

UNIVERSIDAD DE SONORA

División de Ingeniería
Departamento de Ingeniería Industrial

**BBOX4.0 - BANCO TAN FÁCIL COMO JUGAR
XBOX – SHARE.A(NYTHING).**

Reporte de Prácticas Profesionales

Presenta

FRANCISCO JAVIER SALAZAR ESPINOZA

INGENIERO EN SISTEMAS DE INFORMACIÓN

Asesor: Dr. Alonso Pérez Soltero

Hermosillo, Sonora.

Mayo 2017

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	3
1. INTRODUCCIÓN	4
1.1 EXPLICACIÓN DEL PROYECTO	4
1.2 OBJETIVOS	5
1.3 METODOLOGÍA	6
2. DESCRIPCION DEL CONTEXTO	8
2.1 EQUIPAMIENTO E INSTALACIONES	8
2.2 REGLAS DE OPERACIÓN DE LA UNIDAD RECEPTORA	9
2.3 ENTORNO DONDE SE UBICA LA UNIDAD RECEPTORA.	10
2.4 NORMATIVIDAD DE LA UNIDAD RECEPTORA	11
3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS.	12
4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES.	16
4.1 ACTIVIDADES PRELIMINARES	16
4.2 SPRINT 1	18
4.3 SPRINT 2.	18
4.4 SPRINT 3	19
4.5 SPRINT 4	21
4.6 SPRINT 5	22
4.7 SPRINT 6	24
4.8 SPRINT 7	25
5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA	27
5.1 ANÁLISIS GENERAL DEL PROYECTO	27
5.2 ANÁLISIS DE LOS OBJETIVOS DE LAS PRÁCTICAS	27
5.3 ANÁLISIS DE LAS ACTIVIDADES REALIZADAS	28
5.4 ANÁLISIS DE LA METODOLOGÍA UTILIZADA	29
6. CONCLUSIONES Y RECOMENDACIONES	30
7. REFERENCIAS BIBLIOGRÁFICAS	31

ÍNDICE DE FIGURAS

2.1 Sede de Fiducia & GAD IT.	9
4.1 Arquitectura	17
4.2 Código del repositorio de GitHub.	19
4.3 Spring Boot Server.	20
4.4 Spring Boot Server Diagrama de Clases.	21
4.5 Share A Share.	22
4.6 OwlView.	23
4.7 Shake To Share	24
4.8 Front End	25

1. INTRODUCCIÓN

Hoy en día las tecnologías de la información avanzan de manera acelerada y progresivamente van facilitando más y más las comunicaciones e interacciones para el ser humano. Las empresas alrededor del mundo se encuentran en cambios constantes para poder brindar las tecnologías que el mundo digital brinda.

El presente trabajo explica de manera detallada el proceso de desarrollo de un proyecto para una empresa relacionada con el campo de tecnologías de la información para la realización de prácticas profesionales, las cuales son necesarias para acreditar dentro del currículo de la carrera Ingeniería en Sistemas de Información de la Universidad de Sonora.

Para cumplir ese requisito se contactó a Aiesec, una organización internacional encargada de poner en contacto practicantes y empresas de todo el mundo, fue a través de su página en internet que se contactó a la empresa Fiducia & GAD IT AG la cual tiene sede en Karlsruhe, Alemania. Tras una serie de entrevistas respondió positivamente a la solicitud y se decidió realizar las prácticas en dicha empresa.

El proyecto y las funciones se desarrollaron entre los meses de octubre del año 2016 y abril del año 2017, con los propósitos de cumplir con ese requisito curricular, y lograr completar los estudios concernientes a la carrera ingeniería en sistemas de información de la universidad de sonora.

1.1 EXPLICACIÓN DEL PROYECTO

La empresa actualmente cuenta con más de 25 edificios de oficina alrededor de todo el país de Alemania, cada edificio cuenta con incontables oficinas en donde los trabajadores desempeñan sus labores diarias. Mayormente de 3 a 5 trabajadores se encuentran dentro de cada oficina y no es muy común las rotaciones de oficinas entre trabajadores.

Como se explicó en la sección anterior, la empresa posee muy grandes dimensiones y los trabajadores se encuentran principalmente aislados en sus ambientes de trabajo, esto presenta una gran problemática para la empresa debido a la baja posibilidad de interacción que existe entre los diferentes empleados.

Por otro lado, la empresa constantemente se encuentra financiando los viajes de los trabajadores desde su hogar hasta las respectivas ramas de la empresa, esto presenta un gran gasto anual por parte de la compañía. La sociedad alemana se encuentra en ámbito muy individualista en el cual las relaciones interpersonales no son muy comunes en la vida cotidiana. La compañía tiene como objetivo el lograr reducir el gasto anual presente en transporte reduciendo este límite social que existe entre los trabajadores y logrando que 2 o más empleados compartan el mismo medio de transporte.

Por razón de las grandes dimensiones de la empresa, esta tiene como objetivo el crear o reforzar relaciones intralaborales, extralaborales y brindar una manera de mantener comunicados a sus empleados al momento de tomar algún medio de transporte en común. Después de un largo análisis y consultoría, Fiducia GAD logró llegar a la conclusión que es posible lograr su objetivo implementando las nuevas tecnologías de la información que el mundo digital brinda.

La implementación de una aplicación móvil que pueda brindar a sus trabajadores la posibilidad de compartir eventos concernientes a la compañía, o bien, eventos concernientes a las vidas personales de los trabajadores fue la principal idea que surgió tras el largo análisis. Es a través de esta aplicación que la empresa puede lograr conectar la vida social de sus trabajadores, así como lograr que un mayor número de trabajadores tomen el mismo transporte hacia las respectivas compañías logrando así reducir el gran gasto que tiene la compañía en medios de transporte.

1.2 OBJETIVOS

El objetivo general del proyecto es lograr implementar una estrategia eficaz para lograr aumentar las relaciones intralaborales, extralaborales y reducir el gasto

general que se tiene por los empleados al tomar transporte con destino a la empresa.

Se pueden presentar los objetivos específicos del proyecto a través de los siguientes puntos:

- Analizar las circunstancias actuales de la empresa y propuesta de la manera más eficaz de atacar el problema en cuestión.
- Diseñar y desarrollar una aplicación móvil multiplataforma.
- Analizar y seleccionar las mejores tecnologías disponibles para lograr desarrollar el proyecto.
- Implementar una metodología ágil de desarrollo.
- Reducir la cantidad del gasto anual en transporte anual en un 250%.
- Aumentar la frecuencia con la que los trabajadores de la empresa toman transporte público hacia la empresa.
- Aumentar de forma significativa las relaciones intralaborales y extralaborales en los trabajadores de la empresa.

