

UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA



PROYECTO DE GRADO

**DISEÑO DE UNA PLATAFORMA DIGITAL DE
INFORMACION Y CONSULTA A TRAVES DE UNA
APLICACION MOVIL**

CASO: CURSO PREFACULTATIVO (C.P.F.)

POSTULANTE: DIEGO COAQUIRA ILLANES

TUTOR: ING. EDGAR GONZALES LAURA

DOCENTE: ING. ALFONSO JURADO

LA PAZ – BOLIVIA

2021



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERIA**



LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la cita o referencia correspondiente en apego a las normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADAS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A Dios, sin Él nada hubiese sido posible.

A la memoria de mi Señor Padre Orlando, hoy que ya no está junto a mí, le digo: “Papá, misión cumplida”.

A mi Madre María, mujer incansable, gracias por tu abnegación, sembraste la semilla, ahora este es tu fruto.

A mis hermanos Giovanni, Jesús y Leonardo por su cariño y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento gracias.

A Daiana, mi gran amor por ser mi compañera inseparable de cada día.

AGRADECIMIENTOS

A Dios por dejarme ver un nuevo día, despertar y admirar todas las cosas bellas que realizó.

A mi difunto padre, a mi madre y a toda mi familia, por todo su apoyo incondicional que recibí durante mi formación.

A mi docente Tutor Ing. Edgar Gonzales Laura, cuyos consejos y sugerencias a lo largo del desarrollo de este proyecto han sido fundamentales para la conclusión del mismo.

Al Ing. Gustavo Gonzales Gómez, ex Director del Curso Prefacultativo quién me brindó su confianza y con sus sabios consejos me impulsaron a concluir esta etapa.

A mi adorada Facultad de Ingeniería, a la Carrera de Electrónica y a mis Docentes, por haberme acogido y apoyado en mi etapa de formación académica.

Al Curso Prefacultativo (C.P.F.) que me brindó la oportunidad de compartir conocimientos con nuevas generaciones y colaborar en la formación de futuros profesionales.

A mis compañeros que día a día nos superamos para seguir adelante y vivir muchas vivencias en la Universidad.

Resumen

Se presenta el diseño de una plataforma digital de información y consulta para teléfonos móviles inteligentes o *Smartphones*¹, la cual es una aplicación desarrollada para la plataforma Android con la finalidad de gestionar la información de los postulantes del Curso Prefacultativo de la Facultad de Ingeniería perteneciente a la Universidad Mayor de San Andrés, para el ingreso a una de las quince carreras pertenecientes a la Facultad de Ingeniería de la mencionada universidad, misma que brindará la facilidad de recabar información referida a Instructivos, Comunicados, Horarios, Cronogramas de Exámenes y Distribución de Aulas, además de contar con una comunicación móvil en tiempo real sin importar el lugar donde se encuentren, ofreciendo comodidad y evitando aglomeraciones en los predios del Curso Prefacultativo, cuya infraestructura es compartida con el Curso Básico de Ingeniería. El presente proyecto se constituye en una alternativa de información y consulta con el “Panel de Información” general que queda ubicado en la Secretaría del Curso Prefacultativo.

¹ El término *Smartphone* pertenece a la lengua inglesa que en el idioma español se conoce como “teléfono inteligente”, se trata de un dispositivo móvil que ofrece prestaciones similares a las de una computadora (ordenador) y que se destaca por su conectividad de datos.

(<https://definicion.de/smartphone/>)

ÍNDICE DE CONTENIDO

CAPÍTULO I.....	1
MARCO REFERENCIAL	1
1.1. Introducción	1
1.2. Antecedentes	1
1.2.1. Modalidades de Admisión Prefacultativa	2
1.3. Planteamiento del Problema.....	3
1.3.1. Formulación del Problema.....	3
1.4. Objetivos	3
1.4.1. Objetivo Principal	3
1.4.2. Objetivos Específicos.....	3
1.5. Justificación.....	4
1.5.1. Social.....	4
1.5.2. Tecnológica.....	4
1.5.3. Económica.....	5
1.6. Alcances y Limitaciones	5
1.7. Descripción de la Solución Propuesta.....	5
1.7.1. Elementos del Modelo Cliente - Servidor.....	6
1.7.1.1. Subsistema Cliente	6
1.7.1.2. Subsistema Servidor	7
CAPÍTULO II	8
MARCO TEÓRICO	8
2.1. Tecnología Móvil	8
2.2. Sistemas Operativos para Teléfonos Móviles	9
2.3. Sistema Operativo Android.....	11
2.4. Aplicaciones Móviles.....	13
2.4.1. Aplicaciones Nativas	13
2.4.2. Estructura de una Aplicación Android.....	15
2.5. Plataforma de Desarrollo Android	16
2.6. Metodologías de Desarrollo Ágil	18

2.6.1.	Metodología de Desarrollo Ágil RAD	19
2.6.1.1.	Fases de la Metodología RAD	20
2.6.1.2.	Ventajas.....	21
2.7.	Lenguaje UML	22
2.7.1.	Diagramas UML	23
2.7.1.1.	Diagrama de Casos de Uso	23
2.7.1.2.	Diagrama de Clases.....	25
2.7.1.3.	Diagrama de Actividades	26
2.8.	Calidad y Seguridad de Software	28
2.8.1.	Calidad de Software (Norma ISO-9126)	28
2.8.1.1.	Características Propuestas por la ISO-9126	29
2.8.2.	Seguridad de la Información.....	32
2.8.2.1.	Criptografía.....	33
2.8.2.2.	Encriptación de Datos.....	33
2.9.	Emulador Genymotion	34
2.10.	Android Studio	35
2.10.1.	Características de Android Studio.....	35
2.10.2.	Ventajas de Android Estudio.....	36
2.10.3.	Desventajas de Android Estudio	37
2.11.	BASES DE DATOS.....	38
2.12.	XAMPP	38
2.12.1.	MySQL.....	40
2.12.2.	PHP 7.0.....	41
CAPÍTULO III		42
INGENIERÍA DEL PROYECTO.....		42
3.1.	Metodología de la Investigación	42
3.1.1.	Tipo de Investigación.....	42
3.1.2.	Población y Muestra	42
3.1.3.	Recolección de Información	42
3.1.4.	Procesamiento y Análisis de Datos.....	43
3.2.	Diagnóstico sobre el Actual Proceso de Gestión Académica	43

3.2.1.	Análisis FODA.....	46
3.2.1.1.	Fortalezas.....	46
3.2.1.2.	Debilidades.....	46
3.2.1.3.	Oportunidades.....	46
3.2.1.4.	Amenazas.....	46
3.3.	Datos Generales sobre el Curso Prefacultativo.....	47
3.3.1.	Antecedentes Históricos.....	47
3.3.2.	Curso Prefacultativo.....	49
3.3.2.1.	Objetivos del Curso Prefacultativo.....	50
3.3.2.2.	Misión.....	51
3.3.2.3.	Visión.....	51
3.3.2.4.	Organigrama del Curso Prefacultativo.....	51
3.4.	Análisis de la Situación Actual.....	52
3.4.1.	Análisis de Procesos Actuales.....	52
3.4.1.1.	Proceso de Publicación en Página Web de la UMSA.....	52
3.4.1.2.	Proceso de Publicación en Página Web del Curso Prefacultativo.....	53
3.4.1.3.	Proceso de Publicación en Facebook.....	54
3.4.1.4.	Proceso de Inscripción de Postulantes.....	55
3.4.2.	Identificación de Actores.....	56
3.5.	Requerimientos.....	58
3.5.1.	Determinación de los Roles del Equipo.....	58
3.5.2.	Especificación de Usuarios.....	58
3.5.3.	Requerimientos Funcionales de la Aplicación.....	59
3.6.	Requerimientos no Funcionales.....	61
3.7.	Arquitectura de la Solución.....	62
3.7.1.	Arquitectura del Sistema.....	62
3.7.2.	Arquitectura de la Aplicación.....	63
3.8.	Diagramas de Casos de Uso.....	64
3.8.1.	Caso de Uso General del Usuario.....	64
3.8.2.	Caso de Uso del Administrador de la Base de Datos.....	65
3.9.	Diagrama de Actividades.....	66

3.10. Encriptación de <i>Password</i> de Acceso con MD5 en PHP	67
3.11. Diseño de Interfaces	68
3.11.1. Interfaces de la Aplicación Móvil para el USUARIO.....	68
3.11.2. Interfaces del sistema para el ADMINISTRADOR.....	75
CAPÍTULO IV	79
ANÁLISIS DE COSTO DEL PROYECTO	79
4.1. Costos de Investigación	79
4.2. Costo de equipos.....	79
4.3. Costos de Software	80
4.3.1. Punto de Función.....	81
4.3.2. Costos del Sistema Desarrollado.....	84
4.4. Costo Total	86
CAPÍTULO V	87
CONCLUSIONES, RECOMENDACIONES Y MEJORAS.....	87
5.1. Conclusiones	87
5.2. Recomendaciones y Mejoras.....	89
5.3. Bibliografía.....	91
ANEXOS.....	93

ÍNDICE DE FIGURAS

Figura 1.1. Diagrama de bloques de la solución propuesta.....	6
Figura 2.1. Arquitectura del Sistema Operativo Android.....	11
Figura 2.2. Funcionamiento de una aplicación nativa.....	14
Figura 2.3. Estructura de una aplicación Android.....	15
Figura 2.4. Logo de Android.....	16
Figura 2.5. Distribución de versiones de Android.....	18
Figura 2.6. Modelo RAD.....	21
Figura 2.7. Ejemplo de Diagrama de Caso de Uso.....	24
Figura 2.8. Ejemplo de Diagrama de Caso de Clases.....	26
Figura 2.9. Ejemplo de Diagrama de Actividades.....	27
Figura 2.10. Pantalla de Configuración de Dispositivos de Genymotion.....	34
Figura 2.11. Logo de Android Studio.....	35
Figura 2.12. Entorno de desarrollo Android Studio.....	37
Figura 2.13. Panel de control XAMPP.....	39
Figura 2.14. Desarrollo de Bases de Datos en MySQL.....	40
Figura 2.15. Comunicación Cliente-Servidor con intérprete PHP.....	41
Figura 3.1. Página Web de la UMSA.....	43
Figura 3.2. Página Web del Curso Prefacultativo.....	44
Figura 3.3. Acceso al Sistema para inscripción al Curso Prefacultativo.....	44
Figura 3.4. Acceso al Sistema para consultar notas del Curso Prefacultativo.....	45
Figura 3.5. Fan Page en Facebook del Curso Prefacultativo.....	45
Figura 3.6. Edificio del Curso Prefacultativo.....	49
Figura 3.7. Organigrama del Curso Prefacultativo.....	51
Figura 3.8. Proceso Actual sobre las fases de difusión de información.....	52
Figura 3.9. Proceso de publicación de información en Página Web de la UMSA.....	53

Figura 3.10. Proceso de publicación de información en Página Web del Curso Prefacultativo.....	54
Figura 3.11. Proceso de publicación de información en Facebook.....	55
Figura 3.12. Proceso de Inscripción de Postulantes.....	56
Figura 3.13. Arquitectura del Sistema.....	63
Figura 3.14. Arquitectura de la Aplicación.....	63
Figura 3.15. Diagrama de Caso de Uso de Usuario.....	64
Figura 3.16. Diagrama de Caso de Uso del Administrador de la Base de Datos.....	65
Figura 3.17. Diagrama de Caso de Uso del Administrador de la Base de Datos.....	66
Figura 3.18. Generación de Contenido en la Aplicación.....	67
Figura 3.19. Interface principal de la Aplicación Móvil.....	69
Figura 3.20. Pantalla de la opción Convocatorias.....	70
Figura 3.21. Pantalla de la opción Comunicados.....	71
Figura 3.22. Pantalla de la opción Horarios.....	72
Figura 3.23. Pantalla de la opción Cronograma de exámenes.....	73
Figura 3.24. Pantalla de la opción Contáctanos (Login).....	74
Figura 3.25. Pantalla de Ingreso al Sistema Administrador.....	75
Figura 3.26. Pantalla de Envío de Mensajes Administrador.....	76
Figura 3.27. Pantalla de Recepción de Mensajes del Administrador.....	76
Figura 3.28. Pantalla de subida de Archivos del administrador.....	77
Figura 3.29. Líneas de código del fichero Activity_Main.xml.....	77
Figura 3.30. Líneas de código del fichero MainActivity.java.....	78

ÍNDICE DE TABLAS

Tabla 2.1. Sistemas Operativos Móviles más comunes.....	10
Tabla 2.2. Diferencias entre Metodologías Ágiles y No Ágiles.....	19
Tabla 2.3. Características de la Norma ISO-9126.....	29
Tabla 3.1. Actores involucrados.....	57
Tabla 3.2. Determinación de roles.....	58
Tabla 3.3. Especificación de usuarios.....	59
Tabla 3.4. Requerimiento Funcional N° 1.....	59
Tabla 3.5. Requerimiento Funcional N° 2.....	60
Tabla 3.6. Requerimiento Funcional N° 3.....	60
Tabla 3.7. Requerimiento Funcional N° 1.....	60
Tabla 3.8. Requerimiento Funcional N° 2.....	61
Tabla 3.9. Requerimiento Funcional N° 3.....	61
Tabla 3.10. Requerimiento Funcional N° 3.....	64
Tabla 3.11. Requerimiento Funcional N° 3.....	65
Tabla 4.1. Costos de la Investigación.....	79
Tabla 4.2. Costo del Equipo Empleado.....	80
Tabla 4.3. Entradas del Sistema.....	81
Tabla 4.4. Salidas del Sistema.....	81
Tabla 4.5. Peticiones de Usuario.....	82
Tabla 4.6. Archivos de la Interface y la Base de Datos.....	82
Tabla 4.7. Parámetros de Medición del Sistema.....	82
Tabla 4.8. Factor de Complejidad del Sistema.....	83
Tabla 4.9. Factor de Ajuste.....	83
Tabla 4.10. Punto de Función.....	84
Tabla 4.11. Factor LCD/PF de Lenguajes de Programación.....	84
Tabla 4.12. Costo Total.....	86

CAPÍTULO I

MARCO REFERENCIAL

1.1. Introducción

En la actualidad, la Facultad de Ingeniería perteneciente a la Universidad Mayor de San Andrés, contempla quince carreras las cuales son: Ingeniería Civil, Eléctrica, Química, Electrónica, Industrial, Mecánica, Metalúrgica y de Materiales, Petrolera, de Alimentos, Ambiental, Mecatrónica, en Seguridad Industrial y Salud Ocupacional, en Producción Industrial, Petroquímica y Automotriz, todas ellas encaminadas a formar profesionales que contribuyan al desarrollo y bienestar de la sociedad boliviana.

Conforme a resolución N° 184/98 de fecha 09 de septiembre de 1998 del Honorable Consejo Universitario, se permite el ingreso de nuevos estudiantes a la Universidad Mayor de San Andrés previa aprobación de la Prueba de Suficiencia Académica (P.S.A.), o del Curso Prefacultativo (C.P.F.), para lo cual la Facultad de Ingeniería convoca en fechas específicas a todos los bachilleres interesados en postular a cualquiera de las carreras mencionadas².

Cada postulante sigue una secuencia de pasos, desde el momento de la inscripción hasta la conclusión de las evaluaciones escritas, es en tal caso y debido al considerable número de postulantes a una de las carreras de la Facultad de Ingeniería, que el organismo Prefacultativo emite continuamente comunicados relevantes, que deben ser cumplidos estrictamente por los postulantes, para un correcto desenvolvimiento de las instrucciones que les toca realizar, así mismo las interrogantes a ciertos pasos, obliga a mantenerse informado continuamente y de manera presencial, pues no se cuenta con un modo de consulta alterna.

1.2. Antecedentes

Los postulantes a la Facultad de Ingeniería experimentan diversas dificultades en todo el proceso que dura la modalidad de Admisión Prefacultativa, desde el momento de las inscripciones hasta el

² Convocatoria C.P.F. 01/2019.

momento de culminar las pruebas escritas, una en particular, es la continua comunicación existente entre el organismo Prefacultativo y el estudiantado en general, ocasionando aglomeraciones posteriores a la publicación de diversos comunicados de interés dirigidos a los postulantes, sumado a esto, los casos particulares que se presentan con cada postulante.

1.2.1. Modalidades de Admisión Prefacultativa

Las distintas modalidades de Admisión Prefacultativa tienen por objeto nivelar y fortalecer conocimientos de materias básicas, necesarias para el desarrollo formativo en la carrera a la que postula, fortaleciendo así sus aptitudes y habilidades para realizar estudios superiores. Esto se logra a través de métodos didácticos, técnicas de investigación y razonamiento lógico-matemático.

Para este caso, se contemplan dos modalidades de Admisión:

- **Prueba de Suficiencia Académica (P.S.A.):** Consiste en someter a los estudiantes a dicha prueba.
- **Curso Prefacultativo (C.P.F.) Presencial y Virtual:** Consiste en un curso de nivelación de conocimientos básicos para realizar estudios superiores en la Universidad, pudiendo asistir a alguno de los turnos habilitados o en su caso de manera virtual por medio de la plataforma Web del Curso Prefacultativo.

Los principales procesos que se desarrollan dentro del Curso Prefacultativo son:

- Inscripción de postulantes.
- Preparación y realización de pruebas de orientación profesional.
- Preparación y realización de exámenes de suficiencia del Curso Prefacultativo.
- Registro de calificaciones.
- Emisión de certificaciones de habilitación y/o aprobación.
- Registro y control de asistencia de los docentes, auxiliares y administrativos.
- Reportes Estadísticos (cantidad de aprobados, reprobados, abandonos, género, edad entre otros).

Todos los procesos mencionados se realizan de acuerdo a requisitos y procedimientos dentro del Curso Prefacultativo, y se encuentran enmarcados dentro de las normas y reglamentos de la Universidad Mayor de San Andrés.

1.3. Planteamiento del Problema

Debido a la gran cantidad de postulantes a una de las Carreras de la Facultad de Ingeniería, el Curso Prefacultativo procura hacer llegar a los estudiantes todas las notificaciones de interés para el correcto desenvolvimiento de los estudiantes en el período que dura todo el curso, así mismo solucionar las interrogantes e inconvenientes que puedan llegar a presentarse.

La problemática radica en el tiempo invertido que le lleva a los postulantes de informarse de todos los comunicados y actividades a efectuarse, obligando a que este proceso sea de manera presencial, generando aglomeraciones debido al número considerado de postulantes y dificultando la tarea de los Administrativos sin brindar comodidad de respuesta a los postulantes.

1.3.1. Formulación del Problema

¿Qué alternativa de comunicación digital y consulta de información se ofrece a todos los postulantes del Curso Prefacultativo de la Facultad de Ingeniería de la UMSA, además del Panel de Información tradicional con el que se cuenta actualmente?

1.4. Objetivos

1.4.1. Objetivo Principal

Diseñar una plataforma digital para teléfonos móviles inteligentes en *Android*, una aplicación que permita a los postulantes recibir información relevante y consultar aspectos relacionados con la gestión del Curso Prefacultativo de la Facultad de Ingeniería de la UMSA.

1.4.2. Objetivos Específicos

- Realizar un diagnóstico sobre el actual proceso de gestión académica de los postulantes en el Curso Prefacultativo para detectar necesidades y requerimientos de los usuarios.
- Establecer una metodología ágil *RAD* enfocada al desarrollo de una plataforma digital para teléfonos móviles inteligentes.

- Diseñar el sistema a nivel físico y lógico empleando el Lenguaje Unificado de Modelamiento o *UML*, brindando una interfaz amigable con facilidad de uso.
- Desarrollar un prototipo de la plataforma digital para teléfonos móviles inteligentes empleando software libre con la seguridad adecuada para la gestión académica de los postulantes del Curso Prefacultativo.
- Realizar pruebas de la plataforma digital de gestión académica para verificar que cumpla con los requisitos establecidos.

1.5. Justificación

1.5.1. Social

La calidad del servicio que brinde la aplicación móvil, beneficia la gestión parcial del Curso Prefacultativo y a la numerosa cantidad de postulantes que se presentan en las fechas establecidas, apoyando al desempeño eficiente de los procesos de la educación y administración de la información, brindando facilidad de uso y comodidad, generando una continua comunicación con los estudiantes y proporcionando información inmediata y confiable.

1.5.2. Tecnológica

En el presente proyecto, la tecnología es el elemento indispensable, tanto software como hardware y que éstas ayudan al buen manejo de la información, beneficiando así a la institución y al estudiantado.

El Curso Prefacultativo cuenta con equipos de computación necesarios, mismos que se encuentran en la red distribuida de la Universidad Mayor de San Andrés, lo que implica que se tiene acceso a la red tanto de internet como intranet, aspecto fundamental para una futura implementación del proyecto. Asimismo, la Facultad de Ingeniería cuenta con equipos servidores donde alojar la aplicación del sistema, ya sea para el uso dentro de la institución como para el acceso mediante la Web, garantizando un acceso inmediato, toda vez que los equipos servidores se encuentran dentro un segmento dedicado de alta velocidad.

1.5.3. Económica

El funcionamiento de la aplicación móvil permite optimizar el control de la información de forma sencilla trayendo ventajas en cuanto a costos, debido a que el sistema se desarrollará en su integridad con software libre y además el Curso Prefacultativo tendrá todos los derechos de propiedad de la herramienta, y por otra parte la institución cuenta con los equipos necesarios para una implementación futura.

1.6. Alcances y Limitaciones

El alcance temático del presente proyecto comprende: la realización de un diagnóstico de los requerimientos en cuanto a los flujos de información del Curso Prefacultativo, análisis, diseño, simulación y pruebas de la aplicación desarrollada en Android. Asimismo, el alcance institucional comprende únicamente al Curso Prefacultativo de la Facultad de Ingeniería. Sin embargo, muchas de las cualidades de este proyecto se pueden aplicar a la gestión académica requerida de otras carreras pertenecientes a la UMSA.

Cabe mencionar que en la actualidad el sitio Web del Curso Prefacultativo ofrece acceso a un Sistema de Información para los postulantes que deseen inscribirse o consultar sus notas de aprobación, para lo cual cuentan con un servidor que tiene instalada la base de datos que gestiona la información personal de los postulantes, como también los Comunicados, Convocatorias, Horarios y Cronogramas. Por tal motivo, para realizar las pruebas de la aplicación, se realizará previamente la simulación del sistema, donde se instalará en una Laptop el entorno *XAMPP* para que un teléfono inteligente se pueda conectar a la base de datos; posteriormente se implementará el sistema en un servidor profesional (VPS) para su producción, mismo que estará en función de la aprobación de la dirección del Curso Prefacultativo.

1.7. Descripción de la Solución Propuesta

En el presente proyecto, se plantea utilizar el modelo de sistema Cliente-Servidor, donde la parte del subsistema cliente será el dispositivo móvil con la aplicación instalada, y la parte del subsistema

servidor será la que se encarga de procesar las peticiones de registro y *login*³ del cliente, además de almacenar la base de datos de los postulantes, como se puede observar en la siguiente figura:

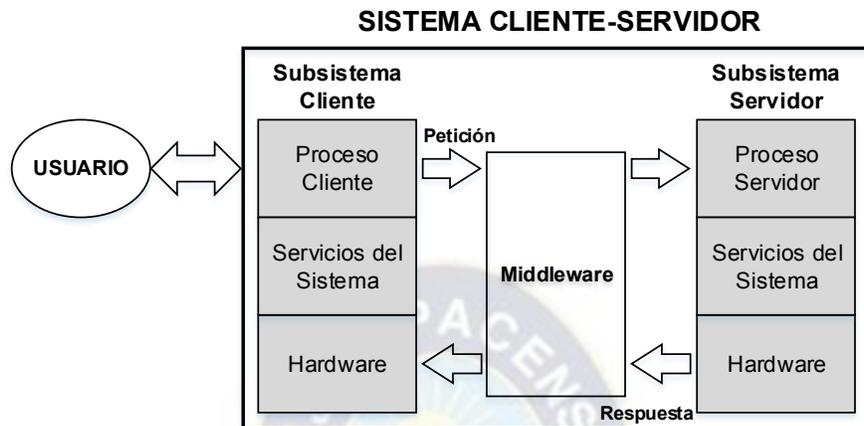


FIGURA 1.1. DIAGRAMA DE BLOQUES DE LA SOLUCIÓN PROPUESTA.

FUENTE: ELABORACIÓN PROPIA, 2021.

1.7.1. Elementos del Modelo Cliente - Servidor

1.7.1.1. Subsistema Cliente

Un subsistema cliente es todo proceso que reclama servicios de otro. Una definición un poco más elaborada podría ser la siguiente: un subsistema cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Se lo conoce con el término *front-end*⁴. Las funciones que lleva a cabo el proceso subsistema cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.

³ En el ámbito de seguridad informática, *login* o *logon* (en español ingresar o entrar) es el proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario. (<https://es.wikipedia.org/wiki/Login>)

⁴ El *Front End* es la parte de una web que conecta e interactúa con los usuarios que la visitan. Es la parte visible, la que muestra el diseño, los contenidos y la que permite a los visitantes navegar por las diferentes páginas mientras lo deseen. (<https://neoattack.com/neowiki/front-end/>)

- Recibir resultados del servidor.
- Formatear resultados.

De este modo el cliente se puede clasificar en:

- **Ciente basado en aplicación de usuario.** Si los datos son de baja interacción y están fuertemente relacionados con la actividad de los usuarios de esos clientes.
- **Ciente basado en lógica de negocio.** Toma datos suministrados por el usuario y/o la base de datos y efectúa los cálculos necesarios según los requerimientos del usuario.

1.7.1.2. Subsistema Servidor

Un subsistema servidor es todo proceso que proporciona un servicio a otros denominados clientes. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término *back-end*⁵. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor se enumeran a continuación:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

⁵ El *Back End* es la parte trasera de cualquier página web. Se trata de todo el conjunto del desarrollo que se encarga de que una página funcione y de que lo haga como lo hace, pero que al mismo tiempo es totalmente invisible para el usuario, que solo ve lo visual y gráfico. (<https://neoattack.com/neowiki/back-end/>)

CAPÍTULO II

MARCO TEÓRICO

En este capítulo se describen los conceptos y herramientas para desarrollar el presente proyecto. Asimismo, se proporcionan las consideraciones de implementación, todo ello bajo una metodología formal de desarrollo como es RAD y el lenguaje de modelado UML. También se realiza un diagnóstico sobre el actual proceso de gestión académica de los postulantes en el Curso Prefacultativo para detectar necesidades y analizar los requerimientos de los usuarios finales, la concepción inicial de la solución, los detalles del diseño propuesto.

2.1. Tecnología Móvil

Actualmente en Bolivia, las comunicaciones móviles han experimentado un enorme crecimiento desarrollándose diversas tecnologías y sistemas para dar servicios de comunicación inalámbrica. La primera empresa en proveer servicios de telefonía móvil en Bolivia fue Tigo, llamada Telecel en ese entonces, en el año 1999, a su vez la Empresa Nacional de Telecomunicaciones (ENTEL) ingresó al mercado en 1996 y la empresa Nuevatel (VIVA) en el año 2000. Los servicios de telefonía móvil y de valor agregado son los más representativos del sector de las telecomunicaciones y generan el mayor volumen de ingresos a las mencionadas compañías.

En los inicios (1973), los primeros terminales ocupaban demasiado tamaño y peso, por lo que su uso se limitaba exclusivamente desde los automóviles. El terminal se instalaba en el habitáculo del vehículo y en el maletero se encontraba equipo de radio de gran peso necesario para mantener las comunicaciones del teléfono celular. Además, el excesivo costo hizo que este servicio no proliferara de manera masiva entre los usuarios de aquel momento.

Las tecnologías de comunicación utilizadas por las redes de datos se denominaron con la letra “G” debido a que hace referencia a la generación designada. 1G, 2G, 3G, 4G (empleada actualmente) y 5G (en un futuro próximo) tienen diferencias sustanciales que cambiaron la manera en que se comunican las personas.

A continuación, se realiza una breve descripción de cada generación de telefonía móvil:

1G (1979-1989): Es un hito que permitió la comunicación entre teléfonos celulares. El gran avance tecnológico significó la movilidad y el hecho de que los equipos no necesitaban una conexión física.

2G (1991-2000): A la comunicación de voz agregó una segunda tecnología que fue la facultad de mandar mensajes cortos (*SMS*, por sus siglas en inglés); y la encriptación de mensajes. Con eso se provee de un sistema seguro y privado.

3G (2001-2008): Significó la llegada del *Internet* móvil en los equipos de usuario. Comenzaron las videollamadas y se habilitó la señal de *TV* en los teléfonos celulares.

4G (2009-2018): Llegó junto con las videoconferencias, la televisión de alta definición (*HDTV*, por sus siglas en inglés); la capacidad de jugar en línea y en general el *Internet* móvil rápido, incluso habilitó la capacidad de llevar el *Internet* a otros dispositivos.

5G (2019): Habilitará la subida y baja de archivos más rápida, el uso de aplicaciones (*apps*) de realidad aumentada y trabajos a distancia que requieren precisión, como la conducción de vehículos a distancia.

Definitivamente la posibilidad de acceso a *Internet* (*Internet* móvil) es el factor que más ha incidido para que los *Smartphones* logren tener el nivel de penetración alcanzado en el mercado, la sensación de conectividad que ofrecen las redes sociales, así como el acceso a mensajería instantánea hace que los usuarios se vean tentados por estos dispositivos. Por otra parte, las numerosas aplicaciones disponibles para teléfonos inteligentes le dan un valor agregado a los usuarios que buscan entretenimiento en sus dispositivos móviles.

2.2. Sistemas Operativos para Teléfonos Móviles

Un sistema operativo (OS, por sus siglas en inglés) móvil es un sistema operativo que controla un dispositivo móvil al igual que los PCs que utilizan *Windows* o *Linux*, los dispositivos móviles poseen sistemas operativos como *Android*, *IOS* (sistema operativo de la multinacional Apple) entre otros. Los sistemas operativos móviles son menos complejos y están más orientados a la conectividad inalámbrica, soportan distintos formatos multimedia para móviles y poseen diferentes maneras de introducir información en ellos. Algunos de los sistemas operativos utilizados en los dispositivos móviles están basados en el modelo de capas.⁶

⁶ dtyoc.com, (2018). “Sistemas Operativos Móviles”. (<https://es.statista.com/estadisticas/600731/cuota-de-mercado-de-sistemas-operativos-para-smartphones-por-pedidos--2020/>)

El sistema operativo móvil de un teléfono o *Tablet* permite la interacción real del usuario a partir de las capacidades del hardware que conforman un equipo. A manera de un traductor, esta plataforma interpreta lo que el usuario desea que el terminal realice y lo ejecuta con mayor inteligencia. A medida que los teléfonos móviles se fueron extendiendo, los sistemas operativos con los que funcionan adquieren mayor importancia. La cuota del mercado de sistemas operativos móviles entre el 2014 y 2020 era el siguiente:

- *Android* 84.6%
- *IOS* 14%
- *Windows Phone* 0.9%
- *BlackBerry OS* 0.4%
- Otros 0.1%

TIPO	CARACTERÍSTICAS	REQUERIMIENTO DE HARDWARE PARA SU INSTALACIÓN
ANDROID	<ul style="list-style-type: none"> • Soporte completo para desarrollar cualquier tipo de aplicación. • Sistema operativo basado en <i>Linux</i>. • Su programación está bajo el <i>framework</i> de <i>JAVA</i>⁷. 	<ul style="list-style-type: none"> • 1 GHz CPU. • 512 MB de RAM. • Pantallas de 3,5".
iOS	<ul style="list-style-type: none"> • Creado por la empresa <i>Apple</i>, derivado de <i>Mac OS X</i>. • Posee la función de <i>Continuity</i> • Incorporación de <i>Quicktype</i>⁸ 	<ul style="list-style-type: none"> • Modelo <i>iPhone 3Gs</i> o superior. • Sistema operativo <i>iOS 3.0</i> o superior. • Instalación en la memoria interna del teléfono (no en tarjeta de memoria)

TABLA 2.1. SISTEMAS OPERATIVOS MÓVILES MÁS COMUNES.

FUENTE: MELKI, 2013⁹.

⁷ Melki Macías, (2013) “Funcionalidad de Sistemas Operativos de Dispositivos Móviles”. (<http://techmi.es/blog/2009/02/02/introduccion-a-las-tecnologias-moviles/>)

⁸ *Ibíd.*

⁹ *Ibíd.*

El sistema operativo *Android* posee la mayor cuota desde enero 2011, con más de la mitad del mercado, experimentando un creciente aumento y en solo dos años (2009 y a comienzos de 2011), pasando a ser el sistema operativo móvil más utilizado.¹⁰

2.3. Sistema Operativo Android

Para comenzar con el desarrollo de aplicaciones en Android es importante conocer como está estructurado este sistema operativo¹¹.

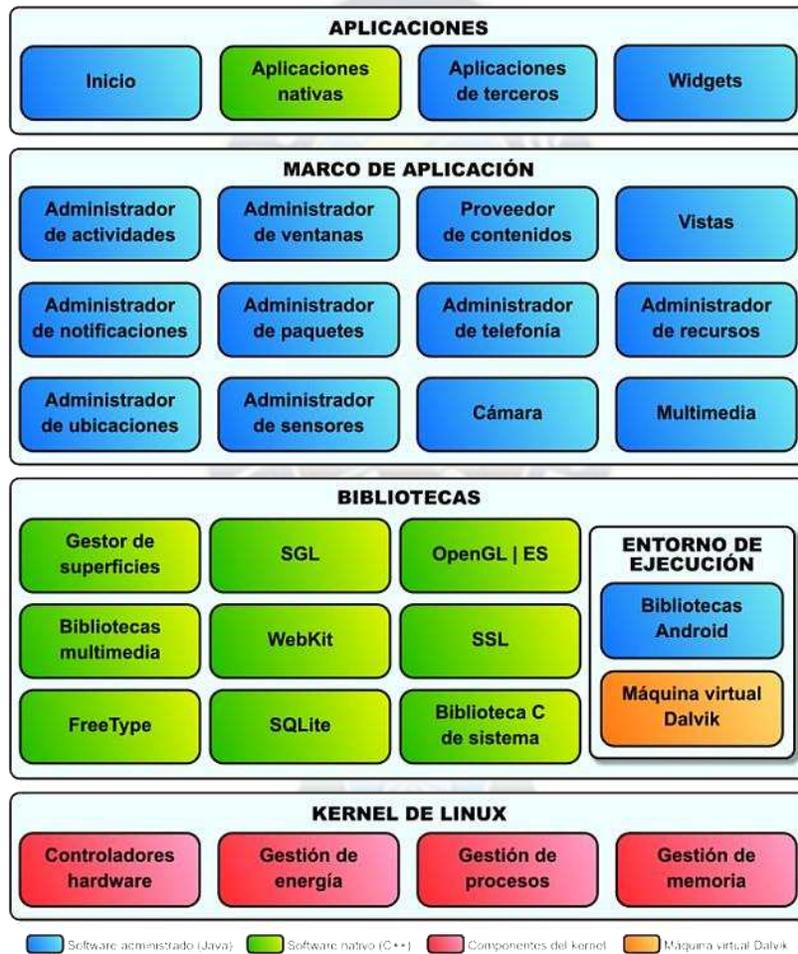


FIGURA 2.1. ARQUITECTURA DEL SISTEMA OPERATIVO ANDROID.

