

## **PRESENTACIÓN:**

### **NOMBRE**

Luis Fermín 2021-1132  
Pamela Villar 2019-8458  
Reiphy Reyes 2019-8363  
Nahomi Núñez 2019-9048  
Samir González 2019-8341  
David Hernández 2020-9388

### **MAESTRO:**

**WILLIS EZEQUIEL POLANCO CARABALLO**

### **MATERIA**

**ELECTIVA 2**

# Índice

Crear cuenta en AWS .....	3
Activar política .....	4
Suscripción en MQTT .....	5
Creación de la base de datos en mongodb.....	5
Conectar Base de datos .....	6
Modificaciones de AWS en el código .....	7
Speaker.....	7
Smartphone vibre .....	8
Notificaciones .....	8

# Crear cuenta en AWS

Lo primero que realizamos fue crear nuestra cuenta en AWS

<https://aws.amazon.com/es/>

- Crear un objeto en AWS
- Se busca IoT Core
- Se le da a administracion
- Se selecciona objetos
- Se le da a crear objetos
- Selecciona crear un único objeto
- Se coloca el nombre del objeto y se dejan los otros parametros iguales y le damos a siguiente
- Generar automáticamente un certificado nuevo y le damos a siguiente



**Políticas (0)**  [Crear política](#)

Seleccione hasta 10 políticas para asociarlas a este certificado.

< 1 > 

<input type="checkbox"/>	Nombre
No hay políticas	
No se encontraron políticas en us-east-1.	

[Cancelar](#) [Anterior](#) [Crear un objeto](#)

- Se le da a crear politica
- Ponemos el nombre

## Crear política [Información](#)

Las políticas de AWS IoT Core permiten administrar el acceso a las operaciones del plano de datos de AWS IoT Core.

### Propiedades de la política

AWS IoT Core admite políticas con nombre para que muchas identidades puedan hacer referencia al mismo documento de políticas.

#### Nombre de la política

Un nombre de política es una cadena alfanumérica que también puede contener caracteres de punto (.), coma (,), guion (-), guion bajo (\_), signo más (+), signo igual (=) y arroba (@), pero sin espacios.

