



FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS

**DETECCIÓN DE ALUMNOS MEDIANTE MECANISMOS
PROACTIVOS PARA EL CONTROL DE ASISTENCIA PARA LA
FACULTAD DE INGENIERÍA Y ARQUITECTURA DE LA
UNIVERSIDAD SAN MARTÍN DE PORRES**

PRESENTADA POR

**CARLOS YOHEL ALVA SANDOVAL
CRISPIN CRISTIAN ESTELA VILLENA**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE COMPUTACIÓN Y SISTEMAS**

LIMA – PERÚ

2015



**Reconocimiento - No comercial - Sin obra derivada
CC BY-NC-ND**

El autor sólo permite que se pueda descargar esta obra y compartirla con otras personas, siempre que se reconozca su autoría, pero no se puede cambiar de ninguna manera ni se puede utilizar comercialmente.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>



ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS

**DETECCIÓN DE ALUMNOS MEDIANTE MECANISMOS PROACTIVOS
PARA EL CONTROL DE ASISTENCIA PARA LA FACULTAD DE
INGENIERÍA Y ARQUITECTURA DE LA UNIVERSIDAD SAN MARTÍN**

DE PORRES

TESIS

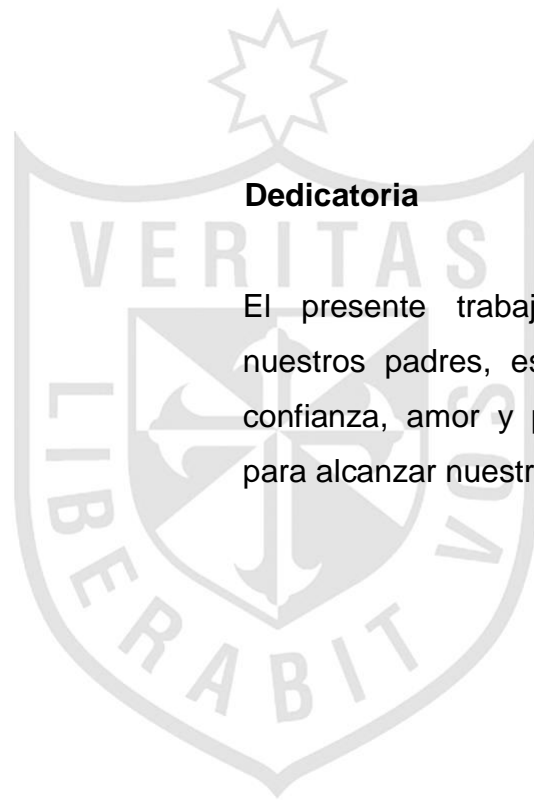
**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE COMPUTACIÓN Y SISTEMAS**

PRESENTADO POR

**ALVA SANDOVAL, CARLOS YOHEL
ESTELA VILLENA, CRISPIN CRISTIAN**

LIMA – PERÚ

2015



Dedicatoria

El presente trabajo está dedicado a nuestros padres, esposa e hijos por su confianza, amor y por ser la inspiración para alcanzar nuestras metas.



Agradecimientos

A Dios, por permitirnos participar en este proyecto y escalar así un peldaño más en nuestra vida profesional.

A todas aquellas personas que con su consejo y opinión colaboraron con el desarrollo y culminación de nuestro trabajo.

ÍNDICE

	Página
ÍNDICE	iv
RESUMEN	ix
ABSTRACT	x
INTRODUCCIÓN	xi
CAPÍTULO I MARCO TEÓRICO	14
1.1 Antecedentes	14
1.2 Bases teóricas	17
1.3 Definición de términos básicos	37
CAPÍTULO II METODOLOGIA	39
2.1 Planificación	42
2.2 Diseño	50
2.5 Mantenimiento	62
CAPÍTULO III DESARROLLO DE LA APLICACIÓN	64
3.1 Planificación	64
3.2 Diseño	81
3.3 Desarrollo	103
3.4 Pruebas	105
CAPÍTULO IV PRUEBAS Y RESULTADOS	111
CAPÍTULO V DISCUSIÓN Y APLICACIÓN	117
CONCLUSIONES	119
RECOMENDACIONES	120
FUENTES DE INFORMACIÓN	121
ANEXOS	124

LISTA DE TABLAS

	Página
Tabla N°1 : Ventajas y desventajas del sistema operativo Android	20
Tabla N°2 : Ventajas y desventajas del sistema operativo iOS	21
Tabla N°3 : Ventajas y desventajas del sistema operativo Windows Phone.	22
Tabla N°4 : Ventajas y desventajas del sistema operativo Symbian.	23
Tabla N°5 : Ventajas y desventajas del sistema operativo Blackberry	24
Tabla N°6 : Aplicaciones Híbridas ventaja y desventaja.	26
Tabla N°7 : Aplicaciones web ventaja y desventaja.	27
Tabla N°8 : Aplicaciones nativas ventaja y desventaja.	28
Tabla N°9 : Arquitectura de aplicaciones para dispositivos móviles	30
Tabla N°10 : Herramientas de redacción y edición	39
Tabla N°11 : Herramientas de repositorio y comunicación	40
Tabla N°12 : Herramientas de desarrollo	40
Tabla N°13 : Herramientas colaborativas usadas en la fase de planificación	42
Tabla N°15 : Herramientas colaborativas usadas en la fase de desarrollo	53
Tabla N°16 : Herramientas colaborativas usadas en la fase de pruebas	57
Tabla N°17 : Herramientas colaborativas	62
Tabla N°18 : Herramientas colaborativas usadas en la fase de planificación	64
Tabla N°19 : Levantamiento de Información	65
Tabla N°20 : Hoja de entrevista no estructurada	66
Tabla N°21 : Hoja de entrevista estructurada	67
Tabla N°22 : Hoja de entrevista estructurada	68
Tabla N°23 : Formato de lluvia de ideas	69
Tabla N°24 : Formato de identificación de mejora de ideas	70
Tabla N°25 : Captura de requerimientos	74
Tabla N°26 : Captura de requerimientos	75
Tabla N°27 : Identificación de los requerimientos funcionales	75
Tabla N°28 : Identificación de los requerimientos no funcionales	76
Tabla N°29 : Cronograma de actividades.	79
Tabla N°30 : Cronograma de actividades	80

Tabla N°31 : Herramientas colaborativas usadas en la fase de diseño	81
Tabla N°32 : Historia de usuario N° 001	82
Tabla N°33 : Historia de usuario N° 002	82
Tabla N°34 : Herramientas colaborativas usadas en la fase de desarrollo	103
Tabla N°35 : Herramientas colaborativas usadas en la fase de pruebas	105
Tabla N°36 : Caso de prueba – registrar dispositivo	106
Tabla N°37 : Caso de prueba – registrar hito de asistencia	107
Tabla N°38 : Herramientas colaborativas	108
Tabla N°39 : Objetivos y resultados obtenidos	115
Tabla N°40 : Objetivos y resultados obtenidos	116
Tabla N°41 : Comparación con otros sistemas de asistencia	118



Tabla de Ilustraciones

	Página
Ilustración 1 : Sistemas operativos móviles más usados.	25
Ilustración 2 : Imagen transversal de un Estimote Beacon	33
Ilustración 3 : Rango de detección de un Estimote Beacon	34
Ilustración 4 : Identificación única de Estimote Beacon	34
Ilustración 5 : Metodología híbrido para el desarrollo ágil	35
Ilustración 6 : Ciclo de desarrollo de la metodología Mobile D	36
Ilustración 7 : Metodología propuesta de desarrollo ágil	41
Ilustración 8 : Etapa de planificación	43
Ilustración 9 : Roles del sistema	47
Ilustración 10 : Simbología del cronograma de actividades	48
Ilustración 11 : Diseño de las historias de usuario	49
Ilustración 12 : Etapa de diseño de la arquitectura de software	51
Ilustración 13 : Etapa de desarrollo esquema general	54
Ilustración 14 : Flujo de trabajo en el proceso de desarrollo	56
Ilustración 15 : Proceso de prueba mediante caja blanca.	58
Ilustración 16 : Etapa de pruebas unitarias	59
Ilustración 17 : Etapa de prueba de concurrencia	60
Ilustración 18 : Pruebas de test de aceptación	61
Ilustración 19 : Prototipos 01 de la aplicación	73
Ilustración 20 : Icono de aplicación	83
Ilustración 21 : Autenticación de estudiante	84
Ilustración 22 : Menú lateral Fuente: Propia	85
Ilustración 23 : Horario de clases Fuente: Propia	86
Ilustración 24 : Asistencias Fuente: Propia	87
Ilustración 25 : Instrucciones Fuente: Propia	88
Ilustración 26 : Activar Bluetooth Fuente: Propia	89
Ilustración 27 : Inicio proceso asistencia Fuente: Propia	90
Ilustración 28 : Registro de hitos Fuente: Propia	91
Ilustración 29 : Mensaje registro satisfactorio	92

Ilustración 30 : Arquitectura del sistema	94
Ilustración 31 : Ubicación de Dispositivos Beacons	95
Ilustración 32 : Arquitectura física	96
Ilustración 33 : Diagrama de clases	97
Ilustración 34 : Diagrama de actividad - registrar asistencia	99
Ilustración 35 : Diagrama de secuencia - registrar asistencia	100
Ilustración 36 : Diseño físico de la base de datos	102
Ilustración 37 : Pruebas de concurrencia	107
Ilustración 38 : Tiempo promedio por cantidad de alumnos en 1.5 horas	112
Ilustración 39 : Porcentaje de reducción de tiempo en la toma de asistencia	113
Ilustración 40 : Cantidad de hitos registrados por alumnos.	114
Ilustración 41 : Calificación del sistema actual del sistema de asistencia.	126
Ilustración 42 : Deficiencias sobre el sistema actual de toma de asistencia	127
Ilustración 43 : Pregunta sobre aceptación del sistema actual	128
Ilustración 44 : Porcentaje de alumnos que posee Smartphone	129
Ilustración 45 : consulta sobre el tipo sistema operativo	130

RESUMEN

La presente investigación está basada en la aplicación de un proyecto que consiste en el desarrollo e implementación de mecanismos de detección que permita reducir los tiempos en la toma de asistencia y controlar la asistencia efectiva del alumno a clase en la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres (FIA). Actualmente; en la facultad, se usa un sistema de control de asistencia que no permite comprobar la permanencia de los estudiantes y su uso diario resta tiempo a las horas académicas.

La solución que se plantea es mediante la aplicación móvil, la cual se desarrolla a través de una metodología ágil. La cual consiste en un marco metodológico de trabajo que permite mejorar la eficiencia y calidad del producto final, la capacidad de respuesta ante el cambio y brindar mayor satisfacción al usuario a través de la entrega temprana de la solución o producto final.

Los resultados obtenidos fueron la implementación del mecanismo de detección a través del uso de tecnología Bluetooth, la reducción de tiempos en la toma de asistencia en un 84.69% y el control de permanencia efectiva en clases de los alumnos que poseen Smartphone.

En consecuencia el mecanismo de detección contribuirá con la reducción de tiempos y el control de permanencia efectiva de los alumnos dentro de las aulas de la Facultad de Ingeniería y Arquitectura de la USMP.

Palabras claves: Mecanismos proactivos, control de asistencia, alumnos, permanencia efectiva, reducción de tiempos.

ABSTRACT

This research is based on the implementation of a project that involves the development and implementation of detection mechanisms that would reduce time in taking assistance and controlling the effective assistance of the student in class at Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres (FIA). Nowadays there is used a control system that does not allow to check the permanence of the student in class and its daily use incurs in taking time away from academic hours.

The solution arises through the mobile application, which was developed through an agile methodology. Which consists of a methodological framework that improves the efficiency and quality of the final product, the ability to respond to change and deliver higher user satisfaction through early delivery of the solution or the final product.

The results were the implementation of the mechanism of detection through the use of Bluetooth technology, the taking assistance time reduction in 84.69% and the control of effective permanence of students in classes who possess a Smartphone.

Consequently the detection mechanism will contribute to reducing time and to control effective assistance for students in the classrooms at Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres.

Key words: Proactive mechanisms, control assistance, students, effective assistance, time reduction.

INTRODUCCIÓN

La tesis se titula “Detección de alumnos mediante mecanismos proactivos para el control de asistencia para la Facultad de Ingeniería y Arquitectura de la universidad San Martín de Porres”.

El campo de estudio es la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres. Se eligió este campo por la experiencia vivencial que se obtuvo durante el periodo de formación universitaria.

Así también debido a la competitividad del sector en el que se encuentra, la institución requiere mejorar su proceso de control de asistencia, motivo por lo cual debe contar con mecanismos de detección eficiente para reducir los tiempos en el control de asistencia de los alumnos y optimizar el tiempo de estudio.

La tesis está estructurada en 5 capítulos. El primero, denominado Marco Teórico, se recopila la información necesaria que sustentará las hipótesis. Segundo, Metodología, se expone los pasos a seguir para lograr el resultado esperado. Tercero, Desarrollo de la aplicación, se detalla cómo se construye la solución planteada. Cuarto, pruebas y resultados, se detalla las pruebas y resultados obtenidos hechos sobre el prototipo desarrollado. Quinto, discusión y aplicación, se detalla comparaciones con soluciones similares resaltando diferenciales sobre la solución propuesta.

1. Planteamiento del problema

Desfasado sistema de control de asistencia el cual no permite controlar la permanencia total o parcial del alumno a clase y su ejecución genera la pérdida tiempo de las horas de clase, la toma de asistencia que se da en las aulas de la facultad de ingeniería y arquitectura de la universidad de San Martín de Porres.

2. Objetivos

2.1 Objetivo general

Mejorar el proceso de control de asistencia a través de la implementación de una aplicación móvil y mecanismos proactivos mediante el uso de la tecnología Bluetooth.

2.2 Objetivos específicos

- a. Facilitar un mecanismo efectivo para la ejecución del proceso de control de asistencia de los alumnos.
- b. Reducir el tiempo que se emplea en realizar el proceso de control de asistencia de los alumnos.
- c. Controlar y medir la permanencia efectiva de los alumnos dentro de las aulas de clase.
- d. Contar con un registro de asistencia del alumno, que permita obtener información detallada y realizar una mejor gestión del mismo

3. Justificación

La calidad de la educación, constituye hoy en día una pieza clave e importante en toda institución, así como el tiempo que se le asigna a los cursos de formación y cómo este es aprovechado en el desarrollo de la clase, motivo por lo cual los responsables de estos servicios deben tomar muy en cuenta, la realización de la mejora continua de sus procesos relacionados con la formación de sus alumnos.

El presente proyecto busca implementar un mecanismo de detección de alumnos con el uso de herramientas tecnológicas a través de una aplicación móvil, la FIA no cuenta con un sistema automatizado del control de asistencia para los alumnos, y ello trae como resultado que los docentes dediquen parte del tiempo de las horas académicas asignadas a realizar el control de asistencia de los alumnos.

Lo que busca este aplicativo es ser una herramienta ágil en el proceso de control de asistencia de los alumnos y aprovechar las nuevas tecnologías de Bluetooth LE, permitiendo así grandes ventajas en la reducción de tiempos.

4. Limitaciones

La limitación que se puede observar es la tecnología ya que no todos los alumnos poseen Smartphone y de estos no todos tienen la versión de Bluetooth 4.0 o superior. Así mismo, la velocidad de ancho de banda provista por la universidad es lenta lo cual hace lento el proceso de toma de asistencia en la solución planteada.

CAPÍTULO I

MARCO TEÓRICO

1.1 Antecedentes

En este capítulo se realizará una breve descripción de la organización, así como la situación actual de su sistema de control de asistencia. Por otro lado, se detallaran las bases teóricas que sustenten la presente tesis y la definición de los términos básicos usados.

1.1.1 Antecedentes de la institución

La Universidad de San Martín de Porres, remonta sus orígenes al Instituto Pro-Deo, una casa dedicada al cultivo de la filosofía y la teología que fundó el Dr. Vicente Sánchez Valer, quien la constituye posteriormente, en Universidad bajo la advocación del santo, el 17 de mayo de 1962. (Universidad de San Martín de Porres, 2015).

A inicio de apertura la FIA llevaba el nombre de Facultad de Ingeniería de Computación y Sistemas (FICS), la cual empezó a ejecutar el control de asistencia de sus alumnos mediante el uso de un documento impreso denominado “Listado de asistencia de alumnos”, el que se pasaba el documento uno por uno a cada alumno presente en la clase para que registren sus datos.

En la actualidad la FIA controla la asistencia de sus alumnos mediante un sistema web, en el cual los docentes llaman uno por uno a todos los alumnos matriculados por cada sección. Si bien es cierto este proceso funciona correctamente, pero no ha evolucionado con el transcurrir del tiempo. Por otro lado, la FIA al ser una facultad de ingeniería, debería ser un referente en el uso de las nuevas tecnologías.

1.1.2 Casos de éxito

Caso 1 - Blueps: Sistema de localización en interiores utilizando Bluetooth

Los sistemas de localización son uno de los campos más prometedores en el campo de la computación móvil. Los sistemas de posicionamiento más comunes basados en satélites GPS o en las redes de telefonía móvil son adecuados para entornos exteriores.

Blueps es un nuevo sistema de localización para entornos de interior. Este sistema usa la tecnología de radiofrecuencia proporcionada por una red Bluetooth. El sistema se basa en la disminución del nivel de la señal transmitida cuando atraviesa una cierta distancia.

Un dispositivo detecta su posición, recibe los niveles de señal de los puntos de acceso cercanos y envía esta información a un servidor central, donde se procesa dicha información para determinar la posición. El uso de un servidor central de proceso permite la implementación de complejos algoritmos y la conexión a este es posible a través de la misma red Bluetooth utilizada para la localización. (Rodriguez, Pece, & Escudero, 2005)

Caso 2 - EDUMÓVIL: Aplicación de la tecnología móvil en la educación

Este aplicativo Edumóvil tiene como objetivo mejorar el proceso enseñanza-aprendizaje de los niños de nivel primaria a través de la incorporación de tecnología móvil en el aula. Edumóvil pretende explotar los beneficios que trae consigo la tecnología móvil en la educación, como el bajo costo, la movilidad, las responsabilidades individuales y la oportunidad de trabajo colaborativo. Edumóvil se enfoca al trabajo en los ejes en los cuales el niño presenta dificultad de aprendizaje, brindando para esto juegos donde los niños pueden trabajar en forma individual, y otros en forma grupal, utilizando para esto el protocolo Bluetooth, y tomando como una buena alternativa la restricción de la tecnología donde el número de integrantes debe ser menor de 8, permitiendo así que todos puedan participar, generando talleres de enseñanza.

Para la comunicación de los diferentes dispositivos que permitan llevar a cabo la enseñanza-aprendizaje colaborativo dentro del aula se contempla el uso del protocolo Bluetooth, por el bajo consumo de energía y su alcance.

La aplicación colaborativa trabaja sobre dispositivos móviles utilizando el protocolo Bluetooth, propiciando el trabajo en equipo y el enriquecimiento de la interacción de los niños entre sus compañeros de clases, y fortaleciendo las relaciones sociales. La aplicación contempla trabajar por medio de un servidor que reparta líneas de tiempo a los diferentes equipos formados en clase, cada uno de ellos ordena los fragmentos de dicha línea trabajando en equipo, y posteriormente lo envía al servidor para ser evaluado. (Gerónimo-Castillo & Rocha-Trejo, 2007)

1.2 Bases teóricas

1.2.1 Cálculo del tamaño de la muestra para una investigación de mercado

Para determinar el tamaño de una muestra en una investigación de mercado se deberán tomar en cuenta varios aspectos relacionados con el parámetro y estimador, el sesgo, el error muestral, el nivel de confianza y la varianza poblacional.

El parámetro se refiere a la característica de la población que es objeto de estudio y el estimador es la función de la muestra que se usa para medirlo.

El error muestral siempre se comete ya que existe una pérdida de la representatividad al momento de escoger los elementos de la muestra. Sin embargo, la naturaleza de la investigación nos indicará hasta qué grado se puede aceptar.

El nivel de confianza, por su parte, es la probabilidad de que la estimación efectuada se ajuste a la realidad; es decir, que caiga dentro de un intervalo determinado basado en el estimador y que capte el valor verdadero del parámetro a medir.

Cuando deseamos estimar una proporción, debemos conocer varios aspectos

- a. El nivel de confianza o seguridad ($1 - \alpha$). El nivel de confianza prefijado da lugar a un coeficiente ($Z\alpha$).
- b. La precisión que deseamos para el estudio.
- c. Una idea del valor aproximado del parámetro que queremos medir (en este caso una proporción). Esta idea se puede obtener revisando la literatura, por estudio pilotos previos. En caso de no tener dicha información utilizaremos el valor $p = 0.5$ (50%). El problema que puede enfrentarse en un estudio de investigación es la cantidad de información con la que se

cuenta; específicamente se pueden tener dos casos: desconocer la población del fenómeno estudiado, o bien, conocerla.

La fórmula para calcular el tamaño de muestra cuando se conoce el tamaño de la población es la siguiente:

$$n = \frac{N * Z^2 * p * q}{d^2 * (N - 1) + Z^2 * p * q}$$

En donde,

N = tamaño de la población

Z = nivel de confianza

P = probabilidad de éxito, o proporción esperada

Q = probabilidad de fracaso

D = precisión (Error máximo admisible en términos de proporción) (Torres, Paz, & Salazar, 2006)

1.2.2 El Smartphone

Un Smartphone es un teléfono móvil con un sistema operativo (OS) avanzado que permite instalar aplicaciones complejas y una amplia personalización de la interfaz gráfica de usuario.

A diferencia de un teléfono móvil estándar, los teléfonos inteligentes tienen un espacio de almacenamiento interno mucho más grande, y la mayoría tiene la posibilidad de introducción de una tarjeta micro-SD, la cual suele tener entre 2 GB y 32 GB, para ampliar la memoria de almacenamiento de los dispositivos.

Un teléfono inteligente está equipado, en la mayoría de los casos, con una pantalla táctil, superior a 3 pulgadas. Además, un smartphone, en general, no tiene un teclado físico. Si es proporcionado por el fabricante, se trata de un formato deslizante y es de tipo QWERTY (hace referencia a las primeras seis letras que aparecen en la esquina superior izquierda de un teclado físico). Hoy

en día un smartphone es considerado un teléfono que ejecuta uno de los siguientes sistemas operativos: Android OS (en aumento, con una gran cantidad de fabricantes que lo utilizan), BlackBerry OS (tan común para personas de negocios como para simples usuarios), Symbian, iOS (software de Apple para iPhone) y Windows Phone. (Timoftii & Lucian, 2013)

1.2.3 Sistemas operativos para dispositivos móviles.

Un Sistema Operativo (SO) es un software que actúa como interfaz entre los dispositivos de hardware y los programas usados por el usuario para manejar un computador. Es responsable de gestionar y coordinar las actividades, llevar a cabo el intercambio de recursos y actuar como base para las aplicaciones que se ejecutan en la máquina.

Un SO móvil es un sistema operativo que controla un dispositivo móvil al igual que las computadoras utilizan Windows o Linux, entre otros. Sin embargo, los SO móviles son bastante más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

Un SO móvil necesita ser fiable y tener una gran estabilidad ya que incidencias habituales y toleradas en computadoras personales, tales como reinicios o caídas, no tienen cabida en un dispositivo móvil. Además, debe adaptarse adecuadamente a las limitaciones de memoria y procesamiento de datos, proporcionando una ejecución exacta y rápida al usuario.

Tabla N°1 : Ventajas y desventajas del sistema operativo Android

Sistema operativo Móvil - Android		
Introducción	Ventaja	Desventaja
Está basado en una versión modificada del kernel de Linux, y es el sistema operativo de Google, actualmente es uno de los sistemas operativos en alza del mundo y se distribuye en numerosos dispositivos de diferentes marcas.	En definitiva al ser desarrollada de forma abierta, se trata de una ventaja tanto para los que desarrollan sus aplicaciones como para sus usuarios. Puedes personalizar tu teléfono al máximo y modificar funciones del teléfono simplemente instalando una aplicación.	Debido a la gran variedad de terminales existen diferentes tipos de pantallas, velocidades, lo que genera que en unos dispositivos existan versiones superiores e inferiores de Android, lo cual genera fragmentación, este es uno de sus principales inconvenientes, por ejemplo en algunos dispositivos se pueden ejecutar juegos y en otros no.

Fuente: (Ana & Gader, 2011)

Tabla N°2 : Ventajas y desventajas del sistema operativo iOS

Sistema operativo Móvil - iOS		
Introducción	Ventaja	Desventaja
Está basado en una variante del kernel de MAC OS X y es la apuesta de la marca Apple para sus dispositivos iPhone.	Buen diseño, funcionalidad, facilidad de uso y una variedad de aplicaciones y juegos enorme lo convierten en un referente, también posee una de las tiendas de aplicaciones y juegos más grande del mundo (App Store) Su perfecta integración con servicios en la nube y equipos de sobremesa, especialmente Mac, es otro de sus puntos fuertes. Tu correo, tus redes sociales... podrás estar siempre conectado.	Además, si quieres disfrutar de un iPhone, sólo te queda desembolsar un precio bastante alto puesto que sólo hay un fabricante y un modelo.

Fuente: (Ana & Gader, 2011)

Tabla N°3 : Ventajas y desventajas del sistema operativo Windows Phone.

Sistema operativo Móvil – Windows Phone		
Introducción	Ventaja	Desventaja
<p>Anteriormente llamado Windows Mobile es un sistema operativo móvil compacto desarrollado por Microsoft, y diseñado para su uso en teléfonos inteligentes (Smartphones) y otros dispositivos móviles.</p>	<p>El sistema gráfico es sencillo y agradable, tenemos acostumbrarnos, y más nos vale, pues el nuevo windows 8 tiene un parecido enorme con este sistema. Un diseño moderno, práctico, atractivo y con características innovadoras han sorprendido ya a más de uno. Windows Phone cuenta con una gran inversión y se ha diseñado para competir con los más grandes fabricantes de dispositivos móviles.</p>	<p>La variedad de móviles con Windows Phone no es tan amplia como la que ofrecen Android o Symbian, aunque está en crecimiento. La cantidad de aplicaciones disponibles en estos momentos es baja, aunque están facilitando el trabajo a los desarrolladores para llenar el hueco rápidamente.</p>

Fuente: **(Ana & Gader, 2011)**

Tabla N°4 : Ventajas y desventajas del sistema operativo Symbian.

