $this->areas_model->get_by_id($id);
        echo json_encode($data);
    }

    public function areas_update() {
        $data = array('success' => false, 'messages' => array());

        $this->load->library('form_validation');
        $this->form_validation->set_rules("nombre_area", "Nombre Área",
            "trim|required");
        $this->form_validation->set_error_delimiters(
            '<p class="text-danger">', '</p>');
        if ($this->form_validation->run()) {
            $data['success'] = true;
            $data1 = array(
                'id_area' => $this->input->post('id_area'),
                'nombre_area' => $this->input->post('nombre_area'),
            );
        }
    }
}
```

Figura 24. Codificación de la vista controlador. Elaborado por: Alejandro Llano

### 2.7.1.6 Otros métodos.

#### 2.7.1.6.1 Autenticación.

##### Vista

Desde aquí se envía las credenciales del usuario para ser verificadas por los métodos de autenticación de las clases controlador y modelo.

##### Clase Vista del método de autenticación

```
<form method="post" action="
  <?php echo base_url() ?>index.php/login/iniciar_sesion_post">
  <div class="input-group">
    <span class="input-group-addon"><i class="fa fa-user"></i></span>
    <input type="text" name="nombre" placeholder="Nombre de usuario"
      class="form-control" required />
  </div>
  <br/>
  <div class="input-group">
    <span class="input-group-addon"><i class="fa fa-lock"></i></span>
    <input type="password" name="contrasena" placeholder="Contraseña"
      class="form-control" required/>
  </div>
  <br/>
  <button type="submit" class="btn col-sm-12"
    >Acceder</button>
</form>
```

Figura 25. Codificación de la clase vista para el método de autenticación. Elaborado por: Alejandro Llano

##### Controlador

Este método obtiene los datos ingresados en el formulario de autenticación, los datos obtenidos son validados en la base de datos, al existir un usuario que coincida se permite el acceso. Si el usuario es de tipo administrador se re direcciona a una vista diferente, caso contrario se dará acceso a las vistas de los usuarios Doctor y enfermera. Si los datos ingresados no coinciden con la información de la base de

datos se niega el acceso y se solicita volver a ingresar la información, una vez iniciada una sesión se puede cerrarla destruyendo todos los datos ingresados en la autenticación

Clase controlador del método de autenticación.

```
public function iniciar_sesion_post() {
    if ($this->input->post()) {
        $nombre = $this->input->post('nombre');
        $contrasena = $this->input->post('contrasena');
        $this->load->model('login_model');
        $usuario = $this->login_model->usuario_por_nombre_contrasena(
            $nombre, md5($contrasena));
        if ($usuario) {

            if ($usuario->nombre_usuario == 'Administrador del Sistema') {
                $usuario_data = array();
                $usuario_data['nombre_usuario'] = $usuario->nombre_usuario;
                $usuario_data['logueado'] = FALSE;
                $usuario_data['admin'] = TRUE;

                $this->session->set_userdata($usuario_data);
                redirect('login/administrador');
            } else {
                $usuario_data = array();
                $usuario_data['nombre_usuario'] = $usuario->nombre_usuario;
                $usuario_data['logueado'] = TRUE;
                $usuario_data['admin'] = FALSE;

                $this->session->set_userdata($usuario_data);
                redirect('login/pagina_principal');
            }
        } else {
            redirect('login/iniciar_sesion');
        }
    } else {
        $this->iniciar_sesion();
    }
}

public function cerrar_sesion() {
    $this->session->sess_destroy();
    $usuario_data = array(
        'logueado' => FALSE,
        'admin' => FALSE
    );
    $this->session->set_userdata($usuario_data);
    redirect('login/iniciar_sesion');
}
```

Figura 26. Codificación del controlador para el método de autenticación. Elaborado por: Alejandro Llano

### 2.7.1.6.2 Subir imágenes.

#### Controlador

En este método cargamos la librería upload propia de CodeIgniter que permitirá subir imágenes, en este método definimos la ruta donde se guardarán las imágenes y los tipos de archivos permitidos.

Clase controlador del método subir imágenes

```
public function upload() {
    if (!empty($_FILES)) {
        $config['upload_path'] = "./public/img/evidencias_fotograficas/";
        $config['allowed_types'] = 'gif|jpg|png|mp4|ogv|zip';

