



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

" E SCIENTIA HOMINIS SALUS "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN SISTEMA PROTOTIPO DE GESTIÓN DE TURNOS MEDIANTE TELÉFONOS MÓVILES CON SISTEMA ANDROID PARA EL DEPARTAMENTO MÉDICO DE LA ESCUELA POLITÉCNICA NACIONAL

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
“INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN”**

ALEX ALEJANDRO MONTENEGRO CHANGOTASI

DIRECTOR: Ing. XAVIER ALEXANDER CALDERÓN HINOJOSA M.Sc.

Quito, febrero 2020

AVAL

Certifico que el presente trabajo fue desarrollado por Alex Alejandro Montenegro Changotasi, bajo mi supervisión.

M.Sc. Xavier Calderón
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Alex Alejandro Montenegro Changotasi, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Alex Alejandro Montenegro Changotasi

DEDICATORIA

A mi madre, quien inculcó en mí los mejores valores. Y quien desde el cielo guía mis pasos para ser una mejor persona.

AGRADECIMIENTO

A mi padre y hermanas, quienes han estado junto a mi en distintas etapas de la vida.

A mis amigos y compañeros con quienes he compartido diversas experiencias para vencer las dificultades presentadas en nuestra etapa estudiantil.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	X
ÍNDICE DE CÓDIGOS	XI
RESUMEN	XIII
ABSTRACT	XIV
1. INTRODUCCIÓN.....	1
1.1 OBJETIVOS	1
1.2 ALCANCE.....	2
1.3 MARCO TEÓRICO.....	4
1.3.1 SMARTPHONES.....	4
1.3.2 APLICACIONES MÓVILES.....	6
1.3.2.1 Aplicaciones Nativas.....	7
1.3.2.2 Aplicaciones Web	8
1.3.2.3 Aplicaciones Híbridas	8
1.3.3 METOLOGÍA ÁGIL DE DESARROLLO KANBAN.....	9
1.3.3.1 Principios básicos de Kanban	10
1.3.3.2 Pasos para configurar la estrategia Kanban.....	10
1.3.4 DESCRIPCIÓN DE HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS. 12	
1.3.4.1 Android.....	12
1.3.4.2 Plataforma de desarrollo de Aplicación Móvil.....	13
1.3.4.3 Base de Datos MYSQL y PhpMyAdmin	14
1.3.4.4 PHP.....	15
1.3.4.5 Amazon Web Services.....	16
1.3.4.6 Servicios Web.....	17
1.3.4.7 JSON.....	18
1.3.4.8 Servicio FTP y Fillezilla.....	19

2. METODOLOGÍA	21
2.1 DISEÑO.....	21
2.1.1 TABLERO DE ACTIVIDADES KANBAN	21
2.1.2 ANÁLISIS DE REQUERIMIENTOS.....	22
2.1.2.1 Requerimientos Funcionales (RF)	23
2.1.2.1.1 Historias de usuario	23
2.1.2.2 Requerimientos No Funcionales (RNF)	26
2.1.3 MÓDULOS DEL PROTOTIPO.....	27
2.1.4 DISEÑO DE LA CAPA DE DATOS.....	30
2.1.4.1 Diagrama Relacional para la Base de Datos.....	31
2.1.5 DISEÑO DE LA CAPA DE NEGOCIO.....	32
2.1.5.1 Diagrama de Casos de Uso (DCU).....	32
2.1.5.2 Diseño del Diagrama de Clases.....	41
2.1.5.3 Diseño de Servicio Web.....	43
2.1.6 DISEÑO DE LA CAPA DE PRESENTACIÓN	44
2.1.6.1 Diseño de las Vistas del sistema prototipo Usuario Paciente	44
2.1.6.2 Diseño de las Vistas del sistema prototipo Usuario Doctor.....	47
2.1.6.3 Diseño de las Vistas del sistema prototipo Usuario Administrador	48
2.2 IMPLEMENTACIÓN	49
2.2.1 ACTUALIZACIÓN DEL TABLERO DE ACTIVIDADES KANBAN	49
2.2.2 ESQUEMA DEL SERVIDOR.....	50
2.2.3 INSTALACIÓN DE HERRAMIENTAS NECESARIAS	51
2.2.3.1 Creación de una instancia de Amazon EC2 en AWS	51
2.2.3.2 Instalación y codificación de MySQL	55
2.2.3.3 Instalación y codificación del servidor Web Apache	56
2.2.3.4 Instalación y codificación del servidor FTP	56
2.2.3.5 Android Studio	57
2.2.4 IMPLEMENTACIÓN DE LA CAPA DE BASE DE DATOS	59
2.2.5 IMPLEMENTACIÓN DE LA CAPA DE NEGOCIO	61
2.2.5.1 Codificación de clases.....	61
2.2.5.2 Codificación de Servicios Web.....	62
2.2.6 CODIFICACIÓN DE LA CAPA DE PRESENTACIÓN	64
2.2.6.1 Codificación de vista del aplicativo móvil.....	65
3. RESULTADOS Y DISCUSIÓN	72

3.1	TABLERO DE ACTIVIDADES KANBAN	72
3.2	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES. ...	74
3.2.1	PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO DE AUTENTICACIÓN.	74
3.2.2	PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO DE VERIFICACIÓN DE TURNOS.	76
3.2.3	PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO GESTIÓN DE TURNOS.....	77
3.2.4	PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO CONSULTA DE TURNOS.....	81
3.2.5	PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO DE NOTIFICACIONES.	83
3.2.6	PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO ADMINISTRACIÓN DE USUARIO.	84
3.3	PRUEBAS DE VALIDACIÓN DE LOS REQUERIMIENTOS NO FUNCIONALES	88
3.3.1	PRESENTACIÓN.....	94
3.3.2	DISPONIBILIDAD.....	94
3.3.3	TIEMPO DE RESPUESTA.....	94
3.3.4	ESCALABILIDAD	95
3.3.5	AUTENTICACIÓN.....	95
3.4	ACTUALIZACION FINAL DEL TABLERO KANBAN.	97
4.	CONCLUSIONES Y RECOMENDACIONES.....	99
4.1	CONCLUSIONES.....	99
4.2	RECOMENDACIONES	100
5.	REFERENCIAS BIBLIOGRÁFICAS	102
6.	ANEXOS.....	108
	ANEXO A. HISTORIAS DE USUARIO.....	108
	ANEXO B. PERMISO DE USO DE LA BASE DE DATOS	108
	ANEXO C. CÓDIGO CLASES.....	108
	ANEXO D. CÓDIGO APLICACIÓN	108
	ANEXO E. CÓDIGO DE INTERFAZ FRÁFICA	108
	ANEXO F. ENCUESTAS A USUARIOS DE SISTEMA PROTOTIPO.....	108
	Las encuestas se encuentran en el CD adjunto a este documento.....	108
	ORDEN DE EMPASTADO	109

ÍNDICE DE FIGURAS

Figura 1.1 Arquitectura del sistema	3
Figura 1.2 Distribución del mercado móvil a nivel global por fabricante.....	5
Figura 1.3 Distribución del mercado móvil por SO a nivel global.....	6
Figura 1.4 Etapas de una aplicación móvil	7
Figura 1.5 Ejemplo de un tablero Kanban	12
Figura 1.6 Arquitectura de un servicio web, usando formato JSON	19
Figura 1.7 Comportamiento del sistema Cliente – Servidor	20
Figura 2.1 Tablero Kanban para fase de Diseño	21
Figura 2.2 Diagrama Entidad-Relación de base de datos	29
Figura 2.3 Actores que interactúan con el sistema prototipo.....	30
Figura 2.4 Caso de uso escoger el tipo de usuario que va a utilizar el prototipo..	31
Figura 2.5 Caso de uso generar clave.....	31
Figura 2.6 Caso de uso autenticar como médico	31
Figura 2.7 Caso de uso autenticar como paciente	32
Figura 2.8 Caso de uso escoger especialidad para cita médica.....	32
Figura 2.9 Caso de uso visualizar información del médico.....	32
Figura 2.10 Caso de uso escoger fecha para cita médica.....	33
Figura 2.11 Caso de uso escoger horario para cita médica	33
Figura 2.12 Caso de uso comprobar disponibilidad de horario	34
Figura 2.13 Caso de uso agendar cita médica	34
Figura 2.14 Caso de uso revisar citas médicas a tratar.....	34
Figura 2.15 Caso de uso visualizar citas médicas vigentes	35
Figura 2.16 Caso de uso visualizar resumen de cita médica agendada.....	35
Figura 2.17 Caso de uso anular cita médica	35
Figura 2.18 Caso de uso recibir notificación de cita médica vigente	36
Figura 2.19 Caso de uso autenticar como administrador	36
Figura 2.20 Caso de uso administrar usuarios	37
Figura 2.21 Caso de uso buscar información de un usuario	37
Figura 2.22 Caso de uso actualizar información de un usuario	37
Figura 2.23 Caso de uso eliminar usuario	38
Figura 2.24 Diagrama de clases del sistema prototipo	39
Figura 2.25 Arquitectura del servicio web.....	40
Figura 2.26 Interfaz de inicio	41
Figura 2.27 Transición de vista autenticación a vista inicio para usuario paciente	42
Figura 2.28 Interfaz lista de áreas médicas.....	43
Figura 2.29 Transición de vista horarios a vista agendar cita.....	43
Figura 2.30 Interfaz vista y anulación de cita agendada.....	44
Figura 2.31 Transición de interfaces para usuario Doctor	45
Figura 2.32 Interfaz gráfica Usuario Administrador	46
Figura 2.33 Tablero Kanban para fase de Implementación	47
Figura 2.34 Arquitectura del servidor.....	48

Figura 2.35	Crear cuenta AWS.....	48
Figura 2.36	Selección de Amazon EC2.....	50
Figura 2.37	Creación de instancia AWS.....	50
Figura 2.38	Elección de Servido Ubuntu 16.04 LTS.....	51
Figura 2.39	Características de instancia EC2 de nivel libre.....	51
Figura 2.40	Opciones de seguridad para la instancia Ubuntu 16.04 LTS.....	51
Figura 2.41	Instancia Ubuntu 16.04 LTS funcionando correctamente.....	51
Figura 2.42	Software Filezilla	53
Figura 2.43	Descarga de Android Studio.....	54
Figura 2.44	Instalación de Android Studio.....	54
Figura 2.45	Creación de nuevo proyecto.....	55
Figura 2.46	Resultado de la consulta a la tabla Docente	57
Figura 2.47	Interfaces gráficas del sistema prototipo	61
Figura 2.48	Interfaz de inicio	62
Figura 2.49	Interfaz Principal para Médico y Doctor.....	64
Figura 2.50	Interfaz Especialidades	64
Figura 2.51	Interfaz Agendar cita médica.....	65
Figura 2.52	Interfaz listar y anular citas médicas.....	65
Figura 2.53	Interfaz administrar usuarios	66
Figura 3.1	Tablero Kanban para la fase de Resultados y Discusión	68
Figura 3.2	Cumplimiento del Requerimiento Funcional RF001	69
Figura 3.3	Cumplimiento del Requerimiento Funcional RF002	70
Figura 3.4	Cumplimiento del Requerimiento Funcional RF003	70
Figura 3.5	Cumplimiento del Requerimiento Funcional RF005	71
Figura 3.6	Cumplimiento del Requerimiento Funcional RF004	72
Figura 3.7	Cumplimiento del Requerimiento Funcional RF006	73
Figura 3.8	Cumplimiento del Requerimiento Funcional RF007	74
Figura 3.9	Cumplimiento del Requerimiento Funcional RF008	74
Figura 3.10	Cumplimiento del Requerimiento Funcional RF009	75
Figura 3.11	Cumplimiento del Requerimiento Funcional RF010	75
Figura 3.12	Cumplimiento del Requerimiento Funcional RF011	76
Figura 3.13	Cumplimiento del Requerimiento Funcional RF012	77
Figura 3.14	Cumplimiento del Requerimiento Funcional RF013	77
Figura 3.15	Cumplimiento del Requerimiento Funcional RF014	78
Figura 3.16	Cumplimiento del Requerimiento Funcional RF015	79
Figura 3.17	Cumplimiento del Requerimiento Funcional RF016	80
Figura 3.18	Cumplimiento del Requerimiento Funcional RF017	80
Figura 3.19	Cumplimiento del Requerimiento Funcional RF018	81
Figura 3.20	Cumplimiento de Requerimientos Funcionales RF019 y RF020	81
Figura 3.21	Encuesta para Usuario Médico	84
Figura 3.22	Encuesta para Usuario Paciente	85
Figura 3.23	Interfaz de ingreso al sistema prototipo.....	90
Figura 3.24	Tablero final Kanban	92

ÍNDICE DE TABLAS

Tabla 1.1 Ventajas y desventajas de Aplicaciones móviles	9
Tabla 1.2 Características De S. O. Android	13
Tabla 1.3 Características De Android Studio	14
Tabla 1.4 Características De Amazon Web Services	17
Tabla 2.1 Formato de una Historia de Usuario	23
Tabla 2.2 Historia de Usuario 001.....	24
Tabla 2.3 Historia de Usuario 002.....	24
Tabla 2.4 Historia de Usuario 003.....	25
Tabla 2.5 Historia de Usuario 023.....	25
Tabla 2.6 Historia de Usuario 024.....	25
Tabla 2.7 Requerimientos Funcionales.....	26
Tabla 2.8 Requerimientos No Funcionales	27
Tabla 2.9 Estimación RF Módulo de Autenticación.....	28
Tabla 2.10 Estimación RF Módulo de Verificación de Turnos	29
Tabla 2.11 Estimación RF Módulo Gestión de Turnos.....	29
Tabla 2.12 Estimación RF Módulo Consulta de Turnos	30
Tabla 2.13 Estimación RF Módulo de Notificaciones.....	30
Tabla 2.14 Estimación RF Módulo Administración de Usuario	30
Tabla 2.15 Características de Amazon EC2 de nivel libre	52
Tabla 2.16 Comparación de recursos entre Windows Server y Ubuntu Server	53
Tabla 3.1 Requerimientos Funcionales del Módulo de Autenticación.....	74
Tabla 3.2 Requerimientos Funcionales del Módulo de Verificación de Turnos....	77
Tabla 3.3 Requerimientos Funcionales del Módulo de Gestión de Turnos	78
Tabla 3.4 Requerimientos Funcionales del Módulo Consulta de Turnos	81
Tabla 3.5 Requerimientos Funcionales del Módulo de Notificaciones	83
Tabla 3.6 Requerimientos Funcionales del Módulo Administración de Usuario ..	84
Tabla 3.7 Lista de cumplimiento de los Requerimientos funcionales y tiempo empleado para su ejecución	87
Tabla 3.8 Respuesta a la Pregunta 1 del Usuario Paciente.....	91
Tabla 3.9 Respuesta a la Pregunta 1 del Usuario Doctor	91
Tabla 3.10 Preguntas y respuestas, encuesta para usuario Paciente	92
Tabla 3.11 Preguntas y respuestas, encuesta para usuario Doctor.....	93
Tabla 3.12 Lista de RNF por cumplir	93
Tabla 3.13 Respuesta a la Pregunta 2.....	94
Tabla 3.14 Tiempo que tarda la aplicación en realizar una acción	95
Tabla 3.15 Cumplimiento de Requerimientos No Funcionales y tiempo empleado para su ejecución	94

ÍNDICE DE CÓDIGOS

Código 1.1 Ejemplo de mensaje JSON	18
Código 2.1 Creación de Tabla Docente	60
Código 2.2 Creación de usuario Docente.....	60
Código 2.3 Consulta de usuario Docente	61
Código 2.4 Código de la clase Docente	62
Código 2.5 Código de la clase Doctor	62
Código 2.6 Ejemplo de mensaje JSON	63
Código 2.7 Código de un Servicio Web.....	64
Código 2.8 Código XML de interfaz gráfica	67
Código 2.9 Código XML de interfaz paciente	69

ÍNDICE DE COMANDOS

Comando 2.1 Instalación de MySQL.....	56
Comando 2.2 Instalación de Apache.....	56
Comando 2.3 Reinicio del servidor Apache.....	56
Comando 2.4 Instalación de servidor FTP	56

RESUMEN

El Departamento Médico de la ESCUELA POLITÉCNICA NACIONAL (EPN) posee 5 ramas o especialidades nombradas a continuación: Medicina General, Odontología, Nutrición, Psicología y ginecología, las cuales están a disposición de los diferentes miembros de la institución, ya sea personal administrativo, docentes, o estudiantes.

En la actualidad, uno de los problemas más importantes que tiene el Departamento Médico de la EPN es que la asignación de turnos para las diferentes áreas médicas de la EPN se lo realiza de forma manual, es por ese motivo que los estudiantes, personal administrativo, o docentes (que en este caso van a ser los pacientes) asisten a los consultorios para revisiones médicas sin previo aviso ni cita médica.

Esto conlleva a que en ocasiones haya una aglomeración de pacientes haciendo que estos estén esperando un tiempo indefinido su turno de ingreso, lo cual conlleva al no aprovechamiento de su tiempo en otras actividades.

Debido a este inconveniente se plantea como solución el desarrollo de un sistema prototipo para teléfonos móviles con Sistema Operativo Android el cual permita gestionar los turnos de los pacientes para las diferentes áreas médicas (medicina general, odontología, nutrición, psicología y ginecología.) dependiendo de los horarios establecidos por el departamento médico, así como de la disponibilidad del personal.

En el Capítulo 1, se describe el objetivo general, objetivos específicos, así como el alcance que tendrá el presente proyecto de titulación. Además, se detallan los fundamentos teóricos de Android Studio, Base de Datos y Servicio Web.

En el Capítulo 2, se detalla la metodología utilizada para el desarrollo del presente Trabajo de Titulación, el mismo que se lo divide en dos partes: diseño e implementación.

En el Capítulo 3, se comprueba el correcto funcionamiento del sistema prototipo para teléfonos móviles con Sistema Operativo Android y se muestran los respectivos resultados de las pruebas realizadas.

En el Capítulo 4, se describe las conclusiones y recomendaciones.

PALABRAS CLAVE: Sistema Prototipo, Base de Datos, Sistema Operativo Android, Servicio Web.

ABSTRACT

The Medical Department of the NATIONAL SCHOOL POLYTECHNIC (EPN) has 5 branches or specialties such as: General Medicine, Dentistry, Nutrition, Psychology and Gynecology, which are available to different members of the institution, whether administrative staff, teachers, or students.

Currently, one of the most important problems that the Medical Department of the EPN has is that the allocation of shifts for the different medical areas of the EPN is done manually, that is why the students, administrative personnel, or teachers (who in this case will be the patients) who attend the offices for medical check-ups without prior notice or medical appointment.

This leads to sometimes an agglomeration of patients causing them to wait indefinitely for their shift, which leads to not using their time in other activities.

Due to this inconvenience, the development of a prototype system for mobile phones with Android Operating System is proposed as a solution, which allows to manage patient shifts for different medical areas (general medicine, dentistry, nutrition, psychology and gynecology.) Depending on the schedules established by the medical department, as well as the availability of the staff.

In Chapter 1, the general objective, specific objectives, as well as the scope of the present project of title is described. In addition, it details the theoretical foundations of Android Studio, Database and Web Service.

In Chapter 2, the methodology used for the development of this thesis project is described which is divided into two parts: design and implementation.

In Chapter 3, the correct operation of the prototype system for mobile phones with Android Operating System is verified and the respective results of the tests performed are shown.

In Chapter 4, the conclusions and recommendations are described.

KEY WORDS: Prototype System, Database, Android Operating System, Web Service.

1. INTRODUCCIÓN

Actualmente es completamente imprescindible que los distintos centros médicos privados y organismos públicos busquen y sobre todo fomenten el uso de herramientas que mejoren su eficiencia y que maximicen el trabajo de sus empleados, permitiendo a estos dar el mejor servicio al cliente final.

Además, el uso de Internet influye en gran medida en casi todos los ámbitos de la sociedad de la misma manera el uso de dispositivos móviles inteligentes los cuales son usados en el día a día por todas las personas.

Un objetivo fundamental de Bienestar estudiantil de la EPN es brindar a la comunidad politécnica atención integral en salud con personal humano calificado brindando servicios eficientes y de calidad.

Por tal razón se plantea el desarrollo de un sistema prototipo en Android para gestionar los turnos de los pacientes del departamento médico de la ESCUELA POLITÉCNICA NACIONAL, porque no existe un sistema que permita realizar esta función.

Este prototipo, de llegar a implementarse, podría ser usado por los estudiantes, personal administrativos y docentes de la ESCUELA POLITÉCNICA NACIONAL, así como por el personal de dicho departamento médico.

1.1 OBJETIVOS

El objetivo general de este Trabajo Técnico es: Desarrollar un sistema prototipo de gestión de turnos mediante teléfonos móviles con sistema Android para el departamento médico de la ESCUELA POLITÉCNICA NACIONAL.

Los objetivos específicos de este Trabajo Técnico son:

- Analizar los conceptos fundamentales de las herramientas que se van a utilizar para la realización del prototipo.
- Diseñar los módulos del sistema necesarios para la elaboración del sistema planteado.
- Implementar los módulos del sistema que van a formar parte del producto final.
- Analizar los resultados obtenidos en las pruebas desarrolladas.

1.2 ALCANCE

Se pretende realizar un Sistema Prototipo Web en Android [1] el cual permita gestionar los turnos de las citas médicas para las diferentes áreas que posee el Departamento Médico de la Escuela Politécnica Nacional.

El Departamento Médico de la EPN [2] posee 5 ramas o especialidades tales como: medicina general, odontología, nutrición, psicología y ginecología las cuales en la aplicación van a estar a disposición de los usuarios con el fin de obtener un turno dependiendo de sus necesidades. Cabe mencionar que habrá horarios de atención dependiendo de la especialidad y de la disponibilidad del personal médico.

Los usuarios al ingresar por primera vez al sistema prototipo deberán generar una clave para poder hacer uso de la aplicación.

Se destaca que solo podrán utilizar la aplicación los usuarios que sean personal activo de la EPN (estudiantes, empleados, o docentes).

En base a lo descrito anteriormente se definen los siguientes roles para los usuarios del sistema:

Paciente: Es el usuario que desea obtener un turno para una determinada cita médica en cualquiera de las especialidades antes mencionadas. Este utilizará la aplicación cliente para acceder al servicio.

Doctor: Es el usuario el cual, dependiendo de la especialidad, estará disponible para la respectiva revisión al rol paciente.

Administrador: Es el usuario que tiene las facultades para editar o eliminar usuarios o su información.

Cada rol tendrá diferente interfaz dependiendo de las necesidades de este.

Toda la información a ser gestionada va a ser almacenada en una base de datos [3], y en el momento que la información sea obtenida será procesada y su resultado se reflejará en el dispositivo, en este caso el resultado será hora, fecha, lugar, especialidad requerida y doctor el cual va a atender.

El sistema prototipo a desarrollar se basará en Kanban y el prototipo estará conformado por: cliente, que será la interfaz por la cual el usuario podrá manipular la aplicación, Servicio Web que estará alojado en Amazon Web Services (AWS) que se encargará de la lógica de

la obtención de turnos, y de una base de datos SQL que almacenará la información correspondiente.

La arquitectura del sistema web se muestra en la Figura 1.1.

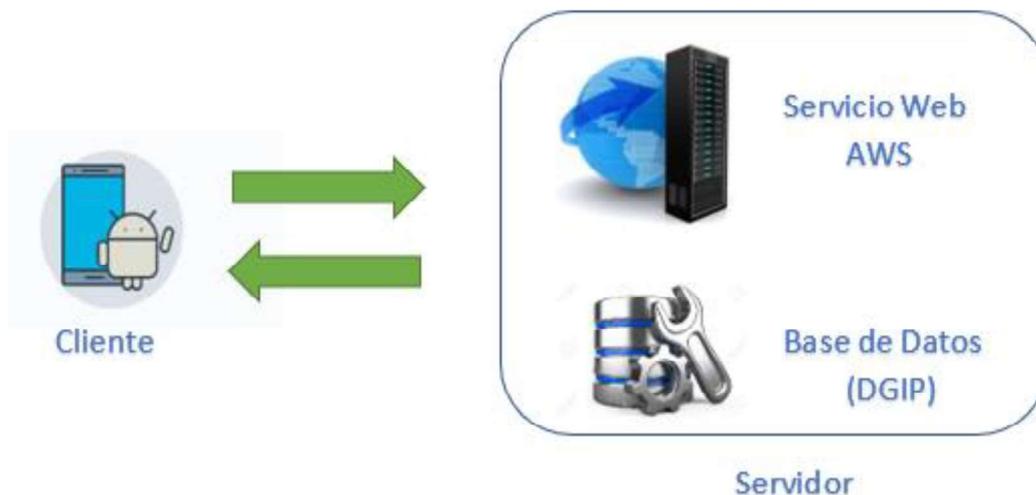


Figura 1.1 Arquitectura del sistema.

Nota: La DIRECCIÓN DE GESTIÓN DE LA INFORMACIÓN Y PROCESOS (DGIP) de la EPN entregará un fragmento de su base de datos del personal activo de la EPN basándose en sus políticas para la realización del sistema prototipo.

El sistema prototipo contará con los siguientes módulos (Requisitos mínimos):

- **Administración de usuario**

Este módulo contará con la capacidad para leer, editar y eliminar usuarios del sistema prototipo.

- **Autenticación**

Este módulo permitirá autenticar a los usuarios de la aplicación móvil con el sistema, esto dependerá si el usuario es doctor, paciente o administrador.

- **Gestión de turnos**

Interfaz donde el usuario paciente pueda escoger la especialidad que este necesite para la obtención del respectivo turno, a la hora y fecha más conveniente para dicho usuario en función de la disponibilidad de horarios.

- **Verificación de turnos**

Este módulo estará disponible para los doctores el cual tendrá una visualización de los turnos que éstos deben atender y a quienes deben atender.

- **Consulta de turnos**

Este módulo estará disponible para los pacientes el cual tendrá a opción de ver que turno tiene para su cita médica, hora, y que doctor lo atenderá.

- **Notificaciones**

Mediante este módulo el paciente tendrá una notificación al celular con S.O. Android, la cual le recordará un día antes que tiene una cita médica con los respectivos detalles de la misma.

Además, se realizará la interfaz gráfica de usuario con la cual se podrá acceder al sistema prototipo.

1.3 MARCO TEÓRICO

Últimamente tanto la salud como la tecnología han avanzado considerablemente, a su vez que estos han trabajado conjuntamente en beneficio de la sociedad, esto ha hecho que variedad de centros de salud hagan de la tecnología un gran aliado.

Además, tomando en cuenta que en la actualidad el uso de teléfonos inteligentes como de aplicaciones de todo tipo sean imprescindibles para las personas, es necesario automatizar la forma en que las personas puedan obtener turnos médicos en cualquier tipo de especialidad que estos necesiten desde cualquier lugar y en cualquier momento.

Esto conlleva a entrar en un concepto muy usado en la actualidad llamado Cloud Computing (Computación en la nube), la cual es una tecnología que nos permite tener acceso remoto a cualquier tipo de software, procesamiento de datos y almacenamiento de archivos y que a su vez es una gran red donde todo se encuentra interconectado [4].

En consecuencia, el presente proyecto de titulación permite desarrollar un sistema prototipo de gestión de turnos, el mismo que permitirá al usuario paciente obtener turnos para citas médicas desde cualquier lugar y en cualquier momento dependiendo de la necesidad de este usuario.

1.3.1 SMARTPHONES

En un principio la idea fundamental de un smartphone o teléfono inteligente fue básicamente unir funciones de un PDA (Personal Digital Assistant) con las funciones de un teléfono con la finalidad de dar comodidad a los usuarios finales. El primer dispositivo móvil

que usó el término “Smartphone” fue el Ericson GS88 desarrollado en 1997, el cual poseía funciones como: correo electrónico, reloj mundial, navegación web, un teclado QWERTY físico, reloj mundial, modo avión, puerto infrarrojo, conexión a PC [5].

En los últimos años el uso de dispositivos móviles inteligentes ha evolucionado de forma considerable, ocasionando así que sean prácticamente indispensables para todos sus usuarios en los distintos ámbitos del día a día.

La potencia de un teléfono inteligente actual se podría comparar y en ocasiones incluso superar a una computadora, estos además deben soportar un SO móvil (sistema operativo móvil) el cual debe tener su propia plataforma de desarrollo para sus aplicaciones, generando de esta forma posibilidades que años atrás no eran pensadas [6].

1.3.1.1 Análisis del mercado móvil

El mercado de los teléfonos inteligentes ha crecido de forma exponencial en los últimos años. Según Xataka [7] en el primer trimestre de 2019 el mayor fabricante de teléfonos inteligentes a nivel mundial sigue siendo Samsung con una cuota de mercado del 22.8%, por primera vez en la historia Huawei se consolida como el segundo mayor fabricante con una cuota mundial del 18.8%, en el tercer escalón se sitúa Apple reduciendo así su fabricación móvil, llegando a tener un mercado mundial de 12,8% [8].

La mayoría de los fabricantes de Smartphones utiliza Android como sistema operativo, a diferencia de Apple el cual utiliza IOS como sistema operativo móvil para sus dispositivos móviles. La Figura 1.2 muestra como está distribuido el mercado móvil por fabricante a nivel global

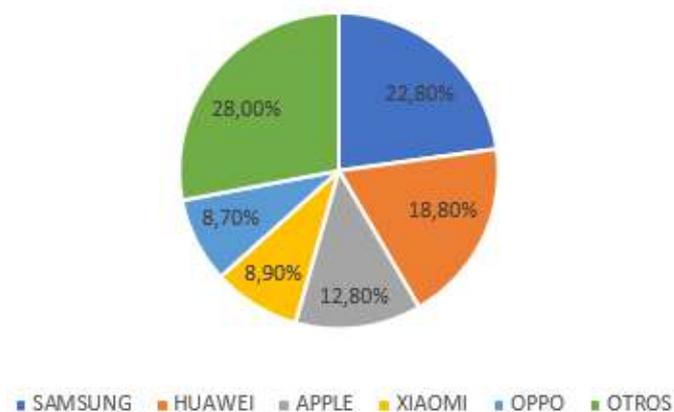


Figura 1.2 Distribución del mercado móvil a nivel global por fabricante.

