

los anglicismos los evitaría por su equivalente en castellano (p.e. entorno en vez de framework), y si es inevitable, en *itálica*

periódicos, entradas de cine, televisión...

Objetivo de este capítulo: describir contexto de tu TFG (el general es el de tecnologías web), motivación de la utilidad (cuenta varias aplicaciones web similares que sean de uso extendido airbnb, blablacar wallapop...), precedentes (este es el contexto cercano, otros TFGs en el mismo ámbito). Y también comentar breve e informalmente el objetivo (con este párrafo mejor redactado valdría).

arquitectura de las aplicaciones web: http, lado cliente, lado servidor... menciona tecnologías de uno y otro lado... html5, aplicaciones de audio y video, páginas dinámicas...

# Capítulo 1

## Introducción

webs intermediarias entre ofertores y demandantes...

### 1.1. Motivación

mejor impersonal, tercera persona. No es una novela, ni una biografía, no interesa el hilo temporal, mejor se plantea un problema y se describe la solución a la que has llegado. Es más ingenieril.

#### ~~1.1.1. Antes de empezar...~~

Antes de empezar a desarrollar una aplicación, debes saber que no será un camino sencillo, pero si muy gratificante. Cuando empecé a desarrollar la aplicación no sabía la cantidad de problemas a los que me iba a enfrentar, pero como tenía la ilusión de poder hacer realidad una idea todo fue más sencillo.

#### ~~1.1.2. Classcity~~

La idea de la que hablo se llama Classcityz consiste en una aplicación web cuya función principal es contactar profesores con alumnos para que puedan quedar y dar clases particulares de cualquier asignatura y curso. Esta idea nace porque llevo muchos años dedicándome a la enseñanza de clases particulares y pienso que hay una gran oportunidad de organizar mejor la comunicación profesor-alumno.

#### 1.1.3. M.E.A.N.

Vas a dedicar todo un capítulo a describir las tecnologías en las que te has apoyado

Classcity ha sido desarrollada con 4 tecnologías diferentes que forman el famoso stack M.E.A.N. haciendo referencia a las siglas Mongo, Express, Angular y Node. Desde el cliente al servidor pasando por la base de datos, todas con el mismo punto en común. Desarrollo end-to-end usando JavaScript tanto en el frontend, backend y la base de datos. Sin entrar en definir complementamente cada una de las tecnologías que lo componen nos encontramos con:

- MongoDB como base de datos que almacena documentos JSON.
- Express como web framework basado en Node.js que nos permite crear API REST, por ejemplo
- Angular como framework para crear la parte cliente de la aplicación en formato Single Page.

¿Qué es API REST? Hay que contarlo antes.

- Node.js como framework JavaScript basado en V8 que proporciona funcionalidades core para nuestra aplicación bajo un modelo asíncrono de eventos.

#### 1.1.4. Javascript

#### JavaScript

MEAN como cualquier plataforma no puede vivir sin su comunidad, pero si tenemos en cuenta de que todas las tecnologías utilizan javascript como lenguaje de programación principal, podemos estar hablando de un stack bastante importante para la comunidad javascript, ya que todos los desarrolladores de front se podrán adentrar dentro del back y así poder hacer su propia aplicación.

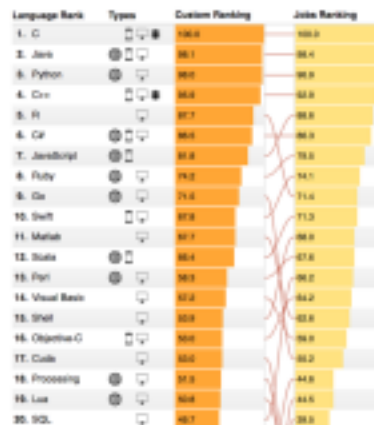


Figura XX

En este cuadro podemos encontrar información relevante de cómo están valorados los lenguajes de programación, en relación a la popularidad general de cada lenguaje de programación, pero también aplicando filtros por plataforma (computadores, móviles, web y sistemas embebidos) y por solicitud de empleo. Este estudio comenzó seleccionando unos 300 lenguajes en Github y escogiendo los más populares hasta quedarse con estos 20 candidatos para después estudiar la popularidad de esos lenguajes en 10 plataformas, que incluyen la propia Github, Google, Stack Overflow y paginas de oferta de empleo como Career Builder o Dice.

Javascript a pesar de estar bien posicionado en el ranking tiene ventajas como que es un lenguaje muy sencillo y versátil, puesto que es muy útil para desarrollar páginas web y aplicaciones web. Otra cualidad que hace importante a javascript es su rapidez de computo, pudiendo ejecutar funciones inmediatamente, y por último y para mi la ventaja más importante es que es el único lenguaje que te permite trabajar modo FullStack en cualquier tipo de desarrollo de programación. Pero no todo puede ser buenas noticias para javascript, también tiene inconvenientes como por ejemplo que en el FrontEnd sus códigos son visibles, por lo tanto pueden ser leídos por cualquier usuario, y por otro lado javascript también tiende a introducir gran cantidad de fragmentos de código en los sitios web.

### 1.1.5. Plataforma Cloud

También tengo que añadir la complejidad de subir una aplicación en la nube a través de la plataforma Amazon Web Service. Hay muchas otras plataforma de cloud como Azure(Microsoft), Google Cloud... Pero elegí AWS debido a que tiene un periodo de prueba gratuito y es bastante mas sencillo de manejar que Azure o Google Cloud. Además AWS es el líder indiscutible, aunque sus competidores últimamente alcanzan mas importancia en el mundo del cloud.

No justifiques aquí las decisiones (JS, AWS...) este capítulo sólo es para describir el contexto y presentar informalmente el objetivo. Aún no has presentado el problema que vas a abordar, ¿cómo se van a entender entonces tus elecciones de diseño?

no lo relaciones aún con tu TFG, simplemente presentalo como una tecnología en la que se pueden apoyar las aplicaciones web, sus ventajas, etc... Menciona ejemplos. En el capítulo 3 ya describirás con un poco más detalle técnico AWS y en el 4 justificarás tu elección.

faltan precedentes...

falta párrafo presentando los siguientes capítulos que esperan en la memoria.

ANTES DE LANZARTE A ESCRIBIR ORDENA EN TU CABEZA QUÉ QUIERES CONTAR Y ARTICULALO EN UNAS SECCIONES