



GRUPO:
2º DAM TARDE

AUTOR:
Alejandro Limiñana Quintana

PROYECTO FULLSTACK

ÍNDICE

Descripción de la aplicación	
Pila tecnológica.....	
Diagrama entidad-relación.....	
Modelo de datos representando las clases gráficamente con UML.....	
Modelo relacional.....	
Descripción de cada tabla/entidad.....	
Mockups.....	
Documentación Postman.....	
Usabilidad.....	
Manual de instalación.....	
Código del proyecto completo	
Comparación de tecnologías	

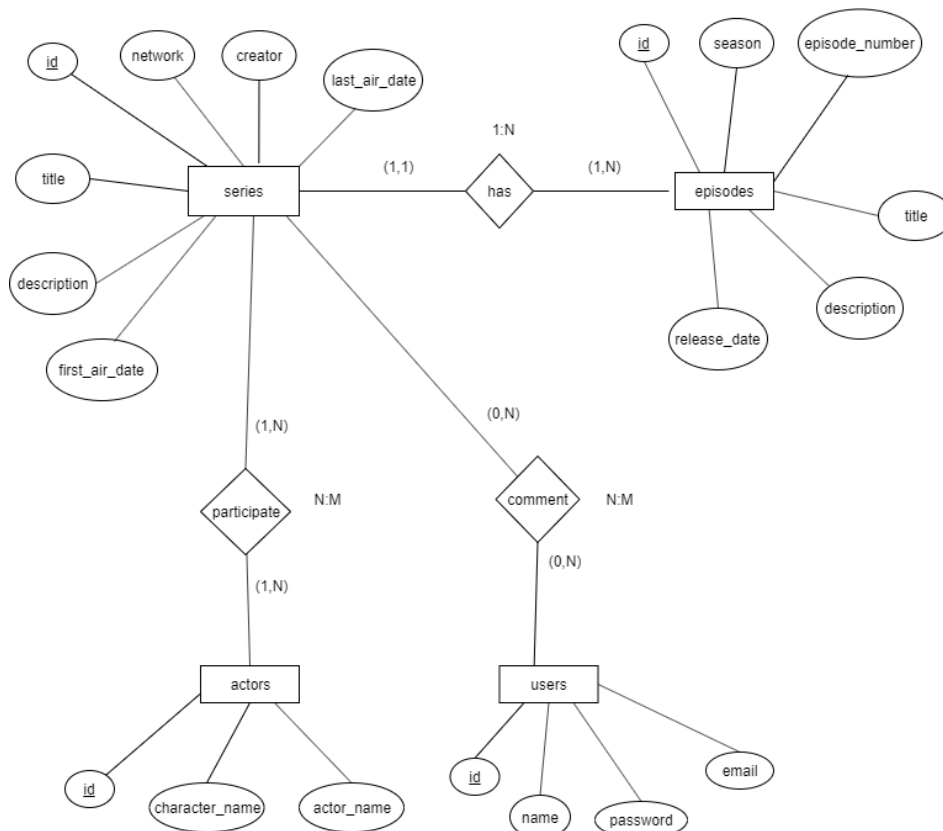
Descripción de la aplicación

La aplicación a desarrollar tratará sobre series. Se mostrarán las series más populares, pudiendo ver información sobre cada una de ellas, además de los actores que componen la serie y los episodios de los que consta. Habrá un apartado de comentarios que solo se podrá usar si primero nos registramos en la aplicación y después nos identificamos.

