

Universidad de Los Andes – Facultad de Ingeniería
Escuela de Ingeniería de Sistemas – Departamento de Computación
Mérida – Venezuela
Octubre 2018

Desarrollo de una app MVP para selfis y optimización de una app para una comunidad de lectores de la empresa GDStudio.

López Rojas Kristo Rafael
V-25.152.160

Índice

Introducción.....	3
Marco de Referencia.....	4
Plan de Trabajo.....	5
Desarrollo del plan.....	6
Vinculación del proyecto con las materias de la carrera.....	8
Conclusiones y Recomendaciones.....	9
Bibliografía.....	10
Anexos.....	11

Introducción

Las pasantías industriales son una oportunidad de poner en práctica todos los conocimientos adquiridos durante los años de formación en un ambiente laboral profesional previa a la obtención del título, en una carrera como Ingeniería de Sistemas, concretamente hablando en la opción de sistemas computacionales, esta experiencia es fundamental para tener una idea de la complejidad a la que se enfrentará el estudiante en el desempeño de su carrera, de las tecnologías actuales que se utilizan en la industria y experimentar en cierto modo los posibles campos de especialización futuros que se deseen realizar.

En GDStudio, la pasantía Industrial tenía el propósito de enfrentar a los pasantes con el desarrollo de un MVP completo de un software para la empresa, colaborar en el desarrollo de nuevas funcionalidades y arreglar errores de un software en producción como parte de un grupo de trabajo. Para ambos casos se planteó el uso de Ionic como framework para el desarrollo de las aplicaciones y Firebase como sistema gestor de base de datos, así como para hacer uso de funciones en backend.

Este informe explicará de manera detallada el trabajo realizado por el pasante durante el período de trabajo comprendido entre el 6 de agosto y el 17 de septiembre del año 2018 en la empresa GDStudio, Inc, el cual contendrá un marco de referencia donde encontrará información útil al momento de entender el contexto, el plan de trabajo inicial con sus modificaciones a lo largo del desarrollo de la pasantía, las actividades desarrolladas y cual es el vínculo de las pasantías con los tópicos de la carrera, para finalizar, estarán expuestas las conclusiones y recomendaciones así como la bibliografía consultada y los anexos necesarios para mostrar el trabajo realizado.

Marco de Referencia

Antes de entrar de lleno en el tema, se empezará por hablar de ciertos puntos útiles para ubicar al lector en el contexto de estas pasantías. Se dividirá el marco referencial en tres partes: La empresa, las tecnologías utilizadas y los proyectos involucrados.

La Empresa.

GDStudio es una empresa joven con una aplicación en estado de producción y con miras a lanzar nuevas aplicaciones al mercado, a lo largo del informe se estará hablando de dos aplicaciones en particular, la aplicación principal de esta empresa, Highlight y la aplicación desarrollada por el pasante llamada Portrait.

Los Proyectos.

La visión de Highlight, es promover la lectura en las personas a manera de red social, es una aplicación donde los usuarios pueden subir fragmentos de algún libro que estén leyendo, resaltando frases en concreto que fueron de su interés y compartirlas con sus amigos y seguidores, de esta forma los usuarios pueden conseguir nuevos libros con contenido relevante para ellos.

Por otro lado, Portrait, busca formar un hábito de tomar selfis de forma diaria al usuario y compartirla con el resto de la comunidad.

Las Tecnologías.

Estas dos aplicaciones fueron realizadas utilizando las mismas tecnologías, Ionic Framework para el frontend de la app y Firebase para el backend. Ionic por su parte, es un framework de javascript que permite de manera sencilla la creación de aplicaciones móviles desarrolladas con tecnologías web como html5, scss y typescript, escritas una sola vez y con posibilidades de exportación a Android y iOS, con una gran cantidad de componentes genéricos que agilizan el proceso del software. Por su parte Firebase es una plataforma para el desarrollo de aplicaciones web y móviles que provee servicios como autenticación, base de datos en tiempo real, almacenamiento en la nube, reporte de errores, envío de notificaciones, entre otros.

Plan de Trabajo

El plan de trabajo se desarrolló sin ningún inconveniente con las mismas actividades como se planteó en la propuesta previa al desarrollo de las pasantías, sin embargo, los objetivos de la semana 4 fueron alterados para atender un cambio urgente en la app de highlight:

Semana 1: Empezar el estudio del framework Ionic y Manejo de Firebase.

