

ANEXO EXAMEN DE CERTIFICACIÓN

Plan de Estudio	Desarrollo Aplicaciones Android Trainee v.2
Tiempo	6 horas
Anexo	Wallet

Caso: “wallet”

Una Entidad Bancaria, ha visto la luz en Wallet(billetera Virtual) de La cual actualmente sólo tenemos el prototipo de diseño. A pesar de ello, al equipo creativo le ha parecido interesante, una posible implementación de esta app, Para el manejo de cuentas y pagos de distintos usuarios. Por lo que, en esta implementación, nos detallan algunos puntos que, los cuales se podrían desarrollar, para poder darle vida a este prototipo.

.

Alcance

Para comenzar con el desarrollo se va a Implementar un mínimo producto viable, donde Nuestros posibles clientes Puedan observar algunas de las funcionalidades de Wallet.

La API desde donde provienen los datos es limitada y cuenta solo con la información mínima para poder mostrarla en la app Android. Por este mismo motivo, aún no se termina de crear ni implementar una forma de recopilar todos los datos de los usuarios, o procesar nuevas funcionalidades.

En primera instancia, se solicita generar una aplicación Android que consuma los servicios proporcionados por la API Clientes. Se solicita que este mínimo producto viable permita con el tiempo poder escalar la aplicación sin mayores inconvenientes.

Características generales:

- Considerar que para esta primera versión se busca tener una gran cobertura de dispositivos manteniendo los costos de mantención bajos, la API mínima es 27 y el target 33.
- La aplicación no cuenta con autenticación de usuario(Login).
- Consumir un servicio **REST** con **2 endpoints**:
 - La primera API entrega una colección de **Usuarios**.
 - La segunda entrega la información de un **Usuario** dado un identificador.
- Se necesita que funcione de forma offline, por lo que se debe guardar en una base de datos local del dispositivo(Room).
- Primera pantalla: Listado de Usuarios disponibles
- Segunda pantalla: Detalle de un Usuario. Se navega a esta pantalla al seleccionar un Usuario del listado.
 - Además del detalle, debe tener un botón para enviar un correo con los datos del usuario, (Este deberá ejecutar un *Intent* implícito a una app de correo electrónico).

Código, arquitectura y dependencias

Dadas las características de este proyecto, la arquitectura debe permitir que la aplicación funcione sin conexión a internet y que se puedan agregar funcionalidades de forma ordenada, es por eso que se requiere cuidar un código legible y que use específicamente la siguiente arquitectura:

- La arquitectura puede ser **MVP-LiveData-ROOM** o **MVVM-LiveData-ROOM**.
- Debe guardar los datos en la persistencia del teléfono (ROOM) y mostrarlos en las vistas correspondientes. (debe guardar el listado de productos y también aquellos que selecciona para ver su detalle).
- Puede utilizar los lenguajes de programación **Kotlin** o **Java**.
 - Si utiliza Kotlin, debe usar Coroutines para manejar el trabajo asíncrono, según corresponda.
 - Si utiliza Java, debe usar alguna alternativa como executors para manejar el trabajo asíncrono, según corresponda.
- Utilizar paquetes representativos, respetar convenciones y estilos de codificación (indentación, reutilización, legibilidad, convenciones de nombre, comentarios cuando sea necesario)
- Se requiere utilizar una actividad y múltiples fragmentos.
- Se recomienda utilizar la biblioteca [navigation](#) para navegar entre fragmentos.
- Todos los textos que no sean obtenidos a partir de la API REST deben ser traducibles.

Sobre las dependencias:

- Los request HTTP deben ser realizados utilizando la librería **Retrofit**.
- Para mostrar imágenes puede utilizar cualquier biblioteca, por ejemplo: Glide, Picasso, Coil.

