

Javier García, Héctor Ruíz, Lucian
Andrei y Juan Picazo

EDT de la metodología SCRUM

7 de marzo, 2025

Curso 2024-2025. UCLM, Albacete
GSI, Gestión de Sistemas Informáticos
Grupo de trabajo 6

Índice

1. Introducción.....	3
2. Estructura del Desglose del Trabajo (EDT).....	3
a. Diagrama del EDT.....	3
b. Descripción de las fases y entregables.....	4
i. Fase 1, inicio y planificación del producto.....	4
ii. Fase 2, planificación del Sprint.....	4
iii. Fase 3, Ejecución y desarrollo del Sprint.....	6
iv. Fase 4, revisión del Sprint y entrega.....	6
v. Fase 5, retrospectiva del Sprint.....	7
3. Justificación de los entregables.....	7
4. Referencias bibliográficas.....	8
5. Anexos.....	9
a. Planificación de la tarea.....	9

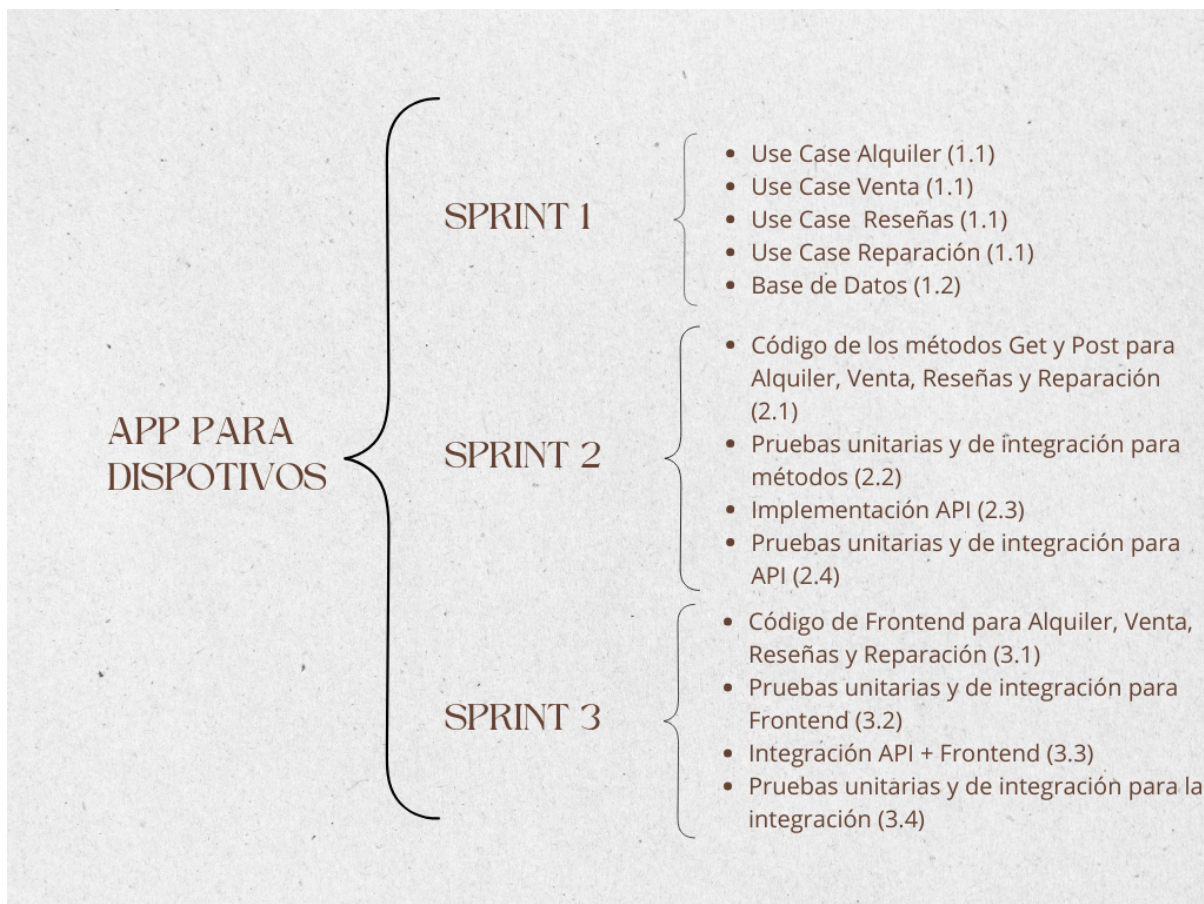
1.Introducción

SCRUM es un marco de trabajo ágil para el desarrollo de software que se centra en la entrega incremental de productos a través de iteraciones cortas llamadas Sprints. En este documento, se presenta la Estructura de Desglose del Trabajo (EDT) para SCRUM, identificando sus fases principales y los seis entregables más relevantes en cada una de ellas.

Para la realización de este trabajo tomaremos como referencia el proyecto desarrollado en la asignatura de Ingeniería del Software II, donde la metodología base utilizada es SCRUM.

2.Estructura del Desglose del Trabajo (EDT)

a. Diagrama del EDT



b. Descripción de las fases y entregables

i. Fase 1, inicio y planificación del producto

En esta fase inicial se sientan las bases fundamentales del proyecto mediante la identificación clara del producto a desarrollar y la organización del trabajo:

- **Product Backlog:** Lista priorizada que recopila claramente los requisitos y funcionalidades del producto. Garantiza el orden y enfoque de los esfuerzos del equipo.
- **Visión del Producto:** Documento que establece de forma sencilla y precisa los objetivos generales del proyecto, asegurando una comprensión común entre todos los implicados.