► **Etiquetas:** *opcional*

- Le damos a json y copiamos y pegamos el código
- Y le damos a crear política

Versiones
Destinos
Falta de conformidad
Etiquetas

Versión activa: 3
Creador
JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect",
        "iot:Receive",
        "iot:Publish",
        "iot:Subscribe"
      ],
      "Resource": "*"
    }
  ]
}

```

- luego seleccionamos la política que creamos y le damos a crear un objeto
- descargamos: Certificado, clave publica, clave privada y punto de enlace

## Activar la política

- Le damos a seguridad
- Luego a políticas
- Seleccionamos la política que estamos usando
- Y la activamos

Todas las versiones (1) <a href="#">Información</a>			
Las versiones activas y anteriores de esta política. Solo puede haber una versión activa. Una política no puede tener más de 5 versiones. Para actualizar una política con 5 versiones, primero debe eliminar una.			
	Eliminar	Establecer como activa	Editar versión
		Ver JSON	
<input type="checkbox"/>	Número de versión	Estado	Creado
<input type="checkbox"/>	1	✓ Activo	August 26, 2022, 12:49:00 (UTC-0400)

# Suscripción en MQTT

- Se le da a administracion
- Luego objetos
- Después seleccionamos el objeto que estamos utilizando
- Después le damos a actividad
- Y luego cliente de prueba de MQTT
- Después escribimos semaphore y luego a suscribir

Filtro de temas **Información**  
El filtro de temas describe los temas a los que usted se desea suscribir. El filtro de temas puede incluir caracteres comodín de MQTT.

semaphore

► Configuración adicional

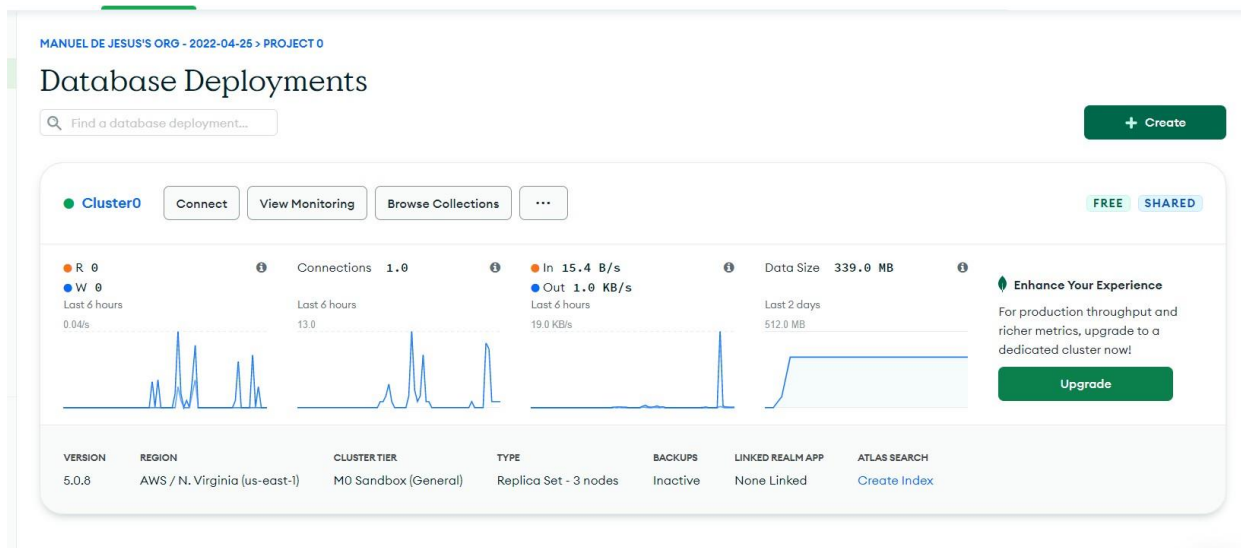
**Suscribir**

Suscripciones	semaphore
\$aws/things/Semaforo/shadow/name/+ /documents	▼ semaphore
\$aws/things/Semaforo/shadow/name/+ /update/+	{
\$aws/things/Semaforo/shadow/name/+ /delete/+	"light": "GREEN"
\$aws/things/Semaforo/shadow/update/documents	}
\$aws/things/Semaforo/shadow/+ /accepted	▼ semaphore
\$aws/things/Semaforo/shadow/+ /rejected	{
\$aws/events/presence/+ /Semaforo	"light": "RED"
\$aws/events/subscriptions/+ /Semaforo	}
semaphore	

# Creación de la base de datos en mongodb

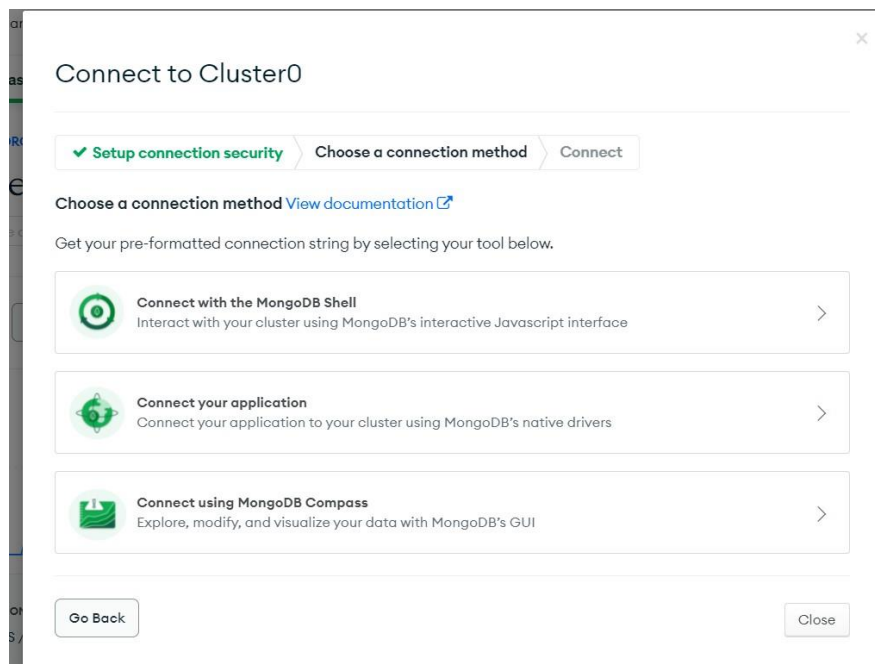
<https://www.mongodb.com/es>

- Entramos a mongodb
- Nos suscribimos
- Creamos un cluster
- Creamos un usuario
- Le damos a browse collections
- Luego le damos a create database
- Le ponemos nombre a la database y a la tabla



## Conectar Base de datos

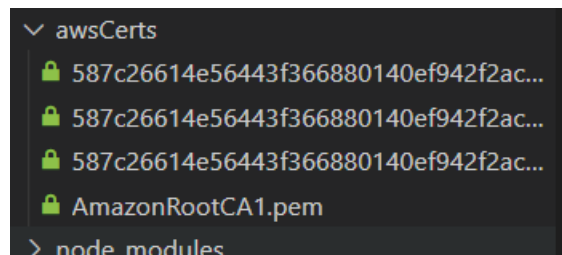
- En mongodb le damos a connect
- Luego a connect your application
- Luego copiamos el código que nos da la pagina
- Ese código lo copiamos en el programa en server.js en el metodo mongoose.connect
- La parte <password> la cambiamos por la contraseña del usuario