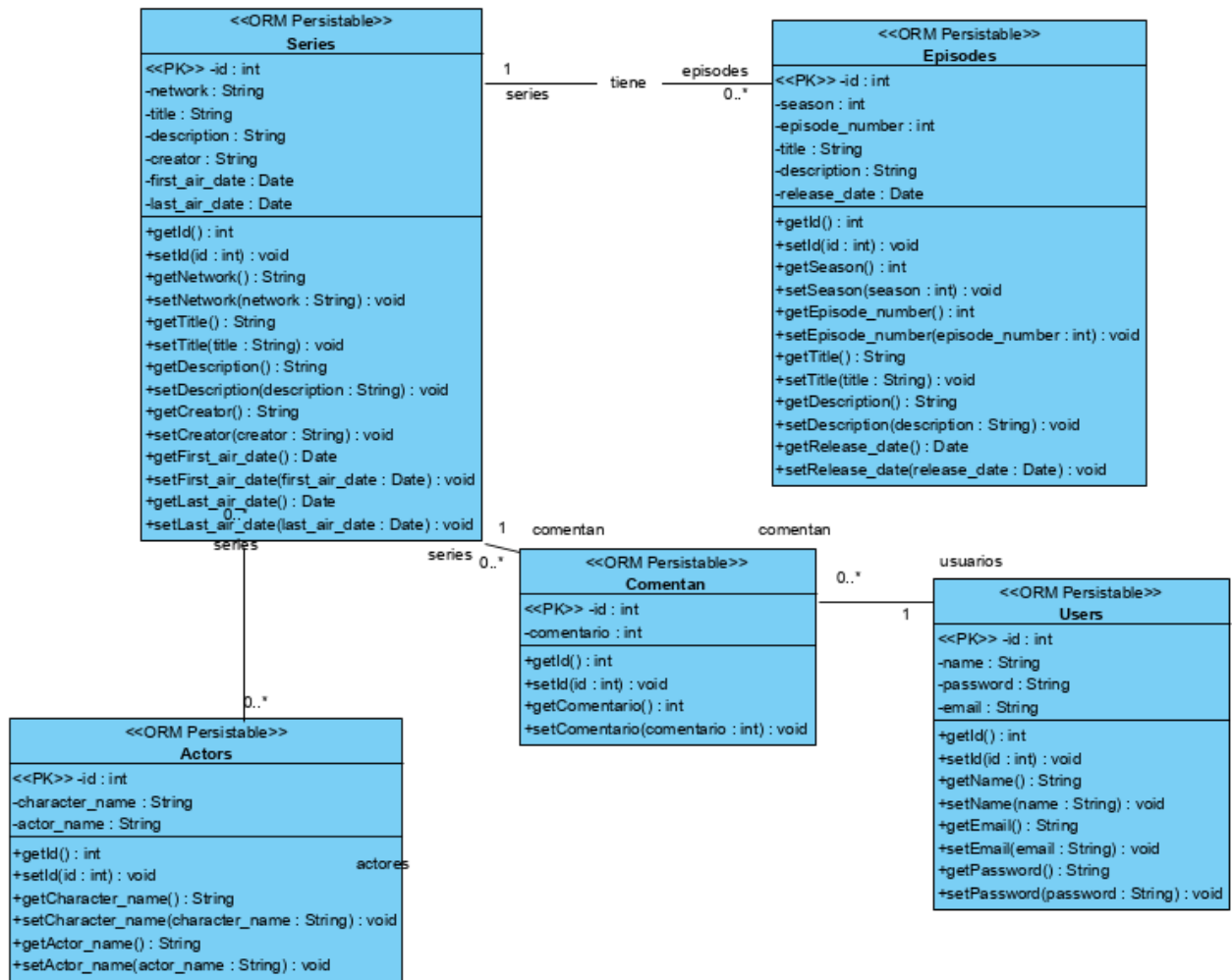
Pila tecnológica



Diagrama entidad relación



Modelo de datos representando las clases gráficamente con UML



Modelo relacional

series (id, network, title, description, creator, first_air_date, last_air_date, id_episodes*)

