TALENTO DIGITAL
INTELIGENCIA HUMANA

ANEXO EXAMEN DE CERTIFICACIÓN

Plan de Estudio	Desarrollo Aplicaciones Android Trainee
Anexo	Caso Elixir Games

Caso “Elixir Games”

La tienda de juegos físicos “ELIXIR GAMES”, tiene la necesidad de incursionar en la venta de juegos digitales, Para ello tienen una idea para un mínimo producto viable, lo cual sería la plataforma de entrada para poder probar este tipo de distribución. La idea principal es mostrar los juegos disponibles en una app móvil.

Se debe mostrar un pequeño catálogo de juegos digitales, para que los interesados puedan visualizar detalles y hacer llegar una petición de compra.

Esta app permitirá probar la idea por lo tanto se te ha encargado realizar un mínimo producto viable en forma de una app Android nativa, para sentar las bases de este proyecto, con la idea de que pueda escalar en un futuro.

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

Alcance del proyecto

El jefe de proyecto determinó que se debe construir un mínimo producto viable, donde los posibles clientes puedan observar los juegos en venta y poder seleccionar alguno, y hacer llegar esta información al departamento de ventas.

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 datos de los usuarios, o procesar las ventas.

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

Esto quiere decir que debe utilizar patrones de arquitectura escalables y buenas prácticas de la industria.

En esta primera versión, la app realizará lo siguiente:

- Debe consumir un servicio **REST**, desde donde provienen los datos a mostrar.
- El servicio REST cuenta con **2 endpoints**, uno que entregará una colección **Juegos** y el segundo que entregará los detalles de estos Juegos dado un identificador (id).
- 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 juegos disponibles, ver los detalles del que seleccione y enviar un correo al área de ventas con información predefinida.

- La primera pantalla es una lista de Juegos (Recyclerview).
- La segunda pantalla muestra la información en detalle del Juego seleccionado.
    - Debe tener un botón para simular la intención de compra (Este deberá ejecutar un intent implícito a una app de correo electrónico).

Código, arquitectura y dependencias

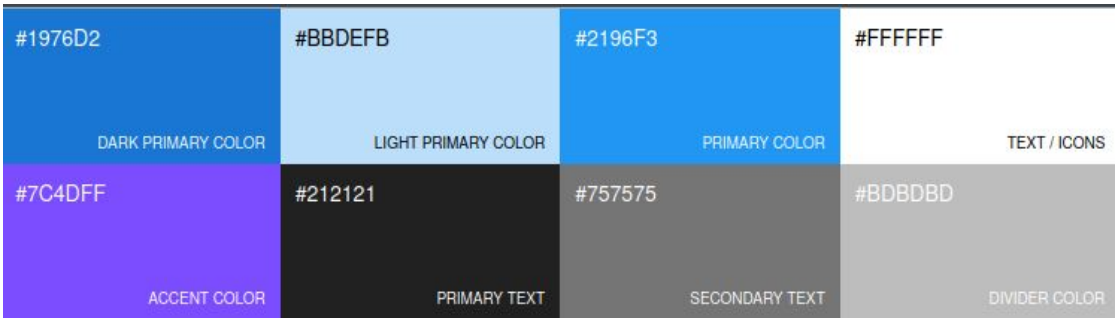
El jefe de proyectos entiende que, aunque el alcance de la aplicación es inicial, la arquitectura que utilice le permitirá seguir expandiendo o modificando el proyecto de acuerdo a los resultados iniciales, es por eso que se requiere cuide un código legible y que use específicamente la siguiente arquitectura 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 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 Runnables o Executors para manejar el trabajo asíncrono, según corresponda.
- Utilice las bibliotecas que estime necesario para hacer TEST.

