

**GDEVLINK**

VERSIÓN 2.0.6

PLAN DE GESTIÓN DE PROYECTO DE SOFTWARE



JUAN PABLO MÉNDEZ PERDOMO

DANIEL HERNÁNDEZ GARCÍA

JUAN CARLOS SUÁREZ JAIMES

DIEGO ANDRÉS BURGOS MELO

GABRIEL GÓMEZ CORREDOR

SANTIAGO ANDRÉS CAROPRESE HIDALGO

16 DE OCTUBRE DE 2020

# Historial de cambios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versión | Fecha de cambio | Secciones Modificadas | Descripción | Responsables |
| 1.0.0 | 29/08/2020 | 1 | Se creó el historial de cambios. | Juan Carlos Suárez |
| 1.0.1 | 29/08/2020 | 11.1 | Se agregaron las reglas para regular el trabajo en equipo. | Todos |
| 1.0.2 | 30/08/2020 | 6.1 | Se agregó la visión del producto | Juan Carlos Suárez |
| 1.0.3 | 30/08/2020 | 7.4 | Creación y realización de Organigrama y descripción de roles. (7.4.2.) | Daniel Hernández |
| 1.0.4 | 30/08/2020 | 6.4 | Descripción de los entregables y creación de la tabla de entregas. | Gabriel Gómez Corredor |
| 1.0.5 | 30/08/2020 | 7.3 | Realización del plan de aceptación del producto | Daniel Hernández |
| 1.0.6 | 31/08/2020 | 7.1 | Descripción de las alternativas consideradas. | Santiago Caroprese Hidalgo |
| 1.0.7 | 31/08/2020 | 6.5 | Descripción de la evolución del plan. | Gabriel Gómez Corredor |
| 1.0.8 | 31/08/2020 | 7.4.1 | Tabla de interfaces externas | Daniel Hernández |
| 1.0.9 | 31/08/2020 | 8.2 | Inicio de Proyecto | Juan Pablo Méndez |
| 1.0.10 | 31/08/2020 | 11.4 | Control de calidad | Diego Burgos, Juan Pablo Méndez |
| 1.0.11 | 1/09/2020 | 6.3 | Supuestos y restricciones | Diego Burgos |
| 1.0.12 | 2/09/2020 | 7.2 | Descripción Lenguajes y Herramientas | Santiago Caroprese Hidalgo y Daniel Hernández |
| 1.0.13 | 3/09/2020 | 6.2 | Se agregaron propósito, alcance y objetivos | Juan Carlos Suárez |
| 1.1.0 | 4/09/2020 | 10 | Se describió el proceso de transición de la entrega de producto | Gabriel Gómez Corredor |
| 1.1.1 | 4/09/2020 | 9.3 | Cierre del proyecto | Juan Pablo Méndez |
| 1.1.2 | 5/09/2020 | 7.1 | Se redactó justificación de ciclo de vida escogido | Juan Carlos Suárez |
| 1.1.3 | 5/09/2020 | 11.2 | Se describieron los procesos de planeación y ejecución de análisis de riesgos | Gabriel Gómez Corredor y Diego Burgos Melo |
| 1.1.4 | 6/09/2020 | 8.3.1 | Se agregó estructura de descomposición de tareas. | Daniel Hernández y Juan Carlos Suárez |
| 1.1.5 | 6/09/2020 | 8.3.2 | Se agregó descripción de las actividades y de su orden de precedencia | Daniel Hernández y Juan Carlos Suárez |
| 1.1.6 | 6/09/2020 | 9.2 | Se agregó contenido de monitoreo y control del progreso. | Santiago Caroprese y Juan Carlos Suárez |
| 1.1.7 | 6/09/2020 | 8.3.3 | Se agrego el flujo de caja del proyecto. | Santiago Caroprese |
| 1.1.8 | 6/09/2020 | 9.1 | Se agrego la administración de requisitos. | Santiago Caroprese |
| 1.1.9 | 7/09/2020 | 6.6 | Creación y desarrollo de la tabla de Glosario | Daniel Hernández |
| 1.1.10 | 7/09/2020 | 8.1 | Métodos y herramientas de estimación | Diego Burgos, Juan Pablo Mendez |
| 1.1.11 | 8/09/2020 | 11.3 | Administración de configuración y documentación | Juan Pablo Méndez |
| 1.1.12 | 8/09/2020 | 8.3.2 | Revisión y mejoras de descripción y relaciones de actividades. | Juan Carlos Suarez, Daniel Hernández |
| 1.1.13 | 8/09/2020 | 13 | Creación y actualización de referencias | Daniel Hernández |
| 1.1.14 | 8/09/2020 | 2 | Resumen | Juan Carlos Suárez |
| 2.0.0 | 6/10/2020 | 11.2 | Actualización de anexo análisis de riesgo y corrección de análisis. | Gabriel Gómez Corredor y Diego Burgos Melo |
| 2.0.1 | 16/10/2020 | 8.1 | Actualización de métodos y herramientas de estimación. | Diego Burgos , Juan Pablo Méndez. |
| 2.0.2 | 16/10/2020 | 8.2 | Actualización de inicio de proyecto. | Juan Pablo Méndez. |
| 2.0.3 | 16/10/2020 | 11.4 | Actualización de control de calidad | Diego Burgos , Juan Pablo Méndez. |
| 2.0.4 | 16/10/2020 | 6.6 | Nuevos términos agregados al glosario | Juan Carlos Suárez |
| 2.0.5 | 16/10/2020 | 8.3.1 | Se arreglaron los WBS | Juan Carlos Suárez |
| 2.0.6 | 16/10/2020 | 8.3.2 | Se corrigió diagrama de Gantt y se actualizaron historias de usuario | Juan Carlos Suárez |
| 2.0.6 | 31/10/2020 | 8.1 | Actualización de nuevas historias de usuario en el anexo y estimaciones. | Diego Burgos |

# Resumen

Este documento incluye la descripción de los componentes del SPMP del proyecto GDevLink, que ilustran el proceso de planeación llevado a cabo para el desarrollo de este proyecto de software. El propósito de este proyecto es desarrollar una aplicación web que sirva como plataforma a través de la cual los desarrolladores de videojuegos aficionados puedan intercambiar conocimiento entre sí, seguir el proceso de desarrollo de proyectos de videojuegos de su interés, encontrar y aplicar para posiciones vacantes en proyectos en los cuales puedan participar y crear sus propios proyectos para formar equipos de desarrollo junto a otros desarrolladores. El alcance de este proyecto abarca la implementación de las funcionalidades de usuario, proyecto, posiciones vacantes y sección de preguntas que permiten realizar este propósito, excluyendo mecanismos de comunicación directa entre usuarios, como podrían ser chats o mensajes directos.

La audiencia esperada del documento son aquellas personas que deseen obtener una vista global del proceso llevado a cabo para planear el desarrollo de un proyecto de software, aplicado en un ambiente académico.

# Tabla de contenidos

[1 Historial de cambios 2](#_Toc50576832)

[2 Resumen 4](#_Toc50576833)

[3 Tabla de contenidos 5](#_Toc50576834)

[4 Lista de figuras 6](#_Toc50576835)

[5 Lista de tablas 6](#_Toc50576836)

[6 Vista general del proyecto 7](#_Toc50576837)

[6.1 Visión del producto 7](#_Toc50576838)

[6.2 Propósito, alcance y objetivos 8](#_Toc50576839)

[6.3 Supuestos y restricciones 9](#_Toc50576840)

[6.4 Entregables 10](#_Toc50576841)

[6.5 Evolución del plan 12](#_Toc50576842)

[6.6 Glosario 12](#_Toc50576843)

[7 Contexto del proyecto 13](#_Toc50576844)

[7.1 Modelo de ciclo de vida 13](#_Toc50576845)

[7.2 Lenguajes y herramientas 16](#_Toc50576846)

[7.2.1 Criterios de Selección 16](#_Toc50576847)

[7.3 Plan de aceptación del producto 20](#_Toc50576848)

[7.4 Organización del proyecto y comunicación 21](#_Toc50576849)

[7.4.1 Interfaces externas o Stakeholders 21](#_Toc50576850)

[7.4.2 Organigrama y descripción de roles 21](#_Toc50576851)

[8 Administración del proyecto 22](#_Toc50576852)

[8.1 Métodos y herramientas de Estimación 22](#_Toc50576853)

[8.2 Inicio del proyecto 24](#_Toc50576854)

[8.3 Planes de trabajo del proyecto 26](#_Toc50576855)

[8.3.1 Estructura de descomposición de tareas 26](#_Toc50576856)

[8.3.2 Descripción y relación de precedencia de las actividades 29](#_Toc50576857)

[8.3.3 Presupuesto del proyecto. 35](#_Toc50576858)

[9 Monitoreo y control del proyecto 35](#_Toc50576859)

[9.1 Administración de requisitos 36](#_Toc50576860)

[9.2 Monitoreo y control del progreso 37](#_Toc50576861)

[9.3 Cierre del proyecto 38](#_Toc50576862)

[10 Entrega del producto 39](#_Toc50576863)

[11 Procesos de soporte 39](#_Toc50576864)

[11.1 Ambiente de trabajo 39](#_Toc50576865)

[11.2 Análisis y administración de riesgos 41](#_Toc50576866)

[11.3 Administración de configuración y documentación 43](#_Toc50576867)

[11.4 Control de calidad 45](#_Toc50576868)

[12 Anexos 52](#_Toc50576869)

[13 Referencias 52](#_Toc50576870)

# Lista de figuras

[Ilustración 1 Organigrama y descripción de roles 21](#_Toc53782953)

[Ilustración 3 Diagrama WBS de fase de sprint 1 27](#_Toc53782954)

[Ilustración 4 Diagrama WBS de sprint 2 27](#_Toc53782955)

[Ilustración 5 Diagrama de Gantt 32](#_Toc53782956)

[Ilustración 6 Diagrama BPMN de subproceso "hacer lista rápida individualmeente" 41](#_Toc53782957)

[Ilustración 7 Diagrama BPMN de proceso de identificación de riesgo 41](#_Toc53782958)

[Ilustración 8 Diagrama BPMN de proceso de control de cambio 44](#_Toc53782959)

[Ilustración 9 Diagrama BPMN de proceso de revisión de documentos 45](#_Toc53782960)

[Ilustración 10 Diagrama BPMN de proceso de pruebas unitarias 46](#_Toc53782961)

[Ilustración 11 Diagrama BPMN de proceso de pruebas del sistema 47](#_Toc53782962)

[Ilustración 12 Diagrama BPMN de proceso de prueba de rendimiento 48](#_Toc53782963)

[Ilustración 13 Diagrama BPMN de proceso de auditoria 49](#_Toc53782964)

# Lista de tablas

[Tabla 1 Tabla de Entregas 10](#_Toc50570394)

[Tabla 2 Glosario 12](#_Toc50570395)

[Tabla 3 Plan de aceptación del producto 20](#_Toc50570396)

[Tabla 4 Interfaces externas o Stakeholders 21](#_Toc50570397)

[Tabla 5 Tabla de roles 21](#_Toc50570398)

[Tabla 6 Especificación de ambiente de trabajo 24](#_Toc50570399)

[Tabla 7 Relación de precedencia 29](#_Toc50570400)

[Tabla 8 Sprints 32](#_Toc50570401)

[Tabla 9 Historias de usuario de sprint 1 33](#_Toc50570402)

[Tabla 10 Historias de usuario sprint 2 34](#_Toc50570403)

[Tabla 11 Presupuesto 35](#_Toc50570404)

[Tabla 12 Riesgos ordenados por prioridad 42](#_Toc50570405)

[Tabla 13 Ítems de configuración 44](#_Toc50570406)

[Tabla 14 Herramientas para control de calidad 51](#_Toc50570407)

# Vista general del proyecto

## Visión del producto

GDevLink busca facilitar el contacto entre desarrolladores de videojuegos y el intercambio de conocimiento entre ellos. Actualmente existe en Internet una gran comunidad de personas que se dedican al desarrollo de videojuegos como hobby. Estos proyectos se llevan a cabo de manera informal y el proceso de búsqueda de desarrolladores y demás participantes se suele realizar a través de diversos medios, como lo son redes sociales y foros de diversos sitios web. Esta diversidad de medios provoca que en ocasiones sea difícil que los desarrolladores entren en contacto.

El desarrollo de videojuegos es una actividad multidisciplinaria en la que se adquiere experiencia trabajando con otras personas, por lo que la dificultad de encontrar otras personas con quienes trabajar puede generar una barrera de entrada para aquellos que buscan participar en proyectos en los cuales puedan adquirir experiencia y armar un portafolio.

GDevLink permitirá solucionar esta problemática, permitiendo que los proyectos de desarrollo de videojuegos puedan publicar posiciones vacantes a las cuales los desarrolladores puedan aplicar, a través de una plataforma que fomente la participación de todos los miembros de la comunidad de desarrollo de videojuegos, desde los desarrolladores principiantes hasta los más experimentados. Esta diversidad de usuarios se verá reflejada, a su vez, en una gran variedad de proyectos de diversos tipos.

A través de GDevLink, los desarrolladores aficionados podrán encontrar a personas con intereses similares con quienes puedan trabajar en proyectos. GDevLink también fomentará la ayuda mutua entre los miembros de esta comunidad a través de espacios en los cuales los desarrolladores puedan preguntarles a colegas más experimentados acerca de diversos temas relacionados con el desarrollo de videojuegos.

## Propósito, alcance y objetivos

**Propósito**

El propósito de este proyecto es desarrollar una aplicación web que sirva como plataforma a través de la cual los desarrolladores de videojuegos aficionados puedan intercambiar conocimiento entre sí, seguir el proceso de desarrollo de proyectos de videojuegos de su interés, encontrar y aplicar para posiciones vacantes en proyectos en los cuales puedan participar y crear sus propios proyectos para formar equipos de desarrollo junto a otros desarrolladores.

**Alcance**

Las características generales del producto incluyen:

* Funciones básicas de registro y autenticación de usuarios.
* Visualización y edición de perfiles de usuarios, en donde se muestra información relevante acerca de la experiencia e intereses del usuario, en relación con el desarrollo de videojuegos.
* Creación, edición y visualización de páginas de proyectos de desarrollo de videojuegos, en las cuales se muestran descripción, género del juego, fase de desarrollo en la que se encuentra, usuarios involucrados con sus respectivos roles, herramientas y frameworks utilizados, imágenes y/o videos y, opcionalmente, un enlace para descargar o jugar el juego.
* Publicación de actualizaciones de progreso de los proyectos de desarrollo de videojuegos.
* Búsqueda de proyectos de desarrollo de videojuegos creados y filtrados según sus características.
* Creación, edición, eliminación y visualización de posiciones vacantes en proyectos de desarrollo de videojuegos.
* Aplicación a vacantes de un proyecto de desarrollo de videojuegos.
* Búsqueda y filtrado de posiciones vacantes.
* Visualización de la información de los aplicantes a una posición vacante.
* Opción de seguir o dejar de seguir a los proyectos de desarrollo de videojuegos.
* Visualización de actualizaciones de progreso de proyectos seguidos desde página de inicio.
* Visualización de posiciones vacantes que le podrían interesar al usuario desde página de inicio.
* Publicación de preguntas.
* Publicación de respuestas a una pregunta.
* Opción de puntuar positiva o negativamente las preguntas y respuestas.
* Opción de escoger la mejor respuesta a una pregunta.
* Búsqueda y filtrado de preguntas.

Es importante aclarar que el alcance del proyecto no contempla que los usuarios se comuniquen directamente a través de la aplicación, por lo que se espera que la comunicación que se llevaría a cabo entre el usuario que publicó una posición vacante y aquel que aplicó a ella se produzca a través de otro medio, utilizando la información de contacto indicada en la aplicación. También cabe aclarar que dentro del alcance del proyecto no se contempla que los usuarios puedan interactuar con las publicaciones a través de “likes” ni comentarios.

**Objetivos**

El objetivo general del proyecto es aplicar las metodologías estudiadas en el curso de Ingeniería de Software a través de una experiencia práctica que integre las diversas tareas involucradas en el proceso de desarrollo de un producto de software. Los objetivos específicos del proyecto incluyen:

* Llevar a cabo el proceso de planeación del desarrollo de un proyecto de software.
* Realizar la elaboración de la especificación de los requisitos de un producto de software.
* Realizar una descripción del diseño de un producto de software.

