a



METODOLOGÍA DE TRABAJO

PROYECTO FINAL

Alumnos:

Bottino, Natanael

Cena, Henry

Pedrotti, Gonzalo

Zanin, Federico

Profesores:

Villafañe, Christian

Cassani, Matías

AÑO DE CURSADO: 2019

Contenido

[Historial de Revisiones: 3](#_Toc8677275)

[Introducción y Justificación: 4](#_Toc8677276)

[Organización del Equipo: 4](#_Toc8677277)

[Integrantes del Equipo de Proyecto Final: 4](#_Toc8677278)

[Definición del lugar de trabajo: 4](#_Toc8677279)

[Días y Horario de Reunión: 5](#_Toc8677280)

[Recursos con los que cuenta el equipo: 5](#_Toc8677281)

[Framework para gestionar el Proyecto: 5](#_Toc8677282)

[Ceremonias a ejecutar: 5](#_Toc8677283)

[Entregables en cada Sprint: 6](#_Toc8677284)

[Roles del Equipo SCRUM: 8](#_Toc8677285)

[Historias de Usuario 8](#_Toc8677286)

[Estimación de las Historias de Usuario 8](#_Toc8677287)

[Criterios de Aceptación (Definition of Ready) 9](#_Toc8677288)

[Criterios de Hecho (Definition of Done) 9](#_Toc8677289)

[Metodología de Trabajo del Sprint 10](#_Toc8677290)

[Flujo de Trabajo en los Sprint 10](#_Toc8677291)

[Planificación del Sprint 11](#_Toc8677292)

[a) Definición/Objetivo del Sprint 11](#_Toc8677293)

[b) Sprint Backlog (Lista de US del Sprint) 11](#_Toc8677294)

[c) Estimación de la Capacidad del Equipo: 12](#_Toc8677295)

d) [Sprint Burndown Chart 13](#_Toc8677296)

[e) Velocidad del Equipo (Puntos Estimados vs Puntos Quemados) 13](#_Toc8677297)

[Gestión del Sprint Backlog 14](#_Toc8677300)

[Definición del Sprint 0 15](#_Toc8677301)

[Gestión de Configuración: Cambios y Versiones 16](#_Toc8677302)

[Estructura del Repositorio: 16](#_Toc8677303)

[Control de versiones de la Documentación en la carpeta “Gestión del Proyecto” 16](#_Toc8677304)

[Control de versiones de la Documentación en la carpeta “Gestión del Producto” 16](#_Toc8677305)

[Bibliografía 17](#_Toc8677306)

# Historial de Revisiones:

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 18/04/2019 | 1.0 | Definición de la Metodología de Trabajo | Equipo de Desarrollo |
| 23/04/2019 | 1.1 | Se incorporó la planificación del sprint y se modificó la gestión del Sprint Backlog | Gonzalo Pedrotti |
| 06/05/2019 | 1.2 | Revisión | Christian Villafañe |
| 08/05/2019 | 1.3 | Se corrigieron errores, se incorporó Sprint 0 y se agregó información en Gestión de Configuración: Cambios y Versiones | Equipo de Desarrollo |

**Definición de Metodología de Trabajo**

# Introducción y Justificación:

Antes de comenzar con el desarrollo del producto, es necesario definir el marco de trabajo que será utilizado para gestionar dicho desarrollo.

Al momento de elegir la metodología de trabajo, el equipo de desarrollo tenía dos opciones de acuerdo a los conocimientos adquiridos a lo largo del cursado de la carrera. Por un lado, elegir un marco de trabajo tradicional, como el Proceso Unificado de Desarrollo o bien, elegir una metodología ágil, como Scrum.

Debido a que el equipo de desarrollo no tiene la suficiente experiencia en la gestión de proyectos y en el desarrollo de sistemas, se ha optado por elegir una metodología ágil, ya que este tipo de marco es más adecuado cuando en el entorno existe una cierta incertidumbre o el mismo es cambiante. Al mismo tiempo, consideramos que la comunicación entre los miembros del equipo es fundamental y cada uno es un valor sumamente importante para que el proyecto se termine de forma correcta.

Si bien existe un gran abanico de metodologías agiles, se considera a Scrum como la más popular dentro de esas opciones y existe mucha documentación sobre cómo llevar a cabo la misma, por lo tanto, el equipo de desarrollo ha elegido como marco de trabajo, la metodología ágil SCRUM.

Otras de las cuestiones por las que se eligió SCRUM, es que permite al equipo auto organizarse, permite entregar resultados en tiempos o iteraciones cortas, y, sobre todo, que el progreso se puede determinar en pocas semanas, de manera que es posible conocer en plazos cortos si se está trabajando correctamente y el proyecto está avanzando hacia los objetivos correctos.



