

**Primera entrega proyecto semestral
Ingeniería de Software**

Plan de Administración de Proyectos

Grupo:

SauveGuard

Integrantes:

Angela Sofia Moreno Rodrgiuez
Kevin Stid Pelaez Nimisica
Juan Sebastian Valderrama
Juan Diego Osorio Hernandez
Federico Torres Mojica

Nombre Proyecto:

SOS Ciclistas

Bogotá, 2019

1 Historial de cambios

Descripción	Fecha	Persona(s) a cargo
Creación del documento (portada, tabla de cambios, resumen, tabla de contenido)	05/08/2019	Sofia Moreno
Vista general de cada sección de la plantilla y escritura de ideas concretas en cada una	06/08/2019	Sofia Moreno, Sebastián Valderrama, Kevin Pelaez, Juan Diego Osorio, Federico Torres
Visión del producto, propósitos, alcances y objetivos. Supuestos y restricciones	6/08/2019	Juan Diego Osorio
Mejora Objetivos, lista de restricciones y supuestos, lista de entregables y glosario.	9/08/2019	Juan Diego Osorio
Mejora resumen, tabla entregables, índice tablas(añadir), adición referencias	12/08/2019	Sofia Moreno
Revisión de ortografía del texto, y glosario completo	12/08/2019	Juan Diego Osorio
Ambiente de trabajo, inicio del proyecto y tabla de aprendizaje	15/08/2019	Juan Diego Osorio
Modelo de ciclo de vida, plan de aceptación de las entregas, organización del proyecto y comunicación	17/08/2019	Sofia Moreno
Inserción de diagramas BPMN para el manejo de cambios, administración y manejo de riesgos con su respectiva explicación escrita y organigrama.	19/08/2019	Kevin Peláez

Revisión general documento en cuanto a redacción y ortografía	20/08/2019	Sebastian Valderrama, Federico Torres
Inserción de diagrama BPMN para el manejo de alteraciones	21/08/2019	Kevin Peláez
Modificación cierre del proyecto, modelo de ciclo de vida, entregables del proyecto y Administración de configuración y documentación	21/08/2019	Sofia Moreno
Datos de estimación y realización anexo de estimación por casos de uso y de dinero	21/08/2019	Federico Torres
Figura capacitación en Uдеми para inicio del proyecto	21/08/2019	Sebastián Valderrama Federico Torres

2 Resumen

A lo largo de este documento nos encargaremos de desarrollar nuestro proyecto semestral, la aplicación móvil SOS ciclistas, siguiendo las indicaciones del Plan de Gestión de Proyectos SPMP. Esto con el fin de establecer las políticas, procedimientos, tareas y recursos del proyecto. Además, se moldeará por medio de diagramas y explicaciones, los pasos a seguir para que la aplicación pensada cumpla en su totalidad con las necesidades presentadas por el cliente.

Así mismo, a medida que avancemos en el desarrollo del proyecto se podrá identificar desde la descripción más general de la aplicación, la creación de modelos explicativos, determinación de alcance y de las fronteras, hasta la explicación de las herramientas utilizadas para estimar, desarrollar, planear y dividir el trabajo entre los integrantes.

En general, en este documento quedarán sentadas las bases del proceder en cualquiera de las situaciones que se puedan presentar durante el desarrollo del proyecto, como los riesgos probables, la manera en la que se manejarán, el modo de distribución de los entregables y su respectivo contenido, etc. Se definirá completamente el modelo de ciclo de vida que se utilizará para este proyecto y daremos razones de su funcionamiento. Del mismo modo, se explicará cómo se administrarán las solicitudes de cambio y otras actividades que se pueden presentar a lo largo del proyecto por medio de diagramas BPMN donde se evidencia qué se debe hacer y quién es el responsable de hacer las modificaciones pertinentes, con el fin de entender qué actores están implicados, de qué manera participan, qué rol juegan, cuál es su importancia en el grupo de trabajo y cuáles son sus respectivas tareas en la organización.

Finalmente, tras todo esto, pondremos en constancia cómo hicimos las estimaciones para hacer evidente la viabilidad del proyecto, y cuál será el proceder en la última entrega con el fin de brindar soporte y hacer un seguimiento a la aplicación.

Se espera que con este documento tanto nuestro cliente, la persona que nos dará seguimiento; como los participantes activos, que en este caso serán los ciclistas; tengan una visión más amplia del trabajo realizado durante los meses de desarrollo y la vista general de lo que es el proyecto y cómo funciona. Y posiblemente que las personas interesadas en desarrollar un proyecto similar al nuestro, o que deseen continuarlo en el futuro, tengan la documentación necesaria para sentar bases, estimar, saber un poco del trabajo que conlleva una aplicación de este estilo y ejecutar cambios sobre la manera de proceder establecida por nuestro equipo de trabajo.

3 Tabla de contenidos

1	Historial de cambios.....	2
2	Resumen	4
3	Tabla de contenidos	5
4	Lista de figuras	6
5	Lista de tablas.....	7
6	Vista general del proyecto	8
6.1	Visión del producto.....	8
6.2	Propósito, alcance y objetivos.....	9
6.3	Supuestos y restricciones	10
6.4	Entregables	11
6.5	Evolución del plan	13
6.6	Glosario.....	14
7	Contexto del proyecto.....	16
7.1	Modelo de ciclo de vida.....	16
7.2	Lenguajes y herramientas	19
7.3	Plan de aceptación del producto	20
7.4	Organización del proyecto y comunicación	22
8	Administración del proyecto.....	25
8.1	Métodos y herramientas de Estimación	25
8.2	Inicio del proyecto	27
8.3	Planes de trabajo del proyecto.....	29
9	Monitoreo y control del proyecto.....	31
9.1	Administración de requisitos	31
9.2	Monitoreo y control del progreso.....	32
9.3	Cierre del proyecto.....	34
10	Entrega del producto	36
11	Procesos de soporte.....	36
11.1	Ambiente de trabajo.....	36
11.2	Análisis y administración de riesgos	38
11.3	Administración de configuración y documentación	41
11.4	Control de calidad	43
12	Anexos	46
13	Referencias	46

4 Lista de figuras

1. Variación actividades modelo RUP.....	17
2. Modelo de ciclo de vida adaptado.....	18
3. Organigrama.....	25
4. Diagrama general de capacitación inicial.....	28
5. Módulos de aprendizaje de Ionic.....	28
6. Work breakdown structure.....	30
7. Work breakdown structure sobre funcionalidad.....	31
8. Modelo para cambios.....	32
9. Migración a otras ciudades.....	39
10. Deserciones.....	40
11. Solución de retrasos.....	40
12. Solución de conflictos.....	41
13. Manejo de alteraciones.....	42

5 Lista de tablas

1. Tabla entregables.....	11
2. Entidades interesadas.....	22
3. Jerarquía organización.....	23
4. Capacitación previa.....	27
5. Artefactos de documentación y código	42
6. Control de calidad de la documentación.....	43
7. Control de calidad del software.....	44

6 Vista general del proyecto

Este proyecto tendrá un gran impacto en la comunidad ciclística, pues se le dará una solución a un problema actual que es la seguridad de los bici-usuarios. Es por este motivo que la aplicación atenderá de manera especial las emergencias anunciadas de modo que se gestionen los casos y solicitudes de los ciclistas y personas del común de forma rápida y efectiva.

La aplicación está dirigida para que un bici-usuario o cualquier persona que quiera realizar una solicitud y/o anunciar alguna emergencia donde haya un tercero involucrado, pueda por medio de un pequeño formulario, explicando de manera breve lo sucedido y enviando la ubicación del accidente, solicitar el servicio de emergencias pertinente (ambulancia, defensa civil, bomberos o policía) e informarse sobre cuál es el proceder en esa situación para evitar que esta empeore. Además, la persona que recibe la solicitud será uno de los coordinadores del grupo de WhatsApp ya existente SOS CICLISTAS. Él se encargará tanto de brindar apoyo moral como de dar instrucciones al usuario sobre el procedimiento que debe realizar para prevenir mayores complicaciones. Todo esto lo hará el coordinador mientras, a su vez, gestiona la solicitud de emergencia y la llegada del servicio al punto donde ocurrió el incidente, dando igual importancia a todos los casos que se presenten al tiempo y atendiéndolos con la misma efectividad. Esto es posible gracias a los múltiples convenios ya establecidos entre el grupo de administración de WhatsApp y las empresas prestadoras del servicio. Al finalizar la gestión de la emergencia se creará un reporte con el cual el administrador a cargo pueda hacer un seguimiento a la(s) persona(s) afectada(s) brindando al servicio de los demás su propio conocimiento y compañía con la finalidad de brindar una asistencia mucho más humana.

6.1 Visión del producto

La aplicación SOS Ciclistas, permitirá al usuario suplir las necesidades de atención de ambulancias en tiempo real y con un seguimiento por parte de los coordinadores del grupo SOS CICLISTAS. También podrá automatizar la atención ante posibles incidentes de la vida real como atracos, incendios, tiroteos, etc. Por medio de una gestión realizada directamente desde la aplicación sin la necesidad de una persona como intermediaria para cubrir dichas emergencias.

El usuario tendrá a disposición una aplicación móvil que le permitirá ponerse en contacto con un administrador, el cual realizará los procedimientos adecuados para gestionar el pedido de una ambulancia, utilizando la información suministrada por el usuario al llenar un pequeño formulario de preguntas básicas sobre el accidente ocurrido. A su vez, la aplicación registrará el tiempo que tarda la gestión de este hecho, brindará automáticamente información oportuna sobre cómo manejar el accidente para evitar riesgos posteriores y permitirá tener la posibilidad de hablar directamente con el administrador por medio de un chat interno y generarle a este un reporte de la solicitud una vez se haya gestionado completamente. Además de

esta solicitud, la aplicación podrá manejar de manera autónoma el envío de otros tipo de riesgos, registrándolos en una base de datos autónoma y transmitiendo la información relevante a todos los usuarios al tiempo.

Igualmente, la aplicación brindará a los administradores la posibilidad de establecer contacto con el usuario que pidió el servicio por cualquier medio de la aplicación, Informar sobre procedimientos frecuentes o capacitaciones, incidentes graves o noticias importantes aparte de las notificaciones automáticas. Finalmente, SOS Ciclistas se encargará de crear reportes sobre las emergencias gestionadas y atendidas con el fin de tener un seguimiento de los incidentes resueltos, permitiendo estadísticas semanales o mensuales de los reportes acontecidos, lo que permitirá una recolección importante de datos y ayudará a una mayor coordinación y la mejora continua en la prestación del servicio por parte de la administración.

6.2 Propósito, alcance y objetivos

- *Propósito:* La aplicación se realizará con el fin de automatizar y gestionar la petición, seguimiento y reporte de servicios de emergencia para los bici-usuarios y será una herramienta de los coordinadores del grupo SOS CICLISTAS para el control adecuado de los accidentes, la comunicación directa con los clientes y el envío de información pertinente a los mismos.
- *Alcance:* La aplicación permitirá enviar las solicitudes de ambulancia a un administrador y las solicitudes de otras emergencias serán atendidas exclusivamente por la aplicación, sin necesidad de una persona detrás. No se encargará por sí misma de enviar una solicitud a una ambulancia, ni tampoco registrar la disponibilidad de las mismas o el lugar en el que se encuentran en un momento determinado. La aplicación no podrá pedir solicitudes de otro tipo automáticas, ya que sería necesario una vinculación con cada una de las entidades que prestan estos servicios, lo que supone realizar trámites entre entidades mucho más complejos. Sin embargo, sería una posibilidad en el futuro para automatizar cualquier envío de solicitud. La aplicación permitirá la elaboración de registros de los incidentes y permitirá informar a cada usuario de la aplicación sobre sucesos que se estén presentando frecuentemente en una zona específica. Del mismo modo, la aplicación generará los reportes de los accidentes solicitados y los guardará en una base de datos o en el medio que el dueño del producto finalmente decida.
- *Objetivos:*
 - *Objetivo general:* Desarrollar una aplicación que permita la automatización y seguimiento de los accidentes de los bici-usuarios y su atención oportuna inmediatamente después del incidente y con los servicios necesarios.
 - *Objetivos Específicos:*
 - Crear una aplicación de un manejo y uso simple, que permita enviar la solicitud de cualquier tipo de emergencia ante una entidad específica (Bomberos, ambulancia, policía y/o defensa civil).

- Permitir la interacción entre el usuario y el administrador de la aplicación por medio de llamadas y mensajería instantánea (chat).
- Facilitar a los administradores la generación de reportes inmediatos una vez sea atendida la emergencia.
- Brindar tanto al usuario como al administrador la información sobre el tiempo promedio de respuesta que tiene la gestión del accidente desde la aplicación.