1.3 METODOLOGÍA

A través de una serie de largas discusiones a principios de la estadía, se llegó a la conclusión de utilizar SCRUM como metodología ágil para el desarrollo de software. Se implementaron los sprint necesarios hasta la satisfactoria finalización del proyecto. A continuación, se pasará a una breve descripción de las iteraciones realizadas.

En el primer sprint se definieron los principios de la aplicación, parte del producto Backlog y el diseño general de la aplicación, formados así los principios básicos de la aplicación. Se experimentaron algunos problemas en la definición del contenido de la aplicación, pero después de una serie de juntas se logró llegar a una firme conclusión.

En el segundo sprint se tomaron en cuenta los cambios realizados en el sprint anterior, la migración del proyecto de la plataforma de Ionic Creator a un entorno de

codificación manual fue la principal actividad desempeñada en este sprint, a su vez se avanzó en la codificación del front end de la aplicación.

En el tercer Sprint se centró en la codificación del servidor de Sprint Boot que soporta nuestra API RESTful para el manejo de la información en la base de datos, se dividió la actividad en diferentes tareas, creación de los modelos de la base de datos, codificación de los servicios necesarios y se desplegó el proyecto a través de servidores soportados por Bluemix IBM.

En el cuarto sprint se implementó la funcionalidad de la característica “Share A Share”, brindando al usuario la posibilidad de compartir eventos con otros usuarios, siendo esta una de las principales características de la aplicación en general.

En el quinto sprint se desarrolló la característica “OwlView”, con la cual, el usuario es capaz de localizar los usuarios que se encuentran cercanos a el mismo, utilizando diversos sensores y la cámara del dispositivo. A su vez se implementó la codificación básica de la característica “Basics for Shake to Share”.

En el sexto sprint se continuó con el desarrollo de la característica previamente explicada, pero en esta ocasión, se codificó el back end de la aplicación, brindando al usuario la posibilidad de conectarse con 2 o más usuarios al momento de sacudir sus dispositivos, a su vez, gran parte del front end fue desarrollado.

En el séptimo sprint se llevaron a cabo las preparaciones necesarias para presentar el proyecto ante los CEO's de la compañía, llevándose a cabo diversas juntas.

2. DESCRIPCION DEL CONTEXTO

Fiducia IT AG era un proveedor alemán de servicios de TI con sede en Karlsruhe. En 2015 Fiducia IT AG se unió con GAD eG a Fiducia & GAD IT AG. La compañía es la más grande de su tipo en la asociación financiera cooperativa (FinanzGruppe). Su actividad principal son los servicios de TI financieros para los bancos cooperativos. Sin embargo, entre los clientes de Fiducia también están los bancos del sector de la banca privada, las instituciones públicas y las empresas libres. En el año financiero 2015, Fiducia alcanzó una facturación de 708 millones de euros y operó su software bancario para 750 bancos.

Acuerdo es una de las soluciones de software estandarizadas ofrecidas por Fiducia IT AG. Es una solución integral de software bancario integral que ofrece múltiples clientes. Desde el 25 de junio de 2007, acuerdo es utilizado por todos los bancos cooperativos apoyados por Fiducia. Por lo tanto, es el sistema bancario más extendido en Alemania.

"Fiducia" es latín y significa "confianza" y "fiabilidad".

2.1 EQUIPAMIENTO E INSTALACIONES

La sede de Fiducia está situada en Karlsruhe con una sucursal en Aschheim, Munich. Basado en el número de empleados, Fiducia es el tercer mayor empleador del sector privado en Karlsruhe, después de Siemens y EnBW.

La empresa fue fundada en 1924 bajo el nombre de Fiducia Contabilidad y Auditoría Instituto AG en Karlsruhe. En 1958 Fiducia llegó a los bancos supervisados con la idea de una fusión en una comunidad de reserva. Fiducia se hizo cargo de la organización como un nuevo negocio. Durante los años siguientes, se utilizaron principalmente tarjetas perforadas. A lo largo de los años sesenta se produjo una

transición gradual a otros medios, tales como tiras de diario, cintas magnéticas y discos magnéticos. En la figura 2.1 se muestra una foto del edificio de la empresa.



Figura 2.1 – Sede de Fiducia & GAD IT.

Actualmente Fiducia opera dos centros de datos activos y totalmente redundantes con más de 8.700 servidores Unix en el área de Karlsruhe. Los centros de datos están equipados con un sistema de energía de emergencia y se ejecutan de forma totalmente independiente.

2.2 REGLAS DE OPERACIÓN DE LA UNIDAD RECEPTORA

La compañía implementa una estructura de operación jerárquica para los diferentes roles entre los trabajadores. Debido a sus grandes dimensiones, esta se divide en una serie de departamentos, cada uno de los cuales, en ocasiones, posee cientos de trabajadores. Debido a razones de seguridad es imposible mostrar un diagrama de la estructura de la empresa en este documento.

La empresa cuenta con 3 principales ramas, sumando un total de 20 edificios a lo largo de todo Alemania y más de 5000 empleados. Fiducia afirma tener una de las mayores redes MPLS para comunicarse con los sistemas locales de sus bancos asociados. En la figura 2.1 se muestra las características que presenta la red MPLS que posee la compañía.

Transmission Speeds	128 kbit/s to 600 Mbit/s (on average 7.4 Mbit/s)
Network Components	8,350 Routers, 9,000 Switches
Network Security	154 Firewalls, 46 Proxies und Virus Scanners
agree Voice (VoIP-Telephony)	15,100 Ports
Internet Customers	Internet- and VPN-access for about 3.8 Million Online Customers

Figura 2.1 – Detalles de la red MPLS.

La firma posee un centro de datos que comprende alrededor de 6.720 metros cuadrados de espacio para equipos informáticos. En la primera fase, alrededor del 60% será ocupado. El nuevo centro de datos se encuentra a unos 10 km del centro de datos de respaldo.

2.3 ENTORNO DONDE SE UBICA LA UNIDAD RECEPTORA.

La unidad de ubica en la ciudad de Karlsruhe perteneciente al estado de Baden-Wurtemberg, Alemania. La actividad principal del grupo son los servicios de TI para Volksbanken y Raiffeisenbanken, así como para otras empresas e instituciones centrales del grupo financiero cooperativo.

Otros clientes son los bancos privados y especiales, así como varias otras empresas comerciales. Fiducia & GAD IT AG ofrece a sus clientes todos los servicios de TI en sus propios centros de datos y, por lo tanto, exclusivamente en

una nube privada. Además de trabajar como un centro de datos y un proveedor de servicios de TI, el desarrollo y la provisión de software específico del banco, así como los servicios de consultoría correspondientes, es una tarea central de la empresa.

2.4 NORMATIVIDAD DE LA UNIDAD RECEPTORA

Fiducia IT& GAD es una gran compañía alemana que guía sus labores por un conjunto de reglas, lineamientos, políticas, manuales y procedimientos, los cuales son definidos internamente por sus propios medios o en acuerdo con sus principales clientes.