- **Roles y Responsabilidades:** Definición detallada de las responsabilidades del Product Owner, Scrum Master y Development Team, asegurando claridad en las tareas y eliminando ambigüedades.
- **Roadmap del Producto:** Planificación a largo plazo, determinando releases, hitos clave y objetivos generales que permiten visualizar el progreso esperado del proyecto.
- **DoD (Definition of Done):** Establece los criterios específicos para considerar un trabajo completado, asegurando estándares de calidad y facilitando la evaluación objetiva del progreso.
- **Plan de Sprints Inicial:** Define claramente el número y duración estimada de los sprints, ofreciendo un esquema organizado que permite anticipar esfuerzos y recursos necesarios.

ii. Fase 2, planificación del Sprint

En esta fase del proceso del EDT se define el contenido de cada uno de los sprints, marcando el Sprint Backlog, el Sprint Goal, las tareas a realizar, las estimaciones de trabajo y esfuerzo, las futuras métricas de éxito y la identificación de riesgos para cada Sprint.

Sprint 1: Requisitos, Use Cases, Modelos y Base de Datos

- **Sprint Backlog:** Toma y análisis de requisitos para hacer nuestra página web, la elaboración de nuestros Use Cases (Venta, alquiler, reparación y opiniones) y el diseño de modelos de datos y estructura de la base de datos.
- **Sprint Goal:** “Definir y documentar los requisitos, Use Cases y el modelo de la base de datos que servirán para el desarrollo de la API y la web”.
- **Tareas desglosadas:**
 - Análisis y definición de requisitos
 - Redacción y validación de los Use Cases
 - Diseño y modelado de la BBDD
 - Revisión de todo lo generado
- **Estimación de esfuerzo y tiempo:** Realizado en la planificación de la tarea.
- **Definición de las métricas de éxito:**
 - Aprobación por parte del Product Owner
 - Diseño coherente y escalable para la BBDD
 - Calidad y realización de los Use Cases
- **Identificación de los posibles riesgos:** Ambigüedad a la hora de hacer los Use Cases e inadaptación al tiempo establecido.