6.3 Supuestos y restricciones

LISTA DE SUPUESTOS:

- Gracias a la información suministrada por la Product Owner, no se tendrán más de 10 peticiones al mismo tiempo. Esto gracias a que el promedio de accidentes que se gestionan en los grupos de WhatsApp es de máximo 5. Sin embargo, se planteará como máximo 20 solicitudes simultáneas, ya que se podrán gestionar automáticamente las demás solicitudes de emergencia.
- El GPS del dispositivo móvil en el cual se pida el servicio debe estar activo. De lo contrario no se podrá acceder a la localización.
- Siempre existirá al menos una ambulancia disponible y un administrador recibiendo las peticiones de ambulancia. La recepción de las demás solicitudes será autónoma de la aplicación, no de los administradores.
- La solicitud de ambulancia se hará llenando un formulario sencillo con datos sobre la ubicación, si existen heridos en el accidente, y una descripción breve del lugar, con la posibilidad de anexar fotos. Una vez enviado el formulario, esta solicitud se continuará gestionando sin importar que se cierre la aplicación o se termine el proceso desde el dispositivo móvil.
- Se cumplirán tanto los horarios, calendarios estipulados, y entregas definidas, como las tareas asignadas a cada uno de los miembros del grupo dependiendo de su rol en la organización.
- El contacto con el product owner será constante a lo largo de todo el semestre para lograr una proyección del proyecto a futuro y desarrollarlo con calidad.
- Es necesario que el usuario complete en su totalidad el formulario de información obligatoria sobre el accidente para poder gestionarlo posteriormente.
- El chat que se tendrá entre el administrador y el usuario será única y exclusivamente usado si alguna de las dos partes necesita comunicarse más a fondo con el otro, ya que no será la herramienta de gestión de la aplicación.

LISTA DE RESTRICCIONES:

- El dispositivo debe contar por lo menos con una conexión a internet para que el usuario pueda enviar la petición desde la aplicación.
- El usuario debe contar en su paquete de telefonía mínimo con minutos o datos para contactar al administrador en caso de que la aplicación falle y no se pueda gestionar el accidente desde la misma. Para ello se incorporarán en la aplicación un listado de números telefónicos pertenecientes a los coordinadores del programa para gestionar la emergencia de manera manual directamente con ellos.
- El usuario no podrá gestionar más de un solo accidente desde la aplicación. Esto no significa que no se puede atender a más de un solo herido, pero si quiere decir que no puede gestionar más de un solo accidente del cual se envió la solicitud.
- Los administradores deberán continuar en comunicación mediante el chat interno de WhatsApp, dado que no es necesario generar una comunicación dentro de la aplicación si se supone que es dirigida únicamente para las emergencias.

Para ampliar información ver en carpeta: Anexo1-Requerimientos.

6.4 Entregables

Realizaremos tres tipos de entregas, las primeras serán las entregas parciales a la persona que llevará semanalmente nuestro proceso (profesor) con el fin de recibir una retroalimentación y una calificación cuantitativa de nuestro trabajo. El otro tipo de entregas serán las que le haremos al cliente final y las terceras serán las entregas internas entre nuestro equipo de trabajo para medir avance:

Nombre entregable	Descripción	A quién va dirigido	Fecha de entrega
"Software Project Management Plan" (SPMP) entrega para revisión previa	Entrega del plan de gestión de proyectos para revisión previa	Profesor	Semana del 12 al 16 de agosto de 2019
"Software Project Management Plan" (SPMP)	Entrega documento final para recibir calificación cuantitativa	Profesor	20 de agosto de 2019
Botones y pantallas	Entrega de la parte principal de la visual del proyecto lista para	Interna	Semana del 17 al 20 de septiembre de 2019

	la segunda entrega y un delante de la funcionalidad		
“Software Requirements Specification” (SRS) entrega para revisión previa	Entrega documento de levantamiento de requerimientos y caso de uso más extenso de la aplicación implementado para revisión previa	Profesor	Semana del 7 al 11 de octubre de 2019
“Software Requirements Specification” (SRS)	Entrega documento de levantamiento de requerimientos y caso de uso más extenso de la aplicación implementado para recibir calificación cuantitativa	Profesor	15 o 17 de octubre de 2019
Funcionalidad principal	Entrega del caso de uso más extenso de la aplicación implementado para retroalimentación y pautas para continuar	Cliente	Semana del 14 al 18 de octubre de 2019
Casos de uso adicionales	Entrega de 2 casos de uso adicionales y un avance sustancial de la aplicación completa	Interna	Semana del 21 al 25 de octubre de 2019
“Software Design Document ” (SDD)	Entrega de documentos de diseño así como los últimos dos documentos realizados	Profesor	Semana del 11 al 15 de noviembre de 2019

	debidamente corregidos y el código terminado y funcionando, para revisión previa		
“Software Design Document ” (SDD)	Entrega de documentos de diseño así como los últimos dos documentos realizados debidamente corregidos y el código terminado y funcionando, para recibir calificación cuantitativa	Profesor	Semana del 18 al 22 de Noviembre de 2019
Aplicación para prueba	Entrega del código completo funcionando para realizar pruebas con usuarios reales y recibir cambios pertinentes antes de la entrega final	Cliente	Semana del 11 al 15 de Noviembre de 2019
Entrega final	Entrega del código final con modificaciones hechas y con documentos requeridos para dar por finalizado el proyecto	Cliente	Semana del 25 al 29 de Noviembre de 2019

1. Tabla entregables

6.5 Evolución del plan

En nuestro equipo de trabajo, al estar constituido por 5 integrantes, debe haber una comunicación asertiva entre los mismos para asegurar una participación equitativa en la formación del plan de gestión del proyecto y actuar sobre las diferentes etapas de este. La manera en la que SauveGuard lo garantiza es por medio de reuniones semanales en salas de la Biblioteca de la universidad reservadas con anterioridad para los martes. Esto con el fin de revisar entre todos las partes del documento ya

realizadas para hacer o no modificaciones, revisar las faltantes para realizar anotaciones a medida que vaya pasando el tiempo y se vaya mirando la viabilidad del proyecto y cada sección, y anotar cosas puntuales para que sólo reste redactarlas mejor.

Todos estos cambios se deben ver oficializados en la tabla de la sección 1.0 que hace referencia al historial de cambios dónde dice por quién fue realizada cada modificación y la fecha de esta.

Aparte de eso se tendrá la flexibilidad de modificación, puesto que se está en constante comunicación con el cliente y por ende con nuevas sugerencias y requerimientos para la aplicación.

En la sección 8.0 llamada “Administración del proyecto” se encontrarán las bases iniciales dadas como acuerdos para empezar y las tareas de cada integrante para el plan de trabajo del proyecto entre las cuales está lo antes mencionado que es pactado en cada reunión.

Finalmente, en la sección 9.0 llamada “Monitoreo y control del proyecto” podemos ver el “estado de salud” del documento y del proyecto así como las acciones de control a ejecutar en casos como los cambios en los requisitos, el monitoreo en general del proceso de cada integrante en el proyecto, el proceder para la identificación de riesgos y el tratamiento de los mismos y la identificación de los principales ítems de configuración y cómo evolucionarán durante el tiempo, particularmente los artefactos de documentación y el código.

6.6 Glosario

Bici-usuario: Persona cuyo principal medio de transporte es la bicicleta para movilizarse en la ciudad.

Accidente: Suceso que le puede ocurrir a una persona o bici-usuario que puede atentar ante su salud o el bienestar de sus pertenencias. Esta puede suceder por causas naturales o humanas que normalmente son imprevistas.

Emergencia: Situación de gravedad que debe ser atendida de manera inmediata por la posibilidad de salirse de control completamente. Puede involucrar tanto un desastre natural, como la posibilidad de que una persona fallezca a causa de un accidente.

Cliente: Persona a la cual está dirigida la aplicación y que hace uso de la misma de manera recurrente. Es el grupo de personas a la cual se le va a vender la aplicación. En este momento son los administradores y coordinadores del grupo SOS ciclistas.

Usuario: Persona natural que posee la aplicación SOS ciclistas y se encuentra registrado en la base de datos de la misma. Propiamente serán los bici-usuarios pertenecientes a los grupos de WhatsApp ya existentes.

Formulario de solicitud: Encuesta breve que necesitan los administradores del grupo SOS ciclistas para poder pedir una ambulancia.

Ambulancia: Vehículo acondicionado con instrumental de primeros auxilios, diseñado con el fin de transportar personas heridas o enfermas a un centro médico para su asistencia completa.

Policía: Cuerpo armado de orden civil diseñado para proteger y salvaguardar el mantenimiento de la convivencia como condición necesaria para el ejercicio de los derechos y libertades públicas de todos los habitantes de un país.

Defensa Civil: Entidad de gestión de riesgos ambientales, sociales y ambientales para mejorar la seguridad, bienestar y calidad de vida de la población en el territorio nacional

Gestión de emergencia: Es cuando el administrador recibe toda la información sobre una emergencia enviada y utiliza sus contactos con la empresa de ambulancias para solicitar una y enviarla al lugar que el usuario agregó al formulario de solicitud

SOS: Señal de socorro universal. Conocida desde la marina como “Save Our Ship” o “Save Our Souls”. Utilizada en cualquier circunstancia para mandar un mensaje de alerta o socorro a cualquier persona.[4]

Requerimientos: Actividad, condición o circunstancia, que es necesaria para obtener un resultado querido. En este caso, son las necesidades primordiales que debe suplir la aplicación y que el usuario necesita automatizar.[5]

Angular: Es un entorno de trabajo para la elaboración de aplicaciones web de código abierto. Se utilizará para desarrollar la aplicación sin necesidad de pagar una licencia de uso o entornos muy robustos para el grupo de trabajo.[6]

Ionic: Es un kit de desarrollo de software para el desarrollo de aplicaciones móviles híbridas de código libre. Esta herramienta permitirá que la aplicación pueda ser utilizada no solamente por un único sistema operativa.[7]

Interfaz: Es una conexión que se presenta entre dos máquinas o dispositivos de modo que sean de utilidad una de la otra. En informática, conocida mucho más como **interfaz de usuario**, es el medio en el que se permite la comunicación directa entre una persona y un dispositivo.[8]

Backlog: En las metodologías ágiles, el backlog o **product backlog** es una lista ordenada de tareas que están pendientes en una organización o desarrollo de un producto-servicio. Estas se dividen entre diferentes grupos para determinar el avance en el desarrollo de cada tarea y tener un control y flexibilidad sobre el producto final.[9]

Scrum: Es una de las metodologías ágiles utilizadas comúnmente en las empresas. Se puede comprender como un conjunto de buenas prácticas en un macro laboral, con el fin de laborar colaborativamente y poder cumplir con las necesidades de un proyecto.[10]

RUP: El proceso unificado racional es una metodología o un modelo de ciclo de vida que fue comúnmente usado para el desarrollo de software. Basado en el uso fundamental de los requerimientos para generar su desarrollo y despliegue, busca agilizar los procesos de desarrollo implementando un requerimiento a la vez a tal punto que esté terminado y poder continuar con otro.[11]

Framework: En español conocido como Entorno de Trabajo, es una aplicación genérica que permite el desarrollo de proyectos y/o aplicaciones de una manera más fácil, ya que se basa en el uso de código reutilizable, junto a un conjunto de

herramientas de desarrollo que permiten un mayor manejo y control del producto de software que se desea desarrollar.[12]

SO: Conocido comúnmente como Sistema Operativo, es el software más importante de los usados en las computadoras. Este coordina y dirige servicios y aplicaciones para los usuarios, específicamente, el hardware y el software de la máquina. A su vez, permiten la interacción entre el computador y el ser humano a partir de los dispositivos como mouse, teclado, pantalla, entre otros.[13]

Android: Android es un sistema operativo para teléfonos móviles, con un núcleo de sistema operativo libre, gratuito y multiplataforma. Está presente en la mayoría de los celulares de todo el mundo y su lenguaje está basado en su mayoría en Java.[14]

IOS: Del inglés “Iphone Operative System”, es un sistema operativo de la empresa Apple Inc. Utilizado en los dispositivos como relojes inteligentes, televisores, tabletas, entre otros. Este es exclusivo de los productos vendidos por esta empresa.[15]

7 Contexto del proyecto

7.1 Modelo de ciclo de vida

Un modelo de ciclo de vida es una estructura aplicada al desarrollo de un producto de software, esto con el fin de seguir el proceder de la estructura elegida y darle forma al desarrollo del producto según las pautas de cada modelo.

El estándar internacional que regula el método de selección, implementación y monitoreo del ciclo de vida del software es ISO 12207.

El fin de estos modelos es el de mejorar la productividad, disminuir costos, tiempo, y errores, y establecer bien la carga y las tareas de cada integrante del proyecto.

Para iniciar el proyecto tuvimos en consideración los siguientes modelos de ciclo de vida:

- **SCRUM:** Tiene como características principales que identifica los roles de cada uno de los miembros del equipo y define su responsabilidad en el proyecto (el quién y el qué) con 3 roles principales: El Product Owner/Dueño del producto es la “voz del cliente” y el responsable de desarrollar, mantener y priorizar las tareas en el *backlog*. El Scrum Master es responsable de asegurarse que el trabajo del equipo vaya bien siguiendo las bases de Scrum. Además, se encarga de remover cualquier obstáculo que pueda encontrar el equipo de desarrollo. Los Development Team Members/Miembros del Equipo de desarrollo son los encargados de escribir y probar el código.