La compañía cuenta con una estructura de trabajo muy jerárquica, en la cual los empleados laboran con diferentes roles, teniendo esto diferente valor autoritario dependiendo de la situación. Es dividida en una serie de grandes departamentos, en los cuales se puede encontrar cientos de trabajadores, en ocasiones, estos ocupan edificios enteros.

Cuenta con una de las más grandes y seguras redes internas privadas de Alemania, en la cual se comunican las diferentes ramas de la empresa situadas en diferentes ciudades a lo largo de todo el país.

3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS.

A continuación, se explican algunos de los principales conceptos relacionados con el proyecto.

Ionic: Ionic es un completo SDK de código abierto para el desarrollo de aplicaciones móviles híbridas. Construido sobre AngularJS y Apache Cordova, Ionic proporciona herramientas y servicios para desarrollar aplicaciones móviles híbridas usando tecnologías Web como CSS, HTML5 y Sass. Las aplicaciones se pueden construir con estas tecnologías Web y luego se distribuyen a través de tiendas de aplicaciones nativas para que se instalen en dispositivos aprovechando Cordova. Ionic fue creado por Max Lynch, Ben Sperry, y Adam Bradley de Drifty Co. en 2013. Ionic Creator es una herramienta de construcción de interfaz de arrastrar y soltar. App Builder para Ionic también es proporcionado por Appery.io.

Cordova: Apache Cordova (anteriormente PhoneGap) es un popular entorno de desarrollo de aplicaciones móviles, originalmente creado por Nitobi. Adobe compró Nitobi en 2011, le cambió el nombre a PhoneGap, y más tarde liberó una versión de código abierto del software llamado Apache Cordova. Apache Cordova permite, a los programadores de software, construir aplicaciones para dispositivos móviles utilizando CSS3, HTML5, y Javascript en vez de utilizar APIs específicas de cada plataforma como Android, iOS, o Windows Phone. Permite encapsular CSS, HTML, y código de Javascript dependiendo de la plataforma del dispositivo. Extiende las características de HTML y Javascript para trabajar con el dispositivo. Las aplicaciones resultantes son híbridas, lo que significa que no son ni una aplicación móvil nativa (porque toda la representación gráfica se realiza vía vistas de Web en vez del framework nativo) ni puramente basadas en web (porque no son solo aplicaciones web, sino que están empaquetadas como aplicaciones para su distribución y tienen acceso a las APIs nativas del dispositivo).

AngularJS: AngularJS es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticos o dinámicos.

AngularJS se puede combinar con el entorno en tiempo de ejecución Node.js, el framework para servidor Express.js y la base de datos MongoDB para formar el conjunto MEAN.

IBM Bluemix: IBM Bluemix es un entorno de plataforma como servicio desarrollado por IBM. Soporta varios lenguajes de programación y servicios así como la metodología de desarrollo DevOps de forma integrada para crear, ejecutar, desplegar y gestionar aplicaciones en la nube. Bluemix está basado en la tecnología abierta de Cloud Foundry y corre sobre la infraestructura de SoftLayer. Bluemix soporta Java, Node.js, Go, PHP, Python, Ruby Sinatra, Ruby on Rails, GeneXus y puede ser extendido a otros lenguajes tales como Scala mediante el uso de 'buildpacks'.

Bluemix fue desarrollado por un equipo localizado en diversas partes del mundo, tardando solo 18 meses desde el concepto inicial hasta llegar a estar disponible al público. Se anunció como beta pública en febrero de 2014 y fue abierto al público en junio. En el momento de su anuncio, Bluemix era uno de los despliegues de Cloud Foundry más grandes del mundo.

IBM CloudAnt: Cloudant es un producto de software de IBM, que se entrega principalmente como un servicio basado en la nube. Cloudant es un servicio de base de datos distribuido y no relacional de código abierto del mismo nombre. Cloudant

se basa en el proyecto CouchDB respaldado por Apache y en el proyecto BigCouch de código abierto.

El servicio de Cloudant ofrece un motor integrado de administración, búsqueda y análisis de datos diseñado para aplicaciones web. Cloudant escala las bases de datos en el marco CouchDB y proporciona alojamiento, herramientas administrativas, análisis y soporte comercial para CouchDB y BigCouch. El servicio CouchDB distribuido de Cloudant se utiliza de la misma manera que CouchDB independiente, con la ventaja añadida de que los datos se distribuyen de forma redundante en varias máquinas.

PouchDB: PouchDB es una base de datos JavaScript de código abierto inspirada en Apache CouchDB diseñada para funcionar bien dentro del navegador.

PouchDB fue creado para ayudar a los desarrolladores web a crear aplicaciones que funcionen tanto sin conexión como en línea.

Permite que las aplicaciones almacenen los datos localmente mientras están fuera de línea, luego sincronizan con CouchDB y servidores compatibles cuando la aplicación vuelve a estar en línea, manteniendo los datos del usuario sincronizados sin importar dónde se conectan.

Visual studio code: Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control Git incorporado, resaltado de sintaxis, terminación de código inteligente, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, atajos de teclado y preferencias. Es de código abierto, aunque la descarga oficial está bajo una licencia patentada.

Spring Framework: Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

La primera versión fue escrita por Rod Johnson, quien lo lanzó junto a la publicación de su libro Expert One-on-One J2EE Design and Development (Wrox Press, octubre 2002). El framework fue lanzado inicialmente bajo la licencia Apache 2.0 en junio de 2003. El primer gran lanzamiento fue la versión 1.0, que apareció en marzo de 2004

y fue seguida por otros hitos en septiembre de 2004 y marzo de 2005. La versión 1.2.6 de Spring Framework obtuvo reconocimientos Jolt Awards y Jax Innovation Awards en 2006. Spring Framework 2.0 fue lanzada en 2006, la versión 2.5 en noviembre de 2007, Spring 3.0 en diciembre de 2009, y Spring 3.1 dos años más tarde. El inicio del desarrollo de la versión 4.0 fue anunciado en enero de 2013. La versión actual es 4.3.7.

4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES.

A continuación, se pasará a describir paso a paso las actividades realizadas a lo largo de toda la estadía de prácticas en la empresa Fiducia GAD.

4.1 ACTIVIDADES PRELIMINARES

La primera semana de actividades en la empresa fue destinada a recibir diferentes tutoriales concernientes a parte de las tecnologías que usamos a lo largo de todo el proyecto. Dichos tutoriales fueron impartidos por personal de la empresa altamente experimentado el cual su principal propósito fue instruirnos lo necesario y brindar apoyo en un futuro si es necesario. Recibimos instrucción en las tecnologías de: Ionic, Java Spring, Java Spring Boot y Swagger. Como resultado, obtuvimos conocimientos básicos en la implementación y configuración de las tecnologías y socializamos con los trabajadores que serían nuestros tutores por el resto de la estadía.

