



BETGAME PROJECT

Desarrollo de un videojuego on-line de pronósticos deportivos

Autor:

Alejandro López González

Consultores:

Antonio Rodríguez Gutiérrez
Carlos Sánchez Rosa

Junio 2017



¿Qué es BetGame?

BetGame es un videojuego para dispositivos móviles que permite a los usuarios crear ligas virtuales con sus amigos para pronosticar resultados y competir entre ellos para demostrar quién sabe más sobre apuestas deportivas.

La finalidad principal de desarrollar esta aplicación es conseguir un videojuego atractivo y ser una opción más dentro de este abanico de aplicaciones.





Objetivos



A nivel general, los objetivos del proyecto son:

- Diseñar y programar una aplicación para dispositivos móviles con Android.
- Planificar y documentar de forma profesional todo el proceso.

A nivel personal, este proyecto me va a servir para:

- Poner en práctica todo lo que he aprendido durante la carrera.
- Introducirme en el desarrollo de aplicaciones móviles.

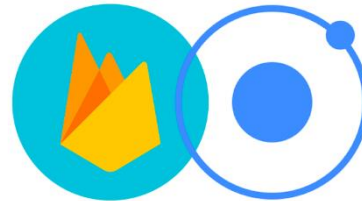


Enfoque y metodología

Al tratarse de un videojuego de “gestión de pronósticos” no requiere de gráficos espectaculares, por ese motivo se ha decidido implementar en forma de aplicación móvil híbrida con una interfaz minimalista y lo más sencilla posible para el usuario.

Después de analizar las diferentes alternativas nos hemos decantado por trabajar con:

- Ionic Framework 3 que nos permite crear aplicaciones híbridas multiplataforma.
- Firebase es un servicio de Google capaz de proveernos un backend en la nube con una base de datos en tiempo real, autenticación, etc.





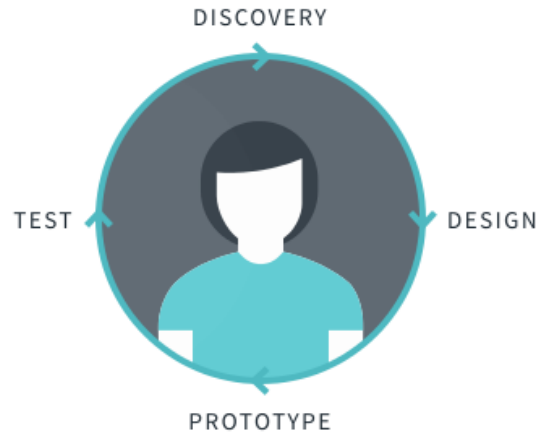
Enfoque y metodología



Además, para poder obtener de forma automática los resultados de los eventos deportivos se ha implementado un pequeño script en Node JS que consulta un Webservice gratuito que nos provee de esta información.



Análisis



Conocer a las personas a las que se dirige nuestra aplicación, para qué la van a usar y en que condiciones nos permite identificar las funcionalidades que tendrá que satisfacer nuestro producto.

Métodos para conocer los perfiles de usuario:

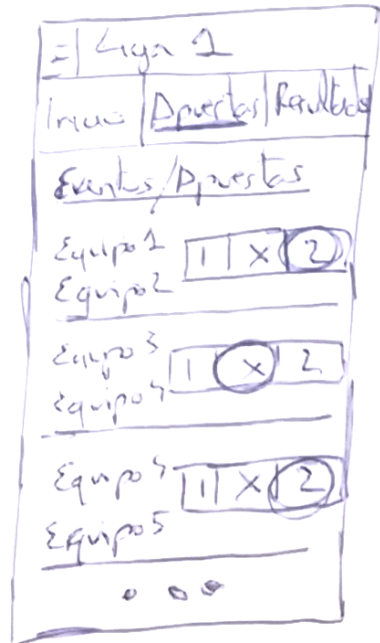
- El método de investigación contextual. Se han realizado varias pruebas para observar la motivación e interacción del usuario con una aplicación similar.
- El análisis competitivo me ha permitido evaluar productos que satisfacen las mismas necesidades para conocer sus ventajas y desventajas.

La realización de este análisis nos ha permitido obtener los principales perfiles de usuarios que utilizarán nuestra aplicación y sus necesidades. Además de algunos contextos de uso habituales.