Sprint 2: Implementación de los Use Cases y Testing de la API

- **Sprint Backlog:** Desarrollo de la lógica de negocio de los Use Cases, implementación de la API para realizar los procesos definidos en los Use Cases y ejecución de pruebas unitarias y de integración para validar el uso de la API.
- **Sprint Goal:** Transformar los documentos del Sprint 2 en un diseño con una API funcional, con test los cuales validarán su funcionamiento y utilidad en el flujo de datos y procesos.
- **Tareas desglosadas:**
 - Paso a código de cada Use Case
 - Integración de la API con los datos y lógica de negocio
 - Configuración y ejecución de pruebas (de integración y unitarias)
 - Documentación de pasos, incidencias y correcciones de estas
- **Estimaciones de esfuerzo:** Realizado en la planificación de la tarea.
- **Definición de métricas de éxito:**
 - Cumplimiento de los criterios del DoD para cada Use Case
 - Resultados correctos en los test
 - Documentación clara de pasos y corrección de incidencias
- **Identificación de los posibles riesgos:** Posibles problemas al integrar la API junto a la BBDD. También podrán aparecer problemas en las pruebas de los test automatizados.

Sprint 3: Implementación en la Web, Diseño y Testing Funcional/Unitario

- **Sprint Backlog:** Desarrollo del frontend de nuestra página web, integración de la interfaz de la página web con la API e implementación de diseño UI/UX junto a la realización de las pruebas funcionales y unitarias.
- **Sprint Goal:** Conseguir una interfaz operativa, atractiva e integrada con la API, asegurando que todo funciona correctamente mediante la realización de los test.
- **Tareas desglosadas:**
 - Desarrollo del diseño de la página web
 - Integración de frontend con la API
 - Realización de pruebas funcionales para UI/UX
 - Realización de pruebas unitarias para los componentes
- **Estimaciones de esfuerzo:** Realizado en la planificación de la tarea.
- **Definición de métricas de éxito:**
 - Interfaz fácil y funcional
 - Use Cases probados y funcionando correctamente
 - Respuesta positiva de los test
- **Identificación de riesgos:** Posibles fallos con la integración de la API con el frontend, al igual que posibles problemas con las pruebas funcionales y unitarias.

iii. Fase 3, Ejecución y desarrollo del Sprint

Se ejecutan las tareas planificadas de cada sprint, con reuniones diarias, seguimiento en el Scrum Board y actualizaciones constantes del Sprint Backlog.

Ejecución del Sprint 1: Requisitos y Modelado

- **Daily Scrum:** Reunión diaria para ver el progreso, identificar bloqueos y ajustar las tareas repartidas.
- **Actualización del Sprint Backlog:** Seguimiento constante de los requisitos, Use Cases y BBDD.
- **Resolución de problemas:** Intervención inmediata sobre cualquier duda o conflicto.
- **Colaboración:** Trabajo en equipo que dará lugar a la validación y coherencia de los entregables.

Ejecución del Sprint 2: Implementación y Testing de la API

- **Daily Scrum:** Seguimiento diario de la implementación y el testing, ver posibles errores.
- **Actualización del Sprint Backlog:** Ajuste de las tareas guardando relación con el desarrollo de la API.
- **Resolución de problemas:** Soporte técnico y colaboración para solucionar problemas de testing e implementación.
- **Validación:** Revisión de los resultados obtenidos y ajustar tiempo y futuras pruebas.

Ejecución del Sprint 3: Desarrollo Web y Pruebas Funcionales/Unitarias

- **Daily Scrum:** Coordinación de frontend y equipo de API para alinear avances y problemas.
- **Actualización del Sprint Backlog:** Revisión de diseño, desarrollo e implementación de test
- **Resolución de problemas:** En relación a la API o interfaz.
- **Validación:** Ejecución de todos los test y pruebas para confirmar los estándares.
- **Tablero Scrum:** Visualizar el estado de cada tarea para ver el seguimiento y la colaboración entre miembros del equipo.

iv. Fase 4, revisión del Sprint y entrega

En esta fase del proceso del EDT se evalúa todo lo que hemos hecho con el proyecto, demostramos el producto y recopilamos información para hacer mejoras. Siempre al final de cada Sprint.

Sprint 1: Documentación y Modelado Validado

- **Sprint Review:** Revisión de todos los documentos de la definición de requisitos y los cuatro Use Cases realizados, además de los modelos y la base de datos.
- **Feedback:** Comentarios por parte del Product Owner y los clientes o posibles clientes de nuestro producto.
- **Registro de cambios:** Actualizaciones y cambios tras el feedback.
- **Entrega:** Modelado y documentación “arreglada”.
- **Actualización del Product Backlog:** Incorporación de lo acordado con el Product Owner y clientes.

