

Tecnológico de Monterrey

Integración de seguridad informática en redes y sistemas de software (Gpo 402)

Sistema para asociación Abuelitos ABP

Documento de Despliegue del sistema

Equipo 4

A01700519, Romel Aldair Vázquez Molina

A00830568, Edgar Alejandro Castillo Palacios

A00831450, Franco Sotomayor Casale

A00830760, José Gerardo Cantú García

A01720588, Marcelo Márquez Murillo

Profesores

David Alonso Cantú Delgado

Rafael Emilio Dávalos Villareal

[DS001]

[v1.0]

21/10/2022

Control de Versiones

| Fecha | Versión | Descripción del Cambio | Autores | Autorizó |
|------------|---------|--|---|--------------------|
| 21/10/2022 | V1.0 | Creación de la primera versión del documento | Edgar Franco José Gerardo Romel Marcelo | Marcelo Marquéz |

Tabla de Contenidos

| | |
|---|----------|
| 1.0 Introducción | 4 |
| 1.1 Propósito | 4 |
| 1.2 Contexto | 4 |
| 1.3 Resumen | 4 |
| 2.0 Glosario | 4 |
| 3.0 Precondiciones | 5 |
| 4.0 Instalación de los componentes del sistema | 5 |
| 4.1 API y Servidor | 5 |
| 4.2 Aplicación Móvil IOS | 7 |
| 4.3 Página web | 11 |

1.0 Introducción

1.1 Propósito

El siguiente documento tiene como objetivo ayudar a cualquier persona que independientemente de su involucramiento con la aplicación, pueda instalar cada uno de los componentes de este proyecto en su ordenador, esto por medio de la creación de servidores locales. Con la finalidad de poder ejecutar las pruebas de evaluación para probar las funcionalidades que se desarrollaron para el socio formador.

1.2 Contexto

Durante el semestre Agosto-Diciembre del 2022 en la materia de “Integración de Seguridad Informática en Redes y Sistemas de Software”, se desarrolló un sistema de software para el socio formador “Abuelitos ABP” con la finalidad de mantener el bienestar integral de los adultos mayores. De esta manera se presenta el siguiente entregable para dar pie a la versión 1.0 desarrollada a lo largo de las 10 semanas que duró la unidad de formación.

1.3 Resumen

A continuación se mostrará un glosario el cual tendrá todas las terminologías empleadas en el despliegue del proyecto, así como las precondiciones que deberán de elaborarse para poder ejecutar el producto. Finalmente se describe una guía paso a paso con la cual el lector podrá probar los ejecutables alojados en el repositorio de Github.

2.0 Glosario

| Término | Explicación |
|-----------------------|---|
| Abuelitos ABP | Asociación sin fines de lucro, a la cual está destinada la creación de este proyecto. |
| Usuario | Toda persona que adquiere un rol en el sistema. |
| Usuario No Registrado | Toda persona que se sitúa en la parte de autenticación de la página web o la aplicación móvil. |
| Adulto Mayor | Usuario que puede realizar pruebas de diagnóstico de su salud, pertenece al grupo de la tercera edad. |
| Paciente | Es un adulto mayor el cual es asignado a un Usuario Doctor. |

| | |
|---------------|---|
| Doctor | Usuario que da seguimiento a un cierto número de adultos mayores y puede administrar los resultados de estos. |
| Administrador | Super Usuario el cual puede ver la información de los doctores y los adultos mayores. |
| DS | Por sus siglas: Despliegue del sistema |
| API | Por sus siglas en inglés: Interfaz de Programación de Aplicaciones |
| Servidor | Computadora capaz de atender las peticiones de un cliente y devolverle una respuesta |

3.0 Precondiciones

Tener un ordenador con el sistema operativo macOS Monterrey versión 12.3.1 o superior para poder ejecutar la parte del API, servidor, aplicación Móvil y Página Web. En cuyo caso de tener un sistema operativo Windows, este deberá de tener la versión 10 o superior, en el cual solo se podrá ejecutar el API, servidor y Página Web.