Sistema operativo Móvil – Symbian		
Introducción	Ventaja	Desventaja
<p>Nokia es una de las marcas más conocidas del mundo y seguro que alguna vez has tenido un móvil de esta marca en las manos</p>	<p>Nokia es una de las marcas más conocidas del mundo y seguro que alguna vez has tenido un móvil de esta marca en las manos. También cuenta con un amplio mercado de aplicaciones externas y con una tremenda variedad de dispositivos disponibles. Sus precio asequible de muchos de sus modelos.</p>	<p>Ha perdido mucho protagonismo con la llegada de iPhone y Android, sobre todo en los smartphones punteros. Por no tener esa variedad de aplicaciones de los que atesoran otras marcas, pero aun así es un sistema operativo que no defrauda.</p>

Fuente: (Ana & Gader, 2011)

Tabla N°5 : Ventajas y desventajas del sistema operativo Blackberry

Sistema operativo Móvil – Blackberry		
Introducción	Ventaja	Desventaja
<p>Es un sistema operativo multitarea (OS) para el BlackBerry, lo que permite un uso intensivo de los dispositivos de entrada.</p>	<p>Principalmente los BlackBerry se utilizan en el mundo empresarial donde están fuertemente arraigados, ya que permite la sincronización mejorada con servicios de servidores de empresas como Microsoft Exchange o Lotus Notes. Perfecto para el uso de correo electrónico, Blackberry destaca también por los aspectos de seguridad y por su principal seña de identidad es el teclado QWERTY permitiendo escribir con bastante comodidad y rapidez</p>	<p>Ha quedado atrás al no poder competir con las tiendas de aplicaciones de los diferentes markets de la competencia; aunque en entornos empresariales sigue siendo el rey, resistiendo a dejar esta cuota de mercado que a sus competidores se les resiste.</p>

Fuente: (Ana & Gader, 2011)

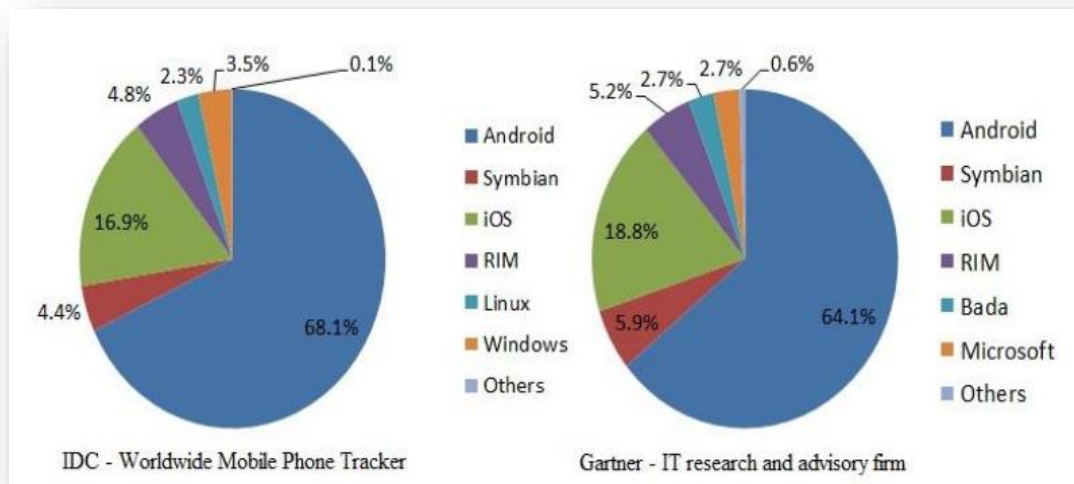


Ilustración 1 : Sistemas operativos móviles más usados.

Fuente: (Timofti, 2013)

1.2.4 Tipo de Aplicaciones para dispositivos Móviles.

En los últimos años el mercado de los dispositivos móviles, en especial Smartphone, ha mostrado un crecimiento notable en Perú (Gestión, 2015). Las plataformas que más han crecido son Android e iOS (El Comercio, 2015). El principal reto para los proveedores de aplicaciones móviles es proporcionar soluciones para todas las plataformas, pero tiene un alto costo. La solución ideal a este problema es crear y mantener una única aplicación para todas las plataformas.

A continuación se presentan tres enfoques en el desarrollo de aplicaciones para dispositivos móviles: un enfoque nativo y dos enfoques multiplataforma (web e híbrido).

Tabla N°6 : Aplicaciones Híbridas ventaja y desventaja.

APLICACIONES HÍBRIDAS PARA DISPOSITIVOS MÓVILES

Las aplicaciones híbridas combinan lo mejor de los dos tipos de aplicaciones anteriores. Se utilizan tecnologías multiplataforma como HTML, JavaScript y CSS, pero se puede acceder a buena parte de las capacidades específicas de los dispositivos.

En resumen, son desarrolladas utilizando tecnología web y son ejecutadas dentro de un contenedor web sobre el dispositivo móvil

Ventaja	Desventaja
Entre las principales ventajas de esta metodología se mencionan la posibilidad de distribución de la aplicación a través de las tiendas de aplicaciones, la reutilización de código para múltiples plataformas y la posibilidad de utilizar las características de hardware del dispositivo.	Una de las desventajas es que, al utilizar la misma interfaz para todas las plataformas, la apariencia de la aplicación no será como la de una aplicación nativa. Finalmente la ejecución será más lenta que la ejecución en una aplicación nativa.

Fuente: (Ana & Gader, 2011)

Tabla N°7 : Aplicaciones web ventaja y desventaja.

APLICACIONES WEB PARA DISPOSITIVOS MÓVILES

Son diseñadas para ser ejecutadas en el navegador del dispositivo móvil. Estas aplicaciones son desarrolladas utilizando HTML, CSS y JavaScript, es decir, la misma tecnología que la utilizada para crear sitios web. (Ana & Gader, Repositorio Graduate, 2011)

Ventaja	Desventaja
Los dispositivos no necesitan la instalación de ningún componente en particular, ni la aprobación de algún fabricante para que las aplicaciones sean publicadas y utilizadas. Solo se requiere acceso a internet.	Por contrapartida, disminuyen la velocidad de ejecución y podrían llegar a ser menos atractivas que las aplicaciones nativas.
Las actualizaciones de la aplicación son visualizadas directamente en el dispositivo, ya que los cambios son aplicados sobre el servidor y están disponibles de inmediato. En resumen, es rápido y fácil de poner en marcha.	Tienden a tener baja performance por problemas de conectividad.
Cuenta con independencia de plataforma. No necesita adecuarse a ningún entorno operativo. Solo es necesario un navegador.	Finalmente este tipo de aplicaciones no pueden utilizar todos los elementos de hardware del dispositivo, como por ejemplo, cámara, GPS, entre otros.

Fuente: (Ana & Gader, 2011)

Tabla N°8 : Aplicaciones nativas ventaja y desventaja.

APLICACIONES NATIVAS PARA DISPOSITIVOS MÓVILES

Son aquellas que se conciben para ejecutarse en una plataforma específica, es decir, se debe considerar el tipo de dispositivo, el sistema operativo a utilizar y su versión. (Ana & Gader, Repositorio Graduate, 2011)

Ventaja	Desventaja
Estas tienen un proceso de auditoría para evaluar si la aplicación se adecúa a los requerimientos de la plataforma a operar	Cuando la aplicación está lista para ser distribuida debe ser transferida a tiendas de aplicaciones específicas de cada sistema operativo
Tienden a interactuar con todas las capacidades del dispositivo (cámara, GPS, acelerómetro, agenda). Además no es necesario poseer acceso a internet. Su ejecución es rápida, puede ejecutarse en modo background y notificar al usuario cuando ocurra un evento que necesite su atención.	Se debe utilizar un lenguaje de programación diferente según la plataforma. Esto conlleva a mayores costos de actualización y distribución de nuevas versiones.

Fuente: (Ana & Gader, 2011)

1.2.5 Apache Cordova

Apache Cordova es un conjunto de APIs de dispositivos que permiten a un desarrollador de aplicaciones móviles acceder a las funciones del dispositivo nativo como la cámara o acelerómetro desde JavaScript. Combinado con un marco de interfaz de usuario, tales como jQuery Mobile, Dojo Mobile, Sencha Touch entre otros, esto permite una aplicación para teléfonos inteligentes que se desarrolló con sólo HTML, CSS y JavaScript.

Cuando se utiliza la API de Cordova, una aplicación puede ser construida sin ningún código nativo (Java, Objective-C, etc) por el desarrollador de la

aplicación. En su lugar, se utilizan tecnologías web, y ellos están alojados en la propia aplicación a nivel local (por lo general no en un servidor http remoto).

Y debido a que estas API de JavaScript son consistentes a través de múltiples plataformas de dispositivos y construido en estándares web, la aplicación debe ser portable a otras plataformas de dispositivos con un mínimo o ningún cambio.

Aplicaciones que utiliza Cordova todavía se empaquetan como aplicaciones que utilizan los SDK de las plataformas, y pueden estar disponibles para la instalación desde la tienda de aplicaciones de cada dispositivo.

Cordova ofrece un conjunto de bibliotecas de JavaScript uniformes que se pueden invocar, con el código de respaldo nativo específico del dispositivo para aquellas bibliotecas JavaScript. Cordova está disponible para las siguientes plataformas: iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada y Symbian. (Apache Cordova™, 2015)

1.2.6 Arquitectura de aplicaciones para dispositivos móviles.

Las aplicaciones móviles y sus arquitecturas pueden ser clasificadas de diversas maneras. Entre las más importantes se encuentran su aplicabilidad, arquitectura, funcionalidad y rango. (Ana & Gader, 2011)

Tabla N°9 : Arquitectura de aplicaciones para dispositivos móviles

Tipos de Arquitectura	Breve Descripción
Aplicabilidad	<p>Una aplicación móvil puede ser soportada por un dispositivo móvil específico o puede ser independiente del tipo de plataforma móvil.</p> <p>Dependiendo de las características del grupo de usuarios es necesario diseñar una aplicación capaz de funcionar en múltiples plataformas.</p> <p>En el desarrollo de una aplicación dependiente de la plataforma, los desarrolladores podrán sacar el máximo provecho del dispositivo haciendo uso de su API, con la desventaja de que todos los usuarios que no disponen de dicha plataforma no podrán correr la misma.</p>
Arquitectura	<p>Un modelo arquitectónico de un sistema trata sobre las partes que lo componen y la manera en que ellas interactúan para lograr la tarea requerida.</p> <p>El objetivo de los modelos arquitectónicos es garantizar que la estructura desarrollada satisfaga los requerimientos actuales y futuros</p>
Funcionalidad	<p>La funcionalidad y la arquitectura son conceptos muy relacionados, se puede clasificar en tres diferentes categorías: sistemas de mensajes, aplicaciones Web y aplicaciones similares a las de escritorio.</p>
Rango	<p>SMS es la tecnología móvil presente en la mayoría de los dispositivos móviles, por lo cual una aplicación basada en un sistema de mensajes SMS estaría disponible para casi la totalidad de los dispositivos móviles.</p>

Fuente: (Ana & Gader, 2011)

1.2.7 Tecnología Bluetooth

Bluetooth es una tecnología inalámbrica de corto alcance que posibilita la transmisión de audio y datos entre varios dispositivos. La estructura uniforme de su especificación, que actualmente se encuentra en la versión 4.0, permite albergar comunicaciones en un amplio abanico de equipos

a) Bluetooth Low Energy

“Bluetooth Low Energy” (BLE) es una nueva prestación de la especificación Bluetooth v4.0 orientada principalmente a aplicaciones de bajo consumo y que no requieren una elevada tasa de transmisión de datos. Estas nuevas características abren la puerta a un amplio abanico de posibilidades en objetos más pequeños como los que existen en el ámbito médico, deportivo, de la seguridad o de la domótica.

En el año 2001, investigadores de Nokia se dieron cuenta que había varios escenarios a los que las tecnologías inalámbricas disponibles en ese momento no podían aplicarse. Para solucionar ese problema, el centro de investigación de Nokia empezó el desarrollo de una tecnología inalámbrica basada en el estándar Bluetooth pero que ofreciera un menor consumo y un menor precio de coste de los componentes. Los resultados se publicaron en 2004 utilizando el nombre de “Bluetooth Low End Extension”.

Tras un desarrollo más exhaustivo trabajando conjuntamente con otras compañías, la tecnología se hizo pública en octubre del año 2006 con el nombre de “Wibree”. Más tarde, en junio de 2007 y tras las negociaciones con los miembros del Bluetooth SIG, se acordó incluir “Wibree” en una futura especificación Bluetooth como una tecnología de muy baja potencia. Ello desembocó, en diciembre del año 2009, en el anuncio de la adopción de esta tecnología en la próxima versión 4.0 de la especificación Bluetooth con el nombre de “Bluetooth Low Energy”.

Esta adopción se hizo para que los futuros dispositivos fueran compatibles con el universo Bluetooth existente en el momento, asegurándose así la continuidad del proyecto y abriendo más posibilidades a la tecnología. Finalmente, el 30 de junio de 2010 se publicaba la especificación v4.0 de Bluetooth. (Morales, 2011)

b) Estimote Beacon

Estimote es la empresa proveedora de los dispositivos Beacon los cuales los ofrecen al mercado 3 por 99 dólares. La empresa ofrece una plataforma web en la cual se puede gestionar la configuración de los dispositivos, así mismo también provee de una aplicación móvil que hace más fácil la configuración de los Beacons.

Técnicamente hablando, Estimote Beacon es una micro computadora. Tiene una poderosa 32-bit CPU ARM® Cortex M0 con la memoria 256 kB flash, acelerómetro, sensor de temperatura y lo que es más importante - 2,4 GHz Bluetooth 4.0 Inteligente (también conocido como BLE o Bluetooth baja energía) de radio bidireccional.

No confunda Bluetooth LE con la primera versión de Bluetooth (la que consume mayor energía). Es un estándar completamente nuevo, rediseñado desarrollado por Nokia™ que ahora se implementa en todos los smartphones modernos, como el iPhone de Apple 4S™ o Samsung Galaxy S III™ y otros dispositivos inteligentes como Google Glass™, rastreadores de Fitbit o smartwatches.

Se llama de baja energía, ya que realmente lo son. Dispositivos BLE pueden durar hasta a 2 a más años con una sola batería. (Estimote Inc, 2015)

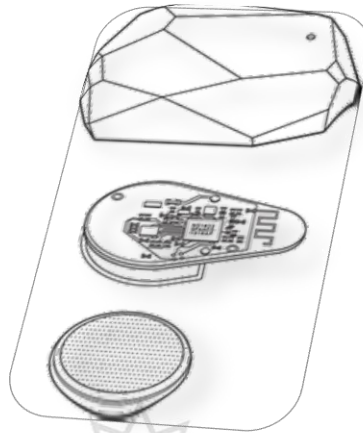


Ilustración 2 : Imagen transversal de un Estimote Beacon
Fuente: (Estimote Inc, 2015)

- **Gama de difusión y proximidad de un Beacon**

Haciendo una analogía se piensa en un faro como una pequeña torre que se instala en una ubicación fija y difunde su presencia a todos los buques (smartphones) alrededor. Podrían ser tan poco como 2 pulgadas y en la medida de como 230 pies (aprox. 70 metros) de distancia.

El alcance máximo exacta depende del entorno, por supuesto. Bluetooth está utilizando el mismo tipo de ondas de radio como routers WiFi 2.4Ghz. La señal puede ser difractada, interferida o absorbida por el agua, (incluyendo el cuerpo humano).

Teléfonos inteligentes u otros dispositivos de la gama alta recogen la señal de radio Bluetooth y estiman su distancia hacia el Beacon midiendo la intensidad de la señal recibida (RSSI). Cuanto más cerca esté del Beacon, más fuerte es la señal. Dependiendo de la implementación, dispositivos podrían sondear la señal de cada segundo (1 Hz) o 10 veces por segundo (10 Hz). Cuanto más a menudo se hace, la señal más estable y la aplicación más sensible es, y la mejor experiencia del cliente se puede lograr.

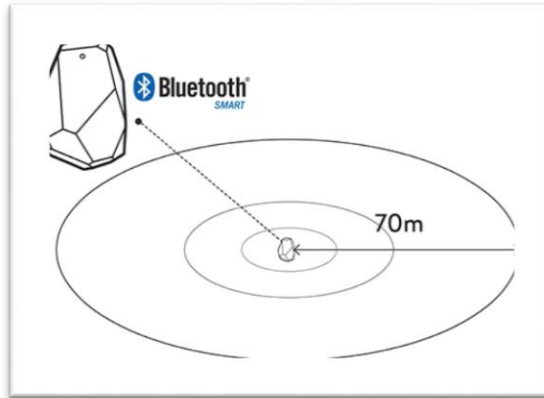


Ilustración 3 : Rango de detección de un Estimote Beacon
Fuente: (Estimote Inc, 2015)

- **Los identificadores únicos**

El Smartphone puede recoger la señal de más de un Beacon al mismo tiempo. Si hay tres o más Beacons dentro de su alcance, el teléfono puede calcular la distancia a cada Beacon y utilizar estos datos para calcular su ubicación relativa. Esta técnica es algo que los marineros han utilizado para las generaciones. Ellos navegaban gracias a marcadores fijos únicos como faros o las estrellas. Estimote Beacon tiene un código único de identificación para evitar conflictos.



Ilustración 4 : Identificación única de Estimote Beacon
Fuente: (Estimote Inc, 2015)

1.2.9 Modelos de desarrollo de software móvil

En el mundo del desarrollo de software existen muchos métodos de desarrollo, cada uno con sus puntos fuertes y sus puntos débiles.

En el caso del desarrollo de aplicaciones móviles sucede lo mismo, y cuando se plantea qué método elegir se deberá escoger en función de las necesidades del negocio. Entre las principales tenemos:

a) Metodología híbrido para el desarrollo ágil

La aproximación metodológica se apoya en una combinación del desarrollo adaptativo de software y el diseño de nuevos productos. Esto supone una decisión crítica para resaltar más del lado del desarrollo de productos que del lado de la gestión de proyectos, lo cual quiere decir que una de las características más sensibles, desde el punto de vista metodológico, para la consolidación de una metodología híbrida propia de un entorno móvil, es la presión de los plazos para llegar al mercado, un mercado volátil y altamente dinámico.

Para empezar, a pesar de ser uno de los pocos intentos metodológicos serios recientes, debemos mostrarnos críticos con la propuesta que encontramos. Se apoya en un proceso iterativo de diseño híbrido de metodologías (Hybrid Methodology Design, HMD) y parte del ciclo de vida tradicional representado en la mayoría de desarrollos de aplicaciones móviles actuales.

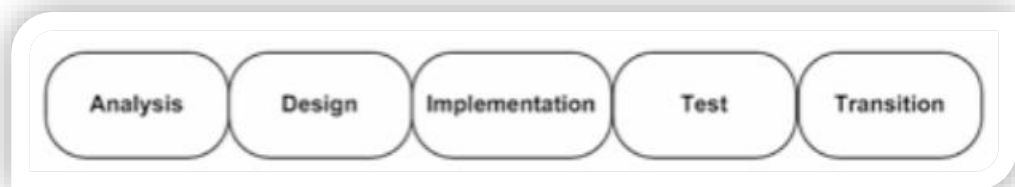


Ilustración 5 : Metodología híbrido para el desarrollo ágil
(Camarero, Fumero, Wertersky, Rodríguez, & Blanco, 2009)

b) Metodología Mobile-D

Se podría pensar que Mobile-D es una creación un tanto antigua, ya que se desarrolló como parte de un proyecto finlandés, ICAROS, allá por 2004. Sin embargo, creemos que vale la pena mencionarlo por dos razones. Primera; fue creado mediante un proyecto de cooperación muy estrecha con la industria. El grueso del trabajo fue realizado por los investigadores del VTT. Aun así la metodología de diseño se elaboró con una participación importante de las empresas de TI finlandesas. Tal como se pudo en algunas investigaciones llevadas a cabo que Mobile-D, no se alejará demasiado de las reglas de desarrollo de las aplicaciones comerciales. Segundo, Mobile-D es una mezcla de muchas técnicas. Tal como se verá luego, los investigadores no dudaron en echar mano de las prácticas habituales de desarrollo software. Pero, al mismo tiempo, consiguieron crear una contribución original para el nuevo escenario del desarrollo de aplicaciones para sistemas móviles.

Ciclo de desarrollo de Mobile-D

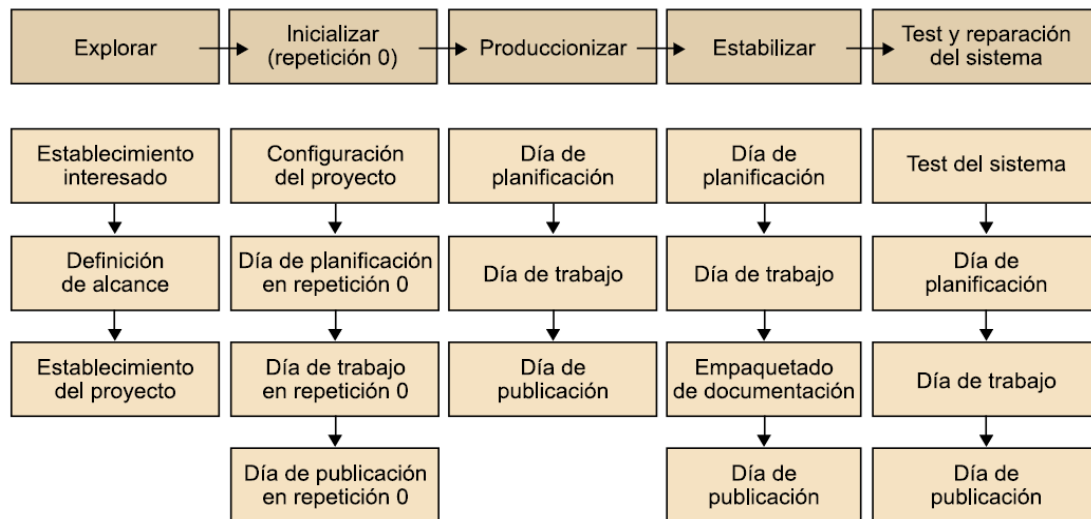


Ilustración 6 : Ciclo de desarrollo de la metodología Mobile D
Fuente: (Camarero, Fumero, Wartersky, Rodriguez, & Blanco, 2009)

c) Metodología enfoque pragmático

Propia de esta metodología dicen haberlo probado obteniendo una certificación CMMI de nivel 2. Esto parece ser una ventaja comparativa importante frente a otras metodologías, puesto que la contratación de empresas para la externalización del desarrollo software se rige por la auditoría de los ciclos y técnicas de desarrollo que utilizan (y CMMI es una de las métricas de aseguramiento de calidad más aceptadas en el sector). Adicionalmente, sus creadores han introducido Mobile-D en numerosos proyectos de desarrollo con clientes reales. La base inicial de 4 casos de estudio se ha desarrollado durante años y afirman sus autores que los ciclos de desarrollo se han actualizado y mejorado a partir de la experiencia obtenida de los enfoques pragmáticos.

(Camarero, Fumero, Wartersky, Rodriguez, & Blanco, 2009)

1.3 Definición de términos básicos

Para la elaboración de esta tesis, se consideraron algunos conceptos que son importantes detallar para una mayor y mejor comprensión del término; estos se describen a continuación:

- **API** es el conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.
- **Beacon:** dispositivo de aproximadamente 2 centímetros que emite una señal en la onda corta de la tecnología Bluetooth, cuyo alcance máximo es de 50 metros.
- **Bluetooth** es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4GHz.
- **E911** The FCCs wireless Enhanced 911 (E911) es una normativa de conexión inalámbrica (Federal Communications Commission, 2015)
- **iBeacons:** Es un Beacon pero de la empresa Apple.

- **Jailbreak** conocido como fuga de la cárcel al proceso de suprimir algunas de las limitaciones impuestas por Apple en dispositivos que utilicen el sistema operativo iOS mediante el uso de kernels modificados, permitiendo al usuario descargar aplicaciones, extensiones y temas que no estén disponibles a través de la App Store oficial.
- **MicroKernel** es un tipo de núcleo de un sistema operativo que provee un conjunto de primitivas o llamadas mínimas al sistema para implementar servicios básicos como espacios de direcciones, comunicación entre procesos y planificación básica
- **Middleware** es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, o paquetes de programas, redes, hardware y/o sistemas operativos
- **PDA** asistente digital personal, computadora de bolsillo, organizador personal o agenda electrónica de bolsillo, es una computadora de mano originalmente diseñada como agenda personal electrónica
- **Radiofrecuencia** también denominado espectro de radiofrecuencia o RF, se aplica a la porción menos energética del espectro electromagnético,
- **SDK** Un kit de desarrollo de software o es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etc.
- **Teclado QWERTY** es el teclado más común que se usan en las Pcs, se diseñó con el propósito de lograr que las personas escribieran más rápido distribuyendo las letras de tal forma que se puedan usar las dos manos para escribir la mayoría de las palabras.
- **Trilateración y Triangulación:** son Técnicas de estimación de la posición
- **UUID:** Identificador único del dispositivo Beacon.
- **RSSI:** El indicador de fuerza de la señal recibida.

CAPÍTULO II METODOLOGIA

2.1 Materiales

A base de la revisión realizada, de manera comparativa entre las metodologías verificadas y validadas para el desarrollo de aplicaciones móviles, que permita integrar en forma transparente y flexible con las aplicaciones del sistema académico y administrativo de la Facultad de Ingeniería de Sistemas de la Universidad de San Martín de Porres.

A continuación se presenta los materiales y método indicado anteriormente.

a) Herramientas colaborativas para la elaboración del proyecto

Tabla N°10 : Herramientas de redacción y edición

HERRAMIENTAS DE REDACCIÓN Y EDICIÓN		
Software	Versión	Descripción
Microsoft Word	Office 2013	Herramienta de Microsoft Office destinado al procesamiento de textos la cual se usara para la redacción de los documentos que formen parte del proyecto.
Microsoft Excel	Office 2013	Herramienta de Microsoft Office que permitió crear hojas de cálculos, tablas y cuadros estadísticos, que muestren información sobre el proyecto
Microsoft Power Point	Office 2013	Herramienta de Microsoft Office que permitió crear las presentaciones para sustentar en manera resumida el contenido del proyecto.
Cacoo	Libre	Cacoo es una herramienta online para hacer diagramas de diversos tipos, desde el esquema de una oficina hasta diagramas UML, pasando por el prototipado de pantallas.
Snagit 12	Libre	Herramienta que permite realizar capturas de pantalla, editar, organizar y compartir, resulta fácil crear efectos visuales atractivos que capten la atención.

Fuente: Propia

Tabla N°11 : Herramientas de repositorio y comunicación

HERRAMIENTAS DE REPOSITORIO Y COMUNICACIÓN		
Software	Versión	Descripción
Zotero	Libre	Herramienta gratuita y fácil de usar para ayudarle a recopilar, organizar, citar y compartir sus fuentes de investigación. (Zotero, 2014)
Google Drive	Libre	Herramienta para almacenar y compartir información de datos en la nube, así como también me permite la edición entre unos o más colaboradores en línea.
Google Académico	Libre	Buscador especializado en artículos de revistas científicas, enfocado en el mundo académico, y soportado por una base de datos disponible libremente en Internet que almacena un amplio conjunto de trabajos de investigación científica de distintas disciplinas y en distintos formatos de publicación.

Fuente: Propia

Tabla N°12 : Herramientas de desarrollo

HERRAMIENTAS DE DESARROLLO		
Software	Versión	Descripción
Netbeans	8.1	Interface de desarrollo para software.
GIT	2.4	Para el versionamiento del código se utilizará la tecnología GIT pues nos permite llevar el control sobre los cambios y poder gestionar correctamente las versiones del proyecto.
Apache cordova	5.0	Permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3. Las aplicaciones resultantes son híbridas.

Fuente: Propia

b) Metodología de desarrollo.

El siguiente gráfico indica en alto nivel la metodología de desarrollo para llevar a cabo la hipótesis planteada.

