



**UNIVERSIDAD DE GUADALAJARA**  
**CENTRO UNIVERSITARIO DE CIENCIAS**  
**EXACTAS E INGENIERÍAS**

**DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN**

**DEPARTAMENTO DE CIENCIAS**  
**COMPUTACIONALES.**

**INGENIERÍA EN INFORMÁTICA.**

**INGENIARIA EN SOFTWARE II**

**SECCIÓN D03**

**CLAVE I5900.**

**MANUAL TECNICO.**

**Equipo 7.**

**Alan Vidal Ocampo Orozco**  
**Miguel Ángel Sandoval Chávez**  
**Luis Enrique Rivera Vargas**

**FECHA: 20/08/2022**

**Prof. ZOILA LILIANA HERRERA LUJAN**

OBSERVACIONES:

---

---

# Índice

## Contenido

Índice .....	2
Tabla De Ilustraciones. ....	4
Presentación.....	5
Resumen.....	5
Objetivo.....	5
Finalidad del manual. ....	5
Introducción. ....	6
1. Aspectos técnicos.....	7
1.1 Herramientas utilizadas para el desarrollo. ....	7
1.1.1 Visual Studio Code.....	7
1.1.2 Node.JS.....	7
1.1.3 React.....	7
1.1.4 Firebase. ....	7
1.1.5 GitHub .....	8
1.2 Frameworks, lenguajes o librerías utilizadas. ....	8
1.2.1 HTML .....	8
1.2.2 CSS.....	8
1.2.3 JavaScript.....	8
1.2.4 React.....	8
2. Aspecto Técnico Desarrollo del sistema. ....	9
2.1 Instalación del proyecto, local y deploy. ....	9
3. Diagramas de modelamiento. ....	16
Diagrama de casos de uso.....	16
Diagrama de clases:.....	17
Diagrama de secuencia. ....	18
Diagrama de estados.....	19
Diagrama de colaboración. ....	20
4. Requisitos recomendados.....	21
Requisitos de hardware.....	21

Requisitos de software .....	21
Bibliografías.....	22

## Tabla De Ilustraciones.

Ilustración 1. Código para crear una app. ....	9
Ilustración 2. Comando iniciar aplicación. ....	9
Ilustración 3. Mensaje de salida, al inicial la app. ....	9
Ilustración 4. Comando instalar react-dom.....	10
Ilustración 5. Instalación de iconos. ....	10
Ilustración 6. Comandos Firebase. ....	10
Ilustración 7. Confirmar autenticación. ....	11
Ilustración 8. Configurar Firestore. ....	12
Ilustración 9. Configurar el SDK.....	12
Ilustración 10. Sustitucion de codigo. ....	13
Ilustración 11. Firebase Login. ....	13
Ilustración 12. Firebase Init. ....	14
Ilustración 13. Firebase Build.....	14

## **Presentación.**

El siguiente manual se ha desarrollado con la finalidad de dar a conocer la información necesaria para realizar mantenimiento, instalación y exploración del software ClonWhatsapp, el cual consta de un chat en tiempo real.

El manual ofrece la información necesaria de ¿cómo está realizado el software? para que la persona (Desarrollador en el framework React) que quiera editar el software lo haga de una manera apropiada, dando a conocer la estructura del desarrollo del aplicativo.

## **Resumen.**

El manual detalla los aspectos técnicos e informáticos del software ClonWhatsapp con la finalidad de explicar la estructura del aplicativo al personal que quiera administrarlo, editarlo o configurarlo.

La siguiente guía se encuentra dividida en las herramientas que se usaron para la creación del software con una breve explicación paso a paso, El aplicativo web maneja diferentes funcionalidades el cual requieren de hardware y software el cual se explicará que funcionamiento realiza cada uno de ellos, dando sugerencias para el debido uso del sistema de información.

## **Objetivo.**

Dar a conocer el uso adecuado del software ClonWhatsapp en aspectos técnicos de manera descriptiva e ilustrada sobre los componentes y funcionalidades que conforman el buen funcionamiento del sistema de información

## **Finalidad del manual.**

La finalidad de este manual técnico es instruir a la persona que quiera administrar, editar o configurar el software ClonWhatsapp usando las debidas herramientas.

## Introducción.

El manual se realiza con el fin de detallar el software ClonWhatsapp en términos técnicos para que la persona que vaya a administrar, editar o configurar el aplicativo lo haga de una manera apropiada. El documento se encuentra dividido en las siguientes secciones:

- **ASPECTOS TEÓRICOS:** Se darán a conocer conceptos, definiciones y explicaciones de los componentes del aplicativo desde un punto de vista teórico para mayor entendimiento por parte del lector sobre el funcionamiento del sistema de información y herramientas.
- **ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA:** Corresponde a la instrucción al lector sobre los componentes del aplicativo desde una perspectiva técnica en los aspectos de almacenamiento de datos, estructura del desarrollo y recomendaciones del uso debido del aplicativo.
- **REQUERIMIENTOS DEL SOFTWARE:** Detalla los requerimientos básicos necesarios para el funcionamiento del software.

