

# **Anexo I: Planificación del proyecto**

## **App para la navegación basada en comandos de voz del robot iRobot Create 3**

Trabajo de Fin de Grado de Ingeniería Informática



**VNiVERSiDAD D SALAMANCA**

Julio de 2023

**Autor:**

Jorge Sánchez Rubio

**Tutores:**

Francisco Javier Blanco Rodríguez

Belén Curto Diego

# ÍNDICE DE CONTENIDOS

1.	INTRODUCCIÓN .....	3
2.	SCRUM .....	3
2.1.	Burn-Down Chart .....	6
3.	PLANIFICACIÓN TEMPORAL DEL PROYECTO.....	8
3.1.	Participantes .....	8
3.2.	Sprint 1.....	9
3.3.	Sprint 2.....	11
3.4.	Sprint 3.....	13
3.5.	Sprint 4.....	15
4.	REFERENCIAS .....	17

## ÍNDICE DE FIGURAS

Figura 1:	Esquema del proceso SCRUM .....	4
Figura 2:	Ejemplo de Burn-Down Chart ideal.....	6
Figura 3:	Burn-Down Chart Sprint 1 .....	10
Figura 4:	Burn-Down Chart Sprint 2.....	12
Figura 5:	Burn-Down Chart Sprint 3.....	14
Figura 6:	Burn-Down Chart Sprint 4.....	16

## ÍNDICE DE TABLAS

Tabla 1:	Participantes .....	8
Tabla 2:	Sprint 1 .....	9
Tabla 3:	Sprint 2 .....	11
Tabla 4:	Sprint 3 .....	13
Tabla 5:	Sprint 4 .....	15

## 1. INTRODUCCIÓN

En este anexo se documenta la planificación del proyecto. A través de esta planificación se identifican las tareas a realizar, así como la asignación de tiempo y recursos que han sido necesarios para el desarrollo del proyecto.

Scrum es un marco de gestión del proyectos de metodología ágil y está especialmente indicado para proyectos en los que se necesita obtener resultados rápidos y donde los requisitos pueden estar continuamente cambiando o estar poco definidos [2]. Esta es la razón por la que se ha elegido seguir este proceso ya que el proyecto a desarrollar se encuentra dentro de este ámbito mencionado.

## 2. SCRUM

Scrum es un marco de trabajo que se basa en el desarrollo iterativo e incremental [3] de los requisitos del proyecto en bloques temporales cortos y fijos denominados iteraciones. Esto permite una flexibilidad en la adopción de cambios y nuevos requisitos durante el desarrollo del proyecto. Al basarse en un control empírico del proyecto [4], se muestra al cliente el resultado real obtenido al final de cada iteración permitiendo además que el equipo se sincronice diariamente para realizar las adaptaciones que sean necesarias.

Se pueden definir tres pilares [5] importantes en el marco de trabajo Scrum:

- **Transparencia:** todos los implicados en el proyecto tienen conocimiento de qué está ocurriendo en él y cómo está ocurriendo, de manera que hay una visión global para todos.
- **Inspección:** los miembros del equipo del proyecto inspeccionan frecuentemente el progreso para detectar posibles problemas.

- **Adaptación:** el equipo ajusta los aspectos que se desvíen de los límites aceptables para conseguir el objetivo del sprint. Es la clave para conseguir el éxito en proyectos complejos.

