

FATEC de Registro

Desenvolvimento de Software Multiplataforma

Adeldivo Alves de Sousa Junior

Aline Coelho Lauriano

Amanda Nogueira de Castro e Silva

CN1 - Atividade 2

AWS Lambda

Registro

2024

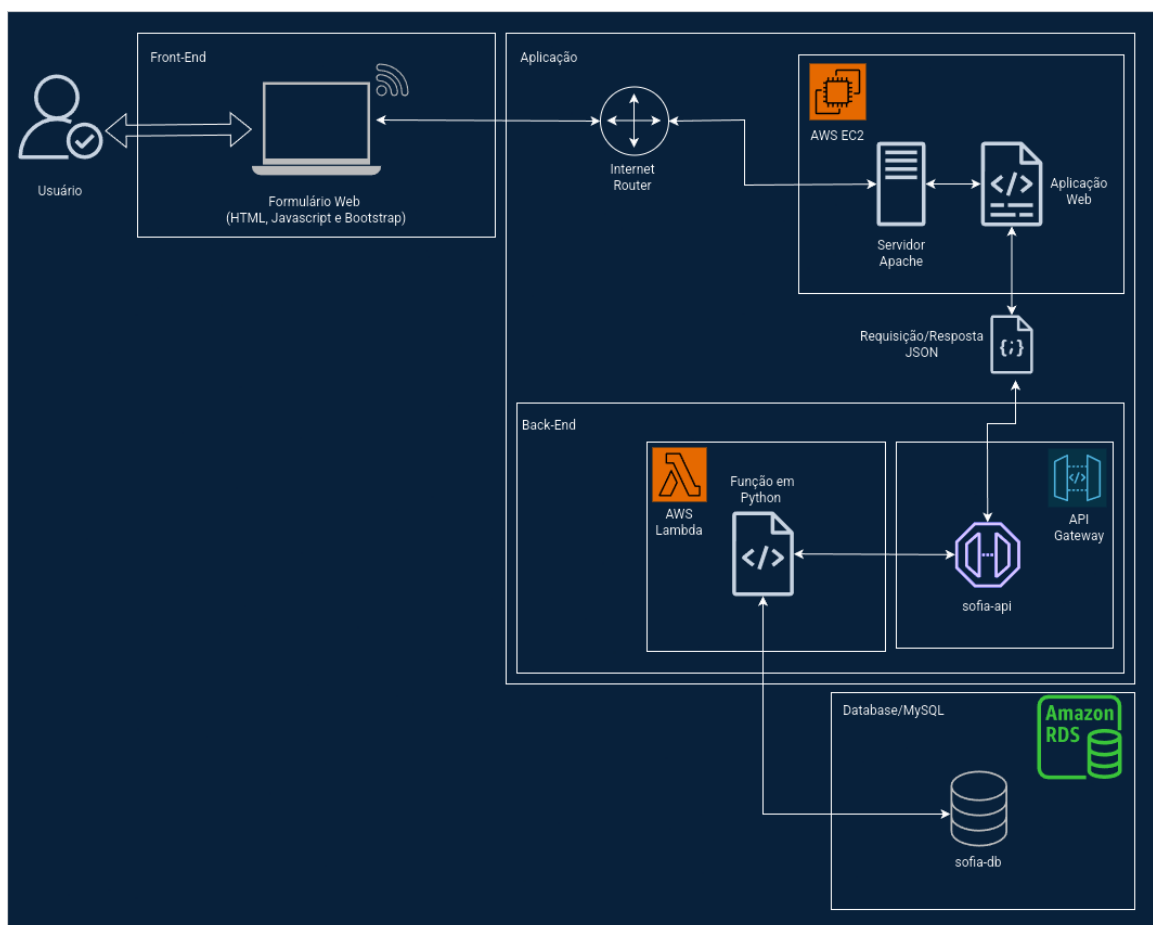
Sumário

Sumário	2
1 SOBRE O PROJETO	3
2 CRIAÇÃO DA INSTÂNCIA DO BANCO DE DADOS NO AMA- ZON RDS	4
3 CRIAÇÃO DA FUNÇÃO LAMBDA E O PROXY	8
3.1 Como criar uma função e um proxy associados	8
3.2 Criação do perfil de execução de função	11
4 CRIAÇÃO DO PACOTE DE IMPLANTAÇÃO DO LAMBDA	14
4.1 Implementação do código em Python	14
5 CRIAÇÃO DA API GATEWAY PARA IMPLEMENTAÇÃO DA FUNÇÃO LAMBDA	21
5.1 Criação da API e do método POST para envio dos dados para cadastro	21
5.2 Configuração do CORS	23
6 CRIAÇÃO DA INSTÂNCIA NO EC2	26
7 EXECUÇÃO DA APLICAÇÃO WEB	29

