



“SISTEMA PARA EL MONITOREO, DETECCIÓN Y ALERTA DE
SOMNOLENCIA DEL CONDUCTOR MEDIANTE VISIÓN ARTIFICIAL,
COMUNICACIÓN INALÁMBRICA Y GEOLOCALIZACIÓN”

Primer Reporte Parcial

Lista de actividades

- Definir rutas del frontend
- Diseño de rutas del backend
- Conexión Backend con Mongo DB
- Sistema de acceso con credenciales
- Creación de la base de datos no relacional
- Investigación de modelos de Redes Neuronales Convolucionales
- Diseño de una red neuronal convolucional capaz de detectar ojos cerrados y abiertos

Autores:

Alan Eduardo Gamboa Del
Ángel
Maite Paulette Díaz Martínez

Asesores:

M.en C. Niels Henrik Navarrete
Manzanilla
Dr. Rodolfo Vera Amaro

Índice

1. Definir rutas del frontend	4
1.1. Objetivo	4
1.2. Descripción	4
1.3. Resultados	5
2. Definir rutas del backend	6
2.1. Objetivo	6
2.2. Descripción	6
2.3. Resultados	6
3. Conexión Backend con Mongo DB	7
3.1. Objetivo	7
3.2. Descripción	7
3.3. Resultados	9
4. Sistema de acceso con credenciales	10
4.1. Objetivo	10
4.2. Descripción	10
4.3. Resultados	12
5. Creación de la base de datos No Relacional	14
5.1. Objetivo	14
5.2. Descripción	14
5.3. Resultados	14
6. Investigación de modelos de Redes Neuronales Convolucionales	15
6.1. Objetivo	15
6.2. Descripción	15
6.3. Resultados	15
7. Diseño de una red neuronal convolucional capaz de detectar ojos cerrados y abiertos	16
7.1. Objetivo	16
7.2. Descripción	16
7.3. Resultados	16
8. Conclusiones	17
9. Bibliografía	18

Índice de figuras

1.	Creación proyecto de react	4
2.	Creación proyecto de react	4
3.	Creación proyecto de react	4
4.	Creación proyecto de react	5
5.	Página web MongoDB Atlas.	6
6.	Página web MongoDB Atlas.	7
7.	Página Principal MongoDB	7
8.	Página Creación de Cluster	8
9.	Página Creación de Cluster	8
10.	Directorio del Backend	8
11.	Directorio del Backend	9
12.	Directorio del Backend	9
13.	Directorio del Backend	9
14.	Directorio del Backend	10
15.	Directorio del Backend	11
16.	Directorio del Backend	11
17.	Directorio del Backend	12
18.	Grupos de usuario	12
19.	Grupos de usuario	13
20.	Grupos de usuario	13

Índice de tablas

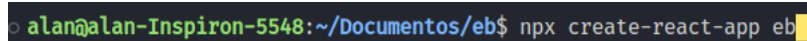
1. Definir rutas del frontend

1.1. Objetivo

Definir e implementar las rutas que tendrá la aplicación, así como si serán públicas o privadas y la información que se desplegará en cada una de las mismas.

1.2. Descripción

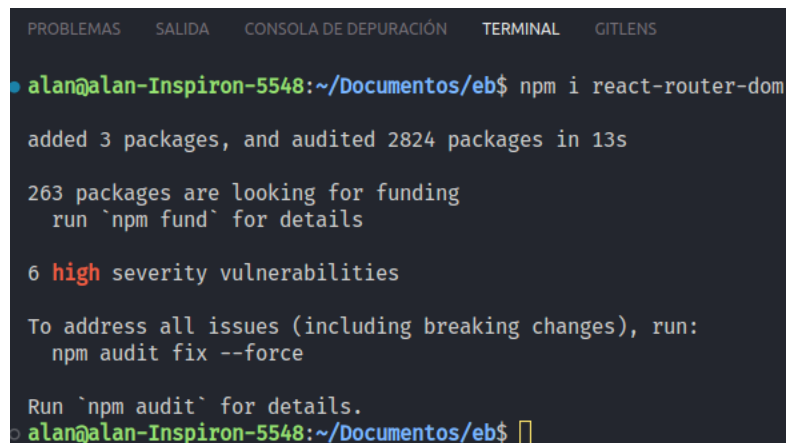
Como primer paso, se necesita crear un proyecto de React utilizando el siguiente comando:



```
alan@alan-Inspiron-5548:~/Documentos/eb$ npx create-react-app eb
```

