



# Implementación de Scrum en Github

## 1. Introducción

### 1.1 ¿Que es Scrum? ¿Y como lo implementamos en Github?

Scrum es un marco de trabajo ágil para la gestión y desarrollo de proyectos. Se basa en la colaboración, la flexibilidad y la entrega incremental de resultados. A continuación, se presenta un resumen de los conceptos clave de Scrum:

#### *Roles:*

1. **Product Owner:** Es responsable de definir y priorizar el backlog del producto, representando los intereses de los stakeholders y asegurando el valor del producto.
2. **Scrum Master:** Es el facilitador del equipo Scrum, encargado de eliminar obstáculos, promover la adopción de prácticas ágiles y asegurar que se sigan los principios y reglas de Scrum.
3. **Equipo de Desarrollo:** Son los encargados de desarrollar el producto. Son autónomos y autoorganizados, colaborando estrechamente para entregar incrementos de valor en cada sprint.

#### *Artefactos:*

1. **Product Backlog:** Es una lista priorizada de las funcionalidades, requisitos y mejoras pendientes del producto. Es responsabilidad del Product Owner y se va refinando y actualizando constantemente.
2. **Sprint Backlog:** Es la selección de elementos del Product Backlog que el equipo se compromete a desarrollar durante un sprint. Es propiedad del equipo de desarrollo y se crea al inicio de cada sprint.
3. **Incremento:** Es la suma de todos los elementos del Product Backlog completados durante un sprint, que se entrega al final de cada iteración.

## Eventos:

1. **Sprint:** Es el corazón de Scrum. Es un período de tiempo fijo (generalmente de 1 a 4 semanas) en el cual se desarrolla un incremento de producto. Incluye la planificación del sprint, las reuniones diarias, el trabajo de desarrollo, la revisión del sprint y la retrospectiva del sprint.
2. **Reunión de Planificación del Sprint:** Al inicio del sprint, el equipo selecciona los elementos del Product Backlog que se compromete a completar durante el sprint y define cómo los entregará.
3. **Reunión Diaria:** Es una reunión breve (generalmente de 15 minutos) que se realiza diariamente durante el sprint. El equipo de desarrollo comparte el progreso, identifica obstáculos y coordina el trabajo futuro.
4. **Revisión del Sprint:** Al final del sprint, el equipo muestra el incremento completado y recibe retroalimentación de los stakeholders. Se revisa el backlog y se ajustan las prioridades y requisitos.
5. **Retrospectiva del Sprint:** Después de la revisión del sprint, el equipo reflexiona sobre su forma de trabajar y busca mejoras en su proceso. Se identifican acciones para aumentar la eficiencia y la calidad en futuros sprints.

## 1.2 ¿Que nos ofrece Github para implementar Scrum?

- Nos ofrece una **wiki** para escribir la documentación del proyecto, podemos escribir en la wiki utilizando el lenguaje Markdown.
- También nos ofrece los **hitos o milestones** son puntos de finalización en el tiempo. Representan metas o logros importantes que se deben alcanzar en un determinado período. Como seres humanos necesitamos plazos para funcionar de manera mas eficiente y productiva. Asociamos un hito a un spring.
- Nos ofrece los **tickets o issues**, a cada issue le puedo asociar un hito, por lo tanto en un Spring a cada tarea le asigno un issue y cada issue esta asociado a un hito que concide con la fecha de finalización del Spring.
- Nos brinda **etiquetas**, con los cuales podemos indicar el tipo de tarea que debemos realizar, por ejemplo para nuestro caso las etiquetas son: Java, Python o Javascript, indicando en que lenguaje debemos programar la tarea.
- Por último, podemos utilizar los **lanzamientos (releases)** que nos ofrece Github para utilizarlos en la metodología Scrum y realizar entregas incrementales. Cada vez que se completa un Spring se puede crear un release y marcar un hito en el progreso del proyecto.

## 1.3 Para implementar Scrum mediante GitHub, debemos seguir los siguientes pasos:

1. **Crear un repositorio en Github:** Crea un repositorio en GitHub que sirva como el espacio central para el proyecto.