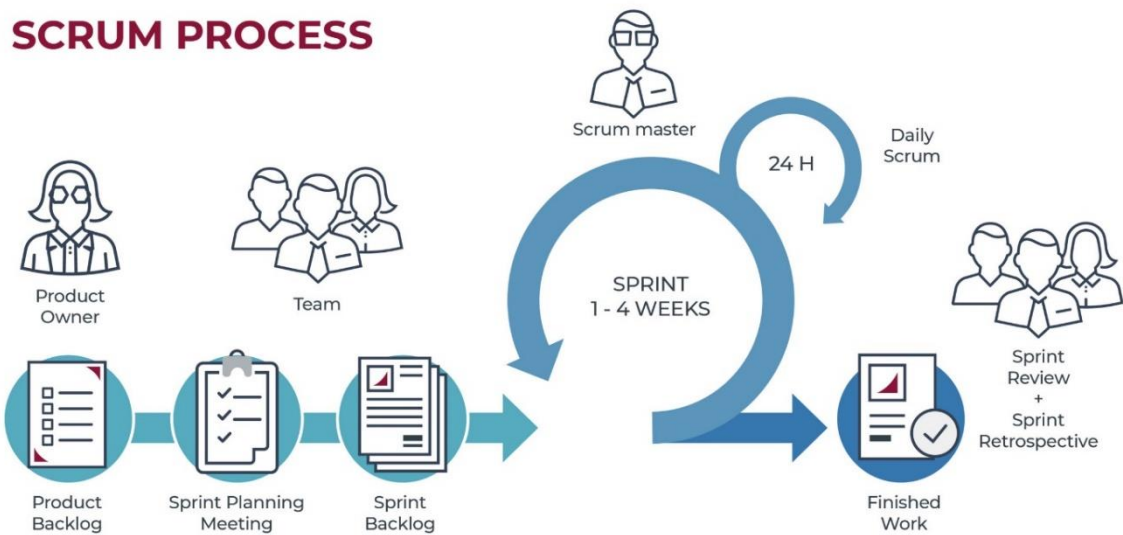


Figura 1: Esquema del proceso SCRUM

Los equipos de un proyecto Scrum son autoorganizados y multifuncionales. Cada miembro es responsable de unas tareas determinadas.

Los tres roles más importantes en un equipo de trabajo Scrum son:

- **Product Owner:** es el responsable de maximizar el valor del trabajo y está en constante contacto con el cliente. Sólo hay uno.
- **Scrum Master:** es el responsable de que las técnicas scrum sean comprendidas y aplicadas.
- **Equipo de desarrollo:** son los miembros encargados de realizar las tareas priorizadas por el *Product Owner*.

Es importante tener en cuenta los siguientes conceptos para el uso del proceso Scrum.

- **Product Backlog:** se trata del listado de tareas (requisitos) que engloban todo el proyecto.
- **Sprint Backlog:** es el grupo de tareas del *Product Backlog* que proporciona una imagen en tiempo real del trabajo que se debe realizar para lograr el objetivo y que se desarrollan dentro de un sprint concreto.
- **Incremento:** es el resultado de sumar todos los elementos del Product Backlog completados durante el sprint actual.
- **Sprint:** es un período de tiempo fijo en el que un equipo de scrum trabaja para completar una cantidad de trabajo establecida. El resultado de un sprint es un producto funcional que el cliente puede ejecutar.

A continuación, se describen algunas de las ventajas y desventajas del marco Scrum.

En cuanto a las ventajas, Scrum es fácil de aprender ya que su filosofía es clara y tiene un objetivo. Además, el cliente puede comenzar a usar el producto rápidamente y el proceso se agiliza ya que las entregas del producto son frecuentes, lo que da lugar también a una menor probabilidad de imprevistos. Es especialmente útil cuando se trabaja en entornos de alta incertidumbre donde la probabilidad de cambios es alta.

En cuanto a las desventajas, Scrum supone un cambio de la cultura de la organización, aunque sea fácil de aprender es difícil de implementar. También la tendencia del equipo a realizar el camino más corto para conseguir un objetivo no siempre ofrece resultados de calidad.

## 2.1. Burn-Down Chart

Otro concepto importante que se debe tener en cuenta al utilizar Scrum para el desarrollo del proyecto es el *Burn-Down Chart*.

Se trata de una gráfica [6] en la que se puede observar el estado del progreso de un sprint y tener así una perspectiva del trabajo. De esta manera se relaciona el trabajo pendiente con un periodo de tiempo determinado.

Esta gráfica se constituye de un eje X en donde se representa el tiempo finito (generalmente en días) de duración de un sprint. En el eje Y se refleja la cantidad total de trabajo comprometido para realizar en el tiempo estipulado. También se representa en la gráfica una línea de referencia o de guía; que es una línea diagonal que une el último valor del eje Y (máximo trabajo comprometido) con el último valor del eje X (fecha máxima para el trabajo comprometido).