Figura 1: Creación proyecto de react

Posteriormente, se realiza la instalación del moudelo *React Router Dom* utilizando el siguiente comando:



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  GITLENS

alan@alan-Inspiron-5548:~/Documentos/eb$ npm i react-router-dom

added 3 packages, and audited 2824 packages in 13s

263 packages are looking for funding
  run `npm fund` for details

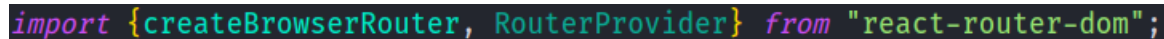
6 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
alan@alan-Inspiron-5548:~/Documentos/eb$
```

Figura 2: Creación proyecto de react

Dentro de nuestro proyecto creado, ingresamos al archivo *index.js*. Dentro de este archivo, importamos dos librerías ofrecidas por el módulo de *react router dom* utilizando la línea de código siguiente:



```
import {createBrowserRouter, RouterProvider} from "react-router-dom";
```

Figura 3: Creación proyecto de react

La importación de estás librerías permitirán diseñar las rutas del frontend que se utilizarán en el proyecto.

En el mismo archivo *index.js* se creará una variable con el nombre de router que contendrá todas las rutas del fronted de la aplicación.

```
const router = createBrowserRouter([
  {
    path: '/',
    element: <App/>
  },
  {
    path: '/home',
    element: <h1>Home</h1>
  },
  {
    path: '/conductor/id',
    element: <h1>Conductores</h1>
  },
  {
    path: '/detalle_incidencia',
    element: <h1>Incidencias</h1>
  },
  {
    path: 'conductor/id/ubicacion',
    element: <h1>Ubicacion Tiempo Real</h1>
  },
  {
    path: 'conductor/id/incidencias',
    element: <h1>Incidencias por conductor</h1>
  }
])
```

You, hace 6 minutos • Uncommitted changes

Figura 4: Creación proyecto de react

1.3. Resultados

- **Path:** /
Descripción: En esta dirección se encontrará la el formulario para poder iniciar sesión o registrarse
- **Path:** /home
Descripción: Esta dirección será la página principal de la aplicación dónde se mostrarán las incidencias más recientes así como una lista de todos los conductores
- **Path:** /conductor/id
Descripción: Esta dirección mostrará el perfil del conductor de id correspondiente
- **Path:** /detalle_incidencia
Descripción: Esta dirección mostrará cada incidencia mostrando detalles como hora, fecha, coordenadas
- **Path:** /conductor/id/ubicacion
Descripción: En esta vista se mostrará la ubicación en tiempo real de cada conductor
- **Path:** /conductor/id/incidencias
Descripción: En esta vista se mostrará todas las incidencias registradas por cada conductor

2. Definir rutas del backend

2.1. Objetivo

Crear las rutas mediante las que el cliente realizará las peticiones y tendrá acceso a las operaciones, así como su funcionamiento en cuanto a obtención de datos y comunicación con el resto de la aplicación.

2.2. Descripción

Para poder utilizar los servicios de Amazon Amplify, necesitamos dirigirnos al directorio root de nuestro proyecto y ejecutar el siguiente comando:



```
npm install -g @aws-amplify/cli
```

copy

Figura 5: Página web MongoDB Atlas.

Como primer paso, se necesita especificar que estará utilizando el *SDK* de Amazon Amplify:

2.3. Resultados

3. Conexión Backend con Mongo DB

3.1. Objetivo

Realizar la conexión de NodeJs con la base de datos MongoDB.

3.2. Descripción

Como primer paso, se debe ingresar a la página web <https://mongodb.com> y registrarse.

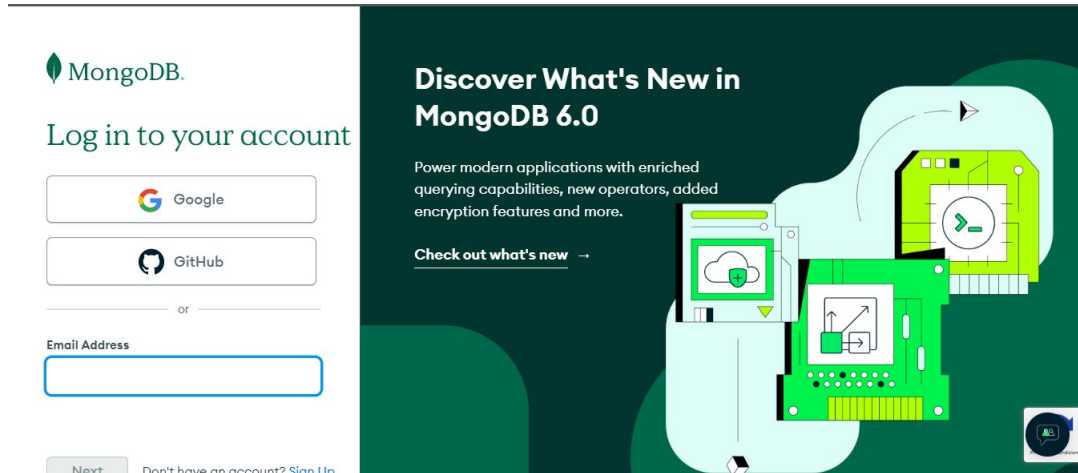


Figura 6: Página web MongoDB Atlas.

Después de haber iniciado sesión, daremos click al botón *Create* para crear un cluster al cuál podremos conectarnos para utilizar los servicios de Mongo Atlas.

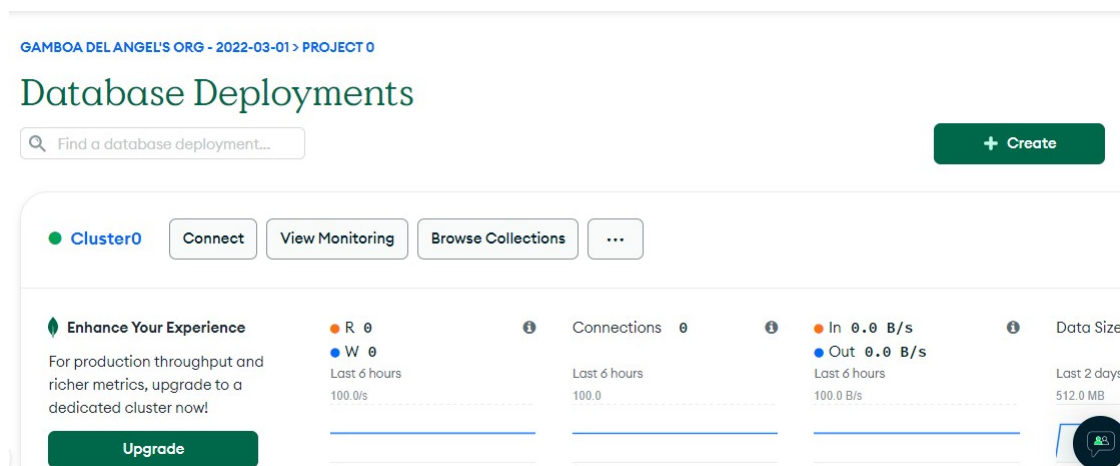


Figura 7: Página Principal MongoDB

Posteriormente, se seleccionarán los ajustes del cluster a crear, será un cluster compartido ya que esta opción es gratis.

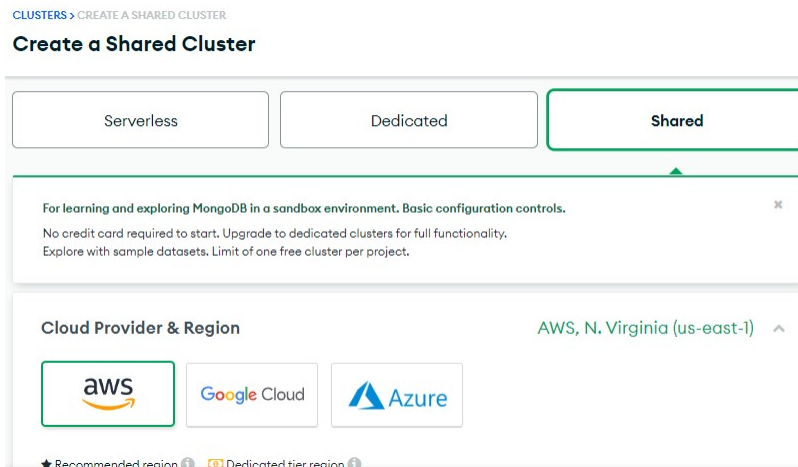


Figura 8: Página Creación de Cluster

De igual manera, se mantendrán los valores por default en cuanto al almacenamiento y la versión de Mongo a utilizar.

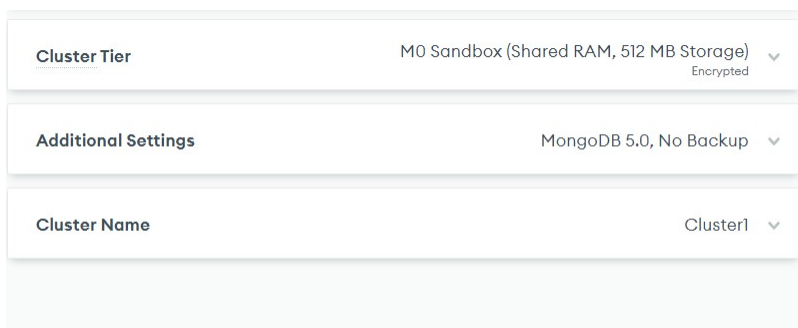


Figura 9: Página Creación de Cluster

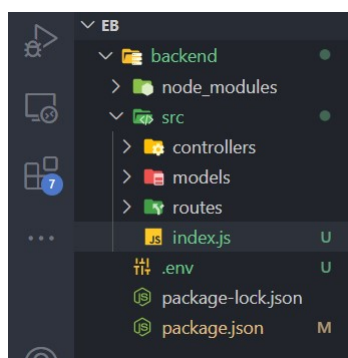


Figura 10: Directorio del Backend

