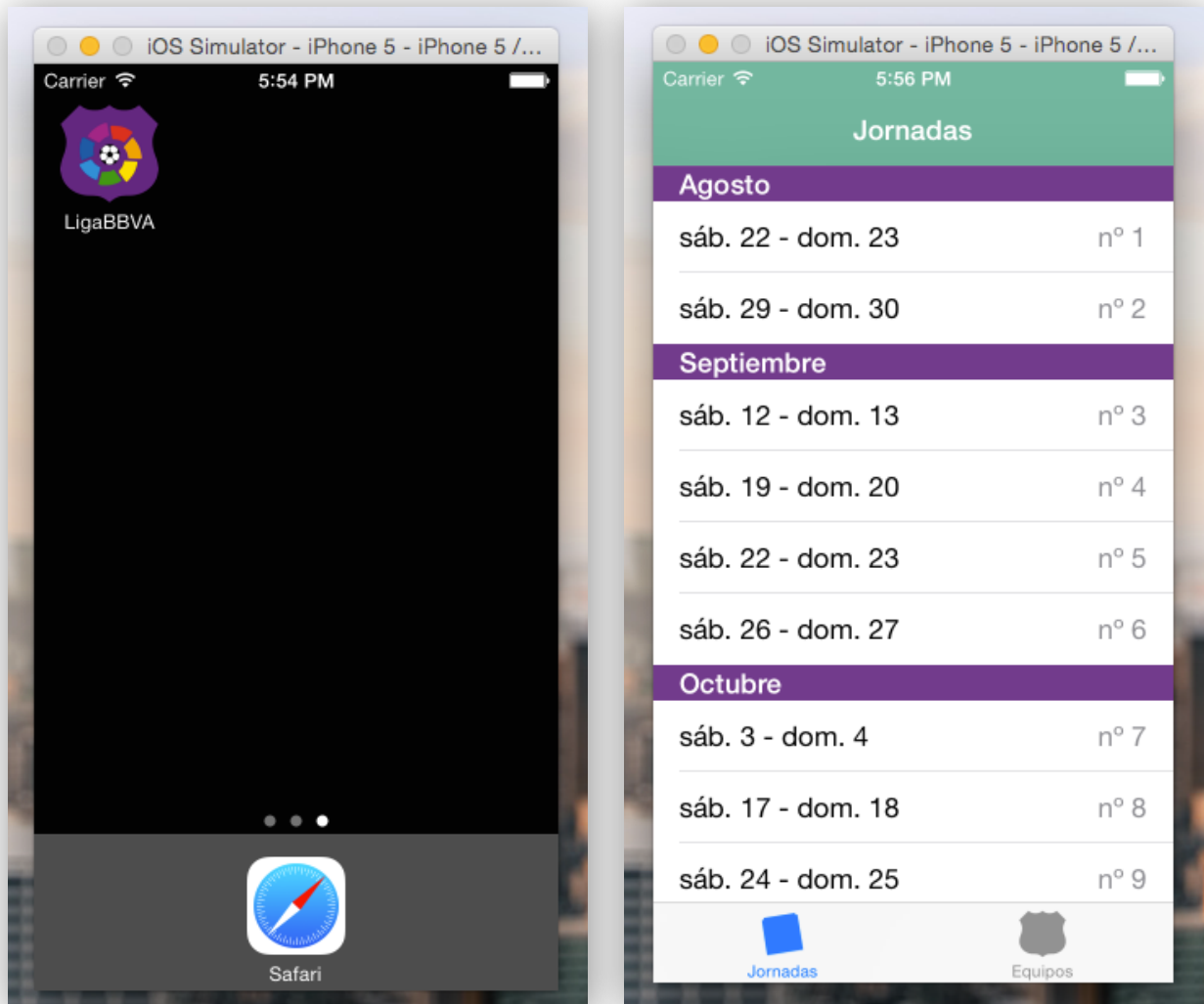


LIGA BBVA



Manual de usuario.

Lo primero es localizar nuestra aplicación "LigaBBVA" en el "escritorio" y lanzarla.



La primera vista que se nos muestra es la lista de Jornadas, organizadas por meses.

Cada jornada es referida por el fin de semana en el que ocurre, y por su número dentro de la liga.

En la barra de abajo podemos observar que también podemos lanzar una vista de los equipos de la liga, pero de momento nos centramos en el flujo de la aplicación a partir de nuestro estado actual.