        $this->load->library('upload', $config);
        if (!$this->upload->do_upload("file")) {
            echo "Archivo no pudo ser subido";
        }
    } elseif ($this->input->post('file_to_remove')) {
        $file_to_remove = $this->input->post('file_to_remove');
        unlink("./public/img/evidencias_fotograficas/" . $file_to_remove);
    } else {
        $this->listFiles();
    }
}

private function listFiles() {
    $this->load->helper('file');
    $files = get_filenames("./public/img/evidencias_fotograficas/");
    echo json_encode($files);
}
```

Figura 27. Codificación del controlador para subir imágenes. Elaborado por: Alejandro Llano

## Vista

Se crea un formulario que debe hacer uso del método antes creado en el controlador para subir imágenes, se declara un elemento tipo file que obtendrá la ruta de la imagen que vamos a subir.

### Clase vista para subir imágenes

```
<div id="d_file">
  <form action="<?php echo site_url("registro_bpm/upload") ?>"
    id="form-upload">
    <div class="fileinput fileinput-new input-group"
      data-provides="fileinput">
      <div class="form-control" data-trigger="fileinput">
        <i class="glyphicon glyphicon-file fileinput-exists"></i>
        <span class="fileinput-filename"></span></div>
      <span class="input-group-addon btn btn-default btn-file">
        <span class="fileinput-new">
          <i class="glyphicon glyphicon-paperclip"></i>
          Select file</span><span class="fileinput-exists">
            <i class="glyphicon glyphicon-repeat"></i></span>
          <input type="file" name="file" id="file"></span>
        <a href="#" class="input-group-addon
          btn btn-default fileinput-exists" data-dismiss="fileinput">
          <i class="glyphicon glyphicon-remove"></i></a>
        <a href="#" id="upload-btn"
          class="input-group-addon btn btn-success fileinput-exists">
          <i class="glyphicon glyphicon-open"></i></a>
      </div>
    </form>
  </div>
```

Figura 28. Codificación de la clase vista para subir imágenes. Elaborado por: Alejandro Llano

### 2.7.1.6.3 Enviar email.

Se realiza las configuraciones según el host que utilizemos en este caso Gmail, se establecen los comentarios predefinidos que se mostraran en los mensajes generados, como remitente, destinatario, titulo o texto.

#### Método para el envío de email

```
$configGmail = array(
    'protocol' => 'smtp',
    'smtp_host' => 'ssl://smtp.googlemail.com',
    'smtp_port' => 465,
    'smtp_user' => '*****@gmail.com',
    'smtp_pass' => '*****',
    'mailtype' => 'html',
    'charset' => 'utf-8',
    'newline' => "\r\n"
);

$to_email = $row->email;
$texto = 'Estimado(a) Señor(a): <b>' . $row->nombre_paciente . '</PRE>';
$subject = 'Cita Médica Agendada UPS';

