

**DISEÑO DE UN APLICATIVO MÓVIL PARA EL CONTROL DE ASISTENCIA  
ESTUDIANTIL EN LAS INSTITUCIONES DE EDUCACIÓN SUPERIOR**

**AUTORES:**

**CRISTIAN ANDRES TOLOZA SIERRA**

**ANGIE MILENA QUINTERO BAYONA**

**UNIVERSIDAD SIMÓN BOLÍVAR**

**FACULTAD DE INGENIERÍAS**

**ESPECIALIZACIÓN EN GESTIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN**

**SAN JOSÉ DE CÚCUTA**

**2021**

**DISEÑO DE UN APLICATIVO MÓVIL PARA EL CONTROL DE ASISTENCIA  
ESTUDIANTIL EN LAS INSTITUCIONES DE EDUCACIÓN SUPERIOR**

Autores:

**ANGIE MILENA QUINTERO BAYONA**

**CRISTIAN ANDRES TOLOZA SIERRA**

*Trabajo de investigación para optar título de: Especialista En Gestión De Tecnologías  
De La Información*

Tutor:

Ing. Johel Enrique Rodríguez Fernández

**UNIVERSIDAD SIMÓN BOLÍVAR**

**FACULTAD DE INGENIERÍAS**

**ESPECIALIZACIÓN EN GESTIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN**

**SAN JOSÉ DE CÚCUTA**

**2021**

## Contenido

<b>RESUMEN</b> .....	9
INTRODUCCIÓN.....	10
CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA.....	11
<b>Problema</b> .....	11
<b>Descripción situación existente</b> .....	13
<b>Árbol de problemas</b> .....	14
<b>Objetivos del proyecto</b> .....	16
CAPÍTULO 2. ESTADO DEL ARTE .....	17
<b>Antecedentes</b> .....	17
<b>Referentes teóricos</b> .....	18
<b>Marco conceptual</b> .....	27
CAPÍTULO 3. GESTION DE LOS STAKEHOLDER .....	30
<b>Equipo de trabajo del proyecto</b> .....	30
<b>Identificación de Stakeholders</b> .....	31
<b>Capacidad del proyecto en beneficiarios</b> .....	31
<b>Localización de desarrollo del proyecto</b> .....	32
CAPÍTULO 4. PLANIFICACION DEL PROYECTO .....	33
<b>Alcance del proyecto</b> .....	33
Criterios de aceptación del proyecto .....	34
<b>Metodología y plan de trabajo</b> .....	34

Metodología Agil .....	34
Plan de trabajo.....	36
<b>Herramientas de Software.....</b>	<b>40</b>
<b>EDT del Proyecto .....</b>	<b>41</b>
<b>Actividades del proyecto.....</b>	<b>42</b>
Descripción de Actividades del proyecto.....	42
<b>Cronograma Actividades.....</b>	<b>46</b>
<b>Características de los usuarios del sistema .....</b>	<b>50</b>
<b>Modelado del sistema.....</b>	<b>50</b>
Representación gráfica de procesos .....	50
Especificación de requisitos.....	52
Diagramas de caso de uso .....	55
Diagramas de secuencia.....	60
Diagrama de clases.....	65
Diagrama Entidad-relación .....	66
Diccionario de datos.....	67
Interfaz del aplicativo “prototipo” .....	71
<b>CAPÍTULO 5. GESTION DE LOS RECURSOS DEL PROYECTO .....</b>	<b>73</b>
<b>Organigrama del proyecto .....</b>	<b>73</b>
<b>Definición de roles.....</b>	<b>73</b>

<b>Estimación de recurso, costos y presupuesto .....</b>	<b>75</b>
Costo del proyecto .....	75
Presupuesto .....	77
<b>CAPÍTULO 6. EVALUACION DE LAS ALTERNATIVAS.....</b>	<b>78</b>
<b>Aseguramiento de control y calidad .....</b>	<b>78</b>
<b>Evaluación y control de riesgos.....</b>	<b>79</b>
Evaluación de los riesgos .....	80
Control de los riesgos.....	82
<b>Planificación de las comunicaciones .....</b>	<b>83</b>
<b>CONCLUSIONES.....</b>	<b>85</b>
<b>RECOMENDACIONES.....</b>	<b>86</b>
<b>REFERENCIAS .....</b>	<b>87</b>

## Lista de tablas

Tabla 1 factores de calidad .....	25
Tabla 2 equipo de trabajo .....	30
Tabla 3 stakeholders .....	31
Tabla 4 Criterios de aceptación.....	34
Tabla 5 herramientas de control y seguimiento .....	37
Tabla 6 ceremonia de metodología scrum.....	39
Tabla 7 herramientas de software .....	40
Tabla 8 cronograma de actividades .....	47
Tabla 9 Actor del sistema: docente.....	50
Tabla 10 Actor del sistema: estudiante.....	50
Tabla 11 Simbología diagrama de flujo.....	51
Tabla 12 requisito funcional autenticación de usuario.....	52
Tabla 13 requisito funcional gestionar clase.....	52
Tabla 14 requisito funcional generar código clase .....	53
Tabla 15 requisito funcional consultar asistencia .....	53
Tabla 16 requisito funcional gestionar estudiante .....	53
Tabla 17 requisito funcional registrar asistencia .....	53
Tabla 18 requisito no funcional seguridad.....	54
Tabla 19 requisito no funcional eficiencia .....	54
Tabla 20 requisito no funcional portabilidad .....	54
Tabla 21 requisito no funcional formato .....	54
Tabla 22 caso de uso autenticación .....	56
Tabla 23 caso de uso registrar asistencia.....	56
Tabla 24 caso de uso gestionar clase .....	58
Tabla 25 caso de uso registro manual.....	58
Tabla 26 caso de uso generar código de acceso .....	59
Tabla 27 caso de uso consultar asistencia .....	59
Tabla 28 especificaciones de entidad usuario .....	67
Tabla 29 especificaciones de entidad clase .....	67
Tabla 30 especificaciones de entidad horario.....	68

Tabla 31 especificaciones de entidad asignatura .....	68
Tabla 32 especificaciones de entidad persona .....	68
Tabla 33 especificaciones de entidad estudiante .....	69
Tabla 34 especificaciones de entidad docente .....	69
Tabla 35 especificaciones de entidad estuxclase.....	70
Tabla 36 especificaciones de entidad asistencia.....	70
Tabla 37 cargos .....	75
Tabla 38 Recurso humano.....	75
Tabla 39 Recurso tecnológico .....	76
Tabla 40 cotizacion Azure App Services .....	76
Tabla 41 presupuesto del proyecto opción 1.....	77
Tabla 42 presupuesto del proyecto opción 2.....	77
Tabla 43 actividades scrum .....	79
Tabla 44 descripción del riesgo.....	81
Tabla 45 matriz de priorización del riesgo .....	82
Tabla 46 matriz de priorización de los riesgos del proyecto.....	82
Tabla 47 control de los riesgos del proyecto .....	83
Tabla 48 PLAN DE COMUNICACIONES .....	84

## LISTA DE ILUSTRACIONES

Ilustración 1 árbol de problemas .....	14
Ilustración 2 árbol de objetivos.....	15
Ilustración 3 matriz de testing agil .....	23
Ilustración 4 Roles de Scrum .....	35
Ilustración 5 Metodología de Scrum .....	36
Ilustración 6 Ejemplo de Microsoft Planner.....	39
Ilustración 7 EDT del proyecto .....	41
Ilustración 8 fases del proyecto .....	42
Ilustración 9 esquema general del aplicativo móvil.....	48
Ilustración 10 Funcionalidad general del aplicativo.....	49
Ilustración 11 caso de uso estudiante.....	55
Ilustración 12 caso de uso docente.....	57
Ilustración 13 Diagrama de secuencia autenticación .....	60
Ilustración 14 Diagrama de secuencia registrar asistencia .....	61
Ilustración 15 Diagrama de secuencia gestionar clase .....	62
Ilustración 16 Diagrama de secuencia registro manual de estudiante.....	63
Ilustración 17 Diagrama de secuencia consultar asistencia.....	64
Ilustración 18 diagrama de clases .....	65
Ilustración 19 diagrama entidad relación .....	66
Ilustración 20 interfaz del aplicativo "prototipo" .....	71
Ilustración 21 interfaz del aplicativo "prototipo" .....	72
Ilustración 22 organigrama del proyecto.....	73
Ilustración 23 estructura gestión del riesgo.....	80

## RESUMEN

En la actualidad el uso de herramientas tecnológicas para la mejora y agilización de procesos a nivel organizacional a sido un gran complemento debido a el gran impacto que generan tanto a la organización como cada uno de los interesados, es por esto que se ha pensado en el desarrollo de una aplicación móvil para resolver el déficit de herramientas tecnológicas en planteles educativos de educación superior, debido a su gran capacidad de alumnos en estos, lo cual hace que procesos como tomas de asistencia a clases impartidas presenciales se realicen de manera rutinaria y con métodos muy convencionales. A través del uso de tecnologías como la planteada en este proyecto se busca mejorar de forma significativa tanto las experiencias de los usuario y docentes como también generar un impacto positivo dentro de la organización a nivel de administración de recursos financieros como también en el desarrollo de procesos. El aprovechamiento de las herramientas tecnológicas disponibles a nivel personal como los dispositivos móviles nos da la facilidad de desarrollar e implementar soluciones a través de estos, lo cual genera confiabilidad debido a la poca inversión de recursos en el proyecto, también es de suma importancia recalcar el uso de herramientas de desarrollo y de gestión innovadoras como la metodología ágil scrum facilitando todo el proceso de gestión y desarrollo, como también herramientas de software libre en este caso Flutter, Azure entre otras, las cuales hacen que el producto final sea confiable y trabaje de manera eficiente durante y después de la implementación.

Palabras claves: herramientas tecnológicas, aplicación móvil, confiabilidad, eficiencia.

## INTRODUCCIÓN

El presente proyecto tiene como finalidad el análisis y diseño de un aplicativo móvil especialmente para el control de asistencia dentro de las aulas en donde se imparten clases, dando solución a la falta de herramientas tecnológicas para gestionar procesos fuente donde vio la necesidad del desarrollar el proyecto.

Con el fin de cumplir el objetivo principal propuesto anteriormente se ah decidido aplicar en su desarrollo las buenas prácticas de la metodología Ágil basándonos en Scrum con el fin de utilizar bases mitológicas como mucho mayor eficiencia en el desarrollo de actividades en el cumplimiento de cada una de las etapas propuestas.

El documento cuenta con un análisis a nivel de ingeniería de software donde se detallan cada uno de los requerimientos establecidos para que el aplicativo cumpla sus funcionalidades, a su vez se analiza y documenta el modelado del sistema con el fin de tener las bases del sistema con el cual el equipo de trabajo establecido en la implementación va a trabajar a lo largo de todo el desarrollo, adicional se propone una planificación del proyecto donde se establece una metodología, plan de trabajo, equipo de trabajo y herramientas de software, la gestión de los recursos del proyecto y la evaluación alternativas donde podemos detallar el aseguramiento de la calidad, controles de riesgo y planificación de comunicaciones dentro del proyecto a implementar.

## **CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA**

### **Problema**

Dentro de las organizaciones interesadas en sistemas inteligentes se evidencia la necesidad de las instituciones educativas de educación superior, esto debido a la gran cantidad de docentes y estudiantes que las conforman, lo que conlleva a la existencia de un sistema de control que gestione automáticamente la asistencia a las aulas de clases el cual en la mayoría de estas se realiza de forma manual, dependiendo de personal y medios físicos para la realización, la cual evidencia una necesidad importante de contar con software automáticos que aportan grandes beneficios tanto para la institución como para sus estudiantes y docentes, mencionados a continuación:

- Minimizar los tiempos de toma de asistencia.
- Mejorar el control de la asistencia del estudiantado.
- Minimizar la inversión en recursos administrativos para las facultades.
- Contar con información en tiempo real de la asistencia para la toma de decisiones.
- Tener un mayor control de las inasistencias prolongadas y facilitar una mejor gestión de la deserción estudiantil.
- Minimizar la posibilidad de suplantación en la asistencia estudiantil.

Entre las necesidades más relevantes también encontramos la implementación de soluciones tecnológicas aplicando software de control debido al método rutinario y convencional utilizado por la gran mayoría de entidades el cual consiste en registrar la asistencia en formatos impresos entregados al docente para su posterior firma de cada uno de los asistentes a la clase los cuales son posteriormente dirigidos a las decanaturas y donde el encargado de la secretaria en dicha dependencia debe sistematizar una por una con el fin

de tener el control de asistencia y de clases impartidas por parte de los docentes, procesos como estos generan una mayor inversión tanto monetaria, tiempo y personal dedicado a la realización de estas tareas.

Herramientas como la propuesta cuya finalidad es controlar la asistencia estudiantil a través de un aplicativo móvil aporta valor agregado tanto en innovación tecnológica como en la reducción de costos a los cuales se le puede dar una utilidad distinta a la antes utilizada, la maximización de la eficiencia y productividad de los procesos de la organización también juega un papel importante no solo en la institución sino también en sus clientes y empleados generando mayor satisfacción, es de vital importancia mencionar que soluciones como la propuesta generan un impacto positivo dentro del medio ambiente reduciendo notoriamente el uso de papel en todo el plantel educativo.

### **Descripción situación existente**

Las instituciones educativas de educación superior dentro de sus procesos para la toma de asistencia cuentan con un sistema convencional basado en formatos básicos para cada clase impartida, donde se detalla la fecha, asignatura impartida por el docente, el docente quien dirigió la clase y el listado de estudiantes por clase, donde cada uno de ellos debe firmar, una vez llena el docente dispone para el uso de esta durante su clase.

El proceso posterior a la toma de asistencia en el formato es la entrega de la misma al finalizar la clase por medio del docente, esta reposa en las decanaturas de determinada carrera para su debida digitalización por parte de la secretaria con el fin de llevar el control de la asistencia, cabe recalcar que las listas no solo tienen esta finalidad, también tiene como sustento que dicha asignatura fue impartida por parte del docente y su evidencia corrobora el pago por parte de la universidad al docente por sus horas de trabajo.

## Árbol de problemas

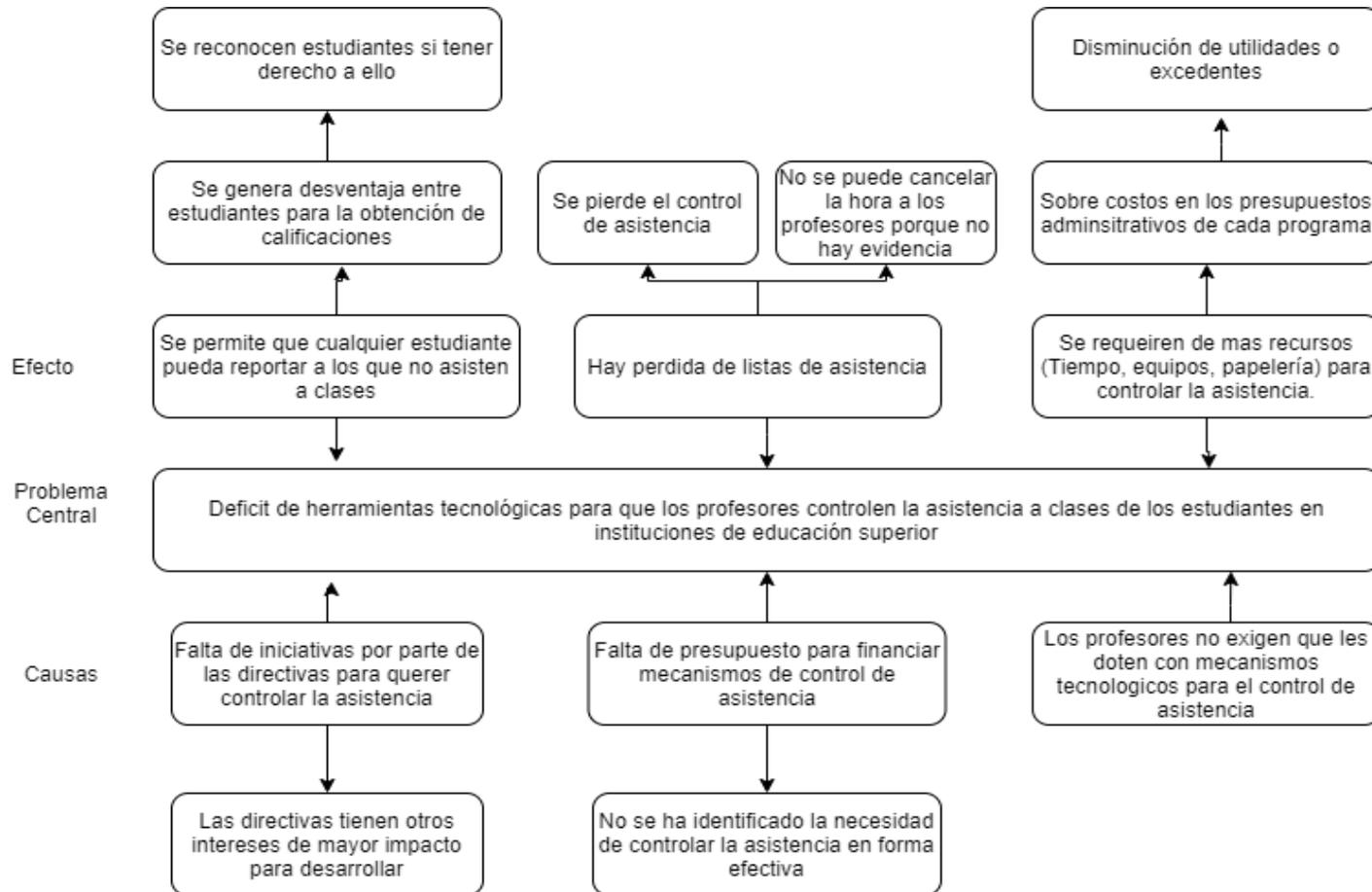


ILUSTRACIÓN 1 ÁRBOL DE PROBLEMAS

*Fuente: elaboración propia*

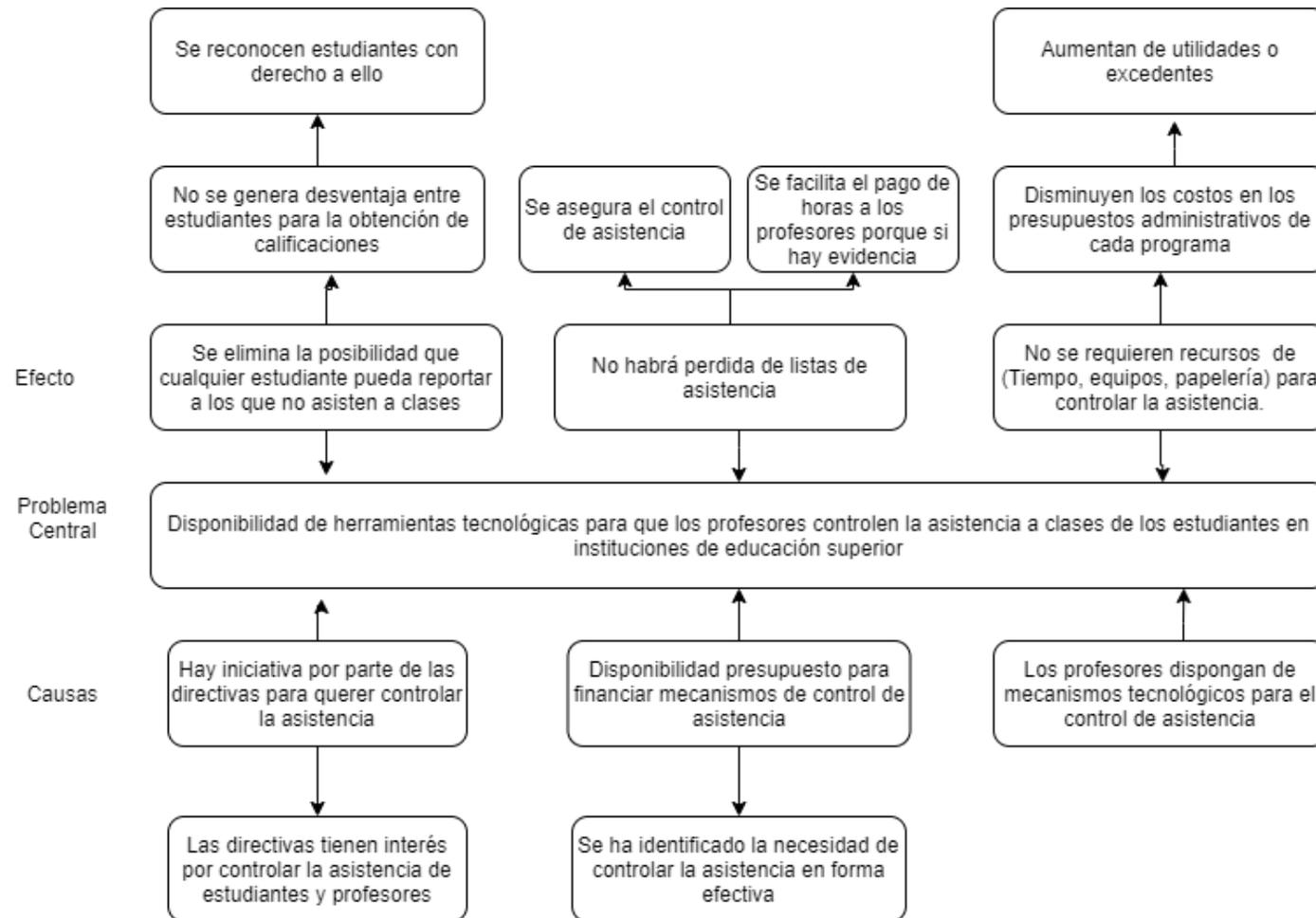


ILUSTRACIÓN 2 ÁRBOL DE OBJETIVOS  
Fuente: elaboración propia

## **Objetivos del proyecto**

En este apartado se presentan los objetivos generales y específicos correspondientes al proyecto

### **Objetivo general**

Diseñar un aplicativo móvil que permita el control y registro de asistencia por parte de los estudiantes en las instituciones educativas.

