



Introducción al desarrollo de aplicaciones híbridas

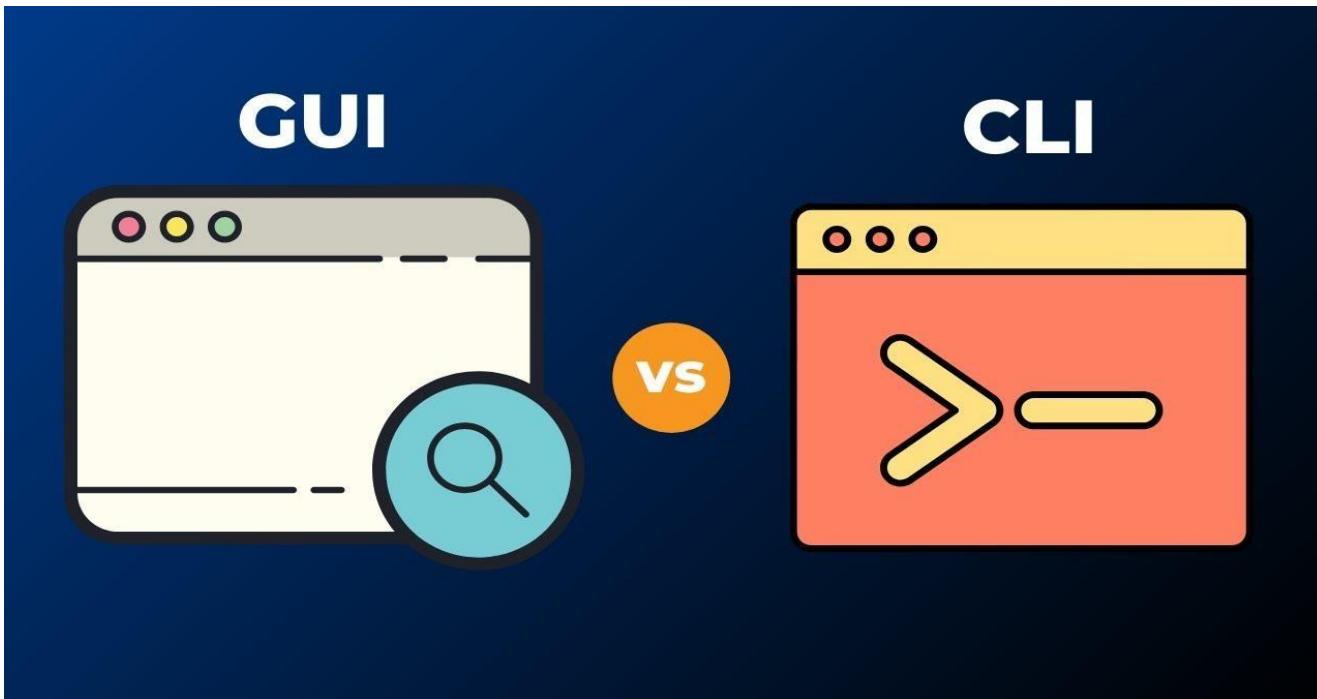
UT4 - Introducción al desarrollo de aplicaciones híbridas con Ionic



Objetivos de aprendizaje

- Comprender el concepto de cross-platform y sus ventajas respecto a un desarrollo convencional
- Distinguir las distintas tecnologías/lenguajes usadas para desarrollar

¿Qué opciones tenemos para desarrollar GUIs?

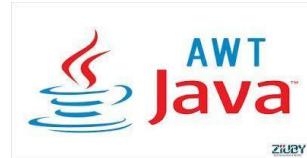


Desarrollo para escritorio (PC)

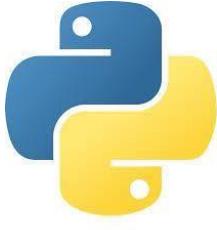
- Implementamos y compilamos una aplicación específicamente para un Sistema Operativo determinado
- A la lógica de la aplicación le añadimos una capa de interfaz (GUI, Graphical User Interface) que proporciona las vistas y la interacción con el usuario



Desarrollo para escritorio (PC)

| Lenguaje | Principales tecnologías para GUI |
|--|---|
|  |    |
|     |   |

Desarrollo para escritorio (PC)

| Lenguaje | Principales tecnologías para GUI |
|--|---|
|  |  |
|  |  Tkinter |

Desarrollo para escritorio (PC)

| Ventajas | Inconvenientes |
|--|---|
| Generalmente optimizadas para la plataforma de destino | La GUI es completamente dependiente del lenguaje que he escogido |
| Tecnologías maduras e históricamente con mucho uso | Muchas de las tecnologías no han evolucionado o no se han mantenido correctamente |
| | No siempre los programas son portables (ej. .NET no era portable a Linux hasta hace poco) |
| | <u>Modelo en desaparición</u> |



Desarrollo nativo (móvil)

- Desarrollamos y compilamos aplicaciones con tecnologías específicas para una plataforma

| Sistema Operativo | Lenguajes | Herramientas |
|--|---|--|
| Android  |  Java  Kotlin  GO |  Android Studio  Android IDE  IntelliJ IDEA |
| iOS  |  Swift  Objective - C |  XCode  AppCode  Atom |

Valorando el desarrollo nativo

| Ventajas | Inconvenientes |
|--|---|
| Aplicaciones más eficientes al estar desarrolladas específicamente para dicha plataforma | Necesidad de aprender un lenguaje separado para cada plataforma |
| Acceso sencillo a las funcionalidades del dispositivo (cámara, GPS, lista de contactos, giroscopio, etc..) y características avanzadas | Aumento en costes y menos productividad (doble mantenimiento, doble velocidad de desarrollo...) |
| GUI de la plataforma de destino | |
| Más seguridad y confiabilidad | |

