Instituto Tecnológico de Buenos Aires

TERCERA ENTREGA: IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL

72.36 - Interacción Hombre-Computadora (HCI) - Grupo 10

Autores:

Bernasconi, Ian - 62867

Feferovich, Jeremías - 62701

Gutiérrez, Agustín - 62595

Índice

1. Introducción

Este documento es la tercera y última de una serie de entregas que consisten en diseñar e implementar parcialmente un sitio Web y una aplicación móvil (Android) que permitan administrar y controlar en forma remota dispositivos inteligentes que se encuentren presentes en un hogar. Esta entrega se enfoca en la implementación de la aplicación móvil.

Basándonos en las pautas que se establecieron en el primer informe, creamos la aplicación móvil siguiendo las necesidades de los usuarios y los modelos de persona previamente definidos. De esta manera, el resultado obtenido se acerca bastante al prototipo final en cuanto a usabilidad y funcionalidad, aunque el diseño visual puede diferir. El diseño es amigable y fácil de usar, lo que coincide con el modelo mental que se espera de los usuarios representativos. Para lograr esto, se utilizó el entorno de desarrollo Android Studio, el lenguaje Kotlin y el conjunto de herramientas de Jetpack Compose para la creación de interfaces de usuario nativas.

2. Requisitos Implementados

2.1. Requisitos funcionales obligatorios

Tipos de dispositivos soportados

Los tipos de dispositivos soportados son los siguientes:

1. Aspiradora
2. Lámpara
3. Puerta
4. Horno
5. Aire acondicionado

Dispositivos

En la pantalla de 'Habitaciones', como se puede ver en la Figura 1, se muestran los dispositivos correspondientes a cada habitación, pudiéndose navegar entre habitaciones deslizando hacia los costados o con el menú superior. También se incluye una "Habitación"que contiene a todos los dispositivos. Seleccionando un dispositivo, se abre un *Dialog* con el estado del mismo, y permite modificarlo. Si el estado fue modificado, una vez cerrado el *Dialog* se muestra un *Snackbar* informando que el estado se actualizó exitosamente.

Figura 1: Pantalla de habitaciones

Notificaciones

Al efectuarse cambios en los dispositivos de manera remota (Desde el sitio web, por ejemplo) el usuario recibe notificaciones al respecto en la barra de estado de su dispositivo móvil. Por defecto, las notificaciones se reciben solamente para eventos que consideramos de alta importancia: El desbloqueo de puertas y el apagado de la aspiradora. La intención fue alertar cuando la aspiradora terminara de cargar, tuviera poca batería, y se quedara sin batería, pero como la API no proporciona eventos al cambiar el nivel de la batería, cambiamos a notificar el apagado, que consideramos se debería producir al agotarse la batería.

Por otro lado, se permite a los usuarios personalizar las notificaciones que desean recibir, pudiendo seleccionar los eventos que cada uno considere importantes desde la pantalla de menú. Esto permite a los usuarios

que desean tener un control más estricto sobre sus dispositivos estar notificados de los cambios producidos en detalle, a la vez que permite reducir la cantidad de notificaciones recibidas a los usuarios a los que no les interesa enterarse de los cambios producidos en sus dispositivos.

2.2. Requisitos funcionales opcionales

Rutinas

En la pantalla de 'Rutinas', como se puede ver en la Figura 2, se muestran todas las rutinas. En cada rutina se pueden ver los dispositivos conectados a la misma, así como ejecutarla apretando el botón de *play*. Una vez ejecutada la rutina se muestra un *Snackbar* informando que la rutina fue ejecutada exitosamente.

2.3. Requisitos no funcionales obligatorios

Idioma

Mediante el uso de los archivos *strings.xml* se generaron los *Strings* de todo el proyecto en los dos idiomas pedidos: español e inglés. Además, el idioma a utilizar es el establecido en la configuración regional del dispositivo, reaccionando ante cualquier cambio en el mismo, como se solicitó en la consigna.

Navegación y Factor de Forma

El tipo de navegación varía dependiendo del factor de forma del dispositivo en el que se está corriendo la aplicación:

- Celular con orientación vertical: Se utiliza una barra de navegación inferior como se muestra en la Figura 2.

