

Aplicación móvil para registrar, calificar  
y llevar un control del feedback que  
realizan los trabajadores sobre su  
desempeño en las actividades de una  
empresa.



Alexander Fernández

Caracas, 1 de Octubre del 2020

## ÍNDICE GENERAL

ÍNDICE DE FIGURAS/TABLAS .....	3
SINOPSIS .....	4
PLANTEAMIENTO DEL PROBLEMA.....	5
OBJETIVO GENERAL .....	6
OBJETIVOS ESPECÍFICOS.....	6
DESCRIPCIÓN DE LA EMPRESA.....	7
METODOLOGÍA EMPLEADA.....	9
RESULTADOS .....	25
CONCLUSIONES Y RECOMENDACIONES.....	26
REFERENCIAS BIBLIOGRÁFICAS .....	27
APÉNDICES.....	28

## ÍNDICE DE FIGURAS/TABLAS

Ilustración 1 .....	10
Ilustración 2 .....	10
Ilustración 3 .....	11
Ilustración 4 .....	16
Ilustración 5 .....	16
Ilustración 6 .....	17
Ilustración 7 .....	18
Ilustración 8 .....	19
Ilustración 9 .....	20

## SINOPSIS

Las evaluaciones del desempeño al personal de trabajo es un factor importante en las empresas para su desarrollo y su estudio determina la calidad de trabajo en la organización. En empresas muy grandes donde los supervisores poseen gran cantidad de trabajadores, normalmente no se puede recordar o mantener el historial completo de desempeño de todos los trabajadores, y solamente se recuerda lo más reciente que sucede, generando un sesgo en la evaluación de cada uno de los trabajadores.

Este tipo de problemas provoca resultados no esperados en el desempeño de los trabajadores, por lo que genera un descontento general en la empresa. Los supervisores se ven entonces con una carga adicional de trabajo, al tener que llevar un registro de todo lo que realiza cada trabajador que tiene a su mando.

El trabajo de la pasantía consistió en el desarrollo de una aplicación móvil de nombre “FeedMe”, cuyo objetivo principal es registrar, calificar y llevar un control del *feedback* que realizan los trabajadores sobre su desempeño en las actividades de una empresa, a través del envío de mensajes entre trabajadores y supervisores sobre las actividades en las que se desenvuelven diariamente.

Estos mensajes son publicados en el perfil del usuario emisor y receptor si ambas partes concuerdan en que el mensaje sea público, de manera que todas las personas registradas vean los mensajes y puedan calificarlos de manera positiva o negativa. El número de calificaciones positivas y negativas en los mensajes y la cantidad de mensajes enviados y recibidos por el trabajador son factores para llevar el control sobre el desempeño en las actividades en la empresa, y estos se ven reflejados en el ranking de trabajadores de la aplicación.

Los supervisores poseen en la aplicación cierta cantidad de trabajadores asignados, en donde se les permite ver los mensajes enviados y recibidos que son públicos y privados, con la finalidad de tener cierto control y ver el desempeño de los trabajadores a su cargo.

## PLANTEAMIENTO DEL PROBLEMA

En muchas empresas se realizan evaluaciones al personal sobre su desempeño en algún período de tiempo, a través de las cuales se realizan análisis para determinar la calidad del trabajo que se desarrolla dentro de la organización.

Existen diversas maneras para obtener información acerca del desempeño realizado por algún trabajador, como lo son los comentarios que realizan los demás empleados sobre los resultados obtenidos dentro de la empresa. Generalmente, los encargados de gestionar o llevar la evaluación de desempeño de los trabajadores son los supervisores de la empresa, donde cada supervisor posee una cantidad específica de trabajadores a su cargo.

Los problemas surgen al momento de tratarse de empresas muy grandes o supervisores con una gran cantidad de trabajadores a su cargo. La principal consecuencia se puede observar en que los supervisores no pueden recordar el historial completo de desempeño de todos los trabajadores a su cargo, y solamente suelen recordar lo más reciente que sucede, generando un sesgo en la evaluación de cada uno de los trabajadores.

La inestabilidad en los resultados del desempeño de los trabajadores, produce un descontento general en la empresa. Además de que los datos que resultan de las evaluaciones no son los esperados, los supervisores se ven con la tarea de llevar una carga adicional de trabajo, al llevar un registro de todo lo que sucede en la empresa referente a los trabajadores a su cargo, durante los ciclos de tiempo establecidos.

La solución propuesta es proporcionar a las empresas una herramienta que tenga las funcionalidades de registrar los *feedbacks* o comentarios que hace un trabajador en la empresa, calificarlos y compartirlos para evaluar el desempeño laboral.

## **OBJETIVO GENERAL**

Desarrollar una aplicación móvil que permita registrar, calificar y llevar un control del feedback que realizan los trabajadores sobre su desempeño en las actividades de una empresa.

## **OBJETIVOS ESPECÍFICOS**

1. Desarrollar el registro y gestión de credenciales integrado con Firebase.
2. Desarrollar la pantalla de envío de un mensaje a un trabajador con control sobre si el mensaje es público.
3. Desarrollar la pantalla que le permite a un supervisor consultar los mensajes que ha recibido y enviado un trabajador.
4. Desarrollar la pantalla de perfil de un usuario donde se pueden visualizar los mensajes recibidos que son públicos.
5. Desarrollar la visión de un administrador general que asigna supervisores a los trabajadores y puede visualizar todos los mensajes enviados y recibidos entre los trabajadores y supervisores.
6. Desarrollar un módulo que contabiliza el número de mensajes enviados, el número de mensajes recibidos y los mensajes calificados como positivos.
7. Desarrollar el ranking de trabajadores basados en el feedback que dan y el feedback que reciben los trabajadores.

## **DESCRIPCIÓN DE LA EMPRESA**

Synergy Vision es una empresa encargada de otorgar formación, tecnología y consultoría en el área de trading, inversión, economía, finanzas y riesgo. Ofrecen formación para prácticas de consultoría y ejercicio profesional especializado para alcanzar certificaciones reconocidas a nivel mundial.

Utilizan técnicas como Data Science, Datamining y Machine Learning para el desarrollo de aplicaciones. También están destinados en asistir a empresas a través de los canales electrónicos con aplicaciones web y móviles y ofrecer servicios en el área de la especulación, inversión, economía, finanzas y riesgo.

La empresa se encarga del desarrollo de sistemas especializados para facilitar la gestión de inversiones, portafolios, trading, simuladores, modelos, finanzas corporativas, finanzas internacionales, valoración, optimización, riesgo, entre otros.

### **Historia**

2007: Inicio de operaciones en Venezuela. Foco inicial Aplicaciones móviles para empresas de Seguros y Comercializadoras.

2008: Synergy-GB se consolida en el sector de seguros a través de soluciones móviles para el ramo de vehículos.

2009: Se abre oficinas en Panamá.

2010: Lanzamiento de Banesco Móvil.

2011: Lanzamiento de Mercantil Móvil. Primera aplicación de banca móvil del mundo en Tablets QNX (PlayBook).