El dispositivo deberá de tener instalado NodeJs, versión 16.18 o superior, así como tener el editor de texto Visual Studio Code versión 1.72 o superior. Para el caso del dispositivo Mac, deberá de tener la versión de Xcode 14 o superior.

Adicionalmente se deberán de descargar todos los archivos alojados en el siguiente repositorio en la rama de “main”.
[tecnologico-de-monterrey-oficial/C.MTY.TC2007B.402.2213.Equipo4 \(github.com\)](https://github.com/tecnologico-de-monterrey-oficial/C.MTY.TC2007B.402.2213.Equipo4)

4.0 Instalación de los componentes del sistema

4.1 API y Servidor

1. Crear una carpeta nueva en el escritorio.
2. Abrir dicha carpeta en el editor de texto Visual Studio Code.
3. Abrir una nueva terminal en Visual Studio Code y ejecutar los siguientes comandos:

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.
Prueba la nueva tecnologia PowerShell multiplataforma https://aka.ms/powershell
PS C:\Users\ravml\OneDrive\Escritorio\septimo semestre\Clase Cantu\prueba_prisma> npm init[]

- a. npm init (Continuar con la instalación predeterminada)
 - b. npm i express
 - c. npm install prisma --save-dev
 - d. npx prisma
 - e. npx prisma init
 - f. npm install --save-dev nodemon
 - g. npm install -g nodemon
 - h. npm i cors
 - i. npm install morgan
 - j. npm install mssql
 - k. npm install @prisma/client
4. Abrir la carpeta C.MTY.TC2007B.402.2213.Equipo4.
 5. Abrir la carpeta Backend.
 6. Copiar los siguientes archivos: controllers, database, logs, routes y server.js; pegarlos en la carpeta creada en el punto 1.

Este equipo > Escritorio > septimo semestre > proyecto > C.MTY.TC2007B.402.2213.Equipo4 > Backend > Server

| Nombre | Estado | Fecha de modificación | Tipo | Tamaño |
|---------------|--------|------------------------|-----------------------|--------|
| controllers | ✓ | 13/10/2022 06:07 p. m. | Carpeta de archivos | |
| database | ✓ | 13/10/2022 06:07 p. m. | Carpeta de archivos | |
| logs | ✓ | 13/10/2022 06:07 p. m. | Carpeta de archivos | |
| node_modules | ✓ | 13/10/2022 06:08 p. m. | Carpeta de archivos | |
| prisma | ✓ | 13/10/2022 06:08 p. m. | Carpeta de archivos | |
| routes | ✓ | 13/10/2022 06:08 p. m. | Carpeta de archivos | |
| .env | ✓ | 13/10/2022 05:24 p. m. | Archivo ENV | 1 KB |
| .gitignore | ✓ | 29/09/2022 10:32 a. m. | Archivo de origen ... | 1 KB |
| db TEC SERVER | ✓ | 29/09/2022 02:13 p. m. | Archivo de origen ... | 1 KB |
| package | ✓ | 13/10/2022 05:14 p. m. | Archivo de origen ... | 1 KB |
| package-lock | ✓ | 13/10/2022 05:14 p. m. | Archivo de origen ... | 153 KB |
| server | ✓ | 17/10/2022 10:23 a. m. | Archivo de origen ... | 22 KB |

7. Abrir el archivo .env de la carpeta del punto 1 y pegar el siguiente texto en la variable DATABASE_URL:
[sqlserver://equipo4.database.windows.net:1433;database=TC2007B;user=Equipo4;password=TECMty@2022E4;encrypt=true;trustServerCertificate=false;hostNameInCertificate=* .database.windows.net;loginTimeout=30"](https://sqlserver://equipo4.database.windows.net:1433;database=TC2007B;user=Equipo4;password=TECMty@2022E4;encrypt=true;trustServerCertificate=false;hostNameInCertificate=* .database.windows.net;loginTimeout=30)

