

Proyecto EA

FreePong App

*Autores:*

Raúl González

Raúl Lorenzo

Jesús Tardío

Carlos Valdés

# Índice

# Introducción

Primeramente es de recibo explicar el contexto en el que se alcanza la idea de diseñar una App cómo Freepong. Dado que se nos impusieron funcionalidades del tipo localización GPS en la que interactuaran los usuarios, y tras barajar diferentes ideas que girarían en torno a ésta funcionalidad principal, llegamos al consenso de diseñar una aplicación orientada a la interactuación entre las personas.

Freepong da la oportunidad de poder hacer quedadas de aficionados al ping pong aprovechando el mobiliario urbano de la ciudad y poder apuntarte a partidas en tiempo real. La localización de las mesas disponibles viene dada ya por la App, ingresadas estas desde el perfil de administrador (que también hemos implementado) dando la libertad y la facilidad a los usuarios registrados de directamente comprobar en la localización de la mesa si hay partidas disponibles, o si en cambio, este mismo puede crear la suya.

Para generar más aliciente a la App en sí, hay disponible un historial de victorias de las partidas cómo desarrollaremos más adelante a lo largo de esta memoria.

Creemos que el punto fuerte de esta App es que se puede extrapolar este sistema a diferentes deportes o aficiones que se desarrollen en una localización fija, dando así una escalabilidad y probabilidad de crecimiento bastante amplia, con relativo poco esfuerzo de desarrollo.

# Objetivos del proyecto

1. **Pila MEAN (MongoDB/Mongoose, Express, Angular i NodeJS)**

Uno de los objetivos principales en el desarrollo de sobretodo la parte Backend aunque también de la Frontend, es decir la API en sí, el motor, la inteligencia de la aplicación ha sido trabajar por primera vez con la pila MEAN.

Hemos podido trabajar y desarrollar una aplicación web sólida basada en esta combinación de diferentes sistemas en los que, una vez bien coordinados y sincronizados, es muy sencillo añadir plugins y funcionalidades.

El sistema de BBDD MongoDB o su integración en la API Mongoose, facilitan mucho el desarrollo a la hora de realizar cualquier tipo de consulta ya que se basa en arquitectura JSON, de forma que la estructura de los datos es más libre y no tan estricta como en cursos anteriores en los que se usaban lenguajes como MySQL.

NodeJS y Express hacen posible la incusión de librerías y estándares desde el inicio del proyecto y permite también, unificar en caso de desarrollo diversificado en diferentes equipos, que todo el mundo trabaje con las mismas versiones. También hay que destacar la utilización del controlador de versiones Bower, que realiza funciones similares a las de Node y Express y en nuestro caso hemos usado para mantener la versión del lenguaje principal AngularJS.

Por último, todo el diseño del FrontEnd ha sido implementado con AngularJS en el que hemos podido comprobar la evolución de JAVAScript y lo realmente escalable.

1. **Aplicaciones híbridas IONIC/CORDOVA**

Una de las particularidades principales de este proyecto es su versión de aplicación para SmartPhones. Ionic es un SDK completo de software libre basado en lenguaje AngularJS y Apache Cordova con el que se puede realizar un desarrollo usando tecnologías web tipo HTML5 y CSS.

En nuestro proyecto se ha integrado un módulo Ionic que “ataca” al servicio principal de nuestra API funcionando de forma paralela.

Éste objetivo nos parece particularmente interesante ya que acerca y simplifica el desarrollo de las aplicaciones para Smartphones con un lenguaje más cercano a nuestros conocimientos, sabiendo de la complejidad de otros lenguajes nativos de los teléfonos inteligentes tipo Android y iOS.

1. **Aplicación RESTFUL**

Como en asignaturas anteriores, en este proyecto también se ha impuesto el sistema RESTFUL que se basa en una estructura cliente/servidor sin estado es decir, que cada mensaje HTTP contiene toda la información necesaria para comprender la petición.

En las diferentes funcionalidades de nuestro proyecto se han aplicado las 4 operaciones principales de los sistemas RESTFUL como son las operaciones CRUD, principalmente los métodos POST, PUT, DELETE y GET para realizar todas las operaciones contra el servidor.

1. **Geolocalización**

Este es un apartado en el que ninguno de los integrantes del equipo hemos desarrollado funcionalidades anteriormente. Dado que la funcionalidad y misión principal de nuestra App gira en torno a esta nueva posibilidad hemos querido apostar sobre seguro y hemos integrado el motor de Goggle Maps en nuestro proyecto. En el desarrollo punto por punto detallaremos la manera de hacer funcionar esta potente herramienta y cómo adaptarla a nuestra App.

Lo interesante de incluir este plugin es que ya funciona como un sistema independiente, es decir que conserva la gran mayoría de sus funcionalidades que le han hecho tan útil y popular como la Street View, por destacar un ejemplo.