Semana 2: Depurar algunos errores en la app de Highlight, añadir una barra de búsqueda en la página de seguidores.

Semana 3: Diseñar y desarrollar la app de Portrait: Añadir la funcionalidad de usar la cámara.

Semana 4: Estudiar acerca de Lazyloading, Rehacer el sistema de búsqueda de highlight cambiándolo a lazyloading

Semana 5: Diseñar y desarrollar la app de Portrait: crear página del usuario con vista de sus publicaciones en formato mosaico/lista, con funcionalidad “Me gusta” y “Comentar”

Semana 6: Diseñar y desarrollar la app de Portrait: Crear un manejador de perfiles que permita a los usuarios editar perfiles, seguridad como reportar contenido abusivo y privacidad.

Nota: El plan descrito previamente es una traducción al español del plan original planteado por la empresa sin embargo se añadió a la semana 4 otras tareas y correr las demás una semana.

Desarrollo del Plan

Semana 1: En esta semana se empezó el estudio del framework Ionic para su manejo en el desarrollo de apps. Esto engloba la arquitectura dada para diseñar la app, las distintas partes que tenía la página, se aprendió el concepto de componentes, directivas y tubo (pipe) que eran las herramientas que teníamos para modificar la página de manera modularizada, los cuales se pueden llamar en cualquier página para evitar la repetición de funciones en varias páginas y así optimizar el código.

Aparte de esto se aprendió el uso de Firebase, una plataforma que ofrece una serie de servicios para el almacenamiento de datos masivos para aplicaciones. Entre estos servicios está el de base de datos relacionales, no relacionales, almacenamiento de imágenes, gestión de cuentas de usuario, sistema de notificaciones para aplicaciones entre otros. Se aprendió el manejo de base de datos a través de la consola proveída en la web, como hacer consultas a la base de datos, y el uso de promesas para encargarse del manejo asíncrono de las consultas.

Para el manejo de promesas se utilizó el lenguaje de programación TypeScript diseñado por Microsoft el cual es un superconjunto de JavaScript. Además de esto, se usó para el desarrollo de la parte funcional de las páginas que se desarrollaron en Ionic.

Semana 2: Luego de la primera semana de haber aprendido lo básico se empezó a añadir la funcionalidad de búsqueda en la página de seguidores, así como la vista de esta página a un tercer usuario. Para la barra de búsqueda se debió pensar en el método adecuado para la recuperación de datos por medio de consultas. El método empleado fue el de paginación, el cual consistía en consultar un número limitado de ítems a la base de datos y al desplazarse hacia el final de la página cargar otro lote de ítems de igual tamaño, esto sucesivamente hasta que no haya más datos recuperar de la base de datos.

Además de esto se pidió arreglar la funcionalidad de “Me gusta” en las publicaciones, estas presentaban un fallo a la hora de mostrar el contador de me gustas debido a que al darle de manera simultánea y sin pausa agregaba a estas como las veces que le dieron “Me gusta” a la publicación, cosa que era errónea pues un usuario solo puede dar una sola vez “Me gusta” a una publicación. Esto se arregló poniendo una condición de diferencia de tiempo mayor de cierto valor entre las veces clickeadas para así cuando la función se ejecutará, tendría que esperar un intervalo de tiempo pequeño para poder volver a hacerla. También se pidió arreglar la página donde se alojaban la información de los que le dieron “Me gusta” a la publicación pues mostraba un problema de paginación pues se repetían los datos de manera infinita.

Semana 3: En esta semana se empezó a desarrollar Portrait la cual era una app con el concepto de que el usuario pudiera publicar sus selfis ante la comunidad y pudiera comentar y dar “Me gusta” a otras publicaciones de selfis de otros usuarios. En esta semana se diseñó el modo en el cual se pudiera usar la cámara del celular dentro de la app para poder tomarse selfis bloqueando la cámara trasera para que solo se pueda usar la frontal, para esto se necesitó dos plugin's soportados por el framework que nos permitían usar la cámara y modificar sus opciones visibles en la previsualización de la toma de foto. Luego de esto se creó una página en la cual se pudiera añadir a la foto una leyenda o comentario que se deseara poner con la foto, y finalmente una opción por si la app se está ejecutando en un página web de modo de prueba en pc.