2012: Se abre sucursal en EEUU. Cambio de nombre a Synergy-GB a nivel corporativo. Lanzamiento de banca móvil del banco Bicentenerio. Lanzamiento de Banesco móvil en RRDD y en Panamá. Finalist en Wireless Achievement Award 2012 de RIM (mBanking). Nominacion a los WSA- World Summit Award 2012 mobile content.

2013: Nace como producto Banca+ (multicanal multiplataforma). Lanzamiento de Banca+ juegos con Banesco Aventura. Lanzamiento de Banca+ transaccional en redes sociales con Banesco Amigos. Crecimiento de Banca+ móvil en banco

Fondo Común y CrediCorp. Lanzamiento de Social+ en Mercantil banco y en Commerce Bank.

2014: Lanzamiento de TodoTicket móvil. Crece Banca+ móvil en banco Activo, BanFanV y banco nacional de Panamá. Lanzamiento de Banca+ home banking en BOD. Firma de alianza latinoamericana con DATAPRO.

2015: Lanzamiento de Banca+ móvil jurídica en Mercantil banco. Lanzamiento de Telco+ móvil en Digitel.

2016: Relanzamiento de Asegura+ móvil para empresas de seguro. Lanzamiento del servicio PUSH integrado a canales digitales de Mercantil banco.

2017: El grupo Synergy-GB decide separar las operaciones para atender la Banca en Synergy-GB. El resto de los segmentos de empresas pasan atenderse en la empresa Apps2GO. Para los aspectos de servicio financiero pasan atenderse en la empresa Synergy Vision.

2018: Se concreta el merge con el grupo Datapro Inc. el proveedor más importante de Core Bancario en Latinoamérica. Se implanta Banca+ móvil personas y jurídica con BNC (Venezuela). Se implanta en la nueva versión de Banca+ Móvil Unibank (Panamá).

### **Misión**

“Habilitar soluciones tecnológicas innovadoras, para apoyar modelos de negocio que mejoren la competitividad y productividad de nuestros clientes de la BANCA”.

### **Visión**

“Hacer de SYNERGY-GB y sus productos 'Banca+ bySynergyGB' la piedra angular de la transformación digital de la Banca en Latinoamérica”.

### **Ubicación**

Centro San Ignacio, Torre Keplet, Oficina 603, Chacao, Miranda - Venezuela



## METODOLOGÍA EMPLEADA

Se empleó una metodología ágil basada en el marco de trabajo Scrum. Esta metodología se enfoca en el método de trabajo que mejor se adapta a la situación debido a que el tiempo establecido es por semanas, pudiendo separar las tareas por *sprints* y poder presentar entregas continuas sobre los avances al Product Owner.

También incita a mantener tanto a los clientes como a los desarrolladores trabajando conjuntamente para poder obtener un producto que sea funcional y que cumplan con los objetivos planteados. Esto hace que el tutor empresarial esté en constante contacto con el desarrollador y pueda dar consejos y solucionar los problemas que se presenten de la manera más rápida y eficaz posible.

Este tipo de metodología clásica también mantiene un orden de ejecución de las tareas y se asocia muy bien a las historias de usuario, aumentando la calidad del software, la satisfacción del cliente con el producto y reduciendo el costo del desarrollo.

La siguiente imagen representa el cronograma establecido para el desarrollo de las actividades establecidas en la pasantía, cuya elaboración fue hecha tomando en cuenta la metodología que sería utilizada para desarrollar la solución.

Como se aprecia en la *ilustración 1*, la metodología de trabajo se planeó principalmente en *sprints* de dos semanas para un total de tres *sprints* cubriendo las primeras seis semanas de la elaboración, teniendo reuniones con el tutor empresarial cada dos semanas. La Primera fase estaría representada por los dos primeros *sprints* y se enfocarían en el análisis y diseño de la aplicación (semana 1) y a la integración con firebase para así desarrollar el registro e inicio de sesión (semana 2).

Sprint	Actividad	Fecha inicio	Fecha final	Agosto				Septiembre	
				S1	S2	S3	S4	S5	S6
1	Levantamiento de información	24-08-2020	24-08-2020						
	Definición de historias de usuario	25-08-2020	25-08-2020						
	Definición de estructura de base de datos no relacional	26-08-2020	26-08-2020						
	Estructura inicial del proyecto - Asignación de datos de prueba en BD	27-08-2020	28-08-2020						
	Integración con Firebase	31-08-2020	31-08-2020						
	Implementación de Registro de Usuarios	1-09-2020	2-09-2020						
	Implementación de Inicio de Sesión	3-09-2020	3-09-2020						
2	Implementación de Barra de Navegación	4-09-2020	4-09-2020						
	Implementación de la vista de perfil de usuario	7-09-2020	8-09-2020						
	Implementar módulo de contabilización de mensajes enviados, recibidos y calificados como positivos del trabajador	9-09-2020	10-09-2020						
	Implementación de la vista de actualización del perfil de usuario	11-09-2020	14-09-2020						
	Implementación de vista de Mensajes enviados	15-09-2020	16-09-2020						
	Implementación de vista de Mensajes recibidos	17-09-2020	17-09-2020						
3	Implementar la pantalla de envío de mensaje a un trabajador	18-09-2020	21-09-2020						
	Implementación del Buscador de usuarios	22-09-2020	22-09-2020						
	Implementación de vistas de otros perfiles de usuario	23-09-2020	23-09-2020						
	Desarrollar módulo de Ranking de Trabajadores	24-09-2020	25-09-2020						
	Implementación de vista de trabajadores asignados al supervisor	28-09-2020	28-09-2020						
	Implementación de vista de mensajes recibidos y enviados de un trabajador asignado a un supervisor	29-09-2020	30-09-2020						
	Desarrollo de vista de administrador general (mensajes enviados y recibidos entre trabajadores y supervisores)	1-09-2020	2-09-2020						

Ilustración 1

La segunda fase contempla plenamente la implementación de la aplicación que abarcaría el *sprint 2* y el *sprint 3* (semanas 3, 4, 5 y 6) cumpliendo con todos los objetivos que se plantearon. La *ilustración 2* contempla de forma general como se plantó la metodología para desarrollar la aplicación.

FASE	SPRINT	SEMANA
Análisis y diseño	1	1 y 2
Implementación	2 y 3	3, 4, 5 y 6

Ilustración 2

En la ilustración anterior, se aprecia que por el tiempo disponible y los requerimientos y objetivos de la aplicación, solo se plantean las tres primeras fases del ciclo del desarrollo de software. Al iniciar el ciclo de pasantías, se mantuvo los objetivos de cada semana, pero se optó por modificar el tamaño de los *sprints*, como se puede apreciar en la *ilustración 3*

FASE	SPRINT	SEMANA
Análisis y diseño	1	1
Implementación	2	2
	3	3
	4	4
	5	5
Pruebas	6	6

**Ilustración 3**

Este nuevo modelo, permite hacer entregas de avances cada semana por lo que el tutor empresarial estará mucho más involucrado con el desarrollo de la aplicación, ya que las reuniones serian establecidas semanalmente.

