



ugr

Universidad
de Granada

MEMORIA DE PRÁCTICAS
MÁSTER EN INGENIERÍA INFORMÁTICA

Prácticas de empresa

Nazaríes IT



Nazaríes IT

Autor

Francisco Javier Bolívar Lupiáñez

Directores

Pedro Villar Castro (tutor académico)

Nombre Apellido1 Apellido2 (tutor en la empresa)



E.T.S. DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada, 23 de julio de 2017

Índice general

1. Introducción	1
2. Descripción de la entidad donde realizó las prácticas	3
3. Trabajo realizado	5
3.1. Primera semana	5
3.2. Segunda semana	6
3.3. Tercera semana	6
3.4. Cuarta semana	7
3.5. Quinta semana	7
4. Valoración personal del trabajo realizado	9
5. Conclusiones	11
Bibliografía	13

Capítulo 1

Introducción

En este documento voy a presentar mi memoria de prácticas de empresa del máster en ingeniería informática de la Universidad de Granada.

Comenzaré con una sección en la que realizaré una descripción de la empresa en la que he realizado tales prácticas, Nazaríes IT.

En la siguiente sección contaré el trabajo que he ido realizando durante las seis semanas indicando las tareas, así como las soluciones aportadas, las herramientas utilizadas y las asignaturas de la titulación que han resultado útiles para resolverlas.

A continuación realizaré una valoración personal del trabajo que he realizado y acabaré con unas conclusiones.

Capítulo 2

Descripción de la entidad donde realizó las prácticas

Nazaríes IT es una empresa granadina del sector de las TIC (Tecnologías de la Información y Comunicaciones) que lleva más de cinco años desarrollando soluciones para clientes de diversa índole.

Muchos de estos proyectos están relacionados con el IoT (Internet de las Cosas, del inglés *Internet of Things*) recogiendo información de sensores, tratándola para posteriormente analizarla y sacar conclusiones.

Además desarrolla muchos otros proyectos de *software* a medida, la mayoría soluciones web.

La empresa cuenta con unos 30 empleados que se separan en tres sectores:

- Ventas y atención al cliente: Se encargan de atender a los clientes así como negociar proyectos.
- Administración: Son los que llevan el tema administrativo y contractual.
- Desarrollo: Es el que tiene el grueso de la plantilla. Se divide a su vez en dos, la parte de desarrollo *software* (más amplia) y la parte de *hardware* y telecomunicaciones.

Todo el equipo se encuentra en las oficinas de un mismo edificio y cada empleado cuenta con su propio ordenador, a su vez, en el departamento de *hardware* cuentan con sensores como los que utilizan en producción para realizar distintas pruebas.

Capítulo 3

Trabajo realizado

Las prácticas de empresa del máster en ingeniería informática cubren un total de 150 horas. Lo que viene a ser seis semanas de lunes a viernes con una jornada laboral de cinco horas.

3.1. Primera semana

Durante la primera semana estuve de formación pues no conocía *Ruby on Rails* que es el *framework* que utilizan en la mayoría de los proyectos web.

El primer día estuve instalando y configurando las herramientas que iba a necesitar en mi ordenador y siguiendo rápidamente un curso *online* gratuito de *Ruby* [1] para refrescar conocimientos, pues es un lenguaje que ya había utilizado en el segundo curso del grado en ingeniería informática en la asignatura de Programación y Diseño Orientado a Objetos. Además, sabiendo que era el que utilizaban, tres semanas antes de comenzar las prácticas pude aprovechar unas horas libres para hacer un curso en el que complementé los conocimientos que adquirí en la asignatura llegando a implementar en *Ruby* un sistema de recomendaciones como el que realicé en *Java* en la asignatura del máster de Gestión de Información en la Web. Este se encuentra disponible en mi cuenta de *GitHub*: [fblupi/recommender-system-cf-ruby](https://github.com/fblupi/recommender-system-cf-ruby).

Como ya tenía conocimientos de *Ruby* realicé muy rápido éste tutorial, lo que me permitió, a partir del segundo día, comenzar a hacer el curso, también *online* y gratuito de *Ruby on Rails* [2] con el que se aprendía a crear un blog desde cero con base de datos, sesiones, etc.

Este curso me llevó el resto de días de la semana pues era bastante más extenso. Además de encontrarme con un fallo en mitad del desarrollo que no llegué a detectar y dejaba parte de la aplicación inutilizable (impedía

llamar a métodos con el verbo *HTTP DELETE*) y no llegué a detectar, y al no haber utilizado un sistema de control de versiones, un error por mi parte, perdí cuatro horas en rehacer el trabajo. Fue entonces, al rehacer lo que llevaba de aplicación de nuevo, cuando detecté que era un problema por la incompatibilidad entre dos gemas que utilizaba.

El proyecto del blog realizado con este curso lo tengo disponible en mi cuenta de *GitHub*: fblupi/blogfacilito.

3.2. Segunda semana

Una vez terminado el primer curso de *Ruby on Rails* empecé con otro [3], bastante más pequeño que me llevó media mañana, pero me permitió afianzar conocimientos. También era de realizar un blog y se encuentra en un repositorio de *GitHub* fblupi/blog-librosweb.

Ya con conocimiento de *Ruby on Rails* comencé a realizar tareas de administración en uno de los proyectos intercalando con un curso de *CoffeeScript* [4] cuyo contenido se puede ver en uno de mis repositorios de *GitHub*: fblupi/learning-coffeescript.