Diseño del Aplicativo

El diseñador es nuevo en el equipo, por lo que para la primera versión de la aplicación no ha podido traspasar el diseño a un programa que le entregue las medidas en formato Android. Por lo que los tamaños y distancias pueden no ser exactos en esta primera versión. Para evitar confusiones, el jefe de proyecto ha decidido enumerar las especificaciones que no pueden faltar:

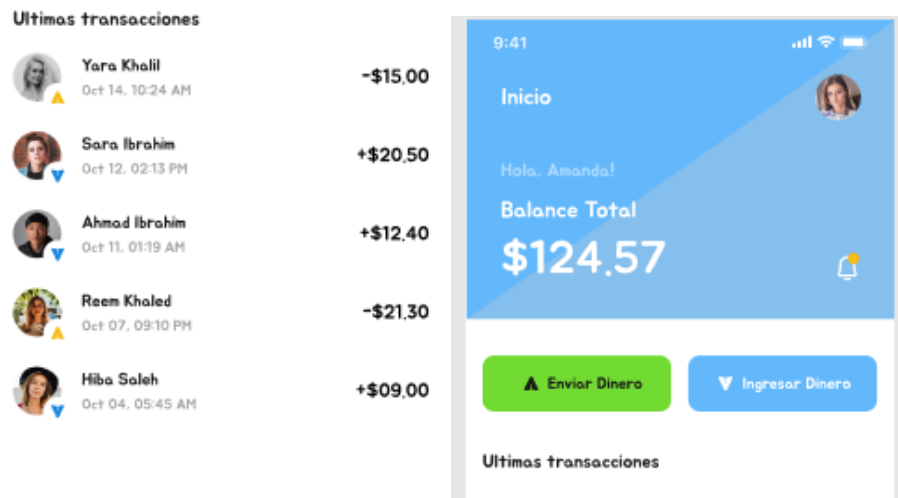
- Use la siguiente paleta para los colores de la aplicación:

#1976D2	#2196F3	#BBDEFB	#FFFFFF
DARK PRIMARY COLOR	PRIMARY COLOR	LIGHT PRIMARY COLOR	TEXT / ICONS
#FF5722	#212121	#727272	#B6B6B6
ACCENT COLOR	PRIMARY TEXT	SECONDARY TEXT	DIVIDER COLOR

- No se especifican los tipos de layouts a utilizar, por lo tanto, es responsabilidad del desarrollador. Debe utilizar su criterio manteniendo una correcta visualización de los elementos.
- Se recomienda que la segunda pantalla tenga la posibilidad de hacer scroll para visualizar el contenido de forma correcta.
- El área de diseño todavía no ha estandarizado la línea gráfica de los íconos, pero debe al menos agregar un ícono usando vectores.
- Para los layouts se recomienda utilizar [Material Design](#)

A continuación, se presentan algunos ejemplos de pantallas que cumplen con lo requerido.

****Solo referenciales, puede cambiar el diseño según estime conveniente en base al aplicativo***



Listado De usuarios

Detalle cuenta de Usuario.

Endpoints y Modelos de Datos

Ambos endpoints se deben acceder a través del verbo HTTP de request GET.

El primer endpoint es para obtener una lista de Super Héroes

- 1) **<https://wallet-afuno76sn-talento-projects.vercel.app/wallet>**
Este endpoint retorna con un listado de usuarios

