



UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES
ESCUELA INGENIERIA CIVIL INFORMÁTICA

Sistema de Asistencia Semi-Automático vía Smartphone

Memoria para optar al título de Ingeniero Civil en Informática

Alumno : Israel Gajardo Recabal

Profesora Guía: Marlene Muñoz Sepúlveda

Enero de 2019

Chillán, Chile

AGRADECIMIENTOS

La Universidad es un lugar que requiere mucho esfuerzo para salir adelante, siempre hay obstáculos que superar y es un constante desafío, por eso, al llegar a este punto de mi carrera quiero agradecer a mis amigos, quienes no me apoyaron académicamente o en momentos difíciles, sino que me hicieron olvidar los problemas sirviendo como ayuda para que a pesar de todo el estrés que conlleva a la Universidad pueda salir adelante.

RESUMEN

El presente proyecto es presentado para dar conformidad a los requisitos de la Universidad del Bío-Bío para finalizar la Carrera de Ingeniería Civil en Informática.

La Universidad del Bío-Bío es una organización de derecho público, perteneciente al Ministerio de Educación, que cumple el objetivo de entregar una educación superior de calidad, formando así personas integrales a través de pregrados y postgrados, además se dedica a investigación fundamental, aplicada y de desarrollo.

La problemática presente en la Universidad se origina en que por cada clase realizada por un docente hay que registrar asistencia a los estudiantes, esta asistencia debe quedar guardada y almacenada en la plataforma institucional Intranet. La responsabilidad de almacenar la asistencia en intranet, recae en los hombros del docente, teniendo la opción de detener la clase para pasar asistencia directamente en intranet o llevarse una lista manual a su oficina para posteriormente traspasarla a intranet. Ambas opciones son perjudiciales ya sea para la clase o para el profesor.

El "Sistema de Asistencia Semi-Automático vía Smartphone" cumple el objetivo de automatizar el proceso de registro de asistencia por medio de un sistema disponible para Smartphones Android e iOS, tanto para Docentes y Estudiantes, el cual interconecta las dos aplicaciones de manera que los alumnos se registren directamente en intranet desde su aplicación, y el profesor gestione este proceso desde la suya. Las bases para la creación del sistema son que la información presentada por este sea Auténtica y el sistema otorgue una buena experiencia de usuario.

El desarrollo del sistema se dio en forma nativa para las plataformas Android e iOS, utilizando los IDE de desarrollo oficiales presentados por estas plataformas, Android Studio para desarrollo de la versión Android y XCode para la versión iOS. Utilizando una arquitectura Modelo-Vista-Controlador(MVC) junto a un enfoque orientado a objetos y la metodología Desarrollo Basado en Funcionalidades(FDD).

ABSTRACT

This project is presented to comply with the requirements of the Universidad del Bío-Bío to complete the Civil Engineering Degree in Computer Science.

The University of Bío-Bío is a public law organization, belonging to the Ministry of Education, which fulfills the objective of delivering a quality higher education, thus forming holistic people through undergraduate and postgraduate courses, in addition to fundamental research, applied and development.

The problem present in the University originates in that each class carried out by a teacher must pass assistance to the students, this assistance must be stored in the Intranet institutional platform. The responsibility of storing the list in the intranet falls on the shoulders of the teacher, having the option of stopping the class to pass assistance directly in the intranet or take a manual list to his office and then transfer it to the intranet. Both options are detrimental to either the classroom or the teacher.

The "Semi-Automatic Assistance System via Smartphone" fulfills the objective of automating the attendance registration process through a system available for Android and iOS Smartphones, both for Teachers and Students, which interconnects the two applications so that Students register directly on the intranet from their application, and the teacher manages this process from theirs. The bases for the creation of the system are that the information presented by this is Authentic and the system provides a good user experience.

The development of the system occurred in a native way for the Android and iOS platforms, using the official development IDE presented by these platforms, Android Studio for the development of the Android version and XCode for the iOS version. Using a Model-View-Controller (MVC) architecture together with an object-oriented approach and the Feature-Driven Development (FDD) methodology.

INDICE

1.	INTRODUCCIÓN.....	1
2.	DESCRIPCIÓN DEL PROBLEMA	3
2.1.	Descripción de la Organización.....	3
2.2.	Ámbito del Problema.....	4
2.3.	Sistema Actual	6
3.	DEFINICIÓN DEL PROYECTO	7
3.1.	Objetivo General y Objetivos Específicos del Proyecto.....	7
3.2.	Requerimientos del Sistema.....	8
3.3.	Ambiente de ingeniería de Software	8
4.	CARACTERIZACIÓN DEL SOFTWARE.....	17
4.1	Alcance del Proyecto	17
4.2	Objetivo del Producto	17
4.3	Descripción Global del Producto.....	18
4.4.	Requerimientos del Sistema	19
5.	ESTUDIO DE FACTIBILIDAD.....	20
5.1.	Factibilidad Técnica.....	20
5.2.	Factibilidad Operativa.....	22
5.3.	Factibilidad Económica.....	24
6.	MODELAMIENTO DEL SISTEMA.....	29
6.1.	Análisis y Diseño de la Solución	29
6.2.	Modelos de Datos	36
6.3.	Conexión con la Base de Datos.....	40
6.5.	Diseño de Interfaz y Navegación	41

6.6. Proceso de Negocio Involucrado.....	59
6.7. Casos de Uso	62
7. CONCLUSIÓN.....	74
8. REFERENCIAS BIBLIOGRÁFICAS.....	76

INDICE FIGURAS

Figura 1: Diagrama con funciones de la Aplicación Original, elaboración propia.....	29
Figura 2: Diagrama con funciones de la Nueva Aplicación, elaboración propia.....	30
Figura 3: Funcionamiento del Sistema, elaboración propia	34
Figura 4: Modelo Entidad Relación.....	37
Figura 5: Mockup Menú de Inicio, Aplicación Docente	41
Figura 6: Mockup Menú Principal, Aplicación Docente	42
Figura 7: Mockup Menú de Pasar lista, Aplicación Docente.....	43
Figura 8: Mockup Recuperar Clase, Aplicación Docente.....	44
Figura 9: Mockup Menú Lista Alumnos Manual/Token, Aplicación Docente.....	46
Figura 10: Mockup Menú Selección Estadísticas, Aplicación Docente	47
Figura 11: Mockup Menú Suspende Clases, Aplicación Docente.....	48
Figura 12: Mockup Menú Estadísticas General, Aplicación Docente	49
Figura 13: Mockup Menú Estadísticas General, Aplicación Docente	50
Figura 14: Mockup Informe Día, Aplicación Docente	51
Figura 15: Mapa de Navegación de Aplicación del Profesor.....	53
Figura 16: Mockup Menú Principal, Aplicación Alumno	55
Figura 17: Mockup Registrarse en Lista, Aplicación Alumno	56
Figura 18: Mockup Ver Estadísticas, Aplicación Alumno	57
Figura 19: Mapa de Navegación Aplicación del Alumno.....	58
Figura 20: BPMN	61
Figura 21: Diagrama de Casos de Uso	63

INDICE TABLAS

Tabla 1: Costos de Inversión en Horas Desarrollo.....	24
Tabla 2: Costos de Inversión en Licencias Android e iOS.....	24
Tabla 3: Costos de Inversión en Utensilios de Desarrollo.....	25
Tabla 4: Costos de Operación del Proyecto.....	25
Tabla 5: Sueldo Aproximado de un Profesor	26
Tabla 6: Beneficio anual con respecto a las clases semanales y el número de usuarios.....	27
Tabla 7: Calculo VAN	28
Tabla 8: Especificación Caso de Uso: Inicio de Sesión Intranet.....	64
Tabla 9: Especificación Caso de Uso: Pasar Lista a Clase Actual en Modo Manual	65
Tabla 10: Especificación Caso de Uso: Pasar Lista a Clase Actual en Modo Token	66
Tabla 11: Especificación Caso de Uso: Recuperar Clase en Modo Manual.....	67
Tabla 12: Especificación Caso de Uso: Recuperar Clase en Modo Token	68
Tabla 13: Especificación Caso de Uso: Suspender Clase	69
Tabla 14: Especificación Caso de Uso: Ver Reporte de la Clase	70
Tabla 15: Especificación Caso de Uso: Ver Estadísticas de Asistencia por Ramo	70
Tabla 16: Especificación Caso de Uso: Ver Estadísticas de Asistencia por Alumno	71
Tabla 17: Especificación Caso de Uso: Almacenar Asistencia del Alumno	72
Tabla 18: Especificación Caso de Uso: Ver Reporte de Asistencia por Asignatura.....	73

1. INTRODUCCIÓN

En el presente informe se da a conocer el proyecto de título Israel Gajardo Recabal, estudiante de la carrera Ingeniería Civil en Informática, Facultad de Ciencias Empresariales, Universidad del Bío-Bío, campus Fernando May.

En la sociedad actual el uso de los teléfonos inteligentes y las aplicaciones móviles son de extrema popularidad, estos se convirtieron en un ícono de nuestra la presente generación y un invento no se eliminará, sino solo evolucionará a futuro. Esta herramienta es muy útil y posee componentes que hace años nunca se hubiera creído podían estar disponibles para cualquier usuario. Esto se menciona con el objetivo de decir, si bien estos dispositivos han tenido un gigantesco impacto cultural, aún pueden seguir desarrollándose, hay muchas tareas que se siguen realizando como hace años atrás, a pesar de tener un dispositivo capaz de realizarlas, tareas cotidianas y repetitivas que son muy optimizables con un Smartphone, solo faltan las ideas y la disposición. Los informáticos y desarrolladores deben buscar mejorar y optimizar lo que ya existe para poder tener más tiempo para otras cosas.

El presente proyecto busca la mejora de una actividad cotidiana en las Universidades, como lo es pasar asistencia, para así facilitar y otorgar más tiempo a las horas docentes. El proyecto es titulado Sistema de Asistencia Automático Vía Smartphone – “Un QR por una Token”, y en términos simples consiste en una adaptación a un sistema ya existente de la Universidad, no muy utilizado, el cual cumple la función de pasar asistencia de manera automática a través de dos aplicaciones móviles, pero que está en desuso dados problemas de usabilidad. Por lo que se busca ofrecer otra manera de realizar las mismas tareas, de forma que los docentes tengan un sistema que les permita pasar asistencia más fácilmente.

El presente informe consta de 7 capítulos los cuales contienen:

- Capítulo 2: Presenta una descripción de la Organización y la problemática que esta posee.
- Capítulo 3: Presenta los objetivos generales y específicos del proyecto, además de los requerimientos y el ambiente de ingeniería de software.
- Capítulo 4: Presenta una descripción del producto, sus interfaces y los requerimientos del sistema
- Capítulo 5: Presenta la factibilidad técnica, operativa y económica del proyecto.
- Capítulo 6: Presenta los distintos modelos del sistema para entender de mejor manera su funcionamiento.
- Capítulo 7: Presenta la conclusión del proyecto.

2. DESCRIPCIÓN DEL PROBLEMA

2.1. Descripción de la Organización

La organización para la que se construye el sistema corresponde a la Universidad del Bío-Bío (UBB), la cual comprende tanto a los 2 campus de la sede Chillán, “La Castilla” y “Fernando May”, además del campus de la sede Concepción, “Concepción”. Esta organización es dependiente del Ministerio de Educación, es de derecho público y cumple el objetivo de entregar una educación superior de calidad, formando así personas integrales a través de pregrados y postgrados, además se dedica a investigación fundamental, aplicada y de desarrollo.

Específicamente, dentro de lo que es la UBB se trabaja con el Departamento de Desarrollo de Sistemas de la Dirección de Informática, la cual está ubicada en el campus de Concepción, el encargado de esta es el Sr. Joan Briones. Esta división de la Universidad es la encargada de parte del desarrollo de sistemas, siendo una de sus tareas la publicación de las aplicaciones de la Universidad. Se trabaja con ellos y no con la sede de Chillán debido a que esta sede publicó originalmente la Aplicación, y el proyecto se centra en mejorar esta.

2.2. Ámbito del Problema

La Universidad posee un sistema de registro de asistencia, por medio de la plataforma institucional Intranet UBB, este sistema mantiene un control sobre cuándo se realiza una clase, qué contenidos se pasaron y quienes asistieron a esta, exigiendo así a los docentes que por cada clase realizada, haya una lista de asistencia. La información obtenida por el sistema de asistencia permite calcular estadísticas y evaluar a los profesores por lo que, para que esta información cumpla su objetivo, debe ser auténtica.

Para poder hacer vigente el sistema de asistencia exigido por la Universidad, hay que ingresar los estudiantes presentes en cada clase a la plataforma intranet, esto se puede hacer de múltiples formas, habiendo dos predominantes, las cuales poseen distintas ventajas y desventajas:

- Pasar lista en la clase registrando asistencia a los alumnos directamente en intranet:
 - **Ventaja:** se ingresa todo directamente al sistema, sin tener que hacer nada a futuro.
 - **Desventaja:** requiere de una interrupción a la clase o tiempo adicional, y un computador para abrir la lista y pasar la asistencia, además de conexión a internet.
- Hacer una lista de asistencia en papel, en la cual los alumnos ponen su nombre y la firman para posteriormente ser ingresada a intranet:
 - **Ventaja:** la clase transcurre normalmente sin interrupciones.
 - **Desventaja:** se requiere un tiempo extra a futuro para traspasar la lista a intranet, siendo esta una tarea larga y engorrosa ya que requiere buscar en intranet a cada alumno que este en la lista en papel.

Se puede ver que ambas opciones poseen desventajas que no son despreciables, y que a la larga son un problema para el profesor. Por un lado se requiere interrumpir el flujo normal de la clase por unos minutos, y otro se tiene llevar el problema a la oficina, requiriendo más tiempo que el usado al interrumpir la clase.

Actualmente la Universidad posee un sistema más automático, que busca cubrir estas falencias, pero que posee grandes problemas de seguridad y usabilidad. Como una de las exigencias al momento de almacenar la lista en intranet, es la autenticidad de la información, una falla en la seguridad resulta fatal para el sistema, además si se tienen problemas de usabilidad que son mayores al problema original, los usuarios preferirán mantenerse con el sistema actual, antes de moverse a la “solución”. Dadas estas falencias, el sistema nunca salió de la versión de pruebas, quedándose así con el problema original intacto,.

2.3. Sistema Actual

El Sistema Actual se conforma de dos aplicaciones móviles, disponibles para Smartphones iOS y Android. Estas aplicaciones, una para el profesor y otra para el alumno, sirven como medio para pasar lista, y almacenarla en Intranet. Funcionan con constante conexión a internet, de modo que la App del Profesor genera un QR para el curso y los estudiantes lo escanean desde su App, con lo que, después de hacer verificaciones de seguridad por medio de GPS, quedan presentes y almacenados en intranet automáticamente, aplacando así las dos grandes desventajas del proceso actual, las cuales son: interrumpir la clase y tener que traspasar la lista a intranet.

Pero como se mencionó previamente, este sistema posee grandes falencias, las cuales son:

- El profesor debe prestar su teléfono y pasar su Smartphone por toda la sala, estudiante por estudiante, para que el sistema pueda funcionar
- Los estudiantes pueden conseguir una foto del QR y escanearlo desde fuera de la sala, quitando autenticidad a la lista almacenada
- Si los edificios son muy grandes, el GPS puede fallar, impidiendo así que un alumno quede presente.

Estos problemas, que se pueden resumir en mala seguridad y mala experiencia de usuario, son la razón de que el sistema nunca haya salido de la versión de pruebas, y que se siga registrando asistencia de la manera tradicional.

3. DEFINICIÓN DEL PROYECTO

3.1. Objetivo General y Objetivos Específicos del Proyecto

General:

El objetivo del proyecto es:

“Crear sistema para pasar asistencia a clases en la Universidad del Bío-Bío en base al ya existente ofreciendo mejoras visuales, funcionales y de seguridad”

Específicos:

- Crear dos nuevas aplicaciones móviles que permitan pasar asistencia de manera simple y funcional, y que además ofrezca una buena experiencia de usuario al momento de ser usada.
- Reutilizar y modificar de los servicios REST ya creados para el sistema anterior.
- Crear un nuevo método de seguridad que proporcione información verídica.

3.2. Requerimientos del Sistema

Requisitos Funcionales:

- El sistema debe estar disponible para plataformas Android e iOS
- El sistema debe permitir inicio de sesión en intranet tanto para alumnos como para docentes
- El sistema debe incluir opciones de almacenamiento de asistencia vía Token y Manual
- La aplicación del profesor debe generar códigos Token
- La aplicación el profesor debe poder ver estadísticas de la asistencia de sus alumnos a sus ramos.
- La aplicación del profesor debe poder dejar una “incidencia de mala conducta” a alumnos que intenten quedar presentes cuando no lo están.
- La aplicación del profesor debe mostrar primero a los alumnos con “incidencia de mala conducta” vigentes.
- La aplicación del alumno debe poder ingresar códigos Token
- La aplicación del alumno debe poder ver estadísticas de su asistencia a clases.
- El sistema debe almacenar la lista de asistencia en intranet

Requisitos No Funcionales:

- El sistema debe ser seguro, de manera que la información entregada por este sea auténtica.
- El sistema debe ofrecer una buena experiencia de usuario.