Al finalizar nuestra semana de tutorías nos vimos con la tarea de reforzar los conocimientos adquiridos y comenzar a desempeñar distintos ejercicios por nuestra propia cuenta. Fue durante este proceso que logramos comprender mejor la tecnología y prestamos atención a los pequeños detalles que fueron omitidos en la previa semana. Una vez más recibimos ayuda de nuestros tutores para aclarar dudas y lograr así un mejor entendimiento de las tecnologías.

Para ser más específico, en Ionic, realizamos una aplicación de ejemplo utilizando el portal *Ionic Creator* el cual nos facilitó el diseño general de la aplicación. En este tutorial aprendimos las diferentes características y los diferentes componentes que ofrece *Ionic Framework*. Seguido, sobre Spring Boot, comprendimos cuál es la función del framework realizando un RESTful API de ejemplo, estudiamos el ciclo de vida de Spring Framework e integramos la tecnología con swagger para realizar fáciles pruebas de nuestra API.

Después de nuestra semana de integración con las diferentes tecnologías, se dio inicio a los tutoriales relacionados con el proceso de desarrollo que utilizaremos en nuestra aplicación, en este caso, recibimos diversos tutoriales sobre SCRUM como metodología ágil para el desarrollo. A su vez nos enseñaron a utilizar JIRA, la herramienta web que utilizamos para llevar seguimiento de SCRUM, en este portal se pueden definir los componentes necesarios para llevar a cabo dicho proyecto, tales como, Sprints, historia de usuario, BackLog, etc.

Una vez que tuvimos definidos los conocimientos básicos para comenzar nuestro proyecto, nos vimos con la tarea de definir los requerimientos del mismo, se realizaron varias sesiones de lluvia de ideas y conceptos para definir los principios de la aplicación, a su vez, se terminó el denominado *White Paper* el cual contiene los principios que la aplicación desempeñará. Una vez definidos los principios de la aplicación y utilizando los conocimientos previamente adquiridos sobre SCRUM se realizó la definición del Product Backlog de nuestro proyecto.

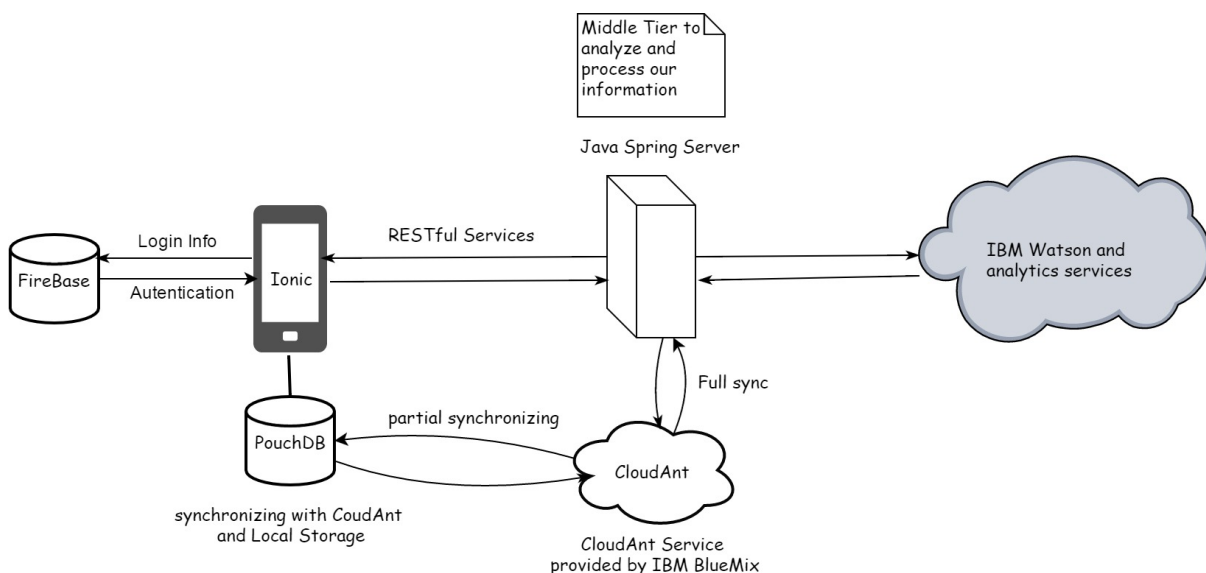


Figura 4.1 - Arquitectura.

Tras las diversas lluvias de ideas realizadas se logró definir la arquitectura general del proyecto la figura 4.1 muestra el diagrama de dicha arquitectura, en resumen, se utilizó ionic como la aplicación de lado cliente, Spring Boot server como un mediador

entre el cliente y la base de datos, para tener un total control de las entradas y salidas de la base de datos y poder desarrollar servicios RESTful. CloudAnt como principal manejador de base de datos y la librería PouchDb para mantener una conexión en tiempo real síncrona con el servidor y la base de datos.

Con la ayuda de nuestro diseñador se llevaron a cabo Mock Ups generales de la aplicación y tuvimos varias sesiones de feedback por parte de nuestros tutores, esto con el propósito de lograr obtener la mejor experiencia de usuarios, y lograr definir el proyecto antes de comenzar con la codificación.

4.2 SPRINT 1

Definidos los principios de la aplicación, parte del product Backlog y el diseño general de la aplicación, se comenzó con el primer Sprint, pero, tan solo días después de comenzar, se volvió al estatus anterior y se llevó a revisión el producto Backlog y los principios generales de la aplicación. Tras diversas lluvias de ideas con nuestro product owner, cambiamos parte del contenido del white paper, es decir, los principios del proyecto, y cambiamos parte del diseño general de la aplicación (MockUps).

Tras la interrupción, logramos terminar el primer sprint y entregar el resultado a como se tenía previsto, es decir, la versión previa antes de los cambios. Los cambios se tomaron en cuenta a partir del siguiente sprint.

4.3 SPRINT 2.

Se comenzaron a tomar en cuenta los cambios hechos en el pasado sprint. Como labor principal en este sprint, fue la migración del proyecto desde la plataforma de ionic framework a un ambiente de codificación manual, utilizando Visual Studio Code como principal IDE. Se codificaron diversas páginas del front end y se avanzó en la estructura general.

```
1 angular.module('app.controllers', [])
2
3
4 /**
5  * Concerning Controller for the side menu.
6  * encapsulate all logic for the side menu here.
7  */
8
9 .controller('menuCtrl', ['$scope', '$stateParams', '$state', 'FireB',
10 function($scope, $stateParams, $state, FireB) {
11
12     // $scope.userData = $ionicUser.details;
13     // $scope.logout = function () {
14
15         FireB.auth().signOut().then(function () {
16             $state.go('login');
17         }, function (error) {
18             alert(error.message);
19         });
20     });
21
22 });
23
24 /**
25  * Concerning Controller for the login page.
26  */
27
28 .controller('loginCtrl', ['$scope', '$stateParams', '$state', 'FireB', 'RandomUser', '$http',
29 function($scope, $stateParams, $state, FireB, RandomUser, $http) {
30     console.log("Login Controller Started");
31 }
```

Figura 4.2 - Código del repositorio de GitHub.

