



Facultad de ingeniería

Departamento de ingeniería de sistemas

Plan de proyecto

Aplicativo web para el apoyo de fundaciones en los procesos de adopción animal en la ciudad de Bogotá.

Juan Camilo Pimiento Gómez

Gabriel Andrés Niño Carvajal

Ricardo Riscanevo Cotrina

Juan Felipe Vanegas Patiño

14/05/2021

Historial de Cambios

Fecha de cambio	Sección	Descripción	Encargado
11/05/2021	6.3 Supuestos y restricciones	Se plantean los diferentes supuestos y restricciones que se utilizarán para llevar a cabo el proyecto.	Gabriel Niño
11/05/2021	6.4 Entregables	Se realizó la tabla respectiva de los entregables del proyecto.	Juan Felipe Vanegas
11/05/2021	6.5 Resumen de Calendarización y Presupuesto	Se realizó una tabla donde se nombraron las etapas del proyecto y los momentos en que deben ser terminadas	Juan Camilo Pimienta
11/05/2021	6.6 Evolución del plan	Se realizó la descripción del proceso llevado a cabo cuando es necesario realizar cambios en el PMP	Juan Felipe Vanegas
11/05/2021	9.1. Métodos y herramientas de estimación	Se realizó la descripción de las herramientas a usar para estimar los tiempos de las historias de usuario	Juan Felipe Vanegas
11/05/2021	8.2 Lenguajes y herramientas	Se realiza una descripción de los diferentes lenguajes y herramientas que podrían ser utilizados para el desarrollo del proyecto.	Gabriel Niño
11/05/2021	8.3 Plan de aceptación del producto	Se describió los criterios empleados para la aceptación de los entregables	Ricardo Riscanevo
12/05/2021	8.4.1 Interfaces externas	Se describieron las interfaces externas con las que interactuaría el equipo de desarrollo	Juan Camilo Pimienta

12/05/2021	8.4.2 Organigrama y descripción de roles	Se realiza el organigrama junto con la asignación y descripción de roles.	Gabriel Niño
12/05/2021	9.2.1 Entrenamiento del personal	Se describió cómo y en qué momento, los integrantes del grupo de capacitarían para el uso de las herramientas necesarias para realizar el proyecto.	Juan Camilo Pimienta
12/05/2021	9.3.1 Descomposición de actividades	Se identificó las principales actividades del proyecto y se modeló empleando un WBS	Felipe Vanegas y Ricardo Riscanevo
12/05/2021	10.1 Administración de requerimientos	Se describió el proceso llevado a cabo al ejecutar un cambio de requisitos	Ricardo Riscanevo
12/05/2021	10.2 Monitoreo y control de progreso	Se identificó el monitoreo que se realiza para medir el progreso del proyecto	Ricardo Riscanevo
12/05/2021	10.3 Cierre del proyecto	Se realiza una breve descripción de los pasos a seguir para llevar a cabo el cierre del proyecto.	Gabriel Niño
12/05/2021	12.2.1 Descripción del plan de riesgo	Se mostró las matrices de riesgos, además de un plan de contingencia para los mismos	Juan Felipe Vanegas
12/05/2021	12.3 Administración y configuración de la documentación	Se realizó la tabla con los momentos de los entregables e ítems de configuración.	Juan Camilo Pimienta
13/05/2021	9.3.2 Calendarización	Se realizó la calendarización de las actividades a desarrollar en el proyecto	Juan Felipe Vanegas y Ricardo Riscanevo

Prefacio

El documento SPMP (Software Project Management Plan) hace referencia al plan de gestión que se llevará a cabo en un proyecto de software. En este documento se mostrará un plan detallado en el cual se otorgarán roles a los distintos miembros del equipo de trabajo, se hablará de las herramientas que se utilizarán para el desarrollo exitoso del mismo, además de un plan de contingencia para los posibles riesgos que puedan presentarse en el desarrollo del mismo, calendarización de las fechas de entrega de documentos y prototipos, y una metodología a llevar a cabo en el proceso antes, durante y después del desarrollo del proyecto.

Tabla de Contenidos

Historial de Cambios.....	3
Prefacio.....	6
Lista de Figuras	9
Vista General del Proyecto	11
6.1 Visión del Producto.....	11
6.2 Propósito, Alcance y Objetivos	11
6.3 Supuestos y Restricciones	11
6.4 Entregables.....	11
6.5 Resumen de Calendarización y Presupuesto.....	12
6.6 Evolución del Plan.....	13
Glosario.....	15
Contexto del proyecto	16
8.1 Modelo de Ciclo de Vida.....	16
8.2 Lenguajes y Herramientas	16
8.2.1 Herramientas y lenguajes de desarrollo.....	16
8.2.2 Desarrollo Web.....	16
8.2.2.1 Base de datos y despliegue de la aplicación.....	17
8.2.3 Herramientas de control de versiones y hosting de versiones.....	18
8.2.4 Herramientas de Modelado.....	19
8.2.4.1 Modelado de procesos de negocio	19
8.2.5 Manejo de documentos.....	19
8.2.6 Manejo del proyecto	20
8.3 Plan de Aceptación del Producto.....	21
8.4 Organización del Proyecto y Comunicación	22
8.4.1 Interfaces Externas.....	22
8.4.2 Organigrama y Descripción de Roles.....	23
Administración del Proyecto.....	25
9.1 Métodos y Herramientas de Estimación.....	25
9.2 Inicio del proyecto	26
9.2.1 Entrenamiento del Personal.....	26
9.2.2 Infraestructura.....	27
9.3 Planes de Trabajo del Proyecto.....	27
9.3.1 Descomposición de Actividades.....	27

9.3.2 Calendarización	27
Monitoreo y Control del Proyecto	28
10.1 Administración de Requerimientos.....	28
10.2 Monitoreo y Control de Progreso.....	29
10.3 Cierre del Proyecto.....	30
Entrega del Producto.....	32
Procesos de Soporte.....	33
12.1 Ambiente de Trabajo	33
12.2 Análisis y Administración de Riesgos	33
12.2.1 Descripción del plan de riesgos	33
Fase 1. Identificación de riesgos.....	33
Fase 2. Análisis de riesgos	33
Fase 3. Planificar la respuesta	35
Fase 4. Supervisar el riesgo	36
12.3 Administración de Configuración y Documentación	36
Métricas y Proceso de Medición.....	37
13.1 Control de Calidad	37
Anexos	42
Referencias	43

Lista de Figuras

Ilustración 1. Evolución del plan.	14
Ilustración 2. Administración de requerimientos.	29
Ilustración 3. Riesgo 1.	34
Ilustración 4. Riesgo 2.	34
Ilustración 5. Riesgo 3.	34
Ilustración 6. Riesgo 4.	34
Ilustración 7. Riesgo 5.	35
Ilustración 8. Riesgo 6.	35
Ilustración 9. Riesgo 7.	35
Ilustración 10. Control de cambios de ítems de configuración.....	37
Ilustración 11. Control de calidad de documentos.	39
Ilustración 12. Control de calidad en el sprint planning.	41

Lista de Tablas

Tabla 1. Entregables.....	12
Tabla 2. Fases del proyecto.....	12
Tabla 3. Presupuesto.....	13
Tabla 4. Glosario.	15
Tabla 5. Plan de aceptación del producto.....	21
Tabla 6. Plan de aceptación.	22
Tabla 7. Stakeholders del proyecto.....	23
Tabla 8. Descripción de roles.	25
Tabla 9. Entrenamiento del personal.....	26
Tabla 10. Infraestructura.	27
Tabla 11. Análisis de riesgos.	33
Tabla 12. Plan de contingencia.	36
Tabla 13. Administración de configuración y documentación.....	36
Tabla 14. Roles para el proceso de control de calidad.	38
Tabla 15. Roles para control de calidad del software.....	40

Vista General del Proyecto

6.1 Visión del Producto

Dirigirse al documento versión final de la propuesta (VFP) en la sección **1.1 Antecedentes, problema y solución propuesta**

6.2 Propósito, Alcance y Objetivos

Dirigirse al documento versión final de la propuesta (VFP) en la sección **1.2 Descripción general del proyecto**

6.3 Supuestos y Restricciones

Supuestos:

- Los intervalos de tiempo para desarrollar los *Sprint* deben ser respetados para cumplir con las fechas de entrega de los entregables establecidos.
- Los tiempos de entrega deben cumplirse de manera obligatoria siguiendo la *diagramación Gantt*.
- Contaremos con el apoyo de ciertas fundaciones/refugios ubicados en la ciudad de Bogotá; estas nos proporcionarán la información de los animales.
- No ocurrirá ningún cambio con respecto al director del trabajo de grado.
- Cada integrante será responsable por aprender a hacer uso de los lenguajes y herramientas seleccionados para la implementación del proyecto.

Restricciones:

- El tiempo de planeación y desarrollo corresponde al primer y segundo semestre académico de 2021 respectivamente.
- El proyecto únicamente funcionará como una plataforma web con diseño adaptable, en cualquier dispositivo electrónico que pueda conectarse a una red de internet.
- La plataforma únicamente estará disponible en Bogotá.
- Algunos elementos del proyecto como el *hosting* y uso de APIs puede ser que requieran hacer un pago para asegurar el mejor funcionamiento del producto final.

6.4 Entregables

En la **tabla 1**, se muestran cuáles son los entregables que se van a presentar a lo largo del proyecto, en donde cada entregable posee una descripción, un cliente y una fecha de entrega.

Entregable	Descripción	Cliente	Fecha
SRS	Especificación de requerimientos de software. Lleva consigo las recomendaciones finales o requisitos de software. Tiene consigo la documentación general para el desarrollo del proyecto.	María Isabel Serrano. Carlos Antonio Navarrete.	24/05/2021
SDD	Descripción del Diseño de Software. Es una descripción de un software, que el equipo escribe con el fin de tener orientación general para el desarrollo del proyecto.	Carlos Antonio Navarrete.	30/06/2021

Prototipos del programa	Es la implementación en código de los casos de uso más relevantes, con el fin de mostrar el comportamiento de la aplicación.	Fundaciones participantes del proyecto. Carlos Antonio Navarrete	15/08/2021 24/10/2021
Documentos de prueba	Este documento define los objetivos de prueba de un sistema, además de establecer y coordinar la estrategia de trabajo para proveer un marco adecuado en la elaboración de una planificación paso a paso de las actividades de prueba.	Carlos Antonio Navarrete	Por definir
Manuales de usuario	Son manuales de instalación y uso, requeridos para que los usuarios tengan una guía adecuada y pertinente para el buen uso e instalación de la aplicación.	Fundaciones participantes del proyecto. Carlos Antonio Navarrete	Por definir

Tabla 1. Entregables.

6.5 Resumen de Calendarización y Presupuesto

En la siguiente tabla se observarán las distintas etapas del desarrollo del proyecto además de los momentos donde se planea que estén terminados.

