Reto React Native

Esta aplicación consiste en el desarrollo móvil utilizando el framework de React Native, para el caso de una app médica, en donde se tiene una serie de sensores con conectividadBluetooth con diferentes proveedores. La aplicación se encarga del manejo de datos por el operario, al cual se le asigna una serie de pacientes, que a su vez, cada paciente tienen asignados unos sensores registrados desde la aplicación central por su identificador MAC, permitiendo tomar de manera independiente los datos de ese sensor. Una vez capturada la información del paciente y confirmada su envío.

Como correr la aplicación

La aplicación puede ser facilmente desplegada con ayuda de npx react-native, para más información vease este documento

Desplegala en vivo

Primero instala todas las dependencias necesarias:

```
npm install
```

Para el uso de la API, necesitas el SECRET KEY, tienes que añadirlo en un archivo llamado .env, el cual debe estar en el principio del proyecto:

.env:

```
SECRET=xxxxxx
```

(Si no se añade este archivo la subida de datos no funcionara!)

Para el siguiente paso necesitas un dispositivo en el cual probar la aplicación, puedes usar uno fisico o crear un ADB. Android Studio hace este proceso muy sencillo.

```
npx react-native start
```

Escoge la opción de android y la app se abrira en el dispositivo que tengas configurado.

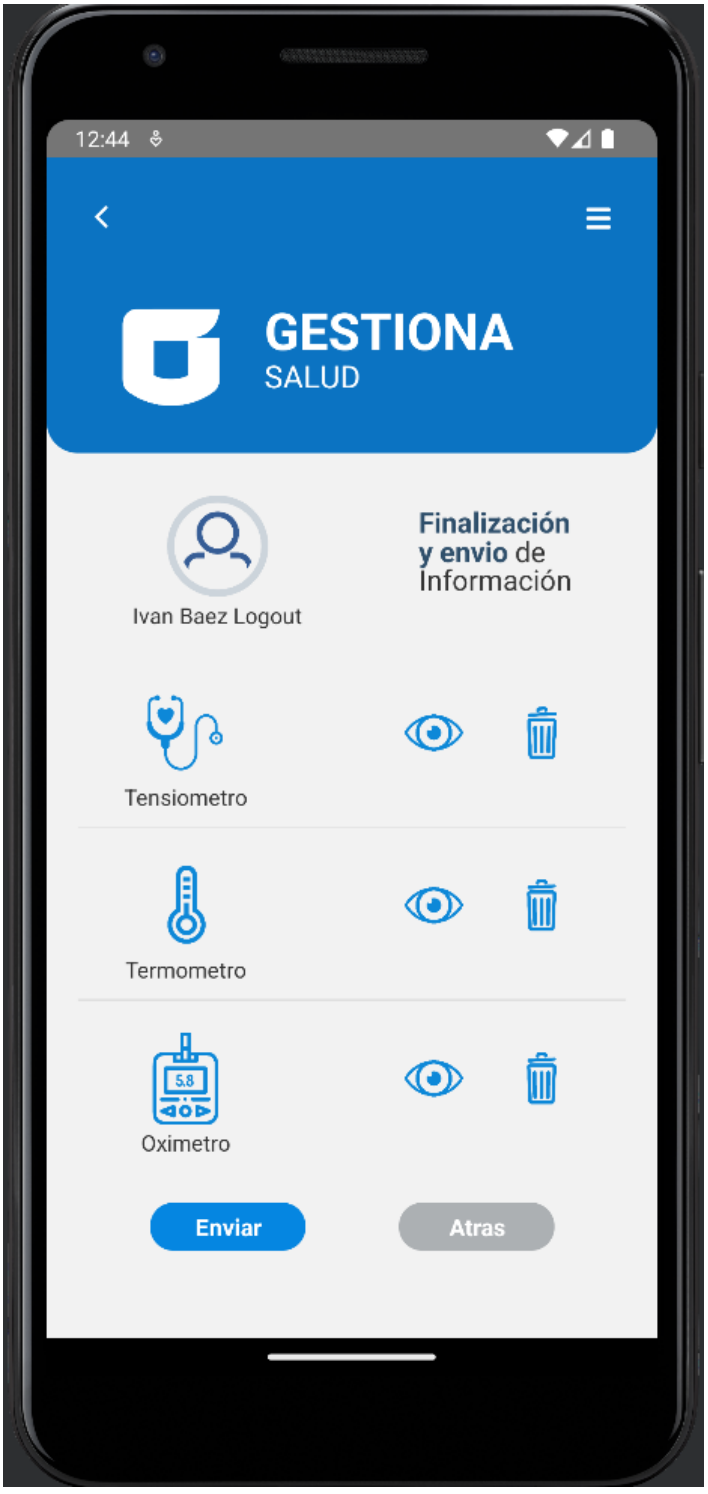
Instala el APK

Tambien puedes instalar el APK directamente en un dispositivo. Puedes encontrarla aquí

Dispositivos y datos adquiridos View

Demostración de capacidad de conceptos:

- Funcionalidad de Vista Previa y de Eliminación de los datos adquiridos
- Funcionalidad de envío del frame de datos capturados al API y la confirmación de su envío



En esta vista se usan multiples componentes, para generar el estilo de la misma.

Por ejemplo (NavBar):

```
const NavBar = () => {
  const navigation = useNavigation();
  return (
    <View style={styles.container}>
      <View style={styles.icons_container}>
        <Pressable onPress={() => navigation.goBack()}>
          <Icon name="chevron-left" size={16} color="#fff" />
        </Pressable>
        <Pressable onPress={() => navigation.toggleDrawer()}>
          <Icon name="bars" size={18} color="#fff" />
        </Pressable>
      </View>

      <View style={styles.title_container}>
        <Image
          source={require('../iconos/logo.png')}
          style={{width: 120, height: 120}}
        />

        <View style={{display: 'flex'}}>
          <Text style={styles.text}>GESTIONA</Text>
          <Text style={{fontSize: 20, color: '#fff', lineHeight: 20}}>
            SALUD
          </Text>
        </View>
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    display: 'flex',
    backgroundColor: '#0b73c2',
    maxHeight: 190,
    borderBottomLeftRadius: 20,
    borderBottomRightRadius: 20,
  },

  icons_container: {
    display: 'flex',
    flexDirection: 'row',
    justifyContent: 'space-between',
    padding: 30,
    paddingBottom: 0,
  },

  title_container: {
    display: 'flex',
```

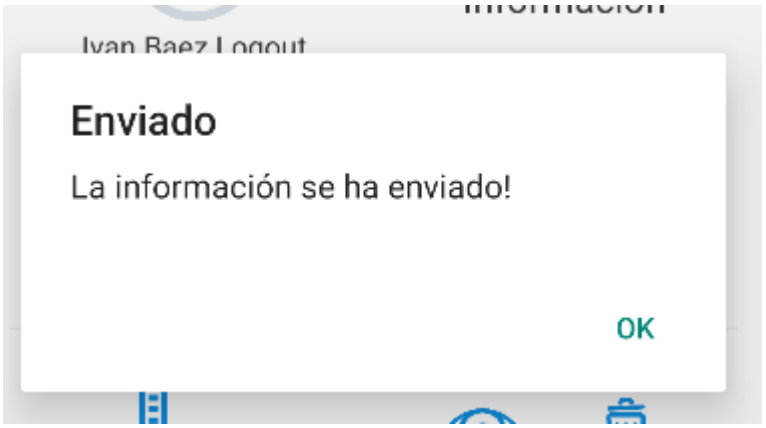
```
    flexDirection: 'row',
    padding: 20,
  },
  text: {
    marginTop: 20,
    fontWeight: 'bold',
    textAlign: 'center',
    fontSize: 33,
    color: '#fff',
  },
});
```