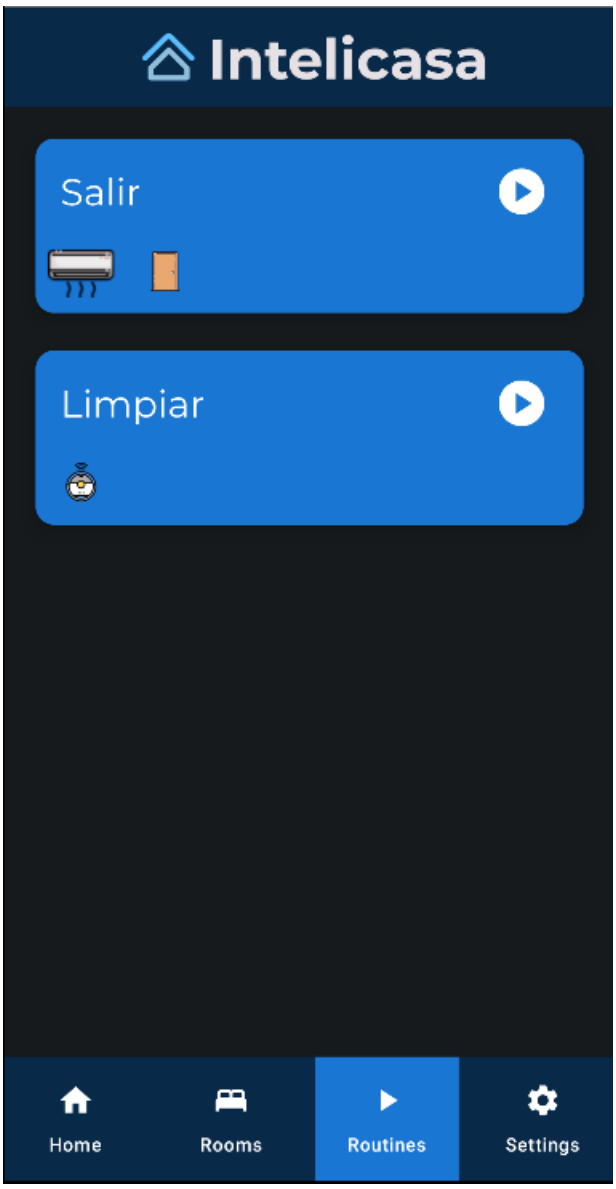

Figura 2: Pantalla de rutinas

- Celular con orientación horizontal: Se utiliza un rail lateral como se muestra en la Figura 3.