Para comprobar mis conocimientos de *CoffeeScript* realicé el mismo sistema de recomendaciones que realicé en *Ruby* en este otro lenguaje y también lo subí a *GitHub*: fblupi/recommender-system-cf-coffee

Las tareas de administración que realicé fueron las de migrar estaciones de una plataforma antigua a una nueva, pues algunas variables de algunos sensores necesitaban que se les aplicase una transformación para mostrar el valor en las unidades que los clientes quieren.

Esto me ayudó a, sin ver el código, tener una visión general del proyecto. Ya que desde el panel de administración de *Rails*, donde realizaba todas las operaciones, podía ver la estructura de clases que se seguía.

3.3. Tercera semana

En esta tercera semana comencé con una tarea sencilla de desarrollo, además aprendí a utilizar *RubyMine* ya que la empresa compra licencias de este IDE (Entorno de Desarrollo Integrado, del inglés *Integrated Development Environment*) que cuenta con muchas funcionalidades que hacen más sencillo el seguimiento de la traza por donde se ejecuta el programa entre todos los distintos ficheros. Además de tener una integración total con *Git*, lo que permite ver quién hizo qué cambio en cada línea para ir a preguntar directamente a ese compañero si surge alguna duda al respecto.

Mi objetivo con esta tarea era crear un nuevo tipo de estación, lo que permitiría especificar funcionalidades para éstas. Mis conocimientos en asignaturas como Programación y Diseño Orientado a Objetos, Sistemas de Información Basados en Web del grado o Sistemas Software Basados en Web del máster me fueron de mucha utilidad para llevar a cabo esta tarea.

3.4. Cuarta semana

Ya con el nuevo tipo de estación creado y funcionando, pasé a crear la funcionalidad específica de éste. A un cálculo de facturas de contador básico que implementó un compañero le añadí nueva funcionalidad como el cálculo de la penalización por energía reactiva y la discriminación del tipo de contador para hacer de una forma u otra el cálculo.

Esta tarea me llevó la semana entera pues había que realizar muchas pruebas y tuve que entender muy bien el proceso del cálculo de la factura de la luz, el cuál no conocía.

Aún así, no pudimos comprobar el cálculo para un tipo de contador específico pues no contábamos con datos suficientes para ello.

3.5. Quinta semana

Durante esta semana realicé pasé de trabajar en el *back-end* del proyecto al *front-end* arreglando las hojas de estilo de la web *responsive* para dispositivos más pequeños pues había varias inconsistencias.

Una vez terminado esto, pasé de nuevo al *back-end* con una tarea que consistía en exportar los informes de las estaciones al formato XLSX (el formato estándar de hojas de cálculo de *Microsoft Excel*) usando la gema *axlsx-rails*. No obstante tuve problemas a la hora de exportar a este tipo de archivo binario, pues los archivos generados con esta gema se encontraban corruptos y no podían ser abiertos.

Los últimos días de la semana los aproveché para realizar un tutorial de *Angular*, pues era una tecnología que iba a hacer falta para un futuro proyecto. El tutorial que seguí fue el que aportaba la web oficial de *Angular* y el código generado lo subí a un repositorio de GitHub: fblupi/tour-of-heroes.

Capítulo 4

Valoración personal del trabajo realizado

Lala

Capítulo 5

Conclusiones

Al realizar estas prácticas de empresa no he necesitado mucho tiempo de formación más allá del que cualquier otra persona habría necesitado. Pues apenas he necesitado una formación extra en *Ruby on Rails* y en *CoffeeScript*, un *framework* y un lenguaje específicos que se puede entender que no se vean durante nuestra formación académica.

Sin embargo, me parece grave que haya estudiantes que terminen su formación sin saber que existen los *frameworks* de desarrollo web. En mi caso, aprendí *Django* en una asignatura del segundo cuatrimestre del máster, pero muchos de los que acaban el grado, si no han escogido una optativa específica terminan sin conocer ningún *framework*, y lo que es más grave, otros, dependiendo de la especialidad que hayan escogido, no han visto ni siquiera nada de desarrollo web el cuál es importantísimo hoy en día y, en mi opinión, debería conocer cualquier ingeniero informático.

Hablo de conocer un *framework* porque pasar de uno a otro es trivial y solo hay que tener en cuenta las peculiaridades que tenga o el lenguaje que se utilice. Ya que la mayoría se basan en rutas, modelos, controladores y vistas.

Otro de los puntos importantes que he tenido que aprender en la empresa es el uso de *Git* de forma colaborativa. En *Cloud Computing: Fundamentos e Infraestructuras* del máster ha sido la primera vez que he tenido que utilizar *Git* de forma obligatoria en una asignatura y aún así, aprendí un uso muy básico pues no era tampoco contenido implícito del temario. Sería interesante tener una asignatura, o varios seminarios y prácticas obligatorios en cursos avanzados del grado para aprender a utilizar un CVS (Sistema de Control de Versiones, del inglés *Control Version System*) pues lo más seguro es que se necesite en un futuro empleo.

Bibliografía

- [1] Código Facilito. Curso de ruby básico gratis. <https://codigofacilito.com/cursos/ruby>.
- [2] Código Facilito. Curso de ruby on rails 4 gratis. <https://codigofacilito.com/cursos/rails>.
- [3] LibrosWeb. Introducción a ruby on rails. https://librosweb.es/libro/introduccion_rails/.
- [4] Siro Días Palazón. Coffeescript desde 0. https://www.youtube.com/playlist?list=PLf46te__lS1_ZLn9ZnPyMUNx0Hen9z5CX.