# 1. Aspectos técnicos.

Se recomienda que el siguiente manual sea manipulado únicamente por la persona que quiera administrar, editar o configurar el software ClonWhatsapp para velar por la seguridad de los datos que se almacenan en la base de datos ya que pueden ser usados para otros fines.

## 1.1 Herramientas utilizadas para el desarrollo.

En esta sección se procede a explicar las herramientas informáticas empleadas para el desarrollo del aplicativo.

### 1.1.1 Visual Studio Code.

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. La versión utilizada para la realización de este software es gratuita.

### 1.1.2 Node.JS.

Node.js y NPM de una manera breve y sencilla. Node.js es un entorno de ejecución de código abierto que utiliza código JavaScript en un servidor, sin necesidad de utilizar navegadores web.

### 1.1.3 React.

React (también llamada React.js o ReactJS) es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. En el proyecto hay más de mil desarrolladores libres.

React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar. React sólo maneja la interfaz de usuario en una aplicación; React es la Vista en un contexto en el que se use el patrón MVC (Modelo-Vista-Controlador) o MVVM (Modelo-vista-modelo de vista). También puede ser utilizado con las extensiones de React-based que se encargan de las partes no-UI (que no forman parte de la interfaz de usuario) de una aplicación web.

### 1.1.4 Firebase.

Firebase básicamente es una plataforma móvil diseñada y creada por Google, teniendo como principal función desarrollar y facilitar la creación de aplicaciones para dispositivos móviles que cuenten con una alta calidad a pesar de su rápida elaboración; esto con la finalidad de que se

pueda incrementar la base de datos de usuarios y de esta manera incrementar la monetización de dicha app (ganar más dinero).

Esta plataforma se encuentra alojada en la nube y, por ende, está disponible para diferentes plataformas como Android, iOS, y web. Así mismo, cuenta con diversas funciones para que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades.

### **1.1.5 GitHub**

GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc.

## **1.2 Frameworks, lenguajes o librerías utilizadas.**

Principalmente este software esta realizado con el framework react combinado con JavaScript, también HTML y CSS para el diseño.

### **1.2.1 HTML**

HTML, siglas en inglés de HyperText Markup Language, hace referencia al lenguaje de marcado para la elaboración de páginas web.

### **1.2.2 CSS**

CSS, en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado

### **1.2.3 JavaScript**

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

### **1.2.4 React**

React es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.

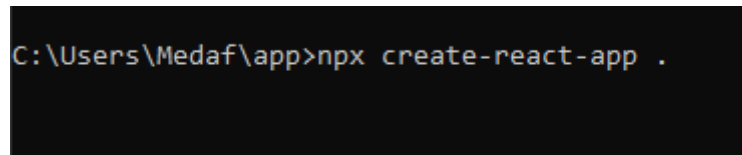


## 2. Aspecto Técnico Desarrollo del sistema.

En la siguiente sección se procede a realizar una descripción detallada sobre los aspectos técnicos del aplicativo, relacionado con la instalación de las herramientas necesarias para realizar modificaciones requeridas de manera ordenada.

### 2.1 Instalación del proyecto, local y deploy.

Antes de empezar todo, asegúrate de tener instalado Node.js ya que es importante para el desarrollo del software.

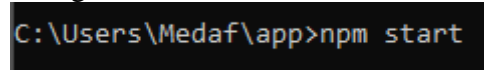


```
C:\Users\Medaf\app>npx create-react-app .
```

**Ilustración 1. Código para crear una app.**

Primero tenemos que abrir una consola de comandos, ubicarnos en la carpeta donde se instalara la app de React con el siguiente código: “npx create-react-app .”, después de insertar este comando tendremos que esperar un poco a que se instale.

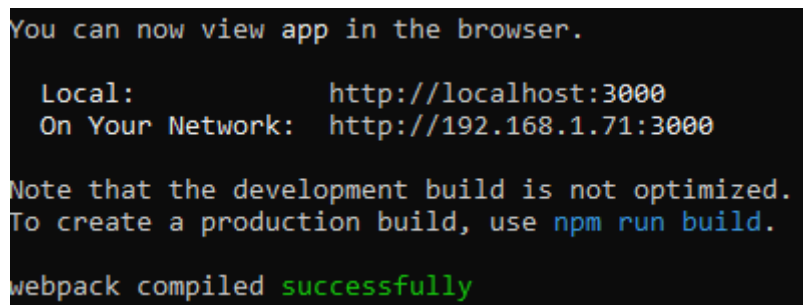
Continuamos dentro de nuestra consola, si queremos ver si nuestra aplicación se instaló de manera adecuada, simplemente introducimos el comando: “npm start” nos saldrá una ventana del navegador con el contenido de nuestra aplicación,



```
C:\Users\Medaf\app>npm start
```

**Ilustración 2. Comando iniciar aplicación.**

Y nos saldrá un mensaje parecido a este cada vez que inicialicemos la app de esta manera (local)



```
You can now view app in the browser.

Local:            http://localhost:3000
On Your Network:  http://192.168.1.71:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

**Ilustración 3. Mensaje de salida, al iniciar la app.**

El siguiente paso importante es instalar una librería de react con el siguiente comando:” npm i react-router-dom”,