1 SOBRE O PROJETO

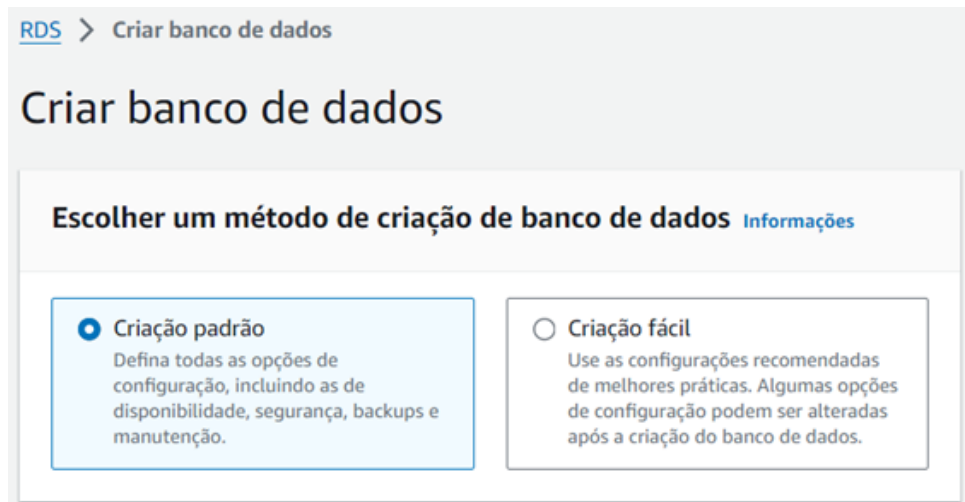
Este trabalho descreve a implementação de uma solução de computação em nuvem utilizando a plataforma **AWS**. Foi desenvolvida uma função **Lambda** em **Python** responsável pelo cadastro de dados em um banco de dados **MySQL**, hospedado no serviço RDS. A integração da função Lambda com o front-end é realizada através de uma **API Gateway**, que recebe dados de um formulário HTML construído com Bootstrap e hospedado em um servidor Apache, operando em uma instância **EC2**. O processo é completado com o método **POST**, que retorna como resposta uma mensagem de sucesso ao usuário.

- Os capítulos 1 a 4 deste documento foram baseados no tutorial disponível em [**AWS: Tutorial Lambda e RDS.**](#)
- Os códigos utilizados neste projeto estão disponíveis em [**aasjunior/aws_atv2.**](#)



2 CRIAÇÃO DA INSTÂNCIA DO BANCO DE DADOS NO AMAZON RDS

1. No console do Amazon RDS, selecione **Criar banco de dados**.



2. Deixe a opção **Criação padrão** selecionada e, em **Opções do mecanismo**, escolha MySQL.



3. Na seção **Modelos**, escolha **Nível gratuito**.



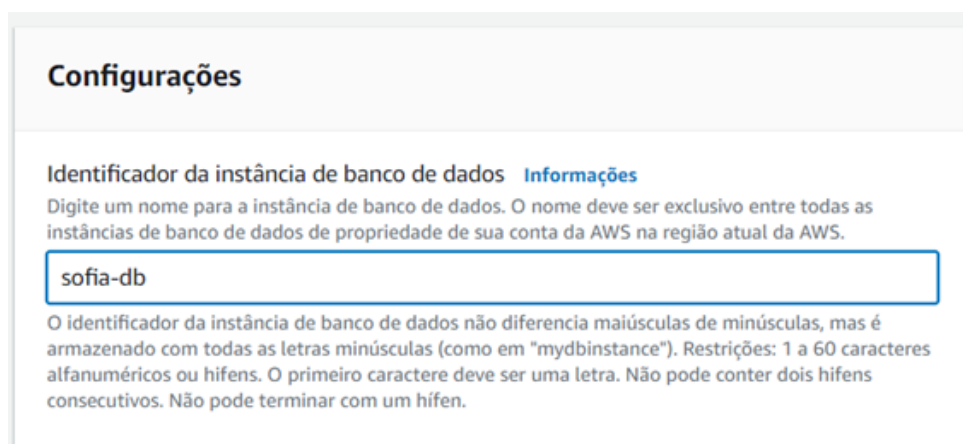
Modelos
Escolha um modelo de exemplo para atender a seu caso de uso.

☐ **Produção**
Use padrões para alta disponibilidade e desempenho rápido e consistente.

☐ **Dev/Test**
Esta instância é planejada para uso de desenvolvimento fora de um ambiente de produção.

☒ **Nível gratuito**
Use o nível gratuito do RDS para desenvolver novas aplicações, testá-las ou obter uma experiência prática com o Amazon RDS. [Informações](#)

4. Em **Configurações**, para o **identificador da instância de banco de dados**, digite **sofia-db**.



Configurações

Identificador da instância de banco de dados [Informações](#)
Digite um nome para a instância de banco de dados. O nome deve ser exclusivo entre todas as instâncias de banco de dados de propriedade de sua conta da AWS na região atual da AWS.

sofia-db

O identificador da instância de banco de dados não diferencia maiúsculas de minúsculas, mas é armazenado com todas as letras minúsculas (como em "mydbinstance"). Restrições: 1 a 60 caracteres alfanuméricos ou hifens. O primeiro caractere deve ser uma letra. Não pode conter dois hifens consecutivos. Não pode terminar com um hífen.

5. Para definir seu nome de usuário e senha, faça o seguinte:

- a. Em **Configurações de credenciais**, deixe o **Nome do usuário principal** definido como **admin**.
- b. Para a **Senha mestra**, digite e confirme uma senha para acessar seu banco de dados.

▼ **Configurações de credenciais**

Nome do usuário principal [Informações](#)
Digite um ID de login para o usuário principal de sua instância de banco de dados.

admin

De um a 16 caracteres alfanuméricos. O primeiro caractere deve ser uma letra.

Gerenciamento de credenciais
Você pode usar o AWS Secrets Manager ou gerenciar suas credenciais de usuário principal.

☐ **Gerenciado no AWS Secrets Manager - mais seguro**
O RDS gera uma senha para você e a gerencia durante todo o ciclo de vida usando o AWS Secrets Manager.

☒ **Autogerenciada**
Crie sua própria senha ou faça com que o RDS crie uma senha para você gerenciar.