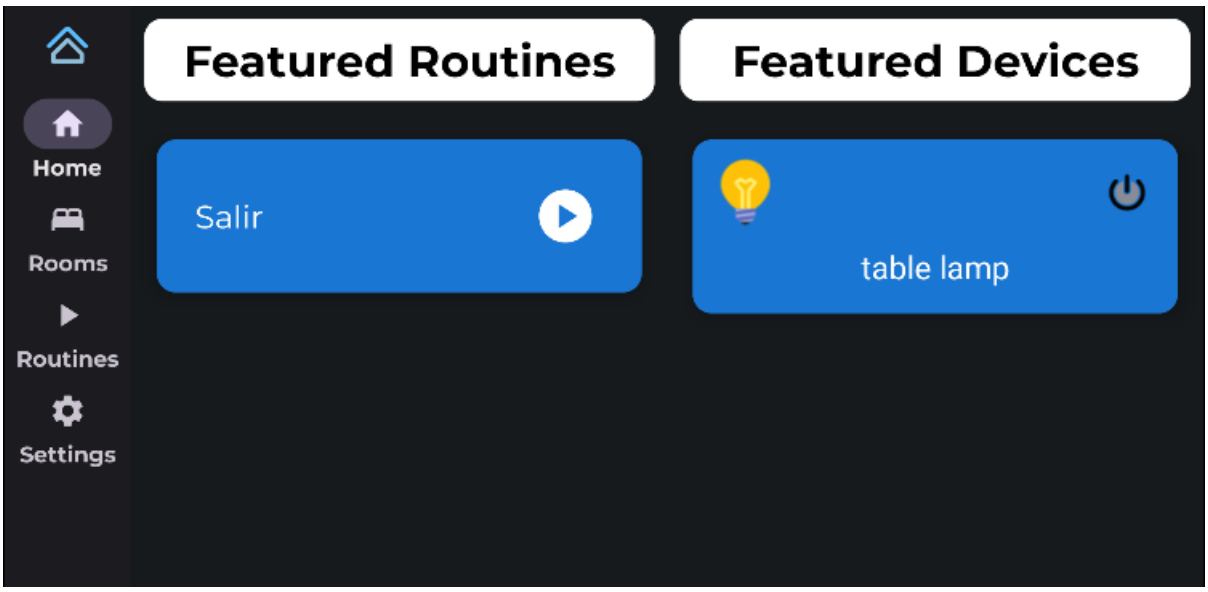

Figura 3: Pantalla con orientación horizontal con celular

Por otra parte, para la Tablet se realizaron varios cambios para que la distribución de los elementos sea más acorde: En la pantalla de inicio, se modificó el *layout* para que la sección de rutinas destacadas y la de dispositivos destacados se encuentren en dos columnas separadas, como se puede ver en la Figura 4

- Tablet con orientación horizontal: Se utiliza un cajón de navegación lateral permanente como se muestra en la Figura 4.

Figura 4: Pantalla de Inicio para Tablet

- Tablet con orientación vertical: Se utiliza una barra de navegación inferior como se mostró en la Figura 5

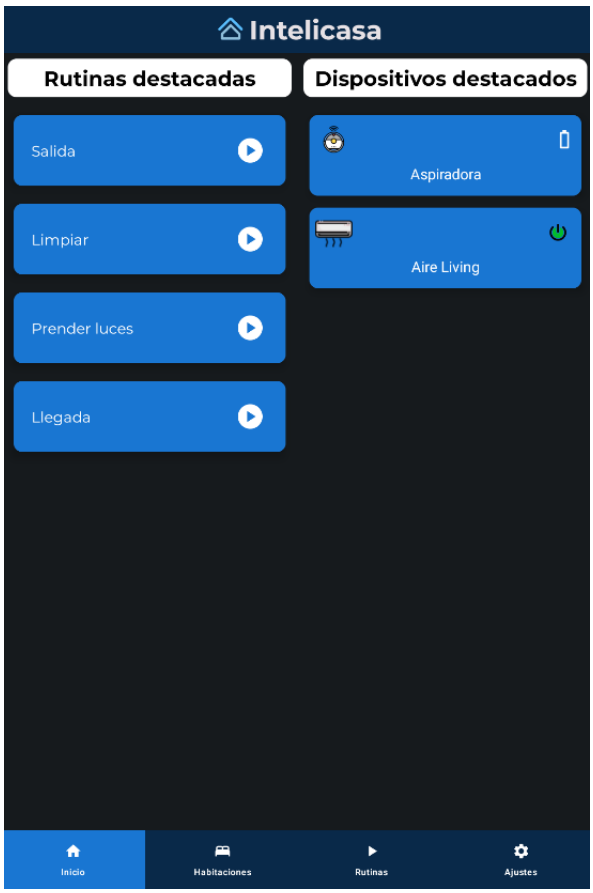

Figura 5: Pantalla con orientación vertical con tablet

En todas las pantallas donde se quería mostrar una lista de elementos, se utilizó una *LazyVerticalGrid*, asignándole al parámetro *columns* el valor *GridCells.Adaptive(minCardWith)*, para que así la cantidad de columnas varíe según el tamaño del dispositivo.

Personalización

Desde la pantalla 'Ajustes', se puede elegir si se desea utilizar el tema del dispositivo o si se lo quiere reemplazar. Si se elige reemplazarlo, se habilita la opción de definir si se quiere el tema claro u oscuro. Asimismo, en esta pantalla hay un desplegable de notificaciones, que le permite al usuario decidir cuáles desea recibir.

Por otro lado, desde la pantalla de habitaciones se puede elegir qué dispositivos se desea destacar para tenerlos en la pantalla de 'Inicio'.

Versión mínima

La aplicación funciona correctamente en dispositivos Pie 9.0 (API 28) o superior.

