

Connect App

Case Study

By
Sergio De La Torre

Overview

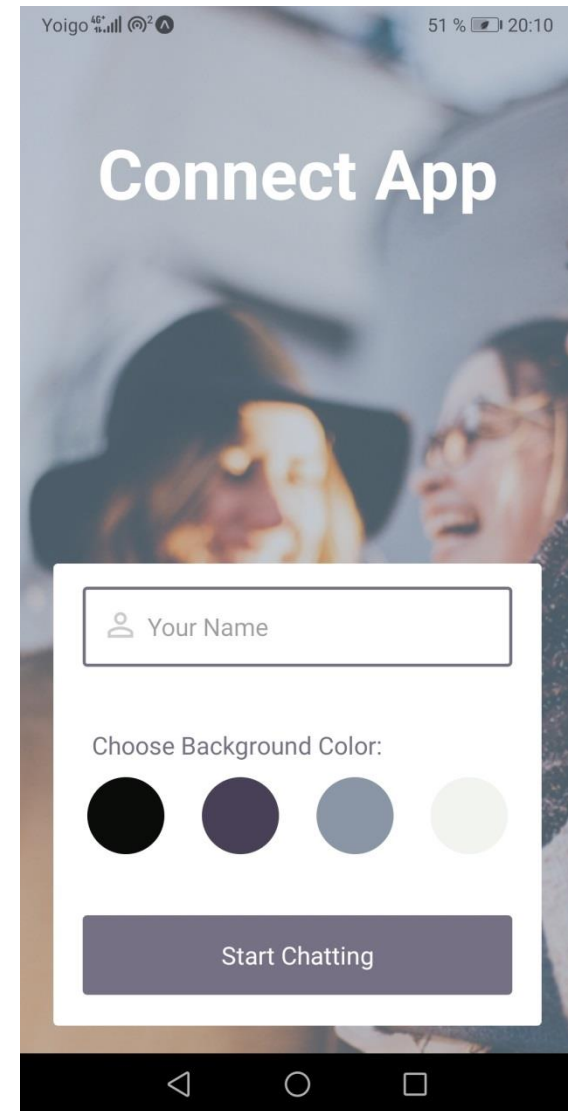
Connect App is a mobile application built using React Native.

The app provides users with a chat interface and options to share images and their location.

Users are able to authenticate anonymously and the chat conversation are stored locally and in the Google Firebase Database.

Check out my GitHub repository for this app here:

[DeLaTorre-23/Connect_App](https://github.com/DeLaTorre-23/Connect_App)



Purpose & Context

I created Connect App as part of my training in Full-Stack Web Development course at CareerFoundry.

This project was specifically part of Native App Development & React module.

This is the first native app I built and also introduced me to the Expo CLI.

The goal of this project was to create a Full-Stack mobile project for Android and iOS that would allow users to easily message each other.

The problem that I was trying to solve was building a native app from scratch using a completely new JavaScript framework for me, React Native, in addition to connecting the server-side with the Google Firebase architecture.

Objective

Client-Side

Setting up a new project using Expo CLI and creating the necessary components was the first step.

After that, I implemented the UI for the homepage as given in the project brief and developed the chat interface on top of the Gifted Chat library.



Would you like to see how I implemented this UI?

Click here:

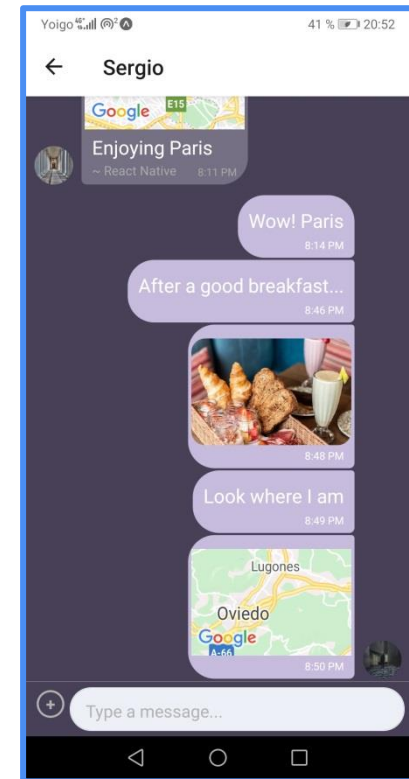
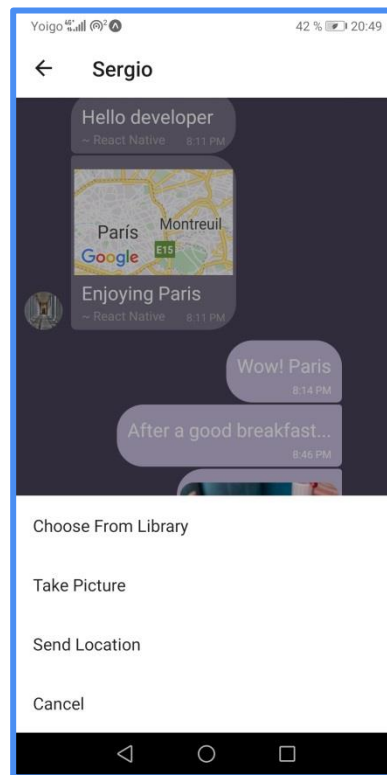
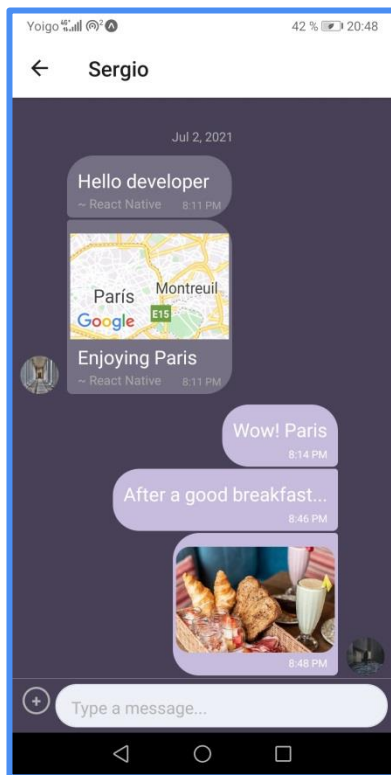
[DeLaTorre-23/Connect_App/Home.js](https://github.com/DeLaTorre-23/Connect_App/Home.js)



Client-Side

To finish up my app, I implemented the necessary communication features.

This meant creating an Action Sheet that contains the code allowing users to take and send images as well as their location over a button near the input field. This grants access to image library on location.



Server-Side

The next part of the project involved implementing the actual chat functionality of my app. This involved setting up a database with Firestore and implementing Gifted Chat, a popular chat UI library.

Connect App ▾

Ir a la docum

Cloud Firestore

Datos Reglas Índices Uso

🏠 > messages > CHIxkBDL0ZApv...

| | | |
|---------------------|--|---|
| connect-app-96277 | messages | CHIxkBDL0ZApvS81oVZu |
| + Iniciar colección | + Agregar documento | + Iniciar colección |
| messages > | CHIxkBDL0ZApvS81oVZu > | + Agregar campo |
| | CXXxN6x5rSGQcU04UQMd j0rq0MizI46fMcaGaapw | <pre>_id: "ea637b69-d581-49b8-825c-05a8ba53e649" createdAt: 16 de abril de 2021 a las 18:47:50 UTC+2 text: "Hi mate!" uid: "bGokHwe2OsgMpGyGH3Eh3OwG0EU2" user _id: 1</pre> |







Server-Side


Every chat message, image and location is stored in Google Firebase Database, allowing for messages to be viewed by users even while offline.


Every message is stored in a JSON Format. Through Google Firebase the users can also authenticate anonymously. While Gifted Chat helped me to greatly improve my app's UI.

<gs://connect-app-96277.appspot.com>

Subir archivo

| <input type="checkbox"/> | Nombre | Tamaño | Tipo | Modificación más reciente |
|--------------------------|--|-----------|------------|---------------------------|
| <input type="checkbox"/> |  0a084289-964a-4311-92c1-a097aa60ff33.jpg | 167.96 KB | image/jpeg | 14 may. 2021 |
| <input type="checkbox"/> |  0b4351f1-1ce2-4f9e-8547-a79cec0282de.jpg | 122.45 KB | image/jpeg | 17 may. 2021 |
| <input type="checkbox"/> |  0e2dd413-7f95-40ff-8769-1c79ac6241db.jpg | 1.58 MB | image/jpeg | 13 may. 2021 |
| <input type="checkbox"/> |  13092fe0-7869-4e6d-a610-379ee4b64f6f.jpg | 167.96 KB | image/jpeg | 14 may. 2021 |
| <input type="checkbox"/> |  146d1ce4-6e3b-4007-8533-1087d8587e2e.jpg | 2.27 MB | image/jpeg | 13 may. 2021 |
| <input type="checkbox"/> |  1ecff97c-aeb6-4569-a019-c6905465ce3e.jpg | 103.29 KB | image/jpeg | 5 may. 2021 |