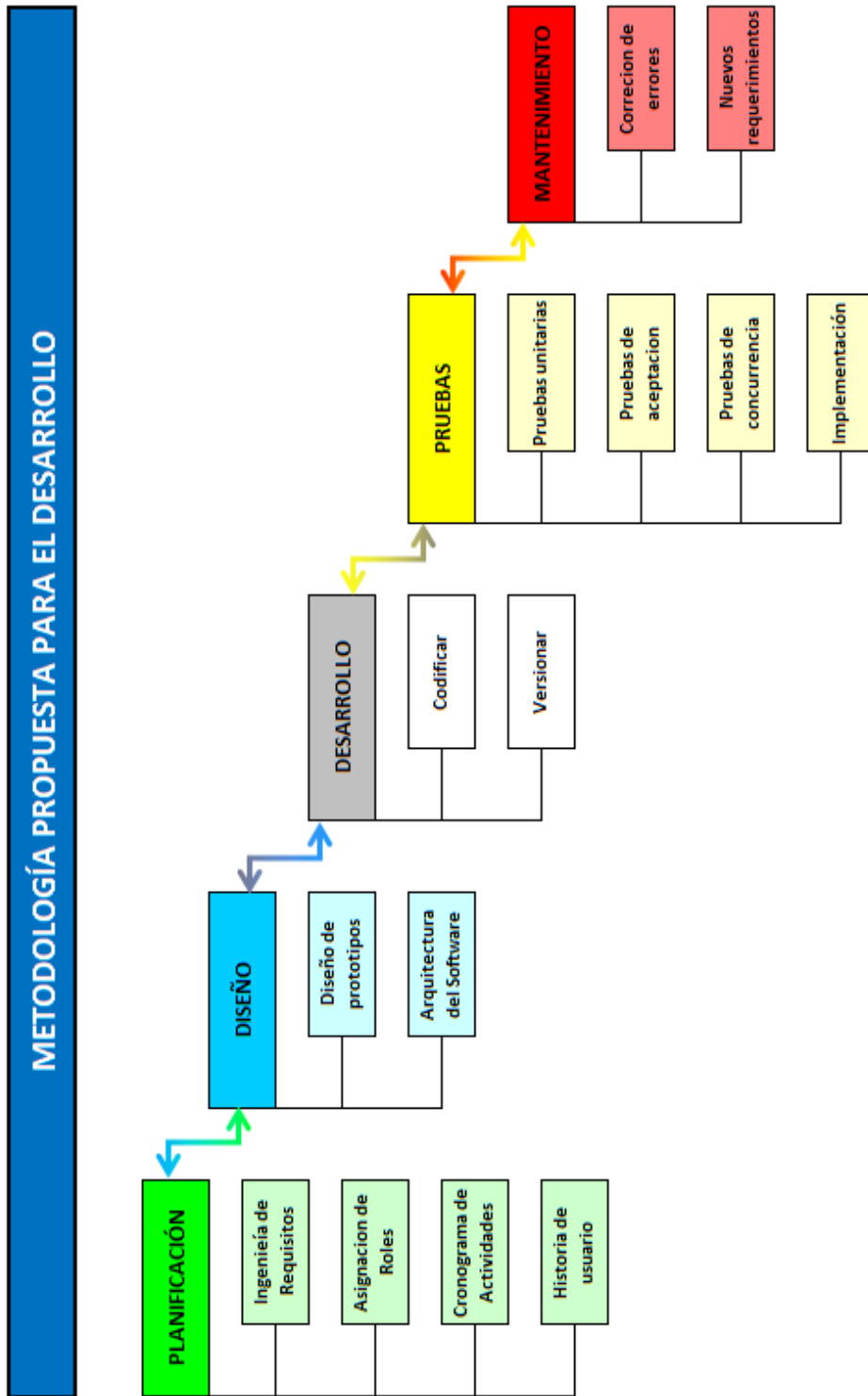


Ilustración 7 : Metodología propuesta de desarrollo ágil
Fuente: Propia

2.1 Planificación

Tabla N°13 : Herramientas colaborativas usadas en la fase de planificación

Herramientas colaborativas.			
Entrada	Rol	Salida	Rol
Asignación de roles	Líder del proyecto	Historia de usuarios	Analista de sistema y analista programador
Ingeniería de requisitos	Líder del proyecto analista de sistema, analista programador	Cronograma de actividades	Líder del proyecto
Entrevistas	Líder del proyecto, analista de sistema y analista programador		
Lluvia de ideas	Analista del sistema y usuarios		
Requisitos del sistema	Cliente , analista de sistema		
Prototipos	Diseñador, analista programador		

Fuente: Propia

En este proceso se captura los requerimientos de los usuarios mediante el uso de técnicas como entrevistas, lluvia de ideas, prototipos y requisitos del sistema, esto permite definir el alcance del sistema y asignar los roles correspondientes necesarios para la implementación de la solución. Este proceso tiene como salida las historias de usuario y el cronograma de actividades.

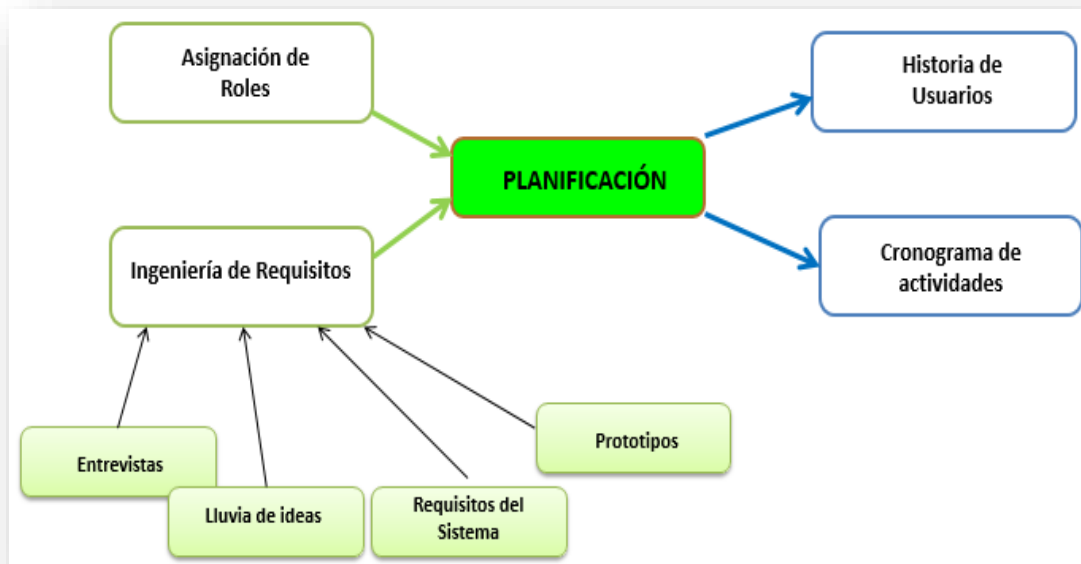


Ilustración 8 : Etapa de planificación
Fuente: Propia

2.1.1 Ingeniería de requisitos

Mediante el uso de la Ingeniería de requisitos se detalla las técnicas o métodos usados en la planificación, la ingeniería de requisitos tiene como propósito producir y analizar requerimientos de los clientes y de la solución.

En esta etapa los usuarios y el analista del sistema, revisan articulan y entienden las características y las restricciones que se dan sobre este y el desarrollo del mismo.

Para nuestro proyecto se elige las técnicas más importantes de la ingeniería de requisitos para llevar a cabo la recolección de requerimientos

a) Entrevistas

Este método se usa para reunir información proveniente de los usuarios mediante una conversación dirigida con un propósito específico la cual se basa en un formato de preguntas y respuestas para conocer aspectos como: metas

de la organización, qué piensa del actual sistema, su compromiso con la organización y su opinión o, entre otras.

Durante la entrevista, el analista del sistema conversa con el entrevistado, aplica una serie de preguntas con formato estructurado y no estructurado, la cual guarda relación con su forma de pensar sobre el sistema actual de asistencia y la solución planteada.

Por lo común, los entrevistados son usuarios del sistema existente, en este caso algunos usuarios son coordinadores académicos, dirección de escuela de ingeniería, docentes y alumnos que proporcionan información para el sistema propuesto.

La información que se obtiene en esta etapa es proveniente de preguntas que son de alto nivel, concretas y puntuales que se realiza en la primera fase de la planeación para obtener información sobre aspectos globales del proceso en estudio, del usuario y posibles mejoras o soluciones.

También con el fin de obtener mayor información se aplica preguntas no estructuradas o abiertas para descubrir sentimientos, opiniones y experiencias personales, o para explorar un proceso o problema, en nuestro caso la falta de control de la asistencia exacta del alumnado al salón de clases. Este tipo de preguntas son siempre apropiadas, además que ayudan a entender la perspectiva del afectado en este caso los clientes/usuarios alumnos que no están influenciados por el conocimiento de la solución.

De esta fase, la calidad de información que se obtiene, depende de habilidad como entrevistador y de preparación para la misma. Los analistas necesitan ser sensibles a las dificultades que se pueden presentar en algunos entrevistados durante la entrevista y saber cómo tratar los problemas potenciales.

b) Lluvia de Ideas

Se aplica la lluvia de ideas con el fin de extraer todas las ideas en torno a un tema o problema planteado que ayuda a estimular la creatividad y participación de los usuarios involucrados a través de reuniones, dinámicas la exposición de manera formal e informal y libre

Hay que tener en cuenta que la dinámica que se aplica en la lluvia de ideas es estructurada básicamente para la aplicación en un entorno presencial donde se asegura la participación de la gran mayoría de usuarios involucrados con la identificación o solución del caso en uso.

c) Prototipos

Esta técnica permite presentar ideas a los usuarios porque ellos pueden ver inmediatamente algunos aspectos claves del sistema, mostrar los prototipos puede provocar que el usuario brinde un mayor número de requerimientos, cambie o elimine algún requerimiento existente.

Los prototipos permiten al usuario ver cómo funciona la solución y lo que pueden hacer con la aplicación propuesta. Además se obtiene más requerimientos cuando el usuario interactúa con los prototipos.

Una forma de presentar los prototipos al usuario es a través de diapositivas y/o prototipos impresos, elaborados por un diseñador o por el mismo analista, la cual incluye una representación o una animación que brinde a los usuarios una visión de las posibilidades del sistema propuesto.

Esta técnica es un estímulo para que los usuarios comiencen a decir lo que necesitan, si ellos expresan demasiados problemas es señal de que se está logrando el cometido ya que cada problema puede conducir a un nuevo requerimiento.

Los prototipos tienen como salida, obtener requerimientos que faltan especificar o aumentar. Es decir se centra la presentación en determinar lo que realmente se quiere del sistema

d) Requisitos del sistema

Una descripción más detallada de los servicios exactos que se proporcionarán y sus restricciones son conocidas como requisitos del sistema. Estos sirven como contrato con los usuarios de la aplicación. A su vez para este proyecto solo se desarrollaron los requisitos funcionales, no funcionales.

- **Requisitos funcionales:**

Se identifica los requisitos funcionales para describir los servicios o funciones que los usuarios esperan de nuestra aplicación, se especifica lo que hace o que proporciona el sistema. Estos requisitos describen también cómo responde el sistema ante distintas entradas por parte del usuario, y su comportamiento frente a situaciones particulares.

- **Requisitos no funcionales:**

Es necesario identificar las restricciones de los servicios del sistema o funciones que ofrece. Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, se define las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

2.1.2 Asignación de roles

Con el objetivo de ayudar a optimizar el presente trabajo se asigna determinados roles que ejercen diferentes funciones y llevan ciertas habilidades para el desarrollo del proyecto de desarrollo de aplicaciones móviles utilizando la metodología ágil. Estos roles son asignados por el líder de proyecto y está en

correspondencia con las características de los miembros del equipo de desarrollo.

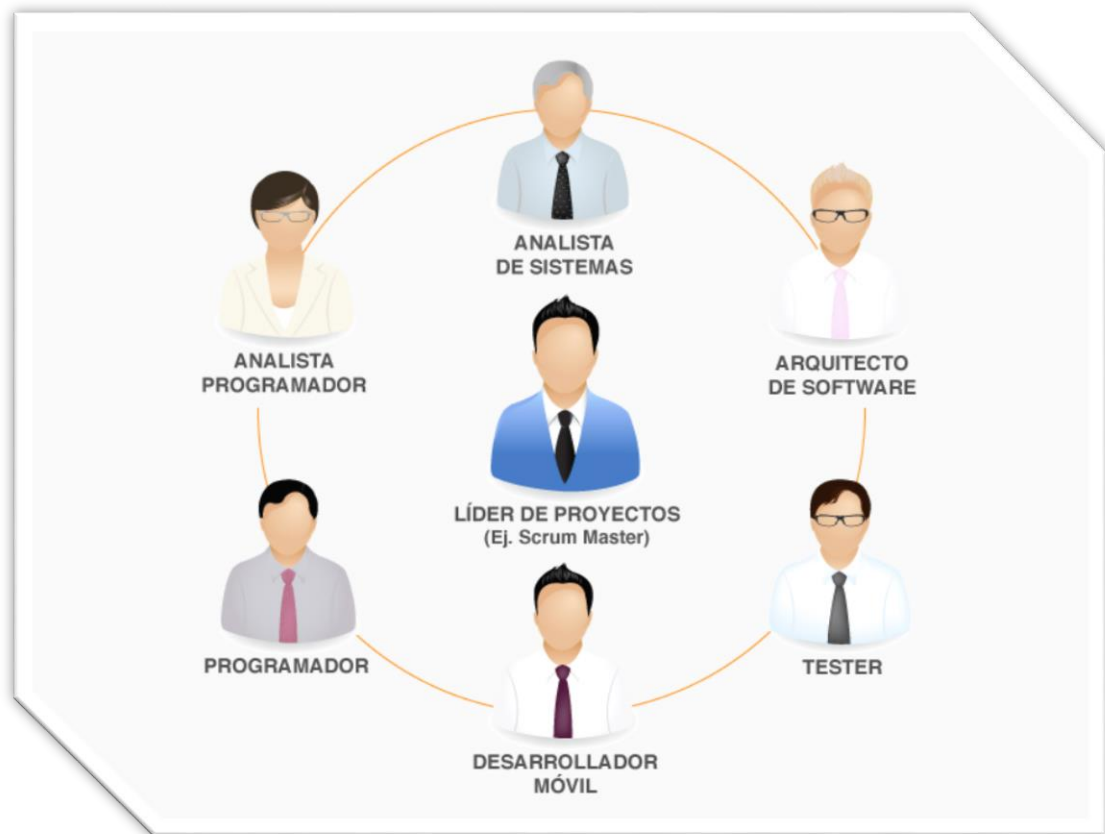


Ilustración 9 : Roles del sistema
Fuente: Propia

2.1.3 Cronograma de actividades

El cronograma de actividades nos permite determinar la duración de cada una de las actividades del proyecto lo cual nos servirá para analizar si el proyecto adecuadamente el tiempo y los recursos asignados al mismo.

- a) **Tarea:** Son actividades de un proyecto que se realizan en una secuencia determinada, pueden clasificarse en:

- b) **Predecesora:** es una tarea que debe comenzar o terminar antes de que otra pueda comenzar.

- a. De Resumen son aquellas que se componen de subtareas.
- b. Duración Es el tiempo que llevará completar una tarea.
- c. Hito: es una tarea sin duración (0 días) que se utiliza para identificar sucesos significativos en la programación como la finalización de una fase importante.

A continuación se presenta la simbología que corresponde a cada caso:




Nombre	Aspecto
Tarea	
División
Progreso	
Hito	

Ilustración 10 : Simbología del cronograma de actividades
Fuente: Propia

2.1.4 Historias de Usuario

Dentro de la etapa de diseño las historias de usuario son usadas en las metodologías de desarrollo ágiles para la especificación de requisitos, son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos, también permiten responder rápidamente a los requisitos cambiantes.

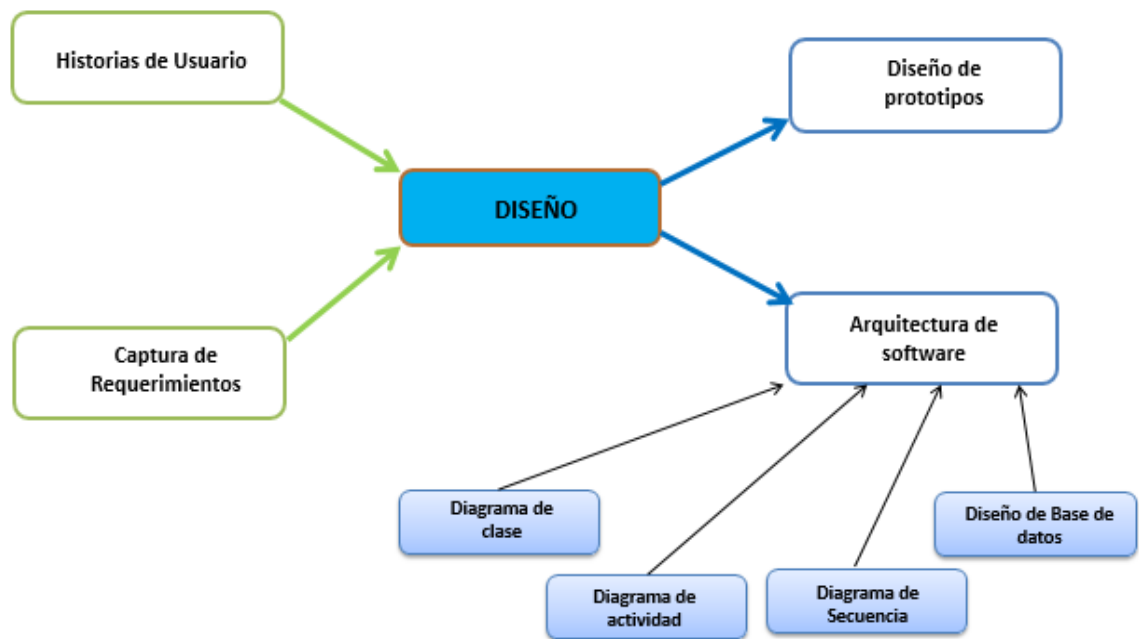


Ilustración 11 : Diseño de las historias de usuario
Fuente: Propia

2.2 Diseño

Tabla N°14: Herramientas colaborativas usadas en la fase de diseño

Herramientas colaborativas.			
Entrada	Rol	Salida	Rol
Historias de usuario	Analista de sistema y analista programador	Arquitectura del software (AS)	Analista de sistema
Cronograma de actividades	Líder del proyecto	Diagrama de actividad	Analista de sistema
Diseño de prototipo		Diagrama de clases	Analista de sistema
		Diagrama de secuencia	Analista de sistema
		Base de datos	Analista
		Prototipos	Programador
			Diseñador

Fuente: Propia

El objetivo de esta etapa es plasmar el pensamiento de la solución mediante diagramas o esquemas, considerando la mejor alternativa al integrar aspectos técnicos, funcionales, sociales y económicos.

2.2.1 Diseño de Prototipo

Una de las técnicas más sencillas para pre visualizar cómo va a quedar la aplicación a desarrollar, en cuestión de diseño y sin tener que empezar a programar ni una línea de código, es hacer un prototipo de App. Hacer un prototipo de App nos va a ahorrar mucho tiempo y trabajo y además va a permitir que veamos de manera muy fiel cómo va a quedar la App, y si es necesario, optimizar el diseño, apariencia y usabilidad de la App, ramas muy importantes en el desarrollo de la aplicación

De esta manera, en caso de que el cliente quisiera realizar cambios en el diseño será mucho más fácil hacerlos desde la herramienta de prototipo de Apps. Incluso podemos realizar varios prototipos para que el cliente escoja el que más se ajusta a lo que quiere.

2.2.2 Arquitectura del software (AS)

La arquitectura de software es una estructura de los componentes de un sistema y se define como relacionan, así mismo las capas que tendrá. En esta etapa el analista de sistemas diseña la arquitectura física y de interfaces.

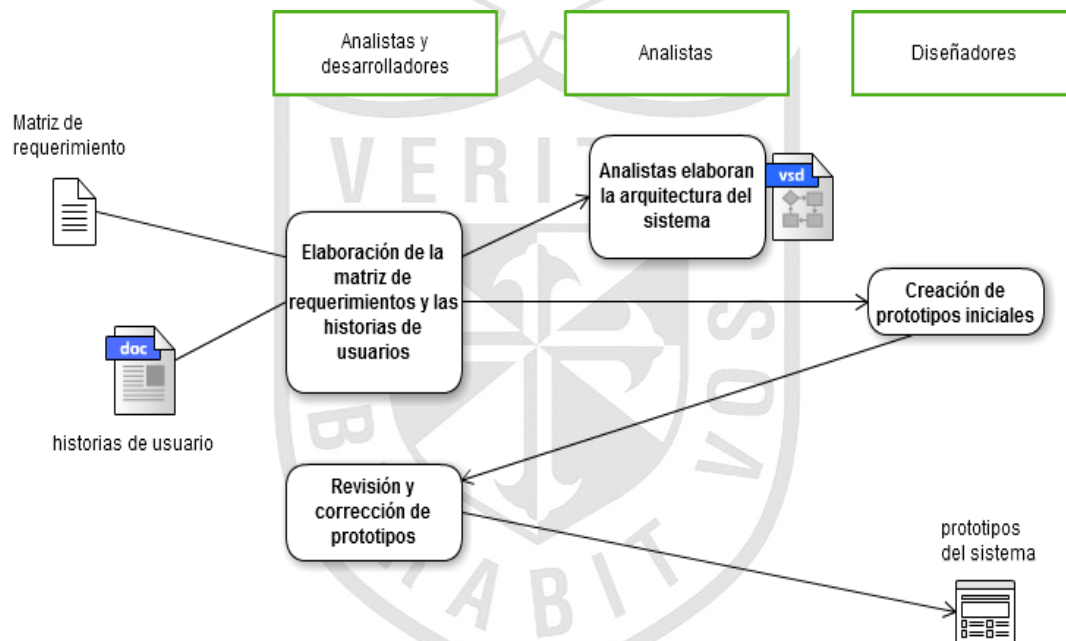


Ilustración 12 : Etapa de diseño de la arquitectura de software
Fuente: Propia

2.2.3 Diagrama de clase

Para la elaboración de nuestro aplicativo se elabora el diagrama de clases por ser un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones y las relaciones entre los objetos. El diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema.

2.2.4 Diagrama de actividad

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, mayormente contienen actividades, que siempre están asociados a una clase, a una operación o a un caso de uso.

Los diagramas de actividad soportan actividades tanto secuenciales como paralelas. La ejecución paralela se representa por medio de iconos de fork/espera, y en el caso de las actividades paralelas, no importa en qué orden sean invocadas (pueden ser ejecutadas simultáneamente o una detrás de otra).

2.2.5 Diagrama de secuencia

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

2.2.6 Base de datos (diseño físico)

Se elabora el diseño físico de la base de datos el cual tendrá todas las tablas correspondientes a la aplicación indicando el tipo de relación que pueden ser de uno a uno, de uno a muchos, de muchos a uno o de muchos a muchos.

Finalmente se genera el script que permita importar en el sistema actual de la universidad.

2.3 Desarrollo

Tabla N°14 : Herramientas colaborativas usadas en la fase de desarrollo

Herramientas colaborativas			
Entrada	Rol	Salida	Rol
Arquitectura del software (AS)	Analista de sistema	Código	Analista programador, programador
Prototipos	Diseñador	versiones	Analista programador, programador
Matriz de requerimientos	Analista del sistema		
Historias de usuario	Analista de sistemas y analista programador		

Fuente: Propia

En este proceso se realiza la codificación respectiva en base a lo planteado en la fase de diseño, el cual está basado en los casos de uso identificados para el control de asistencia de los alumnos , el cual permitirá codificar y realizar la integración de versiones para la aplicación respectiva sobre los diferentes sistemas operativos de los dispositivos móviles. Para el desarrollo de la aplicación móvil se usó apache Córdova que produce aplicaciones híbridas permitiendo la reutilización de componentes.

La ventaja que dio el desarrollo híbrido sobre el desarrollo nativo es que no se necesitó tener un especialista por cada lenguaje sino que con lenguajes comunes como JavaScript, html5 y css3 se creó aplicaciones para diferentes sistemas operativos.

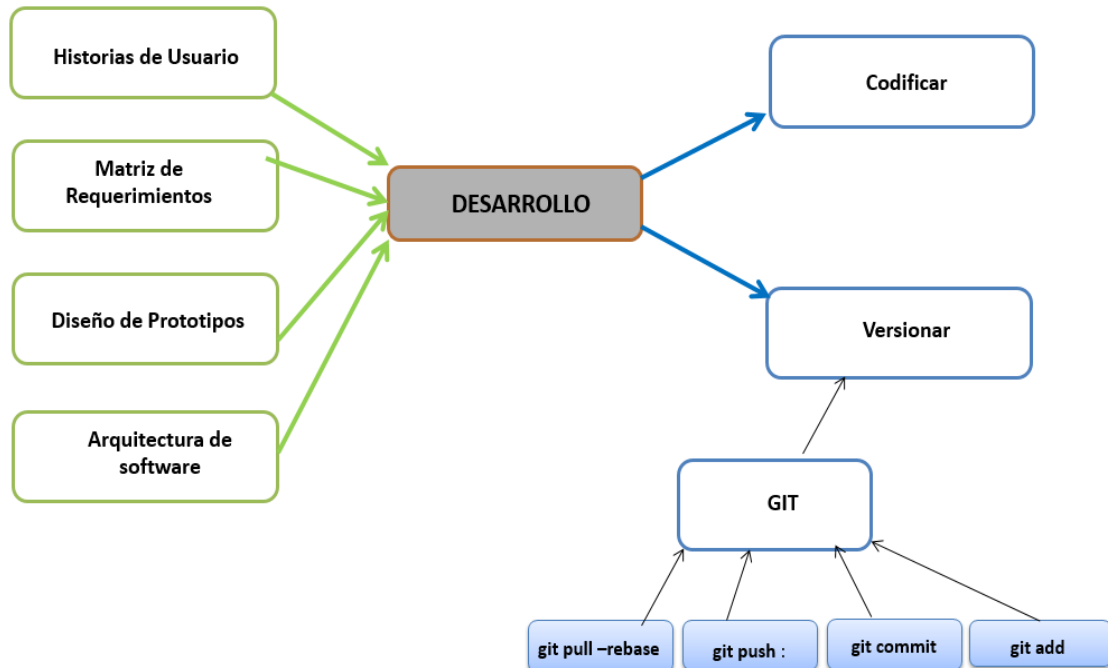


Ilustración 13 : Etapa de desarrollo esquema general
Fuente: Propia

2.3.1 Codificar

La codificación es basada en estándares de calidad y convenciones de formato (Jones, 2010), para que funcionen las convenciones, todos los programadores deben seguirlas.

La matriz de requerimientos sirve a los desarrolladores para saber qué es lo que necesita el cliente y da una idea genérica de la funcionalidad.

Las historias de usuario detallan cómo se desea la funcionalidad de manera escrita y en un lenguaje no técnico que da un alcance más de la funcionalidad a implementarse.

Los diseños de la arquitectura del sistema los cuales contienen el esquema de base de datos, diagramas entre otros proveen información visual y técnica de

dónde deben ubicarse, cómo interactúan y cómo deberán relacionarse los componentes dando una idea clara al desarrollador de la funcionalidad a desarrollarse.

Finalmente los prototipos son fáciles de interpretar por todos los miembros del equipo incluso el cliente que no necesariamente tiene conocimientos técnicos y expresa un gran alcance de cómo debería verse el producto una vez implementado la funcionalidad. Todo este conjunto de información sirve a los desarrolladores para poder plasmar el requerimiento del cliente.

2.3.2 Versionar

Para el versionamiento del código se usa la herramienta Git por ser una tecnología *open source* y permite trabajar de forma distribuida lo cual es una gran ventaja si se desea trabajar con desarrolladores de forma remota.

Así mismo, permite tener revisiones de código retroalimentando al equipo con sugerencias de otros desarrolladores. Los comandos básicos usados en el flujo básico de desarrollo son:

git pull –rebase se utiliza para actualizar un repositorio local con los últimos cambios hechos.

git add : Utilizado para enviar un archivo cambiado a un estado de “stage” es decir listo para ser enviado a la rama de integración.

git commit : Permite guardar una modificación en el repositorio local el cual nos servirá para poder navegar entre diferentes cambios y versiones.

git push : Este comando envía *commits* hacia la rama de integración terminado el envío los cambios podrán ser obtenidos por todos los desarrolladores del equipo.

En la siguiente ilustración se presenta el flujo de trabajo del proceso de desarrollo.