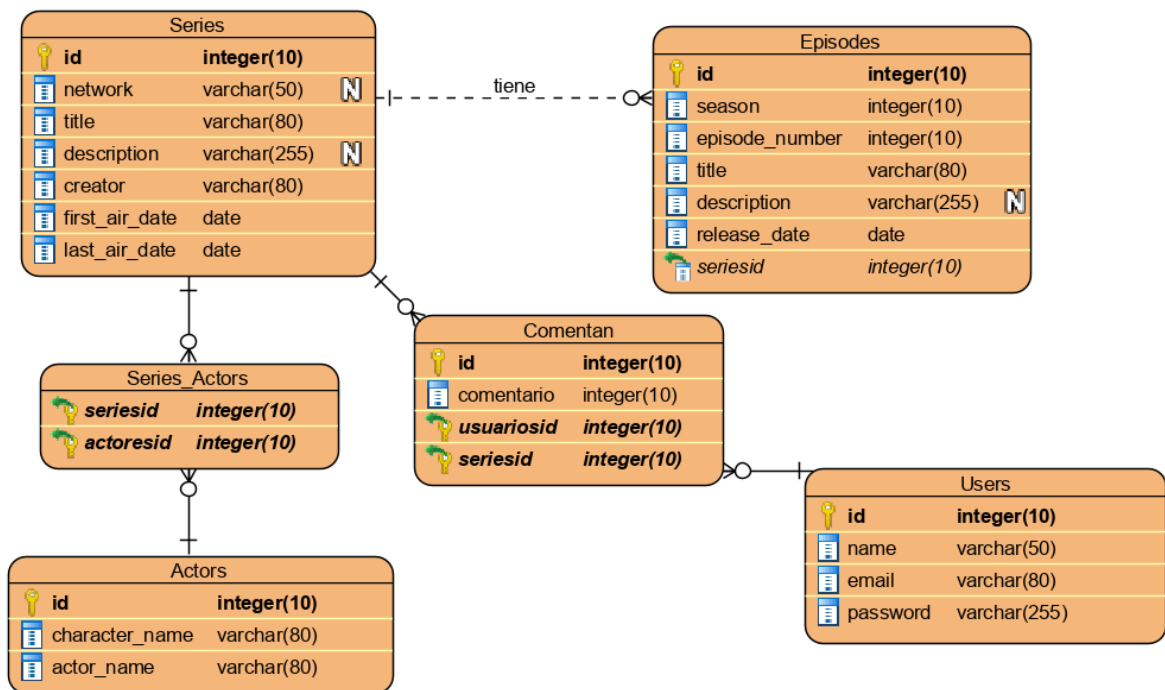
episodes (id, season, episode_number, title, description, release_date)

actors (id, character_name, actor_name)

participate (id_series*, id_actors*)

users (id, name, password, email)

comment (id_series*, id_users*)



Descripción de cada tabla/entidad

Entidades de la base de datos “tv_shows”.

- **Series:** almacena información básica sobre las series.
- **Episodes:** almacena las temporadas y los episodios que contiene cada serie.
- **Actors:** almacena los actores que participan en la serie, así como el personaje al que interpretan.
- **Comment:** almacena los comentarios que los usuarios hagan en el apartado de una serie.
- **Users:** almacena toda la información de los usuarios. Si un usuario no se registra y posteriormente inicia sesión, no se le permitirá comentar.
- **Series_Actores:** entidad que se genera dada la relación de muchos a muchos entre la tabla series y la tabla actores. Almacena las claves primarias y foráneas de cada entidad.

Atributos:

Series:

- id: clave primaria
- network: cadena que emite cada serie (puede ser nulo).
- title: nombre de la serie.
- description: descripción de la serie (puede ser nulo).
- creator: creador de la serie.

Episodes:

- id: clave primaria
- season: temporadas de las que consta.
- episode_number: número de episodio al que hace referencia en la temporada.
- title: nombre de la serie.
- description: descripción de la serie (puede ser nulo).
- release_date: fecha en la que se emitió cada episodio.

Actors:

- id: clave primaria
- character_name: nombre del personaje que interpreta el actor.
- actor_name: nombre del actor.

Users:

- Id: clave primaria
- name: nombre del usuario.
- email: email del usuario.
- password: contraseña del usuario.

Comment:

- id: clave primaria
- comment: comentario que hace el usuario sobre una serie.
- usuariosid: clave primaria y foránea.
- seriesid: clave primaria y foránea.