En la Figura 2 podemos observar un ejemplo de cómo puede ser un *Burn-Down Chart* ideal:

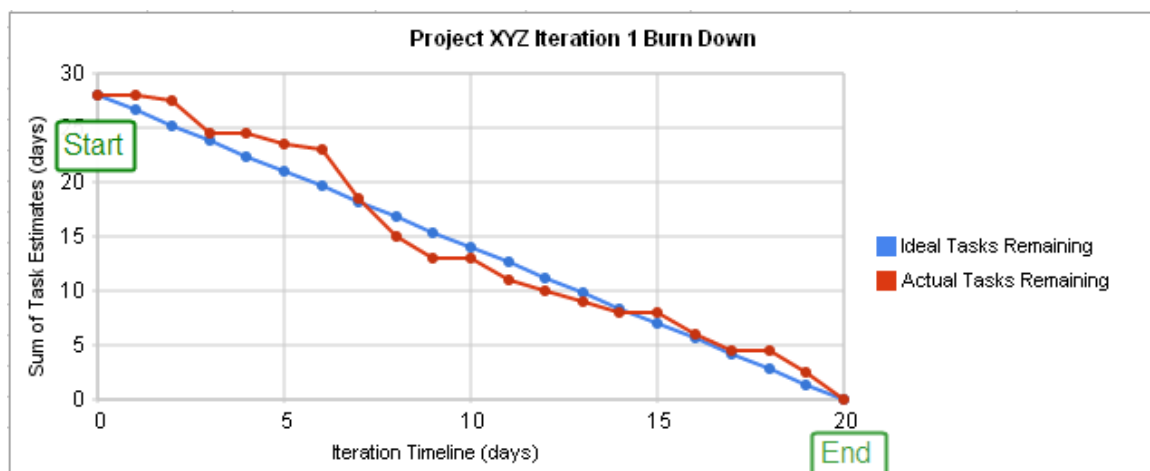


Figura 2: Ejemplo de Burn-Down Chart ideal

Es necesario que las tareas a realizar para el desarrollo del proyecto estén estimadas en tiempo y esfuerzo. Para hacerlo, en Scrum existen diferentes técnicas entre las que destaca la técnica *Planning Poker* [7]; que consiste en unas métricas abstractas llamadas puntos de historia o *story points* regidos por la escala de Fibonacci (1-2-3-5-8-13), donde una historia de usuario con un gran número de puntos de historia (13) debe ser replanteada y dividida en historias más simples.

Los puntos de historia dan al usuario una idea del tamaño y esfuerzo que son necesarios para realizar una historia de usuario concreta (de cada tarea). La suma de los puntos de historia de cada historia de usuario perteneciente a un sprint proporciona el trabajo total de un sprint.

### 3. PLANIFICACIÓN TEMPORAL DEL PROYECTO

La planificación temporal del proyecto se divide en diferentes sprints con un objetivo o hito, que están a su vez compuestos por historias de usuario y estos por tareas.

Las historias de usuario se encargan de definir la funcionalidad del sistema por medio de una explicación no formal y a su vez, cada historia de usuario tiene una serie de tareas que se deben terminar para poder finalizar la historia de usuario correspondiente. Se presenta para cada historia de usuario ,y por lo tanto para cada sprint, un esfuerzo estimado en puntos de historia (o *story points*) basado en el método *Planning Poker* mencionado anteriormente.

#### 3.1. Participantes

En la Tabla 1 se especifican los roles del equipo Scrum de este proyecto.

Rol	Descripción	Personas
Product Owner	Supervisor	Francisco Javier Blanco Rodríguez, Belén Curto Diego
Scrum Master	Director del proyecto	Jorge Sánchez Rubio
Development Team	Desarrollador	Jorge Sánchez Rubio

Tabla 1: Participantes



### 3.2. Sprint 1

Objetivo – Implementar un reconocimiento de voz funcional.

