



TIPOS DE APLICACIONES Y TECNOLOGÍAS WEB

Ubicamos los contenidos del curso dentro del ámbito del desarrollo actual de aplicaciones

MOMENTO ACTUAL (I)

- En los últimos años no ha parado de crecer el número de dispositivos que utilizamos para trabajar y para conectarnos a Internet.
- Algunos ejemplos:
 - Móviles
 - Tablets
 - Smart TV
 - Relojes inteligentes
 - IOT

MOMENTO ACTUAL (II)

- Asociados a estos dispositivos también han aparecido nuevos sistemas operativos.
- Ejemplos:
 - Android
 - IOS
- Eso ha hecho que aparezcan multitud de nuevos términos para poner nombre a una nueva realidad que necesitamos entender.

APLICACIONES NATIVAS

- Uno de esos términos que cobra importancia para los desarrolladores es el de aplicaciones nativas o “*native apps*”.
- Una aplicación nativa es un programa que ha sido desarrollado para un sistema o dispositivo en particular.
- Ejemplos:
 - Las aplicaciones Windows clásicas.
 - Aplicaciones para IOS o Android.

ALGUNAS OPCIONES (I)

- Ecosistema Apple
 - *Swift es un lenguaje de programación poderoso e intuitivo creado por Apple para desarrollar apps de iOS, Mac, Apple TV y Apple Watch.*
 - Herramienta de desarrollo: X-Code
 - <https://www.apple.com/la/swift/>
- Android
 - Java
 - Kotlin
 - Herramienta de desarrollo: Android Studio
 - <https://developer.android.com/training/basics/firstapp?hl=es-419>

ALGUNAS OPCIONES (II)

- Windows
 - Windows Forms: Visual Basic, C#.ul> - <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-7.0>
 - WPF: Visual Basic, C#.ul> - <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-7.0>
- Herramienta de desarrollo: *Visual Studio*
- Linux
 - C++
 - Python
 - <https://developer.gnome.org/documentation/introduction/languages.html>
 - Herramienta de desarrollo: *Gnome*

APLICACIONES NATIVAS.

VENTAJAS

- Ventajas:
 - Al estar pensadas para una plataforma determinada aprovechan el SO anfitrión.
 - Tienen un mejor rendimiento.
 - Permiten usar mejor el hardware del dispositivo, sobre todo en las apps para dispositivos móviles (cámara, gps...).
 - Se benefician antes de las actualizaciones en los sistemas.

APLICACIONES NATIVAS INCONVENIENTES.

- Inconvenientes:
 - Si se quiere una app “universal” hay que desarrollar diferentes versiones de la app para cada sistema o dispositivo.
 - Se disparan los costes y los tiempos.
 - Cada vez hay más dispositivos diferentes.
- **Solución: Frameworks multiplataforma**

FLUTTER

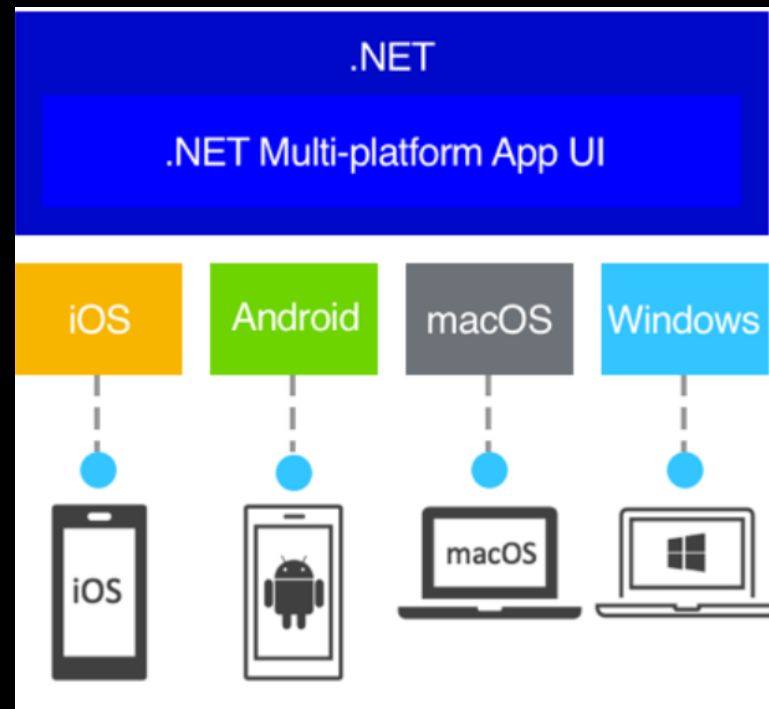
- Es un framework desarrollado por Google que permite crear aplicaciones móviles tanto para Android como para iOS.
 - Muy potente en la creación de interfaces gráficas.
 - Desarrollo rápido.
 - Lenguaje: *Dart*
 - <https://flutter.dev/>
- Herramienta de desarrollo: *Visual Studio Code*