### **Objetivos específicos**

- Gestionar un mejor control en la asistencia tomada por parte de los docentes en las aulas.
- Disminuir los costos en los presupuestos administrativos de cada programa en el uso del papel impactando de forma positiva al medio ambiente.
- Disminuir la desventaja dentro de los estudiantes en relación con la asistencia por motivos de suplantación.
- Implementar soluciones tecnológicas en las aulas que generen mayor impacto en los procesos educativos y la satisfacción del cliente.

## CAPÍTULO 2. ESTADO DEL ARTE

### Antecedentes

El desarrollo de tecnologías aplicadas a sistemas con el fin de agilizar y modernizar procesos y tareas que se han venido realizando de forma manual han tenido una gran participación en la modernización de instituciones a nivel internacional como nacional se han venido desarrollando diferentes aplicativos y estudios para la puesta en marcha de sistemas capaz de sistematizar y modernizar procesos dentro de las entidades educativas a través de diferentes tecnologías. A continuación, se describen diferentes proyectos realizados a nivel internacional y nacional.

Estudiantes de master universitario en Ingeniería del software y sistemas informáticos de la Universidad internacional de la Rioja en España, plantearon el desarrollo de un sistema de asistencia estudiantil mediante reconocimiento facial a través de la integración de servicios Cloud de Amazon Web services, el cual utiliza el reconocimiento facial a través de imágenes o videos cargados, con el fin de sistematizar la asistencia de los alumnos en las aulas de manera masiva. (Salvatierra Tumbaco, Gabriel 2018).

Por su parte en la Universidad Politécnica de Madrid se planteó un sistema de control de asistencia utilizando un dispositivo adicional de tecnología bluetooth LTE llamado “active beacon” el cual se encargaría de generar esta señal en radios omnidireccionales a 100 metros, con el objetivo de asignar clases de manera remota, siendo el objetivo principal del sistema la capacidad del registro de asistencia automático de los alumnos previamente inscritos dentro de la clase generada por el docente, a su vez se cuenta con un registro manual como método de acceso para quienes tengan algún impedimento tecnológico para realizar el registro.(Sánchez Fernández, Miguel Ángel 2017).

## Referentes teóricos

En el desarrollo del marco teórico se aplican distintas teorías que desde el punto de vista profesional permiten la elaboración del contenido del proyecto, las cuales serán descritas a continuación:

### Ingeniería de software

Según Ian Sommerville, software representa “no sólo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que un programa funcione de forma apropiada y correcta”

Para el desarrollo de software hay métodos de ingeniería estructurados que incluyen modelos de sistemas, notaciones, reglas y guías de procesos que sirven como orientación en su correcto desarrollo, lo que conlleva al concepto de Ingeniería de Software

Ingeniería de software según Lawrence Pfieeger es un “conocimiento aplicado de las computadoras y la computación para resolver un problema identificado”, permitiendo desarrollar una metodología para cubrir el ciclo completo de un proyecto informático incluyendo desde la documentación hasta sus implementación y pruebas del mismo.

De acuerdo con lo anterior primero se debe identificar correctamente el problema con un debido y exhaustivo levantamiento de información para conocer el alcance real del problema y sus posibles soluciones, lo cual permite establecer la relación de los conceptos de Sommerville que define Ingeniería de Software como “una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza” y Presman que la define como “una disciplina o área de la informática o Ciencias de la

Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo”

### Metodologías ágiles

El concepto de metodología ágil aplicado al desarrollo de software nace después de una reunión que se realizó en febrero de 2001 en Utah-EEUU con 17 expertos de la industria de software, donde se discutió acerca de realizar un desarrollo de software rápido (Canós, Letelier, & Penadés, 2003) pero que a su vez cumpliera con las expectativas del cliente siendo flexible a los cambios que pueden generarse a través de la construcción del producto y que se centre en “Desarrollar software que funciona más que conseguir una buena documentación” (Canós, Letelier, & Penadés, 2003), siendo este uno de principios descritos en el manifiesto de la alianza creada por este grupo (The Agile Alliance, 2014).

En la última década el movimiento ágil ha crecido notablemente tanto en número de seguidores como en variantes metodológicas que han sido ampliamente difundidas en el sector académico e industrial; a continuación, se presenta un breve análisis de la selección de las principales metodologías ágiles, cuyo criterio de inclusión fueron aquellas reconocidas como las abanderadas del movimiento ágil:

- DSDM (Dynamic System Development Method) es el resultado de la conformación de un consorcio de empresas de TI de la gran Bretaña en el año 1994 sin fines de lucro, motivadas por la necesidad de comprender las mejores prácticas en el desarrollo de aplicaciones, con el fin de enseñarlos e implementarlos ampliamente entre sus miembros, (DSDM Consortium, 2003) no obstante a partir del año 2007 el

pleno apoyo de sus miembros decidieron hacer universalmente disponible DSDM para su lectura y uso. (DSDM Consortium, 2015).

- DSDM se concibe como un Framework para la entrega de proyectos que responde a las necesidades del negocio, proyectos DSDM se entregan dentro del cronograma, con el presupuesto y sin eliminar aspectos importantes.
- El DSDM es un Framework genérico iterativo e incremental basado en la metodología RAD (Rapid Application Development), enfocado en el involucramiento constante del usuario y los involucrados, con la premisa de entrega constantes del producto, en adición de otros principios considerados de la corriente de desarrollo ágil.
- SCRUM Su primera aparición en 1995 (Schawer, 1995) como una mejora al ciclo de desarrollo iterativo e incremental orientado a objetos, en la actualidad SCRUM es concebido como un marco de trabajo (Framework) para el desarrollo y mantenimiento de productos complejos. (SCRUM ORG, 2015) Anteriormente variantes de SCRUM para el proceso de desarrollo de nuevos productos en industrias manufactureras enfocados a un alto rendimiento y rapidez conformados por grupos pequeños fue observado en una serie de empresas japonesas. (Takeuchi & Nokana, 1986). El termino SCRUM es derivado del juego Rugby el cual hace referencia a una formación apretada de delanteros juntos entre sí en especificas posiciones cuando se inicia un reinicio del juego o Scrumdown. (World Rugby, 2015).
- XP La Extrem Programing (XP) desarrollada por Beck en 1996. (Wells, 2015) Propone una metodología ligera diseñada para grupos pequeños o medianos de desarrollo de software orientado a trabajar en un entorno donde los requerimientos

del proyecto puedan ser vagos o cambiar rápidamente. (Beck, Extreme Programming Explained, 1999) Si bien, esta fue la idea principal subyacente de XP en sus inicios, el autor y sus adeptos consideran que XP trata de una disciplina de desarrollo de software que cubre todos los niveles del proceso de desarrollo. (Beck & Andres, Extreme Programming Explained: Embrace Change, Second Edition, 2004).

### Manifiesto ágil

Es el documento que define los principios rectores de las metodologías ágiles de desarrollo de software, fue creado por un grupo de académicos y expertos de la industria del software reunido en Snowbird, Utah, Estados Unidos en febrero de 2001, a través de una iniciativa de Kent Beck.. Tal como lo aclara Cockburn (2007), ninguno estaba interesado en combinar las prácticas para crear una Metodología Liviana Unificada (Unified Light Methodology – ULM). La reunión buscaba un análisis y discusión de los elementos (valores y principios) que permitirían responder de una manera rápida y efectiva a los cambios que surgieran durante el desarrollo de un proyecto.

En esta reunión se decidió adoptar el término Ágil y como resultado se obtuvo un documento conocido como el Manifiesto Ágil (Manifiesto Ágil). El Manifiesto Ágil incluye cuatro postulados y una serie de principios asociados. Tal como hace observar Cockburn (2007), el propio manifiesto comienza diciendo que se están sacando a la luz mejores formas de desarrollar software, no las están inventando. Además, que el resultado se obtuvo de la experiencia directa y la reflexión sobre la experiencia y que se reconoce valor a las herramientas, procesos, documentación, contratos y planes, si bien ponen mayor énfasis en otros valores. Por esto es por lo que las metodologías ágiles cambian

significativamente algunos de los énfasis de las metodologías tradicionales lo que puede apreciarse en los postulados del manifiesto ágil.

El Manifiesto Ágil está soportado en 4 valores o postulados, cuyos conceptos fundamentales son:

- Las personas y su relacionamiento constante.
- Funcionalidad y menos documentación.
- El cliente involucrado en todas las etapas del proceso.
- Soluciones rápidas y efectivas ante los cambios permanentes

#### Pruebas de software con metodología ágil

Pruebas ágiles o agile testing son técnicas esenciales en el desarrollo de software que implementan los principios y prácticas ágiles, estas mejoran la agilidad porque nos permite crear un producto con un mejor diseño, más entendible y en general de mejor calidad. Esto promueve un fácil mantenimiento y la adaptabilidad en un ambiente súper cambiante. Lo que nos lleva a menores costos financieros. Las pruebas ágiles se basan en principios ágiles, lo podemos ver en el proceso empírico de scrum y sus tres pilares:

- Transparencia
- Inspección
- Adaptación

Cuando hablamos de pruebas ágiles hablamos de varias técnicas y de diferentes tipos de pruebas, como pueden ser TDD, BDD, las pruebas unitarias, funcionales, de integración, de rendimiento, de usabilidad, etc, estas maximizan el valor de la solución y minimizan los

riesgos, esto se consigue fallando inmediatamente de manera segura para aprender y adaptarse rápidamente.

Con esto último tenemos cuatro aspectos fundamentales a la hora de hacer pruebas:

- Perspectiva tecnológica
- Perspectiva de negocio
- Soporte y guía al desarrollo
- Crítica del producto

Esos cuatro aspectos se representan en los siguientes cuadrantes:

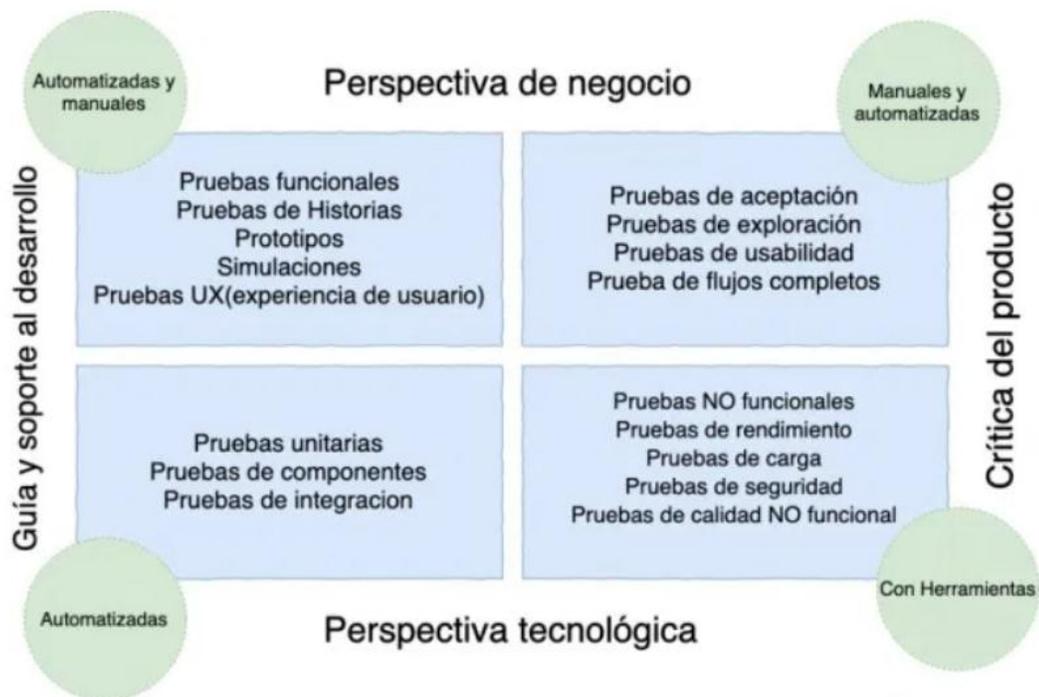


ILUSTRACIÓN 3 MATRIZ DE TESTING AGIL

Fuente: extraído de: <https://www.pensemosweb.com>

## Calidad en el software

Según las Normas UNE la calidad es definida como “el conjunto de propiedades y características de un producto ó servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas”, es por esto que se afirma si un software cumple todos los requisitos exigidos por el cliente y cada uno de sus procesos se ejecuten de manera correcta, se afirma que se está garantizando la calidad del software. La calidad dentro del software implica la utilización de procedimientos estándares para el análisis, diseño, programación y prueba del software con el fin de lograr aspectos de aseguramiento de calidad, a su vez existen factores de calidad analizados para conseguir su propósito, los cuales se definen a continuación:

<b>FACTOR DE CALIDAD</b>	<b>DEFINICIÓN</b>
<b>Cumplimiento</b>	El grado en que un programa satisface sus especificaciones y consigue los objetivos de la misión encomendada por el cliente.
<b>Fiabilidad</b>	El grado en que se puede esperar que un programa lleve a cabo sus funciones esperadas con la precisión requerida.
<b>Eficiencia</b>	La cantidad de recursos de hardware y de código requerido por un programa para realizar su función.
<b>Integridad</b>	El grado en que puede controlarse el acceso al software o a los datos por personas no autorizadas.
<b>Facilidad de uso</b>	El esfuerzo requerido para aprender, trabajar, preparar la entrada e interpretar la salida de un programa.
<b>Facilidad de mantenimiento</b>	El esfuerzo requerido para localizar y arreglar un mantenimiento error en un programa.
<b>Facilidad de prueba</b>	El esfuerzo requerido para probar un programa de prueba forma que se asegure que realiza la función requerida.
<b>Portabilidad</b>	El esfuerzo requerido para transferir el programa desde una configuración de hardware o sistema operativo a otro
<b>Reusabilidad</b>	El grado en que un programa (o partes de él) se pueden reutilizar en otras aplicaciones.
<b>Facilidad de interoperación</b>	El esfuerzo requerido para acoplar un sistema interoperación. a otro.
<b>Facilidad de auditoría</b>	Facilidad con que se puede comprobar la conformidad con los estándares.

<b>Exactitud</b>	La precisión en los cálculos y el control.
<b>Normalización de las comunicaciones</b>	El grado en que se usan el ancho de banda, las comunicaciones. protocolos y las interfases estándar
<b>Compleitud</b>	El grado en que se ha conseguido la total implementación de las funciones requeridas.
<b>Concisión</b>	Lo compacto que es el programa en términos de Líneas de código.
<b>Consistencia</b>	El uso de un diseño uniforme y de técnicas de documentación.
<b>Estandarización datos</b>	El uso de estructuras de datos y de tipos datos estándar.
<b>Tolerancia de errores</b>	El daño que se produce cuando el programa encuentra un error.
<b>Eficiencia en la ejecución</b>	El rendimiento en tiempo de ejecución de un programa.
<b>Facilidad de expansión</b>	El grado en que se puede ampliar el diseño arquitectónico de datos.
<b>Generalidad</b>	La amplitud de aplicación potencial de los componentes del programa.
<b>Independencia del hardware</b>	El grado en que el software es independiente del hardware hardware que usa.
<b>Instrumentación</b>	El grado en que el programa muestra su propio funcionamiento e identifica errores que aparecen.
<b>Modularidad</b>	Grado de facilidad de operación.
<b>Facilidad de operación</b>	La disponibilidad de mecanismos que controlen o protejan los programas o los datos.
<b>Seguridad</b>	Independencia funcional de los componentes del programa.
<b>Auto documentación</b>	El grado en que el código fuente proporciona documentación significativa. El grado en que un programa puede ser entendido sin dificultad.
<b>Simplicidad</b>	La posibilidad de seguir la pista de la representación del diseño de los componentes reales del programa hacia atrás.
<b>Facilidad de trazo</b>	El grado en que el software ayuda a permitir que nuevos usuarios apliquen el sistema.

**TABLA 1 FACTORES DE CALIDAD**  
*Fuente: elaboración propia*

## Marco legal

El objetivo de desarrollar este estudio es poder demostrar que todo lo que realiza y realizara en este proyecto, por los desarrolladores, no transgredieran la ley que impida el desarrollo adecuado del sistema de control y registro, finalmente cabe mencionar que el presente proyecto se está llevando a cabo a algunos artículos de ley que se rige en actualmente en la ley colombiana como la son:

- Decreto número 1360 de 1989, por el cual se reglamenta la inscripción del soporte lógico (software) en el Registro Nacional del Derecho de Autor.
- Ley 1915 de 2018, por la cual se modifica la Ley 23 de 1982 y se establecen otras disposiciones en materia de derecho de autor y derechos conexos.
- Ley 1581 del 17 de octubre 2012 de PROTECCIÓN DE DATOS PERSONALES, sancionada siguiendo los lineamientos establecidos por el Congreso de la República y la Sentencia C-748 de 2011 de la Corte Constitucional.

Como resultado de la sanción de la anunciada ley toda entidad pública o privada, cuenta con un plazo de seis meses para crear sus propias políticas internas de manejo de datos personales, establecer procedimientos adecuados para la atención de peticiones, quejas y reclamos, así como ajustar todos los procesos, contratos y autorizaciones a las disposiciones de la nueva norma.