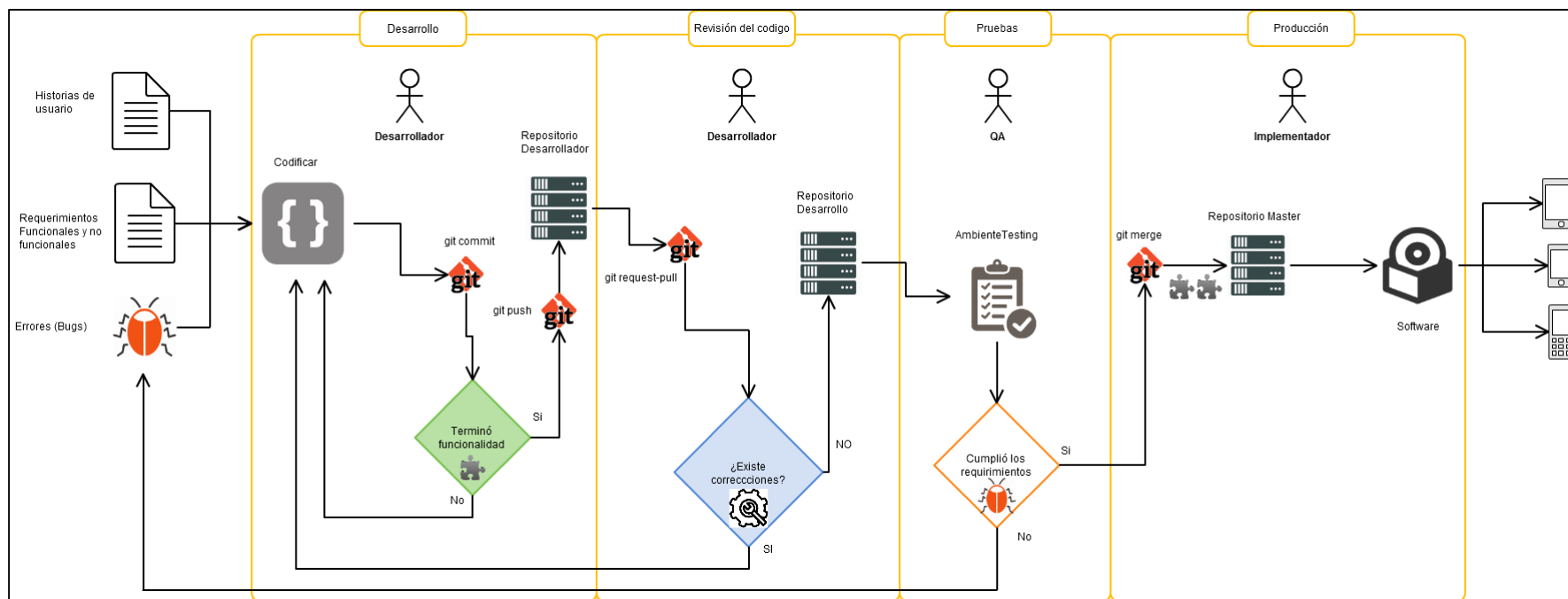


Ilustración 14 : Flujo de trabajo en el proceso de desarrollo
Fuente: Propia

2.4 Pruebas

Tabla N°15 : Herramientas colaborativas usadas en la fase de pruebas

Herramientas colaborativas.			
Entrada	Rol	Salida	Rol
Plan de pruebas	Equipo QA	Pruebas Unitarias	Equipo QA
Escenarios de pruebas	Equipo QA	Pruebas de aceptación	Equipo QA
		Pruebas de concurrencia	Equipo QA
		Implementación	Implementador

Fuente: Propia

Las pruebas de software son las investigaciones empíricas y técnicas cuyo fin es proporcionar información objetiva e independiente sobre la calidad del producto.

Esta actividad forma parte de nuestra metodología ágil porque son básicamente un conjunto de actividades dentro del desarrollo de software y dependiendo del tipo de pruebas, estas actividades son implementadas en cualquier momento del proceso de desarrollo y permite verificar que se cumplan con las especificaciones planteadas desde un inicio por el analista o el propio cliente, y/o eliminar los posibles errores que se hayan cometido en cualquier etapa del desarrollo.

2.4.1 Prueba unitaria

Al desarrollar la aplicación del proyecto, la primera etapa de pruebas a considerar es la etapa de pruebas unitarias o también llamada pruebas de caja blanca (White Box), estas pruebas también son llamadas pruebas modulares ya que nos permiten determinar si un módulo del programa está listo y correctamente terminado.

El objetivo fundamental de las pruebas unitarias es asegurar el correcto funcionamiento de las interfaces, o flujo de datos entre componentes.

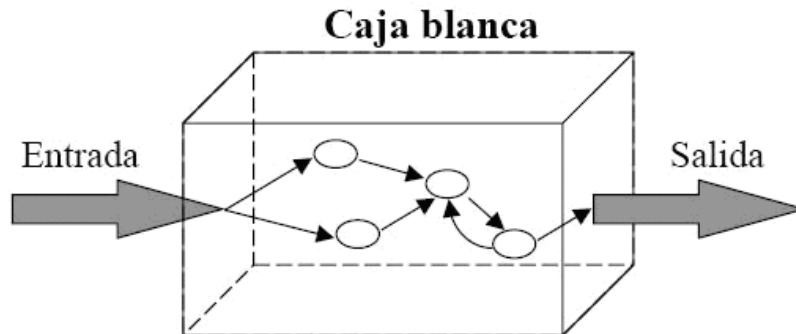


Ilustración 15 : Proceso de prueba mediante caja blanca.
Fuente: (Oré, 2009)

Una prueba unitaria es una forma de comprobar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado.

Características:

Una Prueba Unitaria debe cumplir con los siguientes requisitos:

- a. Automatizable
No debería requerirse una intervención manual.
- b. Completas
Deben cubrir la mayor cantidad de código.
- c. Repetibles o Reutilizables
No se deben crear pruebas que sólo puedan ser ejecutadas una sola vez.
- d. Independientes
La ejecución de una prueba no debe afectar a la ejecución de otra.
- e. Profesionales
Las pruebas deben ser consideradas igual que el código, con la misma profesionalidad y documentación.

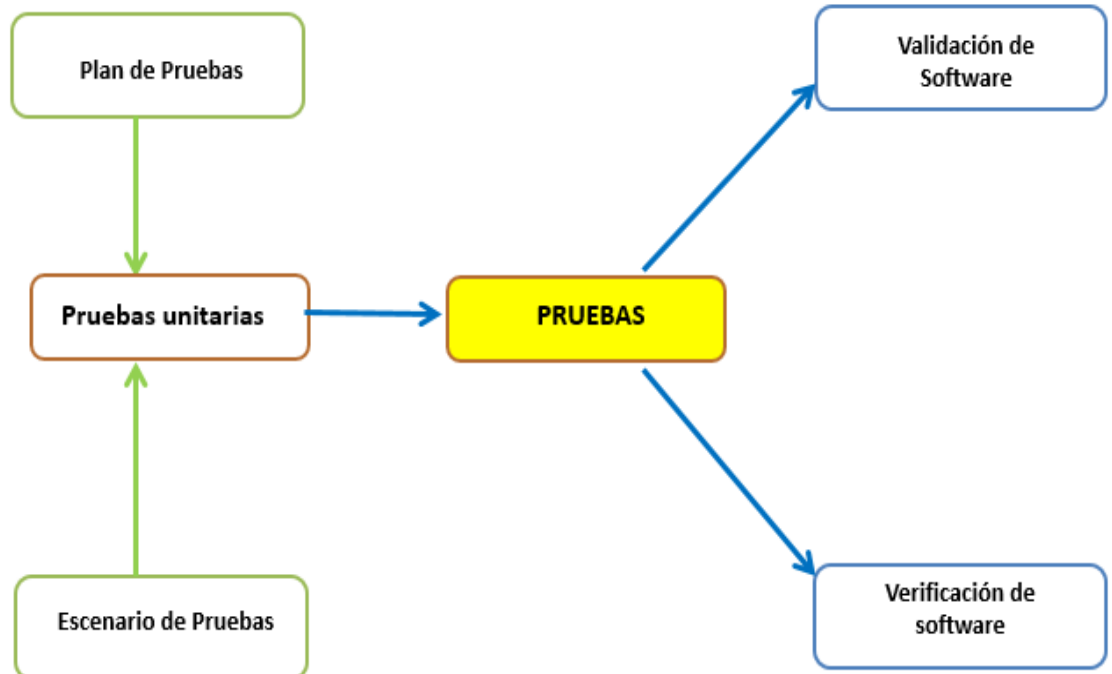


Ilustración 16 : Etapa de pruebas unitarias
Fuente: Propia

2.4.2 Prueba de concurrencia

Para que las prueba de rendimiento o concurrencia cumplan su objetivo, es necesario diseñarlas intentando reproducir los niveles de demanda que la aplicación tendrá en la realidad. El comportamiento de la aplicación, incluido el tiempo de respuesta que perciben los usuarios, dependerá de la concurrencia que exista en un momento determinado sobre la aplicación.

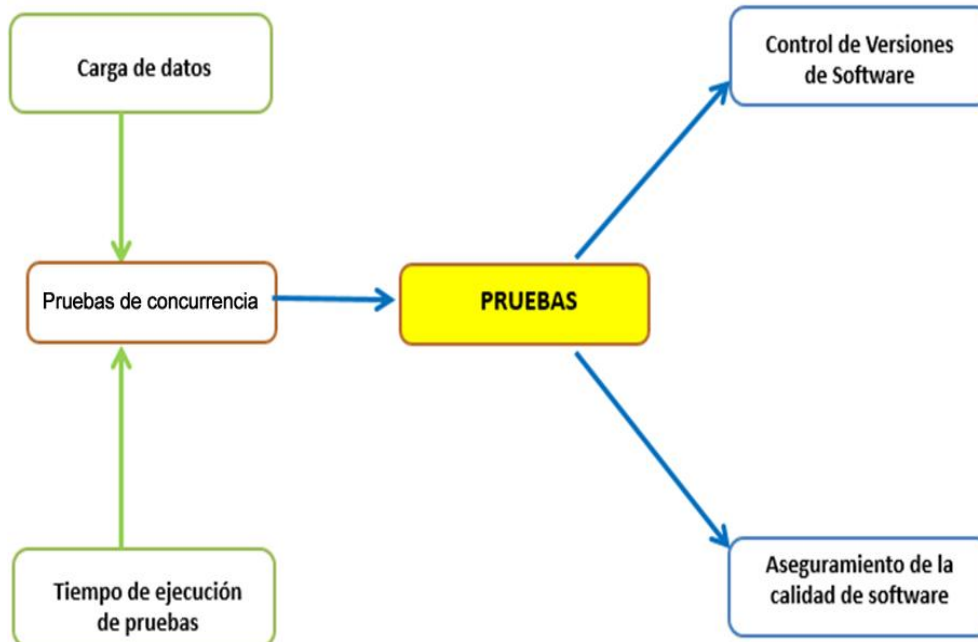


Ilustración 17 : Etapa de prueba de concurrencia
Fuente: Propia

2.4.3 Pruebas de aceptación

Esta prueba se realiza para certificar cada los casos de uso se ejecuten de manera correcta, lo que garantiza que los requisitos funcionales de la aplicación sean eficientes para el usuario, se prueba que el software ofrezca las funciones solicitadas.

El objetivo de esta prueba es verificar que el producto satisfaga sus expectativas.

Una vez que se ha probado que cada módulo funciona bien por separado, que la aplicación puede utilizarse bajo condiciones de operación extremas, que todos los módulos se integran correctamente con el sistema de control de asistencia de la Facultad de Ingeniería y Arquitectura de la USMP, llega el momento de escuchar la opinión del impulsor del proyecto el usuario final, el cual deberá certificar que cumpla con lo requerido.

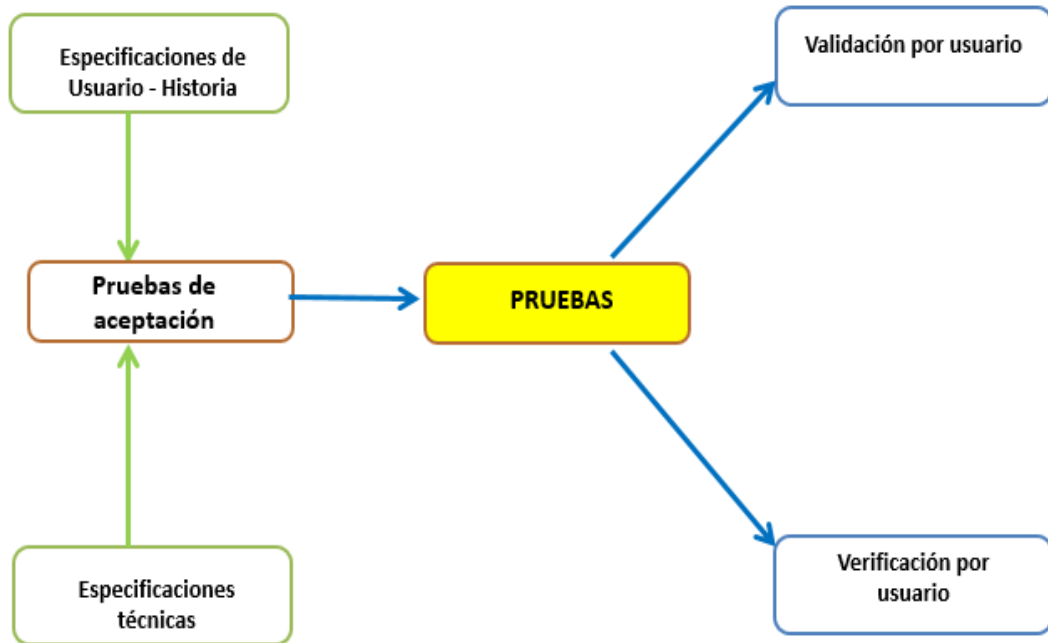


Ilustración 18 : Pruebas de test de aceptación
Fuente: Propia

2.4.4 Implementación

Esta etapa el implementador junta las funcionalidades certificadas por el equipo de QA creando una nueva versión del sistema para luego hacer el despliegue en el ambiente de producción y así el cliente pueda tener un producto final y certificado según los criterios de aceptación provenientes de los requerimientos en las etapas anteriores.

2.5 Mantenimiento

El siguiente cuadro muestra las herramientas colaborativas usadas en el proceso de mantenimiento tabuladas por cara rol.

Tabla N°16 : Herramientas colaborativas

Herramientas colaborativas.			
Entrada	Rol	Salida	Rol
Pruebas Unitarias	Equipo QA	Requerimiento de cambio	Equipo QA
Pruebas de aceptación	Equipo QA	Implementación del cambio	Analista programador y programador
Pruebas de concurrencia	Equipo QA		
Implementación	Equipo QA		

Fuente: Propia

En la etapa de mantenimiento se realiza la corrección de errores, mejoras de las capacidades, eliminación de funciones obsoletas y optimización. Debido a que el cambio es inevitable, se debe desarrollar mecanismos para la evaluación, controlar y hacer modificaciones, así como de ingresar los nuevos requerimientos para el desarrollo de la aplicación móvil. Así que cualquier trabajo realizado para cambiar el software después de que esté en operación es considerado trabajo de mantenimiento.

El propósito es preservar el valor del software sobre el tiempo. El valor puede ser mejorado ampliando la base de clientes, cumpliendo requisitos adicionales, siendo cada vez más fácil de usar, más eficiente y empleando más nuevas tecnología.

Los objetivos fundamentales de la etapa de mantenimiento son:

- a. Hacer cambios predicables en los programas existentes para corregir errores que se cometieron durante el diseño y la implantación del Sistema.

- b. Preservar aquellos aspectos de los programas que fueron ya corregidos. Al contrario, intentaremos evitar la posibilidad de que los arreglos en dichos programas originen que otros aspectos de los mismos funcionen de modo diferente.



CAPÍTULO III

DESARROLLO DE LA APLICACIÓN

3.1 Planificación

Para el Desarrollo de la metodología ágil se aplicará la utilización de formatos y procedimientos con el objetivo de la automatización efectiva del mecanismo de detección automática de alumnos presentes en la Facultad de Ingeniería de la Universidad de San Martín de Porres.

Este proyecto está orientado a implementar un aplicativo móvil, mediante el cual reconocerá proactivamente al alumno, de manera de llevar un control de asistencia automatizado, sin necesidad de que éste intervenga. A continuación, se detallará cada fase y/o etapa:

Tabla N°17 : Herramientas colaborativas usadas en la fase de planificación

Herramientas utilizadas en esta etapa	
Microsoft Word 2013	Para documentar
Microsoft Excel 2013	Para cálculos y análisis de data
Formulario de google	Para hacer encuestas online
Cacoo	Herramienta de dibujo en línea para diagramas, prototipos etc.

Fuente: Propia

Mediante los requisitos del usuario identificamos las necesidades, problema y objetivos del software, también se organizó un plan de actividades en donde se definió el tiempo de desarrollo.

Una vez estudiado el problema se realizó un análisis de las necesidades y se culminó con esta fase de la formulación del plan se llevó a cabo las siguientes fases. En la etapa de Planificación se usó los siguientes formatos y procedimientos:

3.1.1 Ingeniería de requisitos

A. Entrevistas Pasos que se realizó para la ejecución de las entrevistas:

- a. Lectura de antecedentes.
- b. Definición de nuestros objetivos.
- c. Seleccionamos nuestros invitados.
- d. Selección del tipo y estructura de preguntas.

Tabla N°18 : Levantamiento de Información

Tiempo que demora del control de asistencia de los alumnos en las aulas de la USMP - FIA					
N* alumnos	¿Se tomó asistencia?	Profesor	Curso	Tiempo de asistencia	Comentario
90	si	Li Perez	Contabilidad	5'15"	
63	si	Alberto Reyes	Estadística y Probabilidad I	1'45"	Tomó al azar
30	si	Quinto Cáceres	Programación I	2'18"	
20	si	Aparicio Yrala Víctor	Taller de Proyectos	10'23" + 1'26"	Se calcula el tiempo que demora que se pase la hoja de asistencia más el tiempo que demora en registrarlo en el sistema
27	Si	Bernuy Alva	Gestión del conocimiento	1' 33"	
34	si	Henriquez Taboada	Pruebas de software	2'	

Fuente: Propia

Tabla N°19 : Hoja de entrevista no estructurada

Información no estructurada

Opinión sobre el sistema actual del control de asistencia en la FIA USMP

¿Considera usted que es necesario la toma de asistencia en la Universidad, por qué?

Sí, porque nos permite controlar el nivel de asistencia de nuestros alumnos, además porque es un requisito y forma parte de nuestros procesos de calidad y somos evaluados por empresas acreditadoras.

¿Qué piensa usted del sistema actual de asistencia?

Es un sistema que ha venido cumpliendo con su objetivo principal, la toma de asistencia, pero reconocemos que no se ha mejorado o evolucionado y que muchas veces el docente por no tener un control de su tiempo, no realiza este proceso.

¿En qué momento cree usted que se debe realizar la toma de asistencia?

Lo ideal siempre es a la hora de clases, pero lamentablemente es un proceso que así como está, nos muestra deficiencia y es una actividad más para la labor de nuestros docentes

Fuente: Propia

Los resultados de las encuestas los puede verificar en el Anexo 01

Tabla N°20 : Hoja de entrevista estructurada

Información estructurada;		
Opinión sobre el sistema actual del control de asistencia en la FIA - USMP		
<p>Cómo calificaría usted el proceso de toma de asistencia que se ejecuta en la Facultad de Ingeniería y Arquitectura.</p>	<p>A. Bueno B. Regular C. Malo.</p>	<p>x</p>
<p>Si consideras que el sistema actual tiene deficiencias, cuáles serían esas deficiencias:</p>	<p>A. No se escucha bien al docente B. Resta mucho tiempo a la hora de clase C. Probabilidad que el docente marque equivocadamente.</p>	<p>x</p>
<p>Le gustaría contar con un sistema de control de asistencia automatizado:</p>	<p>A. Si B. No</p>	<p>x</p>

Fuente: Propia

Tabla N°21 : Hoja de entrevista estructurada

INFORMACIÓN ESTRUCTURADA; OPINIÓN DE LOS USUARIOS SOBRE EL USO DE SMARTPHONE EN LA FIA - USMP		
¿Posee usted un teléfono inteligente? (Smartphone)	A. Sí	x
	B. No	
En caso de poseer un teléfono inteligente. Con qué tipo de sistema operativo cuenta.	A. Android	x
	B. Windows Phone	
	C. IOS	
	D. BlackBerry OS	
¿Cómo le gustaría que sea un nuevo mecanismo de control de asistencia?	A. Rápido	x
	B. Confiable	
	C. Automático.	

Fuente: Propia

B. Lluvia de Ideas,

La aplicación de este método se ejecuta en tres fases:

Se presenta el problema que se va a tratar los problemas tienen que ser reales y conocidos previamente ya que el objetivo no es estudiar y analizar problemas sino producir ideas para su solución.

Se presentan los objetivos y reglas de la técnica al grupo, los usuarios tienen que aportar libremente, durante un tiempo determinado, todas las ideas y sugerencias que se les ocurran sin analizar si son válidas, correctas, posibles o adecuadas, y sin miedo al ridículo.

- **Desarrollo (producción de ideas)** El grupo durante un periodo de tiempo entre 20 y 30 minutos genera ideas libremente.

El formador hace de coordinador y estimula la producción de ideas, y el propio formador o un ayudante toma nota de las ideas en una lista visible; esto sirve de refuerzo y motivación para la participación de todos.

- **Evaluación de las ideas** Se redactan y clasifican las ideas surgidas. A continuación, se fijan los criterios para seleccionar las ideas mejores y se valora cada idea de acuerdo con los criterios establecidos.

Se eliminan las ideas no válidas y, a partir de una lista breve de ideas válidas, el grupo elige la mejor solución.

Tabla N°22 : Formato de Lluvia de ideas

Obtención de ideas para mejorar el proceso de control de asistencia en la USMP - FIA	
N°	Descripción de Ideas
1	Un control de asistencia que sea rápido e inteligente.
2	Que el docente no pueda caer en error de marcación.
3	Que el control de asistencia se tome a través de huella digital.
4	Que nos permita saber que alumno sale o entra a clases.
5	El control de asistencia no lo debe realizar el docente.
6	Que el aplicativo sea móvil y se conecte al sistema de la Universidad.

Fuente: Propia

Tabla N°23 : Formato de identificación de mejora de ideas

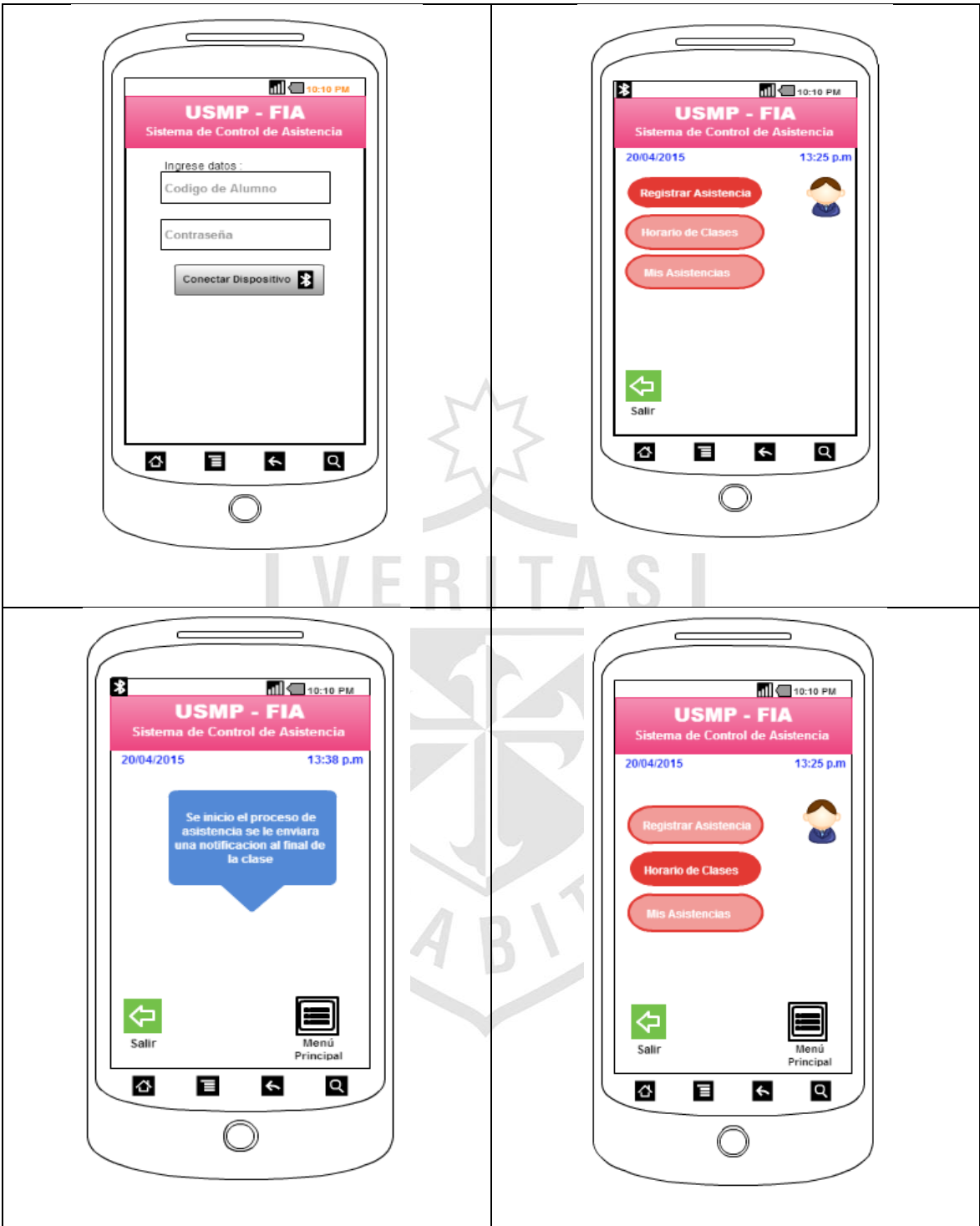
N°	IDENTIFICACIÓN DE LA MEJORA DE IDEAS									
	IDEA -		IDEA -		IDEA -		IDEA -		IDEA -	
	01	02	03	04	05	SÍ	NO	SÍ	NO	
Preguntas Orientadas - Ideas	SÍ	NO	SÍ	NO	SÍ	NO	SÍ	NO	SÍ	NO
Debe evitar errores en el usuario			X							
Conocen de otros sistemas				X						
Desarrollar un aplicativo móvil										X
Debe ser automatizado	X									
Obtener mayor control							X			

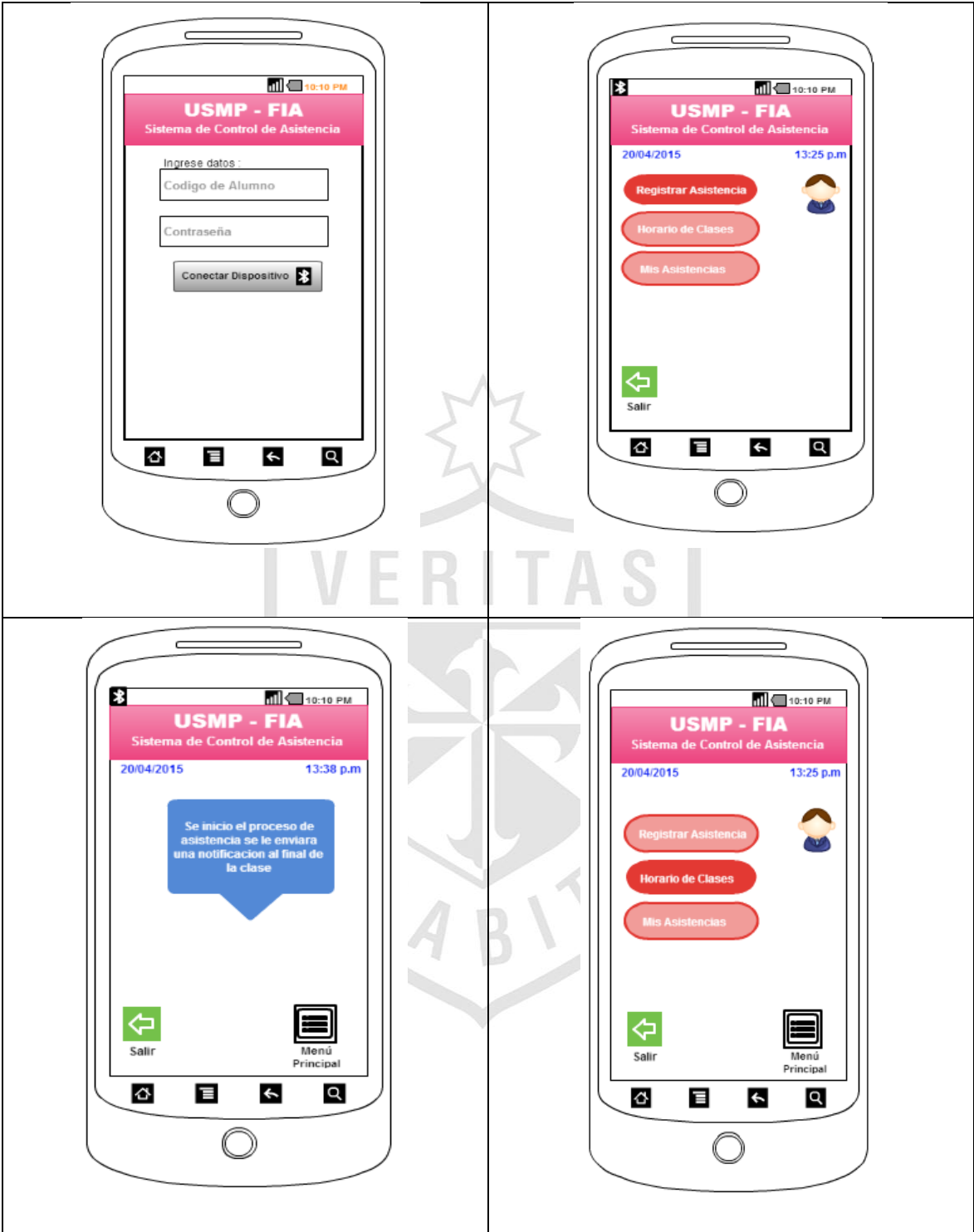
Fuente: Propia

C. Prototipos

Se debe de utilizar la versión preliminar de prototipos para aplicaciones móviles, reconociendo el principio emperico del usuario y del cliente.