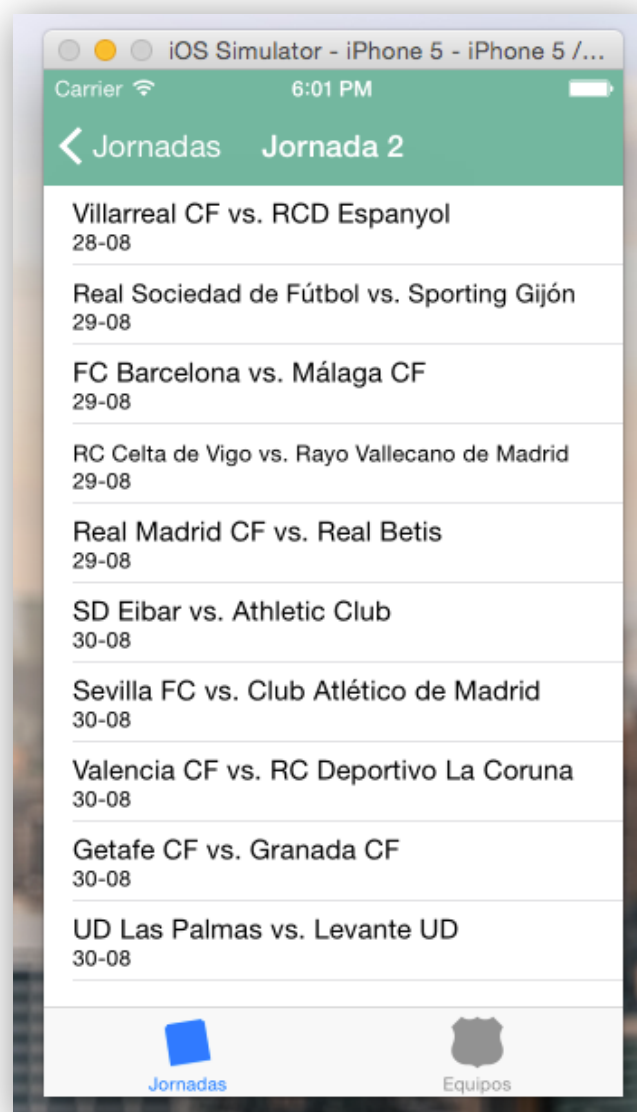
Si hacemos "click" en una jornada, pasaremos a ver la lista de partidos que abarca, como es de esperar.

Esta vista, tendrá como título el número de jornada de liga en la que nos encontremos.

En cada elemento de la lista veremos como título a los equipos que se enfrentan, y, en el detalle podremos ver el día en que el partido fue jugado / se jugará.

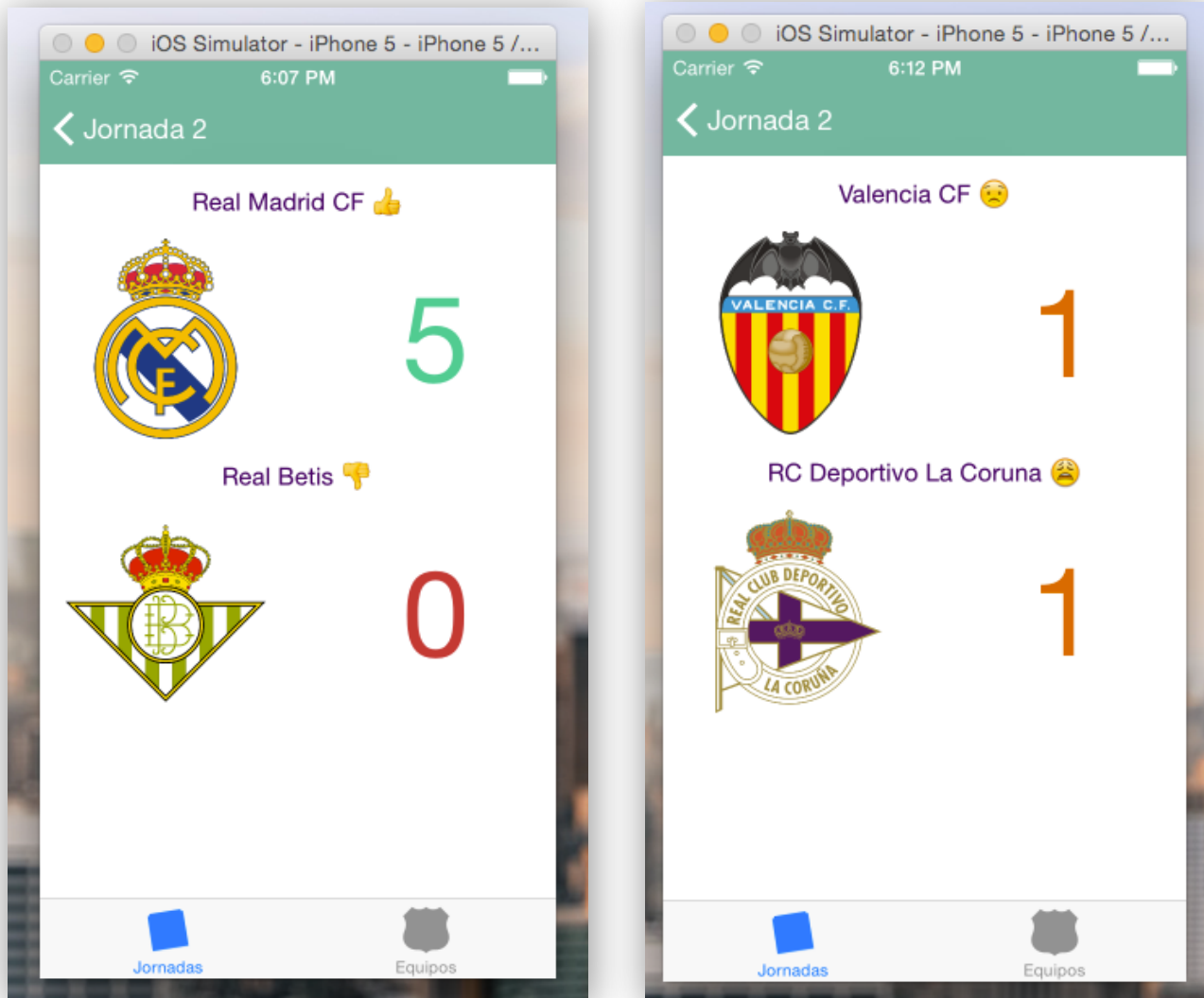
Si "clicamos" en la palabra Jornadas, o deslíamos desde el borde izquierdo de la pantalla hacia el centro, volveremos a la lista de jornadas.