En la figura 4.2 se puede apreciar nuestro el nuestro repositorio de GitHub al cual subimos el código de nuestra aplicación, en los siguientes meses este pasaría a ser nuestra principal manera de tener un control de versiones de nuestro código.

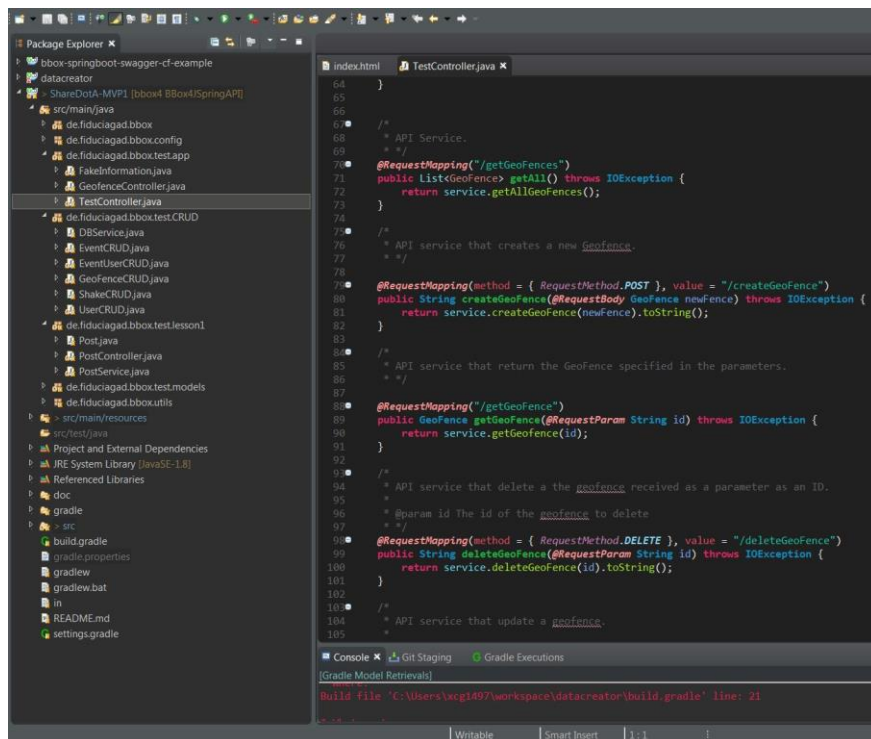
4.4 SPRINT 3

En el tercer sprint, se comenzó con la codificación de nuestro spring boot server, el cual servirá como mediador entre la base de datos y nuestra aplicación cliente. A continuación se pasará a describir las actividades realizadas en este sprint.

Primeramente se creó y configuró el proyecto en spring boot, a su vez se implementó un servicio por parte de IBM para desplegar el proyecto, este servicio tiene acceso al internet público, por lo que, podemos acceder a nuestra API desde cualquier lugar del mundo.

Tras la configuración del servidor se comenzó con la codificación de los modelos de la base de datos en nuestro Spring Boot. Tomando la estructura de nuestros JSON

se creó la representación de estas clases en java. El objetivo detrás de esto es tener un modelo el cual spring boot pueda reconocer e interpretar.



```
64     }
65
66
67     /**
68      * API Service.
69      */
70     @RequestMapping("/getGeoFences")
71     public List<GeoFence> getAll() throws IOException {
72         return service.getAllGeoFences();
73     }
74
75     /**
76      * API service that creates a new GeoFence.
77      */
78
79     @RequestMapping(method = { RequestMethod.POST }, value = "/createGeoFence")
80     public String createGeoFence(@RequestBody GeoFence newFence) throws IOException {
81         return service.createGeoFence(newFence).toString();
82     }
83
84     /**
85      * API service that return the GeoFence specified in the parameters.
86      */
87
88     @RequestMapping("/getGeoFence")
89     public GeoFence getGeoFence(@RequestParam String id) throws IOException {
90         return service.getGeoFence(id);
91     }
92
93     /**
94      * API service that delete a the GeoFence received as a parameter as an ID.
95      */
96     @param id The id of the GeoFence to delete
97     /**
98     @RequestMapping(method = { RequestMethod.DELETE }, value = "/deleteGeoFence")
99     public String deleteGeoFence(@RequestParam String id) throws IOException {
100         return service.deleteGeoFence(id).toString();
101     }
102
103     /**
104      * API service that update a GeoFence.
105      */
```

Figura 4.3 - Spring Boot Server.

La figura 4.3 muestra una sección del código de nuestro Spring Boot server en el cual se implementaron CRUDs básicos para los archivos relevantes en la base de datos con el propósito de poder realizar las operaciones básicas necesarias en cada base de datos. Manteniendo una programación orientada a objetos y las mejores prácticas, se encapsularon los CRUDs, cada uno, en diferentes clases, logrando así tener un orden el código.