2. **Crear un proyecto de GitHub:** GitHub ofrece una funcionalidad de proyectos para organizar y dar seguimiento a tus tareas y sprints. Crea un proyecto de GitHub y configuralo para representar tu tablero Scrum. Podes crear columnas para representar el backlog, el sprint actual y las diferentes etapas de desarrollo.
3. **Configurar el Product Backlog:** Utiliza el sistema de gestión de problemas (issue tracking) de GitHub para crear y organizar las funcionalidades, requisitos y mejoras pendientes en forma de "issues". Cada issue puede representar un elemento del Product Backlog. Prioriza y etiqueta los issues según su importancia y categoría.
4. **Crear un hito (milestone):** son puntos de finalización en el tiempo que se utilizan para rastrear y organizar el progreso de un proyecto. Representan metas o logros importantes que se deben alcanzar en un determinado período. Asociamos un hito a la fecha de finalización de un spring.
5. **Asigna tareas a sprints:** Para cada sprint, selecciona las tareas más prioritarias y muévelas a la columna correspondiente en tu proyecto de GitHub. Esto creará tu Sprint Backlog. Podes asignar los issues a miembros del equipo de desarrollo y establecer fechas de vencimiento para cada uno.
6. **Crear tickets (issues):** para cada tarea, además le asignaremos una etiqueta, indicando que tipo de tarea es, lo asignaremos a un hito que coincide con la fecha de finalización del Spring y le asignaremos un responsable, que será la persona que deberá resolver la tarea.
7. **Realizar seguimiento del progreso:** Utiliza las etiquetas, las asignaciones y los comentarios en los issues para realizar un seguimiento del progreso del equipo. Actualiza el estado de los issues a medida que se van completando y moviéndolos a la columna "Terminado".

## 2. Nuestra experiencia implementando Scrum en Github

1. Creamos el proyecto, con los siguientes pasos:

- Creamos un nuevo repositorio.
- Habilitamos la función de proyectos
- Creamos un nuevo proyecto.
- Configuramos las columnas del proyecto.
- Añadimos nuevos items del proyecto, cada item representa una tarea.

Asignamos un Scrum master para cada spring. La duración de cada Sprint fue entre 2 y 4 semanas.

Creamos el milestone correspondiente al primer Spring

Creamos las etiquetas: Python, Java y JavaScript

Creamos los issues de cada tarea, le asignamos un responsable, una etiqueta y un milestone.

Cada miembro del equipo fue realizando su tarea asignada, una vez realizada la tarea se cierra el issue y se mueve la tarea a Increment, luego el scrum master la revisa y la mueve a la columna Terminado. Así sucesivamente hasta terminar el sprint.

Luego se abre el siguiente sprint y se realiza el mismo ciclo.

### 3. Errores que surgieron en la implementación

Al implementar Scrum en GitHub, es posible que se cometan errores o se encuentren desafíos iniciales por las siguientes causas:

**Falta de conocimiento y experiencia:** Si el equipo no está familiarizado con Scrum o GitHub, es posible que cometan errores debido a la falta de conocimiento y experiencia.

**Mala planificación y estimación de tiempos:** Los errores pueden surgir debido a una planificación y estimación inadecuadas. Esto puede resultar en un desequilibrio entre la capacidad del equipo y la cantidad de trabajo asignado, lo que lleva a retrasos y falta de entrega.

**Falta de comunicación y colaboración:** Los errores pueden ocurrir cuando no hay una comunicación y colaboración efectiva dentro del equipo. Esto puede llevar a malentendidos, falta de alineación y problemas en la coordinación del trabajo.

**Resistencia al cambio:** Es posible que algunos miembros del equipo o partes interesadas muestren resistencia al cambio y se opongan a la implementación de Scrum en GitHub. Esto puede dificultar la adopción exitosa y conducir a errores en la implementación inicial.

En general, es importante abordar los errores de manera constructiva y utilizarlos como oportunidades para aprender y mejorar. Al justificar los errores como parte del proceso de implementación inicial de Scrum en GitHub, se puede crear un entorno que fomente la mejora continua y el éxito a largo plazo.