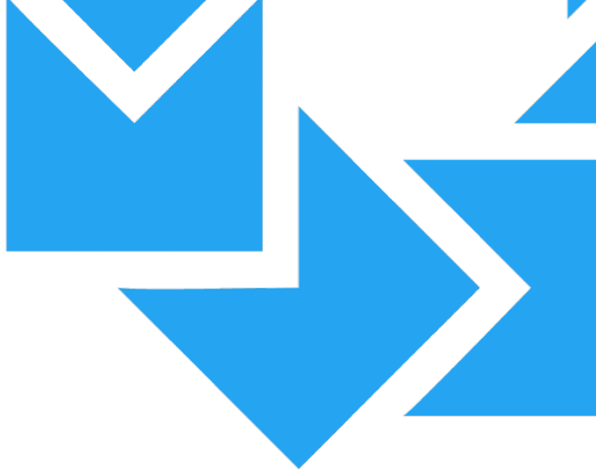
KOTLIN MULTIPLATFORMA EN ACCIÓN

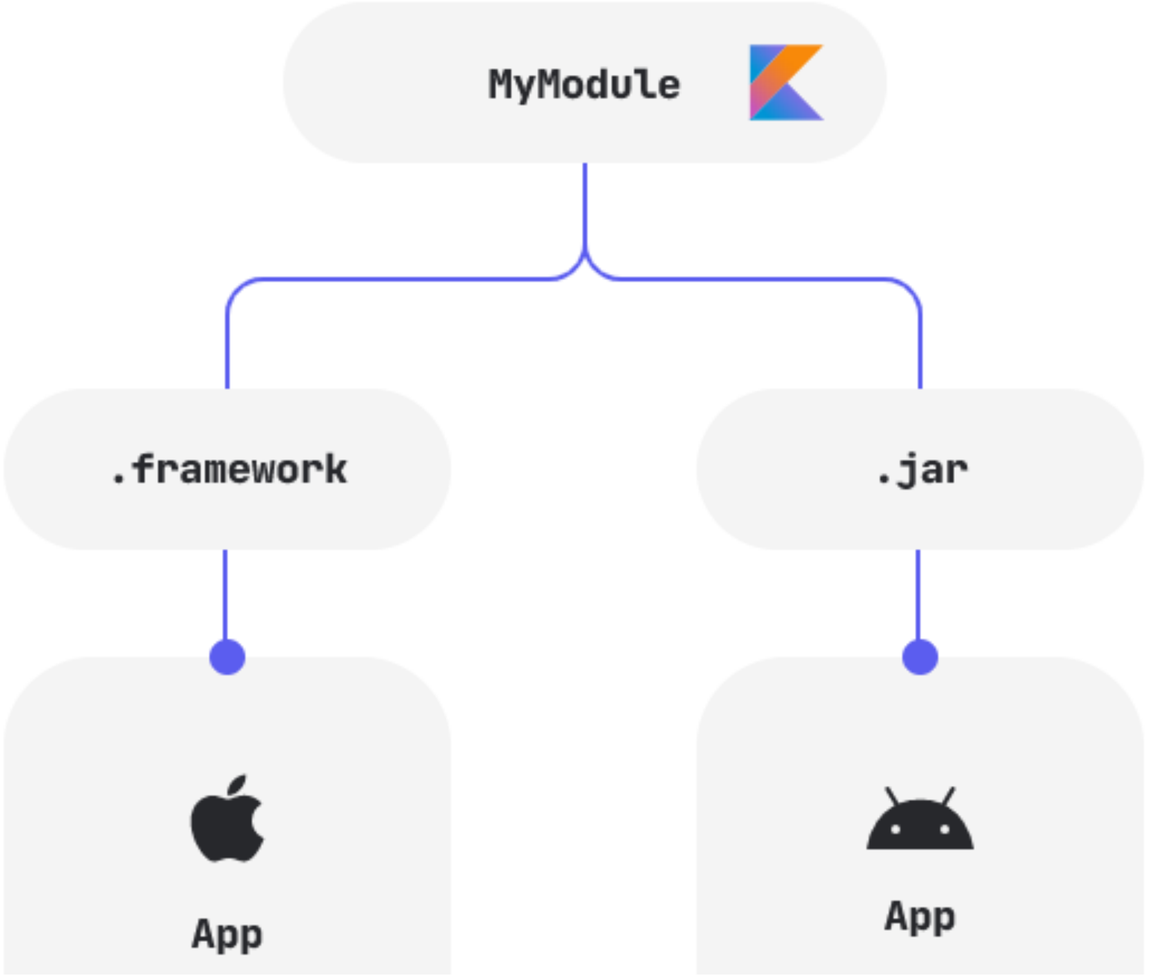
LAURA DE LA ROSA

Kotlin Medellín User Group

Redes: @lauramdelarosa



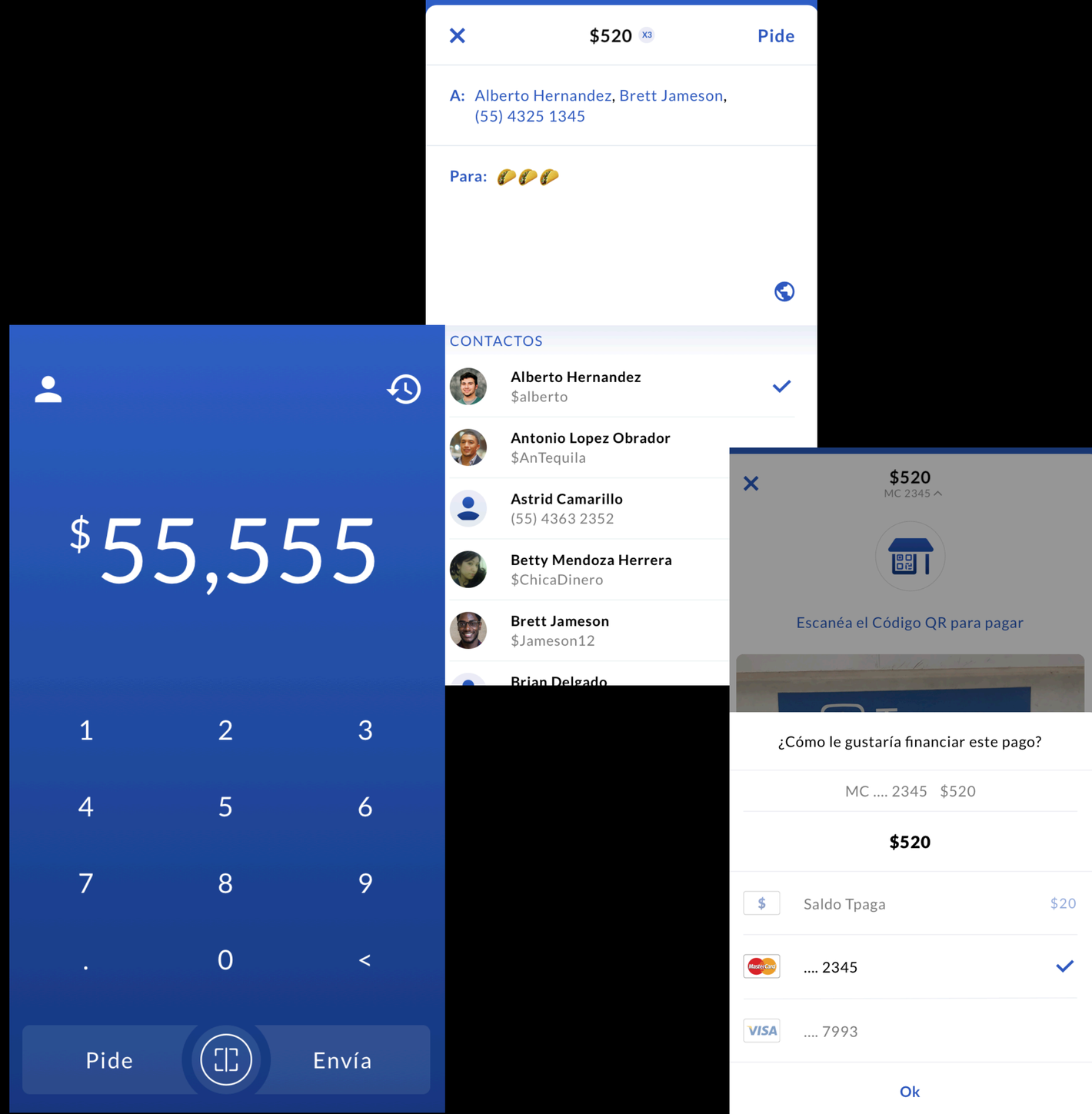
FLUTTER
VS
KOTLIN MULTIPLATAFORMA



https://www.jetbrains.com/lp/mobilecrossplatform/

Tpaga MX

- Envío de dinero
- Pide dinero
- Pagos con Qr
- Formulario de regulación mexicana.
- Ver Informes de Pagos



Tpaga MX

- Usuarios pagan fácil y sin efectivo
- Changarros controlan sus ventas
- Bancarizan los negocios informales

Kotlin Multiplatform

- En producción hace 1 año en Android
- En producción hace 10 meses en iOS
- 18% iOS
- Equipo .

Orígenes

Orígenes

- App de Colombia
- Oportunidad en MX
- Poco tiempo
- Objetivos específicos MX
- Ambas Plataformas

Orígenes

Nueva app con KOTLIN MULTIPLATAFORMA

Proceso y Aprendizaje

Acoplamiento

- Android. ☺️
- iOS. 🙄
- Problemas?
  - Revisando PR
  - Cambiando lógica de negocio

Migración

- Todo en Kotlin
- Migrar la arquitectura actual - Probabilidad de mejorarla .
- Tener buena comunicación ante:
  - Nuevos features .
  - Revisando PR
  - iOS empoderados para aprender y crecer
- Tpaga NO migró

Tip

Cuando desarrollas en código compartido, Piensa en compartir.

¿Qué compartir?

- Depende 🤫
  - Reglas y lógica de negocio
    - Network
    - Manejo de errores
    - Casos de Uso
    - Transformación de datos.