Aspectos claves de la normatividad:

- Cualquier ciudadano tendrá la posibilidad de acceder a su información personal y solicitar la supresión o corrección de la misma frente a toda base de datos en que se encuentre registrado.
- Establece los principios que deben ser obligatoriamente observados por quienes hagan uso, de alguna manera realicen el tratamiento o mantengan una base de datos con información personal, cualquiera que sea su finalidad.
- Aclara la diferencia entre clases de datos personales construyendo las bases para la instauración de los diversos grados de protección que deben presentar si son públicos o privados, así como las finalidades permitidas para su utilización.

- Crea una especial protección a los datos de menores de edad.
- Establece los lineamientos para la cesión de datos entre entidades y los procesos de importación y exportación de información personal que se realicen en adelante.
- Define las obligaciones y responsabilidades que empresas de servicios tercerizados tales como Call y Contact Center, entidades de cobranza y, en general, todos aquellos que manejen datos personales por cuenta de un tercero, deben cumplir en adelante.
- la vigilancia y control de las bases de datos personales a la ya creada Superintendencia Delegada para la Protección de Datos Personales, de la Superintendencia de Industria y Comercio.
- Crea el Registro Nacional de Bases de Datos.
- Establece una serie de sanciones de carácter personal e institucional dirigidas a entidades y funcionarios responsables del cumplimiento de sus lineamientos.

### **Marco conceptual**

**API:** El término API es una abreviatura de Application Programming Interfaces, que en español significa interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

**Front-End:** Es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando

accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas(RWD), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales... y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario.

**Back-End:** Se encarga de todos los procesos necesarios para que la web funcione de forma correcta. Estos procesos o funciones no son visibles, pero tienen mucha importancia en el buen funcionamiento de un sitio web. Algunas de estas acciones que controla el backend son la conexión con la base de datos o la comunicación con el servidor de hosting.

**Framework:** Del inglés framework. Un marco de trabajo es un conjunto de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

**Web Service, o Servicio Web:** Es un método de comunicación entre dos aparatos electrónicos en una red. Es una colección de protocolos abiertos y estándares usados para intercambiar datos entre aplicaciones o sistemas.

**Plugins:** Son pequeños programas complementarios que amplían las funciones de aplicaciones web y programas de escritorio. Por norma general, cuando instalamos un plugin, el software en cuestión adquiere una nueva función.

**NPM (Node Package Manager):** Es un gestor de paquetes desarrollado en su totalidad bajo el lenguaje JavaScript por Isaac Schlueter, a través del cual podemos obtener cualquier librería con tan solo una sencilla línea de código, lo cual nos permitirá agregar dependencias de forma simple, distribuir paquetes y administrar eficazmente tanto los módulos como el proyecto a desarrollar en general.

Flutter: Es un framework de código abierto desarrollado por Google para crear aplicaciones nativas de forma fácil, rápida y sencilla. Su principal ventaja radica en que genera código 100% nativo para cada plataforma, con lo que el rendimiento y la UX es totalmente idéntico a las aplicaciones nativas tradicionales.

Azure: Es una nube pública de pago por uso que te permite compilar, implementar y administrar rápidamente aplicaciones en una red global de datacenters (centros de datos) de Microsoft.

Aplicación móvil: también llamada app móvil, es un tipo de aplicación diseñada para ejecutarse en un dispositivo móvil, que puede ser un teléfono inteligente o una tableta. Incluso si las aplicaciones suelen ser pequeñas unidades de software con funciones limitadas, se las arreglan para proporcionar a los usuarios servicios y experiencias de calidad.

### CAPÍTULO 3. GESTIÓN DE LOS STAKEHOLDER

#### Equipo de trabajo del proyecto

A continuación, se define el equipo de trabajo conformado y relacionado directamente con las fases del proyecto:

FASES DEL PROYECTO	ROL
<b>Inicial</b>	- Director del proyecto
<b>Análisis</b>	- Director del proyecto - Desarrollador - BDA
<b>Diseño</b>	- Director del proyecto - Desarrollador
<b>Implementación de servicios web</b>	- Director del proyecto - Desarrollador - BDA
<b>Desarrollo</b>	- Desarrollador - BDA
<b>Pruebas</b>	- Director del proyecto - QA

**TABLA 2 EQUIPO DE TRABAJO**

*Fuente: elaboración propia*

## Identificación de Stakeholders

A continuación, se define la matriz de stakeholders que se relacionan directamente con el proyecto.

<b>PARTES INTERESADAS</b>	<b>NECESIDADES/REQUISITOS</b>	<b>EXPECTATIVA</b>
<b>Director del proyecto</b>	- Capacidad para monitorizar y controlar el proyecto	- Cooperación con el cliente - Presente en todas las fases del proyecto - Comunicación regular con el sponsor
<b>Equipo de trabajo</b>	- Conocimiento previo de su área de trabajo.	- comunicación regular con el director del proyecto
<b>Institución universitaria</b>	- Aprobación del proyecto - Financiación	- Implementación del proyecto - Cooperación con información requerida de los sistemas de información
<b>Estudiantes y docentes</b>	- Disposición para la utilización del aplicativo móvil	- Usabilidad del sistema

**TABLA 3 STAKEHOLDERS**

*Fuente: elaboración propia*

## Capacidad del proyecto en beneficiarios

Los beneficios que se pueden obtener aplicando el proyecto como prototipo de aplicación para la institución, docentes y estudiantes son los siguientes:

- Este tipo de aplicación móvil nos puede ayudar facilitando la comunicación entre el docente y el estudiante.
- Acceder consultar la información de las clases cualquier hora del día (24/7).
- El docente sea una guía a través de este proceso y la tecnología.

- La aplicación permita hasta 1000 conexiones simultánea en Azure, en el caso de una saturación se replicará otro servidor para normalizar el tráfico sin verse afectado la aplicación.
- La aplicación estará disponible para versiones de iOS y Android desde KitKat 4.4
- El estudiante podrá registrar su asistencia de clases, en cualquier espacio de tiempo y lugar, garantizando la movilidad de usuario.

### **Localización de desarrollo del proyecto**

La localización del proyecto se encuentra planteada para los planteles educativos de educación superior teniendo en cuenta que el propósito principal del proyectos es brindar una solución tecnológica que brinde control y versatilidad a la hora de su uso sin depender estrictamente de restricciones físicas o del infraestructura altamente compleja para su desarrollo podemos implementarlo en cualquier aula de clase o desde la virtualidad con la única restricción de portar con una conexión fija a internet, esto debido a la necesidad del aplicativo de conectarse internamente con los sistemas de datos e información de la universidad.

## **CAPÍTULO 4. PLANIFICACIÓN DEL PROYECTO**

### **Alcance del proyecto**

El propósito de este documento es presentar un plan de desarrollo de software basado en especificación de requerimientos, análisis, planificación, diseño y documentación que están constituidos por un conjunto de procesos que van desde la identificación de necesidades del usuario hasta la entrega del documento final, la necesidad de crear este sistema es de computarizar y automatizar la asistencia del personal para llevar un control adecuado y así brindarles un mejor servicio y ahorrar tiempo y recursos a la entidad.

Para ello se pretende hacer un análisis del sistema tomando en cuenta todos los requerimientos y necesidades de los usuarios y así ofrecer una solución factible. Para esto la aplicación móvil realizará la función principal de toma de asistencia a través de un código único para la clase generado por el docente en su respectivo modulo, este le dará acceso al estudiante a su respectiva clase donde hará su registro de asistencia, dicha aplicación debe ser práctica, versátil y multiplataforma, además la aplicación permitirá crear varias sesiones para docentes y estudiantes mientras mantenga en ejecución el App lo que le permitirá tener múltiples secciones y estar conectado todo tiempo, no importara el número de usuarios conectados, todo dependerá el número de peticiones que se hagan realicen a los servidores con tiempo de respuesta de hasta 1 milisegundo.

Finalmente, la aplicación estará disponible 24/7, solo se podrá interactuar cuando este entre los horarios establecidos por la institución. De igual manera el aplicativo no guardara información en dispositivo utilizado, todas peticiones de guardado se realizarán en tiempo real quedando guardado en las bases de datos de la institución.

## Criterios de aceptación del proyecto

A continuación, se definen los criterios de aceptación a través de una serie de objetivos del proyecto.

OBJETIVOS	CRITERIOS DE ACEPTACIÓN
Elaborar un proyecto de ejecución viable en base a los requerimientos establecidos	El sponsor recibe el documento del proyecto y autoriza su desarrollo e implementación
Diseño del aplicativo móvil acorde a los requerimientos establecidos	Análisis y diseño completo dentro del tiempo y presupuesto establecido del proyecto
Portabilidad del Aplicativo móvil	La aplicación debe ser accesible desde los sistemas IOs y Android, siendo así multiplataforma.
Aplicativo ágil y eficiente	El aplicativo debe cumplir con las pruebas de aceptación del usuario y sus tiempos de respuesta en operación no deben ser mayores a 3 seg.

**TABLA 4 CRITERIOS DE ACEPTACIÓN**

*Fuente: elaboración propia*

## Metodología y plan de trabajo

### Metodología Ágil

La metodología de desarrollo de software se puede definir como un conjunto de técnicas, recomendaciones y verificaciones, que permitan sistematizar los procesos en los que se descompone el proyecto (M. O. Herrera, 2010). Para este proyecto se ha decidido usar una metodología Ágil como Scrum, siendo muy utilizada para desarrollos de software, Scrum se rige por un proceso iterativo que reparte el desarrollo de un producto en distintas iteraciones de unas pocas semanas llamadas sprints. En cada sprint el equipo trabaja sobre una lista de requisitos previamente acordados obteniendo así, al acabar la iteración, un producto entregable. Este enfoque permite optimizar el control de riesgos y aumentar la

previsibilidad debido a que, al tener un producto que pueda ver el cliente, se puede proporcionar un mejor feed back y comprobar si se están cumpliendo sus expectativas.

Los equipos de desarrollo de Scrum son autogestionados y multifuncionales, es decir, en cada equipo debe haber alguno de los miembros que sepa realizar cada una de las tareas a hacer, la idea es tener dos Sprints por semana, lo que va permitir realizar pruebas y correcciones sobre el desarrollo del aplicativo donde se involucre a las partes interesadas (Stakeholder) para que se pueda comprobar y verificar el desarrollo del proyecto.

Los roles básicos de Scrum son los cuatro que se muestran a continuación



ILUSTRACIÓN 4 ROLES DE SCRUM

*Fuente: softeng.es*

- Product Owner: Es la persona que representa al Cliente. Es el responsable de organizar las diferentes tareas que se deben realizar, priorizando las más importantes y gestionando las distintas características del producto. Este rol lo ha desempeña el director de TI.
- Scrum Master: Es la persona que se encarga de la gestión del proyecto. Es el responsable de que el Scrum Team entienda lo que tienen que hacer además de

asegurarse de que haya un buen ambiente de trabajo. Este rol lo ha desempeña de Analista Software designado a este proyecto.

- Scrum Team: Es el equipo encargado de desarrollar el proyecto. No hay una jerarquía interna debido a que el equipo se debe autogestionar. Este es el rol lo desempeña todo el equipo del proyecto.

A continuación, se muestra el flujo de trabajo en la metodología Scrum.

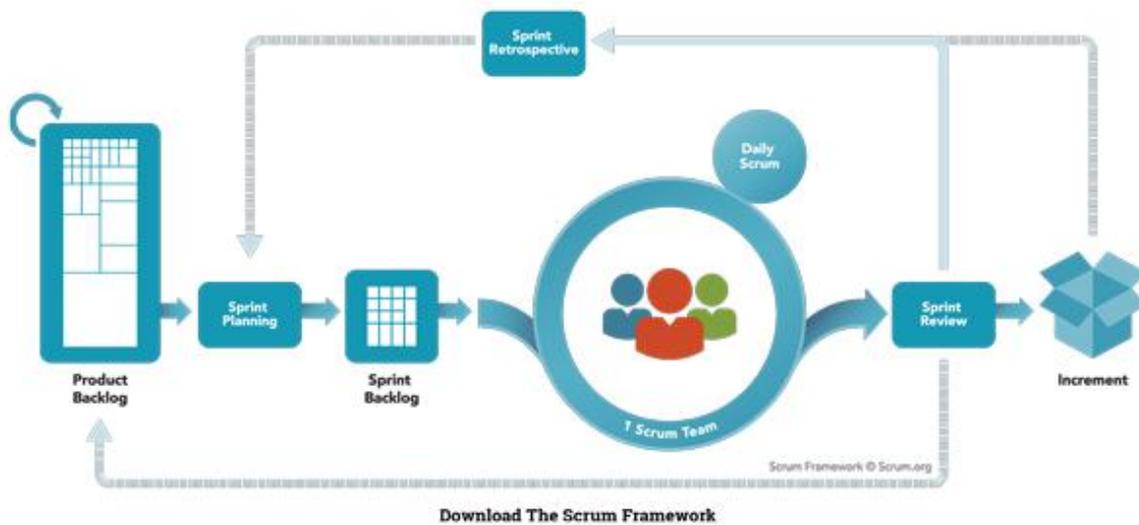


ILUSTRACIÓN 5 METODOLOGÍA DE SCRUM

Fuente: scrum.org

### Plan de trabajo

Con el fin de guiar el uso y la implementación de la metodología Scrum, se propone los siguientes plantillas y buenas prácticas para realizar la inicialización, planeación, ejecución y cierre del proyecto.

HERRAMIENTAS DE CONTROL Y SEGUIMIENTO	FUNCIÓN
<b>Visión del proyecto</b>	Incluye información sustancial sobre los antecedentes del proyecto, los objetivos del negocio y los resultados deseados, un reporte de análisis FODA y de brecha, una lista de los riesgos identificados y las estimaciones de tiempo, esfuerzo y costo
<b>Plantilla Historias Usuario</b>	Se llevan a cabo como parte del proceso de Desarrollo esencial. El Scrum Master facilita estas sesiones.
<b>Plantilla Sprint y Backlog</b>	Permite identificar las historias de usuario como actividades, asignando los tiempos y responsables para su elaboración.

**TABLA 5 HERRAMIENTAS DE CONTROL Y SEGUIMIENTO**

*Fuente: elaboración propia*

A continuación, se define una serie de actividades realizadas en la metodología Scrum definidos como ceremonias.

CEREMONIA	OBJETIVO	DURACIÓN
<b>El Sprint</b>	<ul style="list-style-type: none"> <li>- El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto "Terminado" utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior.</li> </ul>	<ul style="list-style-type: none"> <li>- No se realizan cambios que puedan afectar al Objetivo del Sprint.</li> <li>- Los objetivos de calidad no disminuyen.</li> <li>- El alcance puede clarificarse y renegociarse entre el Dueño de Producto y el Equipo de desarrollo.</li> </ul>
<b>Planificación del Sprint (Sprint Planning)</b>	<ul style="list-style-type: none"> <li>- El trabajo para realizar durante el Sprint se detalla en la Planificación de Sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo.</li> <li>- La Planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes.</li> <li>- Para Sprints más cortos el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al</li> </ul>	<p>La Planificación de Sprint responde a las siguientes preguntas:</p> <ul style="list-style-type: none"> <li>- ¿Qué puede entregarse en el Incremento resultante del Sprint que comienza?</li> <li>- ¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?</li> </ul>

	Equipo Scrum a mantenerse dentro del bloque de tiempo.	
<b>Scrums Diarios (Daily Scrums)</b>	<ul style="list-style-type: none"> <li>- El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para el Equipo de Desarrollo, se lleva a cabo cada día del sprint. el Equipo de Desarrollo planea el trabajo para las siguientes 24 horas. Esto optimiza la colaboración y el desempeño del equipo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección del trabajo del Sprint a realizar a continuación, se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad.</li> <li>- El Equipo de Desarrollo usa el Scrum Diario para evaluar el progreso hacia el Objetivo del Sprint y para evaluar qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la Lista de Pendientes del Sprint.</li> <li>- Este también optimiza las posibilidades de que el Equipo de Desarrollo cumpla el Objetivo del Sprint. Cada día, el Equipo de Desarrollo debería entender cómo intenta trabajar en conjunto como un equipo autoorganizado para lograr el Objetivo del Sprint y crear el Incremento esperado hacia el final del Sprint.</li> </ul>	<p>El Equipo de Desarrollo es el encargado de establecer la estructura de la reunión y esta se puede conducir de diferentes maneras si se enfoca en el progreso hacia la Meta de Sprint. Algunos Equipos de Desarrollo usarán preguntas, algunos se basarán más en discusiones. Aquí hay un ejemplo de lo que podría usarse:</p> <ul style="list-style-type: none"> <li>- ¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr el Objetivo del Sprint?</li> <li>- ¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr el Objetivo del Sprint?</li> <li>- ¿Veo algún impedimento que evite que el Equipo de Desarrollo o yo logremos el Objetivo del Sprint?</li> </ul>
<b>Retrospectiva del Sprint (Sprint Retrospective)</b>	<ul style="list-style-type: none"> <li>- La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y de crear un plan de mejoras que sean abordadas durante el siguiente Sprint.</li> <li>- La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Planificación de Sprint. Se trata de una reunión de, a lo sumo, tres horas para Sprints de un mes. Para Sprints más cortos el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito</li> </ul>	<p>El propósito de la Retrospectiva de Sprint es:</p> <ul style="list-style-type: none"> <li>- Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas.</li> <li>- Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras.</li> <li>- Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.</li> </ul>
<b>Revisión del Sprint (Sprint Review)</b>	<ul style="list-style-type: none"> <li>- Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Basándose en esto y en cualquier cambio a la Lista de Producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.</li> <li>- Se trata de una reunión de, a lo sumo, cuatro horas para Sprints de un mes. Para Sprints más cortos, el evento usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master</li> </ul>	<p>La Revisión de Sprint incluye lo siguiente:</p> <ul style="list-style-type: none"> <li>- Los asistentes son el Equipo Scrum y los interesados clave invitados por el Dueño de Producto.</li> <li>- El Dueño de Producto explica qué elementos de la Lista de Producto se han “Terminado” y cuales no se han “Terminado”.</li> <li>- El Equipo de desarrollo habla acerca de qué estuvo bien durante el Sprint qué problemas aparecieron y cómo fueron resueltos.</li> <li>- El Equipo de Desarrollo hace una demostración del trabajo que ha “Terminado” y</li> </ul>

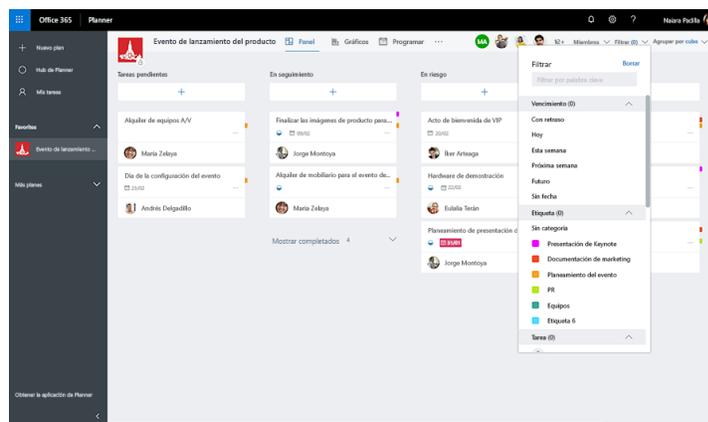
	<p>enseña a todos a mantener el evento dentro del bloque de tiempo fijado.</p>	<p>responde preguntas acerca del Incremento.</p> <ul style="list-style-type: none"> <li>- El grupo completo colabora acerca de qué hacer a continuación, de modo que la Revisión del Sprint proporcione información de entrada valiosa para Reuniones de Planificación de Sprints subsiguientes.</li> <li>- Revisión de cómo el mercado o el uso potencial del producto podría haber cambiado</li> <li>- Revisión de la línea de tiempo, presupuesto, capacidades potenciales y mercado para las próximas entregas de funcionalidad o capacidad prevista del producto.</li> </ul>
--	--	---

**TABLA 6 CEREMONIA DE METODOLOGÍA SCRUM**

*Fuente: Manifiesto Ágil, 2001*

Las herramientas para seguir dicha metodología son las siguientes:

- **Microsoft Planner:** Es una aplicación con un tablero Kanban que nos permite incluir las tareas a realizar y asignarlas a los respectivos responsables.
- **Microsoft Teams:** Es un software basado en un chat de trabajo, que permite realizar reuniones. Permite integrarse con Microsoft planner.