3.3. Ambiente de ingeniería de Software

3.3.1. Desarrollo

Para el desarrollo del producto se crean las aplicaciones desde cero, reutilizando gran parte de los servicios REST ya existentes, realizando modificaciones a estos además de agregar servicios nuevos. Cabe destacar que la conexión con los sistemas de la Universidad funciona consumiendo servicios REST almacenados en los servidores de la Universidad, estos servicios contienen procedimientos almacenados proporcionados por la Universidad, los cuales realizan las consultas a la Base de Datos pertinentes. Durante el proyecto no se tienen control de los procedimientos almacenados, solo de los servicios REST que consume la aplicación.

La decisión de crear las aplicaciones desde cero se da en base a:

- **Cambios Sustanciales en los Requerimientos:** en el presente proyecto se descarta todo lo que es QR, lo cual era la base del proyecto anterior. Además se agregaran funciones referentes a estadísticas. Teniendo así una situación donde gran parte de la aplicación será descartada y otra gran parte será nueva.
- **Enfoque a la Usabilidad del producto:** dado que uno de los pilares de este proyecto es tener una interfaz “limpia e intuitiva”, como se mencionó en los requisitos no funcionales, y que se tienen grandes cambios de los requerimientos del proyecto, modificar el proyecto anterior requiere mantener la misma interfaz de las pocas funciones que se mantienen, generando así un conflicto donde se debe crear una interfaz que sea consistente a la anterior aplicación, solo por ahorrar un poco de tiempo en crear el “Menú de Inicio de Sesión”.
- **Requisitos de la Universidad:** la Universidad como organización ya posee una unidad que desarrolla sistemas, y esta cuenta con sus propios estándares de calidad que van evolucionando con el tiempo. Al momento de crearse el proyecto original, en la parte de Android, se utilizan solo “Actividades” para crear las distintas interfaces requeridas, dado que en ese momento ese era el estándar que poseía la Universidad, y esta fue una de las primeras aplicaciones móviles que se crearon, pero actualmente se sabe que hay un método mejor de programar como lo son los “Fragmentos”, y la Universidad está creando más proyectos móviles, por lo que para que este proyecto se ajuste al estándar de calidad requerido, se debe reemplazar gran parte del código anterior en pos de hacer uso de “Fragmentos”.

Presentados estos puntos, modificar la aplicación original requiere demasiado esfuerzo para solo reutilizar unas pocas interfaces, las cuales de por si deben ser adaptadas a los nuevos requisitos presentados por la Universidad. Por lo que se decidió crear todo el proyecto de Smartphone desde cero, utilizando el proyecto anterior como guía de lo que se debe hacer y reciclando solo algunas partes del código que son de utilidad y fácilmente transferible a otro proyecto, un buen ejemplo de esto son cosas como el manejo del GPS o validación de datos.

Referente a los servicios REST del proyecto, se tomó la decisión de trabajar con los existentes debido a:

- **El proyecto sigue cumpliendo el mismo objetivo:** Si bien el proyecto hace cambios con respecto al anterior, estos cambios se ven reflejados en la parte de las aplicaciones móviles, el proyecto sigue realizando la misma tarea como lo es pasar asistencia, por lo que los servicios creados para el proyecto anterior, son los mismos que se requieren para este proyecto.
- **Se pueden agregar y cambiar detalles con facilidad:** Hacer cambios en esta parte del proyecto no es tan complejo como hacer un cambio a una aplicación ya existente, esto dada la naturaleza de los servicios REST, los cuales no poseen interfaz, ni múltiples clases, solo son un medio por el cual ingresan datos y estos responden con otros datos.

Por estas razones, se reutilizan los servicios REST existentes, realizando pequeñas modificaciones a estos, destacando además que para la parte de estadísticas, se crearán nuevos servicios REST y procedimientos almacenados.

3.3.3. Herramientas Desarrollo

Programación

Para poder realizar las aplicaciones de Smartphones se debe hacer uso de herramientas de desarrollo de aplicaciones móviles oficiales tanto para Android como para iOS, siendo estas:

- **AndroidStudio:** es la herramienta oficial para el desarrollo de aplicaciones Android de forma nativa, desarrollada por Google y JetBrains, está disponible tanto para plataformas Windows, Mac como Linux [4]. Se utiliza para desarrollar la versión de Android, esto dentro de un computador de Sistema Operativo Windows 10, y las pruebas se realizan con un Smartphone Moto G5 con Android 8.0.
- **Xcode:** es la herramienta oficial de Apple para desarrollar software para macOS, watchOS, tvOS e iOS, desarrollada por Apple Inc. y publicada para macOS [5]. Se utiliza para desarrollar la versión de iOS, esto dentro de una Mac, realizando pruebas con XCode iOS Simulator [8].

Además, para todo lo referente a la interconexión entre las aplicaciones y el almacenamiento de datos en Intranet UBB, se hace uso de consultas PHP almacenadas en el servidor de Universidad. Para esto se usa:

- **Sublime Text 3:** editor de texto para poder trabajar las consultas PHP.

Diseño e Interfaz

Dado que uno de los principales objetivos del proyecto es Mejorar la Interfaz de las aplicaciones, se realizan Mockups de la aplicación con el objetivo de usarlos como guía al momento de crear las aplicaciones. Estos Mockups se realizaron con el software:

- **Balsamiq Mockups 3:** software dedicado a la creación de Mockups.

Además, cabe destacar que en este aspecto se solicitó apoyo estudiantes de Diseño Gráfico para asegurar que la interfaz sea intuitiva y ordenada.

Pruebas

Dado que se trabaja con Servicios REST, siendo las aplicaciones consumidoras de estos, estos deben ser testeados para saber que entregan la información requerida, por lo que para realizar estos test se hizo uso del programa

- **POSTMAN:** software utilizado para realizar pruebas a servicios REST vía sus distintos métodos, siendo en este caso el más usado “POST”.

3.3.3. Metodología de Trabajo

Al momento de trabajar se opta por el “Desarrollo Basado en Funcionalidades” (FDD) [11], donde se fueron incorporando las funciones una por una, al mismo tiempo que se incorporaba la interfaz y los controladores.

Se opta por esta metodología dado que al momento de implementar algunas funciones, se requiere de previa investigación, por lo que cada vez que se incorpora algo nuevo al proyecto, se investiga lo referente a esa función. Esto permite que el proceso de aprendizaje requerido para realizar el proyecto no sea una investigación previa al proyecto de todas las tecnologías, sino que un proceso que va de la mano a lo que se busca implementar, donde cada nueva función requiere investigar una nueva tecnología. Además dado que no se posee vasta experiencia trabajando con cosas como GPS, servicios REST, Fragmentos, entre otros, el trabajar con FDD permite que cada vez que se aprende como trabajar con estas tecnologías, se pudiera probar, de manera que el proyecto siempre iba funcionando, sin tener errores de programación fatales.

Además referente al proceso de Desarrollo, se opta por crear el producto en la plataforma Android, dado que se cuenta con conocimientos previos trabajando en la plataforma. Una vez finalizado el proyecto completamente en Android, se realiza la versión de iOS, la cual vendría siendo un porteo de la versión Android.

Las funciones se realizan en el siguiente Orden:

- Inicio de Sesión, Aplicación Docente y Alumno, Android
- Menú Principal, Aplicación Docente y Alumno, Android
- Abrir un Curso para pasar asistencia, Aplicación Docente, Android
- Ver Lista de Alumnos, Aplicación Docente, Android
- Creación de Servicios REST propios para pruebas
- Conexión con servicios REST propios
- Implementación de Fragmentos, Android
- Implementación de Controladores, Android
- Implementación de Mensajería FireBase, Android
- Implementación de Estadísticas, Aplicación Docente y Alumno, Android
- Conexión con el Servidor de la Universidad, Android
- Port iOS

3.3.4. Arquitectura de Software

El patrón de arquitectura utilizado es el Modelo-Vista-Controlador (MVC) [12] quien utiliza el paradigma de la Programación Orientada a Objetos, junto a un modelo Cliente-Servidor.

Por un lado el Servidor funciona por medio de servicios REST los cuales son recursos almacenados en los servidores de la Universidad los cuales son consumidos por los Clientes, estos recursos cumplen la función de almacenar la información en la Base de Datos de la Universidad y entregar esta información a las aplicaciones

Por el lado del Cliente se tienen las aplicaciones, que sirven como interfaz o medio para poder pasar asistencia, el cliente se presenta en forma de las aplicaciones las cuales consumen los servicios REST del Servidor. Las aplicaciones usan MVC, donde se separa todo lo que es el modelo de datos, la interfaz del usuario y los distintos controladores del sistema de manera que estos tres componentes interactúen entre sí para cumplir su objetivo. Cabe destacar que dada la naturaleza de las aplicaciones móviles, todo lo referente a Vista posee sus propios controladores de comportamiento junto a los distintos layouts de la aplicación. Y lo referente a los Controladores se ve reflejado en conexiones con el servidor y control del proceso de asistencia.

Se eligió este patrón dada la facilidad que otorga al trabajar en distintas partes del proyecto, permitiendo modificar partes del proyecto fácilmente sin sacrificar que otras dejen de funcionar. Además este patrón permite que el software sea fácilmente mantenible a futuro dado que entrega una visión clara de lo que hace cada parte del código.

3.3.5. Tecnologías Usadas

Java

Java [16] es un popular lenguaje de programación orientado a objetos desarrollado por Sun Microsystems en 1995. Se utiliza para crear la versión de Android del software, siendo este compatible con AndroidStudio [4].

Swift 4

Swift 4 [15] es un lenguaje de programación multiparadigma desarrollado por Apple y enfocado en el desarrollo de aplicaciones para iOS y macOS. Es el lenguaje requerido para crear la versión de iOS, dado que es el principal lenguaje usado por el software XCode actualmente [14].

XML

XML [18] es un meta-lenguaje que permite almacenar lenguajes de marcas, fue desarrollado por el World Wide Web Consortium. Es utilizado para hacer layouts en el software AndroidStudio [17].

PHP

PHP [19] es un lenguaje de programación de código abierto creado por Rasmus Lerdorf en 1995 y utilizado para hacer scripts en el lado del servidor. Es utilizado para programar los servicios REST almacenados en el servidor y consumidos por las aplicaciones [20].

3.3.6. Pruebas

Para cerciorarse que la función principal del sistema, pasar lista, funcione bien, se realizan sesiones de pruebas en las distintas salas de clases, de manera que se compruebe que el sistema cumpla sus funciones correctamente, y sea fácil de usar para los usuarios finales.

4. CARACTERIZACIÓN DEL SOFTWARE

4.1 Alcance del Proyecto

La función principal del producto es entregar una herramienta que automatice el proceso de asistencia almacenándolo en la plataforma intranet directamente, y que además muestre las estadísticas de asistencia a clases.

El producto está disponible para Smartphones Android, a partir de su versión 5.0 Lollipop, e iOS desde su versión iOS 10, consistiendo de la aplicación para el Docente versión Android e iOS, y la aplicación para el Alumno Android e iOS.

El producto cuenta con dos perfiles principales, Docente y Estudiante, siendo el Docente la persona encargada de realizar la clase y el Estudiante todo perteneciente a un curso. El acceso a estos perfiles se da desde sus aplicaciones respectivas.

4.2 Objetivo del Producto

Objetivo General

Almacenar la asistencia a clases de cada asignatura, de cada carrera de la Universidad bajo los estándares establecidos por esta, siendo estos que la información sea autentica y que quede almacenada en la plataforma Intranet. Además de presentar un medio para ver estadísticas de asistencia a clases.

Objetivo Específico

- Almacenar la asistencia en Intranet
- Gestionar proceso de asistencia
- Mostrar estadísticas de asistencia a clases

4.3 Descripción Global del Producto

El producto consta en dos aplicaciones móviles, una para docentes y otra para alumnos, disponibles cada una para Smartphones Android e iOS, además de un conjunto de consultas PHP que contienen los servicios REST para que estas funcionen.

4.3.1 Interfaz de Usuario

La interfaz consta con opciones simples con títulos descriptivos, de manera que los usuarios puedan navegar fácilmente por el sistema teniendo una clara idea de donde están actualmente y que está ocurriendo. Para el proceso de pasar asistencia se hace uso de etiquetas en forma de “TAGs” las cuales cumplen el objetivo de indicar el estado actual de cada alumno, estos TAGs poseen un color representativo de manera que den una clara idea de lo que representan.

4.3.2 Interfaz de Hardware

Como Hardware requerido para que el sistema funcione se tiene al Smartphone, dispositivo de alta popularidad y uso común en la sociedad actual, del cual dispone prácticamente todo el alumnado y profesores. Cabe destacar que el Smartphone debe ser de marca Apple con Sistema Operativo iOS, o de cualquier otra marca que use el Sistema Operativo Android.

4.3.3 Interfaz de Software

Para el uso del sistema se requiere tener instalada la correspondiente aplicación en sistemas Android o iOS, el sistema funciona de forma nativa dentro de estos Sistemas Operativos Móviles y se presenta como una Aplicación dentro del menú principal del Smartphone.

Además se requiere tener almacenados los servicios REST necesarios para que el sistema funcione dentro de los servidores de la Universidad.

4.4. Requerimientos del Sistema

Smartphone en General

Para el correcto funcionamiento de las aplicaciones se requiere hacer uso de distintas funcionalidades del Smartphone, siendo estas:

- GPS del dispositivo: Con objetivo de capturar la localización geográfica tanto del docente como de los estudiantes, de manera de saber si estos están cerca del docente o no.
- Conexión a Internet: Con motivo de poder realizar las consultas al servidor pertinentes.

Versión Android

Las aplicaciones de Smartphones Android están disponibles en la Google PlayStore [22], requieren de Android 5.0 Lollipop [26] como mínimo, esto debido a que más del 80% del mercado posee Android 5.0 o superior [24] a la fecha diciembre del 2018, además se requiere 5MB de almacenamiento en el dispositivo para aplicación del Docente y 4MB para la aplicación del alumno.

Versión iOS

Las aplicaciones para Smartphones iOS, iPhones, están disponibles en la Apple APPStore [21], requieren de iOS 10 [25] como mínimo, esto debido a que más del 97% del mercado posee iOS 10 o superior [23] a la fecha diciembre del 2018, además se requiere 5MB de almacenamiento en el dispositivo para aplicación del Docente y 4MB para la aplicación del alumno.

5. ESTUDIO DE FACTIBILIDAD

5.1. Factibilidad Técnica

Para evaluar este punto hay que decir que en el mercado de los Smartphones hay dos grandes competidores, por un lado está Apple con un 21,77% del mercado, con sus dispositivos iPhone, los cuales funcionan con el sistema operativo iOS, y por el otro lado esta Android, sistema operativo creado por Android Inc. y actualmente propiedad de Google que posee un 77,42% del mercado, a diferencia de iOS, Android no es exclusivo de una compañía, sino que Smartphones de distintas marcas, lo usan [1 a 3]. Dicho esto a continuación se evaluarán técnicamente ambas plataformas.

Android

Para trabajar con Android se usa el software gratuito “AndroidStudio” [4], el cual funciona en lenguaje Java y está disponible para distintos sistemas operativos como variantes de Linux o Windows.

Respecto al desarrollo para Android, se puede decir que la mayor dificultad al momento de realizar un proyecto como este es la vinculación con un sistema ya existente, siendo en este caso Intranet de la Universidad, pero como solo se trabaja en mejoras a la interfaz y nuevas funciones, esto no es un problema.

Cabe destacar que para publicar la aplicación en la “Google Play Store” [22], tienda oficial de dispositivos Android, se requiere una cuenta en “Google Developers”, la cual requiere una cuota de registro de \$25 Dólares, esta cuota no se considera para la factibilidad, dado que la Universidad ya posee una cuenta de “Google Developers” [6 y 7].

iOS

Al momento de trabajar en la versión de iOS, solo se puede trabajar en una computadora Mac, y utilizar el software gratuito “Xcode” [5], el cual funciona con lenguaje Swift y Swift 2.0.

Si bien esto es una problemática y una dificultad para el proyecto, hay otro problema referente a las aplicaciones en iOS, y es que para publicarlas en la “APP Store” [21], tienda exclusiva y oficial de dispositivos iOS, se requiere una licencia en el “iOS Developer Enterprise Program” [9], la cual cuesta \$299 Dólares, además de una licencia por el “Apple Developer Program” [10] la cual cuesta \$99 Dólares, y es para publicar la aplicación, esta última se paga anualmente. La Universidad actualmente posee una licencia de “iOS Developer Enterprise Program”, por lo que este gasto no se considera para el proyecto, lo que sí es considerado son los costos del “Apple Developer Program”, los cuales deberá pagar la Universidad anualmente para mantener el sistema vigente.

Conclusión

Podemos decir que el proyecto es realizable técnicamente, no supone algo imposible de hacer, sobre todo desde el punto de vista de que son mejoras para ambas versiones, el único y gran inconveniente es referente a las restricciones que presenta Apple al momento de desarrollar aplicaciones para su plataforma.