En cuanto a los Sistemas Operativos Android es el dominador absoluto a nivel mundial ya que cuenta con una cuota del 86.6% de dispositivos que funcionan con este sistema operativo. Apple con su sistema operativo IOS, quien ha reducido su porcentaje con respecto a años anteriores posee el 12,8% del mercado mundial [9]. La Figura 1.3 muestra como está distribuido el mercado móvil por Sistema Operativo a nivel global.

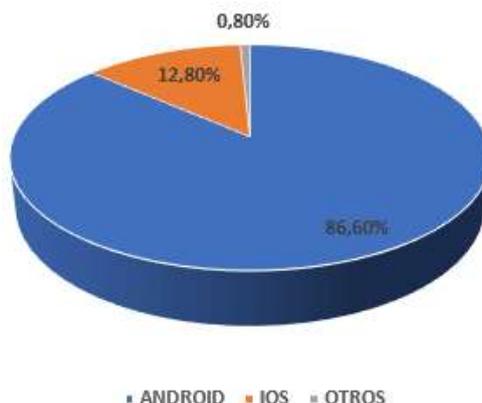


Figura 1.3 Distribución del mercado móvil por SO a nivel global.

El motivo de estas cifras se debe a los precios más cómodos que ofrecen terminales con SO Android en comparación a terminales Apple con IOS, así como la gran variedad de gamas. Sin olvidar que Android ofrece una mayor posibilidad de personalización. Se relaciona que por cada 10 Smartphones casi 9 cargan sistema operativo Android.

Estas fueron razones suficientes para elegir a Android como Sistema Operativo para la realización del presente proyecto de titulación.

1.3.2 APLICACIONES MÓVILES

Las aplicaciones móviles (también llamadas apps) son programas desarrollados para ser descargados y ejecutados en dispositivos móviles las cuales permiten al usuario final realizar múltiples actividades y servicios [10].

Se considera que la primera aplicación se creó a finales de los años 90, claramente no para teléfonos inteligentes sino para teléfonos analógicos, se las consideraban aplicaciones básicas como agenda, juegos, editor de tonos, etc. No obstante, desde que se pudo ingresar a internet por medio de un dispositivo móvil, se expandió la visión de oportunidades para crear distintos tipos de aplicaciones móviles.

La gran demanda de aplicaciones para Smartphones ha provocado que desarrolladores se planteen nuevos desafíos para lograr que estas aplicaciones tengan distintivas características, así como restricciones únicas. Esto conlleva a que tiendas virtuales como Google Play [11] o App Store [12] contengan una gran variedad de aplicaciones móviles de

todo tipo, lo cual permite que este mercado cada vez vaya incrementando con el paso del tiempo y el desarrollo de la tecnología.

Para la realización de una aplicación móvil se tiene que tomar en cuenta distintas etapas las cuales son de vital importancia para que el producto final sea de calidad y brinde los servicios necesarios al usuario final. Estas etapas son: Conceptualización, Definición, Diseño, Desarrollo, Publicación; abarcan desde la concepción de la idea, pasando por el desarrollo hasta el análisis después de la publicación en las respectivas tiendas de los diferentes sistemas operativos [13]. La Figura 1.4 muestra el proceso que deben seguir tanto el diseñador como el desarrollador



Figura 1.4 Etapas de una aplicación móvil.

Existen tres tipos de aplicaciones móviles: aplicaciones nativas, aplicaciones web y aplicaciones híbridas, las cuales las trataremos a continuación para tener una mayor percepción del alcance de estas.

1.3.2.1 Aplicaciones Nativas

Las aplicaciones Nativas son aquellas que se desarrollan con el software que ofrece cada sistema operativo llamado SDK (Software Development Kit), es decir tienen un lenguaje de programación específico [14]. Por ejemplo, para desarrollar aplicaciones para SO Android se utiliza Android Studio y para desarrollar aplicaciones con sistema operativo IOS se utiliza Swift.

Actualmente las aplicaciones nativas son las de más fácil acceso y las más encontradas. Este tipo de aplicaciones se pueden descargar desde cualquier tienda de aplicaciones ya sea en App Store para tecnología IOS y por Play Store para tecnologías Android.

Una característica fundamental de este tipo de apps es que no necesitan de internet para poder funcionar, por lo que su uso es más fluido y completamente integrado al dispositivo móvil, permitiendo así utilizar todo el hardware del dispositivo móvil como cámara, GPS,

acelerómetro, etc. Las aplicaciones más destacadas actualmente que representan a las apps nativas son distintas variedades de juegos, los cuales no necesitan de conexión a internet para su funcionamiento.

El mayor inconveniente que presenta este tipo de aplicaciones es que el costo de desarrollo es mayor que los demás, haciendo que se multiplique dependiendo de los sistemas a los que la aplicación se debe adaptar [15].

1.3.2.2 Aplicaciones Web

Este tipo de aplicaciones están programadas de forma independiente del sistema operativo, para que puedan correr en cualquier navegador a través de una URL, así con una aplicación Web podemos llegar a diferentes tipos de dispositivos sin importar el sistema operativo que estos utilicen.

Para su desarrollo se utilizan lenguajes muy populares como HTML, CSS, conocidos para la programación web. Además, estas aplicaciones no necesitan de un SDK para desarrollarlas por lo que se puede programar independientemente del sistema operativo del dispositivo móvil. A su vez estas no necesitan de una descarga y de instalación ya que corren en el navegador del respectivo dispositivo móvil, y por tal razón no se distribuyen en las tiendas de aplicaciones como las aplicaciones Nativas. Como ejemplos relevantes tenemos a Google Chrome en Android y a Safari en IOS.

Una de sus principales ventajas es que su precio de desarrollo es más económico, pero como sus puntos más débiles esta la necesidad de usar internet para usar todos los recursos del dispositivo móvil, afectando de esa manera su rendimiento [16].

1.3.2.3 Aplicaciones Híbridas

Estas aplicaciones como su nombre lo indica son una combinación de lo mejor de las aplicaciones Nativas y lo mejor de las aplicaciones Web. Cuentan con una estructura interna que se basa en lenguajes de desarrollo Web como HTML, CSS, JavaScript, pero a su vez también poseen una estructura externa basada en lenguajes de programación propios de las diferentes plataformas móviles. Esto hace que este tipo de aplicaciones sean compatibles con cualquier tipo de sistema operativo móvil [17].

En consecuencia, para que una aplicación funcione en los diferentes sistemas operativos no se deberá desarrollarla nuevamente desde cero, sino que solo se deberá modificar la parte externa para hacer así a la aplicación compatible con todas las plataformas. Además, estas aplicaciones si se pueden distribuir a través de las tiendas respectivas.

Un ejemplo bastante popular de estas aplicaciones híbridas es Netflix, el mismo que se ve y funciona de igual manera tanto en Android como IOS, teniendo un diseño visual que no diferencia por el Sistema Operativo [18].

Entre sus ventajas tenemos que estas aplicaciones si se distribuyen a través de las distintas tiendas de aplicaciones y la reutilización de código. Como desventajas se destaca que la interfaz gráfica es la misma para las diferentes plataformas, así como la experiencia de uso no se diferencia en ningún sentido.

La Tabla 1.1 muestra las ventajas y desventajas de cada uno de los diferentes tipos de aplicaciones móviles.

Tabla 1.1 Ventajas y desventajas de Aplicaciones móviles.

TIPO	VENTAJAS	DESVENTAJAS
Aplicación Nativa	<p>Pueden ser distribuidas a través de tiendas.</p> <p>Costos reducidos.</p> <p>No necesitan de conexión a internet.</p>	<p>Funcionan solo en el sistema operativo en el que fueron desarrolladas.</p> <p>Necesitan aprobación de tiendas para ser publicadas.</p>
Aplicación Web	<p>Se pueden usar desde cualquier tipo de dispositivo móvil.</p> <p>Costo de desarrollo menor.</p>	<p>Necesitan de internet para su funcionamiento.</p> <p>Menor rendimiento que aplicaciones Nativas.</p> <p>No usa los recursos del sistema de forma óptima.</p>
Aplicación Híbrida	<p>Aplicaciones multiplataforma.</p> <p>Pueden ser distribuidas en las tiendas de aplicaciones.</p> <p>Reutilización de código.</p>	<p>Interfaz gráfica similar para las diferentes plataformas.</p>

1.3.3 METOLOGÍA ÁGIL DE DESARROLLO KANBAN

Esta metodología nació para procesos de fabricación, pero con el paso del tiempo se la utilizó para el desarrollo de software, en donde se busca obtener un proceso organizado,

efectivo y eficiente para llevar a cabo diferentes tareas [19]. Kanban fue implementado a finales de los años 40 por Toyota traduciéndose literalmente en “señales visuales” el cuál se compone por tres columnas: por realizar, en proceso y realizado, por lo que se lo considera un proceso de mejora incremental ya que producirá un aumento significativo en el rendimiento. El trabajar con bloques a los cuales se los denomina tarjetas, permite ser más resolutivo y a su vez permite reducir tiempos de desarrollo, garantizando así un resultado fiable y de calidad debido a las fases con las que cuenta el sistema.

Los objetivos principales de Kanban son: estímulo de rendimiento, organización y colaboración y mejor distribución de trabajo.

1.3.3.1 Principios básicos de Kanban

Esta metodología se diferencia de las demás debido a que se basa en una serie de principios, los cuales detallamos a continuación [20].

- **Calidad garantizada.**

No debe haber margen de error y se tiene que realizar de forma correcta cualquier tarea que corresponda realizar. En esta metodología no se prioriza la rapidez sino la calidad, ya que en ocasiones cuesta mucho más arreglar que hacer de forma correcta desde un inicio.

- **Reducción del desperdicio.**

Esta metodología tiene como base realizar lo que es justo y no más que lo necesario, pero con la condición de que se lo haga bien. Por lo tanto, con esta premisa se reduce hacer todo lo que es superficial y secundario.

- **Mejora continua.**

Kanban no es solo un simple método de gestión, sino que es un sistema enfocado en mejorar cada vez más el desarrollo de proyectos, dependiendo de los objetivos propuestos al inicio de un proyecto.

- **Flexibilidad.**

Cuando hay que realizar tareas pendientes o acumulados, no siempre es necesario seguir un orden, sino que se debe priorizar aquellas tareas según la necesidad del momento, lo que se conoce como capacidad de respuesta a tareas imprevistas.

1.3.3.2 Pasos para configurar la estrategia Kanban.

Para configurar una estrategia Kanban se debe tener en cuenta los siguientes pasos [21].

- **Definir el flujo de trabajo de los proyectos.**

Se debe crear un tablero el cual debe estar visible para todos los miembros del grupo, así como deberá tener acceso por parte de dicho equipo de trabajo. Cada columna tiene un flujo de tareas que sirve para saber qué tan avanzado va el proyecto.

- **Visualizar las fases.**

Esta metodología se basa en un desarrollo incremental, dividiendo el trabajo en distintas partes representadas con tarjetas. Estas distintas tarjetas harán que se agilite el desarrollo del proyecto. Dichas tarjetas contienen información básica para que el equipo sepa exactamente que tiene que hacer, pueden incluir estimación en horas, responsable u observaciones.

- **Stop Starting, Start Finishing**

Este paso se lo considera como el lema fundamental de Kanban en donde se da prioridad a las tareas que están en proceso en vez de iniciar con nuevas tareas. Con esto se intenta resolver los problemas que habitualmente tienen las empresas que es tener varias tareas abiertas, pero sin haber finalizado muchas de ellas, intentando de esta manera cerrar las tareas abiertas antes de empezar con la siguiente.

- **Control de Flujo.**

Kanban procura que los trabajadores tengan un flujo constante de trabajo permitiendo hacer un seguimiento detallado del trabajo.

Existe variedad de software para realizar los tableros Kanban en línea, un ejemplo es Trello [22] el cual es un software orientado a la metodología ágil Kanban que permite gestionar múltiples tareas. Listas, tableros, tarjetas permiten priorizar y organizar diferentes proyectos de forma flexible.

En este tablero virtual se puede subir ideas, imágenes, tareas, enlaces teniendo como característica principal su fácil utilización y soporte para distintos tipos de Sistemas Operativos. Por estas razones se usó este software para la realización del presente proyecto de titulación. La Figura 1.5 muestra un ejemplo de cómo usar el tablero Kanban.



Figura 1.5 Ejemplo de un tablero Kanban.

1.3.4 DESCRIPCIÓN DE HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.

Para el desarrollo del Prototipo de gestión de turnos médicos es necesario usar varios lenguajes de programación y herramientas. A continuación, se detallarán las herramientas y lenguajes de programación que se usarán en este proyecto de titulación.

1.3.4.1 Android

Desarrollado por Google, es un Sistema Operativo móvil basado en el kernel de Linux. Fue creado en el año 2005 con la finalidad de crear una plataforma para cámaras digitales, pero más tarde se cambió el rumbo concluyendo que el mercado de cámaras digitales era muy pequeño para el Sistema Operativo que se estaba desarrollando, enfocándose así en dispositivos móviles para competir de esta manera con Sistemas Móviles de la época como Symbian y Windows Móvil.

El 23 de septiembre del 2008 se lanzaba al mercado el primer dispositivo móvil inteligente con SO Android y fue el puntal para que hoy en día sea el Sistema Operativo más usado en el planeta [23]. Esta nueva plataforma fue atractiva a los usuarios de inmediato ya que

integraba servicios populares de la empresa como Google Maps o YouTube, logrando convencer a la mayoría de los fabricantes de dispositivos móviles a que usen el sistema operativo de Google. [24].

A continuación, se muestra la Tabla 1.2 donde detalla una serie de características importantes que posee el sistema operativo Android [25].

Tabla 1.2 Características De S. O. Android.

Característica	Descripción
Diseño de dispositivo	Dependiendo de la resolución la plataforma se adapta a cualquier tipo de pantallas.
Almacenamiento	Para almacenamiento de datos usa SQLite.
Navegador Web	Se basa en el motor de código abierto WebKit, enlazando con motor de JavaScript de Google.
Conectividad	CDMA, GSM/EDGE, EV-DO, UMTS, HSPA+, NFC, Bluetooth, Wi-Fi, LTE, HSDPA, GPRS, UMTS y HSDPA+, IDEN.
Entorno de Desarrollo	Usa Android Studio que está basado en java.
Soporte Multimedia	Soporta diferentes formatos multimedia: MP4, MP3, AAC, WAV, JPEG, PNG, GIF y BMP.
Soporte Multi-táctil	Android tiene soporte multi-táctil de forma nativa para pantallas capacitivas
Multitarea.	Tiene multitarea real, una aplicación que no se está ejecutando recibe ciclos de reloj.

1.3.4.2 Plataforma de desarrollo de Aplicación Móvil.

Es un entorno de desarrollo integrado conocido con sus siglas IDE, creado por Google para desarrollar aplicaciones para el Sistema Operativo Android. Este entorno es multiplataforma ya que fue escrito en JAVA. En el 2014 se lanzó la primera versión estable bajo licencia de software libre, actualmente esta plataforma está disponible para Windows, Mac y Linux [26].

Además de las herramientas para desarrolladores y del óptimo editor de código, Android Studio brinda funciones que permiten aumentar la productividad durante la compilación de aplicaciones, como las siguientes [27]:

- Compatible con C++ y NDK.

- Herramientas para detectar problemas del rendimiento.
- Frameworks y herramientas de prueba.
- Emulador bastante rápido.
- Sistema de compilación se base en Gradle.
- El entorno de desarrollo permite realizar aplicaciones para todo tipo de dispositivos que tengan SO Android.

A continuación, se presenta la Tabla 1.3 con las características más importantes de la plataforma de desarrollo Android Studio [28].

Tabla 1.3 Características De Android Studio.

Característica	Descripción
Sistema de construcción y ayudas de codificación.	La refactorización es más potente, edición más fluida y mejoras en el análisis del código.
Previsualización	Se puede revisar de forma rápida y gradualmente la aplicación en diferentes tipos de resoluciones y densidades.
Ayudas para el diseño	Incluye diálogos para facilitar la corrección de errores de forma automática.
Acceso a Servicios de Google.	Google facilita el uso de los diferentes recursos para los proyectos como es el caso de Google Maps.

1.3.4.3 Base de Datos MYSQL y PhpMyAdmin

Es el sistema de gestión de base de datos relacionales más usado a nivel mundial, este gestor de base de datos es multiusuario y multihilo, permitiendo ser usado al mismo tiempo por varias personas [29].

Cuenta con doble licencia, por una parte, se inició como software libre usado hasta la actualidad para todo tipo de usuarios particulares, pero también tiene una distribución pagada usada generalmente para promover datos empresariales por lo que en este caso se puede usar una licencia propietaria que está distribuida por la empresa Oracle. La mayoría del código de MYSQL está desarrollado en C/C++ teniendo una sintaxis de uso simple permitiendo crear base de datos complejas o simples con bastante facilidad [30].

MYSQL se puede ejecutar básicamente en todo tipo de plataforma, ya sea Windows, Linux y UNIX, permitiendo interactuar con lenguajes de programación conocidos como PHP, JAVA o Perl. Además, este gestor de base de datos se lo usa para el desarrollo web, ya que permite a diseñadores y a desarrolladores hacer modificaciones de manera rápida y sencilla, sin tener que modificar directamente el código fuente.

Se asocia generalmente con aplicaciones basadas en la web. Cuando se combina con PHP se pueden realizar aplicaciones cliente/servidor las cuales requieren base de datos rápidas, sencillas de utilizar, seguras y potentes, como es el caso de MYSQL [31].

Por otro lado, PhpMyAdmin [32] es una herramienta visual escrita en PHP que permite realizar el diseño y gestión de base de datos para MYSQL a través del navegador. En la actualidad la mayoría de servidores y hostings compartidos utilizan esta aplicación.

Entre las características más importantes de PhpMyAdmin listamos las siguientes [33]:

- Interfaz fácil de usar e intuitiva.
- Posibilidad de administrar múltiples servidores.
- Permite realizar operaciones básicas de base de datos MYSQL como: crear, editar, eliminar, añadir tablas y campos.
- Importa y exporta base de datos en diferentes formatos.
- Optimiza y repara tablas, las cuales son opciones de mantenimiento.
- Ejecuta sentencias SQL.
- Posibilidad de crear copia de seguridad de una base de datos.

1.3.4.4 PHP.

PHP acrónimo de Hypertext Pre-processor, es un lenguaje de programación de código de abierto popular para el uso de desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP es que el código se lo ejecuta en el servidor, generando código HTML y enviándolo al cliente [34].

Se utiliza principalmente para generar páginas web dinámicas, las cuales contienen su información en una base de datos, pero también son usadas en el desarrollo de páginas web estáticas.

Este lenguaje es procesado en servidores, que son computadores con hardware y software especial. Cuando se escribe una dirección tipo

http://www.servidor_utilizado.com/index.php en un navegador cualquiera, se envían datos de la solicitud a un servidor para procesarlos, una vez reunidos todos los datos el servidor devuelve esos datos en una página HTML simulando como página estática. El esquema es el siguiente [35]:

Páginas estáticas: Petición → Respuesta

Páginas dinámicas: Petición → Procesado y Preparación → Respuesta

Las cuatro grandes características por las que se identifica PHP son las siguientes [36]:

- **Velocidad.**

A parte de que este lenguaje es rápido, no genera retrasos en la máquina, por lo que no requiere grandes recursos del sistema.

- **Estabilidad.**

Es un sistema robusto y estable ya que utiliza su propio sistema para administración de recursos.

- **Seguridad.**

Los niveles de seguridad se los puede configurar en el archivo .ini.

- **Simplicidad.**

Al poseer una alta gama de librerías permite su facilidad en el uso. Se lo considera uno de los lenguajes más fáciles de usar del mercado.

1.3.4.5 Amazon Web Services.

Amazon Web Services (AWS) es un conjunto de servicios y herramientas de Cloud (nube) de la empresa Amazon, sus inicios fueron en el año 2006 y actualmente es la empresa líder en servicios en la nube a nivel mundial. Empresas gigantes a nivel global usan el servicio de Cloud Computing de Amazon, entre las más conocidas aparecen: Netflix, General Electric, Airbnb, etc. Esto se debe al amplio abanico de herramientas y servicios que ofrece la empresa, así como a su madurez frente a sus demás competidores, teniendo como principal rival a Microsoft con su Azure [37].

Uno de los servicios más usados e importantes de AWS es *Amazon Elastic Computer Cloud* (Amazon EC2) que brinda capacidad informática con la posibilidad de escalamiento seguro almacenado en la nube, disponible para los usuarios. Mediante este servicio los usuarios pueden configurar su propio Servidor con las características que este necesite, ya

sea de tipo Ubuntu, Windows, SUSE Linux, entre otras, estableciendo características internas del servidor como memoria RAM, número de CPU, soporte IPV6 o almacenamiento [38]. AWS permite al usuario manejar instancias EC2 de nivel libre conocidas como “*EC2 Free-Tier*”, pero con recursos y duración limitados, generalmente de 12 meses.

La filosofía que mantiene Amazon Web Services es que pagas por lo que utilizas y además pagas poco. En la Tabla 1.4 podemos observar las 5 principales características de AWS [39].

Tabla 1.4 Características De Amazon Web Services.

Características	Descripción.
Auto Servicio en demanda.	El consumidor es quien decide las capacidades de computación, procesamiento, almacenamiento según sus necesidades, sin tener interacción humana con el proveedor de servicio.
Acceso total a servicios.	Se puede acceder a los servicios desde cualquier lugar y desde cualquier dispositivo.
Puesta en Común de servicios.	La asignación y reasignación de recursos físicos y virtuales se realiza dinámicamente y de forma inmediata.
Rápida elasticidad.	Para el consumidor las capacidades que dispone son prácticamente ilimitadas.
Medición transparente.	AWS Permite controlar y monitorear el uso de los recursos para no exceder las capacidades contratadas.

1.3.4.6 Servicios Web.

Se describe como Servicio Web a una tecnología que usa diferentes estándares y protocolos para intercambiar datos entre diferentes aplicaciones a través de la red sin necesidad de la intervención humana con el fin de ofrecer servicios. Esta tecnología permite que distintas aplicaciones que pueden estar desarrolladas en lenguajes diferentes se comuniquen y se integren entre si [40].

El internet es también un espacio de interacción y no solo de información, donde los usuarios solicitan algún tipo de servicio por medio de una aplicación a algún proveedor, pero para que esto funcione debe haber mecanismos de comunicación estándares entre las diferentes aplicaciones que están interactuando, de tal manera que se pueda mostrar la información requerida de forma dinámica al usuario. Básicamente la interacción se trata de solicitudes y respuestas entre un cliente y un servidor por medio del envío de datos [41].

Se clasifican en servicios web SOAP (Simple Object Access Protocol) y servicios web REST (Representational State Transfer); el primero utiliza el protocolo HTTP el cual define como se deben realizar las comunicaciones entre cliente y servidor, usa lenguaje XML que especifica el formato de los mensajes para el intercambio de datos. Por otro lado, los servicios web REST también usan el protocolo HTTP para la comunicación entre cliente y servidor, protocolo que es soportado por todo tipo de sistemas, para esta comunicación utiliza mensajes de formato JSON (JavaScript Object Notation) [42].

En la actualidad la combinación de REST con JSON es la opción más popular y sencilla para la implementación de servicios web

1.3.4.7 JSON.

JSON es el acrónimo de *JavaScript Object Notation*, es un formato basado en texto plano para intercambiar información entre dispositivos, esta información se puede enviar o mostrar desde un sistema y puede ser interpretada por otros sistemas. El formato de JSON al ser independiente de cualquier tipo de lenguaje de programación no necesita hablar el mismo idioma, esto quiere decir que el emisor puede ser JAVA y el receptor puede ser PHP ya que cada tipo de lenguaje tiene librerías propias las cuales decodifican cadenas de JSON [43].

JSON está compuesto por 2 estructuras básicas: objetos y lista de valores ordenados. Los objetos son un conjunto de pares de valores: nombre y valor, estos están separados por comas y su delimitación es por medio de llaves {}. La lista de valores ordenados pueden ser arreglos, vectores o listas, las cuales son una colección de múltiples valores que están separados por comas y delimitados por corchetes "[]" [44].

El código 1.1 muestra un ejemplo de mensaje JSON, donde muestra la información del doctor correspondiente a medicina general que usamos en el proyecto de titulación.

```
 {"doctor": [{"0": "1111111111", "cedula_doctor": "1111111111",  
 "1": "LUIS MALAN", "nombre_doctor": "LUIS MALAN",  
 "2": "MEDICINA GENERAL", "nombre_especialidad": "MEDICINA GENERAL",  
 "3": "nopassword", "password_doctor": "nopassword"}]}
```

Código 1.1 Ejemplo de mensaje JSON.

En la Figura 1.6 se observa la arquitectura que tiene un sistema que utiliza servicio web, los clientes se comunican con un servidor remoto para realizar múltiples peticiones por medio de un servicio web, el servidor devuelve las respectivas respuestas a cada uno de los clientes. El servicio web se lo desarrolla en PHP y la información es enviada por medio

de mensajes JSON para que la aplicación que se encuentra en el cliente pueda entender la información.

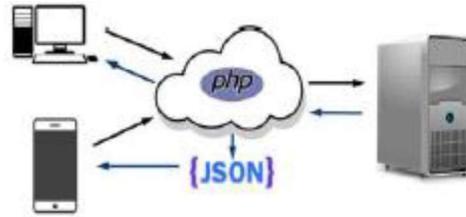


Figura 1.6 Arquitectura de un servicio web, usando formato JSON.

1.3.4.8 Servicio FTP y Fillezilla.

FTP Es un protocolo de red para transferencia de archivos, acrónimo de File Transfer Protocol, el cual está basado en arquitectura Cliente/Servidor [45]. Para el presente trabajo de titulación este protocolo es de suma importancia ya que va a permitir transferir archivos desde nuestro computador hasta la instancia Ubuntu montada en Amazon Web Services, esto debido a que nuestro servidor no tiene interfaz gráfica.

Por otro lado, Fillezilla es una herramienta de software de código abierto el cual permite conectar nuestro computador con un servidor web, que en este caso lo tenemos montado con Ubuntu, para transferir archivos, los archivos que se transfieren son los servicios web que están desarrollados en lenguaje PHP.

Esta herramienta está diseñada para aprovechar el protocolo FTP, por tal razón permite una transferencia de archivos a gran velocidad a través de un servidor. Este software tiene seguridad ya que para poder acceder a él debemos introducir la dirección IP y la respectiva clave del servidor web [46].

La Figura 1.7 muestra un ejemplo básico de como actúa un sistema cliente servidor cuando se despliega un evento, en este caso el evento que se ejemplifica es el botón Agendar Cita, el mismo que hace referencia a almacenar información en la base de datos MYSQL luego de interactuar el cliente con el sistema prototipo elaborado en Android.

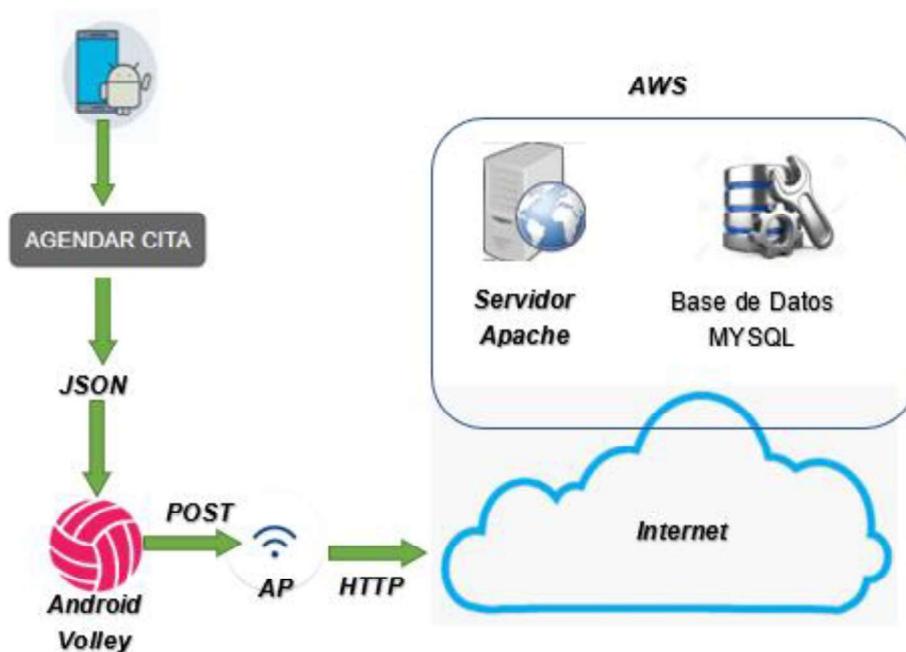


Figura 1.7 Comportamiento del sistema Cliente – Servidor.

Un usuario que interactúa con el sistema prototipo genera un evento llamado Agendar Cita, este evento tiene un mensaje en formato JSON el cual al combinarlo con la librería Volley y haciendo uso de un método POST utiliza el protocolo HTTP para hacer peticiones por medio de internet. Una vez que el servidor comprueba que la petición existe, este se conecta con una base de datos MYSQL para hacer la respectiva consulta. Y devolverla a la aplicación móvil.

2. METODOLOGÍA

El presente capítulo consta de dos apartados, el Diseño y la Implementación. En la fase de Diseño se definen los requerimientos que deben cumplir el prototipo los cuales establecen el comportamiento del sistema. El segundo apartado indica la Implementación del prototipo donde se muestra la instalación de herramientas y la codificación del diseño.

Tanto para la etapa del Diseño, así como para la etapa Implementación los componentes que integran el prototipo son separados en Capa de Datos, Capa de Negocio y Capa de Presentación.

2.1 DISEÑO

En esta parte, se pretende mostrar el diseño realizado para las diferentes capas de este proyecto de titulación, tanto para la Capa de Datos, Capa de Negocio, Capa de Presentación.

Como punto inicial definimos un tablero de actividades Kanban el cual no sirve como guía para el desarrollo del prototipo. Los siguientes pasos para seguir son definir los requerimientos tanto Funcionales como los requerimientos No Funcionales del sistema planteado. En la capa de Datos se realiza un diagrama relacional; para la Capa de Negocio y una vez obtenidos los diferentes requerimientos, se diseñará el diagrama de casos de uso, diagrama de clases y diseño de la arquitectura web; finalmente en la Capa de Presentación se mostrarán las vistas de la aplicación móvil.