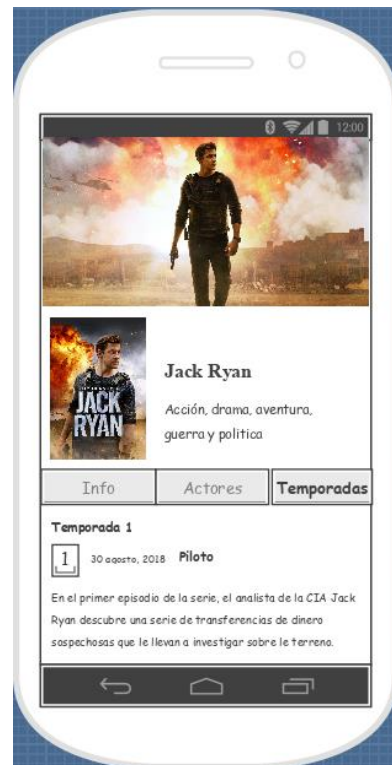
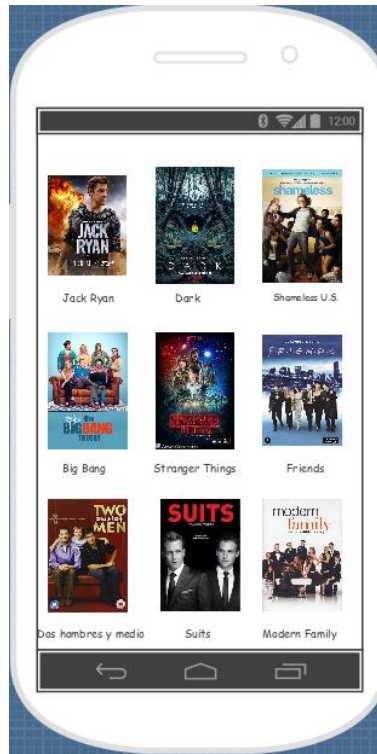
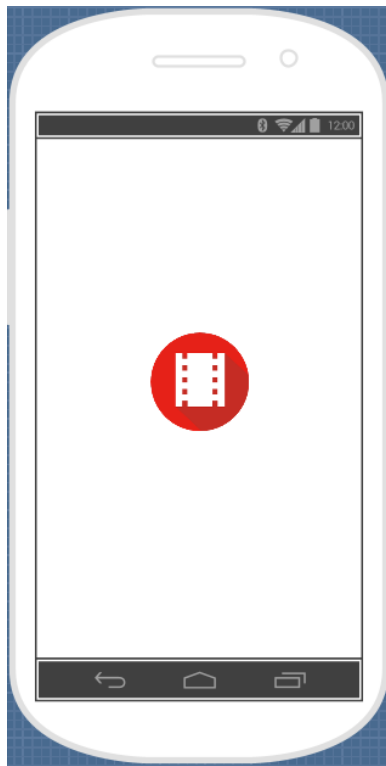
Series_Actors:

- seriesid: clave primaria y foránea.
- actoresid: clave primaria y foránea.

Relaciones:

- ✓ **Series-Episodes (1:N):** relación de uno a muchos, en la que una serie puede contener muchos episodios y un mismo episodio solo puede estar en una serie.
- ✓ **Series-Actors (N:M):** relación de muchos a muchos. Una serie tiene muchos actores y muchos actores participan en muchas series.
- ✓ **Series-Users (N:M):** relación de muchos a muchos. Dado esta relación, surge una nueva tabla llamada Comment. Una serie puede ser comentada por muchos usuarios y un usuario puede comentar muchas series.

Mockups





Registrarse

Nombre

Dirección de correo

Contraseña

¿Ya tienes una cuenta?
[Iniciar sesión](#)

The registration screen features a title 'Registrarse' at the top. Below it are three input fields: 'Nombre' with the value 'administrador', 'Dirección de correo' with 'admin@admin.com', and 'Contraseña' with masked characters '*****'. A dark button labeled 'Registrarse' is positioned below the password field. At the bottom, there is a link '¿Ya tienes una cuenta? Iniciar sesión'.



Iniciar sesión

Dirección de correo

Contraseña

¿No tiene una cuenta?
[Crear cuenta](#)

The login screen features a title 'Iniciar sesión' at the top. Below it are two input fields: 'Dirección de correo' with the value 'admin@admin.com' and 'Contraseña' with masked characters '*****'. A dark button labeled 'Iniciar sesión' is positioned below the password field. At the bottom, there is a link '¿No tiene una cuenta? Crear cuenta'.