Si lo que tocamos es un partido, entraremos a una vista detallada del mismo.



La vista de partido muestra los dos equipos participantes en el partido, junto con un marcador de resultado, que será tematizado dependiendo del resultado del partido.

Verde indica el equipo ganador, rojo el perdedor, y naranja si se ha producido un empate.



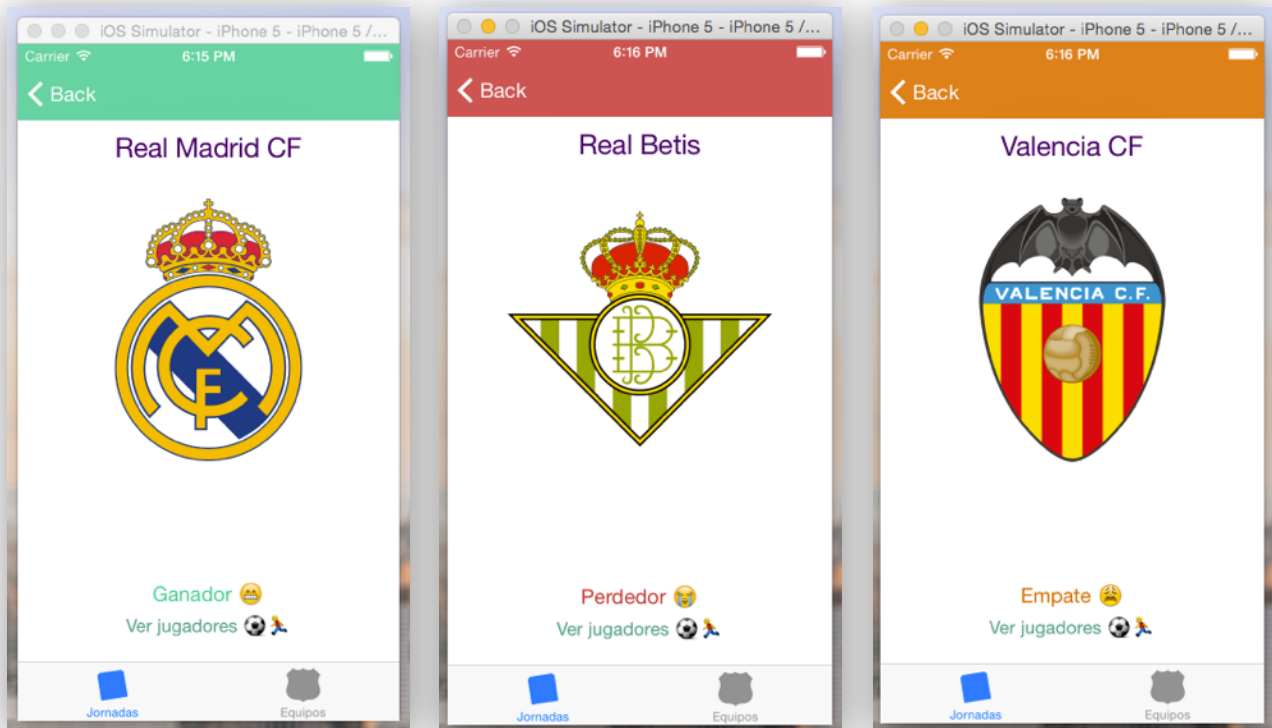
Podemos volver a la jornada, o tocar el escudo o nombre de cualquiera de los dos equipos participantes para ver una vista detallada del equipo pulsado.