2.1.1 TABLERO DE ACTIVIDADES KANBAN

La Figura 2.1 muestra el tablero Kanban el mismo que permite organizar las tareas de la fase de diseño, permitiendo así una visibilidad completa de todas las actividades necesarias para realizar la aplicación móvil.

En la figura se muestran el tablero con tres columnas: la primera columna muestra una lista de tareas que se van a realizar con el título "*Lista de Tareas*"; la segunda columna muestra una lista de tareas que se están realizando actualmente, lleva el título "*Tareas en Proceso*"; finalmente la tercera columna contiene una lista con tareas que ya fueron finalizadas, esta lleva el nombre de "*Tareas Realizadas*" [47].

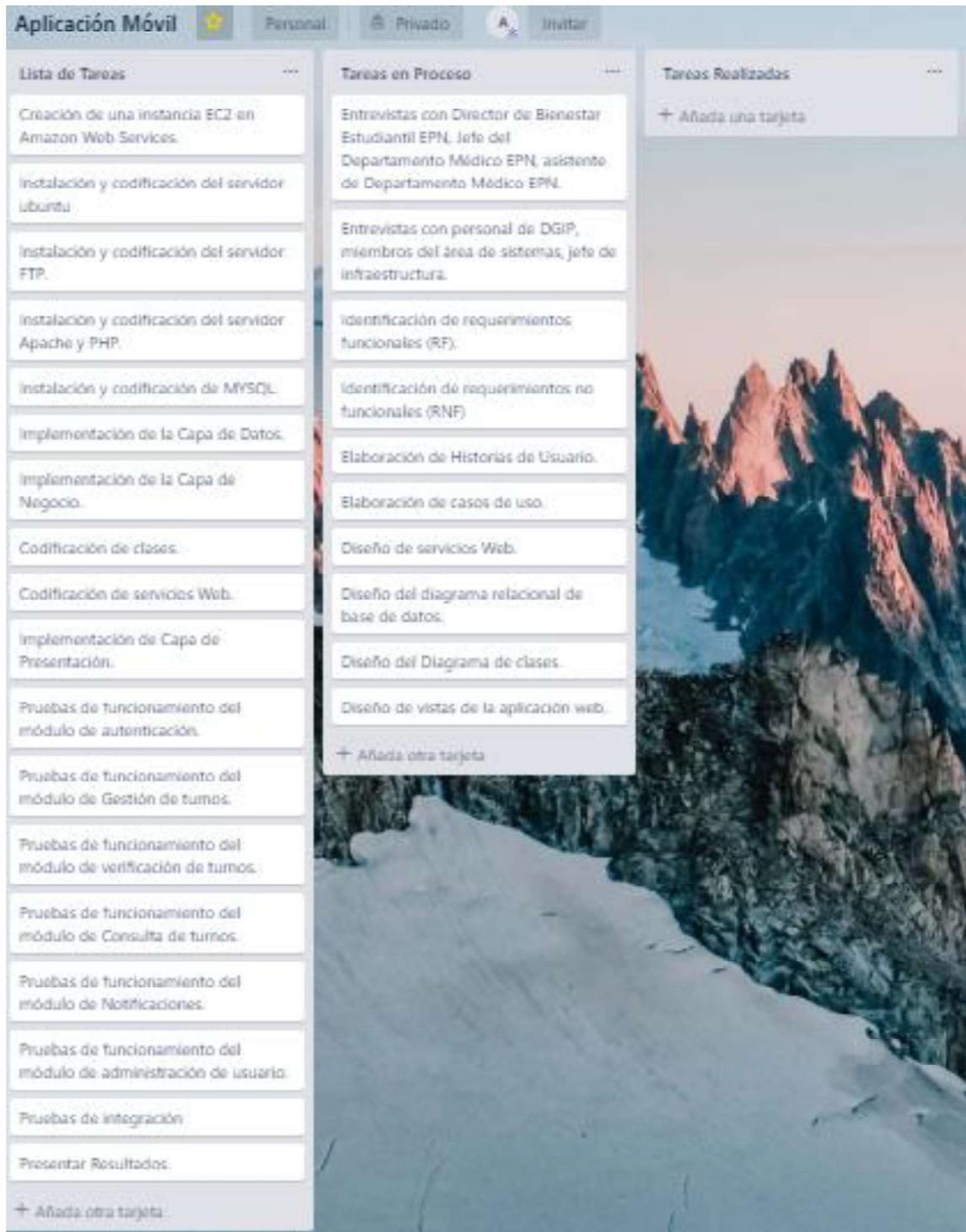


Figura 2.1 Tablero Kanban para fase de Diseño.

2.1.2 ANÁLISIS DE REQUERIMIENTOS.

Los requerimientos para el desarrollo del prototipo fueron definidos en base a diversas entrevistas a miembros del Departamento Médico de la ESCUELA POLITÉCNICA NACIONAL y a miembros de la DIRECCIÓN DE GESTIÓN DE LA INFORMACIÓN Y PROCESOS (DGIP) durante los meses de noviembre, diciembre del 2018 y enero de

2019. Después de analizar cada una de las entrevistas se pudo obtener una serie de requerimientos los mismos que se dividen en dos secciones: Requerimientos Funcionales (RF) [48] y Requerimientos No Funcionales (RNF) [49].

2.1.2.1 Requerimientos Funcionales (RF)

Los requerimientos funcionales son aquellos que describen las distintas funcionalidades que el prototipo móvil debe realizar, en otras palabras, describen como debe funcionar y comportarse el software desarrollado [50].

En relación a las entrevistas realizadas a las siguientes personas funcionarias de la Escuela Politécnica nacional: Dr. Klever Mejía (Director de Bienestar estudiantil y médico del departamento médico), Dr. Gonzalo Garcés (Jefe de departamento médico), Sra. Ivis Cuellar (Asistente de departamento médico), Ing. Fausto Sarrade (Director de bienestar estudiantil a partir de febrero de 2019); así como a los ingenieros: Geovanna Saltos (Jefe de área de sistemas DGIP), Ing. Juan Pablo Ponce (Jefe de Infraestructura DGIP), Ing. Mónica Játiva (Área de Sistemas DGIP); se definen una serie de requerimientos funcionales para el desarrollo del prototipo de la aplicación móvil.

2.1.2.1.1 Historias de usuario

Debido a la flexibilidad para la organización se hará uso de historias de usuario para la toma de requerimientos [51]. Las historias de usuario levantadas se encuentran en el ANEXO A. La tabla 2.1 indica como es el formato de una historia de usuario con sus respectivos elementos.

Tabla 2.1 Formato de una Historia de Usuario.

Número:		Fecha:	
Prioridad:		Estimación:	
Tipo:			
Nombre:			
Descripción:			

La historia de usuario contiene los siguientes elementos:

- **Número:** Es un número único que identifica a cada historia de usuario. Está compuesto por las siglas "HU" (Historia de usuario) y un número expresado en orden. Por ejemplo: HU001

- **Prioridad:** Indica el nivel de importancia de la historia de usuario dentro del desarrollo del proyecto, se define 3 tipos de prioridades: alto, medio, bajo.
- **Tipo:** Indicará si es un requerimiento funcional o un requerimiento no funcional con su respectiva notación.
- **Nombre:** Breve descripción de la historia de usuario.
- **Fecha:** Indica la fecha en la cual se definió la historia de usuario.
- **Estimación:** Indica el número de horas estimado para la implementación.
- **Descripción:** Describe la funcionalidad de la historia de usuario.

A continuación, se muestran algunos ejemplos de historias de usuario obtenidas en la toma de Requerimientos Funcionales. En la Tabla 2.2 se detalla la Historia de Usuario 001.

Tabla 2.2 Historia de Usuario 001.

Número:	HU001	Fecha:	10-may-19
Prioridad:	Alta	Estimación:	14 h
Tipo:	Requerimiento Funcional RF001		
Nombre:	Tipo de usuario		
Descripción:			
Esta historia de usuario permite escoger el tipo de usuario (estudiante, docente, empleado, doctor) que va a utilizar el sistema prototipo.			

En la Tabla 2.3 se detalla la Historia de Usuario 002.

Tabla 2.3 Historia de Usuario 002.

Número:	HU002	Fecha:	10-ago-19
Prioridad:	Alta	Estimación:	12 h
Tipo:	Requerimiento Funcional RF002		
Nombre:	Generar clave		
Descripción:			
Sirve para que el usuario pueda generar su clave de acceso al ingresar por primera vez al sistema prototipo.			

En la Tabla 2.4 se detalla la Historia de Usuario 003.

Tabla 2.4 Historia de Usuario 003.

Número:	HU003	Fecha:	10-ago-19
Prioridad:	Alta	Estimación:	15 h
Tipo:	Requerimiento Funcional RF003		
Nombre:	Autenticar como médico.		
Descripción:			
El usuario médico puede acceder al sistema prototipo haciendo uso de las respectivas credenciales.			

A continuación, se muestran algunos ejemplos de historias de usuario obtenidas en la toma de Requerimientos No Funcionales. La Tabla 2.5 detalla la Historia de Usuario 023.

Tabla 2.5 Historia de Usuario 023.

Número:	HU023	Fecha:	12-may-19
Prioridad:	Media	Estimación:	10 h
Tipo:	Requerimiento No Funcional RNF003		
Nombre:	Tiempo de Respuesta		
Descripción:			
Dependiendo de la velocidad de internet, el tiempo de respuesta debe ser casi inmediato.			

La Tabla 2.6 detalla la Historia de Usuario 024.

Tabla 2.6 Historia de Usuario 024.

Número:	HU024	Fecha:	12-may-19
Prioridad:	Baja	Estimación:	10 h
Tipo:	Requerimiento No Funcional RNF004		
Nombre:	Escalabilidad		
Descripción:			
El prototipo debe ser construido en una plataforma que permita incrementar recursos dependiendo de las necesidades: procesador, almacenamiento, RAM.			

La Tabla 2.7 muestra la lista de los requerimientos funcionales que posee el prototipo, su nomenclatura es RF (Requerimiento Funcional) seguido de un numero expresado en orden ascendente. Por ejemplo: RF001.

Tabla 2.7 Requerimientos Funcionales.

Historia de Usuario	Requerimiento Funcional	Descripción
HU001	RF001	Escoger el tipo de usuario.
HU002	RF002	Generar clave.
HU003	RF003	Autenticar como médico.
HU004	RF004	Revisar citas médicas a tratar.
HU005	RF005	Autenticar como paciente.
HU006	RF006	Escoger Especialidad para cita médica.
HU007	RF007	Visualizar información del médico.
HU008	RF008	Escoger fecha para cita médica.
HU009	RF009	Escoger horario para cita médica.
HU010	RF010	Comprobar disponibilidad de horario.
HU011	RF011	Agendar cita médica.
HU012	RF012	Verificar citas médicas vigentes.
HU013	RF013	Visualizar lista de citas médicas agendadas.
HU014	RF014	Anular citas médicas agendadas.
HU015	RF015	Recibir notificación de cita médica vigente.
HU016	RF016	Autenticar como administrador.
HU017	RF017	Administrar usuarios.
HU018	RF018	Buscar información de un usuario.
HU019	RF019	Actualizar información de un usuario.
HU020	RF020	Eliminar un usuario.

2.1.2.2 Requerimientos No Funcionales (RNF)

Los requerimientos no funcionales son aquellos que no describen las funcionalidades que va a tener el prototipo, sino que describen cualidades observables como: Presentación, tiempo de ejecución; así como cualidades relacionadas con la evolución del prototipo como: Escalabilidad, Disponibilidad [52].

La Tabla 2.8 muestra la lista de los requerimientos no funcionales que posee el prototipo, su nomenclatura es RNF (Requerimiento No Funcional) seguido de un numero expresado en orden ascendente. Por ejemplo: RNF001

Tabla 2.8 Requerimientos No Funcionales.

Requerimientos no Funcionales			
	Historia de Usuario	Requerimiento no funcional	Descripción
Presentación	HU021	RNF001	El prototipo debe ser intuitivo y fácil de utilizar para los usuarios.
Disponibilidad	HU022	RNF002	El prototipo debe estar disponible al menos un 99% del tiempo
Tiempo de respuesta	HU023	RNF003	Dependiendo de la velocidad de internet, el tiempo de respuesta debe ser casi inmediato.
Escalabilidad	HU024	RNF004	El prototipo debe ser construido en una plataforma que permita incrementar recursos dependiendo de las necesidades: procesador, almacenamiento, RAM.
Autenticación	HU025	RNF005	El acceso al prototipo debe estar restringido por usuario y contraseña para todos los usuarios.
Base de Datos	HU026	RNF006	La información de la aplicación está basada en tablas que la DGIP entregó para la realización del prototipo.

2.1.3 MÓDULOS DEL PROTOTIPO.

Para la obtención de los módulos del prototipo se debe agrupar los requerimientos funcionales por afinidad, con la finalidad de que cada módulo cumpla con funciones específicas necesarias que requiera el sistema prototipo de obtención de turnos médicos.

- **Módulo Autenticación:** Este módulo permite autenticar a los usuarios de la aplicación móvil, escogiendo entre cuatro tipos de usuario: estudiante, docente, empleado, doctor; una vez escogido el tipo de usuario se debe ingresar el número de cédula y una contraseña; cabe mencionar que la contraseña debe ser generada al inicio del ingreso de la aplicación.

- **Módulo Verificación de Turnos:** Este módulo está disponible exclusivamente para los doctores, donde dispondrán de una lista de las diferentes citas médicas que estos deben atender.
- **Módulo Gestión de Turnos:** A través de la aplicación móvil, los usuarios pacientes (estudiantes, docentes, empleados) pueden escoger una especialidad médica, dependiendo de sus necesidades, para la obtención de sus respectivos turnos médicos. Acto seguido el paciente puede escoger la fecha y el horario más conveniente para su cita médica, claro está dependiendo de la disponibilidad de dicho horario escogido.
- **Módulo Consulta de Turnos:** A través de la aplicación móvil, los pacientes tienen la posibilidad de ingresar a un apartado donde pueden visualizar las diferentes citas médicas que estos agendaron, cada cita médica esta detallada con la información necesaria para que el paciente decida si sigue adelante con la misma o de lo contrario opte por la opción de anular la cita médica pendiente.
- **Módulo de Notificaciones:** Mediante este módulo, una vez que el paciente reserve una cita médica dependiendo de sus necesidades, llegará una notificación al dispositivo móvil con Sistema Operativo Android, la cual le recordará al usuario que tiene una cita médica pendiente.
- **Módulo Administración de Usuario:** Al presente módulo solo puede acceder un usuario que tenga las credenciales de administrador, aquí el administrador contará con la capacidad de buscar, editar y eliminar cualquier tipo de usuarios o su información. Cabe mencionar que los usuarios ingresados fueron otorgados por la Dirección de Gestión de la Información y Procesos (DGIP), divididos en diversas tablas.

La Tabla 2.9 muestra una estimación aproximada en horas para la ejecución de los requerimientos funcionales para el Módulo Autenticación.

Tabla 2.9 Estimación RF Módulo de Autenticación.

Módulo	Historia de Usuario	Estimación en horas
Módulo de Autenticación	RF001	14
	RF002	12
	RF003	15

	RF005	15
	Total	56

La Tabla 2.10 muestra una estimación aproximada en horas para la ejecución de los requerimientos funcionales por Módulo Verificación de Turnos.

Tabla 10. Estimación RF Módulo de Verificación de Turnos.

Módulo	Historia de Usuario	Estimación en horas
Módulo de Verificación de turnos	RF004	16
	Total	16

La Tabla 2.11 muestra una estimación aproximada en horas para la ejecución de los requerimientos funcionales por Módulo Gestión de Turnos.

Tabla 2.11 Estimación RF Módulo Gestión de Turnos.

Módulo	Historia de Usuario	Estimación en horas
Módulo Gestión de Turnos	RF006	10
	RF007	10
	RF008	16
	RF009	14
	RF010	18
	RF011	12
	Total	80

La Tabla 2.12 muestra una estimación aproximada en horas para la ejecución de los requerimientos funcionales para el Módulo Consulta de Turnos.

Tabla 2.12 Estimación RF Módulo Consulta de Turnos.

Módulo	Historia de Usuario	Estimación en horas
Módulo Consulta de Turnos	RF012	14
	RF013	16
	RF014	14
	Total	44

La Tabla 2.13 muestra una estimación aproximada en horas para la ejecución de los requerimientos funcionales Módulo de Notificaciones.

Tabla 2.13 Estimación RF Módulo de Notificaciones.

Módulo	Historia de Usuario	Estimación en horas
Módulo de Notificaciones	RF015	16
	Total	16

La Tabla 2.14 muestra una estimación aproximada en horas para la ejecución de los requerimientos funcionales Módulo Administración de Usuario.

Tabla 2.14 Estimación RF Módulo Administración de Usuario.

Módulo	Historia de Usuario	Estimación en horas
Administración de Usuario	RF016	8
	RF017	4
	RF018	8
	RF019	8
	RF020	8
	Total	36

2.1.4 DISEÑO DE LA CAPA DE DATOS.

En este apartado se realiza el diseño del diagrama de entidad relación de la base de datos a utilizar, este diagrama contiene tablas (entidad) con sus respectivos campos, también se va a observar cómo estas tablas se relacionan con otras por medio de campos que las tablas tienen en común (relaciones).

Este diseño será utilizado para la creación de la base de datos que se utilizará en el sistema prototipo utilizando MYSQL [53].

2.1.4.1 Diagrama Relacional para la Base de Datos.

La Figura 2.2 muestra el diagrama Entidad-Relación que tendrá la base de datos del sistema de Gestión Turnos Médicos. Este tipo de diagramas se los describen como Diagramas de Flujo, los mismos que muestran entidades como: personas, objetos o conceptos los cuales se relacionan entre si dentro de un sistema [54].

Para la realización de este diagrama se usó el diseñador de modelo entidad-relación de phpMyAdmin el mismo que permite crear la base de datos para acoplarla con el servidor creado en AWS. Las tablas que almacenan la información del sistema prototipo, así como las respectivas relaciones se muestran en la Figura 2.2, las mismas que se detallan a continuación.

Las tablas estudiante, docente y empleado fueron entregadas por parte de la Dirección de Gestión de la Información y Procesos (DGIP), después del respectivo análisis y permiso por parte del Vicerrector de Docencia de la Escuela Politécnica Nacional, como muestra el ANEXO B, estas tablas son independientes y tienen su propio formato y parámetros de acuerdo a la información que la DGIP maneja internamente, por lo que para la realización del prototipo se debió manejar el formato que miembros de la DGIP manifestaron. Estas tablas contienen información de cada uno de los pacientes, información que solo personal autorizado debe manejar.

La tabla doctor permite guardar información de cada uno de los cinco especialistas que tiene el departamento médico de la EPN, esta tabla está relacionada directamente con la tabla especialidad ya que cada doctor posee una especialidad, la misma que está disponible para que un paciente pueda escoger y obtener un turno médico de acuerdo a sus necesidades.

La tabla horario, contiene información de las horas en las cuales cada especialidad está disponible para una determinada cita médica.

Finalmente, la tabla cita es el resultado final de la interacción de los usuarios pacientes (docente, estudiante, empleado), es decir aquí se guarda la información de cada una de las citas médicas que los usuarios en general reservaron, escogiendo previamente fecha, hora, especialidad y verificando la disponibilidad de horarios. Esta tabla se relaciona con cada una de las tablas previamente descritas, ya que esta contiene información de cada una de las anteriores tablas señaladas.

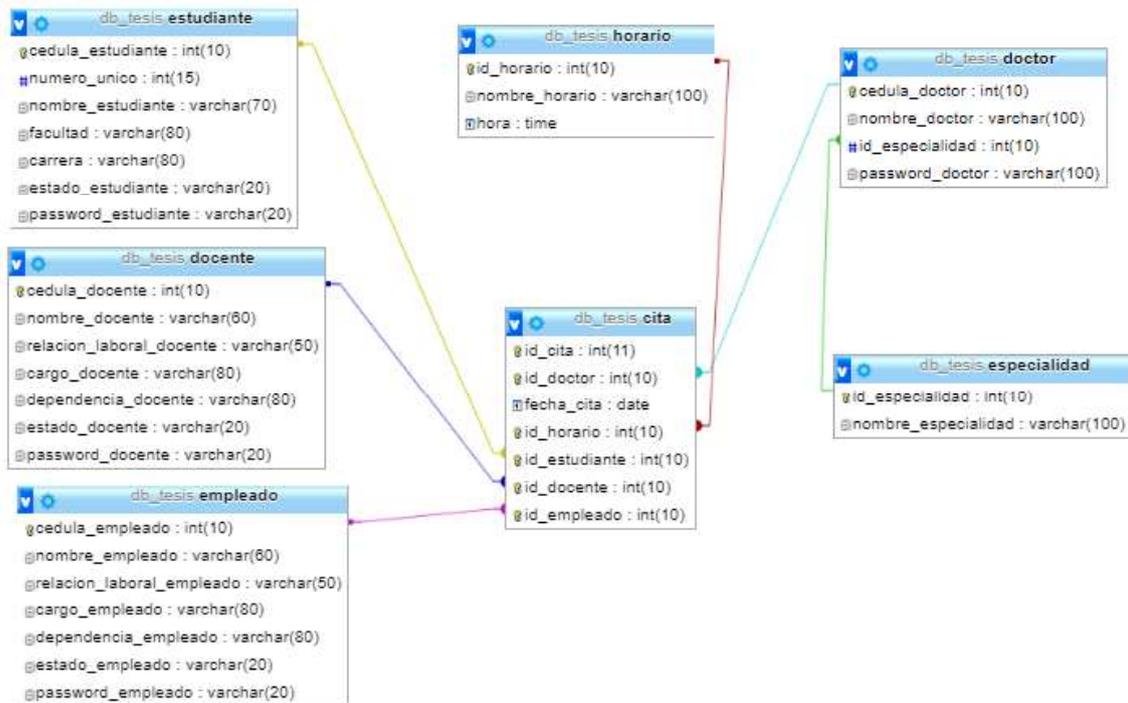


Figura 2.2 Diagrama Entidad-Relación de base de datos.

2.1.5 DISEÑO DE LA CAPA DE NEGOCIO

Esta capa representa la lógica de cada uno de los procesos que permiten implementar el sistema prototipo de Gestión de Citas Médicas, entre los cuales están el diagrama de casos de uso, diagrama de clases, así como el diseño del servicio web.

2.1.5.1 Diagrama de Casos de Uso

Un caso de uso hace referencia a una descripción de las acciones que los usuarios del sistema realizan con la aplicación, este diagrama muestra a los usuarios finales del sistema que interactúan con el sistema prototipo. Básicamente se trata de representar una historia de forma visual y sencilla en formato UML, la cual tiene personajes, acciones y contexto [55].

La Figura 2.3 muestra a los usuarios o personajes que interactúan con el sistema prototipo.

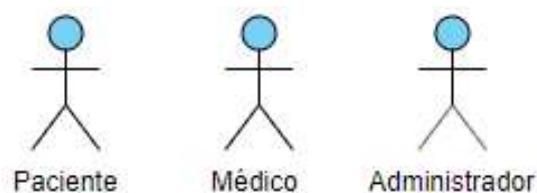


Figura 2.3 Actores que interactúan con el sistema prototipo.

A continuación, se describe la función de cada uno de los actores que interactúan con el sistema prototipo.

- **Usuario Paciente:** En el sistema prototipo final este tipo de usuario corresponde a las personas que requieran reservar una cita médica en cualquiera de las especialidades disponibles, estas personas son estudiantes, docentes o empleados de la EPN, los cuales pueden ingresar al sistema prototipo utilizando su número de cédula y una contraseña para, de acuerdo a sus necesidades, reservar una cita médica en el área deseada dependiendo de su tiempo y horario disponible.
- **Usuario Doctor:** En el sistema prototipo final este tipo de usuario corresponde a los médicos de las diferentes especialidades que posee el Departamento Médico de la EPN (Medicina General, Odontología, Nutrición, Psicología y Ginecología), los mismo que pueden ingresar al sistema prototipo utilizando su número de cédula y una contraseña, para verificar cuando y a qué hora cada médico tiene una cita médica pendiente a tratar.
- **Usuario Administrador:** Este usuario es único con la capacidad de buscar, editar y eliminar información de todos los usuarios anteriormente descritos. Para ingresar al sistema prototipo este usuario tiene predefinido un usuario y una contraseña.

Cada requerimiento funcional es representado por un caso de uso, por este motivo como título de cada caso de uso se usará el nombre del requerimiento funcional correspondiente. Los casos de uso se los clasifica dependiendo de los módulos que se han definido anteriormente.

Los casos de uso correspondientes para el “**Módulo de Autenticación**”, el cual permite autenticar a los usuarios del sistema prototipo, son los siguientes:

La Figura 2.4 muestra el caso de uso para escoger el tipo de usuario (estudiante, docente, empleado, doctor) que va a utilizar el sistema prototipo.

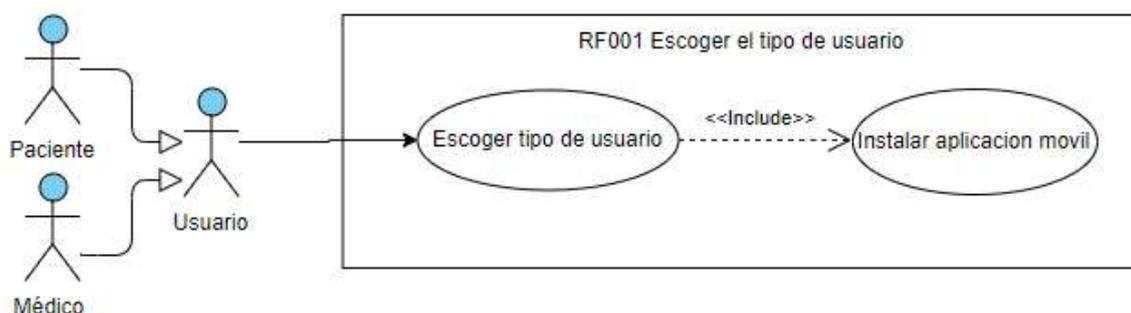


Figura 2.4 Caso de uso escoger el tipo de usuario que va a utilizar el prototipo.

La Figura 2.5 muestra el caso de uso para que el usuario pueda generar su clave de ingreso al sistema prototipo.

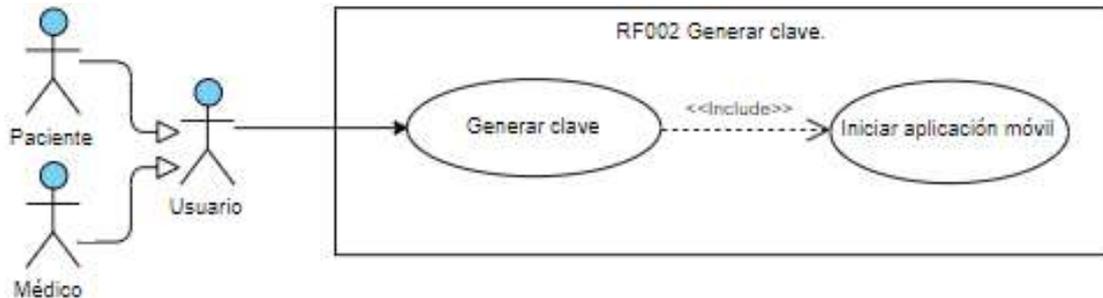


Figura 2.5 Caso de uso generar clave.

La Figura 2.6 muestra el caso de uso para ingresar al sistema prototipo como usuario médico.



Figura 2.6 Caso de uso autenticar como médico.

La Figura 2.7 muestra el caso de uso para ingresar al sistema prototipo como usuario paciente.

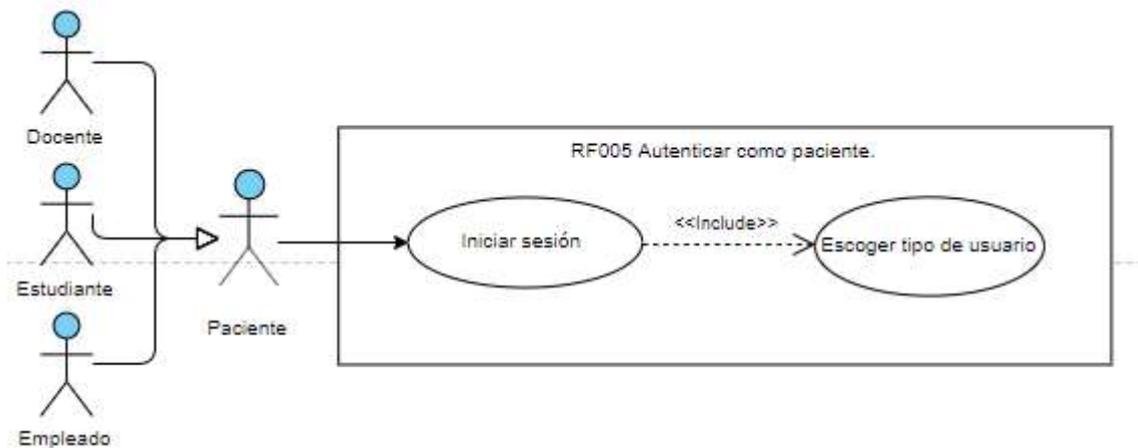


Figura 2.7 Caso de uso autenticar como paciente.

Los casos de uso correspondientes para el “**Módulo Gestión de Turnos**”, el cual permite a los usuarios obtener un turno médico dependiendo de sus necesidades, son los siguientes:

La Figura 2.8 muestra el caso de uso para que el paciente escoja cualquiera de las especialidades que el Departamento Médico ofrece.

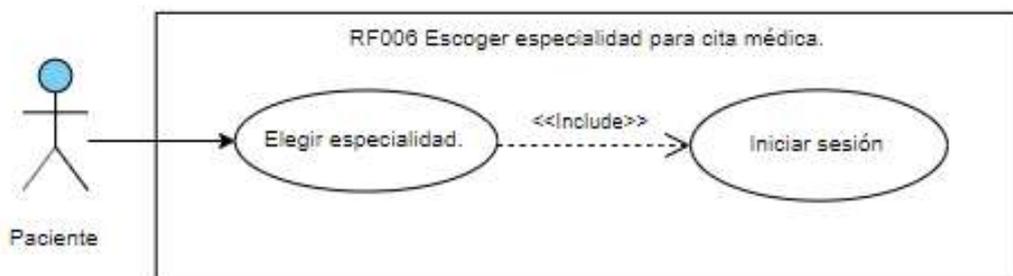


Figura 2.8 Caso de uso escoger especialidad para cita médica.