```
backend > src > .\index.js > ...
1  const express = require("express");
2  const mongoose = require("mongoose");
3  require('dotenv').config({ path: 'env' });
4
5  const app = express();
6  const port = process.env.PORT || 9000;
7  const URI = 'mongodb+srv://root:250997@cluster0.zrhyx.mongodb.net/?retry
8
9  //iniciar servidor
10 app.get("/", (req,res) => {
11     res.send("Servidor Listo")
12 });
13
14
```

Figura 11: Directorio del Backend

```
15 //conexion mongo
16 mongoose.set("strictQuery", true);
17 mongoose.connect(URI, {
18     useNewUrlParser: true,
19     useUnifiedTopology: true
20 }).then(() => console.log("Conexión a mongo exitosa"))
21 .catch((error) => console.log(error));
22
23 //levantar servidor
24 app.listen(port,() => console.log("servidor escuchando en el puerto", port));
25
```

Figura 12: Directorio del Backend

3.3. Resultados

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER  COMMENTS  node

C:\Users\alang\Documents\TT\eb\backend>npm run start

> backend@1.0.0 start
> nodemon src/index.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 src/index.js`
servidor escuchando en el puerto 9000
Conexión a mongo exitosa
```

Figura 13: Directorio del Backend

4. Sistema de acceso con credenciales

4.1. Objetivo

Establecer los roles de cada tipo de usuario con sus respectivos permisos de acceso a la aplicación web utilizando Amazon Cognito

4.2. Descripción

Amazon Cognito funciona utilizando *pools* de usuarios. Un pool de usuarios es un directorio almacenado en los servicios de Amazon. Los beneficios que ofrece estar registrado en un pool de usuarios de Amazon Cognito son los siguientes:

- Servicio de registro e inicio de sesión
- Gestión del directorio de usuarios
- Servicios de seguridad tales como verificación de dos pasos
- Acceso a servicios de la suite de AWS tales como S3 o Dynamodb

