



PROYECTO INTERMODULAR

SPRINT 5

Contexto del sprint

En el **Sprint 5** continuamos con el **modelado del sistema**, centrándonos en representar de forma clara y completa:

- La funcionalidad del sistema (qué hace y para quién).
- El flujo de trabajo de sus procesos principales.
- La interacción entre las distintas partes de la aplicación (FrontEnd y BackEnd).
- La estructura de datos (tanto a nivel lógico como a nivel de intercambio JSON).

Todo el modelado de este sprint debe realizarse con la **herramienta PlantUML**, utilizando la sintaxis de los distintos tipos de diagramas de PlantUML indicados más abajo.

Objetivos del sprint

Al finalizar este sprint, el alumnado deberá ser capaz de:

1. **Modelar los casos de uso principales** del sistema mediante **diagramas de casos de uso**.
2. **Describir los flujos de negocio clave** mediante **diagramas de actividad**.
3. **Representar la interacción temporal** entre actores, FrontEnd y BackEnd mediante **diagramas de secuencia**.
4. **Definir la arquitectura lógica de la aplicación** (capas, módulos y servicios) mediante **diagramas de componentes**, teniendo en cuenta **FrontEnd y BackEnd**.
5. **Modelar los formatos de intercambio de datos** en **JSON**, mediante el diagrama correspondiente.
6. **Diseñar el modelo de datos relacional** del sistema mediante **diagramas IE (Entidad–Interrelación)**.

Nota: El **corazón del modelado funcional** de este sprint se apoya especialmente en el **diagrama de casos de uso** y el **diagrama de actividad**, que deben quedar bien trabajados.

Actividad 1 – Diagrama de Casos de Uso (Use Case Diagram)

- **Objetivo:** Identificar y documentar los **actores** y **casos de uso principales** del proyecto.
- **Herramienta / Tipo de diagrama:**
 - PlantUML usando el **diagrama de casos de uso** (PlantUML: *Use Case Diagram* <https://plantuml.com/es/use-case-diagram>).
- **Alcance mínimo:**
 - Identificar todos los **actores externos** (usuarios, sistemas externos, etc.).
 - Definir los **casos de uso principales** del sistema (no ejemplos genéricos: relacionados con vuestro proyecto real).
 - Incluir relaciones de **inclusión** y **extensión** cuando corresponda.

Actividad 2 – Diagramas de Actividad (Activity Diagram)

- **Objetivo:** Describir los **flujos de actividad** de los procesos más importantes del sistema.
- **Herramienta / Tipo de diagrama:**
 - PlantUML usando **diagramas de actividad** (PlantUML: *Activity Diagram* <https://plantuml.com/es/activity-diagram-beta>).
- **Alcance mínimo:**
 - Elegir al menos **2 casos de uso críticos** del proyecto (por ejemplo: registro de usuario, compra, reserva, gestión de pedido, etc.).
 - Para cada uno, realizar un **diagrama de actividad** que muestre:
 - Estados/actividades principales.
 - Decisiones (ramas condicionales).
 - Posibles bifurcaciones y uniones de flujo.

Actividad 3 – Diagramas de Secuencia (Sequence Diagram)

- **Objetivo:** Modelar cómo se comunican los distintos elementos del sistema a lo largo del tiempo.
- **Herramienta / Tipo de diagrama:**
 - PlantUML usando **diagramas de secuencia** (PlantUML: *Sequence Diagram* <https://plantuml.com/es/sequence-diagram>).
- **Alcance mínimo:**
 - Seleccionar al menos **1 caso de uso** que implique interacción entre:
 - Usuario (actor).
 - **FrontEnd** (por ejemplo: aplicación web, SPA, cliente).
 - **BackEnd** (API, lógica de negocio, base de datos).
 - Representar:
 - Mensajes entre actor y FrontEnd.
 - Llamadas del FrontEnd al BackEnd.
 - Acceso a datos u otros servicios internos si aplica.

Actividad 4 – Diagrama de Componentes (Component Diagram)

- **Objetivo:** Definir la **arquitectura lógica** del sistema diferenciando claramente **FrontEnd** y **BackEnd**.
- **Herramienta / Tipo de diagrama:**
 - PlantUML usando **diagramas de componentes** (PlantUML: *Component Diagram* <https://plantuml.com/es/component-diagram>).
- **Alcance mínimo:**
 - Identificar y representar los **componentes del FrontEnd**, por ejemplo:
 - Aplicación cliente, módulos, páginas principales, etc.
 - Identificar y representar los **componentes del BackEnd**, por ejemplo:
 - API REST, microservicios, módulos de negocio, módulo de acceso a datos, etc.
 - Incluir:
 - Interfaces/proveedores de servicios.
 - Relaciones/dependencias entre componentes de FrontEnd y BackEnd.

Actividad 5 – Modelo de Datos de Intercambio en JSON (JSON Diagram)

- **Objetivo:** Definir los **formatos de datos en JSON** que intercambia el sistema (por ejemplo, entre FrontEnd y BackEnd).
- **Herramienta / Tipo de diagrama:**
 - PlantUML usando el **diagrama JSON** (PlantUML: *JSON* <https://plantuml.com/es/json>).
- **Alcance mínimo:**
 - Elegir al menos **2 estructuras JSON relevantes** del proyecto, por ejemplo:
 - Petición y respuesta de un endpoint importante.
 - Representación de una entidad de negocio principal (pedido, usuario, reserva, etc.).
 - Modelar la estructura de esos JSON (propiedades, tipos de datos, elementos anidados, listas).

Actividad 6 – Diagrama IE (Entidad–Interrelación)

- **Objetivo:** Definir el **modelo de datos relacional** que utilizará el sistema.
- **Herramienta / Tipo de diagrama:**
 - PlantUML usando el **diagrama IE** (PlantUML: *IE Diagram* <https://plantuml.com/es/ie-diagram>).
- **Alcance mínimo:**
 - Identificar las **entidades principales** del sistema (tablas).
 - Definir:
 - Atributos relevantes de cada entidad.
 - Claves primarias.
 - Relaciones (1:1, 1:N, N:M) entre entidades.
 - Reflejar, al menos, las partes del modelo que se están utilizando en los casos de uso trabajados en este sprint.
- **Resultado esperado:**
 - 1 diagrama IE que represente el núcleo del modelo de datos del proyecto.

Entregables del sprint

El alumnado deberá entregar, como mínimo:

1. **Diagrama de casos de uso** (vista global del sistema, por cada historia de usuario un diagrama de caso de uso).
2. **Diagramas de actividad** (mínimo 5).
3. **Diagrama(s) de secuencia** (mínimo 3).
4. **Diagrama de componentes** (arquitectura FrontEnd y arquitectura BackEnd).
5. **Diagramas JSON** (mínimo 2 estructuras).
6. **Diagrama IE** (modelo de datos principal).

Todos los diagramas deben estar realizados con **PlantUML**. Pueden organizarse en un único documento o en varios ficheros, según se indique en la plataforma o por el docente.

La documentación tendrá que estar ubicada en la localización adecuada, es decir, si es información sobre una historia de usuario, pues en la historia de usuario, si es información sobre cómo estaría montado el FrontEnd, pues en el [README.md](#) del repositorio de código del FrontEnd.

Criterios de evaluación

| Descripción | Pts |
|-----------------------------|-----|
| Coherencia entre diagramas | 4 |
| Adecuación al proyecto real | 4 |
| Claridad y legibilidad | 2 |