La Figura 2.9 muestra el caso de uso para observar la información más relevante del médico a cargo de la especialidad escogida.



Figura 2.9 Caso de uso visualizar información del médico.

La Figura 2.10 muestra el caso de uso para escoger una fecha para la cita médica dependiendo de las preferencias del paciente.

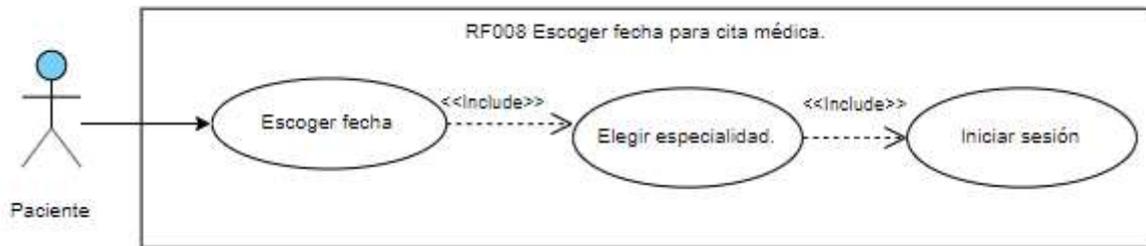


Figura 2.10 Caso de uso escoger fecha para cita médica.

La Figura 2.11 muestra el caso de uso para escoger un horario para la cita médica dependiendo de las preferencias del paciente.

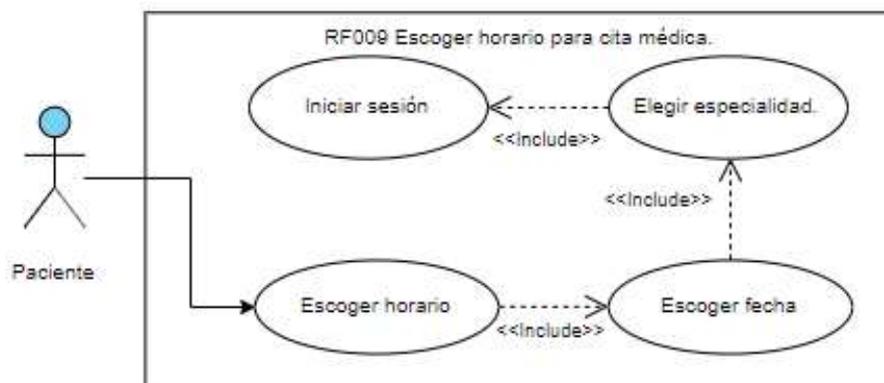


Figura 2.11 Caso de uso escoger horario para cita médica.

La Figura 2.12 muestra el caso de uso para comprobar que la fecha y hora que eligió el paciente estén disponibles para la respectiva cita médica.

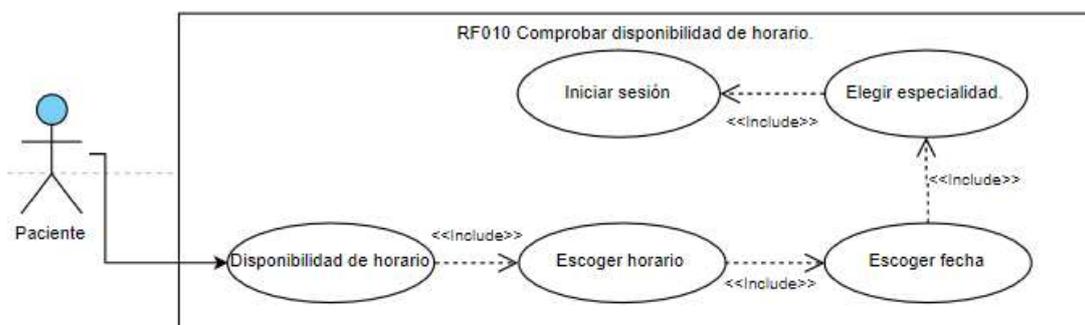


Figura 2.12 Caso de uso comprobar disponibilidad de horario.

La Figura 2.13 muestra el caso de uso para que el paciente pueda agendar la cita médica una vez que esté de acuerdo con las características de la misma.

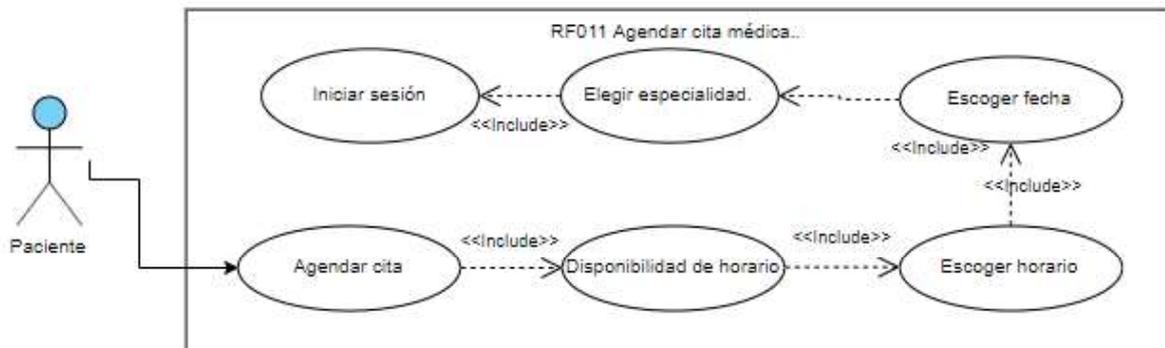


Figura 2.13 Caso de uso agendar cita médica.

El caso de uso correspondiente para el “**Módulo de Verificación de turnos**”, el cual permite al médico revisar las citas médicas que tiene vigentes para tratar, es el siguiente:

La Figura 2.14 muestra el caso de uso para que el doctor revise que citas médicas tiene que tratar.

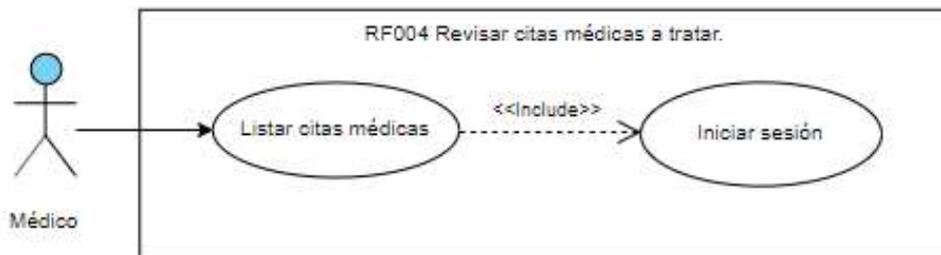


Figura 2.14 Caso de uso revisar citas médicas a tratar.

Los casos de uso correspondientes para el “**Módulo Consulta de Turnos**”, el cual permite a los pacientes revisar la información de las citas médicas que estos agendaron dependiendo de sus necesidades, son los siguientes:

La Figura 2.15 muestra el caso de uso donde el paciente visualice una lista de las diferentes citas médicas que este haya agendado.



Figura 2.15 Caso de uso visualizar citas médicas vigentes.

La Figura 2.16 muestra el caso de uso para visualizar un resumen con los datos de una cita médica específica que el paciente agendó.

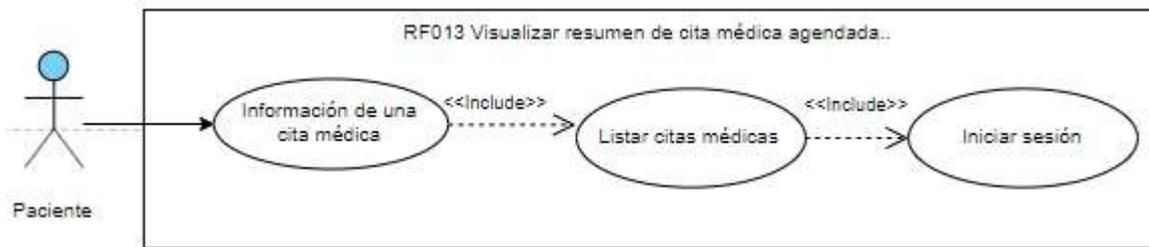


Figura 2.16 Caso de uso visualizar resumen de cita médica agendada.

La Figura 2.17 muestra el caso de uso para que el paciente pueda anular una cita médica específica en caso de que no requiera de la misma.

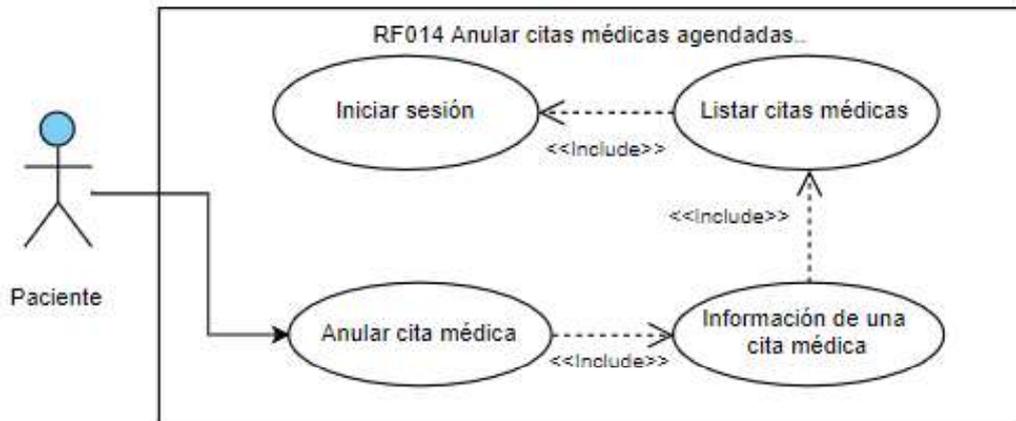


Figura 2.17 Caso de uso anular cita médica.

Los casos de uso correspondientes para el “**Módulo de Notificaciones**”, el cual permite obtener una notificación que llegará al dispositivo móvil con S.O. Android para recordar la cita médica, son los siguientes:

La Figura 2.18 muestra el caso de uso para que el paciente reciba un recordatorio a su dispositivo móvil por medio de una notificación.

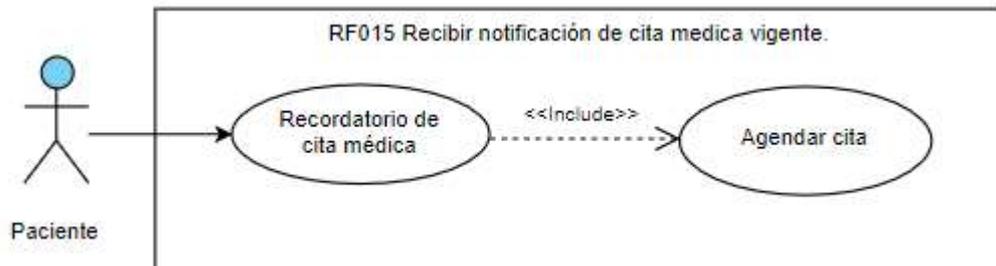


Figura 2.18 Caso de uso recibir notificación de cita médica vigente.

Los casos de uso correspondientes para el “**Administración de Usuario**”, el cual permite a un administrador buscar, editar o eliminar información de cualquier usuario, son los siguientes:

La Figura 2.19 muestra el diagrama de caso de uso donde un administrador se autentique para administrar el sistema prototipo.



Figura 2.19 Caso de uso autenticar como administrador.

La Figura 2.20 muestra el caso de uso donde el administrador escoge que tipo de usuario desea administrar.

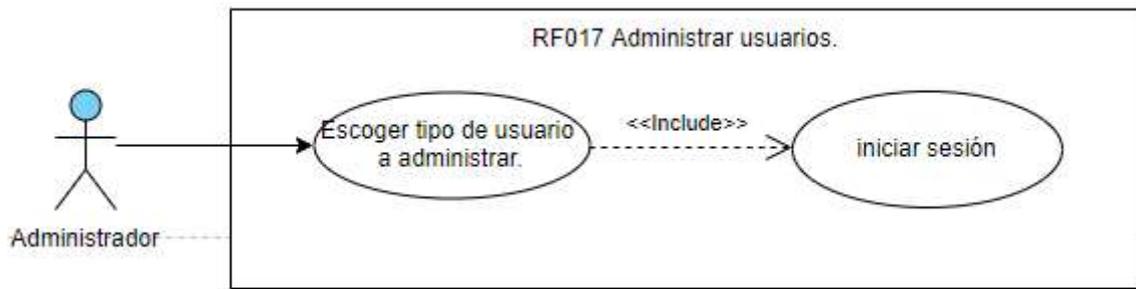


Figura 2.20 Caso de uso administrar usuarios.

La Figura 2.21 muestra el caso de uso donde el administrador por medio de un número de cédula busca la información de un usuario.

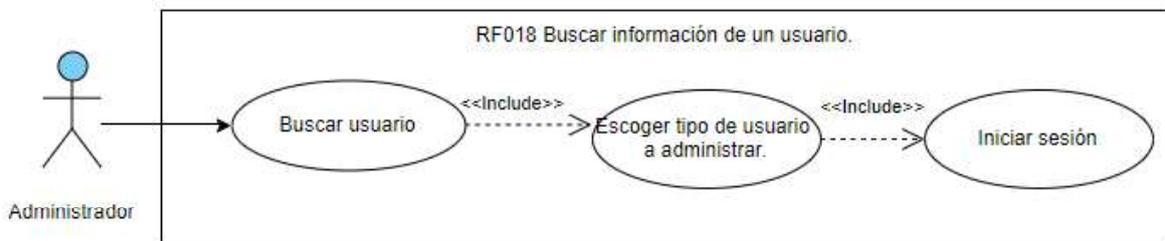


Figura 2.21 Caso de uso buscar información de un usuario.

La Figura 2.22 muestra el caso de uso donde el administrador actualiza la información de un usuario en particular.

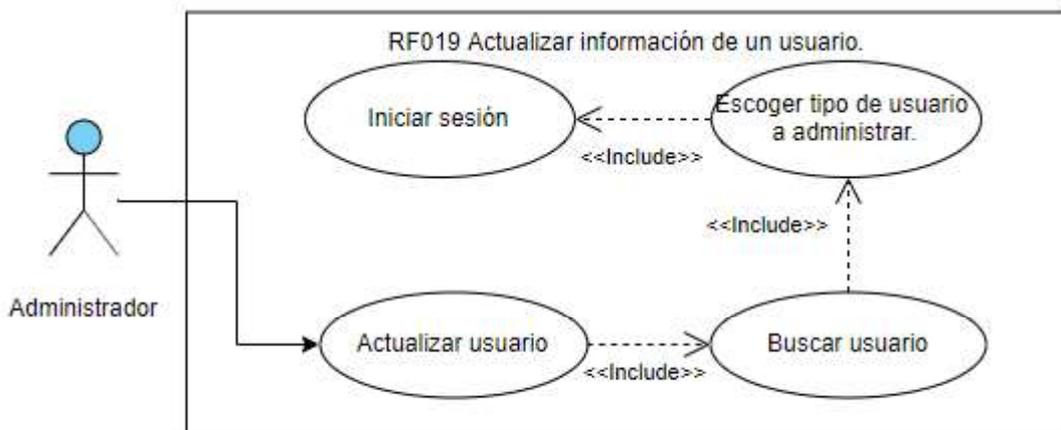


Figura 2.22 Caso de uso actualizar información de un usuario.

La Figura 2.23 muestra el caso de uso donde el administrador elimine un usuario en particular en caso de ser necesario.

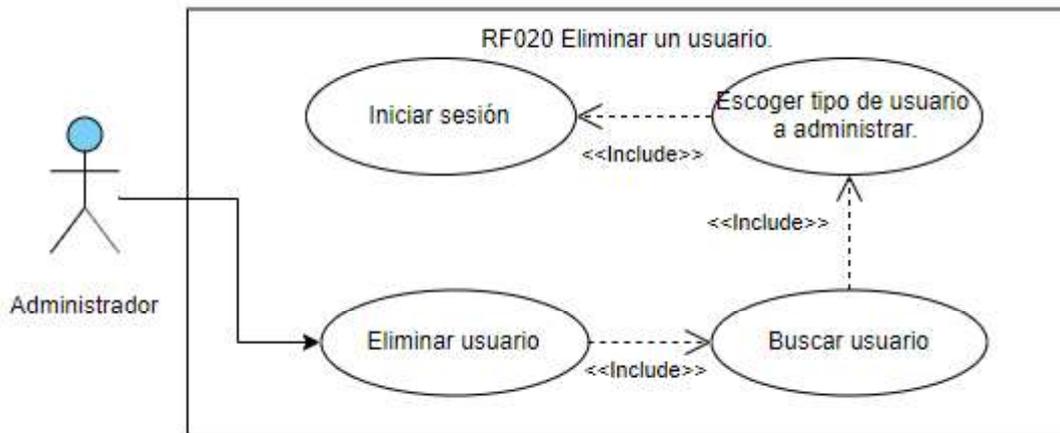


Figura 2.23 Caso de uso eliminar usuario.

2.1.5.2 Diagrama de Clases

El diagrama de clases es una herramienta creada para mostrar el diseño de una aplicación orientada a objetos donde se presentan las clases y sus asociaciones de manera gráfica. En UML las clases se la representa con un rectángulo que posee tres divisiones, la primera contiene el título de la clase, la segunda donde se listan todos sus atributos y finalmente la tercera división contienen los métodos que esta clase realiza [56].

La Figura 2.24 muestra el diagrama de clases con sus métodos más importantes, que sirve para implementar la lógica de la capa negocio del sistema prototipo. Estas clases son utilizadas para la estructura del sistema prototipo móvil.

Se utiliza el software “*Lucidchart*” [57] para la realización del mencionado diagrama de clases, la cual es una herramienta de código abierto especializada en la creación de diagramas UML.

Se observa que las clases “Docente”, “Estudiante”, “Empleado” y “Cita” tienen una cardinalidad de uno a varios lo cual significa que una instancia de las clases “Cita” puede tener varias instancias de las clases “Docente”, “Estudiante” y “Empleado”; y con la clase “Hora_especialidad” tienen una cardinalidad de “uno” a “uno” lo cual significa que una cita solo puede tener un horario en una especialidad. Por otro lado, la clase “Lista cita” hace referencia a que la lista desplegada por los usuarios puede tener tres citas agendadas semanalmente por estos, esto se visualiza mediante la cardinalidad de “uno” a “varios” representada en el diagrama de clases UML.

Finalmente, la clase “Administrador” hereda los atributos de la clase “Empleado” ya que el administrador, que tiene funciones de buscar, modificar y eliminar usuarios, va a ser un trabajador activo de la Escuela Politécnica Nacional.

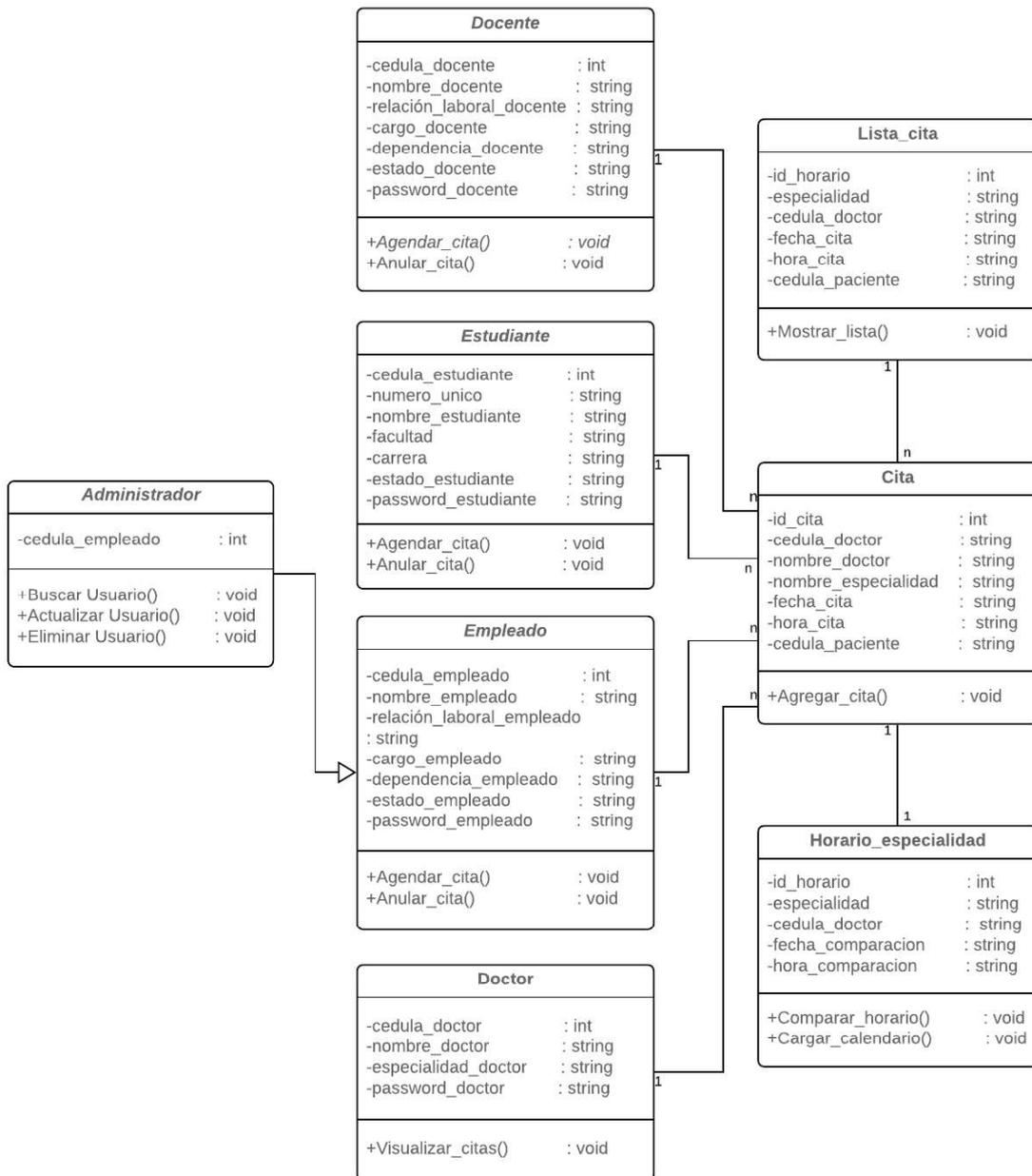


Figura 2.24 Diagrama de clases del sistema prototipo.

2.1.5.3 Arquitectura del Servicio Web

Para el desarrollo de la programación de los servicios web se usa el lenguaje PHP [34], estos servicios permiten darle varias funciones al sistema prototipo de gestión de turnos médicos. El utilizar lenguaje PHP para generar servicios web conlleva algunas ventajas como: PHP es un lenguaje relativamente fácil de usar y entender en comparación a otro tipo de lenguajes como C o JAVA; también PHP se puede acoplar con la mayoría de las plataformas como Linux, Mac OS o Windows.

La Figura 2.25 muestra la lógica del servicio web.



Figura 2.25 Arquitectura del servicio web.

La arquitectura que se representa en la Figura 2.25 muestra a usuarios de dispositivos móviles con sistema operativo Android, realizar peticiones al servidor por medio de mensajes JSON [43]; estas peticiones son enlazadas y procesadas por el respectivo servicio web, el cual se presenta mediante archivos codificados en PHP los mismos que brindan las distintas funcionalidades del sistema prototipo. Una vez que el servicio web recibe todos los datos correctamente procede a realizar las respectivas consultas a la base de datos codificada en MYSQL. Acto seguido el servidor devuelve las respuestas también en formato JSON, las cuales son decodificadas en el lado del cliente para mostrar la información que el cliente haya solicitado. Todos los servicios web contienen la lógica para ejecutar acciones solicitadas por el cliente Android.

2.1.6 DISEÑO DE LA CAPA DE PRESENTACIÓN

En esta capa se encuentran las interfaces gráficas con las cuales el usuario final va a interactuar. Estas interfaces corresponden al sistema prototipo para gestión de turnos médicos, las cuales se trata de diseñar de la forma más amigables posible para el usuario.

El diseño que se plantea cumple con los requerimientos que se establecieron anteriormente. Para el desarrollo de de la Capa de Presentación se utiliza la herramienta online “*Visual Paradigm Online Diagram*” [58] la cual contiene un apartado para diseñar interfaces gráficas para smartphones.

La Figura 2.26 muestra la interfaz de inicio cuando se abre el sistema prototipo, la cual van a observar todos los usuarios (Estudiante, Docente, Empleado, Doctor), estos deben escoger el tipo de usuario que se va a autenticar, escogiendo las opciones en el Spinner inicial.



Figura 2.26 Interfaz de inicio.

2.1.6.1 Diseño de las Vistas del sistema prototipo Usuario Paciente

El sistema prototipo cuenta con diferentes interfaces gráficas. Los diversos bosquejos de las interfaces gráficas que son utilizadas por el usuario paciente (Estudiante, Docente, Empleado) se muestran a continuación. Las vistas se han diseñado tomando en cuenta los requerimientos funcionales y casos de uso planteados en el apartado 2.1.2.1 y en el apartado 2.1.5.1 respectivamente.

La Figura 2.27 muestra la transición desde que el usuario escogió la opción de ingresar como paciente hasta la pantalla de inicio una vez que el paciente se autentica de forma correcta.



Figura 2.27 Interfaz de vista autenticación a vista inicio para usuario paciente.

Como muestra la Figura 2.28, para el caso de que el paciente desee agendar una cita nueva, se muestra una interfaz gráfica donde están listadas todas las áreas médicas que dispone la EPN para que el paciente escoja la que desee dependiendo de sus necesidades, ofertando en este caso las especialidades de Medicina General, Odontología, Nutrición, Ginecología y Psicología.

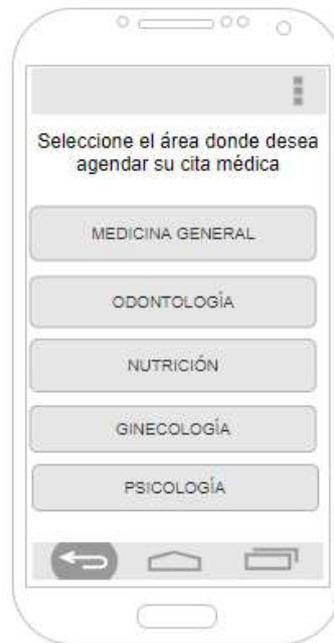


Figura 2.28 Interfaz lista de áreas médicas.

La Figura 2.29 muestra la transición que realiza el sistema prototipo desde que el paciente escoge la fecha, hora de una especialidad en específico para reservar una cita, hasta la interfaz donde el paciente puede agendar la cita médica con las características escogidas anteriormente. De esta manera dependiendo si hay disponibilidad de horarios la cita se guardará.



Figura 2.29 Interfaz de vista horarios a vista agendar cita.

Una vez que un paciente tiene una o varias citas agendadas, tiene la opción de verificar cuales son las citas que el agendó, así también en caso de no estar de acuerdo con alguna de estas va a tener la opción de anularla presionando en el basurero como se muestra en la Figura 2.30



Figura 2.30 Interfaz vista y anulación de cita agendada.

2.1.6.2 Diseño de las Vistas del sistema prototipo Usuario Doctor

Para el usuario doctor el sistema prototipo tiene 2 transiciones, la primera transición empieza desde que el usuario escoge autenticarse como doctor, una vez que las credenciales fueron correctas la interfaz inicio del doctor muestra únicamente una opción, lo que da paso a la interfaz final donde muestra una lista de diversas citas médicas que el doctor debe atender en el área específica a la que este pertenece. Esto se refleja en la Figura 2.31.

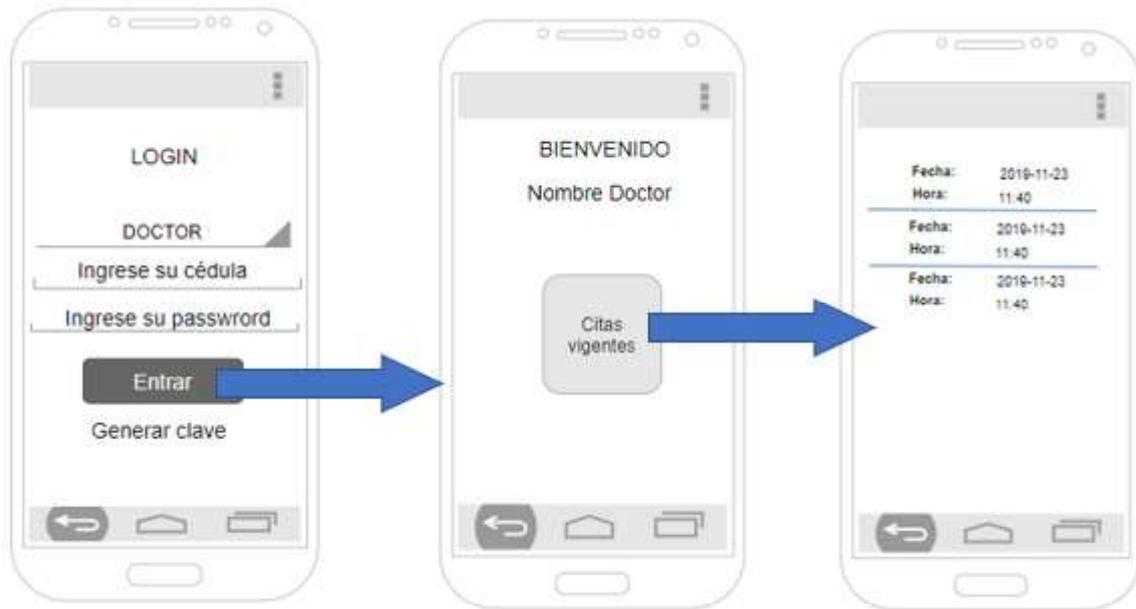


Figura 2.31 Interfaz para usuario Doctor.