☐ **Gerar senha automaticamente**
O Amazon RDS pode gerar uma senha para você, ou você pode especificar sua própria senha.

Senha principal [Informações](#)

.....

Restrições mínimas: pelo menos 8 caracteres ASCII imprimíveis. Não pode conter nenhum dos seguintes símbolos: / ' " @

Confirmar senha principal [Informações](#)

.....

6. Para especificar o nome do banco de dados, faça o seguinte:
- Deixe todas as opções padrão restantes selecionadas e role para baixo até a seção **Configuração adicional**.
 - Expanda essa seção e insira **sofiaDB** como o **Nome do banco de dados inicial**.

▼ Configuração adicional
Opções de banco de dados, criptografia ativado, backup ativado, retroceder desativado, manutenção, CloudWatch Logs, excluir proteção desativado.

Opções de banco de dados

Nome do banco de dados inicial [Informações](#)

sofiaDB

Se você não especificar um nome de banco de dados, o Amazon RDS não criará um banco de dados.

Grupo de parâmetros do banco de dados [Informações](#)

default.mysql8.0 ▼

Grupo de opções [Informações](#)

default:mysql-8-0 ▼

7. Deixe todas as opções padrão restantes selecionadas e escolha Criar banco de dados.

Custos mensais estimados

O nível gratuito do Amazon RDS ficará disponível para você por 12 meses. A cada mês, o nível gratuito permite o uso gratuito dos recursos do Amazon RDS listados abaixo:

- 750 horas do Amazon RDS em uma instância db.t2.micro, db.t3.micro ou db.t4g.micro Single-AZ.
- 20 GB de armazenamento de uso geral (SSD).
- 20 GB de armazenamento para backup automatizado e qualquer snapshot de banco de dados iniciado pelo usuário.

[Saiba mais sobre o nível gratuito da AWS.](#)

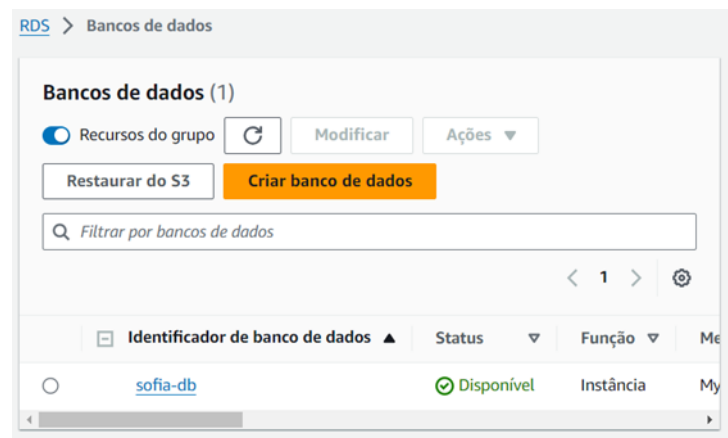
Quando seu uso gratuito expirar ou se seu uso do aplicativo ultrapassar os níveis de uso gratuito, você simplesmente pagará taxas de serviço padrão de acordo com o uso, conforme descrito na [página de preços do Amazon RDS.](#)

ⓘ Você é responsável por garantir que tem todos os direitos necessários para todos os produtos ou serviços de terceiros usados com os serviços da AWS.

Cancelar **Criar banco de dados**

3 CRIAÇÃO DA FUNÇÃO LAMBDA E O PROXY

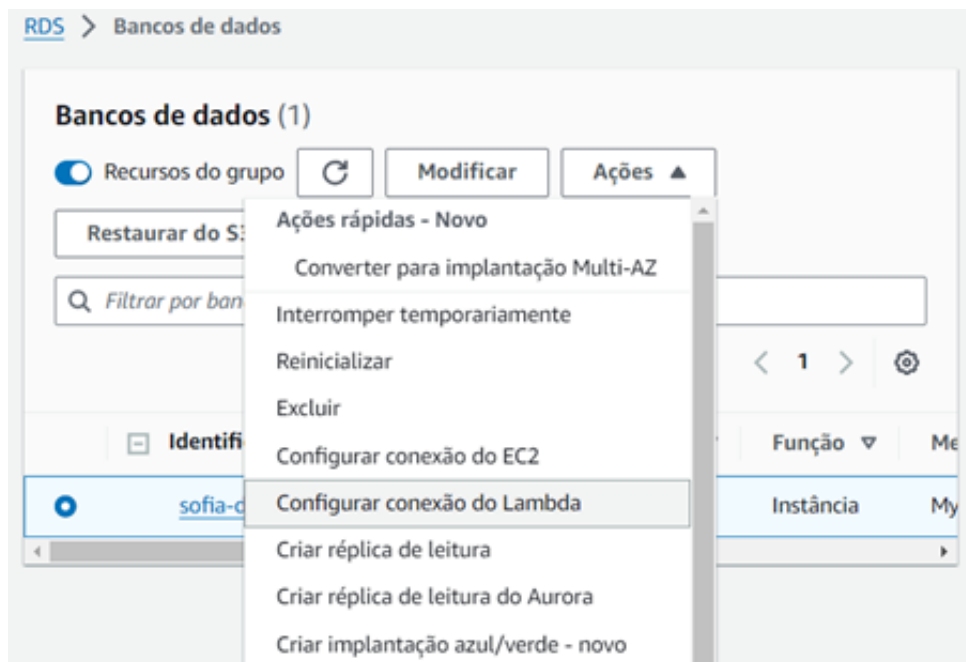
É possível usar o console do RDS para criar uma função do Lambda e um proxy na mesma VPC (Virtual Private Cloud) do banco de dados. Para isso, é preciso aguardar até que o banco de dados esteja com o status **Disponível**.