## Supuestos y restricciones

**Supuestos:**

* Los usuarios que utilicen GDevLink tendrán experiencia básica en interacción con sitios web.
* Los usuarios que utilicen GDevLink contarán con acceso a internet para navegar e interactuar en nuestra aplicación web.
* Las herramientas usadas para el desarrollo del proyecto se mantendrán en funcionamiento durante el proceso de desarrollo.
* Todos los integrantes del equipo de desarrollo poseen o están dispuestos a adquirir conocimiento en los programas, herramientas y lenguajes en los cuales va a ser desarrollado el proyecto.
* La documentación de las herramientas que serán utilizadas para el desarrollo del proyecto se mantendrá disponible de manera gratuita en internet durante el proceso de desarrollo.
* Durante del proceso de desarrollo del proyecto, se mantendrá la compatibilidad entre el sistema de gestión de base de datos y el framework de desarrollo web.
* Durante el desarrollo del proyecto, la totalidad de los integrantes del grupo contarán con disposición de tiempo para desarrollar actividades relacionadas con el proyecto, como tareas, entregas, desarrollo del código, reuniones y presentaciones.

**Restricciones:**

* Los integrantes del grupo deberán limitar al mínimo las reuniones de manera presencial a lo largo del desarrollo del proyecto debido a las dificultades que presenta la actualidad a causa de la pandemia del COVID-19.
* No se utilizarán herramientas de desarrollo que requieran algún tipo de pago a corto o largo plazo.
* El producto final va a tener un objetivo netamente educativo y funciones limitadas, por el corto tiempo que se tiene para desarrollarlo.
* El proyecto no cuenta con un cliente externo, por lo que no será posible realizar de manera realista algunas de las actividades del proceso de desarrollo de software que involucrarían la interacción con el cliente.
* No será posible aplazar fechas de entrega del proyecto, en caso de que el trabajo no haya sido completado para la fecha límite.

## Entregables

Se establecerán fechas de entrega de producto y avance con el cliente para mantenerlo informado y actualizado sobre el progreso del proyecto. A medida que se vayan completando cada uno de los entregables estos serán enviados al cliente por medio de la herramienta de comunicación (Sección 6.2.2) y enviada a más tardar en la fecha preestablecida de la entrega correspondiente. Cada entrega representará una fase de progreso del producto, y se encontrará la información de todas las actualizaciones, cumplimiento de requisitos y otros avances resumidos en una breve descripción (sección 1.2.) [1].

Se establecerán fechas de entrega de producto y avance con el cliente para mantenerlo informado y actualizado sobre el progreso del proyecto. A medida que se vayan completando cada uno de los entregables estos serán enviados al cliente por medio de la herramienta de comunicación (Sección 6.2.2) y enviada a más tardar en la fecha preestablecida de la entrega correspondiente. Cada entrega representará una fase de progreso del producto, y se encontrará la información de todas las actualizaciones, cumplimiento de requisitos y otros avances resumidos en una breve descripción (sección 1.2.) [1].

Adicionalmente, junto con el contenido de progreso, en cada entregable se adjuntará las entregas internas del equipo de trabajo. Estas abarcaran todas las etapas o fases que estuvieron involucradas en el proceso correspondiente de la fecha.

Tabla 1 Tabla de Entregas

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de Entrega** | **Descripción** | **Destinatario** | **Fecha** |
| Entregable/ fase 1 | En esta etapa de progreso se documentará todos los componentes que estructuran el proyecto. Se encontrará la investigación detallada sobre la metodología que más se acomoda al contexto del proyecto y análisis de las herramientas que el equipo de trabajo utilizará para realizar el producto final. El contenido mencionado anteriormente se registrará en un documento SPMP, este será cargado en la plataforma Blackboard. | Cliente | 09/09/20 |
| Entregable/ fase 2 | * En esta etapa de progreso se enfocará en documentar las especificaciones de los requisitos de software que forman parte de los acuerdos entre el cliente y el equipo de desarrollo. En adición, se documentarán todos los avances y actualizaciones relacionados al desarrollo del producto y también se documentará las etapas internas del equipo de trabajo que hayan participado. El contenido mencionado se registrará en un documento SRS [2], este será cargado en la plataforma Blackboard. * Junto con el documento SRS, se anexará una versión del software, esta versión tendrá implementada las actividades correspondientes al sprint 1. Esto le brindará tanto al cliente como al equipo una visión más clara de la ruta que va tomando el software final, como es la primera entrega que contiene código, se le anexarán al cliente el código fuente junto con otros archivos si es necesario y estos estarán publicados en el repositorio del equipo GDevlink en Github. Adicionalmente, se registrarán las modificaciones del SPMP a partir de la retroalimentación recibida de la entrega 1. | Cliente | 16/10/20 |
| Entregable/ fase 3 | * En esta última etapa, se entregará el producto final. En un documento SDD se registrará el diseño físico y lógico del software, se documentará la aplicación de las herramientas utilizadas en el desarrollo del producto, y cómo se llevó a cabo el modelado de datos para satisfacer los requisitos planteados en fases anteriores [3]. En otras palabras, se entregará un documento donde da a conocer la arquitectura del software para su correcto funcionamiento junto a los componentes que lo forman. El documento SDD será cargado en la plataforma Blackboard * Junto con este documento SDD se le adjuntará a la entrega el código fuente junto con sus archivos complementarios, este código contendrá las correcciones a las actividades del sprint 1 en base de la retroalimentación que el cliente haya dado al equipo. También tendrá las actividades del sprint 2, con esto, esta última entrega tendrá todas las funcionalidades implementadas que satisfacen todos los requisitos planteados por el cliente. Por último, estos archivos que contienen el código fuente y sus complementos estarán publicados en el repositorio del equipo GDevlink en Github. | Cliente | 27/11/20 |

## Evolución del plan

Para generar la primera versión del SPMP, los miembros del equipo realizarán una reunión inicial para revisar los componentes del documento, con el objetivo de asegurarse de que todos tengan la misma idea de la manera de la que debe ser elaborado cada uno de ellos y resolver las dudas que puedan existir con respecto a esto. Posteriormente, en esta misma reunión se determinarán las personas en cargadas de cada sección. La gestión y el control de estas actividades serán realizados a través de Trello, como se indica en la sección 7.2.

Cada vez que se agreguen o modifiquen elementos del SPMP, el responsable de este cambio deberá actualizar el historial de cambios incluido en la sección 1, registrando la fecha, sección modificada y responsable del cambio realizado. De igual manera, el estado de las actividades será actualizado en Trello. Una vez una tarea de este tipo haya sido completada, esta deberá ser revisada por otro miembro del grupo antes de ser marcada como terminada. Dado que algunas tareas importantes serán asignadas a dos personas para su realización, también será necesario que otras dos personas se encarguen de su revisión.

Posterior a la entrega de formal de este documento, indicada en la sección 8.3.2, se evaluará la retroalimentación brindada por el profesor para, a partir de ella, aplicarle a este documento las modificaciones o complementos pertinentes. Este tema será abordado en una reunión que se llevará a cabo una vez esté disponible la retroalimentación en cuestión, para de esta manera determinar lo antes posibles las modificaciones que deberán aplicarse.

En esta reunión también se decidirá quiénes serán los encargados de realizar estos cambios y de efectuar la revisión correspondiente una vez hayan sido completados.

Si en etapas posteriores del proceso de desarrollo se identifica alguna circunstancia significativa que modifique el contexto del desarrollo del proyecto a partir del cual se desarrolló el SPMP, o algún elemento importante que se ignoró en la elaboración de este documento, se realizará una reunión entre los miembros del grupo, en la cual se decidirá si esta situación amerita modificar el plan y, de ser así, qué acciones tomar. Al igual que en el caso anterior, para en esta reunión también se decidirá quiénes serán los responsables de realizar estos cambios.

Las versiones del documento que sean elaboradas durante del periodo del primer sprint iniciarán con la versión 2.0, mientras que las versiones del documento que sean elaboradas durante el periodo del segundo sprint iniciarán con la versión 3.0.

## Glosario

Tabla 2 Glosario

|  |  |
| --- | --- |
| **Palabra/Acrónimo** | **Definición** |
| GDevLink | Game Developer Link. Nombre del proyecto (Ver sección 6.1.) |
| Fameworks | Entorno o marco de trabajo. Hace referencia a un conjunto de herramientas o estándares utilizadas para realizar o desarrollar una actividad. |
| SPMP | Software Project Management Plan. Documento donde se tiene el plan de gestión de un proyecto de software. Contiene la información detallada referente a la planeación del proyecto, que consiste en objetivos, itinerarios, estimados, recursos, manejo de riesgo, monitoreo, control, entre otras [4]. |
| SRS | Software requirements specification. Documento con las especificaciones de los requerimientos del software. Entre su información referente a requerimientos funcionales y no funcionales, junto al objetivo de su implementación [4]. |
| SDD | Software Design Documentation. Documento con la descripción del software, el cual provee información respecto al análisis, planificación, implementación y la toma de decisiones [5]. |
| PMP | Project Management Plan. Consiste en un plan detallado con la información referente al manejo de un proyecto en general. |
| BPMN | Business Process Model and Notation. Es una notación gráfica que permite modelar los procesos de negocio, entendido como un flujo de trabajo o acciones. |
| Stakeholder | Es la persona u organización involucrada o interesadas con referente a un objeto, producto, información o proceso. En este caso son aquellos con interés en el desarrollo y creación del producto. |
| Back end | Hace referencia a la capa de acceso a datos dentro de una aplicación. |
| Front end | Hace referencia a la capa de presentación dentro de una aplicación. |
| Proyecto de desarrollo de videojuegos | Proyecto en el que un grupo de personas participa para desarrollar un videojuego. Hace referencia a proyectos informales realizados como hobby, en lugar de trabajos serios. |
| Posición vacante | Posición en un proyecto que no tiene a ninguna persona asignada, por lo que se encuentra disponible para que alguien que actualmente no forma parte del proyecto la ocupe. |
| Aplicante a posición vacante | Usuario que se postula como candidato a una posición vacante. |
| Administrador de proyecto | Tipo de permiso que puede tener un usuario con respecto a un proyecto, que le permite editar la página del proyecto y crear y gestionar posiciones vacantes y actualizaciones de progreso. |
| Administrador maestro de proyecto | Tipo de permiso que puede tener un usuario con respecto a un proyecto, que puede hacer todo lo que hace el administrador de proyecto, pero que adicionalmente puede proporcionar y revocar permisos de administrador de proyecto a los miembros del equipo. Un usuario obtiene estos permisos cuando crea un proyecto. |

# Contexto del proyecto

## Modelo de ciclo de vida

Los modelos de ciclo de vida indican la estructura o la base que seguirá el equipo para la planeación, ejecución y gestión del proceso de desarrollo de software. Por lo tanto, es de suma importancia analizar las metodologías existentes más relevantes e importantes que existen en la actualidad, ya que la estructura de trabajo que se use determinará la dinámica de trabajo del proceso de desarrollo. Es necesario elegir una metodología que se ajuste a las características de la aplicación y del equipo.

Los tipos de metodologías actuales se dividen en cuatro grandes grupos: tradicionales, ágiles, incrementales, e híbridas. Las metodologías tradicionales, como la de cascada, siguen un modelo secuencial e indoblegable, por lo que definen sus requisitos y arquitectura al inicio y no son flexibles para admitir cambios [6]. Los modelos incrementales son más flexibles que los anteriores porque empiezan con una arquitectura definida y una definición tosca de los requisitos para todo el proyecto, pero el software es desarrollado en incrementos que extienden el funcionamiento del sistema [7]. Luego están los modelos ágiles, que se centran más en software funcional, trabajo en equipo, colaboración con clientes y flexibilidad ante los cambios, que en la práctica se aplica a través del rápido desarrollo de software, usando iteraciones cortas [8]. Por último, están los modelos híbridos, que no son más que una integración entre diferentes tipos de modelos. Cada grupo de enfoque tiene diferentes metodologías.

De acuerdo con un estudio realizado en 2016, los enfoques más usados por proyectos son: híbrido con el 45.3%, luego ágil con 33.1%, luego tradicional con un 13.8%, y por último incrementales con un 7.7% [9]. Cabe mencionar que en el mismo estudio se encontró que los enfoques más populares para equipos pequeños y para proyectos de la envergadura y costo de GDevLink son los ágiles y los híbridos.

Las metodologías se evaluarán basadas en su relevancia expresada por [9], [10] y [11]. Del enfoque tradicional solo se evaluará el modelo de cascada, mientras que de las metodologías ágiles se evaluarán Scrum y Kanban.

* Cascada: Esta metodología pertenece al enfoque tradicional y se caracteriza por su estilo estrictamente secuencial, que presenta una falta de flexibilidad por la cual suele ser criticada [12].

En este modelo de ciclo de vida, solo puede empezar una vez haya concluido la anterior. Sin embargo, en el desarrollo de software las fases secuenciales se pueden solapar, y brindar información y retroalimentación a la fase anterior. Es decir, el proceso en la práctica no siempre es completamente lineal [13]. Este modelo es muy usado en proyectos de gobiernos y en muchas empresas grandes. Como este modelo enfatiza la planeación temprana, identifica defectos de diseño antes de que aparezcan. Además, su extensiva documentación y planeación la hacen ideal para proyectos en los que el control de calidad es muy importante [14].

Teniendo en cuenta que para el desarrollo de GDevLink se empleará un equipo pequeño, se considera que no sería de mucha ayuda la extensiva documentación y planeación que caracteriza a este modelo, por lo que se prefiere optar por un modelo más enfocado en el producto de software y que brinde mayor flexibilidad.

* Scrum: Esta es una metodología ágil y se concentra en el funcionamiento de los miembros del equipo para producir flexibilidad en ambientes que cambian constantemente. Scrum cuenta con los siguientes roles definidos en el equipo: Scrum máster, dueño del producto y el equipo de desarrollo [15]. En esta metodología, el equipo se auto organiza y toma decisiones acerca del proyecto, mientras la gerencia se encarga de lidiar con interferencias u obstáculos con los que se podría encontrar el quipo. Se considera que Scrum funciona especialmente bien equipos en pequeños. Esta metodología cuenta con un conjunto de prácticas características que incluyen [6]:
  + Pila del producto: Pila que contiene todas las funciones y cambios por hacer al sistema. El dueño del producto se encarga de mantenerla.
  + Pila del sprint: Es la lista de funciones que están actualmente asignadas a un sprint.
  + Sprint: Periodo entre 15 y 30 días en el que se lleva a cabo la producción de un incremento usable y potencialmente desplegable. Durante el sprint, se deben seguir tres lineamientos: no se hacen cambios que pongan en peligro el objetivo del sprint, los objetivos de calidad no disminuyen, y el alcance puede ser clarificado y renegociado entre el dueño del producto y el equipo conforme se aprende [16].
  + Planeación del sprint: Es una reunión en donde se establecen un conjunto de objetivos y funcionalidades.
  + Scrum diario: Es una reunión diaria que dura 15 minutos. En esta se habla del progreso del equipo y de los obstáculos que este ha encontrado.

Al ser una metodología ágil, Scrum presenta una mejor compatibilidad con las características del proyecto, dado que funciona bien en equipos pequeños y presenta un enfoque hacia el software funcional. Además, la idea de un equipo auto organizado resultaría apropiada para el carácter de aprendizaje del proyecto.

También cabe mencionar que el concepto de sprint se ajusta bastante bien a la modalidad de entregas a la que estará sometido el proyecto y la organización de tareas a través de las diferentes pilas podría ser de gran utilidad para gestionar las funcionalidades identificadas. También se considera que los diferentes mecanismos de comunicación aplicados por Scrum permitirían asegurar la correcta comunicación entre los miembros del equipo, pero resulta evidente que, dado que este proyecto se desarrolla en un ambiente académico, las reuniones diarias no serían apropiadas para el ritmo de trabajo que se llevará.

* Kanban: Al igual que Scrum, esta es otra metodología ágil enfocada en lanzamientos continuos, a través equipos muy colaborativos y eficientes. No tiene roles ni eventos predefinidos, es flexible en cuanto al cambio y depende de desarrollos continuos en sprints definidos [10].

Kanban se basa en tres principios fundamentales: visualizar, limitar el trabajo en proceso y manejar el flujo. Visualizar consiste en crear representaciones de cada elemento de trabajo y realizar su seguimiento a lo largo del flujo de trabajo a través de un tablero cuyas columnas indican el estado en el que se encuentran. Limitar el trabajo en proceso consiste en establecer una cantidad máxima de elementos que se trabajarán al mismo tiempo. Por último, el manejo del flujo consiste en el mejoramiento del flujo de trabajo el cual está presente durante todo el desarrollo, ya que el flujo de trabajo siempre puede ser mejorado [17].

Las principales diferencias que presenta Kanban en relación con Scrum son: la carencia de roles, que se pueden hacer cambios en todo momento, y que es más apropiada en entornos operacionales con un alto grado de variabilidad en las prioridades [18].