2.1.6.3 Diseño de las Vistas del sistema prototipo Usuario Administrador

La Figura 2.32 muestra el diseño de la interfaz gráfica para el usuario Administrador, donde para ingresar como administrador se debe presionar sobre el menú superior, representado por 3 puntos. Una vez que se ingresa al módulo administrador la autenticación previamente definida es la base para poder utilizar las funcionalidades de administración. Ya dentro de la administración, el usuario tiene la capacidad de escoger los diferentes tipos de usuarios que este quiere administrar, las secciones disponibles son: administrar Estudiante, Doctor, Empleado y Docente. Para después realizar opciones de buscar, editar y eliminar algún usuario en particular.



Figura 2.32 Interfaz Usuario Administrador.

2.2 IMPLEMENTACIÓN

En esta fase se muestra el proceso con el cual se lleva a cabo la realización del sistema prototipo para generación de turnos médicos. Primero se inicia con la actualización del tablero Kanban con todas las actividades que se necesitan para realizar esta etapa, luego se presenta la instalación y codificación de las herramientas de software que se utilizan, además se presenta la codificación de lo necesario para que el sistema prototipo de gestión de turnos médicos funcione correctamente como se plantea en la sección 2.1. finalmente se muestra un ejemplo de la codificación de las interfaces gráficas de la aplicación a desarrollar.

2.2.1 ACTUALIZACIÓN DEL TABLERO DE ACTIVIDADES KANBAN

La Figura 2.33 indica el estado de las actividades del tablero Kanban. Estas actividades se han organizado de acuerdo como especifica dicha metodología, donde se puede observar que la fase de diseño está completamente finalizada, todas las actividades que se encuentran en proceso son aquellas que corresponden a la fase de implementación, la cual estamos tratando en este apartado. Finalmente, las tareas que aún están pendientes son aquellas que corresponden a la fase de pruebas.



Figura 2.33 Tablero Kanban para fase de Implementación.

2.2.2 ESQUEMA DEL SERVIDOR

La Figura 2.34 muestra los diferentes servicios que se deben instalar en el servidor Ubuntu en Amazon Web Services. Para almacenar la información del sistema prototipo se utiliza la base de datos MySQL; además, para almacenar los diferentes servicios web se usará un servidor web Apache, estos servicios web son los que permiten realizar diferentes funcionalidades al sistema prototipo; finalmente, se instalará un servicio FTP para la transferencia de archivos directamente desde Windows a nuestro servidor Ubuntu.

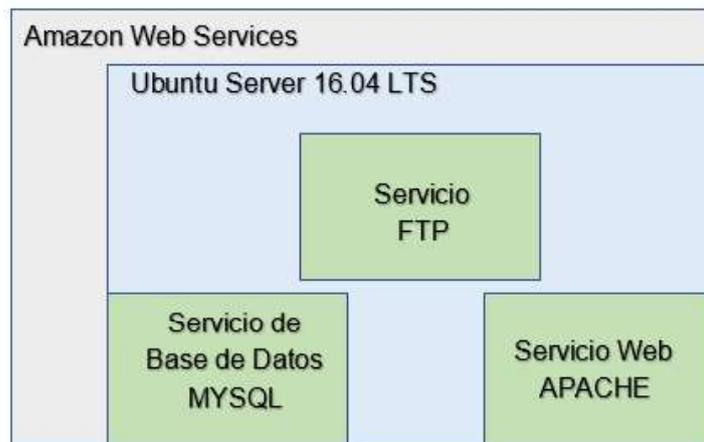


Figura 2.34 Arquitectura del servidor.

Se utiliza un servicio de Base de Datos MYSQL debido a que es el gestor más usado a nivel mundial en el ámbito del desarrollo de aplicaciones, además es multiusuario, es decir que puede ser utilizado al mismo tiempo por varios usuarios; sin olvidarse su compatibilidad con el gestor de Base de Datos phpMyAdmin el cual lo usamos para la realización del presente proyecto de titulación.

Debido a que es un software de código abierto y a sus constantes actualizaciones se hace uso de un Servidor Web Apache, a su vez Apache incorpora compatibilidad con una amplia gama de lenguajes de programación como Perl, PHP, etc.

Finalmente se escoge usar un servicio FTP debido a la facilidad de transportar archivos desde Windows donde se desarrolla el sistema prototipo hacia el servidor utilizado, en este caso un servidor Ubuntu 16.04 LST

2.2.3 INSTALACIÓN DE HERRAMIENTAS NECESARIAS

En esta sección se presenta la instalación de todas las herramientas necesarias que permiten el desarrollo del sistema prototipo, además se mostrará una descripción breve de los detalles más importantes de la instalación y de su respectiva configuración.

2.2.3.1 Creación de una instancia de Amazon EC2 en AWS

Antes de crear una instancia se debe crear una cuenta AWS, la misma que permite una utilización de forma gratuita de ciertos servicios de Amazon Web Services por el periodo de un año. Como requisitos se debe ingresar una dirección de correo electrónico, contraseña y nombre de la cuenta que se va a crear, como lo indica la Figura 2.35. cabe mencionar que en la mayoría de las ocasiones para crear esta cuenta se debe poner en

contacto directamente con personal de Amazon para que la activación de la misma la realicen de forma remota.



Figura 2.35 Crear cuenta AWS.

El servidor a implementarse usará los servicios en la nube de Amazon Web Services (AWS). Antes de la creación del servidor en AWS, como primer paso se debe escoger que tipo de instancia es la más conveniente para montar el servidor. Para el desarrollo del sistema prototipo se ha escogido una instancia Amazon EC2 [59] la cual es de nivel libre, lo que significa que durante un año características limitadas serán gratuitas.

Amazon EC2 (Amazon Elastic Compute Cloud) se refiere a capacidad informática en la nube con tamaño modificable.

La Tabla 2.15 muestra las características de Amazon EC2 en el rango de nivel libre, la cual es la instancia que se ha escogido para la realización del proyecto. Cabe mencionar que no se puede exceder de los límites ya que podría ingresar a un nivel de paga.

Tabla 2.15 Características de Amazon EC2 de nivel libre.

Característica	Capacidad
Almacenamiento	30 GB
Memoria RAM	1 GB
Número de CPU	1 núcleo
Disponibilidad	99.95%
Tiempo de cómputo mensual	750 horas

Se escogió la instancia mencionada debido a que la base de datos manejada para el proyecto de titulación no excede de 1Mb de capacidad por lo tanto las características mencionadas son suficientes para el uso óptimo del sistema prototipo, cabe recalcar que esta instancia es de nivel libre lo que significa que es sin costo.

Acto seguido se debe escoger el tipo de servidor más conveniente debido a las características de nivel libre que posee la instancia escogida. AWS ofrece instancias de servidor Windows y Linux, por lo que se hace una comparación rápida de los recursos para obtener ayuda al escoger el tipo de servidor a montar, específicamente se hace la comparación entre Windows Server y Ubuntu Server como lo muestra la Tabla 2.16.

Tabla 2.16 Comparación de recursos entre Windows Server y Ubuntu Server.

	Ubuntu Server	Windows Server
Tamaño de sistema de archivos	3.6 GB de disco duro de 30 GB disponibles	16 GB de disco duro de 30 GB disponibles
Memoria RAM utilizada	25% de memoria RAM de 1GB disponible	50% de memoria RAM de 1GB disponible
Interfaz	Línea de comandos	Gráfica

Como se observa en la Tabla 2.11 los recursos que utiliza Ubuntu Server en comparación con Windows Server son mucho menores y debido a la capacidad de almacenamiento escasa por ser de nivel libre, se ha decidido usar Ubuntu Server en su versión 16.04 LTS como Sistema Operativo del servidor a montar. Cabe mencionar que al no tener interfaz gráfica se hará uso de herramientas como Putty para acceder al servidor.

Una vez que se tiene claro la instancia y el tipo de servidor que se van a utilizar se procederá a la creación e instalación de las mismas.

En la consola de AWS se escoge la opción “*Compute*” y se marca EC2 que es la instancia que vamos a crear, escogida por las características listadas en la Tabla 2.10. la selección de EC2 está reflejada en la Figura 2.36.

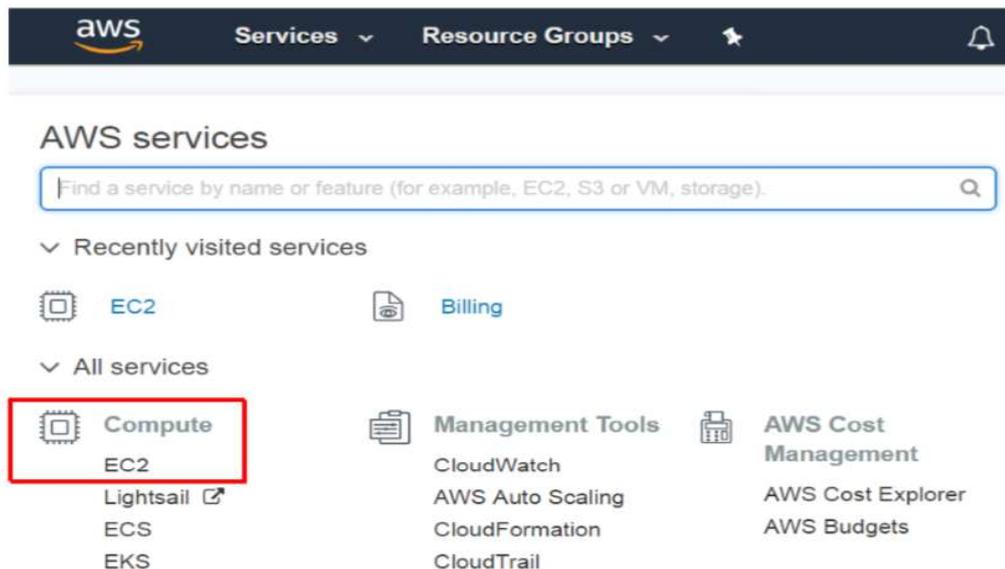


Figura 2.36 Selección de Amazon EC2.

Acto seguido se debe escoger la opción “lanza una nueva instancia”, opción que está dentro de la consola de administración como se observa en la Figura 2.37.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.



Note: Your instances will launch in the US East (N. Virginia) region

Figura 2.37 Creación de instancia AWS.

A continuación, se escoge el tipo de servidor que más se acople a los recursos de nivel libre, la selección del servidor se realizó por medio de la comparativa de la Tabla 2.11. Para este caso el tipo de servidor elegido es Ubuntu 16.04 en su versión LTS (Sin interfaz gráfica). La Figura 2.38 muestra la elección del tipo de servidor.

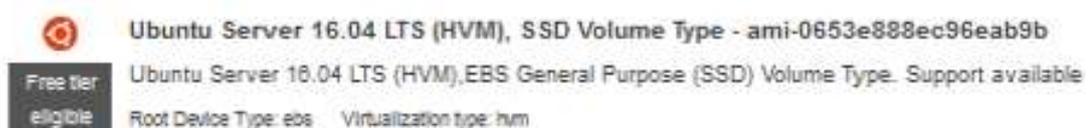


Figura 2.38 Elección de Servido Ubuntu 16.04 LTS.

La Figura 2.39 muestra los detalles de la instancia escogida como: valor de memoria RAM, ancho de banda, números de núcleos del CPU, tamaño de disco duro, etc.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes

Figura 2.39 Características de instancia EC2 de nivel libre.

Finalmente se realiza la configuración de las políticas de seguridad, las cuales regulan el tráfico entrante y saliente como se muestra en la Figura 2.40.

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0, :::0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0/0, :::0	e.g. SSH for Admin Desktop
MYSQL/Aur	TCP	3306	Custom CIDR, IP or Security Group	e.g. SSH for Admin Desktop

Figura 2.40 Opciones de seguridad para la instancia Ubuntu 16.04 LTS.

Se eligió las políticas de seguridad que se muestran ya que para la realización del sistema prototipo se necesita ciertos puertos habilitados.

Como se observa en la Figura 2.41, la instancia está creada y a su vez está activa y en funcionamiento.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-083b2bef8c175480f	t2.micro	us-east-2b	running	2/2 checks ...	None	ec2-18-216-129-59.us-e...

Figura 2.41 Instancia Ubuntu 16.04 LTS funcionando correctamente.

2.2.3.2 Instalación y codificación de MySQL

Para almacenar la información del sistema prototipo se utiliza la base de datos relacional MySQL y el software *phpMyAdmin* [60] como herramienta para administrar la base de datos, por medio de este software se creará la base de datos y las tablas relacionadas con el sistema prototipo.

En el sistema operativo “Ubuntu Server 16.04 LTS” se realiza la instalación de la base de datos MySQL. El Comando 2.1 muestra que en las líneas 1,2 y 3 se realiza primero la actualización respectiva de los repositorios y los paquetes a la versión actual, para después instalar el gestor de base de datos relacional MySQL como se observa en la línea 4. Cabe mencionar que para realizar esta instalación se debe iniciar colocando la contraseña del

archivo raíz (root), la cual es “ubuntu”. Finalmente, se continua con la configuración de MySQL, como lo indica la línea 5; esta línea final permite que solo usuarios de la base de datos puedan acceder a ella.

```
1 Sudo apt-get update
2 Sudo apt-get upgrade
3 Sudo apt-get dist upgrade
4 Sudo apt-get install mysql-server
5 sudo mysql_secure_installation
```

Comando 2.1 Instalación de MySQL.

2.2.3.3 Instalación y codificación del servidor Web Apache

El Comando 2.2 representa los comandos necesarios para instalar el servidor Apache, una vez actualizados todos los repositorios se instala el servidor web apache, con la instalación completa se habilita el puerto 80 y 443 como se muestra en las líneas 2 y 3.

```
1 sudo apt-get install apache2
2 sudo ufw app info "Apache Full"
3 sudo ufw allow in "Apache Full"
```

Comando 2.2 Instalación de Apache.

Para que el servidor Apache se active correctamente se debe reiniciar el servicio, para luego verificar que el servidor este activo, como se indica en el Comando 2.3

```
1 sudo systemctl restart apache2
2 sudo systemctl status apache2
```

Comando 2.3 Reinicio del servidor Apache.

2.2.3.4 Instalación y codificación del servidor FTP

El presente trabajo de titulación usará el protocolo FTP para la transferencia de archivos, esto debido a que el servidor “Ubuntu Server 16.04 LTS” instalado no tiene interfaz gráfica y los editores de texto “vim” y “nano” no son flexibles para redactar código. Por tal razón una vez que los archivos con código estén finalizados, se los sube directamente al servidor a través del protocolo FTP.

La instalación del servicio FTP se la realiza siguiendo la instrucción del Comando 2.4.

```
1 sudo apt-get install vsftpd
```

Comando 2.4 Instalación de servidor FTP.

Para la utilización de este protocolo se usará el software “Filezilla” [61] el cual es un cliente FTP, el mismo que podrá gestionar la conexión con el servidor para realizar transferencia de archivos desde el computador Windows hasta el servidor. Se necesita de la dirección IP del servidor para que el cliente FTP pueda conectarse.

La Figura 2.42 muestra el software Filezilla conectado con el servidor Ubuntu Server 16.04, visualizando los archivos que contiene el servidor.

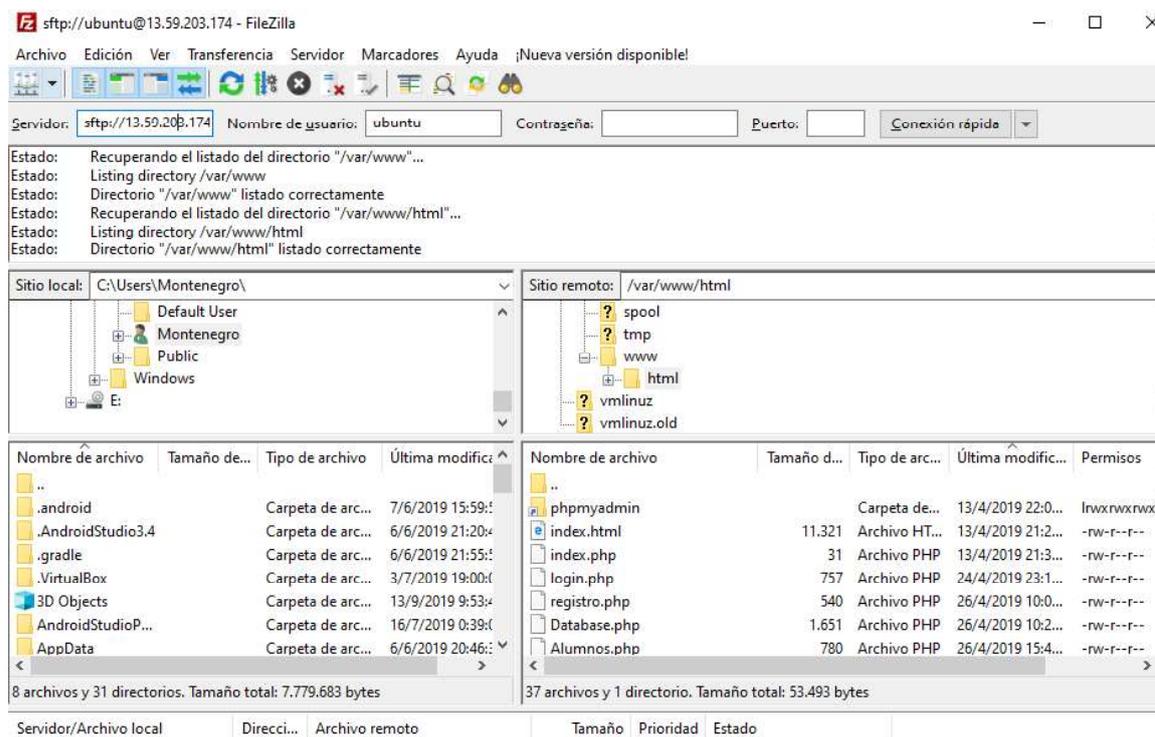


Figura 2.42 Software Filezilla.

2.2.3.5 Android Studio

Android Studio es la herramienta principal para el desarrollo del sistema prototipo. Para empezar, se debe descargar el software desde la página oficial [62] de Android Studio, la versión descargada e instalada es la 3.4 para Windows x64, La figura 2.43 muestra la página donde se descargó la herramienta mencionada.

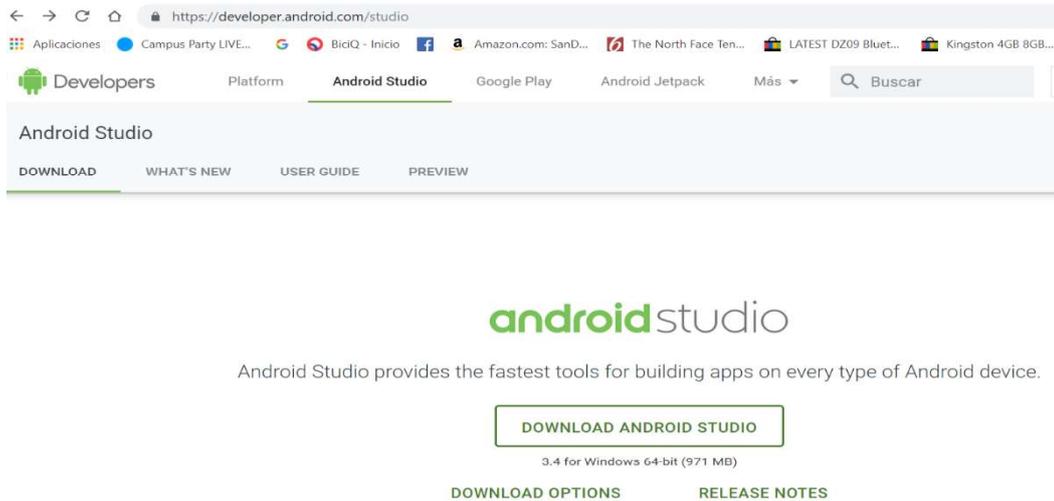


Figura 2.43 Descarga de Android Studio.

La Figura 2.44 muestra el proceso de instalación de la herramienta Android Studio después de haber descargado la misma, cabe indicar que la instalación se la debe realizar como administrador, seguir las instrucciones ofertadas por el software es de vital importancia para la correcta instalación de este. Adicional

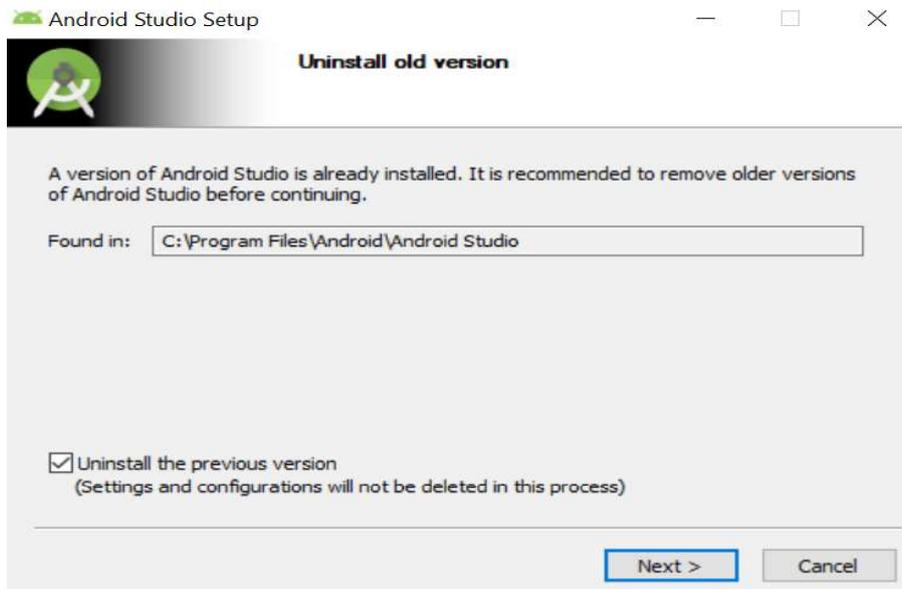


Figura 2.44 Instalación de Android Studio.

Finalmente, una vez que se haya instalado Android Studio de forma correcta, saldrá la pantalla inicial para crear un nuevo proyecto, como lo indica la Figura 2.45. Entonces la herramienta estará totalmente lista para iniciar la creación del sistema prototipo.

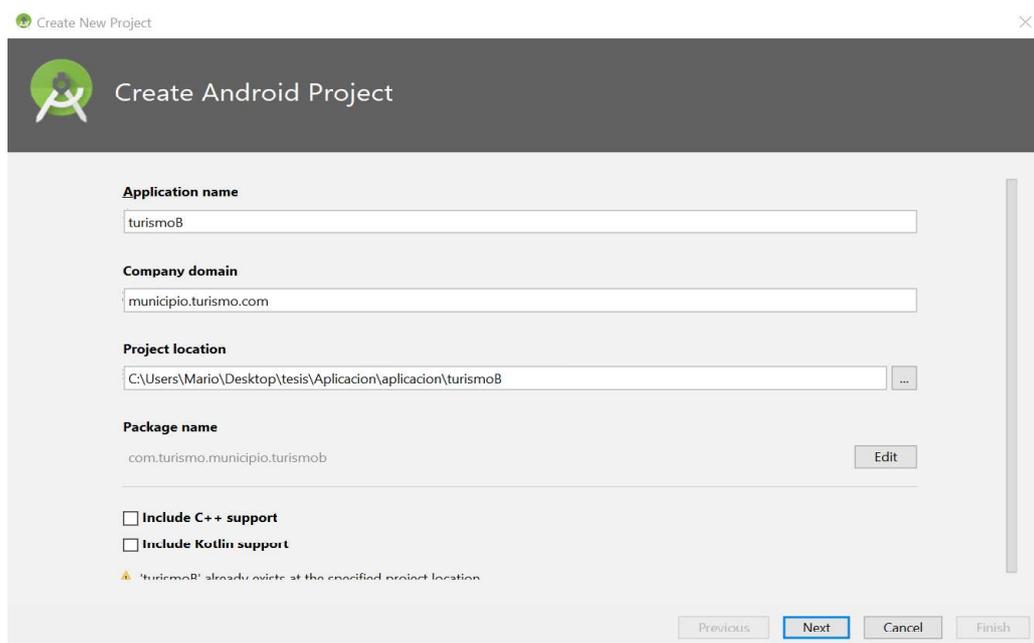


Figura 2.45 Creación de nuevo proyecto.

2.2.4 IMPLEMENTACIÓN DE LA CAPA DE BASE DE DATOS

En esta sección se detalla la codificación de la capa de datos, la misma que se compone por varias tablas. Para la creación de la base de datos se toma como referencia el diseño creado en el apartado 2.1.4.

Se utiliza la herramienta phpMyAdmin para administrar la base de datos, estableciendo una conexión con MySQL usando las credenciales creadas en la instalación de MySQL en el servidor.

El Código 2.5 muestra la codificación para crear la tabla “*Docente*” con sus respectivos atributos, siguiendo el modelo de base de datos relacional del apartado de Diseño, la tabla mencionada está conformada por los siguientes campos:

- *cédula_docente*: es el identificador único de cada usuario docente registrado, este atributo es la llave primaria para la tabla Docente.
- *nombre_docente*: nombre completo del usuario Docente que forma parte de la base de datos para el uso de la aplicación móvil.
- *relación_laboral_docente*: tipo de relación laboral del usuario Docente con la Escuela Politécnica Nacional.
- *cargo_docente*: cargo que ejerce el usuario Docente dentro de la institución.

- *dependencia_docente*: departamento al cual pertenece el usuario Docente dentro de la Escuela Politécnica Nacional.
- *estado_docente*: estado actual del usuario Docente dentro de la institución, puede ser ACTIVO o INACTIVO.
- *password_docente*: *segunda* credencial necesaria para que el usuario Docente pueda ingresar en el sistema prototipo.

La codificación de la tabla Docente es un ejemplo de las diversas tablas con la cuales se conforma la base de datos relacional del sistema prototipo.

```
CREATE TABLE `db_tesis`.`docente` (
  `cedula_docente` INT(10) NOT NULL ,
  `nombre_docente` VARCHAR(60) NOT NULL ,
  `relacion_laboral_docente` VARCHAR(60) NOT NULL ,
  `cargo_docente` VARCHAR(80) NOT NULL ,
  `dependencia_docente` VARCHAR(80) NOT NULL ,
  `estado_docente` VARCHAR(20) NOT NULL ,
  `password_docente` VARCHAR(20) NOT NULL ,
  PRIMARY KEY (`cedula_docente`)
) ENGINE = InnoDB;
```

Código 2.1 Creación de Tabla Docente.

El Código 2.6 indica el código para la creación de un usuario en la tabla Docente, cabe destacar que el ejemplo utilizado es información proporcionada por la Dirección de Gestión de la Información y Procesos (DGIP) por medio del pedido anteriormente expuesto en el ANEXO B.

```
INSERT INTO `docente`
(`cedula_docente`, `nombre_docente`, `relacion_laboral_docente`, `cargo_docente`,
`dependencia_docente`, `estado_docente`, `password_docente`)
VALUES ('1003682182', 'SÁNCHEZ ALMEIDA TARQUINO FABIÁN', 'NOMBRAMIENTO',
'PROFESOR AGREGADO A TIEMPO COMPLETO (NIVEL 1, GRADO 3)',
'DEPARTAMENTO DE AUTOMATIZACIÓN Y CONTROL INDUSTRIAL', 'ACTIVO', 'nopassword')
```

Código 2.2 Creación de usuario Docente.

Finalmente, para la verificar que los usuarios de la tabla Doctor se ingresaron correctamente y la base de datos MySQL funcione de forma satisfactoria se hace las respectivas consultas como lo muestra el ejemplo del Código 2.7.

```
SELECT * FROM docente
WHERE cedula_docente='1104999667' AND
password_docente='nopassword'
```

Código 2.3 Consulta de usuario Docente.

Obteniendo como resultado dentro de phpMyAdmin la siguiente información representada por la Figura 2.46.

	cedula_docente	nombre_docente	relacion_laboral_docente	cargo_docente	dependencia_docente	estado_docente	password_doce
Editar Copiar Borrar	1003682182	SANCHEZ ALMEIDA TARQUINO FABIAN	NOMBRAMIENTO	PROFESOR AGREGADO A TIEMPO COMPLETO (NIVEL 1, GRAD...	DEPARTAMENTO DE AUTOMATIZACION Y CONTROL INDUSTRIA...	ACTIVO	nopassword

Figura 2.46 Resultado de la consulta a la tabla Docente.

El código de las demás tablas utilizadas en el desarrollo del sistema prototipo se encuentran en el ANEXO C.

2.2.5 IMPLEMENTACIÓN DE LA CAPA DE NEGOCIO

En la sección presentada a continuación se muestra la implementación de la capa de negocio, esta sección se divide en 2 partes codificación de clases y codificación de servicios web.

2.2.5.1 Codificación de clases.

Para la codificación de las clases se seguirá el diseño de clases expuesto en el Capítulo 2 Apartado 2.1.5.2.

Se mostrará 2 ejemplos de codificación de clases, el primer ejemplo está representado en el Código 2.8, el cual muestra la implementación de la clase “*Docente*” con sus respectivos atributos, a su vez la clase cuenta con su respectivo constructor el cual permite inicializar el objeto; esta clase implementa los mismos campos que la tabla “*docente*” de la base de datos diseñada en el apartado Diseño de Capa de Datos.

```

1 package com.example.turnos.entidades;
2
3 public class Docente {
4     public int cedula_docente;
5     public String nombre_docente;
6     public String relacion_laboral_docente;
7     public String cargo_docente;
8     public String dependencia_docente;
9     public String estado_docente;
10    public String password_docente;
11
12    public Docente() {
13    }
14
15    public Docente(int cedula_docente, String nombre_docente, String relacion_laboral_docente, String cargo_docente,
16                  String dependencia_docente, String estado_docente, String password_docente) {
17        this.cedula_docente = cedula_docente;
18        this.nombre_docente = nombre_docente;
19        this.relacion_laboral_docente = relacion_laboral_docente;
20        this.cargo_docente = cargo_docente;
21        this.dependencia_docente = dependencia_docente;
22        this.estado_docente = estado_docente;
23        this.password_docente = password_docente;
24    }

```

Código 2.4 Código de la clase Docente.