Tiene sprints, esta es la unidad básica de trabajo para un equipo Scrum. Es una simple iteración llevada a cabo por los miembros del equipo (el dónde y el cuándo). Un Sprint inicia con un equipo que se compromete a realizar el trabajo y finaliza con la demostración de un entregable. El tiempo mínimo para un Sprint es de una semana y el máximo es de 4 semanas. Dentro del

sprint hay unos “eventos scrum” que son: planeación del sprint, reunión de equipo scrum, refinamiento del backlog, y revisión del sprint.

Finalmente, Scrum tiene herramientas particulares utilizadas en su metodología, estas son: El backlog de producto, las historias de usuario, el backlog del sprint, el panel de tareas y la definición de listo. [10]

- **RUP:** Tiene como característica que divide el proceso en 4 fases dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la base de la arquitectura, abarcan más los flujos de trabajo de requisitos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la base de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones

Y finalmente en la fase de Transición ocurre el refinamiento de los objetivos de las fases anteriores e implantación del sistema de información (preparación del producto para su entrega y pasos a producción de versiones no finales). [11]

De esta forma el reparto de esfuerzos entre actividades varía de una fase a otra:

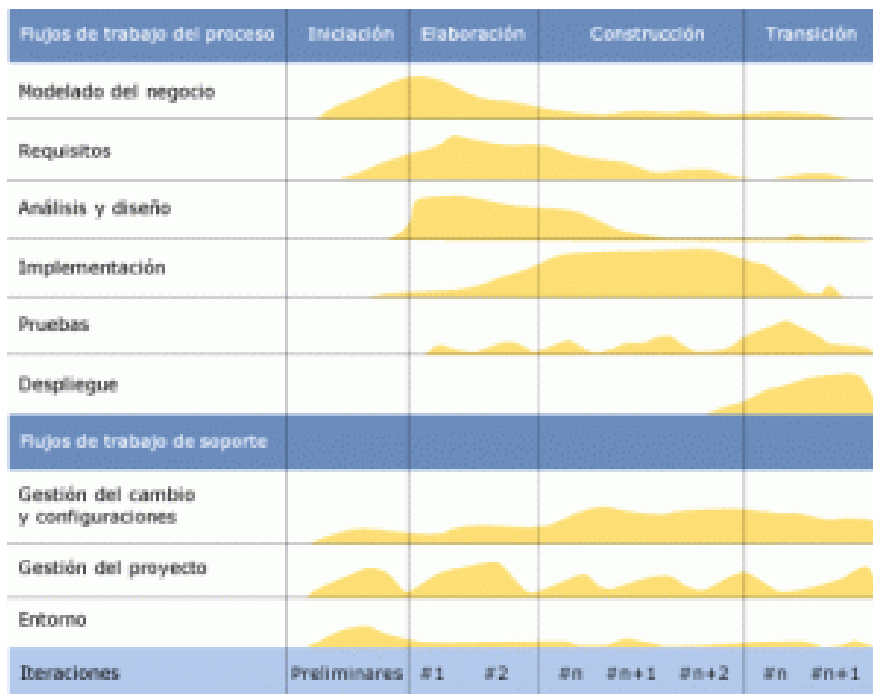


Figura 1. Variación actividades modelo RUP [11]

Tras conocer las características principales de estos dos modelos de ciclo de vida, decidimos que al tener la necesidad de desarrollar un producto de software de alta calidad en un tiempo bastante corto y contar con un cliente real, tomaríamos como referencia para el desarrollo de nuestro proyecto, algunos aspectos del modelo SCRUM y algunos del modelo RUP.

Concluimos que es la mejor alternativa para nosotros puesto que no tenemos demasiado tiempo para realizar los pasos completos del planteamiento del proyecto y de las ideas del SCRUM y tampoco para realizar el número de iteraciones correspondientes a las 4 fases del RUP.

Del RUP compartimos que:

Realizamos las estimaciones guiados por casos de uso, desarrollamos elementos de alto riesgo y alto valor en un periodo de tiempo corto, usamos componentes, antes de sentarnos a ejecutar, planeamos y pensamos objetivos y bases del proyecto, establecemos algunas iteraciones para identificar calidad y hacer seguimiento a procesos a través de una reunión semanal, asignamos tareas y responsabilidades a cada integrante del equipo de trabajo, nos guiamos por las buenas prácticas e identificamos historias de usuario y tareas medibles.

Del SCRUM compartimos que:

Manejamos Trello realizando una estimación afinada, y tenemos una lista objetivos y tareas pendientes planteadas en un Backlog. Manejamos el Burn Down Chart que es una representación gráfica del trabajo por hacer en un proyecto en el tiempo, tenemos un cliente que realiza algunas de las funciones características del Product Owner en la jerarquía scrum, este se encarga de asegurar que el equipo aporte valor al negocio y representa la parte interesada en el producto, existe una persona que hace de las veces de Scrum Master que se encarga de supervisar y revisar el proceso del equipo en cuanto al desarrollo del proceso.

La imagen que mejor describe la manera en la que se ejecutan los procesos en nuestro equipo es la siguiente:

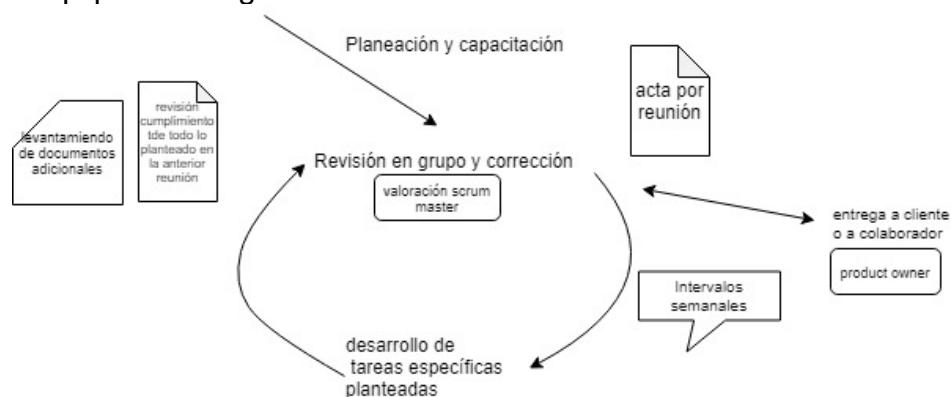


Figura 2. Modelo de ciclo de vida adaptado

Cabe resaltar que dónde se muestra el levantamiento de otros documentos, estos se realizan en cada reunión, se acuerdan en equipo y hacen parte de los documentos o las herramientas antes mencionadas en la descripción de nuestra metodología.

7.2 Lenguajes y herramientas

Para seleccionar el lenguaje de programación y el framework de trabajo que el grupo podría usar para desarrollar la aplicación, se hizo una investigación rápida con el fin de mostrar posibles opciones. Gracias a la experiencia de los semestres anteriores, surgió una primera lista de lenguajes, entre los cuales están:

- Python
- PHP
- Ruby
- Java
- Css
- C++
- C
- Angular
- Django
- Bootstrap
- Ionic
- Mongo
- Android Studio
- Laravel
- Phoenix

Dados los antecedentes de los miembros de la organización, que para efectos prácticos se puede decir que ninguno en este caso ha programado con una herramienta fuera de Java, Python, C y C++ y además, que nadie había hecho un proyecto de las dimensiones que estamos trabajando ahora, se priorizaron y categorizaron los siguientes criterios para escoger las herramientas que requerimos para desarrollar nuestra aplicación:

- El framework debe permitir el desarrollo multiplataforma, es decir, que la aplicación pueda utilizarse con cualquier sistema operativo móvil y no se vea encasillada en uno solo.
- El lenguaje debe ser simple y de aprendizaje sencillo, para que cualquiera de los integrantes pueda aprender a usar esta herramienta sin dificultad.
- La herramienta debe tener tutoriales y/o cursos virtuales para formalizar el estudio de la misma y brindar muchas más herramientas para que los integrantes puedan desarrollar el proyecto de una mejor manera
- Debe ser un software conocido, con buenos manuales de uso y que tenga soporte en este momento. Con el fin de garantizar que el software es de buena calidad y sea sencillo encontrar soluciones en línea a posibles dificultades más adelante.

- La licencia para el uso del framework debe ser gratuita, para evitar gastos en el desarrollo. Esto porque el proyecto dura únicamente el tiempo del semestre y no vale la pena pagar una cantidad de dinero importante por solo unos meses de uso.

Basándonos en los criterios establecidos previamente, el equipo de SauveGuard decidió escoger los siguientes programas para desarrollar la aplicación:

- **Ionic:** Ionic Framework es un set de herramientas de interfaz de usuario (Ui Toolkit) de código abierto para la construcción de aplicaciones móviles y de escritorio de alta calidad utilizando tecnologías web, tales como HTML, CSS y JavaScript. Está enfocado sobre todo el desarrollo del frontend para lograr una mejor interacción con el usuario. Es sencillo de aprender y se puede integrar fácilmente con otras librerías o frameworks, como Angular, o puede mantenerse por sí misma sin el uso del framework usando simplemente incluyendo script [3]. Seleccionamos esta herramienta ya que mirando diferentes ejemplos en internet de aplicaciones hechas, todas cumplen con las expectativas que teníamos sobre un framework para el frontend. Además, fue posible conseguir un curso en Udemy sobre el manejo y uso de esta aplicación para algunos de nuestros desarrolladores y según testimonios, es una herramienta en la que todos podemos entrenarnos para cumplir con las metas esperadas en el proyecto, además de ser un software gratis y con muchísimas herramientas para cumplir con los objetivos.
- **Angular:** Angular es una plataforma y framework para crear aplicaciones de cliente en HTML y TypeScript creada por Google. Facilita a su vez el desarrollo de aplicaciones web SPA, o mejor dicho, permite que las apps creadas con Angular sean reactivas y no tengan que recargar el navegador para cargar la información necesaria, demostrando una fluidez y velocidad en el uso de la aplicación [4]. Seleccionamos Angular gracias a su posible integración con Ionic en el desarrollo del front y del backend, además de ser un software gratuito y que nos permitirá suplir las necesidades de velocidad en la comunicación entre el cliente y el servidor en la aplicación SOS ciclistas, gracias a su fluidez en las llamadas asíncronas al servidor.
- **MongoDB:** Es una base de datos diseñada para facilitar el desarrollo y escalamiento de las empresas y/o proyectos desarrollados. Su alto rendimiento se basa en la posibilidad de tener documentos con información anidada, para evitar el uso elevado de operaciones Entrada/Salida, y su soporte para índices, con la capacidad de crearlos sobre documentos y arrays[5]. Gracias a la necesidad de tener un sistema de registro y validación de datos, así como atributos propios de cada usuario, se decidió utilizar esta herramienta como la base de datos en la que la aplicación sostendrá el módulo de registro y la información de cada persona ingresada.

7.3 Plan de aceptación del producto

"Software Project Management Plan" (SPMP) entrega 1:

- Excelente presentación y ortografía
- Impecabilidad en la redacción
- Formación concreta de proyecto, jerarquía, tareas asignadas, estimaciones, diseños, tiempos estipulados.
- Cumplimiento al pie de la letra de la plantilla sugerida
- Modelos de proceder y para entender la organización sólidos y completos.

Botones y pantallas entrega interna:

- Documentación completa del código (Compodoc)
- Estética de acuerdo a las especificaciones dadas
- Manejo correcto de la información
- Entrega requerida completa
- Sin errores

“Software Requirements Specification” (SRS) entrega 2:

- Corrección completa de particularidades o errores del anterior documento
- Descripción concreta y concisa del comportamiento del sistema que se va a desarrollar
- Diagrama de casos de uso detallado y revisado
- Levantamiento de requisitos completos
- Excelente presentación, ortografía y redacción del documento
- Caso de uso o funcionalidad principal de la aplicación funcionando en su totalidad, sin errores ni defectos.

Funcionalidad principal al cliente, primera entrega oficial:

- Caso de uso o funcionalidad principal de la aplicación funcionando en su totalidad, sin errores ni defectos.
- Avances en diseño de lo restante de la aplicación
- Pantallas para la aplicación

Casos de uso adicionales entrega interna:

- Documentación completa del código (Compodoc)
- Estética de acuerdo a las especificaciones dadas
- Manejo correcto de la información
- Entrega requerida completa
- Sin errores

“Software Design Document ” (SDD), tercera entrega:

- Guía detallada por medio de palabras de la arquitectura de software característica de la aplicación
- Excelente presentación, ortografía y redacción del documento
- Corrección completa de particularidades o errores del anterior documento
- Aplicación completamente funcional, sin fallos, y con todos los casos de uso planteados desde el principio completamente implementados

Aplicación para prueba entregada al cliente:

- Documentos de diseño finalizados
- Aplicación completamente funcional para realizar pruebas y posibles correcciones

Entrega final al cliente:

- Aplicación con correcciones realizadas, con todos los casos de uso y recomendaciones implementadas
- Manejo de concurrencia y respuesta en tiempo real funcionales
- Manual de uso para capacitación

Utilizaremos nuestro calendario para identificar las actividades ya realizadas y listas para entrega, así como el tablero de tareas por hacer dado el tiempo, para definir la calidad de los documentos y el producto antes de cada entrega.