3.1 Como criar uma função e um proxy associados

1. Na página **Bancos de dados**, confira se o banco de dados está no status **Disponível**. Se estiver, vá para a próxima etapa. Caso contrário, espere até que o banco de dados esteja disponível.

2. Selecione o banco de dados e escolha **Configurar conexão do Lambda** em **Ações**.



3. Na página **Configurar conexão do Lambda**, selecione **Criar nova função**. Defina o **Nome da nova função do Lambda** como **LambdaSofiaDB**.

Configurar conexão do Lambda Informações

Simplifique a conexão do seu banco de dados do RDS com a sua função do Lambda.

Selecionar função do Lambda

Banco de dados
sofia-db

Função do Lambda

Escolha a função do Lambda para se conectar a esse banco de dados. Somente as funções do Lambda na mesma VPC do banco de dados são mostradas. Se nenhuma função do Lambda estiver disponível na mesma VPC, você poderá criar uma nova função do Lambda na mesma VPC.

☐ Escolher função existente
Conecte-se a uma função existente na mesma VPC.

☒ Criar nova função
Crie automaticamente uma nova função de amostra na mesma VPC.

Nome da função Informações

Insira um nome que descreva a finalidade da sua função.

O nome da função pode ter até 64 caracteres. Caracteres válidos: A-Z, a-z, 0-9, _ (sublinhado) e - (hifen)

[Criar manualmente a função do Lambda](#)

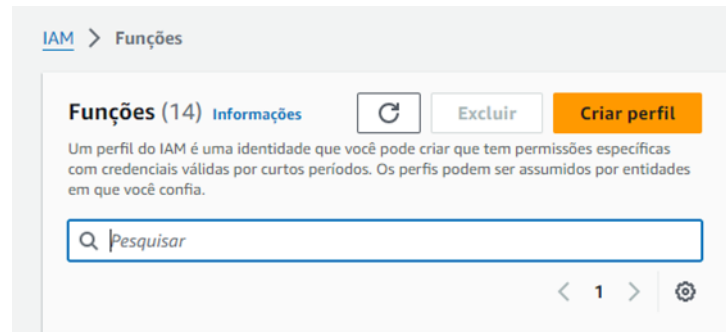
4. Na seção **RDS Proxy**, selecione a opção **Conectar usando o RDS Proxy**. Selecione ainda **Criar proxy**.

- Em **Credenciais do banco de dados**, selecione **Nome do usuário e senha do banco de dados**.
- Em **Nome do usuário**, especifique **admin**.
- Em **Senha**, digite a senha que você criou para a instância de banco de dados.

5. Selecione **Configurar** para concluir a criação da função do Lambda e do proxy. O assistente conclui a configuração e fornece um link para o console do Lambda para você examinar a nova função. Observe o endpoint do proxy antes de mudar para o console do Lambda.

3.2 Criação do perfil de execução de função

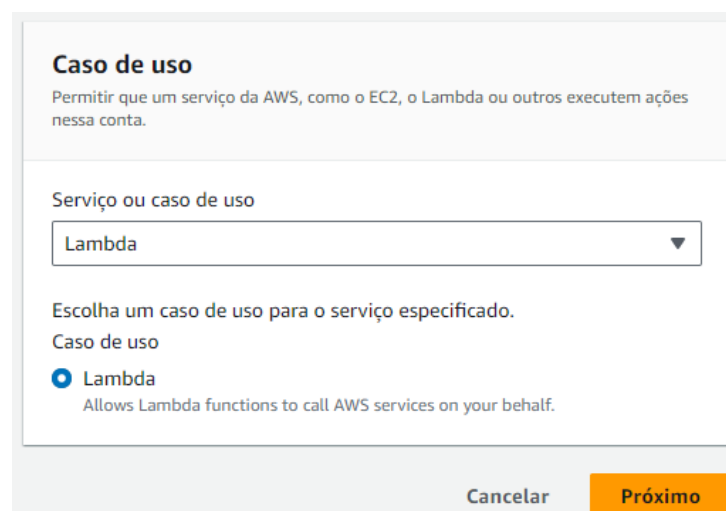
1. Abra a página **Perfis** do console do IAM e escolha Criar perfil