Al igual que Scrum, Kanban cuenta con algunas características que presentan gran compatibilidad con las características de GDevLink y del equipo, como lo son su gran flexibilidad y el uso de sprints. Sin embargo, a diferencia de Scrum, Kanban presenta un mayor nivel de flexibilidad al ignorar roles y muchos de los eventos que son característicos de Scrum. Esto podría ser en cierta medida positivo, pero dado que este proyecto se realizará en un ambiente académico, se considera que tanta flexibilidad podría conllevar a una falta de orden que podría impactar en la calidad del producto.

Luego de haber considerado estos tres modelos de ciclo de vida, se tomó la decisión de que se trabajará con Scrum porque proporciona un buen grado de flexibilidad y el uso de sprints presenta una buena compatibilidad con la estructura de entregas del proyecto. Asimismo, esta metodología se ajusta a las características del equipo y a la dinámica que se busca, dado que funciona bien en equipos pequeños y permite la auto organización de sus miembros. El modelo en cascada fue descartado por su falta de flexibilidad y porque no se ajusta tan bien a las características del equipo. Si bien Kanban presentaba muchas de las ventajas de Scrum, se optó por el segundo en lugar del primero porque se consideró que Scrum presenta una mejor organización que podría ser útil para llevar a cabo la comunicación y la organización de los miembros del equipo a lo largo del desarrollo del proyecto.

Sin embargo, para el uso de Scrum en el desarrollo de este proyecto, se consideró necesario modificar el Daily Scrum para hacerlo semanal, dado que el proyecto se desarrolla en un ambiente académico, por lo que el ritmo de trabajo es diferente y resultaría muy difícil encontrar un especio diario en el que todos los miembros del equipo puedan reunirse. La frecuencia semanal es más apropiada para el ritmo de trabajo que se llevará a lo largo del proyecto, dado que se estima que el trabajo semanal de cada uno de los miembros será de aproximadamente 8 horas.

Para la adopción de Scrum, también resulta necesario adaptar los roles en los que se basa esta metodología. Dado que se busca que todos los miembros del equipo trabajen en la implementación del producto de software, todos formarán parte del equipo de desarrollo. Además, dado que se busca que todos los miembros del equipo compartan el aprendizaje de la aplicación se esta metodología, las funciones de dueño del producto y Scrum máster las compartirán todos los integrantes en cierta medida. No obstante, con el objetivo de que haya mayor organización, se decidió que habrá integrantes con una mayor responsabilidad en estos roles. El líder de proyecto ejercerá en mayor medida las funciones del dueño del producto, mientras que el revisor del proyecto ejercerá en mayor medida las funciones de Scrum máster.

## Lenguajes y herramientas

### Criterios de Selección

A continuación, están presentados implícitamente los criterios que justificaron la selección de los lenguajes y herramientas que se planean usar. Para seleccionar las herramientas que se van a usar es necesario tener en cuenta que es lo que se necesita hacer. Por un lado, tenemos la sección de documentación, luego vendría la parte de diseño y por último tenemos el desarrollo e implementación. También hay que tener en cuenta que hay herramientas que engloban las diferentes secciones del proyecto. Para la selección general de las herramientas se tiene en cuenta que entre más personas ya sepan utilizarla mejor, con el fin de no gastar tanto tiempo en el aprendizaje, y que sea fácil de obtener, sin implicar un gasto económico.

Comenzando por la parte de documentación, lo que se requiere es una herramienta de edición de texto con la capacidad de agregar imágenes, gráficos, tablas y manejar contenido y referencias. Se necesita otra herramienta que permita realizar presentaciones, que facilite la organización y el manejo de la manera en que se presenta la información. Para ambos casos es importante que se pueda compartir la información lo más fácilmente posible y que se pueda realizar el trabajo de manera sincronizada pero remota. Ya que estas son las necesidades principales, la herramienta escogida fue Office 365 ya que todos los integrantes tienen experiencia con la herramienta y ya la poseen. Además, posee una gran variedad de configuración y manejo de la información, que al usar OneDrive permite compartir documentos, modificarlos de manera simultánea sin comprometer la información y el almacenamiento en la nube, por seguridad. También incluye otras aplicaciones que pueden ser útiles en caso de ser necesario como lo podría ser Excel.

Para la parte de diseño se requieren herramientas que permitan realizar diferentes tipos de diseños y diagramas. Estas herramientas son más específicas y en su mayoría se requieren licencias para poder usarlas, por lo que es posible que a medida que se avance en el proyecto, estas herramientas puedan variar. Una de esas herramientas sería Bizagi, para realizar los diagramas BPMN**.** Otra de estas herramientas es Microsoft Project, que en nuestro caso será usado para realizar los diagramas de Gantt, debido a su facilidad de uso.

Para la parte de desarrollo el enfoque está dado por las funcionalidades y requisitos del producto que se va a desarrollar. Para ello, las herramientas están dadas principalmente por los lenguajes e interfaces que se utilizarán. Para empezar, se necesita una manera de desarrollar páginas web, que sea simple, general, dinámica y que permita variar los diseños para poder personalizar la página a gusto. Para ello se utilizará HTML para el contenido de las páginas web y CSS para su diseño. Estas herramientas poseen varias utilidades, pero a su vez son simples de utilizar. En cuanto al framework, se buscaba una alternativa que permitiera desarrollar aplicaciones web, siguiendo un modelo de cliente-servidor, que a su vez permitiera manipular el contenido de los archivos HTML, que tuviera compatibilidad con sistemas de gestión de bases de datos y que permitiera manejar una representación de objetos. Para ello se eligió Django, que utiliza Python como lenguaje de programación del lado del servidor y cumple con los requisitos ya mencionados. Por otro lado, se escogió JavaScript como lenguaje de programación del lado del cliente. La razón de utilizar este entorno de desarrollo se debe principalmente a la existencia de cursos detallados en su uso y libre para el público, específicamente para realizar proyectos como el que se quiere realizar [19]. Además, también influyó en esta decisión el hecho de que el líder de proyecto ya tenga experiencia en desarrollo web a través de estas herramientas. Se decidió no trabajar con frameworks complejos como React, Angular o Vue para el frontend, ya que solo un miembro del equipo posee conocimientos de React, lo cual implicaría gastar tiempo en el aprendizaje de estos, cuando consideramos que no es una herramienta esencial en la construcción de páginas web. Sin embargo, se hará uso de las herramientas de bootstrap, ya que facilita construir páginas web interactivas.

Para poder mantener la información organizada y disponible para todos los integrantes del proyecto, se requiere un repositorio que permita almacenar diferentes tipos de archivos, que maneje el desarrollo de documentos de forma separada y que permita manejar versiones, por lo que se utilizará GitHub. También se necesita una para poder exponer y organizar las actividades que se van a desarrollar para poder completar el proyecto. Esta debe ser lo más sencilla y simple de usar, pero a su vez debe ser entendible e intuitiva. En este caso se utilizará Trello. El uso de estas herramientas se debe principalmente a su facilidad de acceso y de uso. En el caso de GitHub, se puede añadir que es exigida por los stakeholder. Por último, se encuentran las herramientas para la interacción entre los integrantes del proyecto, hablando específicamente de comunicación textual u oral. Por facilidad de uso y adquisición se utilizará WhatsApp, ya que todos los integrantes lo poseen y permite la comunicación individual o grupal, y para los encuentros presenciales se utilizará Hangouts, ya que permite realizar encuentros con audio y video.

**Lenguajes y herramientas:**

A continuación, se detallan los lenguajes y herramientas a usar, escogidos con los análisis previos.

**Documentación:**

1. Office 365: Es un servicio basado en la nube, diseñado para proveer seguridad, fiabilidad, y mejorar la productividad [20].
   1. Word: Es programa de procesamiento de texto con múltiples funcionalidades de edición, revisión y formato [21].
   2. PowerPoint: Es un programa de presentación, que permite organizar y mostrar la información de manera personalizada [21].
   3. OneDrive: Es un espacio de almacenamiento en línea en la nube. Entre sus funcionalidades se encuentra el proteger los archivos de trabajo y acceder a ellos en varios dispositivos. También permite compartir archivos, colaborar en documentos y sincronizar archivos [22].

**Desarrollo:**

1. HTML: El lenguaje de marcado de hipertexto *(Hypertext Markup Language*, HTML) define el significado y la estructura del contenido de una página web [23].
   1. CSS: *Cascading Style Sheet* es usado para describir el diseño y presentación de una página web, como el color y los estilos de letra, además de permitir adaptación al tamaño de la pantalla en que se visualiza [24].
2. Python: Python es un lenguaje de programación de propósito general con semántica dinámica. El syntax de Python es simple y fácil de aprender, que prioriza la legibilidad para así reducir el costo de mantenimiento [25]. Puede ser usado para desarrollar aplicaciones de escritorio, sitios web, y aplicaciones web. También permite centrarse en la funcionalidad de la aplicación al hacerse cargo de tareas comunes de la programación [6]. Python soporta completamente varios paradigmas de programación como la programación orientada a objetos, y la estructurada. Además, Python es un lenguaje interpretado, lo cual permite ejecutar el mismo código en múltiples plataformas sin volverlo a compilar, esto permite ver los cambios al código de forma inmediata sin aumentar el tiempo de desarrollo [26].
3. Django: Django es un *Framework*. Los *Frameworks* es un software diseñado para simplificar el desarrollo web. Típicamente, proporcionan librerías para acceder a bases de datos, manejar sesiones y cookies, crear plantillas para exhibir los HTML y, promover la reutilización del código [27]. Django funciona con Python, es bastante completo y brinda muchas funciones que, en otros Frameworks se acceden mediante librerías. La estabilidad, rendimiento y comunidad de Django han crecido enormemente, lo que se ha traducido en gran disponibilidad de tutoriales detallados y buenas prácticas [28].
4. JavaScript: Es un lenguaje de programación dinámico, que añade interactividad a las páginas web. JavaScript es bastante compacto, y al mismo tiempo muy flexible. Brinda muchas herramientas y funcionalidades con poco esfuerzo [29]. JavaScript es más usado para desarrollar el cliente y el FrontEnd de una página. Puede interactuar con el servidor de la página para enviar y recibir información y así actualizar la interfaz de usuario en tiempo real. Así no se requiere que se recargue la página para obtener nueva información [30].
5. Visual studio code: Es un editor de código fuente, liviano y poderoso. Viene con soporte para lenguajes como JavaScript y tiene un gran ecosistema de extensiones para lenguajes como Python [31].
6. PostgreSQL: Es una base de datos relacional de código abierto. Tiene muchas funciones que facilitan el desarrollo de aplicaciones [32].
7. Bootstrap: Es un framework de CSS, usando para construir páginas web responsivas e interactivas.

**Control de versiones:**

1. GitHub: Es un servicio de alojamiento de repositorios. Los desarrolladores pueden subir su código y trabajar colaborativamente para mejorarlo. GitHub posee un sistema de control de versiones muy robusto. El control de versiones permite a los desarrolladores cambiar el software, potencialmente arreglando bugs o mejorando la eficiencia, sin afectar el mismo software [33].

**Diagramas y diseño:**

1. Bizagi Modeler: Es un programa que permite modelar, simular, publicar y compartir procesos de negocio con notación BPMN [34]. Es una herramienta fácil de usar y conocida por todos los integrantes del grupo.
2. Microsoft Project: Es un programa hecho para la gestión de proyectos, que además permite modelar secuencias de tareas, con fechas y recursos establecidos [35].

**Comunicación y coordinación:**

1. WhatsApp: Provee un servicio de comunicación que permite enviar mensajes de texto, de voz, realizar llamadas y compartir documentos, ya sea individual o grupal [36].
2. Hangouts: es un servicio unificado de comunicaciones que permite iniciar y participar en chats de texto, voz y video, ya sea individual o grupal. Este está construido en conjunto con Google+ y Gmail, y está disponible para iOS, Android y Windows [37].
3. Trello: Es una herramienta colaborativa para organizar proyectos y tareas en tableros. Se puede observar qué tareas están pendientes, en qué se está trabajando, qué tareas han sido completadas y quién está encargado de cada tarea [38].
4. Microsoft Teams: Es una aplicación de comunicación y colaboración, con funciones de chat, reuniones y llamadas de equipos, o personas con que se trabaja, en un solo lugar. Permite transferencia de archivos, y compartir pantallas de participantes en reuniones [39].

## Plan de aceptación del producto

Tabla 3 Plan de aceptación del producto

|  |  |
| --- | --- |
| **Nombre de Entrega** | **Criterios de Aceptación** |
| Entregable/ fase 1 | El documento SPMP deberá estar realizado en su totalidad, siguiendo los requisitos estipulados al inicio por los clientes y aceptado por los integrantes. Deberá tener un nivel de redacción formal y técnico, de acuerdo con la opinión de los clientes y la totalidad de los integrantes del proyecto. El documento deberá ser realizado lo más equitativamente posible, teniendo en cuenta la opinión de los integrantes y basándose en la información de división de trabajo en las herramientas utilizadas. (Ver sección 11.1). Los integrantes del proyecto deberán estar satisfechos con el trabajo realizado. |
| Entregable/ fase 2 | El documento SRS deberá estar realizado en su totalidad, siguiendo los requisitos estipulados al inicio por los integrantes y los clientes. Los requisitos y diseños deberán satisfacer tanto a los clientes como a los integrantes del proyecto. Debe contener toda la información relevante sobre las etapas de desarrollo y las actividades realizadas. Deberá tener un nivel de redacción formal y técnico, de acuerdo con la opinión de los clientes y la totalidad de los integrantes del proyecto. El documento deberá ser realizado lo más equitativamente posible, teniendo en cuenta la opinión de los integrantes y basándose en la información de división de trabajo en las herramientas utilizadas. (Ver sección 11.1).  El cliente realizara una revisión de cada una de las funcionalidades implementadas, para validar si estas están satisfaciendo los requerimientos planteados. Estos requisitos funcionales se evaluarán teniendo en cuenta lo especificado en el monitoreo y control de progreso (ver sección 8.2), y a su vez tendrán un componente cualitativo. De esta manera, el cliente podrá observar el funcionamiento del software con respecto a las actividades correspondientes al sprint 1, dando un veredicto sobre su retroalimentación frente a la entrega. Por último, los integrantes del proyecto deberán estar satisfechos con el trabajo realizado. |
| Entregable/ fase 3 | El producto final deberá funcionar correctamente, de acuerdo con los requisitos planteados previamente, teniendo en cuenta posibles modificaciones que se realicen a los requisitos y deberá concordar con los diseños planteados. El documento SDD deberá estar escrito con un nivel de redacción formal y técnico, de acuerdo con la opinión de los clientes y la totalidad de los integrantes del proyecto y explicará de la manera más clara y concisa posible el funcionamiento del producto y sus componentes. El documento deberá ser realizado lo más equitativamente posible, teniendo en cuenta la opinión de los integrantes y basándose en la información de división de trabajo en las herramientas utilizadas. (Ver sección 11.1).  Al tener la totalidad de las funciones implementadas (actividades de sprint 1 y 2), el cliente realizara pruebas en todas las funcionalidades implementadas en el software, teniendo como referencia lo especificado en monitoreo y control de progreso (ver sección 8.2). Estas funcionalidades deberán satisfacer todos los requisitos planteados por el cliente, se tendrá en cuenta también la presentación de la aplicación y su navegación amigable con el usuario. Por último, una vez el cliente haya probado las funcionalidades dará su veredicto de satisfacción y aceptación frente al software presentado. Los integrantes del proyecto deberán estar satisfechos con el trabajo realizado. |

## Organización del proyecto y comunicación

### Interfaces externas o Stakeholders

Tabla 4 Interfaces externas o Stakeholders

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Responsabilidad Asociada** | **Datos de Contacto** |
| Carlos Andrés Parra | Motivador principal del proyecto y guía principal al momento de desarrollar | Principal responsable de la aceptación del producto en cada entrega establecida. Establece requisitos formales del proyecto | Correo electrónico; Microsoft Teams. |
| Usuarios | Son las personas que harán uso del producto | No poseen ninguna responsabilidad en la fase de desarrollo. | Retroalimentación una vez se tenga un demo funcional. |
| Integrantes del proyecto | Hacen parte del desarrollo del proyecto, pero también son los clientes. | Responsables de establecer requisitos funcionales y la aceptación de estos | Se utilizará WhatsApp para comunicación diaria y para los encuentros síncronos se usará Hangouts. (Sección 11.1.) |

