



**Semana de prácticas**

**Proyecto**

**Curso 2022-2023**

**2º Desarrollo de Aplicaciones Multiplataforma**

**FUNDACIÓN SAN VALERO**  
**SEAS, Centro de Formación Abierta**  
**ZARAGOZA**

# Propuesta de trabajo

Se debe implementar una **aplicación web/API** que gestione información almacenándola y recuperándola de una **Base de Datos Relacional**. Se pensará en un supuesto real, se redactará y se creará la aplicación web/API/Android para gestionarlo.

El supuesto real que se prepare como proyecto deberá contener al menos **2 elementos** de los que se almacene información (lo que equivaldría a 2 entidades/clases en el modelo Entidad/Relación o Modelo de clases Java). A su vez, cada uno de esos 2 elementos/entidades deberá tener, al menos, 5 atributos (de tipos String, numérico, fecha y boolean).

Se proponen una serie de **Requisitos obligatorios** que deben ser implementados y otra serie de **funcionalidades** de entre las que se puede elegir hasta completar la nota.

Todo el desarrollo del proyecto se llevará a cabo utilizando un repositorio en **GitHub** y el gestor de Issues de que dispone para registrar y gestionar las diferentes incidencias.

Además, se realizará un seguimiento exhaustivo de las tareas realizadas por cada componente del grupo utilizando **Trello** (<https://www.trello.com>).

## Requisitos (1 pto cada uno, obligatorios)

- **Modelo de datos:** Prepara un proyecto de aplicación web con Spring Boot para definir el modelo de datos del enunciado y poder trabajar con una base de datos MySQL. Prepáralo todo también para disponer de un fichero log que permita almacenar las trazas de ejecución de la aplicación.
- **Funcionalidad (Aplicación web):** La aplicación web debe listar el contenido de, al menos, un par de tablas de la base de datos en dos páginas diferentes. Se deberá maquetar utilizando HTML + PHP + CSS para presentar correctamente el contenido.
- **Funcionalidad (API):** Se debe diseñar (Open API 3.0) e implementar una API que permita, al menos, realizar las operaciones básicas (registro, listado, modificación, borrado) sobre las 2 entidades del modelo de clases. Además, se deberá realizar una operación que permita ver los detalles de un elemento determinado.
- **Funcionalidad (Datos relacionados):** Define algún tipo de relación entre las clases del model (aunque no la utilices todavía en ninguna de las operaciones desarrolladas en esta primera parte)

- **Funcionalidad (Android):** Desarrolla una aplicación Android que se comunique con la API implementada (utilizando Retrofit) para mostrar, al menos, un listado y la vista detalle de algún elemento del mismo.

## Otras funcionalidades (1 pto cada una)

- **Funcionalidad (Android II):** Amplía la aplicación de Android para que los usuarios puedan registrar, modificar, eliminar y buscar información haciendo uso de la API implementada.
- **Funcionalidad (Android III):** Utiliza Material Design para diseñar algún aspecto de la aplicación.
- **Funcionalidad (Javascript):** Utiliza Ajax para revitar recargar la página cuando realices alguna acción como registrar, eliminar o modificar algún elemento de la misma.
- **Funcionalidad (API II):** Amplía la API con 2 operaciones donde se trabaje con la relación que hay entre las clases del modelo
- **Funcionalidad (Web II):** Amplía la aplicación web para que sea posible registrar información (mediante formularios) y eliminar la existente.
- **Funcionalidad (Programación reactiva):** Crea una pequeña aplicación utilizando programación reactiva que permita visualizar el contenido de al menos un par de operaciones de la API
- **Funcionalidad (seguridad):** Implementa un sistema de autorización mediante token JWT para la API.
- **Git Flow:** Crea un repositorio en GitHub y gestiona el desarrollo del proyecto utilizando la metodología Git Flow y registrando/gestionando las incidencias encontradas en el gestor de issues de GitHub.
- **Inglés:** Traduce todos los textos de la aplicación web para que ésta se muestre completamente en inglés.

## Presentación

La última parte del sábado estará reservada para presentar los proyectos. A lo largo de la misma se establecerá un orden de presentación que os será comunicado junto con la hora aproximada de la misma.

Cada grupo contará con un máximo de 5 minutos para exponer y presentar todo el trabajo realizado para lo que se recomienda preparar algún tipo de presentación (PowerPoint o similar) así como dejarlo todo listo para realizar una demo en directo de las funcionalidades implementadas. Con respecto a este último punto, se recomienda dejar preparados algunos videos que sirvan como demo en caso de que se produzca algún problema que impida realizarla en directo.

# Entrega

La entrega se hará indicando la URL del repositorio de GitHub donde se encuentre TODO el material que se quiera entregar (Proyecto, presentación, documentación, . . .). Recordad dejar configurada como rama por defecto aquella donde se encuentre la última versión de todo el proyecto.

**Tened en cuenta que no se admitirá ningún documento entregado en formato doc o docx.**