FUENTE: ARQUITECTURA DE ANDROID, 2019¹².

¹⁰ Op. cit.

¹¹ pdfslide.net, (2019) “Arquitectura de Android” (<https://pdfslide.net/documents/arquitectura-de-sistemas-operativos-para-moviles.html>)

¹² Ibid.

La arquitectura de Android está formada por varios niveles o capas, lo que facilita el desarrollo de aplicaciones debido a que permite trabajar con las capas inferiores por medio de las librerías evitando programar a bajo nivel y lograr que los componentes de hardware del dispositivo móvil interactúen con la aplicación.

Como se muestra en la anterior figura, los componentes que forman Android se agrupan en capas. Cada una de estas capas utiliza elementos de la capa inferior para realizar sus funciones. Por este motivo, a este tipo de arquitecturas se le denomina pila. Esta es la pila software de Android¹³.

A continuación, se explican las funciones de las capas más relevantes de abajo hacia arriba¹⁴:

Kernel de Linux. El núcleo del sistema operativo Android es un *Kernel Linux*, similar a que puede incluir cualquier distribución de Linux, como Ubuntu, solo que adaptado a las características del hardware en el que se ejecutara Android (normalmente, un *Smartphone*). Asimismo, proporciona una capa de abstracción para los elementos hardware a los que tienen que acceder las aplicaciones. Esto permite que se pueda acceder a esos componentes sin necesidad de conocer el modelo o características precisas de los que están instalados en cada teléfono. De esta forma, si una aplicación necesita, por ejemplo, la brújula, podrá utilizar la que incluya el teléfono, sea cual sea. Para cada elemento hardware del teléfono existe un controlador (o *driver*) dentro de *kernel* que permite utilizarlo desde el software.

Bibliotecas. La capa que se sitúa sobre el *kernel* la componen las bibliotecas nativas de Android. Estas bibliotecas están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono, tarea que normalmente realiza el fabricante, que también se encarga de instalarlas en el terminal antes de ponerlo a la venta. Su cometido es proporcionar funcionalidad a las aplicaciones, para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez garantizando que se llevan a cabo de la forma más eficiente.

Entorno de ejecución. El entorno de ejecución de Android, aunque se apoya en las bibliotecas enumeradas anteriormente, no se considera una capa en sí mismo, dado que también está formado por bibliotecas. En concreto, las bibliotecas esenciales de Android, que incluyen la mayoría de la funcionalidad de las bibliotecas habituales de Java, así como otras específicas de Android. Las

¹³ Op. cit.

¹⁴ *Ibíd.*

aplicaciones Android se ejecutan cada una en su propia instancia de la máquina virtual *Dalvik*, evitando así interferencias entre ellas, y tiene acceso a todas las bibliotecas mencionadas antes y, a través de ellas, al hardware y al resto de recursos gestionados por el *kernel*.

Marco de Aplicación. La siguiente capa la forman todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones y que, obviamente, se apoyan en las bibliotecas y en el entorno de ejecución que ya hemos detallado. La mayoría de los componentes de esta capa son bibliotecas Java que acceden a los recursos a través de la máquina virtual *Dalvik*¹⁵.

2.4. Aplicaciones Móviles

En la actualidad las aplicaciones como los dispositivos móviles están en el día a día, logrando llegar a ser comunes por la variedad de plataformas que se llega a encontrar en el mercado. Así como los *Smartphones* que tienen como función principal servir como plataforma de aplicaciones que cumplen una función dentro del teléfono, ya sea para comunicarse o para el entretenimiento.

Antecedentes tecnológicos hicieron posible la existencia de aplicaciones que evolucionaron de manera desordenada y sin rumbo fijo. Concluyendo actualmente la existencia de tres tipos de desarrollos para aplicaciones móviles: Aplicaciones Nativas, Aplicaciones *Web*, y Aplicaciones Híbridas. La parte del sub Sistema Cliente del presente proyecto viene siendo una Aplicación Nativa, por lo cual es necesario definir a detalle este tipo de aplicación.

2.4.1. Aplicaciones Nativas

Las aplicaciones nativas son desarrolladas únicamente para que funcionen en una sola plataforma móvil, ya sea *IOS*, *Android*, *Windows Mobile*, entre otros, utilizando herramientas y lenguajes de programación específicas de cada plataforma, para acceder a las funciones de los dispositivos tales como cámara, micrófono, parlantes, geolocalización, acelerómetro, notificaciones, llamadas, almacenamiento, conectividad, entre otros¹⁶.

Las aplicaciones nativas ofrecen la mejor experiencia de usuario, mayor flexibilidad, fluidez,

¹⁵ Op. cit.

¹⁶ Korf, M., & Oksman, E. (2012) "Understanding your Mobile Application Development Options". (http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options)

mejores gráficas que hacen que sean más fáciles de usar en comparación con los otros dos tipos de aplicaciones móviles. Estas aplicaciones se ejecutan en el dispositivo móvil y se las pueden descargar desde las tiendas de las diferentes plataformas móviles: *Google Play*, *Apple App Store*, *Windows Phone Marketplace*, entre otros, y dependiendo de la funcionalidad, algunas de ellas pueden trabajar sin conexión a Internet.¹⁷

La mayoría de las aplicaciones que se pueden descargar desde estas tiendas son de tipo nativas y desarrolladas usando lenguajes de programación, como Java para Android, Objective-C para iOS o C# para *Windows Phone*. Para cada plataforma existe un SDK¹⁸ que provee una serie de APIs¹⁹ que permiten al desarrollador acceder al *hardware*²⁰ del dispositivo.

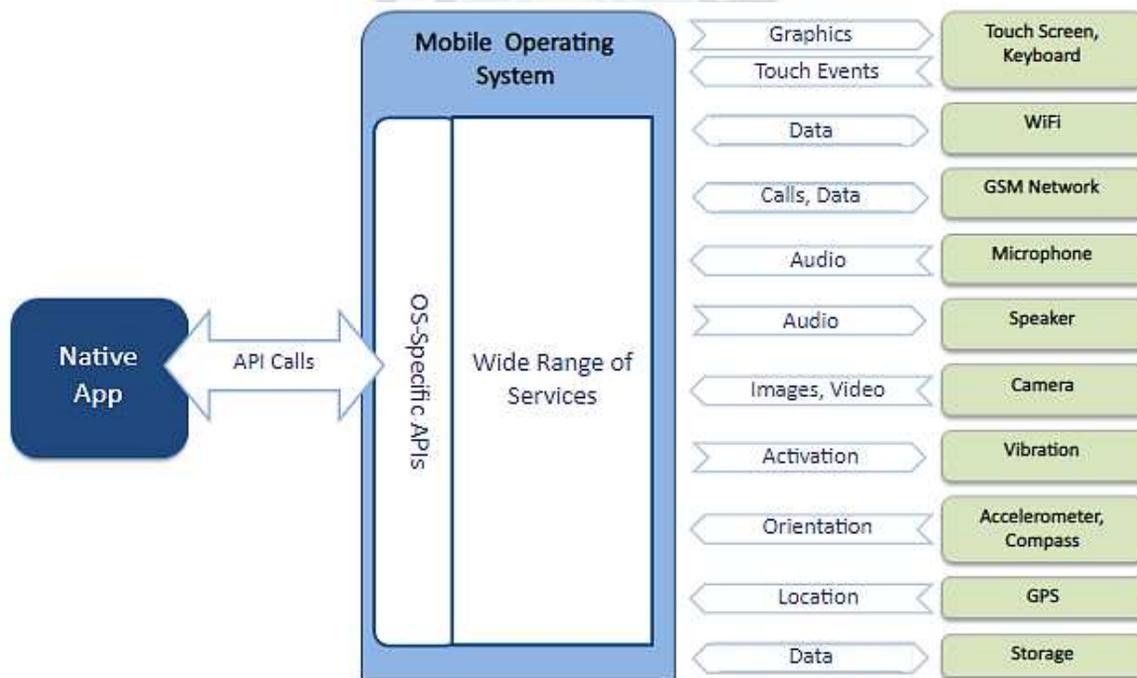


FIGURA 2.2. FUNCIONAMIENTO DE UNA APLICACIÓN NATIVA.

FUENTE: REYNOLDS, 2016.

¹⁷ Gok y Khanna 2016

¹⁸ SDK: *Software Development Kit*, son un conjunto de herramientas para desarrolladores que permiten la creación de aplicaciones para un sistema específico o plataforma.

¹⁹ API: Es el conjunto de procedimientos y funciones que ofrece un paquete de *software* y es usado para la creación de sistemas.

²⁰ *Hardware*: término utilizado para identifica todas las partes tangibles de un dispositivo o sistema informático

Ventajas e Inconvenientes:

- Acceso completo al dispositivo.
- Mejor experiencia del usuario.
- Visibilidad en *App Store*.
- Envío de notificaciones o “avisos” a los usuarios.
- La actualización de la *App* es constante.
- Diferentes habilidades/idiomas/ herramientas para cada plataforma.
- Tienden a ser más caras de desarrollar.
- El código del cliente no es reutilizable entre las diferentes plataformas.

2.4.2. Estructura de una Aplicación Android

Las aplicaciones en Android están basadas en componentes y son los elementos básicos con los que se construyen un proyecto. Una aplicación Android será una combinación de uno o más de estos componentes y se deben declarar en el fichero “*AndroidManifest.xml*” donde se definen todos los componentes de la aplicación, así como los permisos que requiere o los recursos y librerías que utiliza.

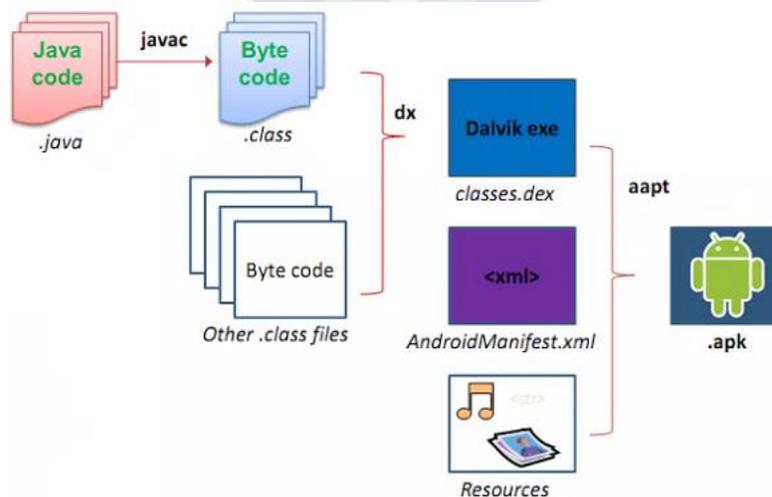


FIGURA 2.3. ESTRUCTURA DE UNA APLICACIÓN ANDROID.

FUENTE: elbinario.net, 2017.

- **Activity:** Es el componente visual de una aplicación, es decir forman la interfaz y se podría decir que son las pantallas o ventanas que ve el usuario.
- **Service:** Son componentes sin interfaz gráfica, se ejecutan en segundo plano y realizan múltiples acciones.
- **Content Provider:** Es la forma que tiene Android de compartir datos entre aplicaciones, compartiendo datos sin la necesidad de dar detalles sobre su almacenamiento.
- **Broadcast Receiver:** Este componente se encarga de detectar y reaccionar a los eventos generales como pueden ser: batería baja, entrada de llamadas, es decir es el encargado de recibir las llamadas que son enviados a cualquier aplicación que este escuchando.

2.5. Plataforma de Desarrollo Android

Android es una solución completa de *software* de código libre (*GNU Linux*) para teléfonos y dispositivos móviles por lo que cuenta con un Kit de desarrollo de *Software* o SDK disponible para todo desarrollador o para alguien que desee experimentar que incluye: el conjunto completo del API soportado por este sistema para crear, probar y depurar aplicaciones para *Android*. Asimismo, es un paquete que engloba un sistema operativo, un “*Runtime*” de ejecución basado en Java, un conjunto de librerías de bajo y medio nivel y un conjunto inicial de aplicaciones destinadas al usuario final. *Android* se distribuye bajo una licencia libre que permite la integración con soluciones de código propietario.



FIGURA 2.4. LOGO DE ANDROID.

FUENTE: GARCÍA, 2012.

Como se había descrito anteriormente, *Android* está basado en *Linux*, disponiendo de un *Kernel* en este sistema y utilizando una máquina virtual sobre este *Kernel* que es la responsable de convertir el código escrito en *Java* de las aplicaciones a código capaz de comprender el *Kernel*. Las aplicaciones para *Android* se escriben y desarrollan en *Java* aunque con unas *APIs* propias por los que programas realizados en *Java* para PC y demás plataformas no son compatibles con *Android*. La cualidad más grande de este sistema operativo es que es de código abierto y se distribuye bajo dos tipos de licencias, una que abarca todo el código de *Kernel* y que es GNU GPL v2, Google también por supuesto tiene otra licencia para el resto de componentes del sistema que se licencia bajo APACHE v2 (JR. Software, 2016).

Sus principales características se detallan a continuación:

- Alta calidad de gráficos y sonido: gran variedad de formatos soportados.
- Las aplicaciones escritas en Java pueden ser compiladas y ejecutadas en la máquina virtual *Dalvik*, máquina virtual diseñada para uso en dispositivos móviles.
- Soporte para hardware adicional (cámaras de video, pantallas táctiles, GPS, acelerómetros, etc.).
- Entorno de desarrollo (emulador, herramientas de depuración, perfiles de memoria y funcionamiento, *plug in* para Eclipse IDE).
- Plataforma realmente abierta al ser basada en Linux y de código libre, Se puede usar y adecuar el sistema sin pagar algún tipo de licencia.
- Portabilidad asegurada: Al desarrollar las aplicaciones en Java y gracias al concepto de máquina virtual, las aplicaciones podrán ser ejecutadas en gran variedad de dispositivos tanto actuales como futuros.
- Las interfaces se hacen en formato XML, lo que permite el uso de una misma interfaz en dispositivos de distintos tamaños de pantallas.
- Gran cantidad de servicios incorporados: reconocimiento y síntesis de voz, localización basada en GPS, potentes bases de datos, etc.
- Nivel de seguridad: Los programas se encuentran separados unos de otros. Cada aplicación dispone distintos tipos de permisos que limitan su tango de actuación.

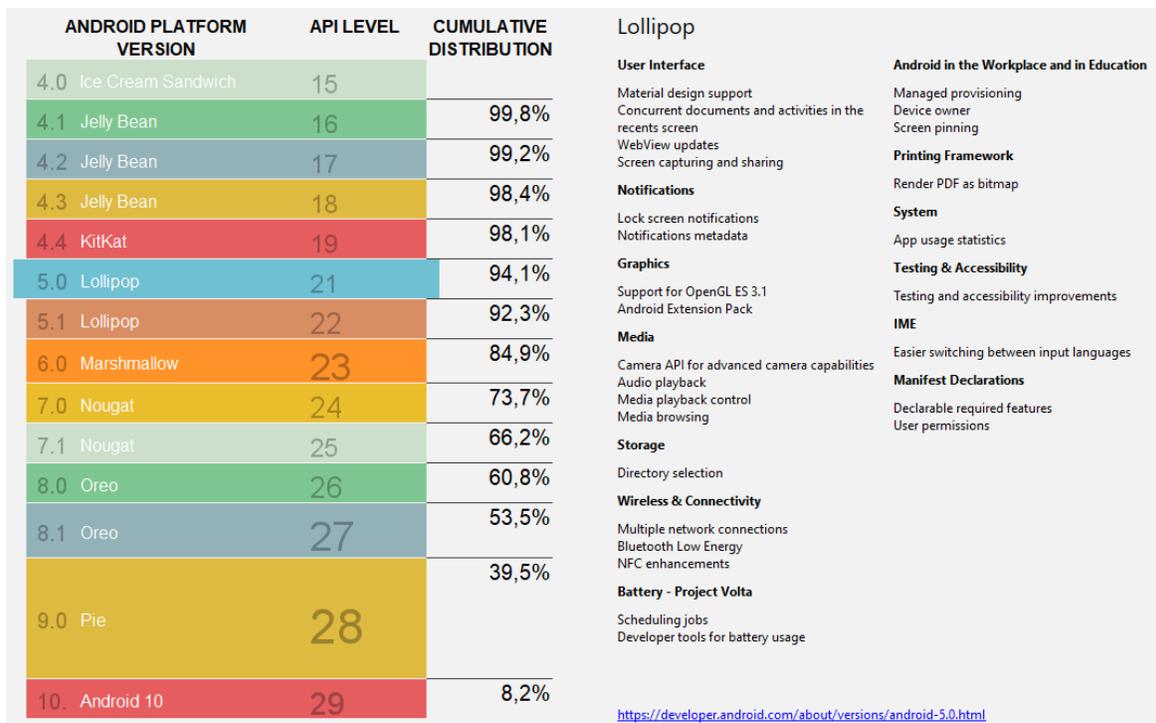


FIGURA 2.5. DISTRIBUCIÓN DE VERSIONES DE ANDROID.

FUENTE: DEVENLOPER, 2021.

2.6. Metodologías de Desarrollo Ágil

Los motivos para desarrollar la Metodología Ágil, fue los continuos errores que conllevaban las metodologías tradicionales, como ser: Alto número de proyectos que se retrasan o fracasan y la baja calidad del software, o bien si el proyecto concluía no se implementaba, porque los requisitos cambiaban en el camino, y las metodologías tradicionales no aceptan el cambio, o la retroalimentación. (Fernández, 2014)

Es por eso que, en febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace en término ágil aplicado al desarrollo de software. El objetivo fue esbozar los valores y principios que deberían permitir a los equipos a desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. La filosofía “ágil” se encuentra plasmada en el “Manifiesto Ágil”, que valora: (Fernández, 2014)

- Al individuo y las iteraciones del equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funciona más que conseguir una buena documentación.

- La colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrechamente un plan. (Fernández, 2014)

En la siguiente tabla, se muestran las diferencias entre las metodologías ágiles y no ágiles:

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

TABLA 2.2. DIFERENCIAS ENTRE METODOLOGÍAS ÁGILES Y NO ÁGILES.

FUENTE: FERNÁNDEZ, 2014.

2.6.1. Metodología de Desarrollo Ágil RAD

En el ámbito educativo universitario de pregrado, debido a los espacios cortos de tiempo, al momento de desarrollar un producto de software de mediana complejidad, como es este caso, se

puede emplear la metodología de Desarrollo Ágil de Aplicación (RAD, *Rapid Application Development*) por la necesidad de construir rápido una aplicación y a la vez que una misma persona hace de analista, diseñador y programador. (Carrillo, 2012).

RAD es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades de Ingeniería de Software Asistida por Computadora (CASE, *Computer Aided Software Engineering*). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución. (Carrillo, 2012)

Este enfoque se caracteriza por el uso de prototipos, que no son más que una aproximación o representación del sistema, que permiten a los usuarios visualizar y responder en función de la implementación realizada. Implementado en primera instancia prototipos, los analistas pueden mostrar a los usuarios las entradas, los estadios intermedios y las salidas del sistema, de una manera que los usuarios entienden. La metodología RAD es un ejemplo que utiliza prototipado. (Carrillo, 2012)

2.6.1.1. Fases de la Metodología RAD

- **Modelado de Gestión:** El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la proceso? (Carrillo, 2012).
- **Modelado de Datos:** El flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos (Carrillo, 2012).
- **Modelado de Proceso:** Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos (Carrillo, 2012).
- **Generación de Aplicaciones:** El RAD asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso

RAD trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software (Carrillo, 2012).

- **Pruebas y Entrega:** Como el proceso RAD enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo (Carrillo, 2012).

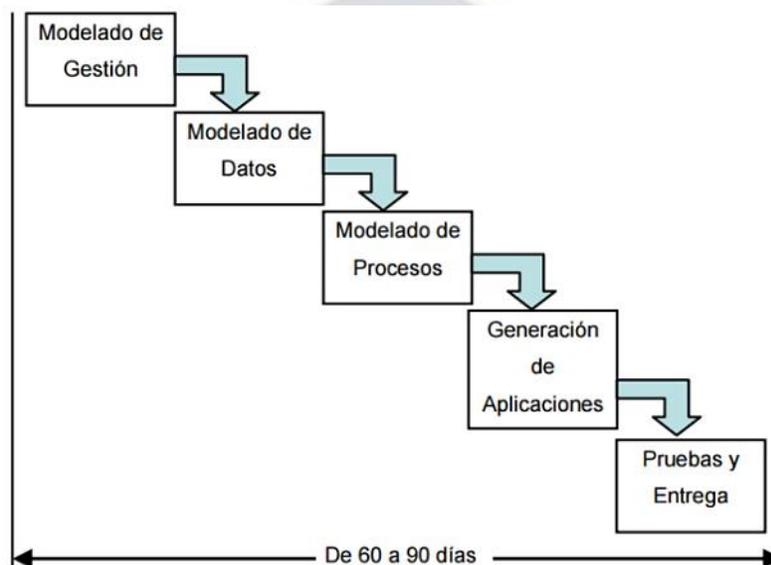


FIGURA 2.6. MODELO RAD.

FUENTE: sisinblog, 2018.

2.6.1.2. Ventajas

Las ventajas de la metodología RAD son las siguientes: (Carrillo, 2012)

- Comprar puede ahorrar dinero en comparación con construir.
- Los entregables pueden ser fácilmente trasladados a otra plataforma.
- El desarrollo se realiza a un nivel de abstracción mayor.
- Visibilidad temprana.
- Mayor flexibilidad.
- Menor codificación manual.

- Mayor involucramiento de los usuarios.
- Posiblemente menos fallas.
- Posiblemente menor costo.
- Ciclos de desarrollo más pequeños.
- Interfaz gráfica estándar.

2.7. Lenguaje UML

El Lenguaje Unificado de Modelado (UML, *Unified Modelling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Grupo de Administración de Objetos (OMG, *Object Management Group*). Es un lenguaje gráfico que permite visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. Es importante remarcar que UML es un “lenguaje de modelado” para especificar o para describir métodos o procesos (Hernández, 2001).

Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como RAD), pero no especifica en sí mismo qué metodología o proceso usar (Hernández, 2001).

- **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos,

como flujos de trabajo (*workflow*) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware. UML es solo un lenguaje y puede ser utilizado como parte de un método de desarrollo de software guiado por casos, centrado en la arquitectura, iterativo e incremental. Un modelo UML está compuesto por tres clases de bloques de construcción: (Hernández, 2001)

2.7.1. Diagramas UML

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas: (Hernández, 2001)

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de actividades.
- Diagrama de secuencia.
- Diagrama de objetos.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de componentes.
- Diagrama de despliegue.

Los diagramas más utilizados son los de casos de uso, clases y actividades.

2.7.1.1. Diagrama de Casos de Uso

Describe lo que hace un sistema desde el punto de vista de un observador externo, debido a esto, un diagrama de este tipo generalmente es de los más sencillos de interpretar en UML, ya que su razón de ser se concentra en un QUE hace el sistema, a diferencia de otros diagramas UML que intentan dar respuesta a un COMO logra su comportamiento el sistema. Un caso de uso es empleado con más frecuencia para:

1. **Determinación de Requerimientos:** Por lo general nuevos requerimientos de sistema generan nuevos usos-casos, conforme es analizado y diseñado el sistema.

2. **Comunicación con el Cliente:** Debido a la sencillez de este tipo de diagramas, son fáciles de emplear para comunicarse con el cliente final del proyecto.
3. **Generación de Pruebas de Sistema:** A través de los diagramas uso-caso se pueden generar una serie de pruebas de sistema.

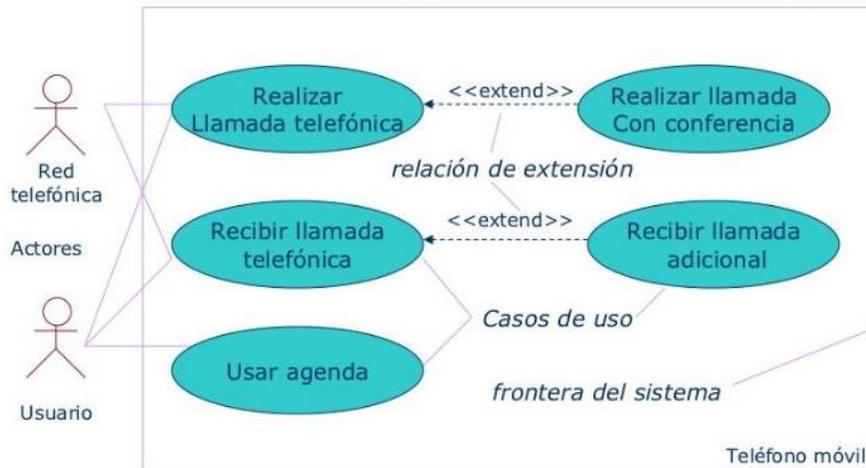


FIGURA 2.7. EJEMPLO DE DIAGRAMA DE CASO DE USO.

FUENTE: FUNDAMENTOS DE INGENIERÍA DE SOFTWARE.

Para modelar el contexto de un sistema,

- ✓ Identificar a los actores que rodean el sistema y los grupos interactúan con los sistemas, y las funciones secundarias de administración y mantenimiento.
- ✓ Llenar un diagrama de casos de uso con estos actores y especificar las vías de comunicación de cada actor para los casos de uso del sistema.

Para modelar los requisitos de un sistema,

- ✓ Establecer el contexto del sistema mediante la identificación de los actores que lo rodean. Para cada actor, tenga en cuenta el comportamiento que cada uno espera o requiere que el sistema.
- ✓ Por cada caso de uso en el diagrama, identificar su flujo de eventos y su flujo excepcional de eventos.

Los diagramas de casos de uso están compuestos por:

- **Actor:** Un actor representa quien o que inicia una acción dentro del sistema, en otras palabras, es simplemente un rol que es llevado a cabo por una persona o cosa. Un Actor en un diagrama Uso-Caso es representado por una figura en forma de persona.

- **Uso-Caso:** El uso-caso en sí representado por un óvalo que describe la funcionalidad a grosso modo que se requiere por el sistema.
- **Comunicación:** Este elemento representa la relación que existe entre un Uso-Caso y un Actor, dicho elemento es representado simplemente por una línea recta que se extiende de la figura del actor hacia el óvalo de uso-caso.
- **Límite del Sistema (*System Boundary*):** Empleado para delimitar los límites del sistema, y representado por un rectángulo con color de fondo distintivo.
- **Generalización:** Una generalización indica que un uso-caso (óvalo) es un caso especial de otro caso, en otros términos, representa una relación padre-hijo, donde el hijo puede ser suplido directamente por el padre en cualquier momento. Este elemento es representado por una línea con flecha que se extiende del uso-caso hijo hacia el uso caso padre (general).
- **Inclusión:** Una inclusión es utilizada para indicar que un uso-caso (óvalo) depende de otro caso, dicho de otra manera, significa que la funcionalidad de determinado caso se requiere para realizar las tareas de otro. Este elemento es representado por una línea punteada con flecha y comentario <<include>> que se extiende del uso-caso base hacia el uso caso de inclusión.
- **Extensión:** Una extensión representa una variación de un uso-caso a otro, aunque similar a una generalización no implica que los usos-casos dependen uno del otro. Este elemento es representado por una línea punteada con flecha y comentario <<extend>> que origina del uso-caso base hacia el uso caso de extensión.

2.7.1.2. Diagrama de Clases

El diagrama de clases muestra un conjunto de clases, interfaces y sus relaciones. Éste es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos. Se pueden usar los diagramas de clases para modelar la vista de diseño estático de un sistema. Los diagramas de clases son también la base para diseñar el diagrama de componentes. Los diagramas de clases son importantes para visualizar, especificar y documentar modelos estructurales. En la figura 2.6 se muestran las clases globales, sus atributos y las relaciones de una posible solución al problema de registro de historiales clínicos (Hernández, 2001).

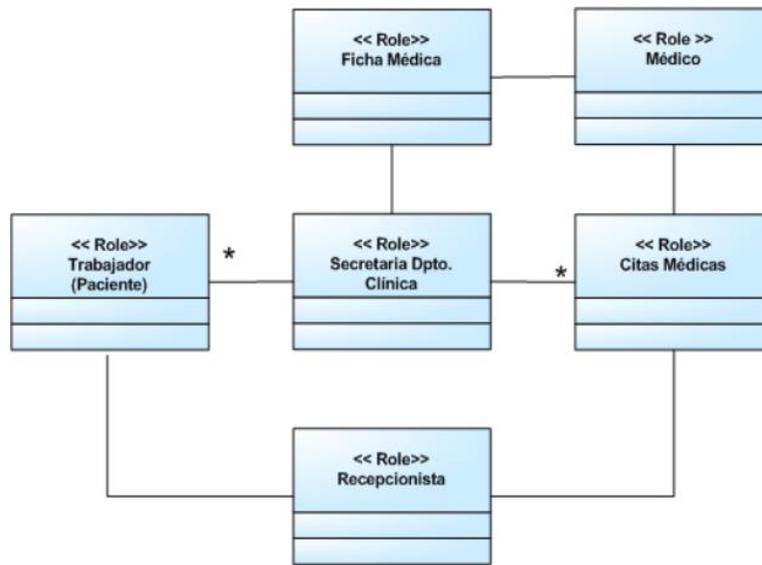


FIGURA 2.8. EJEMPLO DE DIAGRAMA DE CASO DE CLASES.
 FUENTE: APUNTES DE ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN.

2.7.1.3. Diagrama de Actividades

Un diagrama de actividades es esencialmente un diagrama de flujo, que muestra el flujo de control de la actividad a la actividad. Puede utilizar los diagramas de actividad para modelar los aspectos dinámicos de un sistema.

Con un diagrama de actividades, también se puede modelar el flujo de un objeto mientras se mueve de un estado a otro en diferentes puntos en el flujo de control. Diagramas de actividad pueden estar solos para visualizar, especificar, construir y documentar la dinámica de una sociedad de objetos, o pueden ser utilizados para modelar el flujo de control de una operación.

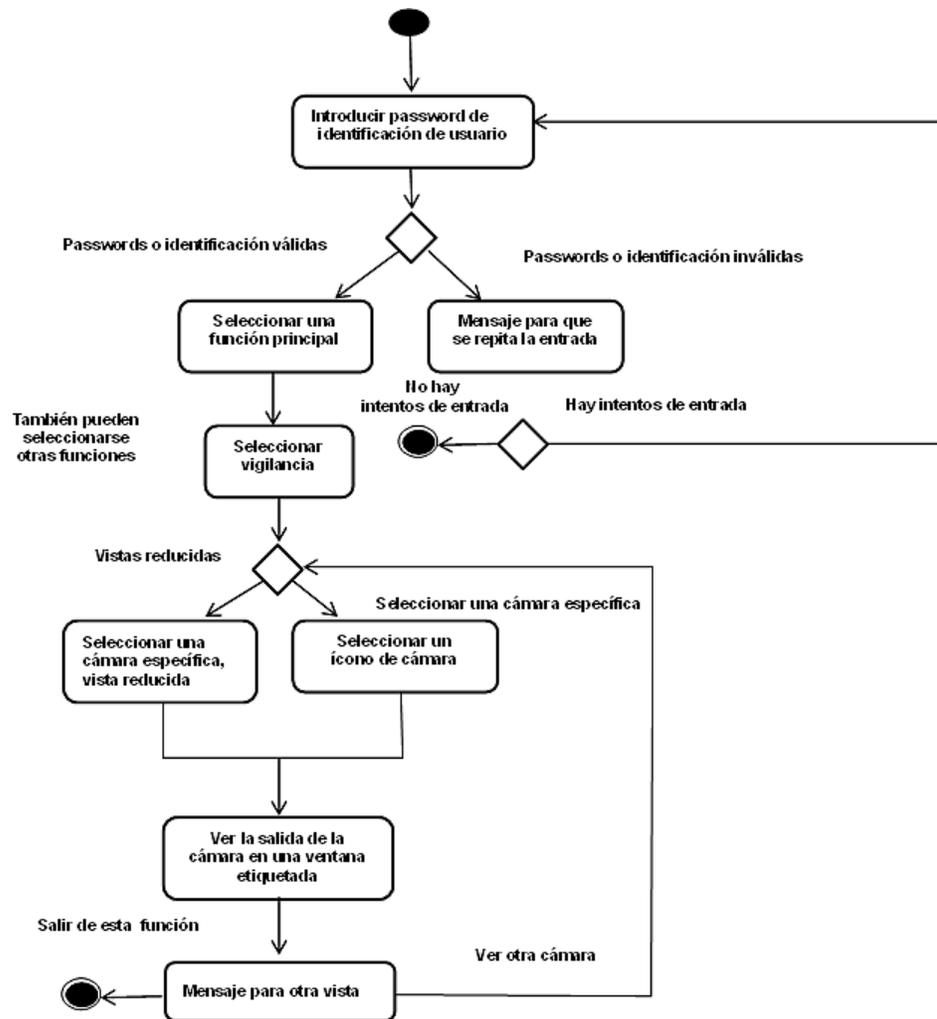


FIGURA 2.9. EJEMPLO DE DIAGRAMA DE ACTIVIDADES.
FUENTE: FUNDAMENTOS DE INGENIERÍA DE SOFTWARE.

Los diagramas de actividades están compuestos por:

- **Inicio:** El inicio de un diagrama de actividad es representado por un círculo de color negro sólido.
- **Actividad:** Una actividad representa la acción que será realizada por el sistema la cual es representada dentro de un óvalo.
- **Transición:** Una transición ocurre cuando se lleva a cabo el cambio de una actividad a otra, la transición es representada simplemente por una línea con una flecha en su terminación para indicar dirección.

- **Ramificación (*Branch*):** Una ramificación ocurre cuando existe la posibilidad que ocurra más de una transición (resultado) al terminar determinada actividad. Este elemento es representado a través de un rombo.
- **Unión (*Merge*):** Una unión ocurre al fusionar dos a más transiciones en una sola transición o actividad. Este elemento también es representado a través de un rombo.
- **Expresiones Resguardadas (*Guard Expressions*):** Una expresión resguardada es utilizada para indicar una descripción explícita acerca de una transición. Este tipo de expresión es representada mediante corchetes ([...]) y es colocada sobre la línea de transición.
- **Fork:** Un *fork* representa una necesidad de ramificar una transición en más de una posibilidad. Aunque similar a una ramificación (*branch*) la diferencia radica en que un fork representa más de una ramificación obligada, esto es, la actividad debe proceder por ambos o más caminos, mientras que una ramificación (*branch*) representa una transición u otra para la actividad (como una condicional). Un fork es representado por una línea negra sólida, perpendicular a las líneas de transición.
- **Join:** Un *join* ocurre al fusionar dos o más transiciones provenientes de un fork, y es empleado para dichas transiciones en una sola, tal y como ocurría antes de un fork. Un fork es representado por una línea negra sólida, perpendicular a las líneas de transición.
- **Fin:** El fin de un diagrama de actividad es representado por un círculo, con otro círculo concéntrico de color negro sólido.
- **Canales (*Swimlanes*):** En determinadas ocasiones ocurre que un diagrama de actividad se expanda a lo largo de más de una entidad o actor, cuando esto ocurre el diagrama de actividad es patrocinada de canales (*swimlanes*), donde canal representa la entidad o actor que lleva a cabo la actividad.