Fase	Descripción	Fecha tentativa
Análisis	Durante esta fase se realizará el levantamiento de los requisitos a las fundaciones, además de planear el proyecto para definir el alcance de este definido.	La fase de análisis debe estar terminada antes del inicio del diseño.
Diseño y arquitectura	Durante la fase de diseño, tal como dice su nombre, se realizará el diseño del sistema con el fin de realizar la arquitectura general del proyecto.	La fase de diseño y arquitectura debe estar terminada al terminar la materia de Planeación del Proyecto Final.
Desarrollo e integración	Durante esta fase se utilizarán los diagramas y modelos creados en la fase previa, esto con el fin de desarrollar los prototipos funcionales en un código de programación.	Se realizará una entrega de código cada 8 días desde el inicio de la materia "Trabajo de grado" hasta la conclusión del proyecto.
Pruebas	Durante la fase de pruebas se verificará la correcta operación de las diferentes funcionalidades que hacen parte del programa a medida que vayan siendo terminadas.	Esta etapa se realizará cada vez que se vaya a realizar una entrega de código nueva.

Tabla 2. Fases del proyecto

Presupuesto

A continuación, se tendrán en cuenta los siguientes ítems en la elaboración del presupuesto para el desarrollo del proyecto.

Ítem	Costos
Almacenamiento	0 dólares hasta 5GB de almacenamiento [1]
API de Google Maps para JavaScript	7 dólares desde 1000 solicitudes [2]
Hosting	3 dólares al mes hasta 10.000 visitas [3]

Tabla 3. Presupuesto.

Ya que el software es intangible, los costos que debe asumir el proyecto se limitan a hosting, APIs y bases de datos.

Ya que en un principio se utilizará la herramienta de almacenamiento para fines de desarrollo y pruebas, no se va a pagar nada ya que no hay pago para uso de hasta 5GB de almacenamiento.

El uso de la API de Google Maps para JavaScript funciona de una manera similar, y como se mencionaba antes, ya que al principio se utilizará únicamente para fines de desarrollo y pruebas. En caso de que en el futuro (después de la culminación del trabajo de grado) se quiera utilizar esta aplicación, las fundaciones deberán asumir los costos de esta.

Finalmente, el precio del hosting será de apenas 3 dólares por mes, esto debido a que en un principio el volumen de visitas no será elevado.

6.6 Evolución del Plan

Para el desarrollo de cada sección del Plan de Proyecto (PMP), se realiza cortas reuniones en las cuales se discute de manera grupal el contenido de cada sección a realizar y se asigna a los integrantes una sección la cual debe ser finalizada en un tiempo límite.

Cuando se requiere realizar un cambio en el Plan del proyecto, se sigue un proceso con el objetivo evitar que los cambios a realizar afecten significativamente las fechas establecidas en otras actividades, además se garantice una coherencia entre las secciones del documento. Para llevar a cabo la ejecución de un cambio se sigue el siguiente proceso:

Informar a los miembros del equipo la sección en la cual se identificó que se requiere realizar un cambio. En el caso de ser identificado al mismo tiempo más de un cambio, se organiza estos cambios para ser analizados en una reunión del equipo.

Realizar una reunión entre todos los integrantes del equipo para discutir los cambios que se van a realizar en el plan de proyecto, priorizando aquellos cambios que son necesarios. Es importante tener en cuenta el impacto que tiene este cambio, debido a que puede desencadenar cambios en otras secciones en desarrollo o ya finalizadas. Debido a esta situación, es necesario tener claro que secciones tienen dependencia de otras.

Asignar a uno o varios integrantes del equipo la actividad para realizar el cambio, teniendo en cuenta el número de actividades que ya tienen asignados los integrantes y el conocimiento que posean para el desarrollo de la sección. Es necesario establecer una fecha límite para la finalizar el cambio, y no perjudicar las demás actividades que ya tiene una fecha de entrega programada. Esta fecha se establece a través de una estimación grupal sobre el tiempo que llevara el desarrollo de la actividad.

Al llegar la fecha límite para la actividad se presenta al equipo cuales fueron los cambios realizados, para que se revise si cumple con la calidad deseada. Esta revisión de calidad es llevada a cabo por el director de calidad asignado en la sección **8.4.2 Organigrama y descripción de roles**, el cual indica a los encargados de la actividad si cumple con los estándares de calidad adecuados para considerar la actividad como finalizada. En caso de no cumplir con la calidad suficiente, es necesario ejecutar una nueva reunión y volver asignar una actividad para llevar a cabo las modificaciones en los cambios realizados.

Al establecer que la actividad fue finalizada y el cambio aceptado, se modifica el historial de cambios, describiendo la sección modificada y la persona que fue asignada a dicho cambio.

A continuación, en la **Ilustración 1** se muestra en un diagrama BPMN el proceso llevado a cabo para realizar un cambio en el PMP. En este proceso todos los integrantes del equipo son participes y conocen los cambios que vas a ser realizados en el Plan de Proyecto.

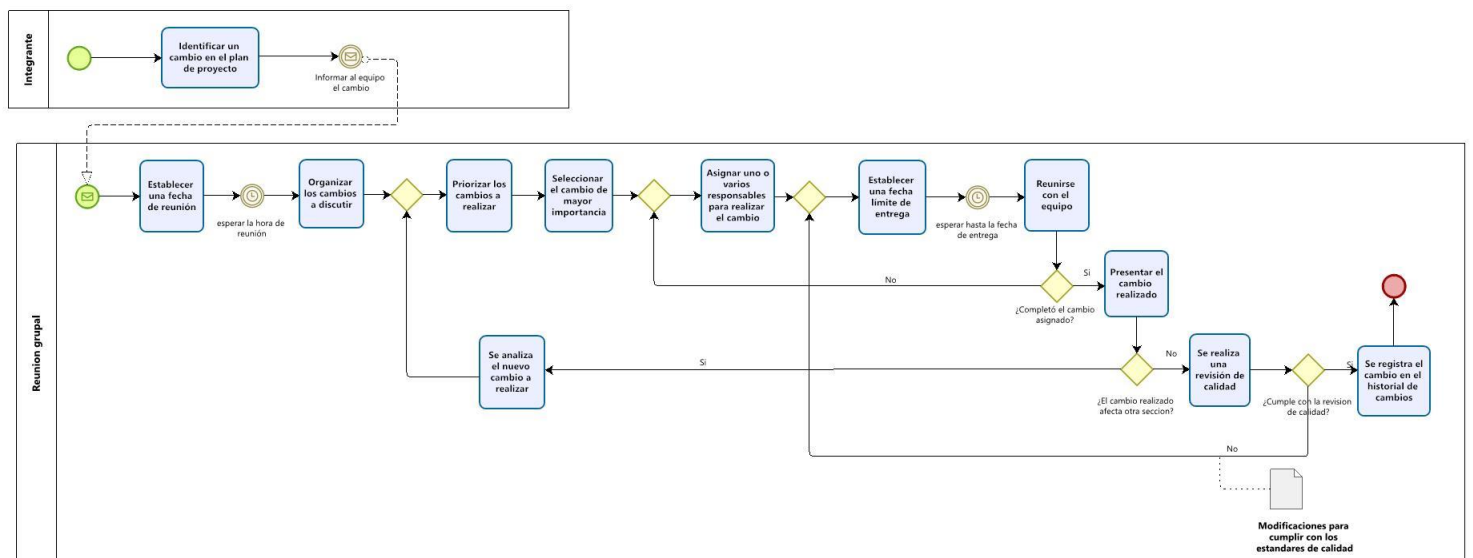


Ilustración 1. Evolución del plan.

Glosario

Término	Definición
Front-end[7]	Es la parte de una aplicación que interactúa con los usuarios; es todo lo que se visualiza en pantalla cuando se accede a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas, teclado, movimientos, efectos visuales.
Back-end[7]	Hace referencia al interior de las aplicaciones que viven en el servidor; consiste en un servidor, una aplicación y una base de datos. Se toman los datos, se procesa la información y se envía al usuario.
Librería [9]	Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.
Framework [8]	Es una estructura conceptual y tecnología de soporte definido, muy útil para la organización y desarrollo de software
BPMN [10]	Son diagramas basados en la técnica de "Diagrama de flujo", diseñado para presentar la secuencia que se sigue en el desarrollo de actividades, a través de diferentes elementos gráficos
API	Application Programming Interfaces por sus siglas en inglés, corresponde a interfaces de programación de aplicaciones, lo cual es un conjunto de definiciones y protocolos que permite la comunicación entre las aplicaciones
Hosting	El hosting es un servicio que provee a los usuarios de Internet un sistema de alojamiento web donde se permitirá la publicación de un sitio o aplicación.
Artefacto	Producto de software que surge en el proceso de diseño y desarrollo de un proyecto. Este puede incluir documentos, diagramas entre otros elementos
Daily Scrum	Reunión diaria en que los integrantes del equipo informan el avance realizado en el transcurso del día
Sprint	Iteración o lapso de tiempo en el cual los integrantes del equipo desarrollar unas actividades asignadas en una reunión
Prototipo [4]	Herramienta que permite agilizar y obtener mejor desempeño en la fase de desarrollo. Se refiere a un producto de aplicación que muestra la funcionalidad actual de un producto, pero en realidad no tiene la lógica ni la calidad que tendrá el producto final.

Tabla 4. Glosario.

Contexto del proyecto

8.1 Modelo de Ciclo de Vida

Dirigirse al documento versión final de la propuesta (VFP) en la sección **3.4.1 Método**

8.2 Lenguajes y Herramientas

A continuación, se presentarán los lenguajes y herramientas a utilizar durante el desarrollo del proyecto; las cuales se seleccionaron con los criterios definidos por el grupo

8.2.1 Herramientas y lenguajes de desarrollo

Para la selección de lenguajes y herramientas se presentan los siguientes criterios:

1. **Curva de aprendizaje:** Los lenguajes y las herramientas seleccionadas deben tener una sintaxis clara, además de contar con soporte y comunidad para una solución eficiente a los problemas que se presenten, y con esto simplificar el trabajo.
2. **Tecnología web:** La tecnología seleccionada permite la ejecución del proyecto en cualquier navegador web, con la posibilidad de agregar el formato específico de cada navegador para su correcta visualización.
3. **Fácilmente escalable:** La tecnología seleccionada permita trabajar en módulos para agregar nuevas funcionalidades al proyecto, sin que se generen conflicto con las versiones anteriores de este.
4. **Nivel de personalización:** La tecnología cuenta con componentes personalizables, los cuales se pueden adaptar con los dispositivos que se estén utilizando.