También se pudo agregar en la última semana la fase de pruebas, en donde si bien no se implementó pruebas unitarias, se utilizó para detectar posibles errores en la aplicación y presentar una aplicación impecable. Esta última semana pudo incorporarse debido a que se contactó con el tutor empresarial antes del comienzo de las pasantías y por tanto se pudo adelantar el trabajo.

Con el fin de llevar un registro de actividades y debido a las circunstancias de salud, se utilizó la plataforma de control de versiones *github* para registrar las actividades que se fueron desarrollando cada semana y en donde se establecieron las historias de usuario y el código fuente de la aplicación, como se aprecia en la *ilustración 4*.

## **Fase de Análisis**

Durante la fase de análisis, correspondiente a las primeras semanas de actividades y previos contactos con el tutor empresarial, se utilizó la plataforma de *Gmail* como medio de comunicación, obtención de información y análisis del dominio en el que estaría centrado el desarrollo del producto.

Además se utilizó la plataforma de *Github* para tener un registro de las historias de usuario que se desarrollaron por cada sprint. Las actividades fueron planteadas a través de reuniones semanales con el tutor empresarial en la plataforma *Zoom*.

### Fase de Diseño

En la fase de diseño, se utilizó las herramientas de *Lucidchart* para la elaboración del modelo de base de datos, que si bien es una base de datos no relacional, se planteó una especie de modelo de datos para representar la estructura general de la base de datos.

También se realizaron esquemas en físico para plasmar las distintas interfaces de usuario en la aplicación, sin embargo la mayoría fue desarrollada semanalmente en seguimiento con las historias de usuario y las recomendaciones de *look and feel* del tutor empresarial.

### Fase de Implementación

Durante la fase de implementación, se utilizaron frameworks de desarrollo web y móvil, orientados a dar solución para distintas plataforma sin necesidad de programar la misma aplicación para cada dispositivo nativo, como sería el caso de utilizar lenguajes nativos como Java o C#. Para el desarrollo de la aplicación se utilizaros las siguientes dependencias:

- **Angular:** Framework *opensource* que utiliza lenguaje en *TypeScript* y facilita la creación de programas destinados a aplicaciones web SPA (*Single Page Application*) separando completamente el *frontend* del *backend* a través del uso de patrones MVC (Modelo-Vista-Controlador). Se destaca por basarse el desarrollo de componentes modulares e independientes, permitiendo en gran medida la escalabilidad y la adaptación al modelo de negocio.
- **Ionic:** Framework *opensource* utilizado en el desarrollo de aplicaciones móviles híbridas (utiliza lenguaje HTML, CSS y Javascript/Typescrip). Permite la sincronización con frameworks para el desarrollo de aplicaciones

SPA como Angular, de tal manera que a través de su sistema *WebView* pueda compilar todo el desarrollo web en código nativo (Java, C#, entre otros), además de permitir a través de plugins el uso del hardware en los dispositivos móviles.

- **Capacitor:** Herramienta desarrollada por Ionic para acceder desde las tecnologías del desarrollo web a los recursos de los dispositivos (cámara, teclado, sensores) optimizando el uso del framework.
- **NodeJs:** Es un entorno de ejecución *opensource* que ejecuta código de JavaScript en los navegadores web y permitir a los desarrolladores escribir scripts del lado del servidor. NodeJs fue utilizado como dependencia del CLI de Angular e Ionic, además de que el *node package manager* (NPM) es utilizado para instalar las dependencias en la aplicación.
- **Firebase:** Plataforma desarrollada por Google utilizada para el desarrollo de aplicaciones web o móviles de manera rápida y efectiva. Permite el almacenamiento de información a través de bases de datos no relacionales en tiempo real, almacenamiento de archivos, hosting, autenticación con servicios de correo electrónico y *Cloud Functions*.
  - **AngularFirestore:** Librería oficial de Angular para el acceso a las funciones de Firebase, utilizada para sincronizar datos en tiempo real, renderizar datos desde el servidor y utilizar funciones de autenticación de usuarios.
- **QuillJs:** Editor de texto enriquecido, construido para desarrolladores y disponible en plataformas móviles y web. Otorga herramientas para generar contenido HTML a través de interfaces de usuario, como letras en negrita, iconos, títulos, margen, entre otros.
- **Ngx-Translate:** Librería para permitir carga dinámica de archivos de traducciones en formato JSON a las aplicaciones. Si bien la aplicación tiene un solo idioma desarrollado, se implementó esta librería para que en un futuro se puedan aplicar traducciones sobre la aplicación.
- **Ngx-Datatable:** Librería de Angular para presentar grandes cantidades de

información en tablas de contenido, permitiendo al cliente ordenar los datos por columnas y mostrar la información en filas.

### **Fase de pruebas**

Volviendo a la idea planteada anteriormente, por el corto plazo de tiempo establecido para el desarrollo de la aplicación y el gran alcance que tiene, se llegó a la decisión de no desarrollar pruebas unitarias en la aplicación. Por otro lado, se pudo tomar tiempo para realizar pruebas de aceptación, validando que se cumplieran con todos los requerimientos funcionales del sistema y hacer pruebas en distintos dispositivos y con diferentes usuarios. Se hicieron pruebas a través del navegador web Chrome y pruebas en dispositivos Android, específicamente en Xiaomi Redmi 8 y Xiaomi Redmi note 8.

## DESARROLLO

Durante el ciclo de desarrollo de la aplicación, al tener conocimientos de que el alcance y objetivos de la solución eran amplios, se decidió iniciar lo más pronto posible con la elaboración de la solución, por lo que se pudo establecer el inicio de las actividades de implementación a partir del 16 de agosto.

### **Fase de Análisis (24/07/2020 - 15/08/2020)**

Durante la fase de análisis, se mantuvieron conversaciones con el tutor empresarial con el fin de establecer el dominio de la solución, una aplicación móvil para evaluar el desempeño de trabajadores en una empresa, a través del *feedback* que realizan los mismos trabajadores y supervisores en las actividades que se realizan diariamente.

Se establecieron las tecnologías a utilizar durante el desarrollo de la aplicación. También se realizó una investigación de preparación para el uso de las herramientas de desarrollo de software, que incluye la búsqueda de guías, documentos oficiales de las herramientas, videos, entre otros.

Las Historias de usuario se plantearon a partir de los objetivos específicos propuestos, en el cual cada semana se creaban tareas e historias de usuario asociadas a las actividades semanales en la que se desarrollaría el proyecto.

También se mantuvo conversación vía *whatsapp* con el tutor empresarial, en caso de presentarse algún inconveniente o duda al respecto del dominio del sistema o alguna mejora en la interfaz de usuario.

### **Fase de Diseño (16/08/2020 - 19/08/2020)**

En la fase de diseño se planteó la arquitectura de la aplicación, el modelo de base de datos (si bien es no relacional, se planteó un modelo de datos para tener una base en el diseño).

