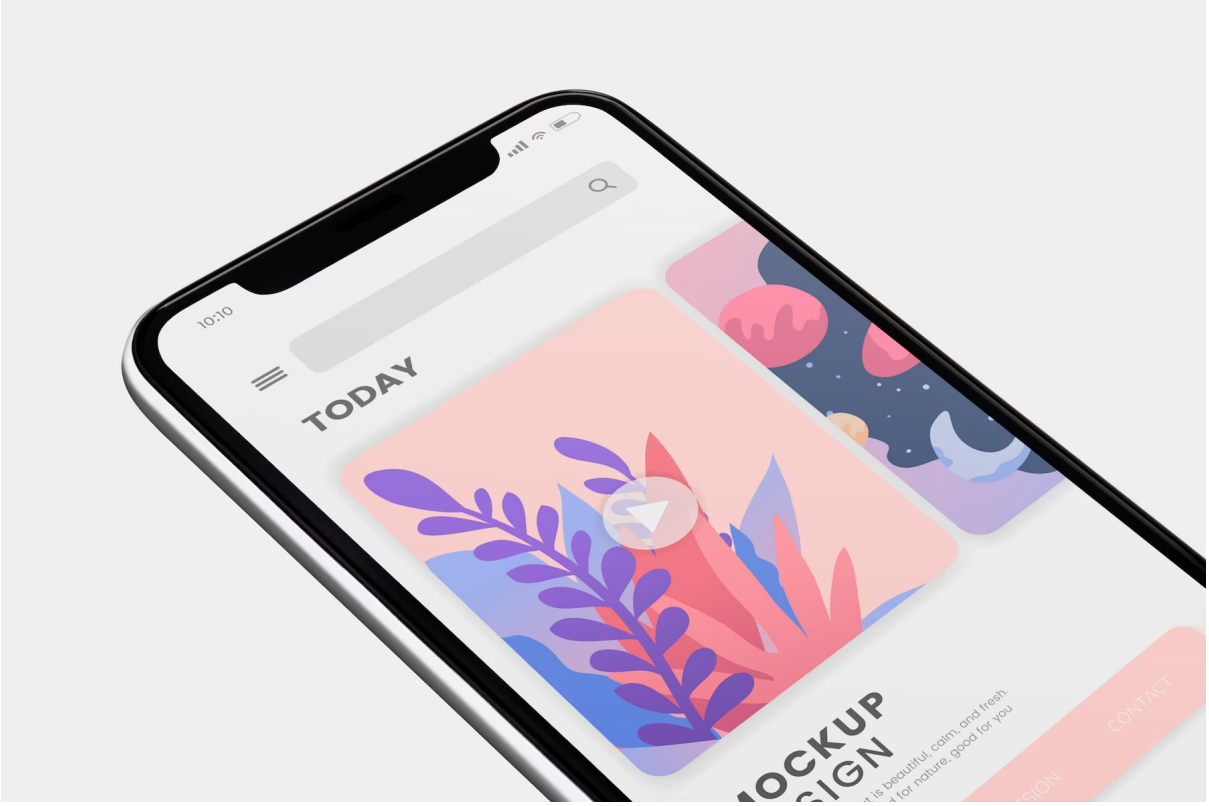
salesianos TRIANA

Colegio San Pedro

PROYECTO INTERDISCIPLINAR

Desarrollo de una app con Flutter conectada con una API REST implementada con Spring Boot



PROYECTO INTERDISCIPLINAR 2º DAM

PSP, AD, PMDM y DI

curso 22-23

C/ CONDES DE BUSTILLO 17 | 41010 SEVILLA | T 954 331 488 | https://triana.salesianos.edu

1. Introducción.

Este proyecto, que se realizará de forma individual, consiste en la implementación de una aplicación móvil con Flutter, que estará intercomunicada con una API desarrollada con Spring Boot, y que incluirá los elementos tecnológicos que se describen a continuación.

2. Elementos tecnológicos

2.1. Flutter

De la aplicación móvil se espera que:

- Siempre que sea posible, se utilizará el patrón BLoC.
- Incluya un mecanismo de registro de usuario, login y logout. Al implementar la autenticación con JWT, se debe de poder almacenar el token de acceso y/o de refresco en el teléfono, para reutilizarlo en posteriores peticiones.
- Si se estima conveniente, también debe de mostrar una pantalla con el perfil del usuario logueado.
- Debe incluir los listados necesarios para mostrar los datos de la aplicación. Desde los listados, se debe tener acceso a la pantalla de detalle, mostrando más ampliamente los datos de la entidad en cuestión.
- En alguno de los listados se podrá realizar búsqueda en base a uno o varios criterios y ordenación de resultados.
- La aplicación debe incluir formularios para el alta de información (además del formulario de registro).
- Se incluirá algún mecanismo de acción, tipo marcar como favoritos, me gusta o likes, comentarios o valoración numérica, …
- Se debe incluir algún mecanismo de subida de ficheros (vía la galería o la cámara del dispositivo).
- Se valorará positivamente el poder realizar algunas de las funcionalidades de la app sin que el usuario esté logueado, y otras que solamente se permitan si el usuario está logueado, o si además tiene un rol específico.