REACT NATIVE

- Es un framework desarrollado por Facebook/Instagram que está basado en *React*.
 - Sencillo de aprender si se conoce *React*.
 - Lenguajes: *JavaScript* o *TypeScript*
 - <https://reactnative.dev/>
- Herramienta de desarrollo: *Visual Studio Code*

.NET MAUI

- **M**ulti-platform **A**pp **U**ser **I**nterface
- Es un framework desarrollado por Microsoft.
- Sencillo de aprender para el desarrollador en .Net
 - Lenguajes: C#
 - <https://learn.microsoft.com/es-es/dotnet/maui/what-is-maui>
- Herramienta de desarrollo: *Visual Studio*



APLICACIONES EJECUTADAS EN EL NAVEGADOR (I)

- Hoy en día los navegadores son sistemas operativos.
- De hecho, la idea de Google con Google Chrome desde el principio era hacer de su navegador un SO (Chrome OS)
- Se puede desarrollar cualquier app para el navegador gracias a *HTML5*, *CSS3* y *JavaScript*.
- Nuestros contenidos se ubican aquí.

APLICACIONES EJECUTADAS EN EL NAVEGADOR (II)

- Ventajas:
 - Estas apps se pueden ejecutar en cualquier dispositivo porque todos tienen un navegador para conectarse a Internet.
 - Una única app para todo.
 - No hace falta aprender multitud de lenguajes.
 - En empresas, no hacen falta varios equipos de trabajo expertos en diferentes lenguajes y plataformas.

APLICACIONES EJECUTADAS EN EL NAVEGADOR (III)

- Ventajas:
 - Se benefician del desarrollo imparable de los estándares web y de las nuevas herramientas para el mundo web.
 - Al conocer los estándares, el aprendizaje de estas herramientas no es complicado.
 - Hoy en día no se contempla que los estándares expiren, luego estas aplicaciones en principio tendrían una vigencia perpetua.

APLICACIONES EJECUTADAS EN EL NAVEGADOR (IV)

- Inconvenientes:
 - Se añade una nueva capa en el funcionamiento del programa, el navegador.
 - La aplicación se conecta al navegador, que a su vez se conecta al SO. Se resiente algo el rendimiento.
 - Las aplicaciones nativas se conectan directamente al SO.
 - Problemas para acceder a algunos elementos del hardware del dispositivo si se quiere utilizar en la app elementos muy concretos (cámara, gps...).

TECNOLOGÍAS

- *HTML5, CSS3 y JavaScript*
- Frameworks:
 - *Angular*
 - <https://angular.io/>
 - *React*
 - <https://es.react.dev/>
 - *Vue*
 - <https://vuejs.org/>
- Estadísticas de uso
 - <https://2022.stateofjs.com/en-US/>
 - <https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/>
 - <https://survey.stackoverflow.co/2023/>
- Herramienta de desarrollo: *Visual Studio Code*

APLICACIONES HÍBRIDAS

- Las aplicaciones híbridas suponen otra alternativa a las nativas y a las orientadas al navegador.
- Se ejecutan en el sistema nativo, pero se ejecutan en un navegador interlo
- Utilizan *HTML5* y *CSS3* para las vistas.
- Utilizan *JavaScript* para acceder a funcionalidades de los dispositivos como la cámara, geolocalización, acelerómetro...