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure named "PRUEBA_PRISMA" containing "controllers", "database", "logs", "node_modules", "prisma", ".env", ".gitignore", "package-lock.json", "package.json", and "server.js".
- Code Editor:** Displays the "schema.prisma" file with the following code:

```

1 # Environment variables declared in this file are automatically made available to Prisma.
2 # See the documentation for more detail: https://pris.ly/d/prisma-schema#accessing-environment-variables-from-the-schema
3
4 # Prisma supports the native connection string format for PostgreSQL, MySQL, SQLite, SQL Server, MongoDB and CockroachDB.
5 # See the documentation for all the connection string options: https://pris.ly/d/connection-strings
6
7 DATABASE_URL= "sqlserver://equipo4.database.windows.net:1433;database=TC2007B;user=Equipo4;password=TECMty@2022E4;encrypt=true;trustServerCertificate=true"
8

```

8. Ir a la carpeta del punto 1 y abrir el siguiente archivo sheme.prisma.
9. Modificar la variable provider en el objeto de datasource db por: "sqlserver".

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure with "schema.prisma" selected.
- Code Editor:** Displays the "schema.prisma" file with the "provider" field in the "db" object changed from "prisma-client-js" to "sqlserver".

```

prisma > schema.prisma
prisma > generator client {
  2 |   provider = "prisma-client-js"
  3 |
  4
  5 datasource db {
  6   provider = "sqlserver"
  7   url      = env("DATABASE_URL")
  8 }
  9
 10 model Accion {
 11   IdAccion     Int          @id(map: "PK_Accion_9845169B82928D83") @default(autoincrement())
 12   Tema        String?
 13   Descripcion String?
 14   Register    DateTime?    @default(now(), map: "DF_Accion_Register_236943A5") @db.DateTime
 15   Is_active   Boolean
 16   AccionPaciente AccionPaciente[]
 17 }
 18
 19 model AccionPaciente {
 20   IdAccionPaciente Int      @id(map: "PK_AccionPa_060E2AA173F0D823") @default(autoincrement())
 21   Completado    Boolean?
 22   Register     DateTime? @default(now(), map: "DF_AccionPac_Regis_26458050") @db.DateTime
 23   Is_active    Boolean
 24   TdPaciente   Int?

```

10. Ejecutar los siguientes comandos en la terminal del punto 3
 - a. npx prisma db pull
 - b. npx prisma generate
11. Ejecutar el server con el siguiente comando en terminal
 - a. nodemon server.js
12. Deberá de aparecer un mensaje como el siguiente

The screenshot shows the terminal window in Visual Studio Code with the following output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnologia PowerShell multiplataforma https://aka.ms/powershell

PS C:\Users\avalvOneDrive\Escritorio\septimo semestre\Clase Cantu\prueba_prisma> nodemon server.js
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node server.js'
Server started running on : 3000
Connected to MSSQL