La parte logica que maneja esta vista es el envío de información al servidor. Se hace por medio de un pressable, que llama la siguiente función:

```
  handlePress = () => {
    try {
      const pulsi = require('../Datasensores/pulsi.json');
      let oxy = pulsi['data']['blood_oxygen'];
      const tensio = require('../Datasensores/tensio.json');

      let sys = tensio['data']['high_pressure'];
      let dia = tensio['data']['low_pressure'];
      let pulso = tensio['data']['heart_rate'];
      let latitude = tensio['latitude'];
      let longitud = tensio['longitud'];

      const customData = require('../Datasensores/temp.json');
      let temp = customData['data']['temperature'];

      const raw = JSON.stringify({
        secret: SECRET,
        type: 'set',
        target: 'sensor',
        business_id: 1000,
        user_id: 5,
        patient_id: 11,
        device_id: 111,
        latitude: latitude,
        longitude: longitud,
        device_time: new Date().toISOString().slice(0, 19).replace('T', ' '), // '20
        data: {
          temperature: temp,
          high_pressure: sys,
          low_pressure: dia,
          heart_rate: pulso,
          blood_oxygen: oxy,
        },
      });
```

```
      const requestOptions = {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: raw,
        redirect: 'follow',
      };

      fetch('https://idco.com.co/sensors2/api.php', requestOptions)
        .then(response => response.text())
        .catch(error => console.log('API REQUEST', error));
      Alert.alert('Enviado', 'La información se ha enviado!');
    } catch (error) {
      console.log(error);
      Alert.alert(
        'No fue enviado',
        'Un error evito que la información se enviara',
      );
    }
  };
```

El codigo toma los datos del .json, lo cual puede ser cambiado para que tome de una API de la misma manera. Y por ultimo envia los datos al servidor.

