

# Trabajo práctico especial: Desarrollo de una app en Flutter

## 73.21 - Desarrollo de Aplicaciones Móviles Multiplataforma

### Objetivo

Desarrollar una aplicación móvil multiplataforma utilizando Flutter que permita a los usuarios buscar oportunidades de voluntariado cercanas a su ubicación y apuntarse a ellas.

### Requerimientos funcionales

#### Autenticación de usuarios

Los usuarios deben poder registrarse e iniciar sesión con su correo electrónico y contraseña.

#### Exploración de oportunidades de voluntariado

Los voluntariados cuentan con la siguiente información:

- Imagen.
- Tipo de voluntariado: lo que en el diseño se ve como acción social.
- Título del voluntariado.
- Misión o propósito
- Detalle de la actividad.
- Ubicación: coordenadas geográficas.
- Dirección.
- Requisitos para participar del voluntariado: Texto enriquecido (puede ser markdown) con información sobre los requerimientos para anotarse en un voluntariado.
- Fecha de creación del voluntariado.
- Vacantes disponibles.

Los usuarios deben poder explorar las diferentes oportunidades de voluntariado disponibles ordenadas por los siguientes criterios:

- Cercanía con el usuario: se le pedirá al usuario que permita a la app obtener su ubicación y, en caso de que se acepte, se ordenarán los voluntariados por cercanía.
- Fecha de creación del voluntariado: de más nuevo a más viejo.

Los voluntariados pueden ser marcados como favoritos (dato que se persiste en el backend) y el ícono de ubicación en la card del voluntariado debe abrir la aplicación Google Maps mediante un deep link con las coordenadas seleccionadas.

Para la búsqueda de voluntariados, se deben tener en cuenta como campos de búsqueda al título, la misión o propósito y el detalle de la actividad

## Detalle de oportunidades de voluntariado

Los usuarios deben poder ver el detalle de las oportunidades de voluntariado.

En caso de no implementar la integración con Google Maps, se utilizará el widget que contiene el mapa (sin el mapa), el cual será tapeable y llevará a la aplicación Google Maps con las coordenadas seleccionadas.

## Postulación a oportunidades de voluntariado

Los usuarios deben poder postularse a la oportunidad de voluntariado de su interés. La postulación sólo será posible si el usuario tiene completo el perfil, no está postulado en ningún otro voluntariado y todavía quedan cupos disponibles.

## Perfil de usuario

Los usuarios deben poder ver y completar su perfil.

## Noticias

Los usuarios deben poder ver un listado de novedades, también ordenadas por fecha de creación. Las novedades deben poder ser compartidas. El contenido que se compartirá será la imagen de la novedad y el subtítulo de la misma.

## Maps

La integración con google maps, o sea, que se muestre el mapa dentro de la app, **no** es un requerimiento base de la aplicación. Solo lo implementará el grupo al que le toque dicha funcionalidad especial.

## Requerimientos no funcionales

### Monitoreo y eventos

Se tiene que implementar monitoreo de errores (crashlytics o equivalente) y seguimiento de métricas de usuario mediante el registro de eventos. Para los eventos, elegir 3 acciones de los usuarios que se consideren pertinentes y documentar el porqué de la elección.

### Seguridad

La app debe utilizar protocolos de comunicación seguros y no se deben exponer API Keys en el repositorio. Además debe estar actualizada hasta la última patch version disponible de Dart y Flutter al menos un mes antes de la entrega.

## Privacidad

La aplicación debe pedir los permisos necesarios en ios (app tracking transparency) para el trackeo de eventos.

## Usabilidad

Si bien la mayor parte recae en el diseño, se tendrán en cuenta aquellas interacciones que queden fuera del diseño que hacen a una buena experiencia de usuario.

## Especificaciones Técnicas

- Design System: Deberán implementar un Design System provisto en Figma.
- State Management: Deberán implementar y documentar la elección de un sistema de manejo de estado.
- Deep Links: Deberán implementar un sistema de ruteo que permita el uso de deep links.
- Integración con Backend: La aplicación deberá integrarse con un backend que contenga la información de los voluntariados, novedades y de los usuarios y sus interacciones con los voluntariados. Documentar la elección de dicho backend.
- Tests: Deberán escribir tests unitarios y golden tests.

## Fechas de Entrega

- Check-in: 11 de octubre
- Entrega Temprana: 13 de noviembre
- Entrega Tardía: 17 de noviembre
- Presentación: 22 de noviembre

La entrega temprana permite corregir errores que surjan durante evaluación funcional de la aplicación, con tiempo hasta la entrega tardía.

## Evaluación

Se evaluará tanto el funcionamiento de la aplicación desde el punto de vista del usuario como la calidad del código, teniendo en cuenta la mantenibilidad, la seguridad y el correcto uso del framework Flutter.

# Funcionalidades especiales

## Feature flagging

Se deberá utilizar feature flagging para habilitar y deshabilitar al menos dos funcionalidades de la aplicación y una vez para realizar AB testing.

## Google Maps

Se implementará la funcionalidad diseñada en Figma que permite al usuario ver los voluntariados en la vista de mapa.

## Push notifications

Se deberán implementar push notifications que se puedan recibir cuando la aplicación está en foreground para los siguientes casos y que contengan deep links que te lleven a la parte correspondiente de la app:

- Cambios en la postulación a un voluntariado. Deep link que te lleva al voluntariado.
  - Aceptación de la postulación
  - Rechazo de la postulación
- Noticia nueva. Deep link que lleva a la noticia.

## Pull to refresh

Se implementará la capacidad de recargar la lista de voluntariados, un voluntariado en particular y la lista de novedades. Personalizar el scrolling widget con los colores del design system.

## Camera

Se implementará el uso de la cámara in app para obtener la foto de perfil del usuario.

## Internationalization (i18n)

Se preparará la app para su traducción completa al inglés (se puede elegir otro idioma si se prefiere). Se agregará un campo extra al voluntariado que represente el costo del voluntariado, el cual se agregará antes del botón de postulación en el detalle del voluntariado junto con la fecha de creación del voluntariado. Estos campos (costo y fecha) deberán ser internacionalizados también.

## Real time

Se implementará sobre el campo de cupo. La aplicación deberá recibir cambios en real time del cupo de los voluntariados y actualizar la UI de manera correspondiente, cambiando la card del voluntariado o la pantalla de detalle de voluntariado en caso que corresponda.

## Offline support

Se agregará al estado de la aplicación el estado de la conectividad a internet, teniendo un efecto en la ui y en las funcionalidades soportadas.

## Accesibilidad

Se hará foco en darle soporte a usuarios con visibilidad reducida para que puedan utilizar la aplicación de manera eficiente