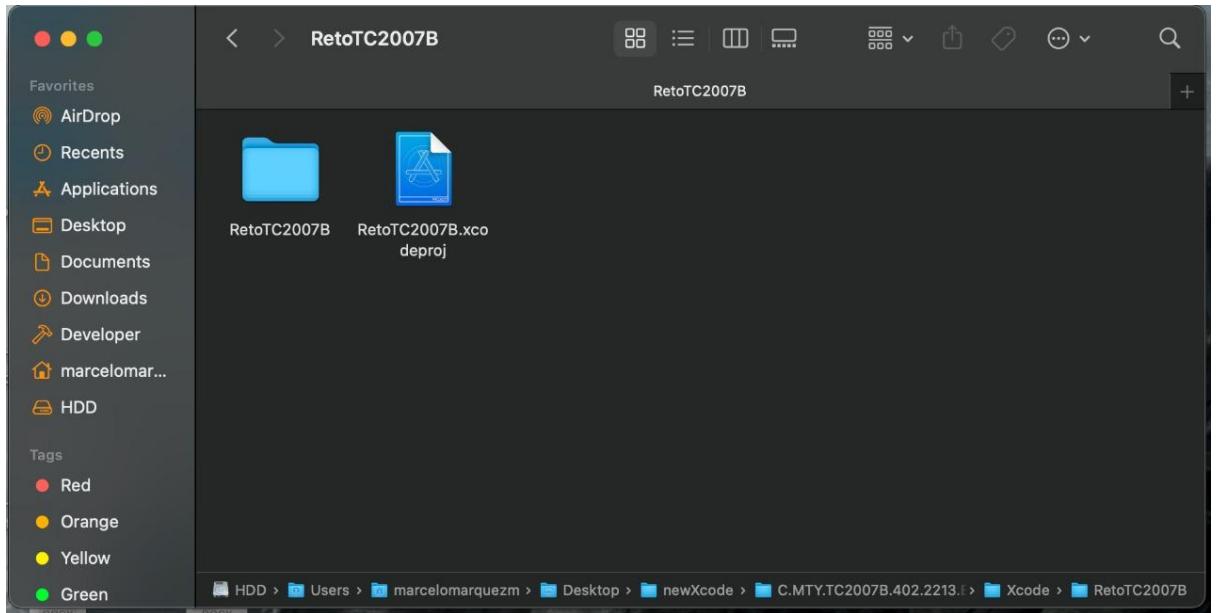
```

Si en cualquier momento se desea detener el servidor, se podrá terminar su ejecución con el atajo CTRL+C.

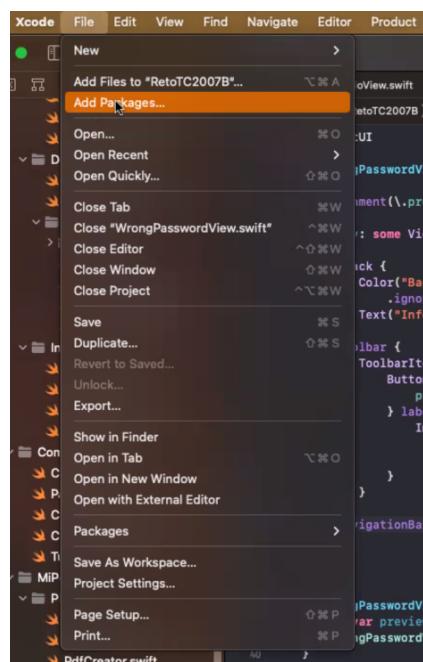
4.2 Aplicación Móvil IOS

1. Abrir la carpeta C.MTY.TC2007B.402.2213.Equipo4
2. Abrir la carpeta XCODE
3. Abrir la carpeta RetoTC2007B

