DESARROLLO DE APLICACIONES 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 (I)

- 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.

APLICACIONES NATIVAS (II)

- 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 (III)

- 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.

ALGUNOS LENGUAJES

- IOS
  - Swift
- Android
  - Android (derivado de JAVA)
  - Kotlin
- Windows
  - Windows Forms: Visual Basic, C#.
  - WPF: Visual Basic, C# + XAML para las vistas.
- Linux
  - Java
  - Phyton

APLICACIONES EJECUTADAS EN EL NAVEGADOR (I)

- Una primera solución a este problema es el uso del navegador.
- 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…).

ALGUNOS LENGUAJES

- Del lado del cliente:
  - HTML5, CSS3 y JavaScript
  - Frameworks *front-end*: Angular, React, Vue.
- Del lado del servidor (si se necesita):
  - Lenguajes .Net (Visual Basic, C#)
  - PHP
  - Phyton
  - Java
  - JavaScript utilizando como servidor Node.js

WEBASSEMBLY (WASM)

- Es un nuevo tipo de desarrollo ejecutable en navegadores modernos.
- Está estandarizado por W3C pero a efectos prácticos todavía le falta tiempo de adaptación.
- Es un lenguaje de bajo nivel, parecido al ensamblador, de formato binario.
- A ese binario se llega compilando lenguajes conocidos, como C++, si bien el abanico se está extendiendo a otros como C#.
- En el navegador se ejecuta entre 10 y 30 veces más rápidos que un lenguaje interpretado como JavaScript.
- Está diseñado para complementar JavaScript y ejecutarse a la par.

APLICACIONES HÍBRIDAS (I)

- Las aplicaciones híbridas suponen otra alternativa a las nativas y a las orientadas al navegador.
- Se ejecutan en el sistema nativo aunque las vistas siguen siendo web.
- Utilizan HTML5, CSS3 para las vistas.
- Utilizan una API JavaScript para acceder a funcionalidades de los dispositivos como la cámara, geolocalización, acelerómetro…
- Los archivos de la aplicación están en el dispositivo local y no en un servidor.

APLICACIONES HÍBRIDAS (II)

- 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, a no ser que se necesite acudir a un servidor web para obtener datos.
  - Las vistas se construyen con HTML5 y CSS3, con todo lo que eso supone.

APLICACIONES HÍBRIDAS (III)

- Inconvenientes:
  - Siguen necesitando capas extra, no es nativo 100%.
  - No es seguro que se pueda acceder a todas las funcionalidades de cada dispositivo.
  - Actualizaciones más complicadas.

APLICACIONES HÍBRIDAS (IV)

- Algunas de estas APIs son:
  - React Native
  - Flutter
  - Ionic (ionic.io)