**SPMP PROYECTO FITT  
  
  
  
  
  
  
  
  
  
  
MIGUEL ANGEL BELTRAN RODRIGUEZ  
MIGUEL ANGEL CHAVES CORTES  
LAURA JULIANA MORA PÁEZ  
JHONNY ALEXANDER PARRA BARRERA  
JUAN SEBASTIAN TRIANA PÉREZ  
  
  
  
  
  
  
  
  
  
  
  
PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE INGENIERÍA DE SISTEMAS**

**BOGOTÁ D.C**

**2019**

# Historial de cambios

|  |  |  |
| --- | --- | --- |
| Fecha | Autor(es) | Descripción del cambio |
| 20/02/2019 | Miguel Beltrán y Jhonny Parra | Creación de la portada, tabla de historial de cambios, tabla de contenidos y lista de tablas. Primera versión de las secciones 6.1, 6.2 y 6.3. |
| 21/02/2019 | Miguel Beltrán y Jhonny Parra | Modificación de las restricciones de la sección 6.3 y primera versión de la sección 6.4. Elaboración del glosario y se agregó una referencia. |
| 23/02/2019 | Sebastian Triana | Actualización de la información de la sección 11.1. |
| 23/02/2019 | Laura Mora | Primera versión de las secciones 7.1, 7.2, borrador secciones 7.3, 7.4 |
| 23/02/2019 | Miguel Angel Chaves | Primera versión de la sección 10. |
| 24/02/2019 | Miguel Beltrán | Revisión y corrección del formato del documento de todas las secciones. Se completo la descripción de algunos términos en el glosario. Creación de la tabla de ilustraciones |
| 24/02/2019 | Jhonny Parra | Primera versión de la sección 8. Se agregaron los dos métodos de estimación. |
| 24/02/2019 | Laura Mora | Se agrego organigrama, diagrama WBS, se actualizaron herramientas y lenguajes |
| 25/10/2019 | Sebastián Triana y Miguel Beltrán | Primera versión de diagramas BPMN sobre la calidad del producto y del documento |
| 24/02/2019 | Miguel Angel Chaves | Actualización y corrección de la sección 10, Primera versión de la sección 9. |
| 26/02/2019 | Sebastián Triana | Primera versión sección 9.2 y corrección sección 11.1 |
| 26/02/2019 | Miguel Beltrán | Creación de los entregables y modificación de referencias. |
| 26/02/2019 | Laura Mora | Descripción roles organigrama, primera versión de presupuestos. |
| 26/02/2019 | Jhonny Parra | Actualización de las herramientas y métodos de estimación. Sección 8. |
| 27/02/2019 | Sebastián Triana | Se corrigió la sección 9.2, se agregó imágenes de la sección en la tabla de ilustraciones, se corrigieron y terminaron los diagramas BPMN de control de calidad. |
| 27/02/2019 | Laura Mora | Se agregaron responsabilidades al organigrama, se agregó sección 7.3 |
| 27/02/2019 | Miguel Angel Chaves | Actualización del punto 9.1 y generación del diagrama BPMN del mismo. |
| 27/02/2019 | Miguel Beltrán | Actualización de entregables y corrección del formato del documento. Primera versión de la sección 6.5 |
| 27/02/2019 | Jhonny Parra | Planes de capacitación sección 8.2.1 |
| 28/02/2019 | Sebastian Triana | Primera versión sección 11.3 |

Tabla 1. Historial de cambios

# Resumen

**Propósito**

Esta sección debe proveer una descripción general del contenido del documento, especificando su organización o distribución de forma que el lector pueda estructurar la información que encontrará en del documento.

**Contenido**

Descripción breve del contenido, propósito y alcance del proyecto. Descripción de la audiencia esperada del documento (para quién es importante leer este documento).

**Tamaño sugerido**

1/2 página.

**Referencias**

[1], sección 5.

# Tabla de contenidos

[1 Historial de cambios 4](#_Toc1944330)

[2 Resumen 5](#_Toc1944331)

[3 Tabla de contenidos 6](#_Toc1944332)

[4 Lista de figuras 7](#_Toc1944333)

[5 Lista de tablas 8](#_Toc1944334)

[6 Vista general del proyecto 9](#_Toc1944335)

[6.1 Visión del producto 9](#_Toc1944336)

[6.2 Propósito, alcance y objetivos 9](#_Toc1944337)

[6.3 Supuestos y restricciones 10](#_Toc1944338)

[6.4 Entregables 11](#_Toc1944339)

[6.5 Evolución del plan 12](#_Toc1944340)

[6.6 Glosario 12](#_Toc1944341)

[7 Contexto del proyecto 13](#_Toc1944342)

[7.1 Modelo de ciclo de vida. 13](#_Toc1944343)

[7.2 Lenguajes y herramientas. 14](#_Toc1944344)

[7.3 Plan de aceptación del producto 16](#_Toc1944345)

[7.4 Organización del proyecto y comunicación. 17](#_Toc1944346)

[8 Administración del proyecto 19](#_Toc1944347)

[8.1 Métodos y herramientas de Estimación 19](#_Toc1944348)

[8.2 Inicio del proyecto 19](#_Toc1944349)

[8.3 Planes de trabajo del proyecto 20](#_Toc1944350)

[9 Monitoreo y control del proyecto 22](#_Toc1944351)

[9.1 Administración de requisitos 22](#_Toc1944352)

[9.2 Monitoreo y control del progreso 22](#_Toc1944353)

[9.3 Cierre del proyecto 23](#_Toc1944354)

[10 Entrega del producto 24](#_Toc1944355)

[11 Procesos de soporte 25](#_Toc1944356)

[11.1 Ambiente de trabajo 25](#_Toc1944357)

[11.2 Análisis y administración de riesgos 26](#_Toc1944358)

[11.3 Administración de configuración y documentación 27](#_Toc1944359)

[11.4 Control de calidad 27](#_Toc1944360)

[12 Anexos 29](#_Toc1944361)

[13 Bibliografía 30](#_Toc1944362)

# Lista de figuras

[Ilustración 1. Organigrama 20](#_Toc2202581)

[Ilustración 2. WBS 27](#_Toc2202582)

[Ilustración 3. Ecuación para calcular el avance de las tareas 32](#_Toc2202583)

[Ilustración 4. Entrega del producto 35](#_Toc2202584)

[Ilustración 5. Listas del tablero de SCRUM. 36](#_Toc2202585)

# Lista de tablas

[Tabla 1. Historial de cambios 3](#_Toc2202570)

[Tabla 2. Entregables 11](#_Toc2202571)

[Tabla 3. Glosario 12](#_Toc2202572)

[Tabla 4. Herramientas 18](#_Toc2202573)

[Tabla 5. Stakeholders 19](#_Toc2202574)

[Tabla 6. Roles 22](#_Toc2202575)

[Tabla 7. Estimación por casos de uso 24](#_Toc2202576)

[Tabla 8. Capacitaciones 26](#_Toc2202577)

[Tabla 9. Nomina 28](#_Toc2202578)

[Tabla 10. Costos de transporte 29](#_Toc2202579)

[Tabla 11. Total de presupuesto del proyecto 29](#_Toc2202580)

# Vista general del proyecto

## Visión del producto

Fitt busca ser la aplicación móvil de apoyo preferida por aquellas personas que disfrutan de hacer ejercicio en su casa o al aire libre, sin necesidad de acompañamiento presencial de un equipo especializado, conectando una red de usuarios con deseos de compartir su experiencia en el mundo del ejercicio y acompañándolos en su intento de conseguir los mejores resultados en sus entrenamientos. Todo esto para contribuir al bienestar de las personas, fomentando un estilo de vida saludable e impulsando la práctica de actividad física que no requiere de equipo y espacios sofisticados, brindando una herramienta que les permita tener el control sobre su actividad física y su progreso. (ver sección 6.2.1)

La aplicación móvil en desarrollo permitirá a sus usuarios crear, compartir y adoptar un amplio número de rutinas físicas, calificarlas y dejar sus comentarios. Así mismo el usuario podrá ver su progreso a través del tiempo, tener un calendario para organizar sus rutinas y encontrar sitios cercanos para entrenar. Los usuarios dispuestos a ir un poco más allá podrán pagar una membresía que les permitirá acceder a entrenadores personales, vía web y estos se encargarán de brindar asesoría especializada a sus clientes (ver sección 6.2).

## Propósito, alcance y objetivos

**Propósito**

Hacer ejercicio para ponerse en forma o llevar una vida saludable es uno de los propósitos más populares y a la vez más difíciles de conseguir para las personas hoy en día, la falta de motivación o tiempo libre y el desconocimiento en el campo de la actividad física termina provocando que muchas personas abandonen sus metas de convertirse en alguien mejor física y emocionalmente antes de tiempo. Fitt tiene como propósito acompañar y hacer más fácil para las personas el proceso de iniciar o continuar en el hábito del ejercicio físico.

**Alcance**

El mundo del fitness y la actividad física es muy extenso, con un amplio espectro de ejercicios físicos y métodos de entrenamiento, Fitt propone concentrarse únicamente en ejercicios que no necesiten de equipo o material especializado, cuando mucho el usuario tendrá que desplazarse a un parque en busca de espacios que mejoren la calidad de su entrenamiento. A continuación, se enuncian algunas características que se espera que tenga la aplicación:

* Crear rutinas a partir de ejercicios que ya están en la aplicación.
* Tener un perfil básico con la información del usuario.
* Tener un calendario para administrar las rutinas.
* Compartir rutinas con otros usuarios.
* Seguir otros usuarios para ver sus rutinas.
* Tener un mapa con el recorrido en tiempo real.
* Tener logros y metas.
* Tener una ayuda audiovisual durante la ejecución de una rutina y ejercicio.
* Ver los parques cercanos con comentarios y calificaciones.
* Se podrá acceder a una cuenta premium para tener asesoría especializada brindada por entrenadores seleccionados por el usuario.
* Contactarse con entrenadores para poder ser asesorados.
* Los entrenadores tendrán una lista de usuarios que asesoran y podrán interactuar con ellos por medio de un chat donde sugerirán rutinas o darán consejos.
* No podrá detectar movimientos durante ejercicios.
* No tendrá un mapa con la ubicación de los otros usuarios
* No estará implementado el sistema de pago.
* No estarán asociadas las redes sociales del usuario a la aplicación.
* Aunque los entrenadores pueden brindar ayuda sobre dieta o alimentación la aplicación no lo contempla.
* La aplicación no complementa ni sustituye un gimnasio o cualquier centro deportivo para actividad física.