4. Abrir el archivo RetoTC2007B.xcodeproj

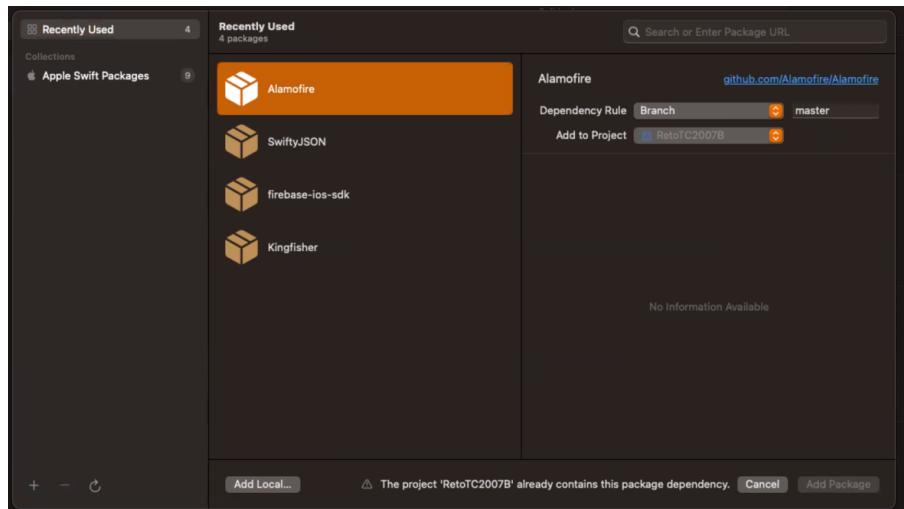


5. Dentro del ambiente de Xcode, ir a la sección de Archivo >> añadir paquetes:

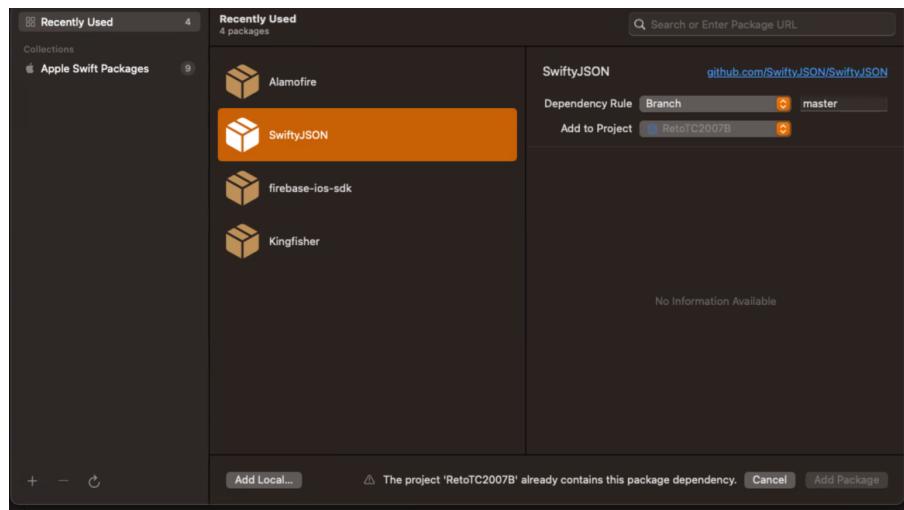


6. Instalar los siguientes paquetes:

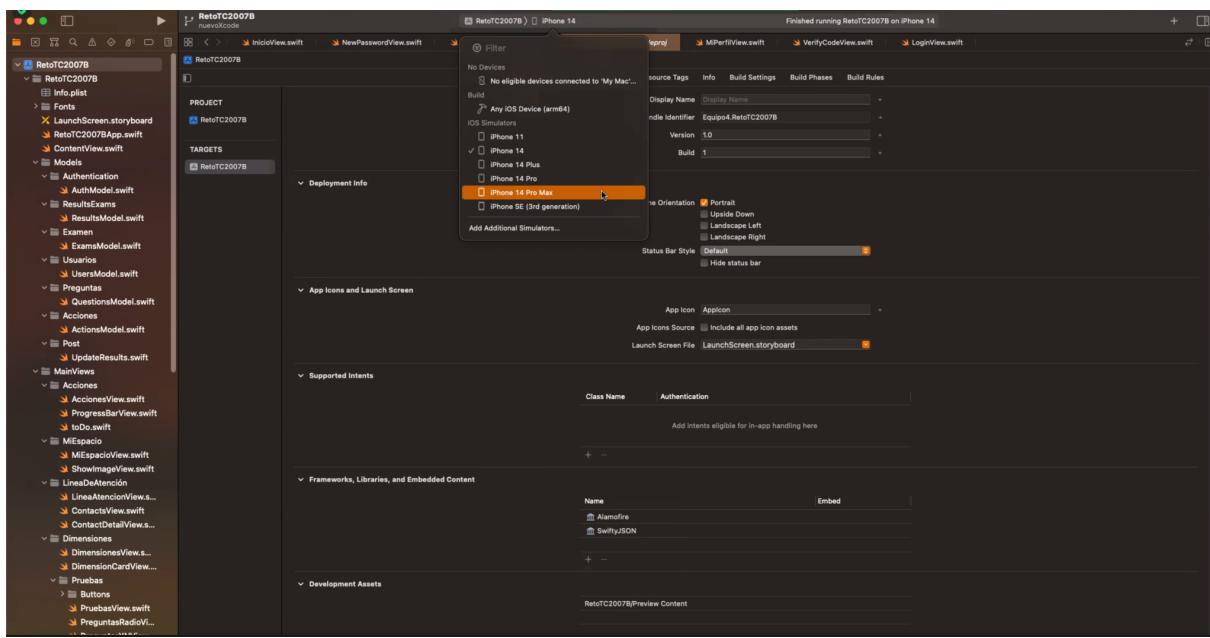