ID	Historia de usuario	Tareas	Esfuerzo estimado
HU-00	Conceptualización del proyecto	Realizar reunión con los tutores para entender el contexto del proyecto y sus objetivos	40
HU-01	Configuración inicial del proyecto y entorno de desarrollo de la aplicación	Instalar y configurar el entorno de desarrollo de Android Studio	60
		Determinar la estructura básica del proyecto	
		Determinar y establecer las dependencias necesarias	
HU-02	Implementar la captación de voz en la aplicación	Documentarse e implementar el <code>SpeechRecognizer</code> de Android para la captación de voz	70
		Diseñar la lógica e interfaz gráfica para la captación de voz	
HU-03	Pruebas y errores	Realizar pruebas y solucionar posibles errores	50

Tabla 2: Sprint 1

En este primer sprint (Tabla 2) se lleva a cabo una primera toma de contacto para informarse a cerca del proyecto y sus objetivos principales. Posteriormente a esta conceptualización se lleva a cabo la instalación y configuración inicial del entorno de desarrollo de la aplicación (Android Studio). Una vez establecidas las dependencias necesarias (permisos de audio, internet ...) y la estructura básica, se da paso a la documentación e implementación de

la captación por voz de órdenes en la aplicación a través del uso de la clase `SpeechRecognizer` que Android ofrece unido al diseño e implementación de una interfaz de usuario básica necesaria.

Por último, se realizan pruebas para comprobar el correcto funcionamiento y solucionar posibles errores.

En el siguiente gráfico *Burn-Down Chart* (Figura 3) se observa cómo ha sido la evolución de trabajo del sprint en función del esfuerzo estimado de cada una de las tareas.

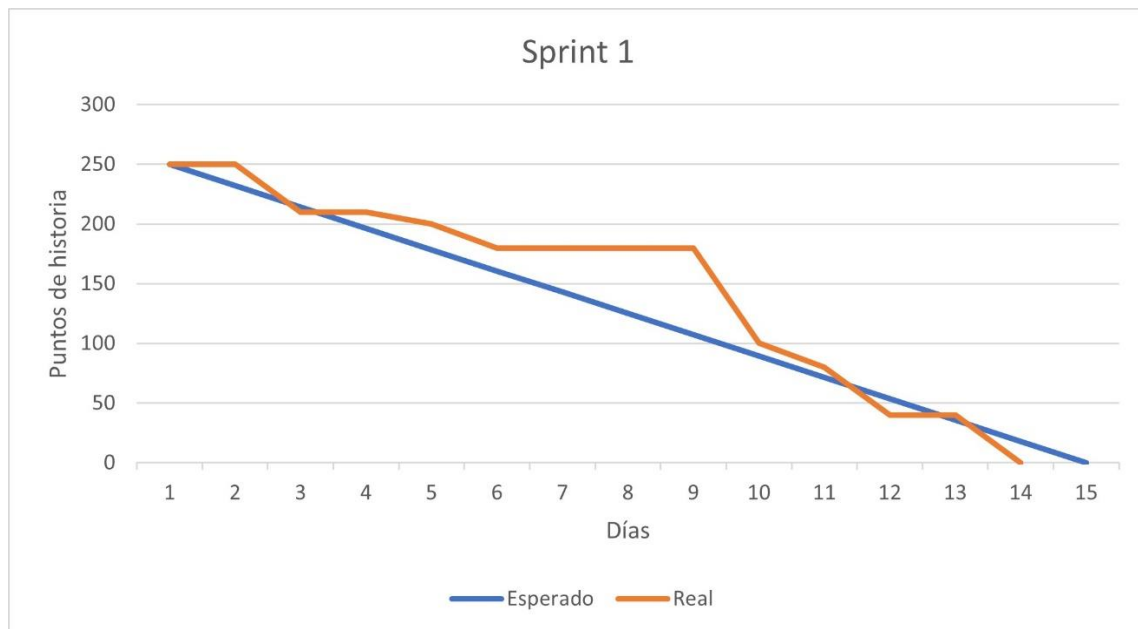


Figura 3: *Burn-Down Chart Sprint 1*

Se puede observar cómo hubo una serie de días en los que el trabajo se estancó debido algunos imprevistos que tardaron en solucionarse y una vez conseguido se pudo acabar incluso antes del tiempo estimado.

### 3.3. Sprint 2

Objetivo – Establecer la comunicación entre la aplicación y el servidor.