Para la realización de estos prototipos se utilizará una excelente herramienta gratuita Invisión, la cual nos permite de forma colaborativa intercambiar opiniones entre los diferentes miembros del equipo.





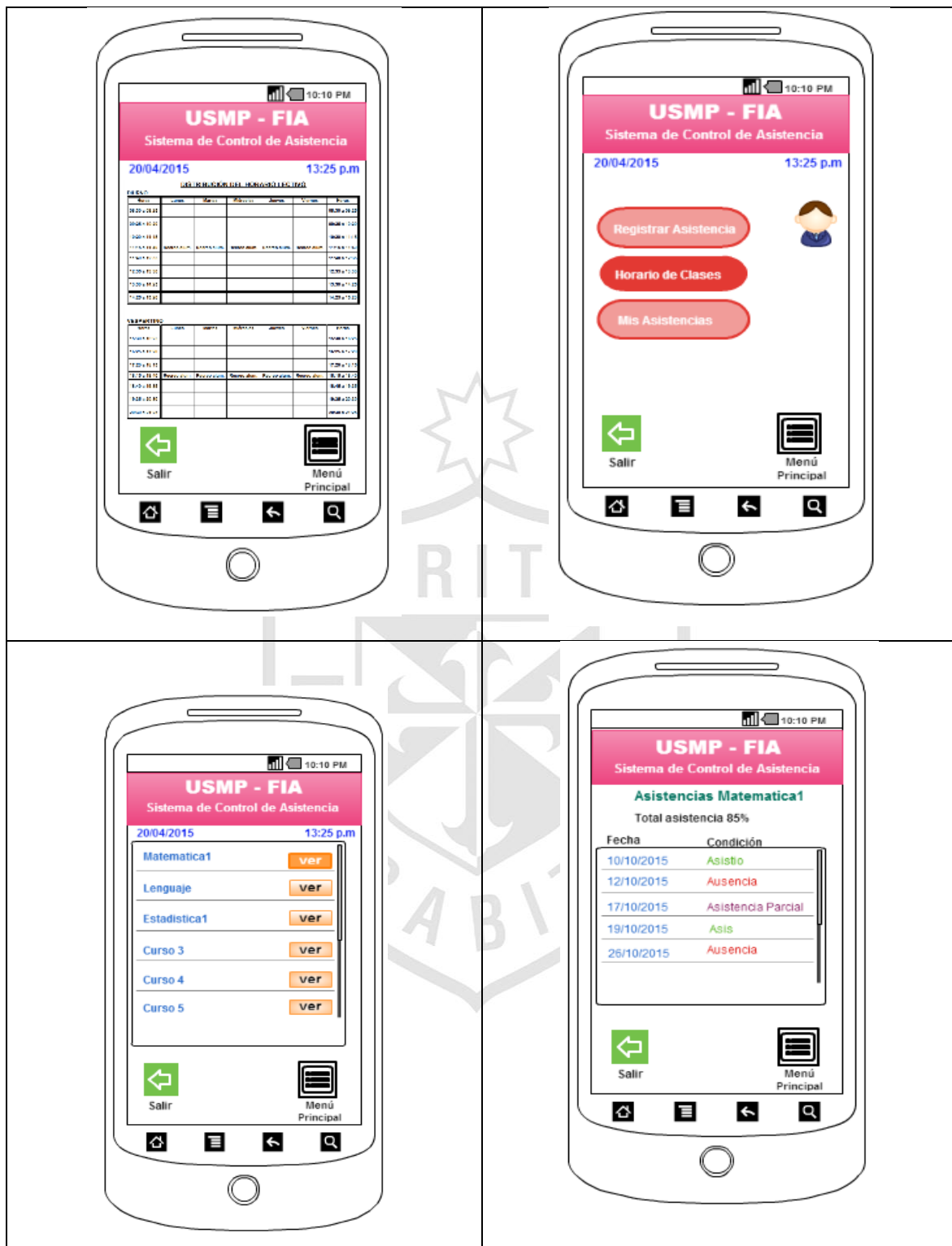


Ilustración 19 : Prototipos 01 de la aplicación
Fuente: Propia

D. Requisitos del sistema (funcional y no funcional)

Para la obtención de los requisitos del sistema primero se optó por identificar las reglas del negocio.

Tabla N°24 : Captura de requerimientos

USMP - FIA

Captura y descripción de requerimientos



Proyecto: Detección de alumnos mediante mecanismos proactivos, para el control de asistencia para la Facultad de Ingeniería y Arquitectura de la USMP

Fecha:

22 de Marzo del 2015

Área:

Coordinación Académica

Usuario:

Ing. Julio Agüero Martínez La Rosa

Cargo del usuario:

Coordinador turno noche

Descripción del Requerimiento:

La aplicación debe integrarse con el Sistema de Gestión Académica para obtener los datos de los alumnos y relacionarlos con sus horarios respectivos, de tal forma que la aplicación reconozca la hora y lugar donde se está controlando la asistencia.

Fuente: Propia

Tabla N°25 : Captura de requerimientos

USMP - FIA
Captura y descripción de requerimientos



Proyecto: Detección de alumnos mediante mecanismos proactivos, para el control de asistencia para la Facultad de Ingeniería y Arquitectura de la USMP

Fecha: 22 de Marzo del 2015
Área: Coordinación Académica
Usuario: Ing. José Cárdenas Martínez
Cargo del usuario: Coordinador turno mañana

Descripción del Requerimiento:

Se deberá considerar a un alumno como ausente cuando el alumno tenga menos de 3 hitos registrados y como asistencia completa cuando tenga al menos 4 tomando como referencia la duración de un curso de 1.5 horas.

Se emitirá un reporte en el cual se pueda apreciar el rango de asistencia efectiva de los alumnos.

Fuente: Propia

• **Requerimientos funcionales**

Tabla N°26 : Identificación de los requerimientos funcionales

RF	La aplicación debe almacenar información sobre las asistencias de los alumnos
	La aplicación debe registrar automáticamente la asistencia del alumno.
	La aplicación debe mostrar al alumno un mensaje de registro de asistencia satisfactorio.
	La aplicación debe permitir a los usuarios en simultáneo buscar y consultar la información sobre sus horarios y asistencias.
	La aplicación debe mostrar los horarios de los cursos matriculados

Fuente: Propia

- **Requerimientos no funcionales**

Tabla N°27 : Identificación de los requerimientos no funcionales

RNF	La aplicación no deberá revelar al personal docente ninguna información de los usuarios del sistema aparte de su nombre apellido y código de alumno.
	La aplicación debe visualizarse y funcionar correctamente en cualquier navegador, especialmente en Internet Explorer, Mozilla.
	La aplicación debe visualizarse y funcionar correctamente en cualquier tipo de sistema operativo móvil.
	La aplicación deberá soportar una conexión de hasta 4000 usuarios en línea.
	La aplicación no debe tardar más de cinco segundos en mostrar los resultados de una búsqueda.
	La aplicación no debe tardar más de cinco segundos en mostrar los resultados de una asistencia.
	La aplicación deberá estar disponible entre las 08:00 am hasta las 10:15 pm
	La aplicación debe integrarse con el sistema actual de la universidad
	La aplicación debe ser presentada al usuario en el idioma de preferencia. Ha de dar soporte al español, al inglés

Fuente: Propia

3.1.2 Asignación de roles:

Se asignó los siguientes roles.

a) Jefe de proyecto: Funciones:

- Gestiona y asigna recursos humanos y de otro tipo.
- Define las prioridades de las tareas relacionadas con el proyecto.
- Coordina las interacciones con los clientes y los usuarios finales.
- Planifica y asigna las tareas de la forma más razonable posible.
- Define la organización y estructura del proyecto.
- Motiva y organiza el equipo de trabajo para lograr un objetivo definido.
- Establece los horarios de trabajo del equipo de desarrollo.
- Planifica y realiza las reuniones de control del equipo de desarrollo.
- Mantiene el control del resultado de estas reuniones
- Informa al grupo sobre el estado actual del proyecto.

b) Líder de desarrollo: Funciones:

- Guía al equipo de desarrollo en una estrategia de desarrollo.
- Establece las estrategias según el tamaño de la tarea y el tiempo disponible, según estimaciones.
- Conduce el desarrollo del producto hacia los requerimientos.
- Guía al equipo en el más alto nivel de diseño.
- Guía al equipo en la especificación del diseño del software.
- Guía al equipo en las pruebas del sistema.
- Guía al equipo en la producción de la documentación del usuario.
- Participa en la producción del reporte de desarrollo.

c) Administrador de configuración: Funciones:

- Define y supervisa el proceso de control de cambios.
- Responsable de proporcionar al equipo de desarrollo la infraestructura general, de gestión de cambios y ambiente de trabajo.
- Establece, controla e informa sobre el control de cambios.

- Responsable del funcionamiento correcto de los servidores y las estaciones de trabajo durante el desarrollo y pruebas del software.
- Responsable de las salvas del proyecto y de la base de datos.
- Responsable de la correcta instalación y configuración del producto en el lugar de destino, así como de los reporte de configuración, herramienta de control de configuración (Source Safe).

d) Especificador de requerimientos: Funciones:

- Lidera al equipo que realiza la captación de requerimientos.
- Se encarga de especificar los detalles de una o varias partes de la funcionalidad del sistema, describiendo uno o varios aspectos de los requisitos.

e) Analista del sistema: Funciones:

- Dirige y coordina el proceso de extracción de requisitos y desarrollo del modelo de casos de uso, definiendo las funcionalidades y límites del sistema.
- Responsable del diseño del sistema, dentro de los límites de: Los requisitos, la arquitectura, y el proceso de desarrollo del proyecto.

f) Diseñador de la interfaz de usuario: Funciones:

- Coordina el diseño de la interfaz de usuario, utiliza los requisitos de uso y crea prototipos candidatos de interfaz de usuario de acuerdo a ellos.
- Encargado de realizar el trabajo artístico que requiera el proyecto (íconos, pantalla de splash, gráficos, etc.)

3.1.3 Cronograma de actividades

Pasos para llevar a cabo la ejecución del cronograma:









- a) Mostrar la lista de actividades en columna
- b) Disponer el tiempo adecuado para el proyecto
- c) Calcular el tiempo para cada actividad

- d) Indicar esos tiempos en forma de barras horizontales
- e) Reordenar cronológicamente
- f) Ajustar el tiempo o secuencia de actividades.

Diagrama de Gantt

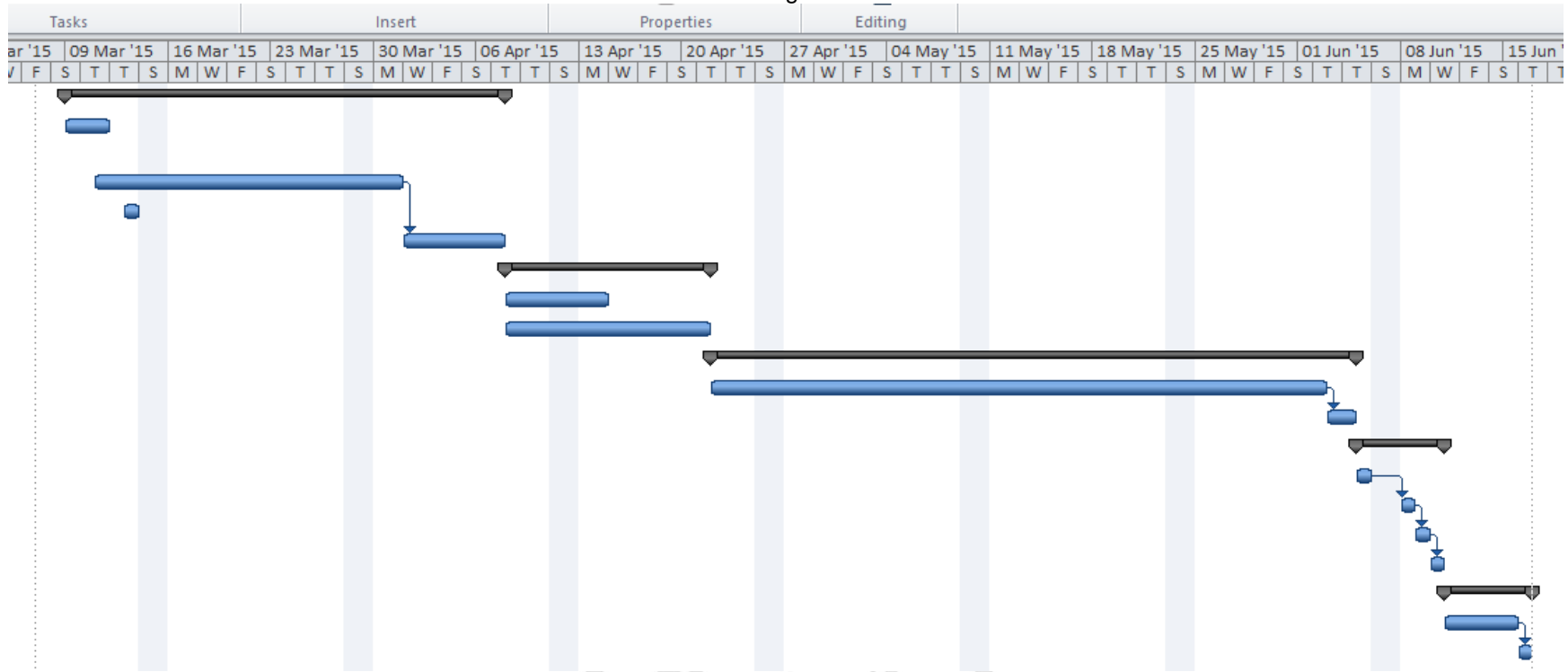
Para el desarrollo del cronograma de actividades se utilizará el diagrama de Gantt, este diagrama nos permite realizar el seguimiento del proyecto en la cual tenemos integradas las diferentes actividades que se realizan en secuencia ordenada, además nos servirá como un adecuado instrumento de comunicación con los usuarios finales.

Tabla N°28 : Cronograma de actividades.

		Task Name	Duration	Start	Finish	Predecessor
1		<input type="checkbox"/> Planificación	22 days	Mon 09/03/15	Tue 07/04/15	
2		Cronograma de actividades	3 days	Mon 09/03/15	Wed 11/03/15	
3		Ingeniería de requisitos	3 wks	Wed 11/03/15	Tue 31/03/15	
4		Asignación de roles	1 day	Fri 13/03/15	Fri 13/03/15	
5		Historias de usuario	1 wk	Wed 01/04/15	Tue 07/04/15	3
6		<input type="checkbox"/> Diseño	10 days	Wed 08/04/15	Tue 21/04/15	
7		Diseño de prototipos	1 wk	Wed 08/04/15	Tue 14/04/15	
8		Arquitectura	2 wks	Wed 08/04/15	Tue 21/04/15	
9		<input type="checkbox"/> Desarrollo	32 days	Wed 22/04/15	Thu 04/06/15	
10		Codificar	1.5 mons	Wed 22/04/15	Tue 02/06/15	
11		Versionar	2 days	Wed 03/06/15	Thu 04/06/15	10
12		<input type="checkbox"/> Pruebas	4 days?	Fri 05/06/15	Wed 10/06/15	
13		Pruebas unitarias	1 day?	Fri 05/06/15	Fri 05/06/15	
14		Pruebas de aceptación	1 day?	Mon 08/06/15	Mon 08/06/15	13
15		Pruebas de estress	1 day?	Tue 09/06/15	Tue 09/06/15	14
16		Implementación	1 day?	Wed 10/06/15	Wed 10/06/15	15
17		<input type="checkbox"/> Mantenimiento	4 days	Thu 11/06/15	Tue 16/06/15	
18		Correccion de errores	3 days	Thu 11/06/15	Mon 15/06/15	
19		Nuevos requerimientos	1 day	Tue 16/06/15	Tue 16/06/15	18

Fuente: Propia

Tabla N°29 : Cronograma de actividades



Fuente: Propia

3.2 Diseño

Tabla N°30 : Herramientas colaborativas usadas en la fase de diseño

Herramientas utilizadas en esta etapa	
Microsoft Word 2013	Para documentar
Navicat Premium 2010	Administrador de base de datos
Android@Screen	Programa que permite visualizar la pantalla de un dispositivo móvil en una computadora.
Cacoo	Herramienta de dibujo en línea para diagramas, prototipos etc.

Fuente: Propia

En esta fase, se estructuró los contenidos obtenidos en la fase de planeación y se determinó los requisitos que serán desarrollados en el software; así mismo, se elaboró un mapa de navegación para el software, el cual permite dividir en partes, cada parte dará como resultado un módulo.

La cantidad y tamaño de los módulos dependerá del caso de uso. En esta fase se desarrolló tres subtarear: historias de usuario, diseño de prototipos y arquitecturas del software.

3.2.1 Historia de Usuario

Las historias de usuario ayudarán a tener una comprensión de lo que serán los requisitos funcionales. A continuación se presenta los más importantes.

Tabla N°31 : Historia de usuario N° 001

Historia de Usuario N° 001	
Número:001	Usuario: Alumno
Nombre historia: Registrar alumno	
Prioridad: Baja	Riesgo en desarrollo: Bajo
Programador responsable: Alva Carlos	
Descripción: El alumno descarga la aplicación desde el repositorio oficial, La aplicación se solicita su código de alumno y contraseña. El alumno ingresa los datos correctamente y la aplicación muestra un mensaje de “registro satisfactorio”.	

Fuente: Propia

Tabla N°32 : Historia de usuario N° 002

Historia de Usuario N° 002	
Número:002	Usuario: Alumno
Nombre Historia: Registrar asistencia	
Prioridad: Alta	Riesgo en desarrollo: Alto
Programador responsable: Alva Carlos	
Descripción: El alumno al ingresar al aula ingresa a la aplicación la cual solicita encender el Bluetooth si es que no lo tiene encendido, así como conectarse a la red wi-fi de la Universidad. La aplicación escanea en background cada 15 minutos la presencia de Beacons; al encontrarlo registra la fecha y hora. El alumno al acumular 4 registros la aplicación envía una notificación PUSH informando que la asistencia ha sido registrada exitosamente.	

Fuente: Propia

3.2.2 Diseño de prototipos

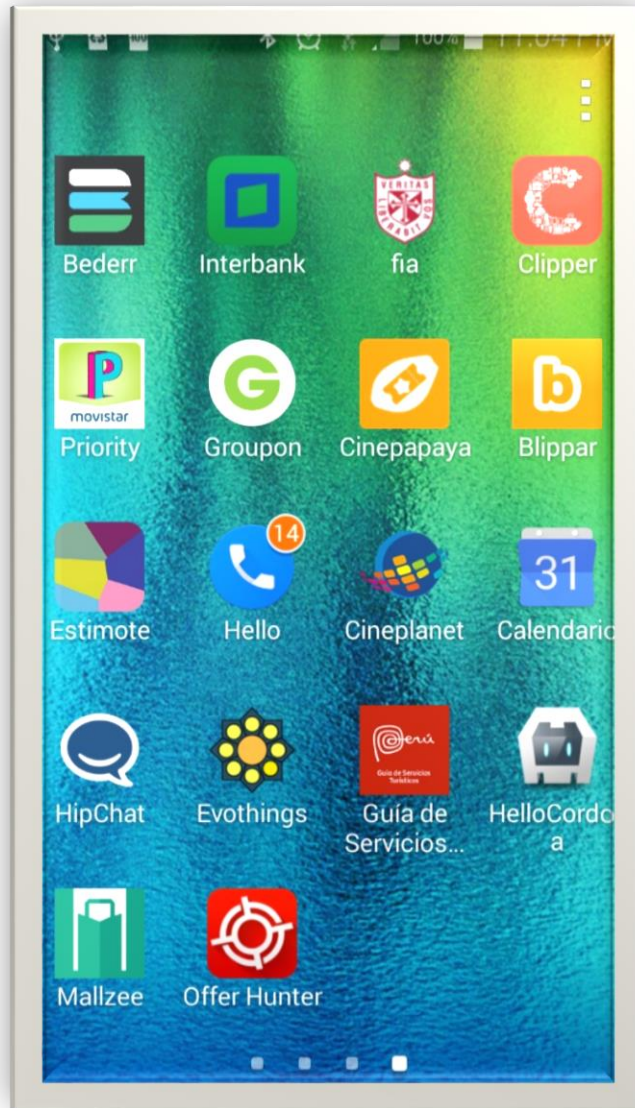


Ilustración 20 : Icono de aplicación
Fuente: Propia

En la ilustración anterior se aprecia el icono de la aplicación que el alumno deberá presionar para poder acceder.



Ilustración 21 : Autenticación de estudiante
Fuente: Propia

En la ilustración anterior el alumno después de ingresar a la aplicación esta le pedirá que se autentique con su código de alumno y contraseña. Serán los mismos que usa para ingresar a la intranet de la universidad.

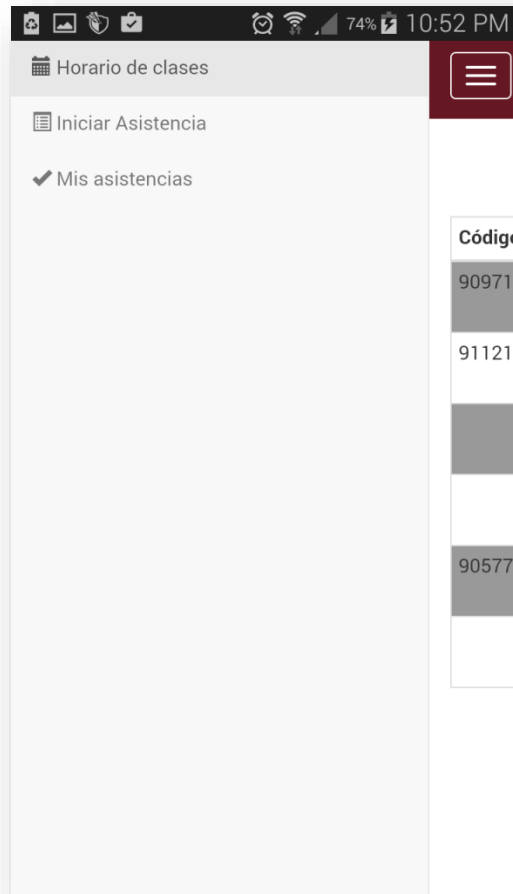


Ilustración 22 : Menú lateral
Fuente: Propia

En la ilustración anterior apreciamos el menú lateral de la aplicación que despliega las opciones de “Horario de clases”, “Iniciar asistencia” y “Mis asistencias”.



Ilustración 23 : Horario de clases
Fuente: Propia

En la ilustración anterior el alumno al presionar la opción “Horario de clases” la aplicación mostrará el horario de clases en el cual está matriculado.



Ilustración 24 : Asistencias
Fuente: Propia

En la ilustración anterior el alumno al presionar la opción “Mis asistencias” la aplicación mostrará el listado de asistencias por cada curso de su horario poniendo un check cuando el alumno asistió y una x cuando faltó.

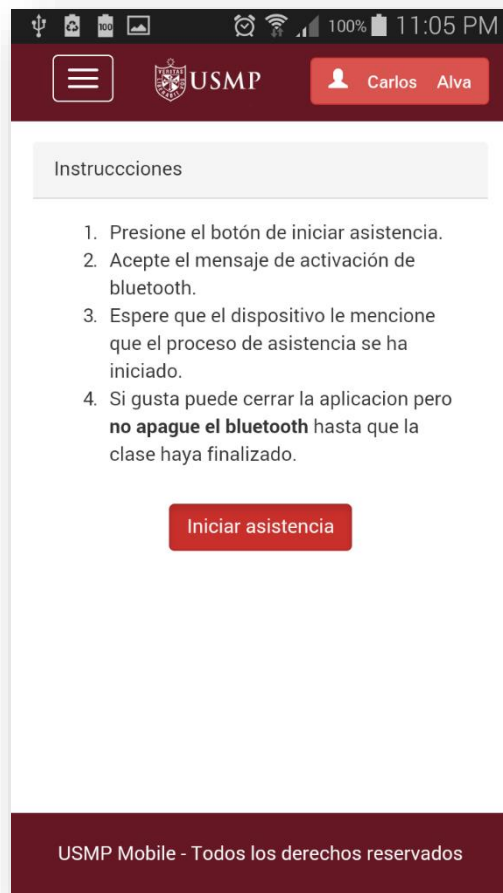


Ilustración 25 : Instrucciones
Fuente: Propia

En la ilustración anterior el alumno al presionar la opción “Iniciar asistencia” la aplicación mostrará la interface de inicio de asistencia con el botón “Iniciar asistencia” y los instructivos a seguir.

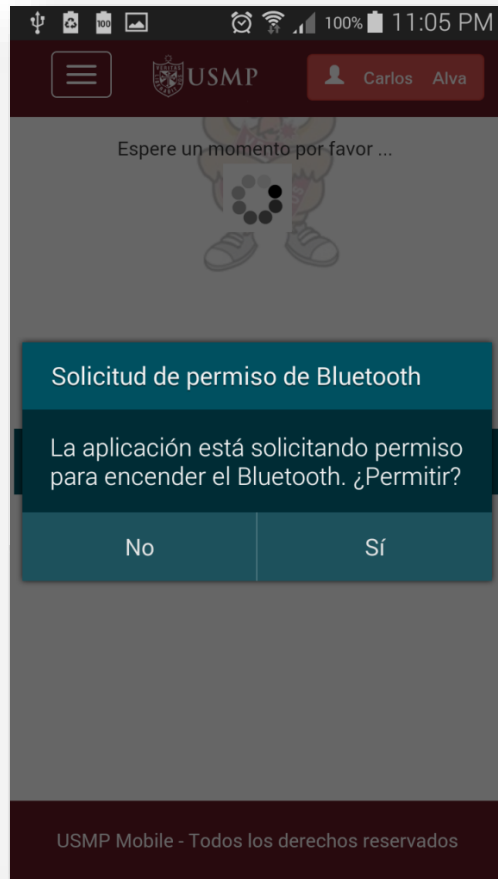


Ilustración 26 : Activar Bluetooth
Fuente: Propia

En la ilustración anterior el alumno al presionar el botón “Iniciar asistencia” la aplicación validará si esta encendido el Bluetooth, en el caso que no lo este, esta le mostrará un mensaje de activación.



Ilustración 27 : Inicio proceso asistencia
Fuente: Propia

De la ilustración anterior si el Bluetooth está encendido, la aplicación comienza a escanear periódicamente la presencia de dispositivos Beacons.