5.2. Factibilidad Operativa

Para poder realizar la factibilidad operativa hay que destacar que el sistema es factible dado que:

- Existe una extrema popularidad de los Smartphones en la sociedad actual.
- Gran parte de los Usuarios saben utilizar correctamente los Smartphones.
- La tarea que realiza supone una ventaja para el docente.
- No se requiere que todos los alumnos posean la aplicación, el docente los puede marcar manualmente.

Esos puntos se resumen como, existe una necesidad, mejorar el proceso de asistencia, y existe un medio extremadamente popular y de fácil uso el cual satisface esta necesidad.

Los principales peros que presenta un sistema de este tipo son:

- **Experiencia de Usuario:** si el sistema no es amigable para el usuario al momento de presentar sus funciones, esto produce que el usuario no lo use. Debe presentarse una mejora sustancial comparada a la forma actual de pasar asistencia si se busca que el sistema sea usado.
- **Autenticidad de los Datos:** si los datos otorgados por el sistema, no son auténticos, o sea, no representan la cantidad real de alumnos presentes en la sala, el sistema será descartado por la Universidad. El sistema debe presentar una forma de que se garantice la autenticidad de la información.

El sistema cubre estas falencias, presentando una interfaz intuitiva y presentando una fácil forma de uso, en forma de los que son los “Tokens”, además, se enfatiza en lo que es presentar información autentica, esto en forma de validación por GPS y Token, además de que no se hará todo automático sino que se requiere verificación previa del docente.

Conclusión

Se puede decir que el sistema satisface una necesidad con medios altamente accesibles para los usuarios, y las grandes falencias que se pueden presentar son cubiertas satisfactoriamente, por lo que no hay razón para que el presente proyecto no sea factible operativamente.

5.3. Factibilidad Económica

Para realizar la Factibilidad Económica se evalúan los costos y beneficios del proyecto, para así calcular el VAN y saber si el proyecto es rentable o no.

Costos de Inversión

El costo asociado al proyecto es:

- Horas de Desarrollo

En cuanto al desarrollo del proyecto se estiman los siguientes datos:

	Horas Estimadas	Valor por Hora	Totales
Análisis del Código Original	20	\$7.000	\$140.000
Creación de versión Android	50	\$5.000	\$250.000
Creación de versión iOS	60	\$5.000	\$300.000
Pruebas a las versiones	20	\$7.000	\$140.000
	Total		\$830.000

Tabla 1: Costos de Inversión en Horas Desarrollo

- Licencias Android/iOS

Con respecto a las los registros para las plataformas Play Store y APP Store, como se mencionó en la factibilidad técnica, los costos son los siguientes:

	Valor en Dólares	Totales
Google Developers	25	\$17.250
iOS Developer Enterprise Program	299	\$206.310
	Total	\$223.560

Tabla 2: Costos de Inversión en Licencias Android e iOS

- Utensilios Desarrollo

Para poder desarrollar el sistema se requiere de distintos equipos tanto para programar como para testear el producto. Los equipos y sus costos son los siguientes:

	Cantidad	Totales
Mac Air	1	\$649.000
Smartphone Moto G5	1	\$80.000
Smartphone iPhone SE	1	\$219.000
Total		\$948.000

Tabla 3: Costos de Inversión en Utensilios de Desarrollo

Por lo que el costo de inversión del proyecto es **\$2.001.560**

Costos de Operación

Los costos de Operación relacionados al proyecto son:

	Valor en Dólares	Totales	Tiempo
Apple Developer Program	99	\$68.310	Anual
Total		\$68.310	

Tabla 4: Costos de Operación del Proyecto

Por lo que el costo de operación del proyecto es **\$68.310 anualmente**

Beneficios

El Beneficio económico del proyecto es:

- Tiempo extra de los docentes al utilizar el sistema
 Para calcular el beneficio se la siguiente información, la cual es una media de lo que se puede demorar un profesor, que son 2 min. para clases de 15 alumnos y profesores que se manejan con la plataforma a 10 min. para clases de 40 alumnos donde los docentes no se manejan.

Sueldo Profesor	\$900.000
Valor 5 Min	\$470

Tabla 5: Sueldo Aproximado de un Profesor

A partir de estos datos se puede calcular la siguiente tabla de ganancias anuales, la cual depende del número de clases semanales que se realicen y cuantos docentes usen el sistema:

		Clases Semanales		
		1	2	3
Nro de Docentes que lo Usan	1	\$18.750	\$37.500	\$56.250
	2	\$37.500	\$75.000	\$112.500
	3	\$56.250	\$112.500	\$168.750
	4	\$75.000	\$150.000	\$225.000
	5	\$93.750	\$187.500	\$281.250
	6	\$112.500	\$225.000	\$337.500
	7	\$131.250	\$262.500	\$393.750
	8	\$150.000	\$300.000	\$450.000
	9	\$168.750	\$337.500	\$506.250
	10	\$187.500	\$375.000	\$562.500
	11	\$206.250	\$412.500	\$618.750
	12	\$225.000	\$450.000	\$675.000
	13	\$243.750	\$487.500	\$731.250
	14	\$262.500	\$525.000	\$787.500
	15	\$281.250	\$562.500	\$843.750
	16	\$300.000	\$600.000	\$900.000
	17	\$318.750	\$637.500	\$956.250
	18	\$337.500	\$675.000	\$1.012.500
	19	\$356.250	\$712.500	\$1.068.750
	20	\$375.000	\$750.000	\$1.125.000

Tabla 6: Beneficio anual con respecto a las clases semanales y el número de usuarios

VAN

Para motivos del VAN se tomó como beneficio el valor que nace a partir de que 11, 12, 13, 14 Y 15 docentes usen el sistema 3 veces a la semana el 1er, 2do, 3er, 4to y 5to año respectivamente. El beneficio en este caso son: \$618.750 para el 1er año, \$675.000 para el 2do año, \$731.250 para el 3er año, \$787.500 para el 4to año y \$843.750 para el 5to año. Además se usara una tasa de 10%.

Año	0	1	2	3	4	5
Beneficios		\$618.750	\$675.000	\$731.250	\$787.500	\$843.750
Costo Operación		\$68.310	\$68.310	\$68.310	\$68.310	\$68.310
Costo Inversión	\$2.001.560					
Flujos		\$500.400	\$501.397	\$498.077	\$491.216	\$481.487
VAN 10%	\$471.017					

Tabla 7: Calculo VAN

Al calcular el VAN a 5 años con 10% de tasa de interés se obtiene **\$471.017**

Con el VAN presentado se tiene que la TIR es **%18,2**

Cabe destacar que desde un beneficio sobre \$618.000 el proyecto ya es rentable, por lo que se requiere que al menos 11 profesores con 3 ramos a la semana usen a la aplicación para que el VAN sea positivo. También decir que mientras más docentes la usen, el VAN crece velozmente, por ejemplo, si el 1er año parten usándola 16 profesores con 3 ramos a la semana, el VAN seria mayor a \$1.000.000, por lo que el VAN depende netamente de que tanto uso se le dé a la aplicación.

Conclusión

El proyecto resulta rentable bajo un mínimo número de docentes que lo usen y cuantas clases realizan a la semana, una vez superado este mínimo los beneficios del proyecto ascienden sustancialmente, lo que se refleja claramente en el VAN y la TIR.

6. MODELAMIENTO DEL SISTEMA

6.1. Análisis y Diseño de la Solución

6.1.1. Funcionalidad

Aplicación Original

Respecto a las aplicaciones actuales se puede detallar que la aplicación del Docente permite crear un curso y generar un QR, y la aplicación del Alumno solo escanea el QR. Ambas aplicaciones permiten hacer Inicio de Sesión en los sistemas de la UBB y hacen gestiones pertinentes de seguridad a través de GPS.

De las versiones originales se puede rescatar todo código referente a consultas para conectarse con los sistemas de la UBB, ya sea: Inicio de Sesión y almacenar la asistencia en Intranet UBB.

Se puede resumir lo que hace la aplicación original tal como se presenta en la Figura 1:

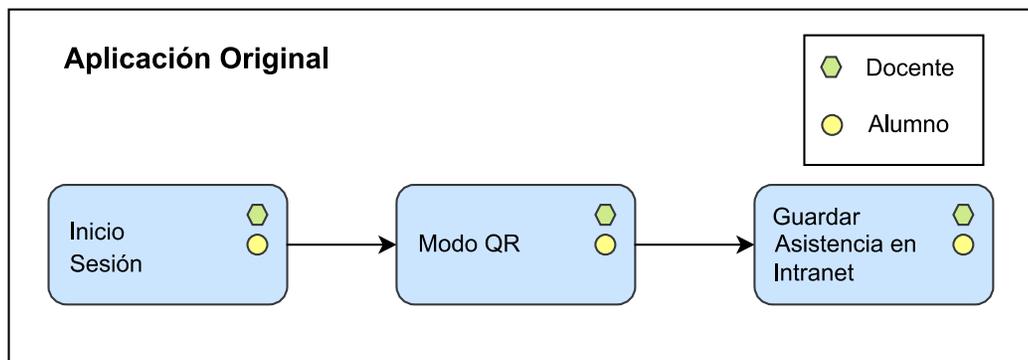


Figura 1: Diagrama con funciones de la Aplicación Original, elaboración propia

Nuevas Aplicaciones

Las nuevas aplicaciones dispondrán de una variedad de funciones y reemplaza el modo QR por nueva manera de pasar lista, siendo esta el modo Token. Esta decisión se toma dada la poca amabilidad con el usuario que posee esta opción, y los requerimientos que conlleva, dado que aunque se modifique, seguiría exigiendo tener un medio para mostrar el QR. Por lo que, si se posee una opción más simple y amigable al usuario, como lo es un Token, no hay razón para mantener la opción de QR.

Además de esto se implementara una opción para ver las estadísticas del curso, de manera que el profesor tenga todo lo referente a asistencia en un solo lugar, y no deba estar calculando las estadísticas manualmente.

Para entender bien lo que se tenía y lo que se tiene ahora, en la Figura 2 se presenta un diagrama de las funcionalidades que posee la nueva aplicación resaltando las que venían de la aplicación actual/original.

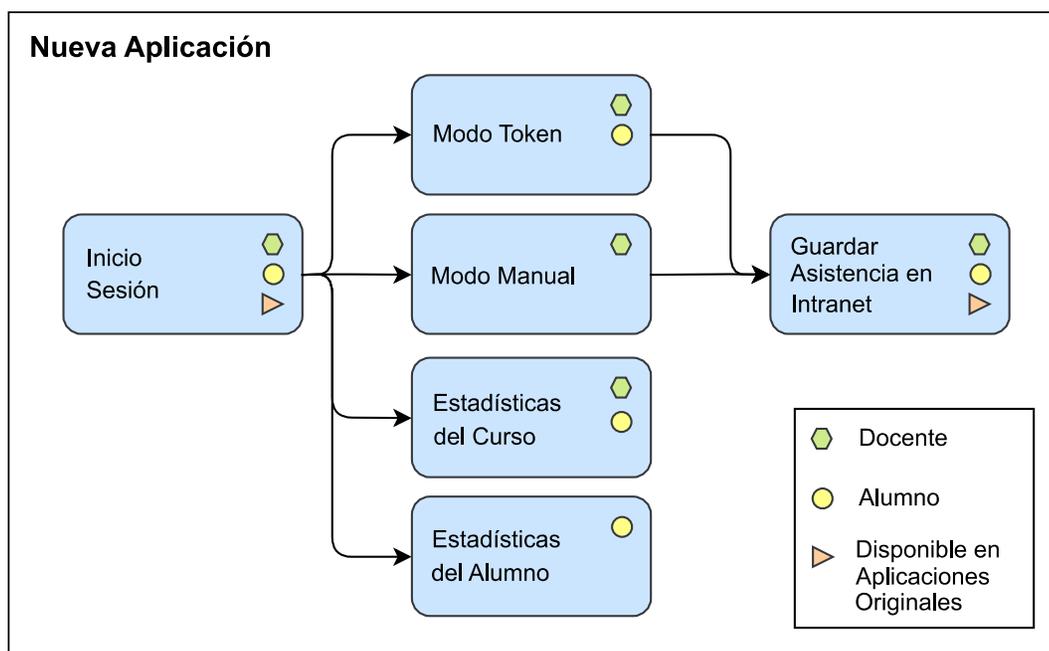


Figura 2: Diagrama con funciones de la Nueva Aplicación, elaboración propia

6.1.2. Seguridad

Por qué se requiere de Seguridad?

La información generada por una lista de asistencia se utiliza dentro de la Universidad con varios fines, siendo los más importantes lo que es una “evaluación docente” y calcular datos estadísticos. Para poder efectuar estas tareas de manera correcta la información de asistencia debe ser verídica y no adulterada, no debe poseer datos erróneos, por lo que todo sistema de asistencia debe garantizar que los presentes sean efectivamente los que están en la sala a la hora de clases, evitando que alumnos que no asisten a clases queden presentes.

Al momento de crear un nuevo sistema que sirva como medio a pasar asistencia se debe tener siempre en consideración este aspecto, ya que en el momento que los datos no sean auténticos, la Universidad simplemente desechará el sistema, dado que la información que este otorga no sirve para obtener la información relevante que se genera a través de una lista de asistencia.

Para poder garantizar que la información sea efectivamente auténtica, el sistema debe poseer medidas de seguridad que aseguren que los estudiantes estén efectivamente en la sala y, que alumnos que no estén en la sala, sino en cualquier otro lugar, no puedan registrarse presentes. También, si bien la seguridad es importante, esta debe presentar un balance entre lo que es seguridad y usabilidad, no sirve tener un sistema que sea 100% seguro y totalmente invulnerable, si requiere que el usuario realice muchas verificaciones o que este realice acciones que no desea, la solución debe ser más usable que el sistema actual e igual o más seguro.

Generación de Autenticidad de Datos

Como medida de seguridad el sistema utiliza un sistema de llave y GPS, donde la llave viene a ser el Token que da el profesor, y el GPS es la ubicación del estudiante en el momento de la clase. Dado esto, para que un alumno quede presente, se requiere estrictamente que la posición GPS del teléfono marque que esté cercano al docente, además de que este disponga del código Token que el profesor ha dado. Garantizando así que cuando estos dos elementos funcionen correctamente, el alumno este efectivamente dentro de la sala y la información sea confiable.

Los problemas de este método de seguridad es que un Token es fácilmente transmisible de un alumno a otro, basta con que un alumno le envíe el Token a otro para que se pierda toda la seguridad que este ofrece. Quedando así como único medio de seguridad la ubicación GPS, la cual es fácilmente alterable con aplicaciones móviles externas. Teniendo así que los alumnos están a un “mensaje de un amigo” y “abrir una app que cambie mi ubicación”, de quedar presentes en un lugar externo a la sala, teniendo como consecuencia de que la información no sea verídica.

Dado que las medidas de seguridad son fácilmente vulnerables, para que la información colectada por el sistema sea “Auténtica”, hay que dejar de pensar del sistema como un sistema automático, y verlo como un sistema semi automático que requiere la gestión del profesor para garantizar que los datos sean verídicos.

La aplicación muestra el número de alumnos que posee el curso, y muestra tanto los ausentes como presentes a tiempo real, finalizada la clase, los profesores deberán revisar que los datos coincidan, contar que los números coincidan es mucho más simple que tener que pasar toda la lista a la plataforma, y en caso de que no coincidan, será un problema las primeras veces, pero una vez que se identifique a las personas que tratan de vulnerar el sistema, es fácil dejarlos ausentes. El profesor tiene el poder sobre quienes quedan presentes y quiénes no. De esta manera, el sistema cumple su función principal de facilitar el proceso de registro de asistencia, y con la gestión del docente se genera información verídica, cumpliendo así sus dos funciones más importantes.

6.1.3. Funcionamiento del Sistema

La aplicación del Docente cumple el objetivo de gestionar todo el proceso de asistencia, por lo que esta aplicación crea la clase por medio de una “lista virtual” con la cual se gestiona a los alumnos presentes recibiendo mensajes desde la aplicación del Alumno

La aplicación del Estudiante consiste en una aplicación que en base a las clases creadas por el profesor, puede mandar una notificación de presente al servidor PHP, quien notifica a la aplicación del Profesor vía mensajería FireBase.

Además de esto, la comunicación entre las aplicaciones se realiza a través de un servidor PHP el cual se comunica con la Base de Datos de la Universidad, todo esto se ubica en los servidores de la Universidad, con el objetivo de no almacenar consultas a la Base de Datos en las aplicaciones. El servidor PHP posee notifica a las aplicaciones sobre el estado del proceso actual de asistencia vía mensajería FireBase. Para más detalles véase la Figura 3.