Funcionamiento

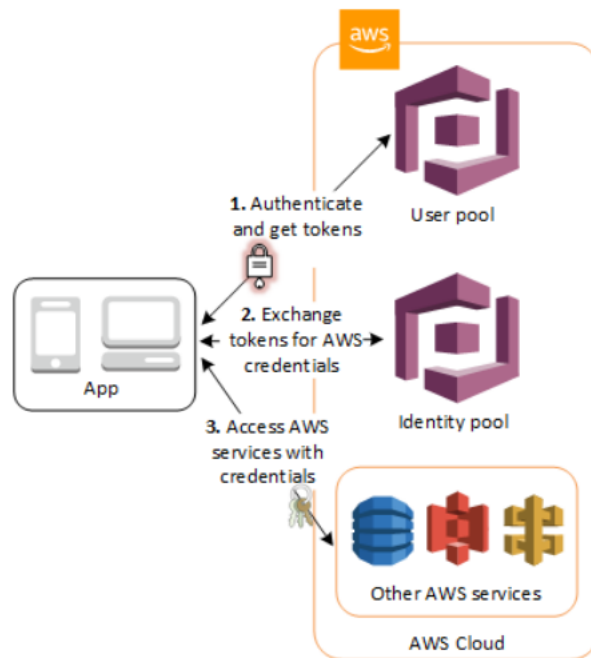


Figura 14: Directorio del Backend

- Como primer paso el usuario inicia sesión a través de un grupo de usuarios y recibe tokens del grupo de usuarios después de una autenticación exitosa.
- Posteriormente la aplicación intercambia los tokens del grupo de usuarios por las credenciales de AWS a través de un grupo de identidades.

- Finalmente, el usuario puede usar esas credenciales de AWS para acceder a otros servicios de AWS, como Amazon S3 o DynamoDB.

Implementación

Para poder hacer uso de Amazon Cognito en nuestra aplicación debemos de introducir el siguiente comando:

```
amplify add auth
```

[copy](#)

Figura 15: Directorio del Backend

Obteniendo el siguiente menú:

```
alan@alan-Inspiron-5548:~/Documentos/eb$ amplify update auth
Please note that certain attributes may not be overwritten if you choose to use defaults settings.

You have configured resources that might depend on this Cognito resource. Updating this Cognito resource could
have unintended side effects.

Using service: Cognito, provided by: awscloudformation
What do you want to do? Walkthrough all the auth configurations
Select the authentication/authorization services that you want to use: (Use arrow keys)
> User Sign-Up, Sign-In, connected with AWS IAM controls (Enables per-user Storage features for images or other
content, Analytics, and more)
  User Sign-Up & Sign-In only (Best used with a cloud API only)
  I want to learn more.
```

Figura 16: Directorio del Backend

Dejamos seleccionado la primera opción, la cual nos permitirá utilizar los servicios de autenticación ofrecidos por Amazon Cognito, además de otros servicios de la suite de AWS.

Finalmente utilizamos el comando *amplify push* para desplegar los cambios a nuestra aplicación.