 0a084289-964a-43... X



Nombre
[0a084289-964a-4311-92c1-a097aa60ff33...](#)

Tamaño
171,988 bytes

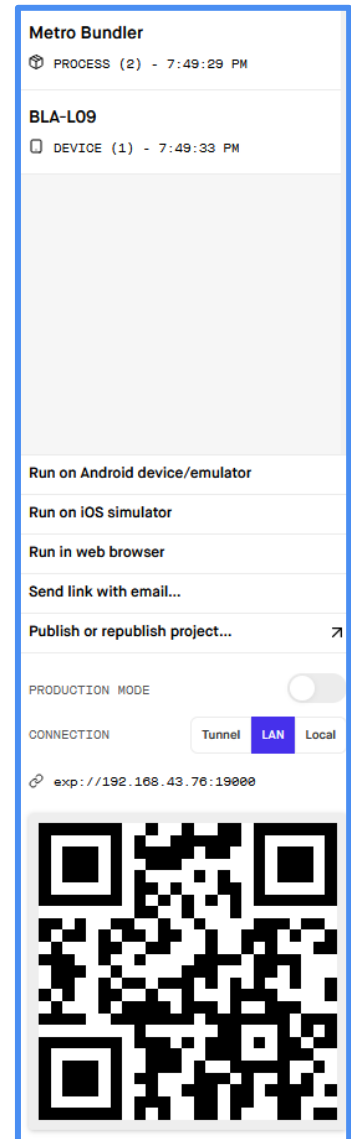
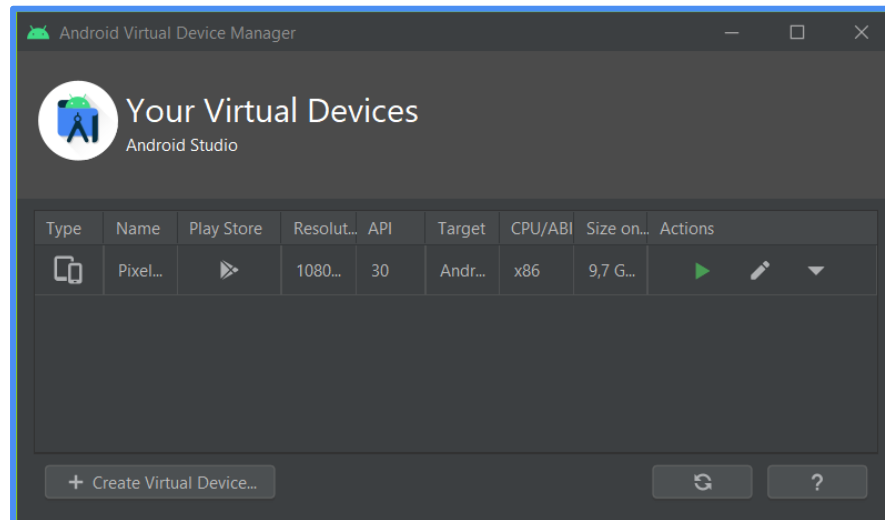
Tipo
image/jpeg

Testing

Next, you can test the app in iOS simulator with XCode and an Android emulator with Android Studio.

Setting up the iOS simulator was fairly straight-forward, while setting up the Android emulator was a bit more complex, as my system settings had to be modified to install the correct tools for the studio.

Once these simulators and emulators were set up, I was able to view my app on these virtual devices as well as my own mobile phone.



Challenge

This was one of my favourite projects in the CareerFoundry bootcamp.

I have enjoyed building a mobile application with Google Firebase architecture and the cross-platform mobile framework React Native.

Developing the client-side, it took me a bit of effort to understand how React Native and Gifted Chat work and how to achieve the desired results. But with help of tutor and self-study, I achieve my goals.

Duration

The development of this project took me 3 weeks.

It was a bit longer than usual because I needed time to learn and develop code with this new technology.

At the end, I am happy with the result and the progress made.

Curious to try the app yourself?
Follow the instructions in my
GitHub account.

[README.md](#)

To check other projects.

Click Here:

[DeLaTorre-23](#)

Credits

Lead Developer

Sergio De La Torre

Tutor

Ebere Iweala

Mentor

Akunna Nwosu