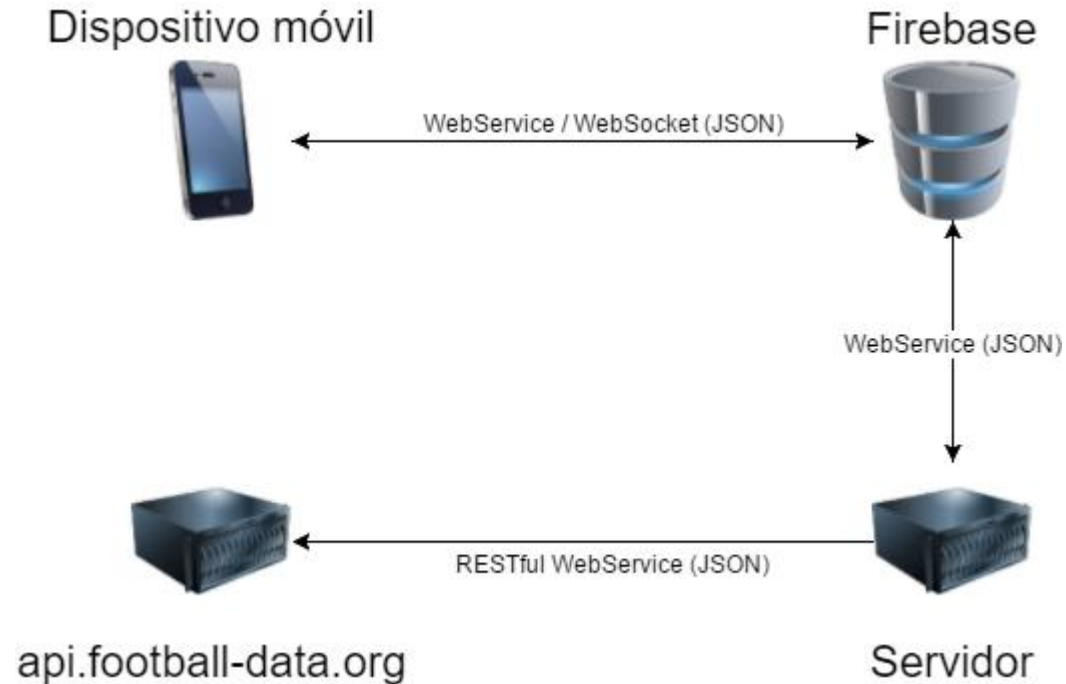
Diseño

- Análisis de tareas: funcionalidades básicas que tiene que tener nuestra aplicación para conseguir los objetivos deseados.
- Flujos de interacción: se diseñan los flujos de interacción generales de la aplicación.
- Prototipado: se realiza un diseño a mano alzada y de alta fidelidad.





Arquitectura



Dispositivo móvil

- Aplicación Android desarrollada con **Ionic Framework 3** (Angular 4 y Apache Cordova)

Firestore

- Base de datos noSQL en tiempo real
- Autenticación de usuarios de manera segura
- Cloud Functions para extender las funcionalidades back-end

Servidor

- Script en **Node JS** que consulta el **WebService** de <http://api.football-data.org/> para obtener los eventos y resultados deportivos



Entorno de desarrollo

Entorno de desarrollo:

- PC de sobremesa Intel Core I7-4790 a 3.60GHz + 16GB RAM.
- Sistema operativo Windows 10 Pro.
- Motorola Moto G4 para realizar las pruebas finales.

Para la implementación de la aplicación hemos trabajado con el editor de texto Sublime Text 3 y con el navegador Google Chrome para la visualización y depuración del código.



Herramientas auxiliares:

- NPM (Node Package Manager)
- Ionic Cli
- StarUML
- GanttProject
- Adobe Photoshop CS 6
- Microsoft Word 2016 y PowerPoint 2016



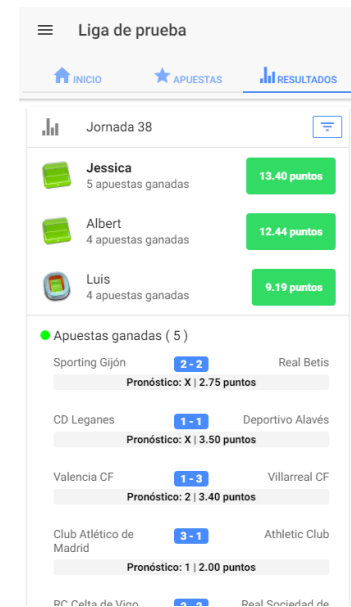
Desarrollo (Front-End)



La aplicación se ha programado en lenguaje TypeScript, HTML y SCSS y para la parte gráfica de la aplicación se han utilizado los componentes que ofrece Ionic con el tema de Material Design.