**ILUSTRACIÓN 6 EJEMPLO DE MICROSOFT PLANNER**

Fuente: microsoft.com

## Herramientas de Software

Para el desarrollo del proyecto se utilizará los siguientes programas como:

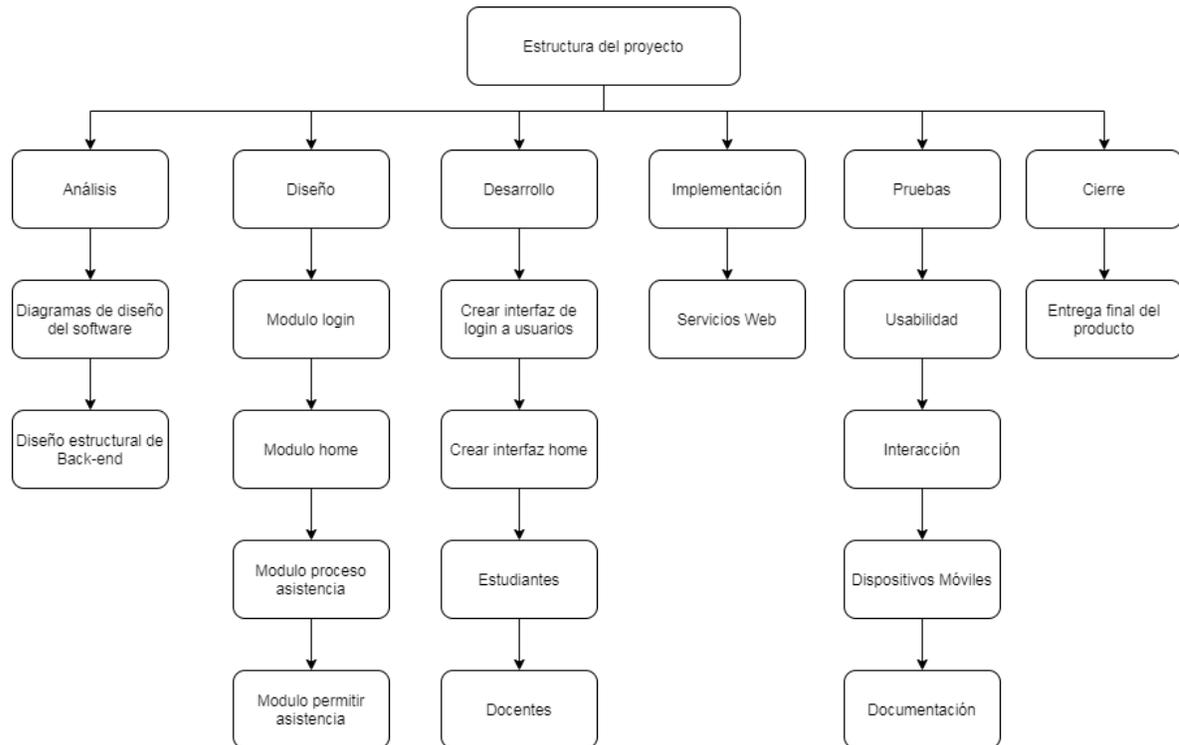
Nombre	Versión	Definición	Motivo
<b>Postman</b>	8.7.0	Es una herramienta que se utiliza, sobre todo, para el testing de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de sistemas	Utilizada para probar API REST
<b>IntelliJ IDEA</b>	2021.1.2	Es un entorno de desarrollo integrado para el desarrollo de programas informáticos. Es desarrollado por JetBrains, y está disponible en dos ediciones: edición para la comunidad y edición comercial	Utilizado para implementación de servicios elaborados en Spring Boot
<b>Visual Studio Code</b>	1.57	Es un editor de código fuente que permite trabajar con diversos lenguajes de programación, admite gestionar tus propios atajos de teclado y refactorizar el código	Utilizado para creación de proyectos en Flutter
<b>GitHub</b>	Online	Plataforma de gestión y organización de proyectos basada en la nube que incorpora las funciones de control de versiones	Utilizado para mejor control de versiones de los proyectos
<b>Balsamiq</b>	4.2.5	herramienta para hacer prototipos de proyectos.	Utilizada para crear Mockups o Wireframing para la creación de estructura de interfaces
<b>Dbeaver Community</b>	21.1.2	Beaver es un gestor universal de BBDD multiplataforma, que ofrece soporte a las bases de datos más conocidas del mercado (MySQL, Oracle, DB2, SQL Server, PostgreSQL, etc ..) , así como algunas NoSQL (MongoDB, Cassandra).	Herramienta de base de datos universal, utiliza para realiza la conexión con la base de datos para una mejor administración
<b>Spring Boot</b>	2.5.2	Spring Boot permite compilar nuestras aplicaciones Web como un archivo .jar que podemos ejecutar como una aplicación Java normal (como alternativa a un archivo .war, que desplegaríamos en un servidor de aplicaciones como Tomcat).	Utilizada para la creación del proyecto y su configuración para creación de los servicios rest

**TABLA 7 HERRAMIENTAS DE SOFTWARE**

*Fuente: elaboración propia*

## EDT del Proyecto

A continuación, se presenta la estructura propuesta para el proyecto y las actividades que se contemplan dentro de cada una de ellas.



**ILUSTRACIÓN 7 EDT DEL PROYECTO**

*Fuente: elaboración propia*

## Actividades del proyecto

Las actividades propuestas dentro del proyecto se encuentran enmarcadas dentro de un grupo de fases como se presentan a continuación:



ILUSTRACIÓN 8 FASES DEL PROYECTO

*Fuente: elaboración propia*

### Descripción de Actividades del proyecto

#### *Fase de análisis*

Dentro del proceso de análisis, es fundamental que, a través de una colección de requerimientos funcionales y no funcionales, el desarrollador o desarrolladores del software comprendan completamente la naturaleza de los programas que deben construirse para desarrollar la aplicación, la función requerida, comportamiento, rendimiento e interconexión. (Pressman,2002). Es de suma importancia que antes de empezar a codificar los programas, se tenga una completa y plena comprensión de los requisitos del software. Por eso dentro del cronograma de actividades se contempla hacer un empalme de la Institución de educación superior en tecnología para conocer tu contexto en cuanto tablas o capacitaciones al personal en caso de que se requiera.

### *Fase de diseño*

El diseño del software es un proceso iterativo por el cual, con los requisitos elaboramos un prototipo para construir el software. De acuerdo con Perssman el objetivo del diseño es producir un modelo o representación de una entidad que se va construir posteriormente. (Perssman,2002).

Para lo cual se propone un diseño creado en con la herramienta de Balsamiq para su posterior construcción, además la utilización se frameworks flutter par a creación de las vistas que se va a mostrar en del aplicativo.

### *Fase de Implementación de servicios web*

En esta fase hay que elegir las herramientas adecuadas, un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de software a construir. Lo cual también se podrá realizar la configuración de las máquinas a utilizar y seleccionar las herramientas candidatas para su posterior desarrollo, para la implementación de los servicios web que desarrollara en lenguaje de programación Java para creación de servicios REST en Sprint Boot, una vez creado cada servicio serán probados mediante Postman, al mismo tiempo con DBA se llevara acabo el manejo de las tablas a utilizar con los permisos requeridos para sus consultar permitiendo una mejor admiración de acuerdo como se tenga en entidad. Si se contrata los servicios de Azure app services, para montar los servicios implementados en la nube para su aprobación.

### *Fase de desarrollo*

En esta etapa se construye la aplicación. El desarrollo se realiza por un equipo de programadores. Se empezará a construir el aplicativo partiendo de los modelos creados de la fase de diseño para crear los componentes del sistema.

Para la construcción de software se utiliza framework de flutter como una herramienta multiplataforma para que el aplicativo funcione en los dos sistemas operativos como lo es Android y iOS.

También utilizar una herramienta de manejador de versiones como lo es github en un repositorio privado, para control del desarrollo del aplicativo.

### *Fase de pruebas*

En esta fase definimos las pruebas a realizar dentro del desarrollo del proyecto las cuales se encuentran enfocadas en 3 tipos diferentes, mencionadas a continuación:

- Pruebas funcionales: El objetivo de las pruebas funcionales es comprobar que el software que se ha creado cumple con la función para la que se había pensado, desde la perspectiva del cliente y no del desarrollador, como se pueden evidenciar en las actividades propuestas para la fase de pruebas del proyecto.
- Pruebas unitarias: se pueden realizar tanto a nivel de back-end como de Front-end, en estos casos y teniendo en cuenta la utilización de las librerías de flutter podemos acceder a su paquete flutter\_test o mockito según sea la necesidad, las cuales nos proporciona utilidades para la realización de estas pruebas.
- Pruebas de integración y de aceptación: técnicas realizadas para probar cómo es la interacción entre dos o más unidades del software en el caso de las de integración y en

cuanto a las de aceptación para comprobar si el software cumple con las expectativas del cliente, con lo que el cliente realmente pidió.

- Pruebas de performance: Las pruebas de estrés deben hacerse con frecuencia, en cada sprint. Cada vez que se haga una funcionalidad, Validan la escalabilidad, estabilidad y velocidad (tiempos de respuesta) del sistema bajo evaluación. Niveles de uso de los recursos que cumplan con los objetivos de desempeño para el sistema.



Crear interfaz de registro de asistencia	6	1	0%	
Crear interfaz de registro manual	7	1	0%	
Generar permisos de usuarios	7	1	0%	
Compilación	7	2	0%	
Implementar datos de pruebas	8	2	0%	
<b>Pruebas</b>				
Usabilidad del usuario docente	9	2	0%	
Usabilidad del usuario Estudiante	9	2	0%	
interacción del usuario Docente	9	2	0%	
interacción del usuario Estudiante	9	2	0%	
Reunión para la validación	9	2	0%	
Pruebas en escenarios reales	9	2	0%	
Pruebas en dispositivos Android	9	2	0%	
Pruebas en dispositivos iOS	9	2	0%	
Correcciones	11	1	0%	
Documentación	11	1	0%	
<b>Cierre</b>				
Entrega final	12	1	0%	

**TABLA 8 CRONOGRAMA DE ACTIVIDADES**

*Fuente: elaboración propia*

## Descripción general del aplicativo móvil

A continuación se describe la funcionalidad del aplicativo, de acuerdo a la ilustración esquema del aplicativo.

- El cliente hará una petición a los servicios REST en Azure.
- La petición será procesada y consultada en la DB.
- Una vez consultada se volverá la respuesta en formato Json,
- La aplicación procesa la lógica que tenga para mostrar dicha información que se realizó
- Los envíos de petición hacia los servicios REST , deberán tener una cabecera para envío de token de validación y un cuerpo para envío de datos o objetos.



ILUSTRACIÓN 9 ESQUEMA GENERAL DEL APLICATIVO MÓVIL

*Fuente: elaboración propia*

Las instituciones de educación superior que no contraten o tengan el servicio en la nube si no local, su lógica de negocio seguirá funcionando de la misma forma que se ilustró anteriormente. El aplicativo móvil tiene como finalidad la toma de asistencia de los estudiantes y docentes adscritos a determinado horario de clases a través de un código generado por el docente activando la clase. El cual será compartido a los estudiantes los

cuales ingresan al aplicativo podrán ver la clase activa e ingresar hacer la toma de asistencia. Una vez ingresado el código validara su validez y permitirá al estudiante ser registrado en la lista del docente del aplicativo para control de estudiante que se van registrando, al finalizar el docente será en el encargado de finalizar la asistencia automáticamente se guardara en base de datos de la Institución.

### Funcionalidad general

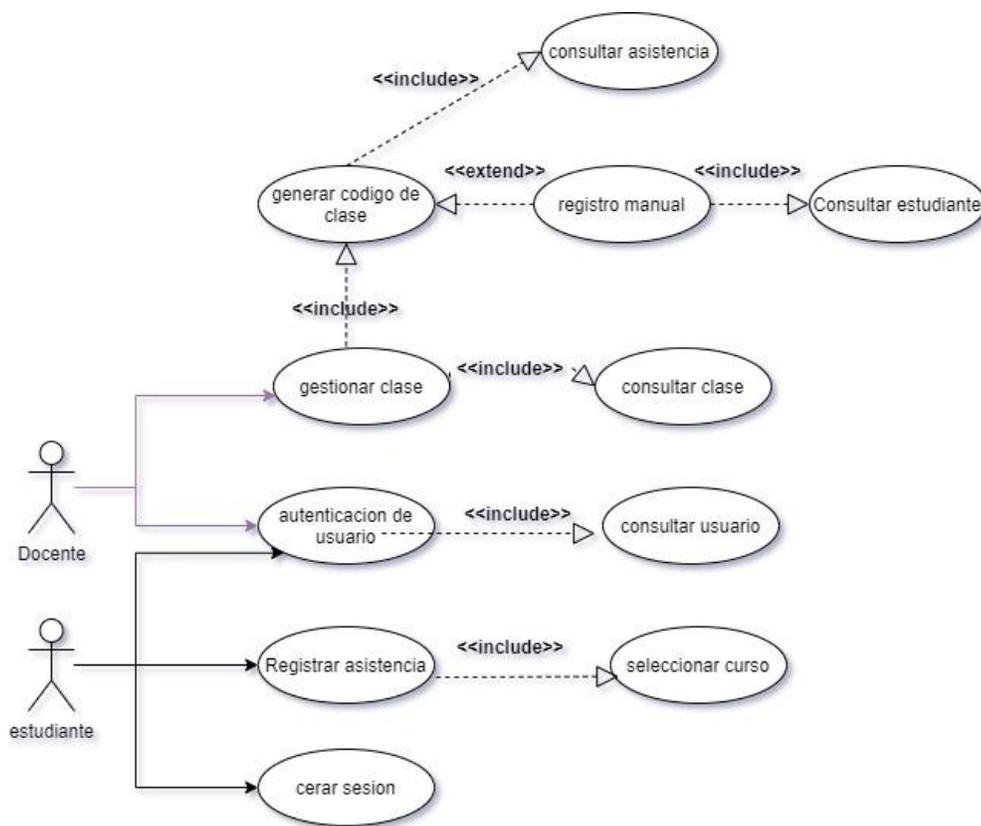


ILUSTRACIÓN 10 FUNCIONALIDAD GENERAL DEL APLICATIVO

*Fuente: elaboración propia*

## Características de los usuarios del sistema

El sistema consta con 2 perfiles con las siguientes funcionalidades cada uno:

<b>ID</b>	<b>AC01</b>
<b>Tipo de usuario</b>	Docente
<b>Formación</b>	Conocimiento intermedio en manejo de sistemas
<b>Actividades</b>	<ul style="list-style-type: none"> <li>- Autenticación en el sistema</li> <li>- Gestión de clases</li> <li>- Generación de código de acceso a clase</li> <li>- Consultar clase</li> <li>- Consultar asistencia</li> </ul>

**TABLA 9 ACTOR DEL SISTEMA: DOCENTE**

*Fuente: elaboración propia*

<b>ID</b>	<b>AC02</b>
<b>Tipo de usuario</b>	Estudiante
<b>Formación</b>	Conocimiento intermedio en manejo de sistemas
<b>Actividades</b>	<ul style="list-style-type: none"> <li>- Autenticación en el sistema</li> <li>- Registrar asistencia</li> </ul>

**TABLA 10 ACTOR DEL SISTEMA: ESTUDIANTE**

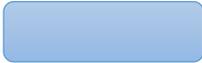
*Fuente: elaboración propia*

## Modelado del sistema

En este capítulo del documento se detallan los modelos utilizados para documentar el proceso del desarrollo de software como: representación de procesos, casos de uso, Requisitos Funcionales y no Funcionales,

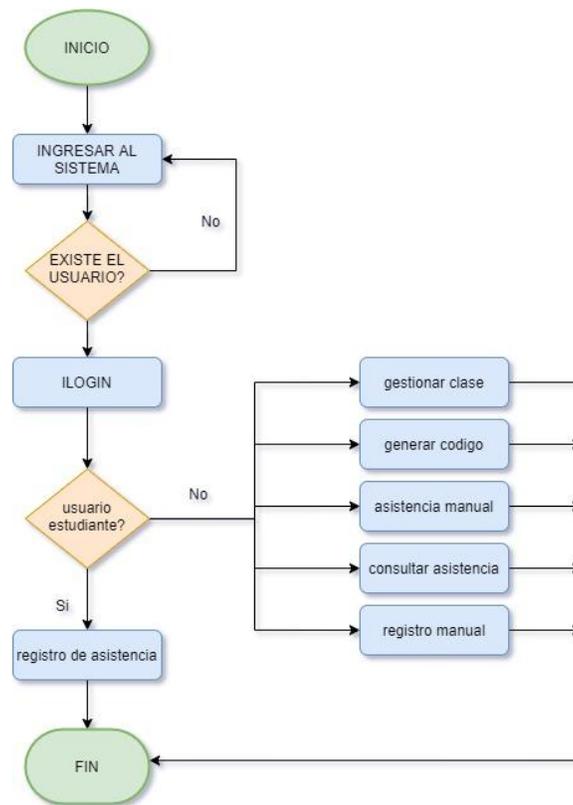
Representación gráfica de procesos

A continuación, se presenta un flujograma de los procesos con los requerimientos iniciales para el desarrollo del aplicativo móvil.

SIMBOLO	PROCESO
	Inicio/Final
	Proceso
	Decisión
	Dirección de flujo

**TABLA 11 SIMBOLOGÍA DIAGRAMA DE FLUJO**

*Fuente: elaboración propia*



**ILUSTRACIÓN 11 DIAGRAMA DE FLUJO**

*Fuente: elaboración propia*

En el diagrama se observa como al ingresar al sistema el usuario debe autenticarse en el aplicativo móvil con sus credenciales para realizar el respectivo LOGIN y dar acceso automático al perfil de usuario correspondiente, sea Estudiante o Docente, dado el caso accederá a su respectivo modulo. El acceso a las funcionalidades varias según el rol al que se acceda.

### Especificación de requisitos

Siendo los requisitos la descripción de las características y funcionalidades del sistema y su comportamiento, se definen entre ellos los requisitos funcionales y no funcionales como también la definición de casos de uso.

A continuación, se detallan cada uno de ellos dentro del sistema web

#### *Requerimientos funcionales de la aplicación*

Descripción de las funciones que debe tener el sistema, las cuales interactúan con los actores.