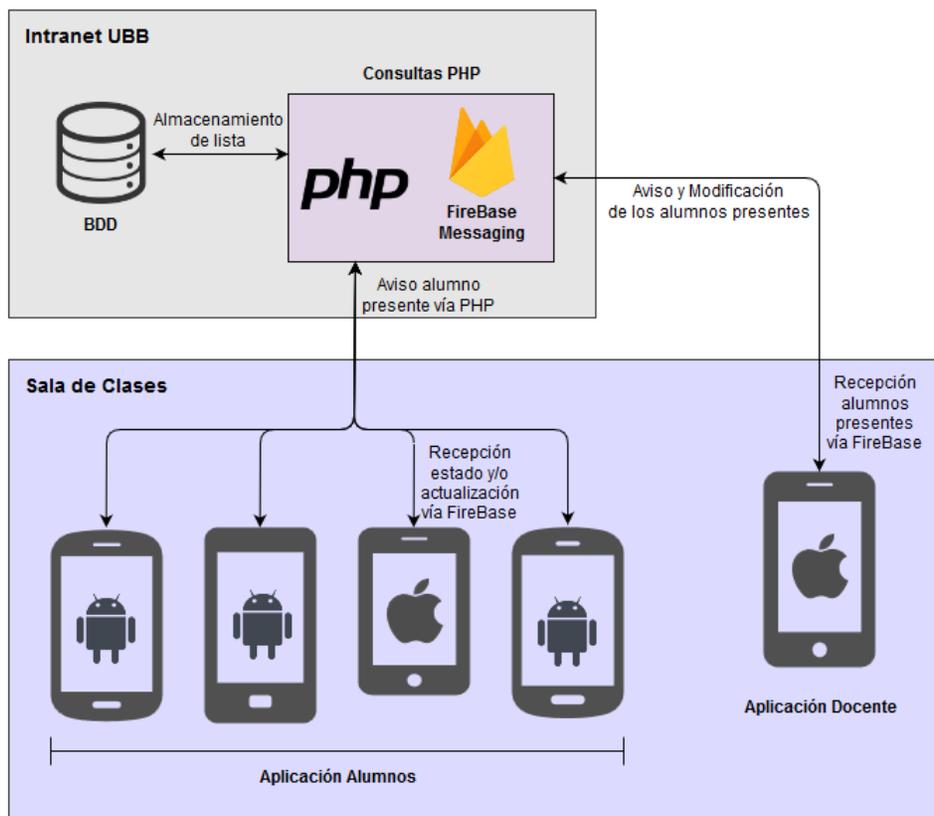


Figura 3: Funcionamiento del Sistema, elaboración propia

6.1.4. Diseño de la Solución

Para poder hacer un producto que se ajuste a los usuarios, durante todo el proceso de desarrollo se fue recopilando información de entrevistas anónimas a distintos docentes de la Universidad, las cuales permitieron ver las necesidades de los docentes e ir ajustando el producto a sus requerimientos. Dentro de lo que más se resalta de estas entrevistas, es la gran aceptación al proyecto que tienen los docentes y las propuestas de mejoras dadas por ellos como lo son el tema de las estadísticas.

Durante el proceso de diseño del producto se trabajó directamente con el Departamento de Desarrollo de Sistemas de la Universidad, de manera que se pueda hacer un producto que se ajuste tanto a los requerimientos de la Universidad como de los Usuarios, descartando y agregando ideas durante el proceso de desarrollo para así obtener el mejor producto posible.

6.2. Modelos de Datos

6.2.1. Modelo Entidad Relación (MER)

Al momento de crear una aplicación que se acopla a un sistema ya existente como lo es la intranet UBB, se trabaja con un Modelo Entidad Relación (MER) ya definido, por lo que el modelo de datos del sistema se acopla a lo ya existente, tomando datos y agregando datos al MER institucional. Dada esta situación, al momento de crear un MER para el sistema (véase Figura 4) se consideraron a rasgos generales las entidades y relaciones que se requieren para que un sistema de este tipo funcione correctamente, muchas de estos datos ya existen dentro del MER de la Universidad, y otras como lo son el Token de 6 dígitos y el Token de FireBase, los cuales no existían dentro del Modelos de Datos de la Universidad antes de crear el proyecto.

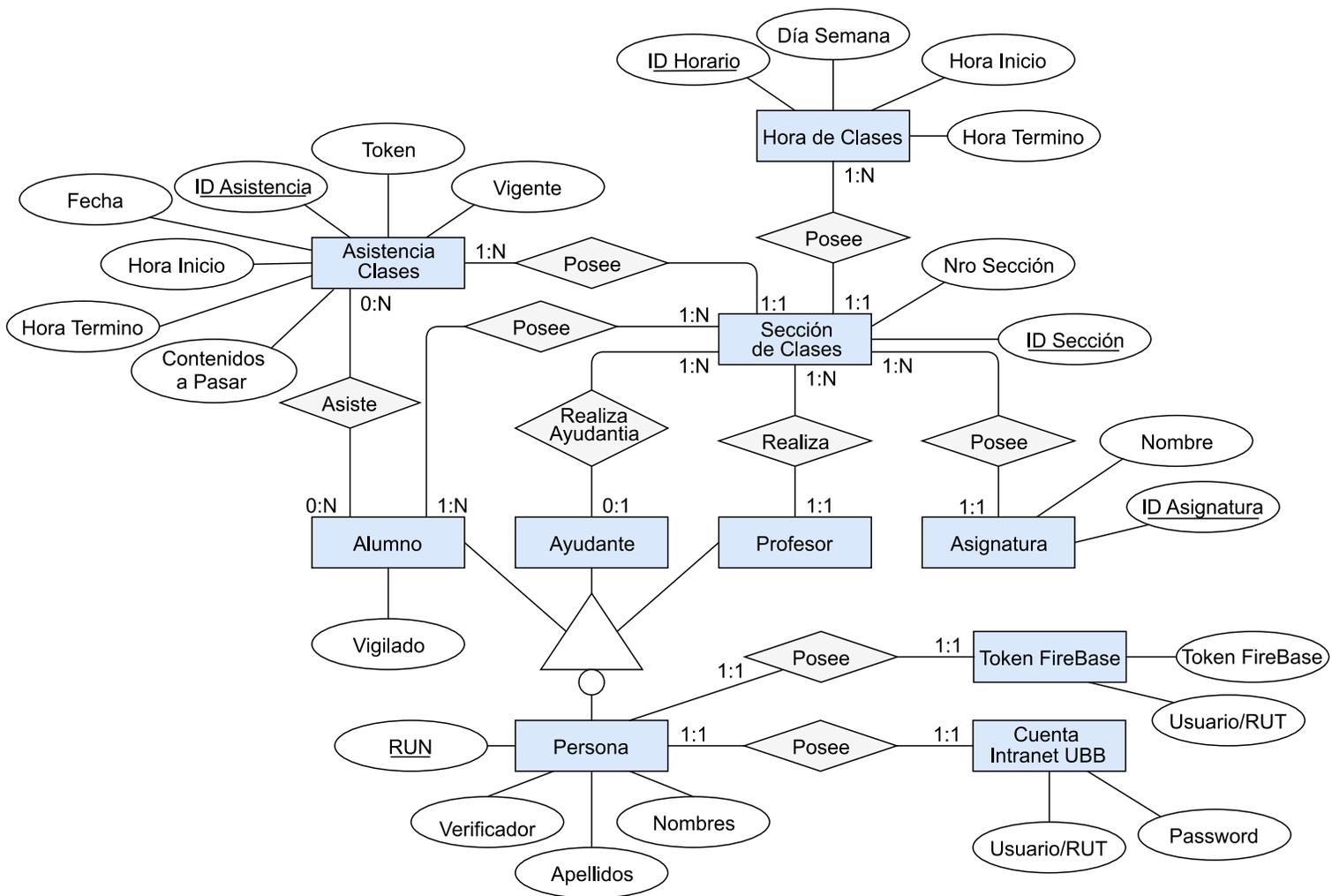


Figura 4: Modelo Entidad Relación

6.2.2. Entidades

- **Alumno:** Son los estudiantes que asisten a clases, siendo más específico, toman una “Asignatura” y asisten a una “Sección de Clases”. Su asistencia se almacena en “Asistencia a Clases”. Alumno solo almacena el campo “Vigilado”, que es una marca que indica que trato de quedar presente cuando no lo estaba, el resto de sus atributos los hereda de “Persona”.
- **Asignatura:** Es una asignatura que imparte la Universidad, posee una o más secciones de clases.
- **Asistencia Clases:** Es la lista que se almacena por cada día de clase impartida.
- **Ayudante:** Es una persona puede ser tanto alumno como profesor, esta entidad existe con el objetivo de indicar una segunda persona que impartirá clases en un ramo.
- **Cuenta Intranet UBB:** Es la cuenta que se le asigna a todo profesor y alumno de la UBB para acceder a la plataforma institucional Intranet UBB.
- **Hora de Clases:** Indica el día de la semana y la hora en la que se debe realizar una clase.
- **Persona:** Se usa para generalizar las entidades alumno, ayudante y profesor. Contiene los datos esenciales de una persona, como RUT, Nombre y Apellido.
- **Profesor:** Son los encargados de realizar las clases, son generalizados de persona.
- **Sección de Clases:** Es una división de la Asignatura con motivos de que muchos alumnos tomaron la Asignatura, dado que siempre se aplica, puede existir una única sección de clases.
- **Token FireBase:** Es el Token que genera FireBase por Smartphone al momento de instalar la Aplicación, sirve para poder mandar mensajería a las aplicaciones desde PHP.

6.2.3. Relaciones

- **Asiste(Alumno – Asistencia a Clases):** Indica que un alumno puede asistir a una o muchas clases, y que a una clase pueden asistir uno o más alumnos. Cabe destacar que el hecho que un alumno esté inscrito en el ramo no quiere decir que asista a este, por eso el alumno “puede asistir”.
- **Posee(Alumno – Sección de Clases):** Son los alumnos los cuales están inscritos en un ramo.
- **Posee(Asistencia a Clases – Sección de Clases):** Indica que una sección de clases posee uno o más alumno y que un alumno tiene una o más clases.
- **Posee(Persona – Cuenta Intranet UBB):** Indica que todas las personas, en este caso “Alumno”, “Ayudante” y “Profesor”, poseen una cuenta de acceso a la Intranet UBB y que esta cuenta de acceso pertenece a una única persona.
- **Posee(Persona – Cuenta Intranet UBB):** Indica que todas las personas que usen el sistema, en este caso “Alumno”, “Ayudante” y “Profesor”, poseen un token de FireBase.
- **Posee(Sección de Clases – Asignatura):** Indica el nro de secciones que posee una Asignatura.
- **Posee(Sección de Clases – Hora de Clases):** Indica que una clase puede poseer una o más horas de clases, y una hora de Clases pertenece a una única clase.
- **Realiza(Profesor – Sección de Clases):** Indica que sección de clases posee un profesor y que este profesor puede impartir una o más asignaturas.
- **Realiza Ayudantía(Ayudante – Sección de Clases):** Indica que toda clase puede poseer un ayudante y que este ayudante puede impartir una o muchas ayudantías.

6.3. Conexión con la Base de Datos

Dado que la Universidad cuenta con una Base de Datos creada y funcional, al momento de trabajar con todo lo referente a almacenar y obtener información de esta, se utilizan procedimientos almacenados en el servidor de la Universidad, los cuales se utilizan desde un servicio REST almacenado los servidores de la Universidad, este servicio es consumido por las aplicaciones móviles, sin existir contacto directo entre las aplicaciones y la Base de Datos institucional, manteniendo así una capa de seguridad con respecto a toda consulta realizada a la Base de Datos. Estos procedimientos son creados por la sección de “Modelado de Datos” de la Universidad, en base a los requerimientos del proyecto, por lo que en el proyecto solo se trabajan las aplicaciones y los servicios REST, dejando todo lo referente a las consultas a la Base de Datos Institucional en manos de la Universidad.

Como se mencionó previamente, tanto los servicios REST, como los procedimientos almacenados serán reutilizados del proyecto anterior. Dado que se han realizado cambios en los requerimientos del proyecto algunos procedimientos han debido ser alterados y otros creados desde cero, lo cual fue solicitado al área de “Modelado de Datos” de la Universidad.

6.5. Diseño de Interfaz y Navegación

6.5.1. Aplicación Docente

Esta Aplicación es la más compleja, ya que debe permitir abrir un curso en distintos modos, manual y Token, mostrar una lista de estudiantes la cual tendrá una distinta interfaz según el modo seleccionado y además mostrar estadísticas generales de asistencia de los cursos que realice el profesor, además de los estudiantes pertenecientes a estos.

Inicio de Sesión

En la Figura 5 se puede ver el Inicio de Sesión, el cual es una interfaz simple que le permite al docente acceder a su cuenta de Intranet UBB, esta recibe como datos el RUT del docente y la clave de intranet. En caso de error en el ingreso de datos, o error de conexión con el servidor, esta interfaz permite mostrar un mensaje de error.

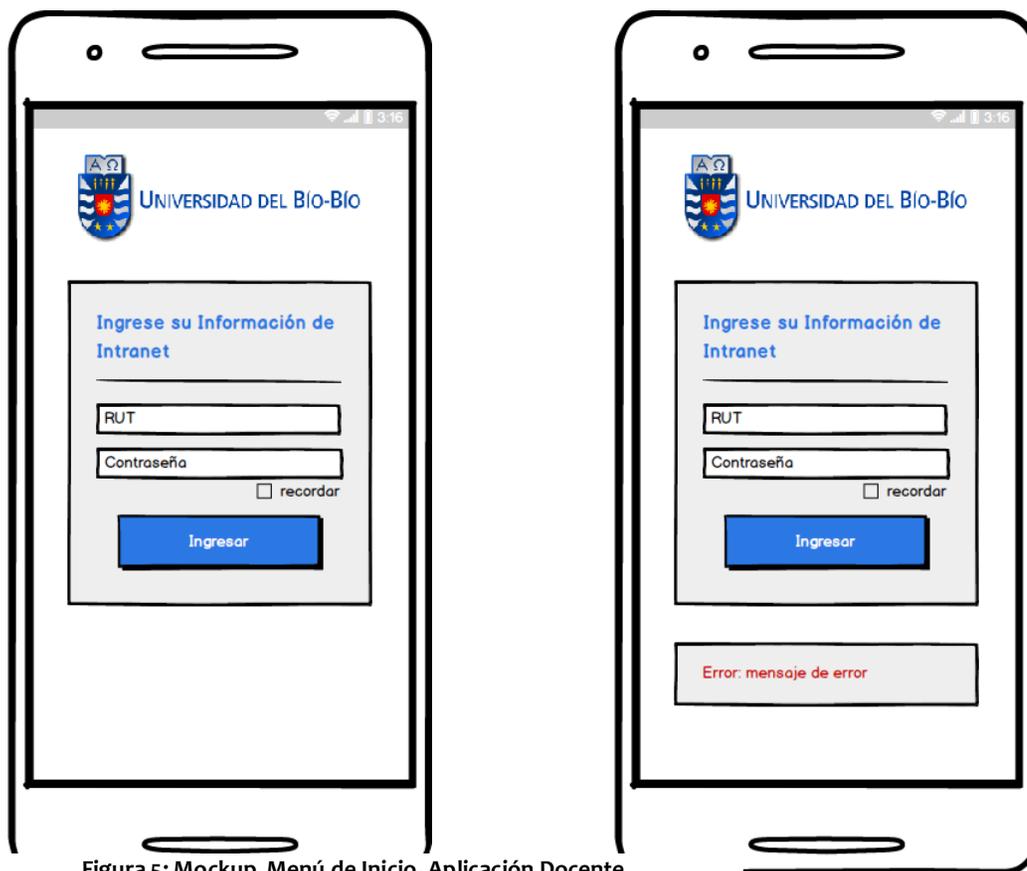


Figura 5: Mockup Menú de Inicio, Aplicación Docente

Menú Principal

En la Figura 6 se presenta la interfaz que sirve de menú principal y posee todos los accesos a las opciones de la aplicación, registrar asistencia, recuperar clase y ver estadísticas.

Además, cuenta con una variante para cuando se está registrando asistencia, en esta variante se deshabilitan las opciones de registro de asistencia y se agrega un acceso a lista que se está pasando actualmente y un botón para cerrar la lista.