**Objetivos**

* **General**: Aplicar los conocimientos adquiridos en la clase de ingeniería de software y la carrera de ingeniería de sistemas para llevar a cabo el proceso de desarrollo de Fitt, una aplicación móvil de apoyo para personas que realizan actividad física.
* **Específicos**:
* Incentivar la actividad física.
* Entender la importancia de realizar una buena planeación de proyecto para construir software de calidad.
* Entender la importancia de trabajar en equipo con roles y responsabilidades definidas y además cumpliendo con los cronogramas acordados.
* Aprender a usar herramientas nuevas de desarrollo de software.
* Aprender a realizar una buena documentación de un proyecto de software.
* Aprender a usar estrategias y metodologías para el desarrollo de un proyecto de software.
* Adquirir experiencia sobre el desarrollo de un proyecto de software utilizando todo lo aprendido durante la carrera.

## Supuestos y restricciones

**Supuestos**

* El grupo contará con una disponibilidad y espíritu de trabajo que permita el desarrollo del cronograma dentro de los marcos temporales establecidos (ver sección 6.5 y 11.1).
* Las herramientas de desarrollo de software seleccionadas serán gratuitas y tendrán la suficiente documentación como para ser usadas a lo largo del proyecto (ver sección 7.2 y 9.2).
* Todos los miembros del grupo cuentan con equipos con los requisitos recomendados para el correcto funcionamiento del software de desarrollo.
* Se cuenta con un profesor que guía el aprendizaje y está pendiente del desarrollo y evolución del proyecto.
* El producto final funcionara en dispositivos Android 5.0 hacia arriba (ver sección 7.2).
* Los usuarios de la aplicación estarán dispuestos a brindar información personal y a dar permisos para acceder a funcionalidades específicas de sus dispositivos.
* Los usuarios harán un uso correcto de la aplicación.

**Restricciones**

* No existe como tal un presupuesto real debido a que es un proyecto universitario con fines académicos enmarcado en el curso de ingeniería de software, por lo tanto, solo se pueden usar herramientas dadas por la universidad o gratuitas (ver sección 6.2 y 7.2).
* El proyecto debe terminarse antes de que finalice el semestre académico.
* El grupo de trabajo estará conformado únicamente por cinco (5) personas, este número es fijo y se espera que no cambie a lo largo del proyecto.
* Los integrantes del equipo de trabajo son estudiantes de ingeniería de sistemas, por lo que no están capacitadas para realizar todas las tareas necesarias para el desarrollo de un proyecto de software, como, por ejemplo: animaciones, diseño gráfico, *marketing*, publicidad, etc. Esto limita algunas características del producto o su calidad final, ejemplo: el diseño de la ayuda audiovisual, interfaces, logos e iconos que tendrá la aplicación (ver sección 6.2).
* Se necesita conexión a internet para que la aplicación funcione.
* La aplicación debe implementar un modelo cliente/servidor.
* La aplicación debe ser desarrollado bajo el paradigma orientado a objetos.
* La aplicación debe tener persistencia de los datos.
* La aplicación debe tener un manejo de interfaz gráfica fuerte.

## Entregables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Cantidad | Descripción | Remitente | Fecha |
| SPMP | 1 | Documento del plan de administración del proyecto. | Cliente | 05/02/2018 |
| Sprint 1 | 1 | Aplicación con la interfaz del caso de uso: crear y compartir rutinas. | Cliente | 19/02/2018 |
| Sprint 2 | 1 | Aplicación con la implementación del caso de uso más importante | Cliente | 02/03/2019 |
| SRS, corrección SPMP y caso de uso más importante | 1 | Documento con la descripción del software y la corrección del SPMP. Aplicación con el caso de uso más importante funcionando correctamente. | Cliente | 16/03/2019 |
| Sprint 3 | 1 | Aplicación con la implementación de los casos de uso que le agregan valor a la aplicación. (ver sección 6.2 Alcance) | Cliente | 30/03/2019 |
| Entrega final | 1 | Aplicación completa con la documentación y corrección del SRS | Cliente | 14/04/2019 |

Tabla 2. Entregables

## Evolución del plan

Durante el desarrollo de este documento se utilizaron dos herramientas para controlar los cambios y modificaciones del documento. El primero de ellos es la utilización del historial de cambios de Word con el OneDrive, el cual tienen gestión de cambios y versiones, además cada integrante debe poner en la sección de historial de cambios cada modificación del documento con su respectiva fecha y descripción. La segunda herramienta es “Trello” (ver sección 8.1) la cual aparte de permitirnos estimar nos brinda instrumentos para poder gestionar tareas las cuales se listan y se van dando por terminadas. Esta última es muy útil para tener control sobre lo que falta sobre este documento, lo que se ha editado y quien lo ha editado.

En la sección **[11.3 - Administración de configuración y documentación]** podemos encontrar un BPM y una descripción sobre el control de la documentación y escritura de este documento.

## Glosario

|  |  |
| --- | --- |
| Concepto | Explicación |
| Entrenador personal | Persona que acompaña, asesora y supervisa el entrenamiento físico de un deportista o alguien que realiza actividad física [1] |
| Ejercicio | En el contexto del proyecto es una actividad física con el propósito de ejercitar el cuerpo y mejorar su estado físico [1] |
| Fitness | Es un estilo de vida saludable en el cual se practica actividades que mantienen y mejoran el estado de salud actual de la persona [2] |
| Aplicación móvil |  |
| Android | Es un sistema operativo enfocado mayormente en dispositivos móviles [3] |
| Ingeniería de software |  |
| Software |  |
| Persistencia |  |
| Interfaz gráfica |  |
| Rutina física | Agrupación de ejercicios físicos que se desarrollan de manera recurrente y continua durante un periodo de tiempo [1] |
| Entrenamiento |  |
| Dispositivo móvil |  |
| Audiovisual |  |
| Parque | Para nuestro proyecto este término solo se delimita a parques con lo necesario para realizar rutinas físicas, es decir, espacios públicos recreativo que tengan suficiente espacio libre y/o que tengan herramientas para hacer ejercicios físicos. |
| Modelo cliente/servidor |  |

Tabla 3. Glosario

# Contexto del proyecto

## Modelo de ciclo de vida.

En esta sección se presentan las metodologías consideradas a seguir, los motivos por los cuales fueron consideradas y la metodología guía que se va a utilizar; para este caso el proyecto se va a realizar con metodologías ágiles debido a su flexibilidad a cambios con lo que se asegura una mejora continua y adecuación constante del producto final.

**Scrum**

Scrum es una metodología que consta de entregas parciales del proyecto final, donde estas entregas están priorizadas por el beneficio que aportan al proyecto. Por esto Scrum es principalmente usado para proyectos con entornos complejos, donde se necesita obtener resultados lo más pronto posible, donde los requisitos pueden estar cambiando o no están bien definidos y donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales [4].

Scrum se compone de los siguientes factores:

* **Roles:** product owner (representa los interesados en el producto final), scrumMaster (responsable del proceso) y scrumTeam.
* **Sprint:** los sprints son las etapas donde se va entregando partes del trabajo, tienen una duración máxima de 4 semanas.
* **Artefactos:** product Backlog (listado de los requisitos del sistema), sprint Backlog (listado de tareas para realizar cada una de las funcionalidades) y grafica de progreso.
* **Comunicación:** se basa en diversas reuniones, daily scrum, reunión de planificación del sprint, reunión de revisión del sprint, retrospectiva del sprint.

**Extreme programming (XP)**

XP es una metodología ágil la cual se puede describir como el conjunto de pasos de diferentes metodologías, acoplados de forma que sean más flexibles para llegar a un desarrollo más agradable y sencillo; por otro lado, sus bases se encuentran en la comunicación, simplicidad, retroalimentación y valentía. Al igual que consta de 12 pilares fundamentales que funcionan como guía para el desarrollo del proyecto, estos pilares son: el principio de pruebas, el proceso de planificación, el cliente en el lugar, integración continua, refactorización, entregas pequeñas, diseño simple, metáfora, propiedad colectiva del código, estándar de codificación y la semana de 40 horas [5].

**Dynamic Systems Development (DSDM)**

DSDM se basa en la filosofía que cualquier proyecto debe estar formado con objetivos bien definidos y se enfoca en entrega temprana de beneficios reales para el negocio. Por otro lado, se conforma por 8 principios, la colaboración, la entrega a tiempo, concentrarse en lo que el negocio necesita, no comprometer la calidad, construir desde bases firmes, el desarrollo iterativo, comunicar contantemente y demostrar control sobre el proyecto [6].

**Enterprise Unified Process (EUP)**

EUP es considerado una extensión del Rational Unified Process (RUP), se basa en realizar mejoras a las deficiencias presentadas por RUP donde no se consideraba el soporte al sistema y el aislamiento de un sistema. EUP tiene en cuenta al desarrollo de software como un proceso independiente, donde al mismo tiempo establece creación, mejoras y reemplazos son parte del ciclo de vida de este. EUP está compuesto por 6 fases, comienzo, elaboración, construcción, transición, producción y jubilación [7].