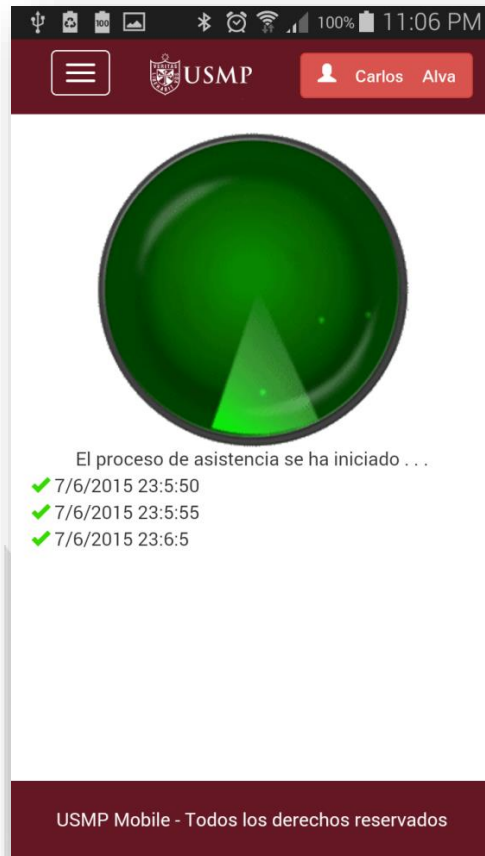


Ilustración 28 : Registro de hitos
Fuente: Propia

De la ilustración anterior al detectar la presencia de dispositivos Beacons y estos estar asociados al horario correspondiente al alumno, la aplicación calcula el tiempo total de la clase dividiéndola en hitos de 15 minutos, suponiendo que es una clase de 1.30 horas entonces serán 6 hitos posibles.

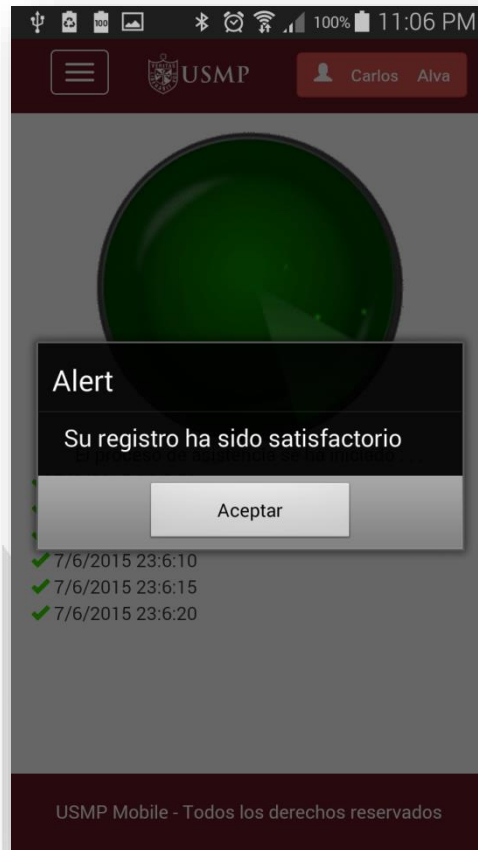


Ilustración 29 : Mensaje registro satisfactorio
Fuente: Propia

Finalmente, de la ilustración anterior una vez transcurrido el tiempo de la clase y haber registrado cada hito de tiempo, la aplicación muestra un mensaje indicando que el registro fue satisfactorio.

3.2.3 Arquitectura de software (AS)

En la Etapa de diseño, hemos desarrollado nuestra arquitectura del sistema propuesto, diseño de prototipos, diagrama de actividad, base del proceso y registro de control de asistencia.

La arquitectura planteada se divide en 3 paquetes (Aplicación, Apache Cordova y USMP).

El paquete Aplicación contiene

- La interface de usuario, la cual interactúan componentes HTML5, CSS3 y JavaScript, es a través de esta capa con la cual la aplicación interactúa con el usuario.
- La capa lógica del negocio, la cual está escrita netamente en JavaScript en la cual esta las funcionalidades de la aplicación y se comunica con el API de Apache Cordova para interactuar con los componentes electrónicos del celular.

El paquete Apache Cordova contiene

- La capa API, la cual está escrita en código nativo por cada sistema operativo y JavaScript que permite la comunicación a través de un lenguaje común (JavaScript).
- La capa Plugins, la cual se encuentran soluciones de terceros escritos en lenguajes nativos y JavaScript que son fácilmente acoplables para su uso inmediato.

El paquete USMP contiene

- La capa RestFul, que permite la comunicación entre la aplicación y el Sistema de Gestión Académica para obtener la información de los alumnos.

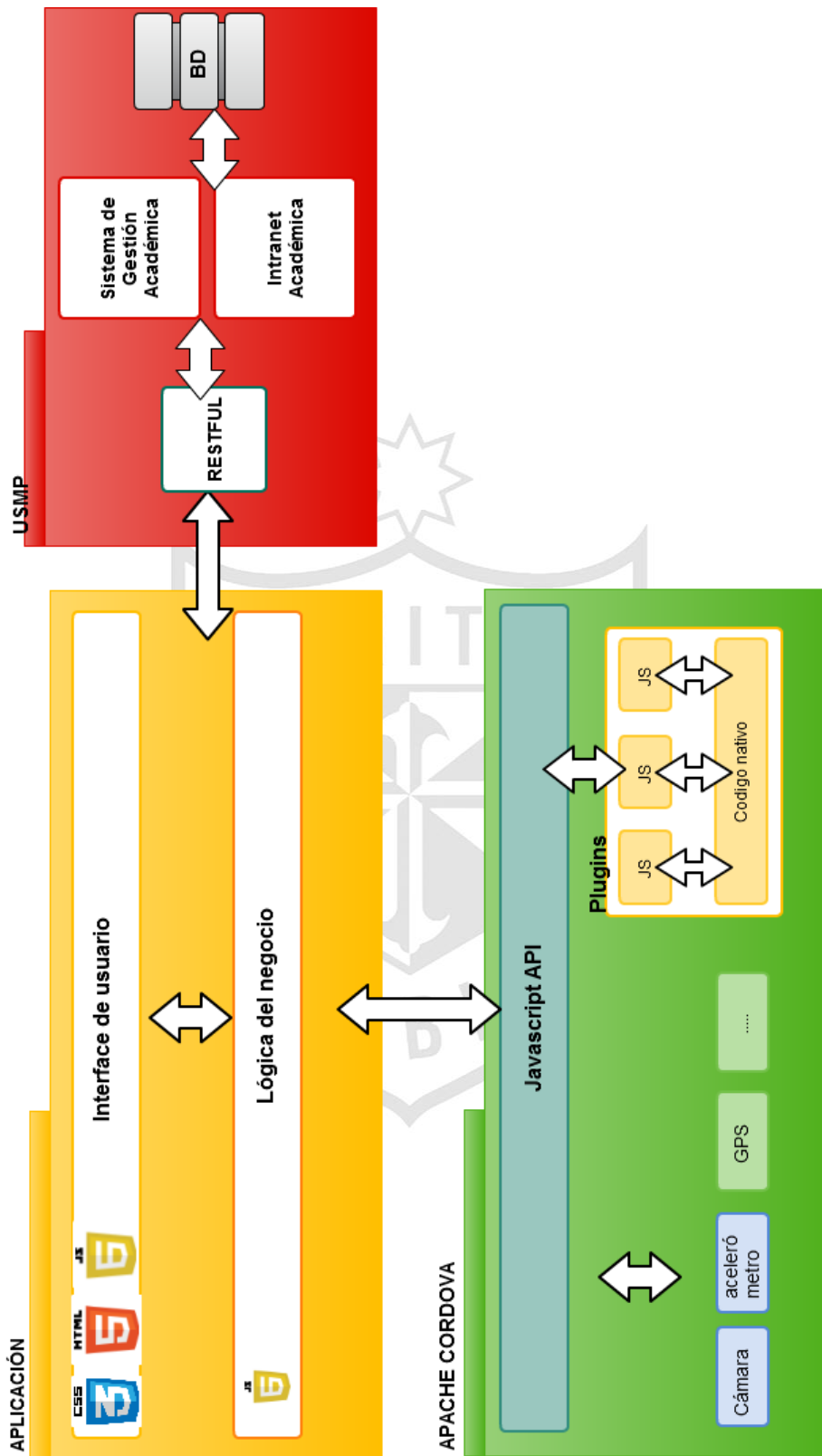


Ilustración 30 : Arquitectura del sistema
Fuente: Propia

Arquitectura de distribución de los dispositivos: Beacon

Para la optimización del reconocimiento de dispositivos Beacon se plantea una distribución forma triangular en el aula de esa forma se asegura que se cubra toda el área del aula.

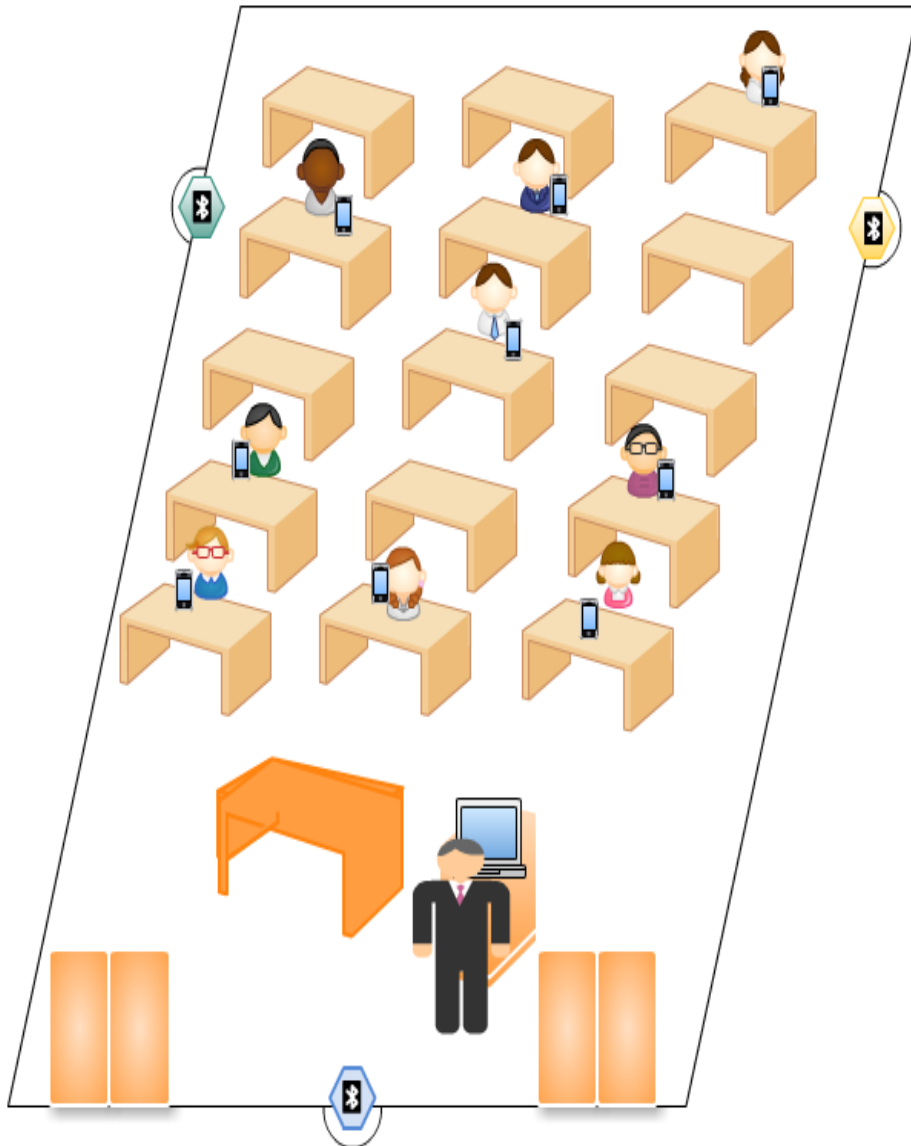


Ilustración 31 : Ubicación de Dispositivos Beacons
Fuente: Propia.

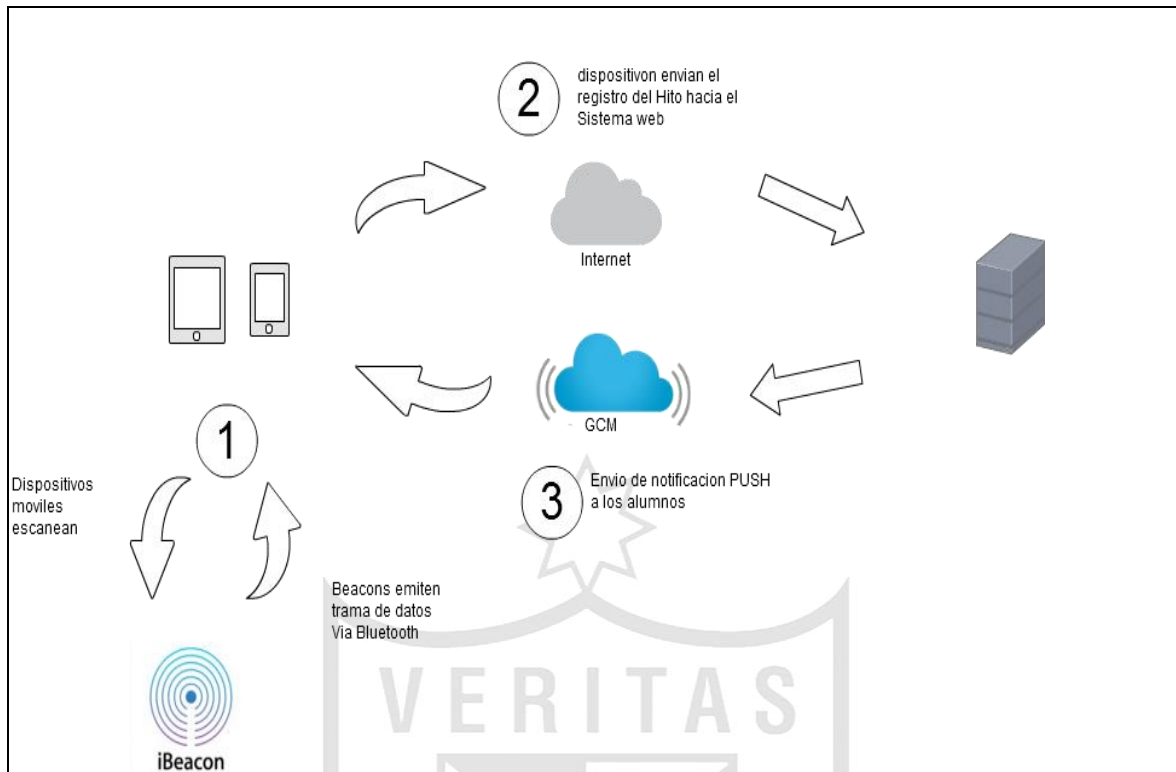


Ilustración 32 : Arquitectura física
Fuente: Propia

Podemos apreciar en la ilustración anterior primero que los dispositivos escanean la presencia de Beacons, segundo si encuentran el dispositivo registrar el hito enviándolo a la base de datos, tercero una vez alcanzado el numero requerido de hitos se le envía una notificación de registro satisfactorio

3.2.4 Diagrama de clase

En el diagrama de clases podemos ver 3 clases principales la clase Dispositivo que contiene los datos del dispositivo móvil asociado al alumno, así como sus métodos de registro y consulta.

La clase Hitos, la cual contiene los atributos de propios de un hito de registro al momento de detectar la presencia de un dispositivo Beacon, y sus métodos de registro y consulta.

Finalmente la clase RestFul, que permitirá la comunicación entre los dispositivos y el Sistema de Gestión Académica el cual se comunica con la base de datos. Tiene métodos de registrar alumno, validación del alumno, buscar alumno y enviar notificación PUSH.

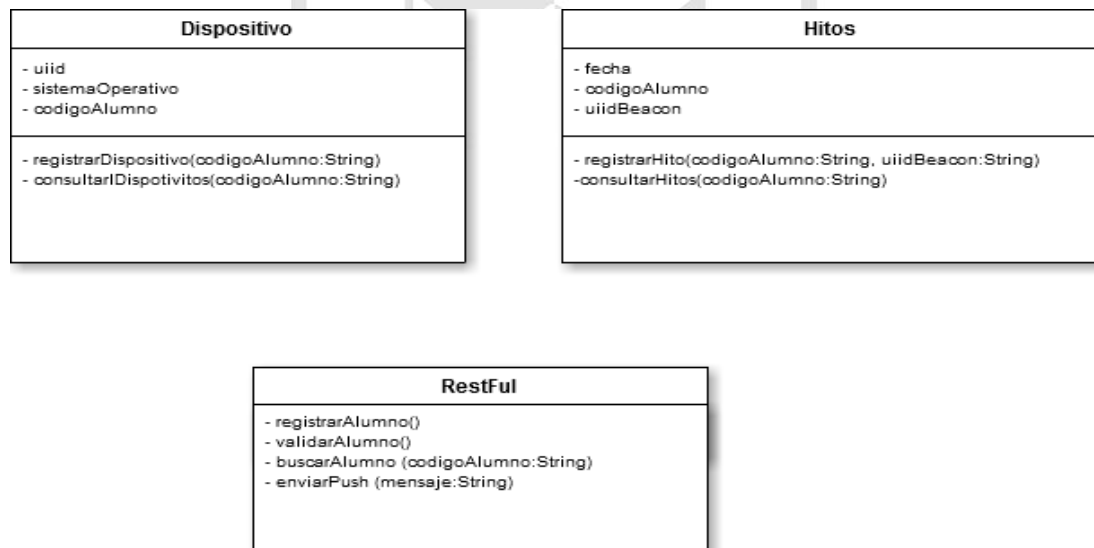
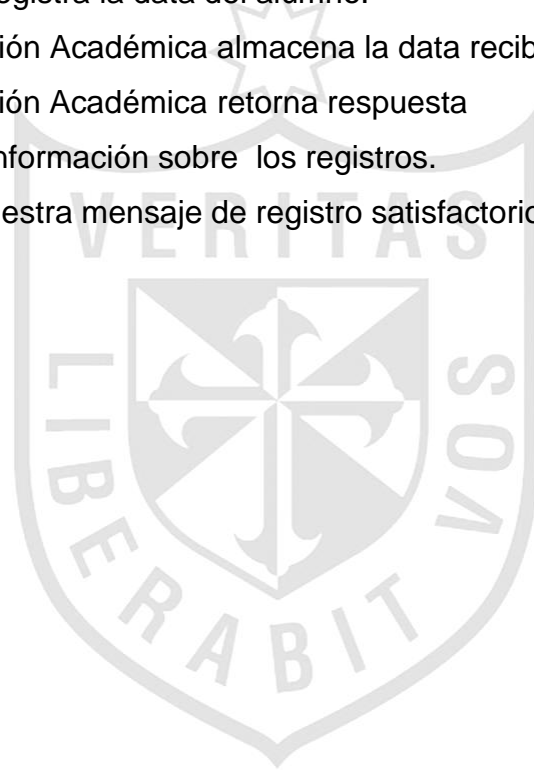


Ilustración 33 : Diagrama de clases
Fuente: Propia

3.2.5 Diagrama de actividad registrar asistencia

En la siguiente ilustración vemos el diagrama de actividad el flujo básico del registro de asistencia.

1. El alumno selecciona la opción “registrar asistencia”
2. La aplicación escanea cada 15 minutos la presencia de dispositivos Beacons.
3. La aplicación envía la data del alumno al Resful Fia.
4. El servicio web registra la data del alumno.
5. Sistema de Gestión Académica almacena la data recibida.
6. Sistema de Gestión Académica retorna respuesta
7. RESFUL envía información sobre los registros.
8. La aplicación muestra mensaje de registro satisfactorio.



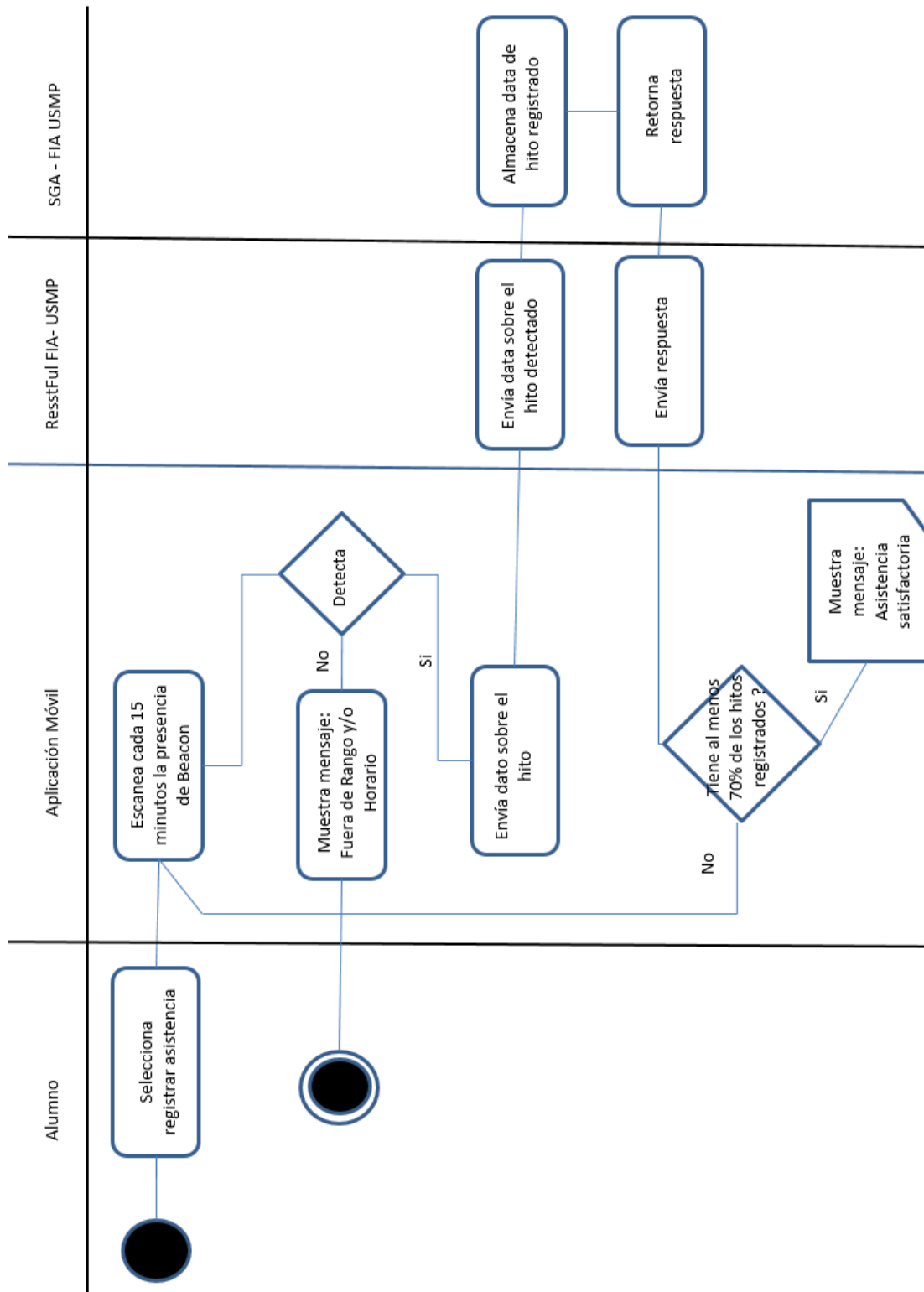


Ilustración 34 : Diagrama de actividad - registrar asistencia
Fuente: Propia

3.2.6 Diagrama de secuencia registrar asistencia:

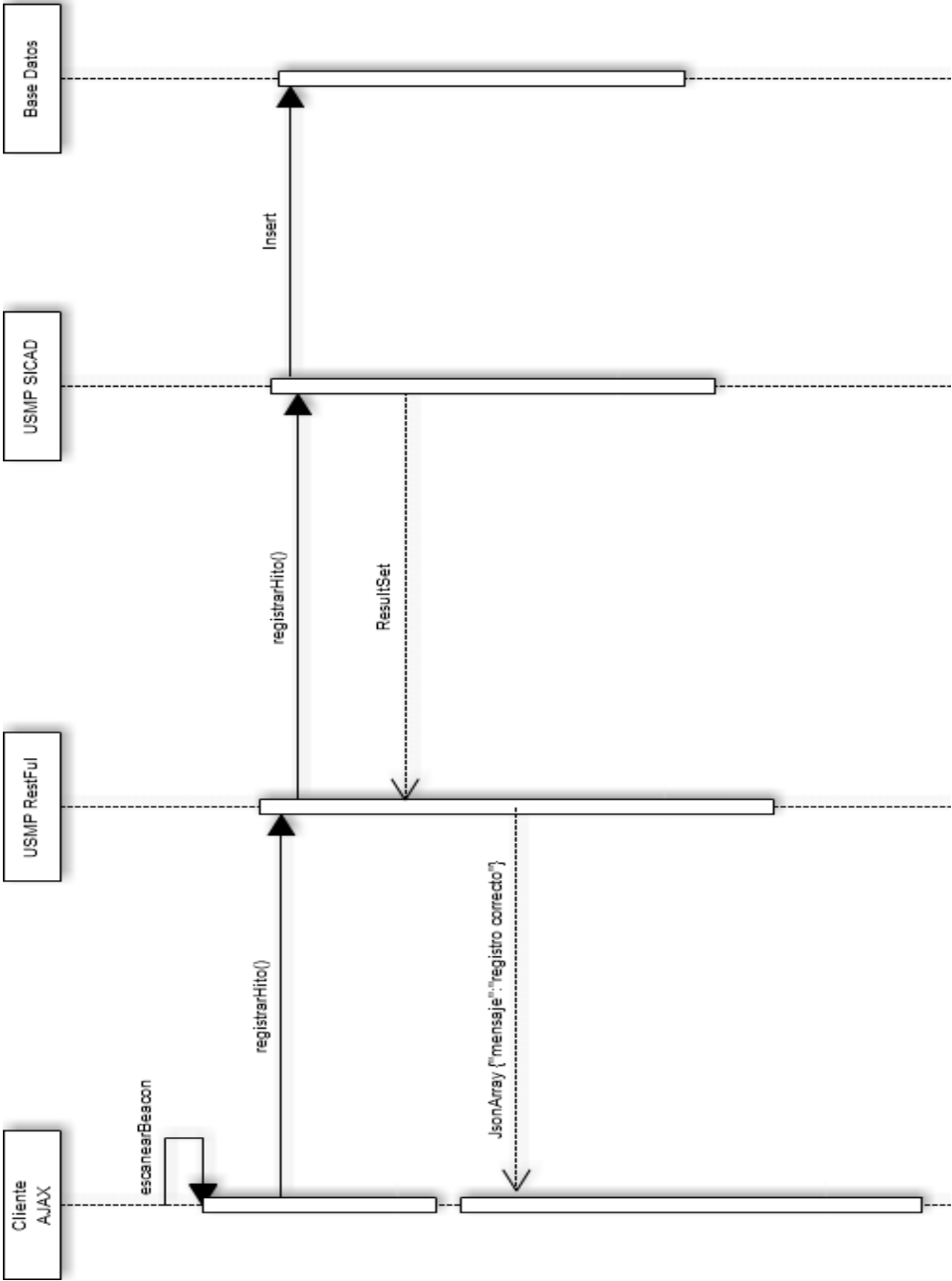


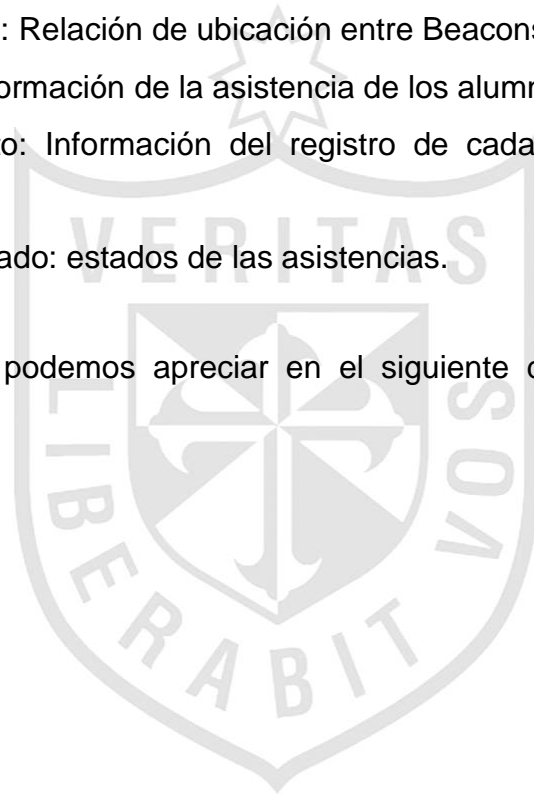
Ilustración 35 : Diagrama de secuencia - registrar asistencia
Fuente: Propia

3.2.7 Base de datos

Para la elaboración de la base de datos se definió las siguientes tablas a continuación una breve descripción de cada tabla

- Estudiante : Información del estudiante
- Curso : Información de los cursos dictados en la FIA
- Horario : Horario académico del alumno
- Salon: Información de los salones en la FIA.
- Beacon: Información de los Becons.
- Beacon_salon: Relación de ubicación entre Beacons y salones.
- asistencia: Información de la asistencia de los alumnos.
- Estudiante_hito: Información del registro de cada hito al escanear los Beacons.
- Asistencia estado: estados de las asistencias.