# Organización del Equipo:

## 

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Definición del lugar de trabajo:

No se tiene un lugar predefinido para llevar a cabo las reuniones del equipo. Dentro de estos lugares se encuentran la FRVM y los distintos departamentos de los integrantes del equipo.

El lugar de reunión del equipo es definido horas antes de llevar a cabo dicha reunión.

## Días y Horario de Reunión:

El equipo define a través de la capacidad del mismo, qué días y horarios serán destinados a trabajar en el proyecto. (Esto se define más en detalle en la planificación de la metodología)

## Recursos con los que cuenta el equipo:

Cada integrante del equipo cuenta con una notebook para realizar el trabajo.

# Framework para gestionar el Proyecto:

Para la gestión del proyecto se utilizará el marco de trabajo SCRUM. Esta metodología requiere de la definición de una iteración de duración fija llamada Sprint. El equipo de desarrollo ha definido como duración del mismo 14 días.

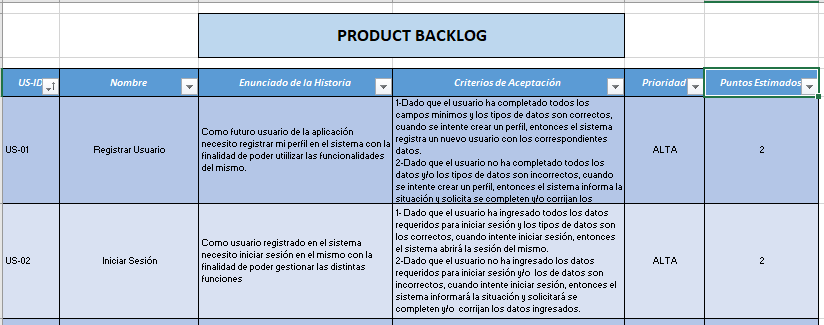
Cada Sprint cuenta con un conjunto de ceremonias llevadas a cabo y artefactos utilizados y generados.

## Ceremonias a ejecutar:

* Story Time: Durante esta ceremonia, el equipo de desarrollo construye el Product Backlog (PB). Esta ceremonia puede realizarse durante el Sprint, después de la Daily Scrum o durante la Sprint Review. Se define el alcance del producto a partir de objetivos expresados como historias de Usuario (User Story).

El Product Backlog es una lista priorizada y estimada de historias. El equipo estima las historias a través del planning Poker para obtener una medida de tamaño relativo llamado Story Point.

Luego de establecer el Story Point, se le asigna una prioridad (valor de negocio para el cliente, la misma puede ser ALTA, MEDIA o BAJA) a cada historia de usuario, y de esa manera, será armado el Product Backlog, con las US de mayor prioridad por encima de aquellas con menor prioridad y dependiendo si existen US dependientes unas de otras.

A continuación, se muestra la estructura que será utilizada para la construcción del Product Backlog:

* Sprint Planning (SP): En esta reunión se define la funcionalidad en el incremento planeado y cómo el equipo de Desarrollo creará este incremento. La salida de este trabajo es definir el Objetivo del Sprint. El producto Owner debe participar de estas reuniones. Durante esta ceremonia, el equipo de trabajo selecciona un conjunto de User Story del Product Backlog que ya se hayan estimado, las cuales serán incluidas en el Sprint Backlog (serie de Historias de usuario que se van a llevar a cabo durante la ejecución del Sprint).

Además, durante esta ceremonia el equipo de desarrollo estima la capacidad del equipo para el sprint que está comenzando.

* Daily Scrum: En esta reunión, el equipo de trabajo comenta lo que cada integrante ha realizado, los problemas que tuvo, y lo que se ha pensado hacer en ese día. Con esta reunión, vamos a poder evaluar el progreso hacia la meta del Sprint.
* Sprint Review: se lleva a cabo al final del Sprint, para inspeccionar el incremento y adaptar el producto que se está construyendo. Se discute lo que funciona, los problemas que hubo y cómo han sido resueltos.
* Sprint Restrospective: Se crea un plan de mejoras para ejecutar durante el siguiente Sprint. Se inspecciona qué es lo que está funcionando bien y que no con el proceso implementado.

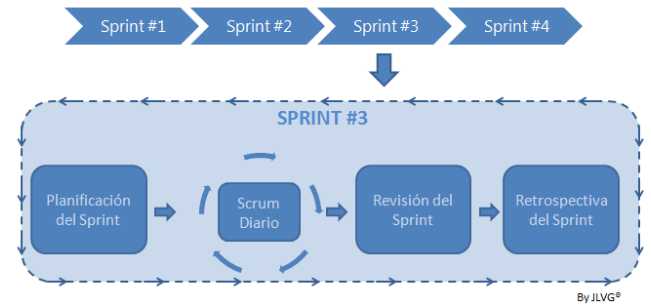


Ilustración 1-Estructura del Sprint

## Entregables en cada Sprint:

El equipo de desarrollo ha decidido incluir en un único documento una lista de puntos que indican como será organizado cada Sprint y lo que se incluye como resultado del mismo.