3. Capturas de pantalla de todas las páginas de la aplicación

3.1. Pantalla de Inicio

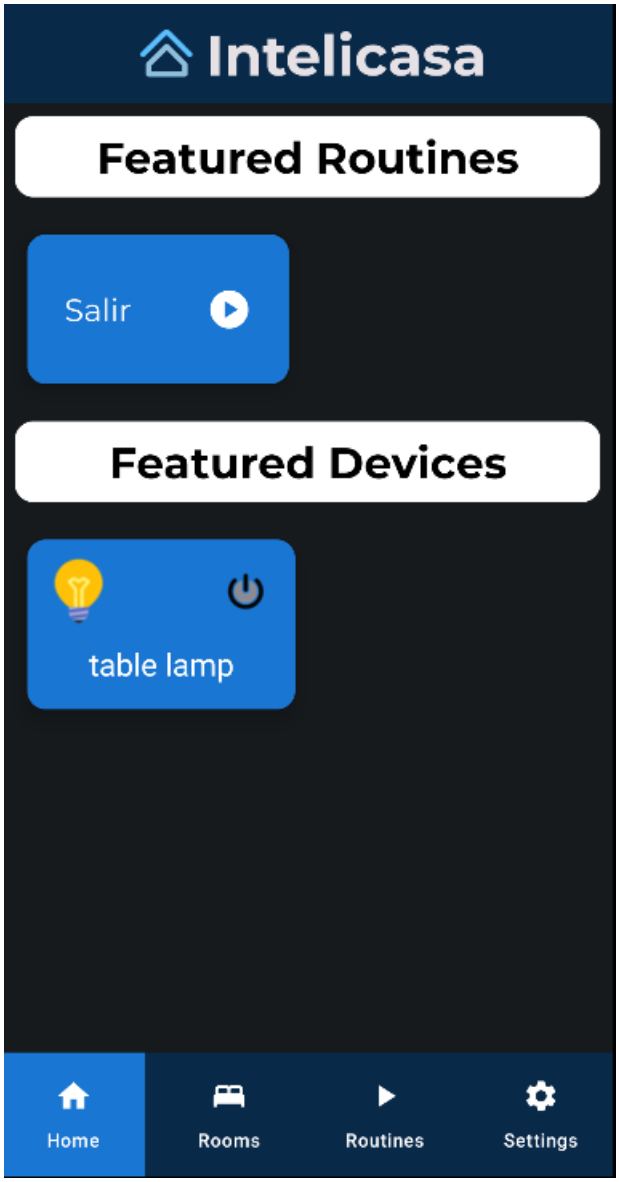

Figura 6: Pantalla de Inicio

En la pantalla de 'Inicio' se pueden observar tanto las rutinas como los dispositivos destacados por el usuario para tener un fácil acceso a ellos. Las rutinas pueden ejecutarse mediante el botón de *play*. En el caso de que no se hayan seleccionado rutinas destacadas o que no se haya agregado ninguna, se muestra un mensaje de información. Dispositivos destacados también maneja esta última funcionalidad. Además, haciendo click en la *card* de un dispositivo, se despliega un *Dialog* con el estado del mismo, pudiendo modificar este.

4. Decisiones de usabilidad

Se estableció que en caso de que el nombre de una rutina sea demasiado largo, se aplique una elipsis en el texto para evitar afectar la apariencia de la tarjeta. En cuanto a las rutinas, no se permite modificarlas ni marcarlas como favoritas. Por otro lado, con respecto a los dispositivos, los usuarios pueden interactuar con ellos libremente, pero no tienen la opción de cambiarles el nombre ni eliminarlos.

4.1. Diferencias presentadas en las pantallas respecto de la versión final de los prototipos

En primer lugar, cabe destacar que existe una gran cantidad de cambios en el aspecto visual de la página en comparación con los prototipos finales. Esto se debe a que, al momento de diseñar los prototipos, no estábamos al tanto de las herramientas disponibles en Android Studio y Jetpack Compose, que nos ofrecía diversas facilidades en términos de diseño. También se ajustó la paleta de colores, de acuerdo a las correcciones del primer trabajo, así como algunos iconos.

Figura 7: Pantalla de Inicio

4.2. Implementación práctica de los temas vistos en la materia

4.2.1. No hacemos pensar al usuario

Al aplicar buenas prácticas en el desarrollo de Android, logramos crear una aplicación que utiliza componentes familiares para los usuarios, como la bottom navigation bar y el Navigation Rail, que suelen encontrarse en otras aplicaciones. Buscamos facilitar la comprensión y usabilidad de nuestra aplicación a través de una cuidadosa jerarquía de colores, íconos e imágenes representativas en todas las pantallas. Por ejemplo, íconos comunes como la casa para la pantalla de inicio o el icono de play para las rutinas. Estos elementos permiten a los usuarios utilizar nuestra aplicación sin requerir un esfuerzo excesivo de pensamiento. Por último, cada botón en el que se puede hacer click presenta una sombra, elevación u otro color de fondo que lo hace fácilmente distinguible. Además, los menúes *dropdown* tienen un formato que corresponde al modelo mental comúnmente utilizado por los usuarios.

