

Sistema de asistencia a cuidadores de personas con discapacidades cognitivas

Planificación y Seguimiento por Sprint

CARRERA: Ingeniería en Sistemas de Información

ASIGNATURA: Proyecto final

CICLO LECTIVO: 2021

CURSO: 5K1

FECHA DE ENTREGA: 27/04/2021

DOCENTES:

- Ing. Cecilia Ortiz
- Ing. María Irene Mac William
- Ing. Lorena Barale

INTEGRANTES:

- Abrego Pérez, Joaquín - 70190
- Barbadillo Incerti, Oriana - 72762
- Lucero, Cristian - 58785
- Pazos, Cesar - 76543
- Primo, Matías - 72078
- Zanotti, Ignacio - 69860

Historial de versiones

N°	Descripción	Autor	Fecha
1.0	Entrega inicial.	Abrego Pérez, Joaquín Barbadillo Incerti, Oriana Lucero, Cristian Primo, Matías Pazos, Cesar Zanotti, Ignacio	27/04/2021

Índice

Historial de versiones	2
Índice	3
Introducción	4
Sprint 0 - Definiciones Generales del Proyecto - Acuerdos	4
Definición de Product Owner	4
Definición del Equipo y Scrum Master	4
Acuerdos	5
Definición del Product Backlog Inicial	5
Story Map	9
Definición de la Herramienta de software para la gestión de proyectos a utilizar	12
Jira	12
Forma de coordinación del trabajo	13
Ciclo de Vida de los artefactos a utilizar en el desarrollo del proyecto	14
Template US (que se va a utilizar para detallar las US)	14
Técnica de estimación a utilizar	16
Definición de la tecnología a utilizar en el desarrollo del producto	16
Métricas del proyecto	16
Burndown Chart	16
Velocidad	18
Pautas de codificación y testing	19
Pautas de Codificación	19
Pautas para Testing	20
Gestión de Configuración	21
Organización de la Documentación	22
Ítems de Configuración	23
Definición de Línea Base	24

Introducción

En este primer sprint, no se comenzará a desarrollar lo concreto del proyecto sino que se redacta una explicación del contexto, así como la metodología de trabajo que se adoptará para llevar adelante el mismo. Se mencionan acuerdos establecidos entre los miembros del equipo, una primera versión del listado de tareas a desarrollar y la fecha estimada de su implementación. Por último, están incluidos aspectos técnicos, como la tecnología a emplear, las métricas para evaluar el avance del proyecto y los lineamientos para implementar y validar las funcionalidades del software.

Sprint 0 - Definiciones Generales del Proyecto - Acuerdos

Definición de Product Owner

El Product Owner de este proyecto será por defecto la Ing. Barale Lorena, dado que no se cuenta con un especialista en el dominio de aplicación actual. Las diferentes instituciones con las que el equipo se comunicó no recibió respuesta alguna para una reunión o charla para discutir la idea del desarrollo de la aplicación.

Definición del Equipo y Scrum Master

- **Scrum Master:** El rol del Scrum Master será desempeñado rotativamente por todo el equipo, dado que ninguno cuenta con experiencia real en conducir un equipo Scrum. De esta forma, se proveerá a todo el personal de la experiencia inicial de cómo dirigir el equipo y controlar su desempeño, así como el análisis de la mejora conjunta en cada sprint.
- **Lead Back-end Developer:** Lucero, Cristian.
- **Lead Front-end Developer:** Primo, Matías.
- **Front-end developer:** Barbadillo, Oriana ; Abrego, Joaquín.
- **Functional Analyst:** Barbadillo, Oriana.
- **Back-end Developer:** Pazos, César.
- **DBA:** Pazos, César.
- **Lead Tester & QA:** Zanotti, Ignacio.
- **UX Designer:** Abrego, Joaquín.

Acuerdos

Team Canvas Basic

Analizar las cuestiones más importantes en las que están de acuerdo al iniciar un equipo de trabajo efectivo y conseguir que los miembros se conozcan mejor entre ellos.

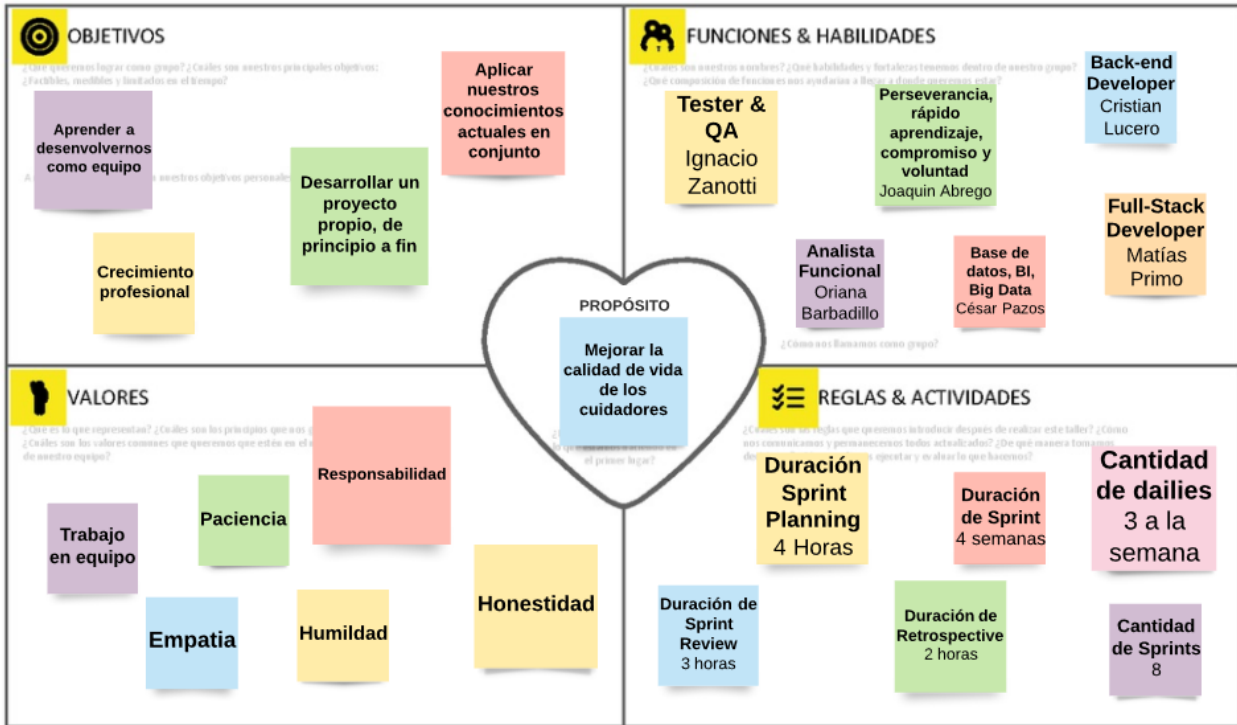
Version 0.8 | theteamcanvas.com | hello@theteamcanvas.com