Sprint 2: Validación de la API

- **Sprint Review:** Comprobar el correcto funcionamiento de la API y todos aquellos procesos que hayamos añadido.
- **Feedback:** Comentarios por parte del Product Owner y los clientes o posibles clientes de nuestro producto y los resultados de nuestras pruebas.
- **Registro de cambios:** Actualizaciones y cambios tras el feedback.
- **Entrega:** Incrementación de la API, ya acabada para integrarse junto al Frontend.
- **Actualización del Product Backlog:** Incorporación de lo acordado con el Product Owner y clientes, más corrección de posibles errores.

Sprint 3: Revisión y validación de la Web

- **Sprint Review:** Comprobar el correcto funcionamiento del Frontend, el diseño y la integración junto a la API.
- **Feedback:** Comentarios por parte del Product Owner y los clientes o posibles clientes de nuestro producto y los resultados de nuestras pruebas.
- **Registro de cambios:** Actualizaciones y cambios tras el feedback.
- **Entrega:** Versión final de la web, totalmente acabada y operativa.
- **Actualización del Product Backlog:** Inclusión de las mejoras realizadas y comentarios ante una posible futura actualización o mantenimiento.

v. Fase 5, retrospectiva del Sprint

- **Revisión de Procesos:** Evaluación objetiva del desempeño general del equipo durante el sprint, para determinar áreas de fortaleza y puntos a mejorar.
- **Identificación de Mejoras:** Análisis detallado para comprender qué aspectos del trabajo han sido efectivos y cuáles requieren atención adicional.
- **Acciones Correctivas:** Definición específica de cambios en la metodología o en el enfoque del trabajo, que permitirán corregir problemas identificados y optimizar futuras entregas.
- **Documentación de la Retrospectiva:** Registro exhaustivo y accesible de aprendizajes clave, asegurando que las lecciones sean aplicadas en los siguientes ciclos de trabajo.
- **Feedback Interno:** Captura y análisis de opiniones y sugerencias por parte de los miembros del equipo, fortaleciendo la comunicación y colaboración internas.
- **Plan de Mejora Continua:** Formulación de estrategias concretas que buscan aplicar los aprendizajes obtenidos para optimizar los próximos sprints, asegurando un avance constante y sostenido del equipo.

3. Justificación de los entregables

Requisitos (Sprint 1): Recoge todo aquello que el cliente quiere y pide, siendo esto como el “padre nuestro” de nuestro proyecto. Gracias a esto podremos comprobar que hemos llegado al objetivo final.

Use Cases (Sprint 1): Ayudan a dividir la idea principal, especificando más detalles y qué hará concretamente cada parte de nuestra Web. Podremos priorizar funcionalidades, planificándolas correctamente.

Modelos (Sprint 1): Definen la estructura y las relaciones del sistema, llegando a aportar escalabilidad y fiabilidad.

Base de Datos (Sprint 1): Podremos almacenar y manejar todo aquello que atraviese nuestra página web.

Lógica de Negocio y Procesos (Sprint 2): Segunda parte de los requisitos y de los Use Cases. Entregable al equipo encargado del desarrollo de la API.

API (Sprint 2): Es el puente entre la BBDD y el Frontend, gracias a esta entrega podemos realizar ciertas acciones como búsquedas de información, muestreo de información, confirmaciones... Uno de los entregables más importantes.

Frontend Web (Sprint 3): Es allí donde nuestro cliente y usuarios podrán ver tangible nuestro trabajo, esta parte luego se relacionará con la API dándole una utilidad total a nuestra página web.

Feedback de las pruebas unitarias y de integración: Se pueden ver como un entregable realizado a nosotros mismos, recibimos la respuesta y la confirmación de que todo funcionó a la perfección, muy útil para confirmar nuestros pasos y poder seguir avanzando a lo largo de todo el proceso.