```
mongoose.connect("mongodb+srv://electiva10:electiva10@cluster0.y4ore.mongodb.net/myFirstDatabase?retryWrites=true&w=majority", {
  useNewUrlParser: true,
  useCreateIndex: true,
});
```

## Modificaciones de AWS en el código

- En la carpeta awsCerts borrar los documentos que ya estan puestos
- Y poner los certificados y claves que descargamos de AWS



- En awsService.js pones la ruta de acceso relativa del certificado en certpath
- Ponemos la ruta de acceso relativa de la clave privada en keypath
- Ponemos la ruta de acceso relativa de permiso de amazon en capath
- En el host ponemos nuestro punto de anclaje (que se consigue configuracion en AWS)
- En clientID ponemos el nombre del objeto
- En la region ponemos la region de nuestra cuenta de AWS

```
const awsIot = require("aws-iot-device-sdk");

const device = awsIot.device({
  keyPath: './awsCerts/587c26614e56443f366880140ef942f2ac3990d85d9ed7a54089edda8b29d640-private.pem.key',
  certPath: './awsCerts/587c26614e56443f366880140ef942f2ac3990d85d9ed7a54089edda8b29d640-certificate.pem.crt',
  caPath: './awsCerts/AmazonRootCA1.pem',
  host: 'a1hjk7t7haucq0-ats.iot.us-east-1.amazonaws.com',
  clientId: 'Semaforo',
  region: 'us-east-1',
});

module.exports = { device };
```

## Speaker

- Utilizando speechsystemutterance



```
var speaker = new SpeechSynthesisUtterance();
speaker.text = totalTime;
speaker.rate = 1.7;
speechSynthesis.speak(speaker);
```

## Smartphone vibre

- Utilizando webkit keyframes

CSS

```
.vibracion {
  position: absolute;
  top: 25%;
  left: 10px;
  -webkit-animation: tiembla 0.1s infinite;
}

@-webkit-keyframes tiembla{
  0% { -webkit-transform:rotateZ(-5deg); }
  50% { -webkit-transform:rotateZ( 0deg) scale(.8); }
  100%{ -webkit-transform:rotateZ( 5deg); }
}
```

JavaScript

```
var tele = document.getElementById('telefono');
tele.className = 'vibracion';
```

## Notificaciones

- Utilizamos un alert

```
alert("No cruzar");
```