2.8. Calidad y Seguridad de Software

2.8.1. Calidad de Software (Norma ISO-9126)

La calidad de software puede ser evaluado midiendo atributos internos (típicamente, medidas estáticas de productos intermedios), o puede ser evaluada midiendo atributos externos (típicamente, medidas de comportamiento del código cuando se está ejecutando).

Característica	Pregunta Central
Funcionalidad	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el qué, ...?
Confiabilidad	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
Usabilidad	¿El software es fácil de usar y de aprender?
Eficiencia	¿Es rápido y minimalista en cuanto al uso de recursos?
Mantenibilidad	¿Es fácil de modificar y verificar?
Portabilidad	¿Es fácil de transferir de un ambiente a otro?

TABLA 2.3. CARACTERÍSTICAS DE LA NORMA ISO-9126.

FUENTE: ABUD, 2012.

La ISO, bajo la norma ISO-9126²¹, ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 con el nombre de “*Information technology-Software product evaluation: Quality characteristics and guidelines for their use*”, en el cual se establecen las características de calidad para productos de software que se describen a través de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de sub características que permiten profundizar en la evaluación de la calidad de productos de software.

2.8.1.1. Características Propuestas por la ISO-9126

A continuación, se detalla cada una de las características que establece la norma ISO-9126.

a) FUNCIONALIDAD.

En este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. Para este propósito se establecen los siguientes atributos:

²¹ <http://www.nacionmulticultural.unam.mx/empresasindigenas/docs/2094.pdf>

- **Adecuación.** Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición;
- **Exactitud.** Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado;
- **Interoperabilidad.** Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
- **Conformidad.** Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
- **Seguridad.** Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.

b) CONFIABILIDAD.

Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. Las sub características que la norma sugiere son:

- **Nivel de Madurez.** Permite medir la frecuencia de falla por errores en el software;
- **Tolerancia a fallas.** Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica;
- **Recuperación.** Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.

c) USABILIDAD.

Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.

- **Comprensibilidad.** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software;
- **Facilidad de Aprender.** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación;

- **Operabilidad.** Agrupa los conceptos que evalúan la operación y el control del sistema.

d) EFICIENCIA.

Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados. Los aspectos a evaluar son:

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

e) MANTENIBILIDAD.

Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad. En este caso, se tienen los siguientes factores:

- **Capacidad de análisis.** Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificarlas partes que deberán ser modificadas;
- **Capacidad de modificación.** Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente;
- **Estabilidad.** Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al software;
- **Facilidad de Prueba.** Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.

f) PORTABILIDAD.

En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos:

- **Adaptabilidad.** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones;

- **Facilidad de instalación.** Es el esfuerzo necesario para instalar el software en un ambiente determinado;
- **Conformidad.** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad;
- **Capacidad de reemplazo.** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.

2.8.2. Seguridad de la Información

La seguridad informática relaciona a diversas técnicas, aplicaciones y dispositivos encargados de asegurar la integridad y privacidad de la información de un sistema informático y sus usuarios. Existen dos tipos de seguridad con respecto a la naturaleza de la amenaza:

- **Seguridad lógica:** aplicaciones para seguridad, herramientas informáticas, etc.
- **Seguridad física:** mantenimiento eléctrico, anti-incendios, humedad, etc.

Amenazas a la seguridad de un sistema informático o computadora:

- **Programas malignos:** Virus, espías, troyanos, gusanos, *phishing*, *spamming*, etc.
- **Siniestros:** Robos, incendio, humedad, etc. pueden provocar pérdida de información.
- **Intrusos:** Piratas informáticos pueden acceder remotamente (si está conectado a una red) o físicamente a un sistema para provocar daños.
- **Operadores:** Los propios operadores de un sistema pueden debilitar y ser amenaza a la seguridad de un sistema no sólo por boicot, también por falta de capacitación o de interés.

Dentro de la seguridad informática, la seguridad lógica hace referencia a la aplicación de mecanismos y barreras para mantener el resguardo y la integridad de la información dentro de un sistema informático. La seguridad lógica se complementa seguridad física.

La seguridad lógica de un sistema informático incluye:

- Restringir al acceso a programas y archivos mediante claves y/o encriptación.
- Asignar las limitaciones correspondientes a cada usuario del sistema informático. Esto significa, no darle más privilegios extras a un usuario, sino sólo los que necesita para realizar su trabajo.

Dentro de la seguridad informática, la seguridad física hace referencia a procesos para proteger el hardware de amenazas físicas. Básicamente, las amenazas físicas que pueden poner en riesgo un sistema informático son:

- Desastres naturales, incendios accidentales, humedad e inundaciones.
- Amenazas ocasionadas involuntariamente por personas.
- Acciones hostiles deliberadas como robo, fraude o sabotaje.

2.8.2.1. Criptografía

La criptografía se refiere a la capacidad de mantener un documento electrónico (o de cualquier otro tipo) inaccesible a todos, excepto a determinadas personas, la autenticidad es la capacidad para determinar si una persona determinada pertenece o no a la lista de personas autorizadas para acceder el documento o no.

2.8.2.2. Encriptación de Datos

En los procesos de almacenamiento y transmisión de la información normalmente aparece el problema de su seguridad. En el almacenamiento, el peligro lo representa el robo del soporte del mensaje o simplemente el acceso no autorizado a esa información, mientras que en las transmisiones lo es la intervención del canal. Hay dos tipos básicos de amenazas de seguridad contra un mensaje secreto enviado o recibido: la pasiva que es su robo o el acceso no autorizado y la activa como es su alteración. La protección de la información se lleva a cabo variando su forma, se llama cifrado (o transformación criptográfica) a una transformación del texto original (llamado también texto inicial o texto claro) que lo convierten en el llamado texto cifrado o criptograma. Análogamente, se llama descifrado a la transformación que permite recuperar el texto original a partir del texto cifrado.

Cuando un usuario quiere acceder a un servicio de un computador, siempre surge la siguiente cuestión: **¿Cómo puede estar seguro el servidor de que el cliente no esté utilizando una identidad falsa?** La solución clásica a este problema pasa por el empleo de *password* (palabras de paso). Claramente, la seguridad de tal esquema depende completamente de habilidad para mantener en secreto las *password*. Los principales puntos débiles de este esquema son: la existencia del

listado de *password* en el computador y la amenaza de escucha. En este esquema mínimamente existe un super usuario que conoce tal información, esta amenaza puede resolverse en la práctica, el computador en realidad no necesita conocer todos los *passwords*, solo debe ser capaz de validarlas y esto puede hacerse mediante una función trampa. Basta con que el computador tenga almacenada las imágenes de los *passwords* según una función trampa. Ni siquiera hace falta que esta función se mantenga en secreto, pero resulta débil si los usuarios usan *password* fáciles de recordar, por lo que se recomienda precaución al momento de elegir un *password*.

2.9. Emulador Genymotion

Genymotion es un emulador de Android que aprovecha la arquitectura x86 para ejecutar de forma fluida y rápida distintos dispositivos *Android*. Olvidando la lentitud del emulador nativo de Android podemos ejecutar todo tipo de aplicaciones y juegos en nuestro Windows, Mac o Linux. Uno de los principales usos de *Genymotion* es facilitar el desarrollo de aplicaciones *Android*. Casi más de 900.000 usuarios registrados usan sus máquinas virtuales para crear aplicaciones usando Eclipse o IntelliJ. Se integran perfectamente con el *ADB* (*Android Debug Bridge*), línea de comandos y los diferentes entornos de desarrollo. Podemos lanzar en apenas un par de segundos las aplicaciones conectadas al *ADT* (*Android Development Tool*) desde el *IDE* (*Integrated Development Environment*) para testearla. Mucho tiempo ahorrado.

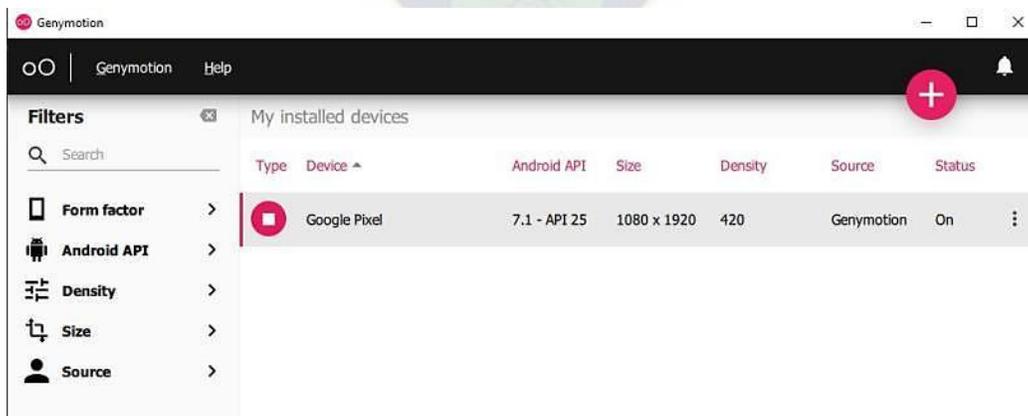


FIGURA 2.10. PANTALLA DE CONFIGURACIÓN DE DISPOSITIVOS DE GENYMOTION.

FUENTE: xatakandroid, 2014.

2.10. Android Studio

Android Studio es un Entorno de Desarrollo Integrado (*IDE*), basado en *IntelliJ IDEA* de la compañía *JetBrains*, que proporciona varias mejoras con respecto al *plugin ADT (Android Developer Tools)* para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma.

Fue presentado por Google el 16 de mayo del 2013 en el congreso de desarrolladores Google I/O, con el objetivo de crear un entorno dedicado en exclusiva a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción. Se trata pues de una alternativa real a Eclipse, el IDE recomendado por Google hasta la fecha, pero que presentaba problemas debido a su lentitud en el desarrollo de versiones que solucionarían las carencias actuales (es indispensable recordar que Eclipse es una plataforma de desarrollo, diseñada para ser extendida a través de *plugins*).

Android Studio se ha mantenido durante todo este tiempo en versión beta, pero desde el 8 de diciembre de 2014, en que se liberó la versión estable de Android Studio 1.0, Google ha pasado a recomendarlo como el *IDE* para desarrollar aplicaciones para su sistema operativo, dejando el *plugin ADT* para Eclipse de estar en desarrollo activo.



FIGURA 2.11. LOGO DE ANDROID ESTUDIO.

FUENTE: LÁZARO, 2012.

2.10.1. Características de Android Studio

Las principales características que incluye Android Studio son:

- Soporte para programar aplicaciones para Android *Wear* (Sistema Operativo para dispositivos corporales como por ejemplo un reloj).

- Herramientas *Lint* (detecta código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones.
- Utiliza *ProGuard* para optimizar y reducir el código del proyecto al exportar a *APK* (muy útil para dispositivos de gama baja con limitaciones de memoria interna).
- Integración de la herramienta *Gradle* encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de *testing*, compilación o empaquetado.
- Nuevo diseño del editor con soporte para la edición de temas.
- Nueva interfaz específica para el desarrollo en *Android*.
- Permite la importación de proyectos realizados en el entorno Eclipse, que a diferencia de *Android Studio (Gradle)* utiliza *ANT*.
- Posibilita el control de versiones accediendo a un repositorio desde el que poder descargar *Mercurial, Git, Github* o *Subversion*.
- Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.
- Vista previa en diferentes dispositivos y resoluciones.
- Integración con *Google Cloud Platform*, para el acceso a los diferentes servicios que proporciona *Google* en la nube.
- Editor de diseño que muestra una vista previa de los cambios realizados directamente en el archivo *XML*.

2.10.2. Ventajas de Android Estudio

Las ventajas del uso Android Studio son:

- Android Studio ha pasado a ser el entorno recomendado para el desarrollo de aplicaciones en *Android*, al tratarse de un *IDE* oficial de *Google* en colaboración con *JetBrains* (compañía de desarrollo software especializada en diseño de IDEs).
- *Android Studio* permite la creación de nuevos módulos dentro de un mismo proyecto, sin necesidad de estar cambiando de espacio de trabajo para el manejo de proyectos, algo habitual en Eclipse.
- Con la simple descarga de *Android Studio* se disponen de todas las herramientas necesarias para el desarrollo de aplicaciones para la plataforma *Android*.

- Su nueva forma de construir los paquetes *.apk*, mediante el uso de *Gradle*, proporciona una serie de ventajas más acorde a un proyecto *Java*:
 - ✓ Facilita la distribución de código, y por lo tanto el trabajo en equipo.
 - ✓ Reutilización de código y recursos.
 - ✓ Permite compilar desde línea de comandos, para aquellas situaciones en las que no esté disponible un entorno de desarrollo.
 - ✓ Mayor facilidad para la creación de diferentes versiones de la misma aplicación, que proporciona numerosas ventajas como puede ser la creación de una versión de pago y otra gratuita, o por ejemplo diferentes dispositivos o almacén de datos.

2.10.3. Desventajas de Android Studio

Las desventajas del uso Android Studio son:

- Aunque ya se ha lanzado la primera versión estable, la v1.0, al estar en una fase inicial, siempre es susceptible de introducirse más cambios que puedan provocar inestabilidad entre proyectos de diferentes versiones.
- Curva de aprendizaje más lenta para nuevos desarrolladores de Android.
- El sistema de construcción de proyectos *Gradle* puede resultar complicado inicialmente.
- En comparativa con Eclipse, menor número de *plugins*.

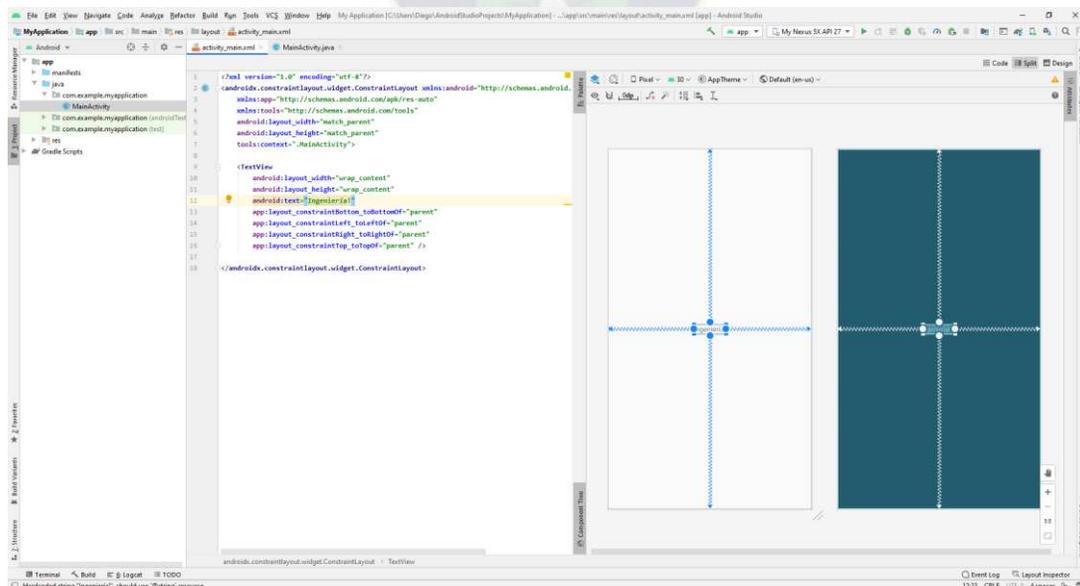


FIGURA 2.12. ENTORNO DE DESARROLLO ANDROID STUDIO.

FUENTE: CAPTURA DE PANTALLA, 2021.

2.11. BASES DE DATOS

Una base de datos es una colección compartida de datos lógicamente relacionados, junto con una descripción de esos datos, que están diseñados para satisfacer las necesidades de una organización y que son gestionados (para lectura/escritura) por un sistema gestor de base de datos (SGBD) (Galindo, 2015).

Así mismo, un sistema de base de datos es básicamente un sistema computarizado para guardar registros; es decir es un sistema computarizado cuya finalidad general es almacenar información, permitir a los usuarios recuperar y actualizar esa información con base a peticiones (Date, 2016).

En efecto, tomando en cuenta las definiciones anteriores se considera como una base de datos al conjunto de datos relacionados entre sí, donde se registran datos con un significado implícito. Sin embargo, la base de datos debe tener una definición que contenga los tipos de datos, las estructuras y restricciones de los datos que se almacenaran en ella. Por ello, se consideran un conjunto de datos no redundantes, almacenados en un soporte informático, organizado de forma independiente de su utilización y accesible simultáneamente por distintos usuarios o aplicaciones.

2.12. XAMPP

XAMPP es un servidor independiente de plataforma de código libre. Permite instalar de forma sencilla Apache en un ordenador, sin importar el sistema operativo que sea posea (*Linux*, *Windows*, *MAC* o *Solaris*). Es una plataforma gratuita que incluye servidores de bases de datos como *MySQL* y *SQLite* con sus respectivos gestores *phpMyAdmin* y *phpSQLiteAdmin*. Incorpora también el intérprete de *PHP*, el intérprete de *Perl*, servidores de *FTP* como *ProFTPD* o *FileZilla FTP Server*, etc. entre otros.

Una de las ventajas de emplear *XAMPP* es que su instalación y utilización es de lo más sencilla. *XAMPP* es una herramienta de desarrollo que permite realizar pruebas de proyectos de *software* (páginas Web o programación por ejemplo) en un ordenador sin necesidad de tener acceso a Internet. El nombre *XAMPP* es un acrónimo compuesto por las iniciales de los programas que lo constituyen como ser:

- **X:** Significa que funciona en cualquiera de los diferentes sistemas operativos existentes hoy en el mercado.
- **Apache:** El servidor web de código abierto es la aplicación usado globalmente para la entrega de contenidos *web*. Las aplicaciones del servidor son ofrecidas como software libre por la *Apache Software Foundation*.
- **MySQL/MariaDB:** *XAMPP* cuenta con uno de los sistemas relacionales de gestión de bases de datos más populares del mundo. En combinación con el servidor web Apache y el lenguaje *PHP*, *MySQL* sirve para el almacenamiento de datos para servicios web. En las versiones actuales de *XAMPP* esta base de datos se ha sustituido por *MariaDB*.
- **PHP:** Es un lenguaje de programación de código de lado del servidor que permite crear páginas *web* o aplicaciones dinámicas. Es independiente de plataforma y soporta varios sistemas de bases de datos.
- **Perl:** Este lenguaje de programación se usa en la administración del sistema, en el desarrollo *web* y en la programación de red. También permite programar aplicaciones web dinámicas.

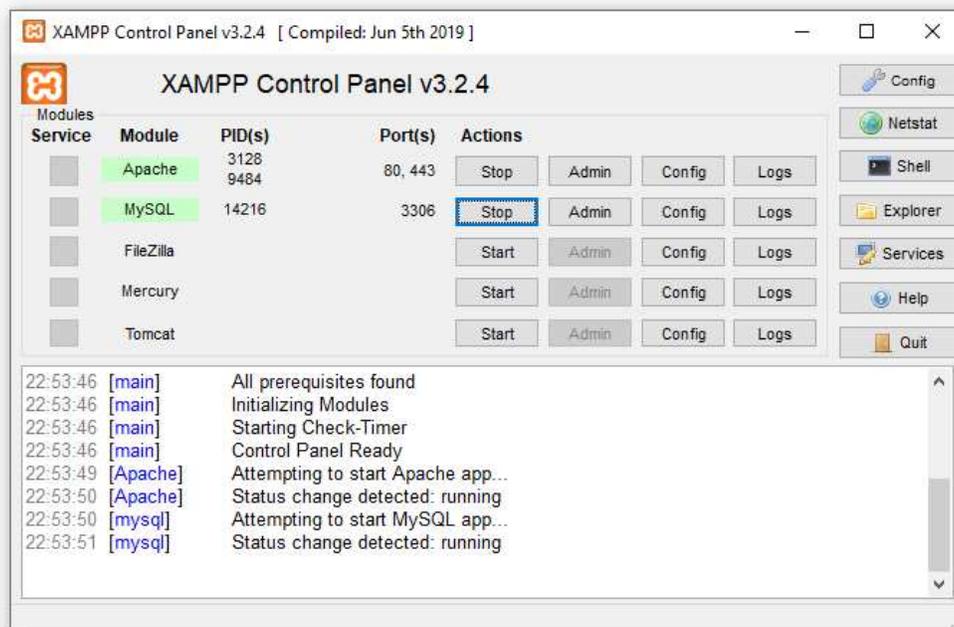


FIGURA 2.13. PANEL DE CONTROL XAMPP.

FUENTE: CAPTURA DE PANTALLA, 2021.

2.12.1. MySQL

MySQL, es un sistema de gestión de base de datos relacional o SGBD. Este gestor de base de datos en multihilo y multiusuario, lo que le permite ser utilizado por varias personas al mismo tiempo, e incluso, realizar varias consultas a la vez, lo que lo hace sumamente versátil (TecInf, 2018).

MySQL nació como una iniciativa de *Software Libre* y aún sigue ofreciéndose como tal, para usuarios particulares. Pero si se desea utilizarlo para promover datos en una empresa, se puede comprar una licencia, como un software propietario, que es autoría de la empresa patrocinante (Actualmente *Oracle Corporation*) (TecInf, 2018).

La mayor parte de su código se encuentra escrito en lenguaje C/C++ y la sintaxis de su uso es bastante simple, lo que permite crear bases de datos simples o complejas con mucha facilidad. Además, es compatible con múltiples plataformas informáticas y ofrece una infinidad de aplicaciones que permiten acceder rápidamente a las sentencias del gestor de base de datos (TecInf, 2018).

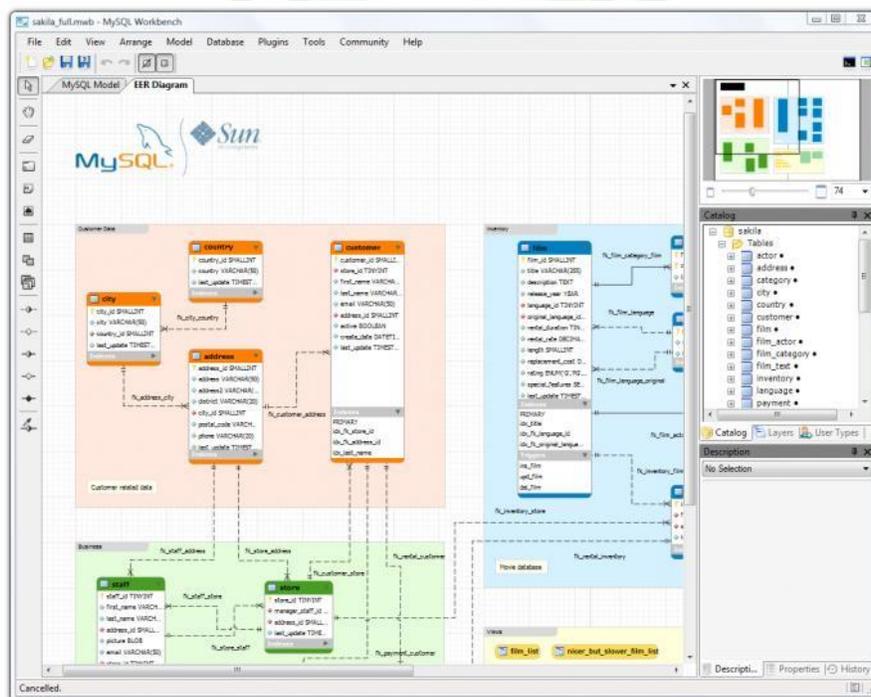


FIGURA 2.14. DESARROLLO DE BASES DE DATOS EN MYSQL.

FUENTE: TecInf, 2018.

2.12.2. PHP 7.0

PHP (Hypertext Preprocessor) es un lenguaje de programación de código abierto, gratuito y multiplataforma creado en 1995 y es usado para generar páginas web dinámicas (dinámica porque el contenido de la página puede cambiar debido a búsquedas, cambios en bases de datos, etc.). Para el caso de una conexión de una aplicación desarrollada en Android existe la *API PHP REST* respaldada con una base de datos *MySQL* el cual es un esquema muy común de una aplicación móvil empresarial. Cuando el escenario requiere que los datos se almacenen de manera centralizada, entonces se debe utilizar esta arquitectura. De lo contrario, la base de datos local en el móvil se puede utilizar para el almacenamiento y recuperación de información. (TecInf, 2018)

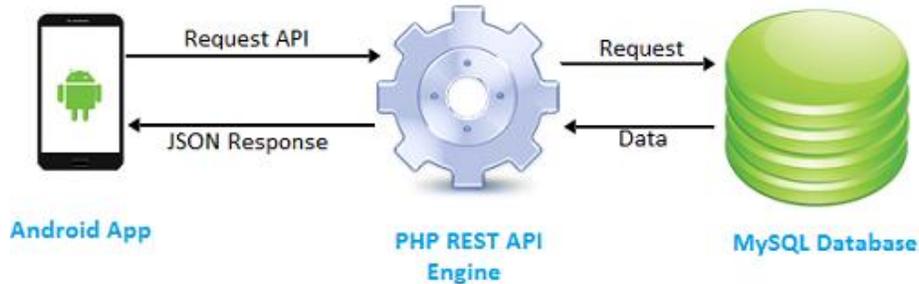


FIGURA 2.15. COMUNICACIÓN CLIENTE-SERVIDOR CON INTÉRPRETE PHP.

FUENTE: TecInf, 2018.

En este ejemplo, se llama a la *API REST de PHP* desde una aplicación de *Android*. En el lado del servidor, el servicio *API* lee datos de la base de datos y envía la respuesta en formato *JSON*. Después de recibir la respuesta, la aplicación de Android muestra la fila de elementos en un *ListView* analizando los datos *JSON*. Una ventaja de *PHP* es que dispone de soporte nativo para la gestión y conexión a bases de datos *SQL* desde su propio código, muy útil para el presente trabajo. (TecInf, 2018).

CAPÍTULO III

INGENIERÍA DEL PROYECTO

En el presente capítulo se realiza un diagnóstico sobre el actual proceso de gestión académica de los postulantes en el Curso Prefacultativo para detectar necesidades y analizar los requerimientos de los usuarios finales, la concepción inicial de la solución, los detalles del diseño propuesto. Luego se darán las pautas para las consideraciones de implementación, todo ello bajo una metodología formal de desarrollo como es RAD y el lenguaje de modelado UML, finalmente se realiza el desarrollo y prueba de la aplicación propuesta.

3.1. Metodología de la Investigación

3.1.1. Tipo de Investigación

El tipo de investigación es aplicada, asimismo se empleó una investigación documental la cual realiza una revisión bibliográfica relacionada con el tema, de esta manera se pretende profundizar de manera conceptual en el desarrollo de la aplicación y su respectiva prueba, además, porque es imprescindible recopilar toda la información necesaria referida a los programas y herramientas utilizadas, lo que permitirá alcanzar de manera eficaz los objetivos planteados.

3.1.2. Población y Muestra

Debido a las características de la presente investigación no se determinó una población y muestra debido a que la aplicación desarrollada en Android será implementada y probada a nivel prototipo por el autor del referido proyecto de grado.

3.1.3. Recolección de Información

La recolección de información se la realizó de fuentes primarias y secundarias, en el primer caso fue de los medios digitales con los que cuenta el Curso Prefacultativo y en el segundo caso de información digital encontrada en la red Internet, libros y demás documentos relacionados con el

tema, para así conocer en forma detallada los requerimientos de los procesos que se consideraron tanto para el desarrollo del aplicativo como para su funcionamiento, obteniendo información adecuada y necesaria para resolver la problemática planteada.

3.1.4. Procesamiento y Análisis de Datos

Lo primero que se realizó al recopilar la información fue analizar los datos obtenidos con relación al problema planteado, donde la misma fue valorada en base a información digital encontrada en la red Internet, libros y demás documentos relacionados con el tema.

3.2. Diagnóstico sobre el Actual Proceso de Gestión Académica

De acuerdo al diagnóstico realizado sobre los medios de comunicación que emplea el curso Prefacultativo se pudo evidenciar que realiza la publicación de sus comunicados de manera física en un panel de información que se encuentra en la puerta de la secretaria del Curso Prefacultativo localizado en predios del edificio ex Príncipe de Paz (en la calle Bueno).



FIGURA 3.1. PÁGINA WEB DE LA UMSA²².

FUENTE: CAPTURA DE PANTALLA, 2021.

Asimismo, realiza la publicación de los comunicados, convocatorias y horarios mediante la página Web de la UMSA²³.

²² <http://prefing.umsa.edu.bo/>

²³ https://www.umsa.bo/72/-/asset_publisher/vMO8v3U0Vo23/content/cpf-fi/20142

A su vez, lo realizan mediante la página Web del Curso Prefacultativo creado para tal efecto, como se muestra a continuación:



FIGURA 3.2. PÁGINA WEB DEL CURSO PREFACULTATIVO²⁴.
FUENTE: CAPTURA DE PANTALLA, 2021.

También, dentro de la página web del Curso Prefacultativo se pudo evidenciar que cuentan con un sistema de información para los alumnos que les provee acceso tanto para inscribirse al Curso Prefacultativo como para averiguar las notas una vez finalizado el mismo.

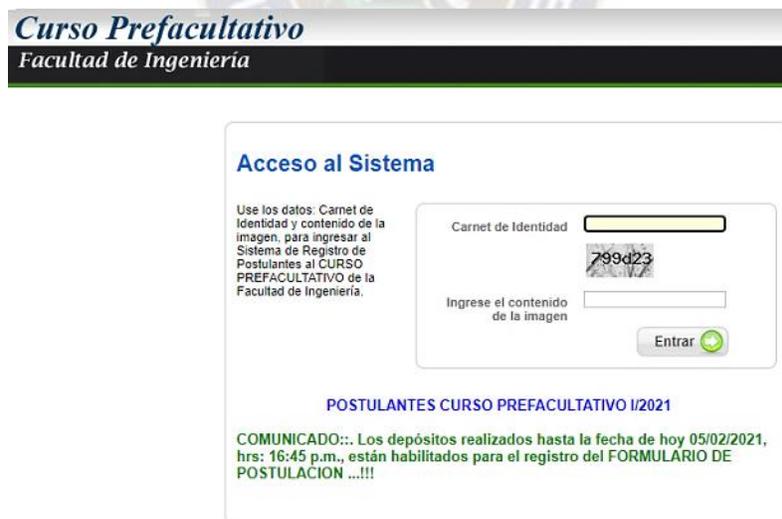


FIGURA 3.3. ACCESO AL SISTEMA PARA INSCRIPCIÓN AL CURSO PREFACULTATIVO²⁵.
FUENTE: CAPTURA DE PANTALLA, 2021.

²⁴ <http://prefing.umsa.edu.bo/>

²⁵ *Ibíd.*

Curso Prefacultativo
Facultad de Ingeniería

Acceso al Sistema

Use su nombre de usuario, clave de su credencial y el contenido de la imagen, para ingresar al Sistema del Curso Prefacultativo.



Usuario

Clave



Ingrese el contenido de la imagen

Entrar 

...: NOTAS CURSO PREFACULTATIVO SEMESTRE I/2021 ...:

FIGURA 3.4. ACCESO AL SISTEMA PARA CONSULTAR NOTAS DEL CURSO PREFACULTATIVO²⁶.
FUENTE: CAPTURA DE PANTALLA, 2021.

Finalmente, el Curso Prefacultativo posee una *Fan Page* en la Red Social Facebook para la publicación de toda la información para los postulantes. De igual forma indican un número de WhatsApp en la modalidad de *Call Center*.



FIGURA 3.5. FAN PAGE EN FACEBOOK DEL CURSO PREFACULTATIVO²⁷.
FUENTE: CAPTURA DE PANTALLA, 2021.

²⁶ Ibid.

²⁷ <https://www.facebook.com/prefing2021/>

3.2.1. Análisis FODA

3.2.1.1. Fortalezas

Las Fortalezas identificadas en el Curso Prefacultativo son:

- Experiencia en la formación de postulantes al Curso Prefacultativo.
- Programa del Curso Prefacultativo en vigencia.
- Contratación de personal administrativo para el apoyo en diferentes áreas.
- Rendimiento óptimo en la cantidad y calidad de postulantes que aprueban de manera satisfactoria el Curso Prefacultativo.
- Calidad académica de los docentes y auxiliares de docencia.
- Prestigio institucional y académico del Curso Prefacultativo.

3.2.1.2. Debilidades

Las Debilidades identificadas en el Curso Prefacultativo son:

- Falta una percepción del ingreso de los nuevos postulantes.
- Trámites burocráticos en la inscripción de nuevos postulantes.
- Demora en los procesos administrativos.
- Carencia en la capacitación y actualización de temas para los docentes.
- Insuficiente capacidad y condiciones de infraestructura.

3.2.1.3. Oportunidades

Las Oportunidades identificadas en el Curso Prefacultativo son:

- Prestigio del Curso Prefacultativo a nivel académico.
- Aumento en la demanda de postulantes.

3.2.1.4. Amenazas

Las Amenazas identificadas en el Curso Prefacultativo son:

- Otras Universidades privadas no ofrecen este Curso Prefacultativo siendo el ingreso de manera directa.
- Competencia de Universidades privadas que ofrecen programas menos exigentes.
- Ineficiente sistema administrativo.

3.3. Datos Generales sobre el Curso Prefacultativo

3.3.1. Antecedentes Históricos

El 14 de octubre de 1929, se creó en la Universidad Mayor de San Andrés la Facultad de Ciencias Físicas y Matemáticas, que comprendía diferentes ramas de la ingeniería. Se tenía como Rector al Dr. Juan Francisco Bedregal y como primer Decano, al ingeniero Vicente Burgaleta. En 1939, con el rector Dr. Héctor Ormachea Zalles, no solo se consolidó la autonomía universitaria y se aprobó un Plan de Reforma de la Universidad, que reconocía que la misma estaba compuesta por tres facultades: Ciencias Biológicas, Ciencias Sociales y Ciencias Exactas, esta última integrada por las Escuelas de Puentes y Calzadas, Escuela de Arquitectura, Escuela de Industrias y Manufacturas, Escuela de Minas y Petróleo y Escuela de Agronomía e Instituto de Ciencias Exactas.