También para verificar calidad del documento en cada reunión anterior a la entrega se realizarán revisiones con todo el equipo de trabajo, así como se le pedirá a una persona experta en el lenguaje y en la escritura para que nos de la retroalimentación y nos hable de posibles errores pasados por alto.

En cuanto a la claridad de los documentos, diagramas y diseños, se les preguntará a personas ajenas al comportamiento interno que den una opinión acerca de estos para saber si se quiso dar a entender la idea y se les pedirá a personas expertas que nos retroalimenten con respecto a la posibilidad de lo planteado y las buenas prácticas de diseño en todo lo realizado.

Finalmente, para verificar funcionalidad total de la aplicación se evaluará lo planteado al inicio puesto en práctica, y la manera en la que funciona cada caso de uso por medio de pruebas, y se mantendrá el código abierto a posibles verificaciones y cambios, por eso debe ser escalable y la documentación debe ser estable y completa.

7.4 Organización del proyecto y comunicación

Entidades interesadas:

Nombre	Descripción	Responsabilidades	Medio de comunicación
Angélica Maria Sanchez Rendón representante SOS Ciclistas	SOS ciclistas es una entidad sin ánimo de lucro encargada de la gestión de ambulancias por medio de una empresa privada, unida a la Liga de	Cliente o Product Owner Encargada de recopilar información de su organización en cuanto a requisitos y peticiones para llevarlo a la	-Mensajes de texto por WhatsApp -Reuniones por skype -Llamadas -Correo electrónico:

	Ciclistas Urbanos y representada por Angélica al mando de otros varios administradores.	realidad por medio de la aplicación. También se encarga de ser el puente directo de comunicación entre ambas partes.	S.o.s.ciclistas.bogota@gmail.com
Biciusuarios	Usuarios finales o público objetivo para la utilización de la aplicación, personas que se espera utilicen el servicio	Interactuar con la aplicación y exponer necesidades para mejorar efectividad y eficiencia	-Charlas en persona
Carlos Andrés Parra Acevedo Profesor	A quien se le harán entregas parciales y quien revisará el proceso semanal	Persona encargada de brindar herramientas para el desarrollo del proyecto, el monitoreo del mismo, y brindar una retroalimentación y una calificación cuantitativa del trabajo realizado	-Reuniones semanales en horario de clase -Correo

Tabla 2. Entidades interesadas

Descripción jerarquía organización:

Nombre	Rol	Descripción	Tareas
Sofia Moreno	Gerente	Puente de comunicación directa con el cliente y quien tiene conocimiento general de todo lo que sucede dentro del proyecto	-Comunicación con cliente -Supervisión de tareas por integrante -Encargada de recibir correcciones y realizar actas por reunión -Programar -Diseñar -Documentar

			-Realizar documento
Kevin Nimisica	Arquitecto	Encargado de diseños de diagramas de modo de proceder y de arquitectura en general del producto	-Encargado de los diseños -Programar -Documentar -Realizar documento -Aportar para el desarrollo del producto
Sebastián Valderrama	Jefe de desarrollo	Encargado de capacitarse para la programación fuerte y orientar a los demás en el desarrollo	-Capacitarse -Guiar -Brindar soluciones para el desarrollo -Programar. -Documentar -Realizar documento -Aportar para el desarrollo del producto
Juan Diego Osorio	Administrador de documentación y configuración	Encargado de la investigación acerca de herramientas necesarias y la escritura y redacción de documentos	-Investigar -Realizar documento y revisar redacción y ortografía -Aportar a los documentos -Programar -Documentar -Aportar para el desarrollo del producto
Federico torres	Director de calidad y manejo de riesgos	Encargado de parte de la programación fuerte y las pruebas	-Capacitarse -Programar -Comprobar requisitos de calidad -Documentar -Aportar a los documentos

			-Aportar para el desarrollo del producto -Probar
--	--	--	---

Tabla 3. Jerarquía organización

Organigrama:

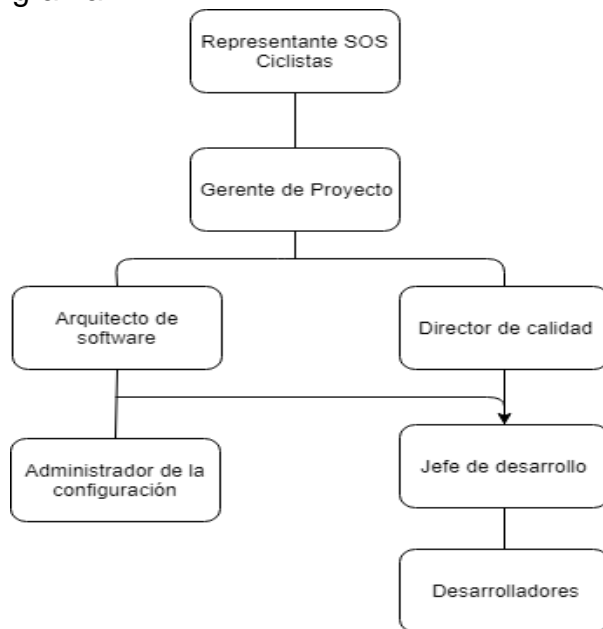


Figura 3. Organigrama

8 Administración del proyecto

8.1 Métodos y herramientas de Estimación

Estimación de la duración del proyecto

Existe una relación entre la complejidad de cada caso de uso y las líneas de código necesarias para implementarlo en un proyecto de software. Por esta razón se decidió usar los puntos de casos de uso como herramienta de estimación de la duración de implementación del proyecto. El objetivo es que entre todos los miembros del equipo se puedan atribuir puntos a cada caso de uso basándose en una escala numérica previamente establecida entre todos los desarrolladores involucrados, teniendo en cuenta los principales criterios que permiten evaluar el tiempo que demanda cada uno. Los criterios son el número y la complejidad de cada caso de uso en el sistema, así como los actores involucrados. Además, se tendrá en cuenta los requerimientos no funcionales que demande cada caso de uso como

puede ser portabilidad, desempeño, concurrencia, mantenimiento y se tendrá también en cuenta el ambiente de desarrollo donde el proyecto se llevará a cabo, como es el lenguaje de programación, el nivel de experiencia de los desarrolladores, la motivación y cohesión del equipo.

Para llevar a cabo esta métrica, fue necesario elaborar entre todos un resumen general para cada caso de uso, diseñando los correspondientes diagramas de actores y todos los subsistemas implicados. Luego, en las reuniones semanales que se llevaron a cabo los miembros del equipo explicaron el funcionamiento de cada caso de uso y cada integrante estableció un puntaje para uno basado en su opinión personal de los criterios previamente mencionados. Posteriormente, se desarrolló un debate en el que cada integrante ofrece una explicación sobre la puntuación establecida y le da a los demás participantes una retroalimentación que permite contemplar los diferentes puntos de vista para finalmente volver a votar y establecer un promedio de puntos para el caso de uso en cuestión.

En nuestro proyecto de software se establecieron 10 casos de uso simples, 4 intermedios y 0 complejos, lo cual establece 96 puntos para el UUCW (Unadjusted Use Case Weight) una vez ponderados con su correspondiente peso como se puede apreciar en la hoja de cálculo adjunta en los anexos. En el caso del UAW (Unadjusted Actor Weight) se estableció que la aplicación no tendrá una API que le permitirá a la aplicación establecer comunicación con terceros porque en principio la aplicación no enviará ni recibirá información de agente externos. Su función es informar al administrador de la aplicación de los sucesos de los bicisuarios. En el nivel intermedio o promedio se encuentran casos de uso como el protocolo de comunicación que nuevamente en nuestro proyecto no tendrá asignado puntos porque el manejo de protocolos de comunicación resulta ser a un nivel básico hasta el punto en el cual en gran medida será la aplicación la encargada implementarlo mediante sus librerías asociadas. Finalmente tenemos la interfaz gráfica que será fundamental en nuestro producto porque deberá garantizar un fácil acceso y operabilidad e caso de emergencias. La puntuación establecida fue de 2, en parte gracias a las utilidades que ofrece el framework de desarrollo ionic, que luego de ser ponderada arroja una puntuación de 6.

Esta misma metodología se usó para establecer puntos para calcular el TCF (Technical Complexity Factor) que mide la complejidad técnica de las principales características del producto. Este resultado fue de 1.005 dejando en evidencia la importancia que tiene las características móviles en el producto, así como también que esté basado en sistemas distribuidos para garantizar el correcto desempeño. Finalmente se evalúa el EF (Environmental Factor) los factores ambientales del desarrollo del proyecto usando las referencias utilizadas en la mayoría de los proyectos de desarrollo de software, arrojando un resultado de 0.7775 resaltando la importancia de tener una buena hoja de requisitos procedente del cliente real que está solicitando la aplicación, así como la motivación del equipo.

Finalmente, se usan estos indicadores para calcular el UCP (Use Case Points) del proyecto. La fórmula es:

$$UCP = UUCW * TCF * EF$$

El resultado arrojado fue de 70.3248 y haciendo una equivalencia con que cada punto son 23 horas de trabajo, se llegó a la conclusión de que serían 1725.30 horas de trabajo, que deberán ser repartidas entre los miembros del equipo de desarrollo. Dado que son 5 integrantes y que en promedio cada uno destinará 30 horas semanales de trabajo se puede afirmar que en total el equipo destinará 150 horas. Teniendo en cuenta además que serán iteraciones de 3 semanas, la duración total será de 3.59 iteraciones con 450 horas totales por iteración.

Para ampliar la información ver en carpeta: Anexo3-Diagrama_CU, Anexo2-Estimaciones.

8.2 Inicio del proyecto

Para poder desarrollar el proyecto de la manera que se espera, es necesario que cada integrante, haciendo uso de sus computadores personales y los recursos brindados por la universidad (portátiles y cubículos) tenga un buen manejo y uso de ofimática para la creación de la documentación respectiva en cada entrega (Uso de Word, Project, Excel, etc.). Las habilidades para desarrollar en este punto son prácticamente nulas, ya que el uso de estas herramientas ya se ha aprendido a lo largo de la carrera. Sin embargo, para desarrollo de la aplicación es necesario la generación de un plan de enseñanza para que cada uno de los integrantes tenga por lo menos un mínimo conocimiento en los Frameworks que se van a utilizar (Angular, Ionic). Por este motivo, es necesario que algunos de los integrantes tengan una formación un poco más a profundidad en estos campos para que sean los líderes encargados en la programación, mientras el resto del equipo servirá como base de apoyo para tareas más sencillas y al tiempo estarán mucho más enfocados en la parte de la documentación, permitiendo una distribución uniforme del trabajo y evitando sobrecargas entre los miembros del equipo. Por consiguiente, el plan de capacitación es el siguiente:

Integrante	Habilidad/Conocimiento	Quién lo enseñará?	Donde lo aprenderá?
Sebastián Valderrama	Aprender Angular formalmente Aprender Ionic formalmente	Udemy	Udemy[1]
Juan Diego Osorio	Aprender Angular y Ionic de manera personal	Tutoriales Manuales de uso	Youtube[2]
	Aprender Angular formalmente		

Federico Torres	Aprender Ionic formalmente	Udemy	Udemy[1]
Kevin Peláez	Aprender Angular y Ionic de manera personal	Tutoriales Manuales de uso	Youtube[2]
Sofia Moreno	Aprender Angular y Ionic de manera personal	Tutoriales Manuales de uso	Youtube[1]

Tabla 4. Capacitación previa

En la siguiente gráfica se podrá apreciar el plan de distribución de enseñanza con los contenidos que se aprenderán en cada uno de los tiempos establecidos:

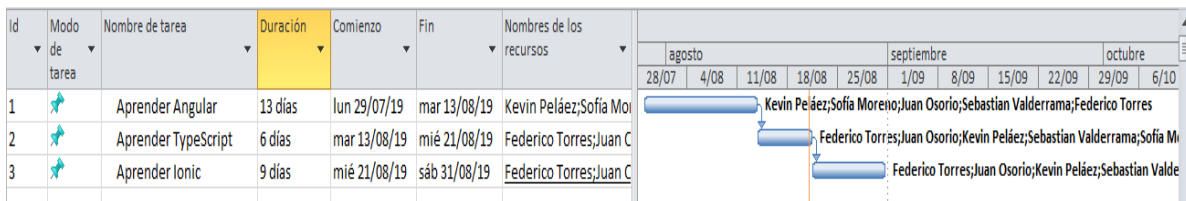


Figura 4. Diagrama general capacitación inicial

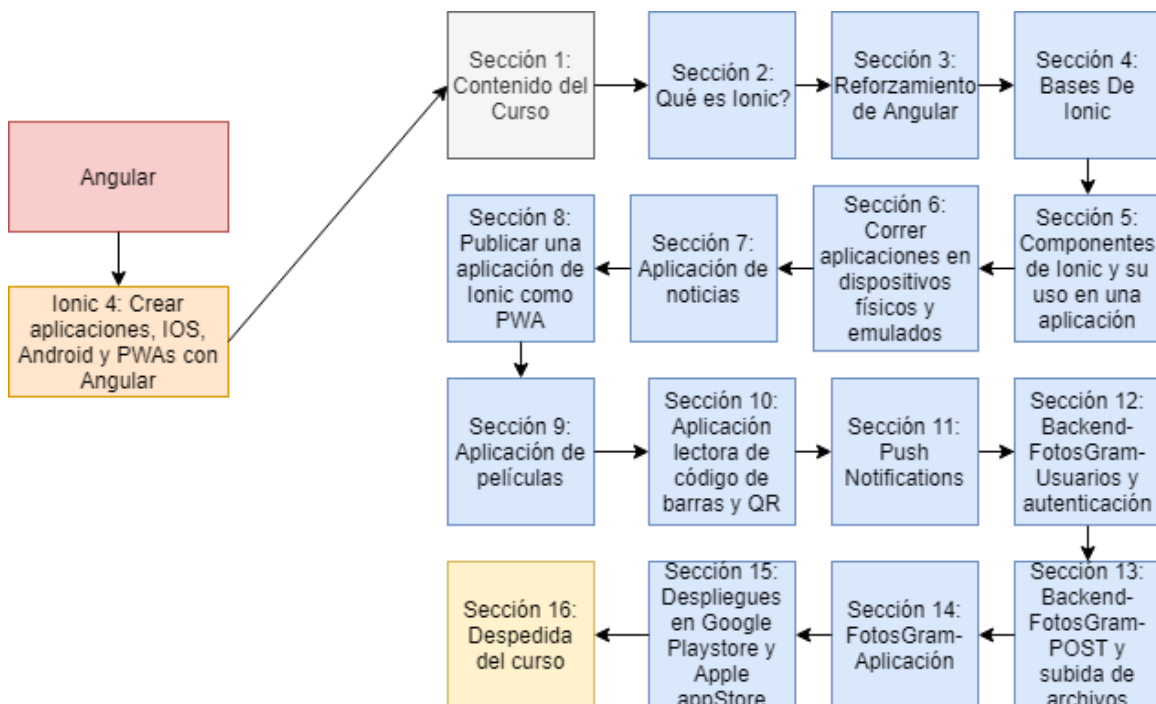


Figura 5. Módulos de aprendizaje de Ionic

8.3 Planes de trabajo del proyecto

Estimación del costo del proyecto

Para calcular el costo total del proyecto, fue necesario indicar los principales datos de entrada que impactan directamente en los costos como son el sueldo básico de los ingenieros, las prestaciones laborales de ley, los costos de transporte, almuerzo y antojos. A eso toca incluir además el costo del servicio en la nube para almacenar los datos de la aplicación, el servicio de capacitación de los ingenieros en el nuevo framework de Ionic y el registro mercantil para constituirnos como una empresa de desarrollo legalmente constituida. Finalmente, se suma el costo de impuestos que eventualmente tocará pagar al gobierno como resultado de los beneficios. Los datos de entrada se suministran con su correspondiente periodo de desembolso que puede ser mensual, bimensual, semestral o anual dependiendo el caso y luego al momento de elaborar el presupuesto de gastos mensuales se unifican los periodos de pago. El resultado obtenido para el presupuesto de gastos mensuales asciende a COP\$20.806.190, incluyendo todos los gastos mencionados anteriormente. Luego al ajustar el costo mensual de desarrollo con el tiempo estimado de duración de implementación del proyecto que es de 2.69 meses, se obtiene que el costo total del proyecto de ingeniería de software asciende a COP\$56.089.054 que es aproximadamente US\$17.473 con la actual tasa de cambio vigente en casas de cambio: US\$1 = COP\$3.210. Es importante aclarar que la mayor parte de los costos de desarrollo es la correspondiente al sueldo de los 5 ingenieros involucrados en la empresa que asciende a COP\$2.500.000 mensuales más prestaciones laborales.

Para ver en detalle los resultados del proceso de estimación de la duración del proyecto como de la estimación de costos recomendamos ver hoja de cálculo anexa al final del documento.

Para ampliar información ver en carpeta: Anexo2-Estimaciones

Work Breakdown Structure (WBS):

Se usa esta herramienta de modelado de tareas para mostrar lo que se debe hacer con respecto a cada entrega que se presentará ante el instructor y ante el cliente:

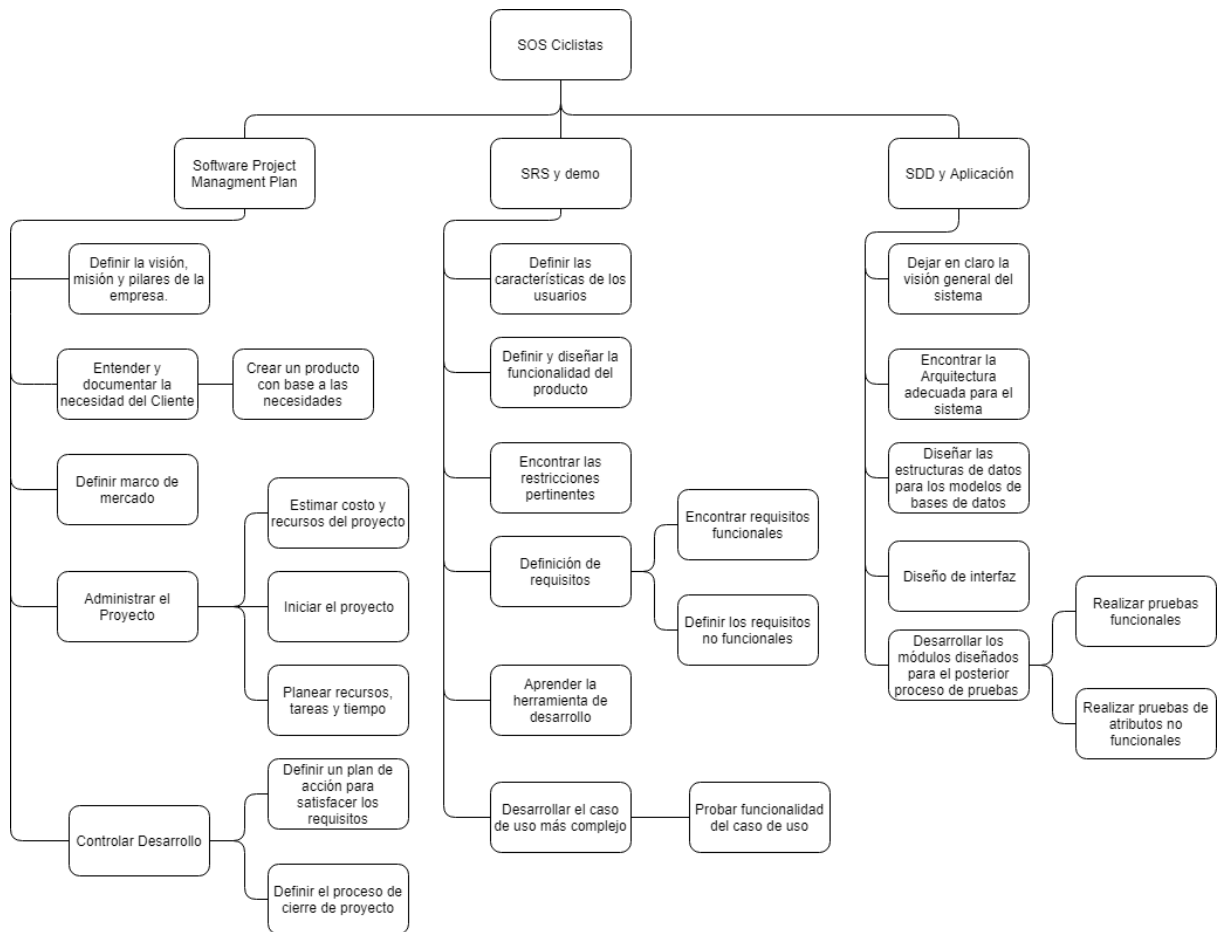


Figura 6. Work Breakdown Structure.

El anterior diagrama se encargó de mostrar las tareas llevadas a cabo frente al instructor y cliente, a continuación, se presenta el correspondiente a las actividades que se llevarán a cabo para el desarrollo de la aplicación y su funcionalidad más específicamente.

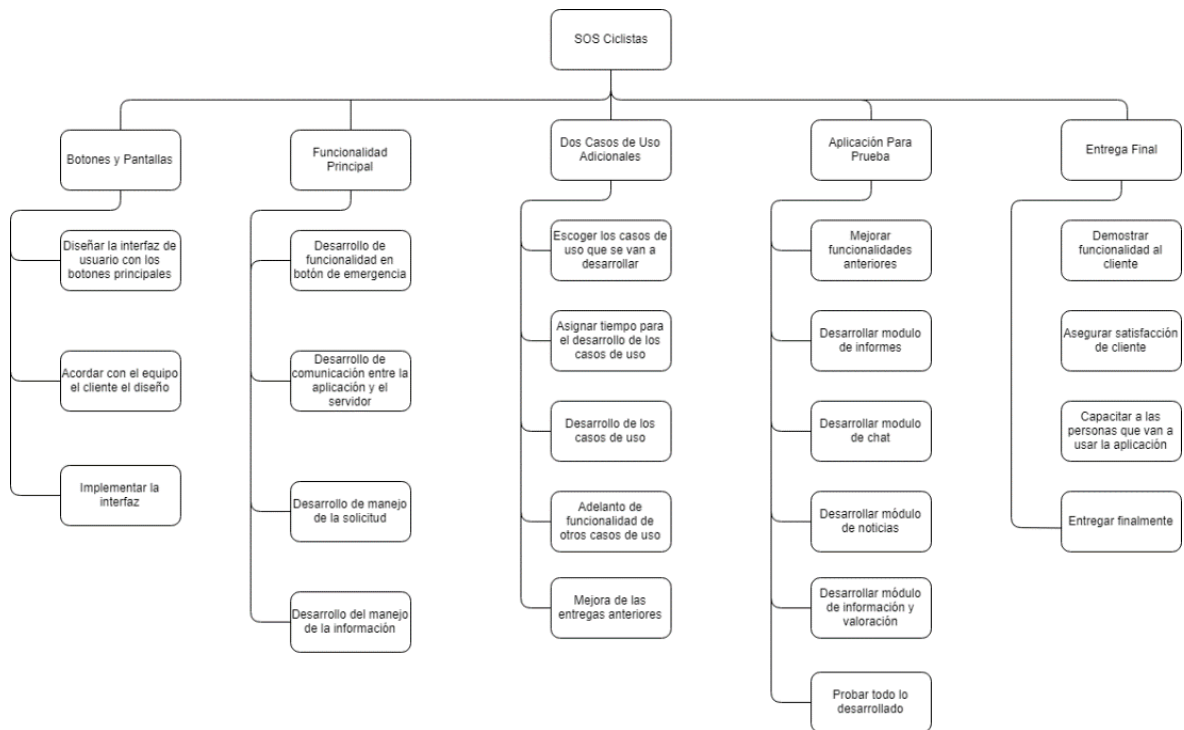


Figura 7. Work Breakdown Structure sobre Funcionalidad

Para ampliar información ver en carpeta: Anexo4-Diagramas_Grantt

9 Monitoreo y control del proyecto

9.1 Administración de requisitos

Modelo para Cambios:

Para llevar a cabo un cambio en el producto de software o un cambio en los requisitos del mismo, este cambio es detectado por el cliente, el cual debe avisar al gerente de proyecto sobre el cambio explicándole todo lo relacionado con él. El gerente se encarga de ver que este cambio en realidad sea necesario con respecto a los intereses el cliente, como por ejemplo que el cambio no sea algo que ya está dentro de los módulos de desarrollo, que este cambio es una funcionalidad que se le quiere agregar o un atributo que se quiera favorecer, pasado esto si el gerente ve el cambio como no necesario le comunica al cliente las razones, si este queda satisfecho, se acaba, en caso contrario puede apelar al gerente; en caso de ser un cambio necesario, se documenta a profundidad los datos del cambio, lo que es entregado al arquitecto, este se encarga de diseñar de manera entendible para el equipo desarrollador y mediante este diseño evitar interferencias con lo desarrollado o hacer que el impacto sea menor. Este diseño se agrega a la cola de desarrollo, que posteriormente será desarrollado por el equipo para ser probado y entregado.