```
C:\Users\Medaf\app>npm i react-router-dom
```

**Ilustración 4. Comando instalar react-dom.**

El siguiente paso importante es instalar el paquete de iconos MUI con los siguientes comandos:

“npm install @mui/icons-material”

“npm install @mui/icons-material @mui/material @emotion/styled @emotion/react”.

También será necesario instalar emojis con el sig. comando: “npm i emoji-picker-react”

```
C:\Users\Medaf\app>npm i emoji-picker-react
```

```
C:\Users\Medaf\app>npm install @mui/material
```

```
C:\Users\Medaf\app>npm install @mui/icons-material @mui/material @emotion/styled @emotion/react
```

**Ilustración 5. Instalación de iconos.**

Después de haber instalado los iconos, ahora vamos con la instalación y configuración de firebase, primero pondremos este comando: “npm install firebase”

Y también necesitaremos

“npm install -g firebase-tools”.

```
C:\Users\Medaf\app>npm install firebase
```

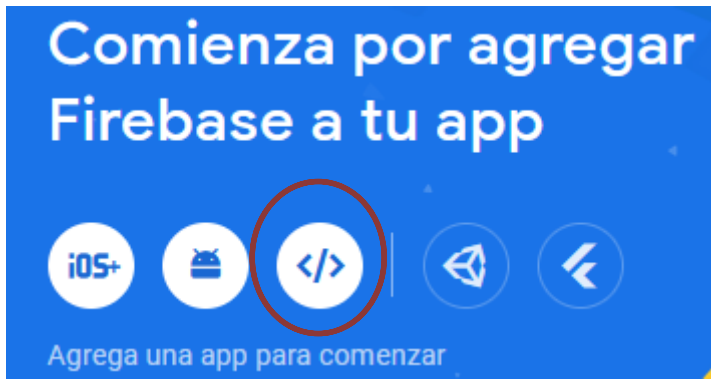
```
C:\Users\Medaf\app>npm install -g firebase-tools
```

**Ilustración 6. Comandos Firebase.**

Cuando hayamos hecho todos estos pasos vamos a descargar el código fuente de mi repositorio de GitHub de la Branch 3: <https://github.com/AlanOcampo/Proyecto---ClonWhatsapp/tree/V3-Firebase-Whats> y una vez hayamos descargado todos los archivos, vamos a copiar la carpeta “src” y pegarla en la carpeta de nuestra aplicación.

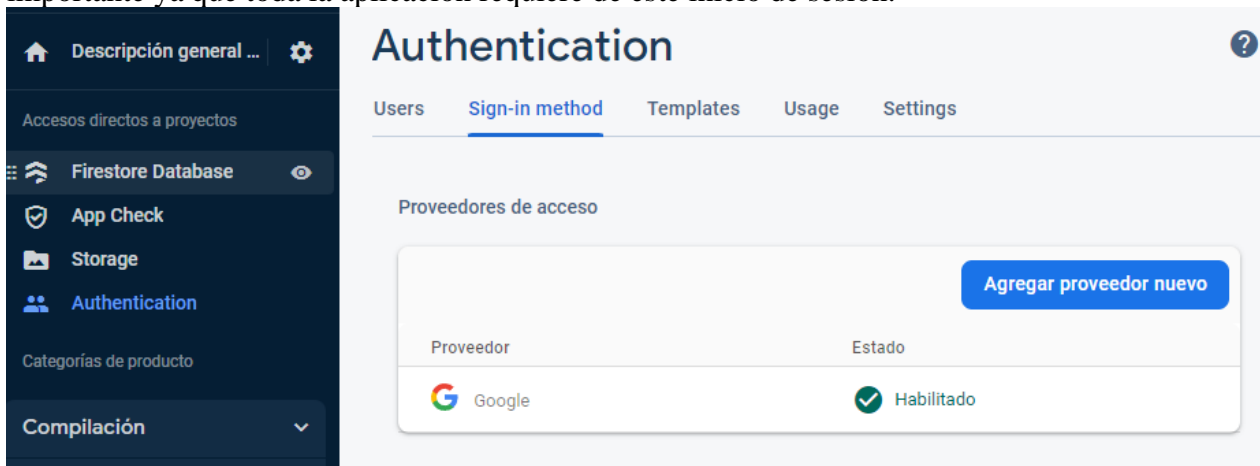
Después de haber instalado todos los archivos necesarios y también introducido los comandos tenemos todo lo necesario para continuar con la configuración de la firebase

Como nuestro software está alojado en un servidor de firebase, tendremos que hacernos una cuenta en firebase, y para ello tendremos que ir a la página de firebase dentro de nuestro navegador, <https://console.firebase.google.com/u/0/?hl=es>, creamos nuestra cuenta y una vez dentro crearemos un nuevo proyecto, con el nombre del software: “clone-whatsapp” o como les parezca apropiado, y le damos siguiente hasta crear el proyecto, una vez que esta creado nuestro proyecto, le damos tenemos que crear una app web, para esto tendremos que seleccionar la opción de app web, de igual manera completamos la información del formulario como con el proyecto, sin olvidar habilitar el hosteo de la aplicación.