$this->email->initialize($configGmail);
$this->email->from('*****@gmail.com', 'CENTRO MÉDICO UPS CAMPUS SUR');
$this->email->to($to_email);
$this->email->subject($subject);
$this->email->message($texto);
for ($i = 1; $i <= 1; $i++) {
    $this->email->send();
}
```

Figura 29. Codificación del método controlador para subir imágenes. Elaborado por: Alejandro Llano

## 2.8 Pruebas de aceptación

Tabla 15. Prueba de aceptación para la gestión de usuarios.

Prueba de Aceptación	
<b>Código</b>	A - 1
<b>Requerimiento evaluado</b>	Gestión de Usuarios
<b>Descripción</b>	El usuario administrador podrá registrar usuarios que podrán interactuar con el sistema.
<b>Resultado esperado</b>	Los usuarios creados podrán acceder al sistema con las credenciales registradas, y hacer uso del mismo según los accesos que sus roles les permitan.
<b>Resultado obtenido</b>	Se obtiene el resultado esperado
<b>Evaluación</b>	Aprobado por: 

Nota: Esta tabla contiene la prueba de aceptación 1.

Tabla 16. Prueba de aceptación Registro de Pacientes

Prueba de Aceptación	
<b>Código</b>	A - 2
<b>Requerimiento evaluado</b>	Registro de Pacientes
<b>Descripción</b>	El sistema requiere el registro de todos los pacientes previo a ser atendidos.
<b>Resultado esperado</b>	Una vez registrado el paciente los usuarios del sistema podrán agendar una cita para el mismo, registrar información relacionada al paciente y esta información se evidenciará en los informes e indicadores.
<b>Resultado obtenido</b>	Se obtiene el resultado esperado
<b>Evaluación</b>	Aprobado por: 

Nota: Esta tabla contiene la prueba de aceptación 2.

Tabla 17. Prueba de aceptación agendamiento de citas

<b>Prueba de Aceptación</b>	
Código	A - 3
Requerimiento evaluado	Agendamiento de Citas
Descripción	El sistema debe lograr una organización de la agenda del doctor, para lo cual se podrá agendar citas mediante un calendario.
Resultado esperado	Los usuarios encargados de agendar citas, podrán visualizar una agenda donde encontrarán todas las citas agendadas y de esta manera tendrán una mejor organización
Resultado obtenido	Se obtiene el resultado esperado
Evaluación	Aprobado por: 

Nota: Esta tabla contiene la prueba de aceptación 3.

Tabla 18. Prueba de aceptación de registro de atención medica

<b>Prueba de Aceptación</b>	
Código	A - 4
Requerimiento evaluado	Registro de atención médica
Descripción	Previo al agendamiento de una cita el Doctor podrá atender las citas según el cronograma para lo cual se le mostrará una vista que contendrá la historia clínica del paciente.