IDENTIFICACIÓN DEL REQUISITO	RF01
<b>Nombre del requisito</b>	Autenticación de usuario
<b>Descripción del requisito</b>	La aplicación móvil debe permitir al usuario el inicio de sesión una vez este se encuentre registrado en la plataforma ingresando su usuario y contraseña
<b>Prioridad del requisito</b>	Alta

**TABLA 12 REQUISITO FUNCIONAL AUTENTICACIÓN DE USUARIO**

*Fuente: elaboración propia*

Identificación del requisito	RF02
<b>Nombre del requisito</b>	Gestionar clase
<b>Descripción del requisito</b>	La aplicación móvil debe permitir al docente la habilitación de la clase previamente asignada dentro de los tiempos establecidos.
<b>Prioridad del requisito</b>	Alta

**TABLA 13 REQUISITO FUNCIONAL GESTIONAR CLASE**

*Fuente: elaboración propia*

IDENTIFICACIÓN DEL REQUISITO	RF03
Nombre del requisito	Generar código de clase
Descripción del requisito	La aplicación móvil permitirá al docente generar un código para la toma de asistencia de los estudiantes a su clase.
Prioridad del requisito	Alta

**TABLA 14 REQUISITO FUNCIONAL GENERAR CÓDIGO CLASE**

*Fuente: elaboración propia*

IDENTIFICACIÓN DEL REQUISITO	RF04
Nombre del requisito	Consultar asistencia
Descripción del requisito	La aplicación móvil debe permitir al docente consultar en tiempo real el listado de los alumnos que se han registrado asistencia a su clase.
Prioridad del requisito	Alta

**TABLA 15 REQUISITO FUNCIONAL CONSULTAR ASISTENCIA**

*Fuente: elaboración propia*

IDENTIFICACIÓN DEL REQUISITO	RF05
Nombre del requisito	Gestionar estudiante
Descripción del requisito	La aplicación móvil debe permitir al docente la opción de marcar la asistencia de un estudiante en caso fortuito que este no pueda hacerlo por falta de un dispositivo, no poder hacer login o falta de acceso a internet.
Prioridad del requisito	Alta

**TABLA 16 REQUISITO FUNCIONAL GESTIONAR ESTUDIANTE**

*Fuente: elaboración propia*

IDENTIFICACIÓN DEL REQUISITO	RF06
Nombre del requisito	Registrar asistencia
Descripción del requisito	La aplicación móvil debe permitir al estudiante marcar su asistencia a la clase al cual el docente le dio acceso.
Prioridad del requisito	Alta

**TABLA 17 REQUISITO FUNCIONAL REGISTRAR ASISTENCIA**

*Fuente: elaboración propia*

### *Requerimientos no funcionales del sistema*

A continuación, se definen los requisitos no funcionales del sistema, siendo aquellos comportamientos o restricciones en el diseño y la implementación del mismo.

IDENTIFICACIÓN DEL REQUISITO	RNF01
Nombre del requisito	Seguridad
Descripción del requisito	La aplicación móvil deberá brindar seguridad tanto al acceso a usuarios, la generación del código de clase. Se necesitará mecanismos de encriptación con el fin de cifrar los usuarios y contraseñas.
Prioridad del requisito	Alta

**TABLA 18 REQUISITO NO FUNCIONAL SEGURIDAD**

*Fuente: elaboración propia*

IDENTIFICACIÓN DEL REQUISITO	RNF02
Nombre del requisito	Eficiencia
Descripción del requisito	La aplicación móvil deberá funcionar de forma eficiente brindando al usuario un tiempo de respuesta máximo de 3 segundos
Prioridad del requisito	Alta

**TABLA 19 REQUISITO NO FUNCIONAL EFICIENCIA**

*Fuente: elaboración propia*

IDENTIFICACIÓN DEL REQUISITO	RNF03
Nombre del requisito	Portabilidad
Descripción del requisito	La aplicación móvil debe ser responsive, con el objetivo que se adapte a la mayor cantidad de dispositivos móviles.
Prioridad del requisito	Alta

**TABLA 20 REQUISITO NO FUNCIONAL PORTABILIDAD**

*Fuente: elaboración propia*

IDENTIFICACIÓN DEL REQUISITO	RNF04
Nombre del requisito	Formatos de colores de la entidad
Descripción del requisito	El sistema deberá utilizar los colores preestablecidos por la entidad.
Prioridad del requisito	Media

**TABLA 21 REQUISITO NO FUNCIONAL FORMATO**

*Fuente: elaboración propia*

## Diagramas de caso de uso

### *Caso De Uso Estudiante*

Su objetivo es registrar la asistencia en la aplicación móvil dentro de la clase en la cual se encuentra

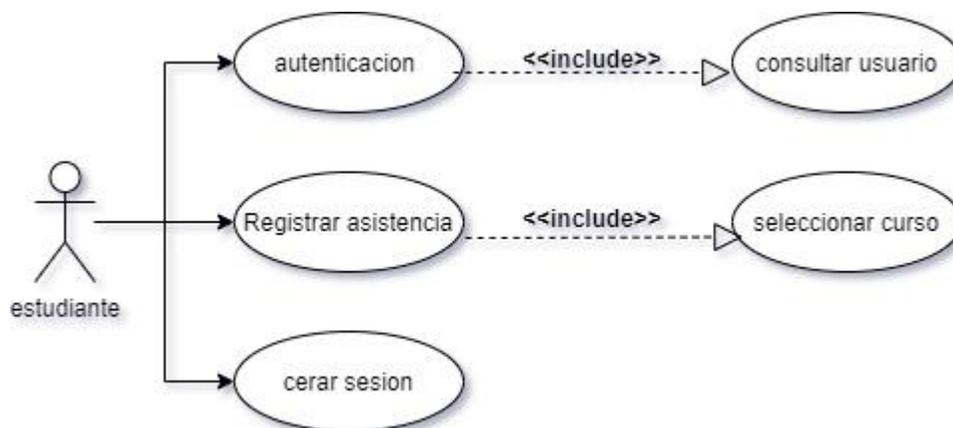


ILUSTRACIÓN 11 CASO DE USO ESTUDIANTE

*Fuente: elaboración propia*

<b>CU-01</b>	<b>AUTENTICACIÓN</b>	
<b>[Versión]</b>	1.0 (01/05/2021)	
<b>Precondición</b>	El usuario debe estar previamente registrado para poder acceder a la aplicación móvil	
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario desee acceder a la aplicación web.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario accede al LOGIN para poder acceder a la aplicación móvil.
	2	El sistema solicita al usuario el nombre de usuario y su respectiva contraseña.
	3	Si el usuario no tiene un dispositivo móvil o en su defecto red de internet
	3.1	Se realiza el caso de uso Registro manual desde el perfil del docente
4	El sistema deja ingresar con éxito al usuario.	
<b>Postcondición</b>	El usuario a iniciado sesión en la aplicación y puede acceder a las acciones que tenga permiso.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el sistema arroja error en la autenticación
	E.1	La aplicación genera un aviso de error
	E.2	El usuario deberá acceder a la opción recordar Usuario y Contraseña

	E.3	El sistema deberá enviar al correo institucional registrado su usuario y contraseña
	E.4	Se cancela el caso de uso

**TABLA 22 CASO DE USO AUTENTICACIÓN**

*Fuente: elaboración propia*

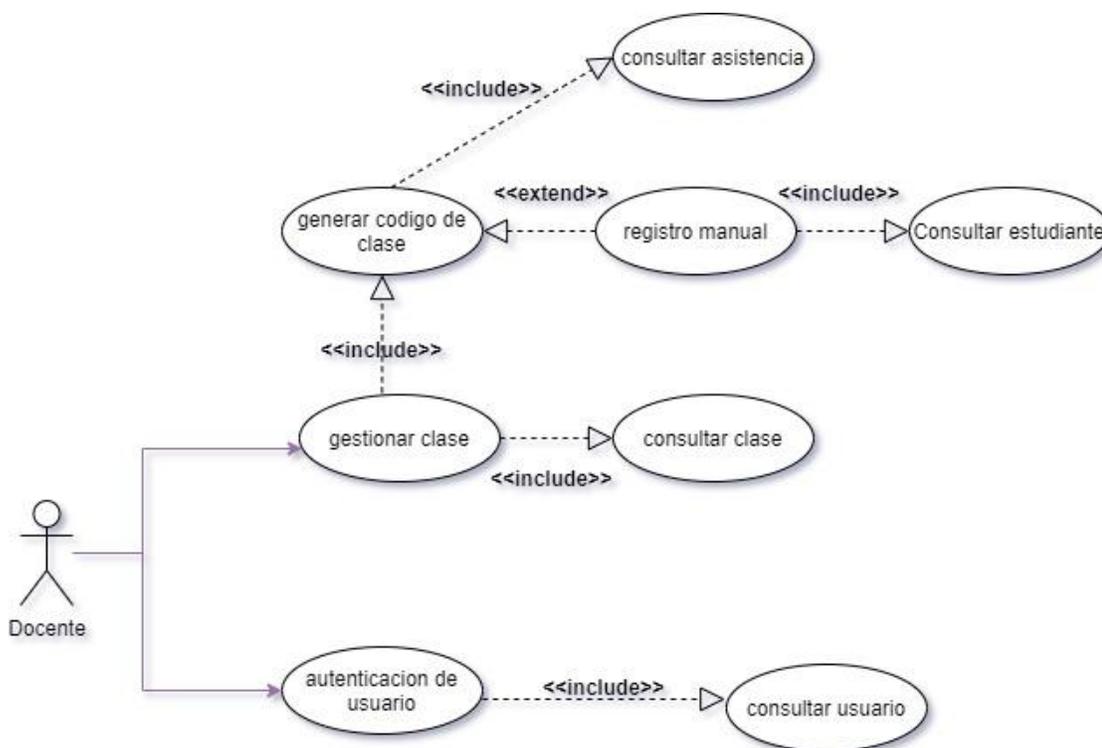
<b>CU-02</b>	<b>REGISTRAR ASISTENCIA</b>	
<b>[Versión]</b>	1.0 (01/05/2021)	
<b>Precondición</b>	El usuario debe estar previamente logueado en la aplicación móvil y debe conocer el código de acceso a la clase.	
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario desee registrar su asistencia a una clase.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario ingresa el código de la clase generado por el docente.
	2	El estudiante selecciona la clase a la cual se va a registrar.
	4	El sistema deja registrar con éxito la asistencia a la clase.
<b>Postcondición</b>	El estudiante ya puede cerrar su sesión activa.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el sistema arroja error en el acceso al código de clase
	E.1	La aplicación genera un aviso de error
	E.2	El usuario deberá volver a ingresar el código de acceso a la clase
	E.3	El sistema deberá enviar al correo registrado su usuario y contraseña
E.4	Se cancela el caso de uso	

**TABLA 23 CASO DE USO REGISTRAR ASISTENCIA**

*Fuente: elaboración propia*

### *Caso De Uso Docente*

Su objetivo dentro de la aplicación es Gestionar la clase a impartir, dentro de esta tiene la facultad de generar el código de clase con el cual se le dará acceso a los estudiantes, realizar un registro manual de un estudiante que por algún motivo no pueda registrar asistencia como: no disponer de un equipo móvil, acceso a internet o fallo en el inicio de sesión.



**ILUSTRACIÓN 12 CASO DE USO DOCENTE**

*Fuente: elaboración propia*

<b>CU-03</b>	<b>GESTIONAR CLASE</b>	
<b>[Versión]</b>	1.0 (01/05/2021)	
<b>Precondición</b>	El usuario debe estar previamente logueado en la aplicación móvil	
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el docente desee dar inicio a una clase.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El docente consulta las clases que tiene asignada en ese horario
	2	El docente selecciona la clase a iniciar.
	3	El docente selección la opción de generar código
	4	El sistema dejar con éxito la clase.
<b>Postcondición</b>	El docente ya puede entregar el código de acceso a los estudiantes.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el sistema no muestra la clase
	E.1	El docente deberá comunicarse con el área encargada
	E.4	Se cancela el caso de uso

**TABLA 24 CASO DE USO GESTIONAR CLASE**

*Fuente: elaboración propia*

<b>CU-04</b>	<b>REGISTRO MANUAL</b>	
<b>[Versión]</b>	1.0 (01/05/2021)	
<b>Precondición</b>	El docente debe haber activado la clase.	
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el docente desee registrar manualmente la asistencia de un estudiante en particular.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El docente selecciona la opción "registro manual de asistencia"
	2	El docente debe digitar en el campo requerido el código de estudiante
	3	Una vez identificado el alumno se selecciona y se selecciona la opción registrar asistencia a clase
	4	El sistema deja registrar con éxito al estudiante de forma manual.
<b>Postcondición</b>	El docente ya puede consultar los asistentes a la clase.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el sistema no muestra la clase
	E.1	El docente deberá comunicarse con el área encargada
	E.4	Se cancela el caso de uso

**TABLA 25 CASO DE USO REGISTRO MANUAL**

*Fuente: elaboración propia*

<b>CU-05</b>	<b>Generar código de acceso</b>	
<b>[Versión]</b>	1.0 (01/05/2021)	
<b>Precondición</b>	El docente debe haber iniciado sesión	
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el docente desee generar el código de acceso a la clase.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>

	1	<i>El docente selecciona la opción “generar código de acceso”</i>
	2	<i>El aplicativo muestra en pantalla el código de acceso generado</i>
	3	<i>El docente debe dar clic en el botón compartir Código</i>
	4	<i>El sistema deja generar y compartir con éxito el código de acceso a la clase.</i>
<b>Postcondición</b>	<i>El docente ya puede entregar el código de acceso a los estudiantes.</i>	

**TABLA 26 CASO DE USO GENERAR CÓDIGO DE ACCESO**

*Fuente: elaboración propia*

<b>CU-05</b>	<b>Consultar asistencia</b>	
<b>[Versión]</b>	1.0 (01/05/2021)	
<b>Precondición</b>	<i>El docente debe haber dado inicio a la clase en el aplicativo</i>	
<b>Descripción</b>	<i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el docente desee consultar la asistencia de los alumnos a determinada clase.</i>	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	<i>El docente selecciona la opción “mostrar asistentes”</i>
	2	<i>El sistema muestra en pantalla con éxito la lista de estudiantes marcados como asistentes a la clase.</i>
<b>Postcondición</b>	<i>El docente ya puede entregar el código de acceso a los estudiantes.</i>	

**TABLA 27 CASO DE USO CONSULTAR ASISTENCIA**

*Fuente: elaboración propia*

## Diagramas de secuencia

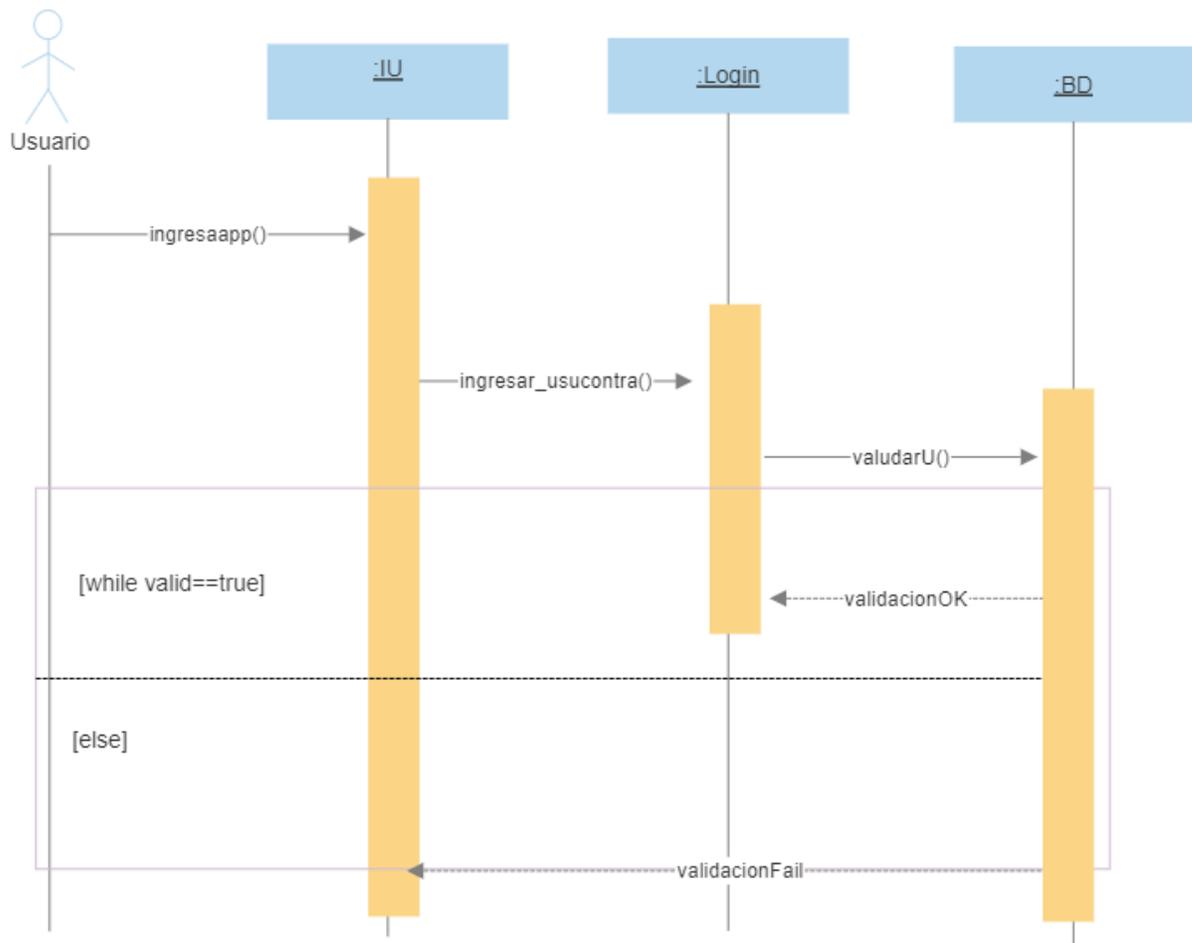
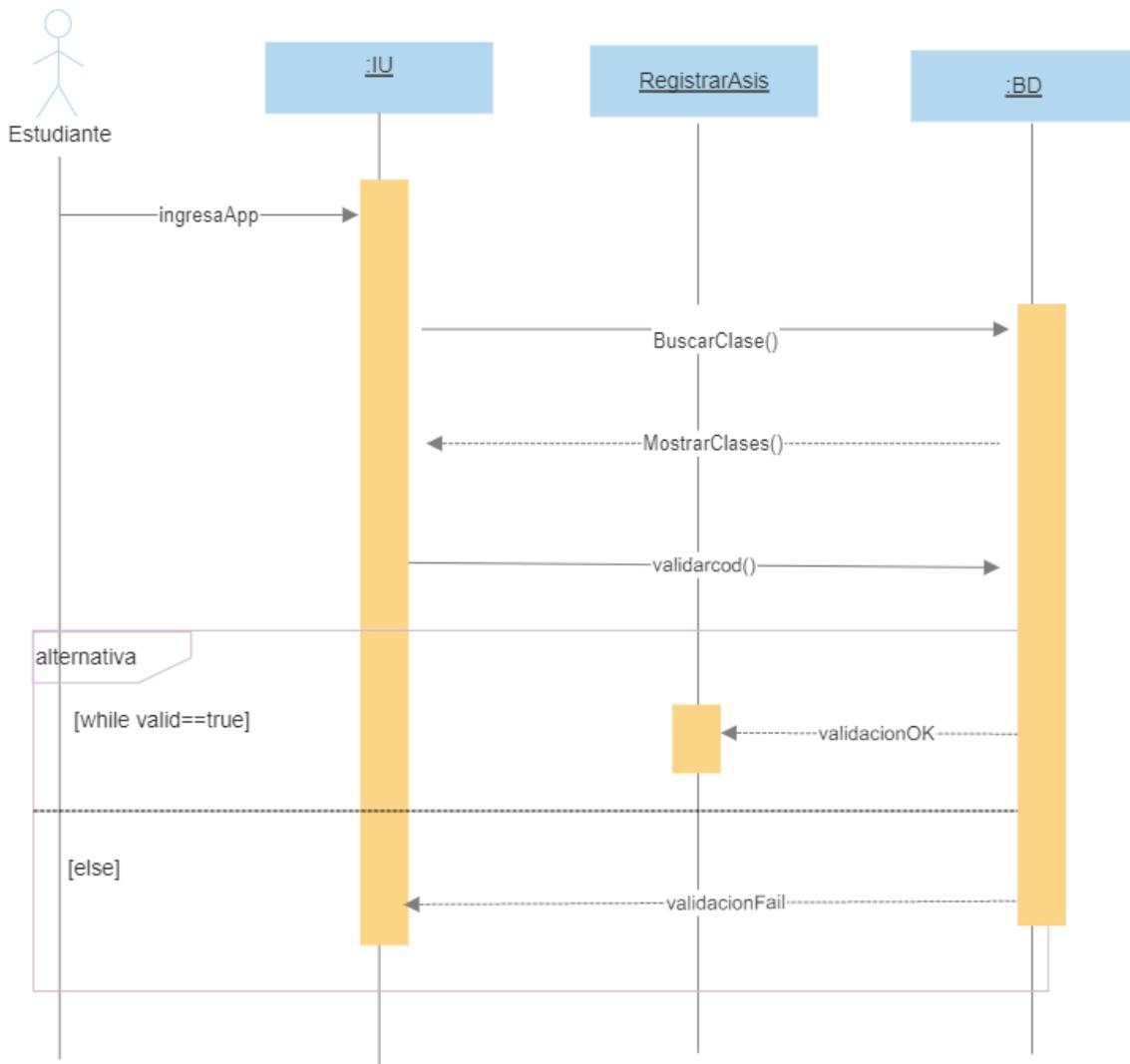
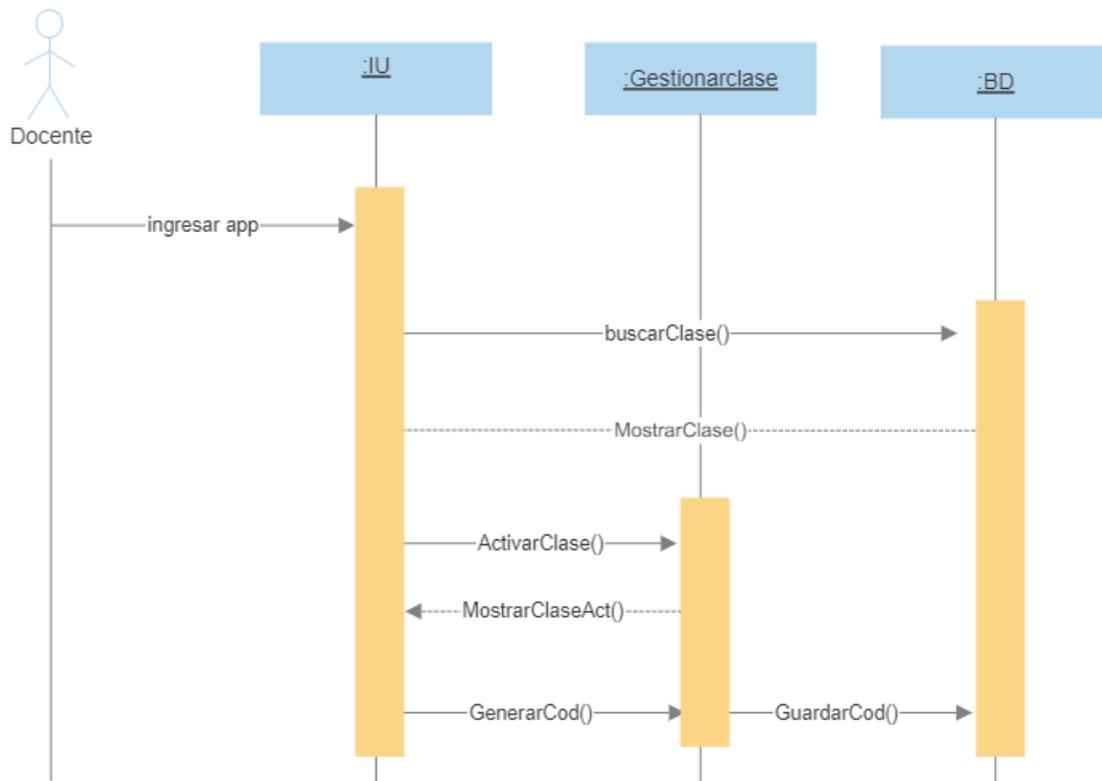
*Diagrama de secuencia autenticación*

