



iOS app development

Denisse Maldonado
denisse.mf@tec.mx

Tec de Monterrey - Campus Querétaro
TC2024 - Desarrollo de Aplicaciones Móviles

El entorno de apps iOS

Julio de 2008

- 1 SWIFT
- 2 XCODE
- 3 APP STORE



1er sistema operativo en mostrar un teclado virtual





SWIFT

Lenguaje Multiparadigma

VARIABLES

MUTABLE:

```
var police: String!           String
```

```
var detective: Bool = true    Int
```

NON-MUTABLE:

```
let placa: String = "9KD"
```

```
let vehiculo = true
```

TIPOS DE DATOS:

Double

Bool

FUNCIONES

```
func arrestar(direccion: String, policias: Int) {  
    print("Buscando al sospechoso en \(direccion)  
    con un equipo táctico de \(policias)  
    policias")  
}  
  
arrestar(direccion: "Epigmenio Gonzalez  
#500", policias: 100)
```

FUNCIONES

```
func interrogar(prueba: String) -> Bool {  
    print("La prueba \(prueba) corresponde al  
asesino")  
    return false  
  
}  
  
print(interrogar(prueba:"Arma de calibre  
45"))
```

ARREGLOS

```
var evidencias = [String]()
evidencias.push("Arma Calibre 45")

func interrogar(prueba: String, evidencia:
[String]) -> Bool {

    for n in 0...evidencia.length-1 {
        if(prueba == evidencia[n]) {
            return true
        }
    }
    return false
}
```

SETS

```
var pruebas = ["Arma Calibre 30", "Gorra",
    "Huella digital", "Sangre", "Zapatos", "Gorra"]

let unique = Array(Set(pruebas))

print(unique)

>> ["Gorra", "Sangre", "Zapatos", "Arma Calibre
30", "Huella digital"]
```

CLOSURES

```
let encerrar = { (placa: String, delincuentes:  
Int) -> Bool in  
  
    return (placa == "9KD" || placa == "1LP")  
    && delincuentes < 5  
  
}  
  
encerrar("9KD", 1)
```

PROTOCOLOS

```
protocol Criminal {  
    func condena()  
}  
  
class Asesino:Criminal {  
    func condena() {  
        print("Cadena perpetua")  
    }  
}  
  
class Ladron:Criminal {  
    func condena() {  
        print("Encerrar por 5 años")  
    }  
}
```

PROTOCOLOS VS OOP

```
protocol Criminal {  
    func condena()  
}  
  
protocol Colabora {  
    func proteccion()  
}  
  
class Narco:Criminal, Colabora {  
    func condena() {  
        print("Encerrar por 5 años")  
    }  
    func proteccion() {  
        print("Protección de Testigos")  
    }  
}
```

```
class Criminal {  
    func condena()  
}  
  
class Ladron:Criminal {  
    override func condena() {  
        print("Encerrar por 5 años")  
    }  
}  
  
class Narco:Criminal {  
    override func condena() { print("Encerrar  
por 5 años") }  
  
    func colabora() { print("Protección de  
Testigos") }  
}
```