Resultado esperado	El usuario podrá realizar la atención, tener acceso a los historiales del paciente, emitir certificados y directrices médicas, en el caso del usuario enfermera podrá tomar signos vitales, registrar exámenes, modificar datos de la cita.
Resultado obtenido	Se obtiene el resultado esperado
Evaluación	Aprobado por: 

Nota: Esta tabla contiene la prueba de aceptación 4.

Tabla 19. Prueba de aceptación del parte diario

<b>Prueba de Aceptación</b>	
Código	A - 5
Requerimiento evaluado	Parte Diario
Descripción	Los datos de este informe generaran en base a las atenciones médicas que el Doctor haya realizado en cada día.
Resultado esperado	Se mostrará un listado con los mismos datos del formato físico que se utiliza en el centro medico
Resultado obtenido	Se obtiene el resultado esperado
Evaluación	Aprobado por: 

Nota: Esta tabla contiene la prueba de aceptación 5.

Tabla 20. Prueba de aceptación de informe de inspecciones

<b>Prueba de Aceptación</b>	
Código	A - 6
Requerimiento evaluado	Informe de inspecciones preventivas
Descripción	Se podrá registrar y generar un informe de las evaluaciones BPMS al momento de inspeccionar el bar de la universidad
Resultado esperado	El usuario podrá registrar en tiempo real las inspecciones, y el sistema calculara automáticamente la calificación. Además una vez registrada la inspección se podrá realizar el respectivo informe de acuerdo al formato solicitado.
Resultado obtenido	Se obtiene el resultado esperado
Evaluación	Aprobado por: 

Nota: Esta tabla contiene la prueba de aceptación 6.

Tabla 21. Prueba de aceptación de Indicadores de morbilidad

<b>Prueba de Aceptación</b>	
Código	A - 7
Requerimiento evaluado	Indicadores de morbilidad
Descripción	El sistema mostrará informes estadísticos acerca de la morbilidad dividido en tres categorías: General, Áreas, CIE 10.
Resultado esperado	El usuario visualizara informes estadísticos con datos obtenidos en las iteraciones de atención médica, están basados en los formatos utilizados en el centro médico.
Resultado obtenido	Se obtiene el resultado esperado
Evaluación	Aprobado por: 

Nota: Esta tabla contiene la prueba de aceptación 7.

Tabla 22. Prueba de aceptación de informes epidemiológicos

<b>Prueba de Aceptación</b>	
Código	A - 8
Requerimiento evaluado	Informes Epidemiológicos
Descripción	El sistema mostrará informes estadísticos epidemiológicos divididos en cuatro categorías: Género, Edad, Patologías más comunes, CIE 10.