Nombre del Equipo

Main Team

Fecha

27/4/2021



Definición de Ready

- La tarea debe tener una estimación de complejidad (Story Points), para su correcta priorización respetando los estándares de SCRUM.
- El detalle funcional, detalle técnico y los casos de prueba de cada tarea deben ser claros para que sean entendibles por cualquier miembro del equipo, así como del PO.
- La tarea debe tener un propósito claro dentro del contexto del sprint, no estar aislados del resto de las tareas, por lo tanto, **la tarea debe poder validarse y verificarse dentro del Sprint**.
- La tarea no debe tener bloqueos que impidan su ejecución, es decir, la tarea no puede estar pendiente por falta de equipos/herramientas, etc.
- Las dependencias deben estar resueltas.

Definición de Done

- Cada historia de usuario/funcionalidad está codificada, compilada, desplegada en el contexto de desarrollo y preproducción actual.
- Las historias de usuario/funcionalidades cumplen todos los criterios de aceptación acordados con calidad(QA).
- Las historias de usuario/ funcionalidades construidas no afectan a las funcionalidades ya entregadas, por lo que las **pruebas de regresión deben ser satisfactorias**.

- Las funcionalidades/historias de usuario comprometidas funcionan en todos los ambientes comprometidos (Desarrollo y Preproducción).
- El trabajo de cada miembro del equipo ha sido revisado por al menos otro miembro del equipo.
- La funcionalidad/historia codificada actualizada en la rama de desarrollo del repositorio.

Definición del Product Backlog Inicial

Especificación de Backlog de tareas	
Nro.	Título
1	Registrar paciente
2	Consultar datos de paciente
3	Listar pacientes
4	Modificar datos de paciente
5	Dar de baja a un paciente
6	Registrar nuevo cuidador
7	Consultar datos de cuidador
8	Listar cuidadores
9	Modificar datos de cuidador
10	Dar de baja a cuidador
11	Registrar nuevo grupo de cuidadores
12	Consultar datos de grupos de cuidadores
13	Dar de baja a un grupo de cuidadores
14	Modificar datos de grupo de cuidadores
15	Agregar cuidador a un grupo
16	Agregar paciente a un grupo
17	Listar grupos de cuidadores

Universidad Tecnológica Nacional
Facultad Regional Córdoba
Dpto. Ingeniería en Sistemas de Información

Asignatura: Proyecto Final

Tema: Sprint 0

18	Listar cuidadores de grupo
19	Listar pacientes de grupo
20	Registrar evento de paciente
21	Consultar datos de evento
22	Modificar datos de evento
23	Postergar evento de paciente
24	Registrar fin de un evento
25	Dar de baja a evento
26	Listar Eventos de paciente
27	Registrar Tipo de Evento
28	Consultar Tipo de Evento
29	Modificar Tipo de Evento
30	Listar Tipo de Evento
31	Registrar baja de tipo de Evento
32	Registrar nueva actividad
33	Consultar datos de actividad
34	Modificar datos de actividad
35	Listar actividades
36	Registrar baja de actividad
37	Registrar inicio de actividad
38	Registrar progreso de actividad
39	Registrar fin de actividad
40	Consultar actividades en progreso
41	Reiniciar actividad
42	Registrar cancelación de actividad

Universidad Tecnológica Nacional
Facultad Regional Córdoba
Dpto. Ingeniería en Sistemas de Información

Asignatura: Proyecto Final

Tema: Sprint 0

43	Registrar Institución
44	Consultar datos de institución
45	Modificar datos de institución
46	Registrar baja de institución
47	Listar instituciones
48	Registrar nuevo Tipo de Institución
49	Modificar Tipo de Institución
50	Consultar Tipo de Institución
51	Listar tipos de Institución
52	Registrar baja de tipo de institución
53	Buscar instituciones por ubicación
54	Buscar instituciones por nombre
55	Buscar instituciones por Tipo
56	Generar reporte de desempeño por actividad y por paciente
57	Generar reporte de turnos de paciente
58	Generar reporte de toma de medicamentos
59	Consultar asistente virtual
60	Enviar notificación de cuidador
61	Enviar notificación de grupo de cuidador
62	Armado de Diagrama de clase
63	Armado de Modelo de datos
64	Armado de modelo de Front-end para pantallas móviles
65	Armado de maqueta de arquitectura back-end
66	Registrar usuario
67	Iniciar Sesión

Universidad Tecnológica Nacional
Facultad Regional Córdoba
Dpto. Ingeniería en Sistemas de Información

Asignatura: Proyecto Final

Tema: Sprint 0

68	Modificar Usuario
69	Consultar Usuario
70	Listar Usuarios
71	Registrar baja de usuario
72	Ingresar a la aplicación
73	Capacitación React-Native
74	Capacitación Microservicios
75	Capacitación MongoDB
76	Configuración Conexiones de base de datos

Story Map

Sprint	N° User Story	Nombre
1	62	Armado de Diagrama de clase
1	63	Armado de Modelo de datos
1	64	Armado de modelo de Front-end para pantallas móviles
1	65	Armado de maqueta de arquitectura back-end
1	73	Capacitación React-Native
1	74	Capacitación Microservicios
1	75	Capacitación MongoDB
2	20	Registrar evento de paciente
2	21	Consultar datos de evento
2	22	Modificar datos de evento
2	23	Postergar evento de paciente
2	24	Registrar fin de un evento
2	25	Dar de baja a evento
2	26	Listar Eventos de paciente
2	28	Consultar Tipo de Evento
2	30	Listar Tipo de Evento
3	1	Registrar paciente
3	2	Consultar datos de paciente
3	3	Listar pacientes
3	4	Modificar datos de paciente
3	5	Dar de baja a un paciente
3	6	Registrar nuevo cuidador
3	7	Consultar datos de cuidador