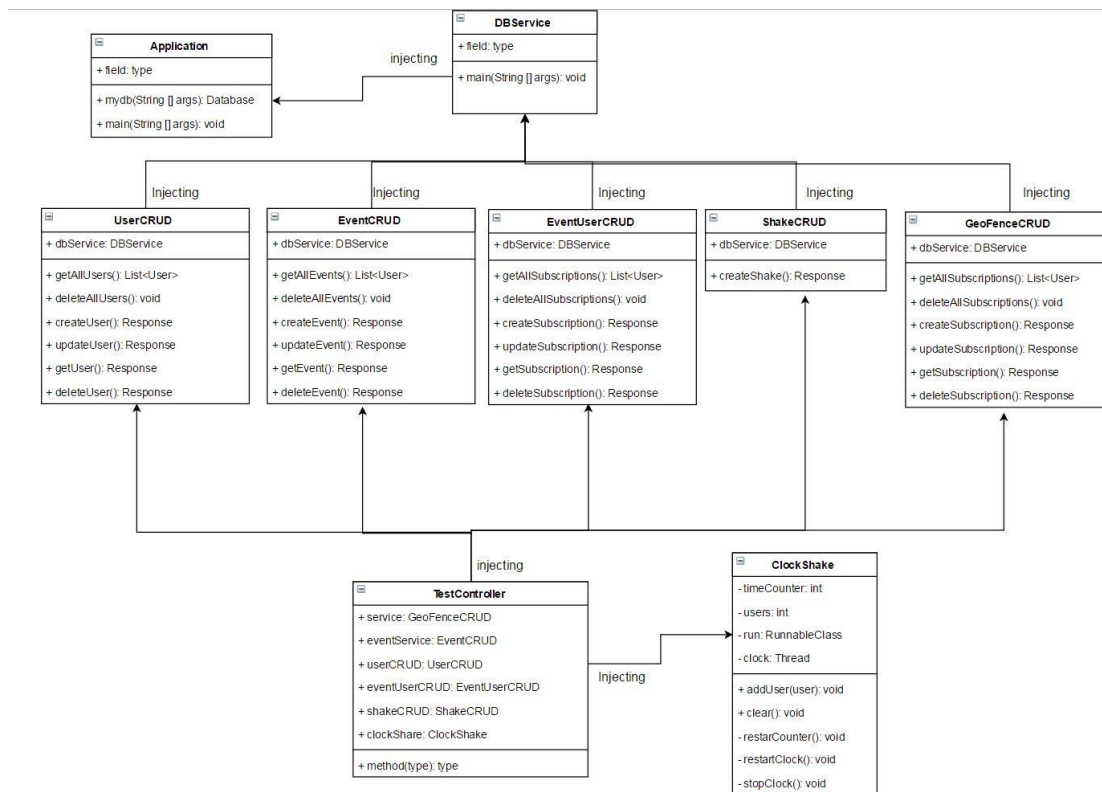


Figura 4.4 - Spring Boot Server Diagrama de Clases.

La figura 4.4 muestra el diagrama de clases que se codificó y desplegó en el servidor de spring.

4.5 SPRINT 4

Siendo una de las características más importantes en la aplicación, en el cuarto sprint, se implementó la característica “Share A Share”, la cual brinda al usuario la habilidad de compartir eventos en tiempo real con cualquier usuario que tenga la aplicación y se encuentra en un área cercana al usuario.

Para el desarrollo de dicha característica se utilizó principalmente la librería PouchDB para conectar los usuarios a través de la implementación de web sockets que la librería ya posee. En una descripción más detallada, si el usuario acepta un evento, el resto de los usuarios que estén en un área cercana serán capaz de ver su companero que aceptó el mismo evento e igualmente unirse si ellos lo consideran. Esta implementación utiliza el sensor gps del teléfono al igual uno de los pilares de la aplicación “geofences”.

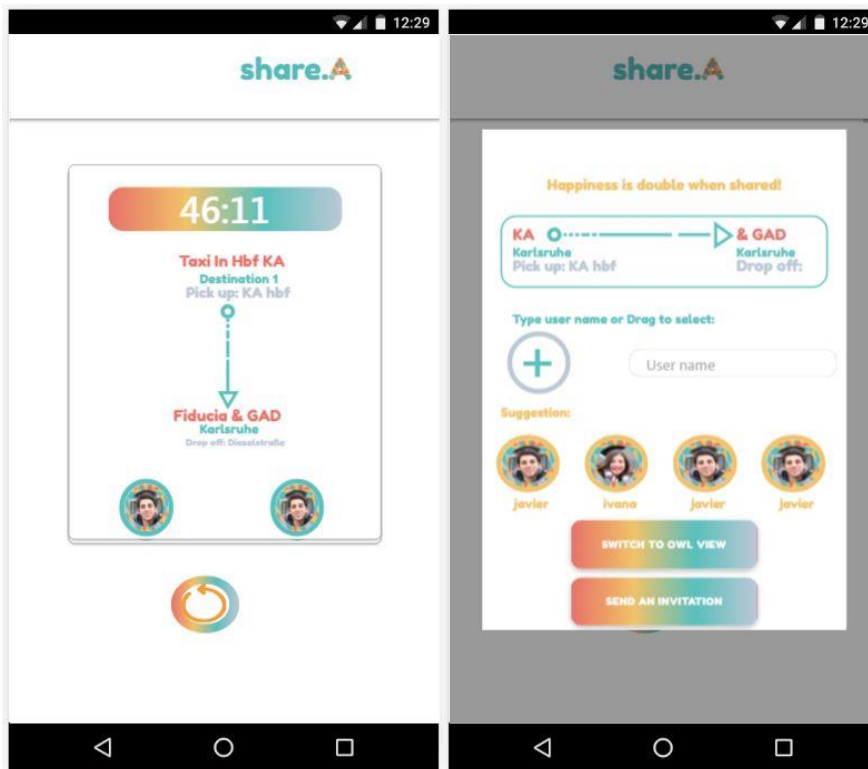


Figura 4.5 - Share A Share.

La figura 4.5 muestra el flow de la característica “Share A Share”, el usuario es capaz de aceptar un evento deslizando las cartas hacia el lado derecho, inmediatamente un pop up describiendo los detalles del evento es lanzado, en este el usuario es capaz de invitar a nuevos usuarios a asistir al mismo evento, existen 2 diferentes maneras de invitar, una es por medio de OwlView y la otra desplazando los iconos de los usuarios hacia la marca correspondiente.

4.6 SPRINT 5

En el quinto Sprint, se implementó el concepto básico de la característica “OwlView” la cual muestra los usuarios que interactúan con la aplicación en distancias cercanas.

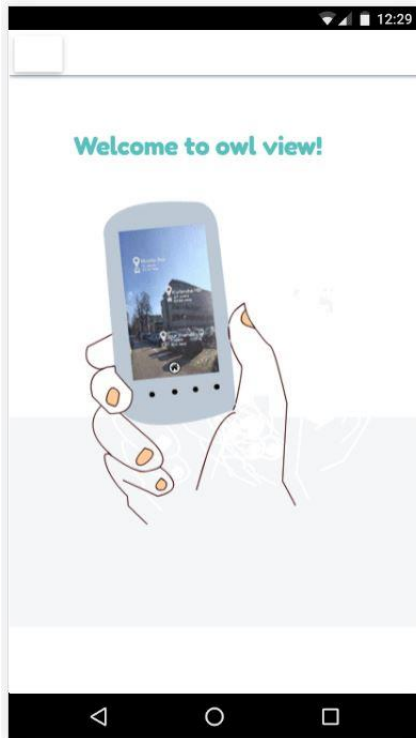


Figura 4.6 - OwlView.

La figura 4.6 muestra la pantalla en donde se desplegó la funcionalidad de “OwlView”, para desarrollarlo se procesó el sensor compás del dispositivo, de esta manera se pudo tomar el ángulo que éste tenía con respecto al norte geográfico de la tierra. Tomando este ángulo se muestra el número de personas cercanas a ti que se encuentren en dicho ángulo, por ejemplo, muestra el número de personas que se encuentren en la dirección de 20 a 50 grados.

El objetivo detrás de dicha característica es poder tener un fácil acceso al resto de los usuarios que interactúan con la aplicación, y brindar la posibilidad de invitar o chatear con los usuarios que se encuentren en un determinado rango de distancia.