APLICACIONES HÍBRIDAS. VENTAJAS

- Su rendimiento es parecido al de las aplicaciones nativas.
- Son universales para todos los dispositivos.
- Es la alternativa ideal para aplicaciones orientadas a dispositivos móviles por la cantidad de elementos del hardware a los que permite acceder la API JavaScript.
- Para programarlas sólo hace falta conocer JavaScript.
- Las vistas se construyen con HTML5 y CSS3, con todo lo que eso supone.

APLICACIONES HÍBRIDAS. INCONVENIENTES

- Inconvenientes:
 - Necesitan un navegador interno, no es nativo 100%.
 - No es seguro que se pueda acceder a todas las funcionalidades de cada dispositivo.
 - Actualizaciones más complicadas.

FRAMEWORKS PARA APLICACIONES HÍBRIDAS

- *Ionic*
 - Un único desarrollo para:
 - IOS
 - Androkd
 - PWA
 - Aplicaciones de escritorio (gracias a *Electron*)
 - <https://ionicframework.com/>
- *Electron*
 - Un único desarrollo para *Windows*, *Mac* y *Linux*.
 - <https://www.electronjs.org/es/>
 - *Whatsapp*, *Teams*, *VS Code* están hechos con *Electron*,
- Herramienta de desarrollo: *Visual Studio Code*

PWA

- Pretenden que las aplicaciones web logren los mismos objetivos de las aplicaciones nativas.
- Wikipedia: *“Aplicaciones web que, de manera progresiva, incorporan funcionalidades y tienen un rendimiento similar al de las aplicaciones nativas”*
- ¿Cómo es posible que una aplicación web pueda hacer lo mismo (casi) que una aplicación nativa?
 - <https://whatwebcando.today/>

PWA. CARACTERÍSTICAS.

- El adjetivo progresiva tiene que ver con la capacidad de la aplicación de aumentar progresivamente sus capacidades según el dispositivo en el que se ejecute.
 - Si nuestra aplicación se ejecuta en un dispositivo que permita ejecutarla hasta un cierto nivel, esta funcione hasta donde pueda llegar.
 - Si ejecuta en otro que permita expresar todas sus capacidades al máximo la aplicación alcance ahí su máximo rendimiento.

PWA VS APP HÍBRIDA

- No son muchas.
- Ambas se desarrollan con tecnologías web.
- Las apps híbridas se desarrollan en plataformas orientadas a dispositivos móviles.
- Las PWA forman parte del desarrollo web normal.
- Estas últimas están pensadas para quienes ya conocen una web y la quieren en su versión móvil.

BACKEND

- En (casi) todas las aplicaciones vamos a necesitar un *backend*.
- Algo que nos permita:
 - Gestionar usuarios/seguridad.
 - Acceder a bases de datos.
 - Gestionar archivos (subir/descargar)
 - Tiempo real.
 - ...

BACKEND. TECNOLOGÍAS Y LENGUAJES

- Java
 - Spring
- Php
 - Symfony
 - Laravel
- .Net C#
 - ASP
- JavaScript
 - NodeJS
- Python
 - Django
- Firebase

OTROS TIPOS DE APLICACIONES

- Industria 4.0
 - Aplicaciones para microcontroladores.
 - Ecosistema muy variado, a menudo multimarca.
 - Arduino con C++.
- Videojuegos
 - *Unity*
 - C#
 - Herramienta de desarrollo: *Unity + Visual Studio*
 - *Unreal*
 - C++
 - Herramienta de desarrollo: *Unreal engine*