Documentación Postman

<https://documenter.getpostman.com/view/8789935/SWEB3w4W?version=latest>

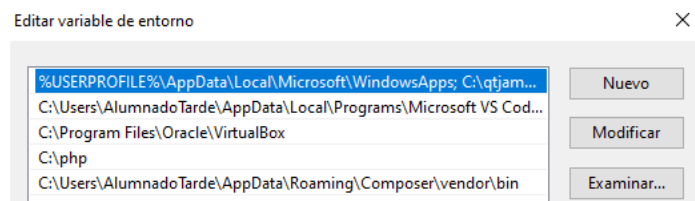
Usabilidad

- **Útil:** la aplicación cumple el cometido por el que fue diseñado, mostrar información sobre series.
- **Fácil de usar:** porque tiene interfaces sencillas con un buen diseño para no perderte en la aplicación
- **Fácil de aprender:** de la forma minimalistas que están hechas las interfaces es muy fácil encontrar la información que se busca.
- **Elegante en su diseño:** las interfaces se muestran limpias con un diseño atractivo para el usuario, sin usar muchas tonalidades y si jugando con 2 o 3 colores principales.
- **Es eficiente:** solo necesitas iniciar sesión y se tendrá a pocos clicks toda la información que se busca sobre una serie.
- **Simplicidad de diseño de la interfaz:** la aplicación solamente contiene la información necesaria, sin sobrecargar las interfaces y que sea difícil para el usuario su uso.
- **El usuario/a, debe ser capaz de personalizar la interfaz:** el usuario tiene a su disposición un modo nocturno, con el cual las interfaces cambian a un tono oscuro.
- **Acceder a todo el contenido:** el usuario puede acceder a todo el contenido en lo que se refiere a información sobre series, pero nunca podrá tener acceso de administrador (no tendría sentido que un usuario pudiese acceder a información confidencial de otro usuario).
- **Interactuar con la aplicación:** el usuario puede interactuar con las distintas interfaces.
- **Estética de diseño:** como ya he mencionado anteriormente, las interfaces son minimalistas, está bien diseñada y utilizando pocos colores para no sobrecargar la aplicación.
- **Consistencia de la interfaz:** la aplicación al completo sigue unas mismas pautas, redondeando los botones, utilizando los mismos estilos, etc.
- **Simplicidad de diseño:** el diseño es minimalista y elegante.
- **Retroalimentación:** se avisa al usuario si ha iniciado sesión correctamente, si ha modificado su perfil, si cierra sesión, etc.
- **Diseño visual:** se utilizan imágenes, iconos en los edit text de iniciar sesión para favorecer la comunicación con el usuario.
- **Color:** la aplicación se centra sobre todo en colores blanco y negro jugando también con el rojo que es el color principal del icono de la aplicación.
- **Disposición de las ventanas tipo formulario:** para que fuese más limpio y más fácil para el usuario, se han prescindido de los text view y simplemente se han colocado varios edit text con una pista (hint) dentro de éstos sobre qué es lo que se debe escribir dentro. Se han colocado uno debajo del otro para que el usuario no tenga problemas.
- **Iconos:** se han colocado varios iconos, uno de un sobre haciendo referencia al email y otro de un candado para la contraseña, haciendo más intuitiva la aplicación.

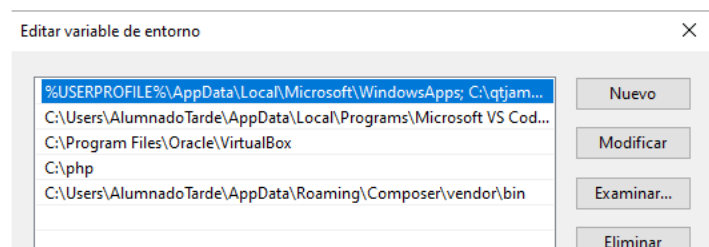
- **Seguridad:** la aplicación utiliza JWT para poder acceder a tu cuenta. Se te da un token y con ese token es con el que se te autoriza o no a acceder. Las contraseñas se encriptan para mayor seguridad.
- **Elementos multimedia:** se utilizan imágenes para mostrar información sobre los actores y sobre las series en general. De este modo la interfaz se hace mucho más agradable para el usuario ya que no es todo texto.
- **Accesible a todas las personas:** se indica en todo momento en que interfaz se encuentra, indicado con un texto en el toolbar.
- **Ser conciso con la información:** se es bastante conciso, se muestran textos ya que es una aplicación de información, pero sin abusar, se muestra de una manera minimalista, sin grandes párrafos, solo la información estrictamente necesaria y también se utilizan imágenes.

Manual de instalación

-
-
-
- Instalar Virtual box. (<https://www.virtualbox.org/wiki/Downloads>) y añadir al path la ruta.



- Instalar Git (<https://git-scm.com/downloads>)
- Instalar Vagrant (<https://www.vagrantup.com/downloads.html>)
- Instalar php (<https://www.php.net/downloads.php>) y en variables de entorno, ir al path y añadir la ruta donde hayamos guardado el programa php.