Este documento será generado al comienzo de cada Sprint y se irán incorporando y actualizando cada uno de los puntos requeridos, de manera que, al finalizar el Sprint, este documento cuente con todos los ítems predefinidos.

A continuación, se detallan los puntos o apartados a incluir en el documento generado en cada Sprint:

Objetivo y Alcance de la Iteración: En este apartado se incluye el estado actual del Product Backlog (PB), es decir la lista de historias de usuarios que se encuentran definidas, estimadas y priorizadas en ese momento. Se incluye, además, el objetivo y definición del Sprint, el grupo de historias de usuarios que serán llevadas a cabo en el Sprint (Sprint Backlog), y la fecha de inicio y fin de dicho Sprint. (Este apartado se define más adelante en los puntos a) y b) de la Planificación del Sprint)

Cronograma y Calendarización de la Iteración: El cronograma y la calendarización define los días y cantidad de horas que cada integrante del equipo le designará al desarrollo del producto durante ese Sprint. (Este apartado se define más en detalle en el punto c) de la Planificación del Sprint).

Sprint Burndown Chart y Velocidad del Equipo: este apartado muestra como fue la dedicación en horas a cada historia de usuario por parte de los integrantes del equipo teniendo en cuenta las horas estimadas y las horas reales de trabajo. Y, por otro lado, se muestra cómo se fueron quemando los puntos a lo largo del Sprint y los Sprints anteriores, dando como métrica la velocidad con la que el equipo está avanzando con el desarrollo del producto. (Este apartado se define más en detalle en los puntos d) y e) de la Planificación del Sprint).

Definición de procesos y/o procedimientos de ayuda para comprender el dominio:

Para aquellas historias de usuario o porciones de funcionalidad en la que se requiera tener una mejor comprensión del dominio y sea necesario contar con una descripción más detallada de dichas funciones, el equipo de desarrollo utilizará descripciones de Casos de Uso en formato de Plantilla de Trazo fino con su prototipo asociado.

Minutas de reuniones de equipo: En cada reunión, el equipo de trabajo comenta lo que cada integrante ha realizado, los problemas que tuvo, y lo que se ha pensado hacer en ese día. Con esta reunión, vamos a poder evaluar el progreso hacia la meta del Sprint.

Documento de Análisis: el equipo de desarrollo implementará como herramienta de análisis un diagrama de clases, el cual se irá actualizando en cada nueva versión del documento, también se utilizará un diagrama de estados, para aquellas clases que contengan muchos estados y sea necesario manejarlos en la aplicación.

Documentación de diseño de iteración: En este apartado, el equipo de desarrollo implementará patrones de diseño y/o patrones arquitectónicos de diseño con subsistemas e interfaces.

Documentación de testeo y pruebas de la iteración: en esta sección, se incluyen qué pruebas se realizaron, los subsistemas probados, errores detectados, cambios sugeridos, etc.

Documentación de implementación y despliegue: el equipo de desarrollo utilizará un diagrama que muestre la Vista arquitectónica de Despliegue, la cual incluya los nodos y componentes de hardware.

Documentación de ayuda para el usuario: Al final de cada iteración, el equipo de desarrollo actualizará el manual de usuario, el cual está destinado a dar asistencia a las personas que utilizan el sistema.

Documentación de aceptación y cierre de la iteración: se describen conclusiones del equipo al final de cada iteración.

## Roles del Equipo SCRUM:

Product Owner (PO): Representante del cliente que utiliza el software. Formaliza las funcionalidades en historias de usuario a incorporar en el Product Backlog y les asigna prioridades. Se cuenta con un conjunto de clientes/expertos en tema.

Scrum Master: Persona que lidera al equipo guiándolo para que se cumplan las reglas y procesos de la metodología. Trabaja con el Product Owner y es parte del equipo.

Equipo de desarrollo: Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historiasa las que se comprometen al inicio de cada sprint.

|  |  |
| --- | --- |
| Miembros del Equipo | Rol |
| Pedrotti, Luis | Product Owner |
| Pedrotti, Gonzalo | Scrum Master |
| Cena, Henry | Miembro del Equipo de Desarrollo |
| Bottino, Natanael | Miembro del Equipo de Desarrollo |
| Zanin, Federico | Miembro del Equipo de Desarrollo |

## Historias de Usuario

Descripción corta de una funcionalidad valuada por un usuario o cliente de un sistema.

Una historia estará compuesta por una tarjeta que incluye un identificador de historia, un nombre, el objetivo/requisito expresado en forma de historia de Usuario (Por ejemplo: Como agricultor quiero registrar un lote con el objetivo de poder gestionar actividades a realizar en el mismo). Cuenta también con un valor o prioridad para el Product Owner, un valor de esfuerzo estimado por el equipo de desarrollo y un conjunto de condiciones o criterios de aceptación redactado también en una frase.