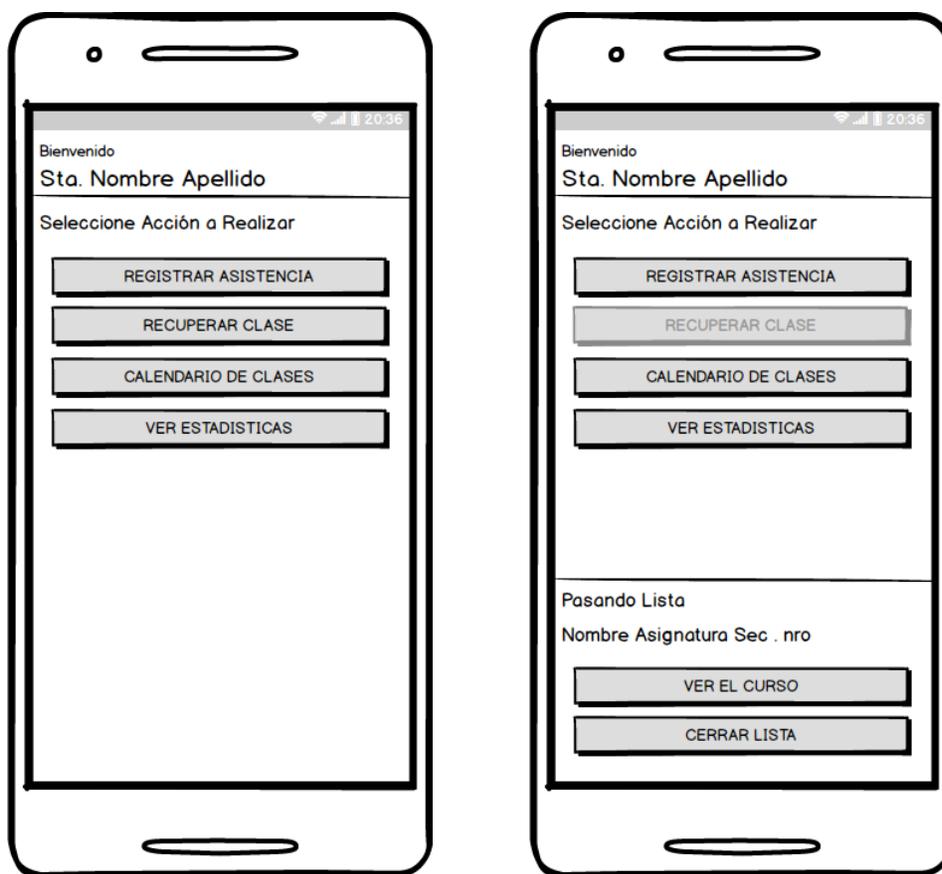


Figura 6: Mockup Menú Principal, Aplicación Docente

Registrar Asistencia

En la Figura 7 se presenta la vista de Registro de Asistencia, la cual permite registrar asistencia a un curso que se está realizando actualmente, muestra los datos de la hora de clases, como el nombre, sección, fecha, hora inicio, hora término. El único campo editable son los contenidos de la clase.

Lo más importante de esta ventana es que permite seleccionar entre registrar asistencia manualmente, o por Token, con cuya opción el profesor genera un Token al azar para luego continuar.



Figura 7: Mockup Menú de Pasar lista, Aplicación Docente

Recuperar Clases

En la Figura 8 se presenta la vista de Registro de Asistencia, la cual cumple la función de presentar un medio para registrar fechas de recuperación de clases, es similar a “Registrar Asistencia” con la gran diferencia que el curso es seleccionado de los cursos que realiza el profesor en el semestre, y la hora de inicio y de término es seleccionada por el profesor.

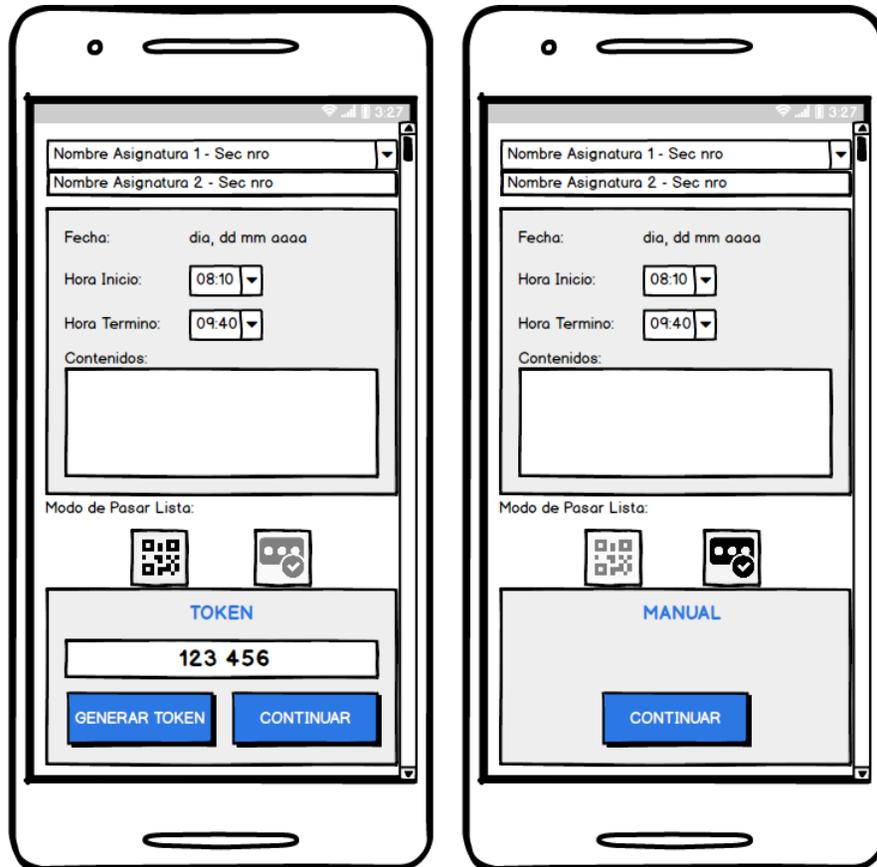


Figura 8: Mockup Recuperar Clase, Aplicación Docente

Lista de Estudiantes

En la figura 9 se presenta la vista Lista de Estudiantes, que se abre al registrar la asistencia de una clase, cumple la función de mostrar la lista de alumnos, donde se pueden marcar o desmarcar a los alumnos presentes por medio de un Switch, de manera que esta sirva como una versión digital y dinámica de la lista de clases que se usa actualmente. Cuenta con dos grandes variantes dependiendo del modo de registrar asistencia, siendo estas el “Modo Token” y el “Modo Manual”.

El “Modo Token” requiere que los alumnos utilicen su aplicación para marcarse presentes, por lo que este modo representa el estado actual de los alumnos por medio de “TAGs” ubicados bajo su nombre y un contador de alumnos totales y según TAG, teniendo así una visión clara de todo el proceso de asistencia. Los TAGs utilizados son los siguientes:

- En rango: Utilizado cuando todo el procedimiento funciona correctamente, ósea, el alumno se marcó presente desde dentro de la sala, por lo que el GPS lo marca como dentro de rango y el Switch queda en presente.
- Fuera de Rango: Utilizado cuando un alumno se trata marcar presente desde fuera de la sala, por lo que el GPS indica que este está fuera del rango sin quedar presente en la clase. Este TAG existe con el objetivo de que el profesor pueda ver los alumnos que tratan de marcarse desde fuera de la sala y en caso de error de GPS este los marque presentes manualmente.
- Manual: Utilizado cuando el profesor marca el Switch por su cuenta, dejando a los alumnos presentes de forma manual.

El “Modo Manual”, no requiere conexión con la Aplicación del alumno, por lo que se descarta el uso de “TAGs”, mostrando solo la lista y el total de alumnos presentes.



Figura 9: Mockup Menú Lista Alumnos Manual/Token, Aplicación Docente

Estadísticas – Selección de Cursos

En la figura 10 se presenta la vista Estadísticas – Selección de Cursos, que sirve como menú de selección de cursos y transición entre el menú principal y las estadísticas de un curso. Muestra los distintos cursos que realiza y ha realizado un profesor, de manera que al seleccionar alguno, se puedan ver las estadísticas de asistencia.

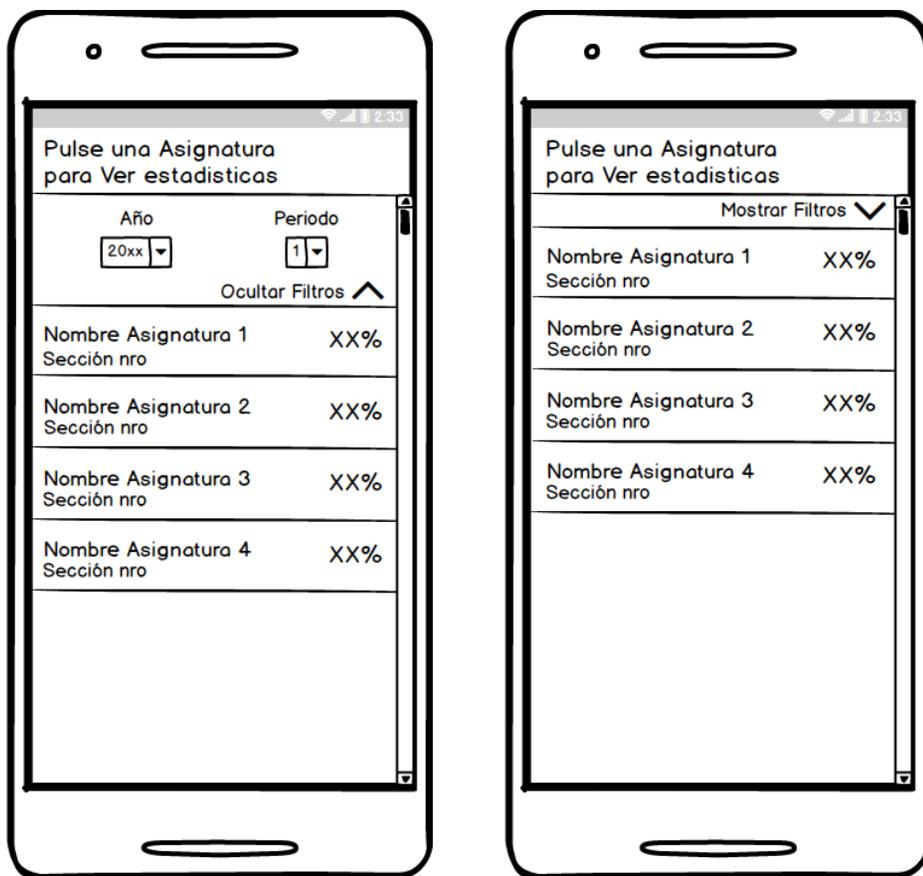


Figura 10: Mockup Menú Selección Estadísticas, Aplicación Docente

Suspender Clases

En la figura 11 se presenta la vista Suspender Clases, la cual permite Suspender las una asignatura, se muestran las asignaturas que se poseen ordenadas por fecha, de manera que presionando el Switch se suspenda la clase.

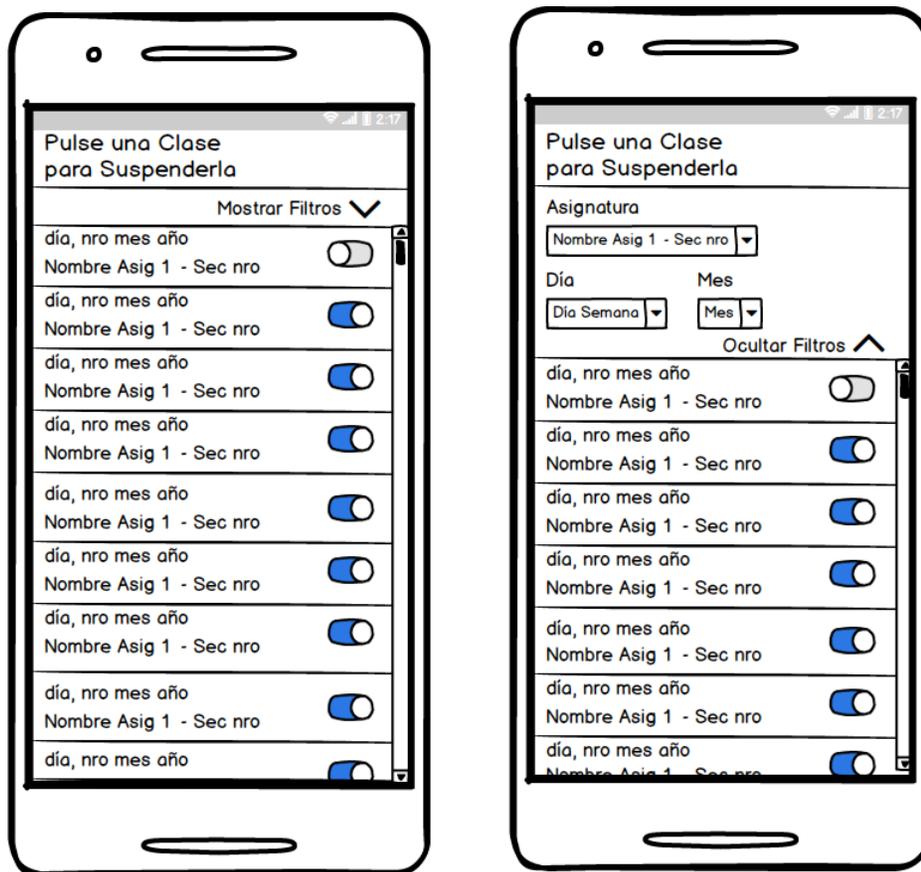


Figura 11: Mockup Menú Suspender Clases, Aplicación Docente

Estadísticas – General/ Alumnos

Las Figuras 12 y 13 presentan la vista Estadísticas, la cual muestra las estadísticas del curso seleccionado, sirviendo así de representación visual del estado actual de un curso en el ámbito de la asistencia a clases.

Cuenta con dos categorías, siendo la primera, “General”, que muestra las estadísticas de asistencia a clases totales del curso durante todo el periodo, “Alumnos”, que muestra las estadísticas del curso desde la perspectiva de los estudiantes. La navegación de estas dos actividades no se da por un menú convencional, sino que se podrá ir desde el menú “General” al menú “Alumnos” con un botón en la parte superior de la pantalla, o deslizando el dedo de izquierda a derecha, de manera que estas dos vistas estén una al lado de la otra.

La Figura 12, muestra la de la Vista “General” de la vista de estadísticas la cual presenta las estadísticas del curso en forma de gráficos de barra que muestra el promedio de asistencia por semestre, mes y semana, donde el semestre presenta un gráfico de barras que muestra los 6 meses del periodo, el mes, muestra un gráfico de barra donde se ven las 4 semanas del mes, y finalmente la semana muestra los días de clases. Dado que para ver el gráfico de mes, se requiere seleccionar uno, la tabla mes solo se muestra cuando se presione el mes del gráfico de periodo, y la tabla de la semana se muestra al seleccionar un mes de la tabla mes.



Figura 12: Mockup Menú Estadísticas General, Aplicación Docente

La figura 13, muestra la de la Vista “Alumno” de la vista de estadísticas, esta muestra la lista de estudiantes del curso junto a su porcentaje de asistencia total del curso, permitiendo “expandir” los datos de un estudiante determinado para ver estadísticas como el número de días asistidos, el porcentaje de asistencia basado en las clases realizadas y el porcentaje de asistencia basado total de clases que se realizan en el curso.

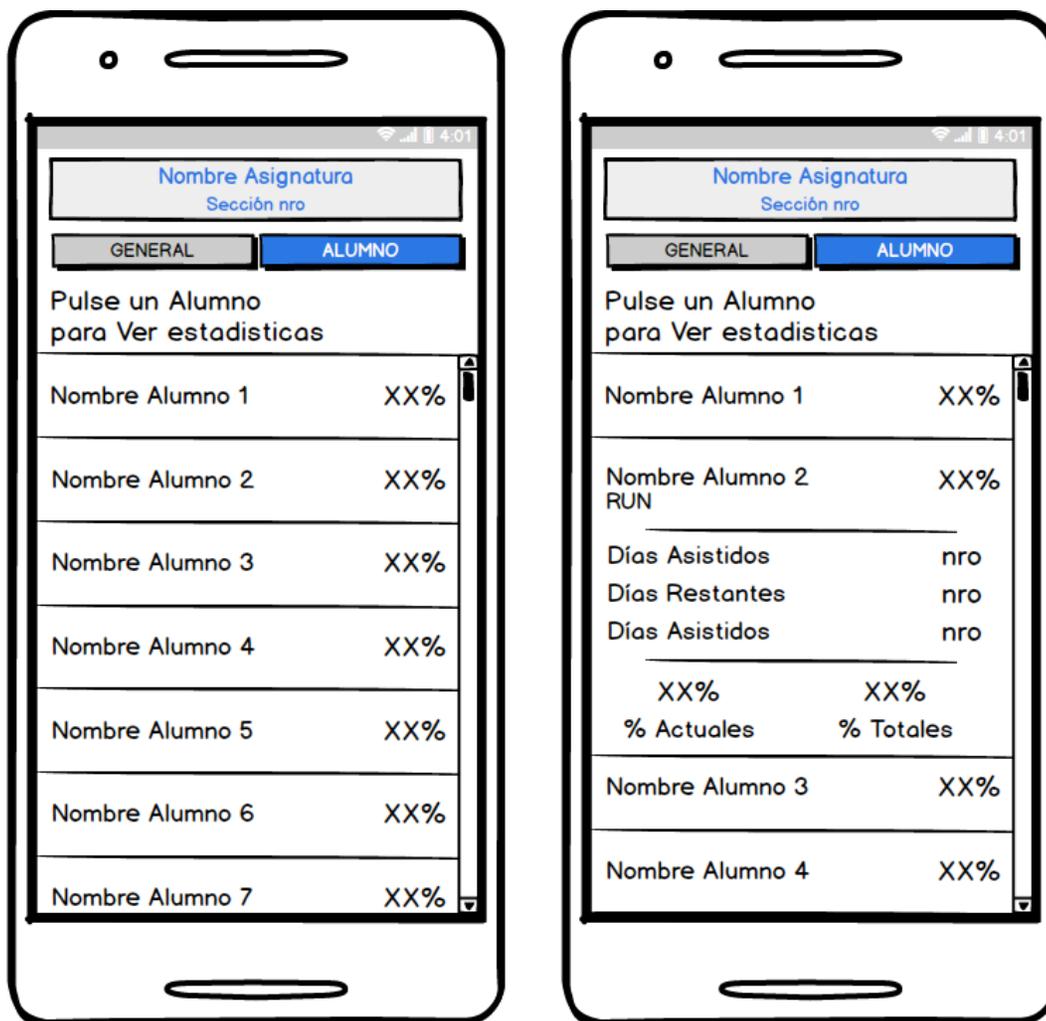


Figura 13: Mockup Menú Estadísticas General, Aplicación Docente

Informe Clase

En la figura 14 se presenta la vista Informa de Clases, la cual presenta todos los datos de una clase ya realizada. El acceso a esta se da desde el menú “General” de Estadísticas, al momento de presionar un día determinado.