- Instalar Composer (<https://getcomposer.org/download/>). Cuando estemos en el proceso de instalación tendremos que seleccionar la carpeta de php que instalamos anteriormente. Si no seleccionamos esa carpeta no podremos continuar con la instalación.
- Instalar Android Studio (<https://developer.android.com/studio>)

Una vez hemos terminado de instalar todos los programas anteriores, ahora tenemos que abrir el git bash y añadir los siguientes comandos:

- Instalar Vagrant:

```
Juan@Juan-PC MINGW32 ~
$ vagrant box add laravel/homestead

$ vagrant box add laravel/homestead
=> box: Loading metadata for box 'laravel/homestead'
    box: URL: https://vagrantcloud.com/laravel/homestead
This box can work with multiple providers! The providers that it
can work with are listed below. Please review the list and choose
the provider you will be working with.

1) parallels
2) virtualbox
3) vmware_desktop
Enter your choice: 2
```

- Instalar Homestead:

```
Juan@Juan-PC MINGW32 ~
$ git clone https://github.com/laravel/homestead.git Homestead
```

- Inicializar Homestead:

```
Juan@Juan-PC MINGW32 ~
$ cd Homestead

Juan@Juan-PC MINGW32 ~/Homestead (master)
$ bash init.sh
Homestead initialized!
```

- Generar el código para validar el acceso a Homestead:

```
Juan@Juan-PC MINGW32 ~
$ ssh-keygen -t rsa -b 4096 -C "cursos@pildorasinformaticas.es"
```

Ahora iremos a C:/usuarios/(tu_usuario) y creamos una carpeta llamada "Code". A continuación, accedemos a la carpeta Homestead y levantamos Vagrant.

```
Juan@Juan-PC MINGW32 ~  
$ cd Homestead  
  
Juan@Juan-PC MINGW32 ~/Homestead (master)  
$ vagrant up
```

Una vez levantada solo nos quedará autenticarnos utilizando el siguiente comando:

```
Juan@Juan-PC MINGW32 ~/Homestead (master)  
$ vagrant ssh  
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-92-generic x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:   https://landscape.canonical.com  
 * Support:       https://ubuntu.com/advantage  
  
3 packages can be updated.  
0 updates are security updates.  
  
vagrant@homestead:~$ |
```

Instalamos Lumen en la máquina virtual.

```
composer global require "laravel/lumen-installer"
```

Para terminar, vamos a crear un proyecto.

```
lumen new blog
```

Código del proyecto completo

FRONTEND: <https://github.com/Alewitox/ProyectoFullStack-Android>

BACKEND: <https://github.com/Alewitox/ProyectoFullStack-Lumen>

Comparación de tecnologías

¿Qué es mejor una aplicación nativa o híbrida?

No hay una respuesta absoluta porque cada proyecto es diferente. Dependiendo del tipo de app que tengas pensado hacer, te conviene una u otra. Es tan sencillo como responder uno por uno a los puntos de esta lista:

Si tu aplicación va utilizar las funcionalidades genéricas de una aplicación sin variar sus funciones dependiendo del dispositivo, entonces te conviene una App híbrida.

Si tu aplicación va lanzarse para más de un sistema operativo móvil, mejor hacerla híbrida.

Si no te importa demasiado la pureza de la apariencia y la rapidez no es algo vital para tu app, un desarrollo híbrido puede ser la mejor opción.

Si tu aplicación necesita acceder al hardware concreto del dispositivo como el sensor de huellas o el desbloqueo facial, mejor será que consideres hacer tu app nativa.

Si quieres usar la apariencia pura y rapidez del sistema de cada móvil, mejor desarrollar la app en nativo.

En conclusión, yo diría que en 2018 los beneficios del desarrollo híbrido para aplicaciones móviles son mayores que las del desarrollo nativo. Hace unos años quizás por el tema de la velocidad de procesamiento de los móviles, podíamos valorar la opción nativa porque ofrecían mejor experiencia de usuario. Pero con dispositivos cada vez más potentes esta diferencia ya es mínima y solo en casos concretos valdrá la pena.

App Nativa

Una aplicación se dice que es nativa cuando se desarrolla en el lenguaje específico de la plataforma original.

En este sentido, si desarrollamos para Android lo haremos con C++, Java y XML o con Kotlin y sin embargo si lo hacemos para dispositivos con iOS lo haremos con Objective-C y Swift que son los entornos de desarrollo homologados para cada una de estas dos plataformas.