**Justificación del modelo.**

Tras analizar las diversas metodologías existentes se llegó a la decisión de utilizar SCRUM, esto debido a que:

* Involucra reuniones donde se analizan las diferentes partes del proyecto y los problemas que se presenten.
* Nos brinda flexibilidad para realizar algunos cambios o agregar funcionalidades que se vean necesarias en las diferentes etapas del proyecto.
* Nos permite tener un control de lo que se está haciendo, lo que ya se hizo y lo que falta por hacer.
* Se ajusta a diferentes tiempos de entrega ya que se encuentra dividido por los diferentes sprints que se presenten.

No se decidió por el uso de las otras debido a que por ejemplo en el caso de XP no podemos tener una definición del costo y tiempo de desarrollo, al igual que la programación en parejas puede llegar a ser compleja ya que todos tienen diferentes modos para programar y no siempre es sencillo coordinar en ideas para la solución de problemas. Por otro lado, para el caso de (DSDM), no se utilizó debido a que ningún sistema es realizado a la perfección en el primer intento, se necesita una alta participación para evitar caer en supuestos. Finalmente, no se utilizó EUP al centrarse únicamente en actividades de alto valor se puede dejar de lado actividades que influyan fuertemente en el desarrollo del sistema.

## Lenguajes y herramientas.

Para elegir estas herramientas se consideraron los siguientes aspectos:

* Un lenguaje que todos los integrantes del grupo conocieran.
* El uso de las herramientas no tuviera un costo muy elevado, preferiblemente que no tuviera un costo.
* Entre las restricciones (ver sección 6.3) se plantea que el proyecto debe hacerse bajo el paradigma orientado a objetos.
* Una herramienta que al menos alguno de los integrantes del equipo conozca su funcionamiento.
* Un lenguaje y herramientas que tengan la suficiente documentación para el uso.
* Compatibilidad entre herramientas.
* Tener un control de versiones.

**Java**

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra [8].

Java se basa en 5 principios:

* Usar el paradigma de programación orientado a objetos.
* Permitir la ejecución de un programa en distintos sistemas operativos.
* Incluir soporte para trabajo en la red.
* Diseño para ejecutar códigos en sistemas remotos de segura.
* Fácil de usar.

**Android Studio**

Android Studio es un entorno de desarrollo integrado para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que facilitan la creación de apps para Android, entre las que encontramos [9]:

* Un sistema de compilación basado en Gradle flexible.
* Un emulador rápido con varias funciones.
* Un entorno unificado donde se puede realizar proyectos para todos los dispositivos Android.
* Instant Run para aplicar cambios mientras la app se ejecuta sin la necesidad de compilar un nuevo APK.
* Integración de plantillas de código y GitHub para ayudar a compilar funciones comunes de las apps e importar ejemplos de código.
* Gran cantidad de herramientas y frameworks de prueba.
* Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
* Compatibilidad con C++ y NDK
* Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine.

**Firebase**

Firebase es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente que se encuentra conectado. Cuando se compilan apps multiplataforma con los SDK de iOS, Android y JavaScript, todos los clientes comparten una instancia de *Realtime Database* y reciben actualizaciones automáticamente con los datos más recientes.

Firebase permite compilar aplicaciones colaborativas, ya que permite el acceso seguro a la base de datos directamente desde el código, manteniendo los datos de forma local. Además, incluso cuando no hay conexión, se siguen activando los eventos en tiempo real, cuando el dispositivo se vuelve a conectar, se sincronizan los cambios de los datos locales con las actualizaciones remotas que ocurrieron mientras se estuvo sin conexión, lo que combina los conflictos de forma automática.

También proporciona un lenguaje flexible de reglas basadas en expresiones, conocido como reglas de seguridad de Firebase *Realtime Database*, para definir cómo se deberían estructurar los datos y en qué momento se pueden leer o escribir. Integrar Firebase Authentication permite a los desarrolladores definir quién tiene acceso a qué datos y cómo acceden a ellos [10].

**Google Cloud**

Google Cloud es una de las diversas herramientas que ofrece Google donde se presentan servicios de nube para cualquier público, Google Cloud es un espacio virtual donde se pueden realizar tareas que antes dependían de cierto Software o Hardware especifico.

Entre las API que serán utilizadas para fitt provenientes de este servicio tenemos:

* Google Maps: esta API combina 18 interfaces de programación en tres interfaces principales, mapas, rutas y lugares permitiendo así una fácil integración en otras aplicaciones [11].
* Cloud Text-To-Speech: permite sintetizar alrededor de 30 voces en varios idiomas y con diversas variantes, generando audios de alta fidelidad con voces naturales, esto gracias a el DeepMind en WaveNet y las redes neuronales que maneja Google [12].

**GitHub**

Es un controlador de versiones público, este repositorio permite guardar una copia del código generado y a su vez tiene la ventaja de que se puede realizar consultas de versiones anteriores de la información guardada.

**Justificación de herramientas**

Se tomó la decisión de utilizar las herramientas mencionadas anteriormente como consecuencia de la verificación del cumplimiento con las condiciones presentadas anteriormente.

Para el caso de Java es un lenguaje de programación que cumple con el requisito planteado en secciones anteriores de que sea orientado a objetos pero a diferencia de otros lenguajes orientados a objetos como c#, objetive c, clarion, entre otros, es un lenguaje que todo el grupo de bittus conoce y tiene experiencia trabajando en el, por otro lado se prefirió sobre C++ porque presenta algunas facilidades como el hecho de que muchas estructuras de datos ya se encuentran implementadas en diversas librerías.

Android Studio, Firebase y Google Cloud se eligieron a causa del gran mercado que abarca, a la experiencia que algunos de los integrantes utilizando estas herramientas, presentan una amplia cantidad de documentación para su manejo y finalmente son adaptables para el uso entre estas herramientas.

Finalmente se tomó la decisión del uso de GitHub debido a que como es una herramienta para el uso colaborativo de repositorios era más sencillo para el manejo de todos los integrantes del grupo, a su vez el poder consultar versiones anteriores de los códigos que se encuentren en el repositorio nos da la seguridad de que en caso de algún cambio alguna funcionalidad deje de funcionar siempre se puede volver a la versión anterior.

## Plan de aceptación del producto

En la presente sección se plantean los acuerdos a los que se llegó con el cliente para que el producto sea aceptado, al igual que se presentan las herramientas para dicho objetivo.

Los entregables se encuentran en la sección 6.4

**Herramientas**

|  |  |  |
| --- | --- | --- |
| **Entregable** | **Criterios** | **Herramienta** |
| SPMP | * Buena redacción. * Buena ortografía. * Citas y Referencias en formato IEEE. * Imágenes, tablas y ecuaciones bien enumeradas. * Información completa. | * Uso de Word Online con OneDrive. * El director de documentación junto con el auditor de documentación se encarga de revisar que todos los lineamientos establecidos se cumplan en el documento. |
| Primer prototipo | * Caso de uso con mayor complejidad. * Seguir lineamientos planteados por el director de desarrollo. * Sin errores de ejecución. * Documentar caso de uso. | * Uso de GitHub para trabajo colaborativo en el desarrollo del software. * El director de desarrollo estará pendiente de que se cumplan cada uno de los lineamientos establecidos. * Cumplir con las normas de calidad. |
| Segundo prototipo | * Sin errores de ejecución y compilación. * 70% de los casos de uso en funcionamiento. * Bien documentados. | * Uso de GitHub para trabajo colaborativo en el desarrollo del software. * El director de desarrollo estará pendiente de que se cumplan cada uno de los lineamientos establecidos. * Cumplir con las normas de calidad. |

Tabla 4. Herramientas

## Organización del proyecto y comunicación.

En esta sección se presenta la organización de Bittus para el proyecto, la especificación del rol que cumple cada integrante del equipo junto con las responsabilidades que conlleva dicho rol. Al igual que se dan a conocer las diferentes entidades que influyen de forma directa o indirecta en el desarrollo del proyecto.

**Interfaces externas o Stakeholders.**

A continuación, se presenta la descripción de las personas interesadas, entidades involucradas que cumplen el rol de stakeholders para el desarrollo de la aplicación Fitt.

|  |  |  |
| --- | --- | --- |
| **Nombre entidad o stakeholder** | **Descripción** | **Responsabilidad** |
| Entrenadores. | Personas interesadas en dar un asesoramiento en rutinas a usuarios por medio de una app. | * Crear rutinas para usuarios a cargo. * Asesorar usuarios a cargo. |
| Clientes. | Personas interesadas en tener rutinas que puedan hacer en cualquier momento y no necesariamente necesiten ir a un gimnasio. | * Consultar rutinas. * Crear rutinas propias. * Seguir usuarios. |
| Otras aplicaciones con el mismo fin. | Aplicaciones que contienen funcionalidades similares a las propuestas. | Competencia.  Monitorear servicios que ofrecen para ser más competitivos. |
| Gimnasios | Gimnasios que los usuarios conozcan o suelan frecuentar | Competencia.  Monitorear servicios que ofrecen para ser más competitivos. |

Tabla 5. Stakeholders

**Organigrama y descripción de roles.**

Para un manejo eficaz del proyecto a cada integrante se le asignó el rol de acuerdo con sus habilidades, fortalezas, gustos y afinidad presentada para cada uno de los roles propuestos.