4.2.2. Navegación

Para la orientación vertical de nuestra aplicación, implementamos una bottom navigation bar que muestra los íconos y textos correspondientes a cada una de las diferentes pantallas. Además, en la pantalla de habitaciones, permitimos al usuario deslizar hacia los costados para navegar entre ellas. En cambio, en el modo horizontal, hemos incorporado un rail en el lado izquierdo de la pantalla que muestra las distintas pantallas disponibles para la navegación del usuario. Al momento de interactuar con acciones, como modificar los atributos de un dispositivo, se mostrará el *modal* visto en las secciones anteriores. Si el usuario desea regresar a la pantalla anterior, debe hacer click fuera del diálogo.

4.2.3. 8 reglas de oro

1. Consistencia: A través de toda la aplicación se mantiene consistencia en cuanto al formato, los botones, la estructura, etc. Los componentes se mantienen al cambiar de orientación el dispositivo móvil.

2. Shortcuts: Los dispositivos y las rutinas destacadas aparecen en la pantalla de inicio para tener un acceso más directo a lo más utilizado por el usuario.

3. Feedback: Implementamos diversas técnicas para evitar cualquier confusión por parte de los usuarios, brindándoles feedback al interactuar con sus dispositivos o rutinas. Por ejemplo, utilizamos un *snackbar* para mostrar mensajes cuando se modifican estados o atributos de los dispositivos, así como al ejecutar una rutina.

4. Cierre de diálogos: Únicamente se utilizan diálogos para acceder a los atributos del dispositivo. Para volver atrás a la pantalla principal se debe presionar afuera del diálogo. Si el dispositivo fue modificado, se mostrará un *snackbar* en la parte inferior de la pantalla para confirmar los cambios realizados. El *snackbar* tiene un tiempo de visualización, pero también posee un botón por si se quiere remover más rápidamente.

5. Manejo de errores: El manejo de errores de nuestro trabajo en la página web eran parte de la creación de dispositivos, rutinas o habitaciones. Al no tener esas funcionalidades en la aplicación Android, no fue necesario el uso de manejo de errores.

6. Revertir fácilmente las acciones: Las acciones que podían revertirse en la web eran el borrado de los componentes creados por el usuario. No fue implementada esa funcionalidad en este trabajo.

7. Sentir el control: No se implementaron mensajes de confirmación, ya que no existen acciones que no sean reversibles. En la web se utilizaba un diálogo de confirmación para confirmar que la acción que iba a producirse es la deseada por el usuario.

8. Reducir lo que el usuario debe recordar: Las pantallas de la aplicación son simples y consistentes, lo que fomenta el reconocimiento de las mismas y la pronta familiarización con el funcionamiento. Además, el uso de dispositivos y rutinas destacados disminuye los pasos necesarios para llegar a los mismos, así como también lo que se debe memorizar.

4.2.4. Estándares de diseño

Es relevante destacar que durante todo el proceso de desarrollo utilizamos los estándares establecidos en Material Design con el objetivo de que nuestros usuarios se sientan cómodos utilizando interfaces gráficas móviles que sean familiar para ellos. De esta forma, se utilizaron colores y tipografías que ya han sido utilizados y validados por la comunidad.

4.2.5. Colores

Una parte crucial de la aplicación es la coherencia de la paleta de colores y su adecuación a los usuarios representativos. Nuestra paleta de colores es coherente con el producto al que está asociada. El azul se asocia comúnmente con la inteligencia, que en este caso hace referencia a un hogar inteligente. Es por eso que la aplicación utiliza este color como primario, mientras que los demás colores se han seleccionado basándonos en este para crear contrastes. Por otra parte, utilizamos iconos de colores saturados para que resalte a la categoría o tipo al que pertenece el componente. El resto de la web mantiene un diseño minimalista.

