

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS



SOFTWARE PROJECT MANAGEMENT PLAN

TECHCAMEL

ALEX DAVID BARRETO ALFONSO

ANDRÉS ESTEBAN DÍAZ DEL CASTILLO LONDOÑO

DIEGO MAURICIO BULLA NUÑEZ

GABRIEL FERNANDO FORERO ORTIZ

LUISA YURANI CONTRERAS VERGEL

SEBASTIAN ROBERTS SERRATO

FECHA: 09 septiembre 2020

# Historial de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Sección N°** | **Secciones del Documento** | **Comentarios** | **Responsable(s)** |
| 1 | Historial de cambios | Teniendo en cuenta la distribución del documento, se completó dicha tabla | Luisa Contreras |
| 2 | Resumen | Completamos el contenido en grupo | Equipo TechCamel |
| 3 | Tabla de contenidos | Completamos el contenido en grupo conforme íbamos avanzando | Equipo TechCamel |
| 4 | Lista de figuras | Completamos el contenido en grupo conforme íbamos avanzando | Equipo TechCamel |
| 5 | Lista de tablas | Completamos el contenido en grupo conforme íbamos avanzando | Equipo TechCamel |
| 6.1 | Visión del producto | Se determinó la visión en grupo | Equipo TechCamel |
| 6.2 | Propósito, alcance y objetivos | Determinamos los ítems en grupo | Equipo TechCamel |
| 6.3 | Supuestos y restricciones | Determinamos los ítems en grupo | Equipo TechCamel |
| 6.4 | Entregables | Determinamos los ítems en grupo | Equipo TechCamel |
| 6.5 | Evolución del plan | Determinamos los ítems en grupo | Equipo TechCamel |
| 6.6 | Glosario | Determinamos los ítems en grupo | Equipo TechCamel |
| 7.1 | Modelo de ciclo de vida | Se determinaron los modelos de procesos | Andrés Díaz del Castillo |
| 7.2 | Lenguaje y Herramientas | Se determinaron el lenguaje en que trabajaremos y las herramientas a usar | Andrés Díaz del Castillo |
| 7.3 | Plan de aceptación del producto | Se determinaron los criterios a cumplir en cada entrega | Gabriel Forero y Diego Bulla |
| 7.4 | Organización del proyecto y comunicación | Se determinaron los roles y las interfaces | Gabriel Forero y Diego Bulla |
| 8.1 | Métodos y herramientas de estimación | Se determinaron que métricas de estimación se usarán a lo largo del proyecto | Gabriel Forero y Diego Bulla |
| 8.2 | Inicio del proyecto | Se determina que tareas debe hacer cada integrante del equipo | Gabriel Forero y Diego Bulla |
| 8.3 | Planes de trabajo del proyecto | Se determina que debe hacer cada integrante del equipo y que herramientas utilizar en cada fase | Gabriel Forero y Diego Bulla |
| 9.1 | Administración de requisitos | Completamos los requisitos por medio de una tormenta de ideas | Alex Barreto, Luisa Contreras y Sebastían Roberts |
| 9.2 | Monitoreo y control de progreso | Se determinó el monitoreo y control del documento | Alex Barreto y Sebastían Roberts |
| 9.3 | Cierre del proyecto | Se determinó que hacer antes de cerrar el proyecto | Alex Barreto y Sebastían Roberts |
| 10 | Entrega del producto | En complemento de la 7.3 se completo el Itém | Luisa Contreras |
| 11.1 | Ambiente de trabajo | Se establecieron las normas de trabajo, contenidas en un manual de reglas | Luisa Contreras |
| 11.2 | Análisis y administración de riesgos | Se determinaron los riesgos y sus análisis por medio de lluvia de ideas | Alex Barreto, Luisa Contreras y Sebastían Roberts |
| 11.3 | Administración de configuración y documentación | Se determinaron las herramientas de configuración y los procesos que se necesitan para su documentación | Luisa Contreras |
| 11.4 | Control de calidad | Se determinó el proceso de control de calidad | Alex Barreto y Sebastían Roberts |
| 12 | Anexos | Completó los documentos correspondientes | Sebastían Roberts |
| 13 | Referencias | Cada uno completó las referencias según sus citas usadas | Equipo TechCamel |

# Resumen

El propósito del documento es explicar el plan de gestión del proyecto de software (SPMP) para el proyecto DO4ME. El objetivo del proyecto es desarrollar una herramienta que permita la comunicación entre el consumidor final y el servicio de niñera o limpieza, permitiendo brindar seguridad, trasparencia y confianza tanto a nuestros clientes finales como también a las personas que trabajan en nuestra aplicación.

El SPMP se explicará a través del documento mediante las estructuras establecidas en la tabla de contenido. Entre los aspectos de la estructura se encuentran la visión general del proyecto, en la que se especifican las características esenciales del proyecto; el contexto donde se describen los elementos y herramientas necesarios para su elaboración; gestión de proyectos, donde se describen los seguimientos y controles a lo largo de la realización del proyecto; soporte de la aplicación donde se encuentran todos los procesos referentes al soporte, anexos, entre otros que dan sostenibilidad a nuestra aplicación.

# Tabla de contenidos

[1 Historial de cambios 2](#_Toc50576547)

[2 Resumen 3](#_Toc50576548)

[3 Tabla de contenidos 4](#_Toc50576549)

[4 Lista de figuras 6](#_Toc50576550)

[5 Lista de tablas 7](#_Toc50576551)

[6 Vista general del proyecto 8](#_Toc50576552)

[6.1 Visión del producto 8](#_Toc50576553)

[6.2 Propósito, alcance y objetivos 8](#_Toc50576554)

[6.3 Supuestos y restricciones 9](#_Toc50576555)

[6.4 Entregables 10](#_Toc50576556)

[6.5 Evolución del plan 11](#_Toc50576557)

[7 Contexto del proyecto 12](#_Toc50576558)

[7.1 Modelo de ciclo de vida 12](#_Toc50576559)

[7.2 Lenguajes y herramientas 16](#_Toc50576560)

[7.3 Plan de aceptación del producto 18](#_Toc50576561)

[7.4 Organización del proyecto y comunicación 19](#_Toc50576562)

[8 Administración del proyecto 21](#_Toc50576563)

[8.1 Métodos y herramientas de Estimación 21](#_Toc50576564)

[8.2 Inicio del proyecto 23](#_Toc50576565)

[8.3 Planes de trabajo del proyecto 24](#_Toc50576566)

[9 Monitoreo y control del proyecto 27](#_Toc50576567)

[9.1 Administración de requisitos 27](#_Toc50576568)

[9.2 Monitoreo y control del progreso 29](#_Toc50576569)

[9.3 Cierre del proyecto 31](#_Toc50576570)

[10 Entrega del producto 33](#_Toc50576571)

[11 Procesos de soporte 34](#_Toc50576572)

[11.1 Ambiente de trabajo 34](#_Toc50576573)

[11.2 Análisis y administración de riesgos 35](#_Toc50576574)

[11.3 Administración de configuración y documentación 38](#_Toc50576575)

[11.4 Control de calidad 41](#_Toc50576576)

[12 Referencias 44](#_Toc50576577)

[13 Referencias 44](#_Toc50576578)

# Lista de figuras

[Figura 1. SCRUM PROCESS 12](#_Toc50485117)

[Figura 2. SCRUM Metodología 14](#_Toc50485118)

[Figura 3. Organigrama del proyecto 18](file:///C:\Users\USER\Downloads\SPMP%20DO4ME.docx#_Toc50485119)

[Figura 4.BPMN para representar el proceso de detectar y controlar los cambios que sufren los requerimientos 25](#_Toc50485120)

[Figura 5. ejemplo grafica para métrica de velocidad 27](#_Toc50485121)

[Figura 6.BURNDOWN 28](#_Toc50485122)

[Figura 7. Diagrama BPMN del proceso de identificación y análisis de riesgo. 31](#_Toc50485123)

[Figura 8. Proceso de Cambio (Mejora) 35](file:///C:\Users\USER\Downloads\SPMP%20DO4ME.docx#_Toc50485124)

[Figura 9.Proceso Documentación 36](#_Toc50485125)

[Figura 10.BPMN para representar la gestión de control de calidad 38](#_Toc50485126)

# Lista de tablas

[Tabla 1.Entregables 10](#_Toc50486227)

[Tabla 2.Entregables y criterios de aceptación 18](#_Toc50486228)

[Tabla 3.StakeHolders 18](#_Toc50486229)

[Tabla 4.Descripcion de roles 19](#_Toc50486230)

[Tabla 5.Plan de capacitación 20](#_Toc50486231)

[Tabla 6.Actividades 21](#_Toc50486232)

[Tabla 7.Diagrama de Gantt 22](#_Toc50486233)

[Tabla 8.Adquisicion de licencias 23](file:///C:\Users\USER\Downloads\SPMP%20DO4ME.docx#_Toc50486234)

[Tabla 9.Salario de los colaboradores 23](file:///C:\Users\USER\Downloads\SPMP%20DO4ME.docx#_Toc50486235)

[Tabla 10.Hardware 23](file:///C:\Users\USER\Downloads\SPMP%20DO4ME.docx#_Toc50486236)

[Tabla 11.Plan de capacitación 23](file:///C:\Users\USER\Downloads\SPMP%20DO4ME.docx#_Toc50486237)

[Tabla 12.Gastos mensuales 24](file:///C:\Users\USER\Downloads\SPMP%20DO4ME.docx#_Toc50486238)

[Tabla 13.Presupuestos 24](file:///C:\Users\USER\Downloads\SPMP%20DO4ME.docx#_Toc50486239)

[Tabla 14.Sprint 1 27](#_Toc50486240)

[Tabla 15.Artefactos de entrega 29](#_Toc50486241)

[Tabla 16.Identificacion de riesgos 32](#_Toc50486242)

[Tabla 17.Analisis y gestión de riesgos 33](#_Toc50486243)

[Tabla 18.Elementos de documentación y código 37](#_Toc50486244)

[Tabla 19.Control de calidad 39](#_Toc50486245)

# Vista general del proyecto

## Visión del producto

Para los padres, madres y dueños de hogar que necesitan ser asistidos en sus labores domésticas, DO4ME es un servicio conveniente, seguro y fácil de usar que les permitirá suplir sus necesidades de cuidado de sus hijos y de mantenimiento de sus hogares.