Hasta ese momento no se tenía un lugar fijo para las distintas carreras, funcionaba en hoteles e incluso billares, funcionando el rectorado junto a Ingeniería en predios ubicados entre las calles Ingavi esq. Jenaro Sanjinés. Ya en 1937, se logra el traspaso del terreno que perteneció al Colegio Militar, para poder construir una Ciudad Universitaria que incluía a casi todas las facultades (excepto la de Medicina que se encontraba ya en Miraflores), pero el proyecto fue cambiado para construir un edificio principal de 15 o 20 pisos y 10 en los otros pabellones, por el reducido espacio que se ocupaba, la primera parte de la obra comprendía solamente el pabellón principal y los de Resistencia de Materiales e Ingeniería Sanitaria, que a la postre fueron los únicos que se pudieron edificar.

El cuerpo central tiene carácter monolítico (monoblock), teniendo un frente con figuras pre coloniales, y formas escalonadas, típicas del mundo pre colonial de los Andes, encargado la obra al Arquitecto Emilio Villanueva, que era el decano por ese tiempo, terminando de erigirse para 1948. En 1943 se creó la Escuela de Ingeniería Industrial, que pasó a formar parte de la entonces existente Facultad de Ciencias Exactas, iniciando sus labores efectivas en 1944. En 1953 se da

autonomía a las Escuelas y se eleva el rango de las mismas convirtiéndolas en Facultades de Ingeniería Civil y de Ingeniería Industrial.

En 1955 se crean como parte integrante de la Facultad de Ingeniería Industrial, las Carreras de Petróleos, Mecánica y Química; años más tarde las carreras de Metalurgia y Eléctrica. Al empezar los sesenta (como ya lo habíamos indicado antes), en forma paralela a la UMSA, se crea el "Instituto Tecnológico Boliviano" con las carreras de Ingeniería Petrolera, Geológica y Minas, las que fueron incorporadas en 1964 a la Facultad de Ingeniería. Luego de que el edificio fue tomado y cedido a la Facultad. En 1972 las Facultades de Ingeniería Civil e Ingeniería Industrial se fusionan para convertirse en Facultad de Tecnología, posteriormente pasa a convertirse en Facultad de Ingeniería.

En 1981, los de Geológicas se trasladan al Edificio Hoy. A partir de 1984, ocho carreras constituyen las especialidades principales en la actual Facultad de Ingeniería, que han dado lugar a la oferta de 12 carreras y programas acreditados y reacreditados por CACEI/México y MERCOSUR: Civil, Eléctrica, Electromecánica, Electrónica, Industrial, Industrial Amazónica (en San Buenaventura, Abel Iturralde), Mecánica, Metalúrgica y Materiales, Petrolera, Química, Ambiental y Alimentos. Contando con 11 institutos: Hidrología e Hidráulica, Ensayo de Materiales, Ingeniería Sanitaria, Transportes y Vías de Comunicación, Investigaciones Eléctricas, Electrónica aplicada, Investigaciones Industriales, Mecánica y Electromecánica, Investigaciones metalúrgicas y de materiales, Investigaciones petroleras e Investigaciones y desarrollo de procesos químicos. La facultad cuenta con edificaciones que están ubicadas en la plaza del Obelisco; en los predios de Cota Cota; en el edificio ex Príncipe de Paz (en la calle Bueno) donde funciona el Curso Prefacultativo y en Villa Fátima.



FIGURA 3.6. EDIFICIO DEL CURSO PREFACULTATIVO²⁸.

FUENTE: VYMaps, 2021.

3.3.2. Curso Prefacultativo

La Universidad Mayor de San Andrés de la ciudad de La Paz (UMSA) tiene una oferta de Cursos Prefacultativos, es decir, aquellos que permiten el ingreso a esta casa de estudios a los alumnos desde mediados de sexto de secundaria. El objetivo de ese proceso es favorecer a los bachilleres de La Paz para que no pierdan tiempo y accedan directamente el próximo año a una carrera universitaria, aprobando el curso o la Prueba de Suficiencia Académica (P.S.A.) (anteriormente denominada Evaluación de Dispensación). Las clases tienen distintos horarios y los estudiantes asisten de acuerdo a sus clases en el colegio.

Conforme Resolución N° 184/98 de fecha 09 de septiembre de 1998 del Honorable Consejo Universitario, el ingreso de nuevos estudiantes a la Universidad Mayor de San Andrés (UMSA) procede previa aprobación del Curso Preuniversitario (C.P.F.), asimismo, requiere con carácter indispensable, la aprobación del Curso Prefacultativo; aunque alternativamente contempla, la aprobación de la PSA.

La Prueba de Suficiencia Académica (P.S.A.), corresponde a la evaluación de conocimientos adquiridos en la formación del Ciclo Secundario y que se consideran necesarios para iniciar

²⁸ <https://vymaps.com/BO/Universidad-Mayor-De-San-Andres-curso-Prefacultativo-De-Ingenieria-107555/>

estudios superiores en Ingeniería. Consiste en pruebas escritas expresamente preparadas para tal objetivo, haciendo énfasis en aspectos conceptuales y de dominio práctico en las asignaturas de:

- Matemáticas.
- Física.
- Química.

Cada una de las tres pruebas se evalúa sobre 100 puntos y la nota mínima de aprobación es de cincuenta y uno por ciento (51%). La aprobación de la Prueba de Suficiencia Académica se determina cuando el postulante aprueba las tres asignaturas.

La asistencia a la Prueba de Suficiencia Académica tiene carácter optativo y el postulante que apruebe esta prueba queda dispensado de realizar el Curso Prefacultativo de cinco meses, habilitándose de manera inmediata para su inscripción en el primer semestre de la carrera elegida. Caso contrario, puede incorporarse al Curso Prefacultativo inmediatamente en el horario elegido en oportunidad de su inscripción.

3.3.2.1. Objetivos del Curso Prefacultativo

- Ambientar al postulante a las exigencias de las actividades académicas del nivel universitario.
- Promover la nivelación de los conocimientos básicos adquiridos en Secundaria de las asignaturas de Ciencias Exactas (Matemáticas, Física y Química) necesarias para el desarrollo formativo en el área de las ingenierías.
- Fortalecer los conocimientos, aptitudes y habilidades necesarias para el desarrollo formativo en la carrera profesional a seguir en la Facultad de Ingeniería de la UMSA.
- Introducir al postulante en el conocimiento de los aspectos relacionados con los principios, estructura y funcionamiento de la Universidad Mayor de San Andrés y sus Facultades.
- Contribuir a superar el criterio individualista y memorístico del modelo tradicional de aprendizaje, incentivando la utilización del método científico mediante un proceso de enseñanza – aprendizaje basado en clases magistrales, proyectos de aula e investigación bibliográfica.
- Proporcionar orientación vocacional, a través de cada una de las carreras que actualmente ofrece la Facultad de Ingeniería.

3.3.2.2. Misión

El Curso Prefacultativo de la Facultad de Ingeniería tiene como misión de nivelar, actualizar y orientar a los bachilleres postulantes a alguna de las carreras de la Facultad de Ingeniería. De tal manera de prepararlos académicamente para comenzar sus estudios universitarios. En ese entendido también es un curso de adecuación al sistema universitario.

3.3.2.3. Visión

El Curso Prefacultativo de la Facultad de Ingeniería pretende ser el referente a nivel departamental y nacional sobre la orientación, nivelación, actualización en las asignaturas de Matemáticas, Física y Química para los postulantes a alguna de las carreras de la Facultad de Ingeniería y para todas aquellas personas que estén interesadas en nivelarse en las mencionadas asignaturas.

3.3.2.4. Organigrama del Curso Prefacultativo

El organigrama del Curso Prefacultativo se estructura de la siguiente manera:

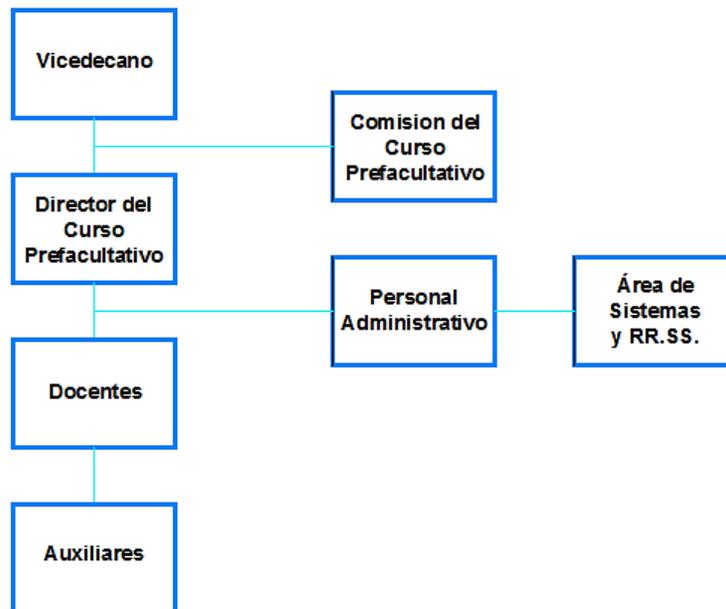


FIGURA 3.7. ORGANIGRAMA DEL CURSO PREFACULTATIVO.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.4. Análisis de la Situación Actual

Como parte del análisis de la situación actual se pretende comprender el entorno e interacciones que se lleva a cabo, para que de esta manera sea posible identificar los procesos y actores que se convertirán en base fundamental para la propuesta del presente proyecto de grado.

3.4.1. Análisis de Procesos Actuales

Las fases que conlleva desde la verificación de inscripción de un nuevo postulante hasta la publicación de comunicados, convocatorias, horarios y cronogramas, tienen como objetivo fundamental brindar interacción e informar sobre todas las actividades que se llevan a cabo en el Curso Prefacultativo.

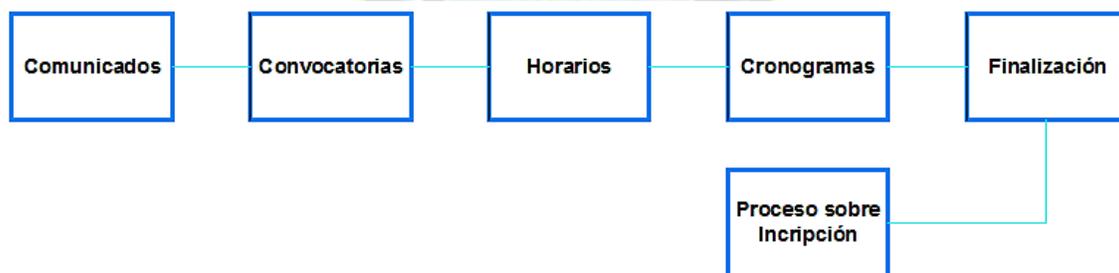


FIGURA 3.8. PROCESO ACTUAL SOBRE LAS FASES DE DIFUSIÓN DE INFORMACIÓN.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.4.1.1. Proceso de Publicación en Página Web de la UMSA

Durante el proceso de publicación de información en la página Web de la UMSA, el Director del Curso Prefacultativo establece un requerimiento de información para docentes y auxiliares para la publicación posterior de comunicados, convocatorias, horarios y cronogramas. Los mismos reciben el requerimiento para su análisis, una vez dado el visto bueno se procede a programar y coordinar la información con el personal administrativo, mismos que proceden a preparar y redactar la información. Finalmente, el personal administrativo pasa esta información digitalizada al área de sistemas para que lo publiquen en la página Web de la UMSA. Este proceso se muestra a continuación:

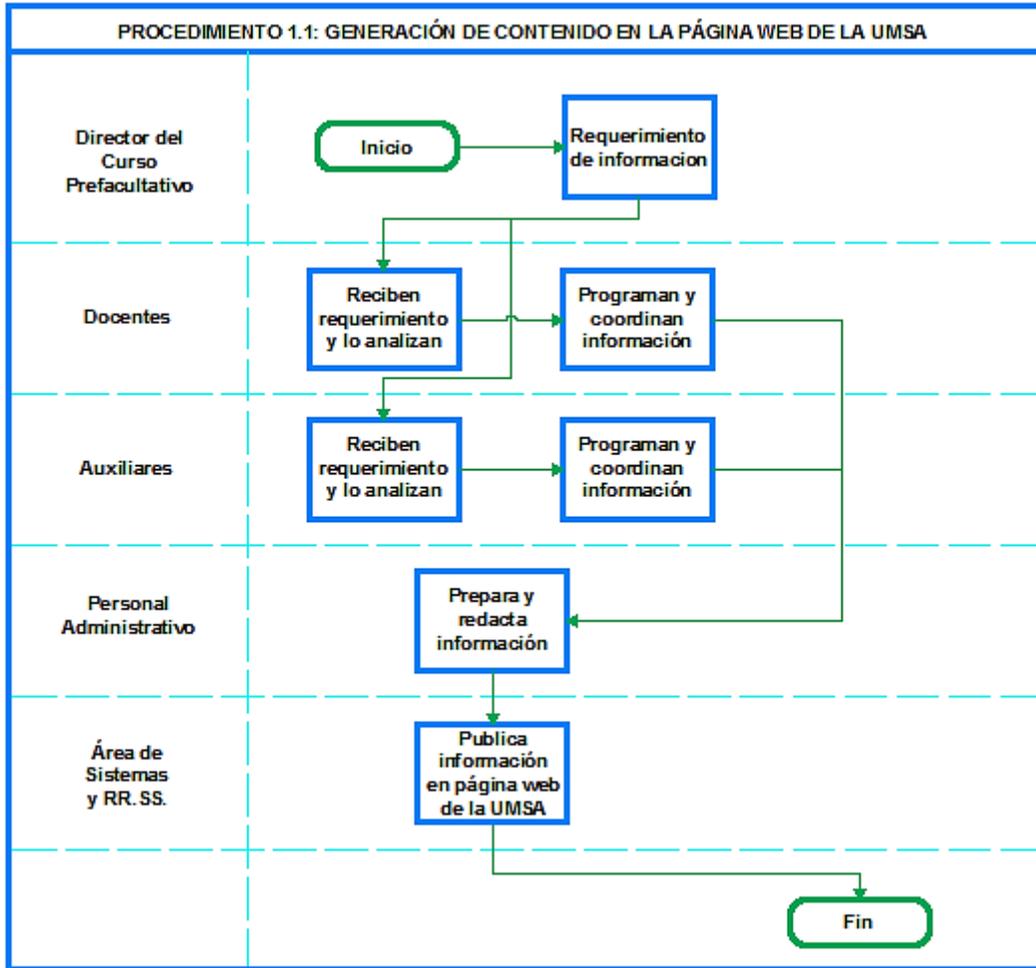


FIGURA 3.9. PROCESO DE PUBLICACIÓN DE INFORMACIÓN EN PÁGINA WEB DE LA UMSA.
FUENTE: ELABORACIÓN PROPIA, 2021.

3.4.1.2. Proceso de Publicación en Página Web del Curso Prefacultativo

Durante el proceso de publicación de información en la página Web del Curso Prefacultativo, el Director del mismo establece un requerimiento de información para docentes y auxiliares para la publicación posterior de comunicados, convocatorias, horarios y cronogramas. Los mismos receptionan el requerimiento para su análisis, una vez dado el visto bueno se procede a programar y coordinar la información con el personal administrativo, mismos que proceden a preparar y redactar la información. Finalmente, el personal administrativo pasa esta información digitalizada al área de sistemas para que lo publiquen en la página web del Curso Prefacultativo. Este proceso se muestra a continuación:

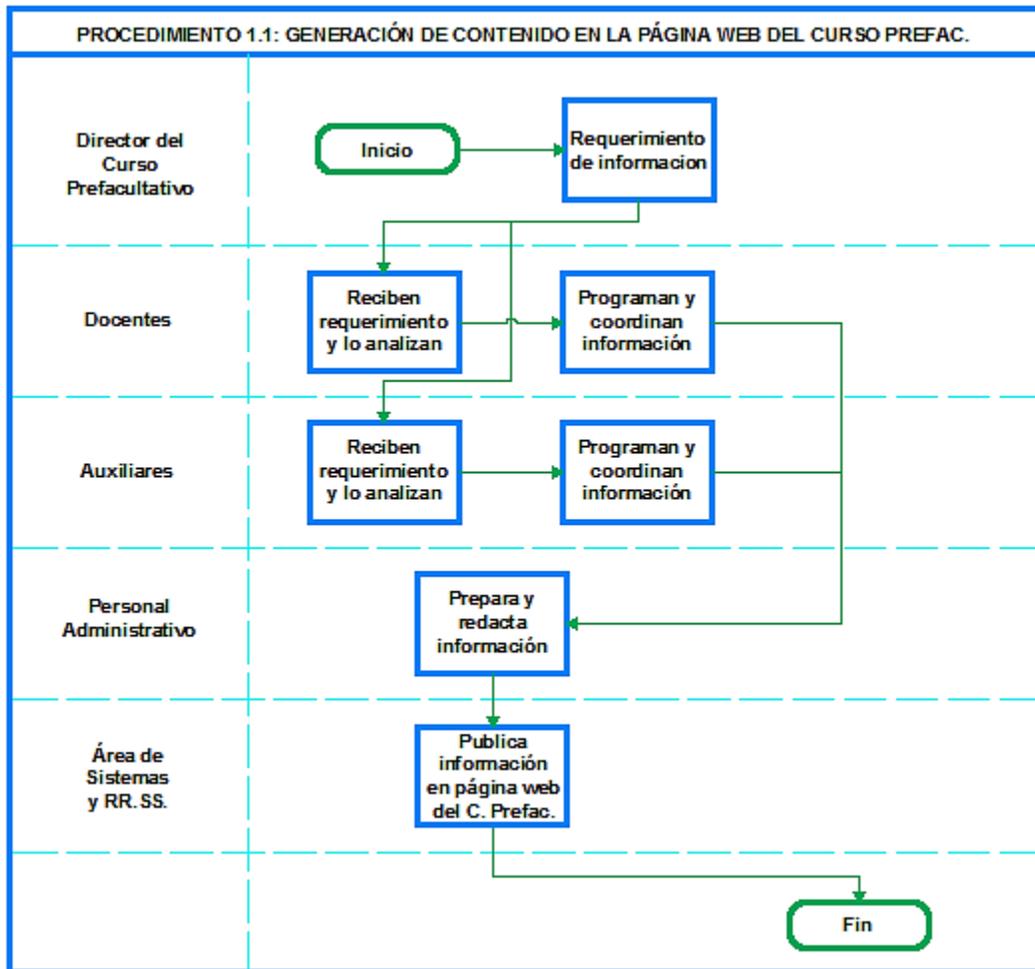


FIGURA 3.10. PROCESO DE PUBLICACIÓN DE INFORMACIÓN EN PÁGINA WEB DEL CURSO PREFACULTATIVO.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.4.1.3. Proceso de Publicación en Facebook

Durante el proceso de publicación de información en Facebook, el Director del Curso Prefacultativo establece un requerimiento de información para docentes y auxiliares para la publicación posterior de comunicados, convocatorias, horarios y cronogramas. Los mismos receptionan el requerimiento para su análisis, una vez dado el visto bueno se procede a programar y coordinar la información con el personal administrativo, mismos que proceden a preparar y redactar la información. Finalmente, el personal administrativo pasa esta información digitalizada al área de sistemas para que lo publiquen en la *Fan Page* de Facebook del Curso Prefacultativo.

El proceso se muestra a continuación:

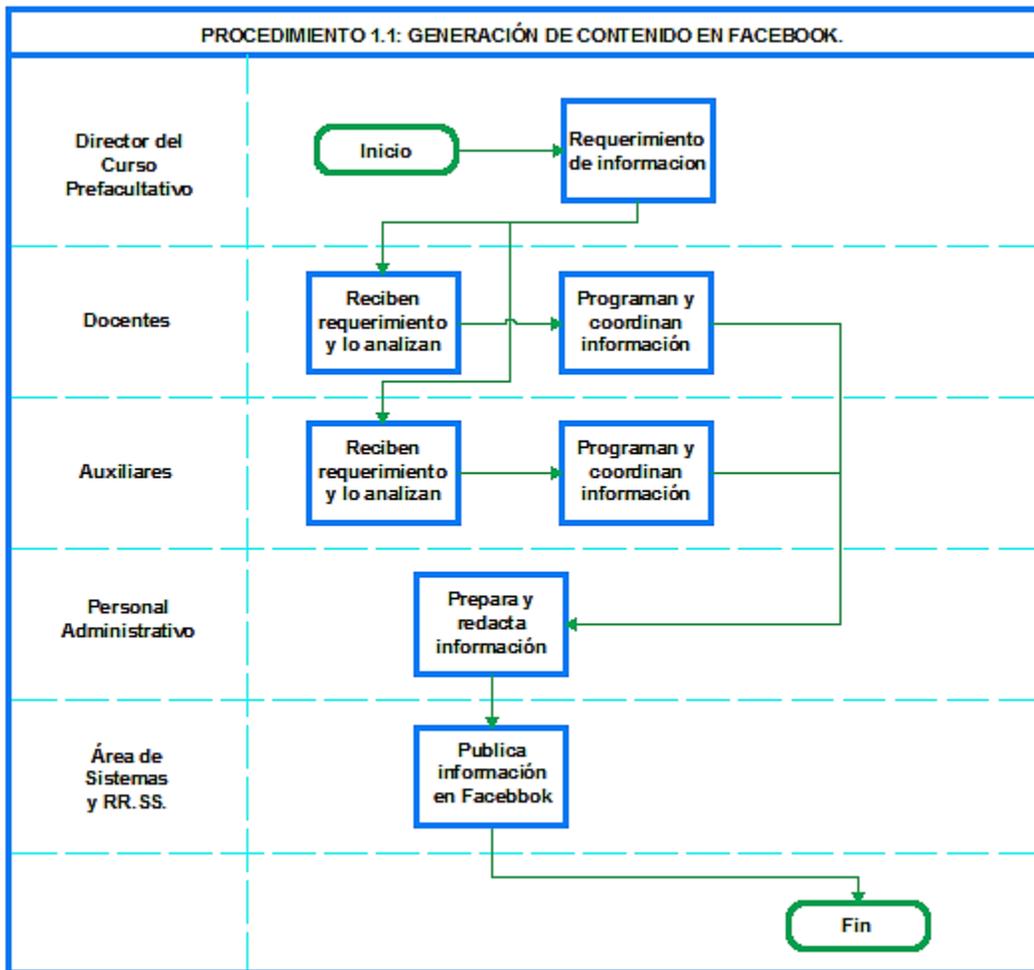


FIGURA 3.11. PROCESO DE PUBLICACIÓN DE INFORMACIÓN EN FACEBOOK.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.4.1.4. Proceso de Inscripción de Postulantes

Durante el proceso de inscripción los postulantes realizan un depósito en una entidad bancaria habilitada para tal propósito. Después, ingresa a la página web del Curso Prefacultativo prefing.umsa.edu.bo y se registra; realiza un click en el formulario de postulación, a continuación, ingresa al sistema, se registra en un formulario y elige su horario, asimismo, registra el recibo de la empresa Síntesis para después subir su fotografía y reimprimir el formulario llenado. Finalmente, entrega sus documentos de manera personal de acuerdo a cronograma establecido por el Curso Prefacultativo.

El proceso se muestra a continuación:

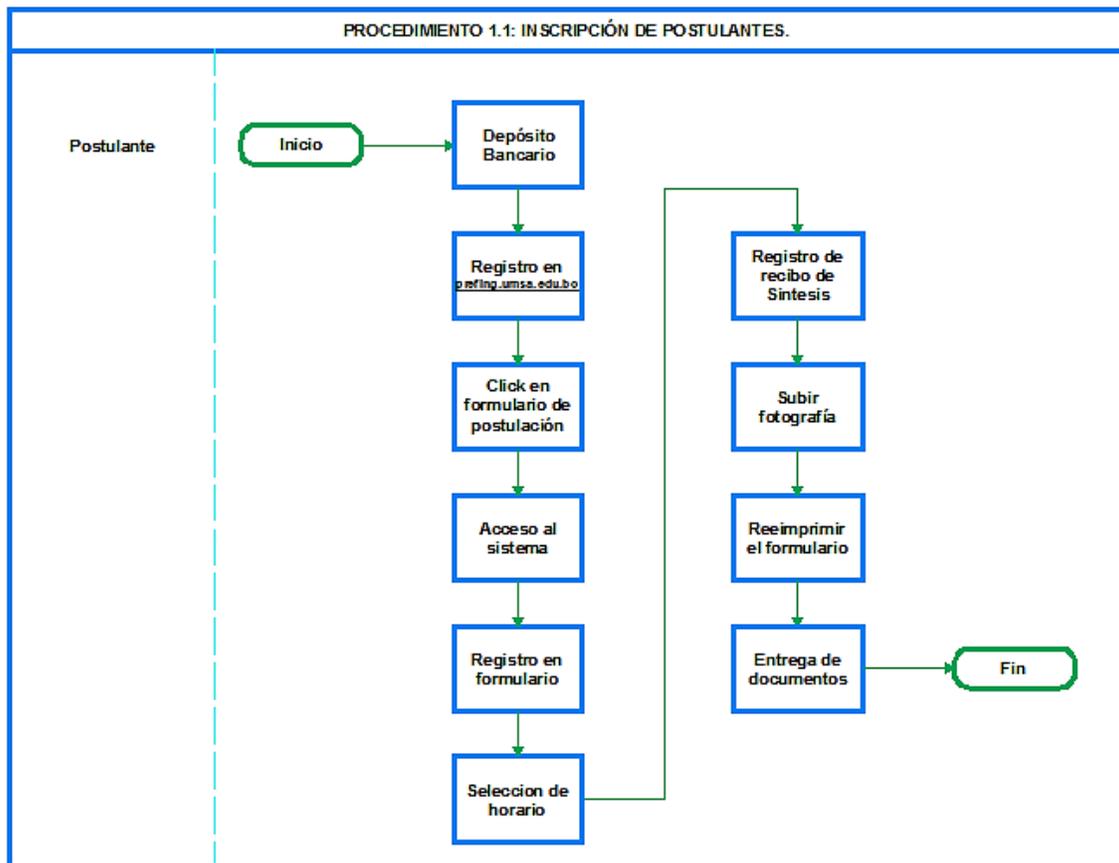


FIGURA 3.7. PROCESO DE INSCRIPCIÓN DE POSTULANTES.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.4.2. Identificación de Actores

A continuación, se procede a determinar los actores que intervienen en el desarrollo de la aplicación para teléfonos móviles inteligentes desarrollada en Android:

Director del Curso Prefacultativo	Elabora y presenta a la COMISIÓN un Plan Global Académico y Administrativo, para el desarrollo de las 2 modalidades de admisión de postulantes a la Universidad. (Prueba de Suficiencia Académica y Curso Prefacultativo).
Docentes	Son designados en cada Facultad, dependen directamente del DIRECTOR y son responsables de regentar su respectiva asignatura y participar activamente en el proceso enseñanza – aprendizaje del mismo.
Auxiliares de Docencia	Son estudiantes universitarios regulares y dependen académicamente del Docente y administrativamente del Director. Tienen la responsabilidad de cooperar, prestar apoyo y asistencia académica al docente en el proceso de enseñanza – aprendizaje; pero, sin perjudicar en la asignación de notas en el proceso de evaluación de estudiantes.
Personal Administrativo	Depende directamente del DIRECTOR y se constituyen en funcionarios que tienen la responsabilidad de cooperar y prestar apoyo en el proceso de gestión administrativa.
Área de Sistemas y RR.SS.	Forma parte del personal administrativo y está encargado de la gestión de la información que se publica en medios digitales como la página web institucional y Facebook del Curso Prefacultativo. Administran también el Sistema de Información de Estudiantes.
Postulantes	Bachilleres que se inscriben al Curso Prefacultativo de la Facultad de Ingeniería de la UMSA.

TABLA 3.1. ACTORES INVOLUCRADOS²⁹.

FUENTE: ELABORACIÓN PROPIA, 2021.

²⁹ <https://turismoumsa.files.wordpress.com/2012/04/4-4-reglamento-admision-facultativa.pdf>

3.5. Requerimientos

El proceso de análisis de requerimientos proporcionará una perspectiva detallada con relación a las necesidades existentes para que de esta manera se pueda ofrecer una solución integral tanto en el proceso de diseño como del desarrollo de la aplicación.

3.5.1. Determinación de los Roles del Equipo

De acuerdo a la metodología ágil de desarrollo RAD adaptada para el presente trabajo, como uno de los primeros elementos de la planificación se considera a la determinación de roles y funciones que cumplirán los integrantes del desarrollo del mismo. Debido a que el presente proyecto solo posee un integrante, asumirá más de un rol al mismo tiempo. En la siguiente tabla se representa dicha asignación de roles:

INTEGRANTE	FUNCIÓN / ROL
Área de Sistemas y RR.SS.	Dueño del producto.
Diego Coaquira Illanes	Líder para el desarrollo de la aplicación.

TABLA 3.2. DETERMINACIÓN DE ROLES.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.5.2. Especificación de Usuarios

De acuerdo a la información recopilada, se procede a determinar: los tipos de usuarios, accesos, privilegios, y las diferentes interacciones que se debe llevar a cabo con la aplicación desarrollada en Android.

TIPO DE USUARIO	FUNCIÓN
USUARIO	APLICACIÓN
Usuario Administrador	<ul style="list-style-type: none"> • Encargado de la administración de las funciones generales del sistema. • Administración de cuentas de usuarios. • Generación de reportes generales. • Publicación de información en todas las plataformas digitales.
Usuario de la Aplicación	<ul style="list-style-type: none"> • Consulta de información. • Inscripción al Curso Prefacultativo. • Acceso a Sistema de Información. • Consulta de Notas.

TABLA 3.3. ESPECIFICACIÓN DE USUARIOS.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.5.3. Requerimientos Funcionales de la Aplicación

Los requerimientos funcionales de los estudiantes y administrador son los siguientes:

1. Requerimientos del estudiante:

Encargado: Diego Coaquira Illanes.	
ID: reg_user	Prioridad: 1
Descripción: Este requerimiento se encarga de todo lo relacionado con el proceso de registro de datos del usuario del curso Prefacultativo en la aplicación móvil.	

TABLA 3.4. REQUERIMIENTO FUNCIONAL N° 1.

FUENTE: ELABORACIÓN PROPIA, 2021.

Encargado: Diego Coaquira Illanes.	
ID: log_user	Prioridad: 2
Descripción: Este requerimiento se encarga de todo lo relacionado con el proceso de inicio de sesión del usuario en la aplicación móvil del curso Prefacultativo.	

TABLA 3.5. REQUERIMIENTO FUNCIONAL N° 2.

FUENTE: ELABORACIÓN PROPIA, 2021.

Encargado: Diego Coaquira Illanes.	
ID: visual_info	Prioridad: 3
Descripción: Este requerimiento se encarga de todo lo relacionado con el proceso de visualización de la información del curso Prefacultativo en la aplicación móvil.	

TABLA 3.6. REQUERIMIENTO FUNCIONAL N° 3.

FUENTE: ELABORACIÓN PROPIA, 2021.

2. Requerimientos del administrador de la base de datos:

Encargado: Diego Coaquira Illanes.	
ID: user_admin	Prioridad: 1
Descripción: Este requerimiento dará acceso directo al administrador para trabajar con aplicación móvil elaborada, desde crear su propio usuario y contraseña para acceder a la base de datos de la aplicación, entre otras opciones.	

TABLA 3.7. REQUERIMIENTO FUNCIONAL N° 1.

FUENTE: ELABORACIÓN PROPIA, 2021.

Encargado: Diego Coaquira Illanes.	
ID: reg_admin	Prioridad: 2
<p>Descripción: Este requerimiento se encarga del proceso de registrar a cada estudiante para que acceda a la información publicada del curso Prefacultativo mediante la aplicación móvil, aparte integra opciones para administrar a los estudiantes inscritos o dados de baja.</p>	

TABLA 3.8. REQUERIMIENTO FUNCIONAL N° 2.

FUENTE: ELABORACIÓN PROPIA, 2021.

Encargado: Diego Coaquira Illanes.	
ID: public_admin	Prioridad: 3
<p>Descripción: Este requerimiento se encarga de todo lo relacionado con el proceso de publicación de la información del curso Prefacultativo en la aplicación móvil por parte del administrador.</p>	

TABLA 3.9. REQUERIMIENTO FUNCIONAL N° 3.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.6. Requerimientos no Funcionales

Los requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste, tales como:

Rendimiento de la Aplicación:

- La aplicación ofrecerá respuesta (datos) al usuario en tiempo real.
- El tiempo de respuesta promedio de la aplicación no debe superar los 10 segundos.

Seguridad:

- Requisito de conexión (Debe verificar que sea un usuario creado para la aplicación por lo que consulta directamente en la base de datos para acceder a la aplicación).

Disponibilidad

- La aplicación estar disponible el 100% del tiempo, ya que se trata de una aplicación nativa que se instalará en el dispositivo móvil.
- La aplicación dependerá de una conexión a internet o plan de datos solo para el envío de información. En caso de no tener conexión a internet la misma se guardará en la aplicación (localmente) y se reenviará la petición una vez que tenga conexión a Internet.

Mantenimiento:

- El sistema estará en constante mantenimiento ya que tiene la apertura para agregar nuevas funcionalidades o realizar modificaciones o correcciones.

Portabilidad:

- Compatibilidad con plataformas. El sistema desarrollado ofrece compatibilidad con otras plataformas Android desde la versión 4.0 ya sea en una Tablet o Smartphone.
- Es un sistema portable ya que puede acceder desde cualquier dispositivo móvil.

Operabilidad:

- La aplicación móvil podrá ser operada por cualquier usuario que la descargue y tenga su debido usuario y contraseña, de igual forma será para el administrador de la base de datos.

3.7. Arquitectura de la Solución

3.7.1. Arquitectura del Sistema

La arquitectura a utilizar será cliente-servidor donde el cliente realizará la petición al servidor y éste a su vez le dará la respuesta. En este caso el sistema cuenta con un servidor de base de datos

donde se almacenarán los registros, pedidos del cliente y a su vez recibir información actualizada, para esto el cliente debe contar con acceso a Internet.