### Organigrama y descripción de roles

Ilustración 1 Organigrama y descripción de roles

Tabla 5 Tabla de roles

|  |  |  |  |
| --- | --- | --- | --- |
| **Rol** | **Encargado** | **Descripción** | **Responsabilidad Asociada al proyecto** |
| Líder de proyecto | Juan Carlos Suarez Jaimes | Es el coordinador del proyecto, gestiona el trabajo del grupo. Monitorea el trabajo realizado. Es el responsable de liderar y gerenciar el proyecto para lograr los resultados esperados en tiempo y forma [40]. | Se comunica con los stakeholders, coordina y documenta reuniones del grupo y el proyecto. Evalúa el progreso del proyecto y revisa que cumpla con los requisitos especificados. |
| Líder de arquitectura | Santiago Caroprese Hidalgo | Responsable de la definición y diseño de la arquitectura [40]. | Se encarga de que el proyecto mantenga una consistencia a lo largo del desarrollo y en todas sus secciones. |
| Líder de desarrollo | Gabriel Gómez Corredor | El responsable por el desarrollo de las piezas de software. Debe diseñar, producir o mantener componentes conforme a especificaciones [40]. | Estar al tanto de los requisitos/ especificaciones del proyecto (Sección 6.3) y asegurar que se apliquen en todas las fases del desarrollo. |
| Líder de configuración | Juan Pablo Méndez Perdomo | El administrador de configuración debe asegurarse de facilitar la revisión del producto y las actividades de seguimiento de cambios y defectos. También es responsable informar las estadísticas de progreso según las solicitudes de cambio [41]. | Realizar registros de cambios y avances del proyecto. |
| Líder de calidad | Diego Andrés Burgos Melo | Encargado de establecer controles de seguridad, y principal encargado de revisar y asegurar la calidad del trabajo realizado por todos los integrantes. | Estar al tanto de manejar y evaluar los aspectos de calidad en las secciones del proyecto. (Sección 10.4) |
| Yo | Daniel Hernández García | Responsable de evaluar los artefactos de planificación y los artefactos de evaluación del proyecto en los principales puntos de revisión del ciclo de vida del proyecto [41]. | Asegurarse de que en cada sesión de planeación y evaluación del proyecto se mantenga un equilibrio entre los integrantes, el desarrollo del proyecto y el resultado del proyecto. Debe estar al tanto de los artefactos del modelo del ciclo de vida que se estén implementando. (Sección 7.1) (Sección 8.3.2) |

# Administración del proyecto

## Métodos y herramientas de Estimación

Una estimación es una actividad para planificar el proyecto determinando de una manera no tan exacta que se enfoca en estos cuatro elementos:

* Dinero
* Esfuerzo
* Recursos
* Tiempo

Estos serán necesarios para el desarrollo de nuestro producto de una manera satisfactoria Aparte de esto, también nos permitirá determinar la viabilidad del proyecto, la asignación de los recursos y hacer planes de una manera realista teniendo en cuenta no exceder el tiempo límite para el desarrollo del producto.

Hablando en términos de dinero, implicaría detalles como:

* Hardware y software.
* Entrenamiento del personal.
* Salarios.
* Costos sociales y de seguros.
* Costos colaterales.

En cuanto al esfuerzo y el tiempo, se analizará qué tan compleja es la tarea para realizar, cuántas horas tardaría esa persona en realizar dicha tarea, y si tienen dependencias o no. Finalmente, hablando de recursos, simplemente se tendrá que ver cuál recurso se necesita, la disponibilidad de este, fecha en la que se requeriría y el tiempo disponible en estas fechas, para que podamos estimar de manera correcta cosas tan fundamentales del proyecto como lo son el tiempo y la complejidad de cada historia de usuario.

Por lo tanto, nacieron unas técnicas y métodos para lograr una estimación más exacta como:

* **Planning poker:** Es una técnica de estimación que se basa en un consenso. Se utiliza para estimar el esfuerzo de las tareas para el desarrollo de software y es una variación del método de Delphi, comúnmente usada en el desarrollo ágil. Este es un método sencillo y entretenido, en el cual todos los miembros del equipo colaboran para poder calcular el esfuerzo de cada tarea. Su dinámica es la siguiente: todos los miembros del equipo se reúnen y uno asume el rol de moderador, el cual no podrá jugar y será asesorado por el gestor del proyecto. El gestor del proyecto es la persona con más conocimientos acerca del tema y dará una breve introducción sobre el ítem a evaluar. Los demás integrantes participarán de la siguiente manera: sobre la mesa se tienen un mazo y cada integrante colocará una carta que contendrá un número: un valor pequeño para indicar que la tarea es poco compleja y un valor grande para indicar que una tarea es muy compleja. También existen dos cartas adicionales que son el signo de interrogación “?” para indicar que tienen incertidumbre acerca de esa historia de usuario y otra con una taza de café la cual indica que el participante está demasiado cansado para continuar y necesita un descanso. Una vez que todos seleccionen sus tarjetas se mostrarán simultáneamente y se debatirá entre las estimaciones más altas con las bajas hasta llegar a un consenso. Esto proceso se repite por cada historia de usuario [42].

Para la estimación de nuestro proyecto escogimos Planning poker, ya que nos pareció una manera dinámica y acertada para nuestro método de desarrollo ágil aplicado a este proyecto, en el cual todos los integrantes del grupo pueden aportar con el mismo peso a cada estimación por historia de usuario, dando un valor que cada uno considerara correcto, con la opción de justificar. Por esto decidimos usar la plataforma [43] “Plan it Poker” el cual trae beneficios como; una buena interfaz, capacidad de almacenar todas las historias de usuario a evaluar y sus respectivos puntajes obtenidos mediante estadísticas de los votos y gráficos circulares.

Este proceso se llevó a cabo en dos sesiones de manera síncrona por medio de llamada en la plataforma hangouts, donde el líder del proyecto fue el “product owner”. Este rol era muy importante, ya que este posee la experiencia en desarrollo de aplicaciones de este tipo y además se encuentra familiarizado con el framework escogido para este proyecto (ver sección 6.2). De esta manera se evaluaron las 35 historias de usuario, una por una, donde cada integrante realizaba la estimación según creyera correspondiente, justificando su decisión. Posterior a esto, nuestro product owner resolvía dudas acerca del trabajo que conllevaba cada historia de usuario y luego se votaba nuevamente. Este proceso se repitió hasta que todos los integrantes del grupo llegaran a un acuerdo (se anexan registro de la votación e información de los cálculos).

Para las estimaciones, se consideró que cada uno de los integrantes le podrá dedicar 8 horas de trabajo semanales a las actividades del proyecto. Por lo tanto, al ser seis integrantes, se determinó que serán en total 48 horas semanales de trabajo.

**Sprint 0:** Este primer sprint consistirá en la planeación y desarrollo del documento llamado SPMP el cual tendremos 3 semanas de elaboración un total de 144 horas. Dicha elaboración del documento se repartirá en los 6 integrantes del scrum team, por lo cual en promedio a cada estudiante le corresponderá un trabajo de total de 24 horas en cuanto al desarrollo de este mismo.

**Sprint 1:** A partir de la experiencia con la elaboración del SPMP, pudimos determinar que el tiempo de elaboración de este tipo de documento lleva entre semana y media, y dos semanas. Examinando las plantillas, se encontró que, en comparación con el SPMP, el SRS presenta una cantidad de trabajo similar, por lo que se estima que su tiempo de elaboración será de dos semanas, es decir, 96 de las 192 horas correspondientes al sprint.

Adicionalmente, para este sprint también se contempla la implementación de 8 historias de usuario, cuyo total de puntos de historia equivale a 67 puntos. Estos puntos se deben distribuir entre las 96 horas restantes del sprint. Mediante la división de las horas estimadas del desarrollo sobre el total de puntos del sprint, se pudo calcular que para este sprint el número de horas por historia de usuario es de 1.4 horas (ver anexo).

**Sprint 2:** Para el caso del SDD, se encontró que presenta una menor longitud que la del SPMP, por lo que se estimó que su elaboración tomará una semana, es decir, 48 de las 240 horas del sprint.

Adicionalmente, para este sprint también se contempla la implementación de 30 historias de usuario, cuyo total de puntos de historia equivale a 179 puntos. Estos puntos se deben distribuir entre las 192 horas restantes del sprint. Mediante la división de las horas estimadas del desarrollo sobre el total de puntos del sprint, se pudo calcular que para este sprint el número de horas por historia de usuario es 1.5 horas (ver anexo).

## Inicio del proyecto

Por los problemas que se están dando en la actualidad a nivel global, se desarrollará la página web desde la casa de cada integrante del grupo, los cuales cuentan con al menos un computador o laptop y si a futuro es posible, contaríamos con los equipos de cómputo disponibles por la Pontificia Universidad Javeriana. Los equipos de cómputo disponibles actualmente cuentan con las siguientes características:

Tabla 6 Especificación de ambiente de trabajo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | PROCESADOR | DISCO DURO | MEMORIA RAM | SO | RED | No. DE EQUIPOS |
| LIDER DE PROYECTO | Intel Core i7 –7700HQ | 1 TB HDD  128 GB SSD | 16 GB | Windows 10 | 60 MB | 2 |
| LIDER DE ARQUITECTURA | Intel Core i5- 8300H | 1 TB HDD | 8 GB | Windows 10 | 5 MB | 1 |
| LIDER DE CALIDAD | Intel Core i7 8 generación | 500 GB HDD | 16GB | Windows 10 | 15 MB | 2 |
| LIDER DE DESARROLO | AMD Ryzen 5 2500U | 1 TB HDD | 16GB | Windows 10 | 50 MB | 2 |
| LIDER DE CONFIGURACION | Intel Core i7 8 generación | 500 GB HDD | 16GB | Windows 10 | 70 MB | 2 |
| REVISOR DE PROYECTO | Intel Core i5 - 4400 | 1 TB  HDD | 8 GB | Windows 10 | 100 MB | 1 |

Como se indica en la sección 7.2, se utilizará el framework Django, junto con los lenguajes de programación Python y JavaScript. En cuanto a la estimación tuvimos en cuenta la técnica de planning poker para calcular el desarrollo de nuestra aplicación web basados en metodologías agiles, con base a esto calculamos un valor de SP en el sprnt 1 de 0.88 horas y para el sprint 2 de 1.5 podemos ver detalladamente este cálculo en la sección 8.1. El líder de proyecto y el líder de desarrollo son los miembros del equipo con mayor experiencia en el uso de estas herramientas, por lo que ellos son los más capacitados en conocimiento y habilidad. Por lo tanto, el resto del grupo (líder de arquitectura, líder de calidad, líder de configuración y revisor del proyecto) se tendrán que capacitar en el desarrollo de páginas web usando Django y Python para la elaboración de nuestro producto, tomando un curso en línea dirigido por la universidad de Harvard llamado “Web Programming with Python and JavaScript”. Este curso tendrá que ser tomado por los integrantes del grupo antes de iniciar el desarrollo del producto, por lo que se estima que tendrán un plazo máximo de dos semanas, para que todos los integrantes del grupo puedan contar con los conocimientos suficientes para la realización de dichas tareas. En caso de surgir alguna duda en el procedimiento de aprendizaje se redirigirá en primera instancia hacia el líder de proyecto o el líder de desarrollo.

La asignación de tareas dependerá del rol de cada integrante del grupo, por lo que serán asignadas como se describe en el organigrama de la sección 7.4 del documento.

Posteriormente, cuando todos los integrantes del grupo se hayan capacitado con el conocimiento necesario y cuenten con las herramientas de trabajo descritas en la sección 7.2, se podrá dar el aval para la implementación de las funcionalidades del producto de software. Posteriormente, el equipo de trabajo se reunirá para analizar cuáles son las tareas que hay que hacer y cuáles de estas son complejas, para asignarles los recursos necesarios. En este orden de ideas, a las tareas se les puede asignar un grupo en específico o un solo integrante. Además, cada integrante o grupo de integrantes tendrá que mostrar avances semanales que serán monitoreadas a través de la herramienta de organización de actividades seleccionada (sección 7.2). El avance del proyecto seguirá procesos estrictamente regulados (sección 9.2) para asegurar la calidad (sección 10.4) y el cumplimiento de requisitos funcionales (sección 6.1 y 6.2).

## Planes de trabajo del proyecto

### Estructura de descomposición de tareas

El componente de desarrollo del sprint 1 se descompuso en dos actividades principales: el desarrollo de las funcionalidades de usuario y el desarrollo de las funcionalidades de proyecto. Por otra parte, se identificó que los principales componentes de desarrollo del sprint 2 serán los relacionados con funcionalidades adicionales de proyecto, funcionalidades relacionadas con las posiciones vacantes, funcionalidades relacionadas con las preguntas y funcionalidades relacionadas con comunicación. Cada una de estas actividades incluye, a su vez, una descomposición más detallada de las funcionalidades que deberán implementarse para cada uno de estos componentes.



Ilustración 3 Diagrama WBS de fase de sprint 1

Ilustración 4 Diagrama WBS de sprint 2

### Descripción y relación de precedencia de las actividades

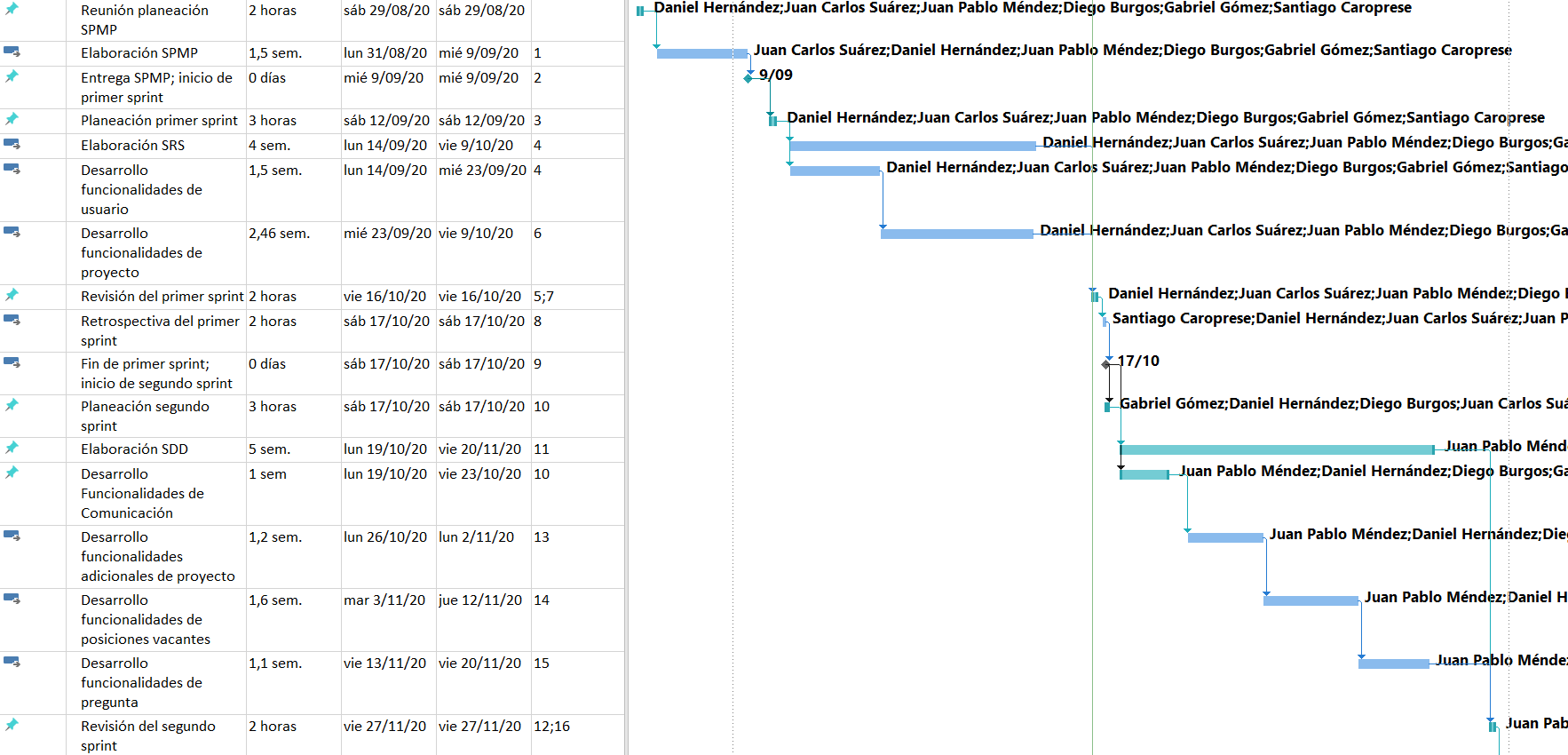
En la siguiente tabla se muestran la duración, actividades predecesoras y fechas de comienzo y fin para las tareas que se realizarán a lo largo del desarrollo del proyecto. Para la elaboración del cronograma se tuvieron en cuenta las estimaciones indicadas en la sección 8.1. Se consideró que, durante la totalidad de cada sprint, se realizarán de manera paralela la elaboración del documento y el desarrollo de la aplicación. A partir de esto, se obtuvo la proporción del tiempo de cada semana que se le destinaría a cada una estas actividades. Por lo tanto, se determinó que, durante cada una de las semanas del primer sprint se destinará el 50% de las horas de trabajo para la elaboración del SRS. Se igual manera, se determinó que para cada una de las semanas del segundo sprint se destinará el 20% de las horas de trabajo para el desarrollo del SDD. En ambos casos, el resto del tiempo semanal se le dedicará al desarrollo de la aplicación. Cabe mencionar que para ambos sprints se realizó el cronograma de tal manera que quedaran algunos días de holgura.

