

Proyecto final Modulo 3

Plan Formativo	Nivel de Dificultad:
Desarrollo de Aplicaciones Móviles Android Trainee	
Nombre del proyecto: Proyecto Final de módulo 3	Tema: App Final Módulo 3.
Objetivo del proyecto:	Construir aplicaciones móviles Android utilizando lenguaje Java, patrones de diseño escalable y elementos de interfaz de aplicaciones nativas de acuerdo a las especificaciones entregadas.
Ejecución: Individual	
Descripción General del Proyecto	
<p>1. La empresa “Evil Corps” le solicita a usted desarrollar una aplicación Android para solucionar un problema en su flujo de trabajo. El rubro de la empresa es revisar entregas de apartamentos (departamentos), confirmando que los elementos básicos estén presentes en las entregas. La empresa necesita realizar un proceso de transformación digital, cambiando un proceso de revisión manual, por uno digital.</p> <p>2. El proceso a mejorar pertenece a la labor realizada por inspectores en terreno, ellos se encargan de revisar nuevos departamentos (apartamentos) entregados por las constructoras asociadas a la empresa y asignarles un puntaje según lo observado. Los inspectores revisan si están presentes elementos básicos como luces, elementos del dormitorio, baño o cocina.</p> <p>En una ficha de papel, van haciendo check a estos puntos, y al final realizan un cálculo de puntaje de entrega. La forma tradicional presenta varios inconvenientes, como la perdida de los comprobantes en papel y también errores en los cálculos asociados al puntaje.</p> <p>3. El puntaje se calcula bajo la siguiente formula:</p> <ul style="list-style-type: none"> a. Aspectos a evaluar: <ul style="list-style-type: none"> i. Luces del apartamento ok = 10, no ok = 0. ii. Elementos del baño ok = 40, no ok = 0. iii. Elementos de la cocina ok = 30, no ok = 0. 	

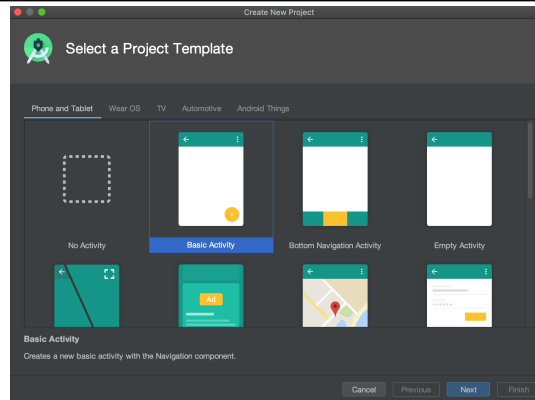
- iv. Elementos del dormitorio = ok 20, no ok 0.
 - v. Estado general de las terminaciones. Normal = 3, Regular = 2, Malas = 1.
4. Una vez que se obtienen los puntajes, se suman y se multiplican según lo siguiente.
- a. Luces, elementos de baño, elementos de cocina y dormitorio.
 - b. A la suma anterior se le realiza una multiplicación basándose en el estado de las terminaciones.
 - c. Ejemplo:
 - i. Luces ok = 10
 - ii. Elementos de dormitorio ko = 0.
 - iii. Elementos de cocina ok = 30.
 - iv. Elementos de baño ok = 40.
 - v. Estado de terminaciones Normal = 3.

$$(10 + 0 + 30 + 40) * 3 = 160.$$

5. Todos los puntajes inferiores a 130 debían ser informados al supervisor de forma inmediata, por llamado telefónico, pero regularmente no era posible la comunicación y se fallaba en este punto.
6. La empresa contratante quiere una solución para este problema, y se acordó la realización de una app móvil Android que pueda ser instalada en los teléfonos de los técnicos y que realice el proceso de check y calculo de puntaje de forma automática.
7. Durante esta **iteración(Periodo de entrega)** se le solicitará construir solo el **mínimo Producto viable** de la app. Por lo tanto algunas funciones como persistencia de datos y conexión a los servidores estará simulada con datos locales generados por la app en tiempo de funcionamiento.
8. Lo mismo ocurrirá con el proceso de envío de información al supervisor en caso de ser necesario, lo cual puede realizarse a través de correo electrónico, SMS o WhatsApp.
9. El cliente Desea que la app se llame "**Check Apartment**" o algo similar.

Descripción particular de la solución.

10. La aplicación debe poder ser instalado en dispositivos con API 23 o superior.
11. Debe seguir en la medida de lo posible algún patrón de arquitectura recomendado (MVP).
12. Existe un diagrama de pantallas inicial propuesto por la compañía, Pero puede ser modificado basándonos en la necesidad de los tiempos de entrega y decisiones técnicas.
13. La app debe tener dos actividades indicadas a continuación:
14. Pantallas de la app: 1. "LoginActivity".
 - a. Esta actividad servirá como punto de partida (Login activity), debe contener los siguientes elementos:
 - i. Un Campo para escribir un correo electrónico.
 - ii. Un Campo para escribir una contraseña.
 - iii. Un Button para ingresar.
 - iv. La validación en esta iteración será solo al password ingresado. Revisando si es igual a "123Pass".
 - v. Si es exitosa, debe llevar a una segunda actividad. Si no, debe indicar que el password no es el indicado, debe tener un contador de errores, si alcanza los tres intentos, debe bloquear el Button de ingreso. Teniendo que reiniciar la app para intentarlo otra vez.
 - vi. Para mostrar información al usuario puede usar Toast o Snackbar.
15. Segunda actividad:
 - a. La segunda actividad deberá ser el contenedor de dos fragmentos.
Hint: puede usar el template Basic Activity.