Cada historia de usuario debe ser estimada antes de ser tenida en cuenta para llevarla a cabo en un Sprint, ya que, si no, no se conoce a ciencia cierta su duración y se corre el riesgo de no poder terminarla en un determinado Sprint.

## Estimación de las Historias de Usuario

La unidad de peso, para el equipo, será el Story Point (SP). Éste es un valor abstracto que determinará el nivel de complejidad, esfuerzo e incertidumbre que representa para el equipo la US estimada. El SP indica la idea del “peso” de cada US y determina cuán grande es.

Nuestra herramienta de estimación es Planning Poker. Utilizaremos la serie de Fibonacci para determinar el valor de los SP. En esta serie la secuencia comienza en 1 y cada número subsecuente es la suma de los dos precedentes (1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, …).

Nuestro rango de valores para utilizar Planning Poker con historias de usuario será:

* 1: funcionalidad pequeña. Usualmente se trata de algo más bien cosmético.
* 2-3: funcionalidad pequeña o mediana. Es lo que queremos.
* 5: funcionalidad media. Es lo que queremos.
* 8: funcionalidad grande. Se puede lograr, pero es mejor dividir en dos o más historias.
* 13: Probablemente nos encontremos frente a una Épica, la cual debería dividirse.
* Deberían indicar algún valor o símbolo que utilizarán para identificar a las historias que no se han estimado aún y que necesita revisión.

## Criterios de Aceptación (Definition of Ready)

Los criterios de aceptación definen los requisitos del Product Owner sobre cómo debe comportarse la aplicación para que una determinada acción se pueda llevar a cabo, es decir, lo que tiene que estar listo para que una historia de usuario pueda entrar a un Sprint.

Estos criterios están relacionados con las pruebas que se realizaran para verificar el cumplimiento de la expectativa de diseño, usabilidad, rendimiento, y la satisfacción del usuario y será definida durante el Story Time para cada historia de usuario en particular.

Criterios de aceptación definidos por el equipo:

* Historias de usuario bien definidas y que posean al menos 4 criterios de aceptación.
* Historias de usuario que posean un tamaño adecuado, es decir, que permitan ser desarrolladas y terminadas dentro de un Sprint.
* Que, si el equipo se encuentra con historias de usuario demasiado complejas, pueda realizar una investigación más profunda (Spike) con el fin de interiorizarse en cuestiones de diseño, tanto arquitectónico como tecnológico que se necesitan para desarrollar tal historia.
* Que la historia de usuario que esté en desarrollo se encuentre alineada con el/los objetivo/s del Sprint, que sean claramente visibles y demostrables.
* Que esté estimada por el equipo.
* Que se pueda testear.

Aclaración: cualquier historia de usuario sobre la que se tenga duda, no será incluida en el siguiente Sprint.

## Criterios de Hecho (Definition of Done)

Los criterios de hecho responden a qué tiene que cumplir la historia de usuario para que sea considerada Hecha o Terminada. Es responsabilidad del equipo que una historia sea marcada como Hecha.

Se utiliza un checklist con los criterios acordados por el equipo para cambiar el estado de una historia de usuario al estado de “Terminada” o “Hecha”.

El Criterio de Hecho utilizado por el equipo será el siguiente:

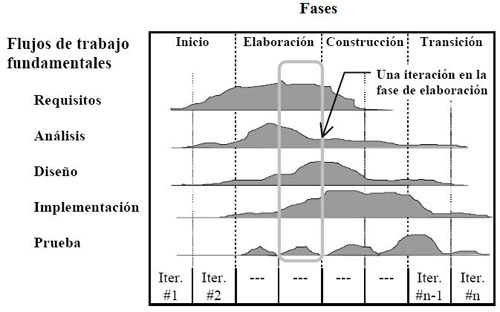
* La historia ha sido analizada y/o diseñado
* El código ha sido completado:
  + Código refactorizado
  + Código con formato estándar
  + Código comentado
  + Código en el repositorio
  + Código inspeccionado
* El análisis, diseño e implementación han sido probados
* Documentación de usuario revisado y actualizado

# Metodología de Trabajo del Sprint

## Flujo de Trabajo en los Sprint

La metodología ágil que utilizará el equipo será SCRUM, y en cada Sprint será llevado a cabo el Proceso Unificado de Desarrollo (PUD), por lo que los flujos de trabajo del PUD, en vez de realizarse en iteraciones, se llevarán a cabo en diferentes Sprint, donde no necesariamente se lleven a cabo todos, sino los que sean necesarios.

Para ver más gráficamente esto, utilizamos el siguiente esquema que define la estructura del Proceso Unificado de Desarrollo, con los correspondientes flujos de trabajo y fases:



En nuestro caso particular, cada iteración del PUD, nos representa un Sprint de SCRUM.