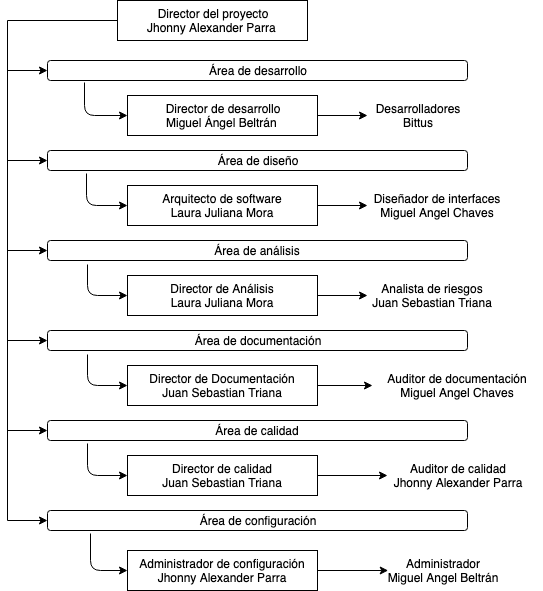


Ilustración 1. Organigrama

|  |  |  |
| --- | --- | --- |
| **Rol** | **Descripción** | **Responsabilidad** |
| **Director del proyecto** | Es el responsable de liderar el equipo en busca de cumplir el objetivo. Debe tener ciertas habilidades como el ser un persona organizada, apasionada y orientada a cumplir las metas, capaz de reconocer riesgos. [13] | * Ser un puente entre el cliente y el equipo de desarrollo. * Unificar ideas con respecto al proyecto. |
| **Director de desarrollo** | Es el encargado de liderar el equipo de desarrolladores en cada uno de los sprints de desarrollo. | * Revisar código para verificar que solo tenga componentes funcionales |
| **Desarrolladores** | Son los encargados de implementar componentes, localizar, corregir defectos, así como desarrollar artefactos de  instalación. [14] | * Desarrollan el código fuente para cada uno de los casos de uso. |
| **Arquitecto de software** | Es una persona con habilidades de diseño, pose madurez, visión y sabe detectar problemas rápidamente y sabe tomar decisiones en ausencia de una completa información. Debe tener habilidades tales como  liderazgo, orientación a metas, comunicación y ser  proactivo. [14] | * Plantear requisitos con base en las solicitudes del cliente. * Plantear diseño técnico. |
| **Diseñador interfaces** | Es quien debe definir los lineamientos visuales, aplicando bases teóricas, resultados del análisis del comportamiento de los usuarios y buenas prácticas de la usabilidad y accesibilidad. Su principal objetivo es brindar la mejor experiencia posible para el público objetivo del proyecto. [15] | * Diseñar GUI para el cliente. |
| **Director de análisis** | Es la persona encarga de liderar los procesos de identificación y gestión de riesgos, acompaña a los miembros del equipo en el manejo de los diferentes riesgos que se pueden presentar. Gestiona y mantiene un registro de riesgos y de reuniones donde su enfoque han sido los mimos. | * Encargado de determinar posibles riegos. * Determinar formas para mitigar riesgos. |
| **Analista de riesgos** | Es la persona encargada en apoyar a el director de riesgos con la definición, implementación y evaluación de acciones a tomar ante los riesgos. | * Verificar si se han presentado riegos y da ideas para mitigarlos. |
| **Director de documentación** | Es el encargado de revisar contantemente la documentación y de definir las normas con las que se va a manejar el documento. | * Establecer lineamientos para la documentación. * Informar a los encargados de realizar la documentación acerca de los lineamientos que deben seguir. |
| **Auditor de documentación** | Es el encargado de apoyar al director de documentación a la hora de revisar que el documento cumpla con las normas establecidas. | * Revisar constantemente el documento para verificar que se cumplan los lineamientos planeados. * Corregir en caso de que no se estén siguiendo los lineamientos. |
| **Director de calidad** | Es el encargado de definir, difundir y mantener la política de calidad y los principios de gestión de calidad. | * Establece lineamientos de calidad. * Revisa contantemente los aportes de los integrantes del equipo. * Comunicar al equipo de trabajo los lineamientos que deben seguir para mantener la calidad. |
| **Auditor de calidad** | Trabaja junto con el director de calidad para asegurar que se cumplan los diferentes requisitos de calidad. | * Detectar fallas, como errores ortográficos, que no se estén siguiendo los lineamientos establecidos. |
| **Administrador de configuración** | Es la persona encargada de controlar las diversas versiones del proyecto y de la documentación. | * Controla las diferentes versiones tanto de documentación como de software. * Determina cuando hay que volver a versiones anteriores. |
| **Administrador** | Junto con el administrador de configuración supervisan las diversas versiones que se pueden presentar. | * Trabaja junto al administrador de configuración para evitar problemas con las diferentes versiones. |

Tabla 6. Roles

# Administración del proyecto

## Métodos y herramientas de Estimación

**Criterios a estimar:**

* **Tiempo:** el estimar tiempos involucra considerar los diferentes procesos para asegurar un buen desarrollo de las distintas actividades, considerando los plazos especificados, así como de las herramientas para el control y seguimiento de la planificación temporal y la programación del proyecto.
* **Esfuerzo.**

**Herramientas de estimación**

Para realizar las estimaciones del proyecto en los diferentes aspectos mencionados anteriormente se utilizaron las siguientes herramientas:

* **Trello:** es una aplicación utilizada para gestionar actividades y entregas con fechas y responsables para cada actividad.
* **Excel:** Microsoft Excel es una herramienta de Office que ofrece una hoja de cálculo en la cual organizar y realizar los cálculos de las estimaciones.
* **Planning Poker App:** es una herramienta que ayuda a la estimación de las historias de usuario usando una metodología ágil [16].

**Estimación de historias de usuario:**

Se usó la estimación de historias de usuario en vista de que es propia de las metodologías agiles como SCRUM que, como se vio anteriormente (ver sección Justificación ciclo de vida), fue seleccionada para el proyecto. Dada la naturaleza de las metodologías ágiles, este método de estimación puede no ser el más estricto y por ende no producir resultados tan precisos como tal vez lo harían otros métodos, sin embargo, se muestra como una alternativa fácil, sencilla e incluso entretenida para efectuar el proceso de estimación de un proyecto pequeño como Fitt.

El proceso de estimación inició proponiendo una serie de historias de usuario, estas son funciones o características que serán incluidas en el transcurso de desarrollo del proyecto. Una vez hecho esto, se asignó un conjunto de tareas a cada historia, esto permitió determinar, de cierta forma, la cantidad de esfuerzo que podía conllevar la implementación de una historia. Luego llego momento de jugar “Planning Poker”, en el que un moderador contrastaba los puntajes asignados a cada tarea por el resto del grupo (Anexo X).

El resultado de la aplicación del método de estimación con puntos de historia colaboró en el proceso de delimitación y definición del proyecto, algunas funcionalidades relacionadas con el manejo de la dieta del usuario, por ejemplo, fueron descartadas. Por otra parte, las historias con más puntos de historia fueron las siguientes:

* “Como cliente quiero crear rutinas para adoptarlas y/o compartirlas”
* “Como cliente quiero llevar un calendario para llevar el control de mi actividad física”
* “Como cliente quiero ver instrucciones para realizar los ejercicios de manera correcta”

Teniendo en cuenta que uno de los entregables importantes (ver secciones entregables) tiene que ver con la implementación del caso de uso más importante, la historia de usuario anteriormente mencionadas son claras candidatas. En total, el costo de la estimación fue de 503 puntos de historia. Sí se asigna una hora de esfuerzo a cada punto de historia, cada miembro del grupo tendría que dedicar 8.4 horas cada semana para terminar el proyecto en el tiempo disponible.

**Estimación con casos de uso:**

La estimación con casos de uso es una forma fácil de estimar el tamaño del proyecto, su premisa es sencilla, toma menos tiempo y esfuerzo codificar un caso de uso sencillo que codificar uno complejo [17]. En adición, este método de estimación tiene en cuenta factores tecnológicos y de entorno.

El resultado de la estimación con casos de uso mostró, contrario a la estimación con historias de usuario, que hay un déficit de tiempo para el proyecto:

|  |  |
| --- | --- |
| UCP | 69 |
| Horas por UCP | 20 |
| Total horas | 1389 |
| Semanas disponibles | 12 |
| Horas por semana por cada miembro del equipo | 23 |

Tabla 7. Estimación por casos de uso

Como se muestra en la tabla se obtuvo un total de 71 UCP (puntos de casos de uso), siguiendo el proceso de estimación, cada UCP representa veinte (20) horas de trabajo. El número de horas de trabajo por cada miembro del equipo en la semana que entrega la estimación es bastante alto en relación al tiempo que se tiene disponible, lo que indica que es necesario reducir las partes funcionales del proyecto, algunos casos de uso candidatos por su baja prioridad son:

Consultar lista de metas.

Consultar guías de rutinas con información audiovisual.

Esta estimación puede no ser la más acertada para el proyecto, teniendo en cuenta que las reglas para determinar el número de transacciones entre el sistema y el cliente en un proyecto son muy imprecisas. Este método se usó dado que el grupo está acostumbrado a trabajar con casos de uso, no obstante, no es muy recomendable en metodologías ágiles [17].

## Inicio del proyecto

### Planes de capacitación

El inicio del proyecto supone que todos los miembros del equipo de trabajo puedan familiarizarse con las herramientas que se utilizaran en el proceso de creación de Fitt. Algunas habilidades y conocimientos serán adquiridos en el transcurso de las clases de ingeniería de software otras, en cambio, deben ser estudiadas y profundizadas fuera del aula de clase. Del mismo modo, la asignatura de ingeniería de software tiene una serie de prerrequisitos que indican que el grupo tiene un conjunto de capacidades que facilitarán el debido desarrollo del proyecto, entre ellos están: análisis y diseño orientado a objetos, sistemas de información y programación orientada objetos.