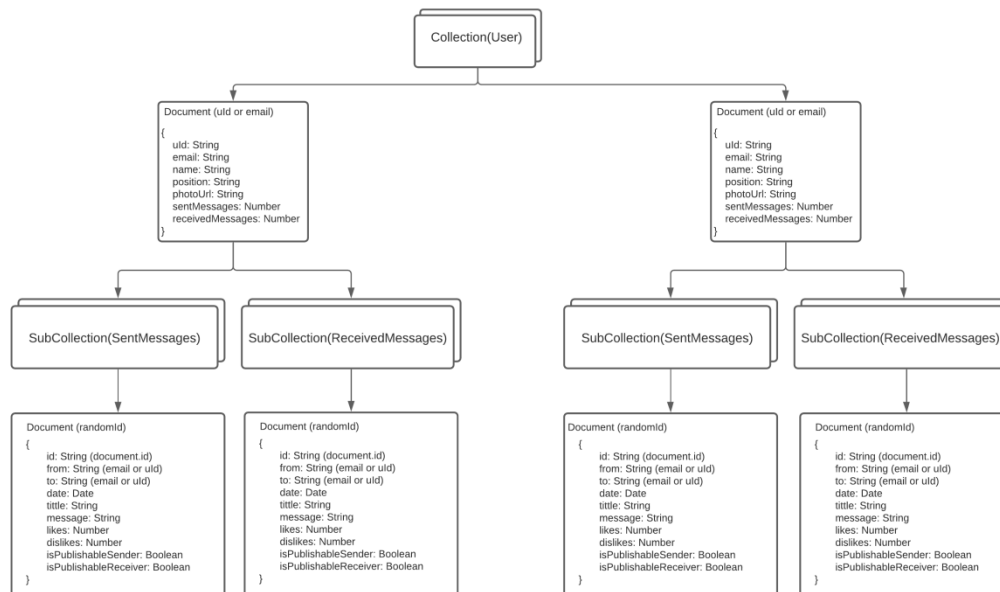


Ilustración 4

La *ilustración 4* contempla el modelo de base de datos no relacional orientada a documentos, donde se aprecian las distintas colecciones, documentos y subcolecciones en la base de datos, así como la forma de representar los datos. Posteriormente se refactoriza la base de datos, debido a que el modelo anterior genera datos duplicados en las subcolecciones de mensajes, dando como resultado el modelo que se puede apreciar en la *ilustración 5*

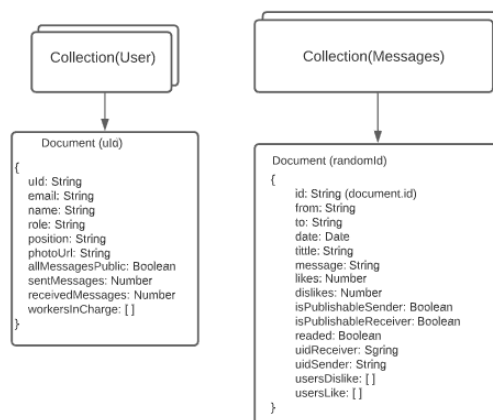


Ilustración 5

En la imagen anterior se aprecia un cambio radical, debido a que



desaparecen las subcolecciones y todos los mensajes están centralizados en una misma colección, los cuales se asocian a los respectivos usuarios a través del id del usuario y se conoce si es enviado o recibido a través del correo, el cual es único para cada uno.

Referente a la arquitectura de la aplicación, se implementó el desarrollo basado en la arquitectura de angular, cuyo modelo se centra en el desarrollo por componentes.

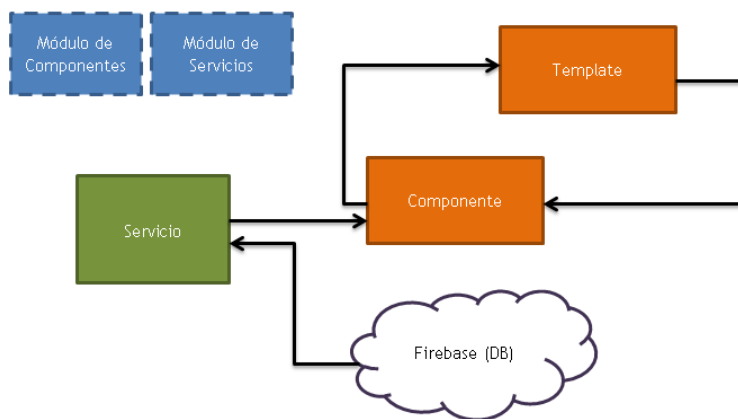


Ilustración 6

En la *ilustración 6* se puede apreciar la idea de la arquitectura de la aplicación. Los componentes manejan la lógica correspondiente al uso de los servicios y manejo de los datos para ser mostrados a través del template, el cual representa al código HTML de la aplicación e interactúa a través de eventos y propiedades con el componente. Los servicios se encargan de mantener la lógica de acceso a los datos contenidos en la base de datos.

Cada aplicación de Angular se caracteriza por ser modular y estos módulos se encargan de manejar los metadatos que describen al módulo, manejar las importaciones, exportaciones y dependencia entre módulos, crear servicios y establecer declaraciones.

#### **Fase de Implementación (16/08/2020 - 15/09/2020)**

La fase de implementación inició con la configuración base del proyecto, se crea a partir del CLI de Ionic utilizando los comandos necesarios. Posteriormente

se instala Capacitor para el uso de los componentes de los dispositivos móviles a través de los plugins que ofrece el Framework.

- Creación de la estructura con el CLI de Ionic.
- Instalación de Capacitor para utilizar los componentes de los dispositivos móviles a través de los plugins que ofrece el Framework.
- Inicializamos Capacitor con la información de la aplicación, que corresponde al "appName" y al "appId".
- Se realiza el Build de nuestra aplicación, esto creará la carpeta "www".
- Agregamos las plataformas de ios y Android que corresponden a los proyectos nativos.
- Se agrega al mismo nivel del *root* de la aplicación, las carpetas "informe" y "doc". Estas carpetas contienen un informe detallado del proceso de desarrollo de la aplicación y el documento de propuesta de pasantía.

Finalizada esta etapa, se procedió a establecer la estructura base del proyecto, utilizando la documentación de Angular referente a las buenas prácticas de desarrollo.

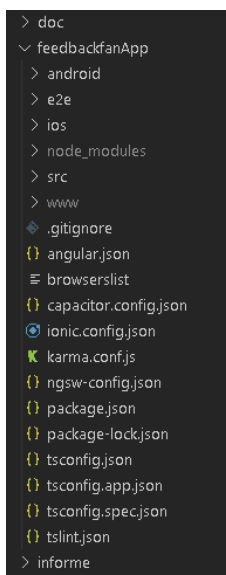


Ilustración 7

En la *Ilustración 7*, se ofrece una vista de la estructura base de la aplicación, en donde podemos analizar los siguientes puntos:

- La carpeta ios y android, corresponden a las aplicaciones móviles nativas configuradas a través del webView de Ionic.
- La carpeta src contiene la aplicación web, desarrollada a través de código HTML, CSS y JavaScript.
- Los archivos .JSON corresponden a configuraciones realizadas por el CLI de Ionic, como por ejemplo las dependencias de la aplicación.