ILUSTRACIÓN 13 DIAGRAMA DE SECUENCIA AUTENTICACIÓN

*Fuente: elaboración propia*

*Diagrama de secuencia registrar asistencia***ILUSTRACIÓN 14 DIAGRAMA DE SECUENCIA REGISTRAR ASISTENCIA***Fuente: elaboración propia*

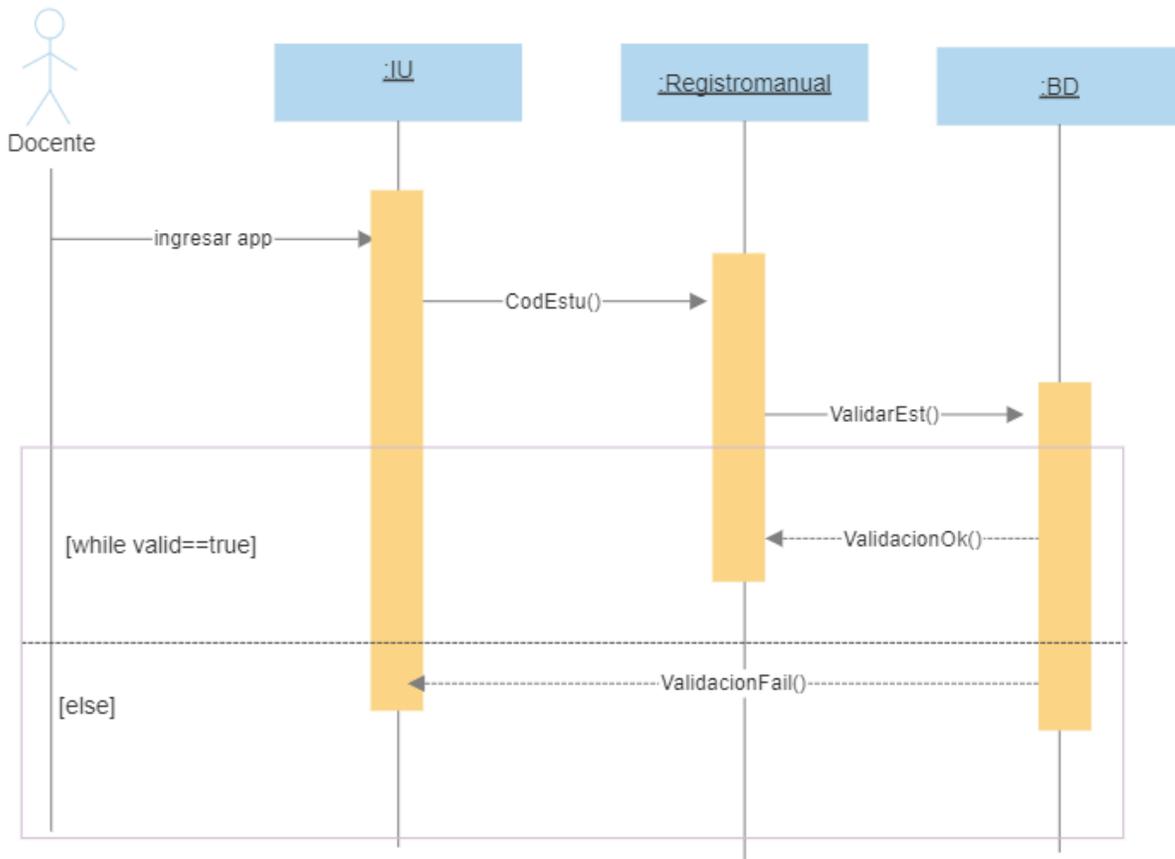
*Diagrama de secuencia gestionar clase*



**ILUSTRACIÓN 15 DIAGRAMA DE SECUENCIA GESTIONAR CLASE**

*Fuente: elaboración propia*

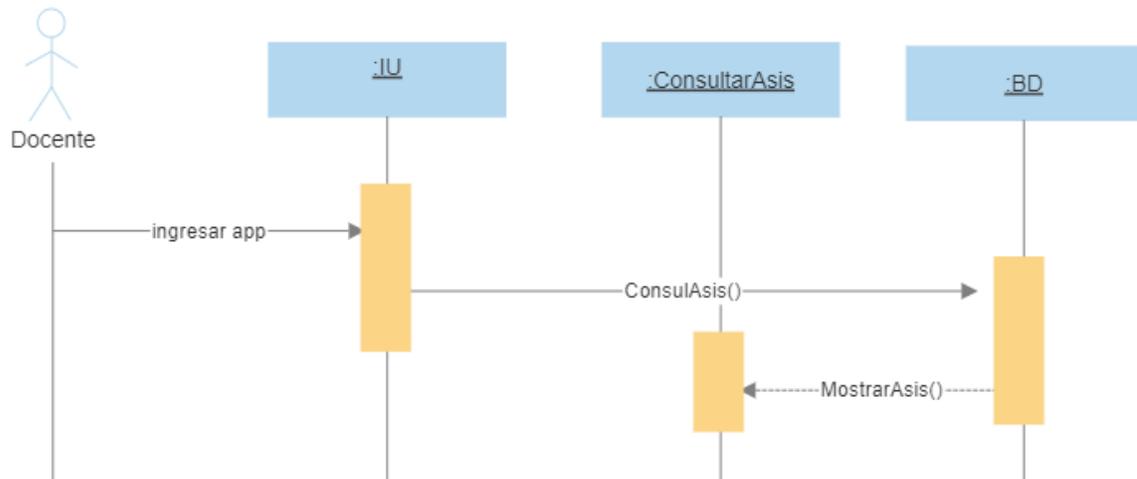
*Diagrama de secuencia registro manual de estudiante*



**ILUSTRACIÓN 16** DIAGRAMA DE SECUENCIA REGISTRO MANUAL DE ESTUDIANTE

*Fuente: elaboración propia*

*Diagrama de secuencia consultar asistencia*



**ILUSTRACIÓN 17** DIAGRAMA DE SECUENCIA CONSULTAR ASISTENCIA

*Fuente: elaboración propia*

## Diagrama de clases

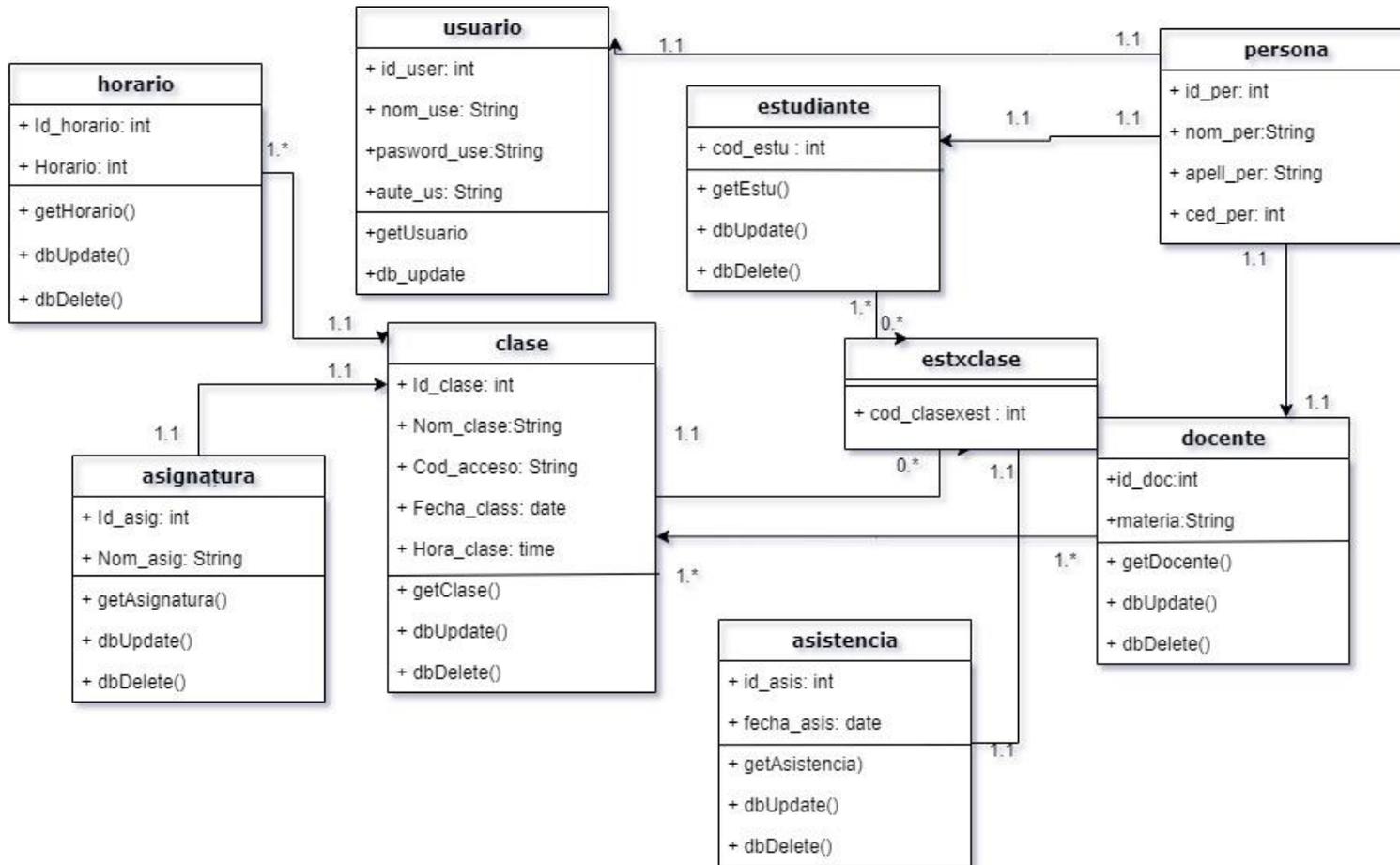


ILUSTRACIÓN 18 DIAGRAMA DE CLASES

*Fuente: elaboración propia*

## Diagrama Entidad-relación

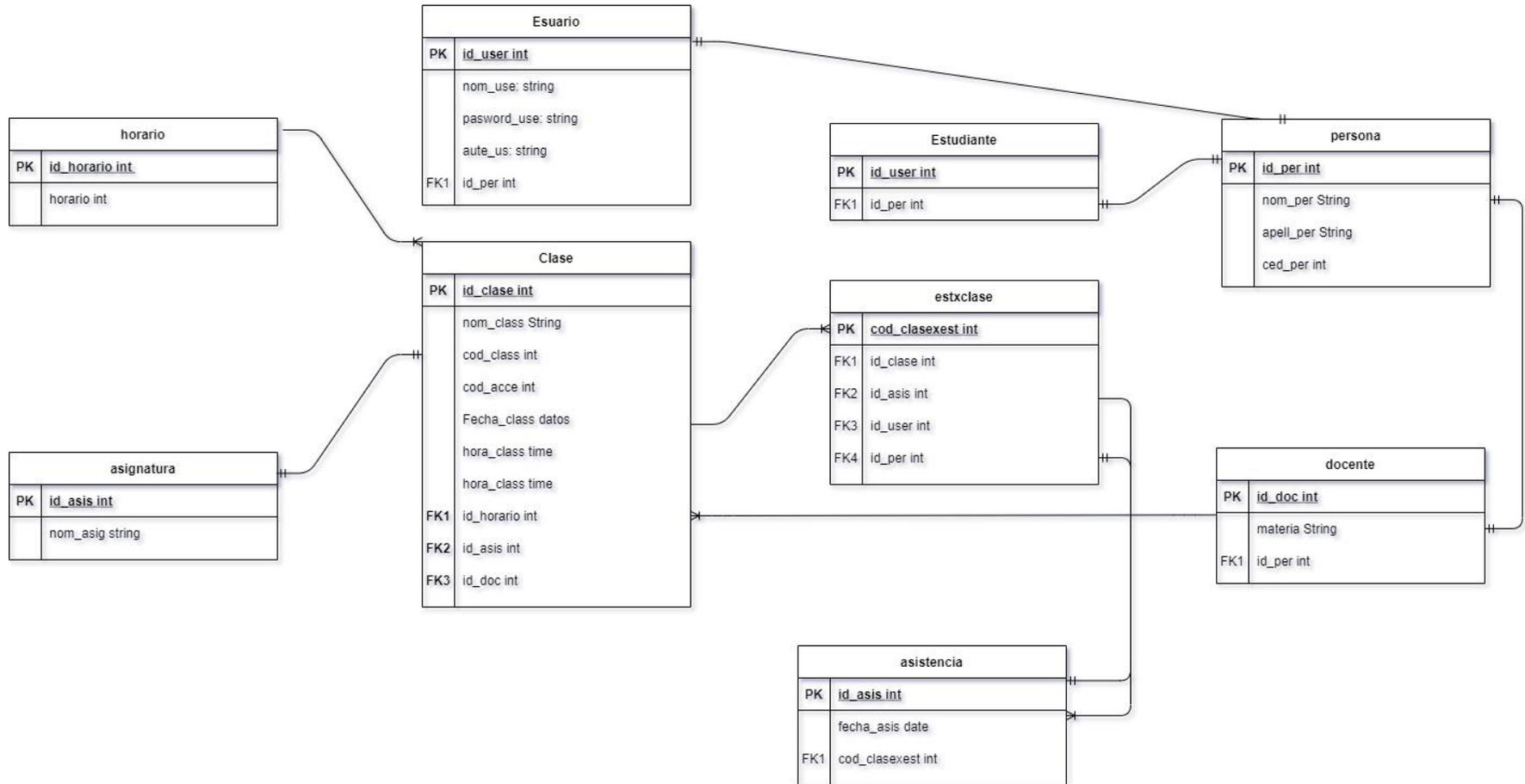


ILUSTRACIÓN 19 DIAGRAMA ENTIDAD RELACIÓN

*Fuente: elaboración propia*

## Diccionario de datos

A continuación, se especificarán cada una de las entidades del modelo entidad-relación.

### *Entidad usuario*

A continuación, se detallarán cada uno de los atributos de la entidad usuario. Esta entidad corresponde a los diferentes usuarios existentes dentro del aplicativo, en este caso se definen los usuarios: docente y estudiante.