ID	Historia de usuario	Tareas	Esfuerzo estimado
HU-04	Establecer la comunicación TCP/IP utilizando sockets	Establecer la conexión del socket entre la aplicación y el servidor de ROS2	70
		Definir el protocolo de comunicación para enviar las órdenes de voz captadas al servidor	
HU-05	Desarrollar el servidor de ROS2 en Linux	Documentarse y configurar ROS2 en Linux	100
		Crear el servidor de ROS2 para recibir las órdenes de voz	
HU-06	Pruebas y errores	Realizar pruebas y solucionar posibles errores	60

*Tabla 3: Sprint 2*

En este segundo sprint (Tabla 3) se procede a definir y establecer el protocolo de comunicación TCP/IP para el envío de las órdenes de voz desde la aplicación utilizando sockets. A su vez, comienza la fase de documentación y configuración de ROS2 en Linux para la creación del servidor de ROS2 con el fin de recibir en este último las órdenes de voz enviadas desde la aplicación.

Por último, se realizan pruebas para comprobar el correcto funcionamiento y solucionar posibles errores.

En el siguiente gráfico *Burn-Down Chart* (Figura 4) se observa cómo ha sido la evolución de trabajo del sprint en función del esfuerzo estimado de cada una de las tareas.

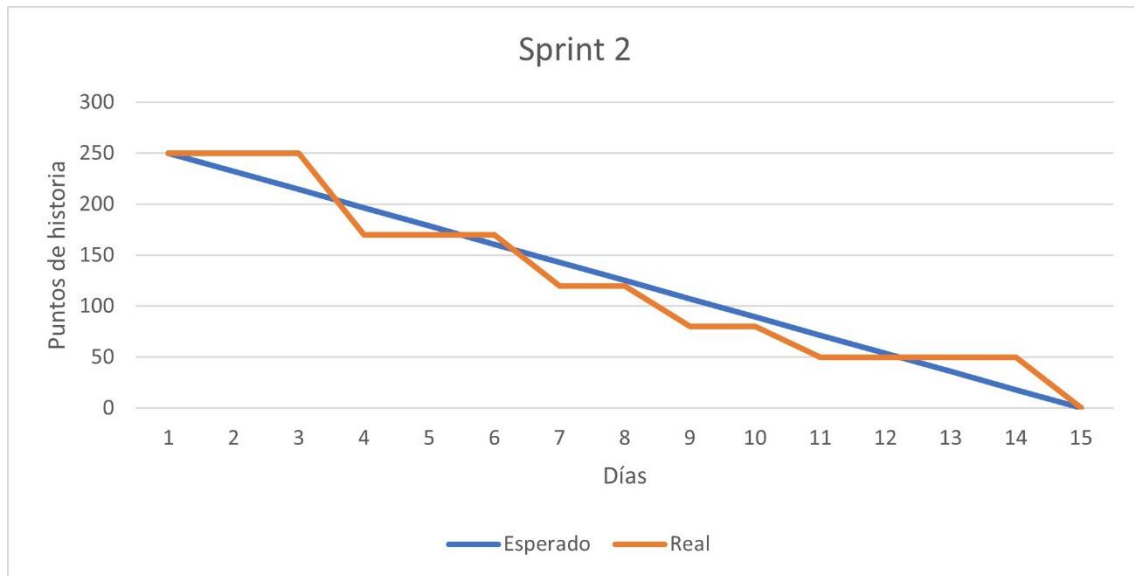


Figura 4: *Burn-Down Chart* Sprint 2

En este caso se puede observar cómo el desarrollo del proyecto fue avanzando sin muchos contratiempos, aunque finalmente hubo una serie de días en los que de nuevo se tuvieron que solucionar algunos problemas, pero con el éxito de poder acabar en el tiempo establecido.

### 3.4. Sprint 3

Objetivo – Controlar el robot y recibir la realimentación en la app.

