



ugr

Universidad
de Granada

MEMORIA DE PRÁCTICAS
MÁSTER EN INGENIERÍA INFORMÁTICA

Prácticas de empresa

Bi 4 Group



Autor

Aythami Estévez Olivas

Directores

Pedro Villar Castro (tutor académico)

Nombre Apellido1 Apellido2 (tutor en la empresa)



E.T.S.I.I.T.

Granada, 24 de septiembre 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
3.6. Sexta semana	8
4. Valoración personal del trabajo realizado	9
5. Conclusiones	11
Bibliografía	13

Capítulo 1

Introducción

En este documento recogeré mi experiencia del desarrollo de las prácticas curriculares en la empresa **Bi4 group**.

Estás tuvieron una duración de 150 horas a 5 horas diarias lo que hace un período de 6 semanas del 3 de Julio de 2017 al 14 de Agosto del mismo año.

El documento se estructurará de la siguiente manera:

- En primer lugar haré una descripción de la empresa, su ámbito de trabajo, estructura y metodologías de trabajo.
- A continuación explicaré el trabajo que he llevado a cabo en mis prácticas semana a semana con un nivel de detalle elevado.
- Proseguiré con una valoración de mi experiencia de prácticas.
- Terminaré con unas conclusiones sobre esta experiencia.

Capítulo 2

Descripción de la entidad donde realizó las prácticas

Bi4 Group se define como una empresa que desarrolla soluciones software a medida para sus clientes [1]. Nació en Ámsterdam en 2012 y actualmente cuenta con unos 30 empleados repartidos entre sus oficinas de Ámsterdam, Madrid y Granada. Aunque la mayor parte de los empleados se concentran en Granada por ser donde se encuentra todo el desarrollo, el resto de oficinas cuentan con personal administrativo o de ventas además de los posibles desarrolladores desplazados al lugar para ponerse en contacto con los clientes.

Su principal área de negocio es la creación de aplicaciones web complejas, incluyendo elementos de *Business Intelligence*, visualización de datos y *Big Data* siguiendo un modelo de *Custom Software as a Service (CSaaS)* que se vale del desarrollo basado en componentes y despliegues en la nube para agilizar el proceso de desarrollo, reutilizando los componentes comunes de las aplicaciones y centrándose en adaptar el producto a las necesidades específicas del cliente.

Como metodología de desarrollo emplean múltiples métodos ágiles como Scrum y Kanban donde el cliente siempre está presente de modo que la comunicación con él es continua y siempre conoce el estado del proyecto. Estos métodos utilizan pequeños equipos de 5-6 personas máximo, que se autoorganizan y permiten llevar a cabo múltiples proyectos en paralelo aún con un pequeño número de empleados.

Capítulo 3

Trabajo realizado

Durante mis prácticas he pertenecido al equipo de desarrollo de un proyecto interno que tenía como objetivo la gestión de servidores en *IaaS*, permitiendo establecer tareas automáticas sobre estos, apagarlos y encenderlos. Todo esto en un formato de *Single Page Application (SPA)* donde existan diversos usuarios y roles que puedan realizar distintas tareas sobre estos servidores. Como *stack* de desarrollo se ha usado *Mongo DB*, *Flask* y *Angular JS*; repartido en contenedores *Docker* y utilizando *Python 3.x*.

3.1. Primera semana

Esta primera semana consistió en una adaptación a las prácticas, se me dio un ordenador Ubuntu recién formateado y estuve instalando las herramientas necesarias como el IDE *PyCharms*, la herramienta de gestion de BD *RoboMongo*, *Docker*, ...

Para poder trabajar en el proyecto donde desarrollaría mis prácticas requería de acceso a varios proyectos situados en el servidor *git* de la empresa, por ello según avanzaba en el despliegue del mismo iban surgiendo problemas que tras lograr solucionar documenté y actualicé las instrucciones del repositorio de la empresa.

Una vez logrado esto me dispuse a realizar un tutorial de Angular JS [2], ya que la parte de *front-end* en mi proyecto utilizaba dicho *framework* y yo no tenía ninguna experiencia previa con este. El desarrollo de dicho tutorial se encuentra disponible en mi *GitHub* en el siguiente enlace.

Para completar la primera semana me dispuse a realizar mi primera tarea del *sprint* actual, esta consistía en realizar un *CRUD (Create, Read, Update, Delete)* utilizando algunas librerías internas de la empresa siendo el objeto de este *CRUD* servidores que correspondería a instancias de EC2

de *Amazon Web Service(AWS)*. Era la primera vez que veía las librerías internas de la empresa e intenté imitar lo que se realizaba con los usuarios, pero debido a mi desconocimiento y a algunos problemas con los usuarios que me impedían comprobar algunas funcionalidades esta tarea quedo sin acabar en este *sprint*.

3.2. Segunda semana

Continuando la tarea anterior comencé desarrollando el *back-end* de mi *CRUD* y testeando la *API Rest* creada con la herramienta *Postman*.

Para aclarar algunos conceptos sobre las librerías de la empresa estuve instalando y probando otros proyectos de ejemplo que mostraban su funcionamiento de una manera más simple, con ello comprendí mejor la lógica de estas librerías y realicé algunas modificaciones en mi tarea *CRUD*.

