

Desarrollo de Aplicaciones para Dispositivos Móviles

Ingeniero Walter Medina
walter.medina@correounivalle.edu.co



Inyección de dependencias con Dagger Hilt

Dagger es una biblioteca de inyección de dependencias para Android, desarrollada por Google. **Hilt** es otra biblioteca que esencialmente trabaja como una capa de abstracción sobre Dagger para simplificar su uso y reducir la cantidad de código.

```
class Car {  
  
    private val engine = Engine()  
  
    fun start() {  
        engine.start()  
    }  
}  
  
fun main(args: Array) {  
    val car = Car()  
    car.start()  
}
```

Sin inyección de dependencias: Car y Engine están estrechamente vinculados: una instancia de Car usa un tipo de Engine, y no se pueden utilizar subclases ni implementaciones alternativas con facilidad.

La inyección de dependencias (DI) es un patrón de diseño en el que las dependencias de un objeto son proporcionadas desde el exterior en lugar de ser creadas por el propio objeto. Este enfoque ayuda a reducir el acoplamiento entre componentes, facilita la prueba unitaria y mejora la modularidad del código.

```
class Car(private val engine: Engine) {  
    fun start() {  
        engine.start()  
    }  
}  
  
fun main(args: Array) {  
    val engine = Engine()  
    val car = Car(engine)  
    car.start()  
}
```

Con inyección de dependencias: Puedes pasar diferentes implementaciones de Engine a Car. Por ejemplo, puedes definir una nueva subclase de Engine, llamada ElectricEngine, para utilizar con Car. Si usas DI, solo debes pasar una instancia de la subclase actualizada de ElectricEngine y Car seguirá funcionando sin más cambios.

Pruebas Unitarias (JUnit / Mockito)

Las pruebas unitarias son una práctica de programación que implica la verificación individual de unidades o componentes de un software (métodos) para asegurar que funcionan correctamente de manera aislada.

JUnit: Es un marco de trabajo (framework) de pruebas unitarias. Unit proporciona anotaciones para identificar métodos de prueba, y proporciona clases y métodos para realizar afirmaciones que validan el comportamiento esperado del código.

Mockito: Es un marco (framework) que se utiliza para escribir pruebas unitarias mediante la creación de objetos simulados (mocks).

Las pruebas unitarias a menudo requieren aislar el código bajo prueba de sus dependencias externas, como bases de datos, servicios web o componentes que aún no han sido implementados. Mockito facilita la creación de estos objetos simulados, permitiendo a los desarrolladores especificar el comportamiento esperado de estos mocks durante las pruebas.



Imagen de internet

Autenticación y uso de la Base de datos Firestore de Firebase

Firebase es una plataforma de desarrollo móvil y web desarrollada por Google. Proporciona una amplia gama de servicios y herramientas que facilitan el desarrollo de aplicaciones, tanto para dispositivos móviles como para la web.