**Ilustración 8. Crear aplicación web, en firebase.**

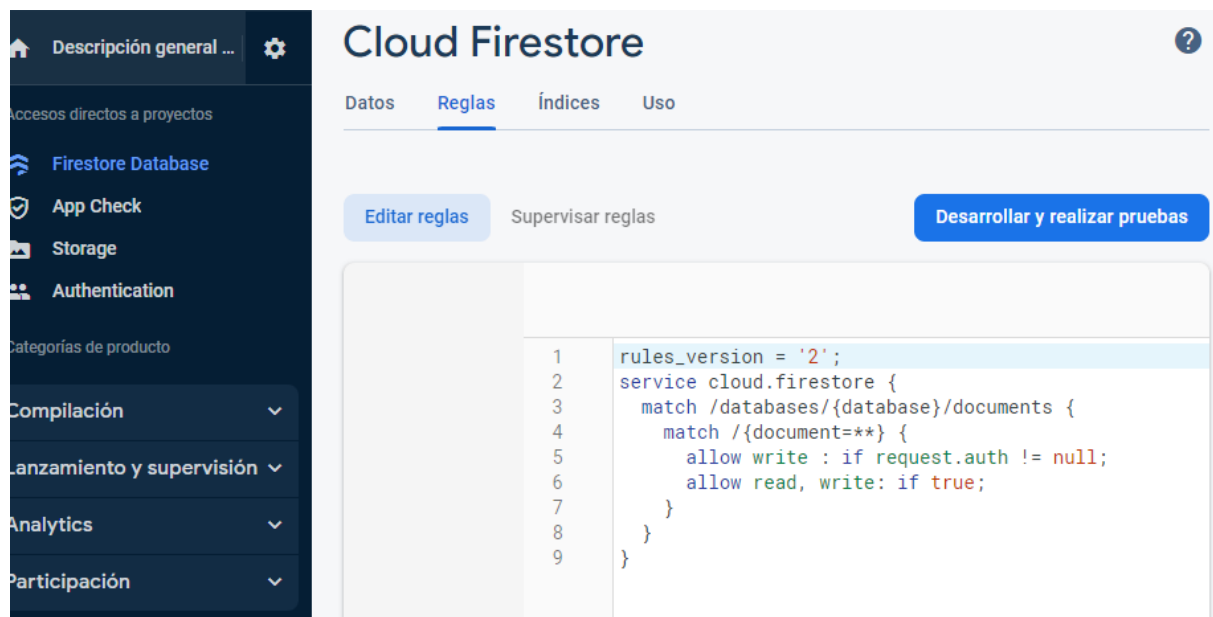
Una vez hayamos creado la aplicación web podremos continuar con la configuración, primero lo que tendremos que hacer es habilitar el inicio de sesión dentro de nuestra aplicación web, para ello iremos a al apartado de autenticación, y habilitar el inicio de sesión con Google, esto es muy importante ya que toda la aplicación requiere de este inicio de sesión.



**Ilustración 7. Confirmar autenticación.**

Después de habilitar el inicio de sesión con Google, iremos al apartado de Firestore Database, también tendremos que habilitarlo, después de hacerlo, iremos a la parte de las reglas y las modificaremos. Y pondremos las siguientes líneas:

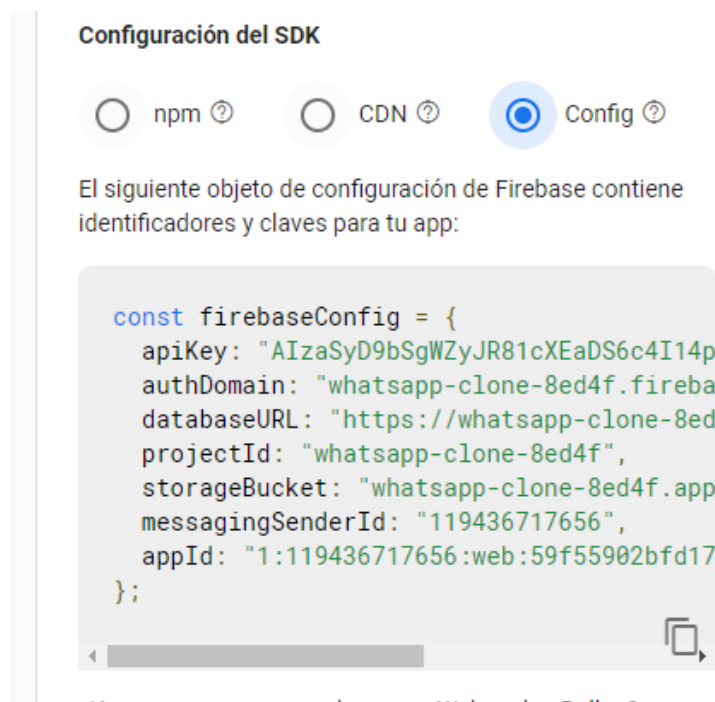
```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow write : if request.auth != null;
      allow read, write: if true;
    }
  }
}
```



**Ilustración 8. Configurar Firestore.**

Una vez hayamos hecho estos procesos ya tenemos configurada la firebase, ahora vamos a la configuración del proyecto e iremos a la parte inferior de la pestaña.

Aquí nos saldrá una ventana con algunos comandos, seleccionamos la opción que dice “Config” y copiamos el texto que viene ahí, atentos a este paso que estos comandos son variados, depende de tu aplicación lo que viene en esta parte.



**Ilustración 9. Configurar el SDK.**

Este texto lo copiaremos y lo sustituiremos, primero iremos a la carpeta contenedora de nuestra app, aquí abrimos la carpeta src que es donde se encuentra el código de todo el software, una vez aquí abriremos el archivo llamado firebase.js y dentro del mismo sustituiremos la parte de firebaseConfig por lo que teníamos copiado anteriormente.