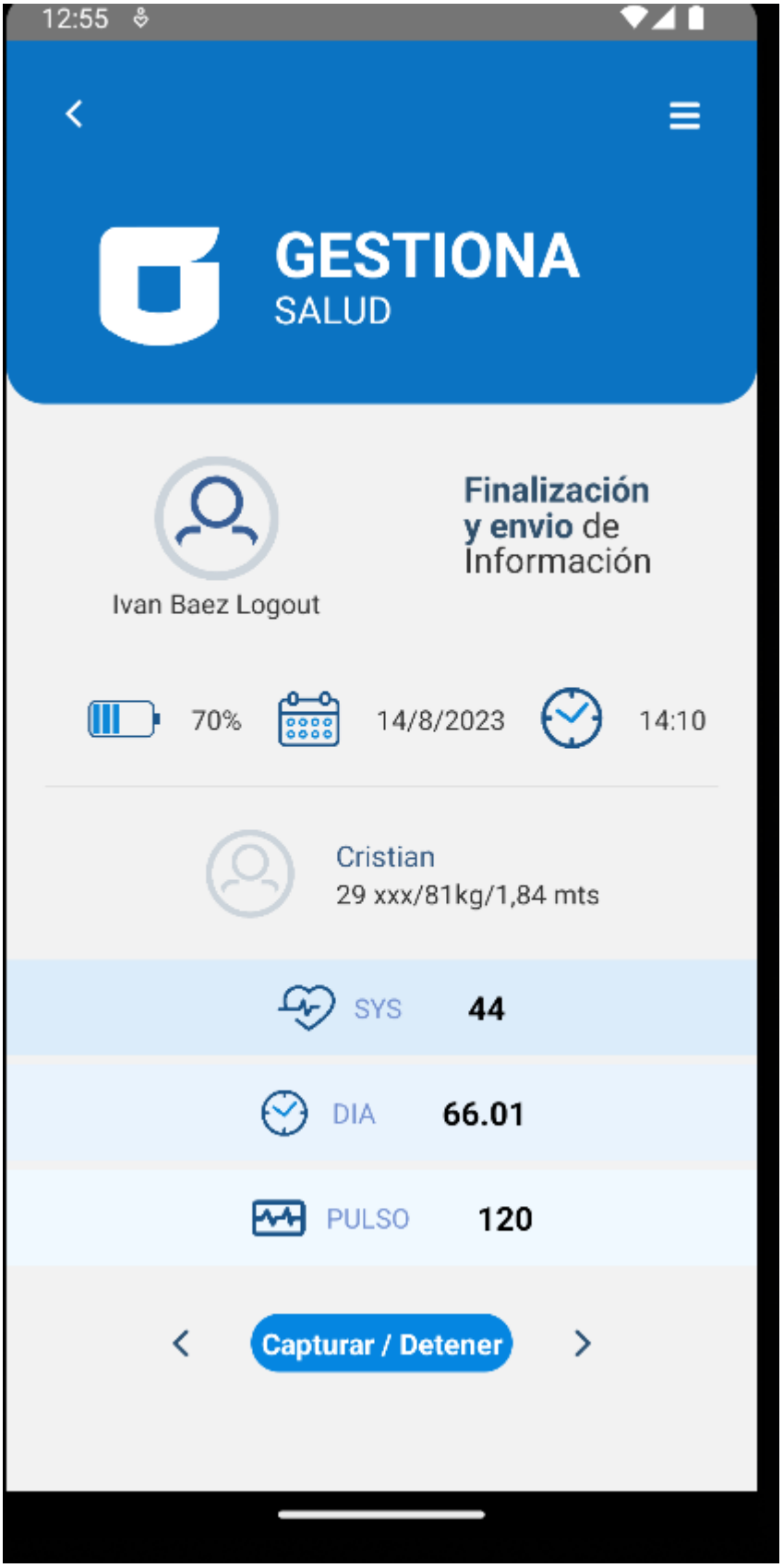