Resultado esperado	El usuario visualizara informes estadísticos con datos obtenidos en las iteraciones de atención médica, están basados en los formatos utilizados en el centro médico.
Resultado obtenido	Se obtiene el resultado esperado
Evaluación	Aprobado por: 

Nota: Esta tabla contiene la prueba de aceptación 8.

Tabla 23. Prueba de aceptación de plan de vigilancia médica

<b>Prueba de Aceptación</b>	
Código	A - 9
Requerimiento evaluado	Plan Vigilancia Médica
Descripción	El sistema pretende lograr una organización de la agenda de planificaciones para vigilancia mediante un calendario.
Resultado esperado	Los usuarios visualizarán la planificación para cada una de las sedes accediendo al sistema
Resultado obtenido	Se obtiene el resultado esperado
Evaluación	Aprobado por: 

Nota: Esta tabla contiene la prueba de aceptación 9.

## 2.9 Capacitaciones

Como parte del proceso de entrega del proyecto es necesaria una capacitación la cual será efectuada según el cronograma, también se realizarán retroalimentaciones en caso de ser necesario.

Ver anexo 4 – Capacitación

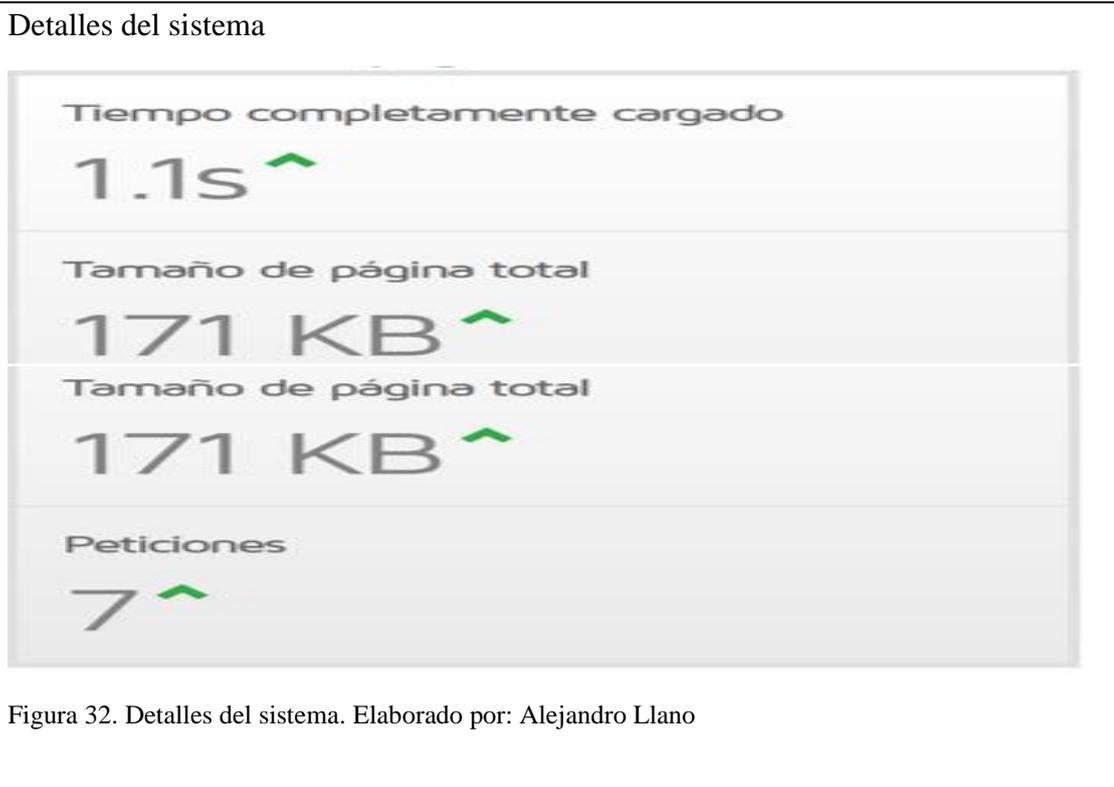
# Capítulo 3

## 3.1 Análisis de resultados con GTmetrix

Se analizará el sistema de control de historias y citas médicas mediante pruebas de carga las mismas que nos permiten analizar el comportamiento del sistema en el momento que el usuario interactúa con el mismo pudiendo ser de manera recurrente es decir cuando varios usuarios interactúan con el sistema al mismo tiempo. Estos escenarios nos mostrarán los tiempos de respuesta de todas las transacciones que se realicen. Para realizar esta prueba se utilizará GTmetrix que es una herramienta online que permite obtener un resumen claro del rendimiento de una página según los indicadores obtenidos.

A continuación, los resultados obtenidos por GTmetrix:





Los resultados nos dejan una sensación positiva ya que el sistema muestra una notable eficiencia en los tiempos de respuesta y tomando en cuenta que su uso no está orientado a una gran carga de usuarios ya que únicamente será usado en el centro médico y la recurrencia de usuarios muy difícilmente podrá sobrepasar los límites que el sistema puede soportar.

**3.2 Pruebas de satisfacción de usuario**

Se han registrado ocho encuestas correspondientes a los usuarios del sistema que ha sido implementado en el centro médico de la universidad politécnica salesiana campus sur, los cuales son Doctor (a) y Enfermero (a). En la encuesta constan preguntas que también tienen en cuenta a la satisfacción del paciente por lo tanto el usuario que registró la encuesta debió valorar la satisfacción de los pacientes en diferentes atenciones ya que las encuestas fueron registradas en varias ocasiones.

