

ANEXO EXAMEN DE CERTIFICACIÓN

Plan de Estudio	Desarrollo Aplicaciones Android Trainee
Anexo	Caso Nova Era

Caso “Nova Era”

La empresa distribuidora de equipos móviles “Nova Era” necesita realizar una aplicación móvil Android. En esta primera instancia necesitan un mínimo producto viable, donde sus usuarios pueda observar un listado de teléfonos y seleccionar alguno.

Para esto, se ha formado un equipo de proyectos del cual usted forma parte como desarrollador Android.

En el siguiente documento, usted encontrará el briefing de la aplicación Android que usted tiene que desarrollar.

Alcance del proyecto

La primera tarea que llevó a cabo el jefe de proyecto fue definir el alcance de la transformación digital de la empresa “Nova Era”.

Tras analizar las operaciones, el jefe de proyecto determinó que se debe construir un mínimo producto viable, donde los posibles clientes puedan observar los equipos en venta y poder seleccionar alguno.

El equipo de back end logro poner en funcionamiento una API Rest con la información de los equipos y detalles de estos, pero aún no se termina de crear ni implementar una forma de recopilar datos de los usuarios.

En primera instancia, se solicita generar una aplicación Android que consuma los servicios proporcionados por el Back end. Dado el poco tiempo disponible, se solicita un mínimo producto viable, pero que permita con el tiempo poder escalar la aplicación sin mayores inconvenientes.

En esta primera versión, la app realizará lo siguiente:

- La app Android consumirá un servicio Rest, que le proporcionara los datos a mostrar.
- El servicio REST contará con 2 end points a consumir, uno que entregará una colección de equipos móviles y el segundo que entregará los detalles de los equipos dado un identificador.

- Considerando que para esta primera versión se busca tener una gran cobertura de dispositivos manteniendo los costos de mantención bajos, la API mínima es 23 y el target 30.

Lo que se espera que haga la aplicación es que cualquier usuario que la instale sin necesidad de autenticarse, pueda ver una lista de los equipos, ver los detalles del que seleccione y enviar un correo al área de ventas con información predefinida.

- La primera pantalla es una lista de equipos.
- La segunda pantalla muestra la información en detalle de un equipo.
- La segunda pantalla tiene un botón que al presionarlo envía el correo descrito.

Código, arquitectura y dependencias

El jefe de proyectos entiende que, aunque el alcance de la aplicación es inicial, la arquitectura que utilice le permitirá seguir expandiendo o modificando el proyecto de acuerdo a los resultados iniciales, es por eso que se requiere cuide un código legible y que use específicamente la siguiente arquitectura en conjunto con estas dependencias:

- Respetar las convenciones y estilos de codificación (indentación, reutilización, comentarios, legibilidad, convenciones de nombre, etc.)
- La arquitectura del aplicativo 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.
- Los request HTTP deben ser realizados utilizando la librería Retrofit.
- Puede utilizar Kotlin o Java.
- Si utiliza Kotlin, debe usar Coroutines para manejar el background thread, según corresponda.
- Si utiliza Java, debe usar alguna alternativa como executors para manejar el background thread, según corresponda.
- Utilice las bibliotecas que estime necesario para hacer testing.
- Al menos 1 test unitario.
- Al menos 1 test instrumental.





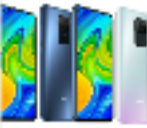

Diseño del Aplicativo

El diseñador fue contratado hace poco, 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:


#0097A7 DARK PRIMARY COLOR	#B2EBF2 LIGHT PRIMARY COLOR	#00BCD4 PRIMARY COLOR	#FFFFFF TEXT / ICONS
#FF5252 ACCENT COLOR	#212121 PRIMARY TEXT	#757575 SECONDARY TEXT	#BDBDBD DIVIDER COLOR

- Aunque los tamaños pueden variar, la diagramación de los layouts se debe mantener, esto incluye las filas de la lista y la segunda pantalla
- El área de diseño todavía no ha estandarizado la línea gráfica de los íconos, por lo que si le está tomando mucho tiempo encontrar uno, puede utilizar el que más le convenga. Para los íconos tiene que importar scalar vector graphics. Se recomienda utilizar el asistente de Android Studio.
- La primera pantalla es una activity y un fragment que contiene la lista de equipos móviles.
- La segunda pantalla es una scrolling activity y 2 fragments, uno para la cabecera colapsable y otro para el cuerpo. (Puede usar un fragmento para mostrar la pantalla de detalles, queda en sus manos decidir si es necesaria otra actividad o fragmento).
- Todos los textos que no sean obtenidos a partir de la API REST deben ser traducibles.
- Se adjuntan algunos ejemplos de pantallas tipo:


Plaplix		
		
Samsung Galaxy A21s 64GB \$ 157.250	Huawei Nova 7 SE 128GB \$ 347.760	Apple iPhone 7 32GB \$ 329.760
		
Galaxy Note 20 256GB \$ 757.750	Xiaomi Redmi 9 64GB \$ 167.750	Motorola Moto E6S 32GB \$ 131.760

Plaplix

HUAWEI NOVA 7 SE 128GB



Antes \$347.960
Ahora \$347
Acepta Crédito 

[Contactar Area de Ventas](#) 

Tamaño 6,5"
Densidad 405 ppi
Resolución de pantalla 1080 x 2400 pixeles

Solo referenciales, puede cambiar el diseño según estime conveniente.

End Points y Modelos de Datos

Ambos endpoints se deben acceder a través del verbo de request HTTP GET. El primer endpoint es para obtener una lista de los productos

1) <https://my-json-server.typicode.com/Himuravidal/FakeAPIdata/products/>

En esta respuesta se encuentran los productos agrupados por una colección donde cada producto tiene la data reducida:

```
{
  "id": 1,
  "name": "Samsung Galaxy A21s 64GB",
  "price": 167253,
  "image":
  "https://images.samsung.com/is/image/samsung/es-galaxy-a21s-sm-a217fzkoeub-262755098?$PD_GALLERY_L_JPG$"
}
```

Ilustración 1: Reduced equipos Model

A continuación, puede ver un ejemplo con 3 objetos:

```
[{
  "id": 1,
  "name": "Samsung Galaxy A21s 64GB",
  "price": 167253,
  "image":
  "https://images.samsung.com/is/image/samsung/es-galaxy-a21s-sm-a217fzkoeub-262755098?$PD_GALLERY_L_JPG$"
},
{
  "id": 2,
  "name": "Huawei Nova 7 SE 128GB",
  "price": 347760,
  "image":
  "https://consumer.huawei.com/content/dam/huawei-cbg-site/common/mkt/pdp/phones/nova7-se/img/mob/huawei-nova7-se-mob.png"
},
{
  "id": 3,
```

```
{
  "name": "Apple iPhone 7 32GB",
  "price": 323760,
  "image":
    "https://daisycon.io/images/mobile-device/?width=250&height=250&color=ffffff&mobile_device_brand=apple&mobile_device_model=iphone+7+32gb&mobile_device_color=silver"
}
```

Ilustración 2: Example Response

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

2) <https://my-json-server.typicode.com/Himuravidal/FakeAPIdata/details/1>

En esta respuesta se encuentra tan solo 1 objeto producto que corresponde al ID, la diferencia es que tiene todos los datos.

```
{
  "id": 1,
  "name": "Samsung Galaxy A21s 64GB",
  "price": 167253,
  "image":
    "https://images.samsung.com/is/image/samsung/es-galaxy-a21s-sm-a217fzkoeub-262755098?$PD_GALLERY_L_JPG$",
  "description": "Tamaño 6,5''\n Densidad 294 ppi\n Resolución de pantalla 720 x 1600",
  "lastPrice": 177253,
  "credit": true
}
```

Ilustración 3: Full product Model

Como habrá notado hay 1 atributos que es booleano, pero en el diseño debe aparecer como texto:

- credit
 - true: Acepta Crédito
 - false: Sólo Efectivo

Funcionalidad de Envío de Correo

Cuando el usuario está viendo el detalle de un equipo al hacer click en el fab tiene que enviarse un correo con la siguiente información pre llenada:

- Destinatario: info@novaera.cl
- Asunto: Consulta {PRODUCT_NAME} id {PRODUCT_ID}
- Mensaje:

“Hola

Vi la propiedad {PRODUCT_NAME} de código {PRODUCT_ID} y me gustaría que me contactaran a este correo o al siguiente número _____

Quedo atento.”

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

No es necesario que el usuario llene el número 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

El jefe de proyectos es sumamente ágil y quiere lanzar a producción la aplicación en lo mínimo viable para ver su factibilidad en el tiempo. Sin embargo, está preocupado por la calidad del código y estabilidad de la app, por lo que ha definido un mínimo de test a llevar a cabo.

Algunos tipos de test que se pueden implementar son:

- 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 función

Como mínimo debe ser 1 test unitario y 1 test instrumental.