Por otro lado, se desarrolló la tarea “Basics for Shake to Share” la cual consiste en leer los sensores del teléfono celular y detectar cuando el usuario lo sacude. Una básica implementación de la característica que posteriormente sería mayormente desarrollada.

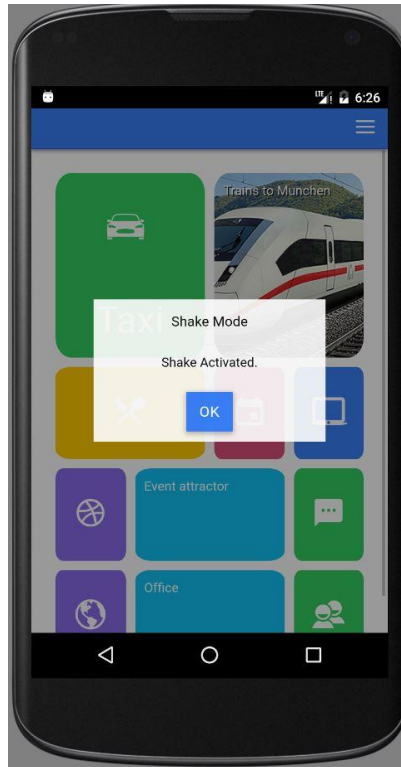


Figura 4.7 - Shake To Share.

En la figura 4.7 se puede apreciar que se le muestra al usuario una vez esté sacude el teléfono.

Se arreglaron diversos bugs encontrados en la característica previamente desarrollada “shake to share” con el propósito de tener una interacción más sólida con el usuario al momento de aceptar el evento.

4.7 SPRINT 6

Sexto Sprint, continuación con el desarrollo de la característica “Shake to share”, en esta ocasión, con la implementación de la funcionalidad del backend, brindando así, la posibilidad a 2 o más usuarios de sacudir sus dispositivos y recibir un mensaje de alerta en ambos dispositivos.

Dicha implementación se desarrolló en nuestro spring boot server. El usuario realiza una petición al servidor cada vez que esté sacude su teléfono y el servidor conecta a ambos usuarios y manda a estos el mismo mensaje.

Se mejoró gran parte del front end de la aplicación, mejorando así la experiencia de usuario.

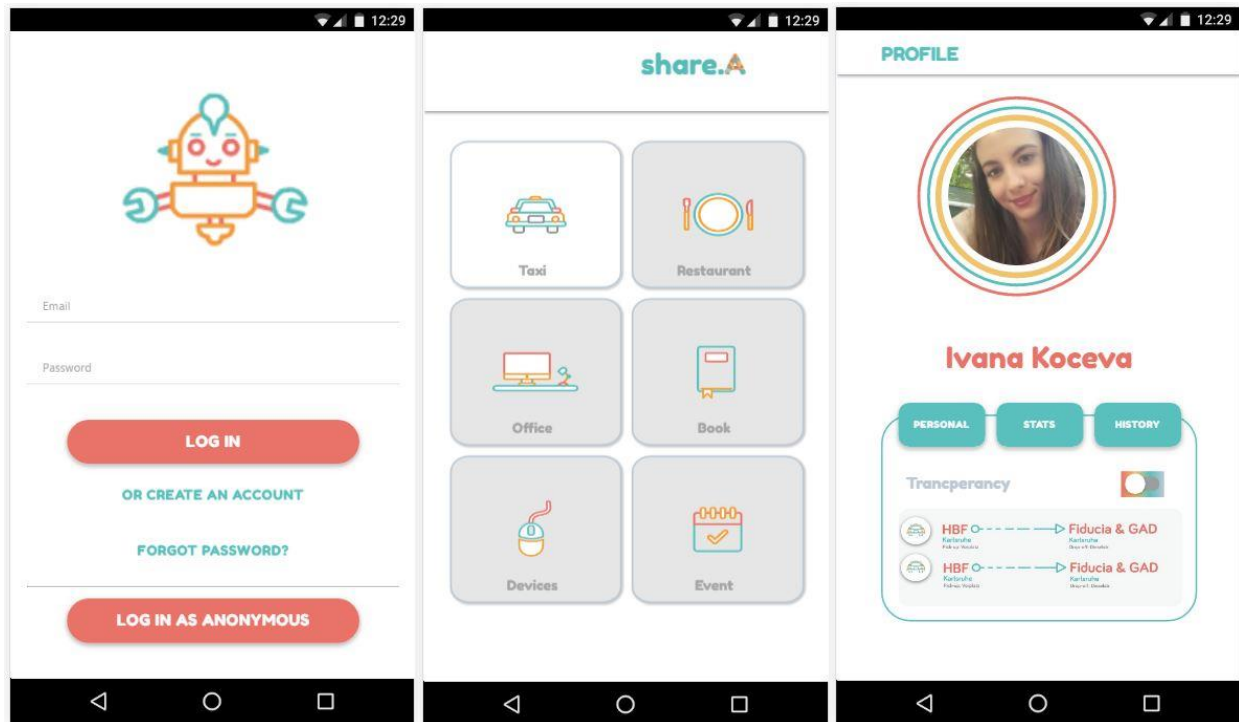


Figura 4.8 - Front End.

La figura 4.8 muestra el nuevo diseño de algunas páginas de la aplicación, se mejoró el login de la aplicación dándole un diseño más amigable al usuario, seguido al iniciar sesión se despliega el menú principal, el cual a su vez se le dio una mejora. La última página muestra los perfiles de usuario de la aplicación, se le agregaron menús de selección y mejores en la imagen de perfil.

4.8 SPRINT 7

Séptimo y último sprint, se continuaron con las mejoras del front end de la aplicación para brindar al usuario la mejor experiencia posible. Se llevaron a cabo diversas juntas para definir el contenido de nuestra presentación final, en donde presentamos a los CEO's de la compañía el proyecto que desarrollamos durante los 6 meses anteriores.

Debido a la magnitud de la compañía, esta posee 6 diferentes CEO's pero debido a falta de tiempo y la complejidad de sincronizar la agenda de estos 6 hombres de negocios, solo tuvimos la oportunidad de presentar nuestro producto final a 2 CEO's.

Tuvimos reuniones semanales para planear el contenido de la presentación final, así como la definición del guion que diríamos y la presentación de PowerPoint que presentamos. La presentación se llevó a cabo y fue un éxito, recibiendo así el visto bueno por parte de los CEO's de la compañía.

5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA

En esta sección se presentará la valoración del aprendizaje y lecciones aprendidas durante la estancia profesional en la empresa Fiducia IT & GAD tanto en términos del proyecto en general, los objetivos que se plantearon para su realización, las actividades desarrolladas y la metodología que se utilizó.