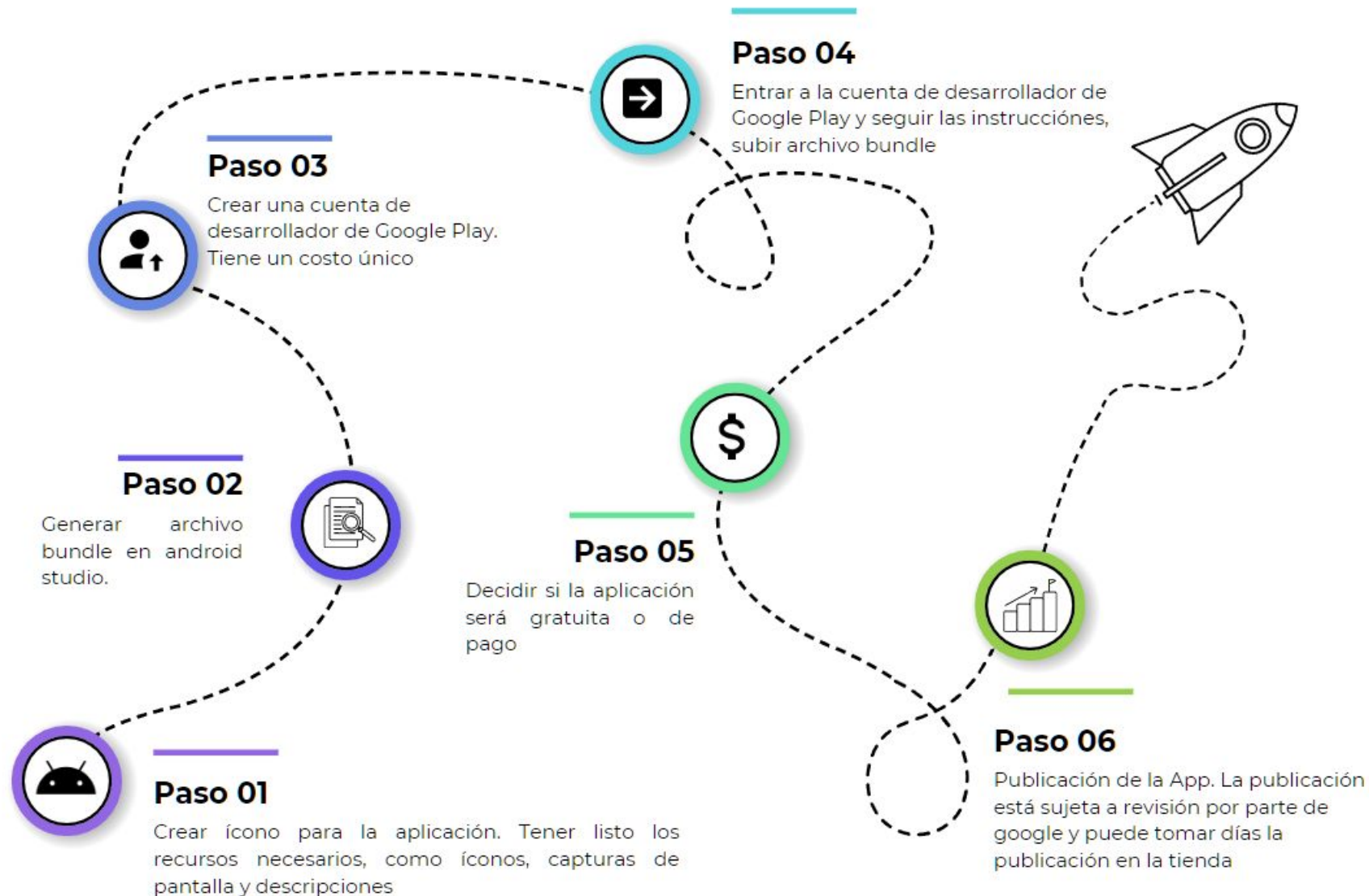
Authentication : Proporciona servicios de autenticación segura y fácil de usar, permitiendo a los desarrolladores implementar fácilmente la autenticación de usuarios a través de métodos como correo electrónico, contraseña, autenticación con Google, Facebook, etc.

Firestore: Es un servicio de base de datos NoSQL. Es una base de datos de documentos que está diseñada para almacenar y sincronizar datos en tiempo real entre aplicaciones web, móviles y servidores.

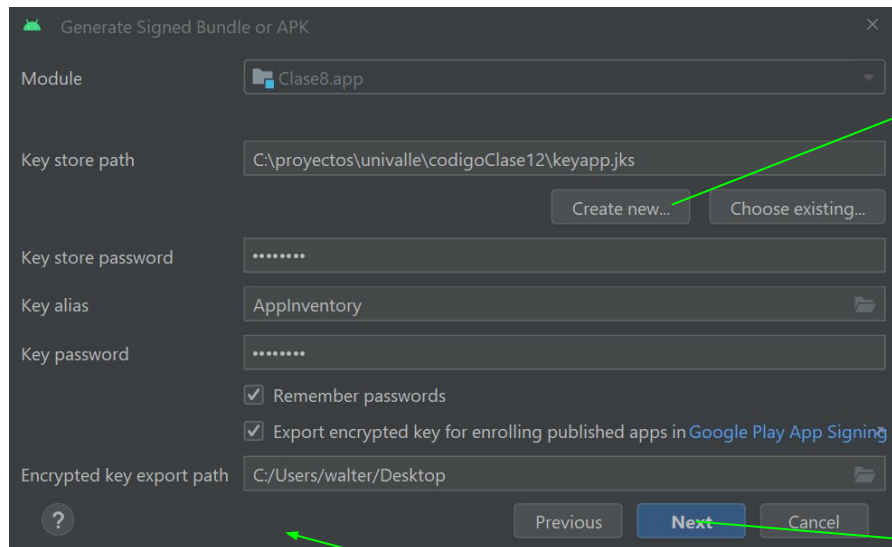


Imagen de internet

Subir una app a la tienda de Google Play



Firmar la app y generar el bundle



Module: Clase8.app

Key store path: C:\proyectos\univalle\codigoClase12\keyapp.jks

Key store password:

Key alias: ApplInventory

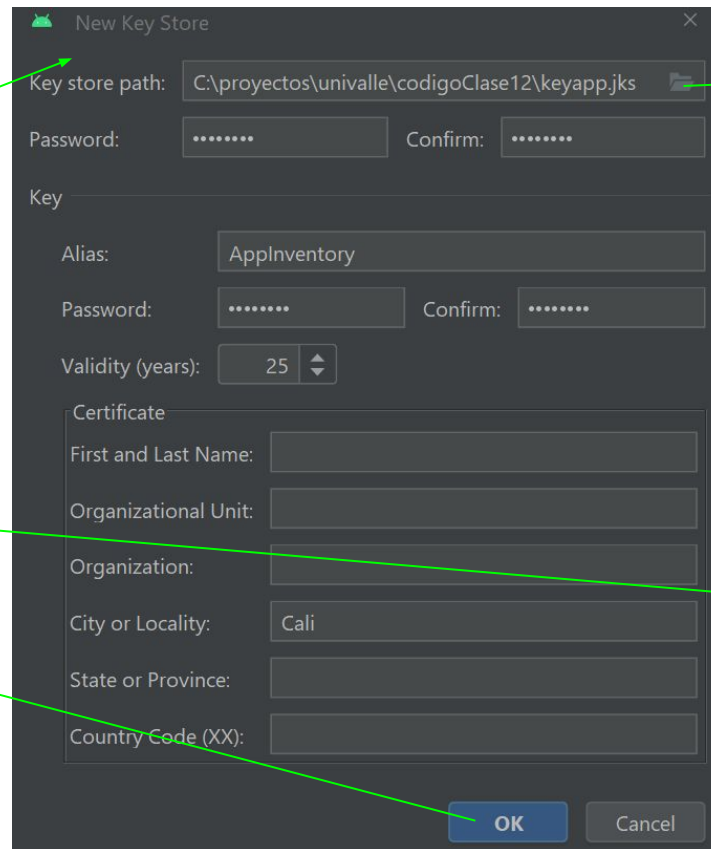
Key password:

☒ Remember passwords

☒ Export encrypted key for enrolling published apps in [Google Play App Signing](#)

Encrypted key export path: C:/Users/walter/Desktop

Buttons: Previous, Next, Cancel



Key store path: C:\proyectos\univalle\codigoClase12\keyapp.jks

Password: Confirm:

Key

Alias: ApplInventory

Password: Confirm:

Validity (years): 25

Certificate

First and Last Name:

Organizational Unit:

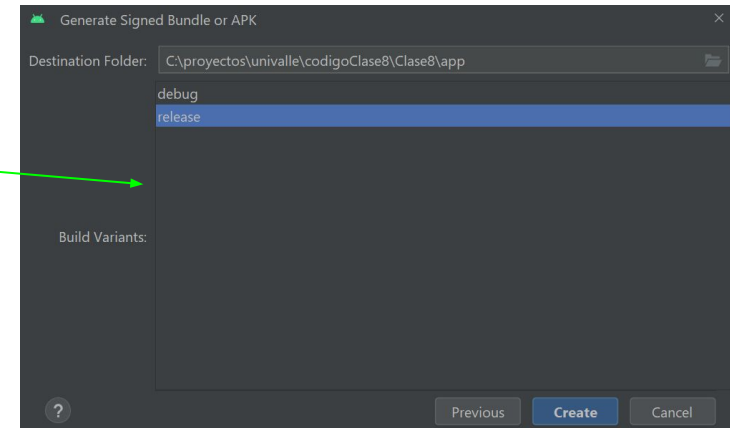
Organization:

City or Locality: Cali

State or Province:

Country Code (XX):

Buttons: OK, Cancel



Destination Folder: C:\proyectos\univalle\codigoClase8\Clase8\app

Build Variants: debug, release

Buttons: Previous, Create, Cancel

Elegir la ruta donde se guardará la key y asignarle un nombre

3

4

1

2

GRACIAS !!

*“El esfuerzo vence al talento cuando el talento
no se está esforzando”.*
Anónimo



Bibliografía

- ❖ <https://developer.android.com/training/dependency-injection/hilt-android?hl=es-419>
- ❖ <https://developer.android.com/codelabs/android-hilt?hl=en#0>
- ❖ <https://developer.android.com/studio/test/test-in-android-studio?hl=es-419>
- ❖ <https://firebase.google.com/>