8.2.2 Desarrollo Web

Para el desarrollo se tuvieron en cuenta los entornos de desarrollo, *frameworks*, lenguajes y librerías. De las cuales se encuentran las siguientes alternativas:

- **Visual Studio:** Es un editor de código fuente desarrollado por *Microsoft*, ligero pero potente que se ejecuta en escritorio y está disponible para *Windows*, *macOS* y *Linux*. Viene con soporte incorporado para *JavaScript*, *TypeScript* y *Node.js* y tiene un rico ecosistema de extensiones para otros lenguajes. [11]
- **Atom:** Es un editor de texto de código abierto creado por *GitHub* (ahora propiedad de *Microsoft*) que permite personalizar fácilmente todos sus aspectos. Toda la aplicación está escrita en *JavaScript*, *CSS* y *HTML*. Este se encuentra disponible para *Windows*, *macOS* y *Linux*. [12]
- **HTML:** Significa **Lenguaje de Marcado de Hipertextos (*HyperText Markup Language*)**, es la pieza más básica para la construcción de la web y se usa para definir el sentido y estructura del contenido en una página web. [13]
- **Bootstrap:** Es un *framework front-end* gratuito para desarrollo web rápido y sencillo, este incluye plantillas de diseño basadas en *HTML* y *CSS* para tipografía, formularios, botones, tablas, navegación, etc, así como complementos de *JavaScript* opcionales. [14]
- **Foundation:** Un *framework* para cualquier dispositivo, medio y accesibilidad. *Foundation* es una familia de marcos *front-end* receptivos que facilitan el diseño de sitios web, aplicaciones y correos electrónicos receptivos que se ven en cualquier dispositivo. *Foundation* es semántico, legible, flexible y completamente personalizable. [15]
- **React:** Es una librería de *JavaScript* para crear interfaces de usuario. *React* facilita la creación de interfaces de usuario interactivas utilizando vistas declarativas que hacen que el código sea más predecible, más simple de entender y más fácil de depurar. Dado que la lógica de los componentes está escrita en *JavaScript* en lugar de en plantillas,

puede pasar fácilmente datos enriquecidos a través de su aplicación y mantener el estado fuera del DOM. [16]

- **Vue:** Es un *framework* progresivo para construir interfaces de usuario. Vue está diseñado desde cero para ser adoptable gradualmente. La biblioteca principal se centra solo en la capa de vista y es fácil de recoger e integrar con otras bibliotecas o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar aplicaciones sofisticadas de una sola página cuando se usa en combinación con herramientas modernas y bibliotecas de soporte. [17]
- **Angular:** Es un *framework* de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones de una sola página eficientes y sofisticadas. Angular está escrito en *TypeScript*. Angular utiliza componentes los cuales definen vistas, que son conjuntos de elementos de pantalla entre los que Angular puede elegir y modificar de acuerdo con la lógica y los datos de su programa. [18]

Las herramientas y tecnologías seleccionadas para este proyecto son:

- **Visual Studio Code:** aunque tanto VSC como Atom poseen gran variedad de *plugins* personalizables, se decidió por escoger VSC como el editor sobre el cual se desarrollará el proyecto, ya que este ofrece mejores *plugins* para el desarrollo web bajo las tecnologías seleccionadas (descritas a continuación). Además, el grupo de trabajo tiene experiencia en el manejo de este editor.
- **HTML, CSS y JavaScript:** estas tecnologías son obligatorias en el desarrollo web, ya que no hay otras opciones en el mercado.
- **Angular:** se optó por utilizar este *framework* debido a la serie de ventajas que trae sobre las otras opciones contempladas. Este permite crear vistas de interfaz de usuario con una sintaxis simple. Además, contiene herramientas que ayudan a añadir componentes, realizar pruebas y previsualizar de forma instantánea la aplicación.
- **Bootstrap:** se escogió este *framework* ya que provee muchas herramientas para la personalización de componentes gráficos, lo que ahorra tiempo en el desarrollo general de la interfaz gráfica de usuario y provee un diseño más amigable, además disminuye la carga de trabajo con CSS. Otra razón por la que se escogió *Bootstrap* es por su facilidad de integración con *Angular*.

8.2.2.1 Base de datos y despliegue de la aplicación

Para seleccionar el manejador de base de datos y el despliegue de esta se tomaron estos criterios:

1. Debe proveer un servicio de *hosting* que permita el fácil despliegue de la aplicación.
 2. La base de datos debe ser en tiempo real.
 3. La herramienta de despliegue debe proveer mecanismos para la autenticación de usuario.
- **Firebase:** Es una plataforma de desarrollo de aplicaciones web y móviles, respaldada por Google. *Firebase* administra su propia infraestructura con un buen conjunto de herramientas para simplificar el flujo de trabajo del desarrollador proporcionándoles kits de desarrollo y un panel en línea. Estos kits de herramientas están interconectados, son escalables e integrables con *software* de terceros para superar desafíos complejos con bloques de construcción estándar. *Firebase* cuenta con *Realtime Database* y *Cloud Firestore* que pueden almacenar datos estructurados en documentos y sincronizar las

aplicaciones correspondientes en milisegundos cada vez que se produce una transformación de datos. [19]

- **MongoDB:** Esta es una base de datos NoSQL basada en documentos. Mongo está diseñada para tener mayor facilidad en desarrollo y escalabilidad de la aplicación. Actualmente cuenta con diferentes ediciones; Mongo DB Community es la edición de software libre disponible. También existe la posibilidad de hacer uso de Atlas el cual es un servicio de la edición Mongo DB *Enterprise* que permite guardar datos en la nube. [20]
- **Azure:** Azure *Database* ofrece una selección de bases de datos relacionales, NoSQL y en memoria totalmente administradas, que abarcan motores patentados y de código abierto, para adaptarse a las necesidades de los desarrolladores de aplicaciones actuales. La administración de la infraestructura, incluidas la escalabilidad, la disponibilidad y la seguridad, está automatizada, lo que ahorra tiempo y dinero. Azure proporciona información sobre el rendimiento con el uso de inteligencia integrada, modifica su escala sin límites y administra las amenazas de seguridad. [21]

8.2.3 Herramientas de control de versiones y hosting de versiones

Para las herramientas de control de versiones y *hosting* se definieron los siguientes criterios:

1. La herramienta debe proveer de manera completa todas las opciones para un correcto manejo de las versiones de los ítems de configuración durante el proyecto.
2. El equipo debe tener estar familiarizado con la herramienta elegida.
3. Debe tener una curva de aprendizaje moderada.

Se consideraron dos herramientas para el manejo del control de versiones en el proyecto:

- **Git:** Es el sistema de control de versiones más utilizado en la actualidad. Git es un sistema de control de versiones distribuido, lo que significa que su copia local del código es un repositorio de control de versiones completo. Estos repositorios locales completamente funcionales hacen que sea fácil trabajar sin conexión o de forma remota. Confirma su trabajo localmente y luego sincroniza su copia del repositorio con la copia en el servidor. La flexibilidad y la popularidad de Git lo convierten en una excelente opción para cualquier equipo. [22]
- **Subversión:** Es una herramienta que se utiliza para mantener versiones actuales e históricas de proyectos. Subversión es un sistema de control de versiones centralizado de código abierto. Tiene licencia de Apache. [23]

Ya planteadas estas dos herramientas de control de versiones, se ha tomado la decisión de trabajar con Git, ya que cumple con los criterios, además de ser una de las herramientas más utilizadas en el mercado laboral.

Para la plataforma de hosting de versiones, se consideraron las siguientes:

- **GitHub:** Es una plataforma de alojamiento de código para el control de versiones y la colaboración. Permite a las personas trabajar juntos en proyectos desde cualquier lugar. [24]

- **GitLab:** Es una plataforma *DevOps* completa, entregada como una sola aplicación. Esto hace que GitLab sea único y crea un flujo de trabajo de software optimizado, liberando a su organización de las limitaciones de una cadena de herramientas ensamblada. GitLab ofrece una visibilidad incomparable y niveles más altos de eficiencia en una sola aplicación a lo largo del ciclo de vida de DevOps. [25]

Se ha elegido GitHub ya que los integrantes han tenido experiencia en esta herramienta. Además, con la explicación dada en clase queda claro las posibilidades que se tiene con la elección de esta aplicación.

8.2.4 Herramientas de Modelado

Para la selección de herramientas de modelado, se definieron los siguientes criterios:

1. Debe proveer la gran mayoría de diagramas *UML*.
2. Debe proveer colaboración para equipos.
3. Los miembros del equipo deben estar familiarizados con la plataforma.

Las herramientas consideradas para el modelado *UML* fueron las siguientes:

- **StarUML:** Es una plataforma de modelado de software que soporta *UML (Unified Modeling Language)*. Se basa en la versión 1.4 de *UML* y proporciona once tipos diferentes de diagramas, y acepta la notación *UML 2.0*. Sobresale por su adaptabilidad al entorno del usuario y tiene una alta extensibilidad en su funcionalidad. Garantiza la maximización de la productividad y la calidad de sus proyectos de *software*. [26]

Se ha tomado la decisión del uso de *StarUML* ya cumple los criterios planteados. Además de que la página provee gran número de diagramas *UML* y permite la colaboración entre los miembros del trabajo en equipo.

8.2.4.1 Modelado de procesos de negocio

En cuanto al modelamiento BPMN las herramientas consideradas fueron:

- **Visual Paradigm:** Es una herramienta de software diseñada para que los equipos de desarrollo de software modelen el sistema de información empresarial y gestionen los procesos de desarrollo. Esta herramienta soporta los principales lenguajes y estándares de modelado de la industria. Ofrece un completo conjunto de herramientas de software que las empresas necesitan para la captura de requerimientos, análisis de procesos, diseño de sistemas, diseño de bases de datos, etc. [27]
- **Bizagi Modeler:** Es una aplicación gratuita para diagramar, documentar y simular gráficamente procesos para negocios que se rigen por las directrices del entorno BPMN 2.0. [28]

8.2.5 Manejo de documentos

Se definieron los siguientes criterios para la selección de las herramientas para el manejo de documentos:

1. Debe tener alojamiento en la nube, que permita la colaboración en tiempo real del equipo de trabajo.
2. Debe proveer una variedad de herramientas ofimáticas.
3. Debe proveer herramientas para el versionado de los documentos.

Las herramientas consideradas fueron las siguientes:

- **G Suite:** Es un conjunto de aplicaciones web creadas por *Google*. A través de esta herramienta tendrá acceso a una serie de potentes aplicaciones de *Google* la cual tendrá la capacidad de compartir hojas de cálculo y documentos, crear videoconferencias con *Hangouts* y utilizar la mensajería instantánea. También puede compartir calendarios con otras personas, lo que facilita la planificación de reuniones. [29]
- **Microsoft Office 365:** Es una versión basada en la web del conjunto de aplicaciones de productividad empresarial de *Microsoft Office*. *Office 365* se entrega a los usuarios a través de la nube e incluye *Exchange Online* para correo electrónico, *SharePoint Online* para colaboración, *Lync Online* para comunicaciones unificadas y un conjunto de *Office Web Apps*, versiones basadas en Web del conjunto de aplicaciones tradicional de *Microsoft Office*. [30]

Se ha escogido la herramienta *Microsoft Office 365*, ya que contamos con la licencia dada por la universidad, además de poder trabajar en tiempo real en sus herramientas ofimáticas, visualizando el trabajo realizado por cada integrante, además esta herramienta nos ofrece la comunicación entre el grupo por medio de *Microsoft Teams* y dejar registro de nuestras reuniones.