2. Em **Tipo de entidade confiável**, selecione **Serviço da AWS**.
3. Escolha **Próximo**

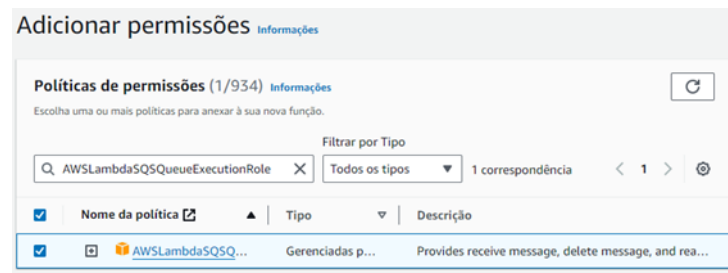


4. Em **Caso de uso**, escolha **Lambda**.

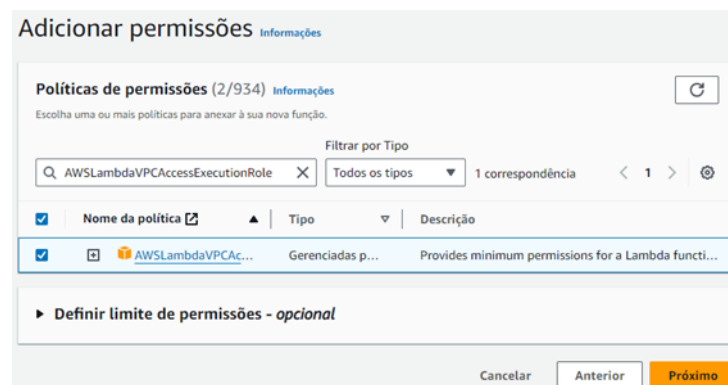


5. Para adicionar as políticas gerenciadas do IAM, faça o seguinte:

- Usando a caixa de pesquisa de políticas, pesquise **AWSLambdaSQSQueueExecutionRole**.
- Na lista de resultados, marque a caixa de seleção ao lado do perfil e escolha **Limpar filtros**.



- Usando a caixa de pesquisa de políticas, pesquise **AWSLambdaVPCLambdaAccessExecutionRole**.
- Na lista de resultados, marque a caixa de seleção ao lado do perfil e escolha **Próximo**.



6. Em Nome do perfil, digite **lambda-vpc-sqs-role**.

Nomear, revisar e criar

Detalhes da função

Nome da função
Insira um nome significativo para identificar esta função.

Máximo de 64 caracteres. Use caracteres alfanuméricos e '+', '@', '-'.

Descrição
Adicione uma breve explicação para esta função.

Máximo de 1000 caracteres. Use letras (A-Z e a-z), números (0-9), tabulações, novas linhas ou qualquer um dos seguintes caracteres: '_', '+', '@', '-', '/', '[', ']', '#', '\$', '%', '&', '*', '!', '~', '^', '<', '>'.

7. Escolha **Criar perfil**.

The screenshot shows the 'Etapa 2: Adicionar permissões' (Step 2: Add permissions) screen in the AWS IAM console. It features a table titled 'Resumo da política de permissões' (Policy summary) with columns for 'Nome da política' (Policy name), 'Tipo' (Type), and 'Anexado como' (Attached as). Two policies are listed: 'AWSLambdaSQSQueueExecutionRole' and 'AWSLambdaVPCLambdaAccessExecutionRole', both managed by AWS and attached as 'Política de permissões' (Permission policy). Below this is 'Etapa 3: Adicionar tags' (Step 3: Add tags), which includes a section for adding optional tags and a button to 'Adicione uma nova tag' (Add a new tag). At the bottom, there are three buttons: 'Cancelar' (Cancel), 'Anterior' (Previous), and 'Criar perfil' (Create profile).

Nome da política	Tipo	Anexado como
AWSLambdaSQSQueueExecutionRole	Gerenciadas pela AWS	Política de permissões
AWSLambdaVPCLambdaAccessExecutionRole	Gerenciadas pela AWS	Política de permissões

Etapa 3: Adicionar tags

Adicionar tags - *opcional* [Informações](#)

Tags são pares de chave/valor que você pode adicionar aos recursos da AWS para ajudar a identificar, organizar ou pesquisar recursos.

Nenhuma tag associada ao recurso.

[Adicione uma nova tag](#)

Você pode adicionar até mais 50 tags.

[Cancelar](#) [Anterior](#) [Criar perfil](#)

Posteriormente no tutorial, você precisará do nome do recurso da Amazon (ARN) do perfil de execução que acabou de criar. Para encontrar o ARN do perfil de execução

1. Abra a página **Perfis** do console do IAM e escolha o perfil (lambda-vpc-sqs-role).
2. Copie o **ARN** exibido na seção **Resumo**.

The screenshot shows the 'lambda-vpc-sql-role' page in the AWS IAM console. It includes a breadcrumb trail 'IAM > Funções > lambda-vpc-sql-role' and a title 'lambda-vpc-sql-role' with an 'Informações' (Information) link. Below the title is a description: 'Allows Lambda functions to call AWS services on your behalf.' There is an 'Excluir' (Delete) button. The 'Resumo' (Summary) section contains a table with details about the role. An 'Editar' (Edit) button is also present.

Resumo	
Data de criação	ARN
June 08, 2024, 12:38 (UTC-03:00)	am:aws:iam::637423181495:role/lambda-vpc-sql-role
Última atividade	Duração máxima da sessão
-	1 hora

[Excluir](#)

[Editar](#)

4 CRIAÇÃO DO PACOTE DE IMPLANTAÇÃO DO LAMBDA

O exemplo de código Python a seguir usa o pacote PyMySQL para abrir uma conexão com o banco de dados. Na primeira vez que você invoca a função, ela também cria uma nova tabela chamada Patient. A tabela usa o seguinte esquema, em que PatientID é a chave primária:

```
Patient(PatientID, FirstName, LastName, BirthDate, Gender)
```

4.1 Implementação do código em Python

1. **Configuração do Logger:** O código começa configurando um logger, que é usado para registrar informações sobre a execução do script.
2. **Configurações do RDS:** Em seguida, o código recupera as configurações do banco de dados RDS a partir das variáveis de ambiente. Isso inclui o nome de usuário, senha, host do proxy RDS e o nome do banco de dados.
3. **Verificação das Variáveis de Ambiente:** O código verifica se todas as variáveis de ambiente necessárias estão definidas. Se alguma delas estiver faltando, o código registra um erro e termina a execução.

4. **Função Lambda Handler:** Esta é a função principal que é chamada quando o Lambda é acionado. Ela recebe um evento e um contexto como parâmetros.