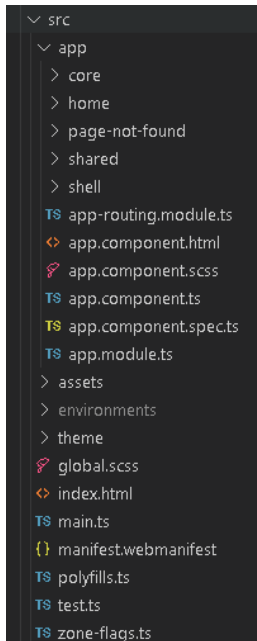


Ilustración 8

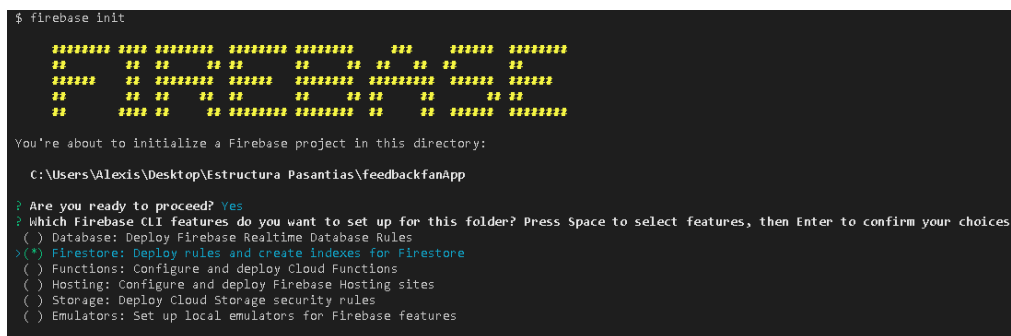
En la *Ilustración 8* se puede apreciar la forma en cómo se estructura la carpeta src, la cual es en donde se implementó la mayoría del código desarrollado, podemos destacar los siguientes puntos:

- La carpeta Core contiene un módulo y funciona para establecer todos los servicios hacia las APIs externas, en este caso, todas las peticiones hacia Firebase. Su módulo se importa en el app.module para utilizarlo en cualquier parte de la aplicación.
- La carpeta home es el componente que se carga al iniciar la aplicación. Este componente será modificado posteriormente.
- La carpeta page-not-found contiene el módulo y las páginas que se va a establecer al momento de no conseguir alguna ruta especificada en la aplicación.
- La carpeta shared va a contiene alertas, notificaciones, pipes y otros componentes que se comparten entre todos los componentes y que pueden reutilizarse.
- La carpeta shell, corresponde a una forma de crear esqueleto de precargado de los datos en la aplicación. También posee el DataStore para hacer de una manera más fluida la obtención y envío de información sin necesidad de bloquear las funcionalidades de la aplicación mientras se espera por los datos.
- La carpeta assets contiene iconos e imágenes estáticas de la aplicación, como por ejemplo la imagen que carga en page-not-found.
- La carpeta environments contiene las variables de entorno y api Keys de la aplicación. Por motivos de seguridad, esta carpeta no será publicada en el repositorio, debido a que es público.
- La carpeta theme, contiene información referente a los estilos scss globales

de la aplicación.

Una vez configurada la estructura inicial del proyecto, se procedió a la configuración de Firebase tanto en la plataforma como en el proyecto, tomado en cuenta los siguientes puntos:

- En la consola del proyecto en Firebase, habilitamos el método de inicio de sesión por correo electrónico/contraseña. Esto va a permitir la autenticación por correo electrónico.
- Configuramos Cloud Firestore para la capa de persistencia de la aplicación. Para esto establecemos su ubicación en us-central.
- Establecemos las reglas de seguridad de la base de datos. En este caso como estamos en etapa de desarrollo, se permitirá agregar y leer documentos de la base de datos.
- Configuramos la aplicación web en Firebase para que nos asignen las claves y variables de acceso a los servicios.
- Instalamos la librería de AngularFire para conectar Firebase con la aplicación. Esta librería también posee lo necesario para hacer las peticiones de autenticación y de Cloud Firestore.
- Agregamos las variables de entorno que nos proporciona Firebase en el environment.ts y environment.prod.ts.
- Iniciamos el proyecto de Firebase en la aplicación. Seleccionamos Firestore para utilizar los servicios de la base de datos.



```
$ firebase init

#####
#                                     #
#  ##  ##  ##  ##  ##  ##  ##  ##  #
#  ##  ##  ##  ##  ##  ##  ##  ##  #
#  ##  ##  ##  ##  ##  ##  ##  ##  #
#  ##  ##  ##  ##  ##  ##  ##  ##  #
#  ##  ##  ##  ##  ##  ##  ##  ##  #
#####

You're about to initialize a Firebase project in this directory:

  C:\Users\Alexis\Desktop\Estructura Pasantias\feedbackfanApp

> Are you ready to proceed? Yes
> Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices.
  ( ) Database: Deploy Firebase Realtime Database Rules
  > (*) Firestore: Deploy rules and create indexes for Firestore
  ( ) Functions: Configure and deploy Cloud Functions
  ( ) Hosting: Configure and deploy Firebase Hosting sites
  ( ) Storage: Deploy Cloud Storage security rules
  ( ) Emulators: Set up local emulators for Firebase features
```

Ilustración 9

Esto creará los archivos de configuración para el proyecto, .firebaserc, firebase.json y firestore.indexes.json. El proyecto estará listo para

utilizar los servicios de Firebase.

A partir de este punto, se comienzan a implementar y desarrollar los componentes para el desarrollo de la aplicación y cubrir los objetivos planteados, correspondiente a las siguientes funcionalidades:

- **Inicio de sesión en la aplicación.**
  - Creación de la carpeta auth, que contiene todo lo relacionado a la autenticación del usuario, olvido de contraseña, registro y reseteo de clave.
  - Creación del componente sign-in dentro de la carpeta auth, contiene toda la lógica del inicio de sesión.
  - Creación de los servicios relacionados al inicio de sesión en la carpeta core.
- **Registro de usuarios en la aplicación.**
  - Creación del componente sign-up dentro de la carpeta auth, contiene toda la lógica del registro de usuarios.
  - Creación de los servicios relacionados al registro en la carpeta core.
- **Perfil de usuario y edición de datos personales.**
  - Creación del componente profile dentro de la carpeta app, contiene toda la lógica del perfil.
  - Creación de un Guard y un Resolver para prevenir que los usuarios entren a esa ruta si no poseen una sesión activa.
  - Creación de los servicios relacionados al usuario en la carpeta core.
- **Olvido de contraseña y reseteo de contraseña.**
  - Creación de métodos para la obtención de datos de los usuarios en los servicios.
  - Se crea el componente de olvido de contraseña en la carpeta auth.
- **Navegación de la aplicación.**
  - Creación del componente tabs dentro de la carpeta app, contiene toda la lógica de la navegación.
- **Internacionalización I18n.**