Universidad Tecnológica Nacional
Facultad Regional Córdoba
Dpto. Ingeniería en Sistemas de Información

Asignatura: Proyecto Final

Tema: Sprint 0

3	8	Listar cuidadores
3	9	Modificar datos de cuidador
3	10	Dar de baja a cuidador
3	76	Conexión con la Base de datos
4	43	Registrar Institución
4	44	Consultar datos de institución
4	45	Modificar datos de institución
4	46	Registrar baja de institución
4	47	Listar instituciones
4	48	Registrar nuevo Tipo de Institución
4	49	Modificar Tipo de Institución
4	50	Consultar Tipo de Institución
4	51	Listar tipos de Institución
4	52	Registrar baja de tipo de institución
4	53	Buscar instituciones por ubicación
4	54	Buscar instituciones por nombre
4	55	Buscar instituciones por Tipo
5	27	Registrar Tipo de Evento
5	29	Modificar Tipo de Evento
5	31	Registrar baja de tipo de Evento
5	32	Registrar nueva actividad
5	33	Consultar datos de actividad
5	34	Modificar datos de actividad
5	35	Listar actividades
5	36	Registrar baja de actividad

Universidad Tecnológica Nacional
Facultad Regional Córdoba
Dpto. Ingeniería en Sistemas de Información

Asignatura: Proyecto Final

Tema: Sprint 0

5	37	Registrar inicio de actividad
5	38	Registrar progreso de actividad
5	39	Registrar fin de actividad
5	40	Consultar actividades en progreso
5	41	Reiniciar actividad
5	42	Registrar cancelación de actividad
6	60	Enviar notificación de cuidador
6	61	Enviar notificación de grupo de cuidador
6	11	Registrar nuevo grupo de cuidadores
6	12	Consultar datos de grupos de cuidadores
6	13	Dar de baja a un grupo de cuidadores
6	14	Modificar datos de grupo de cuidadores
6	15	Agregar cuidador a un grupo
6	16	Agregar paciente a un grupo
6	17	Listar grupos de cuidadores
6	18	Listar cuidadores de grupo
6	19	Listar pacientes de grupo
7	66	Registrar usuario
7	67	Iniciar Sesión
7	68	Modificar Usuario
7	69	Consultar Usuario
7	70	Listar Usuarios
7	71	Registrar baja de usuario
7	72	Ingresar a la aplicación
7	56	Generar reporte de desempeño por actividad y por paciente

7	57	Generar reporte de turnos de paciente
7	58	Generar reporte de toma de medicamentos
7	59	Consultar asistente virtual

Definición de la Herramienta de software para la gestión de proyectos a utilizar

Jira

La herramienta Jira es una solución en línea que brinda soporte para la administración de las tareas de un proyecto, el seguimiento de errores e incidencias y la gestión operativa del proyecto en sí. La misma es provista y mantenida por el proveedor australiano Atlassian, y se creó de forma inicial para el desarrollo de software, siendo un apoyo para la gestión de requisitos, seguimiento del estado de los desarrollos, y finalmente la gestión de errores. Esta última funcionalidad fue añadida en las siguientes versiones de la herramienta.

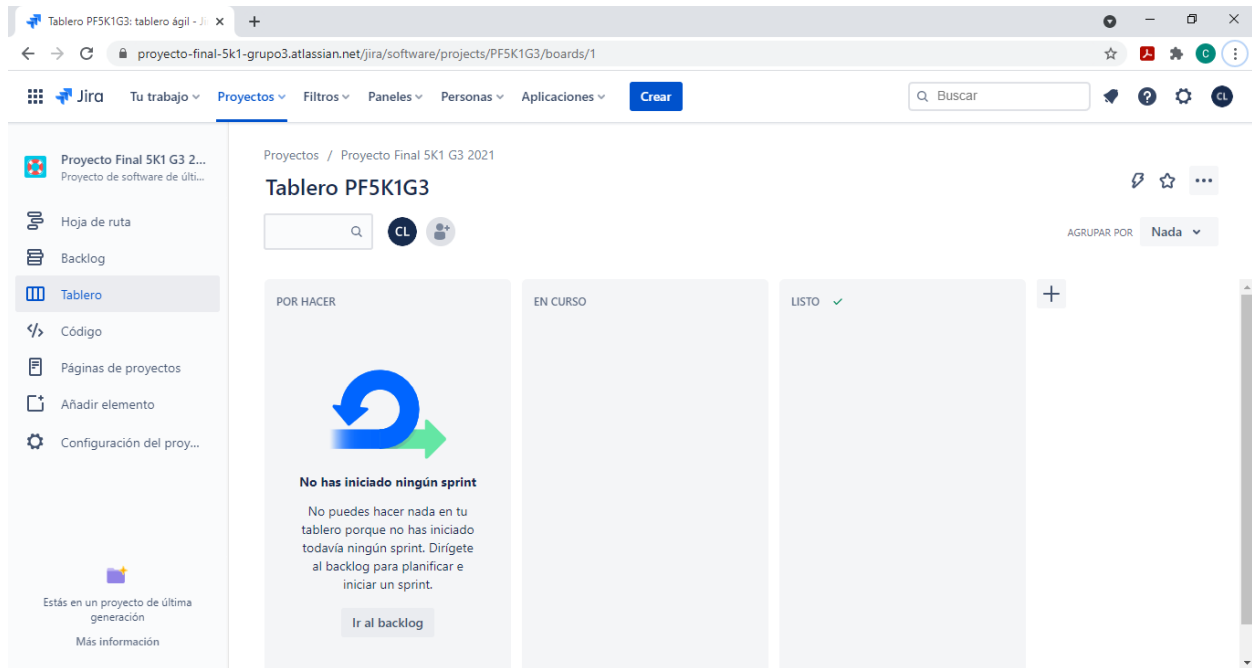
Podemos mencionar los siguientes motivos por los cuales se eligió esta herramienta:

- Es la solución más popular a nivel mundial para la gestión de proyectos, ya que se considera una herramienta completa, que brinda apoyo al ciclo de vida completo de cada uno de éstos.
- Permite aplicar la metodología Scrum para dicha gestión de proyecto, adaptando las características del framework a las necesidades de cada equipo.
- Brinda un espacio gratuito en la nube, posibilitando el trabajo simultáneo de hasta 10 participantes.