Dos miembros del equipo tienen experiencia en Android Studio y Firebase, estos se encargarán de dirigir la capacitación de los demás miembros del equipo, no obstante, cada uno deberá informarse y profundizar acerca del uso de estas herramientas. A continuación, se muestra una tabla con las capacitaciones planeadas al inicio del proyecto:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Objetivo | Encargado(s) | Fecha |
| Instalación Android Studio | Resolver dudas y arreglar inconvenientes en la instalación de Android Studio | Jhonny Parra y Sebastián Triana | 15/02/2019 |
| Uso de herramientas de edición de texto | Aprender a usar algunas funcionalidades disponibles en Word que facilitan el proceso de escritura de los documentos | Miguel Beltrán | 24/02/2019 |
| Uso de Trello | Discutir cómo se usará Trello en el proyecto | Todo el grupo | 24/02/2019 |
| Uso de herramientas para control de cambios | Aprender a utilizar GitHub | Profesor Carlos Parra | 26/02/2019 |
| Funcionalidades básicas de Android Studio | Comprender el funcionamiento básico de Android Studio | Jhonny Parra y Sebastián Triana | 01/03/2019 |
| Funcionalidades básicas de Firebase (Sesión 1) | Comprender el funcionamiento básico de Firebase (Bases de datos no relacionales) | Jhonny Parra y Sebastián Triana | 08/03/2019 |
| Funcionalidades básicas de Firebase (Sesión 2) | Comprender el funcionamiento básico de Firebase (Integración con Android Studio) | Jhonny Parra y Sebastián Triana | 15/03/2019 |

Tabla 8. Capacitaciones

Los encargados de cada una de las capacitaciones ofrecerán al resto de los integrantes del grupo un espacio en el que resolver sus dudas acerca de cada una de las temáticas, sin embargo, cada miembro debe estar constantemente investigando y entrenando sus habilidades en el uso de las herramientas que se usarán en el proyecto. Cada sesión de capacitación tomará una hora y se llevará a cabo en el lapso de las reuniones virtuales o presenciales contempladas.

* Indicar tareas a realizar para mantener la infraestructura del proyecto: obtener las herramientas, ponerlas en marcha para el proyecto y mantenerlas durante el mismo, así como qué roles son los responsables de cada tarea, cuándo se realizará cada tarea y con qué frecuencia.

## Planes de trabajo del proyecto

**Descomposición de actividades (WBS)**

A continuación, se presenta un diagrama para mostrar la división de actividades del proyecto, para la descomposición de las actividades se tuvo en cuenta los diferentes aspectos a tratar en la entrega.

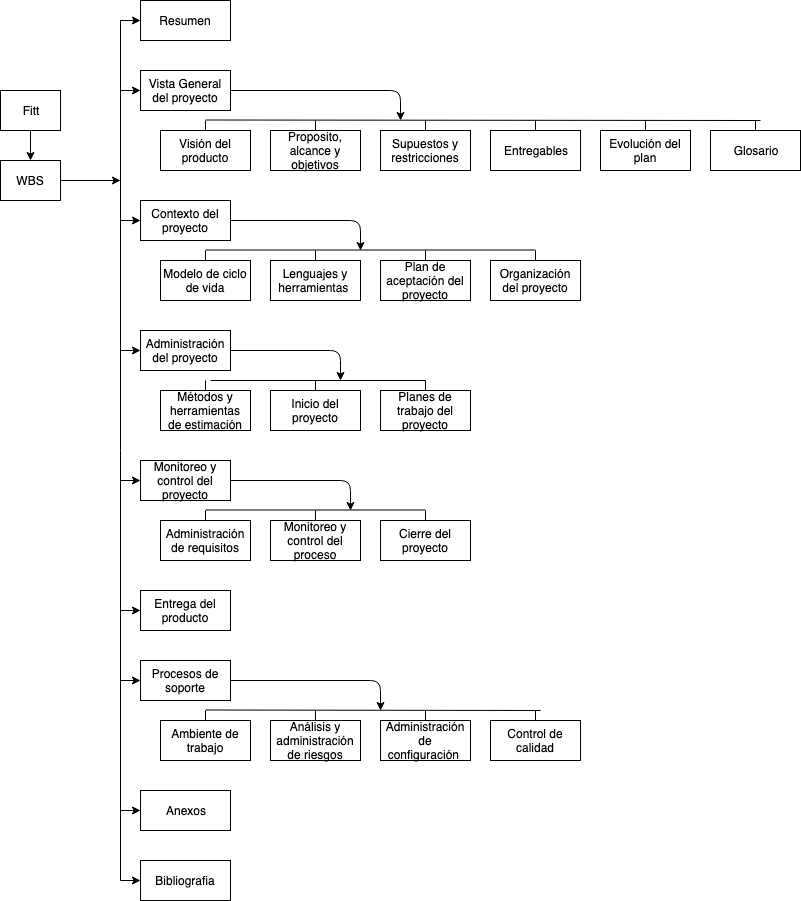


Ilustración 2. WBS

**Calendarización**

Teniendo en cuenta la metodología que se va a manejar, se planea tener diversas reuniones por medio de la aplicación de Discord y una reunión semanal los días lunes en el horario de 11-1 para conocer los avances que se han hecho semanalmente por parte de cada uno de los integrantes del grupo, al igual conocer problemas que se hayan presentado con las actividades asignadas y determinar si ya se solucionaron o si se necesita plantear una solución en equipo. Para tener un control de las actividades hechas, de las que se encuentran en proceso y de las que faltan por hacer se tiene un tablero en Trello con las fechas de entrega de cada una de las actividades.

Para el caso de cada entrega que se deba presentar al cliente, se va a manejar a partir de exposiciones donde se presente un resumen con las actividades realizadas y se espera una retro alimentación por parte del cliente, donde se presenten correcciones para ser mejoradas para las entregas futuras.

Para la entrega final se tiene una presentación junto con una demostración de la aplicación final, donde se verifica si se cumplieron los requisitos planteados desde un principio, y si existe satisfacción por parte del cliente.

**Presupuesto del proyecto.**

Para conocer el presupuesto con el que se cuenta para el proyecto Fitt, se tienen en cuenta costos como la mano de obra que presenta el grupo Bittus, el costo de movilizarse hasta la universidad para las reuniones, al igual que el costo que presentaron cada uno de los dispositivos de hardware con los que se va a trabajar.

Para el caso de los costos de mano de obra por parte de bittus se tienen en cuenta el número de horas que se va a trabajar por integrante a la semana, el salario que tiene un programador junior en Colombia, según la revista dinero “*En promedio este cargo en Colombia gana US$2.400 dólares, unos $6 millones” y “el 10% que está en el fondo de la tabla de ingresos gana menos de US$1.000 mensuales, es decir menos de $2,5 millones”,* para este caso se toma que el programador junior gane alrededor de 2,4 millones por lo tanto cada hora cuesta 15.000 basándonos en esto obtenemos la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| **Empleado** | **Horas trabajadas (semana)** | **Salario semana** | **Salario mes** |
| Miguel Ángel Beltrán | 8 | 120.000 | 480.000 |
| Miguel Ángel Chaves | 8 | 120.000 | 480.000 |
| Laura Juliana Mora | 8 | 120.000 | 480.000 |
| Jhonny Alexander Parra | 8 | 120.000 | 480.000 |
| Juan Sebastián Triana | 8 | 120.000 | 480.000 |
| **TOTAL** | 40 | 600.000 | 2.400.000 |

Tabla 9. Nomina

Se estima que cada integrante trabaje 8 horas por semana para el proyecto y se tiene un estimado de que el proyecto dure 12 semanas, teniendo así un presupuesto de **7.200.000 COP**, finalmente nos basamos en que cada mes tenga 4 semanas, según esto se tiene un presupuesto de mano de obra mensual de **$2.400.000 COP**.

Como se mencionó anteriormente entre los costos se consideró el costo del trayecto hasta la universidad, para esto se considera el precio desde troncales de los servicios de transporte publico de Bogotá, a la fecha el pasaje cuesta $2.400 cop, considerando esto se obtiene la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| **Empleado** | **Cantidad de pasajes semana** | **Precio pasajes semana** | **Precio pasajes mes** |
| Miguel Ángel Beltrán | 2 | 4.800 | 19.200 |
| Miguel Ángel Chaves | 2 | 4.800 | 19.200 |
| Laura Juliana Mora | 2 | 4.800 | 19.200 |
| Jhonny Alexander Parra | 2 | 4.800 | 19.200 |
| Juan Sebastian Triana | 2 | 4.800 | 19.200 |
| **TOTAL** | 10 | 24.000 | **240.000** |

Tabla 10. Costos de transporte

Como se mencionó anteriormente se espera tener 12 semanas para la elaboración del proyecto, a razón de esto se espera un presupuesto de **288.000 cop** para todo el proyecto.

Finalmente se consideró el costo del hardware utilizado para el proyecto, para esto cada integrante tiene su computador, ya sea de escritorio o portátil con los siguientes precios:

* Computador de mesa (sin modelo de referencia): **1.500.000cop**
* Lenovo G400s: **1.200.000 cop**
* Vaio E11: **1.100.000 cop**
* Lenovo G50-30: **950.000 cop**
* MacBook pantalla retina gold: **4.250.000 cop**

Tras esto se tiene un presupuesto de **9.000.000 cop** para el hardware.

Tras considerar los diferentes costos se obtiene la siguiente tabla con las estimaciones de cada componente

|  |  |
| --- | --- |
| **Componente** | **Presupuesto** |
| Mano de obra | 7.200.000 |
| Hardware | 9.000.000 |
| Transporte | 288.000 |
| **Total** | 16.488.000 |

Tabla 11. Total de presupuesto del proyecto

Con esto se estima un presupuesto de **$16.488.000 cop** para las 12 semanas de proyecto.