4. Referencias bibliográficas

Qué es SCRUM. (2008, August 4). Proyectos Ágiles.

<https://proyectosagiles.org/que-es-scrum/>

Ferreiro, A. J. C. (2024). *SCRUM. Teoría e Implementación práctica*. Ra-Ma Editorial.

Enlaces del aula virtual.

Curso de Logística y Transporte desarrollado por la ESIIAB.

Ingeniería del Software II.

5.Anexos

a. Planificación de la tarea

Tarea	Descripción	Resultado de la tarea	Recursos	Tiempo estimado	Tiempo invertido	Causas desviación
Sprint 1: Toma y análisis de requisitos.	Reunirse con el cliente para analizar y crear los requisitos del proyecto.	Documento de requisitos e idea del proyecto.	1.Entrevistas 2.Documento de Word	8:00	8:00	No hubo desviación ya que las estaban ya con horario cerrado.
Sprint 1: Elaboración de Use Cases.	Definir y documentar los cuatro Use Cases (Venta, Alquiler, Reparación, Opiniones) a partir de los requisitos anteriores.	4 Use Cases.	Documento de Word	5:00	6:30	Hubo que rehacer un Use Case entero.
Sprint 1: Diseño y modelado de la BBDD.	Diseñar la estructura de la base de datos y sus relaciones en relación a los Use Cases anteriores	Modelo MER y una BBDD.	Visual Studio 2022	10:00	15:00	Es bastante complejo hacer el diagrama MER y la BBDD.
Sprint 1: Revisión y validación de entregables.	Revisar todas las tareas anteriores del Sprint 1	Documentación y aprobación		3:00	1:00	Todo lo anterior estaba ya muy claro.
Sprint 2: Implementación de la API.	Codificar la lógica de negocio y los métodos (GET/POST) basados en el Sprint 1	API funcional con todos los métodos definidos.	Visual Studio 2022	20:00	22:00	Cada vez que realizamos un pull request nos llevaba tiempo aprobarlo, por lo que se sumó bastante tiempo.

Sprint 2: Pruebas unitarias e integración de la API.	Configurar y ejecutar pruebas unitarias e integradas para validar la API.	Resultado de los test.	Visual Studio 2022	7:00	7:00	Sin desviación.
Sprint 2: Documentación de incidencias y corrección.	Registrar errores y corrección de los mismos.	Documentación de corrección de incidencias.	Documento de Word	2:00	0:30	No hubo prácticamente incidencias.
Sprint 2: Revisión y validación.	Presentar la API al cliente, observar feedback y validar trabajo.	API operativa al 100%.	Reunión	1:30	1:30	El tiempo estaba acordado antes de la reunión y no se tuvo que ampliar ya que no hubo problemas.
Sprint 3: Desarrollo del Frontend.	Crear todo el Frontend, la parte tangible del proyecto.	Frontend 100% operativo.	1.Framework de Blazor 2.Visual Studio 2022 3.ChatGPT	8:00	6:00	El uso de ChatGPT para el código fue muy útil.
Sprint 3: Integración del Frontend con la API.	Enlazar nuestro código de Frontend a la API.	Página web conectada a la API.	1.Framework de Blazor 2.Visual Studio 2022 3.ChatGPT	2:00	1:30	El uso de ChatGPT para el código fue muy útil.
Sprint 3: Pruebas funcionales y unitarias.	Realizar las pruebas de usabilidad, validación de formularios y test unitarios.	Informe de pruebas funcionales y unitarias.	1.Framework de Blazor 2.Visual Studio 2022 3.ChatGPT 4.Documento de Word	4:00	3:00	El uso de ChatGPT para el código fue muy útil.
Sprint 3: Revisión final y entrega de la Web.	Presentar todo nuestro proyecto finalizado al cliente y personas enlazadas a el mismo.	Página web operativa, integrada y lista para su despliegue a los usuarios.	Reunión	2:00	4:00	Debimos explicar el proceso del desarrollo de la página web, no contábamos con ello.

(algún parámetro de tiempo está simulado ya que realmente no se realizó realmente esa acción)