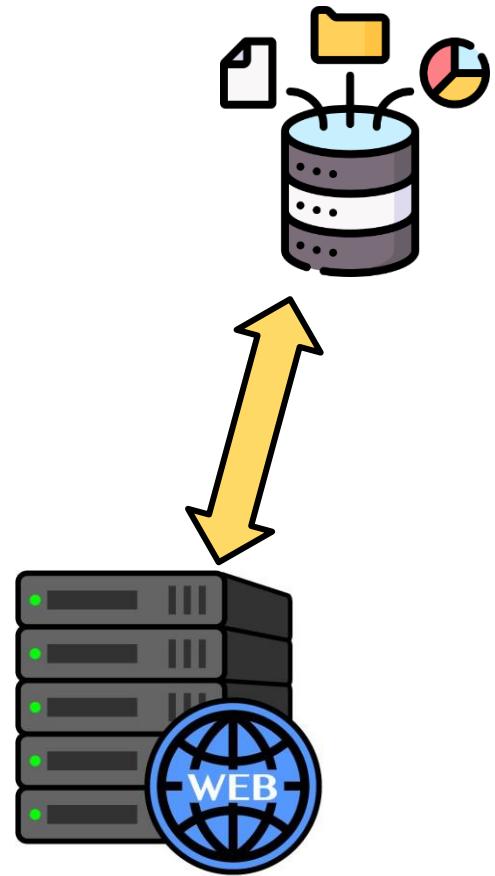
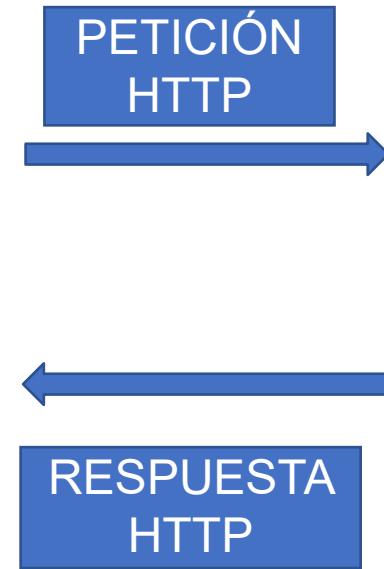


Desarrollo Web

- Dividimos nuestro código en dos partes:
 - Parte servidora
 - Código que se ejecuta en un equipo remoto (servidor).
 - Funciones principales es el manejo de datos (acceso a bases de datos), autenticación, seguridad...
 - El usuario no tiene acceso al código de esta parte
 - Parte cliente
 - Código que se ejecuta en mi propio dispositivo (navegador Web)
- El diálogo entre cliente y servidor se establece a través del protocolo HTTP



Modelo Web



©DESIGNALIKIE

Tecnologías Web

| | |
|---------------|---|
| Lado Cliente |   |
| Lado Servidor |   |

Frameworks

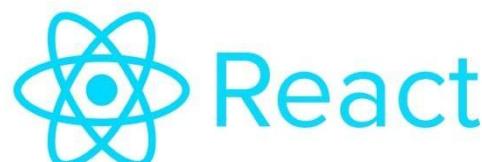
- Un framework es un conjunto de utilidades probadas y validadas que me permiten solucionar problemas que me encuentro de manera común al realizar una aplicación.
 - Organización de archivos, separación de funcionalidad, manejo de rutas, seguridad, acceso a datos, manejo de usuarios....
- Cada plataforma de desarrollo suele tener su propio framework asociado.





Frameworks JS

- Actualmente está muy extendido el uso de 3 frameworks JS
- Son similares pero cada uno tiene sus particularidades y curva de aprendizaje

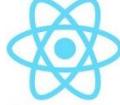
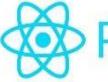


Desarrollo híbrido

- Se programa en un único lenguaje que me sirva para varias plataformas
 - Generalmente se trata de lenguajes Web
- Una vez finalizado el desarrollo, se “envuelve” dentro de una aplicación nativa
 - Generalmente esta aplicación se ejecuta en el marco de un navegador Web.



Tecnologías de desarrollo híbrido

| Tecnología | Lenguajes de programación/Frameworks UI | | | |
|---|---|---|---|--|
|  Ionic |  |  ANGULAR |  React |  Vue.js |
|  React Native  Meta |  |  React | |  Java |
|  Flutter  | |  | Dart | |

Tecnologías de desarrollo híbrido (II)

| Tecnología | Lenguajes de programación/Frameworks UI |
|---|--|
|  Xamarin  .NET MAUI  Microsoft |   |



Valorando el desarrollo híbrido

| Ventajas | Inconvenientes |
|--|---|
| Coste asociado a desarrollo mucho menor | Peor rendimiento de las aplicaciones (cada vez mejor) |
| Coste asociado a mantenimiento mucho menor | Pueden ser más complicadas de depurar |
| Tecnologías útiles en otros mercados | Interfaz de usuario menos personalizada |
| | Más dependientes de Internet |



¿Por qué una aplicación híbrida es más dependiente de Internet?



Resumiendo

| | | | | | | |
|--------------------|-----|-----------------------|-----------------|--|--|--|
| Mobile Apps | iOS | | | | | |
| Web Frontend | | | | | | |
| Web Backend | | | | | | |
| Databases | | Express | | | | |
| Cloud/ Integration | | | | | | |
| | | | | | | |
| | | | | | | |
| | | Google Cloud Platform | Microsoft Azure | | | |