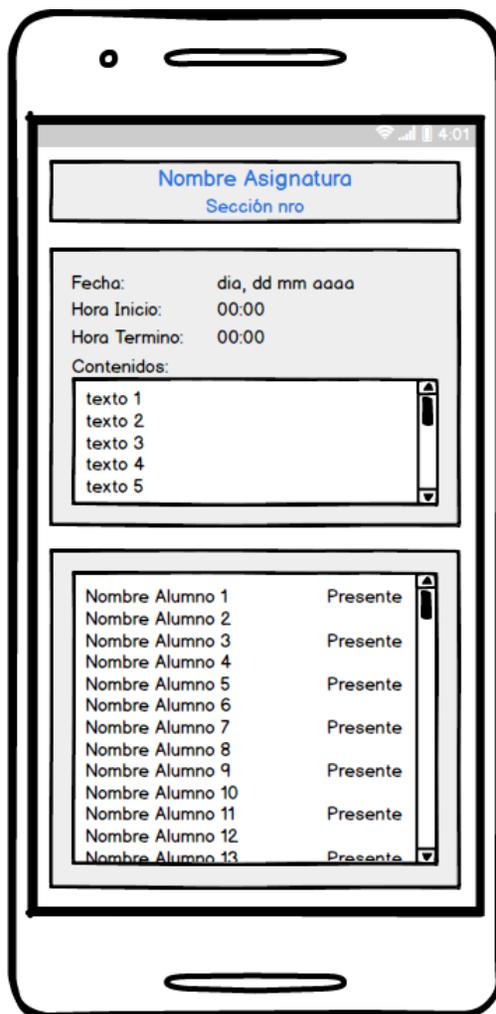


Figura 14: Mockup Informe Día, Aplicación Docente

Navegación de la Aplicación

La navegación por la aplicación (véase Figura 15) se presenta en forma de la vista de Inicio de Sesión, la cual lleva al menú principal, en este menú el profesor puede seleccionar entre las 3 opciones, las cuales son estadísticas, registrar asistencia y recuperar clase.

Las opciones de registrar asistencia y recuperar clases llevan a una vista similar la cual permite abrir una sesión de clase en “Modo Manual” o “Modo Token”.

Una vez seleccionado un modo, se abre la lista del curso, dependiendo de la opción seleccionada. Estando la lista abierta al volver al menú principal, se bloquean las opciones de abrir una lista, y se abre una nueva opción de ver la lista y cerrar el curso.

Referente a estadísticas, esta lleva a una pantalla de selección de curso, donde al seleccionar el curso se abre la pantalla de estadísticas, esta presenta dos opciones que son “General” y ”Alumnos”. Además de esto se puede ver un informe de la clase seleccionando el día de clases desde la opción general.

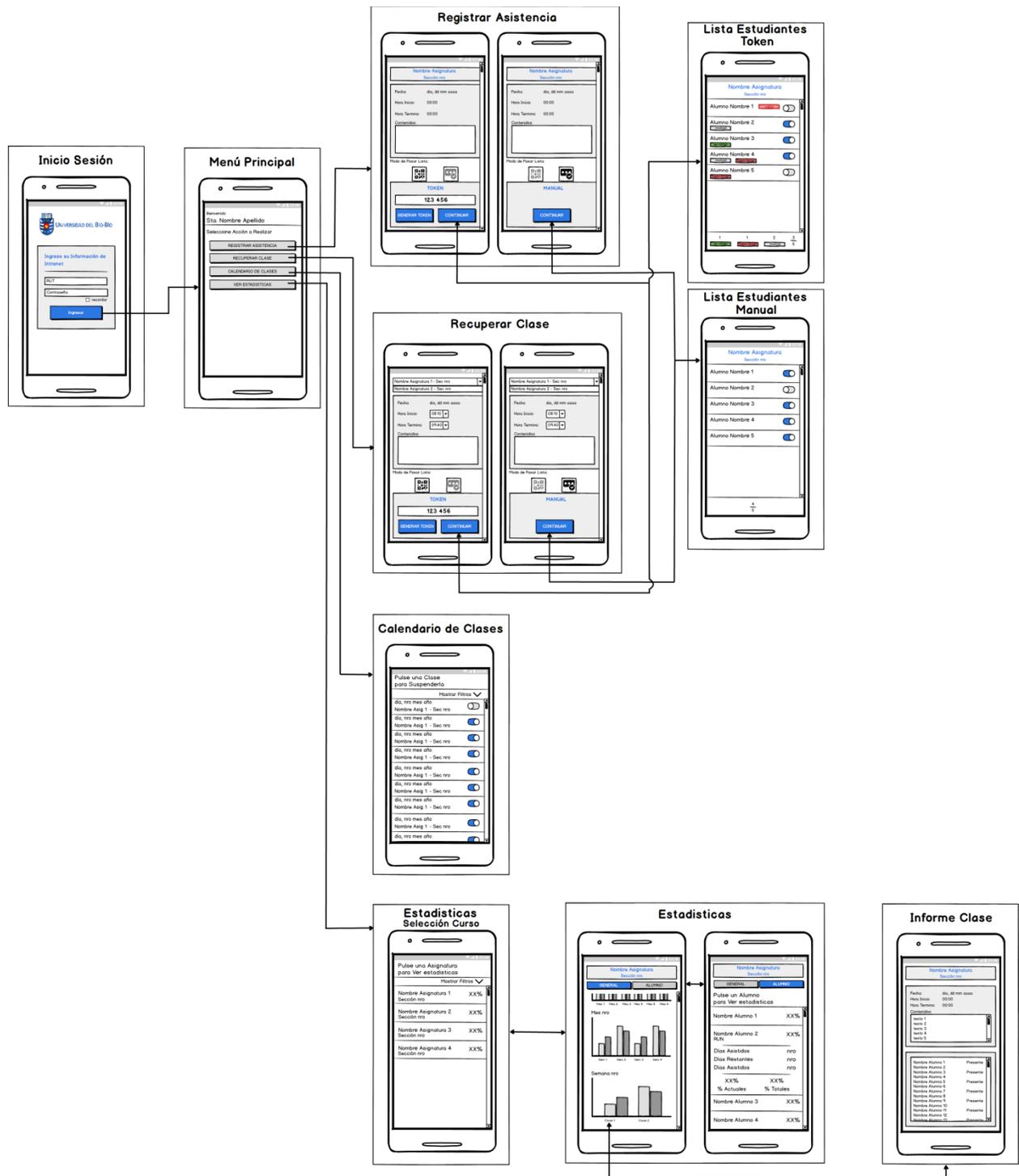


Figura 15: Mapa de Navegación de Aplicación del Profesor

6.5.2. Aplicación Alumno

Esta Aplicación es la más simple, dado que solo cuenta con dos opciones, siendo estas registrarse en la lista creada por la Aplicación del profesor, y ver las estadísticas de un curso.

Inicio de Sesión

El inicio de sesión es idéntico al inicio de sesión del profesor.

Menú Principal

En la figura 16 se presenta la vista Menú Principal, la cual es el acceso a las dos opciones disponibles para la aplicación, “Registrarse en Lista” y “Ver Estadísticas”. Dado que no siempre habrá un curso abierto, la opción de “Registrarse en Lista” no siempre estará habilitada.



Figura 16: Mockup Menú Principal, Aplicación Alumno

Registrarse en Lista

En la figura 17 se presenta la vista Registrarse en Lista , que permite registrarse en la lista creada por la aplicación del profesor, siendo este el medio para que el estudiante se registre presente. Dado que para que esta opción este habilitada el profesor debe haber seleccionado la opción Token en su aplicación, para poder registrarse en la lista, hay que ingresar el Token, por lo que esta vista permite ingresar el Token de 6 dígitos para luego ser enviado. Mejorar redacción

The mockup shows a mobile application interface for registering in a list. At the top, there are two labels: 'Nombre Asignatura' and 'Sección nro'. Below these is a 'Profesor:' label followed by a 'Nombre Apellido' field. A prompt 'Ingrese el Token Dictado por el Profesor' is displayed above a 'TOKEN' input field. A blue 'CONTINUAR' button is positioned below the token input field. The interface is contained within a smartphone frame.

Figura 17: Mockup Registrarse en Lista, Aplicación Alumno

Ver Estadísticas

En la figura 18 se presenta la vista Ver Estadísticas, la cual muestra las estadísticas de los cursos, esto se realiza a través de un filtro que permite cambiar el año y periodo, además de la lista de cursos, la cual da la opción de que un curso se “expanda” para que muestre las estadísticas de dicho curso.

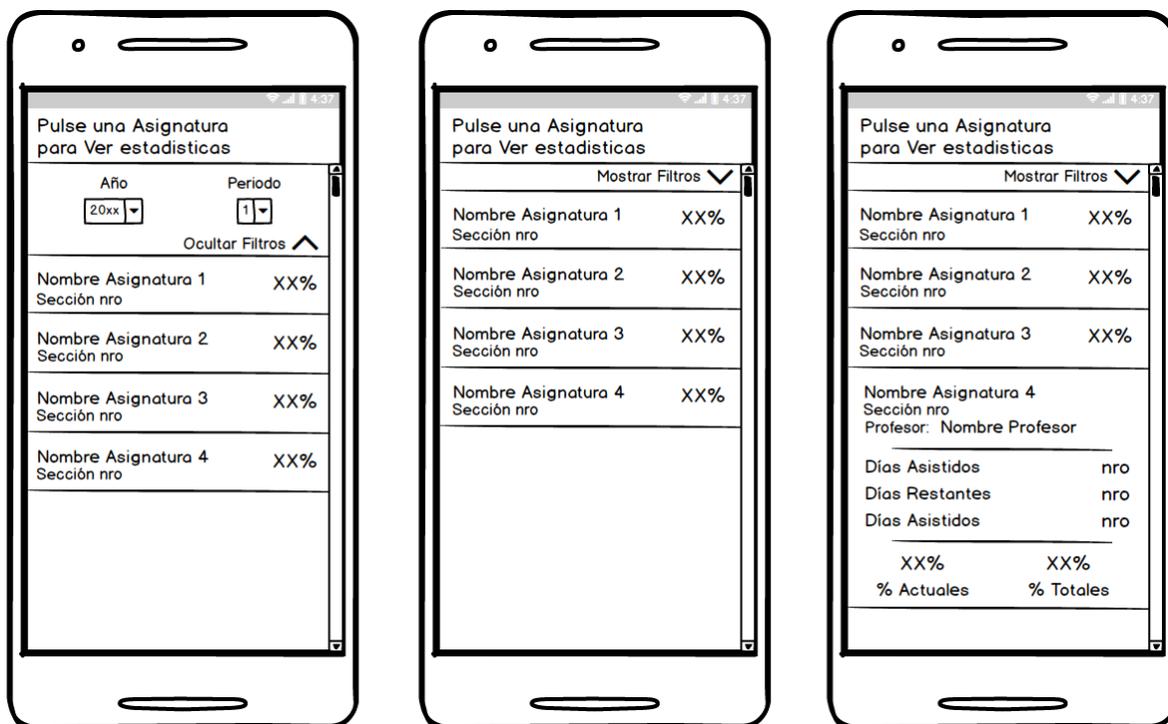


Figura 18: Mockup Ver Estadísticas, Aplicación Alumno

Navegación de la Aplicación

Dado la simpleza de esta aplicación, la navegación de las vistas consiste en muy pocos pasos. Básicamente, al momento de hacer Inicio de Sesión se va al menú principal, el cual lleva a las dos opciones que son “Registrarse en Asistencia” y “Ver Estadísticas“. Véase Figura 19.

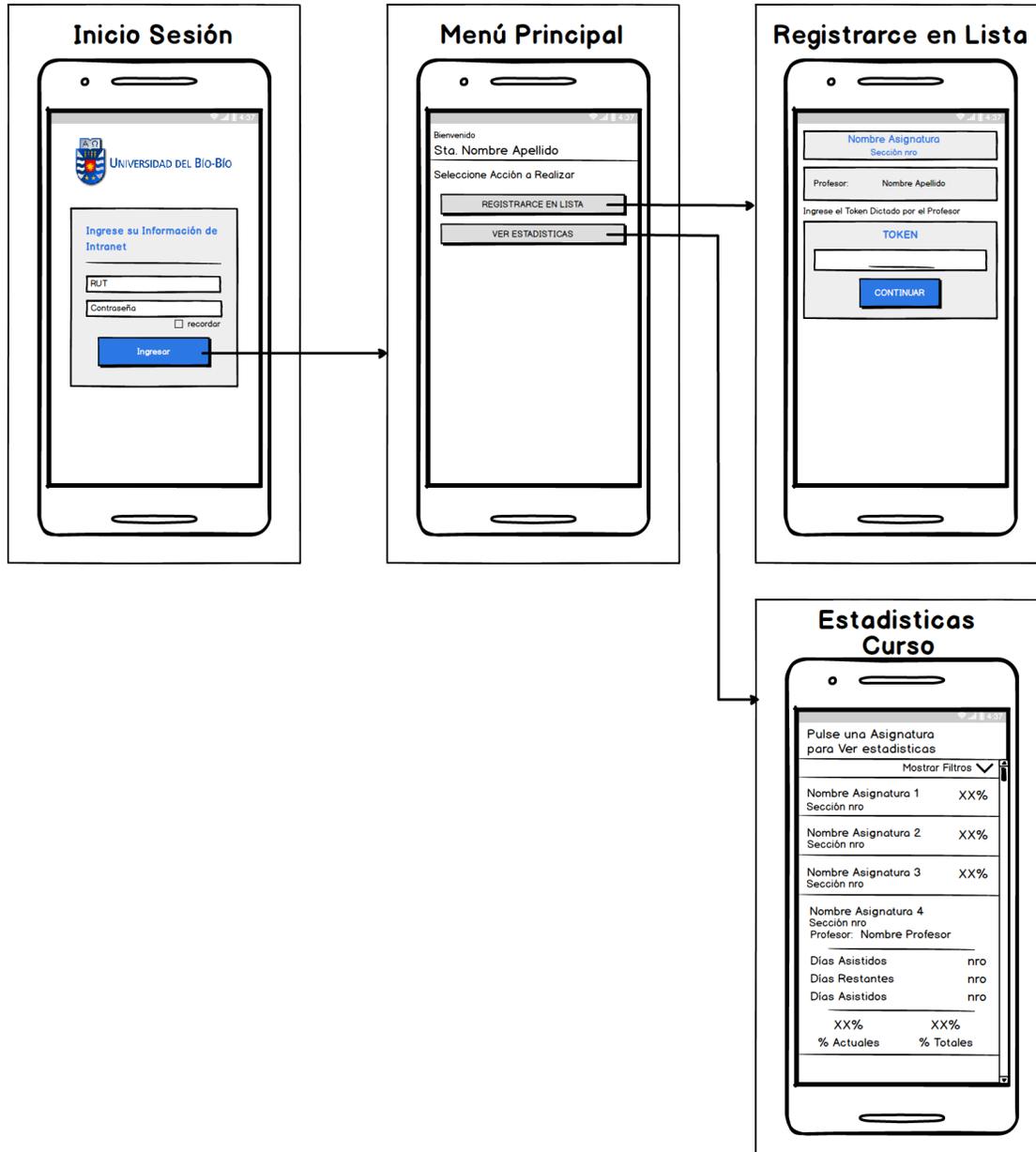


Figura 19: Mapa de Navegación Aplicación del Alumno

6.6. Proceso de Negocio Involucrado

El proceso que se lleva a cabo se puede describir como dos procesos que interactúan entre sí por medio de un servidor (véase Figura 20). Estos procesos son la aplicación del alumno y la del profesor, las cuales hacen consultas a los servicios alojados en un servidor el cual maneja y almacena la información en la Intranet de la UBB.

App Profesor: este es un proceso único que inicia todo el proceso de asistencia. El proceso comienza con el inicio de sesión a intranet por medio a una consulta a los servicios alojados en el servidor, posteriormente se obtienen los cursos de este y si posee una clase actualmente, de manera que este pueda recuperar clase, o registrar asistencia a la clase actual. Una vez seleccionado el curso se obtiene la lista de estudiantes y se crea una sesión de clases, en esta sesión se pueden marcar los alumnos manualmente o también los alumnos pueden marcarse presentes con sus aplicaciones a través del servidor, el cual notifica a la aplicación del profesor cuando un alumno se registra presente. Una vez finalizada la clase, el profesor cierra la sesión de clases y recibe un informe con el registro de los estudiantes presentes en la clase.