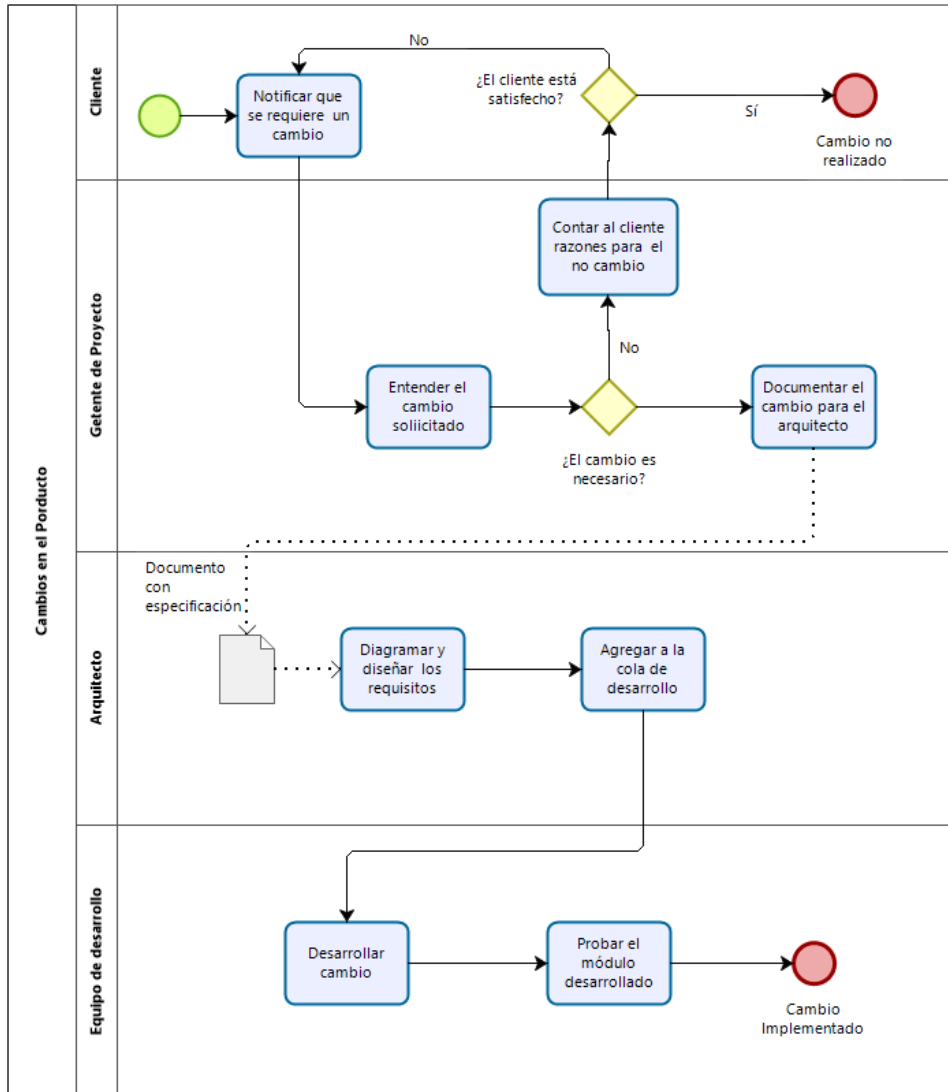


Figura 8. Modelo para cambios.

9.2 Monitoreo y control del progreso

Medidas utilizadas para medir el progreso

- **Horas laboradas:** Gracias a las estimaciones hechas según los casos de uso y la cantidad de horas que según esta estimación serán requeridas para el desarrollo del proyecto, una de las medidas utilizadas para seguir el progreso es la cantidad de horas laboradas por cada uno de los miembros de SauveGuard. A pesar de que esta medida no considera ni posibles errores que pueden presentarse en el código o posibles retrasos en las entregas, nos

servirá como una guía para poder llevar un control del progreso sobre la aplicación.

- *Cantidad de casos de uso implementados:* Ya que en total se pudieron evidenciar 14 casos de uso en las estimaciones y a la vez, cada caso de uso contiene cierta cantidad de pequeñas tareas o actividades pendientes, la implementación en el producto de software de cada uno de estos casos de uso nos permitirá tener una medición acertada del proyecto. Esto se debe a que se desarrollarán en primera instancia los casos de uso más complicados y al final, se complementará el programa con los casos de uso que permiten una mayor calidad del servicio, más no que son fundamentales para el mismo.
- *Cantidad de pruebas satisfactorias:* Es necesario hacer pruebas para determinar si un caso de uso es completamente funcional y es lo suficientemente robusta para poder reaccionar ante los problemas. Se contarán las cantidades de pruebas que se le hagan al código para demostrar su correcto funcionamiento con diferentes posibilidades y respuestas diversas. Así identificaremos si un requisito tiene una implementación exitosa o es necesario que se corrijan posibles errores.

Actividades de reporte

- *Presentación de los requisitos o casos de uso:* El encargado de implementar cada requisito o caso de uso que se le asigne, deberá exponer al resto de sus compañeros en las reuniones que se hacen semanalmente, qué fue lo que hizo y realizará una serie de pruebas con el fin de que el resto de SauveGuard conozca el funcionamiento de este requisito.
- *Sesiones de corrección de documento:* Se harán reuniones vía Skype o Hangouts para verificar los documentos que hacen parte de las entregas, para evitar posibles errores. Estas reuniones se hacen generalmente en la noche y todos verificarán los ítems añadidos a los entregables, haciendo verificaciones de ortografía y redacción o llenando los vacíos que queden en el documento.

Acciones correctivas

- En caso de que el responsable de cualquiera de las tareas asignadas en el momento del desarrollo de software o del documento llegue a la reunión semanal y no haya cumplido con lo acordado, el equipo le dará un plazo máximo de una semana para completar dicha tarea. Si no la completa en este tiempo, cualquiera de los desarrolladores que haya terminado con sus responsabilidades le brindará asistencia al compañero que no pudo terminar con su labor. Si esta tarea es de importancia vital, la asistencia del grupo será inmediata y todos trabajarán en conjunto para solucionar este problema lo más rápido posible.
- Si al momento de la presentación a los administradores de los demos o micro entregas que se harán, existe algún aspecto que no le guste al product owner, se comentará esta insatisfacción en las reuniones semanales y se hará un plan de cambios de esta característica, ceñidos a los controles de cambios, siendo todos igualmente responsables por el desarrollo de este cambio.

- Si el equipo de SauveGuard se ve corto en el tiempo y debe cumplir con una tarea, todos los miembros deberán utilizar el tiempo que tengan para dedicárselo a las entregas, así sea necesario pasar la noche en vela para poder terminar cualquier aspecto del software.

9.3 Cierre del proyecto

Para poder realizar cada una de las entregas, es necesario tener en cuenta cada una de las siguientes actividades:

- *Primera Entrega:* Ya que la primera entrega únicamente es completar el documento “SPMP”. Los insumos necesarios para realizar este plan de gestión son:
 - Computador personal de cada integrante del grupo.
 - Suite de office para gráficas y documentación: (Project, Word)
 - Software especializado para creación de gráficas (Bizagi)
 - Software especializado para creación de historias y actividades por hacer (Trello, Aplicación para las tarjetas de Scrum)
 - Disponibilidad de salas de estudio para las reuniones semanales, junto con los materiales utilizados en la reunión (marcadores, notas adhesivas).

Las actividades realizadas para completar esta etapa fueron las siguientes:

- Reuniones semanales para compartir información sobre el cliente final.
 - Elaboración de estimaciones en grupo para tener un resultado óptimo.
 - Desarrollo del documento en cada reunión para adelantar posibles pendientes.
 - Practicar y aprender a usar los FrameWorks para poder desarrollar la aplicación en las próximas entregas.
 - Crear el reporte gerencial de los resultados obtenidos en la entrega y la propuesta de nuevas ideas para solucionar los inconvenientes presentados
- *Segunda Entrega:* Esta entrega consta de dos partes: El documento SRS de requerimientos y una Demo de la aplicación con la funcionalidad principal. Como esta demostración consta de las pantallas del bici-usuario, que permiten el envío de la información y la petición de los servicios de emergencia y del administrador, que conforman la lista de peticiones por asumir y el formulario enviado por el bici-usuario, los insumos requeridos fueron:
 - Los computadores de cada integrante.
 - Software especializado para Front y Backend donde se desarrolla la aplicación (Ionic y Angular).
 - Software especializado para manejo de base de datos (Mongo).
 - Suite de windows para la documentación (Word y Project).
 - Salas de estudio para las reuniones semanales.
 - Reporte gerencial en cada

Ya que en esta entrega la programación es fundamental, las actividades propias de la entrega estarán más orientadas al desarrollo del software. Estas serán:

- *Etapas post-mortem:* Debido a la implementación de un modelo de ciclo de vida muy similar al Scrum, SauveGuard priorizará el desarrollo de software en etapas pequeñas de entrega, entre las cuales se hará una división del trabajo por semanas y en la reunión que se haga para hacer seguimiento se exigirán avances en cuestión sobre la tarea asignada. Por consiguiente, se realizarán una serie de micro entregas semana a semana para poder completar el proyecto de manera satisfactoria. Una vez terminada la tarea, se asignará al integrante una nueva labor por cumplir y se hará lo mismo para cada integrante que vaya terminando sus propias labores.
- *Tercera Entrega:* La última entrega se compone del documento SDD que habla sobre todo del diseño de software, y finalmente, la aplicación completa con todas las funcionalidades descritas en las dos entregas posteriores. Para esta entrega fueron necesarios los mismos insumos que las entregas pasadas, por lo que no hace falta mencionarlo. Sin embargo, al requerir un mayor compromiso a nivel de desarrollo, no se asignarán a cada estudiante una tarea específica, sino un requerimiento por satisfacer, ya que el tiempo es corto y hay un usuario final al que se le debe entregar el producto. Por ello, se les brindará soporte una vez entregado el proyecto, se les hará una capacitación y se les hará entrega de los manuales de uso. Obviamente, en cada avance que se realice, se le mostrará al cliente final lo que se hizo para estar seguros de que los requerimientos del cliente se están cumpliendo de la manera esperada.

Para hacer seguimiento la entrega al cliente se realizará:

- Una primera entrega de las funcionalidades principales con su respectiva documentación con el fin de recibir sugerencias, enunciar pasos a seguir y mostrar avance en la parte visual y funcional de la aplicación
- Una entrega del producto finalizado con fines de prueba y de resolución de posibles conflictos y errores en ejecución
- Una semana después se entregará la versión oficial y lista para el uso y la migración de los interesados presentes en el grupo de WhatsApp a esta. Con esta entrega final viene una capacitación corta a los administradores para explicación de uso para realizar la capacitación a los usuarios por medio de un video explicativo y la entrega de documentos correspondientes a la documentación del código y los documentos de diseño, seguidos de un comprobante de avance y de calidad de lo entregado.
- Finalmente se mantendrá un contacto directo con la organización para recibir peticiones de posibles cambios y de escalamiento, puesto que es una opción muy viable y probable.
- Se hará seguimiento mensual al producto entregado para confirmar buen funcionamiento

- Y se hará un análisis detallado de posibles herramientas adicionales a añadir a la plataforma, dependiendo de las observaciones y comentarios recibidas por los usuarios o por terceras personas.

10 Entrega del producto

Al tener un cliente verdadero, la entrega del producto comprende diferentes áreas a trabajar:

- Se deberán migrar todos los usuarios del grupo de Whatsapp “SOS ciclistas” a la aplicación desarrollada. Esta migración será posible gracias a la distribución de la app por medio del WhastApp, brindándoles el link de descarga.
- Para entender el correcto funcionamiento de la aplicación, se les brindará a los administradores del grupo la documentación hecha a lo largo del semestre para que tengan un soporte del software por si deciden hacerle alguna mejora en el futuro. Además, se les brindará a todos los usuarios de la aplicación un manual de uso para que sea más sencillo manejar la aplicación.
- También se creará un video explicativo para manejar la aplicación, ya que es mucho más probable que los usuarios vean este tipo de contenido que ver un manual. Esto con el fin de que todos aprendan fácilmente como hacer un correcto uso de la aplicación.
- Para las entregas en la universidad, todo el equipo presentará los documentos requeridos por entrega y los dos demos funcionales en las fechas acordadas en el plan de aceptación. Estos correrán en dos dispositivos móviles de diferente sistema operativo, para garantizar al usuario que no deba regirse únicamente a un dispositivo único.

11 Procesos de soporte

11.1 Ambiente de trabajo

Para lograr tener un ambiente de trabajo exitoso y agradable en el cual todos los miembros de SauveGuard puedan trabajar a gusto y si alguno comete una falta, sea justamente penalizado, se definieron las siguientes normas dentro de la compañía:

- *Reportes:* Para poder tener una alta trazabilidad en el desarrollo del proyecto y poder realizar un seguimiento semana tras semana, SauveGuard utiliza una ficha de reportes de cada una de las reuniones, con el fin de poder evaluar cuál ha sido el avance de cada uno de los integrantes según las tareas que quedaron asignadas desde la sección anterior, poder documentar lo que se hizo y se habló en cada una de las sesiones y dejar repartidas las tareas que

quedan pendientes para la próxima sesión. La persona encargada de la generación de estos reportes es Sofia Moreno y quedarán registrados en la carpeta del grupo junto con los otros.