```

1  import firebase from 'firebase/compat/app';
2  import 'firebase/compat/auth';
3  import 'firebase/compat/firestore';
4
5  const firebaseConfig = {
6    apiKey: "AIzaSyDvt7IkcTRi6yN8EqxiZ6TIUA_d8YzoK_k",
7    authDomain: "clone-whatsapp-d7c6a.firebaseio.com",
8    projectId: "clone-whatsapp-d7c6a",
9    storageBucket: "clone-whatsapp-d7c6a.appspot.com",
10   messagingSenderId: "505642313111",
11   appId: "1:505642313111:web:b851ca7e1aeb0bb3b290d2",
12   measurementId: "G-4Q8VCBKVXN"
13 };
14
15 const app = firebase.initializeApp(firebaseConfig);
16
17 const auth = firebase.auth();
18
19 const db = app.firestore();
20
21 const googleProvider = new firebase.auth.GoogleAuthProvider();
22
23 export { auth, googleProvider };
24
25 export default db;

```

**Ilustración 10.** Sustitución de código.

Una vez hecho esto tendríamos hecho toda la modificación de nuestro proyecto, ahora continuaremos con el deploy.

Una vez hayamos hecho todos los pasos anteriores, tendremos que abrir la consola de comandos, una vez aquí pondremos los siguientes códigos:

“firebase login” e iniciamos sesión con la cuenta donde hayamos configurado nuestra aplicación web.

```

C:\Users\Medaf\app>firebase login
Already logged in as [email]

```

**Ilustración 11.** Firebase Login.

Después introduciremos:

“firebase init” y seguiremos las siguientes instrucciones, primero le daremos “y”, después seleccionamos la opción “Hosting: Configure files for firebase hosting” y le damos un espacio y

después enter, después seleccionamos nuestra aplicación (si solo tenemos una se selecciona automáticamente), a las dos opciones “y” y a la siguiente “n”

```
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space

=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

i Using project clone-whatsapp-d7c6a (clone-whatsapp)

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? y
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
? File y/index.html already exists. Overwrite? No
i Skipping write of y/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...

+ Firebase initialization complete!

C:\Users\Medaf\app>
```

**Ilustración 12. Firebase Init.**

Después de hacer esto introduciremos el siguiente comando:

“npm run build”