5.1 ANÁLISIS GENERAL DEL PROYECTO

La empresa cuenta con más de 5000 empleados a lo largo de todo Alemania, con dicho tamaño es difícil mantener un seguimiento de las relaciones sociales que suceden entre los diferentes empleados a lo largo de todo el país, la empresa está dispuesta a tomar en cuenta las relaciones laborales y no laborales que suceden con sus empleados, con el objetivo de poder brindar una experiencia más sociable y un ambiente más agradable para sus trabajadores. Además de las relaciones sociales, la organización brinda la posibilidad a sus trabajadores de pagar sus medios de transportes, si estos tienen como destino alguno de las diferentes ramas de la empresa que existen a lo largo de todo el país.

En general el proyecto desarrollado es muy visionario y se logró aprender bastante en el transcurso de las actividades, al ser un proyecto tan grande, aconsejaría más inversión por parte de la empresa y la implementación de una mayor cantidad de características/módulos a la aplicación.

5.2 ANÁLISIS DE LOS OBJETIVOS DE LAS PRÁCTICAS

El resultado obtenido fue el desarrollo e implementación de una completa idea, desde la fase concepcional hasta la fase de desarrollo, se llevó a cabo una total arquitectura de software con el principal propósito de brindar al usuario final una aplicación móvil basada en un framework multiplataforma.

Se logró cumplir con las metas que la compañía definió para el equipo desde el principio del proceso. Se vio una fuerte asimilación por parte de los empleado de la

compañía a la nueva aplicación, y se esperaran resultados positivos para los próximos meses.

La compañía tenía como objetivo el desarrollo profesional de los practicantes, por nuestra parte logramos obtener la experiencia profesional de trabajar para una gran compañía y aprendimos nuevas tecnologías con amplio valor profesional. Se vio un amplio desarrollo en los diseñadores y programadores del equipo por igual.

La aplicación desarrollada tiene como objetivo adaptarse a las necesidades del usuario, el cual debe tener control sobre el contenido y poder ser capaz de socializar en una manera innovativa, a como no se vio antes en la compañía.

5.3 ANÁLISIS DE LAS ACTIVIDADES REALIZADAS

Las actividades realizadas en la estadía profesional fueron bastante diversas y complementaron satisfactoriamente los objetivos planteados al principio del proyecto. a través del periodo se realizó un workshop para enseñarnos el funcionamiento de la tecnología Ionic Framework, la cual utilizaríamos a lo largo de todo el proyecto. Considero el desarrollo de este workshop muy importante e interesante en mi desarrollo profesional.

Tutoriales sobre Spring Boot Framework se llevaron a cabo al principio del periodo y se implementó esta tecnología a lo largo de todo el proyecto, considero a esta muy valiosa en mi desarrollo y con un amplio rango de acción en el mundo del desarrollo de software. A lo largo del proyecto Bases de datos no SQL fueron muy utilizadas, anteriormente no había tenido el privilegio de trabajar con alguna de estas tecnologías e igualmente considero a estas muy importantes para mi desarrollo profesional.

Además de las diversas tecnologías aprendidas, obtuve la experiencia de realizar diversas presentaciones con diferentes departamentos de la compañía a lo largo de todo el periodo, dándome la oportunidad de conocer a personas importantes y diferentes puntos de vista dentro de la misma empresa.

5.4 ANÁLISIS DE LA METODOLOGÍA UTILIZADA

La mecánica de trabajo aprendida y tomada fue la metodología ágil de desarrollo de software SCRUM. Primeramente, se creó el denominado “Product BackLog” dividiéndose así el proyecto completo en tareas asignando así diferentes puntos de complejidad a estas. Tomando en cuenta la complejidad de las tareas, se crearon diferentes ciclos de trabajo denominados Sprint, en donde cada uno de los miembros del equipo tuvo la oportunidad de elegir las actividades que consideraba posibles para el periodo de 2 semanas.

Diariamente se llevaron a cabo juntas para conocer el estatus de las tareas siendo desarrolladas por cada uno de los miembros del equipo, con el objetivo de que cada miembro supiera las actividades que se realizan y el tiempo que falta para terminar el actual sprint. En la finalización de cada sprint, aproximadamente 2 semanas laborales, se realizaron juntas para conocer los resultados obtenidos en el periodo de trabajo y, a su vez, para realizar la planificación de siguiente sprint.

La planificación del siguiente sprint consiste en la selección de las actividades que el propietario del proyecto considera necesarias para la siguiente iteración y en la complejidad que los desarrolladores pueden realizar para el periodo de 2 semanas laborales de trabajo. Al encontrarse ambas partes de acuerdo el sprint planning es terminado y el sprint es puesto en marcha.

Considero la metodología muy versátil y con un amplio valor curricular, debido a que gran parte de las empresas de desarrollo de software alrededor del mundo utilizan esta metodología como principal mecánica de trabajo.

6. CONCLUSIONES Y RECOMENDACIONES

Las prácticas profesionales se desarrollaron en una gran compañía y con una enorme cantidad de empleado en todo Alemania, que se rige por normas y políticas formalmente establecidas, considero que tiene bastantes áreas de oportunidad, un amplio desarrollo social y un increíble estabilidad y fortaleza, pero al igual, existen algunas áreas concernientes al progreso de sus practicantes que considero descuida.

Los trabajadores de la empresa tienen una amplia cantidad de oportunidades y trabajan siguiendo detalladas instrucciones, los practicantes de la empresa tienen un futuro incierto y no tienen una instrucción sobre lo que tienen que desarrollar. Personalmente no considero a la empresa como una desarrolladora de software, debido a que esa solo ofrece servicios a los bancos de Alemania y desarrolla gran cantidad de sus productos en la misma plataforma, de tal manera que no existe diversidad de conocimientos sobre IT en la compañía.

El fondo técnico de nuestra aplicación es muy diferente a las tecnologías que la empresa implementa, esto nos llevó a un gran problema, debido a que nadie conocía nuestra tecnología, tuvimos muchos problemas al momento de solicitar ayuda o tutoría en la implementación.

Como total conclusión considero mi estadía profesional como exitosa, debido a que logre aprender bastante sobre diferentes tecnologías y la experiencia profesional de trabajar en una enorme compañía, pero, considero que pudiera haber sido mejor si este proyecto no se hubiera alejado tanto del rubro principal de la compañía.

7. REFERENCIAS BIBLIOGRÁFICAS

<https://pouchdb.com/>

<https://cloudant.com/>

<https://www.ibm.com/cloud-computing/bluemix/>

<http://ionicmaterial.com/>

<https://ionicframework.com/>

<https://angularjs.org/>

<https://spring.io/>

https://es.wikipedia.org/wiki/Spring_Framework

<https://es.wikipedia.org/wiki/Bluemix>

<https://es.wikipedia.org/wiki/AngularJS>