La vista que se abre también estará tematizada dependiendo del resultado del equipo en el partido.

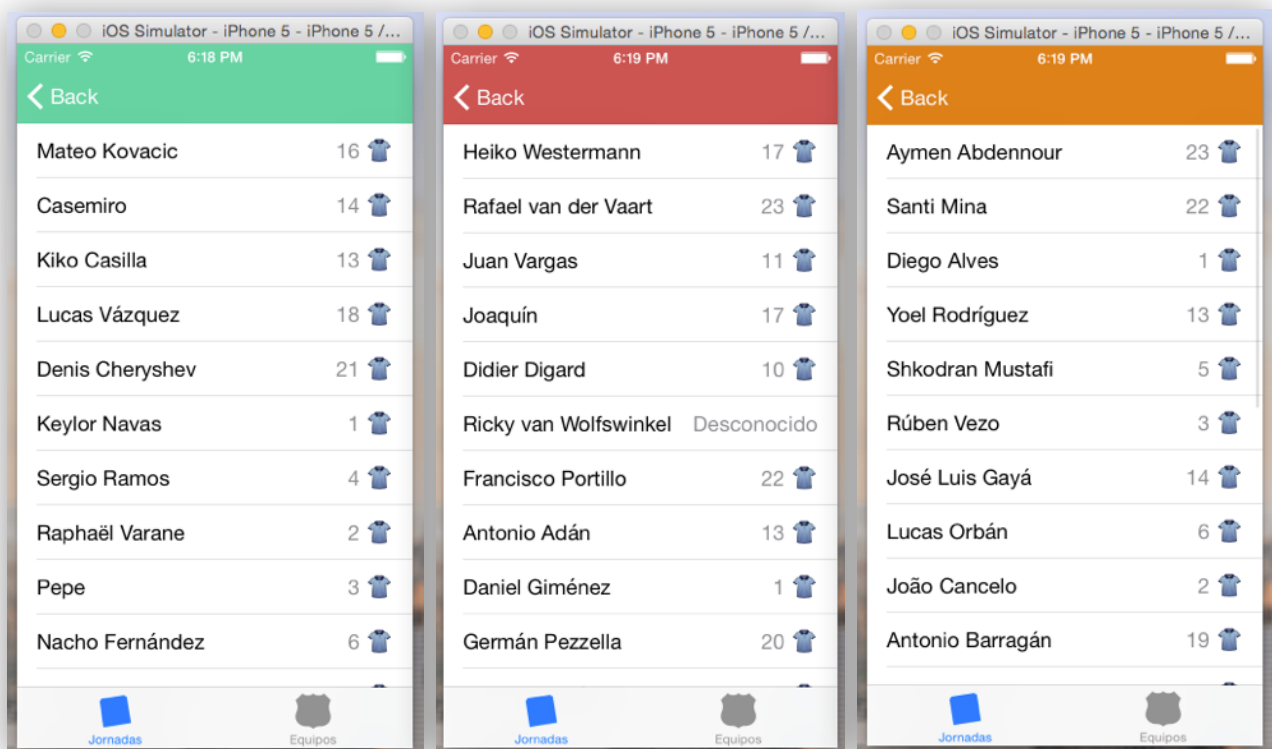
La tematización alcanzará hasta la barra superior, cambiando el tono de verde si hemos entrado a ver el equipo ganador, roja si se trata del perdedor, o naranja si hubo empate.

También se mostrará un mensaje recordando el resultado del equipo.

Desde esta vista podremos volver al partido, o ver los jugadores del equipo.



Si decidimos entrar a ver los jugadores, veremos una lista en la que se indica el nombre de cada uno, junto con su dorsal. La tematización continúa.