8.2.6 Manejo del proyecto

Para la selección de una herramienta para el manejo del proyecto, se especificaron los siguientes criterios:

1. Debe proveer una forma clara y sencilla para el manejo de proyectos.
2. Debe proveer facilidad a la hora de asignar tareas a los miembros del equipo.
3. Los miembros del equipo deben tener experiencia con la herramienta seleccionada.
4. Debe proveer notificaciones sobre las tareas asignadas en tiempo real.

Una vez planteados estos criterios se consideraron las siguientes alternativas:

- **Trello:** Es una herramienta de colaboración que organiza proyectos en tableros. A través de Trello podrá saber cuáles son las tareas que se llevan a cabo, quién trabaja en una tarea determinada y cuál es el estado de un proceso. [31]
- **Monday:** Es un sistema operativo de trabajo (Work OS) que permite a los equipos ejecutar proyectos y flujos de trabajo con confianza. Es un simple, pero intuitivo, Sistema Operativo de Trabajo para que los equipos den forma a los flujos de trabajo, se ajusten a las necesidades cambiantes, creen transparencia y se conecten en colaboración. Este es personalizable y puede ser usado para cualquier caso de uso. [32]

Se ha tomado la decisión de trabajar con ambas herramientas ya que Trello nos ofrece ventajas que facilitan la integración con otras herramientas. Esto es una gran ventaja a la hora de centralizar todas las tareas y procesos en materia de administración y gestión del proyecto. Para

poder realizar el seguimiento de las tareas en un diagrama de Gantt se ha utilizado *Monday* ya que es gratuita y permite llevar un seguimiento a cada actividad con una gran personalización.

8.3 Plan de Aceptación del Producto

Para hacer entrega de los diferentes productos o entregables del proyecto, es necesario revisar que cada uno de ellos cumplan con ciertos criterios de calidad. A continuación, se explica cada uno del criterio que aplican en los entregables del proyecto relacionados con documentación.

- **Completitud:** Los requisitos del entregable se satisfacen en su totalidad
- **Consistencia:** La información del documento es coherente entre todas sus secciones
- **Forma:** La redacción, presentación, lenguajes y criterios previamente nombrados son utilizados de forma adecuada.
- **Argumentación:** La información contenida dentro del entregable está debidamente sustentada y referenciada de ser el caso.
- **Investigación bibliográfica:** La información recolectada es extraída de fuentes confiables como libros, revistas o páginas web científicas.

En la sección **6.4 Entregables**, se explica cada uno de los entregables del proyecto que van a ser mencionados a continuación. En la **tabla 5** se indica los criterios que debe cumplir cada uno de los entregables para ser aceptados y entregados al cliente. Los entregables corresponden a la fase de análisis, diseño y pruebas mencionados en la sección **6.5 Resumen de calendarización y presupuesto**.

Entregable	Criterio
SRS	Completitud Consistencia Forma Argumentación Investigación bibliográfica
SDD	Completitud Consistencia Forma Argumentación Investigación bibliográfica
Documentos de prueba	Completitud Forma
Manuales de usuario	Completitud Forma

Tabla 5. Plan de aceptación del producto.

Además de los criterios de calidad establecidos para los documentos del proyecto, también se establecen criterios de calidad de software. A continuación, se presenta una lista de los criterios [33], junto con la **tabla 6** que indica los criterios que debe cumplir los entregables relacionados con el desarrollo de software.

- **Funcionalidad:** Que el software satisface las necesidades del usuario que fueron planteadas al inicio el proyecto
- **Confiabilidad:** Tiempo que está disponible para el usuario el software, brindándole un servicio tolerante a fallos
- **Usabilidad:** Grado en que el software sea fácil de usar para el usuario a nivel estético y operatividad
- **Eficiencia:** El software emplea los recursos del sistema óptimamente.

En la **tabla 6** se muestra los entregables de software a realizar en el proyecto y el criterio de calidad que debe cumplir.

Entregable	Criterio
Prototipos del programa	<ul style="list-style-type: none"> - Funcionalidad - Confiabilidad - Usabilidad - Eficiencia

Tabla 6. Plan de aceptación.

8.4 Organización del Proyecto y Comunicación

8.4.1 Interfaces Externas

Los *stakeholders* o clientes que hay en el proyecto se muestran en la siguiente tabla:

Nombre	Descripción	Responsabilidades	Datos de contacto
Fundaciones	Se estará en contacto desde el principio con las fundaciones que deseen hacer parte en un principio del proyecto. Estas fundaciones jugarán el papel de clientes en nuestro proyecto, ya que a partir de ellos se realizará la captura de requisitos.	Las fundaciones tienen la labor de darnos una guía sobre los procesos que realizan y los procesos que realizan los posibles adoptantes a la hora de querer adoptar una mascota.	-Entrevistas -WhatsApp -Instagram
Carlos Antonio Navarrete Puentes	Es a quien se entregará cada uno de los entregables, verificando que se cumpla con lo prometido.	Hará una retroalimentación luego de cada entrega, además de realizar la	-Teams -Correo electrónico -WhatsApp

		retroalimentación en cada reunión que se realice.	
--	--	---	--

Tabla 7. Stakeholders del proyecto.

8.4.2 Organigrama y Descripción de Roles

Para que el grupo tuviera una distribución de carga justa y eficiente se llevó a cabo el proyecto por medio de la metodología scrum, esto quiere decir que para cada miembro se estableció un rol específico (*Scrum master, Product Owner, Development team*). En la sección de anexos se puede encontrar el organigrama que describe la distribución de los roles del equipo.

A continuación, se presenta la **tabla 8** en la cual se pueden apreciar los diferentes roles que se van a considerar para el desarrollo de este proyecto. Así mismo, se presenta una breve descripción y responsabilidades que le corresponde a cada rol.

Rol	Descripción	Responsabilidades
<i>Product owner</i>	Se encarga de coordinar principalmente la gestión de requisitos. Es el que tiene la última palabra en las decisiones del proyecto	<ul style="list-style-type: none"> - Revisar el progreso de las funcionalidades en cada requisito. - Verificar el desempeño de todos los miembros del equipo. - Encargado de revisar las recomendaciones planteadas por el docente. - Diseñar planes para el manejo y control de riesgos.
<i>Scrum master</i>	Responsable de asegurar que <i>Scrum</i> es entendido y adoptado a lo largo del proyecto. También se debe mantener al equipo unido, solucionando los problemas que puedan ocurrir entre los miembros del equipo a lo largo del proyecto.	<ul style="list-style-type: none"> - Hacer cumplir el cronograma. - Proteger al equipo de interrupciones y distracciones. - Resolver conflictos internos. - Hacer respetar las decisiones tomadas democráticamente. - Asegurar la implementación de <i>Scrum</i>. - Organizar reuniones del equipo.
Director UI/UX	Se encarga de desarrollar las interfaces de la aplicación y la experiencia del usuario con la misma.	<ul style="list-style-type: none"> - Recopilar y evaluar los requisitos de los usuarios, en colaboración con el <i>Scrum Master</i> y <i>Project Manager</i>. - Ilustración de ideas de diseño utilizando guiones gráficos, flujos de procesos y mapas de sitio. - Diseñar elementos gráficos de la interfaz de usuario, como menús, pestañas y <i>widgets</i>.
Director documentación	Responsables de asegurar y monitorear la documentación del proyecto en todas sus etapas, tanto en el código de fuente como en el diseño del sistema.	<ul style="list-style-type: none"> - Definir el tipo de documentación que se necesita (técnica, manuales). - Monitorear la documentación tanto del código como del diseño.

		<ul style="list-style-type: none"> - Informar sobre la documentación al director de proyecto. - Asociarse con los directores de desarrollo y configuración para generar los manuales de instalación
Arquitecto <i>software</i>	Encargados de planear la arquitectura tanto lógica como física del proyecto de acuerdo con los requisitos planteados.	<ul style="list-style-type: none"> - Elegir las tecnologías para el proyecto a lo largo del semestre. - Desarrollar las funcionalidades con buenos estándares de programación, en lo posible el uso de patrones de diseño. - Limitarse a la arquitectura diseñada desde un principio. - Verificar que todos los componentes estén bien acoplados. - Comunicar cualquier novedad al <i>Scrum Máster</i>. - Levantar requerimientos. - Plantear la arquitectura del sistema.
<i>Frontend developer</i>	Desarrollador enfocado en la parte principal del sistema, enfocado en mejoras visuales y la interacción del usuario con el sistema.	<ul style="list-style-type: none"> - Traducir los requisitos e ideas en maquetación responsiva - Verificar el correcto funcionamiento de componentes visuales - Lograr una correcta adecuación y comunicación entre los componentes lógicos y funcionales con los componentes gráficos
<i>Backend developer</i>	Desarrollador enfocado en el funcionamiento trasero de la aplicación, su función es crear funcionalidades a nivel funcional sin importar la calidad gráfica del proyecto.	<ul style="list-style-type: none"> - Generar módulos acoplables y funcionales que pueden ser incorporados por los desarrolladores <i>front end</i>. - Verificar correcta funcionalidad de bases de datos y su correcta integración - Conexión de parte frontal con funcionalidades no visibles para clientes.
Administrador bases de datos	Encargado de la gestión de la base de datos a utilizar durante el proyecto.	<ul style="list-style-type: none"> - Administrar la estructura de la Base de Datos. - Administrar la actividad de los datos. - Administrar el Sistema Manejador de Base de Datos. - Establecer el Diccionario de Datos. - Asegurar la confiabilidad de la Base de Datos. - Confirmar la seguridad de la Base de Datos.
Director calidad	Se encargan de coordinar y monitorear las actividades relacionadas con gestión y control	<ul style="list-style-type: none"> - Revisar ortografía, forma, coherencia y cohesión del trabajo escrito.

	de la calidad en todas sus áreas. Así como los posibles riesgos para el proyecto y de gestionar planes para el tratamiento de éstos.	<ul style="list-style-type: none"> - Verificar el desempeño de todos los miembros del equipo. - Diseñar e implementar los índices de calidad. - Encargados de revisar las recomendaciones planteadas por el docente. - Diseñar planes para el manejo y control de riesgos.
Analista calidad	Miembro de apoyo y soporte para gestión de procesos de calidad y sostenibilidad del aplicativo	<ul style="list-style-type: none"> - Verificar mediante una segunda opinión la integridad y funcionamiento del aplicativo frente a diversos problemas, esto de la mano del director de QA
Director QA	Se asegura de que la aplicación cumpla adecuadamente la administración del cambio, la gestión de versiones, construcción del sistema y gestión de entregas.	<ul style="list-style-type: none"> - Guiará a los demás integrantes del grupo en el manejo del sistema de gestión de versiones. - Creará los repositorios necesarios para gestionar el almacenamiento de la información. - Encargado de llevar al tanto del directorio de versiones de los ítems de configuración.