Este último ítem resulta fundamental para el proyecto a llevar a cabo, ya que nos da la flexibilidad de poder sumar a los docentes e interesados en el mismo, con la posibilidad de dar seguimiento al trabajo en cualquier momento que se desee.

El espacio de trabajo en la nube será el siguiente:

<https://proyecto-final-5k1-grupo3.atlassian.net/>



Esta última imagen nos muestra la página inicial del proyecto, en el cual Jira nos genera de manera automática un tablero al estilo Kanban, para poder organizar las tareas asociadas al desarrollo de dicho proyecto. Dicho tablero puede ser configurado a medida, en base a las necesidades del equipo.

Por último, podemos destacar además la fuerte integración que tiene Jira con otras herramientas que hacen a la ejecución de las tareas propias del proyecto, como ser aplicaciones para la comunicación del equipo, documentación en repositorio, versionado de componentes, etc.

Forma de coordinación del trabajo

Semanalmente, se realizarán 3 reuniones entre el equipo y el Scrum Master, a través de Discord, para llevar a cabo la Daily. Scrum define que las dailies son reuniones diarias, pero debido a cuestiones de tiempo y las exigencias académicas y laborales de los miembros del equipo, esto no es posible. Estas reuniones se llevarán a cabo los martes, jueves y sábado de cada semana, en las cuales el equipo comentará lo que logró hacer hasta el momento, los problemas con los que se encontró y lo que hará hasta la próxima reunión.

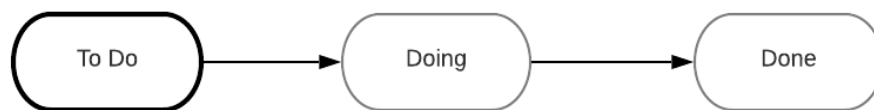
Por otro lado, el resto de las reuniones (Sprint Planning, Review y Retrospective) se llevarán a cabo según lo especificado por Scrum, al comienzo y fin del sprint, respectivamente. Además de las reuniones obligatorias, llevaremos a cabo un Backlog Refinement, previo a cada sprint.

Respecto a los sprints propiamente dichos, trabajaremos con una duración de sprint de 4 semanas, ya que el equipo no es experto en programación, por lo que debemos contemplar spikes para capacitarnos en el uso de ciertas tecnologías.

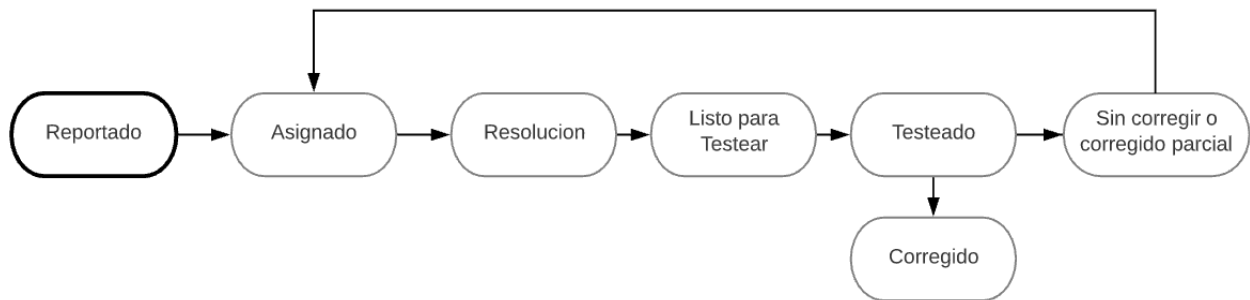
Ciclo de Vida de los artefactos a utilizar en el desarrollo del proyecto

Los artefactos del proyecto son los siguientes:

- Definidos por Scrum:
 - **Product Backlog:** irá evolucionando de manera continua a través de una reunión de Grooming donde se encontrarán nuevas User Stories y se mejorarán las siguientes.
 - **Sprint Backlog:** se definirá en el Sprint Planning y, una vez definido, no cambiará. Comprenderá únicamente aquellas US que hayamos determinado como próximas a desarrollar, siendo siempre, la US bien definida y estimada.
 - **Incremento del producto:** habrá un incremento del producto al final de cada Sprint el cual comprende la funcionalidad ya desarrollada en Sprints previos y la nueva funcionalidad que originó como salida el último Sprint.
- **User Stories:** estas irán evolucionando a medida que avance el proyecto. Las mismas serán descompuestas en tareas más concisas que nos ayudarán con el desarrollo. El Ciclo de Vida de dichas tareas y las Users será el siguiente:



- **Defectos:**

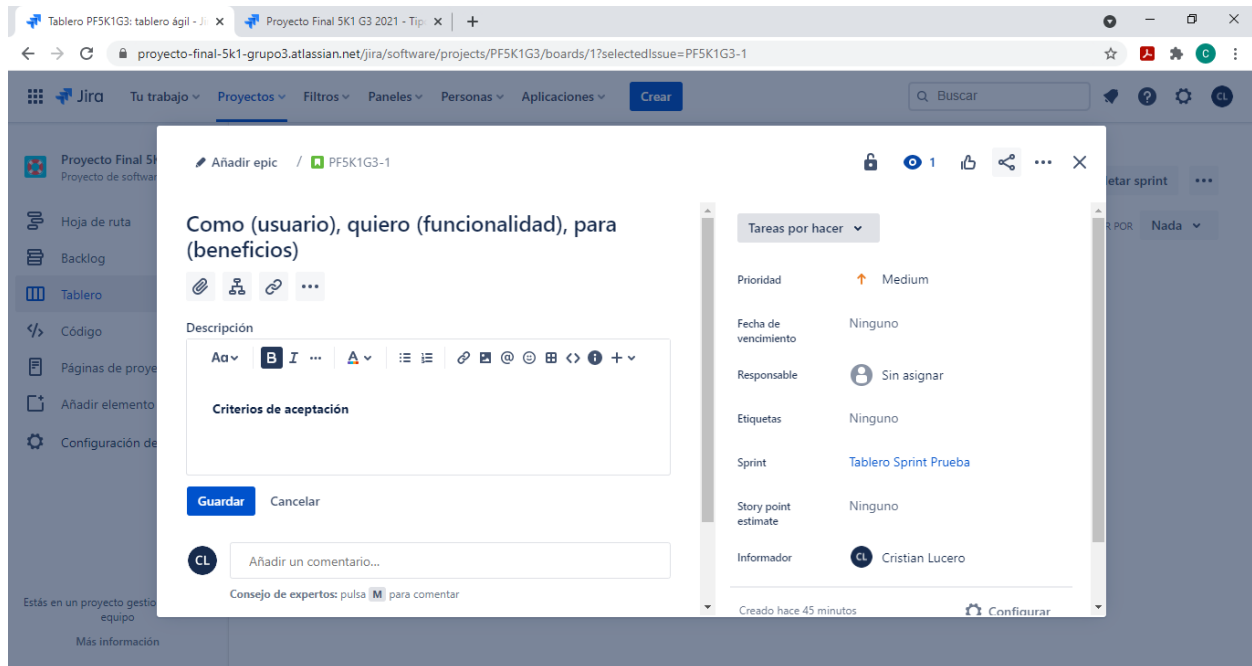


Template US (que se va a utilizar para detallar las US)

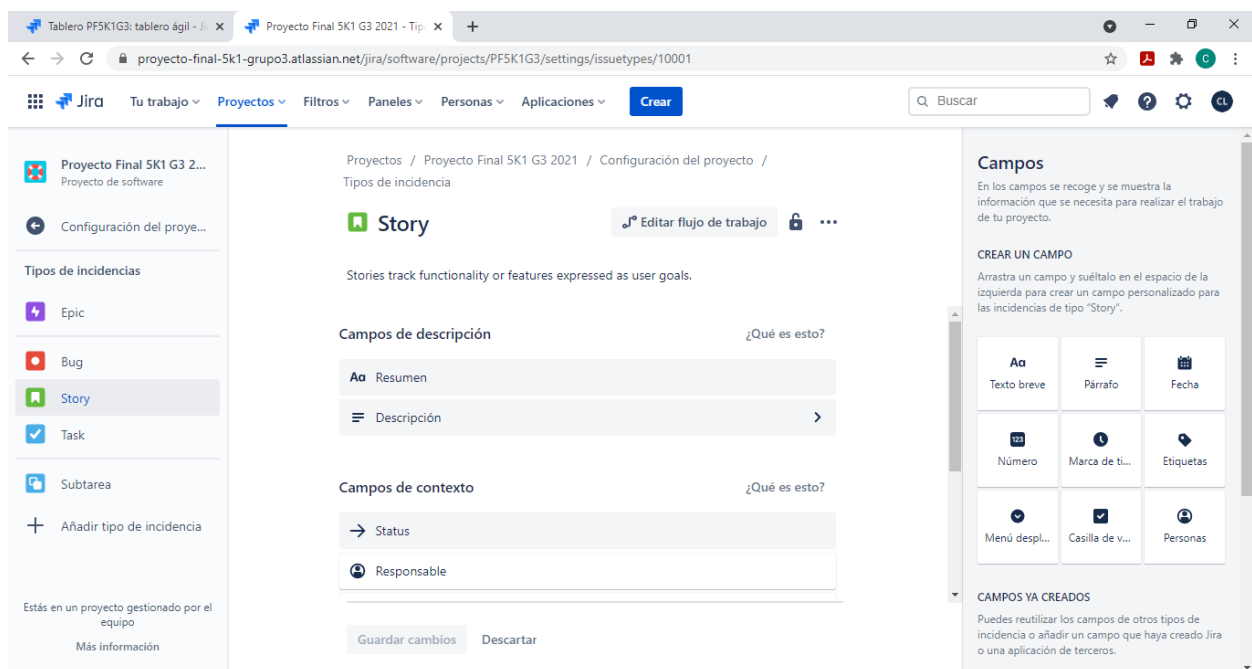
Nº - Nombre de User Story	Prioridad	Estimación
Como (usuario), quiero (funcionalidad), para (beneficio).		
Criterios de aceptación		

El template presentado en el apartado anterior será configurado en la herramienta Jira, aprovechando las funcionalidades de adaptación que brinda la misma.

En la siguiente imagen se verifica el formato por defecto que ofrece Jira para la US, y la opción de configuración para realizar una mayor adaptación, en caso de ser requerido:



En la siguiente visualización se pueden observar las opciones de configuración que dispone la herramienta:



Técnica de estimación a utilizar

En cada baraja hay una pseudo-secuencia de Fibonacci modificada. Cada participante tendrá las siguientes cartas: 0, 1/2, 1, 2, 3, 5, ?, e infinito. El cero significa que la historia ya está hecha o no requiere ningún esfuerzo, el interrogante significa que nos falta información para estimar esa historia o tarea y el infinito es que es demasiado grande y hay que dividirla en historias más pequeñas.

Durante la ceremonia, una por una se leen y discuten las User Stories. Una vez todos tienen claro en qué consiste cada una, se elige una carta en función del esfuerzo que se prevé requerirá esa historia. Solo estiman los que después desarrollan (ni el Scrum Master ni el Product Owner estiman, solo resuelven dudas).

Al final de la sesión el resultado es una estimación consensuada y validada por todo el equipo para cada una de las historias o tareas seleccionadas.

Definición de la tecnología a utilizar en el desarrollo del producto

Lenguaje de Programación:

- **React-Native:** Se utilizará react-native (que involucra Node.js, HTML, CSS, etc.) como principal lenguaje de programación para el proyecto. Esto se debe a que la aplicación debe ser adaptable a diferentes tipos de pantalla, desde tablets, dispositivos móviles y desde un navegador de escritorio. Por lo tanto y debido a la capacidad de adaptarse al tipo de dispositivo consideramos que react-native es la mejor decisión.