La estructura de la aplicación se ha dividido en:

- Pages: componentes que contienen la lógica de cada “vista” de nuestra app.
- Templates: ficheros HTML con el aspecto visual de cada componente Page.
- Services/Providers: clases que nos permiten compartir funcionalidades con los componentes y en donde tenemos encapsulada la comunicación con el Back-End.
- Model: clase en donde tenemos las interfaces de los objetos de nuestro modelo de datos.





Desarrollo (Back-End)



Para la autenticación de nuestra aplicación se ha utilizado la autenticación vía email/password que nos ofrece Firebase



Se han seguido las reglas que recomienda Google para desplegar el modelo de datos en una base de datos noSQL:

- Se ha intentado que la jerarquía de nodos sea mínima, lo ideal una tabla con un nodo padre y N hijos.
- Evitar las “joins”, si tenemos que acceder muchas veces a una tabla cuando cargamos otra es mejor redundar la información.
- Se han creado algunas tablas con estructuras que nos facilitan aplicar las reglas de seguridad, además tenemos tablas auxiliares que nos sirven de catalogo para validar la información que ingresa el usuario en el sistema.



Las Cloud Functions nos han permitido agregar triggers y ejecutar un script en “node.js” para dotar de lógica mas compleja a nuestra aplicación. Por ejemplo, cuando creamos una apuesta tenemos que modificar varias tablas principales y auxiliares.



Desarrollo (Back-End)



Para conectar la aplicación Ionic con Firebase hemos utilizado la librería oficial AngularFire2, gracias a esta librería hemos podido trabajar con programación reactiva utilizando eventos de JavaScript para construir flujos de datos (librería RxJS). El uso de esta tecnología nos ha permitido sincronizar colecciones de objetos y listas solo cuando ha habido un cambio sin tener que recargar constantemente la misma información.

Módulo de sincronización

- Este módulo se ha programado con Node JS y utiliza los paquetes request y promise para consumir de forma asíncrona la api de football-data.org y actualizar los eventos en la base de datos de FireBase.





Producto

☰ Ajustes

😊 Perfil de usuario

Nickname
Jessica

 [SELECCIONAR AVATAR](#)
[QUITAR AVATAR](#)

[GUARDAR CAMBIOS](#)

👤 Cuenta

[CAMBIAR CONTRASEÑA](#)

[CAMBIAR EMAIL](#)

☰ Liga de prueba

[INICIO](#) [APUESTAS](#) [RESULTADOS](#)

★ Jornada 38

Granada CF - RCD Espanyol
5/19/2017, 8:45 PM | L

1 (4.33 puntos) X (4.00 puntos) 2 (1.83 puntos)

Sporting Gijón - Real Betis
5/20/2017, 5:00 PM | W

1 (2.50 puntos) X (3.30 puntos) 2 (2.75 puntos)

RC Deportivo La Coruna - UD Las Palmas
5/20/2017, 7:00 PM | L

1 (1.70 puntos) X (4.00 puntos) 2 (4.50 puntos)

CD Leganes - Deportivo Alavés
5/20/2017, 7:00 PM | W

1 (2.10 puntos) X (3.40 puntos) 2 (3.50 puntos)


Sevilla FC - CA Osasuna
5/20/2017, 9:00 PM | L


1 (1.12 puntos) X (8.50 puntos) 2 (19.00 puntos)


☰ Liga de prueba

[INICIO](#) [APUESTAS](#) [RESULTADOS](#)

📊 Jornada 38

 Jessica
5 apuestas ganadas **13.40 puntos**

 Albert
4 apuestas ganadas **12.44 puntos**

 Luis
4 apuestas ganadas **9.19 puntos**

● Apuestas ganadas (5)

Sporting Gijón **2 - 2** Real Betis
Pronóstico: X | 2.75 puntos


CD Leganes **1 - 1** Deportivo Alavés
Pronóstico: X | 3.50 puntos

Valencia CF **1 - 3** Villarreal CF
Pronóstico: 2 | 3.40 puntos

Club Atlético de Madrid **3 - 1** Athletic Club
Pronóstico: 1 | 2.00 puntos

RC Celta de Vigo **2 - 2** Real Sociedad de

← Editar liga



Nombre
Liga de prueba

Competición:
⚽ Primera Division 2016/17

Cambiar administrador:

Jessica

Luis

Albert

[ACEPTAR](#)



Muchas gracias por vuestra atención