- b. En el primer fragmento deberá mostrar un listado de departamentos por revisar, esta información por el momento será entregada de forma estática (local en la app), pero en futuras iteraciones será enviada a través de un servidor, por lo tanto debe estar preparada para mostrar X cantidad de elementos.

Hint: utilizar un RecyclerView.

- c. Al interior del “item_view” del recyclerView debe mostrar los siguientes campos:

- i. Nombre del proyecto o edificio.
- ii. Unidad o número del departamento.
- iii. Dirección del proyecto.
- iv. Debe utilizar un LinearLayoutManager con orientación vertical para el listado.

- d. El segundo fragmento, deberá abrirse al seleccionar el departamento del listado inicial

Hint: Pasar el objeto al hacer click sobre el recyclerView.

- e. Este fragmento deberá contener los elementos para que el técnico pueda ir marcando un check sobre lo solicitado. (Estado de luces, estado de elementos de cocina etc.)

Hint: utilizar checkBox o Switch para estos elementos.

- f. Utilizar **RadioButtons** para seleccionar el estado general de las terminaciones.

- g. Es recomendable mostrar la imagen del proyecto en esta pantalla, puede cargar la imagen con la biblioteca externa Glide.

También debe indicar el nombre del proyecto y unidad la cual está revisando. (La URL será informada en el listado de datos estáticos).

- h. Este fragmento debe contener un TextView que debe indicar el total del cálculo realizado basándonos en lo seleccionado, el cálculo se debe ser actualizado cuando el técnico marque la opción en el radio Button(estado de las terminaciones).
 - i. Debe existir un Button guardar y otro llamado enviar alerta.
 - j. El botón enviar alerta debe solo activarse o permitir el click si el cálculo realizado es menor a 130 puntos, y si se cumple la condición debe realizar un **Intent** hacia una aplicación de correo electrónico o mensaje de texto presente en el sistema.
 - k. El Button guardar debe mostrar un Toast indicando el puntaje final obtenido.
16. Debe utilizar un sistema de control de versiones Git y un repositorio remoto para ir subiendo su trabajo y avance.

Requisitos Opcionales

- 1. Realizar Test unitarios al modelo que realiza el cálculo de puntaje de los departamentos.
- 2. Realizar Test unitarios al o los presentadores que utilice.

Consejos para comenzar

- 1. Realice un esquema o dibujo del modelo que representara los datos que utilizara la App.
- 2. Comience añadiendo al proyecto las actividades que necesitara.
- 3. Añada las dependencias del proyecto y realice un commit inicial en la rama master. Añada una segunda rama llamada development para ir realizando cada una de las tareas asignadas, cuando termine una tarea suba un commit al repositorio remoto.
- 4. Utilice ViewBinding para manejar el acceso a los elementos de las vistas desde Java.
- 5. Utilice la biblioteca de navegación para los fragmentos.

Requerimientos de los participantes

Conocimientos previos	Actitudes para el trabajo	Valores
<ul style="list-style-type: none">• Estructura base de proyectos Android.• Componentes básicos de diseño.• Manejo de activity y layout• Uso y manejo de eventos de botones.• MVP• Fragmentos• Intent	<ul style="list-style-type: none">• Cumplimiento de plazos• Buenas prácticas de codificación• Diseño y Estructura• Trabajo individual• Optimización del tiempo	<p>Tiempo de resolución.</p> <p>Enfoque al requerimiento.</p> <p>Estructura de Solución.</p>
Objetivo General de Aprendizaje	<p>El participante al finalizar el proyecto será capaz de:</p> <p>Diseñar y ejecutar código útil en Android basado en normas de y patrones de diseño.</p>	
Objetivos particulares	<ul style="list-style-type: none">- Resolución de problemas planteados.- Utilización de Android Studio.- Utilizar patron MVP.- Utilizar Intent explícitos e implícitos.- Utilización de fragmentos.- Utilizar RecyclerView en un proyecto.- Utilizar callback(Interface) para pasar los datos.- Utilizar elementos views o widgets de Android.	
Duración del proyecto	<p>Tendrá dos clasaes para realizar la evaluación. Durante este tiempo puede preguntar y consultar a facilitador lo que sea necesario.</p> <p>Debe ir realizando commits de sus progresos. Debe compartir el proyecto en gitHub.</p>	
Tips o listado de Preguntas Guía		
Productos a obtener durante la realización del proyecto		

Proyecto de Android Studio archivo de respaldo
Especificaciones de desempeño
Deberá realizar la actividad según requerimientos técnicos y en un plazo máximo de 3 clases, el resultado deberá ser entregado como un proyecto de Android Studio. La solución deberá ser gestionada a repositorio GitHub para su posterior revisión.
Cronograma de actividades
Sugerencias bibliográficas para la investigación
- https://developer.android.com/guide