Dagger2

1.

¿Qué es la inyección de dependencias?

“

Es un patrón de diseño que ayuda a hacer a nuestras aplicaciones más modulares, escalables, fáciles de testear y mantener.

2.

Grafo de dependencias

“

En Dagger2 la creación del grafo se ha adelantado al tiempo de compilación por lo que se pueden descubrir los errores mucho antes.

3.

Anotaciones

Anotaciones

- ***@Module.*** Clases que proporcionan dependencias
- ***@Provides.*** Marca los métodosen el módulo que devuelven dependencias
- ***@Component.*** Interfaz de puente entre los módulos y la inyección
- ***@Inject.*** Solicita una dependencia

4.

Problemas en Android

Problemas

- Métodos ***onCreate*** y ***onResume.***
- Cuidado con el tamaño del grafo
- Android puede destruir las dependencias cuando quiera.

5.

Dependencias en build.gradle

Añadimos la dependencias y el plugin *kotlin-kapt*

```
implementation "com.google.dagger:dagger:$daggerVersion"
kapt "com.google.dagger:dagger-compiler:$daggerVersion"
```

```
apply plugin: 'kotlin-kapt'
```

```
kapt {
    generateStubs = true
}
```

6.

Crear módulo

“

Módulo llamado *AppModule* que recibirá un objeto de tipo *Application*

```
@Module
class AppModule(val app: MyApplication) {
    @Provides
    @Singleton
    fun provideApp() = app
}
```

7.

Crear componente

“

Tiene un método para inyectar nuestra clase *Application* en nuestro grafo inicial

```
@Singleton
@Component(modules = arrayOf(AppModule::class))
interface AppComponent {
    fun inject(app: MyApplication)
}
```

8.

Generación del grafo

Pasos

1. Creamos una variable lazy para inicializar el *AppComponent*
2. En onCreate accedemos a la variable y hacemos uso de su método inject.

“

Nos creamos una variable *lateinit* ya que necesitamos que no sea *null* y sea inicializada posteriormente por nuestro inyector de dependencias

```
@Inject
lateinit var myClassInjectable: MyClassInjectable
```

9.

¡Cuidado!

“

Si intentamos acceder a una variable *lateinit* antes de que sea inicializada lanzará una excepción de tipo ***UninitializedPropertyAccessException***