- **Conexão com o Banco de Dados:** A função tenta estabelecer uma conexão com o banco de dados MySQL usando as configurações do RDS.
- **Processamento do Evento:** A função verifica se o evento contém uma chave 'body'. Se 'body' for uma string, ela é convertida em um dicionário. Se 'body' já for um dicionário, ele é usado diretamente. Se a chave 'body' não existir, o evento é usado diretamente.
- **Preparação dos Dados:** A função extrai os dados do dicionário para inserção no banco de dados.
- **Operações SQL:** A função executa várias operações SQL. Primeiro, ela cria uma tabela chamada 'Patient' se ela não existir. Em seguida, ela insere os dados na tabela 'Patient'. Finalmente, ela recupera todos os registros da tabela 'Patient' e os registra.
- **Tratamento de Erros:** Se ocorrer um erro durante as operações do banco de dados, a função registra o erro.
- **Fechamento da Conexão:** Antes de a função terminar, ela verifica se a conexão com o banco de dados ainda está aberta. Se estiver, ela fecha a conexão.
- **Resposta da Função:** A função retorna uma resposta indicando que os registros foram adicionados com sucesso à tabela do banco de dados.

```

1 import sys
2 import logging
3 import pymysql
4 import json
5 import os
6
7 # Configuração do logger
8 logger = logging.getLogger()
9 logger.setLevel(logging.INFO)
10
11 # rds settings
12 user_name = os.environ.get('USER_NAME')
13 password = os.environ.get('PASSWORD')
14 rds_proxy_host = os.environ.get('RDS_PROXY_HOST')
15 db_name = os.environ.get('DB_NAME')
16
17 # Verifica se todas as variáveis de ambiente necessárias estão definidas
18 if not all([user_name, password, rds_proxy_host, db_name]):
19     logger.error("ERROR: Environment variables are not set properly.")
20     sys.exit(1)
21
22 def lambda_handler(event, context):
23     """
24     This function fetches data from the event and writes records to the RDS MySQL table.
25     """
26     # Estabelece a conexão com o banco de dados RDS MySQL
27     try:
28         conn = pymysql.connect(host=rds_proxy_host, user=user_name, passwd=password, db=db_name, connect_timeout=5)
29         logger.info("SUCCESS: Connection to RDS for MySQL instance succeeded")
30
31         # Verifica se a chave 'body' existe no evento
32         if 'body' in event:
33             # Se 'body' for uma string, converte para um dicionário
34             if isinstance(event['body'], str):
35                 data = json.loads(event['body'])
36             else:
37                 # Se 'body' já for um dicionário, usa diretamente
38                 data = event['body']
39         else:
40             # Se a chave 'body' não existir, usa o evento diretamente
41             data = event
42
43         # Prepara os dados para inserção
44         FirstName = data.get('FirstName')
45         LastName = data.get('LastName')
46         BirthDate = data.get('BirthDate')
47         Gender = data.get('Gender')
48
49         # String SQL para inserção de dados
50         sql_string = 'INSERT INTO Patient (FirstName, LastName, BirthDate, Gender) VALUES (%s, %s, %s, %s)'
51
52         # Executa as operações SQL
53         with conn.cursor() as cur:
54             cur.execute('''
55                 CREATE TABLE IF NOT EXISTS Patient (
56                     PatientID INT NOT NULL AUTO_INCREMENT,
57                     FirstName VARCHAR(255) NOT NULL,
58                     LastName VARCHAR(255) NOT NULL,
59                     BirthDate VARCHAR(10) NOT NULL,
60                     Gender VARCHAR(6) NOT NULL,
61                     PRIMARY KEY (PatientID)
62                 )
63             ''')
64             cur.execute(sql_string, (FirstName, LastName, BirthDate, Gender))
65             conn.commit()
66             cur.execute('SELECT * FROM Patient')
67             logger.info('The following items have been added to the database:')
68             for row in cur:
69                 logger.info(row)
70             conn.commit()
71     except pymysql.MySQLError as e:
72         logger.error("ERROR: Unexpected error during database operations.")
73         logger.error(e)
74     finally:
75         # Verifica se a conexão ainda está aberta antes de tentar fechá-la
76         if conn.open:
77             conn.close()
78
79     # Retorna uma mensagem de sucesso
80     return {
81         "statusCode": 200,
82         "body": json.dumps("Records successfully added to RDS for MySQL table")
83     }
84

```

Para criar um pacote de implantação .zip

1. Salve o exemplo de código como um arquivo denominado **lambda_function.py**.
2. No mesmo diretório em que você criou seu arquivo **lambda_function.py**, crie um novo diretório chamado **package** e instale a biblioteca PyMySQL.


```
mkdir package
pip install --target package pymysql
```

3. Crie um arquivo zip contendo o código do seu aplicativo e a biblioteca PyMySQL. No Linux ou no macOS, execute os comandos da CLI a seguir. No Windows, use sua ferramenta de zip preferida para criar o arquivo **lambda_function.zip**. O arquivo de código-fonte **lambda_function.zip** e as pastas que contêm as dependências devem ser instaladas na raiz do arquivo .zip.