AlamoFire: <https://github.com/Alamofire/Alamofire>



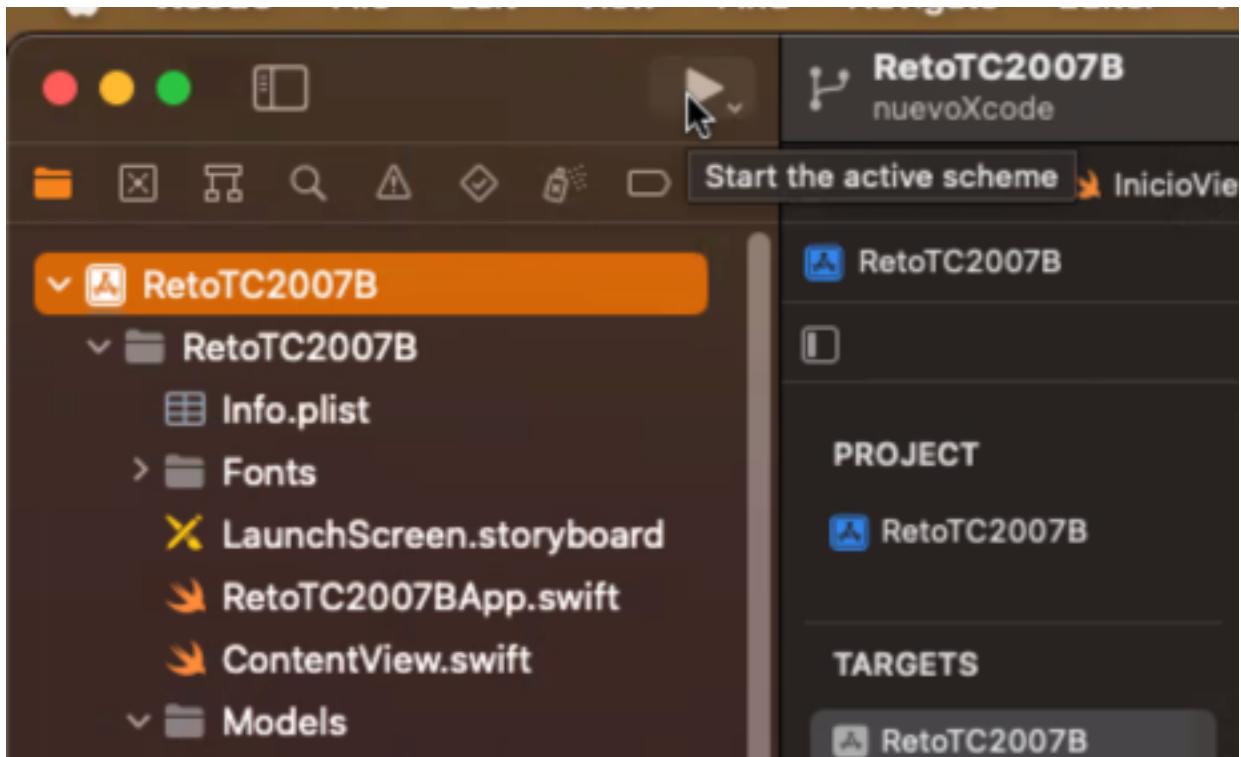
SwiftyJSON: <https://github.com/SwiftyJSON/SwiftyJSON>

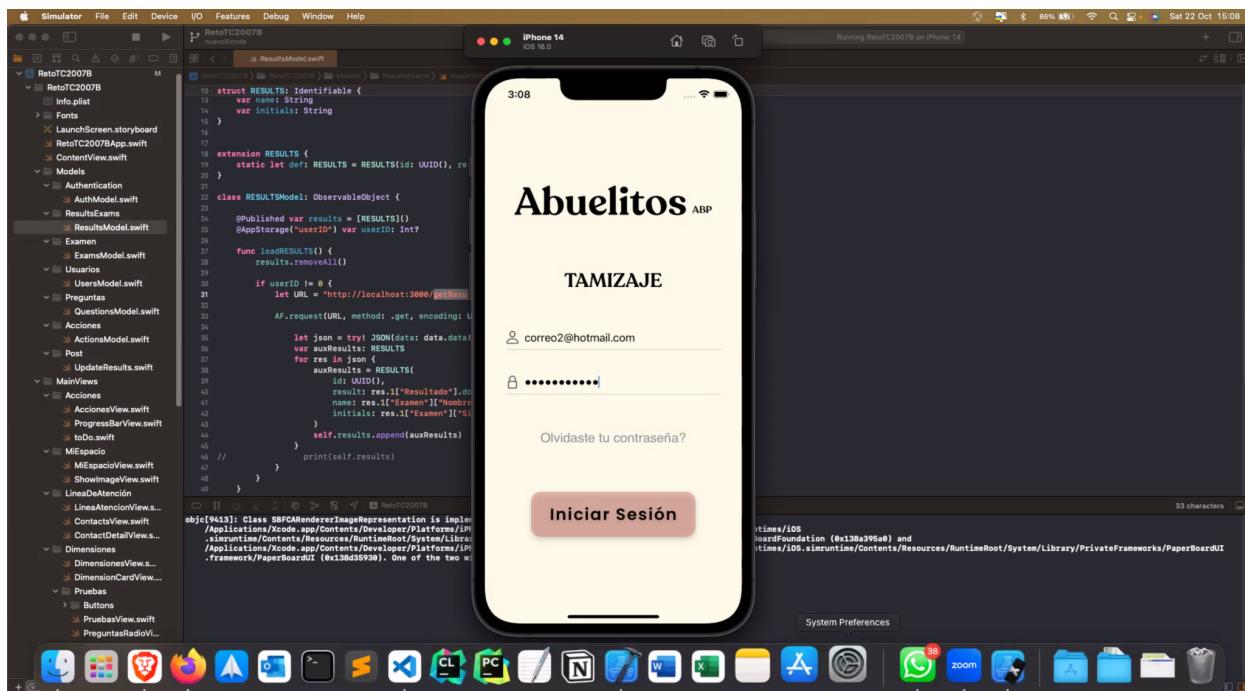


7. Seleccionar el simulador de Iphone 14 pro Max



8. Ejecutar el proyecto con el botón de “Run”, desplegando una pantalla como la siguiente:





Si en algún momento se quiere terminar la ejecución de la aplicación, solo se debe de oprimir el botón de “stop”, ubicado en la esquina superior izquierda.

4.3 Página web

1. Abrir el editor de texto Visual Studio Code.
2. Abrir la carpeta de WEB dentro del editor de texto Visual Studio Code.