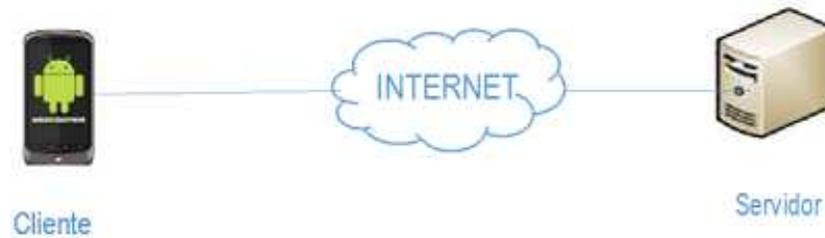


FIGURA 3.13. ARQUITECTURA DEL SISTEMA.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.7.2. Arquitectura de la Aplicación

El proyecto en su totalidad se desarrolló en dos partes. La primera parte es la aplicación móvil para Android y para su funcionamiento se empleará un Smartphone con este sistema operativo, con la aplicación instalada y con acceso a Internet ya que se debe consultar desde cualquier parte del mundo, esta se conecta mediante la aplicación que será descargada de *Google Play Store*³⁰, y la segunda parte es que la comunicación con la base de datos del curso Prefacultativo de la Facultad de Ingeniería de la UMSA la cual almacena la información de los requerimientos del sistema y la publicación de la información, como se muestra a continuación:

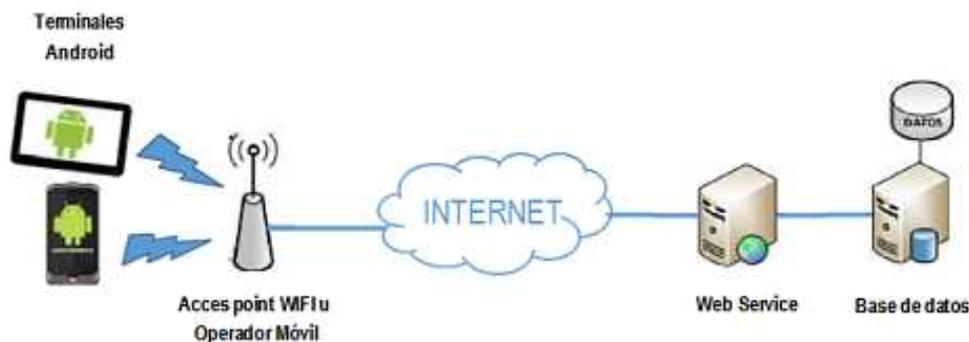


FIGURA 3.14. ARQUITECTURA DE LA APLICACIÓN.

FUENTE: ELABORACIÓN PROPIA, 2021.

³⁰ Play.google.com

3.8. Diagramas de Casos de Uso

3.8.1. Caso de Uso General del Usuario

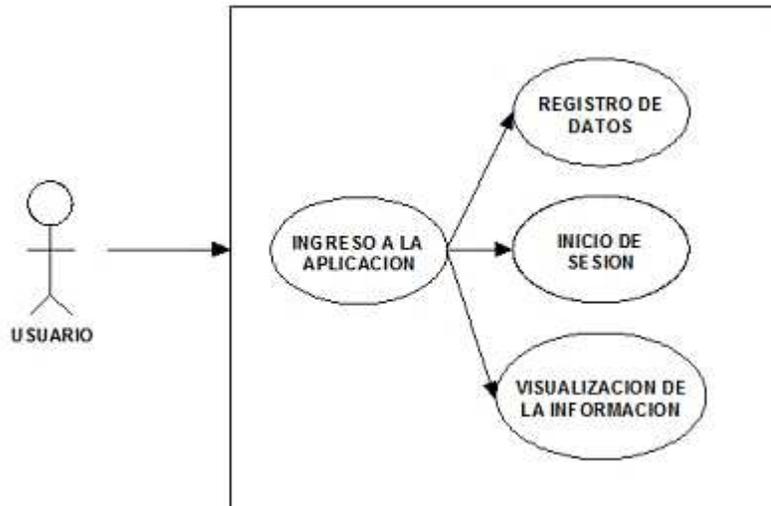


FIGURA 3.15. DIAGRAMA DE CASO DE USO DE USUARIO.

FUENTE: ELABORACIÓN PROPIA, 2021.

Caso de uso: Aplicación móvil para consulta de información.	
Actor:	Usuario.
Descripción: El usuario tendrá que registrar sus datos en la aplicación, para luego hacer un inicio de sesión con un nombre de usuario y contraseña e ingresar a realizar la petición de consulta de información.	
Actividades:	<ul style="list-style-type: none">• Consulta de información.• Inscripción al Curso Prefacultativo.• Acceso a Sistema de Información.• Consulta de Notas.

TABLA 3.10. REQUERIMIENTO FUNCIONAL N° 3.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.8.2. Caso de Uso del Administrador de la Base de Datos

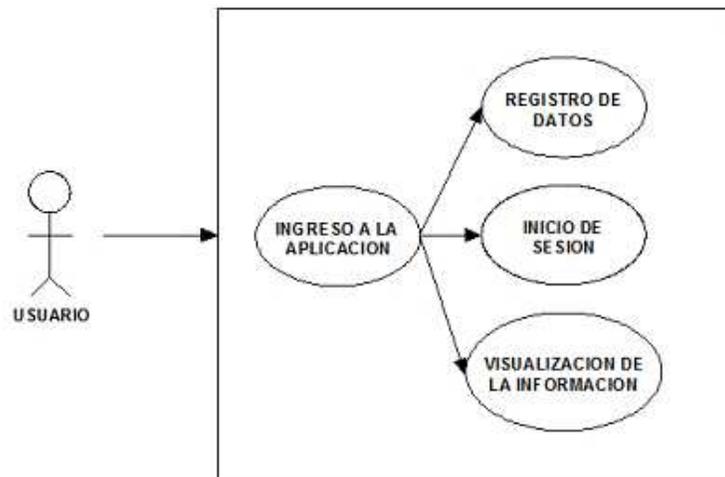


FIGURA 3.16. DIAGRAMA DE CASO DE USO DEL ADMINISTRADOR DE LA BASE DE DATOS.

FUENTE: ELABORACIÓN PROPIA, 2021.

Caso de uso: Base de datos para almacenamiento de información.	
Actor:	Administrador de la base de datos.
Descripción: El usuario tendrá que registrar sus datos en la aplicación, para luego hacer un inicio de sesión con un nombre de usuario y contraseña e ingresar a realizar la petición de consulta de información.	
Actividades:	<ul style="list-style-type: none"> • Encargado de la administración de las funciones generales del sistema. • Administración de cuentas de usuarios. • Generación de reportes generales. • Publicación de información en todas las plataformas digitales.

TABLA 3.11. REQUERIMIENTO FUNCIONAL N° 3.

FUENTE: ELABORACIÓN PROPIA, 2021.

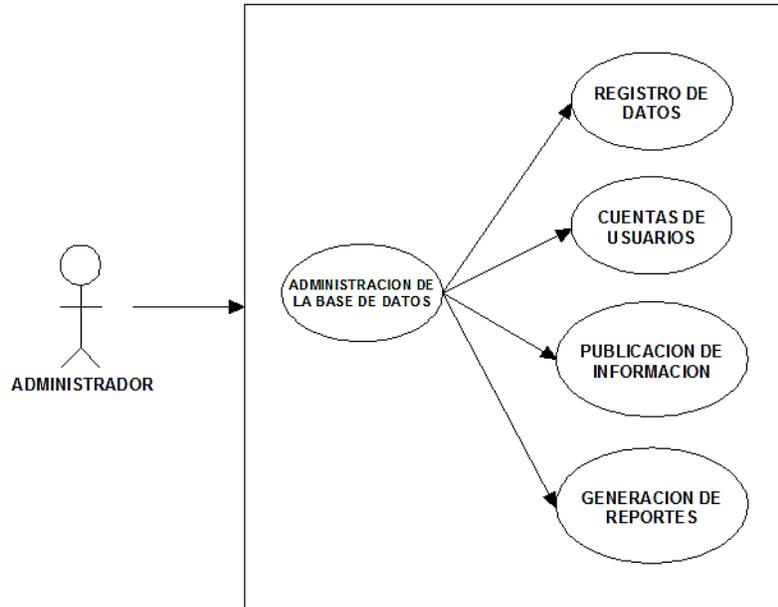


FIGURA 3.17. DIAGRAMA DE CASO DE USO DEL ADMINISTRADOR DE LA BASE DE DATOS.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.9. Diagrama de Actividades

Durante el proceso de publicación de información en la aplicación del Curso Prefacultativo, el Director del mismo establece un requerimiento de información para docentes y auxiliares para la publicación posterior de comunicados, convocatorias, horarios y cronogramas. Los mismos reciben el requerimiento para su análisis, una vez dado el visto bueno se procede a programar y coordinar la información con el personal administrativo, mismos que proceden a preparar y redactar la información. Finalmente, el personal administrativo pasa esta información digitalizada al área de sistemas para que lo publiquen en la aplicación del Curso Prefacultativo. Este proceso se muestra a continuación:

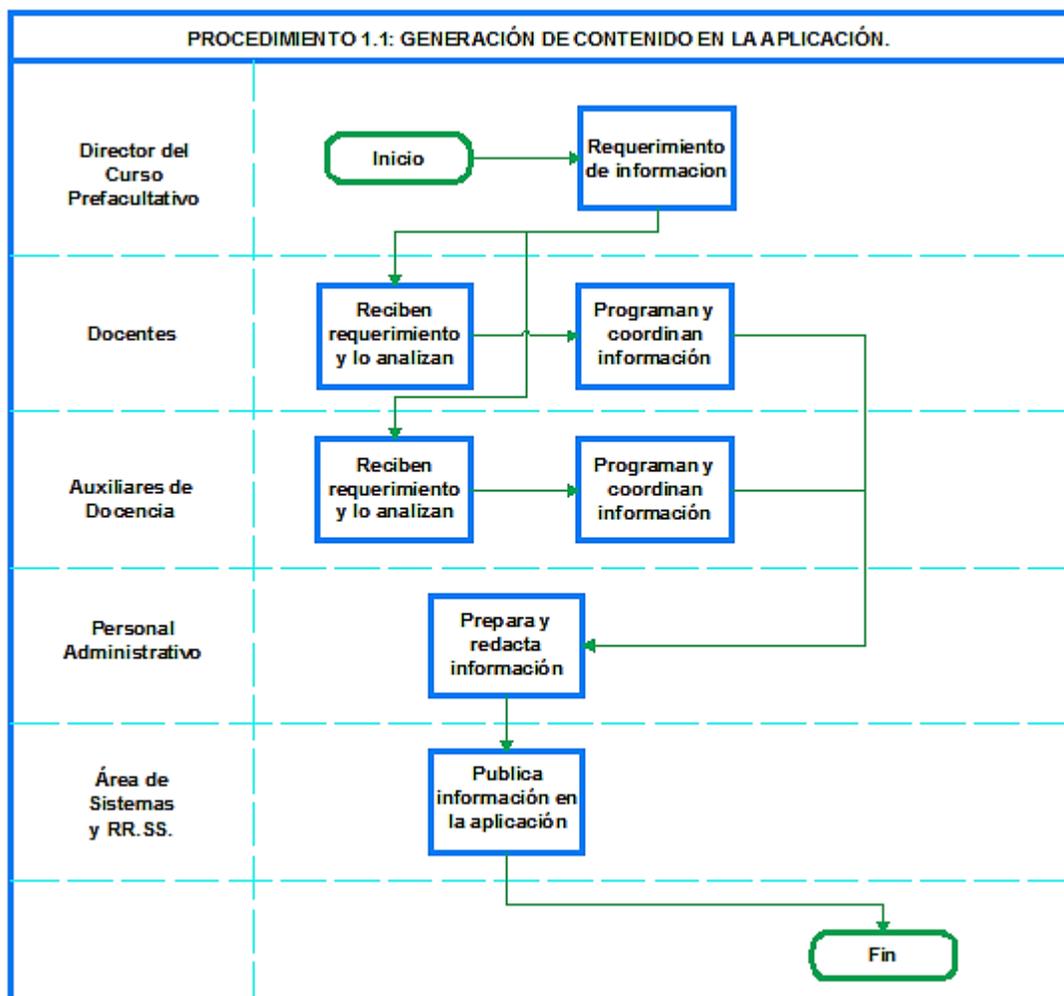


FIGURA 3.18. GENERACIÓN DE CONTENIDO EN LA APLICACIÓN.

FUENTE: ELABORACIÓN PROPIA, 2021.

3.10. Encriptación de *Password* de Acceso con MD5 en PHP

A diferencia de otros lenguajes de programación *PHP* permite de forma nativa (sin el uso librerías externas) encriptar en *MD5*. El objetivo de la encriptación de datos es volver ilegible información considera importante o sensible en un sistema informático. Para cumplir con este objetivo en *PHP* vamos a utilizar la función *MD5()*. En el siguiente ejemplo se procede a encriptar una posible contraseña ingresada por un usuario:

```

1 <?php
2 // Password que puede ingresar un usuario
3 $password = "password1234";
4
5 // Encriptando el password para guardarlo en la base de datos
6 $hash = md5($password);
7
8 echo $hash;
9 ?>

```

Ahora para saber si la contraseña ingresada por un usuario coincide con la registrada en la base de datos se debe hacer lo siguiente:

```

1 <?php
2 // Password que puede ingresar un usuario para iniciar sesión
3 $passwordIngresado = "password1234";
4
5 // Password encriptado guardado en la base de datos
6 $passwordEnBaseDeDatos = "bdc87b9c894da5168059e00ebffb9077";
7
8 // Encriptando el password ingresado
9 $hash = md5($passwordIngresado);
10
11 // Verificando si coinciden los password
12 if($hash == $passwordEnBaseDeDatos){
13     echo "El password coincide";
14 }else{
15     echo "El password no coincide";
16 }
17 ?>

```

3.11. Diseño de Interfaces

3.11.1. Interfaces de la Aplicación Móvil para el USUARIO

Es necesario realizar el diseño de la interfaz del usuario del sistema en las primeras fases, ya que hacerlo una vez desarrollado el sistema, representa un error. Lo que el usuario requiere es una aplicación transparente y amigable que le permita navegar por las distintas opciones. El primer contacto con el usuario debe ser satisfactorio, procurando que el mismo se encuentre cómodo desde el inicio, sin problemas técnicos en lo posible.

En la siguiente figura se muestra la interface principal de la aplicación, cualquier usuario puede consultar la información de su interés puesto, no así la opción “Contáctanos” que esta reservado para estudiantes ya inscritos y registrados en la base de datos del Curso Prefacultativo.



FIGURA 3.19. INTERFACE PRINCIPAL DE LA APLICACIÓN MÓVIL.

FUENTE: ELABORACIÓN PROPIA, 2021.

Al elegir la opción “Convocatorias” nos direcciona a otra ventana en la que se muestra todas las convocatorias actuales en curso con su respectiva bitácora, el usuario elige la convocatoria de su interés y esta se despliega de tamaño completo para recabar la información pertinente.

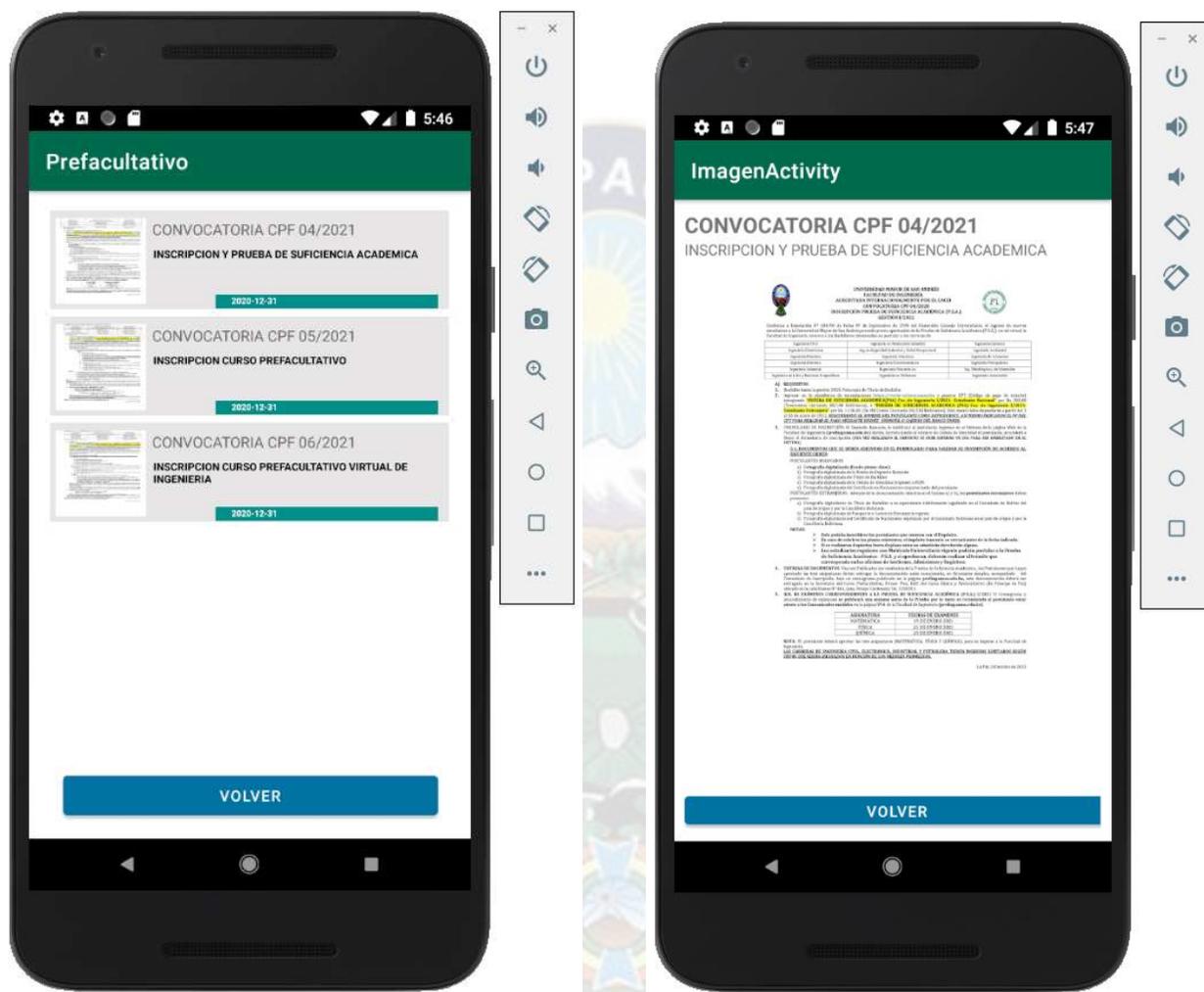


FIGURA 3.20. PANTALLA DE LA OPCIÓN CONVOCATORIAS.
FUENTE: CAPTURA DE PANTALLA, 2021.

Al elegir la opción “Comunicados” nos direcciona a otra ventana en la que se muestra todos los comunicados actuales en curso con su respectiva bitácora, el usuario elige el comunicado de su interés y esta se despliega de tamaño completo para recabar la información pertinente.

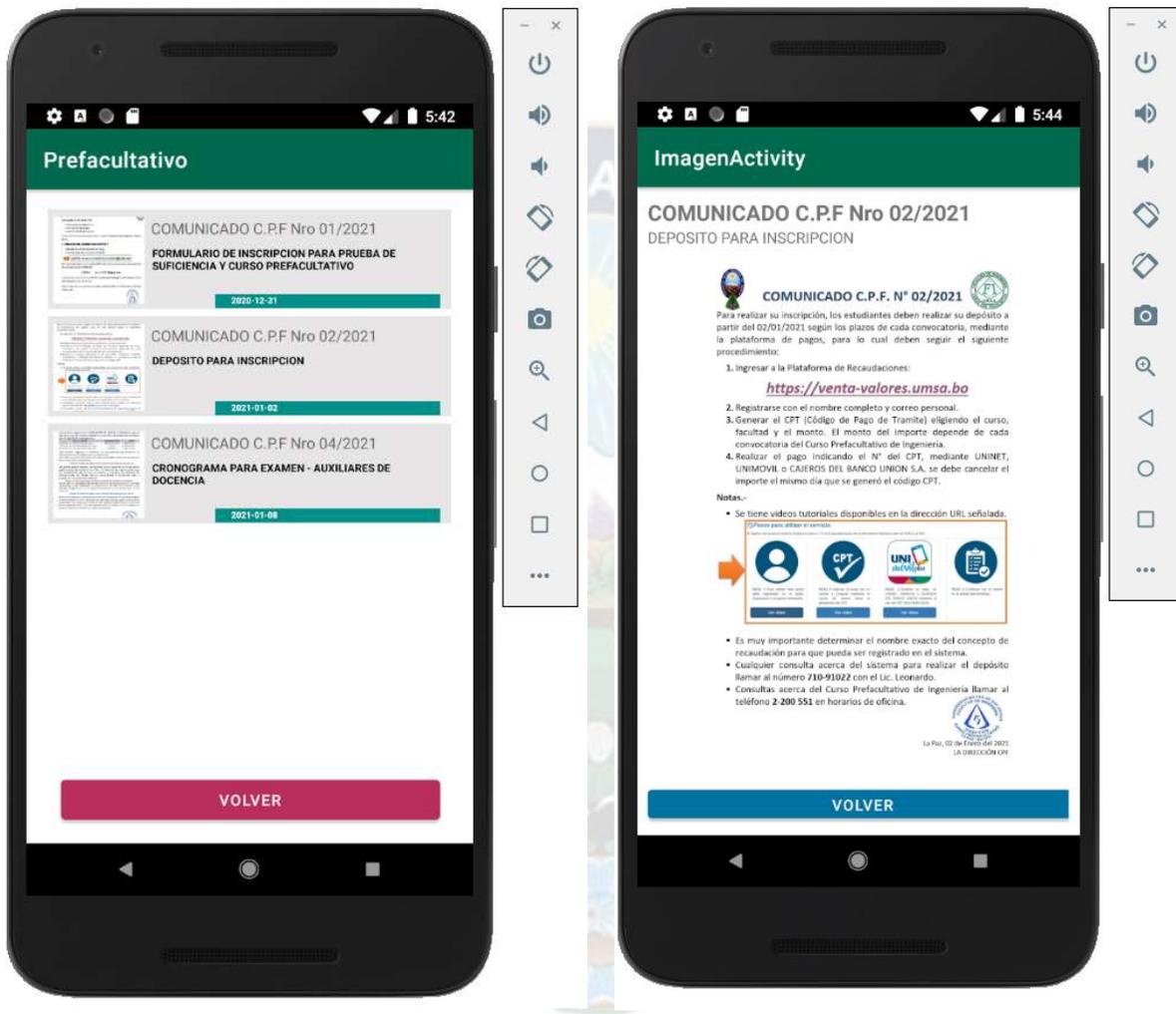


FIGURA 3.21. PANTALLA DE LA OPCIÓN COMUNICADOS.

FUENTE: CAPTURA DE PANTALLA, 2021.

Al elegir la opción “Horarios” nos direcciona a otra ventana en la que se muestra todos los horarios actuales en curso con su respectiva bitácora, el usuario elige el horario de su interés y esta se despliega de tamaño completo para recabar la información pertinente.

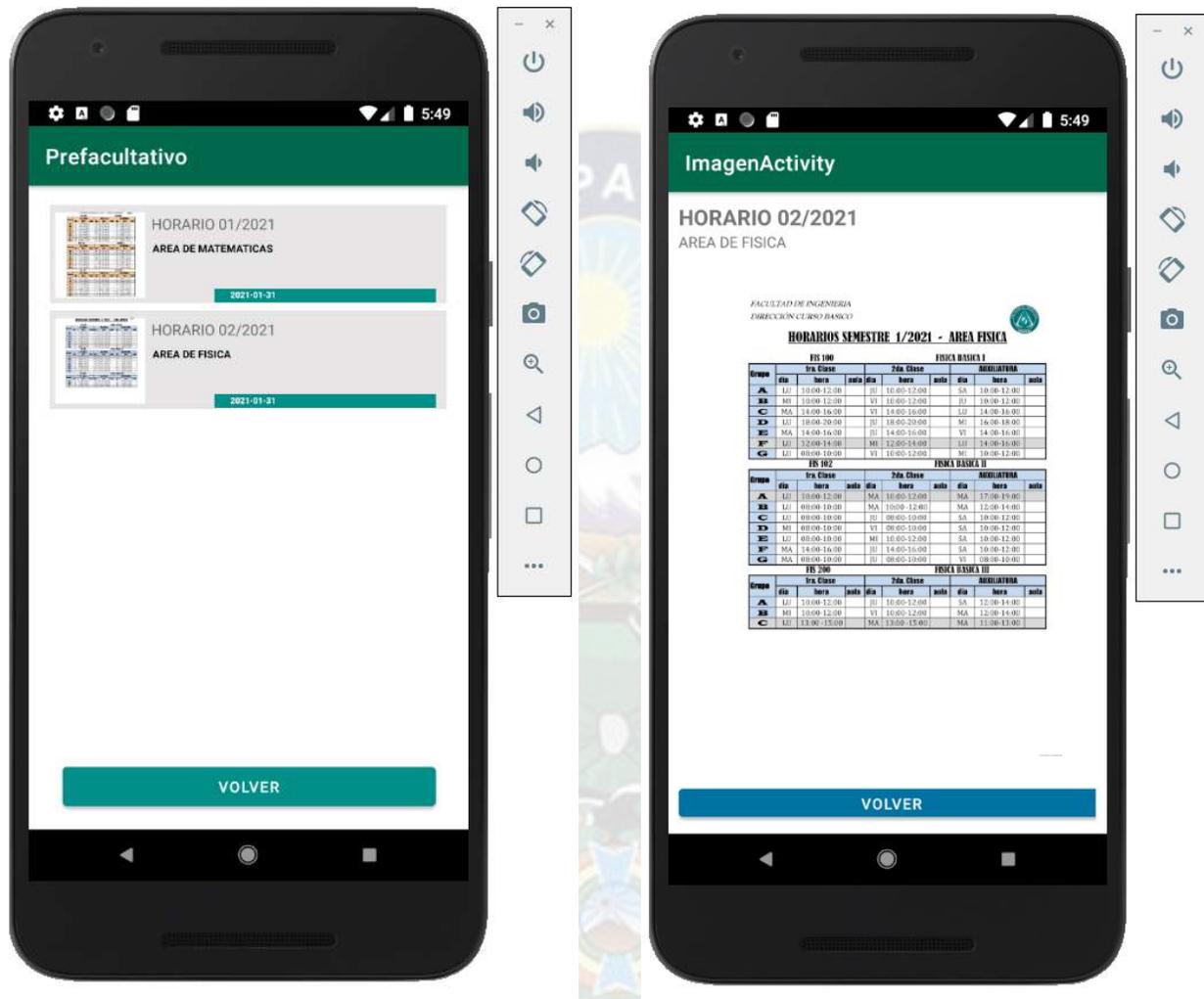


FIGURA 3.22. PANTALLA DE LA OPCIÓN HORARIOS.
FUENTE: CAPTURA DE PANTALLA, 2021.

Al elegir la opción “Cronograma de Exámenes” nos direcciona a otra ventana en la que se muestra todos los cronogramas actuales en curso con su respectiva bitácora, el usuario elige el cronograma de su interés y esta se despliega de tamaño completo para recabar la información pertinente. Del mismo modo lo realiza la opción “Distribución de Aulas”.

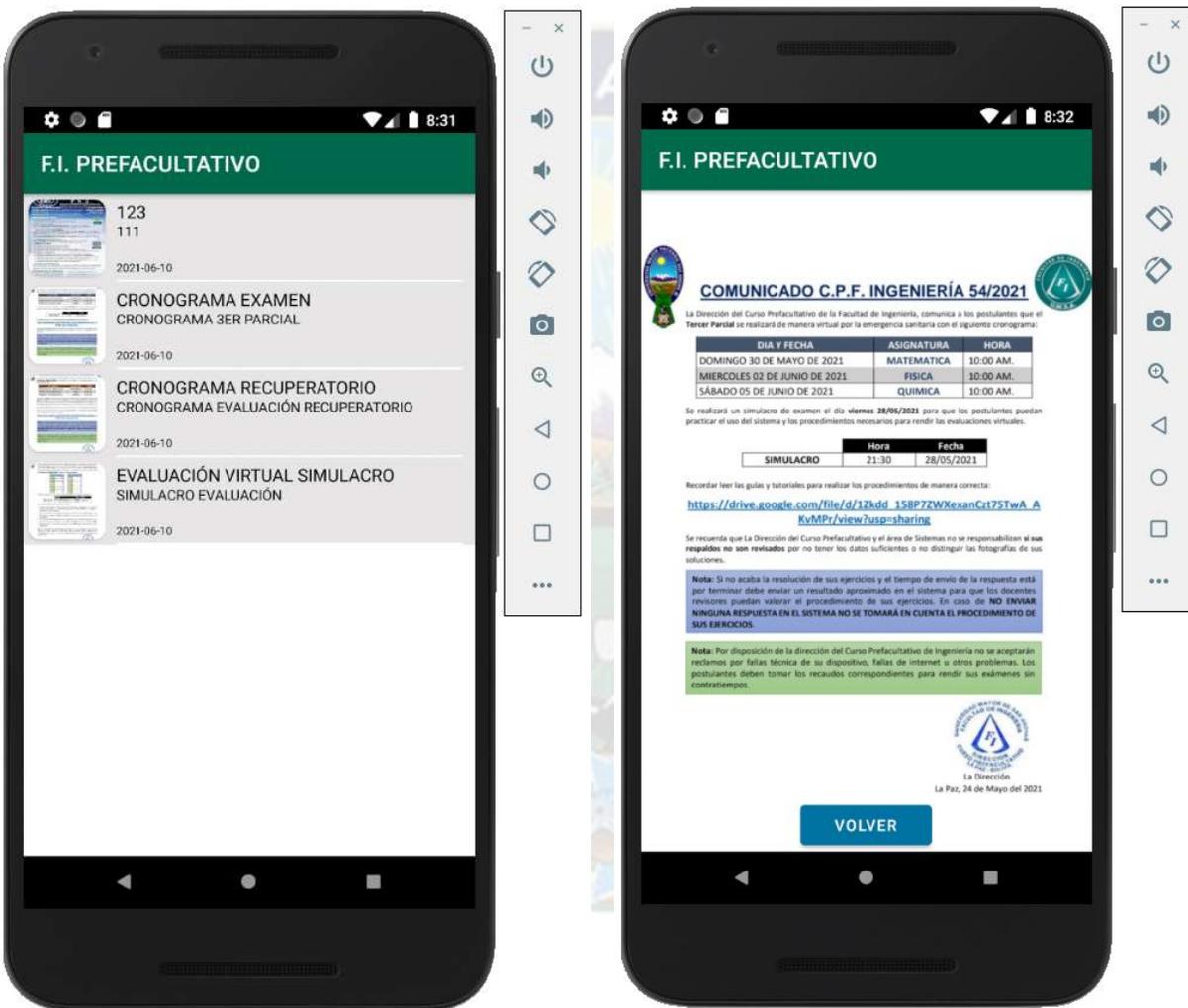


FIGURA 3.23. PANTALLA DE CRONOGRAMA DE EXÁMENES.

FUENTE: CAPTURA DE PANTALLA, 2021.

Al elegir la opción “Contáctanos” es necesario que el usuario esté inscrito y registrado como estudiante del Curso Prefacultativo, donde se le asigna un Usuario y Contraseña personal, con el cual puede ingresar al *Login* para tener una comunicación directa con el personal administrativo y poder realizar sus consultas de carácter personal.

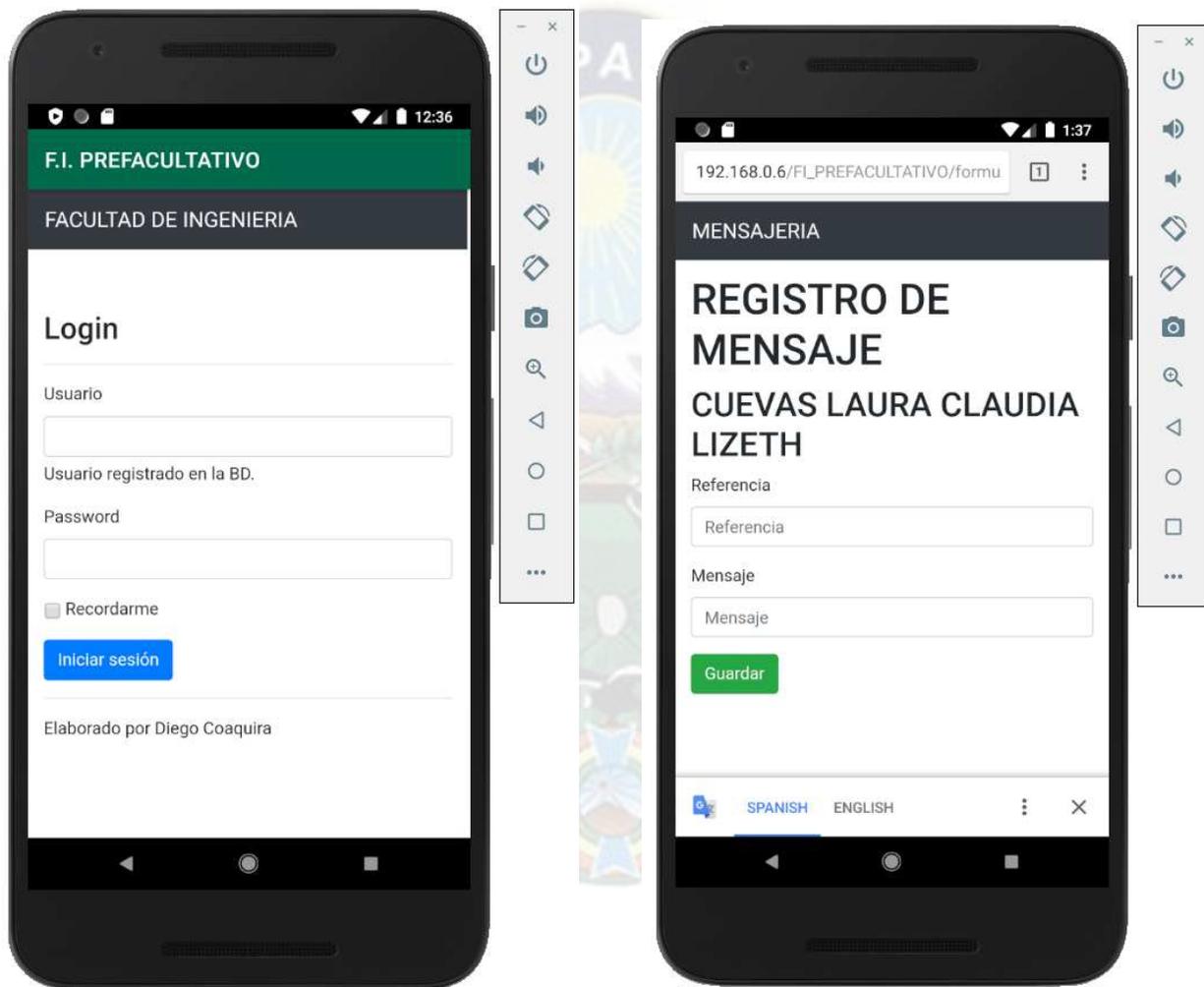


FIGURA 3.24. PANTALLA DE LA OPCIÓN CONTÁCTANOS (LOGIN).

FUENTE: CAPTURA DE PANTALLA, 2021.