Bases de datos:

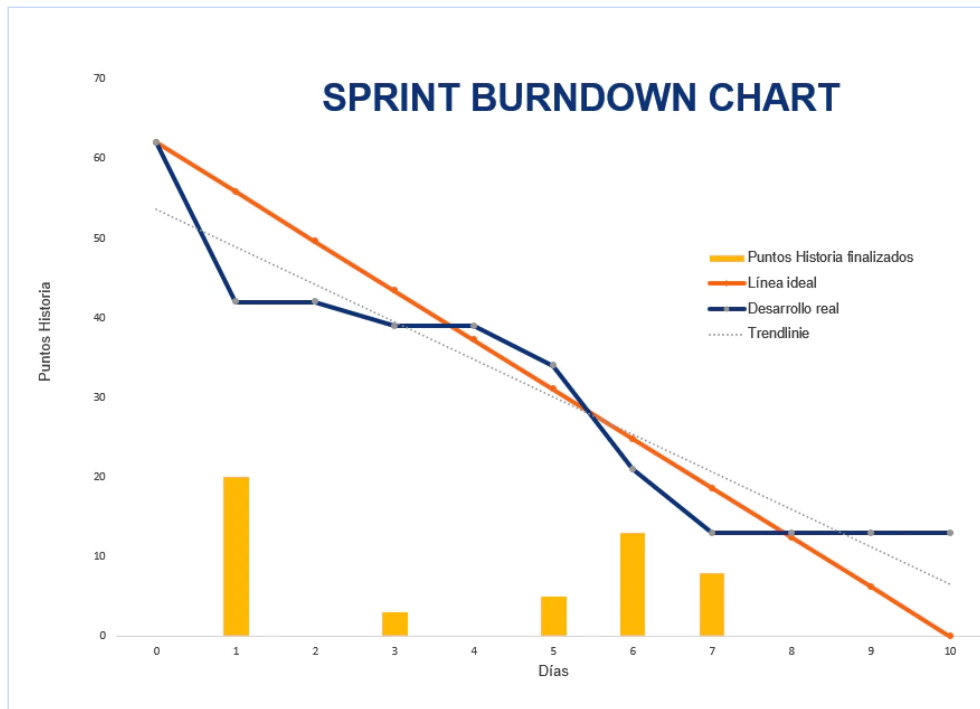
- **MongoDB:** Base de datos no relacional, que permitiría almacenar datos de manera flexible y consultarlos de manera sencilla. El principal objetivo del uso de bases de datos no estructuradas es la capacidad de explotación de datos que no involucran transacciones de negocio, como es el caso de esta aplicación.
- **SQL Server / Oracle / MySQL:** Base de datos relacional, orientada a almacenar datos generales de actividades, inicio de sesión, seguimiento, entre otras. Esta tecnología es la más conocida por el equipo y sería, por lo tanto, sencilla de utilizar y aplicar para el almacenamiento de datos.

Métricas del proyecto

Burndown Chart

Los equipos ágiles que siguen las pautas de SCRUM definen una serie de sprints con un tiempo asignado de compleción, al inicio de cada sprint el equipo define un pronóstico de cuánto trabajo puede completar en el tiempo del mismo.

Una de las principales métricas utilizadas en SCRUM (y en casi todos las metodologías ágiles), consiste en poder visualizar de manera gráfica la cantidad de trabajo pendiente durante la ejecución de un sprint. Esta métrica es conocida como Burndown Chart.



De esta forma, en el eje de las abscisas se muestra el tiempo que pasa (usualmente, en días), mientras que en el eje de las ordenadas se muestra la cantidad de esfuerzo pendiente en el sprint, el cual se representa en los story points restantes del mismo.

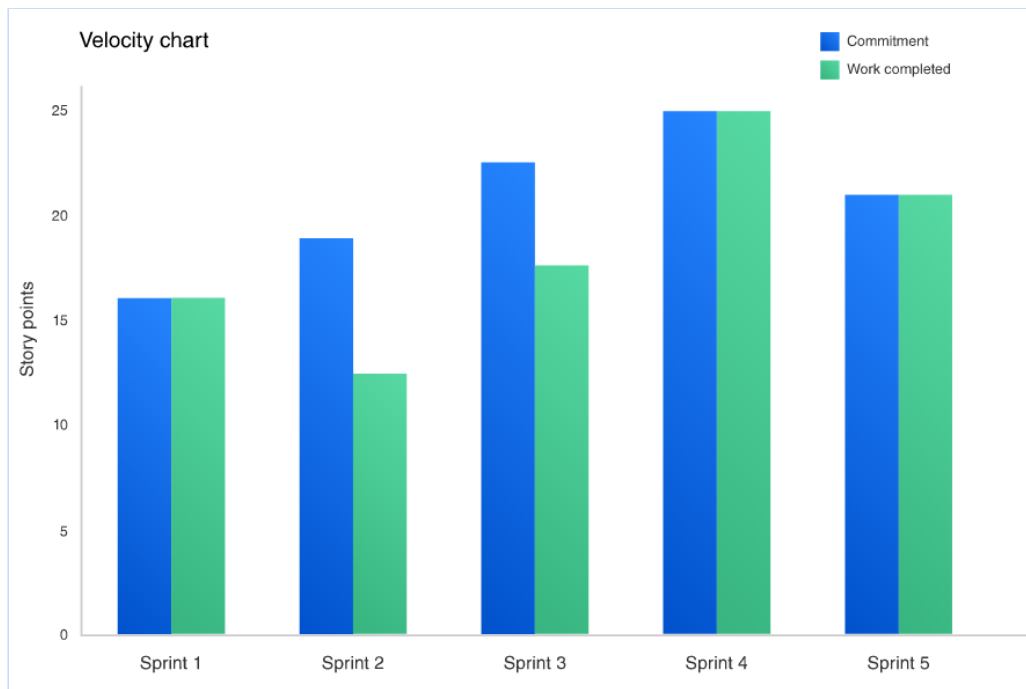
De todas formas, aparte de servir como un indicador del trabajo restante, y como herramienta para definir la cantidad de trabajo que el equipo puede tomar, es muy importante tener en cuenta los siguientes patrones posibles:

- Si el equipo finaliza todas las tareas del sprint antes de tiempo esto indica que no se asume el trabajo suficiente en el sprint. Y al revés ocurre de forma similar (no se cumplen las previsiones del sprint si se asume demasiado trabajo).
- Si la línea de trabajo finalizado tiene caídas muy profundas es un indicador de que el trabajo no fue dividido de forma granular (User Stories que no se dividieron en otras).
- Se debe tener cuidado cuando se modifica el alcance del sprint debido a cambios en las decisiones del Product Owner.