Al contrario de los métodos tradicionales para conseguir a una persona de confianza que cumpla con las labores de asistencia doméstica, DO4ME facilita los canales de comunicación entre los afiliados (trabajadores) y los clientes implementando fuertemente medidas de para garantizar el cumplimiento de las responsabilidades de cada parte y además facilitando el proceso de procesamiento de pagos.

El factor diferenciador de DO4ME es enfocar la oferta laboral en mujeres cabeza de hogar, encontrando a las posibles afiliadas en bolsas de empleo de fundaciones. Además, para garantizar que las trabajadoras estén dando un buen servicio y los solicitantes cumplan con sus responsabilidades con DO4ME y la trabajadora, se utilizará un sistema de calificaciones mediante el cual, al finalizar un servicio, ambas partes podrán darse una puntuación que permitirá a futuro determinar si la persona puede seguir trabajando con DO4ME y/o si el solicitante puede seguir utilizando el servicio.

Uno de los principales problemas con los que nos encontramos en el mercado de servicio doméstico es la desconfianza que pueden tener los clientes frente a personas desconocidas que ingresen a su hogar para hacer aseo, o más aún para cuidar a sus hijos. Para esto, el sistema de calificación será esencial y además se hará una profunda investigación de las trabajadoras de modo que se garantice la seguridad de los clientes.

## Propósito, alcance y objetivos

**Propósito**

TechCamel es consiente que nos encontramos en una ciudad que lucha por mantener la igualdad de oportunidades y las buenas condiciones laborales para algunos de sus habitantes. Por esta razón, introducimos DO4ME, un intento desde el punto de vista de la ingeniería de sistemas para mejorar las condiciones laborales de aquellas mujeres que no tienen tantas oportunidades en el mundo laboral de hoy. Estamos al tanto también que nuestros clientes no tienen disponibilidad completa de horarios, por lo que ellos mismos escogen su disponibilidad y capacidad de trabajo

**Alcance**

El alcance de este proyecto es hacer que DO4ME sea una herramienta popular y confiada por los usuarios para la adquisición de servicios domésticos en el marco de los objetivos mencionados en el numeral anterior, como lo son medidas de calidad para el cliente y la trabajadora. Además, DO4ME busca tener un gran impacto social, pues se permite exclusivamente que mujeres cabeza de hogar puedan tener la posibilidad de trabajar en la plataforma, en tiempos flexibles y bajo su propio horario.

Suposiciones y restricciones

En esta sección se describirán las suposiciones hechas a lo largo del proyecto y de su diseño que permitirán realizarlo correctamente. Se asume que las siguientes suposiciones son verdaderas y necesarias para lograr llevar a cabo el proyecto, y que las restricciones serán componentes que podrían interferir con un buen desarrollo y/o ejecución del mismo.

**Objetivos**

Desarrollar el sitio web DO4ME, una herramienta en la cual se ofrecen dos servicios de asistencia doméstica: aseo de hogar y niñera

Especificación de los objetivos:

* Reducir la fricción y riesgos relacionados a las transacciones monetarias para el pago y reembolso de servicios
* Garantizar medidas de calidad para las afiliadas y los clientes
* Ofrecer una plataforma en la que mujeres cabeza de hogar pueden afiliarse para ofrecer servicios de asistencia doméstica
* Facilitar el proceso de adquisición de servicios domésticos provistos por DO4ME acorde a el tiempo y requisitos de los clientes.
* Excluir a trabajadoras y clientes del servicio que tengan malas calificaciones

## Supuestos y restricciones

En esta parte, describiremos las suposiciones hechas a lo largo de este documento y el diseño del proyecto, tomando las suposiciones como cosas que se cree que son ciertas para ejecutar el proyecto con éxito. Y las restricciones, tomadas como los factores que pueden interferir con el desarrollo del proyecto o el lanzamiento del producto.

Supuestos:

La siguiente es una lista de supuestos que se mantendrán como verdaderos durante el ciclo de vida del proyecto.

* En DO4ME confiamos en la buena voluntad de los involucrados en el proyecto, por lo que se supone que los clientes actúan bajo el marco de la ley.
* Un cliente puede ver la calificación del empleado antes de solicitar el servicio
* Los miembros de TechCamel tendrán una disponibilidad que permita el desarrollo del horario dentro de los tiempos establecidos.
* Todos los integrantes cuentan con el equipo con los requisitos recomendados para el correcto funcionamiento del software de desarrollo.

Restricciones:

La siguiente es una lista de restricciones para el proyecto.

* Asumimos que toda la información proporcionada por los usuarios es de confianza, esto incluye certificado de antecedentes, documentación y hoja de vida.
* El proyecto está limitado por un plazo de 4 meses, donde la última etapa del ciclo de vida es la entrega del proyecto sin abarcar la etapa de mantenimiento y control
* El producto final debe cumplir con las leyes sobre el oficio de aseo y cuidado de niños.
* Solo se pueden utilizar herramientas cedidas por la universidad o gratuitas, porque no hay un presupuesto real, ya que es un proyecto universitario con fines académicos.
* El grupo de trabajo estará compuesto por seis (6) personas, este número es fijo y se espera que no cambie a lo largo del proyecto.
* Se requiere conexión a Internet para que el sitio web funcione.
* Todos los valores económicos son estimaciones ya que el proyecto se limita a un ejercicio académico.

## Entregables

|  |  |  |
| --- | --- | --- |
| Nombre | Descripción | Fecha |
| Sprint 1 | Definir el alcance del proyecto, desarrollo del plan de proyecto y desarrollo del plan de gestión del proyecto | 7/09/2020 |
| SPMP | Documento del plan de gestión del proyecto | 9/09/2020 |
| Sprint 2 | Diseño y planeación del caso de uso mas importante | 9/10/2020 |
| Sprint 3 | Desarrollo del caso de uso más importante | 16/10/2020 |
| SRS y el caso de uso más importante | Caso de uso más importante, correcciones del SPMP y descripción del software | 30/10/2020 |
| Sprint 4 | Realizar pruebas y desarrollo de la interfaz del proyecto | 27/11/2020 |
| Entrega Final | Aplicación terminada con la documentación requerida y correcciones del SSR | 4/12/2020 |

Tabla 1.Entregables

## Evolución del plan

Durante el proceso de elaboración del proyecto, es necesario que cualquier cambio en el plan pase por un proceso de verificación y control, esto esta explicado con detalle en la sección 11.3 “*Administración de configuración y documentación”.* Sin embargo, es necesario que los cambios sean informados a los responsables de las secciones para que puedan actualizar las tablas y diagramas correspondientes, adicionalmente estos cambios se deben registrar en el historial de versiones.

Como se está utilizando SCRUM como metodología ágil descrita en la sección 6.2. Todos los cambios requeridos se tienen que hacer al final de cada Sprint. Solo en caso de que el cambio tenga un gran impacto en el proyecto se hará durante el Sprint.

# Contexto del proyecto

## Modelo de ciclo de vida

En esta sección se definirán las metodologías con procesos claramente descritos que serán utilizados para llevar a cabo el desarrollo de software de este proyecto y se describirán las razones por las cuales se eligen ciertas metodologías sobre otras. Este proyecto se llevará a cabo mediante la aplicación de metodologías agiles, pues con estas se está anticipando a la posibilidad de que aparezcan cambios a lo largo del desarrollo del proyecto y permite mucha mayor flexibilidad que los métodos tradicionales.

**SCRUM**

SCRUM es indiscutiblemente la metodología ágil más popular. El Framework de scrum está diseñado para enfocarse en dos puntos críticos del desarrollo de software: velocidad y cambio en los requerimientos del cliente. Con esta metodología, el desarrollo de software para el proyecto es llevado a cabo por etapas, y cada etapa se conoce como un sprint. El framework de scrum es heurístico – basado en cambios continuos, aprendizaje continuo y ajustes a distintos factores el proyecto. Reconoce que el equipo no tiene conocimiento de todo al comienzo del proyecto y evolucionará con la experiencia. A pesar de que scrum puede parecer sumamente estructurado, no es estrictamente rígido, pues su ejecución puede adaptarse a las necesidades de cualquier organización. [1]

SCRUM está compuesto por tres artefactos principales, los cuales son herramientas para resolver un problema y serán descritos a continuación:

Product backlog: Es la lista maestra de trabajo que debe ser llevada a cabo y mantenida por el dueño del producto o el administrador del producto. Es en esencia una lista de “cosas por hacer” del equipo de trabajo.