ID	Historia de usuario	Tareas	Esfuerzo estimado
HU-07	Implementar la lógica de control del robot en el servidor de ROS2	Documentarse sobre las APIs de ROS2, el robot y cómo controlarlo	120
		Implementar la lógica de control para ejecutar las órdenes de movimiento del robot	
HU-08	Construir la interfaz gráfica de usuario para mostrar el estado de la orden	Diseñar la interfaz gráfica para mostrar el estado de la orden en la aplicación	70
		Integrar la interfaz gráfica de la aplicación con el servidor de ROS2 para obtener la realimentación requerida	
		Implementar en el servidor de ROS 2 la realimentación del robot al usuario	
HU-09	Pruebas y errores	Realizar pruebas y solucionar posibles errores	50

Tabla 4: Sprint 3

En el tercer sprint (Tabla 4) se procede a implementar en el servidor de ROS2 la lógica de control del robot. Para ello, primero es necesario documentarse sobre las APIs de ROS2 y el robot tanto a nivel hardware como software. Una vez implementada la lógica, se procede a modificar la interfaz gráfica para añadir la posibilidad de mostrar el estado de la orden. También se configura el servidor de ROS2 para que el robot muestre una realimentación visual al usuario sobre la orden dada.

Por último, se realizan pruebas para comprobar el correcto funcionamiento y solucionar posibles errores.

En el siguiente gráfico *Burn-Down Chart* (Figura 5) se observa cómo ha sido la evolución de trabajo del sprint en función del esfuerzo estimado de cada una de las tareas.

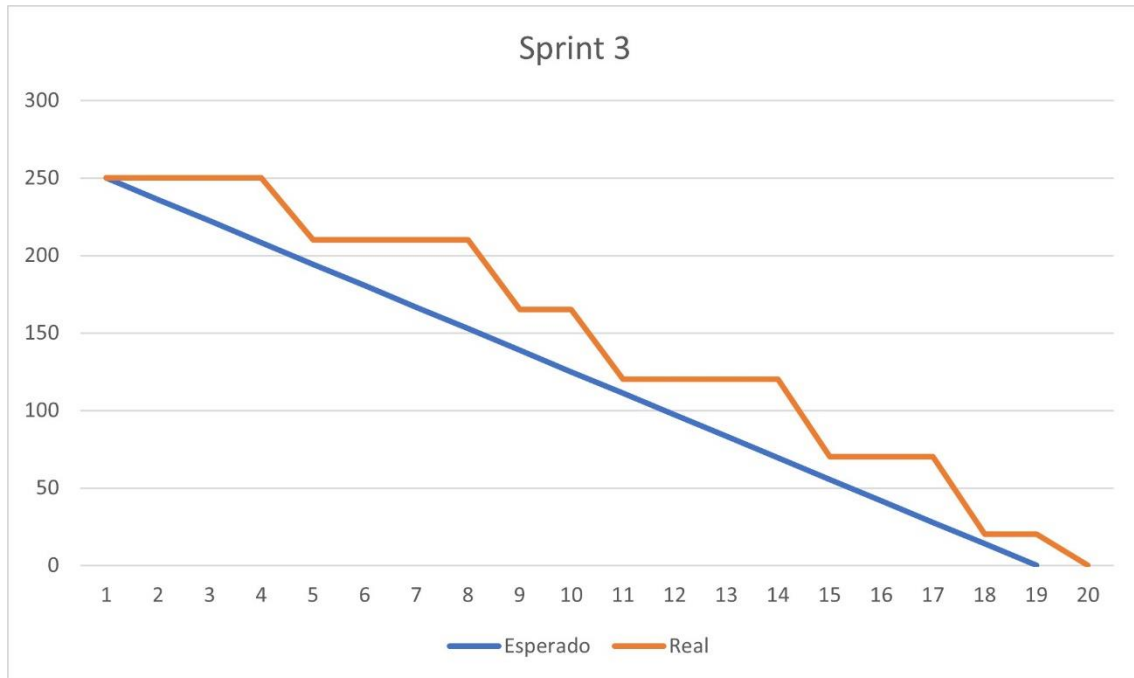


Figura 5: *Burn-Down Chart Sprint 3*

En la gráfica anterior (Figura 5) se observa cómo esta vez el desarrollo del proyecto fue escalonado, con algunos días en los que se taró un poco más de lo esperado, lo que finalmente dio lugar a terminar un día más tarde de lo que se había estimado en un primer momento.

### 3.5. Sprint 4

Objetivo – Realizar pruebas y mejoras.