Semana 4: Debido a la nueva actualización del framework ionic, se decidió trabajar en esta semana de nuevo en la app de highlight debido a que la actualización iba a dejar de soportar el sistema de navegación tipo “pila” e iba a usar un sistema de navegación “Enrutado”. Previamente para poder hacer esto se necesitó primero implementar a las páginas, componentes, pipes, directivas y entre otros (modal y popovers) el módulo “Lazyloading”, esto además que era necesario para hacer el cambio de navegación a uno enrutado, optimizaba la velocidad de despliegue de páginas de la app considerablemente. Sin embargo, se debió primero estudiar este método y luego de ello, desmantelar los módulos encargados de la carga de los elementos e ir modificando las declaraciones de estas en todas las partes necesarias. Luego de esto se procedió a testear y depurar los errores que llegasen a surgir durante la implementación.

Semana 5: de nuevo en la app de Portrait, se creó la vista del usuario en la cual consistía en su página de perfil y en la página de publicaciones que los demás publicaban en la plataforma. En la del perfil se tenía que implementar una vista “mosaico”, la cual permitía ver las fotos en un tamaño reducido y clickeables para poder verlas de manera extendida en una carta. Tanto en la página de perfil como en la de publicaciones se implementó la vista tipo “lista”, este consistía en una lista de cartas el cual era un componente que se diseñó para colocar la publicación y así mostrarlas en la página que se instancia. Cabe añadir que ambas formas de mostrar las publicaciones se les fue implementado el método de paginación antes explicado.

Semana 6: Finalmente en esta semana se depuraron algunos bugs, y se diseñó la parte de edición del usuario, el cual permite cambiar el nombre, y la foto de perfil que pueda tener el usuario en la app.

Vinculación del proyecto con las materias de la carrera

Para el desarrollo de las actividades de hechas en las pasantías fue indispensable el conocimiento de algunos conceptos ya vistos en otras materias tales como lo fueron:

- Programación 2: Para el conocimiento de clases y métodos utilizados en TypeScript, además del manejo de atributos privados, públicos, Herencia y constructores de la clase
- Base de Datos: Para el Manejo de la base de datos no relacionales y relacionales, concepto de consultas, y estructuración de datos.
- Ingeniería de Software: Para la organización de tareas en un ambiente de trabajo para el desarrollo del software y manejo básico de frameworks.

Conclusiones y Recomendaciones

El trabajo hecho en las pasantías es una experiencia valiosa que debería inclusive fomentarse con mayor frecuencia para que más estudiantes logren hacerlas antes de graduarse. Es importante entender que existe una brecha que separa el estudio que se hace en la universidad, y el ambiente laboral que se logró reconocer en este primer acercamiento.

También las pasantías sirvieron para un crecimiento más personal por el hecho que exige un desarrollo de aptitud laboral correcta y responsable que será importante para nuevas oportunidades de trabajo en otras empresas, así como el aprendizaje mas a fondo del uso de frameworks y el lenguaje de programación TypeScript.

Cabe añadir que al ser una empresa extranjera en la que se realizó las pasantías nos dio también un intercambio cultural importante. del cual pudimos aprender a través de la comunicación cotidiana con nuestro jefe directo acerca de algunas costumbres y experiencias que nos relataba con el objetivo de nuestro crecimiento como persona profesional en el campo de desarrollo de aplicaciones orientadas a servicios y aún más en general, el estilo de negocio startups enfocado al desarrollo de aplicaciones.