```
C:\Users\Medaf\app> npm run build
```

**Ilustración 13. Firebase Build**

Aquí es cuando nos pueden salir errores, al momento de hacer este “build”, si tenemos algún error y no compila favor de comunicarse al siguiente correo, [alan.ocampo4510@alumnos.udg.mx](mailto:alan.ocampo4510@alumnos.udg.mx), o de igual manera si quieren solucionar algún error que pudiera surgir, pueden ejecutar el programa de manera local, simplemente introduciendo el siguiente comando:

“npm start” con este comando inicializaremos la aplicación web de manera local, pero si han seguido los pasos debería funcionar de manera correcta, aquí también saldrán los errores que salieron en el build (en el caso de que los hubiera) pero aquí sale en pantalla el error que está causando este fallo.

Una vez no haya errores iremos con el deploy por lo que introduciremos el siguiente comando: “firebase deploy” una vez hecho esto tendremos finalizada la instalación del software para sus modificaciones, actualizaciones, etc.

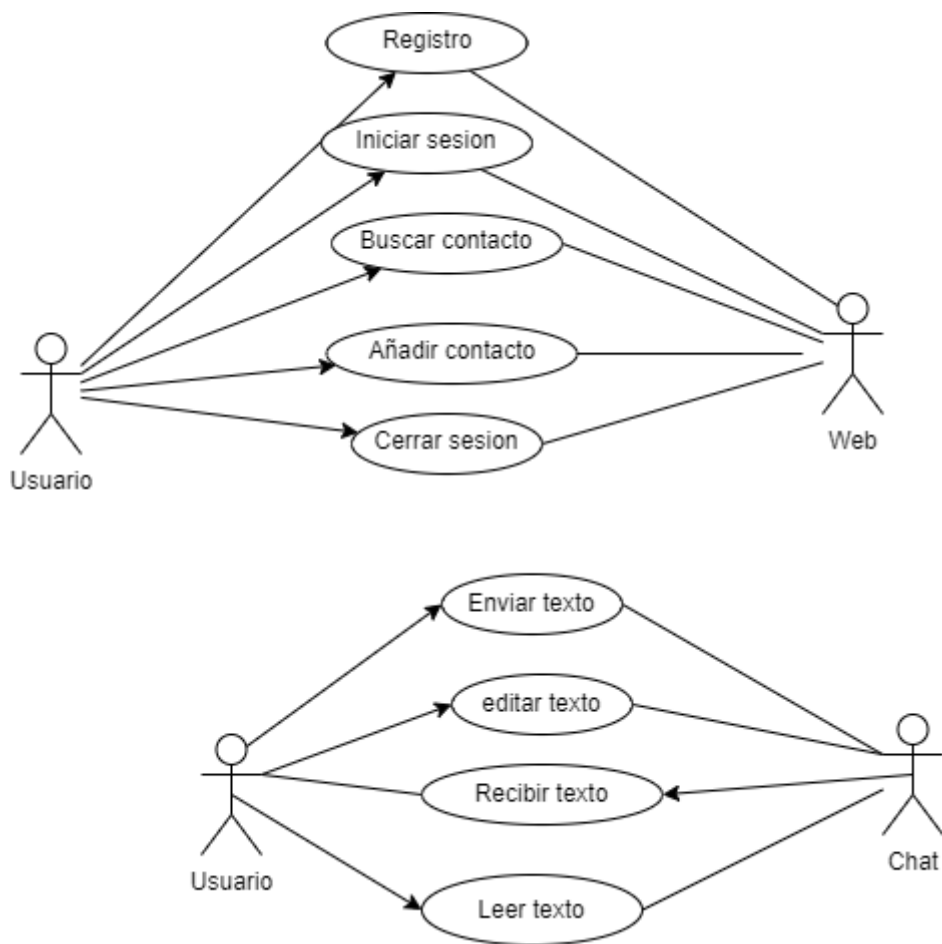
Y con esto terminaríamos esta parte de instalación, y también se puede manejar de manera local una vez hayamos hecho todos los pasos(excluyendo el deploy).

Y con esto concluye esta guía para poder instalar este software, si tienes alguna duda o error como ya dije antes se pueden comunicar conmigo a través del correo:

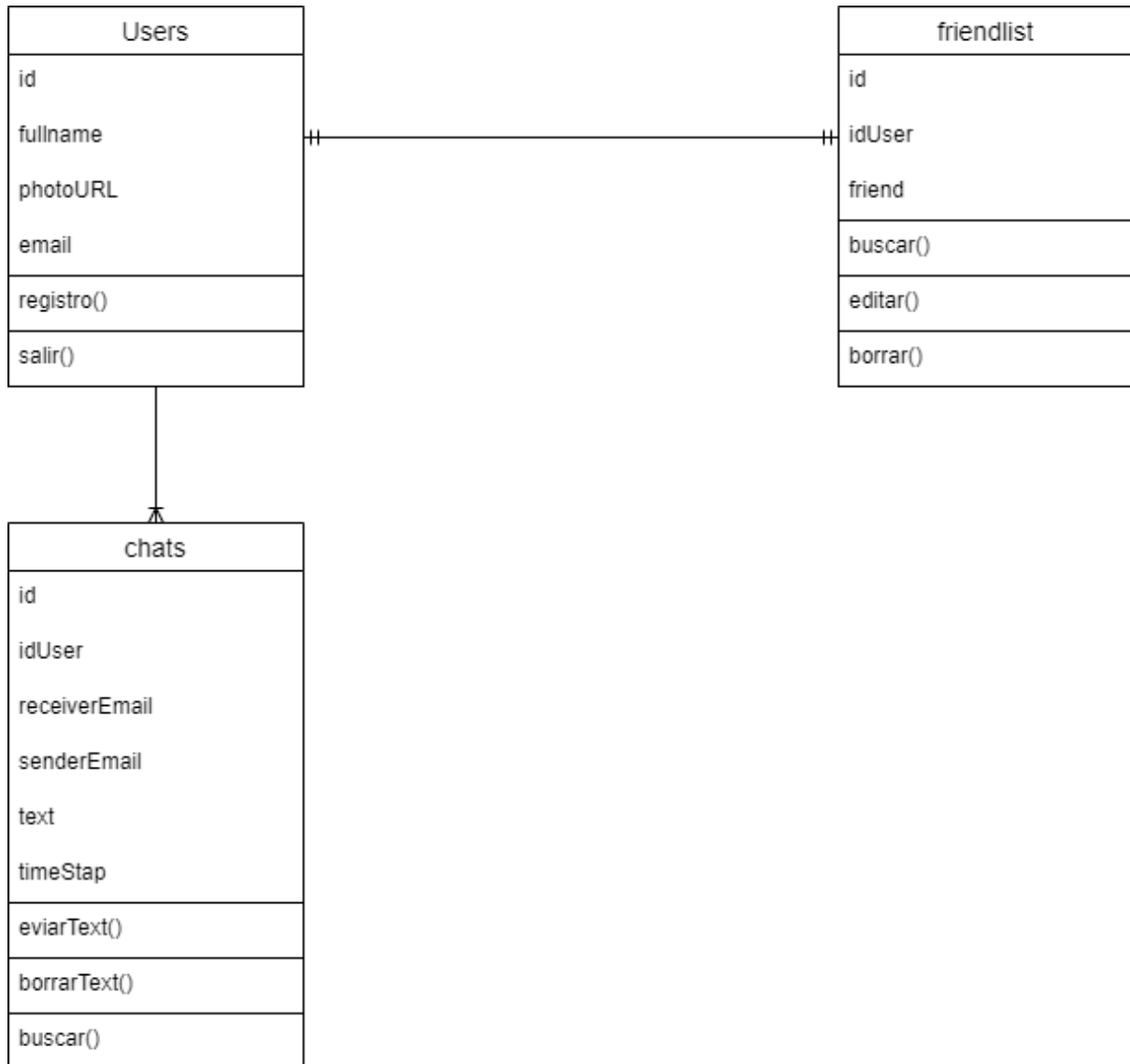
[alan.ocampo4510@alumnos.udg.mx](mailto:alan.ocampo4510@alumnos.udg.mx), y tratare de responder a todas sus dudas sobre el software.

### 3. Diagramas de modelamiento.

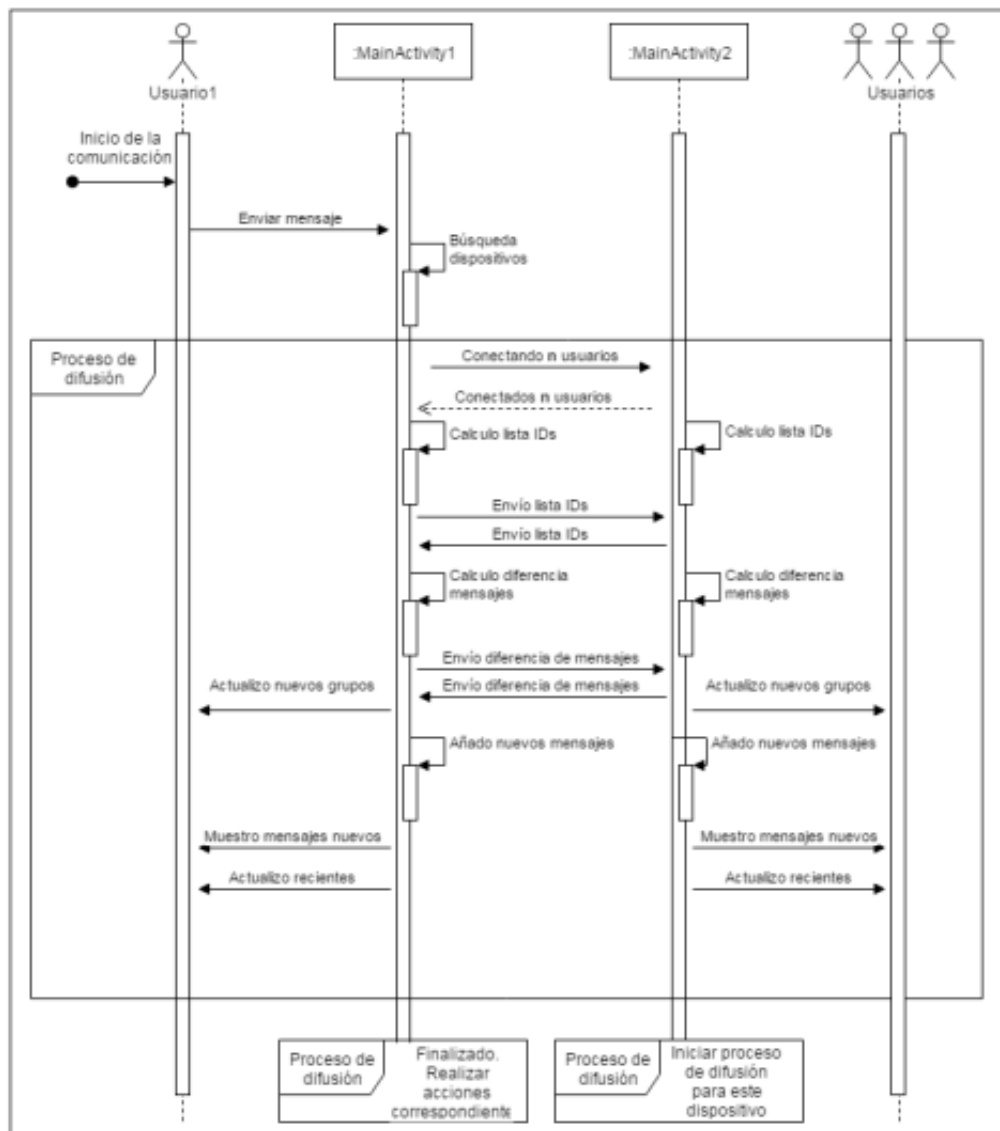
#### Diagrama de casos de uso.

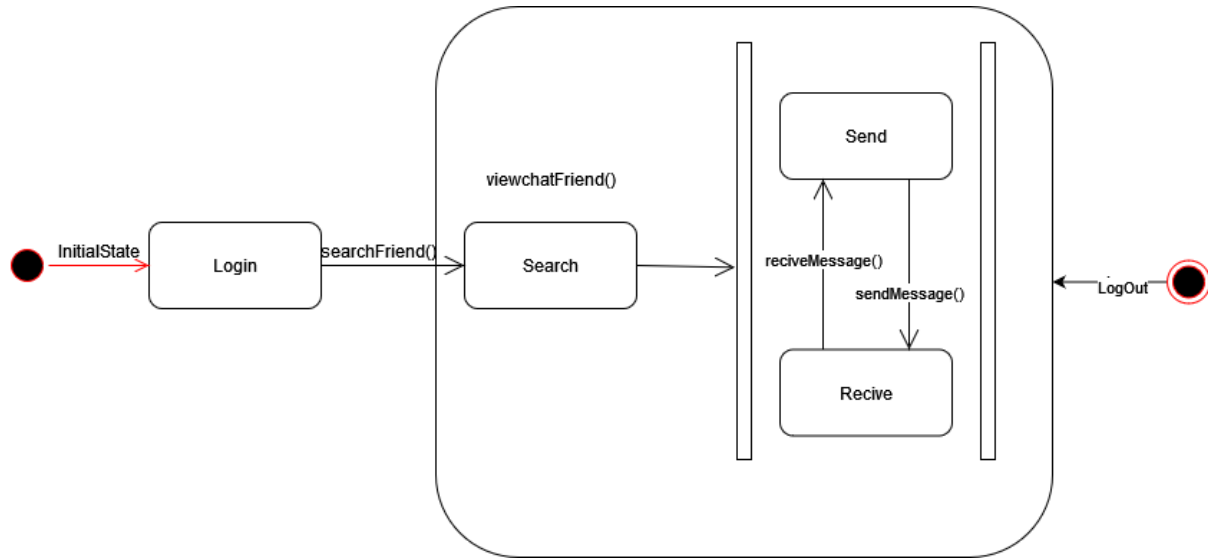