4.3. Resultados

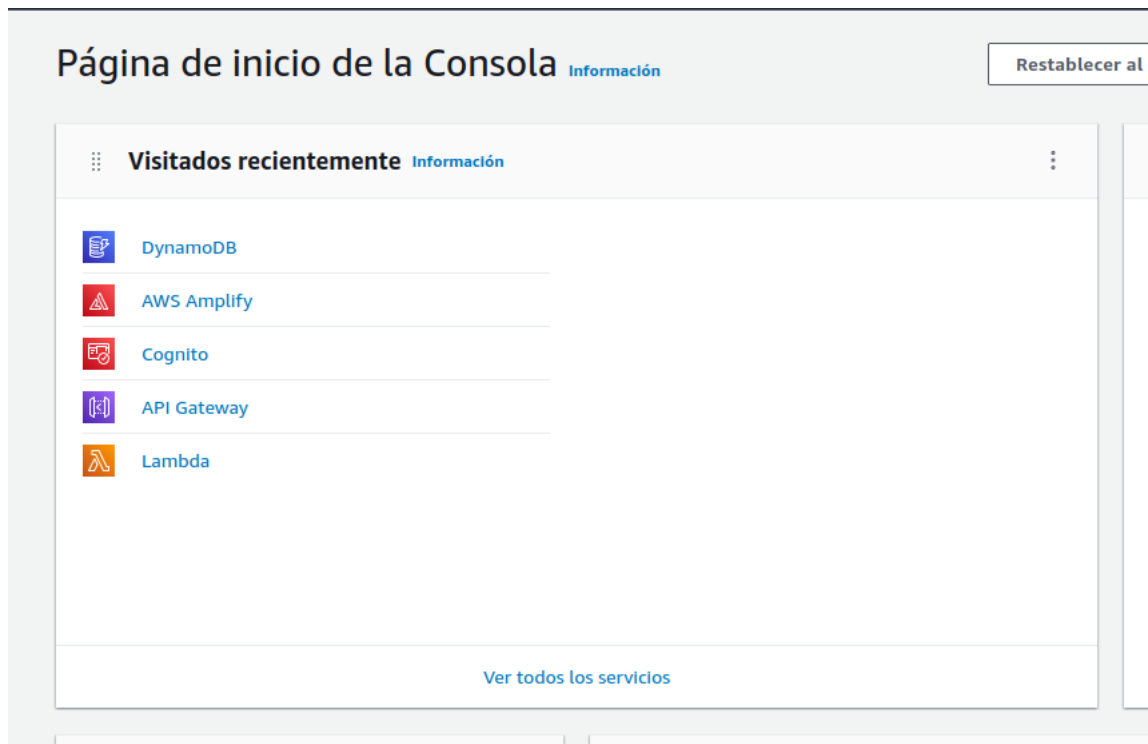


Figura 17: Directorio del Backend

Entrando a nuestra consola de AWS, en la sección de Amazon Cognito, se puede observar dos grupos de usuarios, uno de tipo Administrador, el cual puede realizar cambios a la configuración de la aplicación, y otro de tipo de Usuario, el cual sólo puede hacer uso del sistema de inicio de sesión y registro ofrecido por Cognito.