# Monitoreo y control del proyecto

## Administración de requisitos

Las acciones de administrar, detectar, reportar y controlar los requisitos para Fitt, marcaran la guía hacia la solidificación del proyecto; por tanto, es necesario que Bittus defina un procedimiento de los cambios referentes a este aspecto. El proceso de administración de requisitos está compuesto de 4 pasos, sobre los cuales se adaptará y adecuara el nuevo requisito; cabe remarcar que los requisitos que se definieron al inicio del proyecto no pasaran a través de este proceso ya que estos han sido atacados con anterioridad.

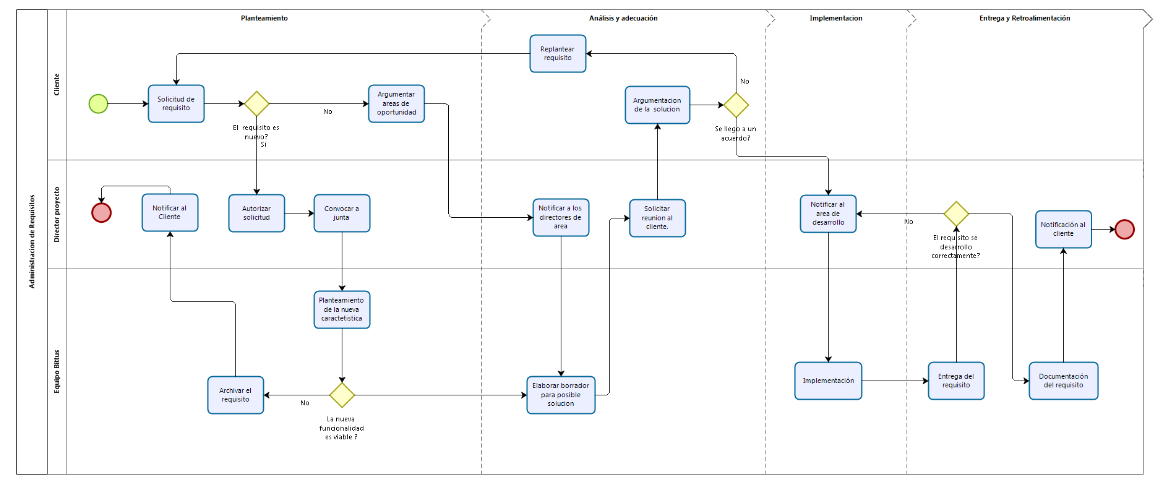
1. **Planteamiento:** Esta fase empieza en el momento en el que el cliente pretende realizar una solicitud de cambio o adición de un requisito. Dicha persona tendrá que exponer las razones por las cuales considera necesario dicho cambio o adición en una reunión entre el director del proyecto y el, para luego convocar a una reunión con el resto de los integrantes del grupo de trabajo donde se determinará si dicho cambio es viable o no para el proyecto en general; en caso de que este sea aceptado pasara a la siguiente fase de lo contrario será archivada y descartada.

1. **Análisis y adecuación:** Tras ser aprobada, se elaborará un primer borrador de ejecución, donde se evalúan todas las implicaciones que conllevara realizarla, analizando de la mejor forma para el tiempo dado. Como resultado se programa una reunión con el cliente donde se le explicará el resultado esperado y las consecuencias de la implementación del mismo, luego de analizar estos aspectos, podrá llegarse a un acuerdo sobre la petición o en último caso reformular el requisito y el cliente volver a postularlo al director del proyecto.

1. **Implementación:** El grupo de trabajo encabezado por del director del proyecto es convocado a una reunión donde se expondrá el plan de acción a seguir. Durante la reunión cada área generara las nuevas tareas asociadas y empezara la marcha del requisito.

1. **Entrega y retroalimentación:** Cada vez que se quiera mostrar un avance o el requisito terminado, se tendrá que consultar al director del proyecto antes que, al cliente, ya que posee la autoridad de dar por terminado el requisito o no, en caso de ser satisfactorio el resultado, se mandara a documentar para así finalmente notificar al cliente que se ha desarrollado a la perfección.

**Diagrama BPMN de este proceso**.



## Monitoreo y control del progreso

A medida que se realizan las diferentes tareas propuestas para el desarrollo del proyecto surge la necesidad de medir su progreso. La finalidad de estas mediciones es la de conocer factores importantes como son los tiempos necesarios para terminar tareas, el proyecto, etc. De esta manera se pueda obtener una percepción clara del desarrollo de este. A continuación, se presentarán las medidas y actividades para monitorear y controlar el progreso del proyecto.

**Unidades para medir el progreso**

La metodología ágil seleccionada (SCRUM) permite establecer dos unidades de medida para evaluar el progreso del proyecto con base a las tareas del tablero y los intervalos durante el cual se hacen incrementos del producto.

* **Tareas realizadas:** Esta es la principal medida del proyecto pues permite evidenciar el avance de la entrega con base en cada una de las tareas, esta permite medir el avance tanto para el documento como para el desarrollo del proyecto se calcula a través de la siguiente ecuación:

Ilustración . Ecuación para calcular el avance de las tareas

Este resultado expresa en términos porcentuales lo completado del proyecto.

* **Velocidad:** Esta medida permite estimar la cantidad de trabajo que podemos hacer en los próximos sprints, además, ofrece un aproximado de la fecha de finalización del proyecto en términos de la cantidad de sprints faltantes para terminar. Esta se calcula a través promediando el número de puntos de historia completados en los sprints anteriores.

Esta última unidad es sacada de otra metodología ágil llamada Kanban la cual está orientada a la gestión de proyectos especialmente complejos, dentro de esta se encuentra la siguiente medida:

* **Lead time:** Esta mesura tiene como propósito medir el tiempo comprendido entre el momento en que se inicia una tarea y el momento de su entrega. Esta servirá para complementar la medida anterior determinando cuanto nos estamos demorando y si debemos agilizar el desarrollo de tareas.

**Actividades para reportar progreso**

Con el fin de retroalimentar el grupo de trabajo con los avances realizados en el documento y en el desarrollo del código se plantearon las siguientes actividades:

* **Reuniones semanales:** Se realizarán reuniones semanales de duración mínima de cuatro horas ya sean virtuales o físicas. En ellas se tratarán temas como el avance realizado. Una tarea realizada será tomada como implementada exitosamente una vez esta sea desplazada a la columna de terminados, además, se dará control a las actividades propuestas midiendo continuamente semana a semana el porcentaje realizado y la velocidad con la que se trabaja. Las personas encargadas de llevar acabo esta actividad son Jhonny Parra junto con Sebastián Triana
* **Revisiones a los tableros de Trello:** Cada vez que se termine o se comience una tarea se ordenará las tarjetas en sus debidas columnas, Trello permitirá observar desde una vista general el avance general, por otro lado, al ser una plataforma publica, mantendrá al grupo constantemente informado acerca del desarrollo de tareas. Las personas responsables del cumplimiento de esta actividad es Miguel Ángel Beltrán ayudado de los demás compañeros de trabajo.

**Acciones correctivas**

Si las tareas se atrasan generando que el proyecto no vaya según lo esperado se tendrán una serie de estrategias que permitirán tomar la mejor decisión para volver a encaminar el proyecto hacia lo ideal. A continuación, se enuncian estas estrategias:

* En el caso en que esto suceda debido a la falta de responsabilidad y compromiso de un miembro del grupo se discutirá con los miembros del equipo cual será la sanción, la gravedad de esta dependerá de la cantidad de trabajo que no haya sido realizado, siguiente a esto se seleccionara a las personas que tengan mayor tiempo disponible (que hayan terminado su parte del trabajo) para que ayuda en al desarrollo de la tarea, en caso de que ninguno de los integrantes pueda brindar una ayuda, cada uno de los integrantes del equipo dedicaran un poco de su tiempo para adelantar la tarea atrasada. En cuestión de los procesos de calidad tendrán que minimizarse para que la tarea con problemas no traiga más retrasos al proyecto.
* En el caso de que sea por fallos en el momento de la planeación, el equipo discutirá la mejor solución durante las reuniones semanales, estas propuestas dependerán totalmente de la tarea, siguiente a esto se buscaran las personas que estén disponibles para que apoyen el desarrollo de la tarea mal planificada y como en el caso anterior si no hay personas disponibles todos los miembros del grupo deberán sacar un tiempo para poder atender la urgencia, la diferencia entre estas 2 está en la persona disponible para ayudar, esta será recompensada con un dulce presupuestado por todos los integrantes. En cuestión de los procesos de calidad tendrán que minimizarse para que la tarea con problemas no traiga más retrasos al proyecto.

## Cierre del proyecto

Para cada entrega del proyecto, en la que esté involucrada el cliente con Bittus, una vez acabado todo lo correspondiente a la entrega y después de finalizada cada una de estas, se convocará a una reunión, en donde se discutirán las acciones correspondientes a realizar consecuencia de las observaciones creadas por ambas partes, así mismo se realizará una inspección final de calidad a cada una de las secciones hechas por los miembros del equipo bajo los parámetros establecidos en la sección **[11.4 - Control de Calidad].**

Para elaborar el reporte gerencial y el análisis post-mortem de la entrega final, se tomará en cuenta toda la información recopilada durante cada una de las reuniones semanales realizadas a lo largo del semestre, el desempeño de cada integrante en cuanto a la productividad, el trabajo realizado, la participación y los recursos e insumos utilizados por cada uno y de igual forma, mediante la opinión de cada uno de los integrantes de Bittus se usaran para dar a conocer una retrospectiva general de cómo fue planeado y ejecutado el trabajo, esto con el fin de identificar fortalezas, fomentar el crecimiento personal y en equipo, áreas de oportunidad y acciones de mejora; con las cuales se pretende establecer nuevos lineamientos a tener en cuenta en el trabajo restante y futuro de la aplicación Fitt.