A pesar de esto no era capaz de hacer funcionar totalmente el *back-end* por algunos problemas con las peticiones HTTP POST, esto se debía a que no estaba añadiendo algunos campos requeridos por las librerías de la empresa para realizar un *CRUD*. Una vez hecho esto logré hacer funcionar el *back-end*, tras lo que añadí código disponible en otros repositorios de la empresa para conectarse a *AWS* y recuperar la lista de instancias asociada a unos credenciales haciendo que estos se guarden en la BD y se puedan refrescar al hacer otra petición a *AWS*.

Como es lógico ahora pasé a desarrollar el *front-end* utilizando otros como ejemplos. Esta semana logré mostrar los servidores en la BD en una tabla con sus campos más destacados y un botón que lanzaba una petición a *AWS* para refrescar la lista, pero me faltó implementar la búsqueda y la ordenación por columnas de la tabla.

3.3. Tercera semana

Los compañeros del proyecto me comentaron en el *daily stand-up* que sería necesario añadir un botón en la tabla de servidores para parar o arrancar, dicho botón llama a un método de la *API* que utiliza el paquete *python boto* para realizar la tarea deseada sobre la instancia seleccionada además de notificar por *Slack* la acción aplicada sobre la instancia.

Adicionalmente implementé la búsqueda y la ordenación por columnas pendiente la semana previa así como diversas mejoras de usabilidad como mostrar mensajes de confirmación al realizar una acción sobre un servidor o mostrar un *spinner* en caso de que se encontrará realizando una acción.

Tras esto cogí una nueva tarea del *sprint* que se basaba en hacer diversos

cambios en el *front-end*, como crear una nueva ventana de login con un logo para la aplicación, resolver algunos problemas con las pestañas o seleccionar la página por defecto al acceder a la aplicación.

Para la instalación de la aplicación existía un *script* para inicializar la BD con los datos por defecto necesarios para que empiece a funcionar la aplicación, pero dicho *script* se encontraba en proceso de migración de *Python 2* a *Python 3* por lo que no funcionaba correctamente, por ello cogí esa tarea y estuve retocando cosas y entendiendo la lógica de inicialización que debería de seguir todo.

3.4. Cuarta semana

Como comenté en la primera semana 3.1 existían problemas con el *CRUD* de usuarios que imposibilitaba añadirlos desde el *front-end*, este *bug* resultó ser más complejo de lo esperado, debido a que la clase usuario heredaba de una librería interna de la empresa donde se producía el fallo.

También estuve realizando una tarea de gestión de roles de usuario con el objetivo de que existieran dos perfiles (administrador y usuario estándar) asignables a cada usuario y que esto limitara su acceso a servidores haciendo que los usuarios estándar solo pudieran manejar servidores marcados como accesibles por parte de algún administrador.

3.5. Quinta semana

He comenzado esta semana realizando una tarea del sprint de gestión de credenciales de *AWS* de modo que se puedan configurar desde el *front-end*, para ello se ha creado un *CRUD* de credenciales, teniendo la precaución de almacenar en la BD la *secret access key* encriptada. Todo esto se ha acompañado de una serie de mensajes de éxito o error para guiar al usuario en la resolución de posibles problemas. Por ejemplo si se acaba de desplegar la aplicación por primera vez no habrá credenciales fijados, por lo que la recuperación de la lista de servidores fallará, esto causará un mensaje de error indicando al usuario que el error seguramente se deba a la falta de credenciales e indicándole donde puede fijarlos.

Aunque en la aplicación se recogían *logs*, estos no contenían el nombre del usuario, las fechas estaban en un formato muy poco legible y su visualización no era la más indicada, por ello estuve retocándolos y mejorándolos.

3.6. Sexta semana

La última semana he estado haciendo algunos retoques para dejar el sistema funcionando correctamente:

- Inicialización consistente con la BD: cuando los contenedores se paran y se vuelven a arrancar los ficheros de credenciales y las tareas automáticas (utilizando *crontab*) no se vuelven a crear, por ello en arranque se comprueba la BD asociada y realizan las tareas necesarias para que el estado del contenedor sea consistente con el de la BD.
- Configuración de token *Slack* para crear un *bots* que notifiquen en un grupo concreto las acciones llevadas a cabo en los servidores, esto será una fuente de información muy importante para poder conocer el estado del sistema sin tener que acceder a los *logs* de la aplicación.
- Creación de las tareas automáticas utilizadas en la empresa como el arranque automático de los servidores de desarrollo a las 8 de la mañana (cuando empieza la jornada de trabajo) y la parada de los mismos al final de la jornada.
- Añadir un reporte del uso de los servicios *AWS* utilizando el paquete *boto* mencionado previamente, este reporte muestra datos de uso de todos los servicios utilizados por esas credenciales *AWS* así como un resumen de la facturación por meses. Esto será muy útil para comprobar como el uso de este sistema puede repercutir en la reducción de la factura.

Capítulo 4

Valoración personal del trabajo realizado

Lala

Capítulo 5

Conclusiones

Falta flexibilidad elegir empresas

Agradecimientos plantilla

Falta de información por parte de los coordinadores

Bibliografía

- [1] Bi4 Group. About us. <https://www.bi4group.com/about-us/>.
- [2] Code School. Shaping up with angularjs. <https://www.codeschool.com/courses/shaping-up-with-angularjs>.