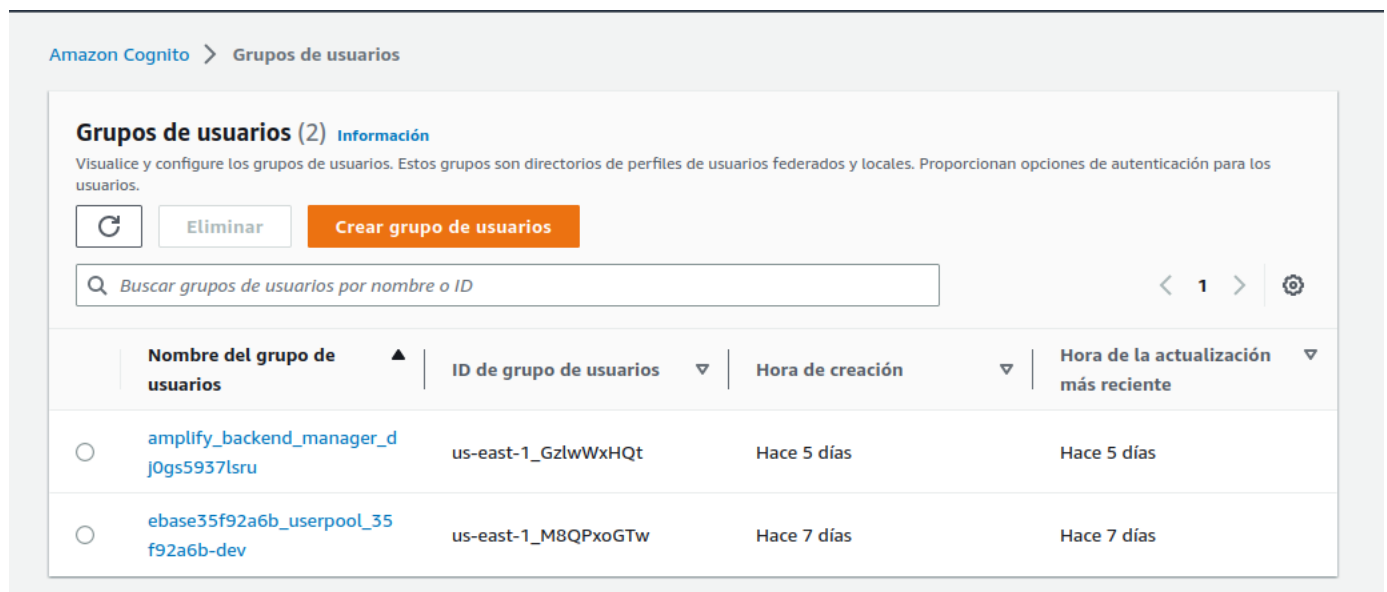


Figura 18: Grupos de usuario

Información general sobre el grupo de usuarios

Nombre del grupo de usuarios
amplify_backend_manager_dj0gs5937lsru

ID de grupo de usuarios
us-east-1_GzlwWxHQ

ARN
arn:aws:cognito-idp:us-east-1:387678068467:userpool/us-east-1_GzlwWxHQ

Número estimado de usuarios
1

Hora de creación
2 de marzo de 2023, 13:46 GMT-6

Hora de la actualización más reciente
2 de marzo de 2023, 13:46 GMT-6

Introducción

Usuarios

Grupos

Experiencia de inicio de sesión

Experiencia de inscripción

Mensajería

Integración de aplicaciones

Propiedades del grupo de usuarios

Usuarios (1) Información

Vea, edite y cree usuarios en el grupo de usuarios. Los usuarios habilitados y confirmados podrán iniciar sesión en el grupo de usuarios.

Nombre de usuario

Buscar usuarios por atributo

< 1 >

Nombre de usuario	Dirección de correo...	Correo electrónico ...	Estado de la confirmación	Estado
<input type="radio"/> aws-amplify-admin	-	No	Confirmado	Habilitado

Figura 19: Grupos de usuario

Usuarios

Grupos

Experiencia de inicio de sesión

Experiencia de inscripción

Mensajería

Integración de aplicaciones

Propied

Usuarios (5) Información

Vea, edite y cree usuarios en el grupo de usuarios. Los usuarios habilitados y confirmados podrán iniciar sesión en el grupo de usuarios.

Eliminar usuario

Crear usuario

Nombre de usuario

Buscar usuarios por atributo

< 1 >

Nombre de usuario	Dirección de correo elec...	Correo electrónico verifi...	Estado de la confirmación	Estado
<input type="radio"/> 19bd77dc-441b-4d3e-8...	maite.diazm98@gmail.com	Sí	Confirmado	Habilitado
<input type="radio"/> 473bad4c-b959-45e4-8...	alangam97@gmail.com	No	No confirmado	Habilitado
<input type="radio"/> 849ca9f3-6fd3-4750-a8...	mayte_dm@hotmail.com	No	No confirmado	Habilitado
<input type="radio"/> d1df2a48-8374-4fd2-be...	mayte_dm@hotmail.es	No	No confirmado	Habilitado
<input type="radio"/> e6bf3d4f-1807-4325-ba...	alangam97@gmail.com	Sí	Confirmado	Habilitado

Figura 20: Grupos de usuario

Proyecto Terminal 2

13

5. Creación de la base de datos No Relacional

5.1. Objetivo

Crear la base de datos en MongoDB.

5.2. Descripción

5.3. Resultados

6. Investigación de modelos de Redes Neuronales Convolutionales

6.1. Objetivo

Determinar distintos modelos de redes neuronales convolucionales que ofrezcan mejor eficiencia al clasificar imágenes.

6.2. Descripción

6.3. Resultados

7. Diseño de una red neuronal convolucional capaz de detectar ojos cerrados y abiertos

7.1. Objetivo

Diseñar y realizar pruebas de los modelos de redes neuronales convolucionales previamente investigados para determinar el rendimiento y la precisión de cada uno.

7.2. Descripción

7.3. Resultados

8. Conclusiones

Para el desarrollo de la Estación base, se decidió utilizar la suite de herramientas de Amazon Amplify. Se hará uso de Amplify Hosting, el cuál tiene una integración directa con Github, esto quiere decir, que los cambios que se realicen en el repositorio se reflejarán de manera automática en la aplicación web. Además, AWS Amplify ofrece su servicio de almacenamiento en la nube S3, este será de gran ayuda para almacenar contenido multimedia, en este caso los videos de incidencia de los conductores. Para el manejo de credenciales, se utilizará Amazon Cognito, que se encargará de administrar las credenciales de acceso a la aplicación. Para el manejo de datos, se estará utilizando MongoDB, un manejador NoSQL que trabaja con documentos. Para el desarrollo de la aplicación, se decidió utilizar el lenguaje de programación Javascript, junto con NodeJs que nos ayudará a manejar varias peticiones al mismo tiempo. Finalmente el análisis del sistema de comunicaciones, en un principio sólo se había contemplado el análisis de telemetría pero al ir realizando las actividades del tercer reporte e ir profundizando en algunos temas de comunicaciones, se decidió que también que se necesitaba el análisis de la cobertura y de los datos, es decir, de la transmisión de los fotogramas por lo que se incluyeron también en este reporte.

9. Bibliografia

Referencias