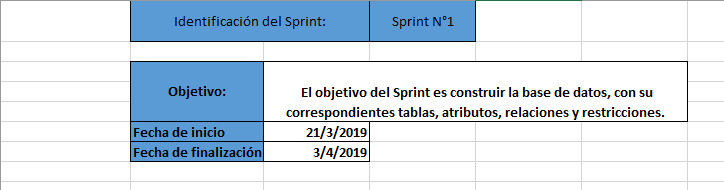
## Planificación del Sprint

La planificación del Sprint cuenta con un conjunto de datos y herramientas utilizadas para llevar el control del mismo y planificar diversos aspectos importantes. Esta planificación será llevada a cabo utilizando un documento a través de la herramienta Excel, el cual contiene cuatro apartados, y pertenecen al mismo documento generado al final del Sprint llamado “Planificación del Sprint”.

Un quinto apartado será incluido en un documento de Excel externo a los apartados anteriores, el cual se irá actualizando con el tiempo conforme se lleven a cabo los sprints y es denominado “Velocidad del Equipo”.

### Definición/Objetivo del Sprint

Al comienzo del Sprint, se define durante el Sprint Planning la identificación del mismo, cuál es el objetivo de dicho Sprint y su fecha de inicio y fin.



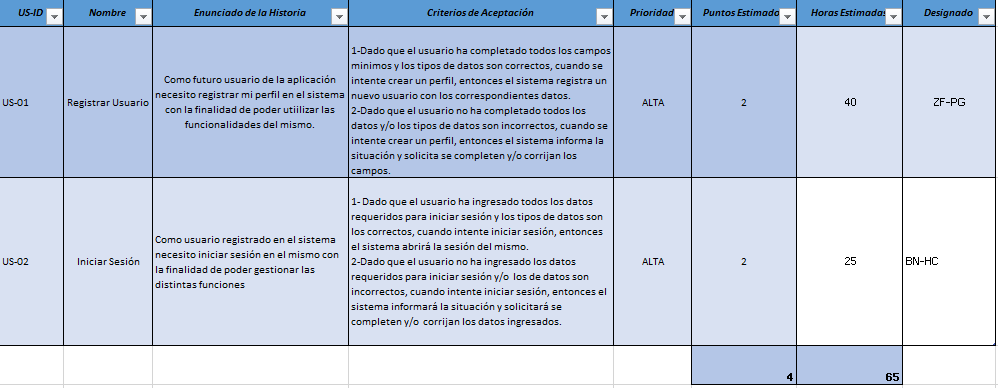
### Sprint Backlog (Lista de US del Sprint)

El Sprint Backlog es una serie de historias de usuario que ya se encuentran priorizadas y estimadas seleccionadas del Product Backlog. El equipo lo elabora durante el Sprint Planning.

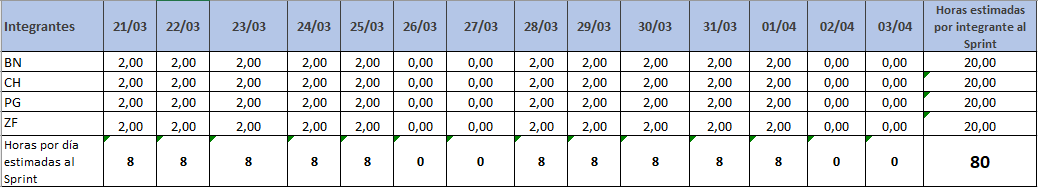
Para cada una de las historias de usuario seleccionadas para llevar a cabo en dicho Sprint, se muestran la identificación de la historia, el nombre de cada historia, el enunciado de la historia, los criterios de aceptación, la prioridad asignada, los puntos estimados (Story Points) para dicha historia, y se incluye, además, una estimación en horas y el responsable de dicha historia, es decir el integrante designado. Al encontrarse dentro de la planificación de un Sprint, es posible conocer en qué Sprint fue realizada cada una de las historias de usuario.

Las historias de usuario están ordenadas según la prioridad asignada por el cliente. Si una historia es dependiente de otra, se coloca por debajo de la que depende.

En la siguiente tabla se muestra un ejemplo de Sprint Backlog para el correspondiente Sprint:



### Estimación de la Capacidad del Equipo:

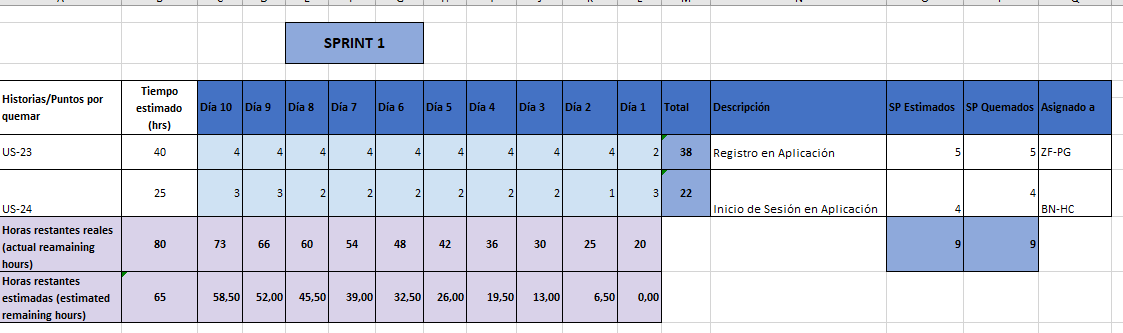
La estimación de la capacidad del equipo se define al comienzo del Sprint, en la cual se estima la cantidad de horas que cada integrante del equipo designará al desarrollo del producto durante ese Sprint. A modo de ejemplo, en la siguiente imagen se muestra cómo se realizará la estimación de la capacidad del equipo:

La tabla está formada por la columna izquierda con el nombre de cada integrante del equipo y las columnas siguientes representan a cada uno de los días correspondientes a la duración total del Sprint. En el ejemplo, el Sprint dura 14 días. Para cada uno de los integrantes en cada uno de los días se asigna la cantidad de horas que se estima que dicho integrante le dedicará al desarrollo del producto en dicho día.

La última columna, localizada a la derecha, muestra la cantidad de horas que le dedicará cada integrante al desarrollo del proyecto.

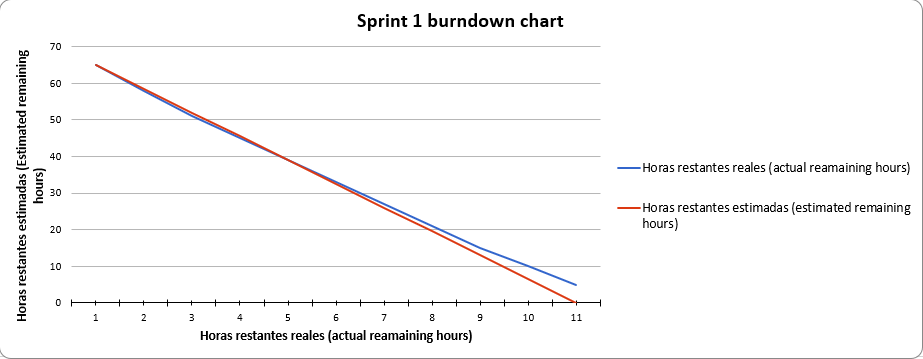
La fila inferior, muestra la sumatoria de horas que le dedicará el equipo completo por día al proyecto, y la sumatoria de estos valores se muestra en el recuadro inferior derecho, tal valor representa la cantidad de horas que el equipo completo le dedicará al proyecto.

### Sprint Burndown Chart

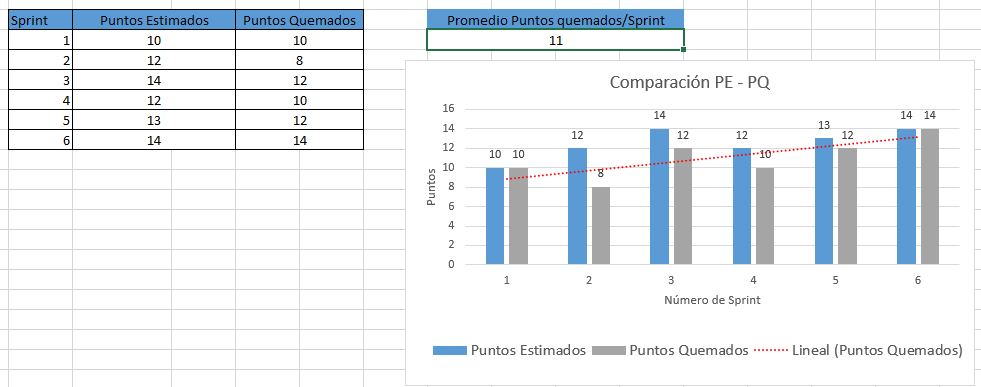
El cálculo de la velocidad del equipo es una medida que se obtiene al final de cada Sprint. Este valor muestra la velocidad a que está completando las historias de usuario planificadas y permite comprender si el trabajo se está realizando en el tiempo estimado, o se están haciendo estimaciones de tiempo erróneas (ya sea que se estima mucho o poco tiempo para terminar las historias de usuario). Se tienen en cuenta dos parámetros, la cantidad de puntos estimados a quemar en dicho Sprint (trabajo pendiente) y la cantidad de horas pendientes para completar el Sprint.

El gráfico anterior muestra para cada historia de usuario, el tiempo estimado anteriormente en el Sprint Backlog, la cantidad de horas dedicadas (horas reales) por día a cada una de las historias, y el total de horas dedicadas a cada historia de usuario durante el Sprint. Además, para cada historia se muestra la cantidad de puntos estimados y quemados en dicho Sprint, conjuntamente con el asignado a desarrollar dicha historia.

