

Informe PAWN

Eric Cabrera Cruz

Contenido

Desarrollo..... 3

Opinion sobre el Codelab..... 5

Problemas técnicos. 7

Desarrollo.

El presente informe tiene como objetivo documentar el proceso y los resultados del proyecto GreetingCard, que se desarrolló tomando como punto de partida la solución previamente implementada en el proyecto HappyBirthday. En este informe, se describirá cómo se utilizó la solución proporcionada por los desarrolladores de Android de Google como plantilla y se combinará con los conocimientos adquiridos a lo largo del curso para abordar y resolver el proyecto GreetingCard.

Como he comentado, en el proyecto GreetingCard, se decidió utilizar la solución proporcionada por los desarrolladores de Android de Google en HappyBirthday como una plantilla inicial. Esta decisión se basó en la eficacia probada de la solución de Google y en la familiaridad adquirida con su funcionamiento durante el desarrollo de HappyBirthday. Utilizar una plantilla existente permitió acelerar el proceso de desarrollo y garantizar una base sólida para la nueva aplicación.

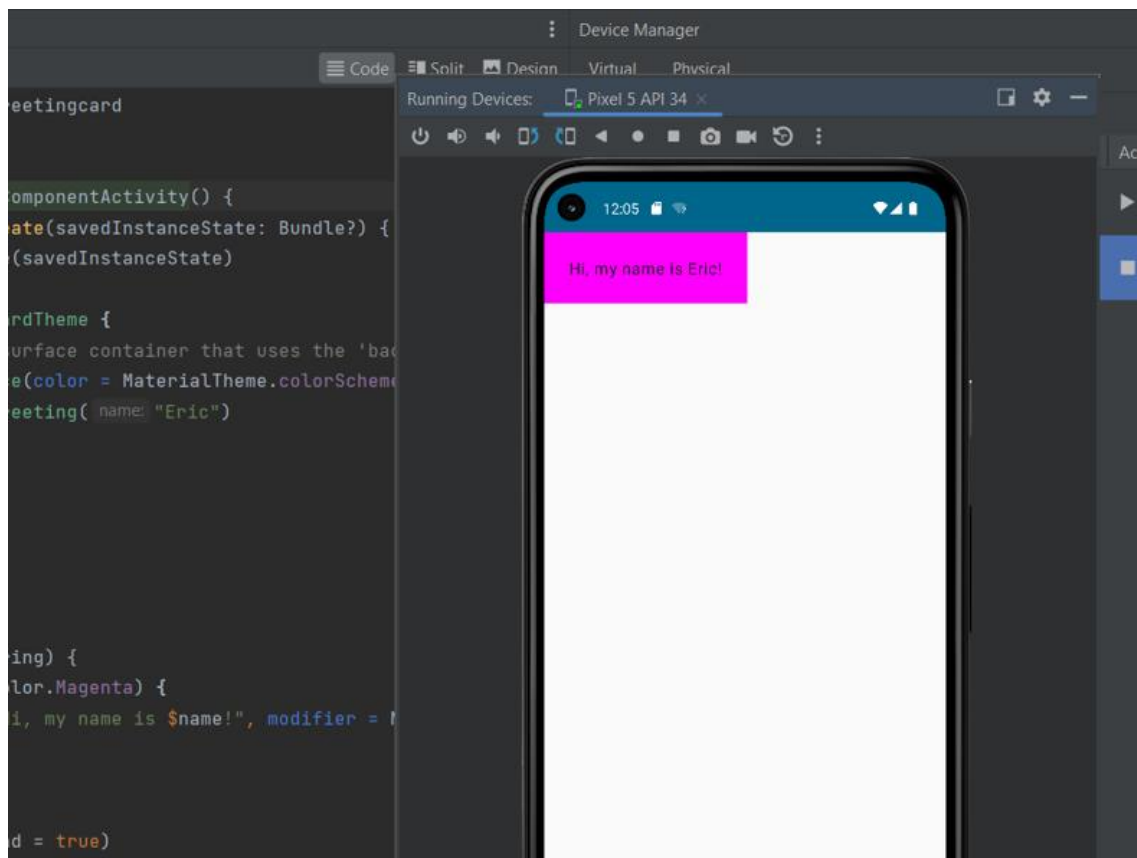
A lo largo del curso, se adquirieron conocimientos esenciales sobre el desarrollo de aplicaciones móviles en el entorno Android. Estos conocimientos incluyeron la programación en Kotlin, las vistas, la implementación de interfaces de usuario. Todos estos conocimientos se aplicaron en el desarrollo de GreetingCard para mejorar la funcionalidad y la usabilidad de la aplicación.

Durante el desarrollo de GreetingCard, se llevaron a cabo los siguientes cambios respecto a la plantilla usada. En primer lugar, se agregó una imagen del logo de Android, basada en el enunciado. A continuación, se agregó un fondo distinto para darle un toque personal al proyecto y finalmente se agregaron nuevos textos acorde a lo pedido en el enunciado.

Todo esto se puede apreciar en la siguiente captura de pantalla.