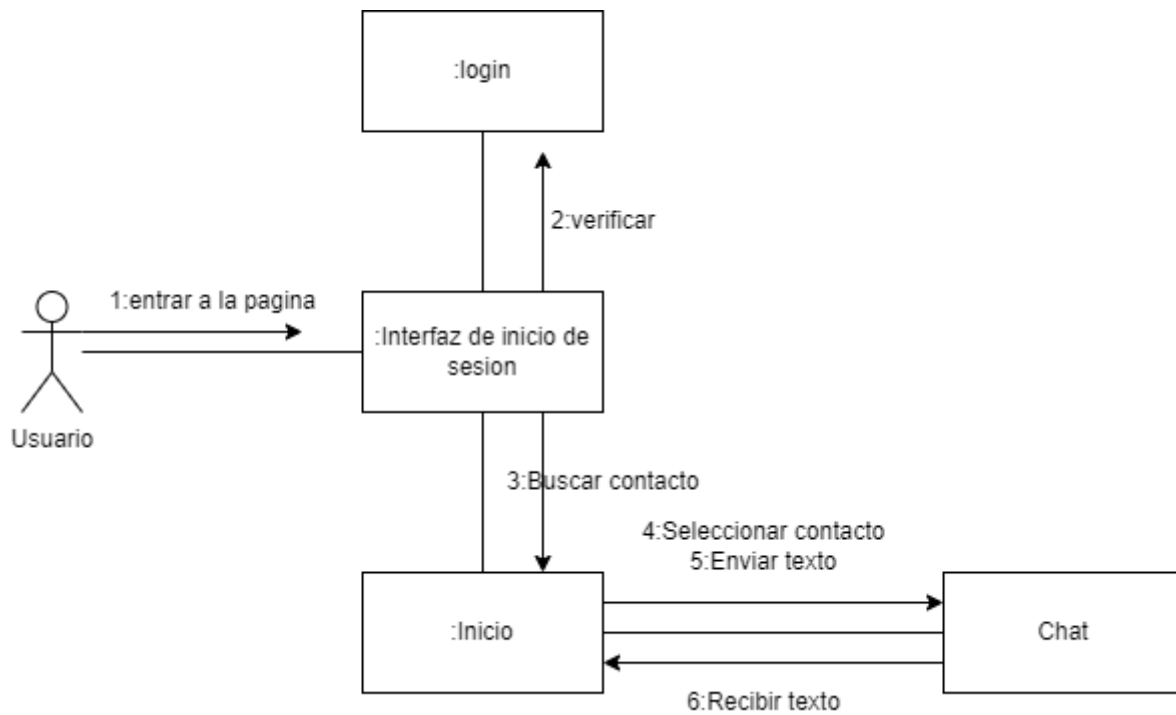
**Diagrama de clases:**

## Diagrama de secuencia.



**Diagrama de estados.**

## Diagrama de colaboración.



## 4. Requisitos recomendados.

### Requisitos de hardware.

- Procesador: Athlon o superior.
- Memoria RAM 4gb.
- Disco Duro 10gb.
- Resolución de pantalla 1280x720 o superior.
- Periféricos, teclado, mouse.

### Requisitos de software.

- Sistema operativo Windows8 o superior.
- Editor de texto (Visual Studio Code)
- Node.js (versión más reciente de ser posible).
- Explorador de internet (Microsoft Edge)

## Bibliografías.

React. (2022). EMPEZANDO. 22/08/22, de Reactjs Sitio web: <https://es.reactjs.org/docs/getting-started.html>

Google. (2022). Documentación y contenido de aprendizaje. 22/08/22, de Google Sitio web: <https://firebase.google.com/docs>