2.2. Spring Boot

De la API REST se espera que:

- Incluya el mecanismo de seguridad (autenticación y autorización) basado en JWT. Se deben tener varios roles de usuario (ADMIN, MANAGER, USER, UPLOAD, …)

- Utilice como base de datos Postgresql en un contenedor de docker levantado con Docker Compose (no es obligatorio, pero se pueden tener dos perfiles de ejecución: dev, con H2, y prod, con Postgresql).
- Incluya una gestión unificada de errores, utilizando excepciones propias allá donde la lógica de negocio lo necesite.
- Haga uso de la validación de las entradas, proporcionando los mensajes de error de forma externalizada (fichero de *properties*); también se deben crear los validadores personalizados donde los proporcionados por las librerías no lleguen a cubrir las necesidades de la lógica de negocio.
- Tenga un modelo de datos que pueda cubrir las necesidades de la app, incluyendo, además del usuario, alguna/s entidad/es más que se puedan relacionar con el mismo.
- Que defina los mecanismos de consulta necesarios para dar respuesta a las necesidades de la app, incluyendo siempre los listados de forma paginada.
- Incluya endpoints para la subida y manejo de ficheros.
- Se agradece que los endpoints estén documentados para poder consultar la misma con swagger.
- Que no utilice las entidades como respuesta en los endpoints, y haga uso del patrón DTO, usando los mecanismos necesarios para minimizar el número de clases necesarias (herencia, Json Views, …)

2.3. Se valorará

- La implementación de otras funcionalidades que se añadan a las descritas anteriormente.
- Un buen diseño y experiencia de usuario.
- La implementación a través de un código limpio, ordenado y documentado.
- La originalidad, tanto a nivel de diseño, navegabilidad y experiencia de usuario, como de código.
- El manejo conveniente de errores provenientes de la API.

3. Metodología y organización del código

- No será obligatorio, pero sí recomendable, identificar las historias de usuario y usar algún mecanismos de organización de las mismas, que permita enumerar las tareas de cada una de ellas y verificar si está completada o no.
- Se tendrá al menos un repositorio de Github donde subir todo el código fuente. Si es más cómodo para el alumnado, se podrá dividir en dos repositorios: uno para flutter y otro para Spring Boot.
- Se pueden usar metodologías como git-flow para la gestión del código.

4. Temática del proyecto y alcance

La temática del proyecto será “libre”, con un alcance realizable pero que trate de incluir los elementos tecnológicos anteriormente mencionados en el apartado 2.

En la medida de los posible no habrá dos proyectos con la misma temática, para fomentar la originalidad del producto final.

Algunas posibles ideas para los proyectos podrían ser:

- Una red social enfocada a un sector concreto de la población: gente de una zona (Triana, Sevilla, …), de una condición (futboleros, fans del manga, …). Aquí tendríamos usuarios, que podrían publicar textos o imágenes, que podrían tener valoraciones o likes, …
- Una aplicación de chollos (ropa, productos tecnológicos, comida, …) que permita a los usuarios publicar chollos (¿organizados por categorías?) y que los demás los puedan votar, valorar o comentar.
- Una aplicación de alquiler de viviendas, donde los usuarios con rol de dueño puedan publicar viviendas en alquiler, y los usuarios con rol de inquilino las puedan consultar, marcar como favoritas, … (como es lógico, no es obligatorio incluir la tramitación del alquiler)
- Una aplicación de comunicación con los padres de bebés de una guardería. El personal de la guardería puede publicar, para cada bebé, si ha comido bien o mal, si se le ha cambiado el pañal o si ha dormido, y los padres pueden ver solamente los datos de su bebé, y no de otros.
- Una galería de arte, donde los usuarios de tipo artista pueden publicar sus obras, y los usuarios de otro tipo las pueden consultar, marcar como favoritas, valorar, comentar, …
- Una aplicación de compra-venta de coches de segunda mano, donde los usuarios con rol de tipo vendedor podrán publicar coches, y los usuarios de tipo comprador los podrán visualizar, filtrar, marcar como me gusta, valorar, … (como es lógico, no es obligatorio incluir la gestión de la venta).
- Otra idea similar que se te ocurra, que cumpla con los elementos tecnológicos solicitados, pero que sea acotada y asumible.

La temática y alcance del proyecto se comentará con el profesor y se registrará en el formulario correspondiente publicado en Google Classroom.

5. Entrega y calificación

La fecha de entrega del trabajo será antes de las 14:30 del viernes 24 de Febrero, a través del repositorio (o repositorios) de código.

El trabajo producirá una calificación en varios referentes de evaluación en los módulos de Programación Multimedia y Dispositivos Móviles, Desarrollo de Interfaces, Programación de Servicios y Procesos y Acceso a Datos.
Para calificar, se utilizará una rúbrica de evaluación, que quedará publicada en Google Classroom en los próximos días.

Es importante resaltar que centrarse solamente en una parte (backend o frontend) puede significar que la calificación en los otros módulos sea negativa, por lo que es mejor plantear un enfoque transversal a la hora de desarrollar.