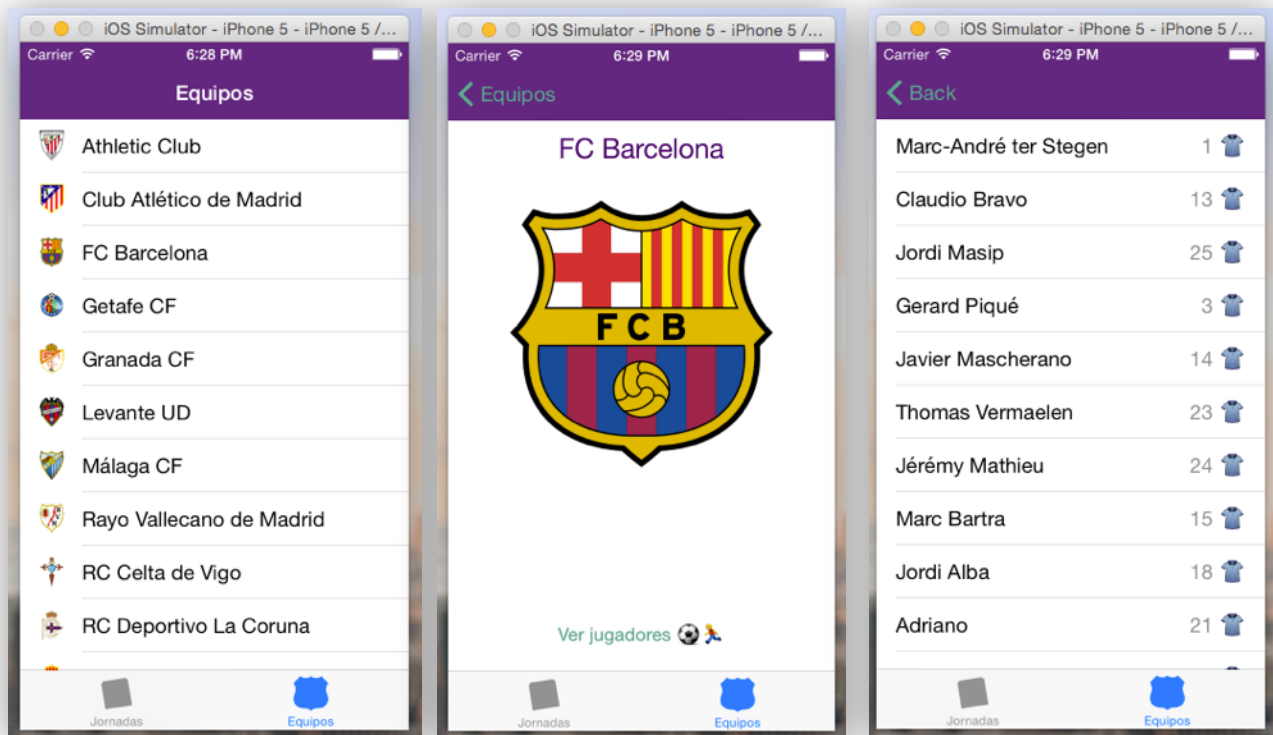


La otra funcionalidad que ofrece la aplicación es la de consulta de equipos de la liga.

Si volvemos a la pantalla inicial (o desde cualquier subvista en la funcionalidad de “Jornadas”), podemos tocar el icono situado a la derecha en la barra para acceder a la función de “Equipos”.

El cambio entre ambas funcionalidades es inmediato, y se mantiene el progreso para que al volver podamos continuar con lo que estábamos haciendo.

Esto es interesante para poder por ejemplo hacer una consulta rápida a un equipo sin tener que navegar todas las vistas si estamos en la función Jornada.



Como es lógico aquí la tematización desaparece, puesto que estamos simplemente consultando los equipos sin partir de ningún resultado.

Manual de ingeniero.

Distinguiremos tres partes de esta aplicación.

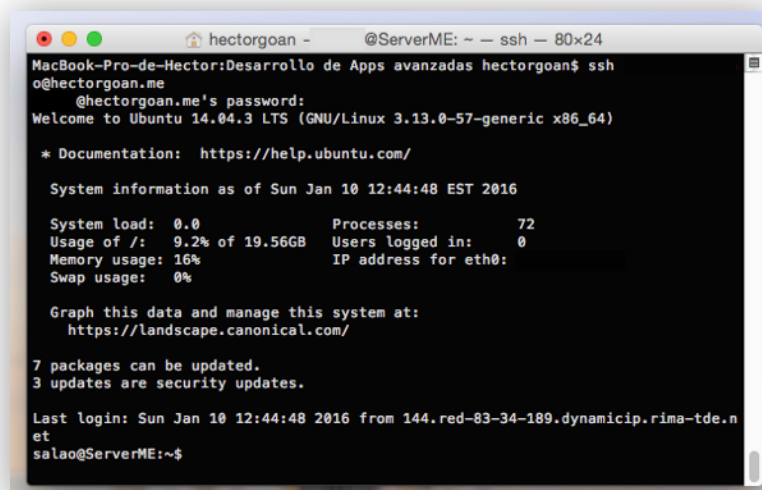
Modelo (En servidor - JSON)

Toda la parte de back-end en el servidor se ha hecho personalizado para esta aplicación.