FUNCIONAL VS IMPERATIVA

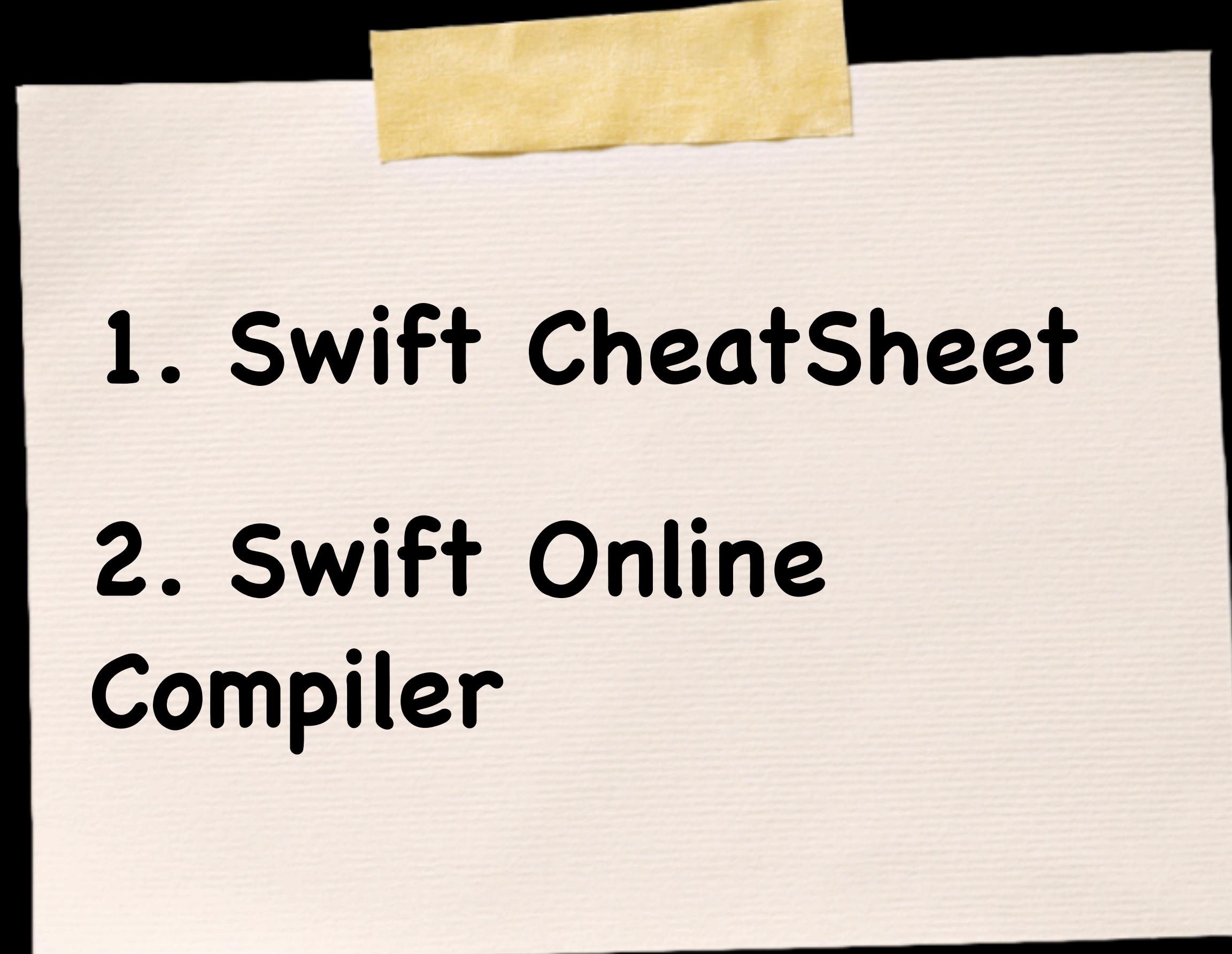
```
let celdas = [12, 19, 48, 93, 7]  
  
les celdasLibres = celdas.filter  
{ $0 % 2 == 0 }  
  
print(celdasLibres)  
  
>> [12, 48]
```

```
let celdas = [12, 19, 48, 93, 7]  
  
var celdasLibres = [Int]()  
  
for celda in celdas {  
  
    if(celda % 2 == 0) {  
  
        celdasLibres.push(celda)  
  
    }  
  
}  
  
print(celdasLibres)
```



