

RETO DE APLICACIÓN BOOTCAMP MOBILE

El reto consiste en poder crear una app (App Sophos) que cumpla con todos los requerimientos expuestos en el documento “Reto Sophos Mobile”. En este documento se especifican las relaciones de funcionalidades expuestas con un respectivo ejemplo de interfaz de usuario a crear. Estas están separadas por Historias de Usuario(HU) las cuales se van a estar evaluando con sus criterios de entrega.

También indica cuál de los servicios de nuestra API se van a usar para cada funcionalidad. El archivo “EspecificaciónServiciosSemillero.json”, con la URL “<https://6w33tkx4f9.execute-api.us-east-1.amazonaws.com/>” es la API que se va a usar para la información de las funcionalidades de la app, pueden verla en un IDE o en el editor de swagger (<https://editor.swagger.io/>).

Les recomendamos que prueben cada uno de los end points de la API en “postman”, para que se familiaricen de cómo realizar los pedidos del consumo del API.

Se debe de usar los colores indicados en la herramienta de Figma para el desarrollo de los componentes de la aplicación. Para poder visualizar las características de cada componente expuesto en Figma se debe de iniciar sesión, si no se tiene una cuenta en Figma, pedimos el favor de crear una para tener el acceso necesario. Link del Figma (<https://www.figma.com/file/EUdoKVxvTd7eTEr3n5obdk/Reto-Bootcamp-Mobile-Sophos-2022?node-id=0%3A1>). Esta es una guía para la creación de las pantallas.

Si necesitan descargar un ícono se debe de exportar en formato .SVG para la importación en el Android studio o XCode.

Se encuentran expuestas unas **HU Opcionales**, estos requerimientos mejoran la experiencia de usuario y demuestran un compromiso adicional al momento de desarrollar el reto.

- **Requerimientos no funcionales:**

- .1. Arquitectura:**

- Es de suma importancia que la App se realice con la arquitectura MVVM (Model View ViewModel) este es un criterio que se va a evaluar en conjunto a las HU ya expuestas.

- .2. Test Unitarios (Opcional)**

- **La aplicación debe implementar test unitarios utilizando la librería Junit y las herramientas nativas de Android. Estas pruebas deben ser aplicadas a los ViewModels.**

- .3. Idioma de desarrollo**

Todo el desarrollo deberá realizarse en idioma inglés, es decir, el nombramiento de variables, clases, métodos, etc. Por ejemplo para la clase “Usuarios” debe ser una variable “Users”.

- **Criterios de aceptación**

- Los proyectos se deben desplegar exitosamente en ambiente local para su respectiva sustentación.
 - El código de fuente final de los proyectos debe estar versionado y alojado en GitHub.
 - La UI del proyecto debe ofrecer una buena experiencia de usuario.
 - Se evaluará la implementación de todos los requerimientos funcionales y no funcionales obligatorios, es decir, que no estén señalados como opcionales.