El segundo ejemplo está representado en el Código 2.9, al igual que el ejemplo anterior se muestra la implementación de la clase “*doctor*” con sus respectivos atributos, además la clase cuenta con su respectivo constructor el cual permite inicializar el objeto doctor; esta clase implementa los mismos campos que la tabla “*doctor*” de la base de datos diseñada en el apartado Diseño de Capa de Datos. El código de las demás clases utilizadas en el desarrollo del sistema prototipo se encuentran en el ANEXO D

```

1 package com.example.turnos.entidades;
2
3 public class Doctor {
4
5     int cedula_doctor;
6     String nombre_doctor;
7     String especialidad_doctor;
8     String password_doctor;
9
10    public Doctor() {
11    }
12
13    public Doctor(int cedula_doctor, String nombre_doctor, String especialidad_doctor, String password_doctor) {
14        this.cedula_doctor = cedula_doctor;
15        this.nombre_doctor = nombre_doctor;
16        this.especialidad_doctor = especialidad_doctor;
17        this.password_doctor=password_doctor;
18    }
19
20 }

```

Código 2.5 Código de la clase Doctor.

2.2.5.2 Codificación de Servicios Web

Para la implementación del código de servicios web se usará Notepad++ [62] en su versión 7.6.6 la misma que permite la creación de archivos PHP.

Los servicios web entran en vigor cuando un cliente por medio del sistema prototipo necesite hacer una acción la cual conlleve a hacer consultas a la base de datos. Existe una secuencia para que un servicio web pueda funcionar de forma correcta, primero se necesita hacer la conexión a la base de datos; como segundo paso el servicio web debe recibir información con la cual quiera hacer la respectiva consulta, finalmente en caso de que se haga la consulta devuelve la información en forma de mensajes en formato JSON para así realizar la funcionalidad requerida. Un mensaje de tipo JSON se lo representa como lo indica el Código 2.10.

```
["doctor":[{"0":"1111111111","cedula_doctor":"1111111111",
            "1":"LUIS MALAN","nombre_doctor":"LUIS MALAN",
            "2":"MEDICINA GENERAL","nombre_especialidad":"MEDICINA GENERAL",
            "3":"nopassword","password_doctor":"nopassword"}]]
```

Código 2.6 Ejemplo de mensaje JSON.

El Código 2.11 muestra un servicio web codificado en lenguaje PHP, donde se observa cómo se guardan los parámetros necesarios para realizar la respectiva conexión con la base de datos, las credenciales necesarias para la conexión son: el enlace o dirección ip, la misma que se creó al momento de montar nuestro servidor en AWS; el nombre de la base de datos a la que nos queremos conectar; el nombre de usuario y la respectiva contraseña, estas credenciales están codificadas en las líneas de 2 a 5.

Una vez realizada la conexión como lo indica la línea 13, se hará la respectiva consulta que cumplirá con la funcionalidad que el cliente necesita, así lo indican las líneas 15 y 16.

Específicamente para el caso del ejemplo, el servicio web recibe desde el aplicativo móvil dos parámetros para realizar las consultas, líneas de 8 a 11, una vez realizada la consulta devuelve una respuesta como mensaje JSON como indican las líneas 18 a 24.

El Código de los servicios Web utilizados en el desarrollo del sistema prototipo se muestran en el ANEXO E.

```

1 <?php
2 $hostname_localhost = "dbtesis.cqgmp22bptwa.us-east-2.rds.amazonaws.com";
3 $database_localhost = "db_tesis";
4 $username_localhost = "dbtesis";
5 $password_localhost = "12345678";
6 $json=array();
7
8 if(isset($_GET["cedula_docente"]) && isset($_GET["password_docente"]))
9 {
10     $cedula_docente=$_GET["cedula_docente"];
11     $password_docente=$_GET["password_docente"];
12
13     $conexion = mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_localhost);
14     $consulta="select cedula_docente, nombre_docente, relacion_laboral_docente, cargo_docente, dependencia_docente,
15 estado_docente, password_docente from docente where cedula_docente= '{$_GET['cedula_docente']}' AND password_docente='{$_GET['password_docente']}'";
16     $resultado=mysqli_query($conexion,$consulta);
17
18     if($resultado){
19         if($registro=mysqli_fetch_array($resultado))
20         {
21             $json['docente'][]=$registro;
22         }
23         mysqli_close($conexion);
24         echo json_encode($json);
25     }
26     else
27     {
28
29         $resultar["cedula_docente"]=0;
30         $resultar["nombre_docente"]="no registra";
31         $resultar["relacion_laboral_docente"]="no registra";
32         $resultar["cargo_docente"]="no registra";
33         $resultar["dependencia_docente"]="no registra";
34         $resultar["estado_docente"]="no registra";
35         $resultar["password_docente"]="no registra";
36         $json['docente'][]=$resultar;
37         echo json_encode($json);
38     }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }

```

Código 2.7 Código de un Servicio Web.

2.2.6 CODIFICACIÓN DE LA CAPA DE PRESENTACIÓN

En la presente sección se detalla la implementación de la Capa de Presentación tomando como base el diseño desarrollado en el Apartado 2.1.6. Android Studio crea por defecto múltiples carpetas que son necesarias para la realización del sistema prototipo, entre ellas tenemos una de las más importantes llamada *layout*, en la cual se almacenan todas las interfaces gráficas que contiene el sistema prototipo, estas interfaces gráficas están desarrolladas en lenguaje XML.

La Figura 2.47 muestra todas las interfaces gráficas que contiene el sistema prototipo. Se mostrará un ejemplo de código para la codificación de las interfaces gráficas, esto debido a que el código completo estará en el ANEXO F.

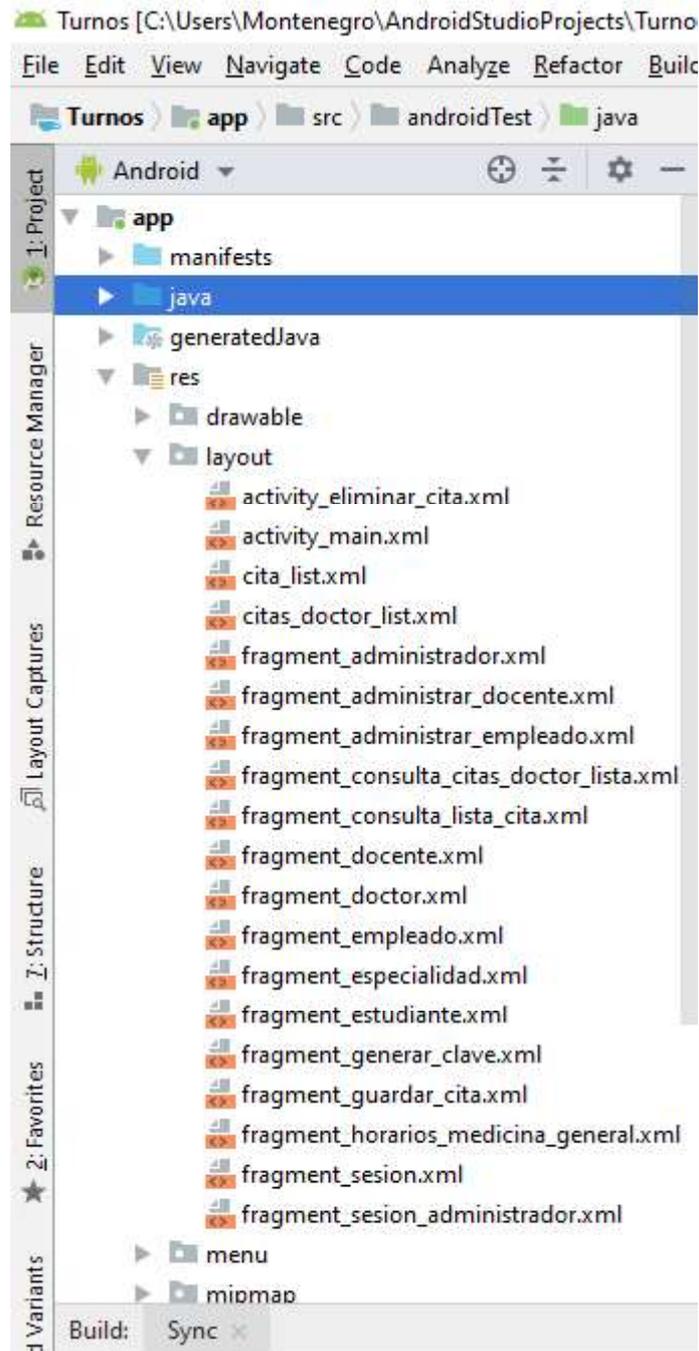


Figura 2.47 Interfaces gráficas del sistema prototipo.

2.2.6.1 Codificación de vista del aplicativo móvil.

En la Figura 2.48 se observa la interfaz gráfica de inicio, donde un usuario puede autenticarse para hacer uso de las funcionalidades del sistema prototipo. debe escoger entre las siguientes opciones en el Spinner del inicio: Estudiante, Docente, Empleado o Doctor.

Además, el Código 2.8 es un ejemplo de codificación para la interfaz gráfica de autenticación que se observa en la Figura 2.48. Como se observa tanto en el código como en la interfaz gráfica se está utilizando componentes como EditText, TextView, Button, y Spinner que es el combo donde se carga el tipo de usuario que va a usar la aplicación.

La línea número 11 muestra que el contenedor de toda la interfaz gráfica es un LinearLayout, el cual tiene una orientación vertical, cierta separación con el margen superior y una distribución completa tanto de ancho y largo, como se puede observar en las líneas 12 a 15.

Dentro de este contenedor se encuentran los diferentes componentes nombrados anteriormente los mismo que poseen características particulares como es el caso del TextView el mismo que tiene sus características representadas en las líneas 18 a 24.

El código de todas las interfaces gráficas codificadas en lenguaje XML utilizadas en el desarrollo del sistema prototipo se muestran en el ANEXO F.

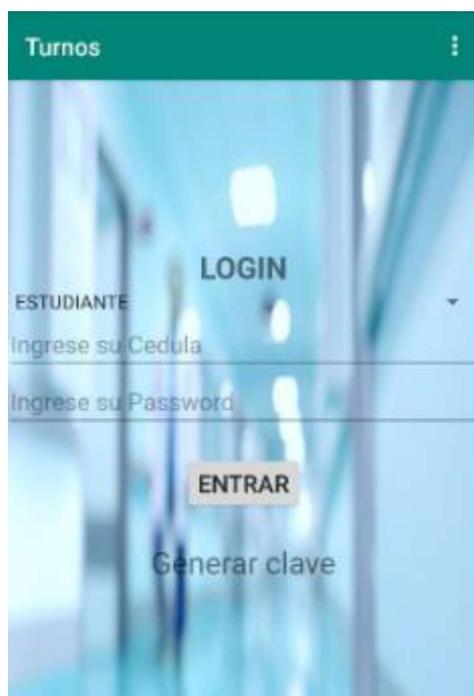


Figura 2.48 Interfaz de inicio.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:background="@drawable/logo2"
7
8   tools:context=".fragment.SesionFragment"
9   android:orientation="vertical">
10
11   <LinearLayout
12     android:orientation="vertical"
13     android:layout_marginTop="120dp"
14     android:layout_width="match_parent"
15     android:layout_height="wrap_content">
16
17     <TextView
18       android:layout_marginTop="20dp"
19       android:textSize="25dp"
20       android:layout_gravity="center"
21       android:text="LOGIN"
22       android:textStyle = "bold"
23       android:layout_width="wrap_content"
24       android:layout_height="wrap_content" />
25
26     <Spinner
27       android:scrollbarSize="40dp"
28       android:id="@+id/id_spinner"
29       android:layout_gravity="center"
30       android:layout_width="match_parent"
31       android:layout_height="wrap_content">
32
33     </Spinner>
34
35
36
37     <EditText
38       android:textSize="20dp"
39       android:id="@+id/txt_cedula_login"
40       android:hint="Ingreso su Cedula"
41       android:layout_width="match_parent"
42       android:layout_height="wrap_content" />
43
44     <EditText
45       android:textSize="20dp"
46       android:inputType="textPassword"
47       android:id="@+id/txt_password_login"
48       android:hint="Ingreso su Password"
49       android:layout_width="match_parent"
50       android:layout_height="wrap_content" />
51
52   </LinearLayout>

```

Código 2.8 Código XML de interfaz gráfica.

La Figura 2.49 muestra la interfaz gráfica principal, una vez que la autenticación como usuario Paciente o Doctor es correcta, en la imagen se puede visualizar componentes llamado CardView los cuales permiten escoger si desea agendar una nueva cita o si quiere revisar citas agendadas previamente en el caso de Pacientes, o la opción de revisar citas médicas a tratar en el caso de Doctor respectivamente.

Por su parte el Código 2.9 representa la codificación de la interfaz gráfica principal para el usuario Paciente de la figura en la Figura 2.49. Como se observa tanto en el código como en la interfaz gráfica se está utilizando componentes como CardView, TextView e ImageView.

Como se puede observar en la línea 56 el contenedor de elementos LinearLayout tiene una orientación horizontal ya que es aquí donde se colocan los CardViews Agendar Cita y Citas Vigentes, estos elementos tienen características propias tanto de dimensiones como en funcionalidad como se observa en las líneas 66 a 70, también se habilita la opción onClick en los CardView para que simule la función de un botón al presionar sobre este elemento.

Se introduce un nuevo elemento llamado ImageView el mismo que permite mostrar una imagen dentro de la tarjeta CardView, de esta manera uniendo los dos elementos se puede obtener un botón con una imagen.

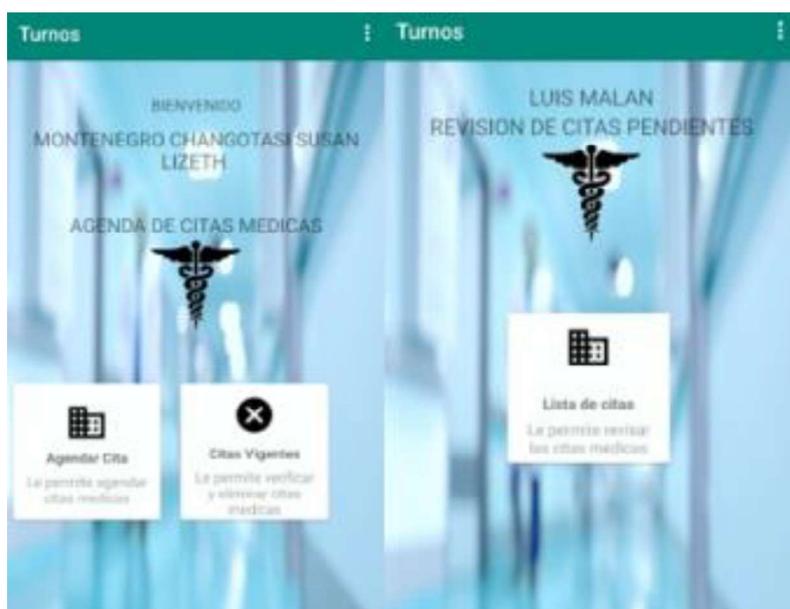


Figura 2.49 Interfaz Principal para Paciente y Doctor.