App Alumno: se puede describir a la aplicación del alumno como un proceso completamente dependiente de la aplicación del profesor, el cual no se puede llevar a cabo sin este. El proceso del alumno parte con un inicio de sesión en intranet, al igual que la App Profesor. Una vez en el sistema el estudiante recibe un mensaje de que hay una sesión de clases abierta y este puede ingresar el Token. Posteriormente se emitirá un mensaje hacia el servidor quien notifica a la aplicación del profesor, concluyendo el proceso de esta aplicación.

En resumen, el proceso es la interacción del proceso receptor del profesor y muchos procesos emisores de los alumnos, quienes consumen servicios alojados en un servidor, el cual sirve como medio de comunicación entre ambas aplicaciones. Para que el sistema funcione debe haber siempre un profesor y el servidor debe estar funcionando, de esta manera los estudiantes pueden registrarse presentes.

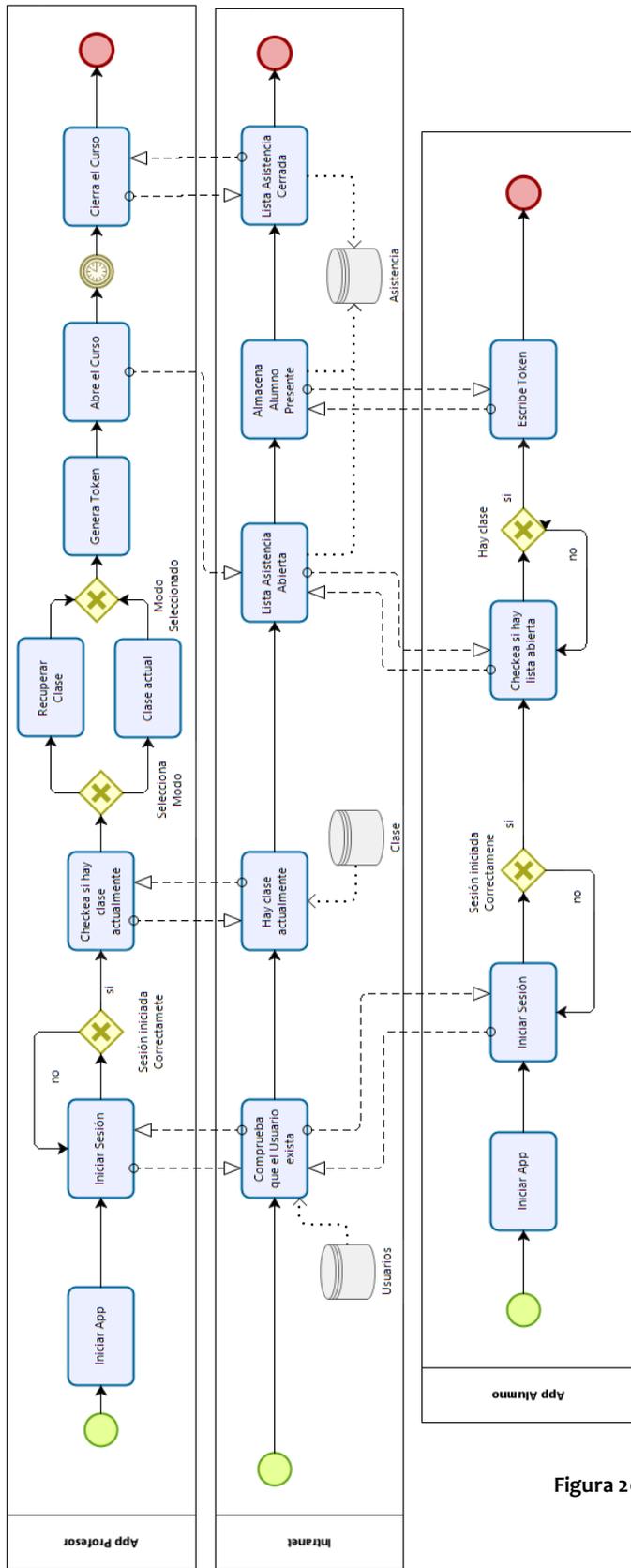


Figura 20: BPMN

6.7. Casos de Uso

6.7.1 Diagrama de Casos de Uso

Para la creación de los Casos de Uso (véase Figura 21) se consideran dos actores principales que interactúan con el sistema, estos son el Profesor y el Alumno, sirviendo uno como emisor de un “medio para pasar lista”, ya sea “Manual” o “Token” y el otro un receptor de este “medio”. El simple hecho de recibir la señal y ser un alumno del curso, hace que este quede registrado en la “lista virtual”. Una vez acabada la hora de clases, el profesor recibe un informe editable y pulsa “Guardar” para almacenar la lista. Además se consideran opciones de estadísticas tanto para el alumno como para el profesor. Para ver la respectiva especificación de cada caso de uso, véanse las tablas desde la 8 a la 18.

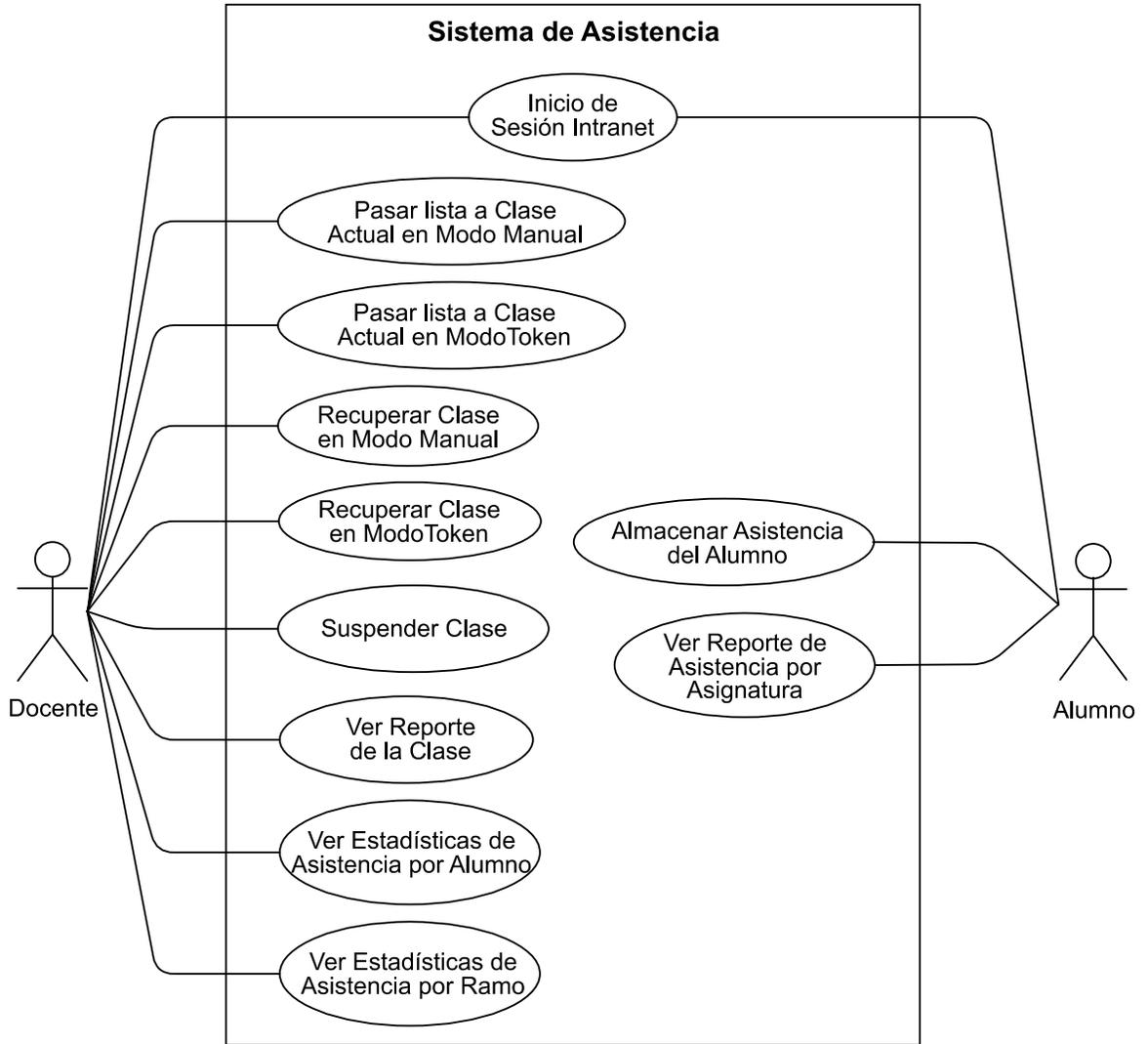


Figura 21: Diagrama de Casos de Uso

6.7.2 Especificación de Casos de Uso

Caso de Uso	Inicio se Sesión Intranet	
Actor	Docente y Alumno	
Descripción	Se hace ingreso a la aplicación a través de un inicio de Sesión que solicita los datos de Intranet UBB	
Pre Condiciones	<ul style="list-style-type: none"> • Estar registrado en la intranet de la Universidad • Ser un Profesor o Alumno para los sistemas de la Universidad, dependiendo del caso 	
Flujo Básico		
Actor	Sistema	
1. Se ingresa el RUT y clave 2. Se pulsa “Ingresar”	3. Se envía la consulta al servidor 4. El servidor da una respuesta positiva. 5. Se abre el “menú principal”	
Flujo Alterno		
Actor	Sistema	
RUT Invalido		
1. Se ingresa un RUT invalido	2. Muestra un mensaje de error indicando que el RUT es invalido	
Falta ingresar Clave		
1. Se ingresa un RUT pero no una clave	2. Muestra un mensaje de error indicando que uno de los campos está vacío	
Falta ingresar RUT		
1. Se ingresa una clave pero no el RUT	2. Muestra un mensaje de error indicando que uno de los campos está vacío	
El RUT no está en el sistema		
1. Se ingresa un RUT valido y clave, pero no está registrado en intranet	2. Se manda la consulta al servidor, este responde con un código de error 3. Muestra un mensaje de error indicando que el RUT no está en los sistemas de la Universidad	
Clave Invalida		
1. Se ingresa un RUT valido, que está registrado en Intranet, pero la clave no coincide	2. Se manda la consulta al servidor, este responde con un código de error 3. Muestra un mensaje de error indicando que el RUT no coincide con la clave	
No hay Internet		
1. Se ingresan un RUT y clave validas pero no se tiene activada la conexión a Internet	2. Muestra un mensaje de error indicando que debe estarce conectado a Internet para usar el sistema	
Servidor caído		
1. Se ingresan un RUT y clave validas	2. Se manda la consulta al servidor, y este no responde 3. Muestra un mensaje de error indicando que hay un problema de conexión con el servidor	
Post Condiciones	No existen Post Condiciones	

Tabla 8: Especificación Caso de Uso: Inicio de Sesión Intranet

Caso de Uso	Pasar Lista a Clase Actual en Modo Manual	
Actor	Docente	
Descripción	Se abre el menú de la clase que se debe realizar actualmente con objetivo de iniciar esta y pasar lista en modo Manual	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Estar en el menú principal de la Aplicación • Tener que realizar una clase Actualmente • No estar pasando lista en otra clase actualmente 	
Flujo Básico		
Actor	Sistema	
<ol style="list-style-type: none"> 1. Se selecciona la opción “Pasar Lista” desde el menú principal de la Aplicación 3. (Opcional) Se ingresan los contenidos de la clase 4. Se Selecciona el Modo “Manual” pulsando el ícono de una Lista 6. Se pulsa el botón “Continuar” 9. Se marcan los alumnos presentes, presionando el Switch al lado del nombre del alumno 	<ol style="list-style-type: none"> 2. Abre el menú de “Pasar Lista” 5. Se abre el sub menú “Manual” 7. Se abre la lista de alumnos a nivel de sistema 8. Se muestra la “Lista de Alumnos” en modo Manual 10. Se almacena el alumno como presente dentro de Intranet 	
Flujo Alterno		
Actor	Sistema	
Se decide no pasar asistencia		
<ol style="list-style-type: none"> 1. Se vuelve al menú anterior con el botón “atrás” 	<ol style="list-style-type: none"> 2. Abre el menú principal 	
Se decide pasar asistencia en modo Token		
Véase el caso de Uso “Pasar Lista a Clase Actual en Modo Token” desde el paso 4		
Post Condiciones	<ul style="list-style-type: none"> • Cerrar la clase manualmente, en caso contrario se cierra automáticamente 	

Tabla 9: Especificación Caso de Uso: Pasar Lista a Clase Actual en Modo Manual

Caso de Uso	Pasar Lista a Clase Actual en Modo Token	
Actor	Docente	
Descripción	Se abre el menú de la clase que se debe realizar actualmente con objetivo de iniciar esta y registrar asistencia en modo Manual	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Estar en el menú principal de la Aplicación • Tener que realizar una clase Actualmente • No estar pasando lista en otra clase actualmente 	
Flujo Básico		
Actor	Sistema	
1. Se selecciona la opción “Pasar Lista” desde el menú principal de la Aplicación 3. (Opcional) Se ingresan los contenidos de la clase 4. Se Selecciona el Modo “Token” pulsando el ícono de un Token 6. Se pulsa el Botón “Generar Token” 8. Se informa del Token a los alumnos 9. Se pulsa el botón “Continuar” 12. (Opcional) Se marcan los alumnos presentes, presionando el Switch al lado del nombre del alumno	2. Abre el menú de “Pasar Lista” 5. Se abre el sub menú “Token” 7. Se genera y muestra un Token al azar 10. Se abre la lista de alumnos a nivel de sistema 11. Se muestra la “Lista de Alumnos” en modo Token, mostrando los alumnos con mala conducta primero 13. (Opcional) Se almacena el alumno marcado manualmente en intranet 14. Se reciben mensajes de Presente desde la Aplicación del Alumno(Véase el caso de uso “Almacenar Asistencia Alumno”) 15. Se muestra los cómo alumnos que se van marcando presentes	
Flujo Alterno		
Actor	Sistema	
Se decide no pasar asistencia		
1. Se vuelve al menú anterior con el botón “atrás”	2. Abre el menú principal	
Se decide pasar asistencia en modo Manual		
Véase el caso de Uso “Pasar Lista a Clase Actual en Modo Manual” desde el paso 4		
Post Condiciones	<ul style="list-style-type: none"> • Cerrar la clase manualmente, en caso contrario se cierra automáticamente 	

Tabla 10: Especificación Caso de Uso: Pasar Lista a Clase Actual en Modo Token

Caso de Uso	Recuperar Clase en Modo Manual	
Actor	Docente	
Descripción	Se abre el menú de la clase que se recupera con objetivo de iniciar esta y registrar asistencia Manualmente	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Estar en el menú principal de la Aplicación • Tener que realizar una clase Actualmente • No estar pasando lista en otra clase actualmente 	
Flujo Básico		
Actor	Sistema	
1. Se selecciona la opción “Recuperar Clase” desde el menú principal de la Aplicación 3. Se selecciona el curso del que se desea recuperar clase 4. Se ingresa una hora de Inicio y de Termino 4. (Opcional) Se ingresan los contenidos de la clase 6. Se Selecciona el Modo “Manual” pulsando el ícono de una Lista 8. Se pulsa el botón “Continuar” 11. Se marcan los alumnos presentes, presionando el Switch al lado del nombre del alumno	2. Abre el menú de “Pasar Lista” 7. Se abre el sub menú “Manual” 11. Se muestra la “Lista de Alumnos” en modo Manual 13. (Opcional) Se almacena el alumno marcado manualmente en intranet 14. Se reciben mensajes de Presente desde la Aplicación del Alumno(Véase el caso de uso “Almacenar Asistencia Alumno”) 15. Se muestra los cómo alumnos que se van marcando presentes	
Flujo Alterno		
Actor	Sistema	
Se decide no pasar asistencia		
1. Se vuelve al menú anterior con el botón “atrás”	2. Abre el menú principal	
No se ingresa la Hora		
1. No se ingresa la hora de inicio ni de termino	2. No se permite abrir la clase 3. Se muestra un mensaje de error indicando que se debe seleccionar la hora	
Se decide pasar asistencia en modo Token		
Véase el caso de Uso “Recuperar Clase en Modo Token” desde el paso 6		
Post Condiciones	<ul style="list-style-type: none"> • Cerrar la clase manualmente, en caso contrario se cierra automáticamente 	