ENTIDAD: USUARIO			
Atributo	tipo	longitud	Descripción
<b>Id_user</b>	int	10	Identificador del usuario
<b>Nom_user</b>	String	100	Nombre del usuario
<b>Pasword_use</b>	String	100	Contraseña del usuario
<b>Aute_us</b>	String	100	Campo de autenticación

**TABLA 28** ESPECIFICACIONES DE ENTIDAD USUARIO

*Fuente: elaboración propia*

### *Entidad Clase*

A continuación, se detallarán cada uno de los atributos de la entidad Clase. Esta entidad corresponde a la información registrada para determinada clase gestionada por el docente, la cual tiene los siguientes atributos:

ENTIDAD: CLASE			
Atributo	tipo	longitud	Descripción
<b>Id_class</b>	int	10	Identificador de la clase
<b>Cod_class</b>	Int	10	Codigo de la clase
<b>Cod_acces</b>	Int	10	Codigo de acceso a la clase
<b>Fecha_class</b>	Date		Fecha de la clase impartida
<b>Hora_class</b>	time		Hora de la clase

**TABLA 29** ESPECIFICACIONES DE ENTIDAD CLASE

*Fuente: elaboración propia*

### *Entidad horaria*

A continuación, se detallarán cada uno de los atributos de la entidad horario

ENTIDAD: HORARIO			
Atributo	tipo	longitud	Descripción
Id_horario	int	10	Identificador de horario
Horario	Int	10	Código de la clase

**TABLA 30** ESPECIFICACIONES DE ENTIDAD HORARIO

*Fuente: elaboración propia*

### *Entidad asignatura*

A continuación, se detallarán cada uno de los atributos de la entidad asignatura. Esta entidad corresponde a la información registrada de la asignatura a impartir por el docente y de asistir el estudiante, la cual tiene los siguientes atributos:

ENTIDAD: ASIGNATURA			
Atributo	tipo	longitud	Descripción
Id_asig	int	10	Identificador de la asignatura
Nom_asig	String	100	Nombre de la asignatura

**TABLA 31** ESPECIFICACIONES DE ENTIDAD ASIGNATURA

*Fuente: elaboración propia*

### *Entidad persona*

A continuación, se detallarán cada uno de los atributos de la entidad persona. Esta entidad corresponde a la información registrada de la persona como datos básicos, la cual tiene los siguientes atributos:

ENTIDAD: PERSONA			
Atributo	tipo	longitud	Descripción
Id_persona	int	10	Código del estudiante
Nom_per	String	100	Nombre de la persona
Apellido_per	String	100	Apellido de la persona
Ced_per	Int	10	Cedula de la persona

**TABLA 32** ESPECIFICACIONES DE ENTIDAD PERSONA

*Fuente: elaboración propia*

### *Entidad estudiante*

A continuación, se detallarán cada uno de los atributos de la entidad estudiante. Esta entidad corresponde a la información registrada del estudiante, la cual tiene los siguientes atributos:

Entidad: estudiante			
Atributo	tipo	longitud	Descripción
<b>Cod_estu</b>	int	10	Codigo del estudiante
<b>Correo_estu</b>	String	100	Correo institucional del estudiante

**TABLA 33** ESPECIFICACIONES DE ENTIDAD ESTUDIANTE

*Fuente: elaboración propia*

### *Entidad docente*

A continuación, se detallarán cada uno de los atributos de la entidad docente. Esta entidad corresponde a la información registrada del docente en relación con las materias que dicta, la cual tiene los siguientes atributos:

Entidad: docente			
Atributo	tipo	longitud	Descripción
<b>Id_doc</b>	int	10	Código del docente
<b>Materia:</b>	String	100	Materia que dicta

**TABLA 34** ESPECIFICACIONES DE ENTIDAD DOCENTE

*Fuente: elaboración propia*

### *Entidad estxclase*

A continuación, se detallarán cada uno de los atributos de la entidad estaxclase. Esta entidad corresponde a la información registrada del estudiante por clase en relación con las tablas de estudiante y clase siendo esta una relación de muchos a muchos, la cual tiene los siguientes atributos:

Entidad: docente			
Atributo	tipo	longitud	Descripción
Cod_clasexest	int	10	Código de la clase por estudiante

**TABLA 35 ESPECIFICACIONES DE ENTIDAD ESTUXCLASE**

*Fuente: elaboración propia*

### *Entidad asistencia*

A continuación, se detallarán cada uno de los atributos de la entidad asistencia. Esta entidad corresponde a la información registrada de la asistencia registrada por los estudiantes a determinada clase, la cual tiene los siguientes atributos:

Entidad: asistencia			
Atributo	tipo	longitud	Descripción
Id_asis	int	10	Id de la asistencia
Fecha_asis	date		Fecha de la asistencia

**TABLA 36 ESPECIFICACIONES DE ENTIDAD ASISTENCIA**

*Fuente: elaboración propia*

## Interfaz del aplicativo "prototipo"

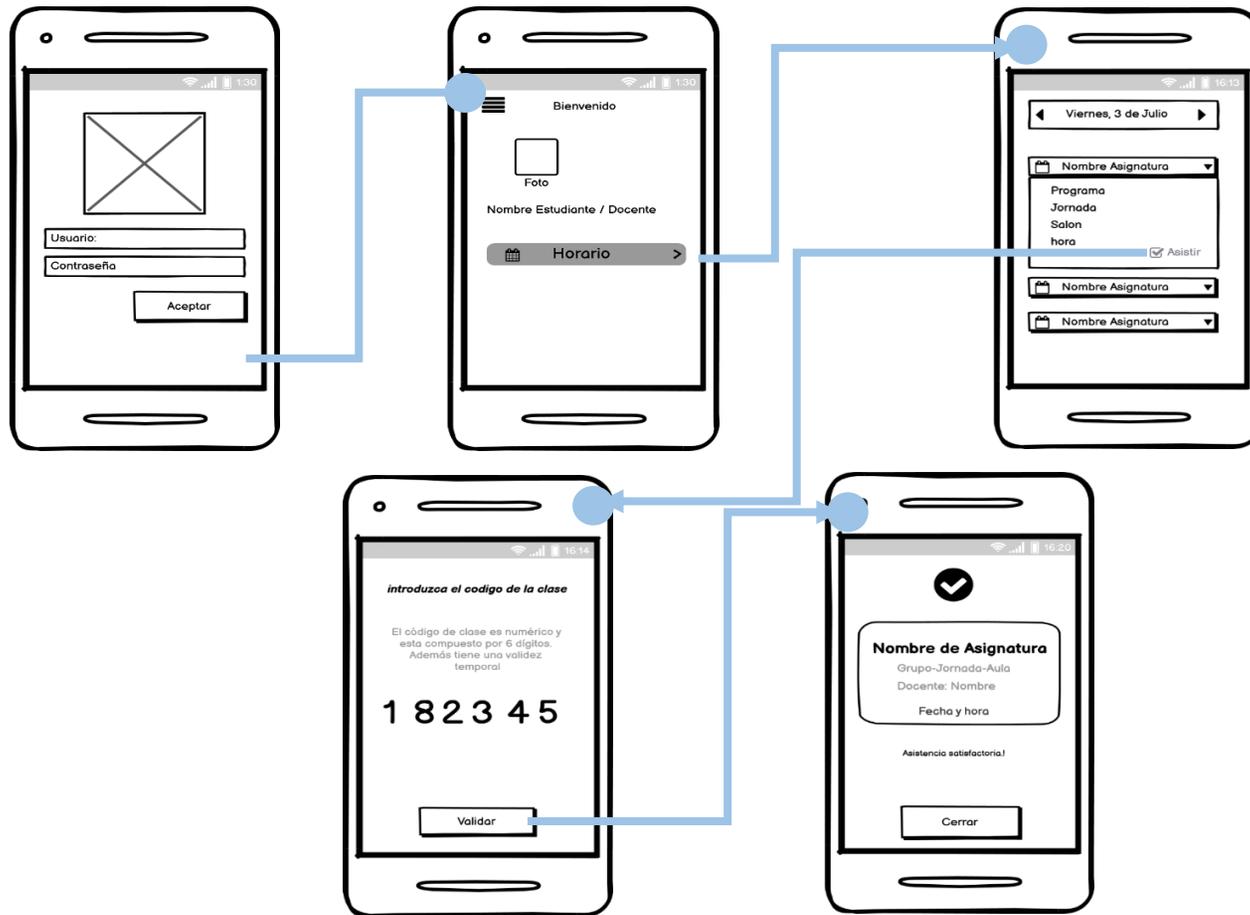


ILUSTRACIÓN 20 INTERFAZ DEL APLICATIVO "PROTOTIPO"

Fuente: elaboración propia

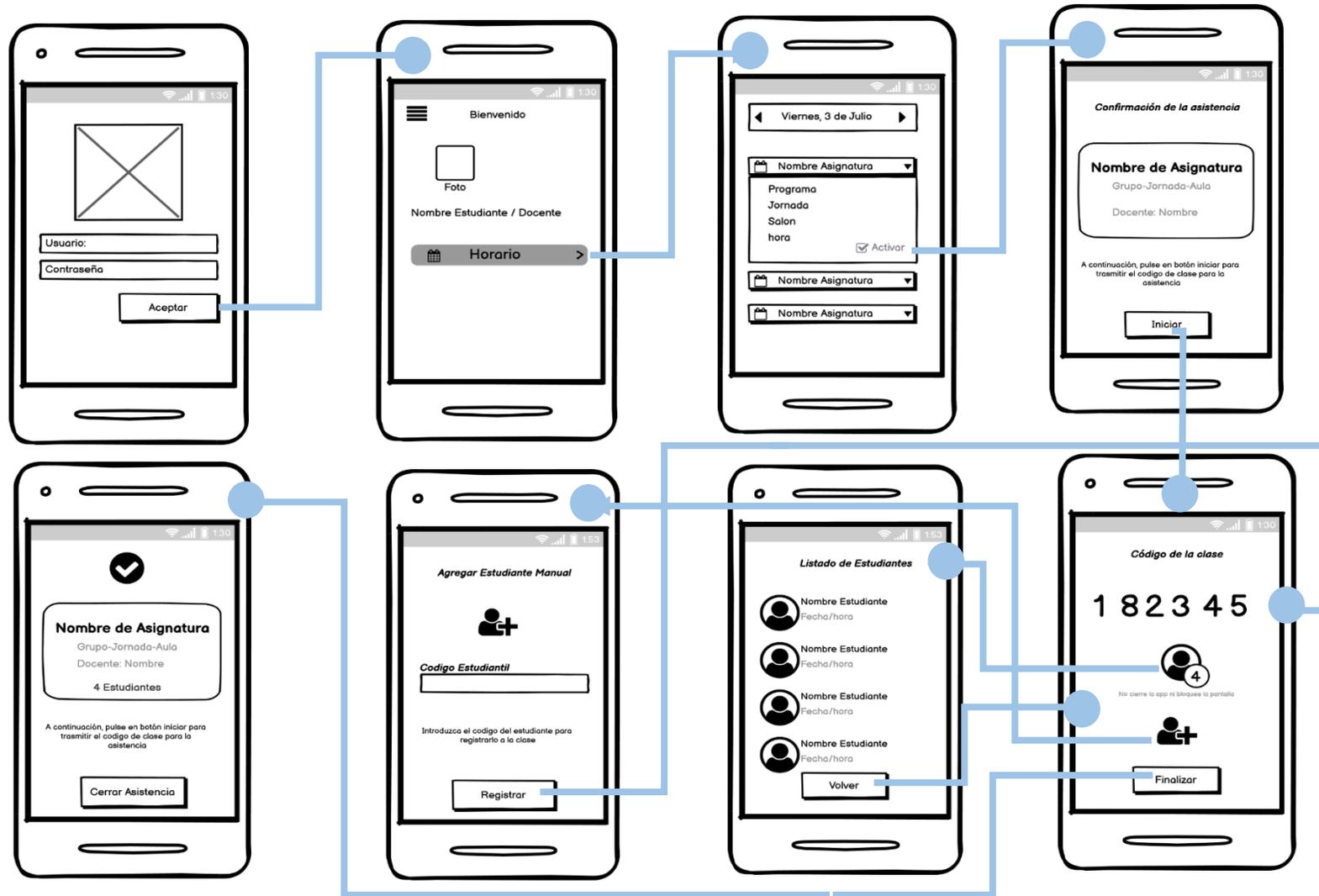


ILUSTRACIÓN 21 INTERFAZ DEL APLICATIVO "PROTOTIPO"

*Fuente: elaboración propia*

## CAPÍTULO 5. GESTIÓN DE LOS RECURSOS DEL PROYECTO

### Organigrama del proyecto

A continuación se muestra la estructura de roles en que se trabaja el proyecto

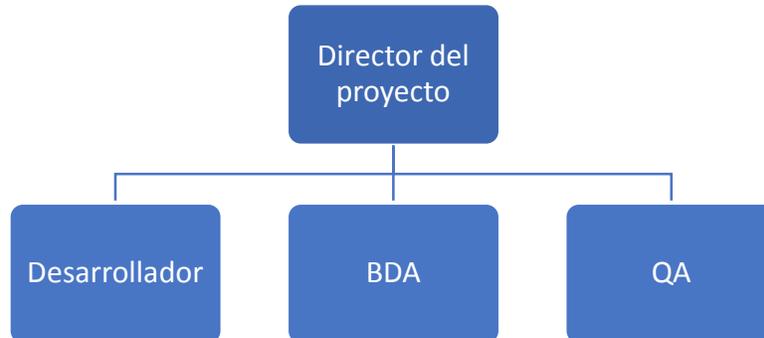


ILUSTRACIÓN 22 ORGANIGRAMA DEL PROYECTO  
*Fuente: elaboración propia*

### Definición de roles

**Director del proyecto:** El director del proyecto se encargará del definir, gestionar y coordinar la estructuración del proyecto, se encuentra presente en todas las fases monitoreando el cumplimiento de los objetivos propuestos para la implementación. Debe tener conocimientos previos en: gestión de proyectos tecnológicos, arquitectura de software, programación y Gestión de base de datos.

**Desarrollador:** El desarrollador de software será quien reciba la documentación creada por el arquitecto y el director del proyecto, y quién implemente el producto según ésta, este perfil conoce y es capaz de realizar todas las tareas de desarrollo. Debe tener conocimientos previos en:

BDA: El Administrador de base de datos es el encargado de desarrollar la base de datos del proyecto, interviene en la fase de análisis donde estudiara la documentación necesaria para comprender el objetivo y el diseño del aplicativo móvil. Debe tener conocimientos previos en:

QA: Se encargará de asegurar que los requisitos definidos del proyecto se cumplen en la implementación del producto o servicio realizado en las fases de diseño, implementación y desarrollo.

## Estimación de recurso, costos y presupuesto

Costo del proyecto

*Recurso Humano:*

ID	CARGO	REQUISITO MÍNIMO	CANTIDAD
1	Director proyecto	Experiencia mínima de 2 años dirigiendo proyectos de TI con certificación en Scrum	1
2	Desarrollador	Experiencia mínima de 2 años como back-end en java	1
3	Desarrollador	Experiencia mínima de 2 años como front-end en desarrollo de aplicaciones móviles multiplataforma	1
4	Dase Datos	Experiencia en creación de DB relacional	1
5	QA	Experiencia mínima 1 año aplicando pruebas	1

**TABLA 37 CARGOS**

*Fuente: elaboración propia*

ID	SALARIO MENSUAL	DESCUENTO SEGURIDAD SOCIAL	SALARIO PAGO NETO	PRESTACIONES SOCIALES	SALARIO MENSUAL CON PRESTACIONES	TIEMPO (MESES)	SALARIO TOTAL	
1	\$ 4.500.000	\$ 451.793	\$ 4.048.207	\$ 982.332	\$ 5.482.332	3	\$ 16.446.996	
2	\$ 3.200.000	\$ 325.293	\$ 2.874.707	\$ 698.547	\$ 3.898.547	3	\$ 11.695.642	
3	\$ 2.800.000	\$ 293.293	\$ 2.506.707	\$ 611.229	\$ 3.411.229	3	\$ 10.233.686	
4	\$ 3.600.000	\$ 357.293	\$ 3.242.707	\$ 785.866	\$ 4.385.866	3	\$ 13.157.597	
5	\$ 2.200.000	\$ 245.293	\$ 1.954.707	\$ 480.251	\$ 2.680.251	1	\$ 2.680.251	
	<b>TOTAL</b>							<b>\$ 54.214.172</b>

**TABLA 38 RECURSO HUMANO**

*Fuente: elaboración propia*

Recurso tecnológico hardware y Software:

PRODUCTO	CANTIDAD	ESPECIFICACIONES	VALOR	LICENCIA/MES	TOTAL
PC HP	2	i5 8th 16 RAM ,500 GB HDD	\$ 3.555.195		\$ 7.110.390
PC HP	2	i7 8th, 16 RAM 500 GB HDD, 500GB SSD	\$ 5.332.792		\$ 10.665.584
Azure App Services	1	Standard Tier; 1 S3 (4 Core(s), 7 GB RAM, 50 GB Storage) x 730 Hours; Linux OS	\$1.146.548	12	\$13.758.576
Samsung Galaxy A21S	1	Memoria interna: 128 GB, externa: 512GB RAM: 4 GB	\$ 757.900		\$ 757.900
iPhone XS 64GB 4G	1	Memoria interna: 64 GB,Red: 4G WiFi pantalla de 5.8"	\$ 2.900.000		\$ 2.900.000
Microstf 365	5	Incluye paquete Microstf Planner y Teams	\$ 213.750	4	\$ 4.275.000
Apple Store	1	Para publicar la aplicación en Apple Store	\$ 371.250		\$ 371.250
Play Store	1	Para publicar la aplicación en Play Store	\$ 93.750		\$ 93.750
			<b>TOTAL</b>		\$ 39.174.550

**TABLA 39 RECURSO TECNOLÓGICO**

*Fuente: elaboración propia*

Microsoft Azure Estimate					
Your Estimate					
Service type	Custom name	Region	Description	Estimated monthly cost	Estimated upfront cost
App Service		West US	Standard Tier; 1 S3 (4 Core(s), 7 GB RAM, 50 GB Storage) x 730 Hours; Linux OS	\$277,40	\$0,00
Support			Support	\$29,00	\$0,00
			Licensing Program	Microsoft Online Services Agreement	
			<b>Total</b>	<b>\$306,40</b>	<b>\$0,00</b>

**TABLA 40 COTIZACIÓN AZURE APP SERVICES**

*Fuente: <https://azure.microsoft.com/pricing/calculator/>*

## Presupuesto

RECURSO	VALORES	
HUMANO		\$ 54.214.172
TECNOLOGICO		\$ 39.174.550
SUBTOTAL		\$ 93.388.722
ADMINISTRATIVOS E IMPREVISTOS	30%	\$ 28.016.617
<b>TOTAL</b>		<b>\$ 121.405.339</b>

**TABLA 41 PRESUPUESTO DEL PROYECTO OPCIÓN 1**

*Fuente: elaboración propia*

A continuación, se presenta otra alternativa del presupuesto del proyecto sin la adquisición de Azure App Services.

RECURSO	VALORES	
HUMANO		\$ 54.214.172
TECNOLOGICO		\$ 25.415.974
SUBTOTAL		\$ 79.630.146
ADMINISTRATIVOS E IMPREVISTOS	30%	\$ 23.889.044
<b>TOTAL</b>		<b>\$ 103.519.190</b>

**TABLA 42 PRESUPUESTO DEL PROYECTO OPCIÓN 2**

*Fuente: elaboración propia*

## CAPÍTULO 6. EVALUACIÓN DE LAS ALTERNATIVAS

### Aseguramiento de control y calidad

La calidad dentro del software implica la utilización de procedimientos estándares que para el análisis, diseño, programación y prueba del software con el fin de lograr aspectos de aseguramiento de calidad como lo son:

- |                   |                 |               |
|-------------------|-----------------|---------------|
| - Seguridad       | - Usabilidad    | - Modularidad |
| - Comprensión     | - Fiabilidad    | - Eficacia    |
| - Portabilidad    | - Adaptabilidad | - Robustez    |
| - Protección      | - Reutilización | - Complejidad |
| - Experimentación | - Flexibilidad  | - Aprendizaje |

No obstante, es importante aclarar que no todos los atributos son aplicables, por lo tanto, debe priorizar los más relevantes dentro del software aplicable.

Dentro del aseguramiento de la calidad del software se encuentran dos roles de suma importancia debido a su gran participación dentro del proyecto como lo son los desarrolladores y el QA, es por esto que se ven inmersos dentro de las actividades propuestas en las actividades de calidad a continuación:

ETAPAS DE CONTROL Y CALIDAD	ACTIVIDAD
<b>Aplicación de metodologías del software</b>	Aplicación y supervisión de metodología agile scrum. - Supervisión en la planificación de Sprint y scrum diario.
<b>Revisiones técnicas formales</b>	Las revisiones que se plantean para los miembros del equipo de desarrollo para conseguir los niveles adecuados de calidad del producto son:  - Pruebas unitarias: consisten en aislar una parte del código y comprobar que funciona a la perfección. - Programación en pareja: técnica muy importante en la que una persona desarrolla y otra se sienta a su lado para revisar el código en tiempo de escritura.