Tabla 7 Relación de precedencia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Número | Nombre de tarea | Duración | Comienzo | Fin | Predecesoras | Nombres de los recursos |
| 1 | Reunión planeación SPMP | 2 horas | sáb 29/08/20 | sáb 29/08/20 |  | Daniel Hernández;Juan Carlos Suárez;Juan Pablo Méndez;Diego Burgos;Gabriel Gómez;Santiago Caroprese |
| 2 | Elaboración SPMP | 1,5 sem. | lun 31/08/20 | mié 9/09/20 | 1 | Juan Carlos Suárez;Daniel Hernández;Juan Pablo Méndez;Diego Burgos;Gabriel Gómez;Santiago Caroprese |
| 3 | Entrega SPMP; inicio de primer sprint | 0 días | mié 9/09/20 | mié 9/09/20 | 2 |  |
| 4 | Planeación primer sprint | 3 horas | sáb 12/09/20 | sáb 12/09/20 | 3 | Daniel Hernández;Juan Carlos Suárez;Juan Pablo Méndez;Diego Burgos;Gabriel Gómez;Santiago Caroprese |
| 5 | Elaboración SRS | 4 sem. | lun 14/09/20 | vie 9/10/20 | 4 | Daniel Hernández;Juan Carlos Suárez;Juan Pablo Méndez;Diego Burgos;Gabriel Gómez;Santiago Caroprese |
| 6 | Desarrollo funcionalidades de usuario | 1,5 sem. | lun 14/09/20 | mié 23/09/20 | 4 | Daniel Hernández;Juan Carlos Suárez;Juan Pablo Méndez;Diego Burgos;Gabriel Gómez;Santiago Caroprese |
| 7 | Desarrollo funcionalidades de proyecto | 2,46 sem. | mié 23/09/20 | vie 9/10/20 | 6 | Daniel Hernández;Juan Carlos Suárez;Juan Pablo Méndez;Diego Burgos;Gabriel Gómez;Santiago Caroprese |
| 8 | Revisión del primer sprint | 2 horas | vie 16/10/20 | vie 16/10/20 | 5;7 | Daniel Hernández;Juan Carlos Suárez;Juan Pablo Méndez;Diego Burgos;Gabriel Gómez;Santiago Caroprese |
| 9 | Retrospectiva del primer sprint | 2 horas | sáb 17/10/20 | sáb 17/10/20 | 8 | Santiago Caroprese;Daniel Hernández;Juan Carlos Suárez;Juan Pablo Méndez;Diego Burgos;Gabriel Gómez |
| 10 | Fin de primer sprint; inicio de segundo sprint | 0 días | sáb 17/10/20 | sáb 17/10/20 | 9 |  |
| 11 | Planeación segundo sprint | 3 horas | sáb 17/10/20 | sáb 17/10/20 | 10 | Gabriel Gómez;Daniel Hernández;Diego Burgos;Juan Carlos Suárez;Santiago Caroprese |
| 12 | Elaboración SDD | 5 sem. | lun 19/10/20 | vie 20/11/20 | 11 | Juan Pablo Méndez;Daniel Hernández;Diego Burgos;Gabriel Gómez;Juan Carlos Suárez;Santiago Caroprese |
| 13 | Desarrollo Funcionalidades de Comunicación | 1 sem | lun 19/10/20 | vie 23/10/20 | 10 | Juan Pablo Méndez;Daniel Hernández;Diego Burgos;Gabriel Gómez;Juan Carlos Suárez;Santiago Caroprese |
| 14 | Desarrollo funcionalidades adicionales de proyecto | 1,2 sem. | lun 26/10/20 | lun 2/11/20 | 13 | Juan Pablo Méndez;Daniel Hernández;Diego Burgos;Gabriel Gómez;Juan Carlos Suárez;Santiago Caroprese |
| 15 | Desarrollo funcionalidades de posiciones vacantes | 1,6 sem. | mar 3/11/20 | jue 12/11/20 | 14 | Juan Pablo Méndez;Daniel Hernández;Diego Burgos;Gabriel Gómez;Juan Carlos Suárez;Santiago Caroprese |
| 16 | Desarrollo funcionalidades de pregunta | 1,1 sem. | vie 13/11/20 | vie 20/11/20 | 15 | Juan Pablo Méndez;Daniel Hernández;Diego Burgos;Gabriel Gómez;Juan Carlos Suárez;Santiago Caroprese |
| 17 | Revisión del segundo sprint | 2 horas | vie 27/11/20 | vie 27/11/20 | 12;16 | Juan Pablo Méndez;Daniel Hernández;Diego Burgos;Gabriel Gómez;Juan Carlos Suárez;Santiago Caroprese |
| 18 | Retrospectiva del segundo sprint | 2 horas | sáb 28/11/20 | sáb 28/11/20 | 17 | Juan Pablo Méndez;Daniel Hernández;Diego Burgos;Gabriel Gómez;Juan Carlos Suárez;Santiago Caroprese |
| 19 | Fin de segundo sprint | 0 horas | sáb 28/11/20 | sáb 28/11/20 | 18 |  |

En la siguiente figura, se muestra el diagrama de Gantt que ilustra la información de la tabla anterior.

Ilustración 5 Diagrama de Gantt



En la siguiente tabla se incluyen las descripciones de las actividades significativas que se llevarán a cabo para cada sprint.

Tabla 8 Sprints

|  |  |
| --- | --- |
| **Actividad** | **Descripción** |
| Planeación del sprint | Dado que ya se definieron los componentes que se trabajarán en cada sprint, esta reunión entre los miembros del equipo será destinada para detallar las tareas que componen las actividades relacionadas con las historias de usuario que forman parte del sprint y definir sus criterios de aceptación. Esta reunión busca aclarar las dudas que puedan existir acerca de la forma de la que se realizarán las actividades, con el objetivo de llegar a un consenso entre todos los miembros del equipo de desarrollo. |
| Desarrollo del sprint | Proceso con duración cercano a un mes, en el cual los miembros del equipo se ocupan del desarrollo de un incremento usable del producto de software y de la elaboración de documentación. Durante el desarrollo de esta actividad, los miembros del equipo realizarán reuniones semanales (“Weekly” Scrum) y gestionarán el sprint backlog, actualizando el estado de las actividades. Las actividades que componen el desarrollo incluyen la elaboración de un documento (SRS en el sprint 1 y SDD en sprint 2) y la implementación de historias de usuario agrupadas en componentes de funcionalidad. |
| Revisión del sprint | Reunión en la que participan los miembros del equipo y el profesor para llevar a cabo una demostración del incremento del producto desarrollado a lo largo del sprint. El líder de proyecto explica cuáles elementos del backlog del producto han sido completados y cuáles están pendientes. Además, los miembros del equipo de desarrollo explican, en términos generales, la dinámica del trabajo y los problemas a los que se enfrentaron y cómo los solucionaron [16]. |
| Retrospectiva del sprint | Reunión dirigida por el Scrum máster en la que se examina en qué medida el desarrollo el sprint fue satisfactorio, identificando elementos positivos y posibles mejoras. En esta reunión también se elabora un plan para implementar las mejoras identificadas [16]. |

En la siguiente tabla, se indican las historias de usuario que pertenecen a cada uno de los componentes de funcionalidad identificados para el primer sprint.

Tabla 9 Historias de usuario de sprint 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Componente** | **Historia de Usuario** | **Descripción** | **Número de Horas** |
| Usuario | 1 | Como USUARIO que NO ha hecho LOG IN, quiero REGISTRARME para CREAR UNA CUENTA-PERFIL | 17.6 |
| 2 | Como USUARIO que NO ha hecho LOG IN, quiero hacer LOG IN para ACCEDER A FUNCIONALIDADES DE USUARIO | 1.76 |
| 3 | Como USUARIO que ha hecho LOG IN, quiero hacer LOG OUT, para CERRAR MI PERFIL | 1.76 |
| 4 | Como USUARIO que ha hecho LOG IN, quiero ENTRAR A MI PERFIL, para VER MI INFORMACION | 2.64 |
| 5 | Como USUARIO que ha hecho LOG IN, quiero EDITAR MI PERFIL, para ACTUALIZAR MI INFORMACION | 11.44 |
| 6 | Como USUARIO, quiero ENTRAR AL PERFIL de OTROS USUARIOS, para VER SU INFORMACIÓN | 0.88 |
| Proyecto | 7 | Como USUARIO que ha hecho LOG IN, quiero CREAR UN PROYECTO, para REPRESENTAR MI PROYECTO y CONVERTIRME EN SU ADMINISTRADOR | 17.6 |
| 8 | Como ADMINISTRADOR de proyecto, quiero EDITAR MI PROYECTO, para ACTUALIZAR SU INFORMACION | 11.44 |
| 9 | Como USUARIO que ha hecho LOG IN, quiero VER PROYECTOS en los que PARTICIPO, para CONSULTAR SU INFORMACIÓN | 4.4 |
| 10 | Como ADMINISTRADOR de proyecto, quiero AGREGAR MIEMBROS a mi proyecto, para INCLUIRLOS en el equipo de desarrollo | 7.04 |
| 15 | Como ADMINISTRADOR de proyecto, quiero QUITAR A UN MIEMBRO, para INDICAR que ya no hace parte del EQUIPO DE DESARROLLO | 1.76 |
| 12 | Como ADMINISTRADOR de proyecto, quiero HACER ADMINISTRADOR a OTRO MIEMBRO del proyecto, para que EL PUEDA EDITAR EL PROYECTO | 1.76 |
| 16 | Como ADMINISTRADOR de proyecto, quiero QUITAR PERMISOS DE ADMINISTRADOR, para que NO puedan EDITAR EL PROYECT | 1.76 |
| 13 | Como USUARIO, quiero ENTRAR A LA PAGINA DE UN PROYECTO, para CONSULTAR SU INFORMACION Y VER SUS ACTUALIZACIONES | 1.76 |
|  | 11 | Como ADMINISTRADOR DE PROYECTO, quiero PUBLICAR las ultimas actualizaciones para COMPARTIR EL PROGRESO DEL PROYECTO | 11.44 |

En la siguiente tabla, se indican las historias de usuario que pertenecen a cada uno de los componentes de funcionalidad identificados para el segundo sprint.

Tabla 10 Historias de usuario sprint 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Componente** | **Historia de Usuario** | **Descripción** | **Número de Horas** |
| Comunicación | 36 | Como USUARIO que ha hecho LOG IN, quiero CREAR UNA CONVERSACIÓN CON OTRO USUARIO, para ESTABLECER UN CANAL DE COMUNICACIÓN CON Él. | 12 |
| 37 | Como USUARIO que ha hecho LOG IN, quiero VER MIS CONVERSACIONES CON OTROS USUARIOS, para VISUALIZAR LOS MENSAJES QUE HE ENVIADO Y LOS QUE HE RECIBIDO. | 19.5 |
| 38 | Como USUARIO que ha hecho LOG IN, quiero ENVIAR UN MENSAJE EN UNA DE MIS CONVERSACIONES, para COMUNICARME CON OTROS USUARIOS. | 7.5 |
| Proyecto | 14 | Como USUARIO que ha hecho LOG IN, quiero SEGUIR un proyecto para RECIBIR SUS ACTUALIZACIONES EN MI PAGINA DE INICIO | 30 |
| 17 | Como USUARIO, quiero BUSCAR y FILTRAR PROYECTOS, para ENCONTRAR PROYECTOS de mi interés | 12 |
| 26 | Como USUARIO que NO ha hecho LOG IN, quiero VER EN MI PAGINA DE INICIO los PROYECTOS CON MAS SEGUIDORES, para VER LOS PROYECTOS que me pueden interesar | 4.5 |
| Posiciones Vacantes | 18 | Como ADMINISTRADOR de proyecto, quiero CREAR UNA POSICION VACANTE, para ENCONTRAR un nuevo MIEMBRO DEL EQUIPO | 12 |
| 19 | Como ADMINISTRADOR de proyecto, quiero EDITAR UNA POSICION vacante, para ACTUALIZAR LA INFORMACION | 7.5 |
| 20 | Como ADMINISTRADOR de proyecto, quiero ELIMINAR una POSICION vacante, para INDICAR que YA NO ES NECESARIA | 3 |
| 21 | Como USUARIO, quiero BUSCAR y FILTRAR POSICIONES VACANTES, para ENCONTRAR UNA POSICION de mi interés | 12 |
| 22 | Como USUARIO que ha hecho LOG IN, quiero APLICAR A UNA POSICIÓN vacante, para que me CONSIDEREN para la POSICIÓN | 3 |
| 23 | Como ADMINISTRADOR de proyecto, quiero VISUALIZAR LAS POSICIONES vacantes de mi proyecto, para REVISAR LA INFORMACION | 4.5 |
| 24 | Como ADMINISTRADOR de proyecto, quiero VER LOS APLICANTES de una POSICION vacante, para VISUALIZAR SU INFORMACION y DATOS DE CONTACTO | 7.5 |
| 25 | Como USUARIO que ha hecho LOG IN, quiero VER en mi página de inicio LAS POSICIONES VACANTES que me pueden INTERESAR, para CONOCER las POSICIONES a las que puedo aplicar | 12 |
| Preguntas | 27 | Como USUARIO que ha hecho LOG IN, quiero CREAR una pregunta para RESOLVER MI DUDA | 7.5 |
| 28 | Como USUARIO que ha hecho LOG IN, quiero PUNTUAR positivamente una pregunta para PODER INDICAR QUE FUE DE UTILIDAD | 4.5 |
| 29 | Como USUARIO que ha hecho LOG IN, quiero PUNTUAR negativamente una pregunta para PODER INDICAR QUE NO FUE DE UTILIDAD | 4.5 |
| 30 | Como USUARIO, quiero CONSULTAR la sección de preguntas para PODER VER LAS PREGUNTAS MAS POPULARES | 4.5 |
| 31 | Como USUARIO que ha hecho LOG IN, quiero CREAR una respuesta para PODER COMPARTIR MI SOLUCIÓN | 4.5 |
| 32 | Como USUARIO que ha hecho LOG IN, quiero PUNTUAR positivamente una respuesta para PODER INDICAR QUE FUE 5DE UTILIDAD | 4.5 |
| 33 | Como USUARIO que ha hecho LOG IN, quiero PUNTUAR negativamente una respuesta para PODER INDICAR QUE NO FUE DE UTILIDAD | 4.5 |
| 34 | Como USUARIO que ha hecho LOG IN, quiero ESCOGER la mejor respuesta a mi pregunta para SATISFACER mi duda | 3 |
| 35 | Como USUARIO quiero FILTRAR en la barra de búsqueda algunos criterios para ENCONTRAR UNA PREGUNTA | 4.5 |

### Presupuesto del proyecto.

Actualmente el proyecto no estima ingresos, por lo que en la tabla 11 solo están los valores estimados de los gastos. Estos están basados en el número de horas requeridos por cada actividad, definidas en la sección 7.1 y sección 8.1. Con un valor de $50.000 pesos el costo de la hora de cada miembro. El costo total del proyecto es de $31.680.000.