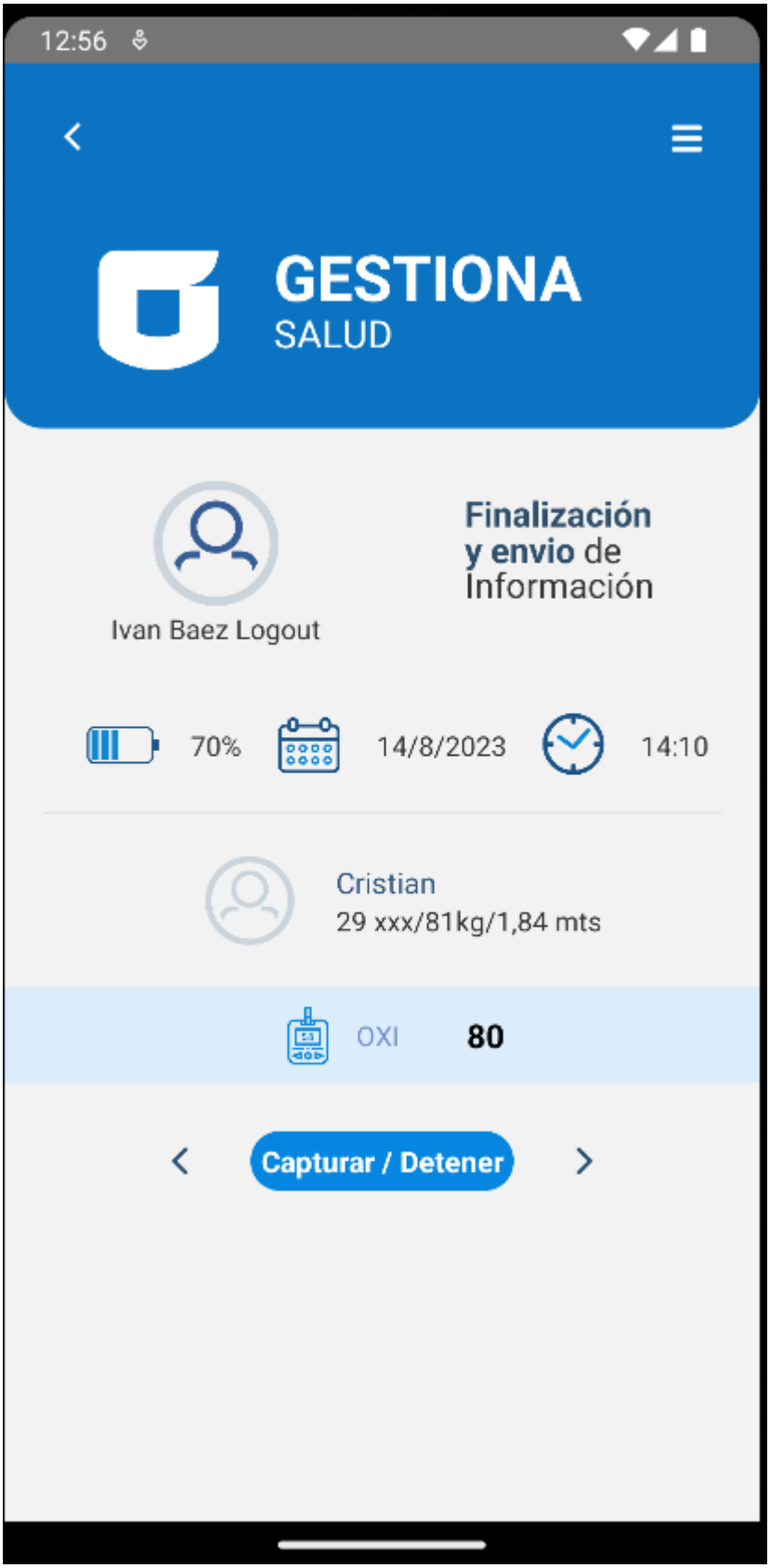
Alerta con la información.