Recursos de Apoyo



- 
1. Swift CheatSheet
 2. Swift Online Compiler

iOS ARCHITECTURE

Exploring the iOS Software Development Kit

Cocoa Touch

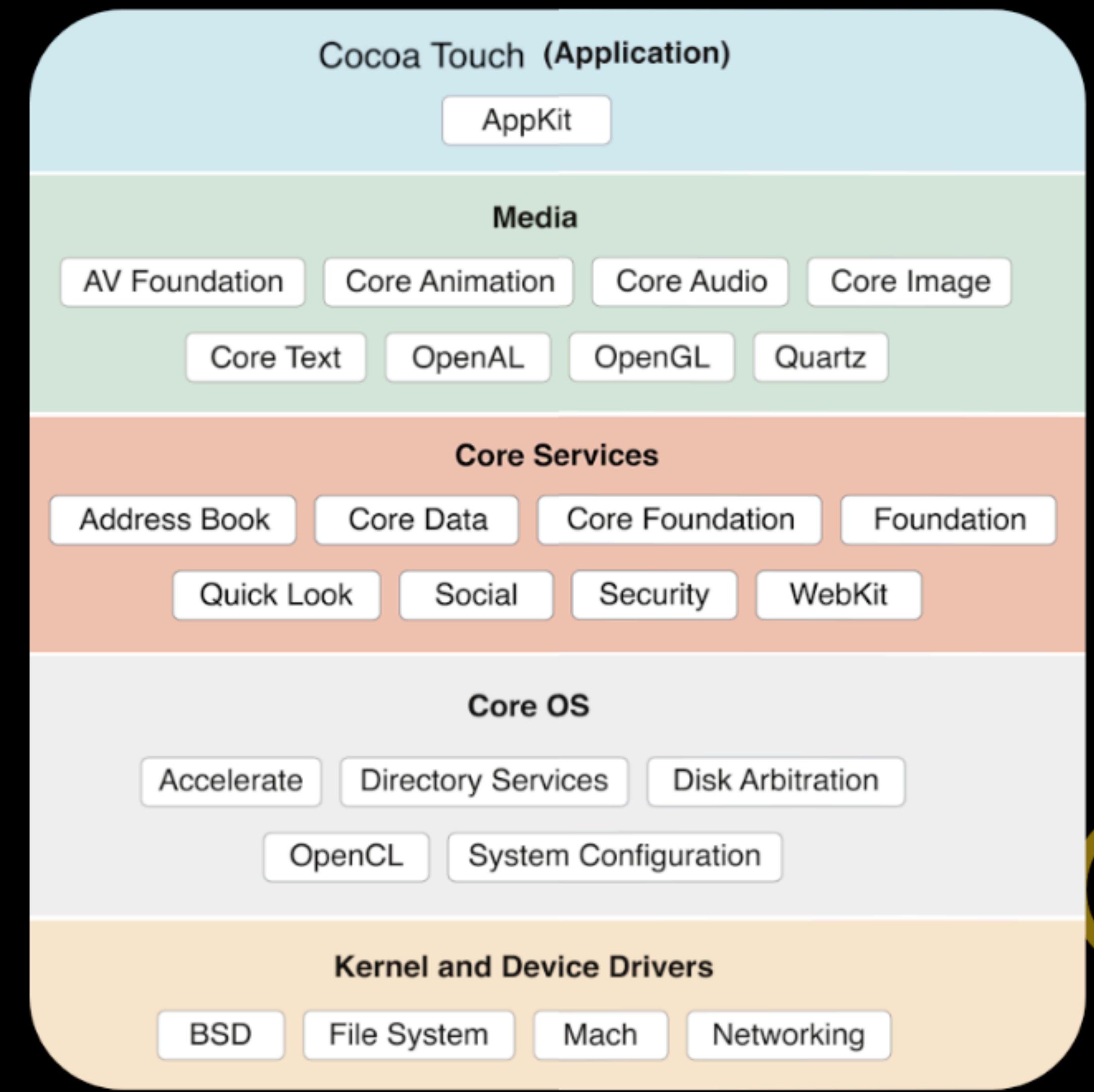
UIKit

Media

Core Services

Foundation

Core OS



APPKIT

Construye y maneja de manera gráfica
y a través de eventos del usuario la
interfaz de la aplicación.

AV FOUNDATION

Framework que proporciona la facilidad de
integrar, escuchar y editar elementos
audiovisuales en tu aplicación

CORE DATA

Es un marco de persistencia en el que se
pueden guardar objetos de manera
estructurada.

SYSTEM CONFIGURATION

Permite a las aplicaciones el acceso a las configuraciones de red. Uso de Wi-Fi y conectividad del teléfono.

FILE SYSTEM

Acceso al árbol de archivos del sistema
operativo