- *Hablar de todo:* Es necesario que todos los integrantes de SauveGuard comenten y le cuenten a los demás de su propio proceso, de si está cumpliendo con las expectativas y las labores asignadas y que si necesita ayuda en algún punto pueda comentarlo lo antes posible, de esta manera no quedar a destiempo en ninguna de las entregas y que todos como equipo puedan solucionar los problemas existentes en cualquier momento o etapa del proyecto.
- *Llegadas puntuales:* Es de total importancia que todos los miembros lleguen temprano o puntualmente a las reuniones establecidas y que si en algún momento llega a existir un percance por el cual no pueda asistir o que su llegada tarde se vea justificada, lo exprese con anterioridad a las reuniones. Esto se debe a que es fundamental que todos los integrantes del grupo estén informados de todo lo correspondiente al proyecto y que todos sepan cómo va el desarrollo de este. Todo esto es posible en las reuniones semanales, y, por consiguiente, si alguien llega a faltar, existe un gran riesgo de que la persona se pierda de mucha información importante para todos los integrantes
- *Multas:* Se cobrará una sanción económica a la persona que cometa alguna de las siguientes infracciones:
 - *Llegar tarde a las reuniones:* El máximo que se esperará a un integrante en una reunión será de 10 minutos (hora local) desde que se entra en la sala de estudio. De no cumplirse deberá pagar \$1000 pesos inicialmente, y otros \$1000 por cada 10 minutos más de retraso los cuales se utilizarán al final del semestre para la actividad de cierre del proyecto.
 - *No decir la verdad en las reuniones y no cumplir tareas:* Si alguno de los miembros de SauveGuard no dice toda la verdad sobre sus avances o la cuenta parcialmente, omitiendo información relevante a los demás compañeros, o no termina las actividades y tareas que se le habían asignado en la reunión pasada, deberá pagar una porción de pizza o una empanada (taco ranchero o equivalente al valor de la porción de pizza) a cada integrante del grupo. Puesto que esta falta es perjudicial para todos y no cumplir con sus deberes debe traer consecuencias.
 - *El uso del celular:* Los integrantes que estén utilizando mucho más de la cuenta el celular, no con un uso netamente académico en las reuniones que se desarrollan deberá pagar \$500 pesos, que a su vez serán utilizados al final de semestre para la actividad de cierre de proyecto.

Estas multas serán recogidas por el tesorero del grupo que será Sebastián Valderrama, el cual tendrá la obligación de pedir el pago de las respectivas sanciones y los guardará de la forma que él desee. Sin embargo, en el cierre

del proyecto, debe traer todo el dinero recogido para el desarrollo de la actividad de cierre.

- *Escuchar al otro y toma de palabra:* La comunicación es clave para poder desarrollar el proyecto de manera eficiente, por ello se tiene como norma en la empresa escuchar y no interrumpir a la persona que esté hablando en las reuniones. La opinión de todos es necesaria para llegar a un consenso y siempre aportar las mejores ideas en el desarrollo del proyecto. Cualquiera puede tomar la palabra si no está interrumpiendo a alguien
- *Retroalimentación entre todos:* Por cada elemento agregado tanto al proyecto como al resto de los documentos se hará una retroalimentación por parte de todos los integrantes de SauveGuard para corregir errores. La persona que haya hecho esta parte deberá escuchar al resto de los participantes de la reunión para hacer los cambios respectivos en el proyecto, siempre y cuando se hagan con respeto y de buena manera los comentarios, para evitar problemas entre los compañeros

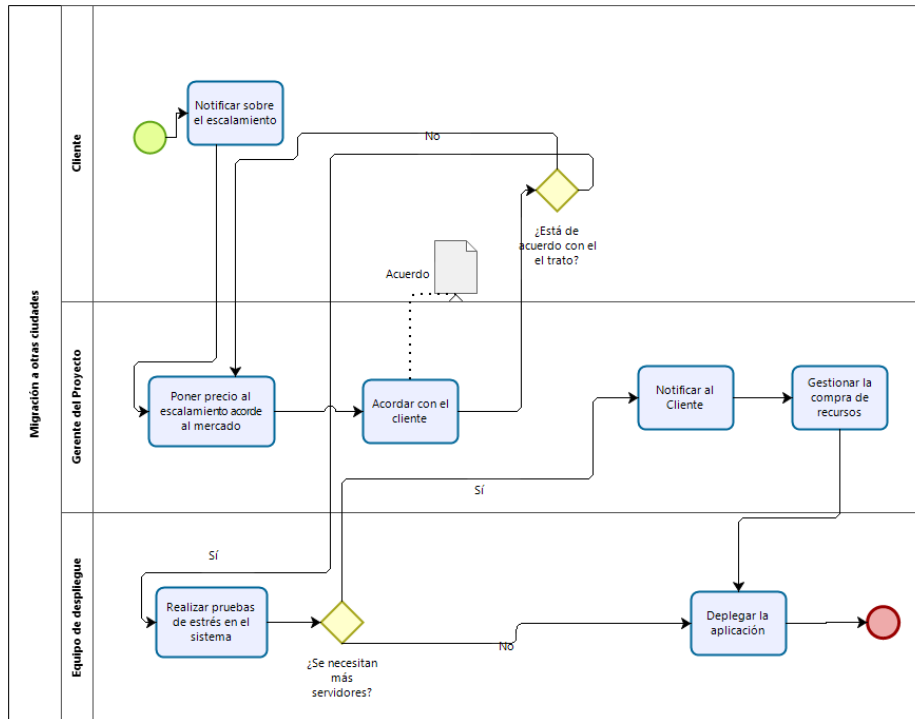
11.2 Análisis y administración de riesgos

Para el análisis de riesgos se tomaron los riesgos positivos y negativos, estos son:

1. La aplicación puede escalar a nivel de funcionar en varias ciudades.
2. Durante el desarrollo del proyecto puede haber deserciones en el equipo.
3. Es posible presentar retrasos en el desarrollo del proyecto.
4. Conflictos entre los integrantes que dañan el ambiente.

Manejo de riesgos:

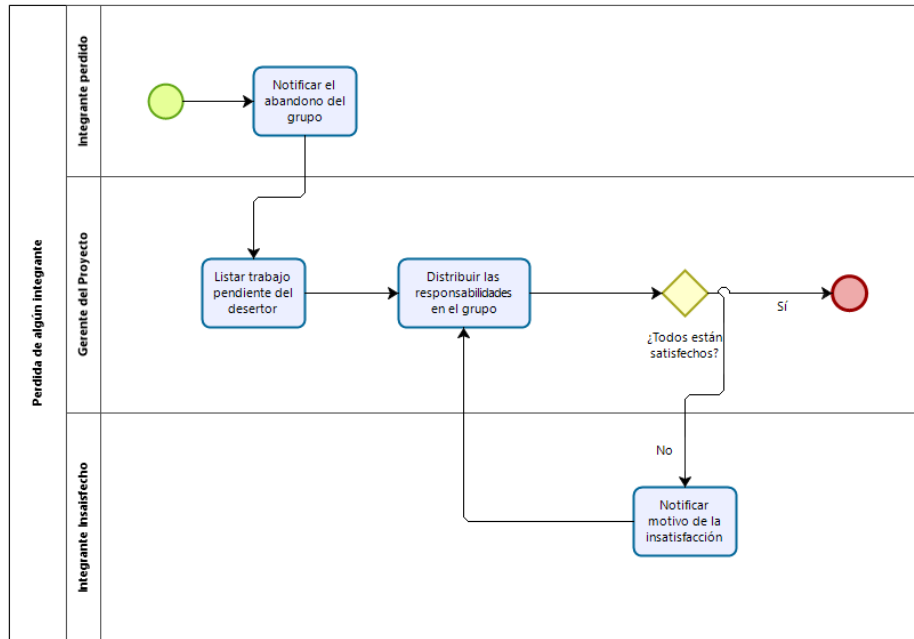
1. *Escalamiento:* Este proceso se inicia por parte del cliente debido a la necesidad de despliegue desde otras ciudades, el cliente notifica al gerente del proyecto, el cual se encarga de poner un precio adecuado con respecto al mercado y realiza una negociación con el cliente, acordado esto se realizan pruebas de estrés en el sistema para saber si es necesario adquirir más infraestructura, en caso de que sea necesaria, el gerente notifica al cliente y gestiona la adquisición de estos recursos y se despliega la aplicación. En caso de que no se necesite más infraestructura, se pasa directo a realizar el despliegue de los cambios necesarios para prestar el servicio a estas ciudades.



Powered by
bizagi
Modeler

Figura 9. Migración a otras ciudades.

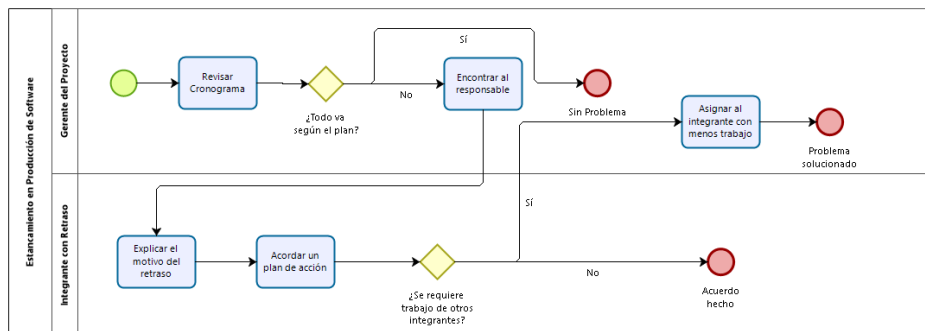
2. *Deserciones*: Para el manejo de una deserción el integrante que va a abandonar el equipo de trabajo notifica al gerente de proyecto, este se encarga de listar el trabajo pendiente para asignarlo al equipo de desarrollo finalmente si todos están satisfechos, la deserción fue solucionada, en caso contrario se negocia el trabajo de esta. Esto, representado en la figura 10.



Powered by
bizagi
Modeler

Figura 10. Deserciones.

3. **Retrasos:** El gerente de proyecto se encarga de revisar el avance semanal del trabajo mediante el cronograma, en caso de que haya un retraso en alguna entrega encuentra al responsable, si es posible soluciona solamente con el responsable y en un caso extremo se asigna trabajo a otros integrantes para poder cumplir con la entrega. Como se evidencia en la figura 11.



Powered by
bizagi
Modeler

Figura 11. Solución de retrasos.

4 **Conflictos:** Para el manejo de los riesgos que puede presentar el manejo de un ambiente de trabajo se va a tomar como medio el dialogo, este dialogo será llevado a cabo por el gerente, quien se encargará de notar la tensión en el ambiente, para lo cual encontrará a los involucrados en el conflicto escuchando el problema y una posible solución que ajustará con respeto al caso para que las

dos partes del conflicto queden satisfechas y se pueda continuar con un buen ambiente laboral, Como se observa en la figura 12.

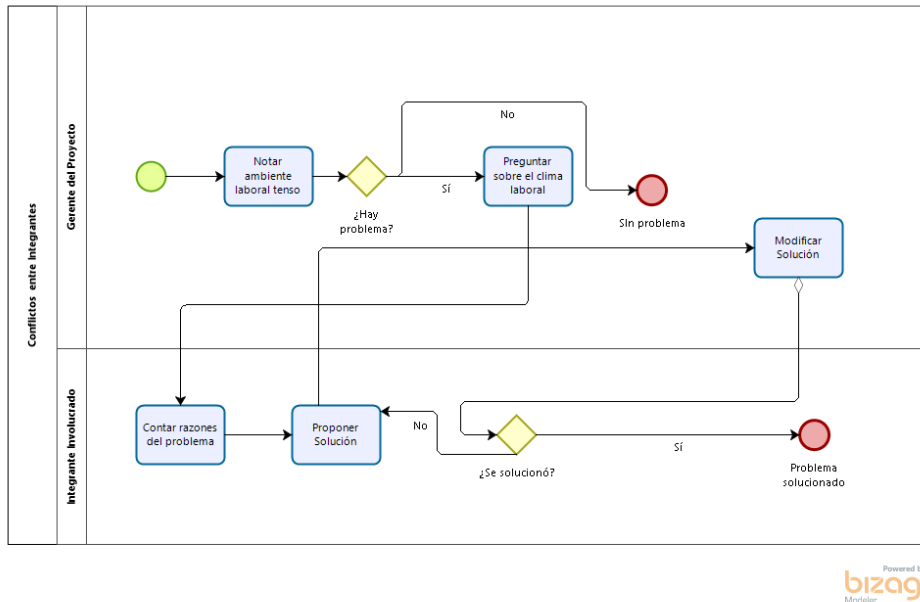


Figura 12. Solución de Conflictos.

11.3 Administración de configuración y documentación

Teniendo en cuenta que un ítem de configuración hace referencia a cualquier activo, servicio o componente que será controlado por la administración de configuraciones, podemos definir una lista de ítems útiles en el desarrollo de nuestro proyecto, estos son:

- Plan de gestión de proyectos necesario para sentar las bases acerca de la forma en la que se constituye la empresa, cómo se procederá frente a los riesgos, los cambios y las cosas que sucedan durante el trabajo en equipo, el producto de software y el proyecto en general en su funcionamiento, organización, desarrollo de las distintas fases, aquí también se incluyen instructivos, y acuerdos entre el cliente y la empresa.
- Especificación de requerimientos de software, necesario para estructurar mejor el contenido explícito de cada parte del producto de software a realizar, además de orientar el desarrollo y las herramientas a utilizar para suplir las necesidades y cumplir los requerimientos
- Documento de diseño de software, el cual es importante para mostrar de manera gráfica cómo se compuso y se formó el producto final de software
- Documentación en Github para manejo de control de versiones en el código, tablas de modificaciones y reportes por reunión para saber avance y tareas finalizadas

Para el manejo de alteraciones en los ítems de la configuración vamos a manejar el mismo proceso, este es iniciado por el cliente pidiendo la alteración, la cual será evaluada por el gerente y un integrante que tenga mayor conocimiento sobre la alteración, frente a esto se van a levantar riesgos, implicaciones y un precio correspondiente que será notificados al cliente, este aceptará si desea continuar con la alteración o desistir, en caso de que continúe el gerente se encarga de definir las tareas correspondientes para llevar a cabo la alteración, delega a los integrantes del grupo y finalmente se acaba el manejo de la alteración.