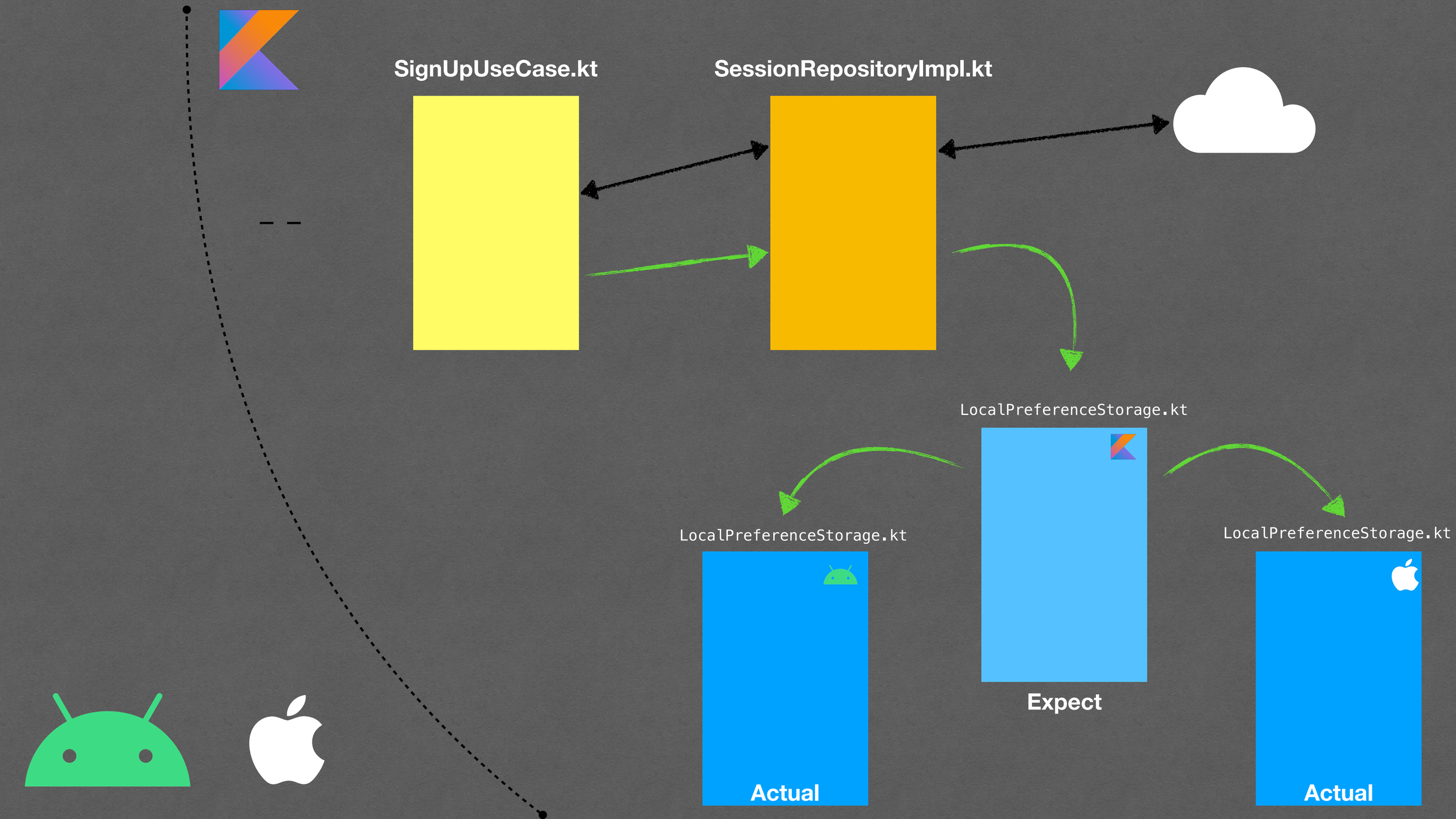
Tip

Estudia muy bien las librerías open source, antes de implementarlas.

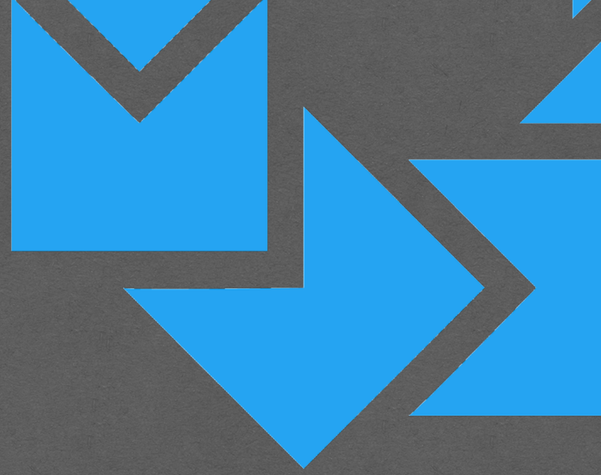
- **No son estables**
- **Representan riesgos**

¿De qué dependo?

Expect / Actual

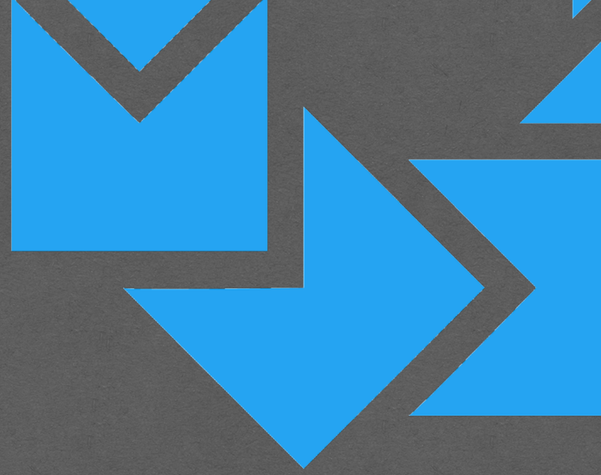

SignUpUseCase.kt
SessionRepositoryImpl.kt
LocalPreferenceStorage.kt
LocalPreferenceStorage.kt
LocalPreferenceStorage.kt
Expect
Actual
Actual

Consideraciones



- iOS lo debe implementar en Kotlin
- No es Kotlin puro por la interoperabilidad con Objective C -> código raro

expect / actual



- Donde los uso?
  - Casos de usos
  - Presenters
- Cuales tenemos?
  - Time
  - DeviceInfo
  - LocalPreferenceStorage
  - Contactos
  - Dispatchers
  - UUID …
  - Problemas ? -> Location