La herramienta utilizada para registrar y analizar estas encuestas fue el formulario de Google ya que es una aplicación muy completa y para nuestro objetivo es totalmente adecuada.

### 3.2.1 Formato Encuesta.

Encuesta de Satisfacción de Usuario del sistema web implementado en el centro médico de la universidad politécnica salesiana campus sur.

Encuesta

---

¿Considera usted que el sistema satisface eficientemente todos los requerimientos solicitados? \*

Sí       No

¿En una escala del 1 a 5, considera que la aplicación de este sistema en los procedimientos del centro medico ha sido favorable y que ha aumentado la satisfacción del personal medico y de los pacientes? \*

	1	2	3	4	5
1 nada favorable - 5 totalmente favorable	<input type="radio"/>				

¿Esta satisfecho con la manera en la que el sistema organiza la agenda de atención? \*

	1	2	3	4	5
1 nada favorable - 5 totalmente favorable	<input type="radio"/>				

¿En la escala del 1 al 5 que tan intuitivo y fácil de usar es el sistema? \*

	1	2	3	4	5
1 nada favorable - 5 totalmente favorable	<input type="radio"/>				

¿Ha tenido que solicitar asistencia para utilizar el sistema? \*

Sí       No

¿El sistema ha presentado algún error durante su utilización?

Sí       No

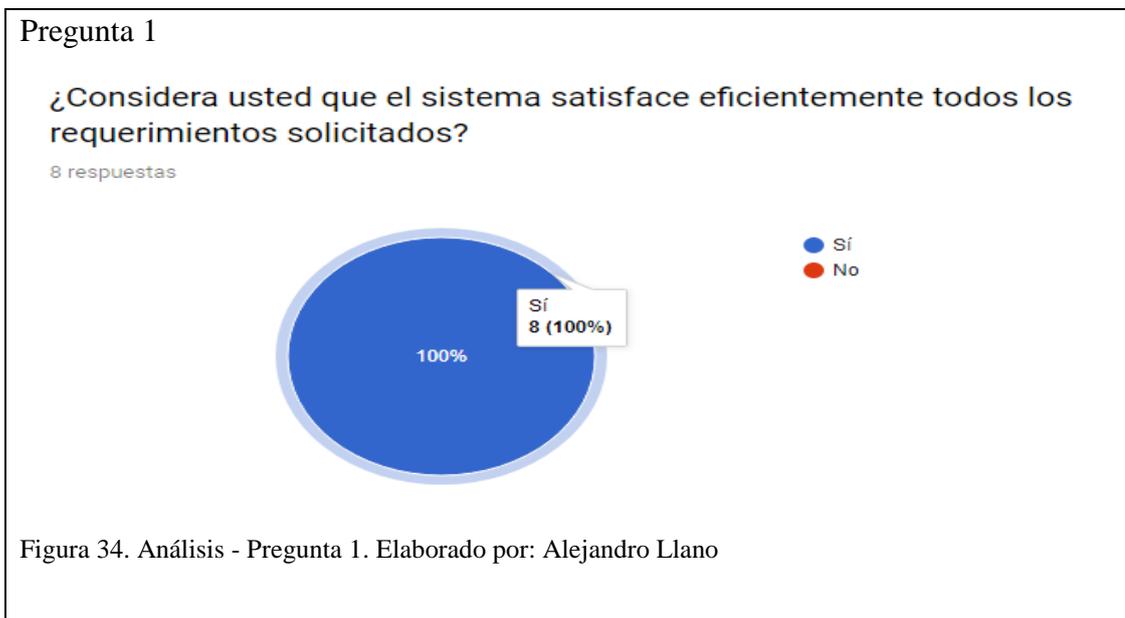
Figura 33. Encuesta de satisfacción. Elaborado por: Alejandro Llano

### Anexo 3 - Encuestas

### 3.2.2 Análisis de respuestas obtenidas en las encuestas.

#### Pregunta 1

Las 8 respuestas obtenidas en distintos momentos y situaciones según el uso de los usuarios nos muestran que hay una total satisfacción del usuario con los procesos que realiza el sistema.



#### Pregunta 2

De las 8 respuestas obtenidas 5 consideran que la implantación del sistema ha sido totalmente favorable corroborando la pregunta anterior, sin embargo el 37.5% considera que el sistema no ha sido totalmente eficiente y se puede suponer que el motivo esta respuesta es debido a que el sistema se encuentra alojado en un servidor web no tan potente y además porque el servicio de internet que se utiliza en el departamento médico no siempre es estable lo que puede ocasionar lentitud en los procesos que realizan.

### Pregunta 2

¿En una escala del 1 a 5, considera que la aplicación de este sistema en los procedimientos del centro medico ha sido favorable y que ha aumentado la satisfacción del personal medico y de los pacientes?

