

## Capítulo 6

# Conclusiones

En los capítulos anteriores se ha hecho una descripción de las tecnologías empleadas y se ha presentado la aplicación desarrollada. Además, se ha argumentado tanto la elección final del diseño como su arquitectura. En este capítulo se analizarán las conclusiones del trabajo realizado y se proponen posibles mejoras futuras de desarrollo.

### 6.1. Conclusiones

Tras analizar el trabajo realizado se puede apreciar que se ha conseguido satisfactoriamente el objetivo general. Se ha creado una aplicación web capaz de poner en contacto alumnos y profesores para dar clases particulares. Esta disponible en [www.classcity.es](http://www.classcity.es). Además se han cubierto los cuatro sub-objetivos marcados en el capítulo 2.

- **Front-End:** El lado cliente de la aplicación se ha desarrollado satisfactoriamente utilizando Angular, una tecnología cuya curva de aprendizaje ha sido superada a pesar de la grandes dificultades que contiene. Angular nos ha permitido tener una estructura de la aplicación lo bastante sólida y escalable para poder introducir cualquier tipo de mejora en un futuro.
- **Back-End:** En cuanto al lado servidor se ha diseñado y programado un servidor web gracias a la facilidad que tiene Node y su entorno Express en su aprendizaje, además de la numerosa comunidad de desarrolladores que hay detrás. Todo esto nos ha supuesto que crear un servidor desde cero, sea algo más sencillo de lo que esperábamos. El servidor incluye autenticación de usuarios, varios roles (profesor y alumno), permite el filtrado de profesores cercanos al área geográfica del estudiante (usando el API de geolocalización), además de ofrecer la posibilidad de chats en directo entre alumno y profesor cuyos perfiles cuadran entre sí usando websockets.
- **BBDD:** La integración del servidor con MongoDB ha sido todo un éxito gracias a moongoose, una librería que nos ha permitido enlazar el servidor con la base de datos sin demasiadas complicaciones. Aunque comprender el modelo NoSQL y sus beneficios, no ha sido algo trivial. Se diseñó y programó un modelo de datos con estudiantes, profesores, temática de las clases, georreferencias, etc. utilizando un modelo de datos no relacional.
- **Despliegue en la red:** El último de los sub-objetivos consistía en subir la aplicación web a producción. Este sub-objetivo se ha desarrollado satisfactoriamente utilizando

Amazon Web Services. Si accedemos a [www.classcity.es](http://www.classcity.es) podemos ver nuestra aplicación corriendo en producción.

## 6.2. Trabajos Futuros

Este TFG ha sido diseñado para poder extender sus funciones en un futuro sin necesidad de cambiar la arquitectura. Hemos identificado cuatro funcionalidades nuevas que harían a nuestra aplicación mas interesante.

1. **Añadir WEB RTC:** WebRTC es una API que está siendo elaborada por la World Wide Web Consortium para permitir a las aplicaciones del navegador realizar llamadas de voz, chat de vídeo y uso compartido de archivos P2P sin plugins. Una de las mejoras que podría tener esta aplicación sería el uso de web RTC para que los profesores pudieses impartir sus clases a partir de videoconferencia.
2. **Añadir Comentarios y valoraciones a los profesores:** Otra de las mejoras que podrían ser empleadas es la valoración de los profesores por parte de los alumnos, utilizando el potencial de mongoDB para almacenar información.
3. **Añadir Calendario:** Otra funcionalidad extra que se puede introducir en la aplicación es la utilización de calendarios para que los alumnos puedan consultar la disponibilidad de cada profesor
4. **Añadir un método de pago:** Por último se podría añadir la mejora de monetizar el uso de la aplicación. Añadiendo un método de pago a partir de la aplicación cada vez que un alumno quiera recibir una clase particular.