Las relaciones las podemos apreciar en el siguiente diagrama de base de datos.



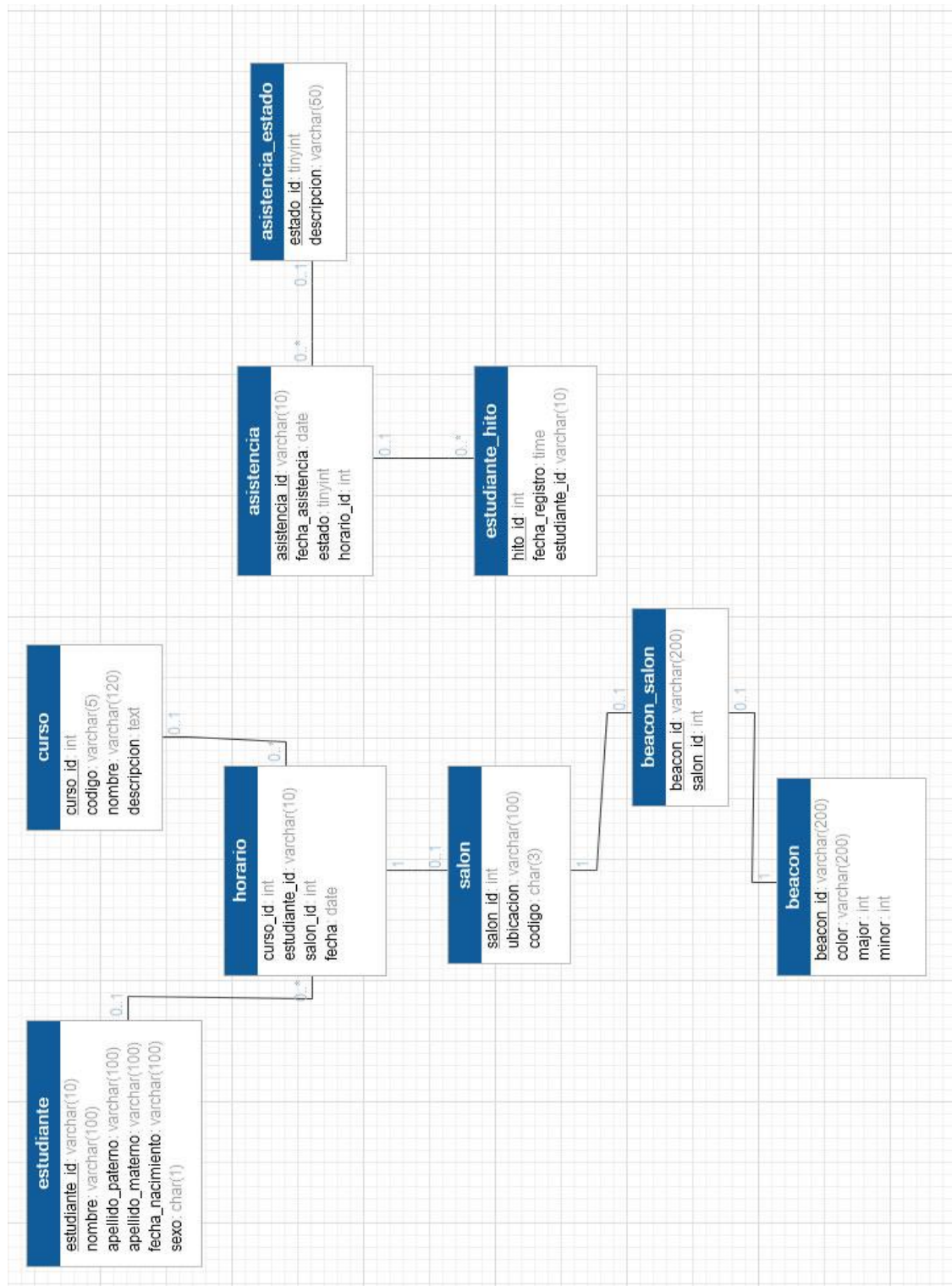


Ilustración 36 : Diseño físico de la base de datos
 Fuente: Propia (Ver anexo 02)

3.3 Desarrollo

Tabla N°33 : Herramientas colaborativas usadas en la fase de desarrollo

Herramientas utilizadas en esta etapa	
Netbeans 8.1	IDE de desarrollo
Navicat Premium 2010	Administrador de base de datos
Android@Screen	Programa que permite visualizar la pantalla de un dispositivo móvil en una computadora.
GIT	Sistema gratuito y de código abierto distribuido de control de versiones.
Evothings Studio Workbench	Herramienta de desarrollo de móvil que permite visualizar la aplicación antes de compilarla.
Apache Cordova Command-Line Interface.	Permite a través de líneas de comandos acciones como generar Nuevo Proyecto, compilar la aplicación, agregar y remover plugins entre otros.
Postman - REST Client	Herramienta online para probar servicios web

Fuente: Propia

En la Etapa de Codificación (Desarrollo de aplicación), se señala el tipo de lenguaje utilizado y las funciones utilizadas para el desarrollo de la aplicación.

En esta fase se construyó el software, empezando con la creación de la base de datos, así como la compilación del código, se integran los elementos multimediales produciéndose las distintas interfaces, se crearon y se enlazaron los elementos correspondientes.

3.3.1 Codificar

EL desarrollador a medida que va desarrollando cada funcionalidad utilizó de manera reiterativa la herramienta de versionamiento GIT. Cada cambio es guardado localmente mediante el comando `git : commit -m "mensaje"` de esta forma se puede navegar entre el historial de cambios una vez terminada la funcionalidad el desarrollador sube sus cambios locales hacia su repositorio remoto mediante el comando `"git push"`.

Seguidamente solicita que su cambio sea integrado al repositorio de desarrollo, para esto ejecuta el comando `"git request pull"`; ahora el código para al estado de "revisión de código" esperando la aprobación de un evaluador que generalmente es un miembro del equipo de desarrollo, si el código es aprobado pasara al siguiente proceso de versionamiento.

3.3.2 Versionar

El versionar el código es importante ya que así se mantiene un orden y si hay problemas en una versión se puede regresar fácilmente a una anterior que sea estable.

Una vez hecha la petición de "revisión de código" si el evaluador aprueba que el código cumple con las convenciones de calidad. El código pasa a ser integrado al repositorio de desarrollo; es decir, pasa del repositorio del desarrollador al repositorio de desarrollo para que después pueda ser evaluado por el equipo de QA, aprobación con el cliente y certificar que los requerimientos hayan sido cumplidos.

3.4 Pruebas

Tabla N°34 : Herramientas colaborativas usadas en la fase de pruebas

Herramientas utilizadas en esta etapa	
Microsoft Word 2013	Para documentación
Apache JMeter™	Aplicación Java diseñada para cargar comportamiento funcional de prueba y medir el rendimiento.
Navicat Premium 2010	Administrador de base de datos
Android@Screen	Programa que permite visualizar la pantalla de un dispositivo móvil en una computadora.
Evothings Studio Workbench	Herramienta de desarrollo de móvil que permite visualizar la aplicación antes de compilarla.
Postman - REST Client	Herramienta online para probar servicios web

Fuente: Propia

En el proceso de pruebas el equipo de QA recibe las funcionalidades que han pasado por la “revisión del código” en la fase de desarrollo. Por lo tanto, en esta etapa se harán las pruebas unitarias, de concurrencia y de aceptación del cliente lo cual permitirá una certificación de la funcionalidad implementada para posteriormente pasar finalmente al implementador y el producto pueda ser lanzado.

Cabe mencionar que si en esta etapa se encuentra algún error la funcionalidad será devuelta a la etapa de desarrollo para ser corregida y vuelva a empezar el flujo de desarrollo. Los tipos de pruebas que se efectuaron en esta etapa son:

3.4.1 Pruebas unitarias

Aquí se evalúa los requerimientos del sistema y se evalúan cada caso de uso

Tabla N°35 : Caso de prueba – registrar dispositivo

Escenario 1:	Validar que el alumno pueda asociar su dispositivo móvil a la aplicación de control de asistencia.
Entrada:	Usuario: a2008110986 Clave : 1234abcd
Salida Esperada:	El alumno queda registrado en la aplicación guardándose en la base los siguientes datos: Código de alumno, id de dispositivo, fecha de registro.
Salida Obtenida:	Se validó que el evento fue registrado satisfactoriamente.
Defectos:	Ninguno.

Fuente: Propia

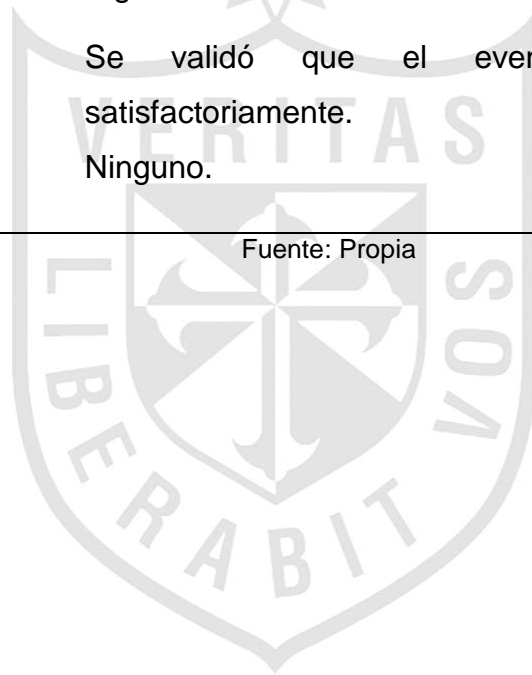


Tabla N°36 : Caso de prueba – registrar hito de asistencia

Escenario 1:	Validar que el dispositivo registre hitos de asistencia cada 15 minutos en el lapso de 1 hora estando el dispositivo móvil dentro de un rango de 5 metros
Entrada:	UIID Beacon: f7826da6-4fa2-4e98-8024-bc5b71e0893e Código alumno: a2008110986
Salida Esperada:	El hito de asistencia queda registrado con los siguientes datos Código de alumno, uiid, fecha de registro
Salida Obtenida:	Notificación de éxito en el dispositivo
Defectos:	Ninguno.

Fuente: Propia

3.4.2 Prueba de concurrencia

Así mismo se realizó las pruebas de concurrencia del Restful para saber cuántas peticiones podía soportar; para se utilizó la herramienta Apache Jmeter obteniéndose como resultados

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Media de Byt...
prueba de e...	1000	95298	229	623213	65821,60	10,60%	1,6/sec	55,09	35173,1
Total	1000	95298	229	623213	65821,60	10,60%	1,6/sec	55,09	35173,1

Ilustración 37 : Pruebas de concurrencia

Fuente: Propia

3.4.3 Pruebas de aceptación

Esta prueba se le enseñó al cliente que la funcionalidad cumpla con el requerimiento del sistema y de su aprobación para ser certificado por el equipo de QA y pase a manos del implementador.

3.4.4 Integración de versiones

Finalmente el implementador junta todas el código certificado por el equipo de QA y lanza una versión del sistema el cual pasa al repositorio de producción teniendo una versión del software funcional.

3.5 Mantenimiento

Tabla N°37 : Herramientas colaborativas

Herramientas utilizadas en esta etapa

Microsoft Word 2013	Documentación
Navicat Premium 2010	Administrador de base de datos
Android@Screen	Programa que permite visualizar la pantalla de un dispositivo móvil en una computadora.
Evthings Studio Workbench	Herramienta de desarrollo de móvil que permite visualizar la aplicación antes de compilarla.
Postman - REST Client	Herramienta online para probar servicios web

Fuente: Propia

En esta fase se tomó en cuenta los resultados de las pruebas para mejorar, corregir, agregar nuevos módulos de tal manera que se pueda tener a futuro una serie de versiones que vayan adaptándose a cambios de currículo, nuevos hardware, nuevos dispositivos, etc.

Finalmente, una vez puesto en marcha nuestro proyecto, se llega a la etapa final de nuestro modelo de desarrollo. Cuando los requisitos del sistema

cambien y haya que actualizar las aplicaciones de nuestro sistema entonces, la realización de la etapa de mantenimiento se realizará en base a los resultados de las pruebas de software obtenidos en la etapa anterior y teniendo en cuenta las siguientes consideraciones.

3.5.1 Corrección de errores

Se aplicó un mantenimiento correctivo si ocurre un fallo cuando el comportamiento del sistema es diferente del establecido en los requerimientos, este tipo de mantenimiento se aplicó cuando los fallos observados sean generados por problemas de procesamiento, programación y/o rendimiento.

Solo al activar la etapa de mantenimiento se volverá a reanudar desde el primer paso de la metodología si fuera necesario, o si hubiese un mejor escenario se volverá a revisar la etapa de diseño.

Se invertirá mayores recursos en la corrección de errores, aun cuando se emplee menos esfuerzo en corregir defectos. Este hecho puede parecer paradójico, pero se debe simplemente a que el sistema se usará más que los demás sistemas existentes en la facultad debido a que los procesos de asistencia son diarios con frecuencia muy cortos en casi todo el día. Por tanto, llegarán más propuestas de modificación y mejora.

3.5.2 Nuevos requerimientos

Una forma de desarrollar la fase de nuevos requerimientos será ejecutando un mantenimiento adaptativo para lo cual se debe adaptar a los nuevos requerimientos que nacen en base a las nuevas necesidades del usuario.

Además también cuando se esté en la etapa de nuevos requerimientos se tendrá que añadir nuevas funcionalidades, esta fase conllevará a una retroalimentación en todas las etapas del desarrollo debido a la experiencia y

necesidades que el usuario a adquirido porque el software ya se encuentra en ejecución.



CAPÍTULO IV

PRUEBAS Y RESULTADOS

4.1 Pruebas

En el presente capítulo se demostró si la solución planteada en la tesis cumplió con los objetivos propuestos.

Para esto, se midió los tiempos antes y después de la implementación de la propuesta, a una muestra de 220 alumnos los cuales están involucrados en el proceso de registro de asistencia.

Para determinar el tamaño de muestra se tuvo como base la siguiente fórmula estadística cuando se conoce la población en estudio; en este caso, 3800 alumnos matriculados en el semestre académico 2015-1 en la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres.

$$n = \frac{N * Z_{\alpha}^2 * p * q}{d^2 * (N - 1) + Z_{\alpha}^2 * p * q}$$

(Torres, Paz, & Salazar, 2006)

Donde

N = Tamaño de población

Z = nivel de confianza,

P = probabilidad de éxito, o proporción esperada

Q = probabilidad de fracaso

D = precisión (Error máximo admisible en términos de proporción)

Entonces reemplazando en la fórmula se obtiene el tamaño de la muestra de estudio.

$$n = \frac{3800 * 1.96^2 * 0.05 * 0.95}{0.03^2 * (3800 - 1) + 1.96^2 * 0.05 * 0.95} = 192.5292705$$

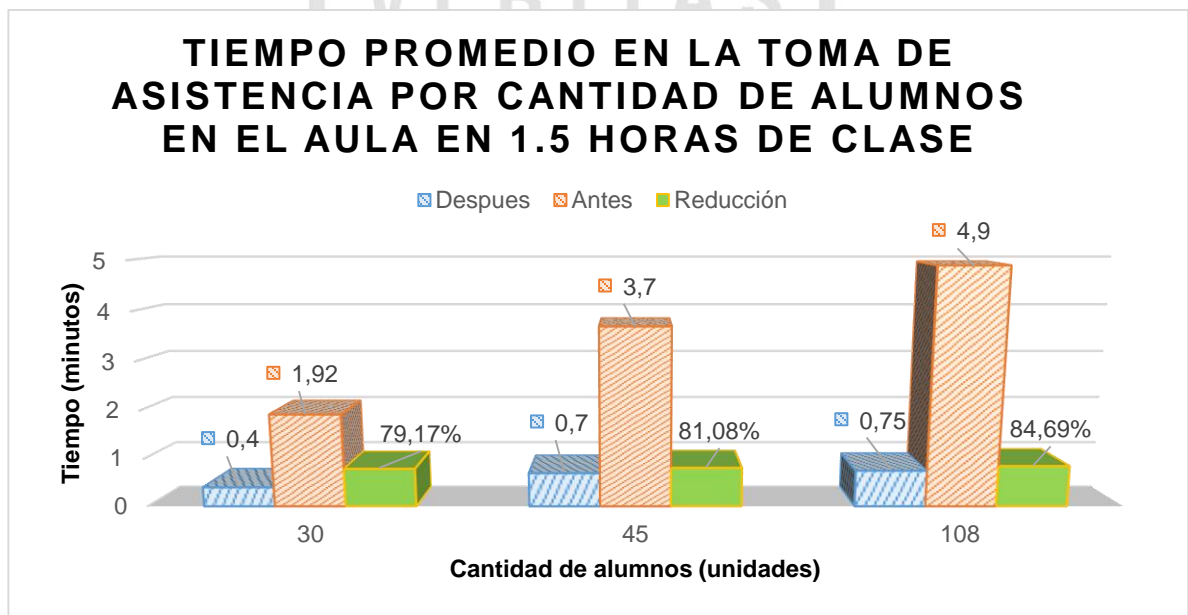


Ilustración 38 : Tiempo promedio por cantidad de alumnos en 1.5 horas
Fuente: Propia

Se puede observar que los tiempos se han reducido hasta 0.4 minutos (24 segundos) en promedio en un aula de 30 alumnos y hasta 0.75 minutos (45 segundos) en un aula de 108, debido a que la aplicación permite registrar simultáneamente a los alumnos; por otro lado se ve claramente que los tiempos son más óptimos a medida que el número de alumnos se incrementa.

Además, si hacemos un cálculo aproximado de cuánto tiempo se ahorraría en todo un ciclo académico tan solo en estudios generales FIA partiendo del hecho que en el semestre 2015-1 hay 12 aulas de 108 alumnos, en las cuales se toma asistencia en promedio 540 veces a la semana el cual multiplicado por las 16 semanas que dura un semestre académico son 8640 veces que se toma asistencia tan solo en estudios generales. Entonces si calculamos el tiempo de ahorro tendríamos 8640 por 0.069 horas (4.15 minutos) que nos da un resultado final de horas ahorradas de 597.6 horas

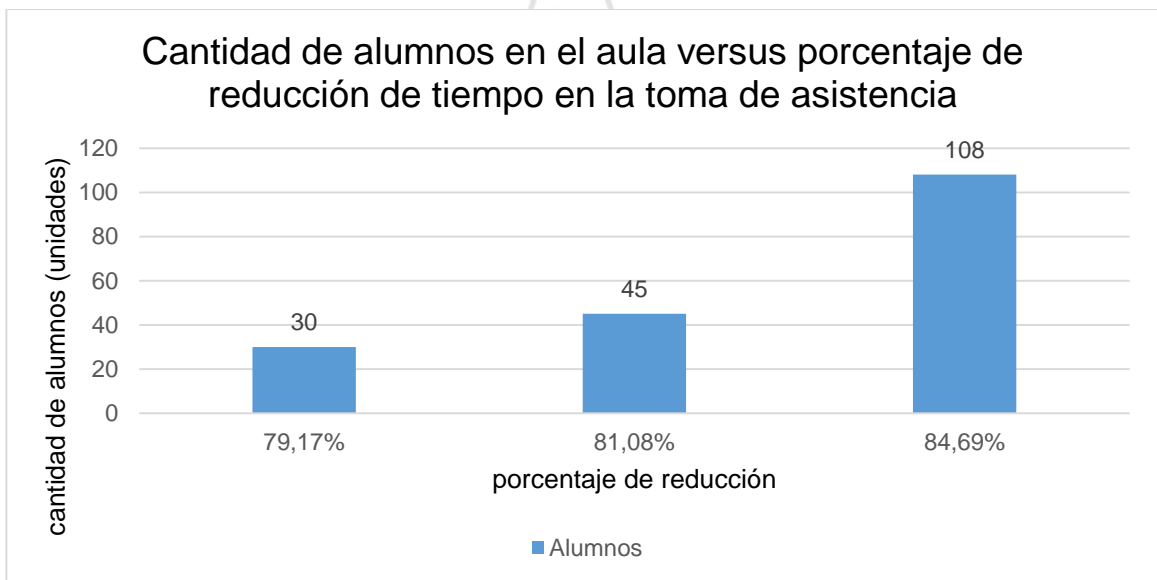


Ilustración 39 : Porcentaje de reducción de tiempo en la toma de asistencia
Fuente: Propia

Se aprecia que se logra reducir los tiempos hasta un 84.69% en un aula de 108 alumnos que es donde se ocupa mayor tiempo en la toma de asistencia 4.9 minutos en promedio.

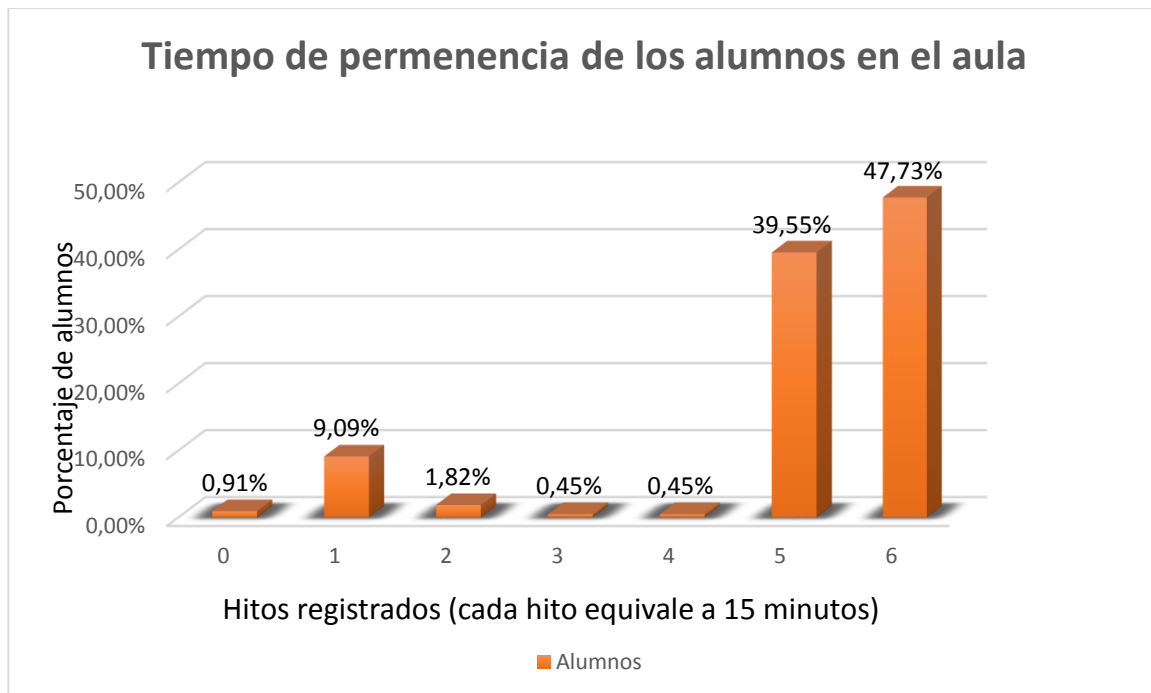


Ilustración 40 : Cantidad de hitos registrados por alumnos.
Fuente: Propia

Se puede apreciar que el 87.73% permaneció en el aula entre 1 y 1.5 horas (4 a 6 hitos) lo cual se puede interpretar que este grupo permaneció toda la clase.

Por otro lado, se aprecia que el 11.36% permaneció en el aula entre 15 a 45 minutos lo cual se puede interpretar que este grupo estuvo parcialmente en el aula.

Por último apreciamos en el 0.91% estuvo ausente en el aula.

4.2 Resultados

Tabla N°38 : Objetivos y resultados obtenidos

OBJETIVO	RESULTADO ESPERADO	RESULTADO OBTENIDO
Facilitar un mecanismo efectivo para el control de asistencia de los alumnos.	Poder realizar la ejecución del proceso de control de asistencia de manera automática y obtener de manera rápida la información actual del mismo.	Se pudo realizar mediante el mecanismo de detección el control de asistencia de alumnos de manera automática. Se visualiza el estado detallado de asistencia actual de cada uno de ellos.
Reducir los tiempos en un mínimo del 65% en el control de asistencia de los alumnos.	Poder reducir los tiempos para la realización del control de asistencia de los alumnos	Se pudo reducir los tiempos hasta un 84.69% para el registro de asistencia de los alumnos hasta 0.4 minutos (24 segundos) en promedio en un aula de 30 alumnos y hasta 0.75 minutos (45 segundos) en un aula de 108 alumnos

Fuente: Propia

Tabla N°39 : Objetivos y resultados obtenidos

OBJETIVO	RESULTADO ESPERADO	RESULTADO OBTENIDO
<p>Controlar la permanencia efectiva a clases de los alumnos.</p>	<p>Poder controlar y medir el tiempo de permanencia de los alumnos en el salón de clases</p>	<p>Se pudo obtener y controlar el tiempo que los alumnos se quedan dentro del aula de clase, de los cuales alumnos con 4 hitos de registro o más fueron el 87.73% y 11.36% tiene entre 1 y 3 hitos registrados</p>
<p>Contar con información de asistencia del alumno, que permita una mejor gestión del mismo.</p>	<p>Contar con información detallada y disponible las 24 horas del día sobre la asistencia de los alumnos.</p>	<p>El sistema permite mostrar el registro de asistencia de cada alumno, así como también obtener datos de los alumnos con más o menos faltas.</p>

Fuente: Propia

CAPÍTULO V

DISCUSIÓN Y APLICACIÓN

5.1 Discusión

En el presente proyecto, se observó el tiempo en que se incurre y se le resta tiempo a las horas académicas en el proceso de toma de asistencia de alumnos, por lo tanto se observó la necesidad de la USMP por querer mejorar el proceso de asistencia, con la finalidad de reducir tiempos de las actividades realizadas en el control de asistencia, mediante una adecuada coordinación entre las áreas involucradas. Esto se definió con la finalidad de brindar un mejor servicio a los alumnos de la USMP, puesto que mientras más rápido y mejor se realice el control de asistencia se mejorara la calidad de los servicios.

5.2 Aplicación

De una población de 3800 alumnos se tomó una muestra de 220 alumnos, a los cuales se aplicaron encuestas y entrevistas de los cuales se llegó a la conclusión, que lo mencionado en el primer párrafo del presente capítulo se lograría mediante la automatización del proceso de control de asistencia, es decir, detección de alumnos mediante mecanismos proactivos para el control de asistencia.