3.11.2. Interfaces del sistema para el ADMINISTRADOR

Del mismo modo, es necesario realizar el diseño de la interfaz del Administrador, puesto que dicho actor puede ser una persona sin conocimientos en programación. Lo que el Administrador requiere es una interfaz amigable que le permita colgar las distintas opciones para el usuario. El Administrador tendrá su propio Usuario y Contraseña para administrar los diferentes comunicados.

En la siguiente figura se muestra la interfaz del Administrador, para su ingreso debe colocar su Usuario y Contraseña personal.

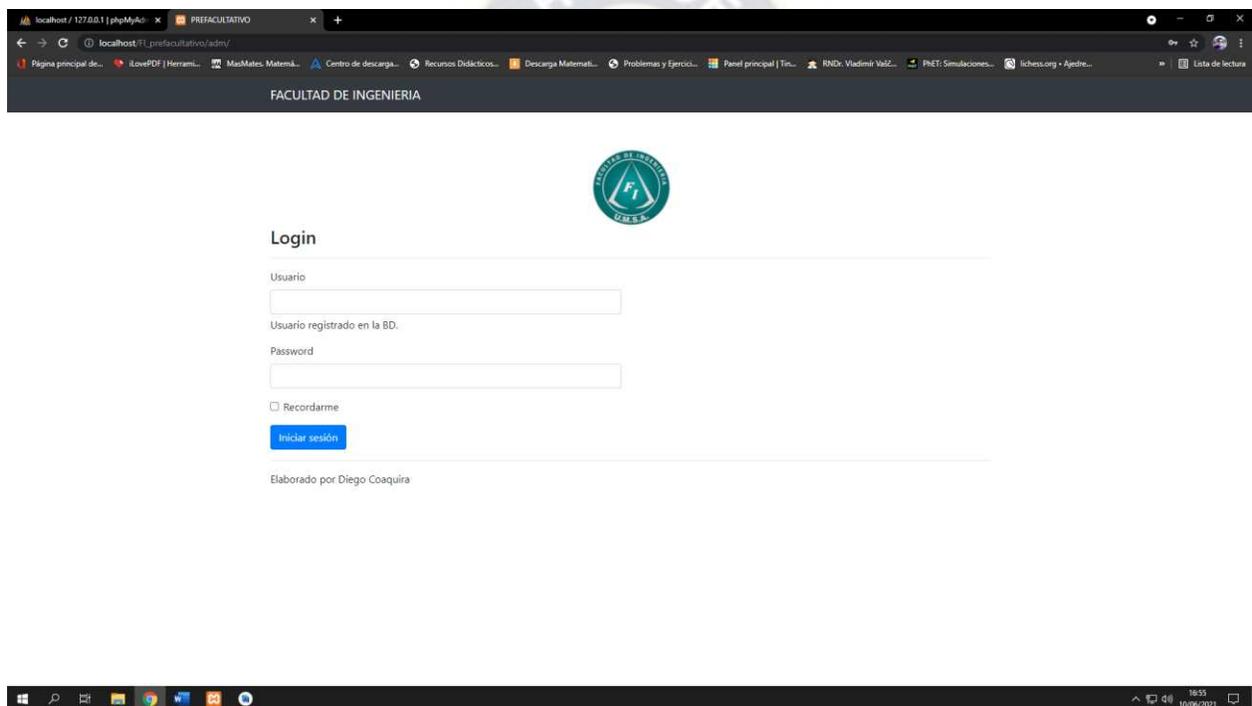


FIGURA 3.25. PANTALLA DE INGRESO AL SISTEMA ADMINISTRADOR.
FUENTE: CAPTURA DE PANTALLA, 2021.

El administrador tiene las siguientes opciones: Leer y responder los mensajes recibidos, enviar mensajes a todos de manera simultánea,

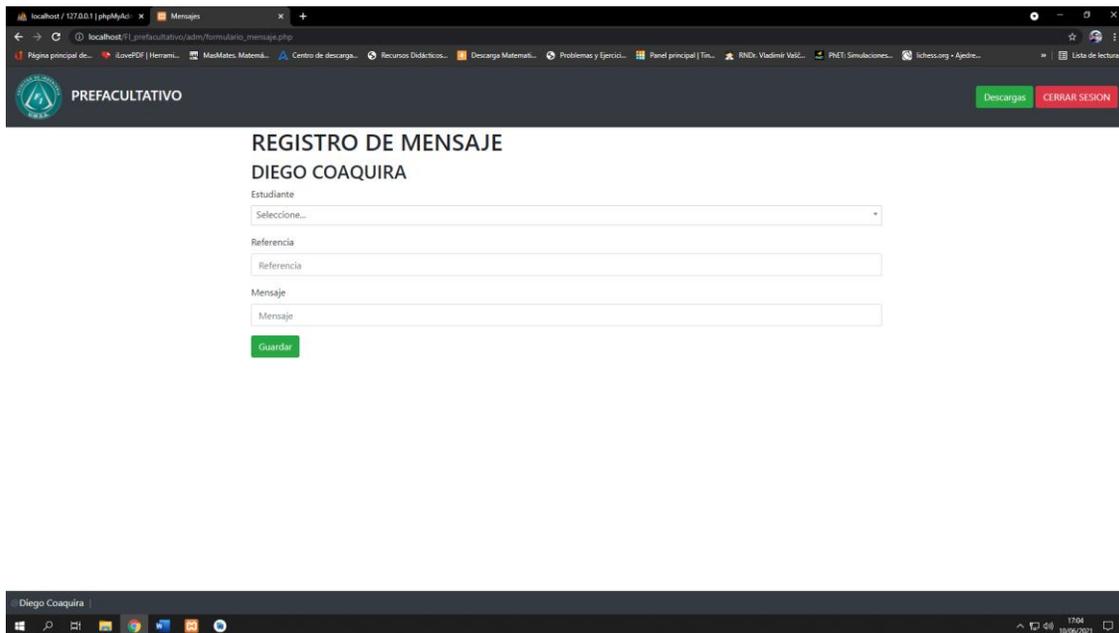


FIGURA 3.26. PANTALLA DE ENVÍO DE MENSAJES DEL ADMINISTRADOR.
FUENTE: CAPTURA DE PANTALLA, 2021.

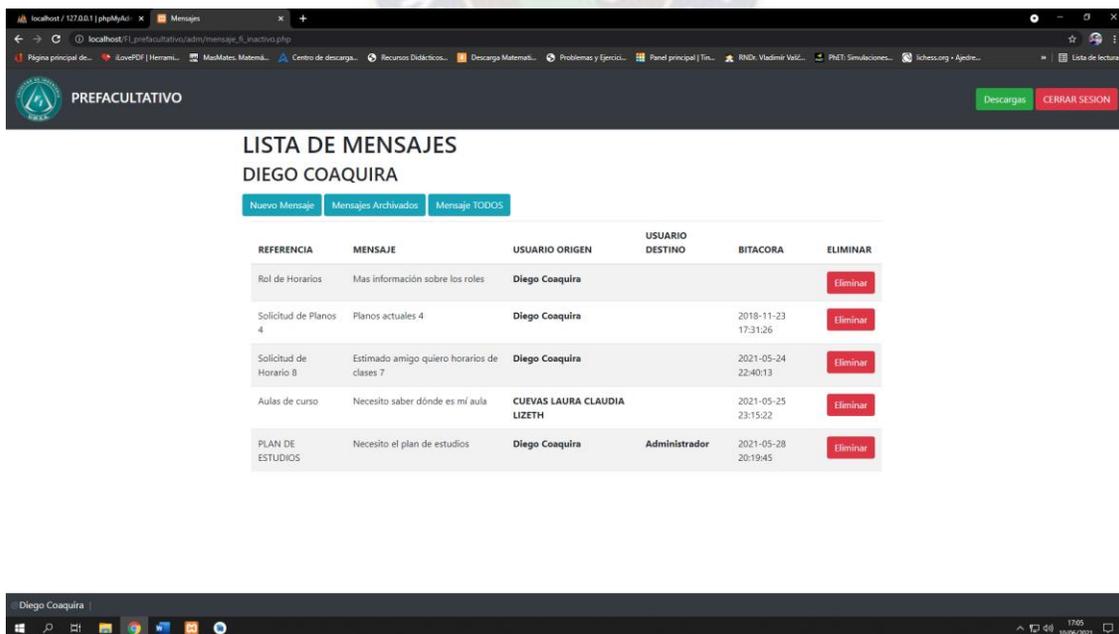


FIGURA 3.27. PANTALLA DE RECEPCIÓN DE MENSAJES DEL ADMINISTRADOR.
FUENTE: CAPTURA DE PANTALLA, 2021.

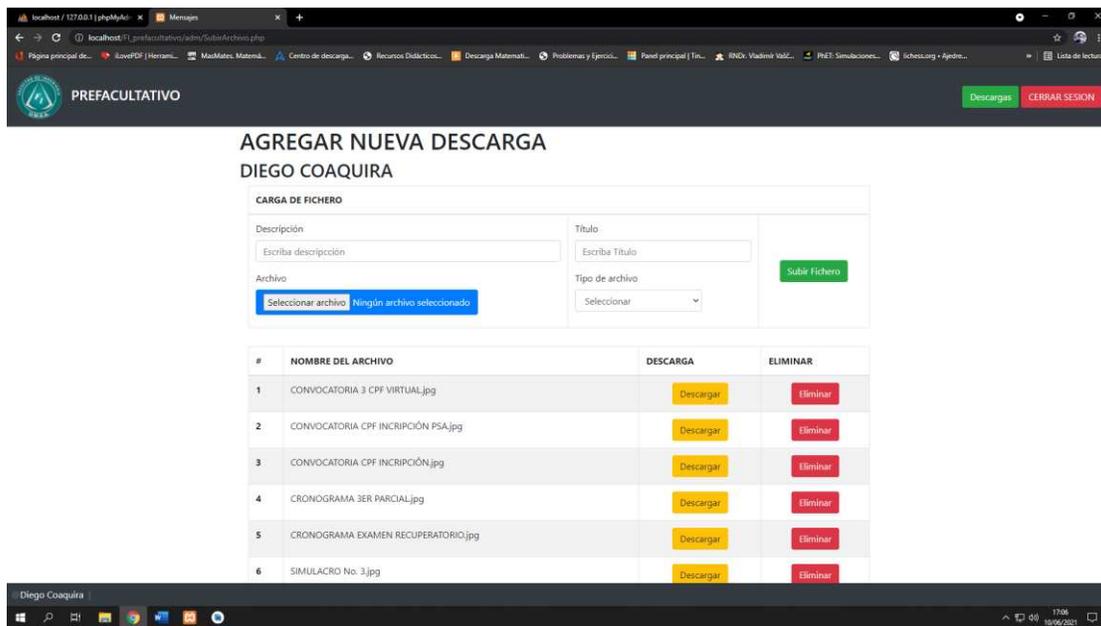


FIGURA 3.28. PANTALLA DE SUBIDA DE ARCHIVOS DEL ADMINISTRADOR.
FUENTE: CAPTURA DE PANTALLA, 2021.

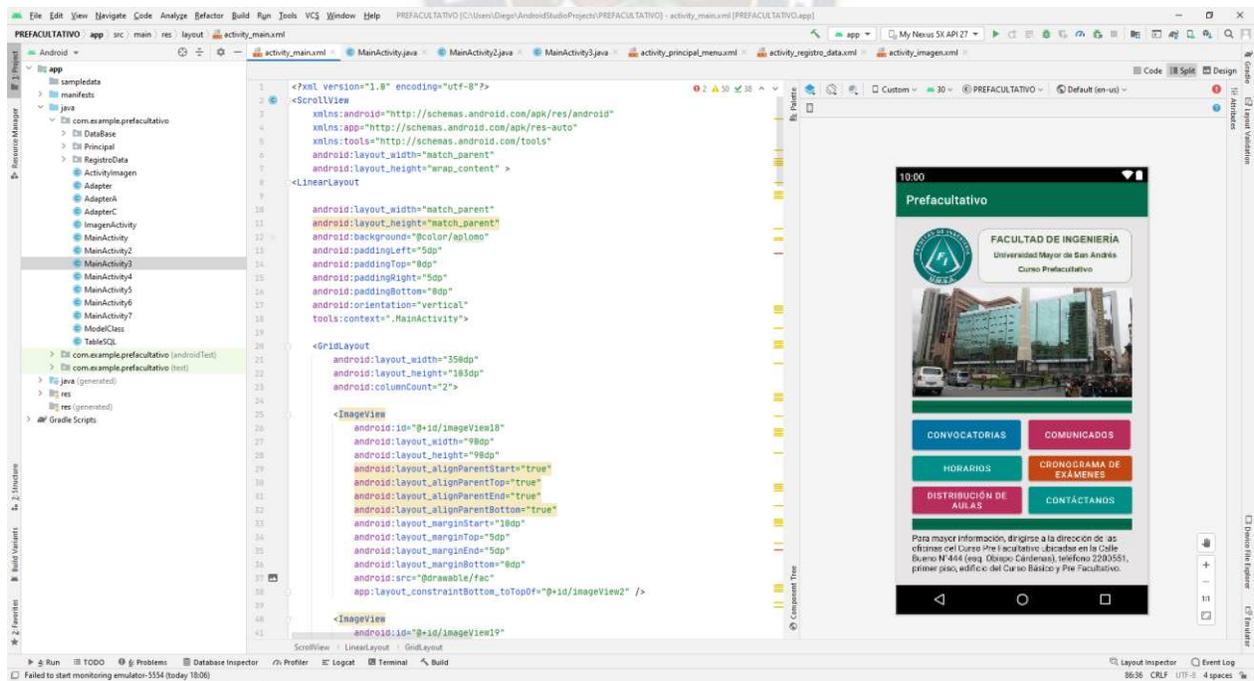


FIGURA 3.29. LÍNEAS DE CÓDIGO DEL FICHERO ACTIVITY_MAIN.XML.
FUENTE: CAPTURA DE PANTALLA, 2021.

```
1 package com.example.prefacultativo;
2
3 import ..;
4
5
6
7
8
9
10
11
12
13
14 public class MainActivity extends AppCompatActivity {
15
16     Button btnSgte, btnSgte1, btnSgte2, btnSgte3, btnSgte4, btnSgte5;
17     private String url;
18
19
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25         btnSgte = (Button) findViewById(R.id.btnPantalla2);
26         btnSgte1 = (Button) findViewById(R.id.btnPantalla3);
27         btnSgte2 = (Button) findViewById(R.id.btnPantalla4);
28         btnSgte3 = (Button) findViewById(R.id.btnPantalla5);
29         btnSgte4 = (Button) findViewById(R.id.btnPantalla6);
30         btnSgte5 = (Button) findViewById(R.id.btnPantalla7);
31         btnSgte.setTextColor(Color.parseColor("ffffff"));
32         btnSgte1.setTextColor(Color.parseColor("ffffff"));
33         btnSgte2.setTextColor(Color.parseColor("ffffff"));
34         btnSgte3.setTextColor(Color.parseColor("ffffff"));
35         btnSgte4.setTextColor(Color.parseColor("ffffff"));
36         btnSgte5.setTextColor(Color.parseColor("ffffff"));
37
38
39
40
41         btnSgte.setOnClickListener(new View.OnClickListener() {
42             @Override
43             public void onClick(View view) {
44                 Intent intent = new Intent( packageContext: MainActivity.this, MainActivity2.class);
45                 startActivity(intent);
46             }
47         });
48
49         btnSgte1.setOnClickListener(new View.OnClickListener() {
50             @Override
51             public void onClick(View view) {
52                 Intent intent = new Intent( packageContext: MainActivity.this, MainActivity3.class);
53                 startActivity(intent);
54             }
55         });
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

FIGURA 3.30. LÍNEAS DE CÓDIGO DEL FICHERO MAINACTIVITY.JAVA.
FUENTE: CAPTURA DE PANTALLA, 2021.

CAPÍTULO IV

ANÁLISIS DE COSTO DEL PROYECTO

4.1. Costos de Investigación

Para la realización de la presente investigación se tienen las siguientes actividades con su respectiva estructura de costos:

ACTIVIDADES	CANTIDAD	VALOR UNITARIO Bs.	VALOR TOTAL Bs.
Prueba del sistema			
Smartphone Samsung J2 Pro	1	600,00	600,00
Investigación documental			
Adquisición de bibliografía	Global	300,00	300,00
Fotocopiado	Global	100,00	100,00
Transporte	Global	50,00	50,00
Conexión a Internet VDSL	Global	310,00	310,00
Tratamiento y análisis de datos			
Material de escritorio	Global	100,00	100,00
Uso de computadora	Global	70,00	70,00
Elaboración del documento de tesis			
Material de escritorio	Global	100,00	100,00
Empastados y anillados	Global	300,00	300,00
Actividades imprevistas			
Material de escritorio	Global	50,00	50,00
Defensa de tesis	Global	200,00	200,00
TOTAL			2180,00

TABLA 4.1. COSTOS DE LA INVESTIGACIÓN.

FUENTE: ELABORACIÓN PROPIA, 2021.

4.2. Costo de equipos

El equipo empleado fue una PC de escritorio como se detalla a continuación:

ACTIVIDADES	CANTIDAD	VALOR UNITARIO Bs.	VALOR TOTAL Bs.
Laptop marca HP que cuenta con: <ul style="list-style-type: none"> • Intel Core i5-1035G1 (frecuencia base de 1,0 GHz, hasta 3,6 GHz con tecnología Intel Turbo Boost, caché L3 de 6 MB, 4 núcleos) • 15.6" diagonal HD, BrightView, micro-edge, retroiluminación WLED, pantalla táctil, 220 nits, 45% NTSC (1366 x 768) 8 GB DDR4-2666 SDRAM (1 x 8 GB) 1 TB 5400 rpm SATA HDD • Gráficos Intel UHD; cámara HP TrueVision HD con micrófono digital integrado de doble matriz • 1 velocidad de señalización USB tipo C de 5 Gbps; 2 velocidad de señalización USB tipo A de 5 Gbps; 1 HDMI 1.4; 1 RJ-45; 1 pin inteligente de CA; 1 combo de auriculares/micrófono • Windows 10 Home 	1	4000,00	4000,00
TOTAL			4000,00 Bs

TABLA 4.2. COSTO DEL EQUIPO EMPLEADO.

FUENTE: ELABORACIÓN PROPIA, 2021.

4.3. Costos de Software

Para el análisis de costos de software se empleó COCOMO II que es un modelo de estimación de costos para el presente proyecto que determina el valor de la inversión para el desarrollo e implementación de la base de datos de la aplicación de consulta de información, combinando los métodos de estimación de esfuerzo, tiempo y costo que son las más utilizadas en el desarrollo de proyectos informáticos.

4.3.1. Punto de Función

Primero se debe hallar el punto de función para poder obtener el costo del proyecto, para ello se analiza cada una de las pantallas del sistema.

Número de entradas de usuario

Son los que proporcionan datos al sistema para realizar las distintas operaciones de altas, bajas, registros, modificaciones, acceso, etc.

Nro.	ENTRADAS DE USUARIO
1	Pantalla de ingreso a la aplicación
2	Registro de usuarios
3	Edición de datos de usuarios
4	Solicitud de reportes de usuarios
5	Ingreso de Usuario y <i>Password</i>

TABLA 4.3. ENTRADAS DEL SISTEMA.

FUENTE: ELABORACIÓN PROPIA, 2021.

Número de salidas de usuario

Son las que proporcionan información procesada por el sistema que son gestionadas para el usuario.

Nro.	SALIDAS DE USUARIO
1	Pantalla de bienvenida al usuario
2	Listado de usuarios
3	Reporte sobre comunicados, convocatorias, horarios y cronograma
4	Confirmación de datos de usuarios
5	Bandeja de entrada de correo electrónico

TABLA 4.4. SALIDAS DEL SISTEMA.

FUENTE: ELABORACIÓN PROPIA, 2021.

Número de peticiones de usuario

Es una entrada de tipo interactiva que proporciona una salida o respuesta al usuario.

Nro.	PETICIONES DE USUARIO
1	Autenticación de usuario
2	Obtener listado de usuarios
3	Obtener reporte sobre comunicados, convocatorias, horarios y cronograma
4	Obtener solicitud de datos de usuarios
5	Obtener ingreso a correo electrónico

TABLA 4.5. PETICIONES DE USUARIO.

FUENTE: ELABORACIÓN PROPIA, 2021.

Número de archivos

Es el grupo lógico de datos que puede ser parte de la interface y de la base de datos.

Nro.	ARCHIVOS
1	<i>fragment_first.xml</i>
2	<i>MainActivity.java</i>

TABLA 4.6. ARCHIVOS DE LA INTERFACE Y LA BASE DE DATOS.

FUENTE: ELABORACIÓN PROPIA, 2021.

Una vez contando con todos los datos sobre las entradas, salidas, etc., se pasa a agruparlas para obtener el parámetro de medición.

Parámetros de medición	Cuentas	Factor de ponderación			Total
Nro. de entradas	5	3	4	6	15
Nro. de salidas	5	4	5	7	20
Nro. de peticiones	5	5	4	6	20
Nro. de archivos	2	7	10	15	14
TOTAL EN CUENTA					69

TABLA 4.7. PARÁMETROS DE MEDICIÓN DEL SISTEMA.

FUENTE: ELABORACIÓN PROPIA, 2021.

Cálculo de factor de ajuste de la complejidad

Los valores son respondidos en la escala de 0 a 5, donde el 0 significa que no es importante y el 5 que es fundamental para el sistema.

FACTOR DE COMPLEJIDAD	VALOR
Requiere copias de seguridad	3
Necesita comunicación de datos	3
Existe funciones de procedimiento distribuido	2
Rendimiento critico	2
Se ejecuta en un entorno operativo existente	2
Se requiere entrada de datos en línea (on-line)	4
Transacciones de entradas en múltiples pantallas	3
Archivos maestros actualizados en línea (on-line)	4
Complejidad de valores del dominio de información	3
Complejidad del procedimiento interno	3
Código diseñado para la reutilización	4
Conversión/instalación en diseño	2
Instalaciones Múltiples	2
Aplicación diseñada para el cambio	4
SUMA de $f() = \sum fi()$	41

TABLA 4.8. FACTOR DE COMPLEJIDAD DEL SISTEMA.

FUENTE: ELABORACIÓN PROPIA, 2021.

Ahora se procede a calcular el factor de ajuste.

FACTOR DE AJUSTE	$0.65 + 0.01 * \sum fi ()$
Factor de ajuste	$0.65 + 0.01 * 41$
Factor de ajuste	1.06

TABLA 4.9. FACTOR DE AJUSTE.

FUENTE: ELABORACIÓN PROPIA, 2021.

Calculando el punto función con los datos obtenidos.

PUNTO DE FUNCIÓN	CUENTA TOTAL*FACTOR DE AJUSTE
Punto de función	69*1.06
Punto de función	73.14

TABLA 4.10. PUNTO DE FUNCIÓN.
FUENTE: ELABORACIÓN PROPIA, 2021.

4.3.2. Costos del Sistema Desarrollado

Se utilizó como punto función $PF = 73.14$ a continuación, realizamos la conversión de punto función a miles de líneas de código mediante la siguiente tabla.

LENGUAJE	FACTOR LCD/PF
Java	53
Java Script	47
Visual Basic	46
ASP	36
Visual C++	34
PHP	12
Ensamblador	320
C	150

TABLA 4.11. FACTOR LCD/PF DE LENGUAJES DE PROGRAMACIÓN.
FUENTE: QSM, 2017.

Ahora se aplican las fórmulas básicas de esfuerzo, tiempo calendario y personal requerido. Las fórmulas de COCOMO II empleadas son las siguientes:

$$E_D = 2,4 (KDLC)^{1,05}$$

$$T_D = 2,5 (KDLC)^{0,38}$$

Donde:

E_D = Esfuerzo aplicado en personas por mes.

T_D = Tiempo de desarrollo en mes.

KLDC = Número estimado de líneas de código distribuidas.

KLDC = PF * Factor LDC

KLDC = 73.14 * 53 = 3876.42/1000 = 3.87

KLDC = 3.87

Para la evaluación de software existen tres categorías de desarrollo de software:

Orgánicos

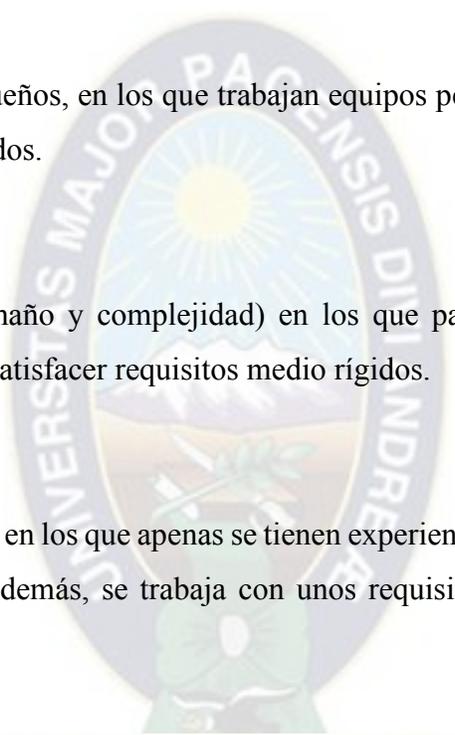
Relativamente sencillos y pequeños, en los que trabajan equipos pequeños con experiencia, sobre un conjunto de datos poco rígidos.

Semiocoplados

Proyectos intermedios (en tamaño y complejidad) en los que participan equipos con variados niveles de experiencia, deben satisfacer requisitos medio rígidos.

Empotrados

Proyectos bastantes complejos, en los que apenas se tienen experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.



Tipo de proyecto	a	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semiocoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

El presente proyecto se adecua a la categoría de orgánico por lo tanto reemplazando en las fórmulas anteriores se tiene:

$$E_D = 2,4 (3,87)^{1,05}$$

$$E_D = 8,94 \text{ persona/mes}$$

$$T_D = 2,5 (8,94)^{0,38}$$

$$T_D = 5,98 \text{ meses}$$

El personal requerido para el desarrollo del proyecto se obtiene de la siguiente fórmula:

$$\text{Numero de programadores} = \frac{E_D}{T_D} = \frac{8,94}{5,98} = 1,49 \approx 1$$

Por lo tanto, se necesita 1 programador para el desarrollo de la aplicación. El costo de salario para un programador es de 600 \$us por mes. Por lo tanto, con este dato la estimación del costo del software se calculó con la siguiente fórmula:

$$\text{Costo Software} = \# \text{ de programadores} * \text{Salario de Programador} * \# \text{ de meses}$$

$$\text{Costo Software} = 1 * 700 * 6 \text{ (6 meses)} = 4200 \text{ $us o } 29022 \text{ Bs.}$$

Por lo tanto, el costo de desarrollo del software es de 4200 \$us o 29022 Bs.

4.4. Costo Total

Para estimar el costo total de la solución se tomarán en cuenta los siguientes costos: costo de la elaboración del proyecto, costo del equipo informático empleado y costos de desarrollo de la aplicación móvil.

ACTIVIDADES	CANTIDAD	VALOR UNITARIO Bs.	VALOR TOTAL Bs.
Costo de investigación	Global	2180,00	2180,00
Costo de equipos	Global	2000,00	4000,00
Costo de software	Global	2000,00	29022,00
TOTAL			35202,00 Bs

TABLA 4.12. COSTO TOTAL.
FUENTE: ELABORACIÓN PROPIA, 2021.

CAPÍTULO V

CONCLUSIONES, RECOMENDACIONES Y MEJORAS

5.1. Conclusiones

- *Realizar un diagnóstico sobre el actual proceso de gestión académica de los postulantes en el curso Prefacultativo para detectar necesidades y requerimientos de los usuarios.*

A partir del diagnóstico realizado sobre los medios de comunicación que emplea el curso Prefacultativo se pudo verificar que realiza la publicación de sus comunicados de manera física en un panel de información que se encuentra en la puerta de la secretaria del Curso Prefacultativo localizado en predios del edificio ex Príncipe de Paz (en la calle Bueno). Asimismo, realiza la publicación de los comunicados, convocatorias y horarios mediante la página web de la UMSA la página web del Curso Prefacultativo. También, dentro de la página web del Curso Prefacultativo se pudo constatar que cuentan con un sistema de información para los alumnos que les provee acceso, tanto para inscribirse al Curso Prefacultativo, como para averiguar las notas una vez finalizado el mismo. Finalmente, poseen una *Fan Page* en la Red Social Facebook para la publicación de toda la información para los postulantes. De igual forma indican un número de WhatsApp en la modalidad de *Call Center* para absolver dudas de los alumnos. Como se pudo observar no cuentan con una aplicación móvil para la publicación de información del Curso Prefacultativo.

- *Establecer una metodología ágil RAD enfocada al desarrollo de una plataforma digital para teléfonos móviles inteligentes.*

El modelo de desarrollo rápido de aplicaciones (*RAD*, por sus siglas en inglés) es una metodología ágil de desarrollo de software que prioriza a las entregas e iteraciones rápidas de prototipos. A diferencia de la metodología de cascada, *RAD* toma más en cuenta el uso de software y la opinión del usuario que la planificación rigurosa y el registro de los requisitos, que es lo más importante al momento de utilizar la aplicación. En el ámbito educativo universitario de pregrado, debido a los espacios cortos de tiempo, al momento de desarrollar

un producto de software de baja complejidad, como es este caso, se puede emplear esta metodología por la necesidad de construir rápido una aplicación y a la vez que una misma persona hace de analista, diseñador y programador. Este enfoque se caracteriza por el uso de prototipos, que no son más que una aproximación o representación del sistema, que permiten a los usuarios visualizar y responder en función de la implementación realizada. Implementado en primera instancia prototipos, los analistas pueden mostrar a los usuarios las entradas, los estadios intermedios y las salidas del sistema, de una manera que los usuarios entienden.

- *Diseñar el sistema a nivel físico y lógico empleando el Lenguaje Unificado de Modelamiento o UML, brindando una interfaz amigable con facilidad de uso.*

El Lenguaje Unificado de Modelamiento (*UML*, por sus siglas en inglés) es una herramienta gráfica que se emplea para especificar métodos o procesos realizados por un sistema, a través de una serie de diagramas. Posee una serie de herramientas que permiten mostrar el programa en sus diferentes etapas o procesos, delimitarlos y organizarlos de tal forma que sean entendibles por la persona que va a desarrollar el sistema. Cabe resaltar que UML no es un lenguaje de programación, más bien es un lenguaje de modelado que permite modelar la estructura del programa para posteriormente poder codificarlo a un lenguaje específico.

- *Desarrollar un prototipo de la plataforma digital para teléfonos móviles inteligentes empleando software libre con la seguridad adecuada para la gestión académica de los postulantes del curso Prefacultativo.*

Para el desarrollo de la plataforma digital para teléfonos inteligentes se empleó la arquitectura cliente-servidor donde el cliente realizará la petición al servidor y éste a su vez le dará la respuesta. En este caso el sistema cuenta con un servidor de base de datos donde se almacenarán los registros, pedidos del cliente y a su vez recibir información actualizada, para esto el cliente debe contar con acceso a Internet.

El proyecto en su totalidad se desarrolló en dos partes. La primera parte es la aplicación móvil para Android y para su funcionamiento se empleará un Smartphone con este sistema operativo, con la aplicación instalada y con acceso a Internet ya que se debe consultar desde cualquier

parte del mundo, esta se conecta mediante la aplicación que será descargada de *Google Play Store*, y la segunda parte es que la comunicación con la base de datos del curso Prefacultativo de la Facultad de Ingeniería de la UMSA la cual almacena la información de los requerimientos del sistema y la publicación de la información. Para el desarrollo de la aplicación móvil se empleó Android Studio que a pasado a ser el entorno recomendado para el desarrollo de aplicaciones en *Android*, asimismo para la base de datos se utilizó *XAMPP* que es un servidor independiente multiplataforma de código libre. Incluye el servidor Apache, la base de datos *MySQL* y *SQLite* con sus respectivos gestores, el lenguaje de programación *PHP* y el intérprete *Perl*.

- *Realizar pruebas de la plataforma digital de gestión académica para verificar que cumpla con los requisitos establecidos.*

Se implementó a nivel prototipo una aplicación móvil bajo la plataforma Android para la publicación de información e inscripción de los postulantes al Curso Prefacultativo de la Facultad de Ingeniería de la UMSA, la cual coadyuva a los mismos a contar con información sobre comunicados, convocatorias, horarios y cronogramas de actividades, entre otros, independientemente en el lugar donde se encuentren. La aplicación desarrollada e implementada a nivel prototipo cumple con el propósito para la cual fue creada además está basada en la normativa vigente de la UMSA.

5.2. Recomendaciones y Mejoras

- La aplicación a futuro puede ser vinculada al file personal del estudiante, con la finalidad de mostrar sus notas, asignar grupos, mostrar horarios, etc., datos que darán seguridad al estudiantado.
- Se recomienda tanto a los usuarios y administradores de la aplicación móvil, revisar la funcionalidad del sistema con respecto a la consulta de información y el acceso al Sistema de Información para la verificación de notas e inscripción con el propósito de que no existan errores que perjudiquen a los postulantes.

- Si se implementa el prototipo en el área de sistemas del Curso Prefacultativo, es recomendable utilizar herramientas de desarrollo proporcionadas por la misma, ya que brindan estándares de diseño y desarrollo, así como soporte y documentación.
- Se debe utilizar metodologías ágiles de desarrollo de software con un control de versiones para futuras actualizaciones según se incrementen o cambien los requerimientos institucionales o reglamentarios.
- Es recomendable implementar la aplicación en otras plataformas móviles, como IOS, aumentando algunas funcionalidades y servicios de actualización automáticos para mantener concordancia en la información.



5.3. Bibliografía

(Anthony-Mauricio-Villanueva-Arrestegui, 2018)

Anthony-Mauricio-Villanueva-Arrestegui. (2018, julio 6). Arquitectura de Sistemas Operativos Para Móviles. Recuperado 16 de mayo de 2021, de Pdfslide.net website: <https://pdfslide.net/documents/arquitectura-de-sistemas-operativos-para-moviles.html>

(«Arquitectura Cliente Servidor», s. f.)

Arquitectura Cliente Servidor. (s. f.). Recuperado 16 de mayo de 2021, de Ecured.cu website: https://www.ecured.cu/Arquitectura_Cliente_Servidor

(«Arquitectura de Android», s. f.)

Arquitectura de Android. (s. f.). Recuperado 16 de mayo de 2021, de Androidcurso.com website: <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/99-arquitectura-de-android>

(Ballesteros, s. f.)

Ballesteros, A. C. (s. f.). CURSO PREFACULTATIVO. Recuperado 16 de mayo de 2021, de Edu.bo website: <http://prefing.umsa.edu.bo/>

(Castellanos, 2016)

Castellanos, L. R. (2016, octubre 3). Sistemas Operativos Móviles. Recuperado 16 de mayo de 2021, de Dtyoc.com website: <https://dtyoc.com/2016/10/03/sistemas-operativos-moviles/>

(«Definición de smartphone — Definición.de», s. f.)

Definición de smartphone — Definición.de. (s. f.). Recuperado 16 de mayo de 2021, de Definición.de website: <https://definicion.de/smartphone/>

(Figuroa, s. f.)

Figuroa, M. A. A. (s. f.). Calidad en la Industria del Software. La Norma ISO-9126. Recuperado 16 de mayo de 2021, de Unam.mx website:

<http://www.nacionmulticultural.unam.mx/empresasindigenas/docs/2094.pdf>

(«Front end», 2020)

Front end. (2020, agosto 7). Recuperado 16 de mayo de 2021, de Neoattack.com website: <https://neoattack.com/neowiki/front-end/>

(Gómez & del C. López, s. f.)

Gómez, A., & del C. López, M. (s. f.). UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE. Recuperado 16 de mayo de 2021, de Wordpress.com website: <https://blogadmi1.files.wordpress.com/2010/11/cocom0llfull.pdf>

(Hernández, 2014)

Hernández, J. (2014). Análisis y Desarrollo Web. Jesús Hernández.

(«Metodología RAD», s. f.)

Metodología RAD. (s. f.). Recuperado 16 de mayo de 2021, de Scribd.com website: <https://es.scribd.com/document/297175443/Metodologia-RAD>

(Rodríguez, 2014)

Rodríguez, T. (2014, marzo 17). Genymotion, el emulador más rápido de Android. Recuperado 16 de mayo de 2021, de Xatakandroid.com website: <https://www.xatakandroid.com/roms-android/genymotion-el-emulador-mas-rapido-de-android>

(«SQLite para Android: La herramienta definitiva», 2018)

SQLite para Android: La herramienta definitiva. (2018, marzo 23). Recuperado 16 de mayo de 2021, de Openwebinars.net website: <https://openwebinars.net/blog/sqlite-para-android-la-herramienta-definitiva/>

ANEXOS

Programación y Prueba de la Aplicación

Las líneas de código del fichero *activity_main.xml*:

```
1 <? xml version = "1.0" encoding = "utf-8"?>
2 <ScrollView
3     xmlns: android = "http://schemas.android.com/apk/res/android"
4     xmlns: app = "http://schemas.android.com/apk/res-auto"
5     xmlns: tools = "http://schemas.android.com/tools"
6     android: layout_width = "match_parent"
7     android: layout_height = "wrap_content" >
8 <LinearLayout
9     android: layout_width = "match_parent"
10    android: layout_height = "match_parent"
11    android: background = "@color/aplomo"
12    android: paddingLeft = "5dp"
13    android: paddingTop = "0dp"
14    android: paddingRight = "5dp"
15    android: paddingBottom = "0dp"
16    android: orientación = "vertical"
17    herramientas: contexto = ".MainActivity" >
18 <GridLayout
19     android: layout_width = "350dp"
20     android: layout_height = "103dp"
21     android: columnCount = "2" >
22 <ImageView
23     android: id = "@ + id / imageView18"
24     android: layout_width = "90dp"
25     android: layout_height = "90dp"
26     android: layout_alignParentStart = "true"
27     android: layout_alignParentTop = "true"
28     android: layout_alignParentEnd = "true"
29     android: layout_alignParentBottom = "true"
30     android: layout_marginStart = "10dp"
31     android: layout_marginTop = "5dp"
32     android: layout_marginEnd = "5dp"
33     android: layout_marginBottom = "0dp"
34     android: src = "@drawable/fac"
35     aplicación:layout_constraintBottom_toTopOf = "@ + id / imageView2" />
36 <ImageView
37     android: id = "@ + id / imageView19"
38     android: layout_width = "230dp"
39     android: layout_height = "80dp"
40     android: layout_alignParentStart = "true"
41     android: layout_alignParentTop = "true"
42     android: layout_alignParentEnd = "true"
43     android: layout_alignParentBottom = "true"
44     android: layout_marginStart = "10dp"
45     android: layout_marginTop = "10dp"
46     android: layout_marginEnd = "10dp"
47     android: layout_marginBottom = "0dp"
48     android: src = "@drawable/booo"
49     aplicación:layout_constraintBottom_toTopOf = "@ + id / imageView2"
50     aplicación: layout_constraintEnd_toEndOf = "padre"
51     app: layout_constraintHorizontal_bias = "0,834"
52     aplicación: layout_constraintStart_toStartOf = "padre"
53     aplicación: layout_constraintTop_toTopOf = "padre"
54     app: layout_constraintVertical_bias = "0,057" />
55 </ GridLayout>
56 <ImageView
57     android: id = "@ + id / imageView1"
```

```

58     android: layout_width = "330dp"
59     android: layout_height = "wrap_content"
60     android: layout_alignParentStart = "true"
61     android: layout_alignParentEnd = "true"
62     android: layout_marginStart = "10dp"
63     android: layout_marginTop = "0dp"
64     android: layout_marginEnd = "10dp"
65     aplicación: srcCompat = "@ drawable / img3" />
66 <ProgressBar
67     android: id = "@ + id / progressBar2"
68     style = "? android: attr / progressBarStyleHorizontal"
69     android: layout_width = "330dp"
70     android: layout_height = "15dp"
71     Android: layout_marginStart = "10DP"
72     android: layout_marginTop = "0DP"
73     android: layout_marginEnd = "10DP"
74     android: layout_marginBottom = "5 dp"
75     androide: Fondo = "@ Color / verr"
76     aplicación: layout_constraintEnd_toEndOf = "padre"
77     aplicación: layout_constraintStart_toStartOf = " parent "
78     app: layout_constraintTop_toTopOf = "@ + id / imageView2 " />
79 <GridLayout
80     android: layout_width = " wrap_content "
81     android: layout_height = " wrap_content "
82     android: paddingLeft = " 5dp "
83     android: paddingTop = " 0dp "
84     android: paddingRight = "5dp"
85     android: paddingBottom ="0dp"
86     android: columnCount = "2" >
87     <Botón
88         android: id = "@ + id / btnPantalla2"
89         android: layout_width = "150dp"
90         android: layout_height = "80dp"
91         android: layout_alignParentStart = "true"
92         android: layout_alignParentTop = " true "
93         android: layout_alignParentEnd = " true "
94         android: layout_alignParentBottom = " true "
95         android: layout_marginStart = " 10dp "
96         android: layout_marginTop = " 0dp "
97         android: layout_marginEnd = " 10dp "
98         android: layout_marginBottom = " 0dp "
99         android: backgroundTint = "@ color / azul"
100        android: textSize = "12sp"
101        android: text = "CONVOCATORIAS" />
102     <Botón
103         android: id = "@ + id / btnPantalla3"
104         android: layout_width = "150dp"
105         android: layout_height = "80dp"
106         android: layout_alignParentStart = "true"
107         android: layout_alignParentTop = "true"
108         android: layout_alignParentEnd = "true"
109         android: layout_alignParentBottom = "true"
110         android: layout_marginStart = "10dp"
111         android: layout_marginTop = "0dp"
112         android: layout_marginEnd = "10dp"
113         android: layout_marginBottom = "0dp"
114         android: textSize = "12sp"
115         android: backgroundTint = "@ color / arosa "
116         android: text = " COMUNICADOS " />
117 </GridLayout>
118 <GridLayout
119     android: layout_width = "wrap_content"
120     android: layout_height = "wrap_content"
121     android: paddingLeft = "5dp"
122     android: paddingTop = "0dp"
123     android: paddingRight = "5dp"
124     android: paddingBottom = "0dp"
125     android: columnCount = "2" >
126     <Botón
127         android: id = "@ + id / btnPantalla4"
128         android: layout_width = "150dp"

```

```

129         android:layout_height = "80dp"
130         android:layout_alignParentStart = "true"
131         android:layout_alignParentTop = "true"
132         android:layout_alignParentEnd = "true"
133         android:layout_alignParentBottom = "true"
134         android:layout_marginStart = "10dp"
135         android:layout_marginTop = "0dp"
136         android:layout_marginEnd = "10dp"
137         android:layout_marginBottom = "0dp"
138         android:textSize = "12sp"
139         android:backgroundTint = "@ color / averde"
140         android:text = "HORARIOS" />
141     <Botón
142         android:id = "@ + id / btnPantalla6"
143         android:layout_width = "150dp"
144         android:layout_height = "80dp"
145         android:layout_alignParentStart = "true"
146         android:layout_alignParentTop = "true"
147         android:layout_alignParentEnd = "true"
148         android:layout_alignParentBottom = "true"
149         android:layout_marginStart = "10dp"
150         android:layout_marginTop = "0dp"
151         android:layout_marginEnd = "10dp"
152         android:layout_marginBottom = "0dp"
153         android:textSize = "12sp"
154         android:backgroundTint = "@ color / anaranja"
155         android:text = "CONTACTANOS" />
156 </GridLayout>
157 <GridLayout
158     android:layout_width = "wrap_content"
159     android:layout_height = "wrap_content"
160     android:paddingLeft = "5dp"
161     android:paddingTop = "0dp"
162     android:paddingRight = "5dp"
163     android:paddingBottom = "0dp"
164     android:columnCount = "2" >
165     <Botón
166         android:id = "@ + id / btnPantalla5"
167         android:layout_width = "150dp"
168         android:layout_height = " 80dp "
169         android:layout_alignParentStart = "true"
170         android:layout_alignParentTop = "true"
171         android:layout_alignParentEnd = "true"
172         android:layout_alignParentBottom = "true"
173         android:layout_marginStart = "10DP"
174         android:layout_marginTop = "0DP"
175         android:layout_marginEnd = "10DP"
176         android:layout_marginBottom = "0DP"
177         android:textSize = "12sp"
178         android:backgroundTint = "@ color / arosa"
179         android:text = "CRONOGRAMA DE EXÁMENES" />
180     <Botón
181         android:id = "@ + id / btnPantalla7"
182         android:layout_width = "150dp"
183         android:layout_height = "80dp"
184         android:layout_alignParentStart = "true"
185         android:layout_alignParentTop = "true"
186         android:layout_alignParentEnd = "true"
187         android:layout_alignParentBottom = "true"
188         android:layout_marginStart = "10dp"
189         android:layout_marginTop = "0dp"
190         android:layout_marginEnd = "10dp"
191         android:layout_marginBottom = "0dp"
192         android:textSize = "12sp"
193         android:backgroundTint = "@ color / azul"
194         android:text = "DISTRIBUCIÓN DE AULAS" />
195 </GridLayout>
196 <ProgressBar
197     android:id = "@ + id / progressBar2"
198     style = "? android:attr / progressBarStyleHorizontal"
199     android:layout_width = "330dp"

```

```

200     android: layout_height = "15dp"
201     android: layout_marginStart = "10dp"
202     android: layout_marginTop = "5dp"
203     android: layout_marginEnd = "10dp"
204     android: layout_marginBottom = "5dp"
205     android: background = "@ color / verr"
206     aplicación:layout_constraintEnd_toEndOf = "principal"
207     aplicación: layout_constraintStart_toStartOf = "principal"
208     app: layout_constraintTop_toTopOf = "@ + id / imageView2" />
209 <TextView
210     android: id = "@ + id / textView4"
211     android: layout_width = "330dp"
212     android: layout_height = "87dp"
213     android: layout_alignParentStart = "true"
214     android: layout_alignParentTop = "true"
215     android: layout_alignParentEnd = "true"
216     android: layout_alignParentBottom = "true"
217     android: layout_marginStart = "10dp"
218     android: layout_marginTop = "5dp"
219     android: layout_marginEnd = "10dp"
220     android: layout_marginBottom = "0dp"
221     android: justificationMode = "inter_word"
222     android: textSize ="14sp"
223     android: text = "Para mayor información, dirigirse a la dirección de las oficinas del
224 Curso Pre Facultativo ubicadas en la Calle Bueno N ° 444 (esq. Obispo Cárdenas), teléfono
225 2200551, primer piso, edificio del Curso Básico y Pre Facultativo. "
226     android: textColor = "# 070707" />
227 </LinearLayout>
228 </ScrollView>

```

Las líneas de código del fichero *MainActivity.java* se muestra a continuación:

```

1  paquete com . ejemplo . prefacultativo ;
2
3  importar androidx.appcompat.app.AppCompatActivity ;
4  importar android.content.Intent ;
5  importar android.graphics.Color ;
6  importar android.net.Uri ;
7  importar android.os.Bundle ;
8  importar android.view.View ;
9  importar android.webkit.WebSettings ;
10 importar android.webkit.WebView ;
11 importar android.widget.Button ;
12
13 público de clase MainActivity extiende AppCompatActivity {
14
15     Botón btnSgte , btnSgte1 , btnSgte2 , btnSgte3 , btnSgte4 , btnSgte5 ;
16     URL de cadena privada ;
17
18     @Override
19     protected void onCreate ( Bundle savedInstanceState ) {
20         super . onCreate ( savedInstanceState ) ;
21         setContentView ( R . diseño . activity_main ) ;
22         btnSgte = ( Button ) findViewById ( R . ID . btnPantalla2 ) ;
23         btnSgte1 = ( Button ) findViewById ( R . ID . btnPantalla3 ) ;
24         btnSgte2 = ( Button ) findViewById ( R . ID . btnPantalla4 ) ;
25         btnSgte3 = ( Button ) findViewById ( R . ID . btnPantalla5 ) ;
26         btnSgte4 = ( Button ) findViewById ( R . ID . btnPantalla6 ) ;
27         btnSgte5 = ( Botón ) findViewById ( R . id . btnPantalla7 ) ;
28         btnSgte . setTextColor ( Color . parseColor ( "#ffffff" ) ) ;
29         btnSgte1 . setTextColor ( Color . parseColor ( "#ffffff" ) ) ;
30         btnSgte2 . setTextColor ( Color . parseColor ( "#ffffff" ) ) ;
31         btnSgte3 . setTextColor ( Color . parseColor ( "#ffffff" ) ) ;
32         btnSgte4 . setTextColor ( Color . parseColor ( "#ffffff" ) ) ;
33         btnSgte5 . setTextColor ( Color . parseColor ( "#ffffff" ) ) ;
34         btnSgte . setOnClickListener ( nueva Vista . OnClickListener () {

```

```

35         @Override
36         public void onClick ( Vista de vista ) {
37             Intención Intención = nueva Intención ( MainActivity.this,MainActivity2.class );
38             startActivity ( intent );
39         }
40     });
41     btnSgte1 . setOnClickListener ( nueva Vista . OnClickListener () {
42         @Override
43         public void onClick ( Vista de vista ) {
44             Intención Intención = nueva Intención ( MainActivity.this,MainActivity3.class );
45             startActivity ( intent );
46         }
47     });
48     btnSgte2 . setOnClickListener ( nueva Vista . OnClickListener () {
49         @Override
50         public void onClick ( Vista de vista ) {
51             Intención Intención = nueva Intención ( MainActivity.this,MainActivity4.class );
52             startActivity ( intent );
53         }
54     });
55     btnSgte3 . setOnClickListener ( nueva Vista . OnClickListener () {
56         @Override
57         public void onClick ( Ver vista ) {
58             Intención Intención = nueva Intención ( MainActivity.this,MainActivity5.class );
59             startActivity ( intent );
60         }
61     });
62     btnSgte4 . setOnClickListener ( nueva Vista . OnClickListener () {
63         @Override
64         public void onClick ( Vista de vista ) {
65             Intención Intención = nueva Intención ( MainActivity.this,MainActivity6.class );
66             startActivity ( intent );
67         }
68     });
69     btnSgte5 . setOnClickListener ( nueva Vista . OnClickListener () {
70         @Override
71         public void onClick ( Vista de vista ) {
72             Intención Intención = nueva Intención ( MainActivity.this,MainActivity7.class );
73             startActivity ( intent );
74         }
75     });
76     }
77 }

```

Las líneas de código del fichero *MainActivity2.java* se muestra a continuación:

```

1     paquete com . ejemplo . prefacultativo ;
2
3     importar androidx.appcompat.app.AppCompatActivity ;
4     importar androidx.recyclerview.widget.LinearLayoutManager ;
5     importar androidx.recyclerview.widget.RecyclerView ;
6     importar android.graphics.Color ;
7     importar android.os.Bundle ;
8     importar java.util.ArrayList ;
9     importar java.util.HashMap ;
10    importar java.util.List ;
11    importar androidx.annotation.Nullable ;
12    importar android.content.Intent ;
13    importar android.os.Bundle ;
14    importar com.android.volley.Request ;
15    importar com.android.volley.RequestQueue ;
16    importar com.android.volley.Response ;
17    importar com.android.volley.VolleyError ;
18    importar com.android.volley.toolbox.JsonArrayRequest ;
19    importar com.android.volley.toolbox.JsonObjectRequest ;
20    importar com.android.volley.toolbox.StringRequest ;

```

```

21 import com.android.volley.toolbox.Volley ;
22 import org.json.JSONArray ;
23 import org.json.JSONException ;
24 import org.json.JSONObject ;
25 importar android.util.Log ;
26 importar android.view.View ;
27 importar android.widget.Button ;
28 importar android.widget.Toast ;
29
30 public class MainActivity2 extiende AppCompatActivity {
31
32     RecyclerView mrecyclerView ;
33     LinearLayoutManager layoutManager ;
34     Lista < ModelClass > userList ;
35     Adaptador adaptador ;
36     Botón btnVolver ;
37
38     String URL_tables = "http://192.168.0.6/Ingenieria/obtenerConvocatorias.php" ;
39     RequestQueue privado requestQueue ;
40
41     @Override
42     protected void onCreate ( Bundle savedInstanceState ) {
43         super . onCreate ( savedInstanceState ) ;
44         setContentView ( R . diseño . activity_main2 ) ;
45         initData () ;
46         initRecyclerView () ;
47
48         btnVolver = ( Button ) findViewById ( R . ID . btnVolver ) ;
49         requestQueue = Volley . newRequestQueue ( esto ) ;
50         btnVolver . setTextColor ( Color . parseColor ( "#ffffff" ) ) ;
51
52         btnVolver . setOnClickListener ( new View . OnClickListener () {
53             @Override
54             public void onClick ( Ver vista ) {
55                 Intent volver = new Intent ( MainActivity2 . this , MainActivity . class ) ;
56                 startActivity ( volver ) ;
57             }
58         } ) ;
59     }
60     privado void initRecyclerView () {
61         mrecyclerView = findViewById ( R . ID . RecyclerView ) ;
62         layoutManager = new LinearLayoutManager ( esto ) ;
63         layoutManager . setOrientation ( RecyclerView . VERTICAL ) ;
64         mrecyclerView . setLayoutManager ( layoutManager ) ;
65         adaptador = nuevo Adaptador ( userList ) ;
66         mrecyclerView . setAdapter ( adaptador ) ;
67         adaptador . notificarDataSetChanged () ;
68     }
69     private void initData () {
70         userList = new ArrayList <> () ;
71         loadconvocatorias () ;
72     }
73     public void loadconvocatorias () {
74
75         String url = "http://192.168.0.6/Ingenieria/obtenerConvocatorias.php" ;
76
77         RequestQueue requestQueue = Volley . newRequestQueue ( getApplicationContext () ) ;
78         Tostadas . makeText ( MainActivity2 . this , "Conectando con la Base de Datos" , Toast
79 . LENGTH_SHORT ) . mostrar () ;
80         StringRequest stringRequest = new StringRequest ( Request . Method . POST , url ,
81             new Response . Listener < String > () {
82                 @Override
83                 public void onResponse ( respuesta de cadena ) {
84                     try {
85                         JSONObject jsonObject = new JSONObject ( respuesta ) ;
86                         JSONArray jsonArray = jsonObject . getJSONArray ( "Tablas" ) ;
87
88                         for ( int i = 0 ; i < jsonArray . length () ; i ++ ) {
89                             JSONObject jsonObject1 = jsonArray . getJSONObject ( i ) ;
90
91                             String auxTitulo = jsonObject1 . getString ( "titulo" ) ;

```

```

92         String auxDescripcion = jsonObject1 . getString ( "descripcion"
93 );
94         Cadena auxFecha = jsonObject1 . getString ( "fecha" );
95         String auxImagen = jsonObject1 . getString ( "imagen" );
96
97         auxImagen = auxImagen . reemplazar ( "", "% 20" );
98         auxImagen = auxImagen . reemplazar ( "localhost" ,
99 "192.168.0.6" );
100         userList . add ( new ModelClass ( auxTitulo , auxDescripcion ,
101 auxFecha , auxImagen , "
102         )
103         );
104         adaptador= nuevo Adaptador ( userList );
105         mRecyclerView . setAdapter ( adaptador );
106         adaptador . notificarDataSetChanged ();
107     } captura ( JSONException e ) {
108         e . printStackTrace ();
109     }
110 }
111 }, nueva respuesta . ErrorListener () {
112 @Override
113 public void onErrorResponse ( error VolleyError ) {
114     error . printStackTrace ();
115 }
116 });
117 requestQueue . agregar ( stringRequest );
118 }
119 }

```

Las líneas de código del fichero *MainActivity3.java* se muestra a continuación:

```

1  paquete com . ejemplo . prefacultativo ;
2
3  importar androidx.appcompat.app.AppCompatActivity ;
4  importar androidx.annotation.Nullable ;
5  importar androidx.recyclerview.widget.GridLayoutManager ;
6  importar androidx.recyclerview.widget.LinearLayoutManager ;
7  importar androidx.recyclerview.widget.RecyclerView ;
8  importar android.content.Intent ;
9  importar android.os.Bundle ;
10 import java.util.ArrayList ;
11 import java.util.List ;
12 import com.android.volley.Request ;
13 import com.android.volley.RequestQueue ;
14 import com.android.volley.Response ;
15 import com.android.volley.VolleyError ;
16 import com.android.volley.toolbox.StringRequest ;
17 import com.android.volley.toolbox.Volley ;
18 import org.json.JSONArray ;
19 import org.json.JSONException ;
20 import org.json.JSONObject ;
21 importar android.view.View ;
22 importar android.widget.Button ;
23
24 public class MainActivity3 extiende AppCompatActivity {
25
26     Botón btnVolver ;
27
28     ArrayList < TableSQL > tableList ;
29     RecyclerView recyclerView ;
30     String URL_tables = "http://192.168.1.16/Ingenieria/obtenerComunicados.php" ;
31
32     @Override
33     protected void onCreate ( Bundle savedInstanceState ) {

```

```

34     super . onCreate ( SavedInstanceState ) ;
35     setContentView ( R . diseño . activity_main3 ) ;
36
37     btnVolver = ( Button ) findViewById ( R . ID . btnVolver ) ;
38     recyclerView = ( RecyclerView ) findViewById ( R . ID . recyclerView ) ;
39     RecyclerView . setLayoutManager ( nuevo LinearLayoutManager ( esto ) ) ;
40
41     tableList = new ArrayList < TableSQL > ( ) ;
42
43     loadconvocatorias ( ) ;
44     btnVolver . setOnClickListener ( new View . OnClickListener ( ) {
45         @Override
46         public void onClick ( Ver vista ) {
47             Intent volver = new Intent ( MainActivity3 . this , MainActivity . class ) ;
48             startActivity ( volver ) ;
49         }
50     } ) ;
51 }
52 public void loadconvocatorias ( ) {
53     RequestQueue requestQueue = Volley . newRequestQueue ( getApplicationContext ( ) ) ;
54
55     StringRequest stringRequest = new StringRequest ( Request.Method.POST,URL_tables ,
56         new Response . Listener < String > ( ) {
57             @Override
58             public void onResponse ( String response ) {
59                 try {
60                     JSONObject jsonObject = new JSONObject ( respuesta ) ;
61                     JSONArray jsonArray = jsonObject . getJSONArray ( "Tablas" ) ;
62
63                     for ( int i = 0 ; i < jsonArray . length ( ) ; i ++ ) {
64                         JSONObject jsonObject1 = jsonArray . getJSONObject ( i ) ;
65
66                         String auxTitulo = jsonObject1 . getString ( "titulo" ) ;
67                         String auxDescripcion = jsonObject1 . getString ( "descripcion" ) ;
68                         Cadena auxFecha = jsonObject1 . getString ( "fecha" ) ;
69                         String auxImagen = jsonObject1 . getString ( "imagen" ) ;
70
71                         auxImagen = auxImagen . reemplazar ( "" , "% 20" ) ;
72                         auxImagen = auxImagen . reemplazar ( "localhost" , "192.168.1.16" ) ;
73
74                     tableList . agregar ( nuevo TableSQL ( auxTitulo,auxDescripcion,auxFecha auxImagen ) ) ;
75                 }
76                 } captura ( JSONException e ) {
77                     e . printStackTrace ( ) ;
78                 }
79             }
80         } , nueva respuesta . ErrorListener ( ) {
81             @Override
82             public void onErrorResponse ( error VolleyError ) {
83                 error . printStackTrace ( ) ;
84             }
85         } ) ;
86     requestQueue . agregar ( stringRequest ) ;
87 }
88
89 }

```

Las líneas de código del fichero *Mensajes.php* se muestra a continuación:

```

1 <? php
2 class Mensajes
3 {
4     privado $ id , $ id_usuario , $ referencia , $ mensaje , $ nombre_usuario ;
5
6     función pública __construct ( $ id_usuario , $ referencia , $ mensaje , $ nombre_usuario
7 , $ id = null )

```