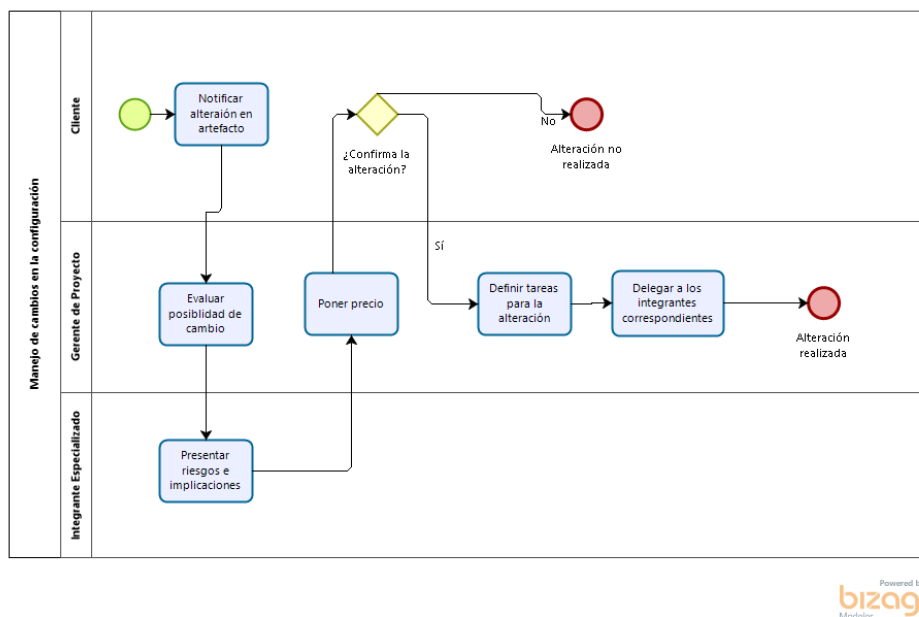


Figura 13. Manejo de Alteraciones.

Artefactos:

Nombre artefacto	Tipo	Descripción	Momento definición
SPMP	documentación	Plan de gestión del proyecto	Desde inicios de agosto de 2019
SRS	documentación	Especificación de requerimientos de software	Desde inicios de septiembre de 2019
SDD	documentación	Documentación de diseño de software	Desde inicios de octubre de 2019
Reporte gerencial	documentación	Documento post mortem acerca de las particularidades	Al finalizar cada entrega parcial al profesor

		de cada documento	
Compodoc	documentación	Herramienta de documentación del código Ionic y Angular	Al iniciar el código
Ejecutable de la funcionalidad principal del proyecto	código	Elemento de pruebas para ver avance del proyecto y comprobar funcionalidad	Mediados de Septiembre de 2019
Ejecutable del proyecto completo y funcional	código	Elemento de pruebas para comprobar funcionalidad completa del proyecto	Mediados de Noviembre de 2019

Tabla 5. Artefactos de documentación y código

11.4 Control de calidad

Para SauveGuard, la calidad es uno de los pilares que sostienen la compañía, Por consiguiente, las revisiones periódicas de los avances tanto de los documentos como de la propia aplicación y los métodos con los cuales garantizaremos una gran calidad en cada una de las entregas son los siguientes:

Para la documentación:

Proceso/Actividad de verificación de calidad	Momento del proyecto en que se hará	Responsables de la revisión
Revisión del documento por parte del grupo: El grupo se reunirá a releer todo el documento nuevamente, para evitar errores gramaticales, desaciertos en la redacción o para complementar los contenidos.	Se realizará semana a semana, hasta el día anterior de la primera revisión no calificable.	Todos los integrantes: Federico Torres Sofía Moreno Juan Diego Osorio Sebastián Valderrama Kevin Peláez
Revisión del documento por parte del profesor: Se le brindará al Ing. Parra los documentos de cada		Ing. Carlos Andrés Parra

entrega para que haga una revisión de ellos y una retroalimentación sobre el trabajo hecho, para poder corregirlo antes de la entrega definitiva.	Se hará la semana anterior a cada una de las entregas, en las salas de ingeniería.	(Profesor de ingeniería de software)
Revisión del documento por parte de la Product Owner: Le brindaremos una copia de la documentación a la dueña del proyecto, con el fin de que sea claro para ella cómo está estipulada SauveGuard como empresa y cuales serán las directrices del proyecto.	Se hará la misma semana de la entrega final, en una reunión entre la dueña del proyecto y la gerente de la organización.	Angélica María Sánchez (Product owner) Ángela Sofía Moreno (Gerente general)
Revisión del documento por parte de un tercero experto: Se hará una reunión con una persona experta en términos de escritura y redacción para que lea el documento y podamos corregir falencias de esta índole.	Será el día anterior a la fecha de cada entrega, para tener tiempo de hacer las otras revisiones y ajustes.	David Santiago Moreno (Estudiante de lenguas modernas) Juan Diego Osorio (Documentador)

Tabla 6. Control de calidad del documento

Para verificar la calidad del software:

Proceso/Actividad de verificación de calidad	Momento del proyecto en que se hará	Responsables de la revisión
Reunión con un experto para pedir consejos y recomendaciones: Como grupo buscaremos a un ingeniero de la universidad para que nos pueda brindar información sobre buenas prácticas que podemos tener a la hora de la programar y a su vez, pedir consejos para cumplir las expectativas y los tiempos.	Esto se hará en la semana siguiente a la entrega final del documento SPMP, en la universidad Javeriana.	Ing. Jaime Pavlich (Profesor de la facultad de ing. De sistemas) Federico Torres (Tester y programador)
División del caso de uso en partes mucho menos		Todos los integrantes:

complejas: Esto con el fin de poder hacer una división más exacta del trabajo y que se pueda verificar más fácilmente si existen errores y donde detectarlos antes de que se conviertan en fallas mucho más complicadas	Se hizo a la par del documento SPMP, cuando se estaban levantando las historias y los to do del proyecto.	Federico Torres Sofía Moreno Juan Diego Osorio Sebastián Valderrama Kevin Peláez
Pruebas unitarias por caso de uso: Una vez se complete un caso de uso, se harán pruebas características de este requerimiento con el fin de verificar el correcto funcionamiento de esta parte del software.	En el momento del desarrollo por caso de uso a lo largo del semestre, se harán las correspondientes pruebas para verificar el funcionamiento.	Todos los integrantes: Federico Torres Sofía Moreno Juan Diego Osorio Sebastián Valderrama Kevin Peláez
Prueba de la aplicación en general: Al completar todos los casos de uso e integrarlos a la aplicación, se hará una prueba general para verificar cada uno de los posibles resultados que previmos en las pruebas anteriores.	Se hará en la semana anterior a la entrega final no calificable en las reuniones del grupo.	Todos los integrantes: Federico Torres Sofía Moreno Juan Diego Osorio Sebastián Valderrama Kevin Peláez
Pruebas con una persona externa: Se buscará un bici-usuario para que pueda hacer pruebas sobre el software y poder evidenciar si el uso de la aplicación es sencillo para cualquier usuario, además de poder verificar realmente si cumple con lo esperado por parte del cliente final.	Se hará después de hacer la revisión de la aplicación por parte de SauveGuard. En una reunión con uno de los bici-usuarios	Erick Duarte Bici-usuario Kevin Peláez (Diseñador y analista)
Pruebas con el profesor: Se harán dos pruebas de la aplicación antes de cada entrega para que el profesor nos brinde diferentes recomendaciones o mejoras.	Se hará la semana anterior a cada una de las entregas de los demos funcionales.	Ing. Carlos Andrés Parra (Profesor de ingeniería de software)
Realizar ajustes según cambios ocurrentes: Cada		Angélica María Sanchez (Product Owner)

vez que exista un cambio posible en la aplicación, se discutirá en el grupo de trabajo y se realizará la asignación de esta tarea a alguno de los miembros.	Se hará a lo largo del semestre, dependiendo de las preferencias del product owner.	Todos los integrantes Federico Torres Sofía Moreno Juan Diego Osorio Sebastián Valderrama Kevin Peláez
Pruebas de error voluntario: Se harán pruebas con datos incorrectos o cometiendo errores voluntarios, de modo que podamos hacer una medición de la capacidad propia de la aplicación de responder ante estas situaciones.	Se hará en la misma reunión que se haga para hacer las pruebas de la aplicación en general.	Todos los integrantes: Federico Torres Sofía Moreno Juan Diego Osorio Sebastián Valderrama Kevin Peláez

Tabla 7. Control de calidad del software

12 Anexos

Anexos en carpeta:

Anexo 1 – Requerimientos

Anexo 2 – Estimaciones

Anexo 3 – Diagrama_CU

Anexo 4 – Diagramas_Grantt

13 Referencias

- [1]TutorialsHand, "Software Project Management Plan - SPMP Document", *tutorialsinhand.com*, 2019. [Online]. Available: <https://tutorialsinhand.com/tutorials/software-engineering-tutorial/software-project-management/spmp-document.aspx>. [Accessed: 22- Aug- 2019].
- [2]Junta De Andalucía, "Especificación de Requisitos del Sistema | Marco de Desarrollo de la Junta de Andalucía", *Juntadeandalucia.es*, 2019. [Online]. Available: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/407>. [Accessed: 22- Aug- 2019].
- [3]T. Dinesh, "Software Design Documentation (SDD)", *Ecomputernotes.com*, 2019. [Online]. Available: <http://ecomputernotes.com/software-engineering/softwaredesigndocumentation>. [Accessed: 22- Aug- 2019].
- [4]"¿Qué significa realmente SOS? La señal de socorro que usa todo el mundo desde la tragedia del Titanic", *Es.gizmodo.com*, 2019. [Online].

Available: <https://es.gizmodo.com/que-significa-realmente-sos-la-senal-de-socorro-que-se-1820764733>. [Accessed: 22- Aug- 2019].

- [5]"Definición de requerimiento — Definicion.de", *Definición.de*, 2019. [Online]. Available: <https://definicion.de/requerimiento/>. [Accessed: 22- Aug- 2019].
- [6]Angular, "Angular", *Angular.io*, 2019. [Online]. Available: <https://angular.io/guide/architecture>. [Accessed: 19- Aug- 2019].
- [7]Ionic, "What is Ionic Framework? - Ionic Documentation", *Ionic Docs*, 2019. [Online]. Available: <https://ionicframework.com/docs/intro>. [Accessed: 19- Aug- 2019].
- [8]"¿Qué es Interfaz de usuario?", *Workana.com*, 2019. [Online]. Available: <https://www.workana.com/i/glosario/interfaz-de-usuario/>. [Accessed: 22- Aug- 2019].
- [9]BBVA, "Qué es un 'product backlog' y cuál es su función", *BBVA NOTICIAS*, 2019. [Online]. Available: <https://www.bbva.com/es/que-es-un-product-backlog-y-cual-es-su-funcion/>. [Accessed: 22- Aug- 2019].
- [10]"¿Cómo funciona la metodología Scrum?", *platzi.com*, 2019. [Online]. Available: https://platzi.com/blog/metodologia-scrum-fases/?utm_source=google&utm_medium=paid&utm_campaign=intention&utm_content=&adgroup=71299386940&gclid=EAlaIQobChMIs-jm4Lf-4wIVk1qGCh2RsATkEAAAYASAAEgKsf_D_BwE. [Accessed: 12- Aug- 2019].
- [11]"Desarrollo de software. Ciclo de vida RUP (Rational Unified Process)", *Jummp*, 2019. [Online]. Available: <https://jummp.wordpress.com/2011/04/06/desarrollo-de-software-ciclo-de-vida-rup-rational-unified-process/>. [Accessed: 17- Aug- 2019].
- [12]"¿Qué es un 'framework'?", *jordisan.net*, 2019. [Online]. Available: <https://jordisan.net/blog/2006/que-es-un-framework/>. [Accessed: 22- Aug- 2019].
- [13]"Sistema Operativo: Concepto, Usos, Componentes y Funciones", *Concepto.de*, 2019. [Online]. Available: <https://concepto.de/sistema-operativo/>. [Accessed: 22- Aug- 2019].
- [14]A. Gonzalez, "¿Qué es Android?", *Xatakandroid.com*, 2019. [Online]. Available: <https://www.xatakandroid.com/sistema-operativo/que-es-android>. [Accessed: 22- Aug- 2019].
- [15]J. Filgueira, "¿Qué es iOS? | Gabit", *Gabit.org*, 2019. [Online]. Available: <http://www.gabit.org/gabit/?q=es/que-es-ios>. [Accessed: 22- Aug- 2019].