

*Alumno: Fran Adrover Vallejo*

*Centro: IES Francesc de Borja Moll*

*Modulo: DIW*

*Docente: Alberto Soto*

# IonicAp



## Índice de contenido

Presentación y propósito.....	3
Requerimientos.....	4
Caso de Uso.....	4
Enfoque técnico y la descripción.....	5
Estructura del proyecto.....	7
Flujo y desarrollo de la aplicación.....	9

## Presentación y propósito

En nuestro día a día el mundo de los gadgets y su contenido están cada vez mas en auge, redes sociales, apps informativas referentes a cualquier temática, contenidos multimedia o servicios en la nube son solo unos ejemplos del abanico que nos ofrece este mercado.

En el presente proyecto hemos decido desarrollar una aplicación que nos permita inmortalizar cualquier momento en cualquier lugar, pudiendo posteriormente compartir nuestra experiencia con el resto del mundo.

Se trata de una aplicación móvil que nos permita realizar fotografías para poder posteriormente subirlas a nuestro servidor, además nos ofrecerá la opción de elegir entre varias categorías y visualizar todo sus items subidos por el resto de usuarios.

## Requerimientos

Para el desarrollo del proyecto es necesario disponer de un ordenador cuyo SO es independiente, (a no ser que la aplicación en cuestión vaya dirigida a sistemas IOS, donde es necesario disponer de un computador MAC), un IDE para nuestro desarrollo en cuestión y un server con servicio web service, Rest en nuestro caso.

## Caso de Uso

El usuario deberá ejecutar la aplicación, logearse si previamente no lo ha hecho, a continuación la aplicación nos llevara al home y el siguiente paso sería pulsar el botón que activará la cámara del dispositivo y posteriormente realizar la fotografía. Una vez hecha la fotografía la aplicación nos ofrece la opción de etiquetarla, añadiendo, su respectivo nombre, una etiqueta y finalmente la categoría a la que pertenece. Si el usuario lo desea tiene la opción de menú en las que aparecen todas las categorías, acción que le permitirá filtrar por la categoría deseada y visualizar todo su contenido.

## Enfoque técnico y la descripción

**Ionic** es un framework desarrollado bajo tecnologías Javascript que permite desarrollar aplicaciones celulares híbridas. A grandes rasgos, cuando hablamos de aplicaciones celulares híbridas nos estamos refiriendo a que son aplicaciones que no son del tipo nativas, es decir, que no son desarrolladas con la tecnología nativa con la que ejecuta en el sistema operativo del móvil.

Particularmente este framework posee grandes características que lo diferencian de muchos de otros frameworks similares. Entre estas ventajas enumeramos las siguientes:

- Posee un numero importante de componentes propios de aplicaciones móviles nativas desarrolladas en **HTML5**.

Esto nos permite no solo obtener una interfaz que emula casi a la perfección una aplicación nativa sino que también, nos permite ver la aplicación desde un navegador web y los mismos componentes no perderán calidad ni se deformarán sin importar que resolución tengamos. Punto importante en la portabilidad.

- Implementa un patrón MVC para dividir las capas de la aplicación haciendo uso de las buenas prácticas.

El patrón de diseño MVC utilizado en el framework es aportado por otro framework muy importante, que es **AngularJS** desarrollado por la gente de Google, aplicando la buena práctica de modularización de las vistas sin necesidad de embeber el template en el mismo index que es como lo van a ver en muchos de los ejemplos existentes en internet y que no son del todo prolijos ya que no utilizan toda la fuerza que brinda **AngularJS**.

- Utiliza **Apache Cordova** como wrapper o capa de abstracción del OS del celular nativo en el cual se quiere desarrollar.

**Apache Cordova** es una plataforma que se utiliza para la construcción de aplicaciones nativas de celulares o dispositivos móviles en general utilizando HTML, CSS y Javascript.

- Permite customizar el front-end de forma fácil, rápida y muy sencilla al permitir el diseño de la aplicación con CSS y HTML.
- Trabaja en conjunto con la tecnología conocida como Bower que funciona como repositorio de modulos javascript.

Con este framework de dependencias javascript nos permite instalar módulos o plugins más fácilmente para ser utilizados por **AngularJS**

#### Instalación

Para la instalación de este framework basta solamente con tener instalado **NodeJS** y el **NPM** (*Node Packaged Modules*).

modulo de **npm** para de esta forma instalar finalmente el Framework **ionic**

***sudo apt-get install nodejs nodejs-legacy npm***

Luego de ejecutada la misma, ya tenemos instalado tanto NodeJS como su manejador de paquetes necesario para la instalación del framework ionic.

Para instalar ionic solo basta con ejecutar el siguiente comando en consola:

***sudo npm install -g cordova ionic***

Con el comando anterior, se instala no solo ionic, sino también, la plataforma de Apache Cordova necesaria para instalar y hacer uso de las funcionalidades nativas del OS del dispositivo móvil donde este o estará corriendo la aplicación.