Tabla 11: Especificación Caso de Uso: Recuperar Clase en Modo Manual

Caso de Uso	Recuperar Clase en Modo Token	
Actor	Docente	
Descripción	Se abre el menú de la clase que se recupera con objetivo de iniciar esta y registrar asistencia Automáticamente	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Estar en el menú principal de la Aplicación • Tener que realizar una clase Actualmente • No estar pasando lista en otra clase actualmente 	
Flujo Básico		
Actor	Sistema	
<ol style="list-style-type: none"> 1. Se selecciona la opción “Recuperar Clase” desde el menú principal de la Aplicación 3. Se selecciona el curso del que se desea recuperar clase 4. Se ingresa una hora de Inicio y de Termino 4. (Opcional) Se ingresan los contenidos de la clase 5. Se Selecciona el Modo “Token” pulsando el ícono de un Token 7. Se pulsa el Botón “Generar Token” 9. Se informa del Token a los alumnos 10. Se pulsa el botón “Continuar” 13. (Opcional) Se marcan los alumnos presentes, presionando el Switch al lado del nombre del alumno 	<ol style="list-style-type: none"> 2. Abre el menú de “Pasar Lista” 6. Se abre el sub menú “Token” 8. Se genera y muestra un Token al azar 11. Se abre la lista de alumnos a nivel de sistema con el curso y horas seleccionadas 12. Se muestra la “Lista de Alumnos” en modo Token, mostrando los alumnos con mala conducta primero 14. (Opcional) Se almacena el alumno marcado manualmente en intranet 15. Se reciben mensajes de Presente desde la Aplicación del Alumno(Véase el caso de uso “Almacenar Asistencia Alumno”) 16. Se muestra los cómo alumnos que se van marcando presentes 	
Flujo Alterno		
Actor	Sistema	
Se decide no pasar asistencia		
<ol style="list-style-type: none"> 1. Se vuelve al menú anterior con el botón “atrás” 	<ol style="list-style-type: none"> 2. Abre el menú principal 	
No se ingresa la Hora		
<ol style="list-style-type: none"> 1. No se ingresa la hora de inicio ni de termino 	<ol style="list-style-type: none"> 2. No se permite abrir la clase 3. Se muestra un mensaje de error indicando que se debe seleccionar la hora 	
Se decide pasar asistencia en modo Manual		
Véase el caso de Uso “Recuperar Clase en Modo Manual” desde el paso 6		
Post Condiciones	<ul style="list-style-type: none"> • Cerrar la clase manualmente, en caso contrario se cierra automáticamente 	

Tabla 12: Especificación Caso de Uso: Recuperar Clase en Modo Token

Caso de Uso	Suspender Clase	
Actor	Docente	
Descripción	Se suspende una clase que este calendarizada	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Estar en el menú principal de la Aplicación • Tener Clases Calendarizadas 	
Flujo Básico		
Actor	Sistema	
1. Se selecciona la opción “Calendario de Clases” desde el menú principal	2. Abre el calendario de clases, el cual muestra una lista de las clases calendarizadas para el docente ordenadas por fecha	
3. Se pulsa el Switch correspondiente a la clase que se desea suspender	4. Muestra un dialogo de confirmación con un espacio para ingresar el motivo	
5. (Opcional) Se ingresa el motivo	7. Se suspende la clase	
6. Se pulsa Suspender	8. Se envía una notificación a los alumnos	
Flujo Alterno		
Actor	Sistema	
Se decide no suspender la Clase		
1. Se vuelve al menú anterior con el botón “atrás	2. Abre el menú principal	
Post Condiciones	No existen Post Condiciones	

Tabla 13: Especificación Caso de Uso: Suspender Clase

Caso de Uso	Ver Reporte de la Clase	
Actor	Docente	
Descripción	Se muestra un reporte de la clase con la fecha, contenidos y alumnos, detallando si están presentes o no	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Haber seleccionado un día desde el menú estadísticas general o haber cerrado una clase 	
Flujo Básico		
Actor	Sistema	
1a. Se cierra una clase desde el botón cerrar clase en el menú principal 1b. Se pulsa un día desde la sección de estadísticas, en el gráfico día	2. Se abre un informe con el detalle de la clase	
Flujo Alterno		
No existe flujo alternativo		
Post Condiciones	No existen Post Condiciones	

Tabla 14: Especificación Caso de Uso: Ver Reporte de la Clase

Caso de Uso	Ver Estadísticas de Asistencia por Ramo	
Actor	Docente	
Descripción	Se muestra un reporte gráfico con las estadísticas de un curso	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Estar en el menú principal de la Aplicación 	
Flujo Básico		
Actor	Sistema	
1. Se selecciona la opción “Estadísticas” desde el menú principal 3. Se pulsa el curso del cual se desean ver estadísticas 5. (Opcional) Se pulsa un mes en el gráfico de del semestre 7. (Opcional) Se pulsa una semana en el gráfico del mes	2. Se abre el menú de Estadísticas, el cual presenta la lista de cursos a realizar 4. Se abre la pantalla de Estadísticas Generales del curso, la cual presenta un gráfico con las estadísticas del semestre 6. Se abre un gráfico con el detalle del mes abajo del gráfico de semestre 8. Se abre un gráfico con el detalle de la semana abajo del gráfico de mes	
Flujo Alterno		
Actor	Sistema	
Se desea ver el detalle de otro mes		
1. Se pulsa otro mes desde la tabla semestre	2. Abre el gráfico del mes	
Se desea ver el detalle de otra semana		
1. Se pulsa otra semana desde la tabla mes	2. Abre el gráfico de la semana	
Se desea ver la estadística del curso según alumnos		
1. Se pulsa el botón “Alumnos” debajo del nombre del curso Véase el caso de Uso “Ver Estadísticas de Asistencia por Alumno” desde el paso 6	2. Abre la pantalla de estadísticas por alumno	
Post Condiciones	No existen Post Condiciones	

Tabla 15: Especificación Caso de Uso: Ver Estadísticas de Asistencia por Ramo

Caso de Uso	Ver Estadísticas de Asistencia por Alumno	
Actor	Docente	
Descripción	Se muestra un reporte gráfico con las estadísticas de un curso por Alumno	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Estar en el menú principal de la Aplicación 	
Flujo Básico		
Actor	Sistema	
1. Se selecciona la opción “Estadísticas” desde el menú principal	2. Se abre el menú de Estadísticas, el cual presenta la lista de cursos a realizar	
3. Se pulsa el curso del cual se desean ver estadísticas	4. Se abre la pantalla de Estadísticas Generales del curso, la cual presenta un gráfico con las estadísticas del semestre	
5. Se pulsa el botón Alumnos debajo del Nombre del Curso	6. Se abre la pantalla de estadísticas por Alumno del Curso, la cual muestra la lista de alumnos, junto al porcentaje de asistencia	
7. (Opcional) Se pulsa un alumno para ver detalle	8. Se expande el gráfico con los detalles de la asistencia del alumno a clases	
Flujo Alterno		
Actor	Sistema	
Se desea ver el detalle de otro alumno		
1. Se pulsa otro alumno	2. Expande al alumno, cerrando el que estaba expandido	
Se desea ver la estadística del curso en general		
1. Se pulsa el botón “General” debajo del nombre del curso	2. Abre la pantalla de estadísticas del ramo	
Véase el caso de Uso “Ver Estadísticas de Asistencia por Ramo” desde el paso 4		
Post Condiciones	No existen Post Condiciones	

Tabla 16: Especificación Caso de Uso: Ver Estadísticas de Asistencia por Alumno

Caso de Uso	Almacenar Asistencia del Alumno	
Actor	Alumno	
Descripción	El Alumno registra su asistencia en un curso	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Estar en el menú principal de la Aplicación • El Docente debe haber abierto una lista de clases de un curso al que se pertenezca en modo Token • Se debe conocer el Token 	
Flujo Básico		
Actor	Sistema	
1. Se selecciona la opción “Registrarse en Lista” desde el menú principal 3. Se ingresa el Token en el campo Token 5. Se pulsa el botón “Continuar”	2. Se abre en menú de Registrarse en Lista, el cual tiene un campo para ingresar el Token 6. Se envía la consulta al servidor el cual responde positivamente 7. Se envía un mensaje de que se quedó registrado en la lista correctamente	
Flujo Alterno		
Actor	Sistema	
El Token no coincide		
	1. El servidor envía un mensaje de que el Token es erróneo 2. Se muestra un mensaje de error	
Post Condiciones	No existen Post Condiciones	

Tabla 17: Especificación Caso de Uso: Almacenar Asistencia del Alumno

Caso de Uso	Ver Reporte de Asistencia por Asignatura	
Actor	Alumno	
Descripción	Se muestra un reporte con las estadísticas de los cursos	
Pre Condiciones	<ul style="list-style-type: none"> • Haber iniciado Sesión en la Aplicación • Estar en el menú principal de la Aplicación 	
Flujo Básico		
Actor	Sistema	
1. Se selecciona la opción “Estadísticas” desde el menú principal	2. Se abre el menú de Estadísticas, el cual presenta la lista de cursos a realizar	
3. Se pulsa el curso del cual se desean ver estadísticas	4. Se abre la pantalla de Estadísticas Generales, la cual muestra la lista de cursos, junto al porcentaje de asistencia	
5. (Opcional) Se pulsa un curso para ver detalle	6. Se expande el curso con los detalles de la asistencia del alumno a clases	
Flujo Alterno		
Actor	Sistema	
Se desea ver el detalle de otro curso		
1. Se pulsa otro curso de la lista de cursos	2. Expande el curso seleccionado cerrando el curso expandido	
Post Condiciones	No existen Post Condiciones	

Tabla 18: Especificación Caso de Uso: Ver Reporte de Asistencia por Asignatura

7. CONCLUSIÓN

Con respecto al producto, se puede decir que el sistema los cumple con los objetivos planteados, presentando así una herramienta que facilita el trabajo de los docentes al momento de registrar asistencia de los estudiantes de un curso, otorgando así una mejor calidad en el registro de los datos y mejor experiencia en clases, ya que los docentes no deberán llevarse trabajo extra a su oficina y no se requiere detener una clase para el registro de asistencia directamente en la plataforma.

Dentro de las mejoras que se pueden agregar al proyecto en trabajos futuros se propone hacer que el sistema funcione de manera offline en el modo manual, de manera que el profesor pueda marcar la lista desde el sistema cuando no posea conexión a internet, esto se puede realizar almacenando los alumnos en un archivo de texto dentro del Smartphone y almacenando la lista en Intranet una vez que haya conexión a Internet.

Otra mejora es la recopilación de información captada por el sistema para saber que salas de clase presentan una mayor tasa de fallos del GPS, de manera que se obtengan datos estadísticos útiles para poder mejorar el sistema o crear nuevo sistemas a futuro.

Durante todo el proyecto se tuvo un objetivo claro, el cual es “Mejorar el proceso de registro de asistencia a clases de la Universidad”, sirviendo esto como meta a medida que se buscaba la mejor manera de cumplirla. Esta búsqueda fue una de las mayores dificultades del proyecto y se puede separar en dos grandes ámbitos, conocimientos técnicos para llevar a cabo el proyecto, y la experiencia que este le entregue al usuario.

Respecto a los conocimientos técnicos, al momento de comenzar el proyecto se poseía experiencia desarrollando para la plataforma Android, pero esta experiencia no fue suficiente, durante todo el proceso de desarrollo se debió aprender cosas nuevas, teniendo una constante batalla entre lo que se “busca realizar” contra un “cómo se realiza”, como solución a esto se tuvo la investigación, constante “prueba y error”, y la paciencia para llevarlo a cabo. La mayor dificultad técnica que presentó el proyecto fue el desarrollo para iOS, siendo esta una plataforma desconocida para la cual nunca se desarrolló en el pasado, donde al momento de trabajar en esta, se tenía completamente claro lo que se debía realizar, siendo el mayor problema fue traducir esas ideas a la nueva plataforma.

Dentro de todo este proceso, trabajar con una metodología ágil resulto extremadamente beneficioso, ya que dada su naturaleza, permite desarrollar ciertas partes del proyecto a medida que se investiga lo que se busca hacer a futuro, y además mostrar avances los cuales permiten obtener el feedback necesario para que el producto se ajuste a los requerimientos de la Universidad. Este feedback se ganó durante reuniones con el área de desarrollo de sistemas de la Universidad, donde se presentaba la función que se quería realizar, para así buscar la mejor forma de llevar a cabo esta.

El segundo gran desafío fue la interfaz y la navegación por el sistema, esto se debe a que se estudió para ingeniero, no para diseñador, lo que para los ingenieros se ve simple, para muchas personas puede ser complejo o tosco. El centrarse en la experiencia de usuario fue como un cambio de paradigma a lo que fue enseñado durante la carrera, fue dejar de pensar en el proyecto como un “conjunto de funciones que debe llevar a cabo”, y verlo más como “de qué manera hago que estas funciones sean cómodas para los usuarios?”, dando así un paso más a lo requerido, y ofrecer el mejor producto posible.

Finalmente se puede decir que el presente proyecto y todo el trabajo que conllevó logra aplicar todos los conocimientos adquiridos durante el proceso universitario de la carrera Ingeniería Civil en Informática, además de conocimientos adquiridos personalmente por medio de investigación, siendo esta una de las experiencias más productivas que me a entregado la Universidad y demostrando así que la carrera nos deja preparados para el futuro convirtiéndonos en entes que saben lo necesario respecto a las ciencias de la computación y que son capaces de adquirir nuevos conocimientos cuando es necesario, se tiene la base y la capacidad de aprender.

8. REFERENCIAS BIBLIOGRÁFICAS

1. GlobalStats, Mobile Operating System Market Share Chile, Recuperado el 9 de Diciembre del 2018, de <http://gs.statcounter.com/os-market-share/mobile/chile>
2. Wikipedia, iOS, Recuperado el 9 de Diciembre del 2018, de <https://en.wikipedia.org/wiki/iOS>
3. Wikipedia, Android, Recuperado el 9 de Diciembre del 2018, de [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
4. Android Developers, AndroidStudio, Recuperado el 9 de Diciembre del 2018, de <https://developer.android.com/studio/>
5. Apple Developers, XCode, Recuperado el 9 de Diciembre del 2018, de <https://developer.apple.com/app-store/submissions/>
6. Google, Sube una App, Recuperado el 9 de Diciembre del 2018, de <https://support.google.com/googleplay/android-developer/answer/113469>
7. Google, Google Play Console, Recuperado el 9 de Diciembre del 2018, de <https://play.google.com/apps/publish/signup/>
8. Apple Developers, iOS Simulator, Recuperado el 9 de Diciembre del 2018, de https://developer.apple.com/library/archive/documentation/IDEs/Conceptual/iOS_Simulator_Guide/GettingStartedwithiOSSimulator/GettingStartedwithiOSSimulator.html
9. Apple Developers, Apple Developer Enterprise Program, Recuperado el 9 de Diciembre del 2018, de <https://developer.apple.com/programs/enterprise/>
10. Apple Developers, Apple Developer Program, Recuperado el 9 de Diciembre del 2018, de <https://developer.apple.com/programs/how-it-works/>

11. Coad P., de Luca J., Lefebvre E., (1999). *Java Modeling Color with Uml: Enterprise Components and Process with Cdom* (Capítulo 6). Upper Saddle River, New Jersey, USA: Prentice Hall PTR.
12. Krasner G., Pope S., (1988). *A cookbook for using the model-view controller user interface paradigm in Smalltalk-80*. Denville, New Jersey, USA: SIGS Publications.
13. Apple Developers, Swift, Recuperado el 9 de Diciembre del 2018, de <https://developer.apple.com/swift/>
14. Apple Developers, Swift, Recuperado el 9 de Diciembre del 2018, de <https://developer.apple.com/swift/>
15. Swift, About Swift, Recuperado el 9 de Diciembre del 2018, de <https://swift.org/about/#swiftorg-and-open-source>
16. Java, What is Java?, Recuperado el 9 de Diciembre del 2018, de https://www.java.com/es/download/faq/whatis_java.xml
17. Android Developers, XML Layout, Recuperado el 9 de Diciembre del 2018, de <https://developer.android.com/guide/topics/ui/declaring-layout>
18. W3C, Extensible Markup Language (XML), Recuperado el 9 de Diciembre del 2018, de <https://www.w3.org/XML/>
19. PHP, What is PHP?, Recuperado el 9 de Diciembre del 2018, de <http://php.net/manual/en/intro-what-is.php>
20. PHP, What can PHP do?, Recuperado el 9 de Diciembre del 2018, de <http://php.net/manual/en/intro-whatcando.php>
21. Apple, APP Store, Recuperado el 9 de Diciembre del 2018, de <https://www.apple.com/lae/ios/app-store/>
22. Google, Google Play, Recuperado el 9 de Diciembre del 2018, de <https://play.google.com/store?hl=en>

23. GlobalStats, Mobile & Tablet iOS Version Market Share Worldwide, Recuperado el 9 de Diciembre del 2018, de <http://gs.statcounter.com/ios-version-market-share/mobile-tablet/worldwide>
24. GlobalStats, Mobile & Tablet Android Version Market Share Worldwide, Recuperado el 9 de Diciembre del 2018, de <http://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide>
25. Wikipedia, iOS versión history, Recuperado el 9 de Diciembre del 2018, de https://en.wikipedia.org/wiki/IOS_version_history
26. Wikipedia, Android versión history, Recuperado el 9 de Diciembre del 2018, de https://en.wikipedia.org/wiki/Android_version_history