Tabla 11 Presupuesto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Actividad** | **Agosto** | **Septiembre** | **Octubre** | **Noviembre** |
| **Ingresos** | **$ -** | **$ -** | **$ -** | **$ -** |
| **Gastos** | **$ 2.400.000,00** | **$ 8.100.000,00** | **$ 13.284.000,00** | **$ 7.584.000,00** |
| Planeación del Sprint | $ - | $ 900.000,00 | $ 900.000,00 | $ - |
| Desarrollo del Sprint | $ - | $ 2.400.000,00 | $ 7.200.000,00 | $ 4.800.000,00 |
| Revisión del Sprint | $ - | $ - | $ 600.000,00 | $ 600.000,00 |
| Retrospectiva del Sprint | $ - | $ - | $ 600.000,00 | $ 600.000,00 |
| SPMP | $ 2.400.000,00 | $ 2.400.000,00 | $ - | $ - |
| SRS | $ - | $ 2.400.000,00 | $ 2.400.000,00 | $ - |
| SDD | $ - | $ - | $ 1.200.000,00 | $ 1.200.000,00 |
| Alojamiento web | $ - | $ - | $ 192.000,00 | $ 192.000,00 |
| Base de datos |  |  | $ 192.000,00 | $ 192.000,00 |
| **Flujo de caja libre** | **-$ 2.400.000,00** | **-$ 8.100.000,00** | **-$ 13.284.000,00** | **-$ 7.584.000,00** |

# Monitoreo y control del proyecto

## Administración de requisitos

Las metodologías de desarrollo ágil han sido diseñadas para tratar con el cambio constante de requisitos durante los procesos de desarrollo. En estas metodologías, cuando se proponen cambios de requisitos, estos no pasan por un proceso formal de gestión de cambios [1]. Como ya se mencionó, Scrum no permite cambios durante el Sprint que interfieran con el objetivo de este (Sección 7.1). Por este motivo, solo se admitirán cambios en los requisitos durante un Sprint cuando estos no sean muy significativos. Un ejemplo de requisito no significativo sería: “el sistema debe presentar la temperatura en grados Celsius, Fahrenheit y kelvin”. Estos requisitos serían mencionados en el Scrum semanal y acompañados por una discusión sobre su implementación. En el caso de requisitos más significativos, estos deberán ser discutidos durante la planificación del Sprint y seguirá un proceso detallado.

Si algún Stakeholder, cliente, dueño del producto, o miembro del equipo identifica que un requisito no fue analizado o identificado correctamente, o fue omitido inicialmente, en el caso de los stakeholders o clientes, se debe seguir el siguiente proceso (especificado con diagrama BPMN en el anexo AdministracionRequisitos):

1. Informar al dueño del producto sobre el requisito (puede ser durante el sprint). Este lo analizará y definirá su importancia. Luego expresará sus opiniones al cliente.
2. Si el requisito es muy complejo, el dueño del producto puede analizar y discutir el diseño del requisito con el equipo de desarrollo, durante el scrum semanal.
3. Durante la planeación del Sprint, si no hubo reflexión previa sobre el requisito, este pasará a ser tema de discusión. Si ya fue discutido en el paso 2, el dueño del producto o un miembro del equipo puede intervenir con alguna duda o inquietud con respecto al requisito.
4. El requisito o su descomposición en tareas es añadida a la cola de productos o es descartada, según las conclusiones a las que se haya llegado.

## Monitoreo y control del progreso

Debido a que el desarrollo del proyecto se realizará utilizando una metodología ágil, el progreso de este se medirá a partir del software funcional desarrollado (Sección 7.1). Las unidades utilizadas para medir esto serán los puntos de historia asignados a cada historia de usuario, utilizando el método descrito en la sección 8.1. Si bien los puntos de historia no indican directamente el número de horas que tomará cada actividad, permiten obtener una buena estimación de la cantidad de trabajo que representa una historia de usuario o un conjunto de ellas, con respecto a un sprint o a la totalidad del proyecto. Esto resulta especialmente útil para darle un valor a la cantidad de trabajo faltante en un sprint, que consistiría en la suma de los puntos de historia de todas las historias de usuario correspondientes al sprint en cuestión, que todavía no han sido completadas.

Para representar esta información gráficamente se utilizará un Burndown Chart, que ilustra la cantidad de trabajo restante a lo largo del tiempo. Esta gráfica representa el tiempo en el eje x y la cantidad de trabajo faltante en el eje y, que en este caso será representada a través del total de puntos de historia de las historias de usuario restantes. Por lo tanto, esto permite visualizar la correlación entre la cantidad de trabajo restante en cualquier punto del tiempo y el progreso del equipo de desarrollo en la reducción de esta cantidad de trabajo/tareas. Para cualquier punto en el tiempo, la intersección entre la línea de tendencia de la gráfica y el eje horizontal indicará la fecha de finalización más probable [44].

Teniendo en cuenta que cada sprint tiene una fecha de finalización fija (sección 8.2.2), en cada punto en el tiempo es posible trazar una recta que atraviese el punto actual y la fecha límite, obteniendo una representación del comportamiento que debería tener la gráfica en el tiempo restante del sprint para cumplir con la fecha límite. Si los puntos de la gráfica se ubican por encima de esta recta, significa que el proyecto no está progresando según lo esperado y resulta necesario tomar acciones correctivas por parte del líder del proyecto y el revisor del proyecto (sección 6.1 y 6.4.2).

En el desarrollo de cada sprint, el Burndown Chart será actualizado cada vez que una historia sea completada, y este será revisado por el equipo en cada uno de los Scrums semanales. En cada una de estas reuniones, se trazará la recta que indica el comportamiento que debería tener la gráfica en el tiempo restante para lograr cumplir con la fecha límite. Si en la reunión se identifica que los puntos más recientes se encuentran por

encima de esta recta, se debe identificar la razón. Para esto, los integrantes del grupo expondrán las posibles causas de esto, como podrían ser una estimación incorrecta de la complejidad de la historia de usuario, falta de organización en el equipo o factores externos. Una vez identificadas las causas de estas deficiencias en el progreso del sprint, el scrum máster, en conjunto con el líder de proyecto, tomarán la decisión de qué acciones correctivas tomar, también con la participación de los demás miembros del equipo de desarrollo (sección7.1 y 7.4.2).

Para el progreso, también se tendrán en cuenta los criterios de aceptación, que son las condiciones que un producto de software debe satisfacer para ser aceptado por el cliente (sección 7.3). En la reunión de planeación de sprint, se establecerán los criterios de aceptación de cada una de las historias de usuario que se implementarán en ese sprint. Estos criterios se escribirán siguiendo un formato de Dado/Cuando/Entonces: “dado un escenario, cuando se realiza esta acción, entonces se espera que pase esto”. Es clave que en estos criterios sean independientes de la implementación, por lo que se debe especificar qué es lo que tiene que ocurrir en cada caso, sin entrar en detalle acerca de qué manera implementar la funcionalidad [45].

Para que una historia de usuario sea considerada como completa, se debe revisar que la implementación siga ciertos parámetros de calidad. En primer lugar, se deben cumplir con los requisitos funcionales y los criterios de aceptación indicados en la especificación de la historia de usuario, pero además se deben completar exitosamente las pruebas unitarias indicadas en la sección 10.4 y, cuando aplique, contar con la documentación mínima y completar exitosamente las pruebas del sistema, las pruebas automatizadas y las pruebas de rendimiento explicadas en la misma sección [46].

## Cierre del proyecto

El objetivo del cierre del proyecto es llevar un seguimiento del trabajo que se está realizando para evaluar cada entrega y así poder detectar errores que se están cometiendo o qué pueden mejorar para garantizar un producto de calidad.

Previo a cada entrega se realizará una reunión entre los integrantes del grupo en la cual se revisarán los siguientes aspectos:

* Calidad de los documentos.
* Cumplimiento de las tareas asignadas a cada integrante.
* Coherencia y consistencia de la información.
* Desempeño de cada rol.
* Comunicación por parte del grupo.
* Cumplimiento del cronograma.
* Uso de la bibliografía y relación con el documento.

A partir del resultado de los aspectos mencionados, se harán los cambios necesarios para corregir y ajustar las tareas y el cronograma en las entregas posteriores. De cada reunión realizada se tendrá un acta en el cual se anotará cuando fue la reunión y cuál fue el motivo de esta, junto con las conclusiones a las cuales se llegaron en esa reunión. Además, el Líder de proyecto tendrá que llevar a cabo el reporte gerencial del cual se sacaran conclusiones para tenerlas en cuentas a lo largo del proyecto y poder hacer el respectivo post-mortem de la fase.

Además, se analizará las causas de los problemas que surjan, los impactos de esos problemas, acciones para solucionar los problemas, el desempeño de los integrantes del grupo, y lecciones aprendidas en el desarrollo de cada entrega para futuros proyectos y trabajos.

# Entrega del producto

Para que la entrega del producto final sea satisfactoriamente recibida por el cliente, el equipo de desarrollo recorrerá un proceso de actividades, tareas y metodologías que ayudarán a la organización para la entrega final del producto. Dividiendo el proceso de desarrollo del proyecto en tres entregas, las primeras dos entregas constaran de un documento (PMP Y SRS respectivamente) donde es registrado los avances y requisitos cumplidos acordados entre el equipo de desarrollo y el cliente (ver sección 6.4). El último entregable consistirá en entregar el software completo funcionando e implementado junto con un documento SDD. Todas las entregas tendrán un historial de cambios donde se visualizará las modificaciones de los responsables de actualizar el documento, es decir, los miembros de equipo de proyecto. También se tendrá en cuenta una fecha limite acordada con el cliente.

Una vez el software final se encuentre listo para entregar, se harán unas pruebas para monitorear el funcionamiento del software en la organización del cliente. Una vez el equipo de trabajo haya realizado estas pruebas y esté satisfecho, se realizará una reunión con el cliente para desplegar e introducir el producto, en donde el equipo de desarrollo explicará en detalle todas las características del software implementado. Esta reunión tendrá como objetivo capacitar al cliente para que se familiarice con el nuevo software.

Por último, el cliente probará el producto y navegará a través de la aplicación, probando todas las funcionalidades como usuario para que tenga claridad de como funcionara el nuevo sistema en su organización. Después de eso, se hará un seguimiento para monitorear el acoplamiento del cliente y el nuevo sistema y para proveer el soporte necesario. Este seguimiento comenzará siendo semanal, pero a medida que pase el tiempo, los periodos entre seguimientos se harán más extendidos.

# Procesos de soporte

## Ambiente de trabajo

**Reglas para regular el trabajo en equipo**

1. Los integrantes del grupo se tratarán respetuosamente entre sí.
2. Todas las semanas se realizarán reuniones a las que deberán asistir todos los integrantes.
3. Los integrantes del grupo se esforzarán para que los trabajos presentados tengan el mayor nivel de calidad.
4. Se deberán utilizar explícitamente las herramientas de comunicación especificadas en la sección 7.2.
5. Los integrantes del grupo deberán responder lo antes posible a los mensajes relacionados con las actividades del grupo.
6. La planificación de las tareas a realizar, sus fechas y responsables se realizarán exclusivamente a través de las herramientas especificadas en sección 7.2.
7. Para la elección de una fecha de finalización de una tarea, deberá haber consenso entre los integrantes del grupo.
8. Para posponer la fecha de finalización de un trabajo, deberá haber consenso entre los integrantes del grupo.
9. Al momento de asignar labores, los integrantes del grupo deberán evitar comprometerse a realizar tareas cuando estos tengan conocimiento de que en realidad no podrán cumplirlas.
10. Los integrantes del grupo deberán avisar con 2 días de anticipación, como mínimo, si consideran que no podrán cumplir con una tarea previamente asignada.
11. En caso de que un integrante del grupo deba ausentarse repentinamente como resultado de una emergencia debidamente justificada, este deberá informarle al grupo lo antes posible y los demás integrantes deberán reasignar las tareas urgentes previamente asignadas al integrante ausente.
12. Los integrantes del grupo deberán comprometerse a ayudar a otros integrantes que soliciten ayuda en labores del grupo siempre que hayan avisado con suficiente anticipación que no pueden llevar a cabo la tarea en cuestión.
13. Los documentos que se trabajen deberán estar subidos en los repositorios especificados en la sección 7.2, para que sea posible trabajar de manera conjunta y también controlar y revisar el trabajo realizado por los demás integrantes del grupo.
14. Si alguno de los integrantes presenta un comportamiento que va en contra de estas reglas, el caso puede ser sometido a votación para determinar si a este miembro se aplica una falta grave. Si el 50% de los integrantes no involucrados deciden aplicar la falta grave, esta será aplicada. El líder del proyecto es el encargado de llevar el registro de las faltas graves de los integrantes.
15. Si un integrante acumula 3 faltas graves, será expulsado del grupo. El líder de proyecto será el encargado de informarle esto al profesor.

**Estrategias de trabajo**

1. Al momento de realizar las actividades, mantenerse conectado a los canales de comunicación establecidos, para asegurar un buen ritmo de trabajo, asegurar la colaboración entre los miembros y facilitar el avance.
2. Realizar o proponer citas de trabajo entre los integrantes del grupo para realizar un trabajo constante y consistente.
3. Al momento de planear la realización de una tarea, en la medida de lo posible, se establecerá una fecha de finalización del trabajo previa a la fecha de entrega, para que haya suficiente tiempo para llevar a cabo la revisión y evitar que el trabajo se acumule.
4. Al momento de realizar las revisiones, procurar cambiar de integrantes para asegurar la integración colaboración y calidad.
5. Al momento de realizar actividades extensas, buscar intervalos de pausas para mantener el ritmo y la calidad del trabajo.
6. Sin importar lo insignificante o pequeña que parezca una duda su opinión, consultarla con el grupo.
7. Las tareas de mayor dificultad y/o importancia serán realizadas y revisadas por más de una persona.

## Análisis y administración de riesgos

En cuanto al análisis de riesgos, ante una posible desviación de lo esperado con consecuencias positivas o negativas, el equipo de desarrollo tomará varios procesos de identificación, planeación y plan de tratamiento para estos riesgos que se puedan presentar durante el desarrollo del proyecto. Se tendrá en cuenta las categorías en donde se clasifican los diferentes tipos de riesgos, utilizando técnicas y herramientas de vital importancia para el adecuado análisis e investigación de dichos riesgos.

En primer lugar, se debe realizar una identificación estricta de los riesgos. Para ello, se investigarán varias técnicas y herramientas, teniendo como objetivo encontrar las técnicas que mejor se ajusten al contexto del proyecto y de la actualidad. Las técnicas y herramientas que van a ser usadas en el proceso de identificación serán las listas rápidas y reuniones virtuales (talleres de riesgos), estas dos son las que mejor se acoplan al contexto del proyecto.

Para generar la lista rápida, de manera individual cada miembro del equipo analizará todos los posibles riesgos que se puedan presentar, posteriormente se realizara una reunión general donde se compartirán dichos riesgos encontrados y el equipo llegar a un consenso de cuales riesgos van a ser parte de la lista final.

Este es el subproceso “hacer lista rápida individualmente” donde cada integrante del grupo realiza individualmente la identificación de riesgos y luego la realización de una lista rápida que será utilizada en el siguiente proceso.

Ilustración 6 Diagrama BPMN de subproceso "hacer lista rápida individualmeente"