A continuación, un ejemplo con 2 elementos en el listado:

```
[
  {
    "id": 1,
    "nombre": "Josefina Martinez",
    "pais": "ALEMANIA",
    "imagenLink":
"https://static6.depositphotos.com/1008939/631/i/950/depositphotos_6318561-stock-photo-smiling.jpg",
    "cuenta": "Chequera Electrónica",
    "saldo": 2,
    "depositos": true
  },
  {
    "id": 2,
    "nombre": "Pedro Carcuro",
    "pais": "Chile",
    "imagenLink": "https://www.12del9.cl/wp-content/uploads/2017/09/pedro-carcuro.jpg",
    "cuenta": "Cuenta Rut",
    "saldo": 150,
    "depositos": false
  },
  {
    "id": 3,
    "nombre": "Raquel Argandoña",
    "pais": "Chile",
    "imagenLink":
"https://www.chilevision.cl/chilevision/site/artic/20230104/imag/foto_0000000120230104153614/raquel.jpg",
    "cuenta": "Cuenta de Ahorro",
    "saldo": 5.999,
    "depositos": false
  },
  {
    "id": 4,
    "nombre": "Fredy Turbina",
    "pais": "Dinamarca",
    "imagenLink": "https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQrXTNJRIkkMuASDUDDLBr31bbVHEFWi9XN56H54eYl4LZxtFYbbm
Jic07Nw&s",
    "cuenta": "Chequera Electrónica",
    "saldo": 3.9999,
    "depositos": true
  },
  {
    "id": 5,
    "nombre": "Victor Barra",
    "pais": "Peru",
    "imagenLink":
"https://media.revistagq.com/photos/621343d2c789a63cc825a58b/16:9/w_1920,c_limit/virgen%2
0a%20lo%2040.jpeg",
    "cuenta": "Cuenta a Plazo",
    "saldo": 4.99999,
    "depositos": true
  },
  {
    "id": 6,
    "nombre": "Marcelo Salaz",
    "pais": "Chile",
    "imagenLink":
"https://www.latercera.com/resizer/et1Y9I_35NcsOVStKTjLQjAw9YY=/900x600/smart/cloudfront-
us-east-1.images.arcpublishing.com/copesa/NYI5DD3YOBdYLC04N5EROBAZ7E.jpeg",
    "cuenta": "Chequera Electronica",
    "saldo": 6.999998877,
    "depositos": true
  }
]
```

El segundo endpoint corresponde al detalle y se accede indicando el ID específico:

`https://wallet-afuno76sn-talento-projects.vercel.app/wallet/5`

- 2) En esta respuesta se encuentra tan solo 1 objeto usuario que corresponde al ID. A continuación, el resultado cuando se indica el id número 5

```
{
  "id": 5,
  "nombre": "Victor Barra",
  "pais": "Peru",
  "imagenLink":
"https://media.revistagq.com/photos/621343d2c789a63cc825a58b/16:9/w_1920,c_limit/virgen%20a%20lo%2040.jpeg",
  "cuenta": "Cuenta a Plazo",
  "saldo": 4.99999,
  "depositos": true
}
```

Como habrá notado hay 1 atributos que es booleano, pero en el diseño debe aparecer como texto o como un icono.

- traducción
 - true: Cuenta con disponibilidad de deposito
 - false: Cuenta no disponible para transferencias.

Funcionalidad de contacto

Cuando el equipo de marketing está viendo el detalle de algún usuario, al hacer click en un botón del detalle, tiene que tener la posibilidad de enviar un correo electrónico al usuario, En el cual se debe consultar si desea ser contactado.

Destinatario: usuarioSeleccionado@hotmail.com

- Asunto: Formulario de Contacto {nombre}
- Mensaje:

“Hola

Somos parte del equipo de contacto de Wallet, Te animas a que podamos contactarte, para poder recibir información importante.

Número de Contacto: _____

Gracias.”

Recuerde reemplazar lo indicado entre paréntesis cursivos por la información correspondiente al Usuario seleccionado. No incluya los paréntesis cursivos en correo.

No es necesario que el usuario llene el Pais de ante mano, conserve los guiones bajos u otro símbolo que le indique al usuario que ahí tiene que escribir su número cuando esté viendo el mensaje en la aplicación de correos.

Testing

Preocupados por la calidad del código y estabilidad de la app, se ha definido un mínimo de test a llevar a cabo.

- Al menos un test unitario
- Al menos un test de instrumental.

Sugerencias:

- Test unitario que verifique la respuesta de los endpoints usando un servidor de pruebas como mockwebserver.
- Test instrumental que compruebe la persistencia de datos con ROOM.
- Test unitarios sobre cualquier