En un principio pensé en utilizar la api de fútbol que localizamos en clase, pero me resultaba algo molesta por lo siguiente:

- Demasiada información: Para la funcionalidad de esta aplicación no necesitaba tantísima información como la mostrada por la api.
- JSON de estructura incómoda: La estructura JSON de los archivos devueltos por la api me resultaban demasiado enrevesados a la hora de leer, cuando yo lo único que necesitaba eran arrays de objetos.
- Uso de datos y memoria en la app: Al descargar mas información de la realmente necesaria se está haciendo un uso indebido del internet del cliente de la app. Con mis JSON se reduce el consumo de datos y por otra parte, al guardar menos datos, estamos usando menos memoria en el teléfono. Debemos recordar que estamos tratando con dispositivos móviles y, aunque cada vez son mas potentes, no dejan de ser más limitados que un pc.
- Me gusta mas tratar los datos en casa: Bien es cierto que esta aplicación no utiliza datos personales del cliente, pero opino que es buena práctica tener la información de nuestra mano en vez de delegarla a terceros.

Se ha utilizado un servidor NGINX corriendo en un Ubuntu Server 14.04.3 para el servicio de archivos JSON.



```
MacBook-Pro-de-Hector:Desarrollo de Apps avanzadas hectorgoan$ ssh  
o@hectorgoan.me  
@hectorgoan.me's password:  
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-57-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
System information as of Sun Jan 10 12:44:48 EST 2016  
  
System load:  0.0          Processes:    72  
Usage of /:   9.2% of 19.56GB  Users logged in:  0  
Memory usage: 16%          IP address for eth0:  
Swap usage:   0%  
  
Graph this data and manage this system at:  
https://landscape.canonical.com/  
  
7 packages can be updated.  
3 updates are security updates.  
  
Last login: Sun Jan 10 12:44:48 2016 from 144.red-83-34-189.dynamicip.rima-tde.net  
salao@ServerME:~$
```

La estructura de servicio de los archivos es la siguiente:

hectorgoan.me/FOOTBALL/...

misEquipos.json : Contiene un array de objetos JSON con los datos de los equipos de la liga.

misJornadas.json : Contiene un array de objetos JSON con los datos de las jornadas de la liga.

Equipos/ : Contiene una carpeta para cada equipo de la liga, y dentro de el un JSON con sus jugadores, llamado players.json

Jornadas/ : Contiene un JSON para cada jornada, con el nombre: **numJornada**Jornada.json así, para la jornada 1 su archivo será 1jornada.json

Logos:/ : Contiene los escudos de los equipos, no obstante, estos se empaquetan con la aplicación como recursos gráficos, psique no es necesario tenerlos aquí.



Vistas (En app iOS - StoryBoard)

Storyboard de la aplicación:

Se parte de un Tab Bar Controller, que nos permite tener las dos funcionalidades de la app en el "dock"

El primer elemento del Tab Bar nos lleva a la funcionalidad de Jornadas, toda ella englobada en un Navigation Controller.

El otro elemento nos lleva a Equipos, que también está en un Navigation Controller.

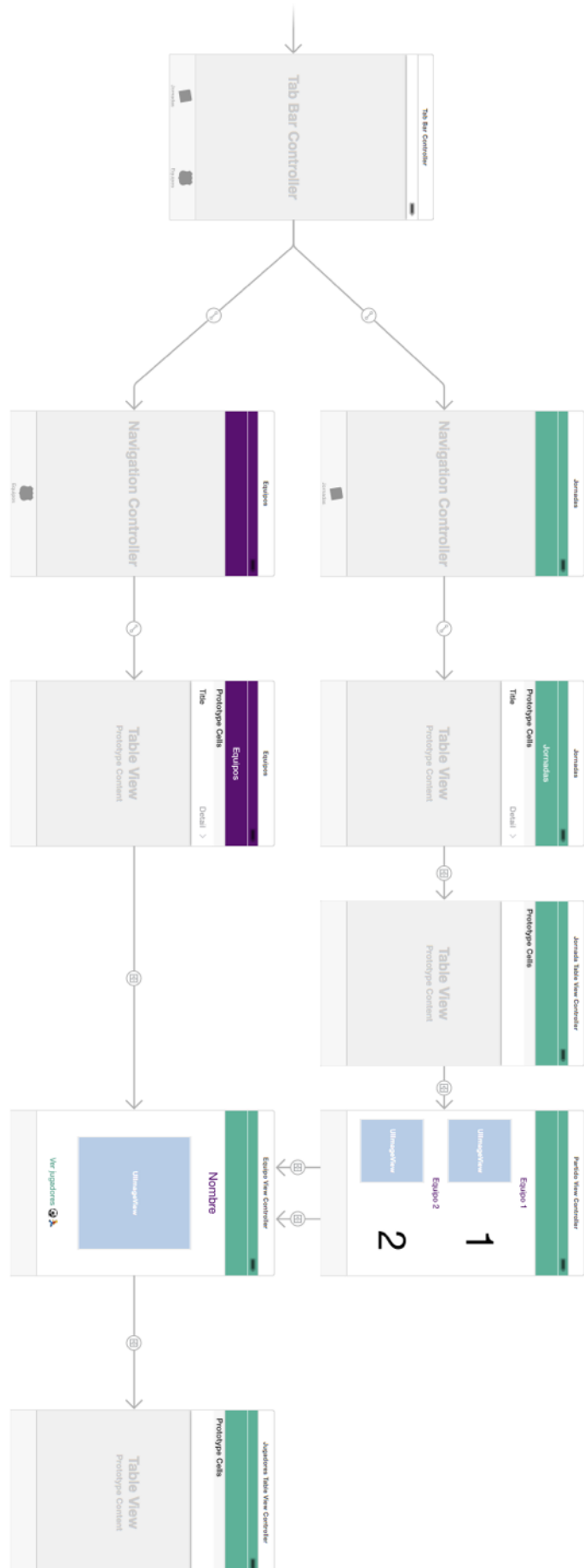
Jornadas:

Empezamos por un TableView que contiene las jornadas, controlado por HGAJornadasTableViewController

Al elegir un elemento del TableView Jornadas, pasamos a la vista detallada de Jornada, en la que se ven los partidos de la misma. Table View gobernada por HGAJornadaTableViewController

Si elegimos un partido pasamos a la vista del partido, controlada por HGAPartidoViewController que es un UIView compuesto por dos botones para los nombres de los equipos, dos UIImageView para los escudos, y dos UILabel para el marcador.

Al hacer click en alguno de los botones, o de los escudos pasamos a ver el UIView del equipo, controlado por HGAEquipoViewController.



La vista del equipo se compone por un UILabel para el nombre, un UIImageView para el escudo, otro label para mostrar el mensaje de resultado (inicializado a vacío por si venimos de la lista de equipos), y un UIButton para poder ver los jugadores.

Si pulsamos el botón de ver jugadores se nos lleva a un Table View gobernado por HGAJugadoresTableViewController.

Equipos:

Es un Table View controlado por HGATeamsTableViewController.

Al pulsar una celda, estamos otra vez en la vista de equipo.

Controladores (En app iOS - Objective-C)

Esta es la estructura del proyecto.

ligaBBVA/

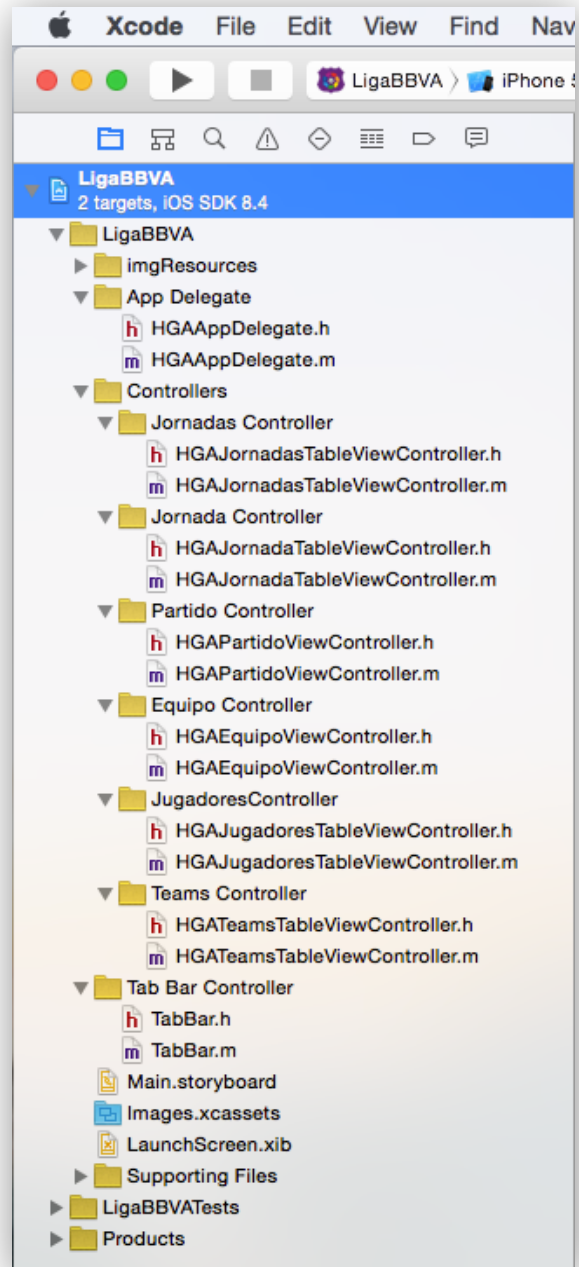
imgResources/ : Contiene los escudos de los equipos en formato .png