Sprint backlog: Es la lista de elementos, historias de usuario o solución de errores seleccionados por el equipo de desarrollo para la implementación en el ciclo actual del sprint. Antes de cada sprint, el equipo elije en cuales elementos se trabajará del producto backlog.

Increment: El increment u objetivo del sprint es el producto final utilizable de un sprint. También se refiere a un hito, meta u objetivo, pero todo dependerá de la definición que el equipo de trabajo de le a hito.

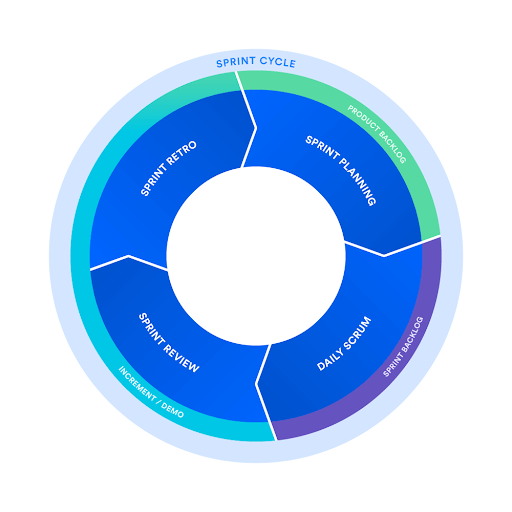


Figura 1. SCRUM PROCESS

En la metodología SCRUM también existen 3 roles esenciales para que se pueda llevar acabo exitosamente el proyecto: Dueño del producto, scrum máster y equipo de desarrollo. Debido a que los equipos de scrum son interdisciplinarios, el equipo de desarrollo incluye a ingenieros de devops, testers, diseñadores, ingenieros de UX y otros roles adicionales a los desarrolladores. [1]

**Extreme programming (XP)**

En esta metodología se busca producir software de mayor calidad y mayor calidad de vida para el equipo de desarrollo de software. XP es la metodología ágil más específica con respecto a prácticas de ingeniería para desarrollo de software. Las principales características de XP descritas por Don Wells son:

* Requerimientos de software cambian dinámicamente
* Riesgos causados por proyectos con marcos de tiempo cortos usando nuevas tecnologías
* Equipos de desarrollo pequeños y distribuidos
* La tecnología que se esté utilizando permite realizar pruebas unitarias automatizadas y pruebas funcionales

**Kanban**

La metodología Kanban tiene como objetivo reducir la cantidad de trabajo en progreso. Se busca manejar el trabajo balanceando la demanda con la capacidad disponible, y mejorando el manejo de estancamientos o cuellos de botella a nivel del sistema. Los elementos de trabajo son visualizados para darle a los participantes del proyecto una vista del progreso y de los procesos de comienzo a fin.

**Feature driven development (FDD)**

FDD es un framework ágil que organiza el desarrollo de software alrededor de hacer progreso en las funcionalidades. En el contexto de FDD, las funcionalidades no son necesariamente funcionalidades particulares del producto, pues también pueden ser procesos completos sin necesidad de llegar a la granularidad de sus componentes [1]

**Metodología ágil para usar: SCRUM**

La razón para utilizar metodologías agiles fue para ayudar a incrementar el desempeño del equipo de trabajo, mejorar la satisfacción del cliente e incrementar la versatilidad del proyecto. Tras una ardua investigación de las distintas metodologías agiles, la que se utilizará para el desarrollo de este proyecto será SCRUM. [1]

A continuación, se encuentran las principales razones por las que se eligió SCRUM como la mejor metodología ágil para este proyecto:

1. Promueve el trabajo en equipo: Gracias a la flexibilidad con relación a la división y asignación de trabajo y roles, SCRUM nos ayuda a trabajar de forma eficiente en el proyecto.
2. Es liviano porque tiene pocos elementos predefinidos: 3 roles, 3 tipos de reuniones y 3 artefactos.
3. Es ágil porque maximiza la solución y respuesta que se le da a las necesidades y requerimientos cambiantes de los clientes.
4. Es un framework de procesos por que no es un proceso, pero una colección de prácticas y procesos a partir de los cuales puede ser construido un proceso.

A pesar de que las otras metodologías también tienen sus beneficios, necesitan de equipos altamente experimentados en desarrollo como lo es en el caso de FDD para lograr los hitos relacionados a las funcionalidades, o se enfocan en equipos más grandes como Kanban y XP.

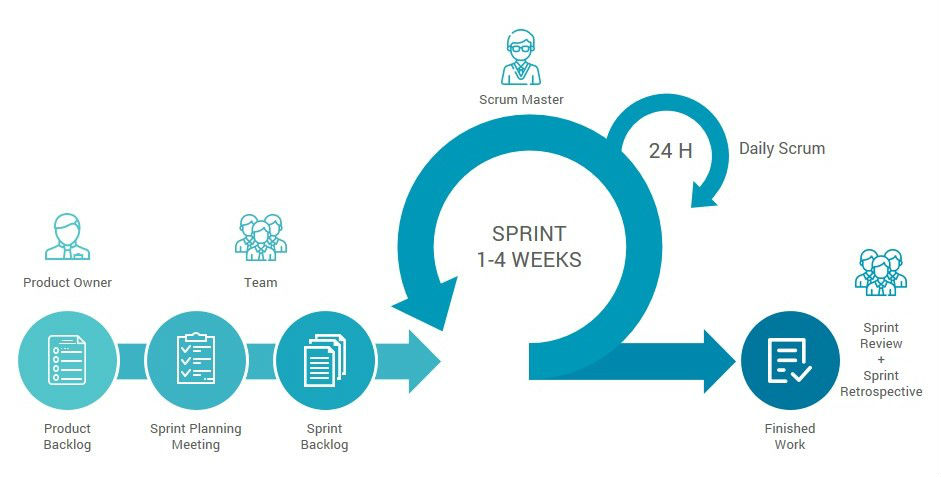


Figura 2. SCRUM Metodología

En la ilustración 2 se encuentran los distintos procesos que componen el proceso de desarrollo de software utilizando la metodología SCRUM. En total existen 19 procesos indicados por el SBOK (Scrum body of knowledge) [1] los cuales están agrupados en las 5 fases a continuación:

**Iniciación**: Esta fase incluye los procesos relacionados a la iniciación del proyecto – crear la visión del proyecto, identificar al scrum máster y los stakleholders, crear un equipo de scrum desarrollar los hitos, crear un backlog priorizado del producto y llevar a cabo la planeación de lanzamiento del producto.

Duración estimada: 14 días

**Plan y estimación**: Esta fase consiste en los procesos relacionados a la planeación y estimación de tareas, los cuales incluyen creación de historias de usuario, aprovechar estimar y usar historias de usuario, crear tareas, estimar tareas y crear el sprint backlog.

Duración estimada: 14 días

**Implementación**: Esta fase está relacionada a la ejecución de las tareas y actividades para crear el producto del proyecto. Estas actividades incluyen crear varios entregables, conducir reuniones diarias que se llevarán a cabo en el grupo de MS Teams de DO4ME, revisar, mejorar y actualizar el backlog del producto. Cabe resaltar que todos los miembros de CamelTech participarán en el desarrollo e implementación de DO4ME.

Duración estimada: 45 días

**Revisión y retrospectiva:** Esta fase está enfocada en revisar los entregables y el trabajo que ha sido finalizado y determinar formas para mejorar las prácticas y métodos utilizados para que el proyecto funcione.

Duración estimada: 7 días

**Lanzar**: Esta etapa se enfoca en entregar el producto aceptado al cliente e identificar, documentar y finalmente internalizar lo aprendido del proyecto

Duración estimada: 7 días

## Lenguajes y herramientas

Las herramientas y lenguajes de programación mencionadas a continuación se decidieron basados en las capacidades y conocimientos de los integrantes del grupo. Además de las capacidades de los integrantes, se tuvieron en cuenta 7 factores para la selección de las herramientas a utilizar:

* **Tipo de aplicación:** Aplicaciones web, móviles, embebidas etc. Necesitarán distintas funcionalidades y tendrás una enorme influencia en el lenguaje a utilizar. En el caso de DO4ME se está desarrollando una aplicación web motivo por el cual se seleccionaron herramientas y lenguajes de programación compatibles con este tipo de aplicaciones.
* **Complejidad de la aplicación:** El tamaño y la complejidad del proyecto tiene un rol importante en la decisión de cual stack de tecnología se utilizará, lo cual conlleva al lenguaje de programación que será usado.
* **Cultura del equipo de trabajo:** Cada grupo de trabajo tiene preferencias con respecto a usar software libre, privado, uso de propiedad intelectual externa y herramientas específicas para ciertas funcionalidades. Se decidió utilizar herramientas y software libre para desarrollar DO4ME.
* **Tiempo para lanzar un producto funcional:** Se utilizarán herramientas bien establecidas en el mercado, pues introducir nuevas tecnologías o tecnologías experimentales puede tener un impacto negativo en el cumplimiento de los plazos establecidos para el lanzamiento del proyecto debido a que se debe tener presente el tiempo necesario para aprender a usar la herramienta.
* **Mantenibilidad:** Cada stack de tecnología viene con un ecosistema de librerías y un proveedor que les da mantenimiento. Queremos enfocarnos en las tendencias para garantizar que los lenguajes y herramientas seleccionados son actuales y se mantendrán en el mercado por un buen tiempo.
* **Escalabilidad y desempeño:** La escalabilidad está más relacionada a las herramientas que a los mismos lenguajes de programación. En las siguientes secciones se discutirán los lenguajes y herramientas escalables a utilizar.
* **Seguridad:** Cada aplicación tiene requerimientos distintos en términos de seguridad. Para el desarrollo de DO4ME se tiene en cuenta lenguajes de programación pertenecientes al mundo de software post-monolitico, es decir que puedan funcionar es un ecosistema en el cual las decisiones de tecnología no sean mutuamente excluyentes y causen problemas críticos de seguridad.
* Herramientas Y Frameworks:
* **Node JS**: Es el framework de javascript para el backend de DO4ME. Está diseñado para construir aplicaciones web escalables. En el caso de este proyecto de software, será utilizado para manejar todos los servicios y solicitudes de la plataforma. Se ha elegido este framework ya que el equipo de trabajo está familiarizado con JavaScript.
* **Sass**: Es un procesador de CSS que nos permitirá utilizar elementos como variables, reglas anidadas, importar y reutilizar componentes de diseño y mucho más. Nos permitirá crear una plataforma con mejor apariencia y además nos permitirá mantener un orden y crear estilos con mayor velocidad.
* **Wompi API:** El api de wompi será necesario para que los clientes puedan pagar un servicio de aseo de forma conveniente, segura y rápida. La razón por la que elegimos wompi es que la pasarela de pago permite que una pyme la utilice sin necesidad de generar muchos ingresos por mes, y por que permite pagos directamente en pesos colombianos sin la necesidad de hacer una conversión a dólares.
* **React JS:** React es el framework de javascript usado para desarrollar el frontend de DO4ME. Permite crear UIs profesionales, dinámicos y responsivos de forma fácil.
* **Git:** Serán las herramientas usadas para hacer control de cambios y mantener un registro de versiones del proyecto. Se eligió esta herramienta debido a que el equipo está familiarizado con su uso.
* **Github:** El proyecto se encontrará en un repositorio de GitHub.
* **PostgreSQL:** Sistema para administración de bases de datos que se basa en SQL.

**Lenguajes de programación:**

* **HTML:** HTML es utilizado para crear la página web de DO4ME y sus componentes de frontend estáticos.
* **CSS:** Puesto a que se busca ofrecer un servicio vendible, es indispensable que la plataforma sea atractiva para los clientes. CSS permitirá crear un sitio web interactivo, atractivo y llamativo.
* **Javascript:** Es el lenguaje utilizado tanto para el desarrollo del frontend como el backend del software. Se usará para la lógica de negocio del proyecto. Fue seleccionado como lenguaje de programación a utilizar debido a su simplicidad, seguridad, escalabilidad y por cumplir con todos los factores establecidos para seleccionar un lenguaje de programación.
* **SQL:** Es necesario para las bases de datos y es en lo que está basado PostgreSQL

## Plan de aceptación del producto

En este apartado se establecerán criterios, técnicas, herramientas y entregables que cumplan con los requerimientos del cliente, para que el producto sea aceptado. Esto lo puede ver en la tabla 2 y los entregables están descritos en la sección 6.4.

|  |  |  |
| --- | --- | --- |
| **Entregable** | **Criterio** | **Técnicas y herramientas** |
| SPMP, SRS, SDD | * Notación bien utilizada * Coherencia en los documentos * Buena escritura * Buena redacción * Buena ortografía * Referencias en IEEE | * Herramienta Zotero o JabRef para definir referencias. * El documento debe ser aprobado por el director de documentación |
| Primer prototipo | * Implementar el caso de uso más importante * Control de documentación * Código sin errores de compilación * Código documentado * Control de versiones * Código sin errores en tiempo de ejecución | * El director de configuración y desarrollo están encargados de verificar que todos los criterios están completados * Cumplir todos los estándares de calidad * Git hub para el control de versiones |
| Segundo prototipo | * Implementar el 70% de casos de uso * Control de documentación * Código sin errores de compilación * Código documentado * Control de versiones * Código sin errores en tiempo de ejecución | * El director de configuración y desarrollo están encargados de verificar que todos los criterios están completados * Cumplir todos los estándares de calidad * Git hub para el control de versiones |

Tabla 2.Entregables y criterios de aceptación

## Organización del proyecto y comunicación

En esta sección se establecerán los Stakeholdes (tabla 3), el organigrama del proyecto (ilustración 3) y la descripción de cada rol junto a sus responsabilidades del proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Descripción | Responsabilidades | Datos de contacto |
| PSE | Empresa que se encarga de las transacciones a través de la aplicación | Ofrecer un pago seguro a los clientes finales | <https://www.pse.com.co/inicio> |
| 4HOME | Es una empresa que se encarga de garantizar la profesionalidad de nuestras empleadas | Contratar empleadas calificadas para un buen servicio | <https://www.4home.com.co/> |

Tabla 3.StakeHolders

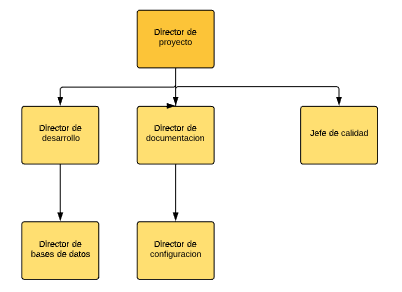


Figura 3. Organigrama del proyecto

|  |  |
| --- | --- |
| Rol | Descripción |
| Director de proyecto | Lider de un equipo con la función de conseguir alcanzar los objetivos. Es la persona que tiene la responsabilidad total del planeamiento y ejecución de cualquier proyecto. |
| Director de desarrollo | Liderar al equipo en el desarrollo y la elaboración del software y orientar el aprendizaje de las herramientas utilizadas |
| Director de documentación | Rol que permite controlar y mantener ordenada la documentación que se va generando a lo largo del proyecto. |
| Jefe de calidad | Dirigir todas las actividades relacionadas con asegurar la calidad de los procesos dentro del proyecto , analizar y diseñar planes de mejora que resuelvan los errores detectados. |
| Director de configuración | Este rol gestiona la infraestructura global de la configuración y el entorno del equipo de desarrollo del producto. |
| Director de bases de datos | Administrador de base de datos Mantener la integridad en los datos |

Tabla 4.Descripcion de roles

# Administración del proyecto

En la sección de administración del proyecto, se tiene como objetivo que los integrantes del proyecto entiendan los aspectos detallados del plan de proyecto y su respectiva gestión. Para esto se definen los métodos y herramientas de estimación, las condiciones iniciales del proyecto y el plan de trabajo de proyecto.

## Métodos y herramientas de Estimación

Para el desarrollo del proyecto, se usarán dos herramientas de estimación: “Estimación basada en casos de uso” y “Sprints”. Estas se aplicarán al “*Burn down chart*”, dado que se está utilizando como metodología ágil Scrum. [2]

* Estimación basada en casos de uso: Se presentará gráfico detallado de los casos de uso desarrollados por el equipo del proyecto. Esto será útil al momento de proporcionar una referencia de cómo es la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas.
* Sprints: Se hará un gráfico que permitirá hacer una estimación de la carga de trabajo que tomará todos los sprints del proyecto. Este cuadro se utilizará como una herramienta que permitirá conocer el trabajo de todo el proyecto (en términos de código y documentación) que aún falta por entregar.

|  |  |  |
| --- | --- | --- |
| Nombre del caso de uso | Cantidad de transacciones | Categoría |
| Asignar trabajador a cliente | 7 | Promedio |
| Mostrar disponibilidad de trabajadores | 3 | Simple |
| Realizar pago | 7 | Promedio |
| Solicitar servicio | 7 | Promedio |
| Iniciar sesión en la plataforma | 3 | Simple |
| Programar disponibilidad de trabajadores | 7 | Promedio |
| Calificar servicio | 3 | Simple |
| Crear PQR | 3 | Simple |
| Registrar usuario | 3 | Simple |

Tabla 5. Clasificación de casos de uso

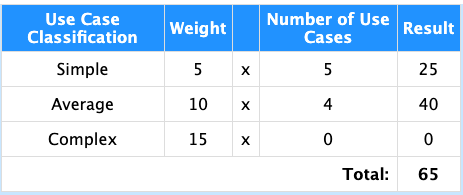


Tabla 6. Unadjusted Use Case Weight  
  
**Unadjusted Use Case Weight**

Determinar el impacto que tendrá cada caso de uso en un sistema es fundamental para el tamaño del proyecto. El número de transacciones que maneja un caso de uso deberá encontrarse para cada caso de uso. Hay tres clasificaciones en las que un caso de uso puede ser: simple, promedio o complejo. [3]

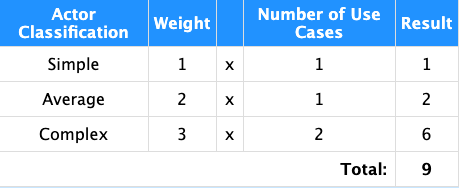


Tabla 7. Unadjusted Actor Weight

**Unadjusted Actor Weight**

Los actores contribuyen al tamaño del proyecto, por lo tanto, todos los actores deben estar bien pensados ​​para todo el sistema. Hay tres clasificaciones que puede tener un actor: simple, promedio y complejo. [3]

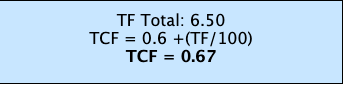


Figura 4. Technical Complexity Factor  
  
**Technical Complexity Factor**

toma en consideración los requisitos técnicos que se implementarán [3]

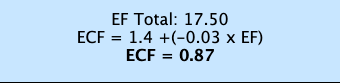


Figura 5. Environment Complexity Factor

**Environment Complexity Factor**

Esto estima factores que no se implementarán en un proyecto, pero lo impactarán severamente. ECF se ocupa del equipo del proyecto y su capacidad para completar el proyecto. [3]

El UCP se calcula con base a la siguiente fórmula:

UCP = (UUCW + UAW) x TCF x ECF

UCP = (65,00 + 9,00) x 0,67 x 0,87

UCP = 43,06

El proyecto contiene 43.06 puntos de casos de uso.