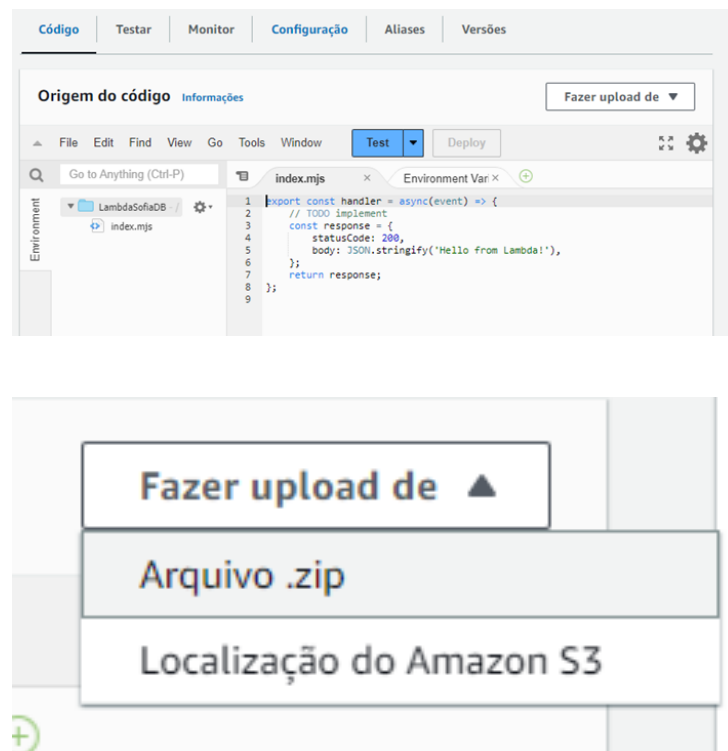
```
cd package
zip -r ../lambda_function.zip .
cd ..
zip lambda_function.zip lambda_function.py
```

Como atualizar a função do Lambda

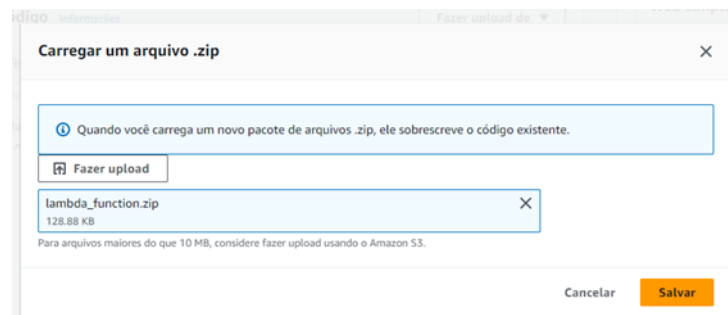
1. Abra a página **Funções** do console do Lambda e selecione sua função **LambdaSofiaDB**.
2. Na guia Configurações de tempo de execução, selecione Editar para alterar o Tempo de execução da função para Python 3.10.
3. Altere o **Manipulador** para **lambda_function.lambda_handler**.

The screenshot shows the 'Editar configurações de tempo de execução' (Edit execution configuration) page in the AWS Lambda console. The breadcrumb trail at the top is 'Lambda > Funções > LambdaSofiaDB > Editar configurações de tempo de execução'. The main title is 'Editar configurações de tempo de execução'. Below this, there are two tabs: 'Configurações de tempo de execução' (selected) and 'Informações'. The 'Configurações de tempo de execução' section has a sub-header 'Tempo de execução' and a note: 'Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.' A dropdown menu shows 'Python 3.10' with a refresh icon to its right. Below this is a blue notification box with a circular icon containing an 'i' and the text: 'Novo tempo de execução disponível. Um novo tempo de execução está disponível para o idioma de sua função. Python 3.12'. The 'Manipulador' section has a sub-header 'Manipulador' and a text input field containing 'lambda_function.lambda_handler'. Below this is the 'Arquitetura' section with a sub-header 'Arquitetura' and a note: 'Escolha a arquitetura do conjunto de instruções desejada para o código da função.' There are two radio buttons: 'x86_64' (selected) and 'arm64'. At the bottom of the configuration section is another blue notification box with a circular icon containing an 'i' and the text: 'Você pode alterar o tempo de execução da função ou a arquitetura do conjunto de instruções em uma só atualização. Para atualizar ambos, repita o processo de atualização.' At the bottom right of the page are two buttons: 'Cancelar' and 'Salvar'.

4. Na guia **Código**, selecione **Fazer upload de** e, depois, **arquivo .zip**.



5. Selecione o arquivo **lambda_function.zip** que você criou no estágio anterior e escolha **Salvar**.



Como configurar o perfil de execução da função.

1. Na página **Funções** do console do Lambda, selecione a guia **Configuração** e escolha **Permissões**.
2. No painel **Papel de execução**, selecione **Editar**.
3. Em **Função existente**, selecione a função de execução (**lambda-vpc-sqs-role**).
4. Escolha **Salvar**.

Código | Testar | Monitor | **Configuração** | Aliases | Versões

Configuração geral | Gatilhos | **Permissões** | Destinos | URL da função | Variáveis de ambiente | Tags

Papel de execução [🔄] [Editar] [Visualizar documento de função]

Nome da função
LambdaSofiaDB-1717859241638 [🔗](#)

Resumo de recursos

Para visualizar os recursos e as ações que sua função tem permissão de acessar, escolha um serviço.

[🔗](#) Amazon CloudWatch Logs
3 actions, 1 resource

Função existente
Escolha uma função existente que você criou para ser usada com esta função do Lambda. A função deve ter permissão para fazer upload de logs no Amazon CloudWatch Logs.

[🔗](#) lambda-vpc-sql-role [🔄]

[Visualize a função lambda-vpc-sql-role no console do IAM.](#)

Cancelar **Salvar**

Como configurar as variáveis de ambiente da função

1. Na página **Funções** do console do Lambda, selecione a guia **Configuração** e, em seguida, escolha **Variáveis de ambiente**.
2. Selecione a opção **Editar**.