8 respuestas

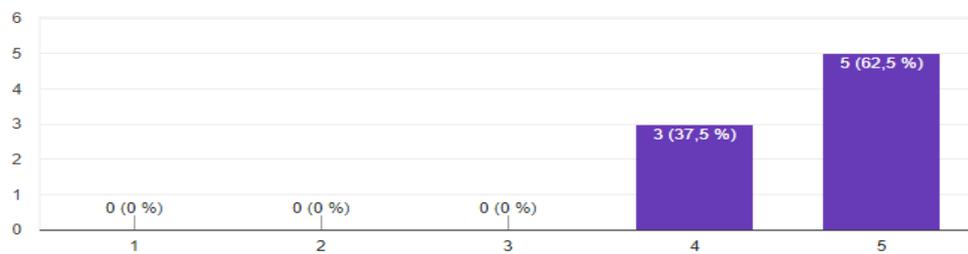


Figura 35. Análisis - Pregunta 2. Elaborado por: Alejandro Llano

### Pregunta 3

El 87.5% de los encuestados considera que el sistema organiza de manera óptima la agenda de atención el 12.5% restante considera que la organización es normalmente buena lo cual después de haber tratado este tema se llegó a considerar que su respuesta se basa en que les resulta algo difícil realizar este procedimiento mas no en que el resultado no sea el esperado.

### Pregunta 3

¿Esta satisfecho con la manera en la que el sistema organiza la agenda de atención?

8 respuestas

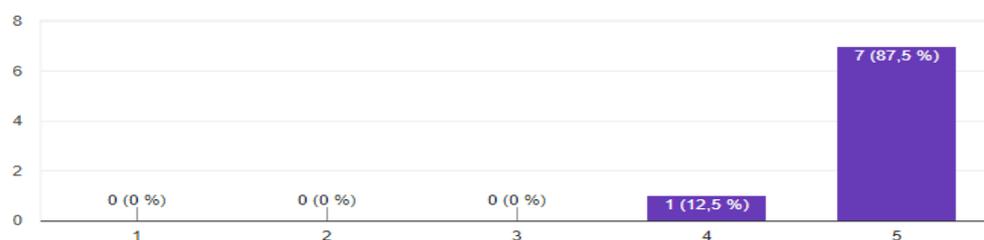
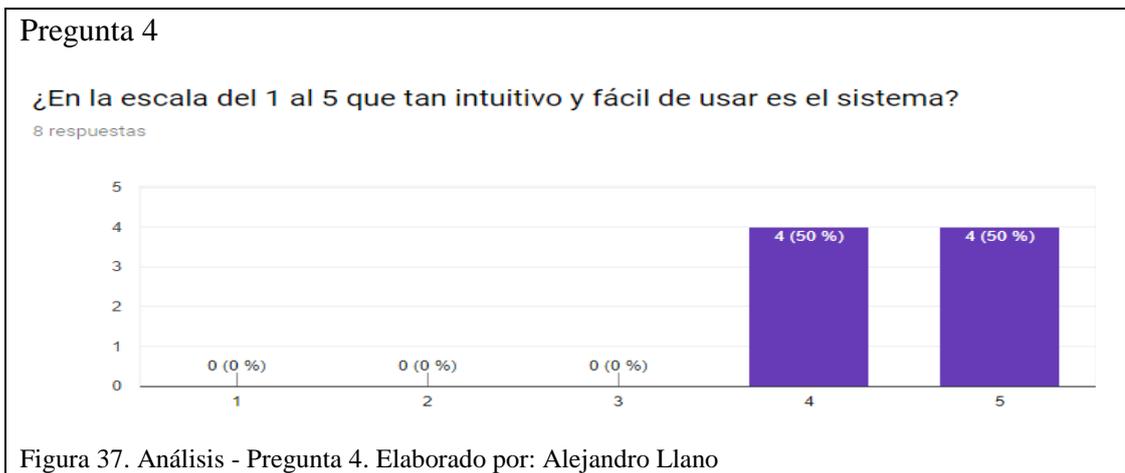


Figura 36. Análisis - Pregunta 3. Elaborado por: Alejandro Llano

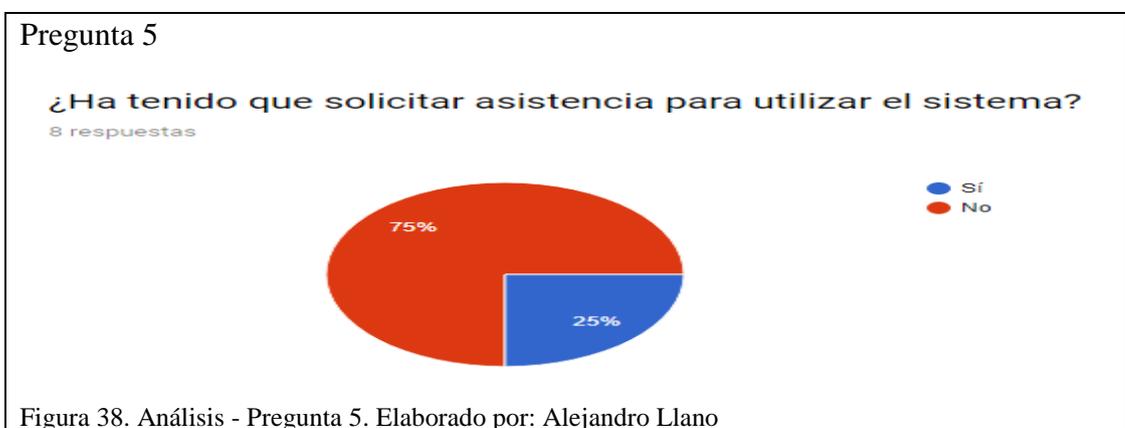
#### Pregunta 4

Como se aprecia en la pregunta 3 el sistema muestra un alto nivel intuitivo y al ser favorables las respuestas se considera que a medida que el usuario se vaya familiarizando con el sistema se considerará totalmente intuitivo.



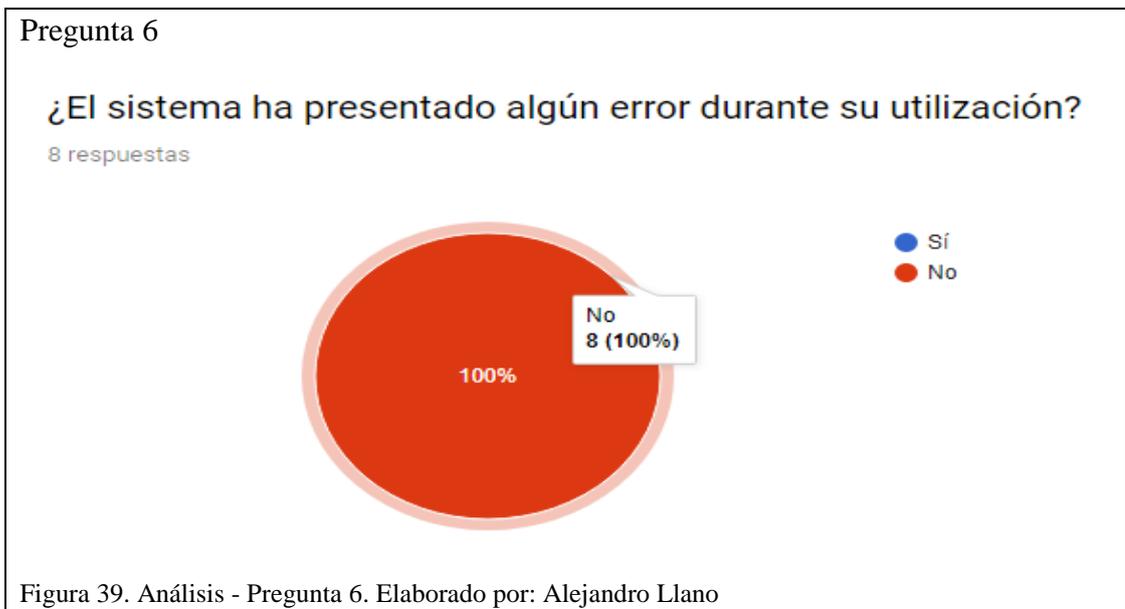
#### Pregunta 5

El 25% de las ecuaciones en las que se ha usado el sistema han sido necesarias asistencias las cuales han sido netamente por desconocimiento del funcionamiento debido a una omisión de pasos en el manual de uso o de olvido por parte de los usuarios.



## Pregunta 6

Pese a alguna lentitud de carga debido al servidor web en el cual se encuentra alojado el sistema o a alguna intermitencia en el servicio de internet el sistema no ha mostrado algún tipo de error en su funcionamiento.



## CONCLUSIONES

- Se consiguió desarrollar un software que haciendo uso de una metodología de desarrollo como es Scrum ha permitido tener como resultado un producto de calidad que a pesar de ser un sistema robusto no exige requerimientos especiales para su eficiente funcionamiento.
- Se evidenció satisfacción por parte del usuario no solo por la funcionalidad sino también en la forma en la cual el proyecto fue desarrollado ya que en un principio el cliente no tenía definidos los requerimientos y en base a esta metodología pudo tener una visión más clara de lo que realmente se necesitaba.