ID	Historia de usuario	Tareas	Esfuerzo estimado
HU-10	Realizar pruebas y depurar errores	Realizar pruebas de funcionamiento del sistema completo	70
		Identificar y solucionar posibles errores o problemas hallados	
HU-11	Mejoras y ajustes	Implementar mejoras de la interfaz gráfica de la aplicación para mejorar la experiencia del usuario	40
		Realizar ajustes de la lógica de control del robot si es necesario	

Tabla 5: Sprint 4

Con este cuarto y último sprint (Tabla 5), la última fase del proyecto se centra en la realización de pruebas de funcionamiento del sistema completo y la depuración de posibles errores; seguido de la implementación de ciertas mejoras y ajustes tanto en la interfaz gráfica de usuario como en la lógica de control del robot en caso de ser necesario.

En el siguiente gráfico *Burn-Down Chart* (Figura 6) se observa cómo ha sido la evolución de trabajo del sprint en función del esfuerzo estimado de cada una de las tareas.

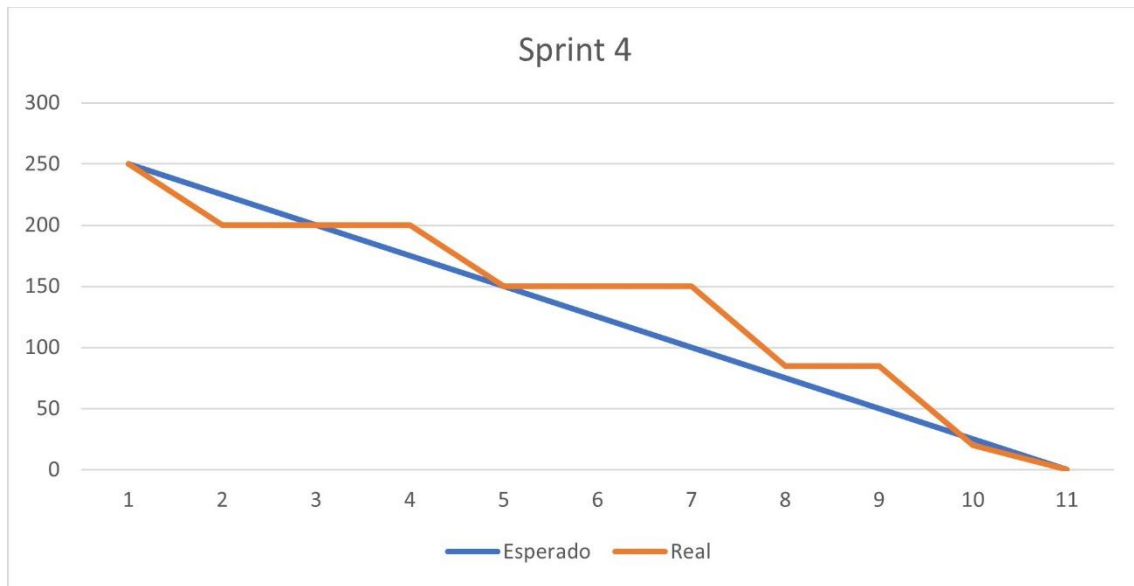


Figura 6: *Burn-Down Chart* Sprint 4



## 4. REFERENCIAS

- [1] «¿Qué es la metodología ágil?» <https://www.redhat.com/es/devops/what-is-agile-methodology> (accedido 5 de junio de 2023).
- [2] «Scrum: qué es, cómo funciona y por qué es excelente». <https://www.atlassian.com/es/agile/scrum> (accedido 5 de junio de 2023).
- [3] «Qué es SCRUM – Proyectos Ágiles». <https://proyectosagiles.org/que-es-scrum/> (accedido 5 de junio de 2023).
- [4] «Fundamentos de Scrum – Proyectos Ágiles». <https://proyectosagiles.org/fundamentos-de-scrum/> (accedido 5 de junio de 2023).
- [5] «Scrum: el marco de trabajo ágil para adaptarse a los cambios». <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html> (accedido 5 de junio de 2023).
- [6] «Burndown Chart (Scrum): qué es, cómo hacer uno y ejemplos». <https://profile.es/blog/burndown-chart/> (accedido 6 de junio de 2023).
- [7] «Planning Poker: metodología scrum para una estimación ágil». <https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/planning-poker> (accedido 6 de junio de 2023).