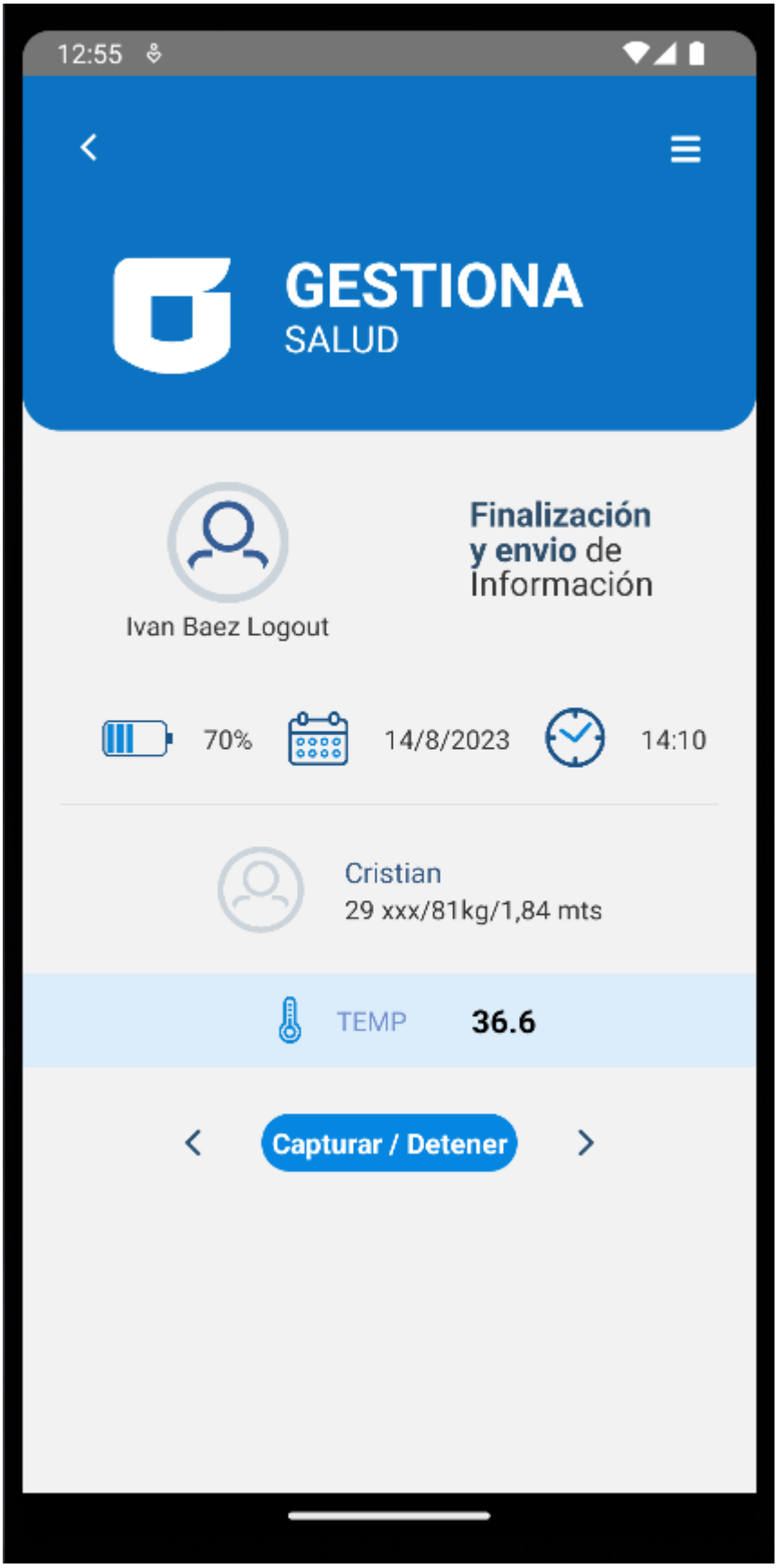
Por ultimo podemos revisar en https://idco.com.co/sensors2/view.php, la información subida:

Device Time (-5 TZ-COL)	Server Time (-5 TZ-COL)	Business	User	Patient	Device	Temperature	High Pressure	Low Pressure	Heart Rate	Blood Oxygen	Latitude	Longitude	Map
2023-08-14 07:50:18	2023-08-14 07:50:27	1000	5	11	111	36.6	44	66.01	120	80	7.1165783333333	-73.112495	Ver
2023-08-13 22:17:03	2023-08-13 22:17:03	1000	5	11	111	36.6	44	66.01	120	80	7.1165783333333	-73.112495	Ver
2023-08-13 22:08:48	2023-08-13 22:08:48	1000	5	11	111	36.6	44	66.01	120	80	7.1165783333333	-73.112495	Ver
2023-08-13 21:30:13	2023-08-13 21:30:14	1000	5	11	111	36.6	44	66.01	120	80	7.1165783333333	-73.112495	Ver

Captura de información (Tensiometro, Termometro, Oximetro)


12:55
GESTIONA
SALUD
Finalización
y envio de
Información
Ivan Baez Logout
70%
14/8/2023
14:10
Cristian
29 xxx/81kg/1,84 mts
SYS
44
DIA
66.01
PULSO
120
Capturar / Detener


12:56
GESTIONA
SALUD
Finalización
y envio de
Información
Ivan Baez Logout
70%
14/8/2023
14:10
Cristian
29 xxx/81kg/1,84 mts
OXI
80
Capturar / Detener



Estas vistas se basan en diferentes componentes para generar los estilos necesitados, tambien añade datos del mismo .json abriendo el archivo y generando un objeto con los mismos datos.

TO DO:

- Logeo a la aplicación (relacion paciente-operario, paciente-dispositivo)
- Manejo de datos locales (cache)
- Añadir las ventanas emergentes personalizadas