## Inicio del proyecto

Para el desarrollo del proyecto es indispensable que los integrantes del equipo tengan conocimientos en las aplicaciones y lenguajes de programación descritos en la sección 7.2. Por esta razón, la tabla 5 muestra el plan de capacitación para poner en marcha el proyecto.

|  |  |
| --- | --- |
| **Habilidad /Conocimiento** | **Proceso de enseñanza** |
| HTML | introducción al desarrollo web |
| CSS |
| JavaScript | Curso Udemy |
| SQL | Desarrollo y gestión de BD |

Tabla 8.Plan de capacitación

Herramientas para poner en marcha el proyecto:

* Inicialización del repositorio: Se creará un repositorio en Git Hub para el control de versiones, el cual será masterizado por todo el equipo
* Net beans: Es un entorno de desarrollo integrado libre
* Oracle data base: Es un sistema de gestión de bases de datos de tipo objeto-relacional desarrollado por Oracle corporación

## Planes de trabajo del proyecto

El plan de trabajo de proyecto consiste en 3 actividades clave que permitirán visualizar y llevar a cabo el proyecto de una manera adecuada. La primera es la descomposición por actividades, la cual se puede observar en la ilustración YYYY, donde las tareas están descompuestas mediante la estructura WBS



Tabla 9.Actividades

La segunda es el diagrama de Gantt donde se muestra las fechas de inicio y finalización de cada tarea en cada Sprint del proyecto. Y se pueden ver las dependencias en la tabla de actividades.



Tabla 10.Diagrama de Gantt

Finalmente se encuentra el presupuesto del proyecto, para esto se tuvieron en cuenta diferentes aspectos en el proyecto como lo son:

* Adquisición de licencias (El pago es mensual): Se necesitan 3 licencias para el desarrollo del proyecto

Tabla 11.Adquisicion de licencias

* Salario de los colaboradores (Pago mensual): Dado que el contrato es por prestación de servicios los colaboradores asumirán los gastos como: seguridad social, auxilio de transporte, pensión etc.

Tabla 12.Salario de los colaboradores

* Compra de computadores (solo se hace una vez)

Tabla 13.Hardware

* Plan de capacitación: Plan de capacitación que esta descrito en la sección anterior

Tabla 14.Plan de capacitación

* Gastos mensuales: Un arriendo en Bogotá y los servicios como agua, luz, teléfono, internet y gas

Tabla 15.Gastos mensuales

El presupuesto del proyecto se calculó con una duración estimada de 4 meses

Tabla 16.Presupuestos

# Monitoreo y control del proyecto

## Administración de requisitos

Es innegable que la flexibilidad y el dinamismo en lo que se refiere a requerimientos en proyectos de software es uno de los desafíos más comunes, los clientes cambian de opinión, nuevas tecnologías surgen, o el contexto cambia de un momento a otro.

Partiendo de que estamos desarrollando el proyecto basándonos en los lineamientos propuestos por Scrum, existe una comunicación constante entre el cliente y el equipo, abriendo la posibilidad a que se modifiquen los requerimientos o se replanteen las prioridades.

Los cambios o actualizaciones en los requerimientos deben ser notificados el mánager/líder del proyecto, y luego de que el producto owner le haya manifestado los cambios que desea, el manager debe notificar a las partes involucradas para que los cambios sean realizados en la siguiente iteración o sprint. [1]

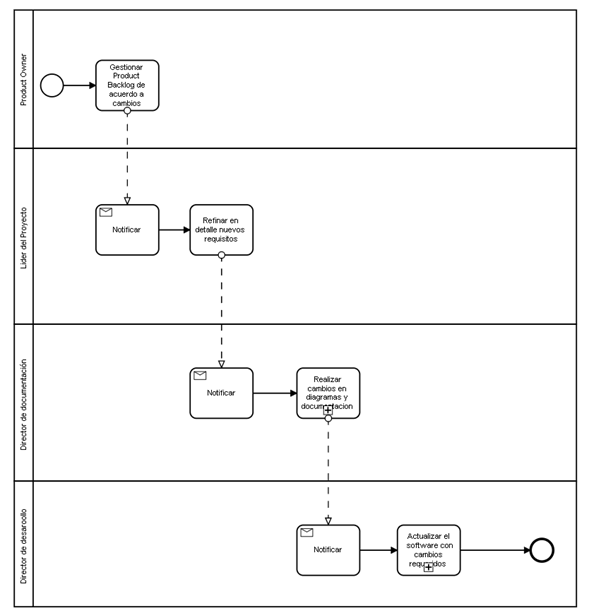


Figura 6.BPMN para representar el proceso de detectar y controlar los cambios que sufren los requerimientos

## Monitoreo y control del progreso

Durante la consulta al SBOK, encontramos que se definen tres métricas iniciales para hacer seguimiento a los sprint, la velocidad, el valor comercial entregado y número de tareas. [1]

La velocidad se refiere al número de requerimientos y/o funcionalidades realizadas o entregadas al cliente por unidad de tiempo (día o semana). Esto se mide por cada sprint, donde idealmente se espera que aumente con respecto al tiempo, así mismo, esta medida permite tener un estimado de la fecha final del proyecto. El propietario del producto puede usar la velocidad para predecir la rapidez con la que un equipo puede trabajar a través de la acumulación, porque el informe rastrea el trabajo pronosticado y completado en cada sprint; cuantos más sprint medidos, más preciso es el pronóstico. Adicionalmente, también se mide el tiempo que toma al equipo desarrollar únicamente una tarea para así conocer la velocidad por cada tarea y no únicamente la velocidad por cada sprint.

El valor comercial entregado busca medir el valor de los requerimientos y funcionalidades completados desde una perspectiva comercial, igualmente se espera hacer esta medición por cada sprint. Como propietario de un producto, también desea tener una estimación del valor comercial de cada requerimiento, adicionalmente, las estimaciones de valor comercial ayudan a crear una acumulación más racional y maximizar el valor que ofrece el equipo.

Por último, el número de tareas implica la cantidad de requisitos o funcionalidades entregadas al cliente por cada sprint, relacionándose también con la tasa de finalización la cual se puede definir como la cantidad de tareas completadas en un sprint versus la cantidad de tareas que se esperaba completar, es decir, si el equipo pone diez tareas en un sprint y completa siete de ellas, tiene una tasa de finalización del 70%. Esta es una métrica que desea que sea bastante alta ya que las tareas son pequeñas y pueden completarse en un sprint.

La manera en que se implementan estas métricas es a partir de gráficas que relacionan las medidas mencionadas en cada definición, por lo tanto, en la velocidad se realiza una gráfica donde se ubica el valor de la velocidad respecto a la unidad de tiempo seleccionada, acompañado de una tabla donde relaciona no solo estas dos medidas sino también el número de requerimientos entregados. Como se muestra a continuación. [1]

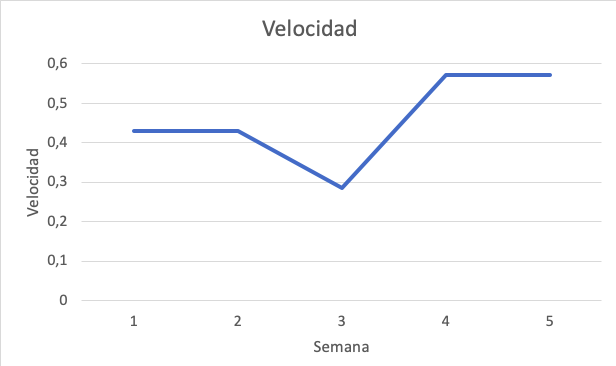


Figura 7. ejemplo grafica para métrica de velocidad

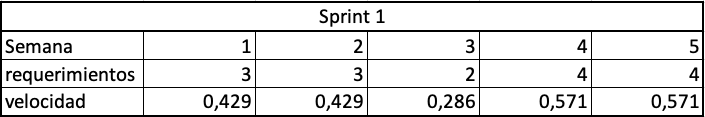


Tabla 17.Sprint 1

Ahora bien, respecto al valor entregado y al número de tareas se representan por medio de un cuadro burndown. En la siguiente imagen se muestra un burndown chart, donde se relacionan las tareas completadas en cada sprint, las tareas pendientes del sprint, las tareas agregadas al sprint y una estimación de las tareas de los futuros sprints. Este cuadro se realiza cada semana a fin de evaluar el rendimiento del equipo frente al progreso del proyecto.