Tabla 8. Descripción de roles.

Administración del Proyecto

9.1 Métodos y Herramientas de Estimación

Un proyecto de software se compone de distintas piezas, y se conocen varias técnicas de estimación; como lo son las empíricas o las que usan modelos de algorítmicos de costos del proyecto, en los cuales se tienen en cuenta el tamaño del software [5]

A pesar de dichas estimaciones anteriormente mencionadas, se debe aplicar técnicas de descomposición, ya sea para un problema o un proceso, ya que de esta forma el proyecto resulta ser más fácil de organizar. La estimación en la cual se basa el proceso se concentra en estimar el esfuerzo necesario para culminar cada uno de estos [6]. No obstante, la estimación basada en el problema busca dimensionar cada elemento de software por medio de métricas de productividad las cuales son las líneas de código y puntos de función. [6]

Estas estimaciones mencionadas realmente son las más clásicas, sin embargo, como se ha mencionado, para la posible realización de este proyecto se utilizará la metodología de scrum, por ende, técnicas de estimaciones ágiles. Así que, se implementará la técnica de *Planning Poker*.

9.1.1 Planning Poker

Escogimos este método principalmente porque permite conocer la opinión de cada miembro del grupo y le concede a este mismo poder tener un control sobre las actividades que se han llevado a cabo a lo largo de la semana.

Para la posible realización es necesario tener en cuenta el tiempo con el que se cuenta disponible para la finalización del proyecto, ya que con este se plantea el tiempo límite.

Para la posible realización de esta actividad de manera virtual, se va a hacer uso de la herramienta de *planning poker*, la cual permitirá votar por cada historia de usuario con el valor que se crea pertinente. El *scrum master*, que en este caso es Ricardo Riscanevo, es el encargado de llevar a cabo esta actividad, este también se encarga de revisar los valores obtenidos al final de la votación, en caso de tener una diferente valoración numérica se lleva a cabo una discusión en la cual los miembros del grupo que tengan el valor más alto y más bajo expresen las razones por las cuales dijeron dicho valor. Esta valoración numérica representa la cantidad de días ideales en los que se debe completar una historia de usuario en específico.

9.2 Inicio del proyecto

9.2.1 Entrenamiento del Personal

En esta sección se van a especificar los conocimientos en cuanto a *frameworks* y lenguajes de programación corresponde a cada uno de los miembros del grupo para el eficiente desarrollo del código del proyecto, que a su vez es la parte central del mismo. Por ello, a continuación, en la **tabla 9**, se presentarán los tiempos de aprendizaje que corresponden a cada herramienta utilizada para el desarrollo del proyecto, además de la estrategia a utilizar para la ejecución de este.

Herramienta	Miembros aprendices	Método de aprendizaje	Fechas tentativas
TypeScript	Todo el equipo	Tutoriales, foros de desarrollo, libros, guía del director de tesis.	13-06-2021/10-07-2021
MongoDB	Gabriel Niño	Tutoriales, guía del director de tesis, libros académicos.	13-06-2021/10-07-2021
Angular	Todo el equipo	Tutoriales de desarrollo en Udemy. Bootcamp de desarrollo web con el director de tesis a mitad de año.	13-06-2021/10-07-2021
GitHub	Todo el equipo	Bootcamp de desarrollo web con el director de tesis a mitad de año. Desarrollo de aplicaciones pequeñas utilizando control de versiones	13-06-2021/10-07-2021

Tabla 9. Entrenamiento del personal.

Por último, cabe resaltar que las tentativas fechas finales no son estrictas ya que se planea que durante el desarrollo del proyecto se pueda seguir aprendiendo de estas herramientas dado que, aunque no son complejas de usar, tienen muchas funciones detrás de sí, imposibles de aprender en su totalidad durante 2 meses. Lo que se planea es que, durante el periodo vacacional de mitad de año, se realice una capacitación exhaustiva de cada uno de los integrantes del grupo, para que a la hora de empezar el semestre se pueda dar comienzo al desarrollo del prototipo de la manera más eficiente posible.

9.2.2 Infraestructura

En la **tabla 10** se muestra los equipos de software y hardware que posee cada uno de los integrantes para el desarrollo del proyecto. Cada uno de estos equipos son de uso personal de los integrantes.

Nombre	Modelo	Procesador	Memoria RAM	SO
Ricardo Riscanevo	Gigabyte B450 M	AMD Ryzen 5 3400G 3.7 Ghz	16GB RAM 3000 Mhz	Windows 10
Juan Felipe Vanegas	ASUS X407UA	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz	20GB RAM 3000 Mhz	Windows 10
Gabriel Andrés Niño Carvajal	Lenovo IdeaPad C340	AMD Ryzen 5 3500U 2.1 GHz	8 GB RAM	Windows 10
Juan Camilo Pimienta Gómez	Alienware 15 R3	Intel (R) core (TM) i5-7300HQ CPU @ 2.50GHz, 2501 Mhz,	16 GB RAM	Windows 10

Tabla 10. Infraestructura.

9.3 Planes de Trabajo del Proyecto

9.3.1 Descomposición de Actividades

A través de una estructura WBS se representó la descomposición de las principales actividades a desarrollar en el proyecto, indicando de manera jerárquica aquellas actividades sencillas que componen actividades más complejas. En el WBS se analiza las actividades de cada fase en el proyecto y sus respectivos entregables a desarrollar.

Para ver en detalle el WBS dirigirse al siguiente link <https://www.gloomaps.com/rFzovRdXKb>

9.3.2 Calendarización

Para la calendarización del proyecto, se emplea un diagrama Gantt que permite tener un cronograma de las fases del proyecto. En este diagrama se establece los tiempos de las actividades que se identificaron en la sección [9.3.1 Descomposición de Actividades](#) y que fueron plasmadas en el WBS.

Para ver la calendarización a través del diagrama Gantt, dirigirse a la sección de anexos.

Monitoreo y Control del Proyecto

10.1 Administración de Requerimientos

Cuando en el desarrollo del proyecto surge un cambio en los requisitos del sistema, es necesario ejecutar un proceso para llevar un control a la hora de ejecutar estos cambios. Este proceso puede surgir en cualquier momento en la etapa, por lo cual se emplea las reuniones de cada Sprint para discutir los posibles cambios que surjan

Al realizar una reunión grupal, para revisar lo desarrollado al finalizar un Sprint, puede surgir la situación que uno de los integrantes o el *Product Owner* haya indicado previamente a los demás integrantes que es necesario realizar un cambio en los requisitos del sistema, por lo cual se debe discutir en la reunión si se va a realizar el cambio en los requisitos, o es mejor no implementarlo. Si el cambio está relacionado a un requisito en específico, y este ya fue desarrollado en una iteración anterior, se debe analizar de manera grupal que tan primordial es el requisito y cuál sería su impacto al realizar nuevamente la implementación de este, debido a que puede afectar los avances ya realizados en el proyecto, además de retrasar el cronograma de las diferentes etapas del proyecto.

En el caso de que el cambio del requisito no afecte la organización del proyecto debido a su bajo impacto o al ser un nuevo requisito que no había sido identificado antes, se realiza la especificación del requisito para que sea desarrollado en una iteración futura. Este cambio en el requisito implica la modificación algunos elementos que son desarrollados en el SRS, como es la especificación del requisito y su relación con los casos de uso, por lo cual es necesario actualizar el SRS para garantizar una consistencia entre el documento y el desarrollo del software; también es necesario actualizar la matriz de trazabilidad, indicando en cual prototipo se va a desarrollar el requisito cambiado.

A continuación, en la **Ilustración 2** se muestra el diagrama BPMN que explica el proceso llevado a cabo en el momento en que se notifica al equipo el cambio de un requisito. En todo el proceso se analiza de manera grupal el requisito que va a ser cambiado, con el propósito de no afectar la organización del proyecto en el caso de realizar la modificación.

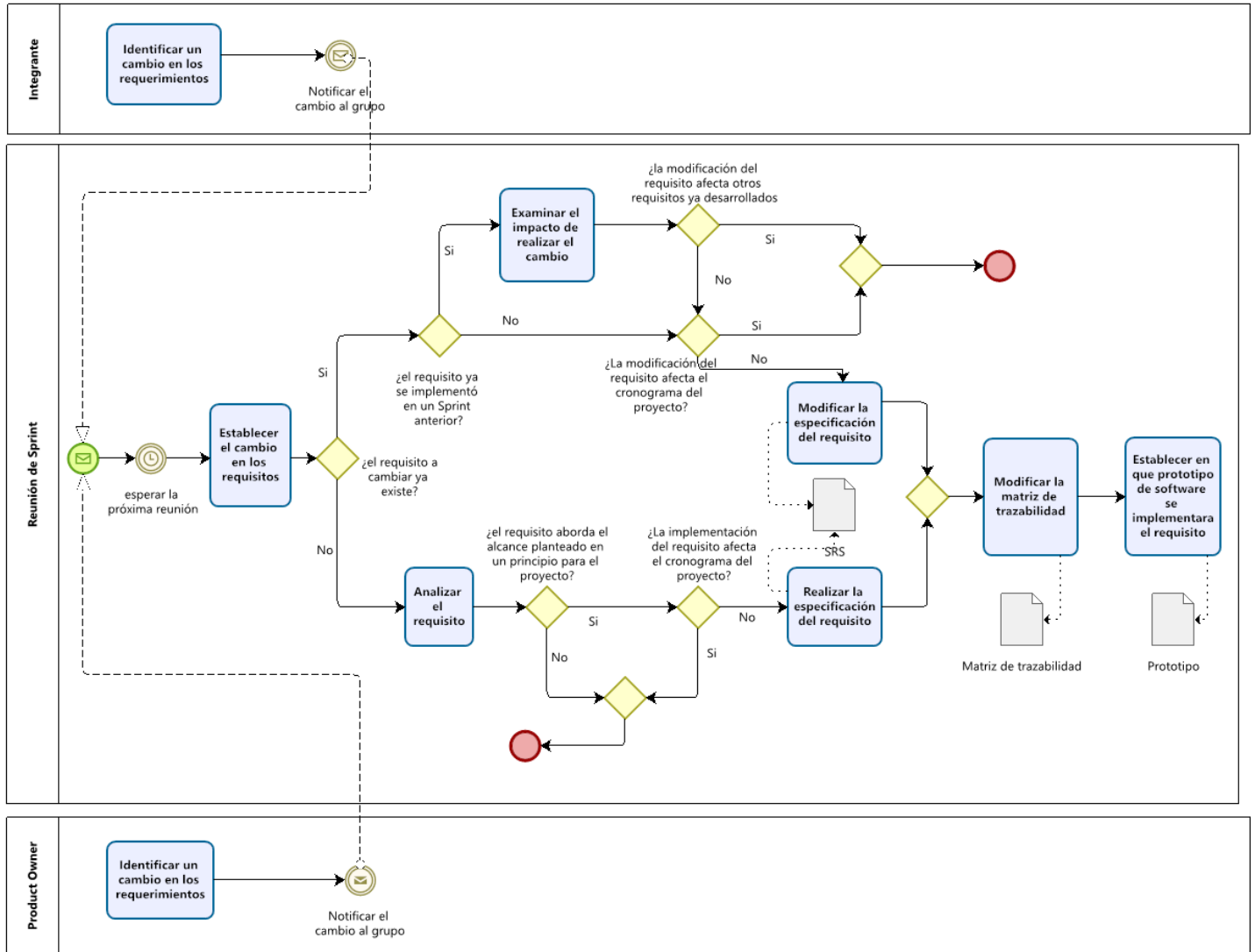


Ilustración 2. Administración de requerimientos.