AppDelegate/ : Contiene la clase *HGAAAppDelegate*, subclase de *UIResponder* y siguiendo el protocolo *UIApplicationDelegate*. Creada automáticamente por Xcode. Nada que destacar.

Tab Bar Controller/ : Contiene la clase *TabBar* que es la controladora de la barra "Tab Bar". Esta clase solo se utiliza para que cada vez que se pulse un elemento la barra de navegación de la vista de Jornada vuelva a su color original. Si no estuviera esta clase, al pulsar un elemento del Tab Bar (el de Jornadas en especial) nos vamos a la vista de Jornadas, y la barra de navegación continuaría tematizada.

Controllers/

Jornadas Controller/ : Contiene la clase *HGAJornadasTableViewController*, subclase de *UITableViewController* y siguiendo los protocolos *UITableViewDelegate* y *UITableViewDataSource*. Es la clase controladora de la vista de la lista de Jornadas. El funcionamiento explicado brevemente es: Coger los datos del JSON, preparar categorías para separar la tabla por secciones en meses, y mostrar la tabla ordenada, permitiendo que al pulsar una celda se haga segue a la vista de Jornada, indicando el número de jornada a *HGAJornadaTableViewController*.



Jornada Controller/ : Contiene la clase HGAJornadaTableViewController, que hace básicamente lo mismo que HGAJornadasTableViewController, solo que no tenemos que hacer la separación en secciones de los elementos de la lista, ósea que leemos del JSON que sabemos cual va a ser dependiendo del número de jornada que nos ha indicado HGAJornadasTableViewController, mostramos la lista, y preparamos para hacer segue a la vista de partido, pasando el título del partido, el equipo local, el visitante, y el número de goles de ambos.

Partido Controller/ : Contiene la clase HGAPartidoViewController, subclase de UIViewController. En el método viewDidLoad, en este caso no tenemos que recoger datos de ningún JSON, sino que simplemente tenemos que centrarnos en las tareas de tematización dependiendo del resultado. Luego prepararemos el segue, pasando a la HGAPartidoViewController el nombre del equipo que ha de mostrar (el pulsado), así como el resultado de este en el partido, para así poder seguir con la tematización en la vista de equipo.

EquipoController/ : Contiene la clase HGAEquipoViewController, que al igual que HGAPartidoViewController descende de UIViewController, y controla un UIView personalizado para la visualización del equipo. Este controlador también se ha de centrar simplemente en tematización, y en preparación de segue para HGAJugadoresTableViewController, pasándole el nombre del equipo.

JugadoresController/ : Contiene la clase HGAJugadoresTableViewController, y, en su método viewDidLoad se encarga de prepararse para mostrar la lista de jugadores que se localizan en el JSON que "calcula" a partir del nombre del equipo.

TeamsController/ : Contiene la clase HGATeamsTableViewController, que muestra una lista de equipos de la liga, cogiendo estos de un JSON, y poniendo a cada uno su escudo (descifrado a partir del nombre del equipo).

Reflexiones:

El desarrollo de esta práctica me ha parecido muy interesante, puesto que hemos podido seguir experimentando con el lenguaje Objective-C, aprendido para la realización de la práctica de la asignatura Interfaces Gráficas, en la que se realizaba un programa con interfaz para OSX, utilizando el framework Cocoa.

Cocoa Touch (iOS) varía respecto a Cocoa (OSX), y el mayor reto que ha supuesto esta práctica ha sido aprender los “trucos” que conlleva el uso de storyboards, ya que para Interfaces Gráficas se utilizó exclusivamente los .xib.

La programación para iOS con StoryBoards es bastante similar a la de WPF debido al uso de la arquitectura MVVM (Model View View-Model) respecto a MVC (Model View Controller) visto en OSX, y esto me parece un avance importante.

Anteriormente a esta asignatura, respecto a dispositivos móviles yo había tocado Android, que tiene de similitud a WPF el tema de Java + .xml (C# is JAVA!), y por tanto, la curva de aprendizaje ha sido mínima gracias a la base de WPF y Android en cuanto a arquitectura, y la de Objective-C gracias a Cocoa.

Estoy aprendiendo Swift. Swift mola mas.