

Desarrollo Android

Introducción

Quiénes somos?



Bruno Frachia



Ingeniero en Informática y CTO de Houlak desde su fundación. Especialista en desarrollo mobile

brunoa.frachia@ucu.edu.uy



Matías González



Analista Programador y Android Developer en Houlak desde hace 4 años.
Licenciado en Negocios Internacionales.

matias.gonzalez@ucu.edu.uy

Y vos?



Objetivos del curso

Objetivos del curso



ANDROID DEV JR

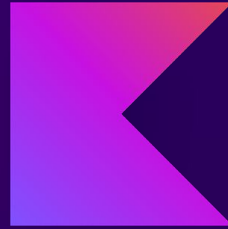
Objetivos del curso



CONSTRUIR APPS
DESDE CERO



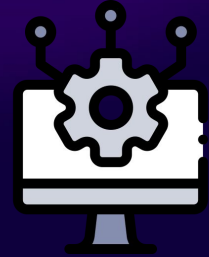
CLEAN CODE



KOTLIN



ARQUITECTURA



APIs

**Qué
esperamos de
ustedes?**

COMPROMISO

**GANAS DE
APRENDER**

Temario del curso

Temario del curso

- Android Studio
- Conceptos fundamentales del lenguaje Kotlin
- Clean code
- Estructura de archivos y carpetas de un proyecto Android
- Cómo construir interfaces de usuario
- Navegación
- Librerías y dependencias
- Comunicación con APIs
- Almacenamiento de datos simples (Preferences)
- Almacenamiento de datos complejos (Base de datos)
- Control de versiones

Evaluación

Evaluación

Actitud y
participación
en clase

Tareas
prácticas

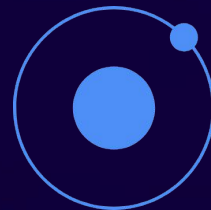
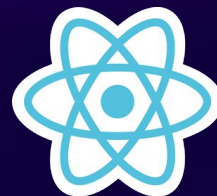
Desafíos

**Qué caminos tengo
para desarrollar
apps Android?**

DESAROLLO NATIVO



DESAROLLO HÍBRIDO



Depende

The logo graphic for Kotlin, featuring a stylized 'K' shape. The left vertical bar of the 'K' is a solid magenta color. The right diagonal bar is split: the upper half has a gradient from magenta to red, and the lower half has a gradient from magenta to blue. The word 'Kotlin' is written in white, bold, sans-serif font, centered horizontally and partially overlaid by the magenta bar of the 'K' graphic.

Kotlin

Kotlin

Creado por
JetBrains

Corre en la JVM
(interoperabilidad)

Null safety

Código más
limpio

Java

POJO

Kotlin

M

```
class Person {  
    private String name;  
  
    public Person(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    // toString...  
  
    // hashCode...  
  
    // equals...  
  
    // copy...  
}
```

```
data class Person(val name: String)
```

Java

Code

Kotlin

M

```
public void createAndPrintPerson() {  
    String name = "Pieter";  
    Person person = new Person(name);  
  
    printName(person.getName());  
    // Prints: Pieter Otten  
}
```

```
fun createAndPrintPerson() {  
    val name = "Pieter"  
    val person = Person(name)  
  
    printName(person.name)  
    // Prints: Pieter Otten  
}
```

Y ahora?

Ejercicio

Crear una lista de personas con los siguientes datos:

- Jorge Perez, 26 años
- Alberto Rodriguez, 30 años
- Norma Pires, 42 años
- Tomás Lorenzo, 12 años
- Alejandra Romero, 29 años
- Norberto Araujo, 68 años
- Omar Junin, 17 años
- Juana Batista, 34 años

A partir de este

- Imprimir los nombres ordenados según la edad (de menor a mayor)
- Imprimir los nombres ordenados según el largo del nombre

Ejercicio 2

Crear una lista de personas con los siguientes datos:

- Jorge Perez, 26 años, Pintor
- Alberto Rodriguez, 30 años, Maestro
- Norma Pires, 42 años, Ingeniera
- Tomás Lorenzo, 12 años, Estudiante
- Alejandra Romero, 29 años, Abogada
- Norberto Araujo, 68 años, Jubilado
- Omar Junin, 17 años, Estudiante
- Juana Batista, 34 años, Escritora

Por cada persona menor a 18 años imprimir en los logs una frase que diga contenga su nombre y su edad.

Por cada persona de entre 18 y 60 años imprimir en los logs una frase que contenga su nombre, edad y profesión.

Por cada persona mayor a 60 años imprimir en los logs una frase que contenga su nombre y su edad.