10.2 Monitoreo y Control de Progreso

Debido a que las actividades tanto del diseño como del desarrollo son distribuidas entre todos los integrantes del proyecto, es necesario, llevar un control del progreso de las actividades que están actualmente en desarrollo.

En este proyecto se utilizan herramientas para llevar un seguimiento de las actividades mencionadas en la sección 8.2.6 Manejo del proyecto. Una de estas herramientas es un tablero virtual que permite medir el desarrollo de una actividad a través de 3 clasificadores: Pendientes, En Curso y Completadas. El objetivo de esta clasificación es llevar un control de las actividades que actualmente están desarrollando los integrantes y cuales aún faltan por realizar. Cada integrante modifica en el tablero virtual el estado de las actividades que tiene asignadas y podrá observar la carga de trabajo que posee los demás integrantes del equipo, la cual puede ser medida de acuerdo con número de tareas que tiene asignados y la estimación realizada para cada

tarea mediante planning poker, mencionado en la sección 9.1 Métodos y herramientas de estimación

Además, se implementa una herramienta para establecer los tiempos de duración para las diferentes actividades y poder realizar un control del tiempo que ha dedicado el integrante del equipo en una actividad y cuánto tiempo aún queda para la fecha de entrega de la actividad. Emplear esta herramienta permite al *Scrum Master* y a los demás integrantes llevar un registro del progreso llevado a cabo en cada una de las iteraciones del proyecto, tomando medidas preventivas cuando se observe que se presenta un estancamiento en el desarrollo de las actividades.

Al adoptar en el proyecto los *Daily Scrum*, se llevará a diario un seguimiento de los avances obtenidos por los integrantes del equipo, sin embargo, no se empleará una herramienta para llevar el registro de los reportes realizados en los *Daily Scrum*. Este seguimiento diario permite informar de manera muy general el avance que ha realizado cada integrante sin emplear alguna métrica o reporte que defina el progreso de una actividad, por lo cual, si es necesario realizar una medición, será necesario apoyarse en las herramientas mencionadas anteriormente.

En el caso que se presente problemas en el cumplimiento de las actividades, será necesario realizar acciones que permitan mitigar el impacto en la organización del proyecto. Si una actividad no es finalizada en la fecha establecida para su entrega, se discute el motivo para que la actividad no fuera finalizada. En el caso de que se reportara que la actividad posee una carga de trabajo muy intensa para los integrantes que la estaban realizando, se reasignará la actividad a aquellos integrantes que tengan la disponibilidad y conocimiento para continuar con la actividad y poder darle fin.

10.3 Cierre del Proyecto

Para llevar a cabo el cierre del proyecto se utilizará la metodología de reporte gerencial el cual permite un mejoramiento continuo en base a las experiencias que deja la realización de este proyecto de *software*. El objetivo del cierre del proyecto es evaluar el desempeño del grupo de trabajo y los problemas que se presentaron durante el desarrollo del producto final, para de esta manera mejorar en próximos proyectos.

Criterios de entrada:

El desarrollo del producto final ha concluido y se han hecho las respectivas pruebas. El grupo de trabajo posee información en cuanto al desempeño durante el trabajo realizado.

Revisión de los datos del proceso:

- Identificar las posibles fallas durante el desarrollo del proyecto
- Comparar la planeación previa al proyecto *versus* el verdadero desempeño una vez terminado el producto
- Evaluar los datos que utilizó cada miembro durante la realización del trabajo.

Al concluir esta revisión obtendremos los problemas que surgieron a lo largo del desarrollo del proyecto, esto con el fin de prepararnos para afrontar estos problemas en un futuro.

Revisión de los datos de calidad:

- Evaluar la calidad del producto.
- Comparar con reportes gerenciales anteriores.

Evaluación de roles:

Evaluar el trabajo realizado por cada miembro de acuerdo con su rol establecido.

En esta sección se evaluará el desempeño de cada miembro del grupo que tuvo dentro del proyecto. Para esto se estará revisando el trabajo que cada miembro realizó, los recursos utilizados, los problemas que aparecieron durante el desarrollo del proyecto, los aspectos que podrían mejorar a futuro y de qué manera se implementarían. La evaluación de roles está a cargo del líder del proyecto, en este caso el *Scrum Master*.

Entrega del Producto

En la sección **6.4 Entregables** se exponen cuáles son los entregables y fechas límites de cada entregable.

Los procesos que se llevarán para la entrega formal del producto final serán mediante una reunión con las fundaciones participantes, explicándoles el funcionamiento de la aplicación y los costos asociados a la misma (*Hosting*, APIs, bases de datos), los cuales deberán asumir una vez esté terminado el proyecto de trabajo de grado.

Por otro lado, otra de las entregas formales que se realizarán será en la sustentación del trabajo de grado, donde se expondrá a fondo la aplicación desde el punto de vista de la Ingeniería de Sistemas, explicando a profundidad cada una de las etapas, desde el levantamiento de requisitos hasta la etapa de pruebas.

Procesos de Soporte

12.1 Ambiente de Trabajo

Dirigirse al documento reglas de trabajo en equipo

12.2 Análisis y Administración de Riesgos

12.2.1 Descripción del plan de riesgos

Los riesgos son una parte inherente en cualquier proyecto, es por esto que se necesita de una metodología la cual nos concientice de ellos, monitorearlos, minimizarlos, analizarlos y mitigarlos para que, de esta manera, no influyan en la culminación exitosa de nuestro proyecto. Para ello se aplicará un plan de riesgos, el cual buscará la manera de que los efectos de estos riesgos sean nulos o mínimos. Este plan posee 4 fases:

Fase 1. Identificación de riesgos

Esta fase se encarga de determinar los riesgos que puedan afectar el desarrollo del proyecto, a continuación, se presentan 7 riesgos que se lograron identificar:

1. Falta de personal con experiencia.
2. Abandono del proyecto por parte de uno de los integrantes.
3. El proyecto no se logró culminar en la fecha específica
4. Personal no disponible en momentos críticos
5. Cambio en los requerimientos que necesitan rehacer el diseño.
6. Pérdida de apoyo con las fundaciones.
7. El personal clave se encuentra enfermo y no disponible en los momentos críticos.

Fase 2. Análisis de riesgos

No solamente es importante la identificación de los riesgos sino también su priorización. La priorización de cada riesgo viene calculada por la relación entre el impacto que tendría en caso de producirse y la probabilidad de que ocurra. En la **tabla 11** se calcula la priorización de los riesgos detectados de acuerdo a la probabilidad y el impacto que puede generar.

N° Riesgo	Probabilidad	Impacto	Priorización
1	Medio bajo (4)	Medio alto (7)	Alto (28)
2	Bajo (1)	Alto (10)	Bajo (10)
3	Medio bajo (4)	Alto (10)	Alto (40)
4	Medio Alto (7)	Medio bajo (4)	Alto (28)
5	Medio bajo (4)	Medio alto (7)	Alto (28)
6	Bajo (1)	Alto (10)	Bajo (10)
7	Medio bajo (4)	Alto (10)	Alto (40)

Tabla 11. Análisis de riesgos.

Ya realizada la priorización, los resultados de este análisis se plasmaron en la matriz de riesgos, como se muestra a continuación desde la **Ilustración 3** hasta la **Ilustración 9**.

1. Falta de personal con experiencia ☒

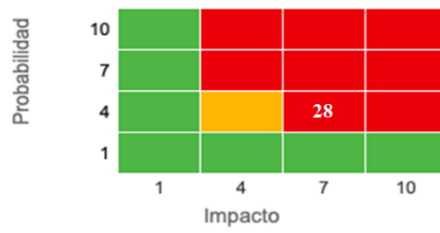


Ilustración 3. Riesgo 1.

2. Falta de personal clave (Abando... ☒

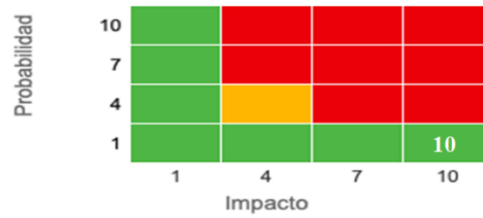


Ilustración 4. Riesgo 2.

3. El proyecto no sea culminado en ... ☺

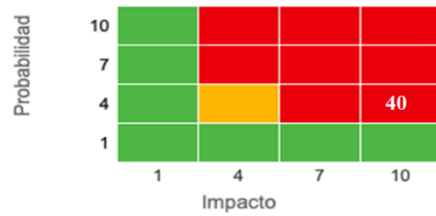


Ilustración 5. Riesgo 3.

4. Personal no disponible en mome... ☒

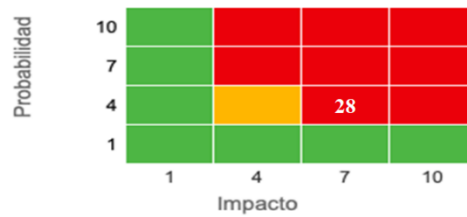


Ilustración 6. Riesgo 4.

5. Cambio en los requerimientos qu...

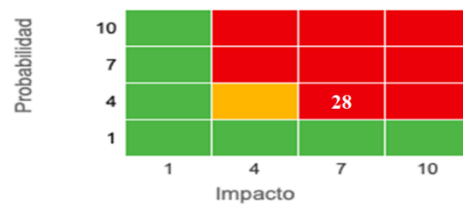


Ilustración 7. Riesgo 5.