Las dos filas inferiores muestran las horas restantes, las reales y las estimadas correspondientemente. La última columna de las horas reales muestra el remanente de horas estimadas. Si ese valor es positivo, nos indica que se está estimando una mayor cantidad de horas de las que realmente se dedicaron al proyecto. Mientras más cercano a cero sea ese valor, mejor será la estimación realizada en la planificación del Sprint. Por otro lado, si ese valor es negativo, significa que estimamos menos horas de las que realmente demandó el proyecto. Por lo tanto, se estimaron erróneamente las horas dedicadas.

Esta comparación se puede visualizar a través de un gráfico que muestra ambas horas restantes:

### Velocidad del Equipo (Puntos Estimados vs Puntos Quemados)

En cuanto a los puntos estimados a quemar y los puntos quemados en el Sprint, también es posible realizar una comparativa que nos ayude a determinar si se están intentando quemar demasiados puntos por Sprint, o si la cantidad de puntos a quemar es insuficiente, quedando tiempo de sobra en cada Sprint. Para poder determinar esto, es necesario contar con un conjunto de Sprints ya realizados que nos permitan obtener valores con mayor certeza. En la siguiente imagen, se muestran la cantidad de puntos estimados y la cantidad de puntos quemados en cada Sprint ya ejecutado. A modo de ejemplo, se muestran un total de 6 Sprints, con sus correspondientes puntos estimados y quemados al final de cada uno junto a un gráfico comparando ambos valores.

La línea roja de puntos muestra cómo se mantiene la cantidad de puntos quemados a lo largo del proyecto. En el ejemplo, nos indica que la cantidad de puntos quemados asciende de Sprint en Sprint.

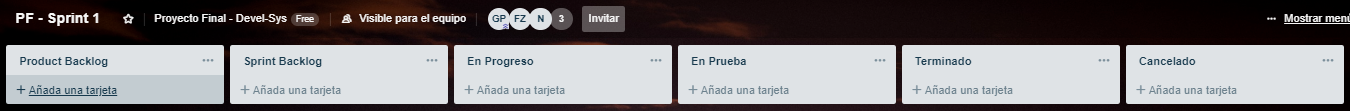
# Gestión del Sprint Backlog

El conjunto de actividades a realizar en cada iteración (Sprint) será gestionada a través del tablero proporcionado por la herramienta GitKraken Glo de GitKraken o Trello, las cuales nos permite definir tareas que deben llevarse a cabo en el Sprint actual y que están relacionadas a una historia de usuario específica.

En este punto, el equipo ha decidido trabajar con cinco listas que indican el estado en el que se puede encontrar una historia de usuario. Estas son: ProductBacklog, Sprint Backlog, En Progreso, En Prueba, Terminada y Cancelada.

* ProductBacklog: en esta lista ubicamos a todas las historias de usuario encontradas y estimadas hasta la fecha en que se va a dar comienzo a un Sprint. Es decir, el total de historias de usuario.
* Sprint Backlog: esta lista contendrá el conjunto de historias de usuario que se van a llevar a cabo en el Sprint actual. Es decir, tomando el ProductBacklog y la priorización para cada historia por parte del Product Owner, se determinarán aquellas historias a llevar a cabo en ese Sprint, pasándolas a esta lista.
* En Progreso: en este apartado se encuentran aquellas historias de usuario que están siendo tratadas por el equipo de desarrollo. Aquí es donde a cada historia de usuario se le realiza su correspondiente análisis, diseño e implementación.
* En Prueba: Luego de haber terminado con el análisis, diseño e implementación, la US para a ser aprobada por el equipo o por otro compañero.
* Terminada: son todas aquellas historias de usuario que han cumplido con las condiciones de hecho.
* Cancelada: Puede que se haya generado un inconveniente que impida continuar con el trabajo sobre esa historia. Se cancela para que esta pueda ser planificada nuevamente. Puede ser cancelada desde cualquier estado.

A modo de ejemplo, se muestra un tablero en la herramienta Trello con los estados en los que se puede encontrar una historia de usuario:



# Definición del Sprint 0

Antes de darle inicio a la gestión de un proyecto, es importante definir ciertos parámetros que permitan enfocar el mismo. En el Sprint 0, se permite reducir la incertidumbre respecto al enfoque tecnológico y también respecto al alcance del producto (en forma de Product Backlog), de manera que ya se pueda comenzar a trabajar con el enfoque ágil.

A partir de la definición de lo que se quiere implementar, en forma de Product Backlog, se tiene que decidir cómo se quiere implementar, esto es, en forma de arquitectura o infraestructura técnica, especificando lenguajes de programación, entornos de desarrollo, sistemas gestores de base de datos, versiones utilizadas, componentes, etc.