## Estructura del proyecto

### Carpetas propias de Cordova:

**hooks:** almacena los scripts para realizar tareas personalizadas como la carga automática de plugins.

**Plugins:** almacena el código fuente de los plugins que vayamos a utilizar.

**www:** almacena el código fuente que va a ser ejecutado dentro de cada proyecto híbrido. En el caso de este generador, es aquí donde va a almacenar la salida de las tareas que ejecute.

**Resources:** carpeta destinada a parámetros de Splash Screen

**config.xml:** es el fichero que almacena información de relevancia para poder configurar los proyectos dependiendo de la plataforma.

### Carpetas propias de la aplicación web:

**css:** almacena todos los archivos correspondientes a los estilos

**img:** almacena las imágenes del proyecto

**js:** almacena los archivos javascript

**app.js:** archivo donde se inicializan los módulos de las diferentes plataformas y contiene todos los estados de los respectivos templates

**controllers:** los controladores contienen todas las funciones que son ejecutadas en la aplicación. Destaca la presencia del scope que es el parámetro que en comunicación con los templates es el que nos brinda el flujo de la aplicación.

**Categories:** es el archivo donde se llevan a cabo la conversión de comandos propios del protocolo HTTP al lenguaje JavaScript.

**Config.js:** este archivo nos permite la conexión a la base alojada en el servidor

**ng-cordova:** archivo que hace que sea posible la interacción entre la aplicación y el dispositivo

**templates:** directorio donde tenemos almacenados todos los archivos html.

**Index.html**

**menu.html**

**home.html**

**playlists.html**

**playlist.html**

**images.html**

**login.html**

**homelogin.html**



## Flujo y desarrollo de la aplicación

Al ejecutar la aplicación el usuario se encuentra frente a un botón que le ofrecerá el acceso a la misma, dicho botón al ser pulsado ejecuta una función llamada `login()`, el acceso a dicha función lo obtendremos mediante el controlador “AppCtrl” que es iniciado al cargar la página inicial establecida en el parámetro “\$urlRouterProvider”, la pagina que contiene el botón, “homelogin.html”. La acción a realizar por esta función (`login()`), es la de mostrarnos el modal, el cual nos permitirá ingresar nuestras credenciales de logeo e ingresar al home de la aplicación activando la función destinada a ello al pulsar el boton de submit del modal, la funcion que nos hace dicho ingreso posible (`closeLogin()`), es caragada por el mismo controlador asociado al archivo “homelogin.html”.

Una vez en el home, dispondremos de un menú que contiene dos opciones, una opción para volver al home y otra para poder visualizar y seleccionar las posibles categorias disponibles. Cabe resaltar que el template del menú lo cargamos inicialmente como aplicación, es decir, es un template el cual esta presente en todo el flujo de la aplicación. Si pulsamos la opción “Home” simplemente accedemos a nuestro home a través de la propiedad “href”, sin mas. Sin embargo al seleccionar la opción “Categorias”, se nos redirecciona al archivo “playlists.html” el cual inicializa el controlador “CategoryListCtrl”, cuya función es la de cargarnos en el scope todas las categorías contenidas en nuestra base de datos actualizando de esta manera la lista desplegable de categorías. Al seleccionar una categoría cualquiera lo que estamos haciendo a parte de redireccionarnos al archivo responsable de mostrarnos los items de dicha categoria “playlist.html”, es cargar el id de la categoría seleccionada, gracias la parámetro establecido en “url” del state de dicho archivo. Este state nos carga un controlador llamado “ItemDetailCtrl”, cuya función es la de recoger todos los ids de los items alojados en el servidor y mediante el scope comparar con el id referente a la categoria seleccionada previamente, el resultado final será una lista de items referentes a la categoría deseada. Situados en la lista de itmes, es decir el archivo “playlist.html”, el siguiente paso es hacer click sobre algún item para poder acceder el avatar entero junto a datos descriptivos al respecto, cuya acción nos llevará al archivo “images.html”, que es el encargado de mostrar dicho contenido. El state de este archivo contiene la carga de un id

y la llamada a un controlador llamado "ImageDetailCtrl", la función de dicho controlador es la misma que el anterior, obtener los id de las imágenes almacenadas en el servidor, acción que es posible gracias a la configuración establecida en los archivos "categories.js" y "config.js" y a la manera en la que interactúan entre ellos, y comparar dicho id con el que nos cargará en el controlador el parámetro "\$stateParams", que es el obtenido al realizar la selección previa.

Otras funcionalidades que tendrá el usuario disponibles son unos iconos situados en el footer de la página. Se trata de un icono para regresar al Home, otro de configuración y un tercero para salir a la página de login. Esto es posible debido a un controlador llamado "homeCtrl" que está situado en el state de carga de inicio inicializándose de esta manera al ser cargada la raíz de la aplicación, dicho state nos cargará el menu y el footer para así permanecer en toda la aplicación. El contenido de dicho controlador simplemente son dos funciones que mediante el parámetro "state.go" mas la ruta del fichero en la aplicación nos redirigirá al lugar establecido.