6. Pérdida de apoyo con las fundaciones

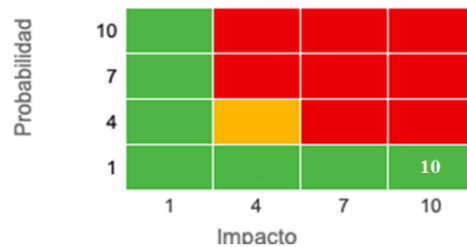


Ilustración 8. Riesgo 6.

El personal clave está enfermo y no...

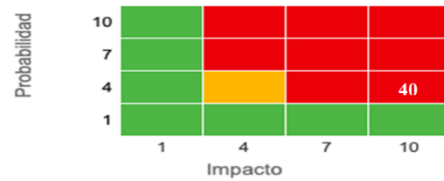


Ilustración 9. Riesgo 7.

Fase 3. Planificar la respuesta

A continuación, en la **tabla 12** se muestran los planes de contingencia que fueron identificados:

Riesgo	Plan de contingencia
Falta de personal con experiencia	En el caso de que un integrante no cuente con el conocimiento necesario en algún tema en específico, se realizará el trabajo de la mano con otro integrante más experimentado. Además, se realizarán reuniones con el fin de aclarar dudas acerca de temas puntuales.
Abandono del proyecto por parte de uno de los integrantes.	Redistribuir las tareas asignadas a los integrantes restantes.
El proyecto no se logró culminar en la fecha específica	Renegociar los términos de entrega estipulados en el contrato, dependiendo de las estimaciones de las versiones
Personal no disponible en momentos críticos	Priorizar actividades con mayor importancia, con el fin de distribuirlas entre los integrantes del grupo
Cambio en los requerimientos que	Añadirlos nuevos requerimientos o los cambios necesarios de forma clara y completa para que se cumpla la funcionalidad.

necesitan rehacer el diseño	
Pérdida de apoyo con las fundaciones	Tener en cuenta distintas alternativas de fundaciones para que, en caso de que alguna se llegue a desinteresar en el desarrollo del proyecto se pueda seguir el proceso con otra.
El personal clave se encuentra enfermo y no disponible en los momentos críticos.	Redistribución de tareas. Esto con el fin de suplir la actividad pendiente del compañero que no pudo estar en ese momento.

Tabla 12. Plan de contingencia.

Fase 4. Supervisar el riesgo

Durante esta fase se realiza un seguimiento del proyecto con el fin de evitar la aparición de riesgos y, por consiguiente, el proyecto se revisará de manera constante y cuidadosa.

Herramientas:

- Software ITM Plattform.

12.3 Administración de Configuración y Documentación

En la **tabla 13** se pueden observar los distintos ítems de configuración y las distintas versiones que tendrán a lo largo del proyecto.

Entregable	Momentos
SRS	24/05/2021
SDD	30/06/2021
Prototipos del programa	15/08/2021 24/10/2021
Documentos de prueba	Por definir
Manuales de usuario	Momento antes del fin del proyecto.

Tabla 1313. Administración de configuración y documentación.

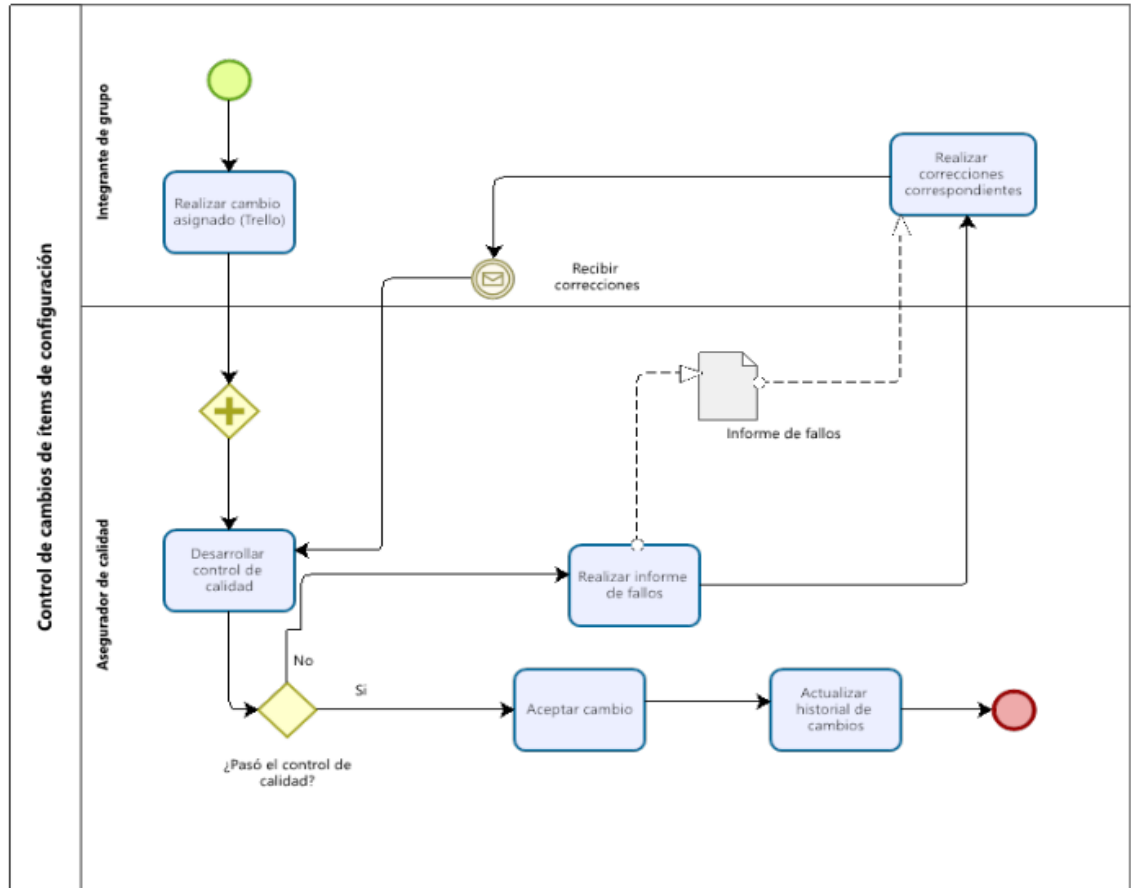


Ilustración 10. Control de cambios de ítems de configuración.

En la **Ilustración 10** el proceso que explica el BPMN correspondiente es que, se realiza un cambio asignado al ítem correspondiente, se hace el control de calidad y si se acepta el cambio, se añade al proyecto. En caso de que el cambio no sea aceptado se realiza un informe de fallos donde el responsable del cambio deberá corregirlos y posteriormente enviar el documento otra vez a control de calidad.

Métricas y Proceso de Medición

13.1 Control de Calidad

En esta etapa, los documentos y código que sean modificados serán sometidos a su correspondiente proceso de control de calidad para cada uno, donde se verificará que cumplan con los resultados esperados. A continuación, en la **tabla 14**, se describirán los roles en el proceso del control de calidad de los documentos.

Tabla 14. Roles para el proceso de control de calidad.

Responsables	Proceso	Frecuencia	Descripción
Director de calidad	Revisión del contenido	Cada vez que se modifique el contenido del documento	El director de calidad verificará que aquello que se le haya modificado al documento no afecte de manera negativa su calidad
Director de calidad y arquitecto	Revisión de diagramas BPMN	Cada vez que cambien los procesos	El analista de la calidad y el arquitecto, verificarán que los procesos que cambien o sean recién transcritos a un diagrama BPMN, sean fieles a lo que se pretende hacer en dicho proceso.
Arquitecto y desarrolladores (front - end y back - end)	Revisión de mockups e interfaz gráfica	Cada que cambien o se generen vistas	El arquitecto y desarrolladores revisarán que los mockups propuestos reflejen una idea fiel de lo que se pretende implementar gráficamente,

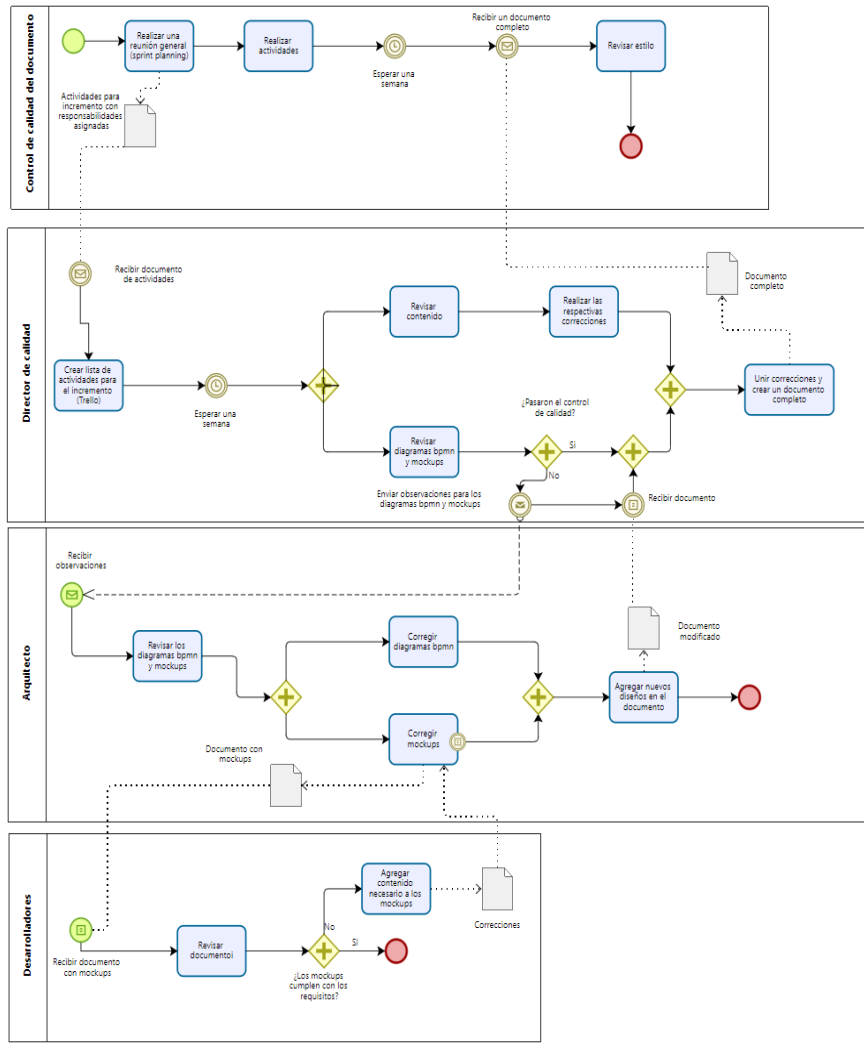


Ilustración 11. Control de calidad de documentos.