La decisión de incluir el Sprint 0, se basa principalmente en la idea de preparar y proporcionar un entorno mínimo que le permita al equipo utilizarlo para sentar la base y tomarlo como punto de partida para la correcta gestión del proyecto.

La descripción del Sprint 0 se encuentra definida como documento extra a este, llamado Sprint 0, y hace hincapié en el enfoque técnico o infraestructura tecnológica, ya que el Product Backlog fue definido con anterioridad en otro documento.

# Gestión de Configuración: Cambios y Versiones

El proceso de Gestión de Configuración de Software debe garantizar las condiciones para que los desarrolladores trabajen en equipo de manera eficiente. Como herramienta para darle apoyo a esta gestión, se ha optado por elegir GitKraken, la cual es una interfaz gráfica que opera sobre Git. La misma no permitirá crear los correspondientes repositorios, y realizar las distintas actividades que se requieran sobre los mismos.

## Estructura del Repositorio:

El equipo implementará en el repositorio dos carpetas, una que incluya la documentación relacionada con la Gestión del Proyecto y otra carpeta relacionada con la Gestión del Producto.

## Control de versiones de la Documentación en la carpeta “Gestión del Proyecto”

El contenido de cada documento creado bajo el control de versiones, será actualizado cuando el documento se modifique, es decir, no se crea una nueva versión del documento, sino que se va actualizando tras realizarle modificaciones. De esta manera, se mantienen las versiones en secuencia, las cuales se identifican mediante números de versiones crecientes, por lo que será posible ver los cambios que se fueron realizando en un documento bajo el control de versiones.

El versionado de los documentos se llevará a cabo utilizando el nombre del documento acompañado de dos números: X.Y

* El primero (X) será conocido como la versión mayor y nos indica la versión principal del documento. Este cambiará si se modifica la estructura básica del documento o se incorporen temas significativos.
* El segundo (Y) será conocido como la versión menor y nos indicará correcciones o actualizaciones de los que ya se encuentra desarrollado. Este cambiará cuando hagamos correcciones menores, como lo puede ser arreglar errores y/o agregar nuevas características que no modifican la estructura básica del documento.

Por ejemplo:

* Definición de la Metodología 1.0
* Definición de la Metodología 1.1

## Control de versiones de la Documentación en la carpeta “Gestión del Producto”

En cuanto a la Gestión del Producto, al finalizar cada Sprint se obtiene un documento que es el entregable correspondiente a ese Sprint y, por otro lado, se tiene la correspondiente versión del software (código fuente, ejecutables, ficheros, etc.).

Luego de investigar sobre el tema en distintos lugares/sitios, el equipo llegó a la conclusión de que no existe una regla o estándar oficial para normar las versiones del software, por lo que se ha llegado a un mutuo acuerdo de seguir la siguiente regla propia del equipo:

* El versionado del software se llevará a cabo utilizando el nombre del ítem acompañado de tres (3) números: X.Y.Z
* El primero (X) será conocido como la versión mayor y nos indica la versión principal de software. Este cambiará cuando agreguemos nuevas funcionalidades importantes, como pueden ser un nuevo módulo o característica clave para la funcionalidad.
* El segundo (Y) será conocido como la versión menor y nos indicará nuevas funcionalidades. Este cambiará cuando hagamos correcciones menores, como lo puede ser arreglar errores y/o agregar nueva funcionalidad que no son cruciales en el proyecto.
* Y el último (Z) será conocido como revisión y nos indicará que se hizo una revisión del código por algún fallo. Este será modificado cada vez que se haga entrega del proyecto luego de corregir errores menores en el código.

# Bibliografía

Ilustración1-Estructura del Sprint:

*Las 5 ceremonias de Scrum: clave para la gestión de procesos*. (13 de 05 de 2019). Obtenido de https://www2.deloitte.com/es/es/pages/technology/articles/ceremonias-scrum.html

Metodo, G. (13 de 05 de 2019). *¿Por qué utilizar Scrum en la Gestión de Proyectos*. Obtenido de http://blog.metodoconsultores.com/scrum-gestion-proyectos/

Plaza, M. (13 de 05 de 2019). *Management Plaza*. Obtenido de https://managementplaza.es/blog/desarrollar\_un\_proyecto\_con\_scrum/

proyectos?, ¿. c. (13 de 05 de 2019). *OBS Business School*. Obtenido de https://www.obs-edu.com/int/blog-project-management/temas-actuales-de-project-management/te-conviene-utilizar-la-metodologia-scrum-en-tus-proyectos

sequal. (13 de 05 de 2019). *¿Por qué utilizar SCRUM?* Obtenido de http://sequal.com.mx/contacto/347-ipor-que-utilizar-scrum.html

Villafañe, C. (2016). *Metodología para el Desarrollo del Producto.* Villa María.