```

56 <LinearLayout
57
58     android:orientation = "horizontal"
59     android:layout_width = "match_parent"
60     android:layout_height = "wrap_content"
61     android:layout_marginTop="40dp">
62
63
64 <!--AGENDAR CITA -->
65 <androidx.cardview.widget.CardView
66     android:clickable = "true"
67     android:id="@+id/cv_agendar_cita_docente"
68     android:layout_width = "150dp"
69     android:layout_height = "140dp"
70     android:layout_margin = "10dp">
71     <LinearLayout
72         android:layout_width = "match_parent"
73         android:layout_height = "match_parent"
74         android:orientation = "vertical"
75         android:background="@drawable/color"
76         android:gravity="center">
77         <ImageView
78             android:layout_width="64dp"
79             android:layout_height="64dp"
80             android:padding = "10dp"
81             android:src="@drawable/ic_business_black_24dp" />
82         <TextView
83             android:layout_width = "wrap_content"
84             android:layout_height = "wrap_content"
85             android:textStyle = "bold"
86             android:textStyle = "bold"
87             android:layout_marginTop = "-5dp"
88             android:text = "Agendar Cita" />
89
90         <TextView
91             android:layout_width = "wrap_content"
92             android:layout_height = "wrap_content"
93             android:gravity="center"
94             android:text = "Le permite agendar citas medicas"
95             android:padding = "5dp"
96             android:textColor = "@android:color/darker_gray" />
97     </LinearLayout>
98
99 </androidx.cardview.widget.CardView>

```

Código 2.9 Código XML de interfaz Paciente.

La Figura 2.50 muestra la interfaz gráfica donde un paciente puede elegir el tipo de especialidad en el que desea agendar su cita, las especialidades se muestran en elementos CardView.



Figura 2.50 Interfaz Especialidades.

En la Figura 2.51 se observa el proceso de elección de fecha y hora para la reserva de un turno médico, una vez que las características de la cita médica sean satisfactorias para el Paciente se procede a agendar la cita médica.



Figura 2.51 Interfaz Agendar cita médica.

la Figura 2.52 muestra como un Paciente puede revisar las diversas citas que agendó dependiendo de sus necesidades, con la opción de anular las mismas en el caso que no esté conforme con alguna de ellas.



Figura 2.52 Interfaz listar y anular citas médicas.

Cuando una persona con permisos de Administrador ingresa a la aplicación tiene las opciones de administrar usuarios, para que dependiendo a qué tipo de usuario quiera administrar como lo indica la Figura 2.53.

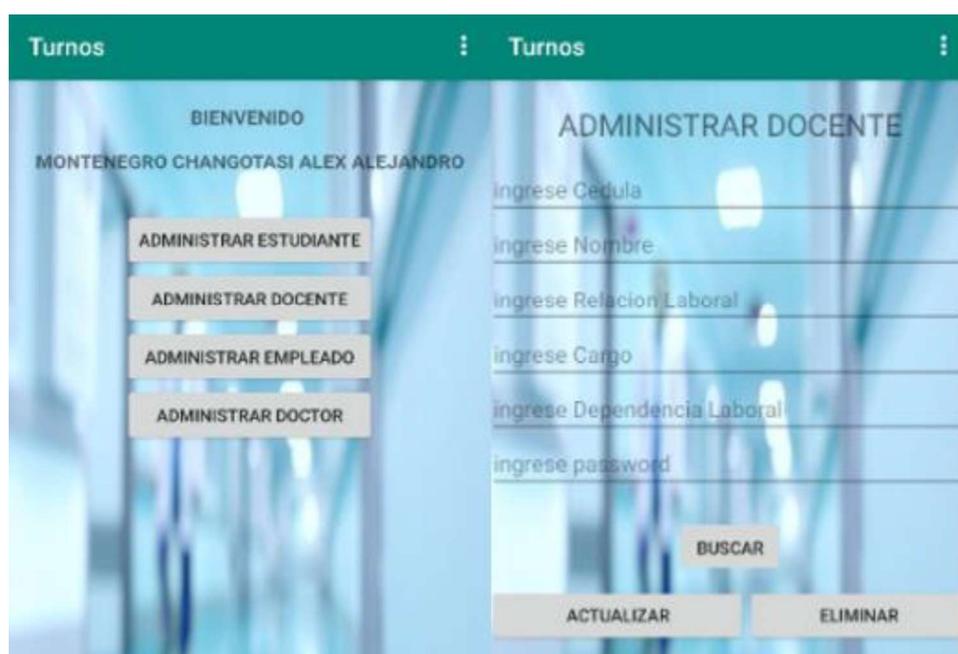


Figura 2.53 Interfaz administrar usuarios.

3. RESULTADOS Y DISCUSIÓN

En el presente capítulo se puede observar las pruebas de funcionamiento realizadas sobre el prototipo implementado. Primero se inicia con la actualización del tablero Kanban para esta fase, colocando las tareas en el lugar correspondiente para esta fase final. A continuación, se validarán tanto los requerimientos funcionales como los requerimientos no funcionales, tal y como se definió en el Capítulo 2 Apartado 2.1.2. Finalmente se analizarán los resultados obtenidos.

3.1 TABLERO DE ACTIVIDADES KANBAN

La Figura 3.1 muestra el tablero de actividades Kanban actualizado para el presente capítulo, donde se observa que ya no existen tareas por realizar; las Tareas en Proceso están ligadas a las pruebas de funcionamiento de cada uno de los módulos planteados; las Tareas Realizadas indican que tanto la fase de diseño como la fase de implementación han sido terminadas satisfactoriamente.

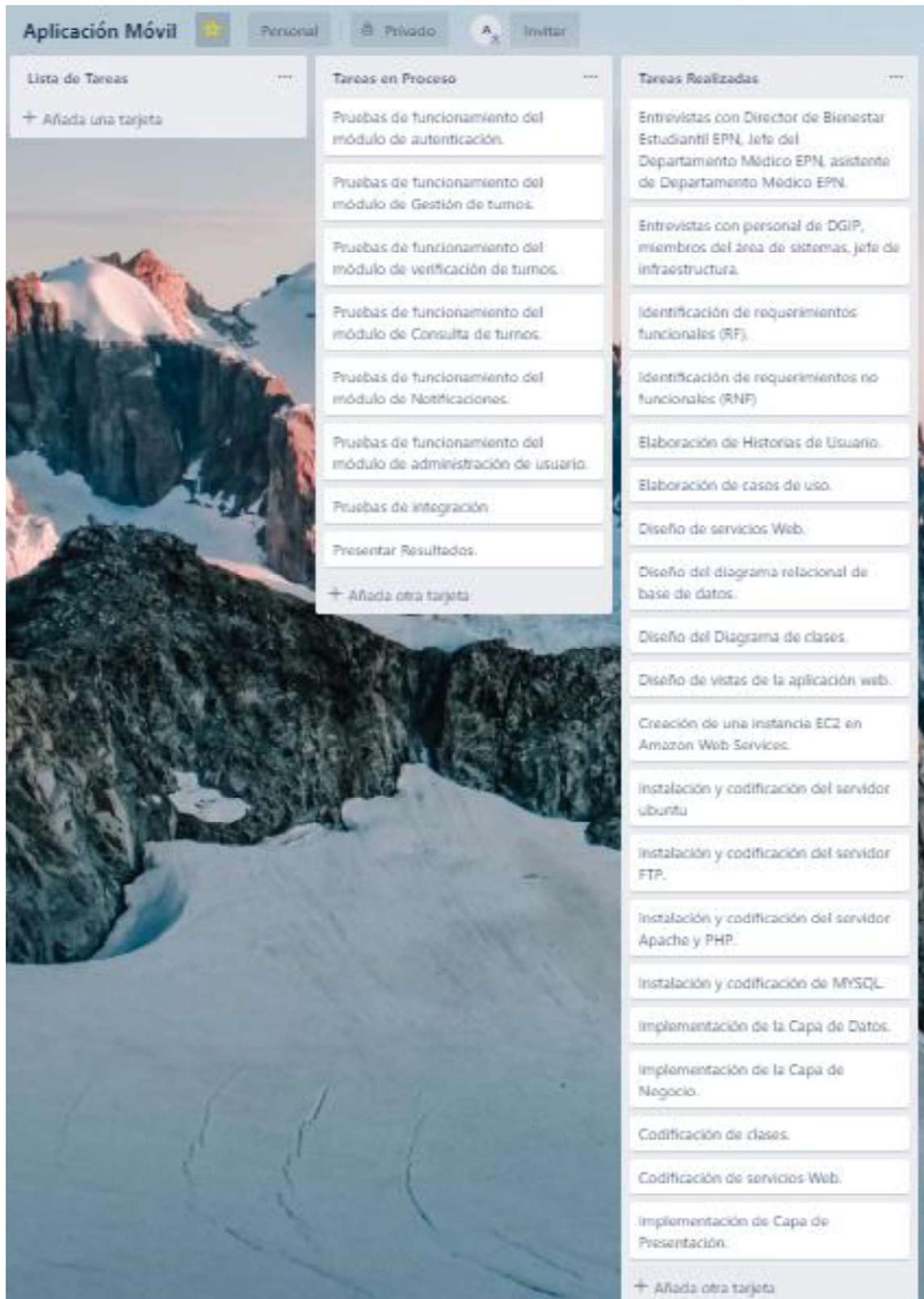


Figura 3.1 Tablero Kanban para la fase de Resultados y Discusión.

3.2 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES.

En este punto se valida la funcionalidad de cada uno de los requerimientos funcionales agrupados en los distintos módulos descritos en el Capítulo 2 Apartado 2.1.3. Para iniciar se detallan los requerimientos funcionales que el sistema prototipo debe cumplir; finalmente se realiza la validación de cada uno de estos requerimientos.

3.2.1 PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO DE AUTENTICACIÓN.

Este módulo permite autenticar a los usuarios de la aplicación móvil, escogiendo entre cuatro tipos de usuario: Estudiante, Docente, Empleado, Doctor. Los requerimientos funcionales que se deben cumplir en este módulo se encuentran en la Tabla 3.1.

Tabla 3.1 Requerimientos Funcionales del Módulo de Autenticación.

Requerimiento Funcional	Descripción.
RF001	La aplicación móvil permite escoger el tipo de usuario que va a ser uso de la misma.
RF002	El usuario al ingresar por primera vez a al sistema prototipo, generará una clave de usuario.
RF003	El usuario médico puede acceder al sistema prototipo haciendo uso de las respectivas credenciales.
RF005	El usuario paciente puede acceder al sistema prototipo haciendo uso de las respectivas credenciales.

En la Figura 3.2 se observa el cumplimiento del Requerimiento Funcional 001 (RF001), donde presionando en el Spinner, se despliega una lista de opciones las cuales incluyen: ESTUDIANTE, DOCENTE, EMPLEADO o DOCTOR.

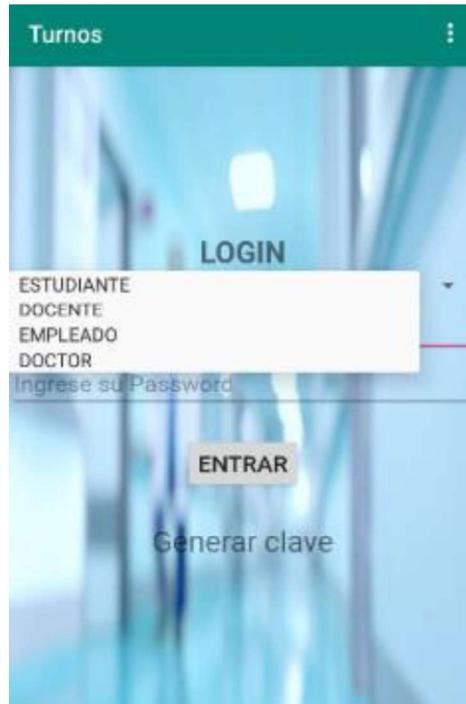


Figura 3.2 Cumplimiento del Requerimiento Funcional RF001.

En la Figura 3.3 se observa el cumplimiento del Requerimiento Funcional 002 (RF002), donde después de que el usuario presione en el TextView "Generar clave", llevará a una interfaz en la cual el usuario puede generar una nueva clave para poder logearse correctamente.

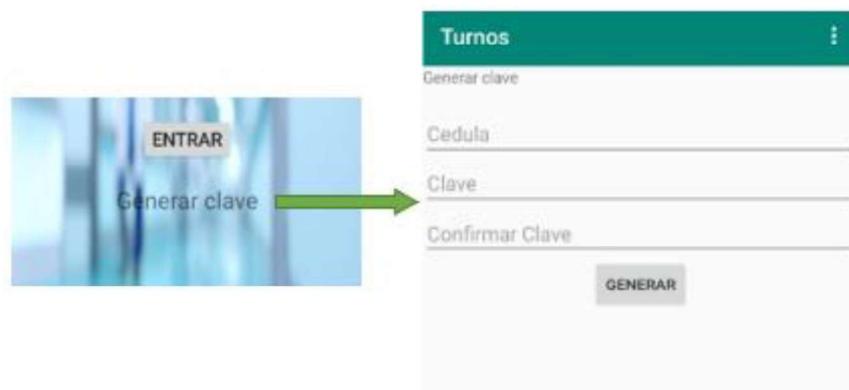


Figura 3.3 Cumplimiento del Requerimiento Funcional RF002.

En la Figura 3.4 se observa el cumplimiento del Requerimiento Funcional 003 (RF003), donde el usuario Médico, con el uso de las credenciales correctas, puede ingresar al sistema prototipo.

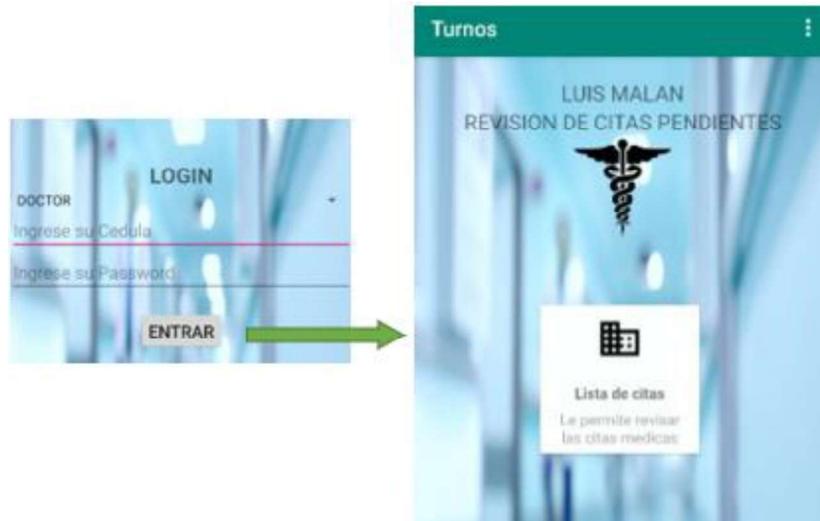


Figura 3.4 Cumplimiento del Requerimiento Funcional RF003.

En la Figura 3.5 se observa el cumplimiento del Requerimiento Funcional 005 (RF005), donde el usuario Paciente, con el uso de las credenciales correctas, puede ingresar al sistema prototipo.

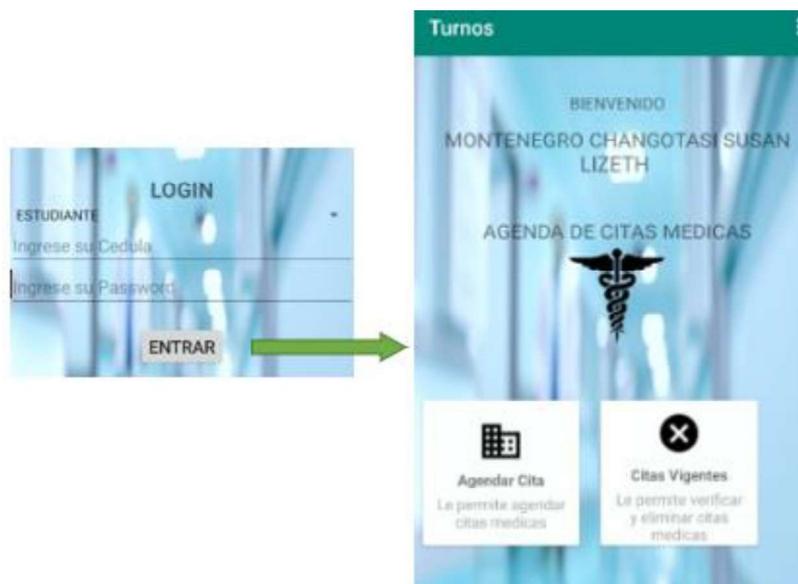


Figura 3.5 Cumplimiento del Requerimiento Funcional RF005.

3.2.2 PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO DE VERIFICACIÓN DE TURNOS.

Este módulo permite exclusivamente a los doctores verificar una lista con las diferentes citas médicas que estos deben atender. El requerimiento funcional que se deben cumplir en este módulo se encuentra en la Tabla 3.2.

Tabla 3.2 Requerimientos Funcionales del Módulo de Verificación de Turnos.

Requerimiento Funcional	Descripción.
RF004	El usuario Médico tiene la opción de revisar la información de las citas médicas que este debe tratar.

En la Figura 3.6 se observa el cumplimiento del Requerimiento Funcional 004 (RF004), donde el usuario Médico puede presionar sobre la tarjeta “*Lista de Citas*”, la cual llevará a una interfaz que contiene una lista con las diferentes citas médicas que el doctor en cuestión debe atender.



Figura 3.6 Cumplimiento del Requerimiento Funcional RF004.

3.2.3 PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO GESTIÓN DE TURNOS.

Este módulo permite a los usuarios pacientes escoger una especialidad médica, hora y fecha para reservar un turno médico, dependiendo de sus necesidades. Si el usuario lo considera necesario tiene la opción de consultar la información del Doctor de la especialidad escogida; la obtención de los turnos médicos dependerá de la disponibilidad de horarios. Los requerimientos funcionales que se deben cumplir en este módulo se encuentran en la Tabla 3.3.

Tabla 3.3 Requerimientos Funcionales del Módulo de Gestión de Turnos.

Requerimiento Funcional	Descripción.
RF006	El paciente puede escoger entre las 5 especialidades ofertadas para la obtención de un turno médico.
RF007	El paciente, si lo considera necesario puede revisar la información del Doctor al cual pertenece la especialidad escogida.
RF008	Dependiendo de las necesidades del paciente, este puede escoger una fecha para su cita médica.
RF009	Dependiendo de las necesidades del paciente, este puede escoger un horario para su cita médica.
RF010	La disponibilidad de los horarios, se valida una vez que el paciente haya escogido un horario y fecha para su cita médica.
RF011	En caso de que haya disponibilidad de horario en la especialidad que el paciente escogió el paciente puede reservar un turno médico.

La Figura 3.7 muestra el cumplimiento del Requerimiento Funcional 006 (RF006), el cual indica que una vez que el usuario paciente presiona sobre la tarjeta “Agendar cita”, inmediatamente cambia de interfaz gráfica, en donde se muestran 5 tarjetas con los respectivos nombres de las especialidades que el departamento médico ofrece a los usuarios, las especialidades son las siguientes; Medicina General, Odontología, Ginecología, Psicología y Nutrición.



Figura 3.7 Cumplimiento del Requerimiento Funcional RF006.

Se observa en la Figura 3.8 el cumplimiento del Requerimiento Funcional 007 (RF007), este requerimiento hace referencia a los pacientes, donde una vez escogida la especialidad que estos necesitan para la obtención de un turno médico, los pacientes pueden visualizar la información del médico al cual corresponde dicha área.



Figura 3.8 Cumplimiento del Requerimiento Funcional RF007.

La Figura 3.9 muestra el cumplimiento del Requerimiento Funcional 008 (RF008), esta tarea la realiza el paciente una vez que escogió la especialidad que este necesita, debe escoger la fecha que más se acople a sus necesidades, cabe mencionar que el calendario para escoger la fecha tiene un despliegue de 30 días a partir de la fecha actual, a su vez los fines de semana no podrá escoger ya que son días no laborables para el departamento médico.

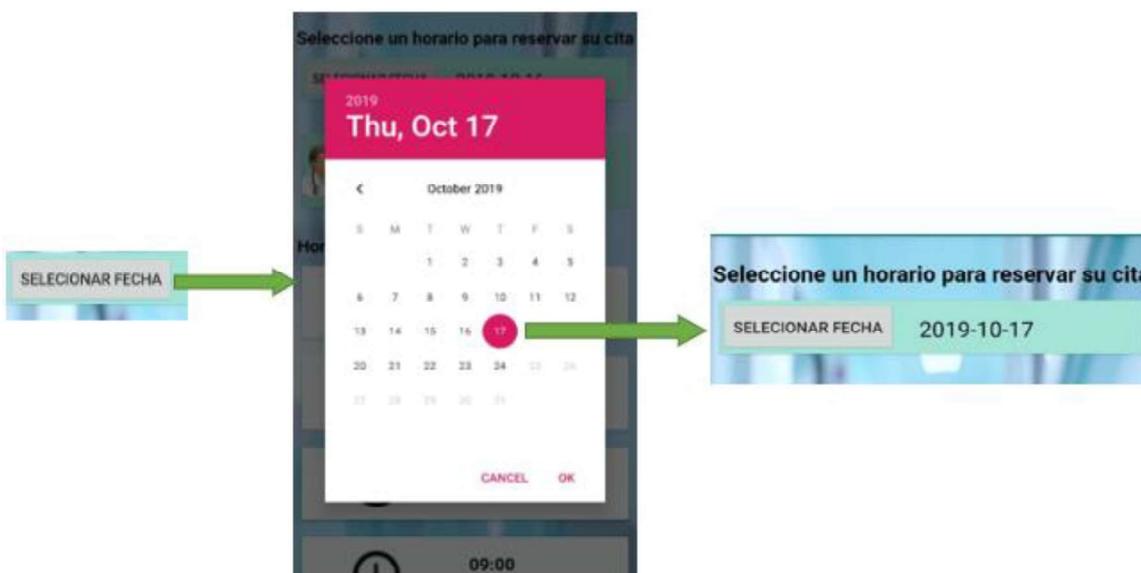


Figura 3.9 Cumplimiento del Requerimiento Funcional RF008.

Se observa en la Figura 3.10 el cumplimiento del Requerimiento Funcional 009 (RF009), este requerimiento lo cumple el paciente ya que una vez que escogió la fecha que más se

acomode a sus necesidades debe escoger un horario en el cual va a reservar un turno médico. Los horarios están previamente definidos en tarjetas dentro de la interfaz gráfica, iniciando a las 8:00 am con la primera consulta y finalizando 16:40 con la consulta final, cada consulta médica tiene una duración de 20 minutos.



Figura 3.10 Cumplimiento del Requerimiento Funcional RF009.

Una vez que el paciente escoge la fecha y la hora para reservar un turno médico, el sistema prototipo valida si hay disponibilidad de horario para reservar dicho turno médico. En caso de que un horario ya esté ocupado por otro usuario, el sistema prototipo advertirá al paciente con un mensaje el cual le informa al paciente que no existe disponibilidad en el horario escogido, y que debe escoger otro horario para poder reservar el turno requerido; la Figura 3.11 refleja el cumplimiento del Requerimiento Funcional 010 (RF010),

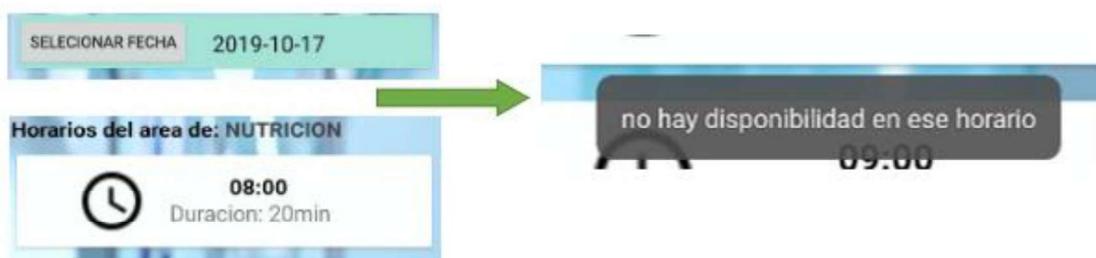


Figura 3.11 Cumplimiento del Requerimiento Funcional RF010.

La Figura 3.12 muestra el cumplimiento del Requerimiento Funcional 0011 (RF011), esta tarea la realiza el usuario paciente, donde en caso de que el horario y fecha escogidos para una determinada especialidad sean validados correctamente por el sistema prototipo, se desplegará una interfaz gráfica presentando un resumen con la información escogida por el usuario paciente, y finalmente si el paciente está de acuerdo con la información que se despliega en la interfaz, puede proceder a agendar y reservar el turno médico.

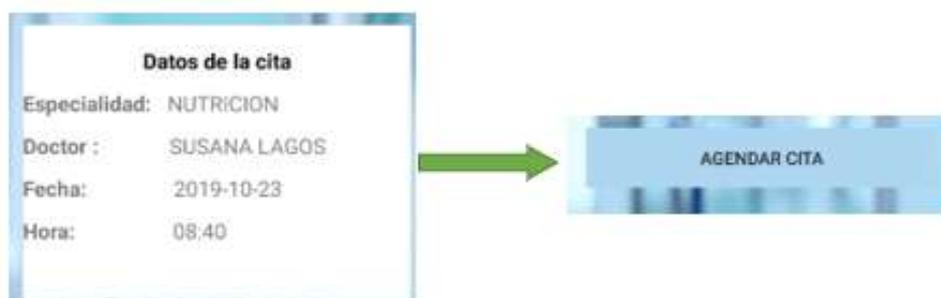


Figura 3.12 Cumplimiento del Requerimiento Funcional RF011.

3.2.4 PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO CONSULTA DE TURNOS.

Este módulo permite a los usuarios pacientes ingresar a un apartado donde pueden visualizar las diferentes citas médicas que estos agendaron, seleccionando la tarjeta “Citas Vigentes”, cada cita médica está detallada con la información necesaria para que el paciente decida si sigue adelante con la misma o en caso contrario opte por la opción de anular la cita médica pendiente, seleccionando en la imagen que representa un basurero. Los requerimientos funcionales que se deben cumplir en este módulo se encuentran en la Tabla 3.4.

Tabla 3.4 Requerimientos Funcionales del Módulo Consulta de Turnos.

Requerimiento Funcional	Descripción.
RF012	El paciente puede visualizar todas las citas médicas que haya reservado, en las diferentes especialidades ofertadas.
RF013	El paciente para recordar las citas médicas vigentes puede desplegar un resumen de cada una de ellas, presentando la hora, fecha, especialidad y doctor que atenderá al usuario.
RF014	En caso de que el usuario no esté de acuerdo con alguna de las citas médicas que este reservó puede anular cualquiera de ellas.

La Figura 3.13 muestra el cumplimiento del Requerimiento Funcional 012 (RF012), este requerimiento lo realiza el paciente donde una vez seleccionada el ítem “Citas Vigentes”, se despliega una lista con todas las citas médicas que el usuario tiene reservado una cita. Estas citas médicas tienen todos los datos importantes de la misma como: especialidad, nombre del doctor, fecha de la cita y hora de la cita.



Figura 3.13 Cumplimiento del Requerimiento Funcional RF012.

Se observa en la Figura 3.14 el cumplimiento del Requerimiento Funcional 013 (RF013), este requerimiento lo cumple el paciente, ya que una vez que tiene desplegada la lista de las citas médicas que reservó se puede obtener un resumen de cualquiera de las citas que este seleccione, en el resumen se presenta de manera ordenada los datos de la reserva y un anuncio debajo de este resumen, advirtiendo el lugar donde atenderá al paciente y con cuantos minutos de anticipación debe estar presente.



Figura 3.14 Cumplimiento del Requerimiento Funcional RF013.

En caso de que el usuario no esté conforme con la fecha, hora, o especialidad de la cita médica que este reservó, tiene la opción de eliminar la misma presionando sobre el icono con forma de basurero, esta figura representa que puede eliminar la cita médica escogida, una vez escogida la cita a eliminar deberá confirmar que desea realizar dicha acción, presionando sobre el botón "Anular Cita". Así lo demuestra la Figura 3.15 la misma que representa el cumplimiento del Requerimiento Funcional 014 (RF014).



Figura 3.15 Cumplimiento del Requerimiento Funcional RF014.

3.2.5 PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO DE NOTIFICACIONES.

Este módulo es exclusivo del usuario paciente, el mismo permite recibir un recordatorio en forma de notificación con todos los datos de la cita médica que este reservó previamente. Los requerimientos funcionales que se deben cumplir en este módulo se encuentran en la Tabla 3.5.

Tabla 3.5 Requerimientos Funcionales del Módulo de Notificaciones.

Requerimiento Funcional	Descripción.
RF015	Siempre y cuando el paciente haya agregado una cita médica de forma satisfactoria le llegará una notificación al dispositivo móvil recordándole los datos de la misma.

Se observa en la Figura 3.16 el cumplimiento del Requerimiento Funcional 015 (RF015), este requerimiento está diseñado para el usuario paciente, el mismo que recibirá una notificación en la barra de herramientas del dispositivo móvil recordándole que tiene una cita médica, este recordatorio contará con datos como hora, fecha, y especialidad de la cita reservada. El recordatorio recibido llegará al teléfono móvil con anticipación de un día para que el paciente tengo el tiempo necesario para coordinar la asistencia a la misma.



Figura 3.16 Cumplimiento del Requerimiento Funcional RF015.

3.2.6 PRUEBAS DE FUNCIONAMIENTO PARA EL MÓDULO ADMINISTRACIÓN DE USUARIO.

Para hacer uso de este módulo se debe tener credenciales de administrador, a su vez para ingresar al mismo se deberá presionar la pestaña “administrar” que se encuentra en la barra menú situada en la parte superior de la pantalla. El administrador tendrá las capacidades de buscar, eliminar y actualizar los diferentes usuarios que se encuentran guardados en la base de datos que la DGIP entregó.

Los requerimientos funcionales que se deben cumplir en este módulo se encuentran en la Tabla 3.6.

Tabla 3.6 Requerimientos Funcionales del Módulo Administración de Usuario.

Requerimiento Funcional	Descripción.
RF016	El usuario Administrador puede ingresar al sistema prototipo, sección administrar con credenciales especiales.
RF017	El administrador tiene la opción de escoger el tipo de usuarios que este quiere buscar, modificar o eliminar.
RF018	Si se quiere buscar un usuario en particular, el administrador lo puede hacer ingresando el número de cédula del usuario que busca.
RF019	Una vez que el administrador encuentre un usuario en particular puede modificar su información.
RF020	Una vez que el administrador encuentre un usuario en particular puede eliminarlo de la base de datos de la aplicación móvil.

La Figura 3.17 muestra el cumplimiento del Requerimiento Funcional 0016 (RF016), para poder administrar cualquier tipo de usuario, ya sea Docente, Estudiante, Empleado o Doctor, se debe ingresar al sistema prototipo con credenciales especiales, las mismas se las conoce como credenciales de administración. A su vez, para poder ingresar en la misma se debe presionar la pestaña “administrar”, la misma que se ubica en el menú de la parte superior del sistema prototipo.



Figura 3.17 Cumplimiento del Requerimiento Funcional RF016.

Se observa en la Figura 3.18 el cumplimiento del Requerimiento Funcional 017 (RF017), esta tarea la realiza el administrador una vez que ingresa al sistema prototipo con las credenciales correctas. En la interfaz el administrador tiene la opción de escoger entre 4 tipos de usuarios para administrarlos: Docente, Estudiante, Empleado o Doctor.

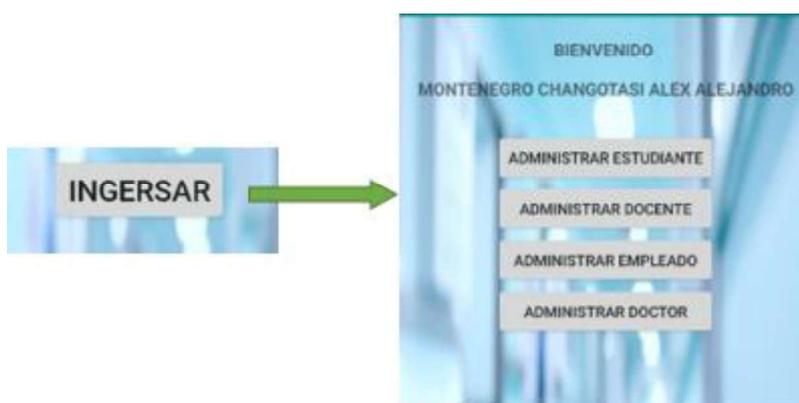


Figura 3.18 Cumplimiento del Requerimiento Funcional RF017.

Para buscar un usuario en particular, el administrador debe saber el número de cédula del mismo, una vez ingresado dicho valor se presiona en el botón buscar, donde inmediatamente se despliega toda la información del usuario buscado. Esto se observa en la Figura 3.19 presenta a continuación, la misma que muestra el cumplimiento del Requerimiento Funcional 018 (RF018).

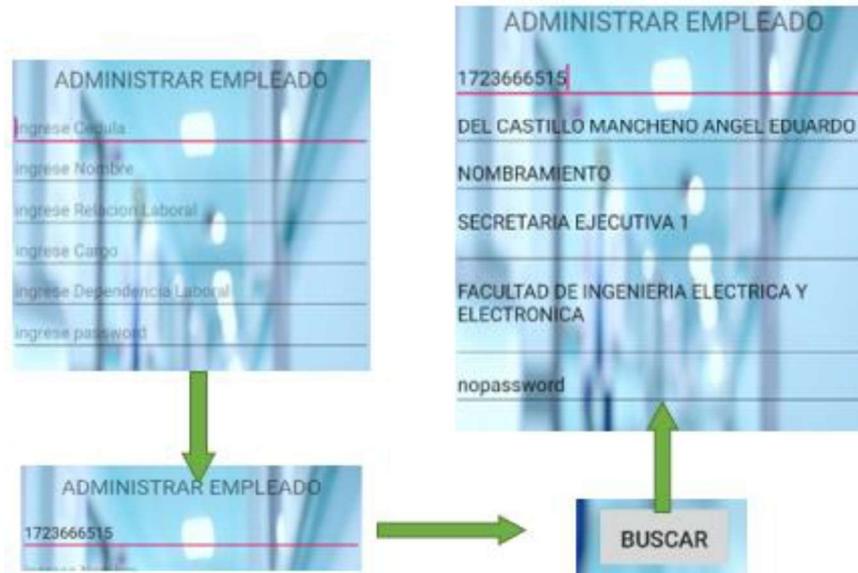


Figura 3.19 Cumplimiento del Requerimiento Funcional RF018.

En la Figura 3.20 se puede observar el cumplimiento de dos requerimientos funcionales: requerimiento RF019 y RF020. Los cuales están relacionados ya que para comprobar la funcionalidad de estos el usuario administrador debe encontrar un usuario en particular al cual quiera modificar su información o eliminar toda su información de la base de datos. Antes de realizar estos cambios el administrador debe estar seguro de que quiera modificar la información de un usuario, ya que una vez modificada o eliminada ya no podrá recuperar esa información.

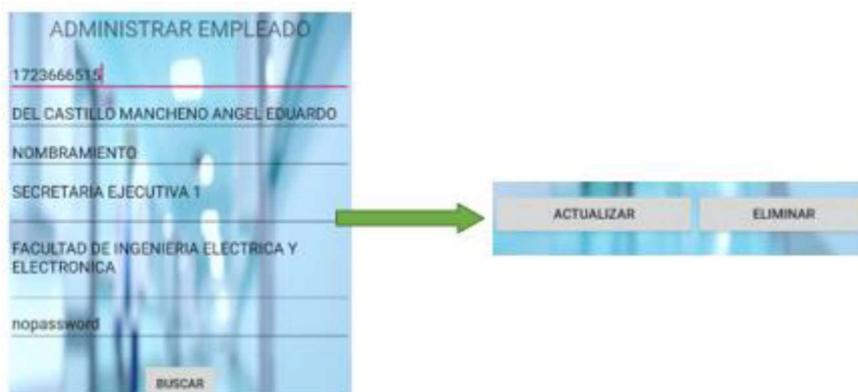


Figura 3.20 Cumplimiento de Requerimientos Funcionales RF019 y RF020.

Una vez que se ha finalizado la validación de todos los requerimientos funcionales, se observa que el sistema prototipo realiza todas las funciones esperadas y cumple con las especificaciones requeridas. La tabla 3.7 muestra la lista de todos los requerimientos funcionales y su respectiva confirmación de cumplimiento, además se muestra el tiempo que llevo implementar cada uno de los módulos listados.

Tabla 3.7 Lista de cumplimiento de los Requerimientos funcionales y tiempo empleado para su ejecución.

RF	Nombre.	Cumple	Horas
RF001	Escoger el tipo de usuario.	Si	10
RF002	Generar clave.	Si	6
RF003	Autenticar como médico.	Si	6
RF004	Revisar citas médicas a tratar	Si	12
RF005	Autenticar como paciente.	Si	6
RF006	Escoger Especialidad para cita médica.	Si	14
RF007	Visualizar información del médico.	Si	16
RF008	Escoger fecha para cita médica.	Si	20
RF009	Escoger horario para cita médica.	Si	20
RF010	Comprobar disponibilidad de horario.	Si	22
RF011	Agendar cita médica.	Si	30
RF012	Visualizar citas médicas vigentes.	Si	18
RF013	Visualizar resumen de cita médica agendada.	Si	18
RF014	Anular citas médicas agendadas.	Si	20
RF015	Recibir notificación de cita médica vigente.	Si	16
RF016	Autenticar como administrador.	Si	6
RF017	Administrar usuarios.	Si	8
RF018	Buscar información de un usuario.	Si	12
RF019	Actualizar información de un usuario.	Si	12
RF020	Eliminar un usuario.	Si	12
		Total	284

3.3 PRUEBAS DE VALIDACIÓN DE LOS REQUERIMIENTOS NO FUNCIONALES

Se ha realizado pruebas de validación del sistema prototipo de generación de turnos médicos, por medio de encuestas, con la colaboración de 10 personas divididas en Estudiantes, Empleados, Docentes y Doctores de la EPN.

Estas encuestas servirán para que el cliente final haga pruebas con el software final y verifique si este cumple con sus expectativas; estas pruebas tienen como objetivo evaluar si se cumplieron o no los Requerimientos No Funcionales, de la misma manera evaluar la dificultad y el tiempo que tomó a los usuarios efectuar las acciones correspondientes en el sistema prototipo. El formato de las encuestas se observa en las Figuras 3.21 y 3.22.

La información total recolectada se encuentra en el ANEXO G.

Mediante la recreación del escenario de pruebas de la Figura 1.1 del Capítulo 1, Apartado 1.2; fueron realizadas las pruebas mencionadas.

Sistema de Gestión de Turnos Médicos (Usuario Doctor)

Saludos Cordiales, el propósito general de esta encuesta es conocer su opinión, valoraciones y sugerencias para mejorar el desempeño de la aplicación móvil. Dedique unos minutos de su tiempo para llenar el siguiente cuestionario, por favor.

Bienvenido y gracias por participar en esta encuesta.

1. ¿Qué calificación le daría al prototipo de Asignación de turnos Médicos?

1	2	3	4	5
<input type="radio"/>				
Muy Mala	Mala	Buena	Muy Buena	Excelente

2. ¿La presentación del sistema prototipo es llamativa e intuitiva?

1	2	3	4	5
<input type="radio"/>				
Muy Mala	Mala	Buena	Muy Buena	Excelente

3. ¿Pudo usted autenticarse como Doctor?

SI	NO
<input type="radio"/>	<input type="radio"/>

4. ¿Pudo usted desplegar la lista con las citas médicas por atender?

SI	NO
<input type="radio"/>	<input type="radio"/>

5. ¿La información de cada una de las citas médicas listadas por atender es clara?

SI	NO
<input type="radio"/>	<input type="radio"/>

6. El tiempo de espera al realizar una acción fue:

Rápido	Lento
<input type="radio"/>	<input type="radio"/>

7. ¿La sensibilidad del teclado es buena?

SI	NO
<input type="radio"/>	<input type="radio"/>

8. ¿Cómo evalúa su experiencia de utilización con este sistema prototipo?

1	2	3	4	5
<input type="radio"/>				
Muy Mala	Mala	Buena	Muy Buena	Excelente

9. ¿Qué añadiría a la aplicación?
-

10. Comentarios.
-

Figura 3.21 Encuesta para Usuario Doctor.

Sistema de Gestión de Turnos Médicos (Usuario Paciente)

Saludos Cordiales, el propósito general de esta encuesta es conocer su opinión, valoraciones y sugerencias para mejorar el desempeño de la aplicación móvil. Dedique unos minutos de su tiempo para llenar el siguiente cuestionario, por favor.

Bienvenido y gracias por participar en esta encuesta.

1. ¿Qué calificación le daría al prototipo de Asignación de turnos Médicos.

1	2	3	4	5
<input type="radio"/>				
Muy Mala	Mala	Buena	Muy Buena	Excelente

2. ¿La presentación del sistema prototipo es llamativa e intuitiva?

1	2	3	4	5
<input type="radio"/>				
Muy Mala	Mala	Buena	Muy Buena	Excelente

3. ¿La información que se presenta, como mensajes de error y mensajes de información, es clara y entendible?

SI	NO
<input type="radio"/>	<input type="radio"/>

4. ¿Pudo usted autenticarse como Estudiante, Empleado o Docente?

SI	NO
<input type="radio"/>	<input type="radio"/>

5. ¿Pudo usted escoger una especialidad para obtener un turno médico?

SI	NO
<input type="radio"/>	<input type="radio"/>

6. ¿Pudo usted escoger una fecha y horario para agendar una cita médica en la especialidad escogida.?

SI	NO
<input type="radio"/>	<input type="radio"/>

7. ¿Pudo usted desplegar una lista con todas las citas médicas que tiene vigentes?

SI	NO
<input type="radio"/>	<input type="radio"/>

8. ¿Pudo eliminar una cita vigente?

SI	NO
<input type="radio"/>	<input type="radio"/>

9. ¿Recibió recordatorios por medio de notificaciones un día antes de la cita médica vigente?

SI	NO
<input type="radio"/>	<input type="radio"/>

10. El tiempo de espera al realizar una acción fue:

Rápido Lento

11. ¿Cómo evalúa su experiencia de utilización con este sistema prototipo?

1 2 3 4 5

Muy Mala Mala Buena Muy Buena Excelente

12. ¿Qué añadiría a la aplicación?

13. Comentarios.

Figura 3.22 Encuesta para Usuario Paciente.

La Tabla 3.8 muestra el resultado de la **Pregunta 1** de la encuesta que se realiza al Usuario Paciente, en donde se observa que el 75% de los usuarios indican que el sistema prototipo es Excelente.

Tabla 3.8 Respuesta a la Pregunta 1 del Usuario Paciente.

USUARIO PACIENTE					
PREGUNTA 1	En una escala del 1 al 5 que calificación le daría al prototipo de Asignación de turnos Médicos.				
VALORACIÓN	Muy Mala (1)	Mala (2)	Buena (3)	Muy Buena (4)	Excelente (5)
NÚMERO DE RESPUESTAS	0	0	0	2	6
PORCENTAJES	0%	0%	0%	25%	75%

De la misma manera, la Tabla 3.9 muestra el resultado la **Pregunta 1** de la encuesta que se realiza al Usuario Doctor, en donde se observa que el 100% de los usuarios indican que el sistema prototipo es Excelente.

Tabla 3.9 Respuesta a la Pregunta 1 del Usuario Doctor.

USUARIO DOCTOR	
PREGUNTA 1	¿Qué calificación le daría al prototipo de Asignación de turnos Médicos?

VALORACIÓN	Muy Mala (1)	Mala (2)	Buena (3)	Muy Buena (4)	Excelente (5)
NÚMERO DE RESPUESTAS	0	0	0	0	2
PORCENTAJES	0%	0%	0%	0%	100%