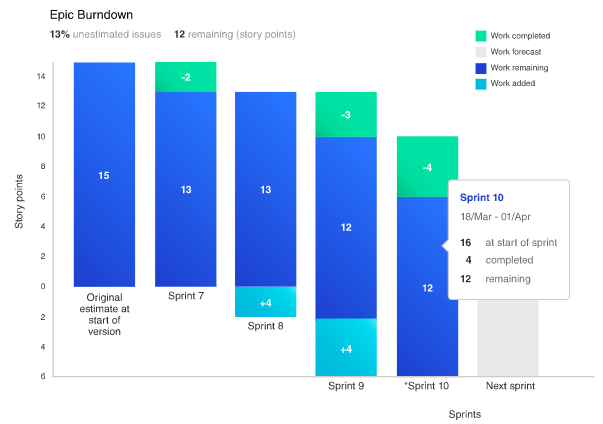


Figura 8.BURNDOWN

## Cierre del proyecto

Al realizar cada una de las entregas correspondientes a los Sprint, y luego de la retroalimentación brindada por el cliente, se realizará una reunión grupal para que las acciones recomendadas por el mismo sean discutidas y realizadas, es decir una reunión de revisión del sprint. Además de ello, parte del protocolo, implica una reunión donde se revisen por última vez la calidad de los entregables para cada iteración. Este control de calidad será descrito en apartados posteriores. [1]

Los aspectos pertenecientes al protocolo incluyen, pero no se limitan a:

-Reporte gerencial que detalle los resultados obtenidos durante la iteración, en este apartado podría incluirse el Burndown Chart actualizado.

-Documentación resultante de los resultados del método para realizar el control de calidad

-Protocolo post mortem de la iteración, para determinar los elementos que fueron exitosos y los que no lo fueron. Su realización en forma de lecciones aprendidas puede servir como herramienta de mitigación de riesgos en futuras etapas.

# Entrega del producto

Para la realización de las entregas junto con él proyecto final, realizaremos actividades como la redacción de documentos relevantes con contribución de cada uno de los integrantes de TechCamel, además todos los integrantes realizarán una capacitación previa en el manejo de creación del software del sitio web de DO4ME, como lo son JavaScript: Nade JS y React, además PostgreSQL o MySQL y por último HTML y CSS dirigida por el director de desarrollo.

Las entregas para realizar junto con sus fechas y sus responsables están plasmadas en la siguiente tabla.

|  |  |  |
| --- | --- | --- |
| Artefacto(s) a entregar | Fecha de Entrega | Responsable(s) |
| SPMP | 9 de septiembre del 2020 | Equipo TechCamel |
| SRS | 16 de octubre del 2020 | Equipo TechCamel |
| SDD | 25 de noviembre-4 de diciembre | Equipo TechCamel |
| Primer prototipo | 16 de octubre del 2020 | Equipo TechCamel |
| Segundo prototipo | 25 de noviembre-4 de diciembre | Equipo TechCamel |

Tabla 18.Artefactos de entrega

# Procesos de soporte

## Ambiente de trabajo

REGLAS DE TECHCAMEL

Objetivo:

Prevenir y regular las posibles situaciones y actividades que se presenten durante el ciclo académico 2020-3, en el grupo TechCamel, con el fin de fomentar el buen trabajo en equipo mediado por el respeto y el aprendizaje colectivo, promoviendo un ambiente laboral agradable y el poder llevar a cabo cada una de las partes del proyecto de la mejor manera, con un protocolo claro.

Reglas:

* Dirigirse a sus compañeros del equipo con respeto, claridad y asertividad.
* Asistir puntualmente a todas las reuniones acordadas por todos sus integrantes.
* Asistir puntualmente a la clase Ingeniería de Software.
* Cumplir con las tareas, compromisos y responsabilidades debidas con puntualidad y excelencia.
* Comunicar con sus compañeros las posibles dificultades que le puedan presentar conflicto con el desarrollo de sus actividades que contribuyen al equipo.
* WhatsApp es el principal canal de comunicación, por lo tanto, deberá estar atento a este y responder cada que sea pertinente.
* Evitar realizar comentarios que afecten el equipo.
* Fomentar el trabajo de manera amable y organizadamente.
* Ser comprensivos y solidarios en caso de dificultades en un integrante del grupo.
* Atender con empatía los comentarios de sus compañeros.

Sanciones:

* Dialogo con el fin de un cambio de actitud del integrante correspondiente
* Llamado de atención: Los integrantes del equipo harán una invitación a su compañero para que mejore su rendimiento o asistencia (ya sea a clases o a las reuniones correspondientes).
* Reunión con el director del curso Ingeniería de Software, con el fin de ver las causas o motivo del comportamiento, y buscar establecer medidas de prevención o correctivas.
* En caso extremo, en el que uno de los integrantes, muestre total desinterés por la contribución con sus compromisos y responsabilidades, se le informará nuevamente al maestro su exclusión para su reubicación o trabajo de manera individual.

## Análisis y administración de riesgos

Para la identificación y el tratamiento de los riesgos hemos utilizado una matriz donde es posible no solamente identificarlos, sino analizarlos y categorizarlos dependiendo de su probabilidad de ocurrencia y de su impacto en los objetivos del proyecto. El proceso de identificación y análisis de los riesgos se realiza al inicio del proyecto, al inicio y finalización de cada sprint.

Este proceso inicia con la identificación de los potenciales riesgos que puedan darse a lo largo del desarrollo del proyecto, posteriormente se analizan estos riesgos, buscando sus posibles causas y consecuencias, conjuntamente, se les otorga un valor que es la relación entre el impacto que tiene el riesgo el los objetivos del proyecto y la probabilidad de que ese proyecto ocurra, esto es guiado por unas métricas ya establecidas y que estos valores no se otorguen aleatoriamente, luego analizamos las acciones que se están tomando en este momento para hacer frente al riesgo, y consecuentemente se define la acción correctiva a realizar, si se evita, mitiga, acepta o terceriza, con esta decisión tomada se busca una solución que se acomode a la forma de resolver el riesgo, definiendo también el responsable de esta acción correctiva y como un proceso iterativo se toma nota de las observaciones referentes al riesgo en análisis; el siguiente BPMN muestra el proceso ya descrito. [2]

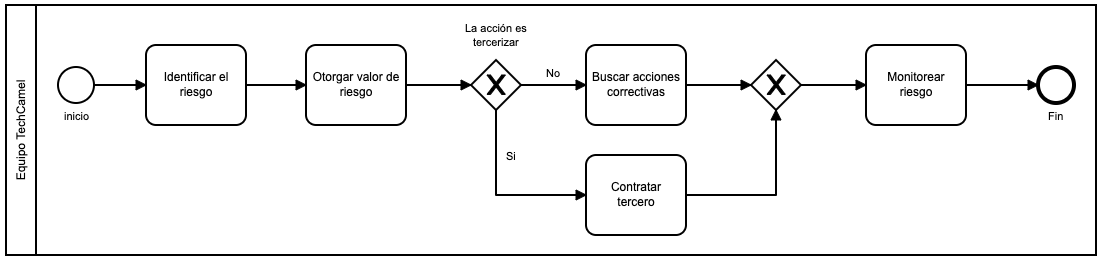


Figura 9. Diagrama BPMN del proceso de identificación y análisis de riesgo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1. Identificación del Riesgo** | | | | | |
| **ID-RIESGO** | **Clasificación de Riesgo** | **Riesgos  (¿qué puede suceder?)** | **Causas  (¿Debido a?)** | **Consecuencias (¿Qué podría ocurrir?)** | **Fecha de Identificación** |
|  |
| T-101 | Técnico | Los integrantes del proyecto eligieron herramientas en las que no se tiene experiencia | Deseo por aprender nuevas herraminetas, voluntad de crecimiento profesional | Baja productividad, baja calidad del producto entregado, redistribución de roles y tareas, sobrecarga de tareas en un miembro, afectación del ambiente laboral | 4/09/20 |  |
| I-103 | Interno | Fallo en la planificación de actividades | Estimación de tiempos y tareas imprecisa, Tareas asignadas de forma ineficiente | Baja calidad del producto entregado, sobrecarga de tareas a un miembro, afectación en el ambiente laboral | 4/09/20 |  |
| I-101 | Interno | Contagio de COVID-19 u otra enfermedad | Situación mundial a nivel sanitario | Probable abandono del proyecto, disminución en la productividad | 4/09/20 |  |
| I-105 | Interno | Abandono de un miembro del equipo | Motivos personales, problemas interpersonales, carencia de tiempo, dar de baja la materia, Peligro inminente de muerte, | Redistribución de roles y tareas, mayor carga para cada integrante | 4/09/20 |  |
| E-103 | Externo | Bajo interés por parte de los potenciales clientes | Alta competencia, desconfianza del servicio, situación mundial a nivel sanitario | Reducción de ingresos, implementar cambios operacionales y/o publicitarios | 4/09/20 |  |
| I-102 | Interno | Incumplimiento de actividades | Licencia por fallecimiento de familiar, Carencia de herramientas para cumplir con las actividades, indisposición respecto al proyecto, conflictos personales | Redistribución de roles y tareas, mayor carga para cada integrante | 4/09/20 |  |
| I-104 | Interno | Explicación poco clara o incompleta de los requerimientos | Poca precisión en la definición de los requerimientos, omisión de detalles | Se incrementa las tareas, se ve afectado el cronograma, baja calidad del producto entregado | 4/09/20 |  |
| E-102 | Externo | Clientes pueden incurrir en asuntos ilegales | El cliente decide realizar actos ilegales | Deterioro de imagen comercial, incurrir en gastos imprevistos, insatisfacción del cliente | 4/09/20 |  |
| I-106 | Interno | El presupuesto económico para el proyecto no fue suficiente | Mala estimación de los recursos necesarios | Se presenta un desbalance financiero en el proyecto | 4/09/20 |  |
| E-101 | Externo | Cambio de normas en la entrega del proyecto | Se presenta un cambio en el cronograma o en los requerimientos | Baja calidad del producto entregado, sobrecarga de tareas a un miembro | 4/09/20 |  |