- Se importa la librería ngx-translate
- Se crea la carpeta i18n en assets que contiene los archivos JSON con las traducciones.
- Se crean los servicios en la carpeta core, correspondiente a la llamada de los archivos de traducciones.
- **Envío de mensajes.**
  - Se importa la librería QuillJs.
  - Se crea el componente de envío de mensajes en la carpeta app, que contiene toda la lógica correspondiente del envío de mensajes.
  - Se crean los servicios en la carpeta core, correspondiente a las funciones de envío de mensajes.
- **Buscador.**
  - Se implementa un pipe en la carpeta shared, que posee la funcionalidad del buscador y se podrá inyectar en cualquier componente que lo necesite.
- **Componente de mensajes enviados.**
  - Se crea el componente de mensajes enviados en la carpeta app, que contiene toda la lógica correspondiente del envío de mensajes.
  - Se crean los servicios en la carpeta core, correspondiente a las funciones de mensajes enviados.
- **Componente de detalle de mensaje.**
  - Se crea un pipe en la carpeta shared, que posee la funcionalidad para saber si un mensaje es o no publicable.
  - Se crea un componente compartido en la carpeta shared denominado message-detail. Este componente será utilizado tanto para el detalle de los mensajes enviados, recibidos y los mensajes del perfil del usuario.
- **Componente de mensajes recibidos.**
  - Se crea el componente de mensajes recibidos en la carpeta app, que contiene toda la lógica correspondiente del envío de mensajes.

- Se crean los servicios en la carpeta core, correspondiente a las funciones de mensajes recibidos.
- **Estadísticas de los mensajes en el perfil de usuario.**
  - Se implementa el conteo de votos positivos y negativos tanto de mensajes enviados como de recibidos en el perfil de usuario.
- **Funcionalidad de votos positivos y negativos en los mensajes.**
  - Se implementa la funcionalidad de *likes* y *dislikes* de mensajes enviados y recibidos en el perfil de usuario.
- **Funcionalidad del administrador para cambiar cargos en trabajadores y supervisores.**
  - Se implementa el componente para cambiar cargos de trabajadores y supervisores llamado change-jobs en la carpeta app.
  - Se crean los servicios en la carpeta core, correspondiente a las funciones de cambio de cargo.
- **Funcionalidad del administrador para asignar trabajadores a supervisores.**
  - Se implementa el componente para cambiar cargos de trabajadores y supervisores llamado workers-to-supervisors en la carpeta app.
  - Se crean los servicios en la carpeta core, correspondiente a las funciones de asignación de trabajadores a supervisores.
- **Funcionalidad del supervisor para ver el perfil de sus trabajadores con los mensajes tanto públicos como privados.**
  - Se implementa el componente para visualizar los mensajes tanto públicos como privados del perfil, denominado employees-in-charge en la carpeta app.
- **Componente de Ranking de usuarios.**
  - Se crea el componente de ranking de trabajadores en la carpeta app, que contiene toda la lógica correspondiente al ranking de usuarios.
  - Se crean los servicios en la carpeta core, correspondiente a las funciones del ranking de trabajadores.

### **Fase de Pruebas (16/10/2020 - 2/10/2020)**

Durante esta fase se implementaron pruebas de aceptación con la finalidad de comprobar que la aplicación cumple con los objetivos para la que fue desarrollada. Durante las pruebas se encontraron errores en casos bordes que se fueron solucionando a medida que se interactuaba con la aplicación. La aplicación fue probada por diferentes usuarios. También se estableció una reunión con el tutor empresarial para realizar las pruebas simultáneamente.

Se hicieron arreglos visuales y se refactorizó el código, tomando en cuenta todas las sugerencias del tutor empresarial, por lo que terminando esta fase se logró el objetivo planteado.



## RESULTADOS

Como resultado obtenido, se presenta una aplicación que cumple con lo solicitado por el tutor empresarial. En el enfoque del modelo de negocio, la aplicación logra registrar, calificar y llevar un registro de los comentarios de los trabajadores y supervisores sobre el desempeño de sus actividades dentro de la empresa, permite calificar estos mensajes con el fin de ver si el desempeño es positivo o negativo, los supervisores pueden llevar un control, hacer un análisis del desempeño de sus trabajadores asignados evitando llevar la carga adicional de trabajo y contabilizar los mensajes a través de estadísticas para llevar un registro del desempeño general de la empresa.

En el punto de vista tecnológico, la aplicación desarrollada utiliza tecnologías actualizadas en el ámbito del desarrollo web. Adopta un modelo por componentes, lo cual hace que la aplicación sea modular, mantenible en el tiempo, reutilizable y con funcionalidades débilmente acopladas.

La solución fue desarrollada siguiendo los estándares y mejores prácticas definidas en las páginas oficiales y documentación de los frameworks utilizados, por tal motivo la aplicación logra ser de fácil adaptación para futuros desarrolladores al momento de actualizar o agregar alguna funcionalidad.

## **CONCLUSIONES Y RECOMENDACIONES**

Las soluciones tecnológicas son un paso indispensable para el desarrollo empresarial. En la actualidad, muchos procesos en las organizaciones son automatizados con la finalidad de establecer un mejor desempeño en las empresas, realizando tareas en el menor tiempo posible y con mejores resultados.

Las aplicaciones han logrado resolver y agilizar no solo problemas complejos que requieren de la ayuda tecnológica, también ha podido dar solución a situaciones cotidianas en la vida de las personas. Para las empresas de alta demanda de trabajadores, la solución presentada cumple un papel fundamental el cual se centra en dar a los supervisores y trabajadores una herramienta para valorar el trabajo y desempeño dentro de una organización.

A partir de la automatización de estos procesos, se logra mantener un historial del desempeño de los trabajadores, manteniendo un orden dentro de la empresa y produciendo un incentivo entre los empleados para mejorar en sus tareas laborales. También ayuda a los supervisores a mantener un historial de las actividades que se realizan dentro de la empresa, disminuyendo la inestabilidad que se presenta al hacer las evaluaciones hacia los trabajadores en los que cada supervisor tiene a cargo.