# Entrega del producto

La entrega final de Fitt como aplicación móvil estable, estará segmentada en cuatro pilares esenciales, la documentación formal del proyecto, el software funcional, la sustentación final del mismo y por último la retroalimentación al proyecto. Cada uno de estos serán entregado conforme a lo establecido en los puntos [**6.4 - Entregables**] y [**7.3 plan de aceptación del producto**] y por el responsable correspondiente.

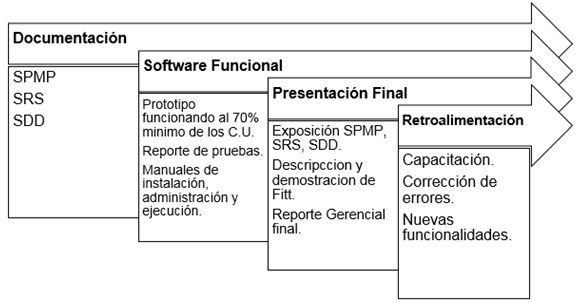


Ilustración 4. Entrega del producto

En cuanto a la documentación es referido el presente documento (SPMP); la Especificación de Requisitos de Software (SRS); por último, la Descripción del Diseño del Software (SDD); además de estos, es importante resaltar y suministrar aquellos documentos que soportan las funcionalidades del prototipo a presentar, así como los que complementan y fundamentan cada entregable dado.

Para el segundo pilar se agrupan aquellos componentes que dan forma y solidifican al software funcional, en este caso para Fitt. Consecuentemente siguen todas aquellas actividades que conformarán la sustentación del proyecto entre el equipo de trabajo y el cliente; para así finalizar con el ultimo componente correspondiente a la retroalimentación en el cual se atenderán aspectos de mejoras y correcciones de Fitt.

# Procesos de soporte

## Ambiente de trabajo

Gran cantidad de los proyectos de desarrollo de software implican trabajo en grupo pues se necesita de una gran gama de conocimientos e ideas para desarrollar un proyecto de calidad, al mismo tiempo, esto permite simplificar tareas y agilizar procesos.

Nunca trabajamos solos, siempre hay un compañero cerca, en este contexto lo mejor es llevarse bien y sacar provecho del potencial de tener a alguien al lado. El establecimiento de reglas y mecanismos de trabajo aseguran el cumplimiento y control de tareas conjuntamente con una buena convivencia en el grupo.

**Reglas y mecanismos para regular y asegurar el cumplimiento de trabajos**

* **Seguimiento de tareas mediante la plataforma Trello:** La plataforma de Trello brinda la facilidad de hacer un seguimiento mediante listas de tarjetas, en ellas se plantea la metodología ágil SCRUM la cual permite mover diferentes tarjetas que contienen historias y tareas a las diferentes listas con el fin de observar el avance en la ilustración 1 se puede apreciar las listas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Historias | Para hacer | En proceso | En revisión | Hecho |

Ilustración . Listas del tablero de SCRUM.

Miguel Ángel Beltrán Rodríguez será el encargado de ir supervisando las tarjetas en la plataforma, en adición, revisará que las tareas e historias estén en sus respectivas listas. Los demás integrantes del grupo también ayudarán en este seguimiento pues moverán las tarjetas con tareas o historias a su correspondiente estado en las listas. En caso de que el integrante no actualice su actividad Miguel será el encargado de realizar un llamado de atención al integrante del grupo que no acate esta medida.

* **Reuniones físicas al menos una vez por semana:** Cada semana se realizará una reunión de máximo cuatro horas por semanas donde se hablarán temas acordes a las tareas y actividades propuestas para esa semana, se discutirán temas como los estados, tiempos, dificultades, posibles soluciones, etc. En adición, si se acaba antes del tiempo estipulado, se aprovechará el tiempo escribiendo código o haciendo las revisiones respectivas a las tareas cuyo proceso ya fue terminado, Juan Sebastian Triana será encargado de supervisar la asistencia a las reuniones y el cumplimiento en cuestiones de la trata de los temas, por otro lado, los integrantes del grupo se encargarán de aportar a las discusiones complementando los temas o ayudado a los demás compañeros. El plazo máximo de llegada tarde es de 15 minutos, después de esto se contará como una falta en donde el infractor deberá traer refrigerio a los demás integrantes del grupo, cada excusa para inasistencia a las reuniones deberá ser revisada por el grupo y aprobada por todos, preferiblemente las excusas deben ser citas médicas, enfermedad, razones de fuerza mayor, eventos respaldados por entidades formales como la universidad.
* **Reuniones virtuales:** Cada semana se realizará una reunión de máximo cuatro horas por semanas donde se hablarán temas acordes a las tareas y actividades propuestas para esa semana, a diferencia de las reuniones físicas están se realizan por medio de internet mediante la plataforma Discord. Esta herramienta permite comunicar un grupo grande de personas con la facilidad de separarlos por canales, es decir, permite separar el grupo en diferentes salas. El plazo máximo de llegada tarde es de 15 minutos, después de esto se contará como una falta en donde el infractor deberá disculparse con los demás integrantes del grupo, cada excusa para inasistencia a las reuniones virtuales deberá ser revisada por el grupo y aprobada por todos, preferiblemente las excusas deben ser citas médicas, enfermedad, razones de fuerza mayor, eventos respaldados por entidades formales como la universidad.
* **Historial de cambios:** Para hacer seguimiento y regulación del documento, este trae la sección de historial de cambios donde se mostrará la fecha, el autor y la descripción del cambio realizado. Es deber de todos los integrantes actualizar esta tabla pues demuestra el trabajo realizado en el documento además de ser parte de este.

## Análisis y administración de riesgos

**Propósito**

Identificar los principales riesgos asociados al proyecto. Definir acciones de prevención y mitigación de dichos riesgos, así como los recursos y responsables. Definir cuántos recursos y tiempo invertir en dichas acciones, de acuerdo a la prioridad de cada riesgo.

**Contenido**

* Descripción del plan de riesgos que se aplicará: responsable, actividades, herramientas, etc.
* Descripción del proceso de identificación de riesgos: diagrama BPMN con su explicación.
* Descripción de los momentos en el proyecto en que se realizará dicho proceso.

Los anexos que se deben mantener y actualizar para todas las entregas son:

* Tabla con los riesgos del proyecto que han sido identificados, indicando probabilidad de ocurrencia e impacto. Los riesgos deben estar ordenados por prioridad, la cual es definida a partir de esos dos factores.
* Tabla con los riesgos más importantes, indicando acciones de prevención y mitigación, así como los responsables y recursos necesarios.

**Tamaño sugerido**

2 páginas máximo

**Referencias**

[1], sección 5.8.3.

## Administración de configuración y documentación

**Propósito**

identificar los principales ítems de configuración y cómo evolucionarán durante el tiempo, particularmente los artefactos de documentación y el código.

**Contenido**

* Descripción de los ítems de configuración identificados.
* Diagrama en BPMN (y su correspondiente explicación) del proceso de control de cambios a los ítems de configuración.
* Tabla que describa los artefactos de documentación y código, y en qué momento se crean o se refinan durante el proyecto.

**Tamaño sugerido:**

2 páginas.

**Referencias**

[1], sección 5.8.4.

Durante el desarrollo del proyecto se puede apreciar una gran cantidad de ítems de configuración basados tanto en el desarrollo del código como en la documentación. Estos poseen diferentes artefactos que soportan el desarrollo de cada uno de los procesos, a continuación, se mostraran como estos ítems se comportan a lo largo del proyecto conjunto a sus respectivos artefactos.

**Identificación de ítems de configuración**

En el desarrollo del proyecto se identificaron los siguientes ítems de configuración:

1. Diagrama de casos de uso.
2. Diagrama de clases.
3. Modelo entidad relación.
4. Código fuente.
5. Software Project Management Plan o Plan de Administración de Proyecto (SPMP, por sus siglas en inglés).
6. Software Requirements Specification o Especificación de Requerimientos de Software(SRS, por sus siglas en inglés).
7. Software Design Description o Descripción de Diseño del Software(SDD, por sus siglas en inglés).
8. Manual de usuario.

## Control de calidad

**Propósito**

Que los integrantes conozcan las actividades relacionadas con todos los procesos de control de calidad del proyecto.

**Contenido**

* Explicación de alto nivel de todos los procesos asociados al control de calidad, lo cual incluye verificación, validación, revisiones y auditorías al proyecto. Por ejemplo: revisiones de calidad de documentos, pruebas unitarias, de sistema, revisiones externas, etc. Para cada proceso indicar: nombre del proceso, momentos del proyecto en que ocurren y quiénes son los responsables. Se recomienda resumir esta información en una tabla.

Describir el detalle de los procesos de control de calidad, usando BPMN y párrafos explicativos.