- Las principales problemáticas que presentaba el centro médico al inicio de este proyecto, como la descoordinación en horarios de atención que provocaba que muchas de las veces los pacientes no puedan ser atendidos o la imprecisión en los reportes estadísticos que se realizan periódicamente fueron solucionados eficientemente, en primer lugar mediante la creación de una agenda virtual que logró organizar el tiempo de los pacientes y el médico. Así también el procesamiento de todos los datos obtenidos en cada iteración logró automatizar la creación de reportes estadísticos y de la misma manera todos los módulos detallados en este proyecto. Dando como resultado final un sistema que cuenta con el total agrado de los usuarios que han evidenciado los resultados de la implementación de este sistema.

## RECOMENDACIONES

- Para la implementación del sistema es recomendable utilizar Apache 2 v2.4 y PHP v7.0 ya que estas fueron las versiones de estas herramientas al momento de desarrollar y probar el sistema con lo cual se garantiza su eficiente funcionamiento.
- El navegador web utilizado durante el desarrollo y pruebas fue Google Chrome mismo que ha mostrado total compatibilidad con el sistema desarrollado, sin embargo también se han realizado pruebas con el navegador Mozilla Firefox mostrando resultados muy positivos, así que se recomienda este navegador como alternativa.
- Pese a que el sistema fue desarrollado con un diseño responsive (adaptativo), no se garantiza su total eficiencia al ser utilizado en dispositivos móviles, por lo que es recomendable usarlo en navegadores web.

## LISTA DE REFERENCIA

- 1and1. (31 de 05 de 2016). www.1and1.es. Obtenido de <https://www.1and1.es/digitalguide/paginas-web/desarrollo-web/caracteristicas-y-ventajas-de-php7/>
- 1and1. (07 de 09 de 2017). www.1and1.es. Obtenido de <https://www.1and1.es/digitalguide/paginas-web/desarrollo-web/codeigniter-framework-php-rapido-y-versatil/>
- Alaimo, D. M. (2013). PROYECTOS ÁGILES CON SCRUM. Ciudad Autónoma de Buenos Aires : Kleer.
- desarrolloweb. (23 de 11 de 2009). desarrolloweb.com. Obtenido de <https://desarrolloweb.com/articulos/codeigniter.html>
- DonWeb. (2014). Obtenido de <http://agencia.donweb.com/las-nuevas-tecnologias-en-el-sistema-de-atencion-medica/>
- iebschool. (10 de 09 de 2015). www.iebschool.com. Obtenido de <http://www.iebschool.com/blog/definicion-y-caracteristicas-del-scrum-master-agile-scrum/>
- Rodrigues, R. J. (06 de 1999). apps.who.int. Obtenido de <http://apps.who.int/medicinedocs/documents/s16596s/s16596s.pdf>
- ultimus. (2017). www.ultimus.com. Obtenido de <http://www.ultimus.com/es/bpms>
- uv. (2017). www.uv.es. Obtenido de <https://www.uv.es/jac/guia/jscript/javascri01.htm>
- vozidea. (26 de 04 de 2013). www.vozidea.com. Obtenido de <http://www.vozidea.com/phpmyadmin-administrador-bases-de-datos>