

University of London  
Imperial College of Science, Technology and Medicine  
Department of Computing

## **Anteproyecto de Grado**

Joe Bloggs

Submitted in part fulfilment of the requirements for the degree of  
Doctor of Philosophy in Computing of the University of London and  
the Diploma of Imperial College, October 2017



## Abstract

Text of the Abstract.



## Acknowledgements

I would like to express (whatever feelings I have) to:

- My supervisor
- My second supervisor
- Other researchers
- My family and friends



## Dedication

Dedication here.

‘Quote text here.’

*Guy Quoted*



# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>1 Marco teórico</b>	<b>1</b>
1.1 Programación orientada a objetos . . . . .	1
1.2 Modelo Entidad-Relación . . . . .	1
1.3 Modelo Vista Controlador . . . . .	2
1.4 Diagrama de Clases . . . . .	2
1.5 Diagrama de casos de uso . . . . .	2
1.6 Front-End . . . . .	2
1.6.1 AngularJs 2.0 . . . . .	2
1.6.2 Sass . . . . .	3
1.6.3 TypeScript . . . . .	3
1.6.4 Apache Cordova (Phonegap) . . . . .	3
1.6.5 Ionic 2.0 . . . . .	3
1.7 Back-End . . . . .	3
1.7.1 Motor de base de datos PostgreSQL . . . . .	4

1.7.2	Java . . . . .	4
1.7.3	JPA . . . . .	4
1.7.4	Servidor de aplicaciones JBoss WildFly . . . . .	4
<b>2</b>	<b>Background Theory</b>	<b>5</b>
2.1	Introduction . . . . .	5
<b>3</b>	<b>Conclusion</b>	<b>6</b>
3.1	Summary of Thesis Achievements . . . . .	6
3.2	Applications . . . . .	6
3.3	Future Work . . . . .	6
	<b>Bibliography</b>	<b>6</b>

# List of Tables



# List of Figures



# Chapter 1

## Marco teórico

### 1.1 Programación orientada a objetos

Paradigma programación orientada a objetos. Representación de la realidad orientada a objetos. (Clases, atributos, métodos). Conceptos (Herencia, encapsulación, polimorfismo).

### 1.2 Modelo Entidad-Relación

Entidad. La entidad es cualquier clase de objeto o conjunto de elementos presentes o no, en un contexto determinado dado por el sistema de información o las funciones y procesos que se definen en un plan de automatización. Relación. Vínculo que permite definir una dependencia entre los conjuntos de dos o más entidades. Esto es la relación entre la información contenida en los registros de varias tablas.

Según el estándar las clases de entidad se muestran con rectángulos, las relaciones con diamantes y la cardinalidad máxima de relación se indica dentro del diamante.

## 1.3 Modelo Vista Controlador

Modelo: Se trata del nucleo funcional que gestiona datos manipulados en aplicación. Vista: se trata de los componentes destinados a representar la informacion al usuario. Controlador: este recibe los eventos que provienen del usuario y los traduce en consultas para el modelo y la vista. diamante

## 1.4 Diagrama de Clases

Diagrama utilizado en POO para representar objetos. Representa sus atributos, métodos y relaciones entre objetos.

## 1.5 Diagrama de casos de uso

Permite describir las acciones que un usuario puede desempeñar dentro de un sistema.

## 1.6 Front-End

Básicamente el software con el que el usuario interactúa. Captura y muestra información. Se piensa realizar el desarrollo de una aplicación móvil híbrida, para garantizar que esta pueda ser ejecutada bajo cualquier sistema operativo móvil (Ej. Android, iOS, etc).

### 1.6.1 AngularJs 2.0

Framework para desarrollo de aplicaciones web. Propietario de Google.



### 1.6.2 Sass

Hojas de estilo. Básicamente una versión mejorada de Css. Compilado. Genera CSS, permite heredar reglas CSS.

### 1.6.3 TypeScript

Mejora de JavaScript. Desarrollado sobre JavaScript. Básicamente aplica paradigma de POO en lenguaje JavaScript. Es compilado, genera código JavaScript.

### 1.6.4 Apache Cordova (Phonegap)

Framework y conjunto de librerías para desarrollo de aplicaciones híbridas. Haciendo uso de tecnología Web. Librerías que permiten acceso a sensores del dispositivo.

### 1.6.5 Ionic 2.0

Framework para desarrollo aplicaciones móviles híbridas. Desarrollado sobre el Framework Angular 2.0.

## 1.7 Back-End

Básicamente el software con el que el usuario no interactúa directamente pero que es vital para proveer servicios de datos que serán consumidos por el Front-End para ofrecer una funcionalidad.

### 1.7.1 Motor de base de datos PostgreSQL

PostgreSQL es un motor de bases de datos relacionales (RDBMS) que verifica integridad referencial con gran funcionalidad como base de datos, aunque un poco más lenta que otros motores. Su licencia es tipo BSD. En esta sección describimos brevemente la instalación y uso en un sistema adJ.

### 1.7.2 Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

### 1.7.3 JPA

Java Persistence API es un conjunto de clases y métodos que persistentemente almacenar la gran cantidad de datos a una base de datos que es proporcionada por Oracle Corporation.

### 1.7.4 Servidor de aplicaciones JBoss WildFly

# Chapter 2

## Background Theory

### 2.1 Introduction

Text of the Background.

# Chapter 3

## Conclusion

### 3.1 Summary of Thesis Achievements

Summary.

### 3.2 Applications

Applications.

### 3.3 Future Work

Future Work.

# Bibliography