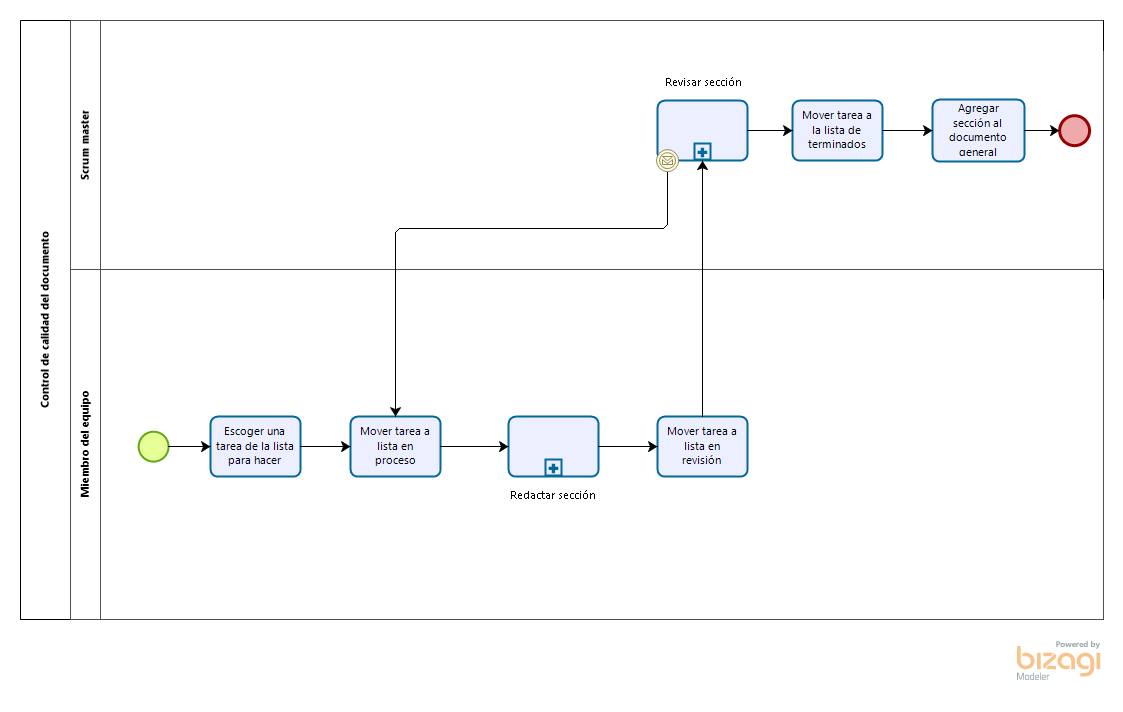
**Tamaño sugerido**

2 páginas.

**Referencias**

[1], secciones 5.8.8 y 5.8.9.

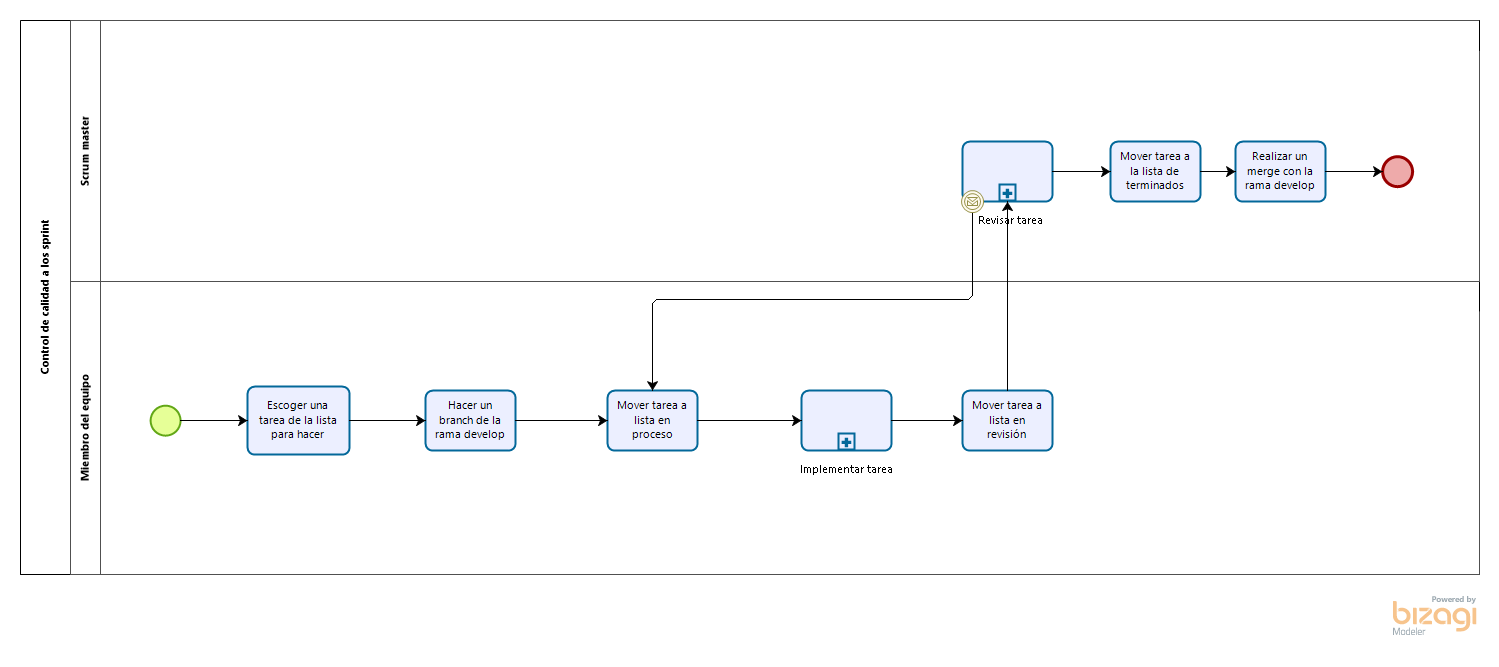
**Diagrama BPMN de control de calidad del documento**

****

Descripción…

En los anexos se encuentran los diagramas BPMN de los subprocesos del diagrama anterior subprocesos.

**Diagrama BPMN de control de calidad de los sprints**

****

Descripción…

En los anexos se encuentran los diagramas BPMN de los subprocesos del diagrama anterior subprocesos.

# Anexos

Agregue aquí cualquier información adicional relevante para el proyecto para que sea referenciada desde las secciones previas.

Es importante tener en cuenta que, hasta antes de esta sección, el documento debe ser autocontenido, es decir, el lector no debe necesitar leer ningún documento adicional o archivo externo para entender el proyecto. La información que se coloque aquí es simplemente un complemento, en caso de que el lector requiriera más detalles sobre algún tema.

Cartas Gantt, diagramas y otros elementos similares no deben ir en esta sección, sino en el lugar que se indique dentro del documento.

# Bibliografía

|  |  |
| --- | --- |
| [1] | M. García Manso, M. Navarro Valdivielso y J. Ruiz Caballero, Bases teóricas del entrenamiento deportivo, Madrid: Gymnos, 1996. |
| [2] | A. Paoli, A. Bianco, M. Neri y A. Palma, What is Fitness:definition, history and helath benefits. A review, Italian Journal of Sport Sciences and Law, 2008. |
| [3] | Google, «Android,» DigitalGlobe, 2014. [En línea]. Available: https://www.android.com/. [Último acceso: 25 02 2019]. |
| [4] | ProyectosAgiles, «proyectosagiles,» [En línea]. Available: https://proyectosagiles.org/que-es-scrum/. [Último acceso: 25 02 2019]. |
| [5] | K. Beck, Extreme Programming Explained, Boston: Addison Wesley, 2004. |
| [6] | Agile Business Consortium Limited, «Agile Business Consortium,» 2019. [En línea]. Available: https://www.agilebusiness.org/what-is-dsdm. [Último acceso: 24 02 2019]. |
| [7] | Techopedia Inc, «Techopedia,» 2019. [En línea]. Available: https://www.techopedia.com/definition/5508/enterprise-unified-process-eup. [Último acceso: 24 02 2019]. |
| [8] | ICTEA, «W-ictea,» 2019. [En línea]. Available: http://www.ictea.com/cs/knowledgebase.php?action=displayarticle&id=8790. [Último acceso: 23 02 2019]. |
| [9] | Google, «Android Developers,» [En línea]. Available: https://developer.android.com/studio/intro/?hl=es-419. [Último acceso: 24 02 2019]. |
| [10] | Google, «Firebase,» 20 02 2019. [En línea]. Available: https://firebase.google.com/docs/database/?hl=es-419. [Último acceso: 25 02 2019]. |
| [11] | Google, «Google Cloud,» [En línea]. Available: https://cloud.google.com/maps-platform/?hl=es. [Último acceso: 24 02 2019]. |
| [12] | Google, «Google Cloud,» [En línea]. Available: https://cloud.google.com/text-to-speech/?hl=es. [Último acceso: 24 02 2019]. |
| [13] | ITM platform, «Itmplatform,» 28 03 2016. [En línea]. Available: http://www.itmplatform.com/es/blog/rol-de-un-director-de-proyecto/. [Último acceso: 25 02 2019]. |
| [14] | M. André Ampuero y Y. López Trujillo, «ROLES EN EL PROCESO DE DESARROLLO DE SOFTWARE PARA LAS EMPRESAS CUBANAS,» 2006. [En línea]. Available: https://www.redalyc.org/html/3604/360433560012/. [Último acceso: 25 02 2019]. |
| [15] | J. Pérez, «IDA,» 2015. [En línea]. Available: https://blog.ida.cl/diseno/que-hace-disenador-interfaces/. [Último acceso: 2019 02 25]. |
| [16] | Google, «Google Play,» 2019. [En línea]. Available: https://play.google.com/store/apps/details?id=br.org.cesar.planningpoker&hl=en. [Último acceso: 25 02 2019]. |
| [17] | M. Cohn, «Estimating With Use Case Points,» *Mountain Goat Software,* p. 12. |

Sección 7.2

<https://www.deustoformacion.com/blog/programacion-diseno-web/que-es-para-que-sirve-github> -> github //está referencia no creo que sea válida.

Sección 8.1

<https://sites.google.com/site/gestiondeproyectossoftware/unidad-3-planificacion-de-proyecto/3-2-estimaciones-de-tiempo> -> tiempo //está sí que no

Sección 7.4

<https://www.recursosenprojectmanagement.com/roles-en-la-gestion-de-riesgos/> -> director y analista de riesgos

<https://sites.google.com/site/recuperaciongestion/tema-1/12-4-conocer-las-funciones-del-director-de-calidad> -> director de calidad //este no

Sección 9.2

<http://managementplaza.es/blog/la-velocidad-sprint-scrum/> -> Velocidad

<https://uv-mdap.com/blog/metricas-en-el-kamban-algo-mas-de-metodologias-agiles/> -> Lead time Kanban