Tabla 19.Identificacion de riesgos

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2. Valoración del  Riesgo** | | | **3. Controles** | **4. Opciones de tratamiento** | **5. Tratamiento del Riesgo** | | | **6. Monitoreo** |
| **Probabilidad** | **Impacto** | **Valor Riesg** | **Controles Existentes** | **MITIGAR ACEPTAR  EVITAR TRANSFERIR** | **ACTIVIDAD** | **RESPONSABLE** | **FECHA** | **Fecha - Observaciones** |
|  |
| 4 | 5 | **20** | Se toman capacitaciones para adquirir conocimiento frente a las herramientas seleccionadas | Evitar | Incurrir en actividades de refuerzo | Director de desarrollo |  | Se realizarán a lo largo de desarrollo del proyecto |  |
| 4 | 4 | **16** | Reuniones de planificación donde se llega a un concenso | Mitigar | Monitoreo constante del progreso personal y grupal, seguir el conducto regular de las reglas, reuniones semanales. | Equipo TechCamel |  | Se realizarán a lo largo de desarrollo del proyecto |  |
| 3 | 4 | **12** | No existe aún un control | Aceptar | Redistribuir tareas en el equipo | Director de desarrollo |  | Se realizarán a lo largo de desarrollo del proyecto |  |
| 2 | 5 | **10** | No existe aún un control | Aceptar | Redistribuir tareas en el equipo | Director de desarrollo |  | Se realizarán a lo largo de desarrollo del proyecto |  |
| 2 | 5 | **10** | Retroalimentación por parte de los clientes | Transferir | Implementar estrategias de publicidad y mercadeo | Empresa experta en publicidad |  | Se realizarán a lo largo de desarrollo del proyecto |  |
| 2 | 4 | **8** | Seguimiento de reglas internas del grupo | Mitigar | Monitoreo constante del progreso personal y grupal, seguir el conducto regular de las reglas, reuniones semanales. | Director del proyecto |  | Se realizarán a lo largo de desarrollo del proyecto |  |
| 2 | 4 | **8** | Reuniones de planificación donde se llega a un concenso | Mitigar | Reunión para redefinir alcance y requerimientos | Equipo TechCamel |  | Se realizarán a lo largo de desarrollo del proyecto |  |
| 2 | 4 | **8** | Verificación de antecedentes | Aceptar |  |  |  | Se realizarán a lo largo de desarrollo del proyecto |  |
| 2 | 4 | **8** | Estudio previo de costos relacionados al proyecto | Mitigar | Realizar un estudio de mercado con una estrategia diferente, replantear el plan de distribución de recursos | Jefe de calidad |  | Se realizarán a lo largo de desarrollo del proyecto |  |
| 1 | 4 | **4** | no existe aún un control | Aceptar | Redistribuir tareas en el equipo | Director de desarrollo |  | Se realizarán a lo largo de desarrollo del proyecto |  |

Tabla 20.Analisis y gestión de riesgos

## Administración de configuración y documentación

Control de versiones

Para controlar las versiones de cada uno de los entregables, se revisará en grupo el trabajo hecho por cada uno de los integrantes, para así tener una mirada objetiva de cada punto, siendo así juzgado y retroalimentado cada parte de los documentos. Además, los aportes de cada integrante serán registrados en la tabla de historial de cambios, junto con el momento en el que se realizó.

Elementos de configuración

Los elementos que identificamos para la configuración son los siguientes:

1. Planes: Cada de estos documentos son relevantes para la ejecución y transcurso del proyecto, pues en ellos se describe el actuar del proyecto. Por lo tanto, es necesaria su actualización y revisión.

* Plan de proyecto
* Plan de calidad
* Plan de gestión de configuración
* Plan de Gestión de Riesgos

1. Especificación de Requisitos: En esta parte del documento se define el comportamiento que tendrá nuestro proyecto. Incluye los casos de uso, junto con las interacciones que va a tener el usuario con el software.

* Requisitos de negocio
* Requisitos de usuario
* Requisitos de sistema

1. Documentos de Diseño:

* Diagrama de Clases
* Diagramas de Caso de Uso

1. Código Fuente: Es el desarrollo del sitio web, por parte de cada uno de los integrantes del equipo. De este depende el funcionamiento del programa
2. Documentos de Manual de Usuario: Este documento hecho por los programadores, le permitirá al usuario aprender el uso del programa.

Proceso de control de cambios para los ítems de configuración

En el desarrollo del proyecto podemos encontrar una gran cantidad de elementos de configuración que abarcan elementos de código y de documentos. En el siguiente diagrama vemos como todos los componentes del proyecto estarán almacenados en un Repositorio, luego de la revisión de cada uno de los elementos hechos por uno de los integrantes, el grupo lo revisará y destinará la aceptación o cambio de este, y nuevamente el documento será actualizado en el repositorio de datos.

Adicionalmente, cuando alguno de los integrantes decida hacer una modificación en alguna de las áreas del proyecto, ya sea para la mejora o para la corrección en cualquiera de las áreas ya mencionadas anteriormente. Este miembro realiza un estudio para él cambio, que posteriormente es presentado al jefe de configuración, el cuál después de analizar determina si es viable o inviable el cambio.

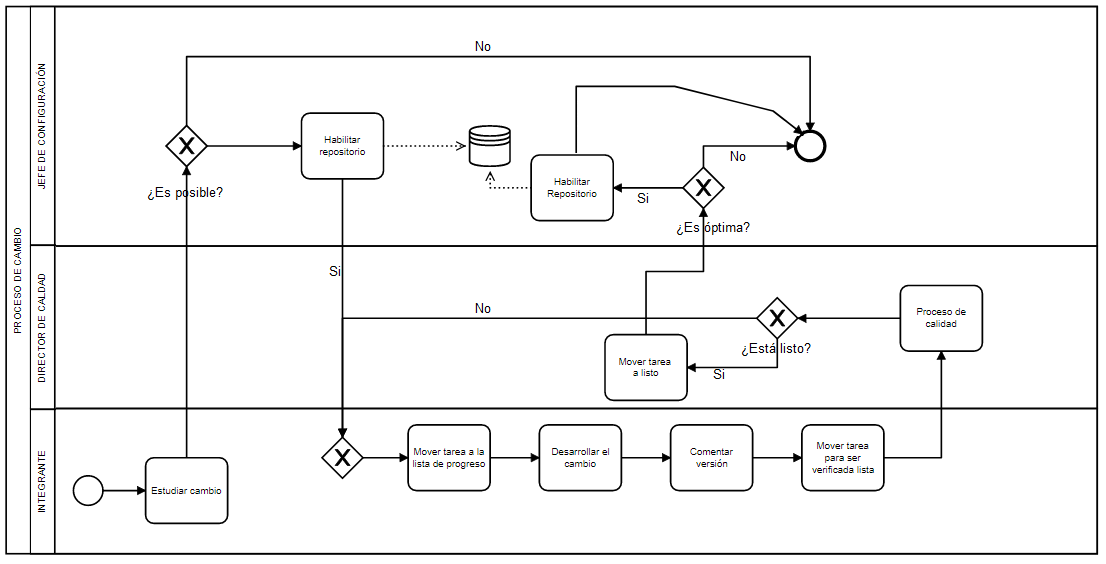
Una vez aprobado el cambio, el jefe de configuración, habilitará el repositorio de datos en la sección “cambios y correcciones” para que el integrante acceda y proceda a hacer el cambio. Una vez realizado el cambio, pasará a ser verificada por el director de calidad quien determinará si la mejora es óptima. Si está es aprobada el jefe de configuración permitirá el cambio en el Repositorio en la sección correspondiente. [1]

Figura 10. Proceso de Cambio (Mejora)

Elementos de documentación

Uno de los integrantes del equipo es el encargado de supervisar la actividad, sin embargo, todos los desarrolladores deben documentar su código y deben ser claros al definir las clases, objetos, variables y demás elementos de la base de datos. Con el propósito de que sea sencillo entender el código que consigo trae el facilitamiento de las tareas del encargado de documentar, junto con los demás programadores para realizar de manera más eficiente su trabajo.