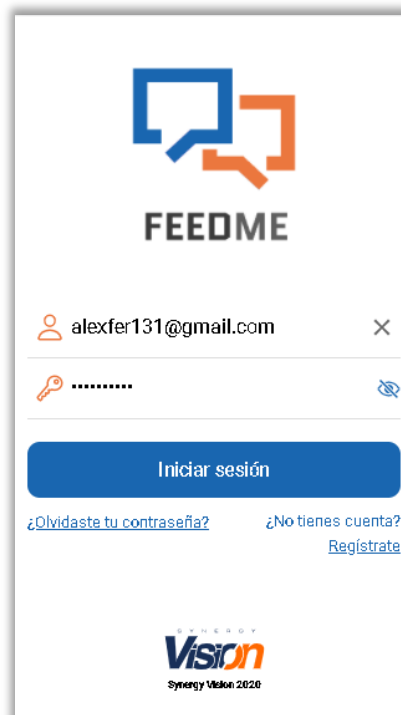
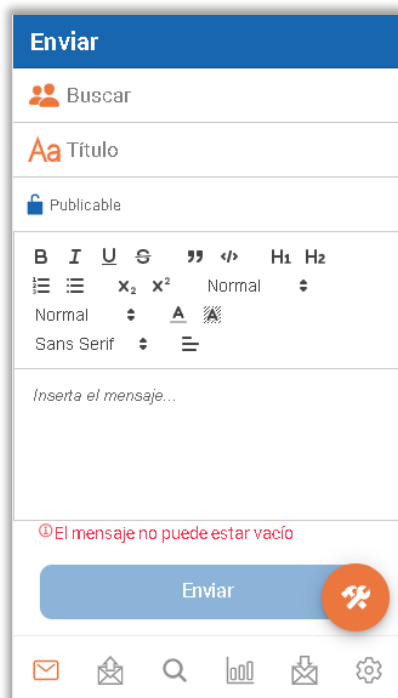
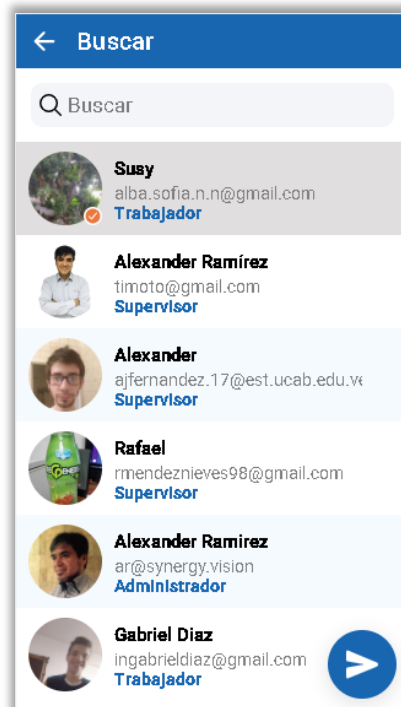
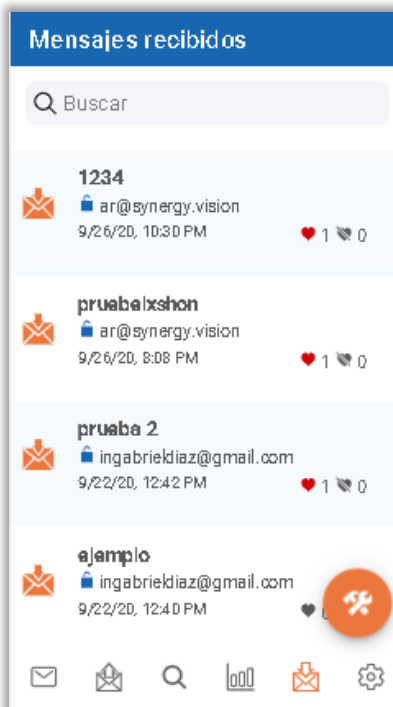
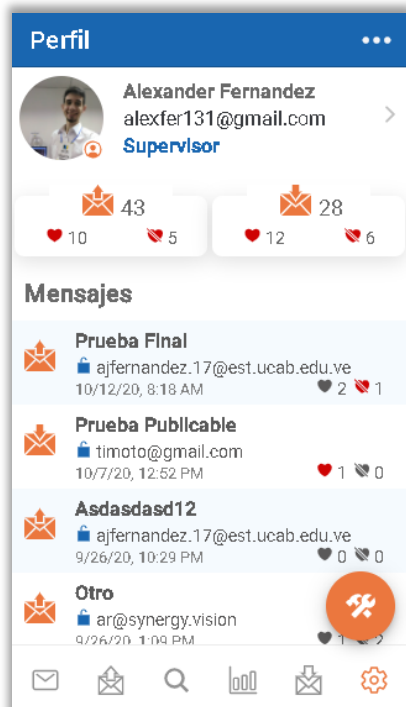
En un futuro, se recomienda que este tipo de soluciones se implementen en cualquier organización y a cualquier nivel empresarial, debido a que resuelven problemas que muchas veces llevan al descontento y producen un descenso en la producción en las actividades de una empresa.

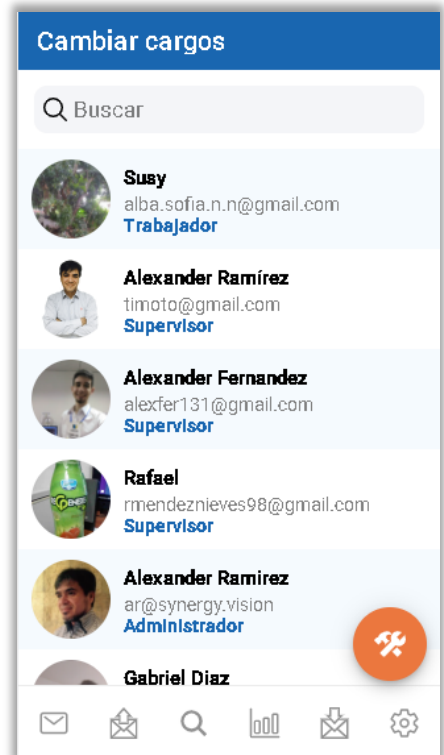
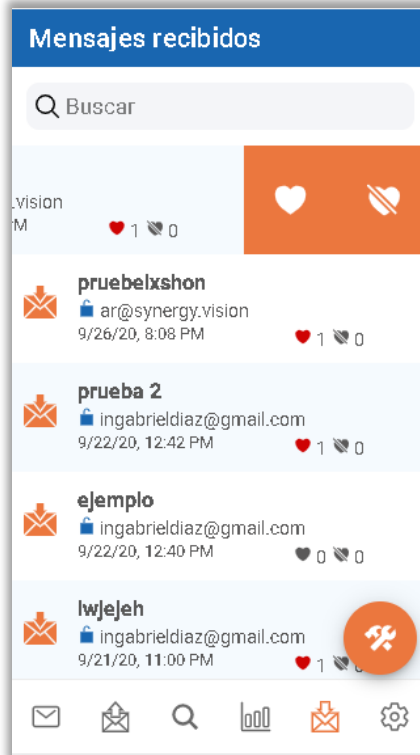
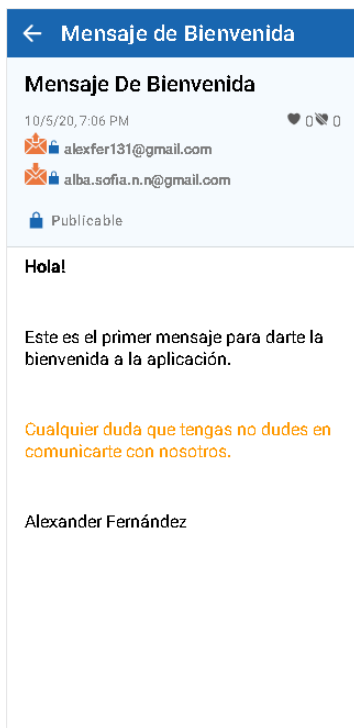
## REFERENCIAS BIBLIOGRÁFICAS