	<ul style="list-style-type: none"> <li>- Refactorizar: realización de prototipos rápidos para mostrar al cliente la funcionalidad, después se procede a refactoriza para tener un código mejor y más limpio</li> <li>- Gestión de requisitos y trazabilidad: herramientas de ticketing que permiten gestionar o dar de alta incidencias y su resolución, y por otro lado gestionar requerimientos funcionales, entre otras cosas.</li> <li>- Code reviews y Code merges: Es una práctica obligatoria y exhaustiva, en la que parte del equipo revisa el código realizado por otros compañeros. Además, sirve de coaching.</li> </ul>
<b>Pruebas de software</b>	<p>Durante esta fase se diseñan casos de prueba que ayudan a la detección de errores producidos en las fases anteriores y no detectados durante la revisión técnica formal.</p> <p>Actividades propuestas dentro de la fase de pruebas del software:</p> <ul style="list-style-type: none"> <li>- Test de integración y de aceptación: técnicas realizadas para probar cómo es la interacción entre dos o más unidades del software en el caso de las de integración y en cuanto a las de aceptación para comprobar si el software cumple con las expectativas del cliente, con lo que el cliente realmente pidió.</li> <li>- Test de performance: Las pruebas de estrés deben hacerse con frecuencia, en cada sprint. Cada vez que se haga una funcionalidad, estresar.</li> </ul>
<b>Control de cambios y modificaciones (mantenimiento)</b>	<p>El proceso de control de cambios contribuye directamente a mantener la calidad del aplicativo, las actividades propuestas para esta sección son:</p> <ul style="list-style-type: none"> <li>- formalizar la petición</li> <li>- evaluar la naturaleza del cambio: si son normales que no comprometen el desarrollo del software o cambios de emergencia con o sin impacto sustancial.</li> <li>- y controlar el impacto: dependiendo del impacto del cambio.</li> </ul>
<b>Mediciones</b>	<p>Elegir medidas a realizar. debe definirse claramente lo que se quiere medir. Con el fin de describir la conducta del software, se establecen las métricas que miden, entre otros aspectos, la competencia, calidad, desempeño y complejidad del software.</p>
<b>Registros e informes</b>	<p>Generación de informes y registros a través de metodología scrum, como también de la documentación del software en cada una de sus etapas de desarrollo e implementación.</p>

**TABLA 43 ACTIVIDADES SCRUM**  
*Fuente: elaboración propia*

## Evaluación y control de riesgos

El análisis de riesgos tiene como objetivo dentro del proyecto la identificación de posibles acciones que generen la inestabilidad en el desarrollo de este, medir su probabilidad y el impacto que generaría. Según la ISO 21500 determina que “La finalidad de identificar los riesgos es determinar eventos de riesgo potenciales y sus características, que, si ocurren, pueden tener un impacto positivo o negativo sobre los objetivos del proyecto”. En esta sección del capítulo se expondrán los riesgos identificados, siendo estos

los posibles causantes de reprimir la implementación del proyecto, es por esto que su identificación y un plan de mitigación con el fin de contrarrestar las afectaciones es de suma importancia. La gestión tendrá el siguiente modelo basado en la siguiente estructura

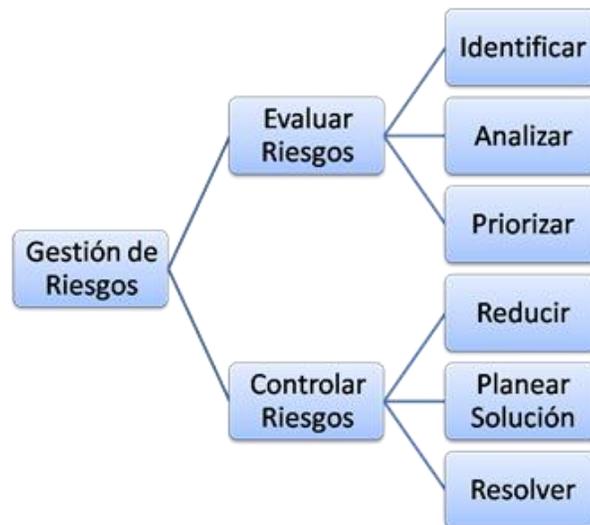


ILUSTRACIÓN 23 ESTRUCTURA GESTIÓN DEL RIESGO

*Fuente: elaboración propia*

Evaluación de los riesgos

*Descripción de los riesgos*

La identificación de los riesgos y su categorización dentro del proyecto se define en las siguientes categorías según la ISO 21500

- Financieros: (mercados, fiscalidad, falta de liquidez, tipos de interés, insuficiencia de capital propio, cambios macro-económicos)
- Normativos: (cambios regulatorios, incumplimiento de acuerdos, códigos éticos y de conducta)

- Estratégicos: (riesgo tecnológico, inadecuación de la estructura organizativa, pérdida de personal clave en la organización)
- Operacionales: (elaboración inadecuada de ofertas, insuficiente control de costes, fallos en el servicio y soporte al cliente, inseguridad en los sistemas de información)

ID	DESCRIPCIÓN DEL RIESGO	CATEGORÍA	ÁREA AFECTADA
1	Cambio de requerimientos por parte del Sponsor	Estratégico	Cronograma/costo
2	Incumplimiento de los hitos de acuerdo con la planificación del proyecto	Estratégico	cronograma
3	Falta de interés en el uso del aplicativo móvil por parte de los usuarios	Motivacional	Proyecto
4	Falta de herramientas tecnológicas para el uso del aplicativo por parte de los usuarios	Técnicos	Proyecto
5	Escaso rendimiento de calidad y tiempo de respuesta del aplicativo móvil	Operacionales	Calidad/costo
6	Fallo en el diseño estructural del aplicativo móvil	Estratégico	Cronograma/costo/calidad
7	Personal poco calificado	Estratégico	Cronograma/costo/calidad
8	Falta de presupuesto para la implementación	financiero	Cronograma/costo

**TABLA 44 DESCRIPCIÓN DEL RIESGO**

*Fuente: elaboración propia*

### *Análisis y priorización de los riesgos*

Dentro del análisis y priorización del impacto se tienen en cuenta factores como lo son:

- Probabilidad: la probabilidad de que un riesgo (amenaza) pueda ocurrir.
- Impacto: o (consecuencia): el efecto en el proyecto si el riesgo ocurre.

A partir de lo anterior se priorizan teniendo en cuenta la siguiente tabla base a través de los niveles: alto, moderado y bajo:

MATRIZ DE PRIORIZACIÓN			
Probabilidad	Impacto		
	Bajo	Moderado	Alto
Alto	2-Media	1-Alta	1-Alta
Moderado	3-Baja	2-Media	1-Alta
Bajo	3-Baja	3-Baja	2-Media

**TABLA 45 MATRIZ DE PRIORIZACIÓN DEL RIESGO**

*Fuente: Protocolo ISO 21500*

ID	DESCRIPCIÓN DEL RIESGO	PROBABILIDAD	IMPACTO	PRIORIDAD
1	Cambio de requerimientos por parte del Sponsor	Moderado	Alto	Alto
2	Incumplimiento de los hitos de acuerdo con la planificación del proyecto	Baja	Moderado	Baja
3	Falta de interés en el uso del aplicativo móvil por parte de los usuarios	Baja	Alta	Media
4	Falta de herramientas tecnológicas para el uso del aplicativo por parte de los usuarios	Moderado	Moderado	Medio
5	Escaso rendimiento de calidad y tiempo de respuesta del aplicativo móvil	Baja	Alto	Medio
6	Fallo en el diseño estructural del aplicativo móvil	Baja	Alto	Medio
7	Personal poco calificado	Baja	Alto	Medio
8	Falta de presupuesto para la implementación	Baja	Alto	Medio

**TABLA 46 MATRIZ DE PRIORIZACIÓN DE LOS RIESGOS DEL PROYECTO**

*Fuente: elaboración propia*

### Control de los riesgos

A continuación, se define el control y tratamiento a realizar a cada uno de los riesgos analizados y priorizados dentro del proyecto

ID	DESCRIPCIÓN DEL RIESGO	MITIGACIÓN	CONTINGENCIA	CUANTIFICACIÓN EN TIEMPO Y DINERO
1	Cambio de requerimientos por parte del Sponsor	Manejar un formato de validación del proyecto por parte del sponsor.	Gestionar análisis de los requerimientos nuevos i/o funcionalidades del sistema	Estimación en tiempo para realizar cambio máximo 2 semanas. Estimación en dinero:
2	Incumplimiento de los hitos de acuerdo con la planificación del proyecto	Resaltar las fechas límites para la entrega de las actividades dentro de la ruta crítica en el cronograma	Agilizar y definir nuevos hitos para la realización de las actividades atrasadas.	Estimación en tiempo: según la ruta crítica Estimación en dinero: \$0
3	Falta de interés en el uso del aplicativo móvil	Divulgación constante sobre el uso y la	Hacer el uso del aplicativo de manera primordial y de	Estimación en tiempo: Constante a lo largo de la

	por parte de los usuarios	importancia del aplicativo web.	carácter obligatorio por parte de los usuarios.	implementación del proyecto. Mínimo 6 meses Estimación en dinero: \$0
4	Falta de herramientas tecnológicas para el uso del aplicativo por parte de los usuarios	Disponer de una estructura tecnológica de red wifi disponible para el uso del aplicativo.		Estimación en tiempo:  Estimación en dinero: \$0
5	Escaso rendimiento de calidad y tiempo de respuesta del aplicativo móvil	Análisis y monitoreo del rendimiento en el aplicativo móvil	Gestionar cambios en el análisis y desarrollo del aplicativo móvil	Estimación en tiempo: Máximo 2 semanas Estimación en dinero: \$0
6	Fallo en el diseño estructural del aplicativo móvil	Disposición de mantenimiento disponible ante posibles fallas en la implementación.	Reestructuración del diseño del aplicativo	Estimación en tiempo: máximo 3 semanas Estimación en dinero: \$0
7	Personal poco calificado	Capacitación constante del equipo de trabajo	Cambio de personal poco productivo.	Estimación en tiempo: Máximo 1 semana Estimación en dinero: \$0
8	Falta de presupuesto para la implementación	Concientizar a los interesados de la importancia en la implementación del proyecto a través de reuniones.	Congelamiento estimado en el desarrollo del proyecto	Estimación en tiempo: Máximo 3 semanas Estimación en dinero: \$0

**TABLA 47 CONTROL DE LOS RIESGOS DEL PROYECTO**

*Fuente: elaboración propia*

### **Planificación de las comunicaciones**

Teniendo en cuenta la importancia que cumplen los stakeholders antes, durante y después del desarrollo del proyecto se ve la necesidad de crear y gestionar un plan de comunicaciones donde se realiza el intercambio de información específica del proyecto creando así canales seguros entre todos los interesados. A continuación, se presenta una tabla exponiendo cada uno de los criterios importantes durante la comunicación como: el mensaje a difundir, interesados, canal de comunicación, formato utilizado, responsable y frecuencia.

<b>MENSAJE</b>	<b>GRUPO DE INTERÉS</b>	<b>CANAL</b>	<b>FORMATO</b>	<b>RESPONSABLE</b>	<b>FRECUENCIA</b>
<b>Presentación del proyecto</b>	Institución	Reunión ejecutiva	presentación	Director del proyecto	Única vez
<b>Divulgación del desarrollo del proyecto</b>	Institución/ Empleados/ Estudiantes	Correo electrónico institucional/ charlas/ Avisos publicitarios	Online/ Presentaciones/	Institución/ Facultades	Semanal
<b>Reunión de avances del proyecto</b>	Institución/ Empleados/	Reunión ejecutiva	presentación	Director del proyecto	1 vez al mes
<b>Finalización del proyecto</b>	Institución/ Empleados/	Reunión ejecutiva	presentación	Director del proyecto	Única vez
<b>Divulgación de la implementación del proyecto</b>	Institución/ Empleados/ Estudiantes	Correo electrónico institucional/ charlas/ Avisos publicitarios	Online/ Presentaciones/	Institución/ Facultades	Semanal

**TABLA 48 PLAN DE COMUNICACIONES**

Fuente: elaboración propia

## CONCLUSIONES

Una vez analizada la problemática expuesta dentro del proyecto se procedió a realizar un amplio análisis de diseños e implementación de una aplicación móvil en cual se introdujo como una posibilidad de mantener un control y registro de estudiantes y docentes descrita en los capítulos anteriores.

Se llevaron a cabo un análisis, el cual pueden demostrar al cliente la factibilidad de desarrollo e implementación del aplicativo móvil en las instituciones de educación superior, ya que existen tecnologías, recursos y conocimientos necesarios para implementarlas.

Al haber seleccionado el Framework flutter, descrito en capítulo 4 del proyecto, nos permite realizar y presentar de forma ágil el desarrollo del aplicativo multiplataforma en los diferentes sistemas operativos móviles como Android y iOS, por lo que se pudo mostrar un avance rápido al cliente.

Además, es esta primera instancia se ha trabajado con distintas tecnologías escogidas (Flutter,git, DBwear, Sprint Boot, Azure etc.) en conjunto con los desarrolladores del proyecto , las cuales en su mayoría pertenecen a tecnologías que hoy día se utilizan demanda el mundo laboral que sirven para ayudas futuras.

Toda la documentación del proceso es dada a conocer para análisis del aplicativo móvil como los diseños del prototipo de aplicación como se pudo evidencias en el capítulo anteriores a lo largo del proyecto.

Finalmente, se ha podido cumplir de manera satisfactoria con los objetivos establecidos, poniendo en práctica los conocimientos adquiridos durante el proyecto.

## RECOMENDACIONES

Para la puesta en marcha e implementación del aplicativo móvil se tendrá que realizar mantenimiento posteriormente a la entrega del aplicativo el cual se calculara su precio de acuerdo con el nuevo requerimiento, con el objetivo de mejorar y optimizar el software, así como también remediar defectos.

El aplicativo móvil queda a mejoras futuras permitiendo que incluyan nuevos módulos como consultas de notas, eventos entre otras, lo que les permite tener diversidad en la aplicación a la hora de utilizarla, lo cual con lleva a tener un mejor flujo de usabilidad.

También se recomienda a los docentes seguir de manera sistemática los procedimientos que se expone en los manales de usuario para conseguir un correcto funcionamiento del aplicativo. realizar capacitaciones a docenes y estudiantes con el objetivo de que conozcan y utilicen el aplicativo móvil adecuadamente. Además, que las instituciones de educación superior cuenten con infraestructura tecnológica de conexión a internet vía Wifi para conectar la aplicación y realizar los registro por parte de los docentes y estudiantes.

## REFERENCIAS

- ¿Qué es Scrum? (Online). Scrum.Org. <https://www.scrum.org/resources/blog/que-es-scrum>
- 1&1 IONOS España S.L.U. (2021, 21 junio). *¿Qué es un plugin?* IONOS Digitalguide.  
<https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-plugin/>
- A. (2021a, marzo 24). *¿Qué es una aplicación móvil?* Anincubator Website.  
<https://anincubator.com/que-es-una-aplicacion-movil/>
- Andalucía, J. U. N. T. A. (s. f.). *Atributos de los requisitos | Marco de Desarrollo de la Junta de Andalucía*. juntadeandalucia. Recuperado 5 de julio de 2021, de  
<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/409>
- Barquinero, J. M. M. (2020, 26 febrero). *Tipos de relaciones en diagramas de casos de uso. UML.* / Blog SEAS. Blog de SEAS. <https://www.seas.es/blog/informatica/tipos-de-relaciones-en-diagramas-de-casos-de-uso-uml/>
- Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change, Second Edition*. Addison Wesley Professional.
- C. (2021, 10 febrero). *¿Estándares, metodologías o marcos de trabajo? ¿Sabes diferenciarlos?* CEOLEVEL. <https://www.ceolevel.com/estandares-metodologias-o-marcos-de-trabajo-sabes-diferenciarlos>
- Canós, J., Letelier, P., & Penadés, M. (2003). *Métodologías Ágiles en el Desarrollo de Software*. Obtenido de [http://noqualityinside.com.ar/nqi/nqifiles/XP\\_Agil.pdf](http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf)
- Cevallos, K. (2015, 26 julio). *UML: Casos de Uso*. INGENIERÍA DEL SOFTWARE.  
<https://ingsoftwarekarlacevallos.wordpress.com/2015/06/04/uml-casos-de-uso/>
- Cockburn, A. (2007). *Agile Software Development The Cooperative Game (2da ed.)*. Boston, USA: Addison Wesley

- Cunningham, W. (2 de mayo de 2015). Manifiesto for agile development. Obtenido de <http://agilemanifesto.org/>
- Devs, Q. (2019, 8 julio). *✓Qué es Flutter y por qué utilizarlo en la creación de tus apps*. Quality Devs. <https://www.qualitydevs.com/2019/07/05/que-es-flutter/>
- Diego Lázaro. (2018). *Introducción a los Web Services*. <https://diego.com.es/introduccion-a-los-web-services>. <https://diego.com.es/introduccion-a-los-web-services>
- DSDM Consortium. (20 de Febrero de 2015). Obtenido de DSDM Consortium From Page: <http://www.dsdm.org/>
- DSDM Consortium. (2003). *DSDM: Busines Focused Development*. Addison-Wesley.
- Fernández, Y. (2019, 23 agosto). *API: qué es y para qué sirve*. Xataka. <https://www.xataka.com/basics/api-que-sirve>
- Ian Sommerville. (2005). *Ingeniería de Software* Pearson Educations S.A p.6
- M. O. Herrera, "Métodos y técnicas para la gestión de proyectos software," Master's thesis, Universidad de Sevilla, España, 2010. [Online]. Available: <http://bibing.us.es/proyectos/abreproy/70193/fichero/3.+METODOLOG%C3%8DAS+DE+GESTI%C3%93N+DE+PROYECTOS.pdf>
- Marketing. (2021, 13 enero). *¿Qué es Microsoft Azure? ¿Cómo funciona?* Tecon. <https://www.tecon.es/que-es-microsoft-azure-como-funciona/>
- Muradas, Y. (2020, 13 abril). *Qué es NPM y para qué sirve*. OpenWebinars.net. <https://openwebinars.net/blog/que-es-node-package-manager/>
- Pressman, r. s. (2002). *ingeniería del software: un enfoque practico (5a. ed.)*. madrid: mcgraw-hill interamericana.
- Proceso y Roles de Scrum. (Online). [www.softeng.es](http://www.softeng.es). <https://www.softeng.es/es-empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>

Rodríguez, E., Rodríguez, E., Rodríguez, E., & Rodríguez, E. (2020, 27 enero). *Qué es backend y por qué es tan importante para tu sitio web*. Seoestudios.

<https://www.seoestudios.es/blog/que-es-backend-web/>

Roggers Perssman. (2010). *Ingeniera de software un enfoque practico*. Mac Graw Hill 2010. P.10

Schawer, K. (1995). SCRUM Development Process

SCRUM ORG. (10 de Marzo de 2015). Scrum.orh The Home of SCRUM. Obtenido de

<https://www.scrum.org/>

Stark, K. (2021, 7 junio). *Qué es la arquitectura de microservicios*. Evaluando Software.

<https://www.evaluandosoftware.com/que-es-la-arquitectura-de-microservicios/>

Stefaniak, P. (2019, 10 septiembre). *¿Qué es Backend y Frontend?* Descubre Comunicación.

<https://descubrecomunicacion.com/que-es-backend-y-frontend/>

The Agile Alliance. (2014). The Twelve Principles of Agile Software. Obtenido de

<http://www.agilealliance.org/the-alliance/the-agile-manifesto/the-twelve-principles-ofagile-software>

Wells, D. (15 de Marzo de 2015). Extreme Programming: A gentle introduction. Obtenido de

<http://www.extremeprogramming.org/>

World Rugby. (12 de Marzo de 2015). Obtenido de <http://www.worldrugby.org/>