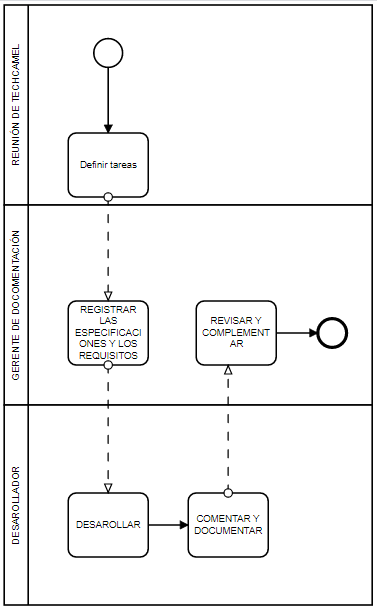


Figura 11.Proceso Documentación

A continuación, podemos observar cada uno de los elementos a documentar, con su descripción y los momentos en los que se crean o refinan a lo largo del desarrollo del proyecto.

|  |  |  |
| --- | --- | --- |
| **Elemento de Documentación/Código** | **Descripción** | **Momento de Creación / Refinación** |
| **Software Project Plan de gestión** | Documento que explica en detalle la planificación del proyecto. Incluye la descripción de los objetivos. estimaciones, programación, recursos, riesgos, planes actividades. | Primer y segunda entrega. |
| **Especificación de requisitos de software** | Descripción detallada del desarrollo del proyecto. Incluye todos los requisitos para el desarrollo del proyecto incluidos los requisitos funcionales y no funcionales. | Segunda y tercera entrega |
| **Descripción del diseño del software** | Descripción del producto de software que proporciona una guía para el equipo de desarrollo sobre la arquitectura y el diseño del proyecto. | Tercera Entrega |
| **Manual del usuario** | Manual que indica la instalación y el uso del producto de software está orientado al cliente. | Tercera Entrega |
| **El código fuente** | Desarrolló un software. | Segunda y tercera entrega |
| **Diagramas de casos de uso** | Diagrama que muestra cómo el cliente utiliza el sistema. Incluye cómo interactúan los elementos. | Segunda entrega. |
| **Diagramas de clase** | Diagrama que muestra los objetos del sistema sus atributos, operaciones y relaciones entre ellos atributos, operaciones y relaciones en.t.re ellos. | Segunda entrega |

Tabla 21.Elementos de documentación y código

## Control de calidad

El control de calidad puede ayudar al equipo a realizar seguimiento del proyecto, y a alcanzar las metas propuestas en términos del plan de aceptación del producto. Definir una forma clara para realizar el control de calidad del producto también reducirá posibles riesgos, que, en etapas avanzadas, podrían llegar a ser muy costosos.

Este proceso correctivo identifica los ítems a mejorar o incompletos en el caso de documentos, y bugs y errores para el código e implica realizar una actividad de validación del producto versus los requerimientos que fueron acordados previamente. Esta validación es responsabilidad del inspector del director de calidad o del equipo de testers.

El proceso para realizar el control de calidad inicia cuando una tarea esta lista para ser verificada, sea documentación o parte del código fuente, luego, las partes encargadas de realizar el control de calidad serán notificadas para que definan las métricas para evaluar la completitud del entregable. Finalmente, estas partes notificaran si los requerimientos fueron satisfechos y de no haberlos cumplido, se realizarán las modificaciones que pertinentes hasta que los cumpla para que la tarea sea clasificada como finalizada. [4]

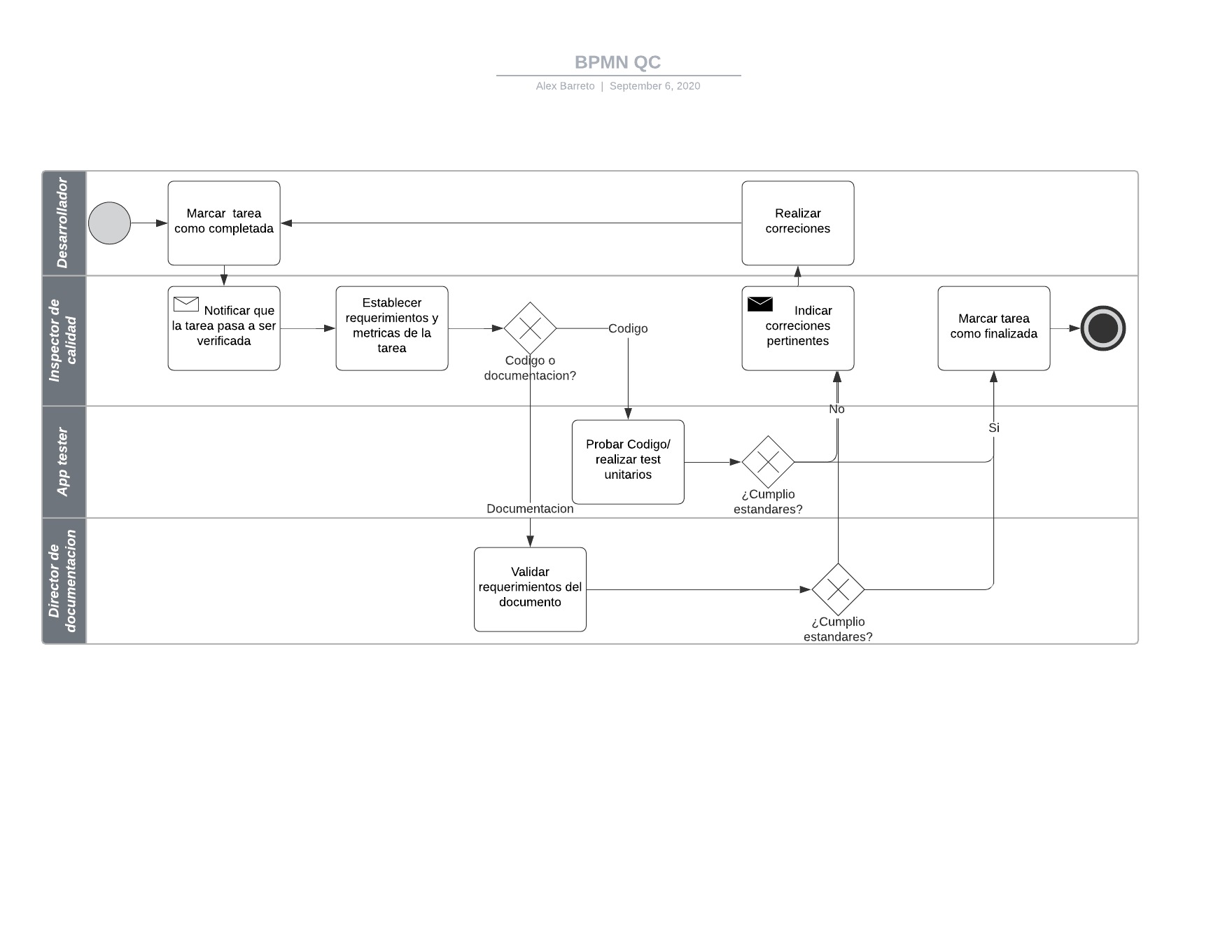


Figura 12.BPMN para representar la gestión de control de calidad

Los procesos que hacen parte del control de calidad (QC) no se limitan a los tests del producto final para que cumpla con el plan de aceptación únicamente. Además de las pruebas, incluye revisiones y análisis de requisitos, planificación, diseño, casos de prueba, entre otros elementos:

|  |  |  |  |
| --- | --- | --- | --- |
| Proceso/Actividad | Descripción /Motivación | Encargado | Momento en el que ocurre |
| Revisión de Requerimientos | Proceso manual donde se revisan posibles anomalías u omisiones en los requerimientos.  Los requisitos bien escritos y la comprensión de estos por parte del equipo son fundamentales para el éxito del proyecto. | Líder del Proyecto junto a la representación del cliente (generalmente es el producto owner). | Idealmente se realizan antes de la reunión de planificación del sprint, para que el contenido de este no cambie drásticamente. |
| Revisión de código | Para mejorar la calidad del código, compartir conocimiento entre el equipo, hacerlo más comprensible para futuras modificaciones.  Es clave para evitar posibles bugs o errores de lógica en el proceso de testing. | Desarrolladores junto al líder de desarrollo | Debido a la metodología usada, el proceso de revisión suele hacerse de forma frecuente, de forma que los errores encontrados sean resueltos cuanto antes. |
| Revisión de documentos | Si bien este es un aspecto secundario en Scrum, se debe hacer un checklist de los elementos de los documentos, sean entregables o para efectos de comprensión del proyecto y posibles modificaciones | Director de documentación | Previo a la entrega del software o de la finalización de cada Sprint. |
| Software Testing | Sigue los principios del desarrollo ágil. Supone una integración continua entre el desarrollo y las pruebas. No es secuencial en el sentido de que se ejecuta solo después de la fase de codificación sino continua. | Todos los desarrolladores junto al líder de desarrollo. | Se realiza de forma iterativa, es decir en todos los Sprint, y paralelamente al desarrollo. |
| Pruebas Unitarias | Significa probar individualmente las partes que componen el software, definiendo resultados que se esperan obtener para ciertas entradas o escenarios y verificar si ocurren |

Tabla 22.Control de calidad

# Referencias

|  |  |
| --- | --- |
| [1] | VMEdu, SBOK, Arizona: SCRUMstudy, 2016. |
| [2] | PMI, PMBOK, Pennsylvania: Project Management Institute, Inc., 2017. |
| [3] | U. o. Michigan-Dearborn, «Calculadora Puntos de Casos de Uso,» [En línea]. Available: http://groups.umd.umich.edu/cis/tinytools/cis375/f17/team9-use-case-pts/Use\_Case\_Point\_Calculator/#top\_page. [Último acceso: 3 Septiembre 2020]. |
| [4] | «Altexsoft,» [En línea]. Available: https://www.altexsoft.com/whitepapers/quality-assurance-quality-control-and-testing-the-basics-of-software-quality-management/. [Último acceso: 2 Septiembre 2020]. |