5.2.1 Otros sistemas de control de asistencia

Tabla N°40 : Comparación con otros sistemas de asistencia

Control de asistencia con sensores Bluetooth - Beacons	Control de asistencia por huella digital / tarjeta RFID
La marcación se puede hacer en simultáneo, es decir el sistema puede detectar y registrar la presencia de varios dispositivos al mismo tiempo.	El registro es uno a la vez lo cual puede generar colas o cuellos de botella en la entrada y/o salida.
La información del alumno en clase posee más hitos de control cada 15 minutos.	Generalmente posee 2 hitos de control a la entrada y a la salida
El docente no debe preocuparse por el marcado de la asistencia de los alumnos en el aula.	El docente no debe preocuparse por el marcado de la asistencia de los alumnos en el aula.
El control de asistencia depende que el alumno use un Smartphone.	El sistema no depende de un Smartphone.
La aplicación es escalable y modular	Generalmente es un sistema cerrado

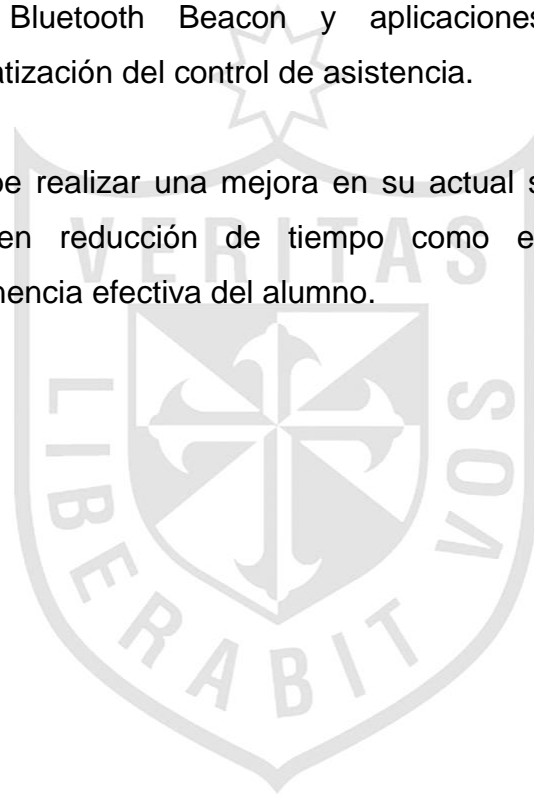
Fuente: Propia

CONCLUSIONES

- Primera:** Se logró Implementar los mecanismos proactivos de detección de alumnos para el control de asistencia en la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres.
- Segunda:** La aplicación de un mecanismo proactivos para la detección de alumnos se redujo los tiempos de toma de asistencia hasta un 84.69%.
- Tercera:** La aplicación provee información detallada sobre la asistencia de los alumnos, para la toma de decisiones, lo cual también permite integrar con el sistema de gestión académica y administrativa de la FIA
- Cuarta:** Al reducirse los tiempos de toma de asistencia esto incrementó proporcionalmente las horas efectivas de clase.
- Quinta:** Con la aplicación móvil los maestros ya no tuvieron que preocuparse por el proceso de toma de asistencia ya que la aplicación lo sistematizó.

RECOMENDACIONES

- Primera:** Para evitar problemas de conectividad de los dispositivos portátiles de los alumnos, con la red de la FIA, se sugiere ampliar el ancho de banda de la institución, para permitir implementar soluciones móviles complementarias al sistema de asistencia.
- Segunda:** La Universidad debe aplicar el uso de nuevas tecnologías tales como Bluetooth Beacon y aplicaciones móviles para la automatización del control de asistencia.
- Tercera:** Se debe realizar una mejora en su actual sistema de asistencia tanto en reducción de tiempo como en el control de la permanencia efectiva del alumno.



FUENTES DE INFORMACIÓN

Ana, H. A., & Gader, I. N. (11 de 06 de 2011). repositorio graduate. Obtenido de repositorio graduate:

www.dit.ing.unp.edu.ar/graduate/handle/123456789/206

Ana, H. A., & Gader, I. N. (06 de 06 de 2011). *Repositorio Graduate*.

Recuperado el 02 de 06 de 2015, de Repositorio Graduate:

<http://www.dit.ing.unp.edu.ar/graduate/handle/123456789/206>

Ana, H. A., & Noel, G. I. (11 de 06 de 2011). *dit.ing.* (F. Cajal, Ed.) Obtenido de

dit.ing: <http://hdl.handle.net/123456789/206>

Apache Cordova™. (3 de 06 de 2015). *Apache Cordova*. Obtenido de Apache

Cordova: <https://cordova.apache.org/#about>

Camarero, J., Fumero, A., Wertersky, A., Rodriguez, P., & Blanco, p. (2009).

Metodología de desarrollo ágil para sistemas móviles. Obtenido de

http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf

El Comercio. (27 de 02 de 2015). *El comercio*. Recuperado el 03 de 06 de 2015,

de El comercio: <http://elcomercio.pe/paginas/smartphones-tablets/tablets-peru-que-marcas-dominan-ahora-mercado-advanced-samsung-noticia-1793558>

Estimote Inc. (2 de Mayo de 2015). *Estimote Beacons — real world context for*

your apps. Obtenido de Estimote Beacons — real world context for your apps: <http://estimote.com/api/>

Estimote. (03 de 06 de 2015). *Estimote Beacons — real world context for your apps*. Obtenido de Estimote Beacons — real world context for your apps:
<http://estimote.com/api/>

Gerónimo-Castillo, G., & Rocha-Trejo, E. (2007). *Revistas UNED*. Obtenido de
<http://revistas.uned.es/index.php/ried/article/view/1014>

Gestión. (03 de 06 de 2015). *Gestión | El diaria de economía y negocios en el Perú*. Recuperado el 03 de 06 de 2015, de Gestión | El diaria de economía y negocios en el Perú:

<http://gestion.pe/noticia/1278159/smartphones-captan-56-mercado-movil>

Jones, C. (2010). *Software Engineering Best Practices 1*. New York: McGraw-Hill, Inc.

Morales, P. R. (08 de 07 de 2011). *upcommons*. Obtenido de upcommons:
<http://upcommons.upc.edu/pfc/bitstream/2099.1/13249/1/memoria.pdf>

Oré, I. A. (2009). *Calidad y software.cvom*. Obtenido de
http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php

Rodriguez, M., Pece, J., & Escudero, C. (2005). *researchgate*. Obtenido de
http://www.researchgate.net/profile/CJ_Escudero/publication/228885933_Blueps_Sistema_de_localizacin_en_interiores_utilizando_Bluetooth/links/00b7d5257a96589b1f000000.pdf

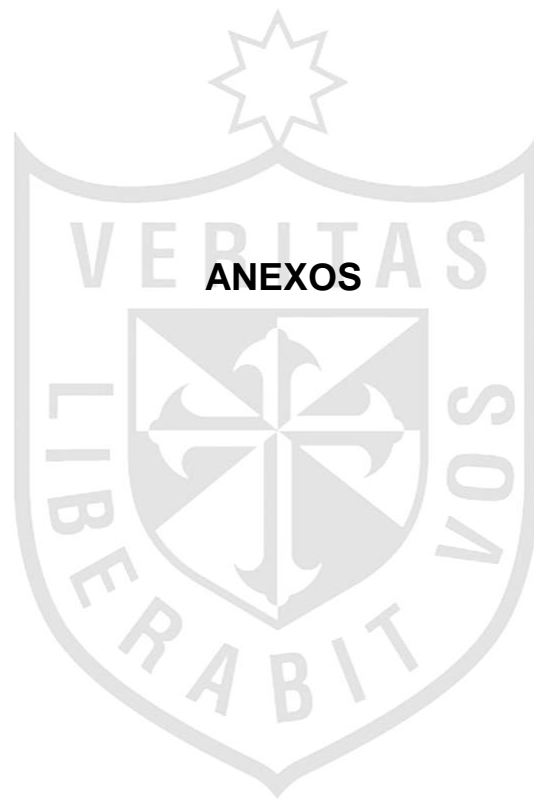
Timoftii, & Lucian, V. (27 de 06 de 2013). *Diseño y desarrollo de una aplicación híbrida para smartphone usando el framework jquery movile y phonegap*. Obtenido de Diseño y desarrollo de una aplicación híbrida para smartphone usando el framework jquery movile y phonegap:
<http://academica-e.unavarra.es/bitstream/handle/2454/7544/578081.pdf?sequence=1>

Timoftii, V. L. (27 de 06 de 2013). Obtenido de <http://academica-e.unavarra.es/bitstream/handle/2454/7544/578081.pdf?sequence=1&isAllowed=y>

Torres, M., Paz, K., & Salazar, F. G. (2006). *Facultad de Ingeniería - Universidad Rafael Landívar*, 2. Obtenido de Facultad de Ingeniería - Universidad Rafael Landívar:
http://cvonline.uaeh.edu.mx/Cursos/Maestria/MTE/seminario_de_tesis/Universidad%203/Lect_Muestra_Invest.pdf

Universidad de San Martín de Porres. (02 de 06 de 2015). *USMP*. Obtenido de USMP: <http://www.usmp.edu.pe/index.php?pag=nuesuniv&sec=resena>





ANEXOS

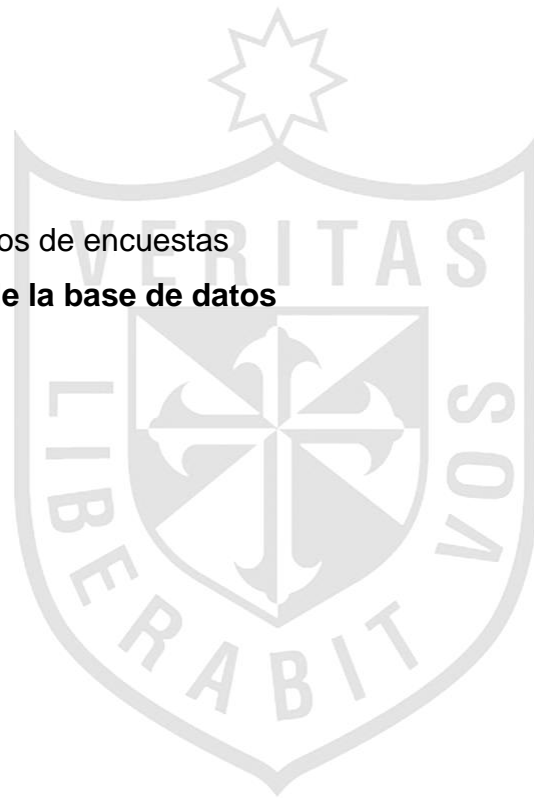
ANEXO 1: Resultados de encuestas

Página

126

ANEXO 2 : Script de la base de datos

131



ANEXO 1: Resultados de encuestas

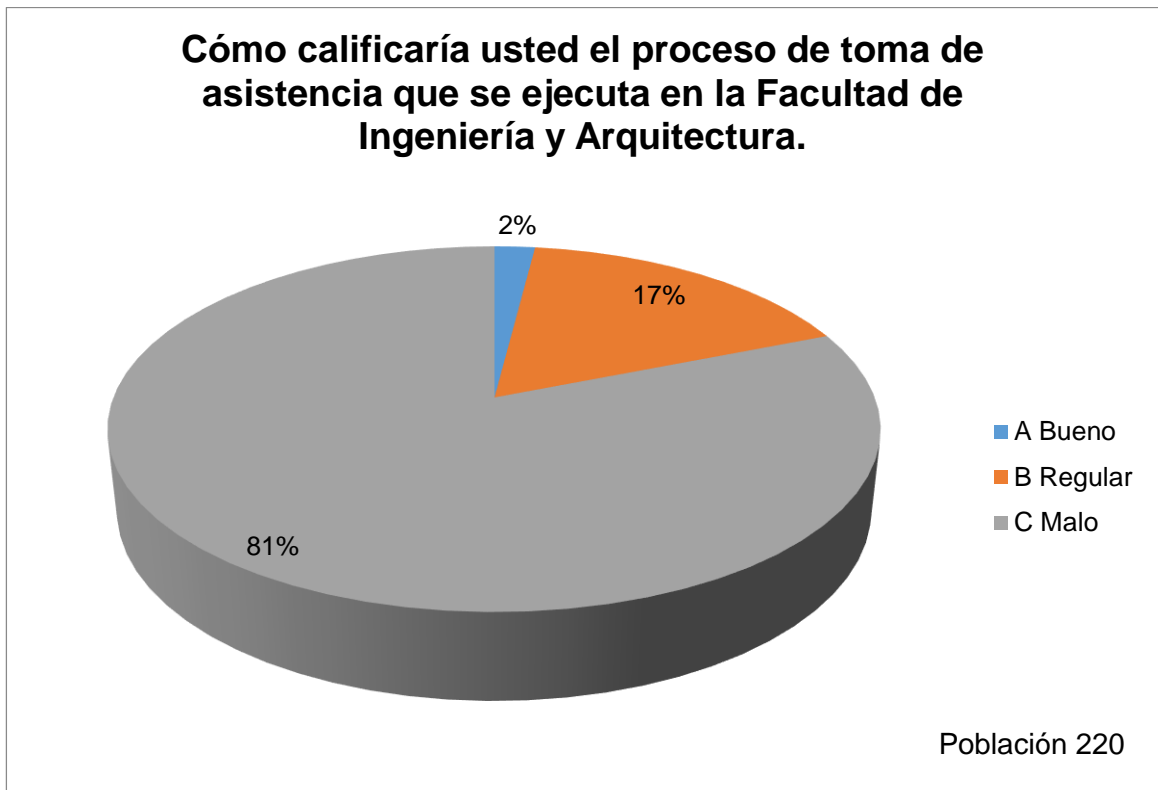


Ilustración 41 : Calificación del sistema actual del sistema de asistencia.
Fuente: Propia

De un total de 220 encuestados el 81% considera que el proceso actual de toma de asistencia es malo , el 17% regular y tan solo el 2% considera que es bueno por lo que se evidencia una gran demanda de una mejor propuesta al proceso actual.

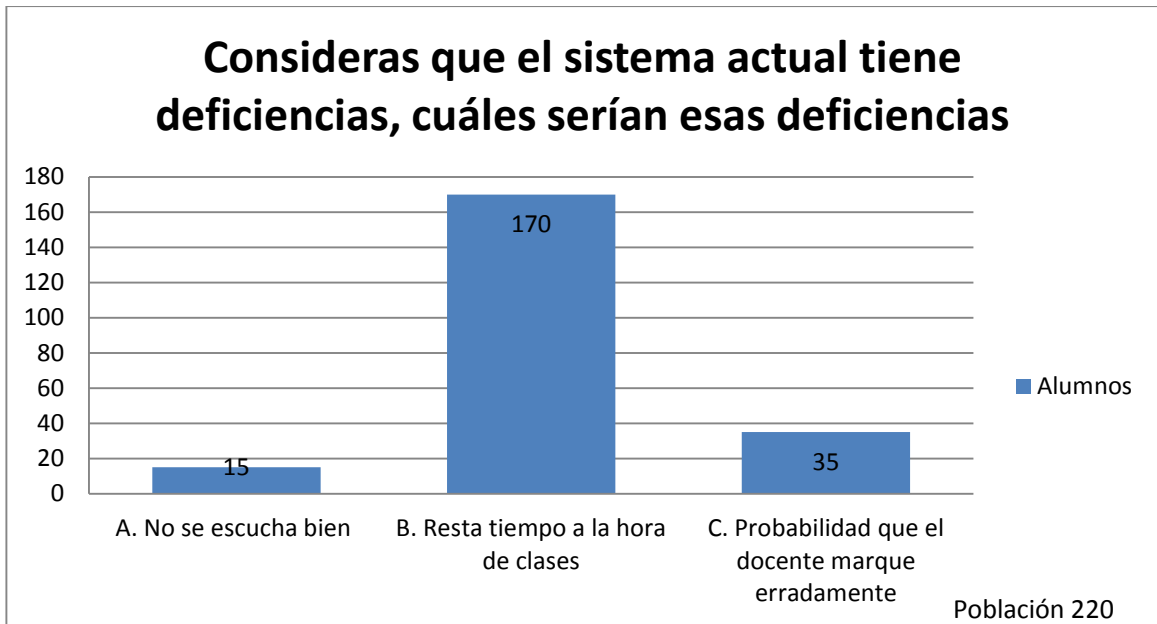


Ilustración 42 : Deficiencias sobre el sistema actual de toma de asistencia
Fuente: Propia.

Se puede notar que 170 alumnos (77%) considera que el sistema actual resta tiempo a la horas de clases, 35 alumnos (16%) piensa que hay probabilidad que el docente marque erradamente y 15 alumnos (7%) considera que no se escucha bien.

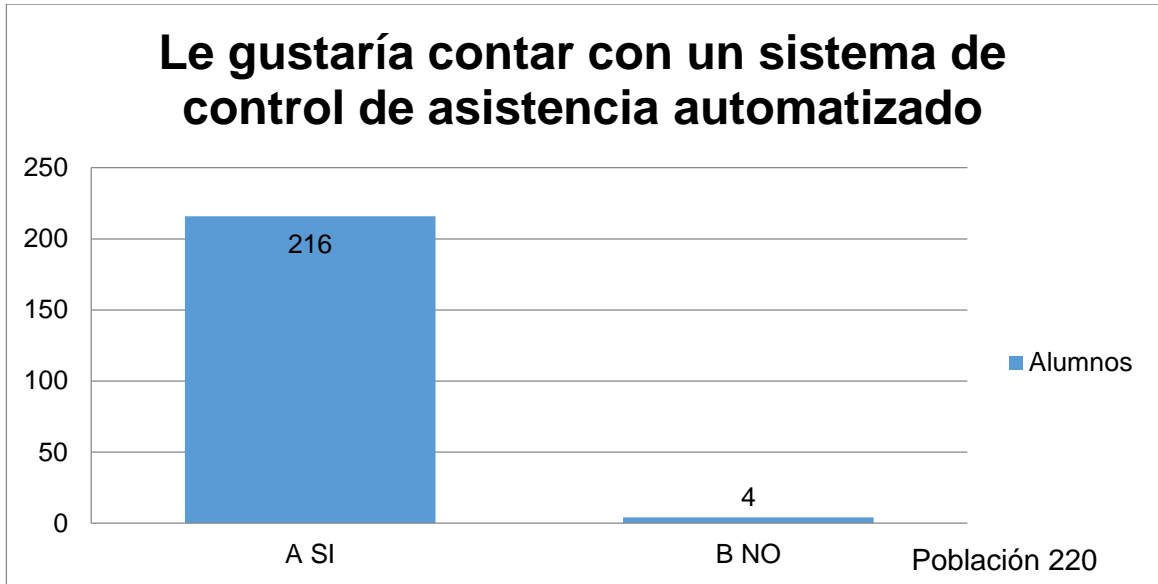


Ilustración 43 : Pregunta sobre aceptación del sistema actual
Fuente: Propia.

Se evalúa la demanda sobre la implementación de un sistema automatizado. Obteniéndose como resultados que 216 alumnos (98%) considera que sí le gustaría contar con un sistema automatizado mientras que 4 alumnos (2%) no.

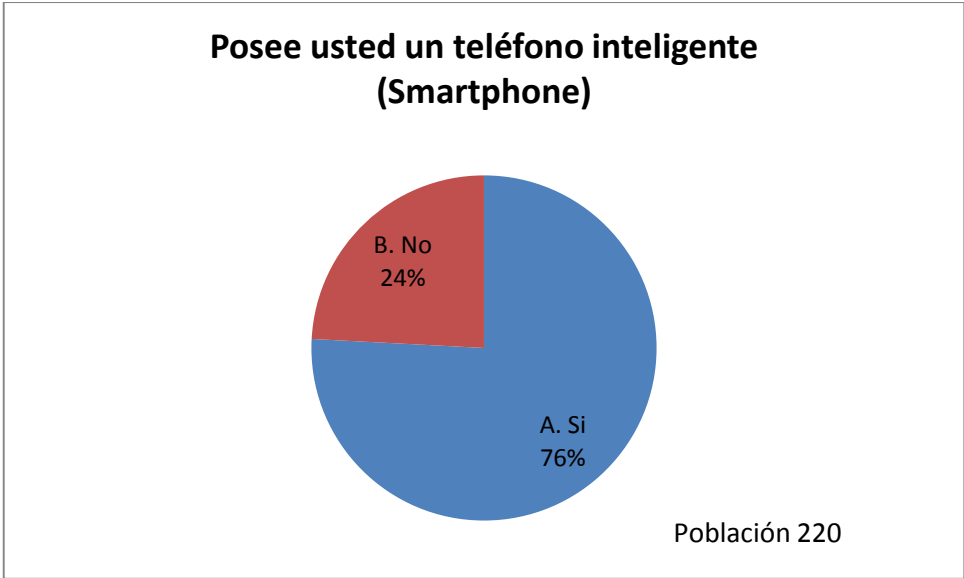
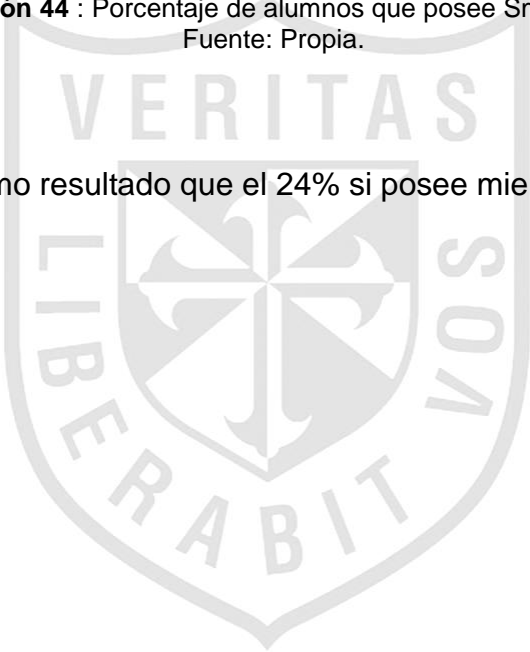


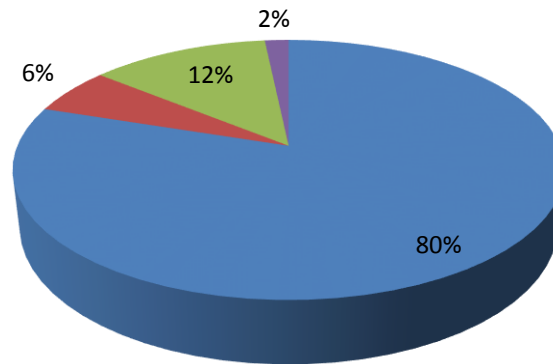
Ilustración 44 : Porcentaje de alumnos que posee Smartphone
Fuente: Propia.

Se obtuvo como resultado que el 24% si posee mientras que el 76% no.



En caso de poseer un teléfono inteligente. Con que tipo de sistema operativo cuenta

■ A. Android ■ B. Windows Phone ■ C. IOS ■ D. Blackberry OS



Población 168

Ilustración 45 : consulta sobre el tipo sistema operativo
Fuente: Propia.

El 80% usa Android el 12% IOS, el 6% Windows Phone y un 2% Blackberry con esta información se puede apreciar que el desarrollo de una aplicación híbrida es viable por la diversidad de sistemas operativos.

ANEXO 2 : Script de la base de datos

```
SET FOREIGN_KEY_CHECKS=0;
```

➤ Table structure for `asistencia`

```
DROP TABLE IF EXISTS `asistencia`;  
CREATE TABLE `asistencia` (  
  `estudiante_id` varchar(10) NOT NULL DEFAULT "",  
  `fecha_asistencia` date DEFAULT NULL,  
  `estado` tinyint(1) DEFAULT NULL,  
  `horario_id` int(11) DEFAULT NULL,  
  PRIMARY KEY (`estudiante_id`),  
  KEY `fk_estado` (`estado`),  
  KEY `fk_horario_id` (`horario_id`),  
  CONSTRAINT `fk_estado` FOREIGN KEY (`estado`) REFERENCES  
  `asistencia_estado` (`estado_id`) ON DELETE NO ACTION ON UPDATE NO  
  ACTION,  
  CONSTRAINT `fk_horario_id` FOREIGN KEY (`horario_id`) REFERENCES  
  `horario` (`horario_id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
- Records of asistencia
```

➤ Table structure for `asistencia_estado`.

```
DROP TABLE IF EXISTS `asistencia_estado`;  
CREATE TABLE `asistencia_estado` (  
  `estado_id` tinyint(1) NOT NULL DEFAULT '0',  
  `descripcion` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`estado_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
-- Records of asistencia_estado
```

➤ **Table structure for `Beacon`**

```
DROP TABLE IF EXISTS `beacon`;  
CREATE TABLE `beacon` (  
  `beacon_id` varchar(200) NOT NULL DEFAULT "",  
  `color` varchar(200) DEFAULT NULL,  
  PRIMARY KEY (`beacon_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
-- Records of beacon
```

➤ **-- Table structure for `beacon_salon`**

```
DROP TABLE IF EXISTS `beacon_salon`;  
CREATE TABLE `beacon_salon` (  
  `beacon_id` varchar(200) NOT NULL DEFAULT "",  
  `salon_id` int(11) NOT NULL,  
  PRIMARY KEY (`beacon_id`,`salon_id`),  
  KEY `fk_salon_id` (`salon_id`),  
  CONSTRAINT `fk_beacon_id` FOREIGN KEY (`beacon_id`) REFERENCES  
  `beacon` (`beacon_id`),  
  CONSTRAINT `fk_salon_id` FOREIGN KEY (`salon_id`) REFERENCES  
  `salon` (`salon_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
-- Records of beacon_salon
```

➤ **-- Table structure for `curso`**

```
DROP TABLE IF EXISTS `curso`;  
CREATE TABLE `curso` (  
  `curso_id` int(5) NOT NULL DEFAULT '0',  
  `codigo` varchar(5) DEFAULT NULL,  
  `nombre` varchar(120) DEFAULT NULL,  
  `descripcion` text,  
  PRIMARY KEY (`curso_id`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- Records of curso
```

➤ -- Table structure for `estudiante`

```
DROP TABLE IF EXISTS `estudiante`;
CREATE TABLE `estudiante` (
  `estudiante_id` varchar(10) NOT NULL DEFAULT '0',
  `nombre` varchar(100) DEFAULT NULL,
  `apellido_paterno` varchar(100) DEFAULT NULL,
  `apellido_materno` varchar(100) DEFAULT NULL,
  `fecha_nacimiento` varchar(100) DEFAULT NULL,
  `sexo` char(1) DEFAULT NULL,
  PRIMARY KEY (`estudiante_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- Records of estudiante
```

➤ -- Table structure for `estudiante_hito`

```
DROP TABLE IF EXISTS `estudiante_hito`;
CREATE TABLE `estudiante_hito` (
  `hito_id` int(11) NOT NULL DEFAULT '0',
  `fecha_registro` time DEFAULT NULL,
  `estudiante_id` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`hito_id`),
  KEY `fk_estudiante_id` (`estudiante_id`),
  CONSTRAINT `fk_estudiante_id` FOREIGN KEY (`estudiante_id`)
REFERENCES `asistencia` (`estudiante_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- Records of estudiante_hito
```

➤ -- Table structure for `horario`

```
DROP TABLE IF EXISTS `horario`;
```

```

CREATE TABLE `horario` (
  `horario_id` int(11) NOT NULL DEFAULT '0',
  `salon_id` int(11) DEFAULT NULL,
  `fecha` date DEFAULT NULL,
  `curso_id` int(5) DEFAULT NULL,
  `estudiante_id` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`horario_id`),
  KEY `salon_id` (`salon_id`),
  KEY `estudiante_id` (`estudiante_id`),
  KEY `curso_id` (`curso_id`),
  CONSTRAINT `curso_id` FOREIGN KEY (`curso_id`) REFERENCES `curso`
(`curso_id`),
  CONSTRAINT `estudiante_id` FOREIGN KEY (`estudiante_id`)
REFERENCES `estudiante` (`estudiante_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- Records of horario

```

➤ **-- Table structure for `salon`**

```

DROP TABLE IF EXISTS `salon`;
CREATE TABLE `salon` (
  `salon_id` int(11) NOT NULL DEFAULT '0',
  `ubicacion` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`salon_id`),
  CONSTRAINT `fk_salon2_id` FOREIGN KEY (`salon_id`) REFERENCES
`horario` (`salon_id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- Records of salon

```