Ventajas de las apps nativas

- Desarrollar apps nativas tiene ciertas ventajas que el usuario puede notar en el uso de la misma.
- Una de sus principales ventajas es la velocidad de ejecución de este tipo de aplicaciones obteniendo el mejor rendimiento posible que con cualquier otro tipo de app.

- Otra de las ventajas para el usuario final es que, al estar desarrollada con el lenguaje específico de la plataforma, tiene la posibilidad de tener acceso a todas las funciones del terminal y dotar de esta forma de mayor funcionalidad a la app. De esta forma se consigue incrementar la experiencia del usuario y sacar el máximo partido al terminal.
- Para los programadores, desarrollar de forma nativa también posibilita una mayor escalabilidad de la app en términos de tamaño y que no afectará al funcionamiento de la misma.
- También debemos destacar que estas apps tienen mayor visibilidad en la App Stores de Android y Apple por lo que su posicionamiento para descargas es mayor.
- Además, estas apps se actualizarán de forma automática cada vez que se publique una nueva versión de la misma.

Desventajas de las apps nativas

Pese a tener muchas ventajas, también hay varios inconvenientes a tener en cuenta.

- El principal inconveniente puede ser el económico, ya que el desarrollo es más costoso. Pensemos que, si queremos publicar la app para Android y para iOS, deberemos programar dos veces las apps ya que cada desarrollo será diferente y necesitaremos desarrolladores de diferente perfil o bien con conocimientos en las dos plataformas.
- El otro inconveniente que vamos a encontrar es que posiblemente no podamos tener dos aplicaciones idénticas para las dos plataformas, pues cada una de ellas tiene sus funciones y por lo tanto los desarrollos no podrán ser idénticos.

App Híbrida

Una aplicación se dice que es híbrida cuando su desarrollo se ha realizado utilizando tecnología web, es decir, HTML, JavaScript y CSS.

Estas aplicaciones son responsive para adaptarse al tamaño de pantalla del dispositivo y se ejecutan sobre el navegador nativo del sistema operativo sobre el que se ejecutan.

Ventajas de las apps híbridas

- La mayor ventaja de este tipo de desarrollo de aplicaciones es que con un solo y único desarrollo, pueden ser utilizadas en las diferentes plataformas, por lo que, para diferentes sistemas como Android o iOS, no deberemos realizar desarrollos diferentes.
- Esta ventaja reduce el coste de desarrollo cuando se pretende lanzar la misma aplicación para diferentes plataformas.
- Al usar tecnologías web, su mantenimiento es también más sencillo y permiten la reutilización de código.
- Sobre su rendimiento, es inferior al de las nativas, pero no es malo, tan solo se ejecutan algo más lentas.

Desventajas de las apps híbridas

Pese a estas ventajas que hemos mencionado, también hay desventajas importantes.

- Como hemos mencionado anteriormente, el rendimiento de este tipo de aplicaciones es menor que el de una aplicación nativa.
- Otro inconveniente es que tienen un acceso limitado a las funciones del dispositivo móvil y no pueden sacarle el máximo rendimiento al mismo.
- Al correr sobre el navegador web, la mayor parte de este tipo de aplicaciones, requiere una conexión a Internet para poder ejecutarse.
- A nivel estético, tampoco son tan atractivas como las nativas debido a las limitaciones de la tecnología de desarrollo utilizada.

Ventajas e inconvenientes de la web app

Si de ventajas de la web app estamos hablando, debemos decir que esta implica un proceso de desarrollo más simple y menos costoso, lo que la convierte en la principal opción, sobre todo, para empresas que dan sus primeros pasos en este segmento.

El hecho de utilizar lenguajes multiplataforma en su desarrollo, hace posible que el mismo código sea reutilizable en distintos sistemas operativos, es decir, con independencia del equipo que se utilice.

Debido a que no es necesario descargarlas de ninguna tienda de aplicaciones, su acceso se da a través del navegador, consumiendo menores recursos de los dispositivos y haciéndolas totalmente accesibles desde cualquier terminal.

No requiere actualizaciones constantes, así es que su coste de mantenimiento es mínimo.

Por supuesto que no todo es positivo con respecto a estas apps, dado que para usarlas siempre se requiere de una conexión a Internet y su velocidad está supeditada a la velocidad del navegador y la calidad de esa conexión de internet.

El acceso a los elementos y características del hardware del dispositivo es limitado, lo que de alguna manera también limita en cierta medida la experiencia del usuario.