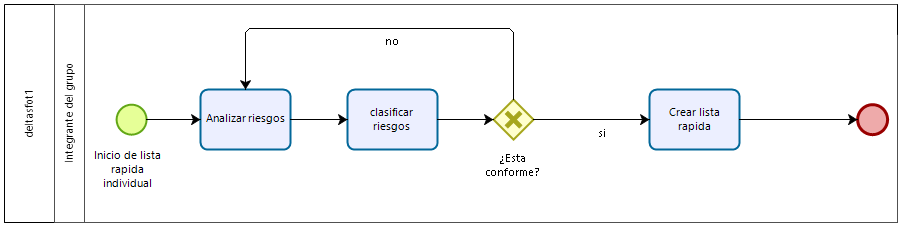
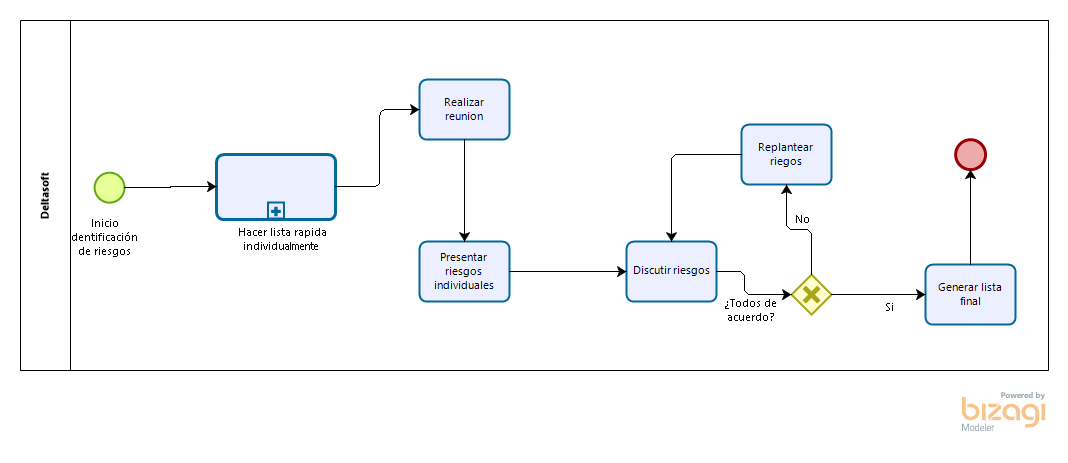
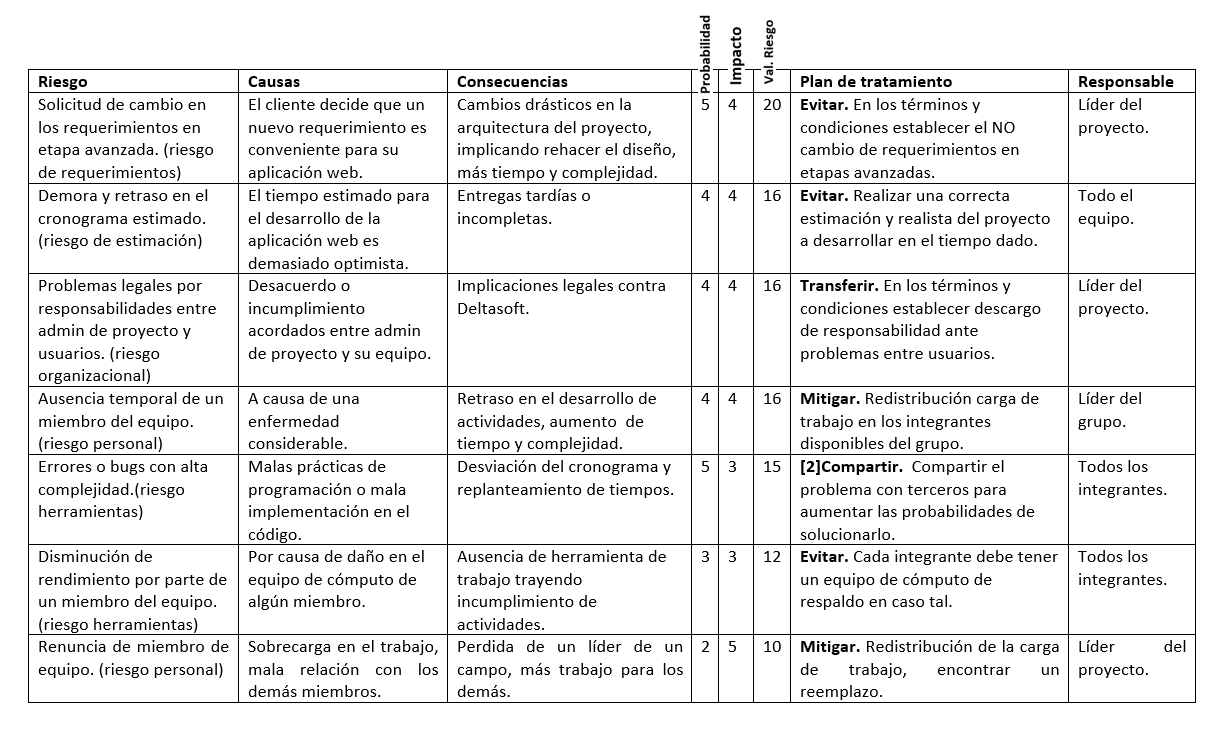
Este es el proceso completo de la identificación de riesgos que los integrantes del grupo realizaran. Cabe aclarar que este mismo modelo de proceso se llevará a cabo a lo largo de todo el desarrollo.

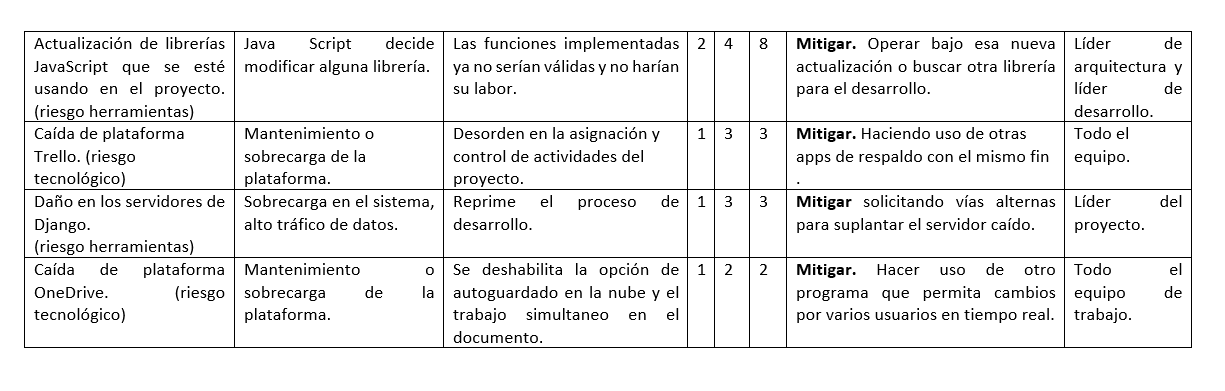
Ilustración 7 Diagrama BPMN de proceso de identificación de riesgo



Para analizar y gestionar el plan de riesgos se realizarán dos tablas. El contenido de la primera consistirá en todos los riesgos identificados por el equipo de desarrollo y estarán en orden por prioridad. Esta prioridad es el resultado del puntaje de la probabilidad y el impacto que tenga el riesgo. La probabilidad y el impacto será medida de 1 –5 siendo 1 poca probabilidad/impacto y 5 mayor probabilidad/impacto, siendo la prioridad la multiplicación de estos dos factores. En cuanto a la segunda tabla, tendrá esta misma información, pero solo de los riesgos más importantes [47].

Tabla 12 Riesgos ordenados por prioridad





## Administración de configuración y documentación

A continuación, se hablará de las actividades correspondientes a la configuración del software y de la documentación que afectan la página web. Estas actividades las hará el líder de configuración, el cual se encargará de identificar la configuración del sistema en los diversos pasos del tiempo, con el fin de mantener la integridad y calidad del producto. Para esto el grupo se guiará mediante el estándar ISO 1007 [48] y el estándar IEEE 828-2005 [49]. Lo que se busca con esto será:

* Tener control sobre cada fase, de los productos generados en estas, y de los cambios evaluados y aprobados.
* Se deberá incluir en el control de configuración a la mayor cantidad de productos posibles, teniendo en cuenta las restricciones y duración del proyecto.
* Al concluir cada fase se realizará una prueba de verificación de funcionalidad.

La primera actividad es identificar cuáles serán los elementos de configuración en nuestro proyecto. Estos son:

* Documento SPMP.
* Documento SRS.
* Documento SDD.
* Código fuente.
* Pruebas Unitarias.

Para estos documentos se tendrá una identificación de versiones que será secuencial simple de tres dígitos enteros no negativos, para distinguir entre cambios funcionales, mejoras y correcciones de errores. El digito de la extrema izquierda indica la entrega del proyecto, el digito del centro indica la versión del documento y el último digito indica modificaciones de información en el contenido, por parte de los integrantes de desarrollo. Esto se verá apreciado en la sección 1.

En cuanto al código fuente y las pruebas unitarias, se tendrán que revisar al finalizar cada sprint y cuando un integrante del grupo termine su tarea respectivamente para obtener un buen desarrollo de nuestro producto.

**Proceso de control de cambios:** Este proceso inicia cuando el cliente o algún integrante del grupo solicita un cambio, el cual tendrá que ser evaluado por el líder de configuración. Si este es denegado tendrá que informar que el cambio no se realizará y terminará el proceso. Si no, se le asignara la tarea a algún integrante del grupo y este hará el respectivo cambio para ser revisado nuevamente por el líder de configuración. Si cumple con los requisitos del cambio se integrará y se finalizará el proceso.

Ilustración 8 Diagrama BPMN de proceso de control de cambio

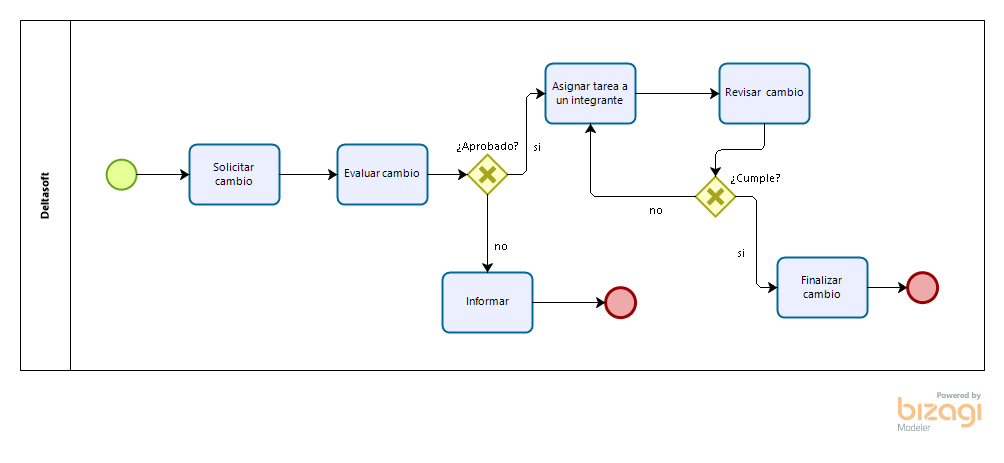


Tabla 13 Ítems de configuración

|  |  |  |
| --- | --- | --- |
| Ítem Identificado | Control | Auditoria |
| Documento SPMP | Este documento tendrá su creación el día 31/08/2020 y finalizará el 09/09/2020. Además, se modificará cada vez que el revisor de proyecto encuentre algún elemento por corregir. | Una vez llegue la fecha de finalización se solicitará una reunión con el cliente para ver si este cumple con los requerimientos especificados. De no ser así este pasara a ser modificado nuevamente. |
| Documento SRS | Este documento tendrá su creación el día 14/09/2020 y finalizará el 09/10/2020. Además, se modificará cada vez que el revisor de proyecto encuentre algún elemento por corregir. | Una vez llegue la fecha de finalización se solicitará una reunión con el cliente para ver si este cumple con los requerimientos especificados. De no ser así este pasara a ser modificado nuevamente. |
| Documento SDD | Este documento tendrá su creación el día 19/10/2020 y finalizará el 20/11/2020. Además, se modificará cada vez que el revisor de proyecto encuentre algún elemento por corregir. | Una vez llegue la fecha de finalización se solicitará una reunión con el cliente para ver si este cumple con los requerimientos especificados. De no ser así este pasara a ser modificado nuevamente. |
| Código fuente | El código fuente será desarrollado desde la segunda entrega hasta la entrega final por lo que se iniciará el 14/09/2020 y finalizará el 20/11/2020 y tendrá modificaciones constantes. | Una vez finalizado el primer sprint, se tendrá que revisar el código teniendo en cuenta la retroalimentación dada por el cliente. De igual manera, al terminar el segundo sprint, este tendrá que estar por lo menos en un 70% de funcionalidad. |
| Pruebas unitarias | Esta actividad será desarrollada cuando un integrante del grupo acabe con alguna tarea de los sprints, por lo que se realizará a lo largo de la segunda y tercera entrega. | Una vez finalizada la prueba, si funciona correctamente no tendrá ningún problema, pero si falla tendrá que ser revisada por el líder de desarrollo hasta encontrar una solución a dicho problema. | |

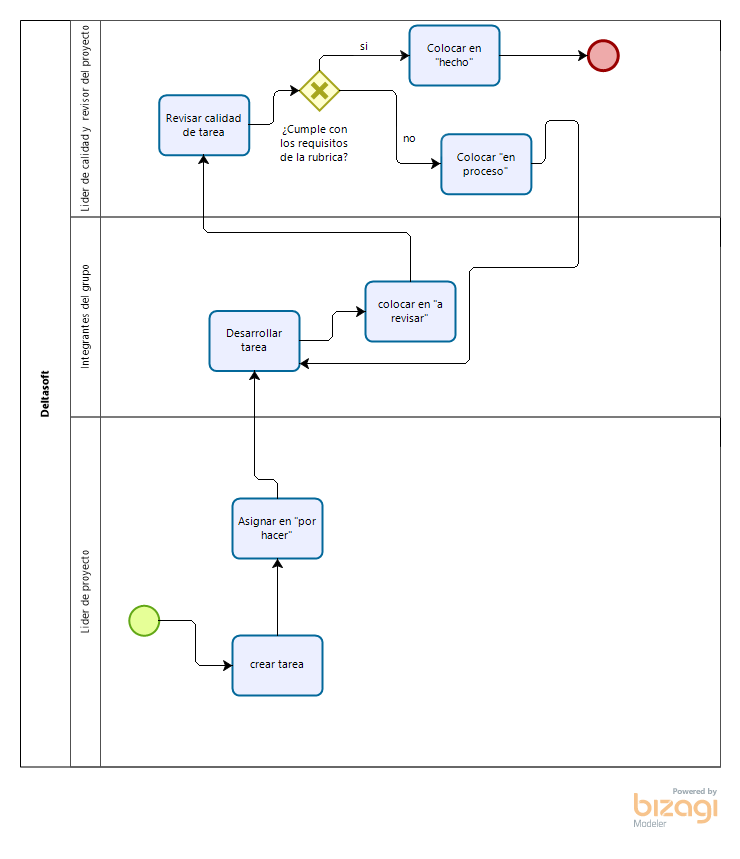
## Control de calidad

Deltasoft implementará diversas técnicas y herramientas en los procesos relacionados con el control de calidad de los avances (verificación, validación, revisión y auditoria) con el fin de supervisar y mejorar la calidad. A continuación, se describe cada herramienta que va a ser utilizada y como va a ser hacer usada mediante párrafos explicativos y diagramas BPMN, con el fin de que todos los integrantes tengan claridad en las labores en cuanto a calidad se refiere. Pero que en ultimas el líder de calidad y el líder del proyecto puedan observar los resultados y evaluarlos para aceptarlos o rectificarlos.

* **Revisión de documentos:** se usará la estrategia de tener un tablero que organice la cantidad de tareas asignadas por el líder de proyecto a integrantes o grupo de integrantes del equipo, alojándose en la columna de tareas “por hacer”.

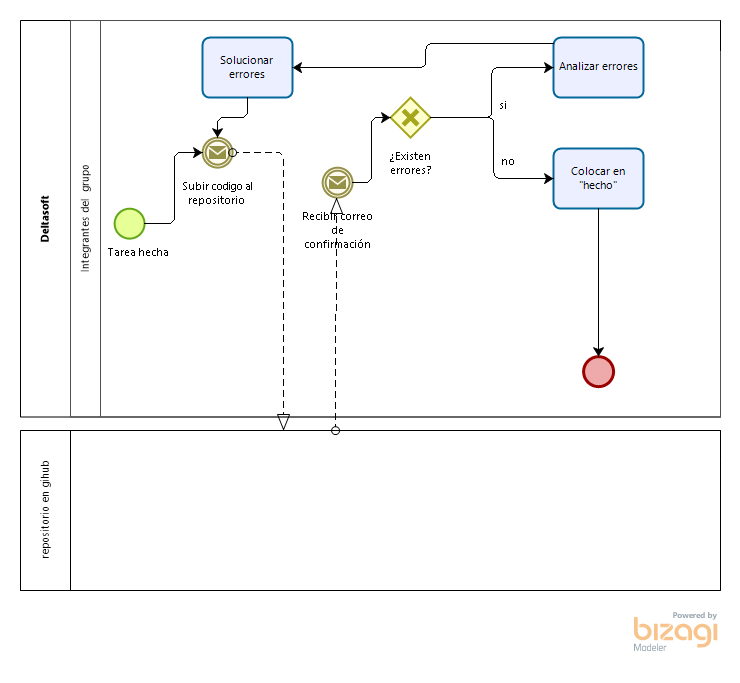
En una siguiente etapa la tarea es desarrollada y puesta en la columna llamada “a Revisar” en la cual entran otro grupo al proceso, este grupo es conformado por dos actores (líder de calidad y revisor del proyecto) los cuales revisan la tarea y manifiestan su opinión en cuanto a la calidad y el contenido, para que de esta manera sea aprobado o sea necesario una refactorización de la tarea. Este proceso se lleva a cabo cada vez que se realice una tarea relacionada con documentación.

Ilustración 9 Diagrama BPMN de proceso de revisión de documentos



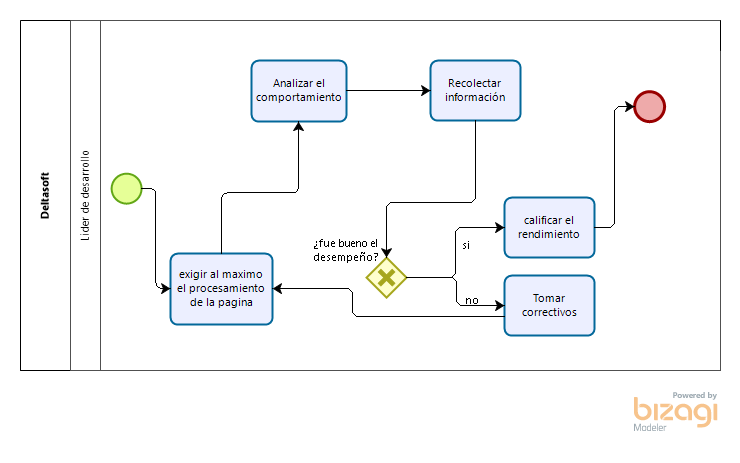
* **Pruebas unitarias**: Las pruebas unitarias se llevarán a cabo por cada integrante del grupo una vez termine el avance de su tarea aislándola del resto de código. Esta prueba se le hará a cada clase creada y se verificará su correcto funcionamiento individual teniendo también en cuenta el manejo de excepciones para garantizar el control de errores y verificación de métodos. Todo este proceso estará acompañado de su respectivo documento de plan de pruebas. Esta tarea se lleva a cabo cada vez que se termine una tarea de desarrollo de código en el tiempo del sprint.