```

8 {
9     $ esto -> id_usuario = $ id_usuario ;
10    $ esto -> referencia = $ referencia ;
11    $ esto -> mensaje = $ mensaje ;
12    $ esto -> nombre_usuario = $ nombre_usuario ;
13    if ( $ id ) {
14        $ esto -> id = $ id ;
15    }
16 }
17 función pública guardar ()
18 {
19     global $ mysqli ;
20     fecha ( 'Ymd H: i: s' );
21     $ sentencia = $ mysqli -> prepare ( "INSERT INTO mensajes
22         (id_usuario, referencia, mensaje, nombre_usuario, bitacora, estado)
23         VALUES
24         (?,?,?,?, now (), 'ACTIVO')" );
25     $ sentencia -> bind_param ( "ssss" , $ this -> id_usuario , $ this -> referencia , $
26 this -> mensaje , $ this -> nombre_usuario );
27     $ sentencia -> ejecutar ();
28 }
29
30 pública estática función Obtener ( $ id )
31 {
32     global $ mysqli ;
33     $ resultado = $ mysqli -> query ( "SELECT id, id_usuario, referencia, mensaje,
34 nombre_usuario, bitacora FROM mensajes donde estado = 'ACTIVO' e id_usuario =" . $ id );
35     // $ resultado-> bind_param ("i", $ id);
36     // $ resultado-> ejecutar ();
37     return $ resultado -> fetch_all (MYSQLI_ASSOC);
38 }
39 función estática pública obtenerUno ( $ id )
40 {
41     global $ mysqli ;
42     $ sentencia = $ mysqli -> prepare ( "SELECT id, id_usuario, referencia, mensaje FROM
43 mensajes WHERE id =?" );
44     $ sentencia -> bind_param ( "i" , $ id );
45     $ sentencia -> ejecutar ();
46     $ resultado = $ sentencia -> get_result ();
47     return $ resultado -> fetch_object ();
48 }
49 función pública actualizar ()
50 {
51     global $ mysqli ;
52     $ sentencia = $ mysqli -> prepare ( "actualizar el conjunto de mensajes id_usuario
53 =?, referencia=?, mensaje=?, nombre_usuario=? donde id =?" );
54     $ sentencia -> bind_param ( "ssssi" , $ this -> id_usuario , $ this -> referencia , $
55 this -> mensaje , $ this -> nombre_usuario , $ this -> id );
56     $ sentencia -> ejecutar ();
57 }
58
59 función estática pública eliminar ( $ id )
60 {
61     global $ mysqli ;
62     $ sentencia = $ mysqli -> prepare ( "BORRAR DE mensajes WHERE id =?" );
63     $ sentencia -> bind_param ( "i" , $ id );
64     $ sentencia -> ejecutar ();
65 }
66 }

```

Descripción:

FEB

01

2021

→ **INSCRIPCIÓN: CURSO PREFACULTATIVO (CPF) - FACULTAD DE INGENIERÍA ✓ GESTIÓN I/2021**

La Facultad de Ingeniería, convoca a los Bachilleres interesados en postular a las carreras de:

- Ingeniería Civil
- Ingeniería en Producción Industrial
- Ingeniería Química
- Ingeniería Electrónica
- Ing. en Seguridad Industrial y Salud Ocupacional
- Ingeniería Ambiental
- Ingeniería Petrolera
- Ingeniería Mecánica
- Ingeniería de Alimentos
- Ingeniería Eléctrica
- Ingeniería Electromecánica
- Ingeniería Petroquímica
- Ingeniería Industrial
- Ingeniería Mecatrónica
- Ing. Metalúrgica y de Materiales
- Ingeniería en Litio y Recursos Evaporíticos
- Ingeniería en Polímeros

→ Regístrase en:

<https://venta-valores.umsa.bo>

→ ✦ Fecha de Depósito:

Del 02 de enero al 30 de enero 2021

→ ✦ Inicio de Clases:

✓ Febrero 2021



UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERÍA
ACREDITADA INTERNACIONALMENTE POR EL CACEI
CONVOCATORIA CPF 05/2020
INSCRIPCIÓN CURSO PREFACULTATIVO DE INGENIERÍA
GESTIÓN I/2021



Conforme a Resolución N° 184/98 de fecha 09 de Septiembre de 1998 del Honorable Consejo Universitario, el ingreso de nuevos estudiantes a la Universidad Mayor de San Andrés procede previa aprobación del Curso Prefacultativo (C.P.F.); en tal virtud la **Facultad de Ingeniería**, convoca a los **Bachilleres** interesados en postular a las carreras de:

Ingeniería Civil	Ingeniería en Producción Industrial	Ingeniería Química
Ingeniería Electrónica	Ing. en Seguridad Industrial y Salud Ocupacional	Ingeniería Ambiental
Ingeniería Petrolera	Ingeniería Mecánica	Ingeniería de Alimentos
Ingeniería Eléctrica	Ingeniería Electromecánica	Ingeniería Petroquímica
Ingeniería Industrial	Ingeniería Mecatrónica	Ing. Metalúrgica y de Materiales
Ingeniería en Lítio y Recursos Evaporíticos	Ingeniería en Polímeros	Ingeniería Automotriz

A. REQUISITOS:

- Ingresar en la plataforma de recaudaciones <https://venta-valores.umsa.bo>. REGISTRÁNDOSE CON EL NOMBRE DEL POSTULANTE COMO DEPOSITANTE y generar CPT (Código de pago de trámite) escogiendo "CURSO PREFACULTATIVO (CPF) Fac. de Ingeniería I/2021- Estudiante Nacional" por Bs. 850,00 (Ochocientos cincuenta 00/100 Bolivianos), ó "CURSO PREFACULTATIVO (CPF) Fac. de Ingeniería I/2021 - Estudiante extranjero" por Bs. 1.650,00 (Un Mil Seiscientos Cincuenta 00/100 Bolivianos). Este monto debe depositarse a partir del 2 al 30 de enero de 2021, **INDICANDO EL N° DEL CPT PARA REALIZAR EL PAGO MEDIANTE UNINET, UNIMOVIL O CAJEROS DEL BANCO UNION**. Una vez generado el CPT realizar el depósito en el mismo día.
- FORMULARIO DE INSCRIPCIÓN.** - El Depósito Bancario habilitará al postulante para ingresar al sistema mediante la página Web de la Facultad de Ingeniería (prefing.umsa.edu.bo) donde introduciendo el número de Cédula de Identidad, tendrá la opción de elegir horario, asignándole un grupo (paralelo) según la disposición de plazas y horarios. En el llenado el FORMULARIO DE INSCRIPCIÓN.

NOTA: Elegido el horario no se aceptan cambios.

2.1. Documentos que el postulante debe adjuntar al formulario para validar su inscripción en el siguiente orden:

POSTULANTES BOLIVIANOS

- Fotografía digitalizada (Fondo plomo claro).
- Fotografía digitalizada del Depósito Bancario.
- Fotografía digitalizada simple del Título de Bachiller. **Para estudiantes que no contarán con el título, (egresados del 6to secundaria gestión 2020) presentar fotografías de la Libreta de 6to de secundaria y nota de compromiso de la presentación de este documento.**
- Fotografía digitalizada de la cédula de identidad (vigente) o RUN.
- Fotografía digitalizada del Certificado de Nacimiento computarizado del postulante.

POSTULANTES EXTRANJEROS: Además de la documentación descrita en el Incisos a) y b), los **postulantes extranjeros** deben presentar:

- Fotografía digitalizada de Título de Bachiller o su equivalente debidamente legalizado en el Consulado de Bolivia del país de origen y por la Cancillería Boliviana.
- Fotografía digitalizada de Pasaporte o Carnet de Extranjería vigente.
- Fotografía digitalizada del Certificado de Nacimiento legalizado por el Consulado Boliviano en el país de origen y por la Cancillería Boliviana.

NOTAS 1:

- Solo podrán inscribirse los postulantes que cuenten con el depósito correspondiente.
- En caso de cubrirse las plazas existentes, el depósito bancario se cerrará antes de la fecha indicada.
- Si se realizaran depósitos fuera de plazo estos no admitirán devolución alguna.
- Los estudiantes regulares con Matrícula Universitaria vigente podrán postular al Curso Prefacultativo y si aprobaran, deberán realizar el trámite que corresponda en las oficinas de Gestiones, Admisiones y Registros.

@UMSAinformacion

NOTA 2: De acuerdo a la evolución de la emergencia sanitaria, el Curso Prefacultativo Presencial se podrá convertir en un curso Semipresencial o totalmente Virtual. En caso de superarse la emergencia sanitaria, los exámenes se llevarán a cabo de manera presencial.

3. **ENTREGA DE DOCUMENTOS** Se publicará un cronograma para la entrega de la documentación antes mencionada en la página prefing.umsa.edu.bo, esta documentación deberá ser entregada en la Secretaría del Curso Prefacultativo, Primer Piso, Edif. del Curso Básico y Prefacultativo (Ex Príncipe de Paz) ubicado en la calle Bueno N° 444, (esq. Obispo Cárdenas); Tel. 2200551.

B. INICIO DE CLASES, ASIGNATURAS Y HORARIOS DE CLASES:

- Inicio Febrero de 2021, conclusión del Curso Junio 2021.
- Asignaturas: **MATEMÁTICA, FÍSICA, QUÍMICA.**
- Horario de clases: A elección del postulante según disposición de plazas:

07:30 a 10:30	11:00 a 14:00	14:30 a 17:30	18:00 a 21:00
---------------	---------------	---------------	---------------

Elegido el horario no se aceptan cambios.

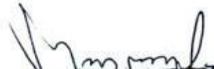
NOTA: El postulante deberá aprobar las tres asignaturas para su ingreso a la Facultad de Ingeniería.
LAS CARRERAS DE INGENIERÍA CIVIL, ELECTRONICA, INDUSTRIAL Y PETROLERA TIENEN INGRESOS LIMITADOS SEGÚN CUPOS QUE SERÁN ASIGNADOS EN FUNCIÓN DE LOS MEIORES PROMEDIOS.
 Para mayor información, dirigirse a la Dirección del Curso Prefacultativo, ubicado en la calle Bueno N° 444, esq. Obispo Cárdenas; Tel. 2200551 y la Página Web de la Facultad de Ingeniería (prefing.umsa.edu.bo).
 La Paz, Octubre de 2020



Ing. Marcelo O. Torres Huancu
 DIRECTOR CURSO PREFACULTATIVO
 FAC. DE INGENIERIA - U.M.S.A.




Ing. Freddy Gutierrez Bareu
 PRESIDENTE a.l.
 HONORABLE CONSEJO FACULTATIVO
 FACULTAD DE INGENIERIA

Ing. M.Sc. Félix Fernando Montenegro Céspedes
 VICERRECTOR a.l.
 UNIVERSIDAD MAYOR DE SAN ANDRÉS



@UMSAinformacion

→ INSCRIPCIÓN: CURSO PREFACULTATIVO VIRTUAL (CPF-V) - FACULTAD DE INGENIERÍA ✓ GESTIÓN I/2021

La Facultad de Ingeniería, convoca a los Bachilleres interesados en postular a las carreras de:

- Ingeniería Civil
- Ingeniería en Producción Industrial
- Ingeniería Química
- Ingeniería Electrónica
- Ing. en Seguridad Industrial y Salud Ocupacional
- Ingeniería Ambiental
- Ingeniería Petrolera
- Ingeniería Mecánica
- Ingeniería de Alimentos
- Ingeniería Eléctrica
- Ingeniería Electromecánica
- Ingeniería Petroquímica
- Ingeniería Industrial
- Ingeniería Mecatrónica
- Ing. Metalúrgica y de Materiales
- Ingeniería en Litio y Recursos Evaporíticos
- Ingeniería en Polímeros

→ Regístrase en:

<https://venta-valores.umsa.bo>

→ Fecha de Depósito:

Del 02 de enero al 30 de enero 2021

→ Inicio de Clases:

✓ Febrero 2021



UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERÍA
ACREDITADA INTERNACIONALMENTE POR EL CACEI
CONVOCATORIA CPF 06/2020
INSCRIPCIÓN CURSO PREFACULTATIVO VIRTUAL DE INGENIERÍA
GESTIÓN I/2021



Conforme a Resolución N° 184/98 de fecha 09 de Septiembre de 1998 del Honorable Consejo Universitario, el ingreso de nuevos estudiantes a la Universidad Mayor de San Andrés procede previa aprobación del Curso Prefacultativo (C.P.F.); en tal virtud la **Facultad de Ingeniería**, convoca a los **Bachilleres** interesados en postular a las carreras de:

Ingeniería Civil	Ingeniería en Producción Industrial	Ingeniería Química
Ingeniería Electrónica	Ing. en Seguridad Industrial y Salud Ocupacional	Ingeniería Ambiental
Ingeniería Petrolera	Ingeniería Mecánica	Ingeniería de Alimentos
Ingeniería Eléctrica	Ingeniería Electromecánica	Ingeniería Petroquímica
Ingeniería Industrial	Ingeniería Mecatrónica	Ing. Metalúrgica y de Materiales
Ingeniería en Lito y Recursos Evaporíticos	Ingeniería en Polímeros	Ingeniería Automotriz

A. REQUISITOS:

- Ingresar en la plataforma de recaudaciones <https://venta-valores.umsa.bo>, REGISTRÁNDOSE CON EL NOMBRE DEL POSTULANTE COMO DEPOSITANTE y generar CPT (Código de pago de trámite) escogiendo "CURSO PREFACULTATIVO VIRTUAL (CPF VIRTUAL) Fac. de Ingeniería I/2021- Estudiante Nacional" por Bs. 850,00 (Ochocientos cincuenta 00/100 Bolivianos), ó "CURSO PREFACULTATIVO VIRTUAL (CPF VIRTUAL) Fac. de Ingeniería I/2021 - Estudiante Extranjero" por Bs. 1.650,00 (Un Mil Seiscientos Cincuenta 00/100 Bolivianos). Este monto debe depositarse a partir del 2 al 30 de enero de 2021, **INDICANDO EL N° DEL CPT PARA REALIZAR EL PAGO MEDIANTE UNINET, UNIMOVIL O CAJEROS DEL BANCO UNION**. Una vez generado el CPT realizar el depósito en el mismo día.
- FORMULARIO DE INSCRIPCIÓN.** - El Depósito Bancario habilitará al postulante para ingresar al sistema mediante la página Web de la Facultad de Ingeniería (prefing.umsa.edu.bo) donde introduciendo el número de Cédula de Identidad realizará el llenado del FORMULARIO DE INSCRIPCIÓN.

2.1. Documentos que el postulante debe adjuntar al formulario para validar su inscripción en el siguiente orden:

POSTULANTES BOLIVIANOS

- Fotografía digitalizada (fondo plomo claro).
- Fotografía digitalizada de la Boleta de Depósito Bancario
- Fotografía digitalizada del Título de Bachiller.
- Fotografía digitalizada de la Cédula de Identidad (vigente) o RUN.
- Fotografía digitalizada del Certificado de Nacimiento computarizado del postulante.

POSTULANTES EXTRANJEROS: Además de la documentación descrita en el Incisos a) y b), los **postulantes extranjeros** deben presentar:

- Fotografía digitalizada de Título de Bachiller o su equivalente debidamente legalizado en el Consulado de Bolivia del país de origen y por la Cancillería Boliviana.
- Fotografía digitalizada de Pasaporte o Carnet de Extranjería vigente.
- Fotografía digitalizada del Certificado de Nacimiento legalizado por el Consulado Boliviano en el país de origen y por la Cancillería Boliviana.

NOTAS 1:

- Solo podrán inscribirse los postulantes que cuenten con el depósito correspondiente.
- En caso de cubrirse las plazas existentes, el depósito bancario se cerrará antes de la fecha indicada.
- Si se realizaran depósitos fuera de plazo estos no admitirán devolución alguna.
- Los estudiantes regulares con Matrícula Universitaria vigente podrán postular al Curso Prefacultativo Virtual y si aprobaran, deberán realizar el trámite que corresponda en las oficinas de Gestiones, Admisiones y Registros.

3. **ENTREGA DE DOCUMENTOS** Se publicará un cronograma para la entrega de la documentación antes mencionada en la página prefing.umsa.edu.bo, esta documentación deberá ser entregada en la Secretaría del Curso Prefacultativo, Primer Piso, Edif. del Curso Básico y Prefacultativo (Ex Príncipe de Paz) ubicado en la calle Bueno N° 444, (esq. Obispo Cárdenas); Tel. 2200551.

B. INICIO DE CLASES Y ASIGNATURAS:

- Inicio febrero de 2021, conclusión del Curso junio 2021.
- **Asignaturas: MATEMÁTICA, FÍSICA, QUÍMICA.**

NOTA: El postulante deberá aprobar las tres asignaturas para su ingreso a la Facultad de Ingeniería.

LAS CARRERAS DE INGENIERÍA CIVIL, ELECTRONICA, INDUSTRIAL Y PETROLERA TIENEN INGRESOS LIMITADOS SEGÚN CUPOS QUE SERÁN ASIGNADOS EN FUNCIÓN DE LOS MEJORES PROMEDIOS.

Para mayor información, dirigirse a la Dirección del Curso Prefacultativo, ubicado en la calle Bueno N° 444, esq. Obispo Cárdenas; Tel. 2200551 y la Página Web de la Facultad de Ingeniería (prefing.umsa.edu.bo).

La Paz, Octubre de 2020



Ing. Freddy Barrera Bareu
DIRECTOR CURSO PREFACULTATIVO
FAC. DE INGENIERÍA - UMSA




Ing. Freddy Barrera Bareu
PRESIDENTE a.l.
HONORABLE CONSEJO FACULTATIVO
FACULTAD DE INGENIERÍA




Ing. M.Sc. Félix Fernando Múzquiz Delgado
VICERRECTOR a.l.
UNIVERSIDAD MAYOR DE SAN ANDRÉS



@UMSAinformacion



COMUNICADO C.P.F. INGENIERÍA 04/2021

La Dirección del Curso Prefacultativo de la Facultad de Ingeniería comunica a los postulantes a **AUXILIARES DE DOCENCIA** para las respectivas asignaturas MATEMÁTICAS, FÍSICA y QUÍMICA, que el examen será de manera **virtual** a causa de la emergencia sanitaria **con el siguiente cronograma:**

DIA Y FECHA	ASIGNATURA	HORA
MARTES 12 DE ENERO DE 2021	MATEMATICA	08:00 AM.
MIERCOLES 13 DE ENERO DE 2021	FISICA	08:00 AM.
MIERCOLES 13 DE ENERO DE 2021	QUIMICA	14:00 PM.

Solo podrán ingresar al examen los postulantes que llenaron el formulario confirmando así su asistencia.

El enlace para los exámenes virtuales se habilitará 20 minutos antes del inicio de los exámenes:

<https://sites.google.com/view/cpf-ingenieria-umsa>

Las guías para el manejo del sistema se encuentran en la siguiente página, además contiene un link a un sistema de práctica para que los postulantes practiquen el ingreso y manejo del sistema de evaluaciones, es deber de los postulantes a auxiliares leer y practicar el manejo del sistema:

<https://sites.google.com/view/guia-examen-auxiliares>

Se realizará un **simulacro de examen** para las tres asignaturas el día **lunes 11/01/2021 a las 10:00 AM**. La asistencia al simulacro de examen es de manera obligatoria, el link para el simulacro es el mismo que el del examen:

<https://sites.google.com/view/cpf-ingenieria-umsa>

Nota: Para ingresar al examen es necesario confirmar su asistencia según el Comunicado C.P.F. N° 03/2021. Se extendió el plazo para confirmar su asistencia a los exámenes hasta el día **SABADO 09/01/2021 a las 23:59** impostergablemente. Los estudiantes que no llenen el formulario no podrán ingresar al sistema ni rendir su examen.



La Dirección

La Paz, 08 de Enero del 2021



COMUNICADO C.P.F. INGENIERÍA 11/2021

La Dirección del Curso Prefacultativo de la Facultad de Ingeniería comunica a los postulantes a la **PRUEBA DE SUFICIENCIA ACADÉMICA** de la gestión **1/2021** que completaron satisfactoriamente su inscripción. Toda la información necesaria para rendir los exámenes de manera **virtual** se publican en la página oficial <http://prefing.umsa.edu.bo>

Se reitera a los postulantes que aún no están en los grupos oficiales de Whatsapp y no llenaron el siguiente formulario deben realizarlo lo antes posible:

<https://forms.gle/PHr1HWM8Sb2Ats7R7>

Nota: El Curso Prefacultativo de Ingeniería no se responsabiliza por el desconocimiento de los postulantes sobre la información necesaria para rendir los exámenes. Toda información es publicada de manera oficial en la página <http://prefing.umsa.edu.bo> y los postulantes tienen la obligación de leer y realizar los procedimientos necesarios para rendir los exámenes.

Para cualquier consulta respecto a la **Prueba de Suficiencia Académica** de la **gestión 1/2021** puede enviar un mensaje de whatsapp al número **74093645**.



La Dirección
La Paz, 17 de Enero del 2021

COMUNICADO C.P.F. INGENIERÍA 13/2021



La Dirección del Curso Prefacultativo de la Facultad de Ingeniería comunica a los postulantes a la **PRUEBA DE SUFICIENCIA ACADÉMICA** de la gestión **1/2021** que completaron satisfactoriamente su inscripción. Un requisito para rendir los exámenes, es contar con una cámara web o cámara frontal en el celular o dispositivo utilizado. Durante los exámenes se pedirá en algún momento que encienda su cámara y podamos fotografiar al postulante rindiendo su examen personalmente. La no respuesta y/o rechazo a este contacto visual, inhabilitará su examen automáticamente. Los respaldos enviados de los ejercicios deben tener la firma del postulante junto a la respuesta, cualquier alteración o signo de manipulación será sujeto a disposición de la dirección del Curso Prefacultativo de Ingeniería. Los postulantes deben portar su carnet de identidad durante todos los exámenes.

De demostrarse que el postulante está siendo suplantado por alguna otra persona ajena, la Facultad de Ingeniería de la UMSA rechazará cualquier intento de inscripción posterior a esta casa Superior de Estudios y se tomarán acciones legales en contra la persona que incurra en falsedad material e ideológica (Código Penal art. 198° y art. 199°).



La Dirección
La Paz, 17 de Enero del 2021



COMUNICADO C.P.F. INGENIERÍA 16/2021



La Dirección del Curso Prefacultativo de la Facultad de Ingeniería comunica a los postulantes que desean inscribirse y tienen **dificultades al generar el código CPT** para realizar el depósito. Pueden aproximarse por la oficina de la Dirección del Curso Prefacultativo cumpliendo con todas las medidas de bioseguridad, para que el personal administrativo pueda ayudar con la generación del código CPT. Para esto debe llevar el carnet de identidad o una fotocopia del carnet de identidad del postulante y tener un correo electrónico activo.

Puede encontrar una guía para generar el código CPT en el siguiente enlace:

<http://prefing.umsa.edu.bo/?p=1710>



La Dirección

La Paz, 22 de Enero del 2021



COMUNICADO C.P.F. INGENIERÍA 21/2021



La Dirección del Curso Prefacultativo de la Facultad de Ingeniería, comunica a los postulantes que **completaron su inscripción** realizando el depósito, adjuntaron las fotos de sus documentos y llenaron el *formulario de inscripción* y aún no están en ningún grupo de whatsapp deben enviar los siguientes datos en un mensaje por whatsapp para registrarlos.

- Formulario de inscripción en imagen (actualizado al 08/02/2021, ingrese al sistema de inscripción - reimprimir formulario)
- Nombre completo
- Número de carnet
- Número de celular con Whatsapp

Debe enviar esta información a los siguientes números dependiendo el grupo al que pertenezca.

Grupos: 1 al 4 -> 79536075

Grupos: 5 al 9 -> 73580197

Grupos restantes: 74093645

Si aún no tiene grupo y realizó su depósito a tiempo debe realizar su registro en la página del Curso Prefacultativo de Ingeniería en la sección de registro a la brevedad posible:

<http://formularios.umsa.edu.bo/>



La Dirección

La Paz, 08 de Febrero del 2021



CRONOGRAMA DE EXAMENES DE COMPETENCIA - VIRTUALES



AUXILIARES DE DOCENCIA GESTION 2021 CURSO BASICO FACULTAD DE

AREA: MATEMATICA

ASIGNATURA	FECHA	HORA	DIA	PLATAFORMA
MAT 100	21/1/2021	08:00	JUEVES	
MAT 101	22/1/2021	08:00	VIERNES	
MAT 102	21/1/2021	08:00	JUEVES	TEAMS
MAT 103	21/1/2021	16:00	JUEVES	TEAMS
MAT 207	21/1/2021	16:00	JUEVES	TEAMS
MAT 218	21/1/2021	14:00	JUEVES	
MAT 313	21/1/2021	08:00	JUEVES	TEAMS
MAT 315	22/1/2021	08:00	VIERNES	

AREA: FISICA

ASIGNATURA	FECHA	HORA	DIA	PLATAFORMA
FIS 100	22/1/2021	08:00	VIERNES	
LAB FIS 100	22/1/2021	11:00	VIERNES	TEAMS
FIS 102	22/1/2021	10:00	VIERNES	
LAB FIS 102	25/1/2021	09:00	LUNES	
FIS 200	22/1/2021	10:00	VIERNES	
LAB FIS 200	20/1/2021	17:00	MIERCOLES	

AREA: QUIMICA

ASIGNATURA	FECHA	HORA	DIA	PLATAFORMA
QMC 100	25/1/2021	14:30	LUNES	
LAB QMC 100	25/1/2021	14:00	LUNES	

QMC 101	23/1/2021	08:00	SABADO	
LAB QMC 101	23/1/2021	08:00	SABADO	
QMC 104	25/1/2021	10:00	LUNES	
LAB QMC 104	25/1/2021	09:00	LUNES	
QMC 106	25/1/2021	10:00	LUNES	
LAB QMC 106	25/1/2021	12:00	LUNES	
QMC 200	23/1/2021	10:00	SABADO	
LAB QMC 200	23/1/2021	08:00	SABADO	
QMC 204	25/1/2021	14:00	LUNES	MOODLE
LAB QMC 204	25/1/2021	14:00	LUNES	MOODLE
QMC 206	23/1/2021	10:00	SABADO	
LAB QMC 206	25/1/2021	16:00	LUNES	

AREA: DIBUJO

ASIGNATURA	FECHA	HORA	DIA	PLATAFORMA
MEC 101	25/1/2021	08:00	LUNES	

TODOS LOS EXAMENES DE COMPETENCIA ESCRITOS SE REALIZARAN DE MANERA VIRTUAL EN LAS PLATAFORMAS DE TRABAJO DE LOS DOCENTES ENCARGADOS, ELLOS CREARAN LOS RESPECTIVOS GRUPOS DE TRABAJO CON LOS DATOS Y CORREOS CORPORATIVOS DE LOS POSTULANTES HABILITADOS EN EL SISTEMA FACULTATIVO.

CUALQUIER DUDA COMUNICARSE CON LA DIRECCION DEL CURSO BASICO TELF. 2200662.
 ATTE. M.SC.ING. MOISES ARTEAGA MIRANDA



COMUNICADO C.P.F. INGENIERÍA 43/2021



La Dirección del Curso Prefacultativo de la Facultad de Ingeniería, comunica a los postulantes que el **Segundo Parcial** se realizará de manera virtual por la emergencia sanitaria con el siguiente cronograma:

DÍA Y FECHA	ASIGNATURA	HORA
DOMINGO 25 DE ABRIL DE 2021	MATEMÁTICA	10:00 AM.
MIÉRCOLES 28 DE ABRIL DE 2021	FÍSICA	10:00 AM.
DOMINGO 02 DE MAYO DE 2021	QUÍMICA	10:00 AM.

Se realizará simulacros de examen para que los postulantes puedan practicar el uso del sistema y los procedimientos necesarios para rendir las evaluaciones virtuales, los postulantes pueden ingresar a cualquier simulacro (se recomienda participar en ambos turnos). Los postulantes que no participen de al menos **UN SIMULACRO** no podrán solicitar la segunda revisión de sus exámenes, el cronograma de los simulacros es el siguiente:

	Inicio	Finalización	Fecha
SIMULACRO 1	06:45 AM.	07:15 AM.	19/04/2021
SIMULACRO 2	21:15	21:45	

Además se habilitará un sistema de pruebas el día **SÁBADO 17/04/2021 a las 08:00 am** hasta el día **DOMINGO 18/04/2021 a las 20:00** para que los postulantes puedan practicar el manejo del sistema y el envío de archivos.

Recordar leer las guías y tutoriales para realizar los procedimientos de manera correcta:

https://drive.google.com/file/d/1_gnwgKd_FhmUL1o08Tbp_hINjatOEaMf/view?usp=sharing

Durante el Primer Parcial muchos postulantes **no siguieron las guías y tutoriales correspondientes** y cometieron muchas faltas, como la mencionadas a continuación:

- Envío de respuestas fuera de la hora prevista para cada pregunta.
- Envío de respaldos de sus ejercicios sin seguir los procedimientos correctos.
- No firmaron las respuestas de sus ejercicios.
- No usaron el formato de hoja del examen.
- Envío de fotos ilegibles.
- Enviaron respaldos con números de carnet errados.
- Enviaron archivos PDF con contraseña que no se pudo abrir.
- Envío de Fotos en lugar de archivos PDF.

Se recuerda que La Dirección del Curso Prefacultativo y el área de Sistemas no se responsabilizan **si sus respaldos no son revisados** por no tener los datos suficientes o no distinguir las fotografías de sus soluciones.

Se recomienda lo siguiente para rendir sus exámenes:

- Cerrar las demás aplicaciones que no se usen para el examen en su celular para evitar que su memoria se llene.
- Practicar sacar fotos a su procedimiento de manera correcta legible.



- Si no acaba la resolución de sus ejercicios y el tiempo de envío de la respuesta está por terminar debe enviar un resultado aproximado en el sistema para que los docentes revisores puedan considerar en valorar el procedimiento de sus ejercicios. En caso de **NO ENVIAR NINGUNA RESPUESTA** en el sistema no se tomará en cuenta el procedimiento de sus ejercicios.

- Si la respuesta enviada al sistema difiere de las respuestas en el respaldo la calificación del ejercicio no será completa.
- Se evidenció que muchos postulantes cometieron demasiados errores al enviar sus respaldos se recomienda revisar las siguientes guías:
 - o <https://drive.google.com/file/d/1l0Ws6wf-dlPte2P3hoVntqx074Ty8rCW/view>
 - o <https://drive.google.com/file/d/1yOl9qE74vj5GrugqmjC4wislejBCbTfo/view>
 - o <https://youtu.be/r2pm1jGcbfQ>
- Si no se puede encontrar sus respaldos por la falta de información tanto en el documento PDF como en el envío a DROPBOX no serán calificados.
- Toda resolución de los ejercicios debe estar completa en los respaldos enviados, tanto en los procedimientos usados como los pasos correspondientes.
- Durante los exámenes el personal de Sistemas atenderá solo los casos graves de los postulantes que tengan dificultades, considérese casos graves a toda acción fortuita no premeditada.
- El personal de Sistemas no se responsabiliza por aquellos postulantes que, al no revisar las guías, tutoriales y no haber participado de los simulacros tengan desconocimiento en el manejo del sistema y los procedimientos necesarios para dar las evaluaciones.
- Toda consulta o reclamo antes, durante y después del proceso de evaluación del Segundo Parcial, es de carácter personal; no se acepta injerencia alguna por parte de terceras personas.

Nota: Por disposición de la dirección del Curso Prefacultativo de Ingeniería no se aceptarán reclamos por fallas técnica de su dispositivo, fallas de internet u otros problemas. Los postulantes deben tomar los recaudos correspondientes para rendir sus exámenes sin contratiempos.



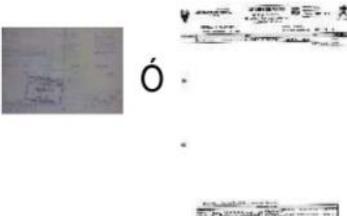
La Dirección
La Paz, 16 de Abril del 2021



COMUNICADO N.º 45



SE COMUNICA A TODOS LOS POSTULANTES QUE NO PUDIERON REALIZAR LA ENTREGA DE DOCUMENTOS HASTA LA FECHA, **SE HABILITA LOS DIAS JUEVES 22 Y VIERNES 23 DE ABRIL DE HORAS 08.30 A HORAS 16:30 DEL PRESENTE AÑO IMPOSTERGABLEMENTE EN CASO DE NO HACERLO NO SERAN HABILITADOS AL SEGUNDO PARCIAL.**

- 1) FORMULARIO DE POSTULACION 
- 2) FOTOCOPIA DE CERT. DE NACIMIENTO 
- 5) FOTOCOPIA DE C.I. 
- 3) FOTOCOPIA DE TITULO DE BACHILLER 
- 4) DEPOSITO BANCARIO Y DOS FOTOCOPIAS DEL DEPOSITO 

ADEMAS DEBE TRAER UNA TARJETA PVC PARA SU CREDENCIAL ESTE LO PUEDE ADQUIRIR EN LA PLANTA BAJA DE LOS PREDIOS DEL CURSO PREFACULTATIVO.


Ing. Angel W. Calderon Beltreros
DIRECTOR CURSO PREFACULTATIVO
FAC. DE INGENIERIA - UASB



La paz 21 de Abril, 2021



CONTRATO SERV. PREFACULTATIVO 05/2011

CONTRATO DE PRESTACIÓN DE SERVICIOS

Conste por el presente contrato de prestación de servicios, suscrito en la forma y lugar estipulados, sin que medie error, violencia o dolo entre partes, según el tenor de las siguientes cláusulas:

PRIMERA.- (ANTECEDENTES) La Universidad Mayor de San Andrés al amparo de lo dispuesto en la Constitución Política del Estado Art. 92, se constituye como entidad autónoma, en cuya condición y en el cumplimiento a disposiciones legales en vigencia, ha elaborado y aprobado el Plan Operado Anual para la presente gestión, el que incluye y contempla la ejecución presupuestaria, al efecto se ha determinado la contratación de servicios de un asistente técnico con destino al Curso Prefacultativo de la Facultad de Ingeniería, sujetándose a los reglamentos internos de la Entidad y disposiciones legales conexas.

SEGUNDA.- (DE LAS PARTES) Intervienen en el siguiente contrato las siguientes partes:

1.- La Universidad Mayor de San Andrés legalmente representada por el Ing. Mario Ramiro Terán Cortéz - Decano de la Facultad de Ingeniería, nombrado mediante Resolución del Honorable Consejo Universitario N° 057/11 de fecha 27 de enero del 2011, que para efectos del presente contrato será denominado UMSA en su calidad de **CONTRATANTE**.

2.- El Sr. Diego Coaquira Illanes, mayor de edad, hábil por derecho, con Cedula de Identidad N° 4267774 L.P., con domicilio en la calle Rosendo Rojas # 285, Zona Villa Victoria de esta ciudad, que para efectos legales del presente contrato administrativo será denominado **CONTRATADO**.

TERCERA.- (OBJETO)- El objeto del presente contrato es la prestación de servicios del **CONTRATADO** en calidad de asistente técnico con destino al Curso Prefacultativo de la Facultad de Ingeniería, debiendo cumplir las funciones y obligaciones descritas en los términos de referencia y el alcance del trabajo que forma parte componente e indivisible del presente contrato.

CUARTA.- (DOCUMENTOS DEL CONTRATO) Forma parte indivisible del presente contrato los siguientes documentos:

- Convocatoria Interna
- Cuadro de Calificaciones
- Certificación Presupuestaria
- Hoja de vida documentada
- Resolución del Honorable Consejo Facultativo N° 057/2011

QUINTA .- (VIGENCIA DEL CONTRATO) El presente contrato de prestación de servicios tendrá vigencia a partir del 03 de Enero de 2011 concluyendo fatalmente el 03 de abril de 2011, sin necesidad de comunicación expresa al **CONTRATADO** constituyendo la suscripción de este contrato la advertencia para la culminación del mismo. El Ing. Mario Ramiro Terán Cortéz, Decano de la Facultad de Ingeniería queda encargado y responsable del cumplimiento del plazo y condiciones estipuladas, bajo su exclusiva responsabilidad para fines de ley, caso de producirse la reconducción y/o ampliación de esta contratación

SEXTA.- (DE LA RENUMERACION) La **UMSA** en contraprestación a la función del **CONTRATADO** se compromete a cancelarle económicamente la suma mensual de Bs 1.750.00 (Un mil setecientos



UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERIA
ÁREA DESCONCENTRADA

cincuenta 00/100 Bolivianos), con cargo al presupuesto del Curso Prefacultativo, sujeto a descuentos de Ley.

SEPTIMA.- (OBLIGACIONES DEL CONTRATADO) El **CONTRATADO** se compromete a cumplir con las tareas asignadas de acuerdo a los términos de referencia y alcance del trabajo.

OCTAVA.- (DE LAS RESCISION) Será causal de rescisión del presente contrato por parte de la UMSA, que el **CONTRATADO**:

- Incumpla cualquiera de las condiciones estipuladas en el presente contrato.
- Que no cumpla los fines para las cuales es contratada.
- Que no encuadre su conducta dentro las previsiones contenidas en los Reglamentos Universitarios.

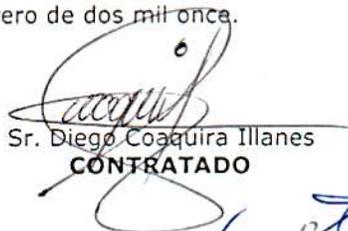
NOVENA (DE LAS OBLIGACIONES TRIBUTARIAS) La **UMSA** se constituye como agente de retención para cargas e imposiciones tributarias a las que está sujeto la contratada de acuerdo a lo previsto por la Ley.

DECIMA.- (IMPROCEDENCIA) Se establece de manera expresa que el presente contrato suscrito entre la **UMSA** y el **CONTRATADO**, no acepta la tacita reconducción y bajo ningún argumento será procedente la suscripción de un nuevo contrato, bajo responsabilidad de la autoridad que ordene, autorice o haga viable dicha prolongación de servicios.

DECIMA PRIMERA.- (DE LA NORMATIVA APLICADA) El presente contrato de prestación de servicios se rige al amparo de lo dispuesto por la Resolución del Honorable Consejo Universitario N° 258/06 del 7 de junio de 2006, la ley del Control Gubernamental, Ley 1178, Estatuto Orgánico, Reglamentos y disposiciones legales de la UMSA, no constituyendo relación laboral de dependencia entre el **CONTRATADO** y la UMSA, por lo que no se aplicará de ninguna manera la Ley General del Trabajo. Asimismo, la UMSA no asume ninguna responsabilidad en cuanto se refiere a beneficios sociales u otros, debido a la naturaleza eventual del contrato.

DECIMA SEGUNDA.- (DEL INCUMPLIMIENTO DEL CONTRATO) El **CONTRATADO** se obliga al cumplimiento del presente contrato, en caso de abandono de funciones o renuncia intempestiva, el **CONTRATANTE** asumirá las acciones legales correspondientes.

DECIMA TERCERA.- (DE LA CONFORMIDAD) Quienes celebran este contrato, la UMSA representada por el Ing. Mario Ramiro Terán Cortéz - Decano de la Facultad de Ingeniería por una parte y el Sr. Diego Coaquira Illanes por la otra, manifiestan haber tomado conocimiento del tenor y contenido de todas y cada una de sus cláusulas, expresando su conformidad a las mismas, por lo que se comprometen a su fiel y estricto cumplimiento. En constancia de su aceptación suscriben el presente, firmado en cinco ejemplares con idéntico fin y tenor, en la ciudad de La Paz, a los catorce días del mes de febrero de dos mil once.


Sr. Diego Coaquira Illanes
CONTRATADO


ASESOR JURÍDICO UMSA


Ing. Mario Ramiro Terán Cortéz
DECANO
FACULTAD DE INGENIERIA



UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE INGENIERÍA
CURSO PREFACULTATIVO



MEMORANDO

C.P.F. MEMO N° 08 - 2012

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE INGENIERIA
CURSO PREFACULTATIVO 2012

Señor:
Diego Coaquira Illanes
Presente.-

La Paz, 06 de Febrero de 2012

Ref.: SU DESIGNACION

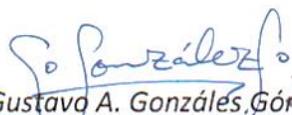
Señor:

Con la respectiva aprobación del Honorable Consejo Facultativo de la Facultad de Ingeniería según Resolución Facultativa N° 082/2012 de fecha 06 de Febrero de 2012, y remitiéndonos a resultados del Concurso de Meritos pertinentes, tenemos a bien comunicarle, que ha sido designado como:

ASISTENTE TÉCNICO DEL "CURSO PREFACULTATIVO DE INGENIERIA 2012"

A partir del 03 de Febrero de 2012 hasta el 03 de Abril de 2012, cumpliendo el horario de tiempo completo de lunes a viernes.

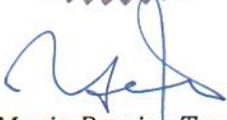
A tiempo de desearle éxito en las funciones asignadas, salud a usted atentamente.


 Ing. Gustavo A. González Gómez
DIRECTOR
CURSO PREFACULTATIVO




 Ing. Fernando Sanabria Camacho
VICEDECANO
FACULTAD DE INGENIERIA




 Ing. Mario Ramiro Terán Cortez
DECANO
FACULTAD DE INGENIERIA



C.c. Archivo
 C.c. File Personal
 GGG/baq



UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERÍA
CURSO PREUNIVERSITARIO



MEMORANDO

C.P.U. MEMO N° 175 - 2013

UNIVERSIDAD MAYOR DE SAN ANDRÉS
 FACULTAD DE INGENIERÍA
 CURSO PREUNIVERSITARIO 2013

Señor:
 Univ. COAQUIRA ILLANES DIEGO
Presente.-

La Paz, 25 de Febrero de 2013

Ref.: SU DESIGNACIÓN

Señor Universitario:

De acuerdo a los resultados obtenidos en el proceso de elección de Auxiliares de Docencia correspondiente a la Primera Convocatoria N° 04/2013 para la Gestión 2013 de esta unidad académica y con la respectiva aprobación del Honorable Consejo Facultativo de la Facultad de Ingeniería mediante Resolución Facultativa 161/2013, en fecha 25 de febrero, tenemos el agrado de comunicarle su designación como:

AUXILIAR DE LA SIGNATURA DE FÍSICA, GRUPO 21
DEL CURSO PREUNIVERSITARIO SEMESTRE ACADÉMICO I/2013 - II/2013

Con una carga horaria de 20 Hrs/mes, a partir del 13 de febrero hasta el 20 de diciembre de 2013.

A tiempo de recomendarle permanente coordinación con el Sr. Catedrático de la citada asignatura hacemos votos para el éxito en el desempeño de sus funciones.




 Ing. Gustavo A. González Gómez
DIRECTOR
CURSO PREUNIVERSITARIO




 Ing. Miguel Ángel Calla Carrasco
DECANO
FACULTAD DE INGENIERÍA


 Ing. Mario José Delgadillo Zurita
VICEDECANO
FACULTAD DE INGENIERÍA



C.c. Archivo
 C.c. File Personal
 GGG/zgq



UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERÍA
CURSO PREFACULTATIVO



MEMORÁNDUM

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERÍA
CURSO PREFACULTATIVO II/2009

C.P.F. MEMO N° 095 - II/2009

Señor:
Univ. Coaquira Illanes Diego
Presente.-

La Paz, 20 de Agosto de 2009

Ref.: SU DESIGNACIÓN

Señor Universitario:

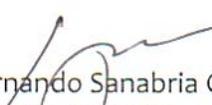
Con la respectiva aprobación del Honorable Consejo Facultativo de la Facultad de Ingeniería bajo **Resolución Facultativa N° 687/2009** de fecha 11 de agosto, y remitiéndonos a resultados del Concurso de Méritos pertinentes, tenemos a bien comunicarle, que ha sido designado como:

**AUXILIAR DE LA ASIGNATURA DE MATEMÁTICA DEL CURSO PREFACULTATIVO
SEMESTRE ACADÉMICO II/2009**

Con una carga horaria de 20 hrs/mes, a partir del 10 de agosto hasta la conclusión del presente Semestre Académico, con cargo al presupuesto de la Dirección del Curso Prefacultativo.

Sin otro particular, saludo a Usted Muy atentamente,


Lic. Jaime Mariscal Ponce
DIRECTOR
CURSO PREFACULTATIVO
La Paz - Bolivia


Ing. Fernando Sanabria Camacho
VICEDECANO
FACULTAD DE INGENIERÍA


Ing. Mario Ramiro Terán Cortéz
DECANO
FACULTAD DE INGENIERÍA