Además, procedo a presentarle la imagen de prueba de que se ha realizado el CodeLab entero, siendo esta una captura del primer ejercicio que promueven los desarrolladores del curso.



Opinion sobre el Codelab.

El codelab es, en general, un recurso bastante sólido. No obstante, me gustaría compartir mi perspectiva sobre cómo podría mejorarse la presentación de Kotlin. Si bien el codelab está bien estructurado y proporciona una visión general de los conceptos, siento que se alarga demasiado.

En particular, el codelab consta de tres secciones, y dos de ellas parecen ser de dificultad extremadamente baja. Esto puede resultar en la pérdida de tiempo, ya que se extienden en contenido que podría ser más conciso y directo. La información podría condensarse de manera más eficiente, permitiendo a los estudiantes avanzar más rápidamente.

Sin embargo, he de destacar que aprecio la simplicidad y la claridad con la que se explican los conceptos clave. La facilidad con la que se presentan las ideas es algo que debería mantenerse. En general, el codelab proporciona una base sólida para comprender Kotlin, pero podría ser aún más efectivo con algunos ajustes en la estructura y la profundidad de la información presentada.

Ejemplo de cómo creo que debería hacerse.

@Composable

```
fun GreetingImage(message: String, from: String, modifier: Modifier = Modifier) {
```

```
    // Create a box to overlap image and texts
```

```
    Box(modifier) {
```

```
        Image(
```

```
            painter = painterResource(id = R.drawable.androidparty),
```

```
            contentDescription = null,
```

```
            contentScale = ContentScale.Crop,
```

```
            alpha = 0.5F
```

```
        )
```

```
        GreetingText(
```

```
            message = message,
```

```
            from = from,
```

```
            modifier = Modifier
```

```
                .fillMaxSize()
```

```
                .padding(8.dp)
```

```
        )
```

```
}  
}
```

Explicación.

@Composable

```
fun GreetingImage(message: String, from: String, modifier: Modifier = Modifier) {
```

Esta es la definición de la función componible `GreetingImage`. Está marcada con la anotación `@Composable`, lo que significa que se utiliza para construir una parte de la interfaz de usuario de la aplicación. La función toma dos cadenas de texto (`message` y `from`) como argumentos y un modificador opcional (`modifier`) que se utiliza para personalizar la apariencia del componente.

```
// Create a box to overlap image and texts
```

```
Box(modifier) {
```

Aquí se crea un contenedor `Box` que se utiliza para superponer una imagen y texto. El modificador que se pasó a la función `GreetingImage` se aplica al `Box`.

```
    Image(  
        painter = painterResource(id = R.drawable.androidparty),  
        contentDescription = null,  
        contentScale = ContentScale.Crop,  
        alpha = 0.5F  
    )
```

Dentro del `Box`, se agrega un componente `Image`. Esta imagen se carga desde un recurso de pintor (`painterResource`) con la ID `R.drawable.androidparty`. El atributo `contentDescription` se establece en `null`, lo que significa que no se proporciona una descripción de la imagen. `ContentScale.Crop` se utiliza para ajustar la escala del contenido, y `alpha` establece la opacidad de la imagen en un 50%.

```
GreetingText(  
    message = message,  
    from = from,
```

```
        modifier = Modifier
            .fillMaxSize()
            .padding(8.dp)
        )
    }
}
```

Luego, se agrega el componente `GreetingText` que has definido previamente. Se pasan las cadenas de texto `message` y `from` como argumentos. Además, se aplica un modificador al `GreetingText` que configura el tamaño y el margen del componente para ocupar todo el espacio disponible y agregar un relleno de 8 densidad independiente de píxeles (dp).

En resumen, la función `GreetingImage` crea un contenedor `Box` que superpone una imagen y un componente de texto (`GreetingText`). La imagen se carga desde un recurso y se ajusta, y el texto se coloca encima de la imagen con ciertos estilos y espaciado configurados mediante el modificador. Esto permite la creación de una tarjeta de cumpleaños con una imagen de fondo y texto superpuesto.

Esto que he usado, es la forma en la que, en mi año de fundamentos de la programación I me lo explicaban aquí, en la universidad para posteriormente mandarnos ejercicios muy sencillitos pero eficientes para practicar lo aprendido, y dotándonos de un playground en donde, no solo podías compilar si no que te ponía nota respecto al ejercicio mandado.

Problemas técnicos.

Entiendo que este apartado es subjetivo, ya que cada cual dispone de un hardware distinto.

Pero después de hablarlo con algunos compañeros, la experiencia ha sido la misma, el Android Studio es muy lento, el emulador personalmente ocasiona que mi portátil tenga serios problemas de ejecución, impidiéndome hacer cualquier cosa mientras tengo el código ejecutándose en segundo plano. Y luego, cada vez que abro la aplicación tarda una eternidad en descargar las dependencias de modo que, para ver, por ejemplo, la estructura de mi proyecto, tengo que esperar 10 minutos.

La experiencia con Android Studio ha sido horrible, porque el tiempo de espera es muy grande y para colmo consume tanta memoria que no puedo aprovechar mi tiempo dejándolo en segundo plano.