Por esta razón, se selecciona la Burndown Chart como métrica para el proyecto, dado que permite tener un panorama completo del desarrollo actual, que a su vez provee de indicadores útiles tanto durante como luego del sprint.

Velocidad

En resumen, la velocidad es la cantidad promedio de trabajo que un equipo lleva a cabo durante un sprint, medida en story points u horas. Es una medida útil para hacer pronósticos para sprints futuros.



Además, esta métrica para supervisar el trabajo previsto en contraste con el completado. De esta forma, el PO podrá definir un Backlog apropiado para la velocidad del equipo. Por otra parte, se podrá observar la evolución de las relaciones entre el equipo, u observar posibles conflictos durante el sprint. Un descenso de la velocidad media generalmente es un indicador de que alguna parte del desarrollo del equipo se ha vuelto ineficiente y debería tratarse en la siguiente retrospectiva.

Así como el Burndown Chart, la velocidad tiene ciertas posibles alertas y advertencias:

- Si la velocidad del equipo presenta variaciones en un largo periodo de tiempo, debemos revisar las prácticas de estimación:
 - Puede implicar algunas dificultades que no se tuvieron en cuenta durante el análisis.
 - Puede haber presión externa sobre el grupo.

Dada la naturaleza del proyecto, es muy probable que la velocidad sea un indicador muy eficaz para denotar aspectos de dificultades durante el desarrollo de la aplicación. La falta de experiencia en el desarrollo real del equipo, o inconvenientes en el aprendizaje, etc.. Por esta razón, se selecciona a la velocidad como métrica del proyecto.

Pautas de codificación y testing

Pautas de Codificación

Se definirán las pautas para el desarrollo de la aplicación, consistiendo estas de los siguientes puntos:

- **Comentarios en Bloque:** Los bloques de comentarios en el código denotan definiciones de archivos, métodos y aplicaciones, para dar una pauta inicial al desarrollador que lee el código, sobre el propósito de cada componente. Se los aplicará para definir la funcionalidad que las diferentes clases y sus respectivos métodos aplicarán.
- **Convenciones de código básicas:** A lo largo del tiempo, respetar los estándares básicos permiten contar con un código más limpio, legible y flexible para modificaciones.
 - **Respetar Niveles de Indentación:** Hace al código más legible.
 - **No repetir bloques de código:** Ayuda a volver el código más reutilizable.
 - **Evitar anidamiento de condiciones o de iteraciones:** Al estructurar los condicionales, se evitan líneas con demasiada indentación y por lo tanto, se obtiene un código más legible y simple de interpretar.
 - **Limitar el ancho de línea de código:** Limitar el ancho de línea cuando se ejecutan consultas SQL, escriben direcciones URL o de archivos, entre otros, permite contar con un código más fácil de leer.
 - **No utilizar Números Mágicos:** Codificar siempre utilizando variables que representen números constantes, ayuda a la reutilización del código, en caso de que cambie el contexto.
 - **Código en inglés:** para mantener el desarrollo del código estandarizado en un lenguaje común, de manera internacional. El proyecto será de código abierto para permitir desarrollos futuros, y código en inglés permite a desarrolladores externos entender el código independientemente del público objetivo del PO.
 - **Variables identificatorias:** Se definirán nombres de variables (y de funciones, métodos y nombres de componentes) según la funcionalidad o propósito que cumplan. De esta forma, se logrará un código más comprensible para nuevos desarrolladores.
- **Granularidad del código:** Considerando los estándares de la cantidad de código que se define como “demasiado” para encapsular las funcionalidades de cada componente del sistema, se tomó la decisión, para tener una mayor cohesión en la estructuración del código, que un método, función o funcionalidad individual no debe contener más de 40 líneas de código por si sola. Esta decisión dependerá de la complejidad del método que se esté desarrollando.

Pautas para Testing

Las pautas y buenas prácticas a tener en cuenta durante el desarrollo (y las pruebas) del proyecto serán las siguientes:

- **Pautas básicas de testing:** El plan de prueba debe ser ejecutado en cada sprint de manera no ambigua, con las siguientes características. Todas estas prácticas básicas permiten contar con un plan de pruebas sólido, que cumpla con los objetivos de calidad solicitados:
 - Los casos de prueba deben tener un objetivo claro y conciso, para poder ser interpretado en pruebas futuras.
 - Los casos de prueba deben abarcar la mayor cobertura de pruebas posibles, así como intentar encontrar la mayor cantidad de errores. Con esto, se reducirá la cantidad de pruebas a realizar en general.
 - Los casos de prueba deben ser autosuficientes, es decir, contar con todos los detalles necesarios respecto a las condiciones del sistema para poder ejecutarlos.
- **Orientar el testing a los usuarios:** Se debe proveer un esquema de testing que esté orientado a priorizar la ejecución de los casos de prueba de requerimientos que el usuario más utilice, de esta forma, se provee una aplicación de calidad y amigable para los usuarios, en las partes clave de la aplicación. En este apartado, se incluye también dedicar el debido tiempo en los casos de prueba asociados al rendimiento del sistema, y a su vez los referidos a la seguridad informática, ya que un software con mal rendimiento puede ser tan poco útil para un usuario como uno poco funcional.
- **Automatizar lo más posible:** Para aquellos casos de prueba que no vayan a ser objeto de muchos cambios a futuro, se contará con un esquema de automatización de testing dedicado a hacer más eficiente el despliegue de nuevas versiones. De esta forma, se puede hacer más eficiente el ciclo de ejecución de pruebas antes de actualizar una versión de la aplicación.
- **Pruebas de regresión:** Ejecutar al menos un ciclo de regresión para los casos de prueba de la aplicación antes de desplegar una versión estable del producto. El testing de regresión permite encontrar errores que quizás se saltaron durante la fase de pruebas de features, y resulta una herramienta muy útil antes de desplegar un producto en una versión estable.