4.3. Implementación respecto a los modelos de persona

En un principio, la aplicación está diseñada para que usuarios como María tengan fácil acceso a los dispositivos que más desean controlar en la pantalla de inicio. De esta manera, María puede controlar la intensidad de las luces rápidamente y obtener información en tiempo real de sus dispositivos destacados mediante un solo toque. Usuarios como María tendrán un gran control sobre la aplicación y podrán experimentarla al máximo.

Además, la aplicación está diseñada para ser útil para usuarios como Miguel. Si el usuario no tiene ningún dispositivo creado, se muestra un mensaje de ayuda que le indica que debe configurar los dispositivos. Cada dispositivo tiene un ícono que representa la categoría a la que pertenece y otro que indica su estado actual. Una vez que el usuario tenga los dispositivos configurados y destacados, ya no es más necesario que el usuario navegue por las distintas pantallas, lo que otorga un funcionamiento sencillo y mecánico de la aplicación. Estas características están diseñadas para ayudar a Miguel a sentirse más cómodo y seguro al utilizar la app.

En cuanto a usuarios con deseos similares a los de Juan, la aplicación está estructurada de manera simple a través de una barra de navegación o rail, y la identificación de tipos de dispositivos y sus estados con iconos, los que permiten proporcionar buen feedback al usuario. Esto hace que la aplicación sea fácil de usar y no requiera conocimientos técnicos para sus características principales. Juan podrá mantener su casa segura al controlar el dispositivo de la puerta recibiendo las notificaciones que desee.

4.4. Justificación haciendo referencia a las sugerencias y/o correcciones realizadas por los docentes.

- *Tener en consideración la posibilidad de representar las tarjetas de rutinas de una forma similar a las tarjetas de dispositivos*

  Se decidió cambiar el color de las tarjetas de rutinas en el inicio para que sean más similares a las tarjetas de dispositivos, dejando una forma diferente para marcar que son elementos diferentes.

- *Tener en consideración que las rutinas son de ejecución inmediata y, por lo tanto, no tienen un estado. Por otro lado, la ejecución de una rutina en forma planificada o ante un cambio en el estado*

  Se decidió cambiar el botón de ejecución de una rutina en el prototipo por uno de play para evitar confusiones y desvincular la opción de estado de una rutina, ya que las rutinas son de ejecución inmediata y no tienen un estado, aunque pueden ser ejecutadas de forma planificada o en respuesta a un cambio de estado.

- *Tener en consideración que algunos de los elementos presentan poco contraste (tarjetas).*

Se decidió modificar la paleta de colores para tener un mayor contraste y permitir mejor usabilidad.

5. Archivos necesarios para construir el instalador de la aplicación

Se adjunta un archivo *.zip* con el código fuente de la app. Para que esta pueda comunicarse con la API, se debe modificar el archivo *build.gradle (:app)*, modificando la variable *API_BASE_URL* que se encuentra en *buildTypes* → *release* por la IP de la computadora en la que se esté corriendo.

Una vez configurada la IP de la computadora, se debe construir la *APK*. Para esto, en el menú superior se debe ir a: *Build* → *Build Bundle(s) / APK(s)* → *Build APK(s)*. Esto producirá un archivo *.apk*, que luego deberá ser copiado al dispositivo en el que se quiera realizar la instalación.

6. Instructivo de instalación

Una vez que el archivo *.apk* se encuentre en el dispositivo, se debe utilizar un explorador de archivos y abrir el archivo mencionado. Posteriormente, seguir los pasos de instalación indicados.

7. Conclusión

Durante el desarrollo de este trabajo, comprendimos la agilidad a la hora de la implementación que nos proporcionó tener un prototipo definido de antemano, ya que nos evitó tener que discutir cuestiones del diseño general para así poder enfocarnos en el desarrollo.

Por otra parte, nos pareció entretenido e interesante adentrarnos en el mundo de mobile y aprender Kotlin para este trabajo, al cual se le puede seguir iterando para mejorar estilos e implementaciones, y agregarle nuevas funcionalidades.

Por último, nos asombró la velocidad con la que pudimos desarrollar una aplicación Android, ya que con nuestra poca experiencia previa pudimos hacerlo en un par de semanas. Además, teniendo experiencia previa con desarrollo Android utilizando *XML* para los layouts, descubrimos que Jetpack Compose presenta muchas mas facilidades y es más cómodo de programar.