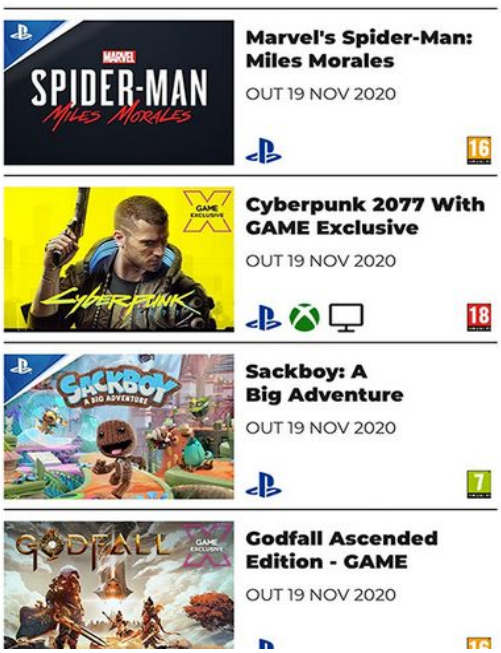
Diseño del Aplicativo

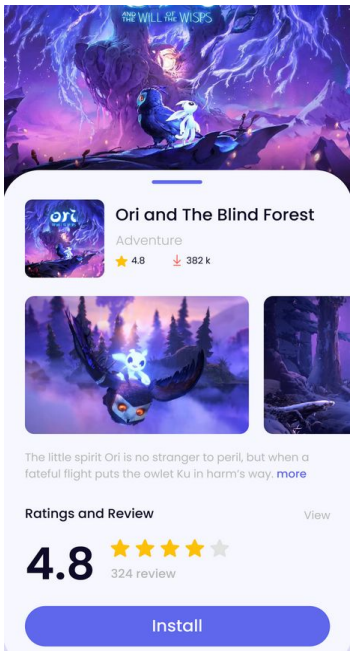
El diseñador fue contratado hace poco, por lo que para la primera versión de la aplicación no ha podido traspasar el diseño a un programa que le entregue las medidas en formato Android. Por lo que los tamaños y distancias pueden no ser exactos en esta primera versión. Para evitar confusiones, el jefe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

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



- Aunque los tamaños pueden variar, la diagramación de los layouts se debe mantener, esto incluye las filas de la lista y la segunda pantalla (Utilizar recomendaciones de Material Design).
- Se recomienda utilizar una actividad y múltiples fragmentos.
- No se especifican qué tipos de layouts debe utilizar, por lo tanto es responsabilidad del desarrollador, debe utilizar su criterio manteniendo como punto importante la correcta visualización de los elementos.
- Se recomienda que la segunda pantalla tenga la posibilidad de hacer scroll para visualizar el contenido de forma correcta.
- Todos los textos que no sean obtenidos a partir de la API REST deben ser traducibles.
- El área de diseño todavía no ha estandarizado la línea gráfica de los íconos, pero debe tratar de utilizar al menos 2 para mejorar el apartado visual de la app. Para los íconos tiene que utilizar vector graphics.
- Se adjuntan algunos ejemplos de pantallas tipo, pero basándonos en el tiempo, debe tomar decisiones que permitan realizar el proyecto y que se ajusten a los datos entregados por el servicio Rest.





Listado. Detalle.

Solo referenciales, Los campos no coinciden con los entregados por la API. Son solo una referencia y no es exigencia que sean iguales.

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 juegos disponibles.

1) **https://my-json-server.typicode.com/himuravidal/gamesDB/games**

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

```
{
    "id": 3498,
    "name": "Grand Theft Auto V",
    "released": "2013-09-17",
    "background_image":
"https://media.rawg.io/media/games/84d/84da2ac3fdfc6507807a1808595afb12.jpg",
    "rating": 4.48,
    "metacritic": 97
  }
```

Ilustración 1: Reduced Game Model

A continuación, puede ver un ejemplo con 3 objetos:

```
[
  {
    "id": 3498,
    "name": "Grand Theft Auto V",
    "released": "2013-09-17",
    "background_image":
"https://media.rawg.io/media/games/84d/84da2ac3fdfc6507807a1808595afb12.jpg",
    "rating": 4.48,
    "metacritic": 97
  },
  {
    "id": 4200,
    "name": "Portal 2",
    "released": "2011-04-18",
    "background_image":
"https://media.rawg.io/media/games/328/3283617cb7d75d67257fc58339188742.jpg",
    "rating": 4.62,
    "metacritic": 95
  },
```

```
  {
    "id": 3328,
    "name": "The Witcher 3: Wild Hunt",
    "released": "2015-05-18",
    "background_image":
"https://media.rawg.io/media/games/618/618c2031a07bbff6b4f611f10b6bcdbc.jpg",
    "rating": 4.67,
    "metacritic": 92
  }
]
```


Ilustración 2: Example Game Response

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

2) https://my-json-server.typicode.com/himuravidal/gamesDB/gameDetails/4200

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


```
{
  "id": 4200,
  "name": "Portal 2",
  "released": "2011-04-18",
  "background_image":
"https://media.rawg.io/media/games/328/3283617cb7d75d67257fc58339188742.jpg",
  "rating": 4.62,
  "metacritic": 95,
  "playtime": 11,
  "platforms": "Xbox One",
  "genres": "Shooter",
  "format": "Digital",
  "price": 11500,
  "lastPrice": 20500,
  "delivery": true
}
```


Ilustración 3: Full Game Detail Model

Funcionalidad de Envío de Correo

Cuando el usuario está viendo el detalle de un juego seleccionado, al hacer click sobre algún botón, tiene que enviarse un correo con la siguiente información pre llenada:

- Destinatario: ventas@elixirgames.cl
- Asunto: Consulta por Juego {NAME} id {ID}
- Mensaje:

> "Hola
>
> Vi el juego {NAME} de código {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 juego 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

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 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 función.