En la **ilustración 11** el proceso que explica el BPMN es el de control de calidad de documentos, donde la revisión empieza en el sprint planning, las actividades que se deben realizar durante el próximo sprint llegarán al director de calidad quien estará encargado de distribuir las entre los miembros del grupo. Los documentos se revisarán sección por sección de igual forma con los mockups y diagramas BPMN. En caso de que alguno de estos artefactos presente problemas, deben corregirse y volver a ser enviados al director de calidad para verificar la corrección.

En la **tabla 15**, se describirán los roles en el proceso del control de calidad del código.

Tabla 15. Roles para control de calidad del software.

Responsables	Proceso	Frecuencia	Descripción
Desarrolladores	Revisión del contenido	Cada vez que se modifique el contenido del documento	El director de calidad verificará que aquello que se le haya modificado al documento no afecte de manera negativa su calidad
Scrum Master	Revisión de diagramas BPMN	Cada vez que cambien los procesos	El analista de la calidad y el arquitecto, verificarán que los procesos que cambien o sean recién transcritos a un diagrama BPMN, sean fieles a lo que se pretende hacer en dicho proceso.
Director QA	Revisión de mockups e interfaz gráfica	Cada que cambien o se generen vistas	El arquitecto y desarrolladores revisarán que los mockups propuestos refleje una idea fiel de lo que se pretende implementar gráficamente,

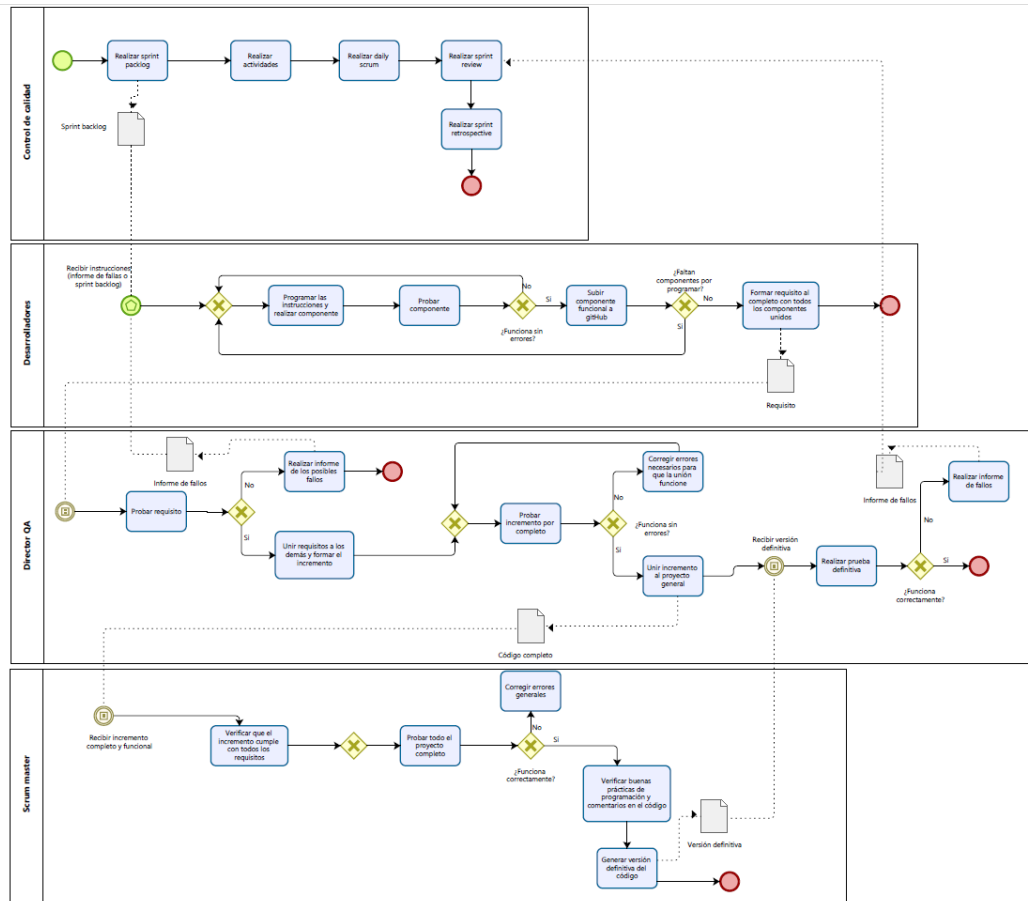


Ilustración 12. Control de calidad en el sprint planning.

En la **ilustración 12** el proceso que explica el BPMN es el de control de calidad de documentos en el *sprint planning*, donde se decidirá que se hará durante el siguiente sprint. Una vez inicia el sprint los desarrolladores tendrán tareas asignadas, luego que el desarrollador realiza su actividad correspondiente, procederá a probar su código, verificando que este funcione adecuadamente para posteriormente subirlo a la herramienta de control de versiones. Una vez se haya concluido con el desarrollo de código, el desarrollador enviará el código al Director QA, quien a su vez verificará que el código funcione adecuadamente. En caso de no pasar satisfactoriamente las pruebas, el Director QA retroalimentará al desarrollador mediante un informe de fallos. Una vez superada la prueba, el código se unificará con el código principal del proyecto y se volverá a probar. Luego de esto, el Scrum Master será el último quien verifique el correcto funcionamiento y posteriormente subirá el incremento a github además de acoplarlo con el resto del proyecto.

Anexos

- A. **Organigrama del equipo** ([ENLACE](#))
- B. **WBS** ([ENLACE](#))
- C. **Gantt** ([ENLACE](#))

Referencias

- [1] MongoDB. 2021. *Pricing*. <https://www.mongodb.com/pricing> (accedido 14 May 2021).
- [2] Google Cloud. 2021. *Precios y planes | Google Maps Platform | Google Cloud*. <https://cloud.google.com/maps-platform/pricing?hl=es-419> (accedido 14 May 2021).
- [3] Hostinger. 2021. *Hosting Web Para Colombia - Rápido, Fácil Y Con Soporte 24/7*. <https://www.hostinger.co/hosting-web> (accedido 14 May 2021).
- [4] «¿Que es un Prototipo de Software? - Software para Todo». <https://softwareparatodo.com/que-es-un-prototipo-de-software/> (accedido may 14, 2021).
- [5] I. Sommerville, *Global Edition Software Engineering Tenth Edition*. 2016
- [6] R. S. Pressman, *Ingeniería del Software. Un Enfoque Práctico*. 2010.
- [7] «Front-End vs Back-End vs Full Stack Web Developers», *Udacity*, dic. 08, 2020. <https://www.udacity.com/blog/2020/12/front-end-vs-back-end-vs-full-stack-web-developers.html> (accedido mayo 13, 2021).
- [8] “¿Qué es un framework y para qué se utiliza? | Orix Systems.” <https://www.orix.es/que-es-un-framework-y-para-que-se-utiliza> (accedido mayo 13, 2021).
- [9] “Biblioteca (informática) - Wikipedia, la enciclopedia libre.” [https://es.wikipedia.org/w/index.php?title=Biblioteca_\(informática\)&oldid=126104731](https://es.wikipedia.org/w/index.php?title=Biblioteca_(informática)&oldid=126104731) (accedido mayo 13, 2021).
- [10] “BPMN Guide Quick Start”, *Bizagi.com*, 2021. https://www.bizagi.com/files/live/sites/bizagi/files/PDFs/BPMN_Quick_Start_Guide.pdf. (accedido mayo 13, 2021).
- [11] “Documentation for Visual Studio Code.” <https://code.visualstudio.com/docs> (accedido mayo 13, 2021).
- [12] “Best of Atom: Features, Plugins, Acting Like Sublime Text — Scotch.io.” <https://scotch.io/bar-talk/best-of-atom-features-plugins-acting-like-sublime-text> (accedido mayo 13, 2021).
- [13] “HTML: Lenguaje de etiquetas de hipertexto | MDN.” <https://developer.mozilla.org/es/docs/Web/HTML> (accedido mayo 13, 2021).
- [14] “Bootstrap 4 Get Started.” https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp (accedido mayo 13, 2021).
- [15] “The most advanced responsive front-end framework in the world. | Foundation.” <https://get.foundation/> (accedido mayo 13, 2021).

- [16] "facebook/react: A declarative, efficient, and flexible JavaScript library for building user interfaces." <https://github.com/facebook/react> (accedido mayo 13, 2021).
- [17] "Introduction — Vue.js." <https://vuejs.org/v2/guide/> (accedido mayo 13, 2021).
- [18] "Angular - Introduction to Angular concepts." <https://angular.io/guide/architecture> (accedido mayo 13, 2021).
- [19] "All you need to know about Firebase." <https://www.the-reference.com/en/blog/pieter-de-busschere/firebase> (accedido mayo 13, 2021).
- [20] «What Is MongoDB?», *MongoDB*. <https://www.mongodb.com/what-is-mongodb> (accedido mayo 13, 2021).
- [21] "Bases de datos de Azure: servicios en la nube totalmente administrados | Microsoft Azure." <https://azure.microsoft.com/es-es/product-categories/databases/> (accedido mayo 13, 2021).
- [22] "What is Git? - Azure DevOps | Microsoft Docs." <https://docs.microsoft.com/en-us/azure/devops/learn/git/what-is-git> (accedido mayo 13, 2021).
- [23] "What is SVN? | Subversion Overview | Perforce." <https://www.perforce.com/blog/vcs/what-svn> (accedido mayo 13, 2021).
- [24] "What is GitHub." https://www.w3schools.com/whatis/whatis_github.asp (accedido mayo 13, 2021).
- [25] "What is GitLab? | GitLab." <https://about.gitlab.com/what-is-gitlab/> (accedido mayo 13, 2021).
- [26] "What is StarUML - StarUML Documentation." https://documentation.help/StarUML/what_is_staruml.htm (accedido mayo 13, 2021).
- [27] "Visual Paradigm Frequently Asked Questions." <https://www.visual-paradigm.com/support/faq.jsp> (accedido mayo 13, 2021).
- [28] "Bizagi Oficial Plataforma de negocios digitales y BPMS." <https://www.bizagi.com/es/plataforma/modeler> (accedido mayo 13, 2021).
- [29] "What Is G Suite? How To Use Google Suite Products | Bluehost Support." <https://www.bluehost.com/help/article/gsuite> (accedido mayo 13, 2021).
- [30] "What is Microsoft Office 365? Webopedia Definition." https://www.webopedia.com/TERM/O/office_365.html (accedido mayo 13, 2021).
- [31] "¿Qué es Trello? - Ayuda de Trello." <https://help.trello.com/article/708-what-is-trello> (accedido mayo 13, 2021).
- [32] "What is monday.com? - Support." <https://support.monday.com/hc/en-us/articles/115005310945-What-is-monday-com-> (accedido mayo 13, 2021).

[33] R. Pressman, *Ingeniería del software*, 7th ed. Mexico, 2010.