En la Tabla 3.10 se puede observar las preguntas y respuestas que se realizaron al usuario Paciente, estas preguntas están orientadas a la experiencia con el uso del sistema prototipo. En base a las respuestas que se pudieron obtener de las encuestas realizadas, se concluye que los usuarios Pacientes fueron capaces de autenticarse de forma correcta, escoger una de las cinco especialidades ofertadas, escoger hora y fecha para reservar una cita médica, verificar si tiene citas médicas vigentes, eliminar una cita médica en particular y recibir recordatorios en forma de notificaciones en el dispositivo móvil. Basándose en los resultados obtenidos en las encuestas, se puede concluir que más del 90% de los pacientes han conseguido utilizar de forma satisfactoria la aplicación móvil.

Tabla 3.10 Preguntas y respuestas, encuesta para usuario Paciente.

USUARIO PACIENTE		
Preguntas	Respuestas	
	SI	NO
3. ¿La información que se presenta, como mensajes de error y mensajes de información, es clara y entendible?	7	1
4. ¿Pudo usted autenticarse como Estudiante, Empleado o Docente?	8	0
5. ¿Pudo usted escoger una especialidad para obtener un turno médico?	8	0
6. ¿pudo usted escoger una fecha y horario para agendar una cita médica en la especialidad escogida.?	7	1
7. ¿Pudo usted desplegar una lista con todas las citas médicas que tiene vigentes?	8	0
8. ¿Pudo eliminar una cita vigente?	8	0
9. ¿Recibió recordatorios por medio de notificaciones un día antes de la cita médica vigente?	7	1

Además, la Tabla 3.11 muestra las preguntas y respuestas que se realizaron al usuario Doctor, estas preguntas están orientadas a la experiencia con el uso del sistema prototipo. En base a las respuestas que se pudieron obtener de las encuestas realizadas, se concluye que los usuarios Doctores fueron capaces de autenticarse de forma correcta, culminar el proceso de registro cambiando la clave de acceso y listar las citas médicas que ellos deben atender. Analizando en los resultados obtenidos en las encuestas, se puede concluir que el 100% de los Doctores han conseguido utilizar de forma satisfactoria la aplicación móvil.

Tabla 3.11 Preguntas y respuestas, encuesta para usuario Doctor.

USUARIO DOCTOR		
Preguntas	Respuestas	
	SI	NO
3. ¿Pudo usted autenticarse como Doctor?	2	0
4. ¿Pudo usted desplegar la lista con las citas médicas por atender?	2	0
5. ¿La información de cada una de las citas médicas listadas por atender es clara?	2	0
7. ¿La sensibilidad del teclado es buena?	1	1

La lista de los requerimientos No funcionales por cumplir se observa en la Tabla 3.12, la misma que fue anteriormente definida en el Capítulo 2 Apartado 2.1.2.2.

Tabla 3.12 Lista de RNF por cumplir.

Requerimientos no Funcionales		
	Requerimiento no funcional	Descripción
Presentación	RNF001	El prototipo debe ser intuitivo y fácil de utilizar para los usuarios.
Disponibilidad	RNF002	El prototipo debe estar disponible al menos un 99% del tiempo
Tiempo de respuesta	RNF003	Dependiendo de la velocidad de internet, el tiempo de respuesta debe ser casi inmediato.
Escalabilidad	RNF004	El prototipo debe ser construido en una plataforma que permita incrementar recursos dependiendo de las necesidades: procesador, almacenamiento, RAM.

Autenticación	RNF005	El acceso al prototipo debe estar restringido por usuario y contraseña para todos los usuarios.
----------------------	--------	---

3.3.1 PRESENTACIÓN.

La presentación del sistema prototipo debe ser amigable, atractiva para la vista, funcional e intuitiva [63], estas características permiten que los usuarios finales puedan realizar acciones en el sistema prototipo de forma sencilla y rápida. Es por ese motivo que la aplicación se diseñó pensando en parámetros visuales como colores, imágenes atractivas, fondos de pantalla amigables.

La tabla 3.13 se observan los resultados de la **Pregunta 2** para el usuario Paciente y usuario Médico, como resultado se tiene que para más del 85% de los usuarios que fueron encuestados la aplicación les resulta atractiva.

Tabla 3.13 Respuesta a la Pregunta 2.

Pregunta 2	¿En la escala del 1 al 5 la presentación del sistema prototipo es llamativa e intuitiva?			
Usuario	Respuestas			
	SI	Porcentaje	NO	Porcentaje
PACIENTE	7	87.5%	1	12.5%
DOCTOR	2	100%	0	0%

3.3.2 DISPONIBILIDAD

Al hablar de disponibilidad, se hace referencia a la utilización de la aplicación móvil en cualquier momento, así como poder acceder a la información de la aplicación cuando el usuario lo requiera, estableciendo así una referencia de que la aplicación funciona de forma correcta [64]. Para la realización de este Proyecto de Titulación se usa un proveedor de servicios AWS [39], el mismo que tiene como una de sus características principales una disponibilidad del 99,99% de tiempo en todas las regiones donde se despliega esta plataforma, basándose en este dato se concluye que el requerimiento de DISPONIBILIDAD se cumple de forma satisfactoria.

3.3.3 TIEMPO DE RESPUESTA

La Tabla 3.14 muestra el resultado de las encuestas realizadas a los usuarios, en la pregunta donde hace referencia a la rapidez al realizar una acción en el sistema prototipo;

arrojando como resultados que más del 85% de los usuarios están satisfechos con el tiempo de respuesta de dicha aplicación. Se debe tomar en cuenta que el tiempo de respuesta se mide desde que el usuario realiza una acción en el sistema prototipo hasta recibir una respuesta exitosa por parte del servidor. Este valor depende de muchos factores para que sea óptimo, como: la calidad de acceso a internet o modelo de teléfono con el que usa la aplicación el usuario final.

Tabla 3.14 Tiempo que tarda la aplicación en realizar una acción.

Pregunta 6	El tiempo de espera al realizar una acción fue:			
Usuario	Respuestas			
	Rápido	Porcentaje	Lento	Porcentaje
PACIENTE	7	87.5%	1	12.5%
DOCTOR	2	100%	0	0%

3.3.4 ESCALABILIDAD

El termino Escalabilidad es usado para explicar la facilidad que posee un sistema para aumentar la capacidad de trabajo sin comprometer el funcionamiento de la aplicación, en resumen, la escalabilidad no es más que la capacidad de un sistema para adaptarse a las diferentes necesidades del usuario final [65]. Como se menciona en el Capítulo 2 Apartado 2.1.2.2, el proveedor AWS ofrece como una de sus características el fácil incremento de recursos, dependiendo de las necesidades del usuario, como: almacenamiento, memoria RAM, número de núcleos; esto hace que el usuario final no tenga que preocuparse por recursos físicos. Debido a este motivo se puede concluir que el Requerimiento no Funcional ESCALABILIDAD se cumple de forma satisfactoria.

3.3.5 AUTENTICACIÓN

Cada uno de los usuarios para hacer uso del sistema prototipo debe autenticarse con las respectivas credenciales, las cuales son “usuario” y “contraseña”. Estas credenciales de acceso deben ser introducidas en la interfaz de inicio de sesión, para luego poder acceder a las diferentes funcionalidades que ofrece el sistema prototipo diseñado para el presente proyecto de titulación. Así lo muestra la Figura 3.23.



Figura 3.23 Interfaz de ingreso al sistema prototipo.

Los Requerimientos No Funcionales validan factores que son externos a la funcionalidad del sistema prototipo, por medio de los cuales se puede realizar la respectiva evaluación al sistema desarrollado. Finalmente, en la Tabla 3.15 se puede observar el cumplimiento de los Requerimientos No Funcionales y el tiempo en horas que llevo la realización de los mismos.

Tabla 3.15 Cumplimiento de Requerimientos No Funcionales y tiempo empleado para su ejecución.

Requerimiento no funcional	Descripción	Cumple	Horas
RNF001	El prototipo debe ser intuitivo y fácil de utilizar para los usuarios.	SI	10
RNF002	El prototipo debe estar disponible al menos un 99% del tiempo	SI	10
RNF003	Dependiendo de la velocidad de internet, el tiempo de respuesta debe ser casi inmediato.	SI	10
RNF004	El prototipo debe ser construido en una plataforma que permita incrementar recursos dependiendo de las necesidades: procesador, almacenamiento, RAM.	SI	10

RNF005	El acceso al prototipo debe estar restringido por usuario y contraseña para todos los usuarios.	SI	10
		Total	50

3.4 ACTUALIZACIÓN FINAL DEL TABLERO KANBAN.

En la Figura 3.24 se muestra cómo están organizadas las tareas realizadas en el tablero Kanban, en el mismo se observa que todas las tareas definidas para la realización del presente Proyecto de titulación han sido culminadas. La columna de “Tareas Realizadas” abarca todas las actividades, dejando vacías las columnas “Lista de Tareas” y “Tareas en Proceso”

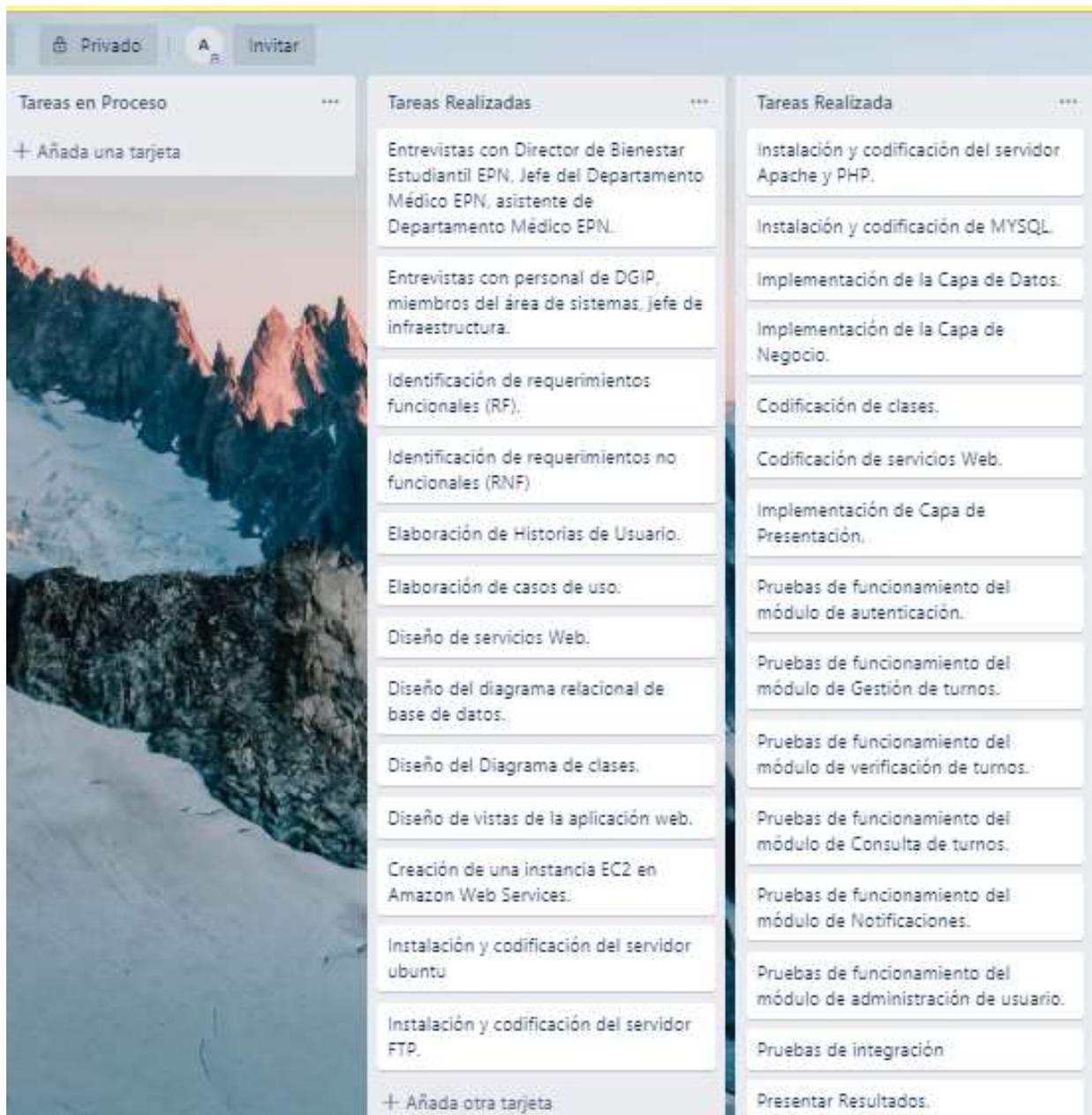


Figura 3.24 Tablero final Kanban.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Concluido el presente Proyecto de Titulación se obtiene como resultado un sistema prototipo de gestión de turnos médicos mediante teléfonos móviles con sistema operativo Android para el Departamento Médico de la Escuela Politécnica Nacional, el mismo que aprovecha el gran despliegue de teléfonos inteligentes que existe en la sociedad actualmente. El sistema prototipo logra mejorar de forma considerable la forma actual de asignación turnos médicos (forma manual), solucionando de esta forma problemas como: aglomeración de pacientes en instalaciones del Departamento Médico, larga espera de tiempo por parte de pacientes, etc. Adicionalmente, toda la información que maneja el sistema prototipo se encuentra respaldada y disponible en la nube, a través del proveedor de servicios AWS.
- Para la elaboración del sistema prototipo de gestión de turnos médicos se basó en la metodología Kanban, permitiendo organizar de forma ordenada las tareas a realizar, de esta manera se logra una optimizar el tiempo para la realización del sistema prototipo. La metodología Kanban permitió que la implementación de las capas definidas: Datos, Negocio y Presentación sean divididas en tareas pequeñas. Como, por ejemplo: análisis de requerimientos funcionales y no funcionales, diseño de casos de uso, etc.
- El presente Proyecto de Titulación permitió profundizar en tecnologías y conceptos tales como: Android Studio, Programación de Servicios Web, PHP, Base de Datos MYSQL, Arquitectura en Capas, etc.; Estos conceptos permitieron mejorar habilidades para el diseño y desarrollo de software, a su vez los conocimientos en Programación y Base de Datos aumentaron gradualmente con la implementación de cada uno de los módulos del sistema prototipo. Todas las tecnologías y conceptos mencionados se utilizaron en el desarrollo del Proyecto de Titulación.
- Para la implementación del servidor Web y almacenamiento de datos del sistema prototipo de gestión de turnos se utiliza el servicio en la nube de Amazon Web Services, el mismo que aporta una disponibilidad del 99.99% del tiempo, por medio de una conexión a internet. Esta característica permite a los usuarios emular un ambiente real y conectarse en cualquier momento y desde cualquier lugar al servidor. Además, AWS permite que los recursos de hardware se adapten de forma dinámica dependiendo de la cantidad de información en el servidor.

- La toma de Requerimientos Funcionales y No Funcionales permitieron hacer un análisis a profundidad de las necesidades reales que tiene el usuario final (Doctores y Estudiantes, Empleados y Docentes), además, estos requerimientos permiten al desarrollador tener un panorama claro acerca del proceso de diseño al momento de elaborar los diversos diagramas de clases, diagrama de base de datos, diagramas de casos de uso, ya que para elaborar dichos diagramas se toma como base que es lo que realmente desea el cliente para el sistema prototipo. Finalmente, el diseño de la interfaz gráfica con la que interactúa el usuario final también se basa en los requerimientos mencionados.
- En la etapa de diseño se planteó una solución que permita que todos los requerimientos que se plantearon se cumplan de forma satisfactoria. Por tal razón en este proceso cada uno de los diferentes requerimientos se agruparon en módulos dependiendo del tipo de actividad o función. Como ejemplo, el módulo Consulta de turnos abarca todas las tareas que se relacionan con desplegar una lista de turnos previamente guardados. De esta manera es como se ha logrado agrupar de forma satisfactoria las funcionalidades que el sistema prototipo debe tener.
- Las pruebas finales que se realizaron al sistema prototipo demuestran que tanto los Requerimientos Funcionales como los Requerimientos No Funcionales se cumplen a cabalidad. Demostrando esto con las encuestas realizadas a usuarios finales divididos entre Estudiantes, Docentes, Empleados y Doctores; las mismas que reflejan una satisfacción de aproximadamente el 90% al uso de la aplicación. A su vez, el 100% de los usuarios encuestados afirman que las funcionalidades de la aplicación son correctas.

4.2 RECOMENDACIONES

- Para que el sistema prototipo tenga una excelente aceptación por parte del usuario final es importante realizar entrevistas periódicas con el cliente que ha solicitado el software, esto se realiza con la finalidad de evitar demoras en la realización del mismo ya que en cada entrevista el cliente puede sugerir algunos cambios o modificaciones en el producto, estas modificaciones pueden ser en la base de datos, diseño de clases o interfaces gráficas.

- Al momento de desarrollar un software de cualquier tipo debe ser imprescindible el uso de una metodología, ya que esta permite al programador llevar de forma ordenada las tareas respectivas para la evolución del proyecto y así entregar a tiempo el producto final. La elección de la metodología debe ser escogida por el desarrollador dependiendo a lo que más se ajuste a sus necesidades, en el caso del desarrollo de este Proyecto de Titulación se utilizó KANBAN.
- Actualmente el desarrollo de software ha tenido un gran impulso en el mercado laboral, por lo que se recomienda que la carrera de Ingeniería en Tecnologías de la Información haga énfasis a materias relacionadas con el desarrollo de software para que así los graduados tengan más probabilidades de trabajo o de emprendimiento.
- Una aplicación de este tipo puede tener miles de usuarios que interactúen con las misma, por lo tanto, la utilización del servidor será muy alta; es por este motivo que se necesita hacer uso de una mayor capacidad tanto de almacenamiento, procesamiento, ancho de banda o memoria RAM. Por lo que se recomienda actualizar el servicio en la nube a uno con mejores características para que el flujo de datos en el servidor no tenga inconvenientes.
- Al implementar peticiones http desde Android hacia cualquier servidor externo, es importante el uso de la librería Volley ya que es la mejor forma de optimizar líneas de código repetitivas, al contrario de realizar peticiones con HttpURLConnection el cual implementa demasiado código para realizar la misma función que Volley. Tomar en cuenta que Volley es la librería que mejor procesa las peticiones http.
- Mediante lo dialogado con el Director de Bienestar Estudiantil y a petición del mismo se recomiendo incentivar la realización e implementación de aplicaciones que el Departamento Médico de EPN necesita, en este caso el director del bienestar estudiantil quiere que se implemente aplicaciones como administrador de fichas clínicas o administrador de inventarios de equipos y medicina del Departamento Médico mencionado.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] J. Nolasco, Desarrollo de Aplicaciones móviles con Android, 2da. Edición, Marcombo Bogotá, 2016, p 128-134.

[2] “Departamento Medico EPN”. [En línea]. Disponible: <https://www.epn.edu.ec/institucion/servicios-internos/bienestar/> [Consultado: 12-julio-2018].

[3] C. Date, Introducción a los sistemas de base de datos, 5ta. Edición, Addison-Wesley Iberoamericana Wilmington, 1993, p 178-188.

[4] “Que es Cloud Computing” [En línea]. Disponible: <https://www.salesforce.com/mx/cloud-computing/> [Consultado: 22-nov-2018].

[5] “Historia y Evolución del Smartphone”. [En línea]. Disponible: <http://smartphoneavancetecnologico.blogspot.com/p/historia-y-evolucion-del-smartphone.html> [Consultado: 22-julio-2019].

[6] “Smartphones” [En línea]. Disponible: <https://histinf.blogs.upv.es/2012/12/03/smartphones/> [Consultado: 22-nov-2019].

[7] “Xataka tecnología”. [En línea]. Disponible: <https://www.xataka.com/>
[Consultado: 23-julio-2019].

[8] “El primer Top 5 de fabricantes móviles de 2019 sitúa a Huawei muy por delante de Apple” [En línea]. Disponible: <https://www.xatakamovil.com/mercado/primer-top-5-fabricantes-moviles-2019-situa-a-huawei-muy-delante-apple> [Consultado: 23-julio-2019].

[9] “Android domina el tercer trimestre de 2018 con un 86,8% de cuota de mercado, según IDC” [En línea]. Disponible: <https://www.xatakandroid.com/moviles-android/android-domina-tercer-trimestre-2018-86-8-cuota-mercado-idc> [Consultado: 23-julio-2019].

[10] “Definición y cómo funcionan las aplicaciones móviles” [En línea]. Disponible: <https://www.servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/> [Consultado: 23-julio-2019].

[11] “GOOGLE PLAY” [En línea]. Disponible: <https://play.google.com/store> [Consultado: 23-julio-2019].

[12] “Apple” [En línea]. Disponible: <https://www.apple.com/retail/> [Consultado: 23-julio-2019].

- [13] “Proceso de diseño y desarrollo de una app” [En línea]. Disponible: <http://appdesignbook.com/es/contenidos/las-aplicaciones/> [Consultado: 24-julio-2019].
- [14] “Aplicaciones Nativas” [En línea]. Disponible: <http://appdesignbook.com/es/contenidos/las-aplicaciones> [Consultado: 24-julio-2019].
- [15] “Tipos de App” [En línea]. Disponible: <http://dchain.com/la-influencia-de-los-colores-en-el-diseno-web-2/> [Consultado: 24-julio-2019].
- [16] “Aplicaciones Web” [En línea]. Disponible: <http://dchain.com/la-influencia-de-los-colores-en-el-diseno-web-2> [Consultado: 24-julio-2019].
- [17] “Los tres tipos de Aplicaciones Móviles” [En línea]. Disponible: <https://okhosting.com/blog/cuales-son-los-3-tipos-de-aplicaciones-moviles/> [Consultado: 24-julio-2019].
- [18] “Aplicaciones Híbridas” [En línea]. Disponible: <http://appdesignbook.com/es/contenidos/las-aplicaciones/> [Consultado: 24-julio-2019].
- [19] “METODO KANBAN: PRINCIPIOS Y VENTAJAS” [En línea]. Disponible: <https://apiumhub.com/es/tech-blog-barcelona/metodo-kanban-ventajas/> [Consultado: 25-julio-2019].
- [20] “4 Principios: Método KANBAN” [En línea]. Disponible: <https://apiumhub.com/es/tech-blog-barcelona/metodo-kanban-ventajas/> [Consultado: 28-julio-2019].
- [21] “Los principios de la metodología Kanban” [En línea]. Disponible: <https://www.iebschool.com/blog/metodologia-kanban-agile-scrum/> [Consultado: 28-julio-2019].
- [22] “Trello” [En línea]. Disponible: <https://trello.com/b/gmgleSCP/bienvenido-a-trello> [Consultado: 28-julio-2019].
- [23] “Historia y Evolución de Android” [En línea]. Disponible: <https://www.xatakandroid.com/sistema-operativo/historia-y-evolucion-de-android-como-un-sistema-operativo-para-camaras-digitales-acabo-conquistando-los-moviles> [Consultado: 30-julio-2019].
- [24] “Cuando Google conoció a Android” [En línea]. Disponible: <https://hipertextual.com/2018/12/historia-origen-android> [Consultado: 30-julio-2019].
- [25] “Características de Android” [En línea]. Disponible: <https://es.wikipedia.org/wiki/Android#Caracter%C3%ADsticas> [Consultado: 30-julio-2019].

- [26] “Conociendo Android Estudio” [En línea]. Disponible: <https://developer.android.com/studio/intro?hl=es-419> [Consultado: 01-agosto-2019].
- [27] “DEVELOPERS” [En línea]. Disponible: <https://developer.android.com/studio/intro?hl=es-419> [Consultado: 01-agosto-2019].
- [28] “IDE para Android” [En línea]. Disponible: <https://academiaandroid.com/ide-android-intellij-android-studio-aide/> [Consultado: 01-agosto-2019].
- [29] “Que es y para qué sirve MYSQL” [En línea]. Disponible: <http://culturacion.com/que-es-y-para-que-sirve-mysql/> [Consultado: 02- agosto -2019].
- [30] “Concepto de MYSQL” [En línea]. Disponible: <http://culturacion.com/que-es-y-para-que-sirve-mysql/> [Consultado: 04- agosto -2019].
- [31] “Utilidades de MYSQL” [En línea]. Disponible: <http://culturacion.com/que-es-y-para-que-sirve-mysql/> [Consultado: 04- agosto -2019].
- [32] “PhpMyAdmin” [En línea]. Disponible: <https://www.phpmyadmin.net/> [Consultado: 04- agosto -2019].
- [33] “¿Qué nos ofrece PhpMyadmin?” [En línea]. Disponible: <https://www.coriaweb.hosting/nos-ofrece-phpmyadmin/> [Consultado: 04- agosto -2019].
- [34] “¿Qué es PHP?” [En línea]. Disponible: <https://www.php.net/manual/es/intro-what-is.php> [Consultado: 05- agosto -2019].
- [35] “¿Qué es PHP y para qué sirve PHP?” [En línea]. Disponible: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:i-que-es-php-y-para-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193 [Consultado: 05- agosto -2019].
- [36] “¿Qué es PHP y para qué sirve PHP?” [En línea]. Disponible: <http://redgrafica.com/El-lenguaje-de-programacion-PHP> [Consultado: 05- agosto -2019].
- [37] “¿Qué es Amazon Web Services?” [En línea]. Disponible: <https://www.ticportal.es/temas/cloud-computing/amazon-web-services> [Consultado: 07- agosto -2019].
- [38] “¿Amazon EC2?” [En línea]. Disponible: <https://aws.amazon.com/es/ec2/> [Consultado: 07- agosto -2019].

- [39] “¿Qué es Amazon Web Services y para qué sirve?” [En línea]. Disponible: <https://amara.org/es/videos/C5xRi8FXyXoK/info/que-es-amazon-web-services/> [Consultado: 07- agosto -2019].
- [40] “¿Qué es un servicio web?” [En línea]. Disponible: <https://medium.com/grupo-carricay/qu%C3%A9-es-un-servicio-web-510be516863> [Consultado: 08- agosto -2019].
- [41] “Servicios Web” [En línea]. Disponible: http://www.hipertexto.info/documentos/serv_web.htm [Consultado: 08- agosto -2019].
- [42] “Que son los web services y que tecnologías usar en su desarrollo” [En línea]. <https://www.arsys.es/blog/programacion/disenio-web/web-services-desarrollo/> [Consultado: 08- agosto -2019].
- [43] “¿Qué es JSON, para que sirve y donde se usa?” [En línea] Disponible: <https://canela.me/articulo/JSON-JavaScript-jQuery/%C2%BFQu%C3%A9-es-JSON-para-qu%C3%A9-sirve-y-d%C3%B3nde-se-usa> [Consultado: 09- agosto -2019].
- [44] “¿Qué es JSON, para que sirve y donde se usa?” [En línea] Disponible: <https://www.json.org/json-es.html> [Consultado: 09- agosto -2019].
- [45] “¿Definición de FTP?” [En línea] Disponible: <https://definicion.de/ftp/> [Consultado: 09- agosto -2019].
- [46] “¿Qué es Fillezilla?” [En línea] Disponible: <https://neoattack.com/neowiki/filezilla/> [Consultado: 09- agosto -2019].
- [47] “¿Ejemplo de tablero Kanban básico?” [En línea] Disponible: https://kanbanize.com/es/recursos-de-kanban/software-kanban/ejemplos-de-tableros-kanban/#basic_kanban_board [Consultado: 12- agosto -2019].
- [48] “¿Identificación de requerimientos funcionales?” [En línea] Disponible: <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales> [Consultado: 12- agosto -2019].
- [49] “¿Requerimientos no funcionales?” [En línea] Disponible: <https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a> [Consultado: 12- agosto -2019].
- [50] “¿Qué son los requerimientos funcionales?” [En línea] Disponible: <https://ingenieriadesoftwareutmachala.wordpress.com/2017/01/20/requerimientos-funcionales-y-no-funcionales/> [Consultado: 15- agosto -2019].

- [51] “¿Historias de Usuario?” [En línea] Disponible: <https://drive.google.com/file/d/0B2eI86yiiYU0NG5IeEowWHY3dk0/view> [Consultado: 15- agosto -2019].
- [52] “¿Qué son los requerimientos no funcionales?” [En línea] Disponible: <http://www.pmoinformatica.com/2013/01/requerimientos-no-funcionales-porque.html> [Consultado: 15- agosto -2019].
- [53] “phpMyAdmin Diseñador” [En línea] Disponible: http://ec2-13-59-203-174.us-east-2.compute.amazonaws.com/phpmyadmin/db_designer.php?db=db_tesis&server=3&token=b29abea5733fde0743fe401179c59458 [Consultado: 21- agosto -2019].
- [54] “¿Qué es un modelo Entidad Relación?” [En línea] Disponible: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion> [Consultado: 21- agosto -2019].
- [55] “¿Diagramas de caso de uso?” [En línea] Disponible: <https://openclassrooms.com/en/courses/4990961-planea-tu-proyecto-con-uml/4996511-diagramas-de-casos-de-uso> [Consultado: 23- agosto -2019].
- [56] “¿Qué es un diagrama de clases?” [En línea] Disponible: <https://okdiario.com/curiosidades/que-diagrama-clases-3323710> [Consultado: 28- agosto -2019].
- [57] “Lucidchart” [En línea] Disponible: https://www.lucidchart.com/documents#docs?folder_id=search&browser=icon&sort=saved-desc [Consultado: 28- agosto -2019].
- [58] “Visual Paradigm Online Diagram” [En línea] Disponible: <https://diagrams.visual-paradigm.com/#proj=0&type=AndroidWireframe> [Consultado: 30- agosto -2019].
- [59] “Amazon EC2” [En línea] Disponible: https://aws.amazon.com/es/ec2/?did=ft_card&trk=ft_card [Consultado: 31- agosto -2019].
- [60] “phpMyAdmin Bringing MySQL to the Web” [En línea] Disponible: <https://www.phpmyadmin.net/> [Consultado: 01- septiembre -2019].
- [61] “FileZilla the best FTP solution” [En línea] Disponible: <https://filezilla-project.org/> [Consultado: 01- septiembre -2019].
- [62] “Notepad++ v7.66 home” [En línea] Disponible: <https://notepad-plus-plus.org/download/v7.6.6.html> [Consultado: 03- septiembre -2019].

[63] “Presentación” [En línea] Disponible: <http://appdesignbook.com/es/contenidos/presentacion/> [Consultado: 17- septiembre - 2019].

[64] “Seguridad de la información” [En línea] Disponible: <https://www.tecon.es/la-seguridad-de-la-informacion/> [Consultado: 17- septiembre -2019].

[65] “La escalabilidad de Cloud Computing” [En línea] Disponible: <https://blog.dataprius.com/index.php/2015/04/10/versatilidad-y-escalabilidad-del-cloud-computing/> [Consultado: 17- septiembre -2019].

6. ANEXOS

ANEXO A. HISTORIAS DE USUARIO

Las Historias de Usuario se encuentran en el CD adjunto a este documento

ANEXO B. PERMISO DE USO DE LA BASE DE DATOS

El permiso se encuentra en el CD adjunto a este documento

ANEXO C. CÓDIGO DE TABLAS DE LA BASE DE DATOS

El código se encuentra en el CD adjunto a este documento

ANEXO D. CÓDIGO CLASES

El código se encuentra en el CD adjunto a este documento

ANEXO E. CÓDIGO DE SERVICIOS WEB

El código se encuentra en el CD adjunto a este documento

ANEXO F. CÓDIGO DE INTERFAZ FRÁFICA

El código se encuentra en el CD adjunto a este documento

ANEXO G. ENCUESTAS A USUARIOS DE SISTEMA PROTOTIPO

Las encuestas se encuentran en el CD adjunto a este documento.

ORDEN DE EMPASTADO