Bibliografía

<https://ionicframework.com/docs/>

<https://firebase.google.com/docs/>

Anexos

The screenshot shows the Ionic App interface with a browser window displaying a profile page for 'Chris Rojas' (@galileo.20). The console on the right shows a warning from Firestore (4.13.0) regarding timestamp changes. The warning text is as follows:

```
Angular is running in the development mode. Call enableProdMode() to enable the production mode. core.js:3688
grid Component grid-post.ts:23
Profile tab profile.ts:35
index.esm.js:272
[2018-10-09T02:14:27.558Z] @firebase/firestore: Firestore (4.13.0):
The behavior for Date objects stored in Firestore is going to change
AND YOUR APP MAY BREAK.
To hide this warning and ensure your app does not break, you need to add the
following code to your app before calling any other Cloud Firestore methods:

const firestore = firebase.firestore();
const settings = { /* your settings... */ timestampsInSnapshots: true };
firestore.settings(settings);

With this change, timestamps stored in Cloud Firestore will be read back as
Firestore Timestamp objects instead of as system Date objects. So you will also
need to update code expecting a Date to instead expect a Timestamp. For example:

// Old:
const date = snapshot.get('created_at');
// New:
const timestamp = snapshot.get('created_at');
const date = timestamp.toDate();

Please audit all existing usages of Date when you enable the new behavior. In a
future release, the behavior will change to the new behavior, so if you do not
follow these steps, YOUR APP MAY BREAK.
```

The screenshot shows the Ionic App interface with a browser window displaying a 'Portrait' post by 'galileo.20' on 'Tue Oct 02 2018 00:19:02'. The console on the right shows a warning from Firestore (4.13.0) regarding timestamp changes. The warning text is as follows:

```
Angular is running in the development mode. Call enableProdMode() to enable the production mode. core.js:3688
grid Component grid-post.ts:23
Profile tab profile.ts:35
index.esm.js:272
[2018-10-09T01:16:52.321Z] @firebase/firestore: Firestore (4.13.0):
The behavior for Date objects stored in Firestore is going to change
AND YOUR APP MAY BREAK.
To hide this warning and ensure your app does not break, you need to add the
following code to your app before calling any other Cloud Firestore methods:

const firestore = firebase.firestore();
const settings = { /* your settings... */ timestampsInSnapshots: true };
firestore.settings(settings);

With this change, timestamps stored in Cloud Firestore will be read back as
Firestore Timestamp objects instead of as system Date objects. So you will also
need to update code expecting a Date to instead expect a Timestamp. For example:

// Old:
const date = snapshot.get('created_at');
// New:
const timestamp = snapshot.get('created_at');
const date = timestamp.toDate();

Please audit all existing usages of Date when you enable the new behavior. In a
future release, the behavior will change to the new behavior, so if you do not
follow these steps, YOUR APP MAY BREAK.
```

The screenshot shows the 'Edit profile' page in a browser. The page has a back arrow, a title 'Edit profile', and a profile image placeholder labeled 'Profile Image'. Below the image is a form with the following fields:

- Basic info
- ID: galileo.20
- First name: Chris
- Last name: (empty)

The browser's developer console is open, showing a warning from Firebase Firestore (4.13.0) regarding the change in behavior for Date objects. The warning text is:

```
[2018-10-09T02:14:27.558Z] @firebase/firestore: Firestore (4.13.0):
The behavior for Date objects stored in Firestore is going to change
AND YOUR APP MAY BREAK.
To hide this warning and ensure your app does not break, you need to add the
following code to your app before calling any other Cloud Firestore methods:

const firestore = firebase.firestore();
const settings = { /* your settings... */ timestampsInSnapshots: true };
firestore.settings(settings);

With this change, timestamps stored in Cloud Firestore will be read back as
Firebase Timestamp objects instead of as system Date objects. So you will also
need to update code expecting a Date to instead expect a Timestamp. For example:

// Old:
const date = snapshot.get('created_at');
// New:
const timestamp = snapshot.get('created_at');
const date = timestamp.toDate();

Please audit all existing usages of Date when you enable the new behavior. In a
future release, the behavior will change to the new behavior, so if you do not
follow these steps, YOUR APP MAY BREAK.
```

Below the warning, there are several log entries for 'grid Component' and 'Profile tab' from various source files.

The screenshot shows the 'Likes' page in a browser. The page has a back arrow, a title 'Likes', and a list of two users:

- galileo.20
- dasfasf2

At the bottom of the page, there is a navigation bar with a home icon and a user profile icon. The browser's developer console is open, showing the same Firebase Firestore warning as in the previous screenshot. Additionally, there is a yellow error message:

```
Failed to decode downloaded font: https://fonts.gstatic.com/stats/Raleway/normal/7 : 8100/s/home/tab-1/like:1
00
```

Below the error, there are several log entries for 'grid Component', 'Profile tab', and 'ionViewDidLoad LikePage' from various source files.