1. Google. *Un Framework de desarrollo móvil y de escritorio*. Angular io. Recuperado el 26 de Julio del 2020 de <https://angular.io/>
2. Papa, J. *Guía de estilo de codificación de angular*. Angular io. Recuperado el 26 de Julio de 2020 de <https://angular.io/guide/styleguide>
3. Google. *Documentación de Firebase*. Firebase. Recuperado el 29 de Julio del 2020 de <https://firebase.google.com/docs/>
4. Drifty. *Ionic Framework*. Ionic. Recuperado el 30 de Julio del 2020 de <https://ionicframework.com/docs/>
5. GitBook. *Ionic 5 Full Starter App*. Ionic 5 Full Starter App. Recuperado el 30 de Julio del 2020 de <https://ionic-5-full-starter-app-docs.ionicthemes.com/>
6. StartApp Labs. *Ionic Tutorial: Build a complete mobile app with Ionic Framework*. Ionic Themes. Recuperado el 30 de Julio del 2020 de <https://ionicthemes.com/tutorials>
7. SauceLabs. *Quill, Your powerful rich text editor*. QuillJs. Recuperado el 2 de agosto del 2020 de <https://quilljs.com/docs/quickstart/>
8. GitBook. *Ngx-Datatable*. ngx-datatable. Recuperado el 4 de agosto del 2020 de <https://swimlane.gitbook.io/ngx-datatable/>

## APÉNDICES

### Imágenes de la aplicación





## Issues / historias de usuario



aaronmirez

commented on 26 Aug

- ✓ Diseñar el flujo de selección de un usuario y envío de un mensaje.
- ✓ Diseñar la pantalla de envío de mensaje con opción de que el mensaje sea publicable.
- ✓ Agregar una opción de configuración en el perfil para decidir la opción por defecto de un mensaje (Publicable o no publicable por defecto)
- ✓ Manejar el formato markdown para enviar los mensajes y para mostrarlos en pantalla.
- ✓ Manejar los datos del mensaje y su visualización como el emisor, fecha, título, publicable o no y contenido en markdown.



aaronmirez

commented on 19 Aug

- ✓ Gestión el cambio de clave a través de una pantalla de perfil.
- ✓ Gestión del olvido de contraseña utilizando los mecanismos provistos por Firebase.



aaronmirez

commented on 19 Aug

- ✓ Agregar pantalla de registro con Nombre, Email y Clave con integración a Firebase.



**aaramirez**

commented on 26 Aug



- ✓ Diseñar la pantalla de mensajes recibidos, resaltando los leídos y los no leídos.
- ✓ Dar la opción para publicar un mensaje en el perfil del receptor.
- ✓ Desarrollar la funcionalidad de calificar el mensaje como positivo o negativo.



**aaramirez**

commented on 9 Sep

- ✓ Diseñar y desarrollar una pantalla de visualización del ranking de los trabajadores utilizando distintos criterios, entre ellos, Ranking por mensajes enviados o recibidos, Ranking por calificaciones positivas realizadas o recibidas, Cruce de criterios.



**aaramirez**

commented on 2 Sep



- ✓ Si el usuario es supervisor tiene la opción de ver las personas que están a su cargo.
- ✓ El Supervisor puede ver los mensajes enviados y recibidos de un trabajador cualquiera tanto públicos como privados.
- ✓ El Supervisor puede calificar los mensajes, tanto públicos como privados.



**aaramirez**

commented 19 days ago

- ✓ Revisar y ajustar la documentación del desarrollo, README.md
- ✓ Elaborar el informe de la pasantía y dejar el documento fuente y en formato PDF.
- ✓ Elaborar una presentación (láminas) del proyecto.
- ✓ Capturar en video los flujos de la aplicación que demuestren toda la funcionalidad implementada.
- ✓ Sesión de pruebas conjunta.
- ✓ Presentación a otro integrante del equipo.



**aaramirez**

commented on 9 Sep

- ✓ Mostrar en la pantalla de perfil de forma visualmente atractiva los mensajes enviados y recibidos.
- ✓ Mostrar en pantalla la suma de las calificaciones de los mensajes enviados y recibidos en la pantalla de perfil.
- ✓ Cada vez que se muestra un mensaje se debe mostrar la cantidad de calificaciones positivas y no positivas de forma visual.



**aaramirez**

commented on 2 Sep

- ✓ Desarrollar la pantalla de perfil de un usuario donde se pueden visualizar los mensajes recibidos que son públicos.
- ✓ Mostrar la lista de los mensajes enviados y recibidos con una leyenda visual. Con opción de calificarlos de manera visual.
- ✓ Mostrar la cantidad de mensajes enviados y recibidos con los likes y dislikes asociados de forma visual.
- ✓ Mostrar los datos del trabajador.
- ✓ Permitir el cambio de contraseña.
- ✓ Mostrar de forma visual las estadísticas de cada mensaje.



**aaramirez**

commented on 19 Aug

- ✓ Configurar la gestión de credenciales en Firebase para que utilice Email y Clave.
- ✓ Configurar las reglas de acceso a la información en Firebase.
- ✓ Agregar a los administradores a la Base de Datos.



aaronmirez commented 26 days ago • edited by alexjose131 ▾

Contributor 😊 ...

- ✓ Revisar la alineación de todos los textos, íconos, títulos y título de la pantalla en cada pantalla para que queden igual a lo logrado en la pantalla de perfil.
- ✓ Cambiar los títulos con los colores y tamaños logrados en la pantalla de perfil.
- ✓ Cambiar el tamaño del ícono de mensaje enviado y recibido al tamaño logrado en la pantalla de perfil.
- ✓ Diferenciar el tamaño del texto y el color del título de un mensaje haciéndolo más grande como lo logrado en la pantalla de perfil.
- ✓ Cambiar el tamaño de los íconos y dejar en una fila los mensajes recibidos y en otra los mensajes enviados con sus likes. Probar colocando un fondo distintos para diferenciar.
- ✓ Hacer que esa homogéneo el look de todas las pantallas. Evitar espacios entre los items que generen vacíos.
- ✓ Verificar la alineación a la derecha y a la izquierda.
- ✓ Evitar el uso del negro para el color de las letras y probar con dimgray (Tonos de gris oscuro).
- ✓ Verificar la funcionalidad de toda la aplicación.
- ✓ Probar que el ícono de los mensajes enviados y recibidos tengan un color distintos.
- ✓ Colocar una pantalla de splash e ícono de la aplicación.
- ✓ Los títulos de los mensajes deben tomar a lo sumo dos líneas y finalizar con ellipsis.



aaronmirez commented on 26 Aug • edited by alexjose131 ▾

Contributor 😊 ...

- ✓ Estudiar otras aplicaciones para captar funcionalidades similares que nos ayuden a darle mayor atractivo a los flujos diseñados hasta ahora.
- ✓ Agregar colores a los elementos visuales como los iconos y botones.
- ✓ Evitar que los componentes de la pantalla se muevan cuando hay formas con validaciones, el espacio donde aparecen las validaciones se debe reservar y evitar que se muevan los elementos cuando hay algún error de validación.
- ✓ Implementar i18n para mantener los textos de toda la aplicación en un lugar centralizado.
- ✓ Revisar los textos y colocar en mayúscula sólo la primera letra.
- ✓ Revisar los títulos de las pantallas y evitar que se corten en el iPhone 5 SE.
- ✓ Diseñar la forma como se va a mostrar las estadísticas en el perfil del usuario.
- ✓ Dejar como pantalla por defecto, inicial, la pantalla de mensajes recibidos.