Gestión de Configuración

Para mantener una estructuración generalizada y sostenible del código desarrollado, así como la documentación del mismo, y llevar un seguimiento riguroso y confiable de las versiones que se desarrollan del proyecto, se debe contar con una herramienta de versionado y configuración de software, para lo cual emplearemos dos herramientas estrechamente relacionadas entre sí.

La primera, Git, es un sistema de control de versiones de código abierto orientado a manejar el código de grandes proyectos de software, así como de pequeños grupos de desarrolladores.

En conjunto se utilizará la plataforma Gitlab, que complementará el servicio brindado por Git. Gitlab es un servicio web de control de versiones y desarrollo de software colaborativo basado en Git. Además de gestor de repositorios, el servicio ofrece también alojamiento de wikis y un sistema de seguimiento de errores, todo ello publicado bajo una Licencia de código abierto.

Organización de la Documentación

- Repositorio
 - 1 - Documentación del Proyecto
 - 1.1 - Documentación
 - 1.1.1 - Estudio Inicial
 - 1.1.2 - Plan de Proyecto
 - 1.1.3 - Gestión de Configuración
 - 1.1.4 - Plan de Pruebas
 - 1.1.5 - Plan de Calidad
 - 1.2 - Gestión de Proyecto
 - 1.2.1 - Product Backlog
 - 1.2.2 - Métricas
 - 1.2.3 - Sprint
 - Sprint 0
 - Sprint 1
 - Sprint N
 - 2 - Documentación del Producto
 - 2.1 - Especificación de Producto
 - 2.1.1 - Especificación de US
 - 2.1.2 - Especificación de Casos de Prueba
 - 2.1.3 - Definición de modelos
 - 2.1.3.1 - Definición de arquitectura de Software
 - 2.1.3.2 - Definición de modelo de datos
 - 2.1.3.3 - Definición del diagrama de clases
 - 2.3 - Código Fuente
 - 2.4 - Licencias
 - 2.5 - Manuales de Usuario

Ítems de Configuración

Ítem de Configuración	Regla de Nombrado	Ubicación Física	Tipo de Ítem
Estudio Inicial	Estudio_Inicial_<Version>	Repositorio/Documentación del Proyecto/Estudio Inicial/Entregas	Proyecto
Plan de Proyecto	Plan_De_Proyecto_<Version>	Repositorio/Documentacion del Proyecto/Plan De Proyecto/Entregas	Proyecto
Gestión de Configuración	SCM_<Version>	Repositorio/Documentacion del Proyecto/Gestión de Configuración/Entregas	Proyecto
Plan de Pruebas	Plan_De_Pruebas_<Version>	Repositorio/Documentacion del Proyecto/Plan De Pruebas/Entregas	Proyecto
Plan de Calidad	Plan_De_Calidad_<Version>	Repositorio/Documentacion del Proyecto/Plan_de_Proyecto	Proyecto
Daily Meeting	Daily_Meeting_<dd_mm_aa>	Repositorio/Documentacion del Proyecto/Sprint/Sprint_<N>/Dailies	Proyecto
Conclusión Sprint	Conclusion_Sprint_<Nro_Sprint>	Repositorio/Documentacion del Proyecto/Sprint<Nro_Sprint>	Proyecto
Product Backlog	Product_Backlog_<Version>	Repositorio/Documentacion del Proyecto/Gestión de Proyecto/Entregas	Proyecto
Métricas	Nombre_Metrica_<Nro_Sprint>	Repositorio/Documentacion del Proyecto/Gestión de Proyecto/Metricas	Proyecto
Diagrama de Clases de Análisis	US_<Nro_US>_Diagrama_de_Clas es_<Version>	Repositorio/Documentacion del Producto/Especificación de producto/Especificación de US	Producto
Diagrama de Secuencia	US_<Nro_US>_Diagrama_de_Sec uencia_<Version>	Repositorio/Documentacion del Producto/Especificación de producto/Especificación de US	Producto
Diagrama de Comunicación	US_<Nro_US>_Diagrama_de_Co municacion_<Version>	Repositorio/Documentacion del Producto/Especificación de producto/Especificación de US	Producto
Diagrama de clases de dominio	Diagrama_clases_dominio_<Versio n>	Repositorio/Documentacion del Producto/Especificación de producto/Definición de modelos/Diagrama de clases	Producto

Modelo de Arquitectura	Modelo_Arquitectura_<Version>	Repositorio/Documentacion del Producto/Especificación de producto/Definición de modelos/Arquitectura de Software	Producto
Modelo de Datos	DER_<Version>	Repositorio/Documentacion del Producto/Especificación de producto/Definición de modelos/Modelo de Datos	Producto
Casos de Prueba	Caso de Prueba_<Nro_Test_Case>	Repositorio/Documentacion del Producto/Especificación de producto/Especificación de Casos de Prueba	Producto
Código Fuente	<Nombre_Proyecto>_<Versión>	Repositorio/Documentacion del Producto/Código Fuente	Producto

Glosario de Nombrado

Expresión	Significado
<Version>	Versión del documento. Formato: ##.##, donde el primer número hace referencia a cambios mayores, el segundo a cambios menores.
<Nro_Sprint>	Es un número que corresponde al número de Sprint
<Nro_US>	Es un número que corresponde al número de la User Story
<Nro_Test_Case>	Número que identifica al número de caso de prueba

Definición de Línea Base

Una Línea Base es una porción de producto o de documentación, la cual ha sido revisada formalmente por el equipo y determinada base para el desarrollo posterior. La Línea Base debe ser actualizada cada un cierto período para poder continuar con el desarrollo futuro del proyecto.

La línea base **será actualizada al finalizar cada Sprint**, habiendo sido revisados todos los cambios en cada uno de los ítems de configuración por parte de todos los integrantes del Comité de Control de Cambios, el cual se detallará más adelante.