Common Gradle

```
apply plugin: 'kotlin-multiplatform'
apply plugin: 'kotlinx-serialization'
apply plugin: 'com.android.library'

kotlin {

    targets {
        fromPreset(presets.android, 'android')

        final def iOSTarget = presets.iosX64
        fromPreset(iOSTarget, 'ios')
    }
}
```

Diferencias entre plataformas

- Localización
  - Android: FuseLocation - GpsLocation
  - iOS: CoreLocation

Notas

- La interoperabildad es con Objetive C y no con Swift
- Valores de parámetros por defecto: Kotlin- Swift - Objective C .
- Compilar en Android studio para generar el build del framework para iOS
- Generics

```
data class Node<T>(val value: T, val next: Node<T>? = null)
```

```
__attribute__((objc_subclassing_restricted))
__attribute__((swift_name("Node")))
@interface EverythingNode : KotlinBase
- (instancetype)initWithValue:(id _Nullable)value next:(EverythingNode * _Nullable)next
__attribute__((swift_name("init(value:next:)")))
__attribute__((objc_designated_initializer));
// more declarations here
@end;
```

```
let node = Node<NSString>(value: "hello!", next: nil)
```

Cannot specialize non-generic type 'Node'

Replace '<NSString>' with ''

Fix

```
object CarNode {
  fun getCarNode(): Node<Car> = Node(Car(2019, "Tesla", "Model S"))
}
```

```
func test() {
    let currentCarNode = CarNode.init().getCarNode()
    let currentCar = currentCarNode.value
    print(currentCar.model)
}
```

Value of type 'Any?' has no member 'model'

```
func test() {
    let currentCarNode = CarNode.init().getCarNode()
    let currentCar = currentCarNode.value as! Car
    print(currentCar.model)
}
```

```
kotlin {
    iosX64 {
        binaries {
            framework {
                freeCompilerArgs += "-Xobjc-generics"
            }
        }
    }
}
```

```
data class Node<T : Any>(val value: T, val next: Node<T>? = null)
```

```objc
__attribute__((objc_subclassing_restricted))
__attribute__((swift_name("Node")))
@interface EverythingNode<T> : KotlinBase
- (instancetype)initWithValue:(T)value next:(EverythingNode<T> * _Nullable)next
__attribute__((swift_name("init(value:next:)")))
__attribute__((objc_designated_initializer));
// more declarations here
@end;
```

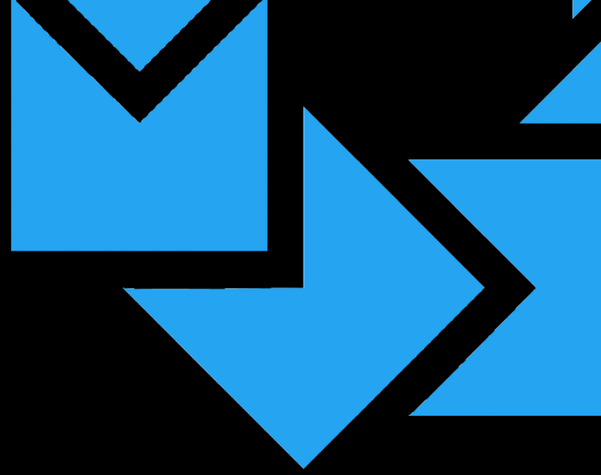
```
func test() {
    let node = Node<NSString>(value: "hello!", next: nil)
    let length = node.value.length
    print(length)

    let currentCarNode = CarNode.init().getCarNode()
    let currentCar = currentCarNode.value
    print(currentCar.model)
}
```