Ilustración 10 Diagrama BPMN de proceso de pruebas unitarias



* **Pruebas del Sistema:** Estas pruebas son llevadas a cabo cuando el sistema ya tenga cierto nivel de funcionalidad (al menos dos unidades estén integradas). en esta prueba a partir de un plan de pruebas el líder de desarrollo integra que se cumpla los requerimientos del cliente y las interfaces funcionen de manera correcta entre ellas. Al ser una tarea con un grado de complejidad elevado, deberá realizarse por dos o más integrantes. Esta tarea se lleva a cabo cada vez que se entregue un adelanto que complementen a algo que ya está hecho dentro del tiempo del sprint.

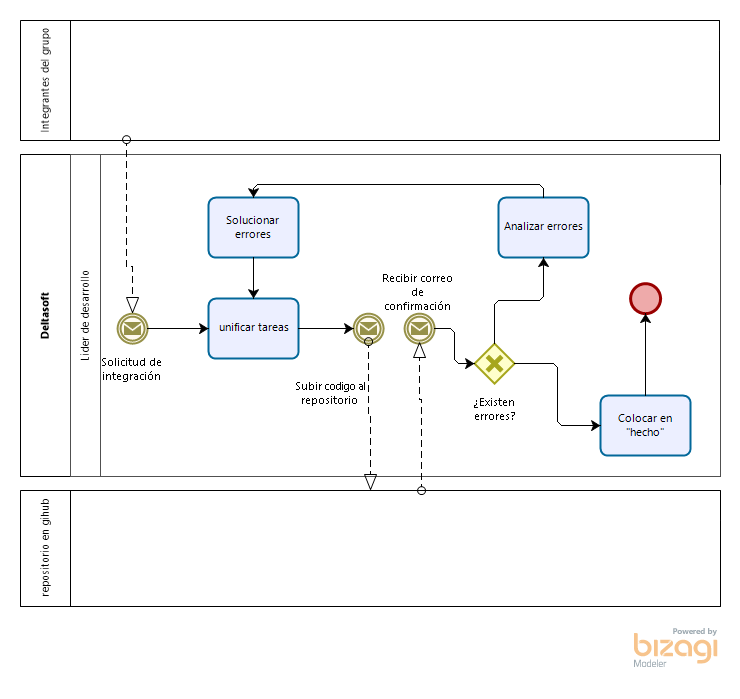
Ilustración 11 Diagrama BPMN de proceso de pruebas del sistema



* **Prueba de rendimiento:** Cuando la aplicación complete el 50% y el 90% de sus funcionalidades desarrolladas se realizará una prueba de rendimiento la cual consiste en que el líder de calidad y líder de configuración sometan a la aplicación web a una carga presada de procesos para así recoger datos del tiempo y el comportamiento de los procesos. Con esto se puede calificar el rendimiento de lo desarrollado y en caso de no ser favorable tomar medidas para mejorar el rendimiento no solo en procesamiento si no en respuestas.

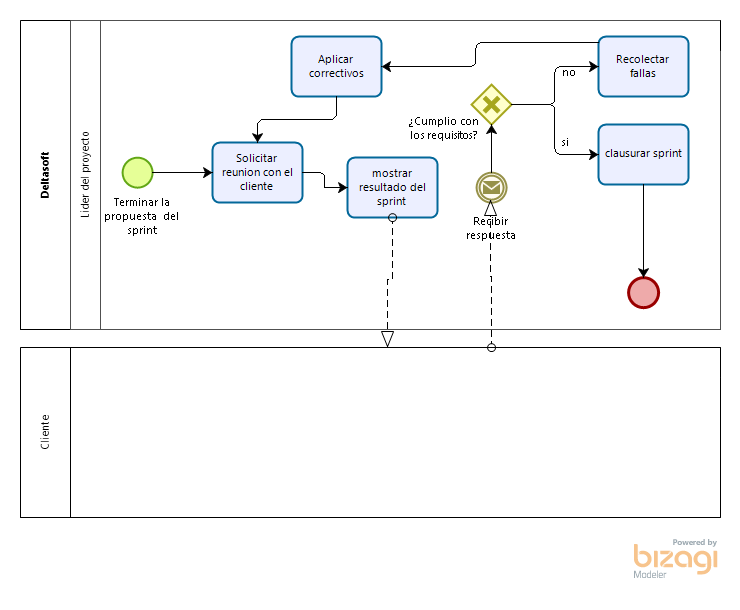
Diagrama de prueba de rendimiento:

Ilustración 12 Diagrama BPMN de proceso de prueba de rendimiento



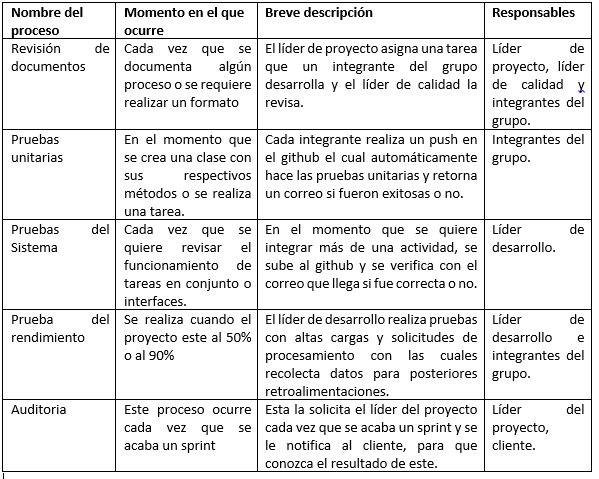
* **Auditoria:** En cuanto a la auditoria que se va a implementar en este proyecto, existen dos tipos: el primer tipo va a ser auditoria con el cliente, la cual nos va a proporcionar una prueba fiable para saber si todo el equipo está cumpliendo el objetivo y los requerimientos por los cuales el cliente nos contrató. Esta prueba se realizará una vez de a conocer el entregable al final de cada sprint. La manera en la cual es evaluado dicho producto está ligada a los entregables planteados para cada sprint, puesto que este tendrá un número determinado de funciones que el usuario podrá aceptar mediante el criterio de aceptación definido en la sesión 8.2 (dado/cuando/entonces), y de esta manera poder concluir que es aprobado si más del 85% de estas historias fueron exitosas, en caso tal que un porcentaje menor a este sea aprobado el entregable no aprobó la auditoria y sería necesario aplicar modificaciones. [50]

Ilustración 13 Diagrama BPMN de proceso de auditoria



Nota: algunas de estas pruebas se harán de manera automatizada usando la herramienta “GitHub Actions”, que nos proporciona la capacidad de realizar pruebas unitarias y pruebas de integración de manera automática cada vez que se realiza un “push” a nuestro repositorio. De ese modo se puede saber de forma inmediata si ese nuevo código falla o, de lo contrario, funciona correctamente y es añadido la rama principal de forma satisfactoria [51].

Tabla 14 Herramientas para control de calidad



# Anexos

1. Documentación de estimaciones
2. Diagrama de Gantt
3. Reporte gerencial (actas de reuniones)
4. Presentación
5. Diagrama BPMN AdministracionRequisitos

# Referencias

|  |  |
| --- | --- |
| [1] | D. W. White, «SPMP (SOFTWARE PROJECT MANAGEMENT PLAN),» PONTIFICIA UNIVERSIDAD JAVERIANA, Colombia, 2013. |
| [2] | Universidad ICESI, «SRS / ERS Especificación de requerimientos de software,» 15 10 2010. [En línea]. Available: https://www.icesi.edu.co/departamentos/tecnologias\_informacion\_comunicaciones/proyectos/lisa/home/analisis/srs/srs#:~:text=El%20est%C3%A1ndar%20IEEE%20830-1998,el%20grupo%20de%20desarrollo%20para. |
| [3] | F. d. J. Nuñez, «DOCUMENTO DE DISEÑO DE SOFTWARE,» UAEH, 2017. |
| [4] | TutorialsInhand, «Software Engineering Introduction,» 2020. [En línea]. Available: https://tutorialsinhand.com/tutorials/software-engineering-tutorial/software-engineering-introduction/software-engineering-home.aspx. |
| [5] | I. C. Society, «IEEE Standard for Information Technology—Systems Design—Software Design Descriptions,» IEEE, New York, 2009. |
| [6] | M. A. Awad, A Comparison between Agile and Traditional Software Development Methodologies, Australia, 2005. |
| [7] | ScienceDirect, Incremental Development, ScienceDirect, 2003. |
| [8] | S. Kashyap, «Traditional vs Agile Project Management Method: Which One is Right for Your Project?,» ProofHub, 13 Noviemre 2018. [En línea]. Available: https://www.proofhub.com/articles/traditional-vs-agile-project-management. |
| [9] | L. R. Vijayasarathy y C. W. Butler, «Choice of Software Development Methodologies,» IEEE COMPUTER SOCIETY, Colorado, 2018. |
| [10] | Roadmunk, «8 popular software development methodologies—with pros and cons,» Roadmunk, 2019. |
| [11] | Velvetech, «Top 12 Software Development Methodologies: Benefits and Drawbacks,» Velvetech, 12 Febrero 2020. [En línea]. Available: https://www.velvetech.com/blog/software-development-methodologies/. |
| [12] | G. Fischer, Automated Software Engineering, University of Colorado, 2002. |
| [13] | I. Sommerville, Software Engineering, England: Pearson Education Limited, 2016. |
| [14] | N. M. A. Munassar y A. Govardhan, A Comparison Between Five Models Of Software Engineering, IJCSI International Journal of Computer Science Issue, 2010. |
| [15] | Y. Obergfell, «Scrum Roles – The Scrum Team - International Scrum Institute,» International Scrum Institute, 2020 . [En línea]. Available: https://www.scrum-institute.org/Scrum\_Roles\_The\_Scrum\_Team.php. |
| [16] | K. Schwaber y J. Sutherland, The Scrum Guide, 2017. |
| [17] | M. Hammarberg y J. Sundén, Kanban IN ACTION, Shelter Island, NY: MANNING, 2014. |
| [18] | J. LEBOW, What Is Kanban? An Introduction to Kanban Methodology, Digital.ai Software Inc., 2020. |
| [19] | B. Yu y D. J. Malan, «CS50’s Web Programming with Python and JavaScript,» Harvard University, 2020. |
| [20] | Microsoft, «Microsoft 365 and Office 365 service descriptions,» 14 7 2020. [En línea]. Available: https://docs.microsoft.com/en-us/office365/servicedescriptions/office-365-service-descriptions-technet-library. |
| [21] | Microsoft, «Office applications,» 14 7 2020. [En línea]. Available: https://docs.microsoft.com/en-us/office365/servicedescriptions/office-applications-service-description/office-applications. |
| [22] | Microsoft, «OneDrive service description,» 16 7 2020. [En línea]. Available: https://docs.microsoft.com/en-us/office365/servicedescriptions/onedrive-for-business-service-description#:~:text=OneDrive%20for%20work%20and%20school,sync%20files%20to%20your%20computer.. |
| [23] | Mozilla and individual contributors, «HTML: Hypertext Markup Language,» 2020. [En línea]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML. |
| [24] | MIT, ERCIM, Keio, Beihang, «HTML & CSS,» W3C, 2016. |
| [25] | Python, «What is Python? Executive Summary,» Python Software Foundation, 2020. |
| [26] | M. Solutions, «Python: 7 Important Reasons Why You Should Use Python,» Medium, 3 10 2017. [En línea]. Available: https://medium.com/@mindfiresolutions.usa/python-7-important-reasons-why-you-should-use-python-5801a98a0d0b. |
| [27] | P. Statz, «Get Started with Web Frameworks,» *WIRED,* 15 2 10. |
| [28] | M. Makai, «Django,» Full Stack Python, 2020. [En línea]. Available: https://www.fullstackpython.com/django.html. |
| [29] | Mozilla and individual contributors, «JavaScript basics,» MDN web docs, 2020. [En línea]. Available: https://developer.mozilla.org/en-US/docs/Learn/Getting\_started\_with\_the\_web/JavaScript\_basics. |
| [30] | J. Rottinger, «The Role of JavaScript in the Modern Web,» Medium, 4 7 2019. [En línea]. Available: https://medium.com/better-programming/the-role-of-javascript-in-the-modern-web-ff0f6961829a. |
| [31] | Microsoft, «Visual Studio Code,» Microsoft, 2020. [En línea]. Available: https://code.visualstudio.com/docs. |
| [32] | The PostgreSQL Global Development Group, «About SQL,» The PostgreSQL Global Development Group, 13 08 2020. [En línea]. Available: https://www.postgresql.org/about/. |
| [33] | L. BRADFORD, «What Is GitHub?,» 24 07 2020. [En línea]. Available: https://www.thebalancecareers.com/what-is-github-and-why-should-i-use-it-2071946. |
| [34] | Bizagi, «Bizagi Modeler,» 2020. [En línea]. Available: http://help.bizagi.com/process-modeler/en/. |
| [35] | Microsoft, «Microsoft Project,» Microsoft, 2020. [En línea]. Available: https://www.microsoft.com/en-us/microsoft-365/project/project-management-software. |
| [36] | WhatsApp Inc., «WhatsApp,» 2020. [En línea]. Available: https://www.whatsapp.com. |
| [37] | M. Rouse, «Google Hangouts,» TechTarget, 2020. [En línea]. Available: https://searchmobilecomputing.techtarget.com/definition/Google-Hangouts. |
| [38] | Trello, «What is Trello?,» Trello, 16 10 2017. [En línea]. Available: https://help.trello.com/article/708-what-is-trello. |
| [39] | Microsoft, «Microsoft Teams,» Microsoft, 2020. [En línea]. Available: https://www.microsoft.com/en-US/microsoft-365/microsoft-teams/group-chat-software. |
| [40] | CESSI ARGENTINA, «PERFILES DESARROLLO DE SOFTWARE,» Cámara de la Industria Argentina del Software., 2020. [En línea]. Available: https://www.cessi.org.ar/perfilesit/perfil-desarrollo-de-software-1. |
| [41] | Rational Software Corp., «Rational Unified Process,» Rational Software Corporation, 25 05 2002. [En línea]. Available: https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/process/workers/ovu\_works.htm. |
| [42] | Visual Paradigm, «What is Planning Poker in Agile?,» Visual Paradigm, 2020. [En línea]. Available: https://www.visual-paradigm.com/scrum/what-is-agile-planning-poker/. |
| [43] | PlanITpoker, «Pure & Simple Planning,» PlanITpoker, 2020. [En línea]. Available: https://www.planitpoker.com/board/#/room/001fcc86f43a46aa9189e76931ca6dcc. |
| [44] | K. Schwaber, «What Is Scrum?,» Volaro. |
| [45] | S. POVILAITIS, «ACCEPTANCE CRITERIA,» Leading Agile, 9 9 2014. [En línea]. Available: https://www.leadingagile.com/2014/09/acceptance-criteria/. |
| [46] | D. HUETHER, «DEFINITION OF DONE,» LeadingAgile, 8 2 2017. [En línea]. Available: https://www.leadingagile.com/. |
| [47] | Nextop, «Como hacer el Plan de Gestión de Riesgos en Proyectos,» Nextop, 18 3 2017. [En línea]. Available: https://nextop.es/como-hacer-plan-de-gestion-de-riesgos/. |
| [48] | ISO , «ISO 10007:2017 Gestión de la calidad — Directrices para la gestión de la configuración,» ISO , 2017. [En línea]. Available: https://www.iso.org/obp/ui#iso:std:iso:10007:ed-3:v1:es. |
| [49] | IEEE, «828-2005 - IEEE Standard for Software Configuration Management Plans,» IEEE, 14 2 2005. [En línea]. Available: https://standards.ieee.org/standard/828-2005.html. |
| [50] | Instituto de Auditoría Interna y Gobierno Corporativo, «AUDITORÍA ÁGIL CON SCRUM,» enel, Chile, 2019. |
| [51] | GitHub, Inc, «GuitHub Docs,» GitHub, Inc, 2020. [En línea]. Available: https://docs.github.com/es/actions. |