```

README.md
1 # Getting Started with Create React App > ## Available Scripts > ### `npm start`
2
3 This project was bootstrapped with [Create React App](https://github.com/facebook/create-react-app).
4
5 ## Available Scripts
6
7 In the project directory, you can run:
8
9 ### `npm start`
10
11 Runs the app in the development mode.\|
12 Open [http://localhost:3000](http://localhost:3000) to view it in your browser.
13
14 The page will reload when you make changes.\|
15 You may also see any lint errors in the console.
16
17 ### `npm test`
18
19 Launches the test runner in the interactive watch mode.\|
20 See the section about [running tests](https://facebook.github.io/create-react-app/docs/running-tests) for more information.
21
22 ### `npm run build`
23
24 Builds the app for production to the `build` folder.\|
25 It correctly bundles React in production mode and optimizes the build for the best performance.
26
27 The build is minified and the filenames include the hashes.\|
28 Your app is ready to be deployed!
29
30 See the section about [deployment](https://facebook.github.io/create-react-app/docs/deployment) for more information.
31

```

3. Abrir una nueva terminal dentro del editor de texto Visual Studio Code y ejecutar los siguientes comandos:

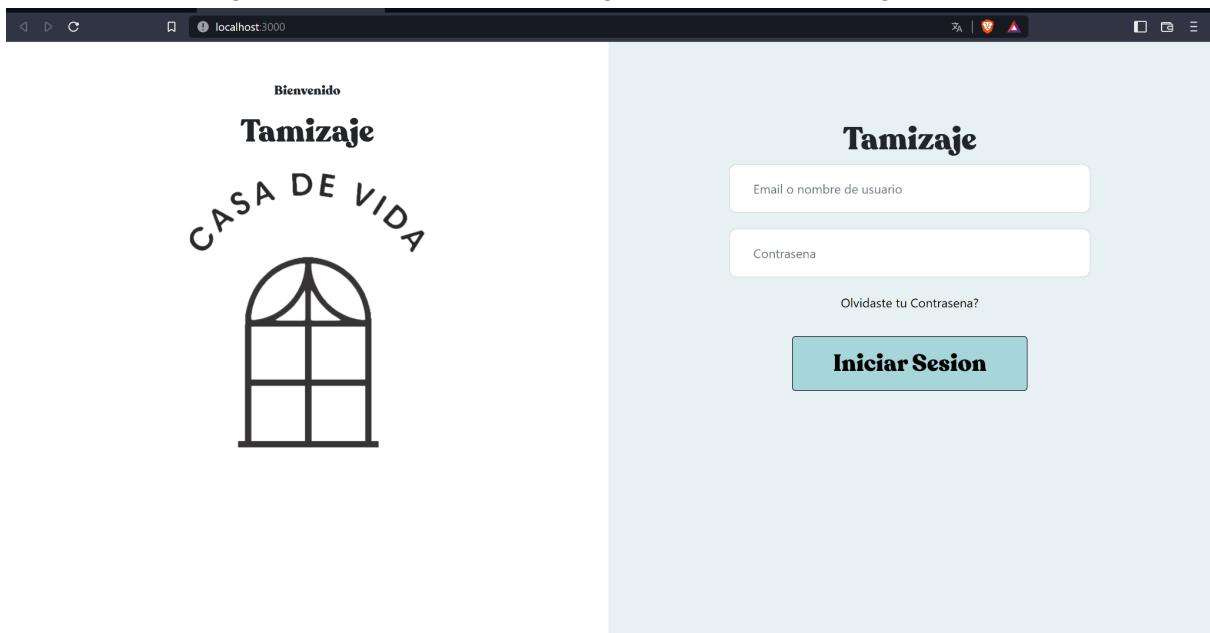
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
webpack compiled with 1 warning
¿Desea terminar el trabajo por lotes (S/N)? s
PS C:\Users\ravml\OneDrive\Escritorio\septimo semestre\ proyecto\ C.MTY.TC2007B.402.2213.Equipo4\WEB> npm i
```

- a. npm i
- b. npm i sweetalert2
- c. npm i xlsx
- d. npm i file-saver
- e. npm i bootstrap
- f. npm i sweetalert2-react
- g. npm install --save sweetalert2-react-content

4. Ejecutar el siguiente comando en la consola: npm start

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
webpack compiled with 1 warning
¿Desea terminar el trabajo por lotes (S/N)? s
PS C:\Users\ravml\OneDrive\Escritorio\septimo semestre\ proyecto\ C.MTY.TC2007B.402.2213.Equipo4\WEB> npm start
```

5. Se desplegará una ventana en el navegador web como la siguiente



Si en cualquier momento se desea detener el servidor, se podrá terminar su ejecución con el atajo CTRL+C e introducir la tecla "s".