Dificultades

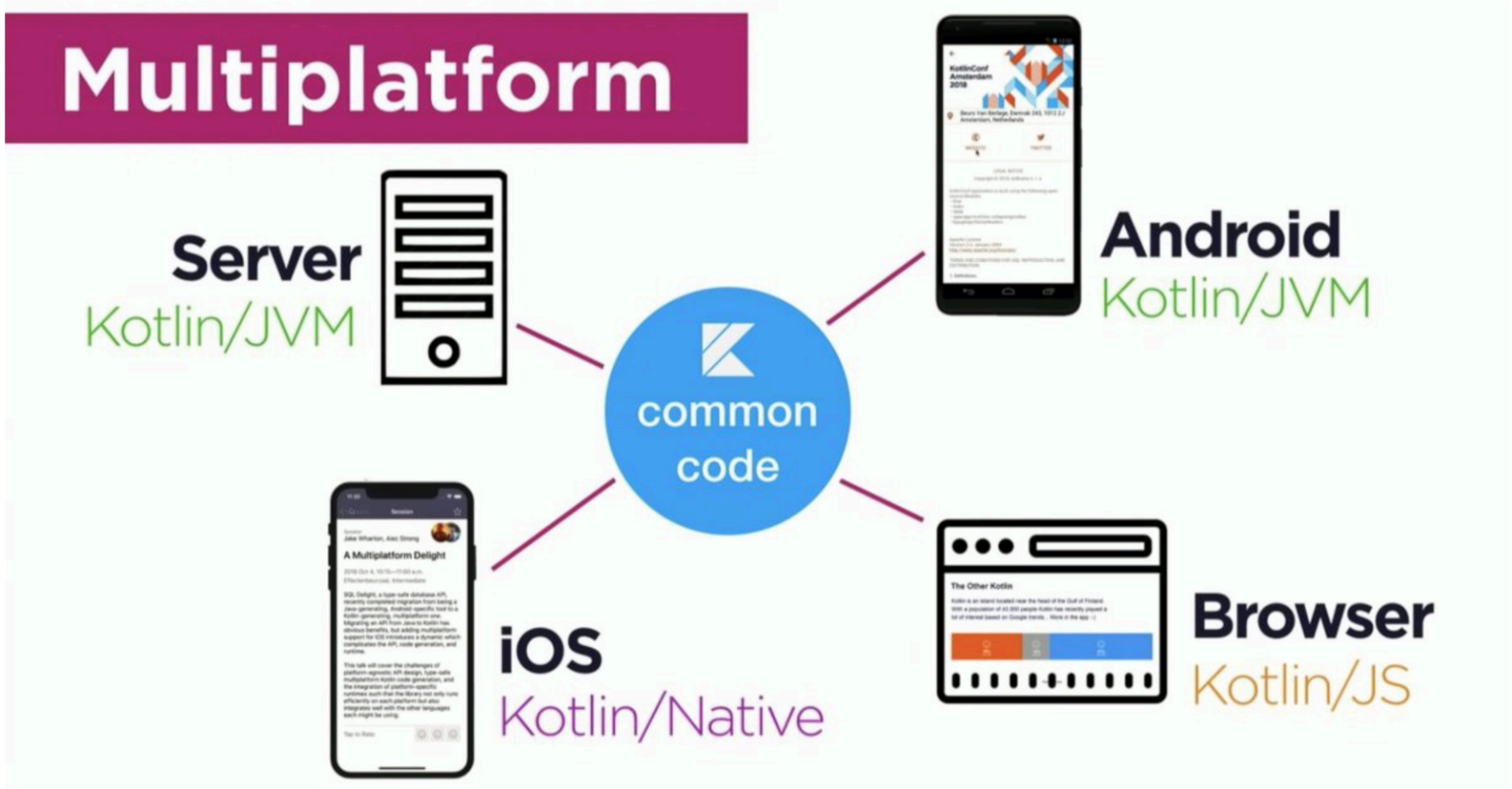
- Debuggear para iOS
- Hacerlo en Android primero

Beneficios



- 1 código base que escribir, testear, mantener
- Errores centralizados
- Implementaciones de diseño nativas
- Android - iOS - Web - Backend (1 sola base de código)

Beneficios


Multiplatform
Server
Kotlin/JVM
common
code
Android
Kotlin/JVM
iOS
Kotlin/Native
Browser
Kotlin/JS

Resumen

- Piensa en compartir
- La comunicación lo es todo
- Empodérate
- Comparte lo que ambas plataformas estén dispuestas
- Empieza de a pocos
- Construye confianza
- Con el tiempo el impacto aumenta

KOTLIN MULTIPLATFORMA EN ACCIÓN

LAURA DE LA ROSA

Kotlin Medellín User Group

Redes: @lauramdelarosa