[Lambda](#) > [Funções](#) > [LambdaSofiaDB](#) > Editar variáveis de ambiente

Editar variáveis de ambiente

Variáveis de ambiente

Você pode definir as variáveis de ambiente como pares de chave-valor acessíveis a partir do código de função. Elas são úteis para armazenar parâmetros de configuração sem precisar alterar o código de função. [Saiba mais](#) [🔗](#)

Não há variáveis de ambiente nessa função.

[Adicionar variáveis de ambiente](#)

► Configuração da criptografia

Cancelar **Salvar**

3. Para adicionar as credenciais de acesso ao banco de dados, faça o seguinte:
- Escolha **Adicionar variável de ambiente** e, em seguida, para **Chave**, insira **USER_NAME**, e para **Valor**, insira **admin**.
 - Escolha **Adicionar variável de ambiente** e, em seguida, para **Chave**, insira **DB_NAME**, e para **Valor**, insira **sofiaDB**.
 - Escolha **Adicionar variável de ambiente** e, em seguida, para **Chave**, insira **PASSWORD**, e para **Valor**, insira a senha escolhida ao criar seu banco de dados.
 - Selecione **Adicionar variáveis de ambiente**. Em **Chave**, insira **RDS_PROXY_HOST** e, em **Valor**, insira o endpoint do RDS Proxy que você anotou anteriormente.
 - Escolha **Salvar**.

Lambda > Funções > LambdaSofiaDB > Editar variáveis de ambiente

Editar variáveis de ambiente

Variáveis de ambiente

Você pode definir as variáveis de ambiente como pares de chave-valor acessíveis a partir do código de função. Elas são úteis para armazenar parâmetros de configuração sem precisar alterar o código de função. [Saiba mais](#)

Chave	Valor	
DB_NAME	sofiaDB	Remover
PASSWORD	[REDACTED]	Remover
RDS_PROXY_HOST	sofia-db.cr0s844gyvvj.us-east-1.rds.am	Remover
USER_NAME	admin	Remover

[Adicionar variáveis de ambiente](#)

► Configuração da criptografia

Cancelar Salvar

Variáveis de ambiente (4) [Editar](#)

As variáveis de ambiente a seguir são criptografadas em repouso com a chave de serviço padrão do Lambda.

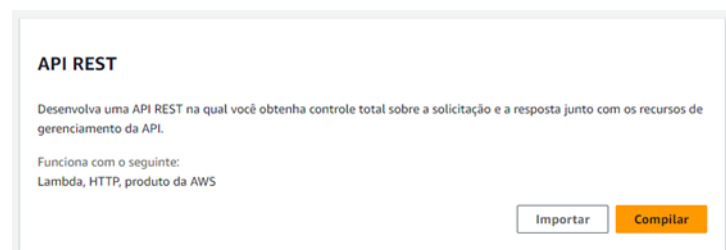
Q Encontrar variáveis de ambiente < 1 >

Chave	Valor
DB_NAME	sofiaDB
PASSWORD	[REDACTED]
RDS_PROXY_HOST	proxy-1717855241638-sofia-db.proxy-cr0s844gyvvj.us-east-1.rds.amazonaws.com
USER_NAME	admin

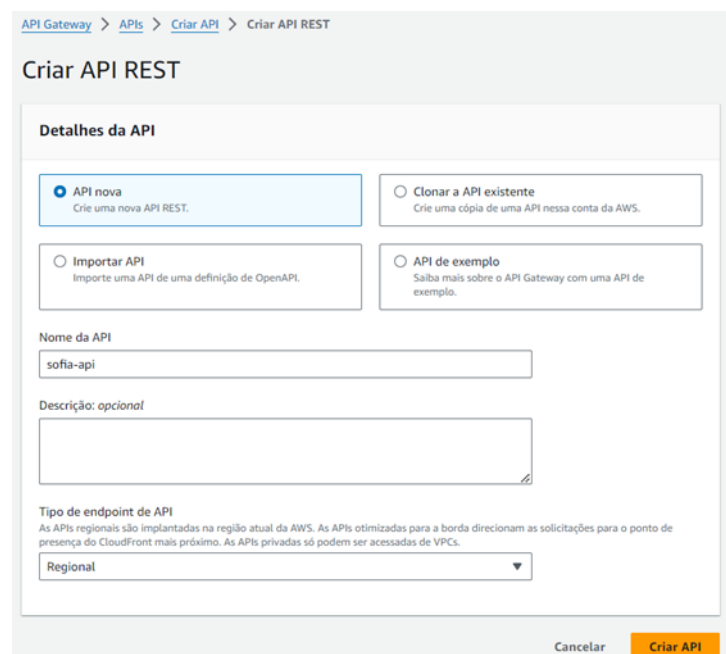
5 CRIAÇÃO DA API GATEWAY PARA IMPLEMENTAÇÃO DA FUNÇÃO LAMBDA

5.1 Criação da API e do método POST para envio dos dados para cadastro

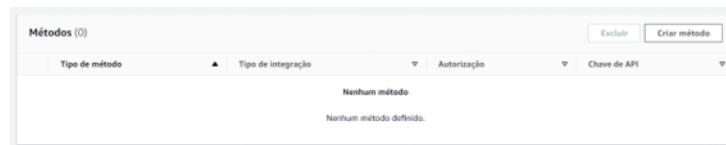
1. Na página API Gateway, encontre **API REST** e selecione **Compilar**.



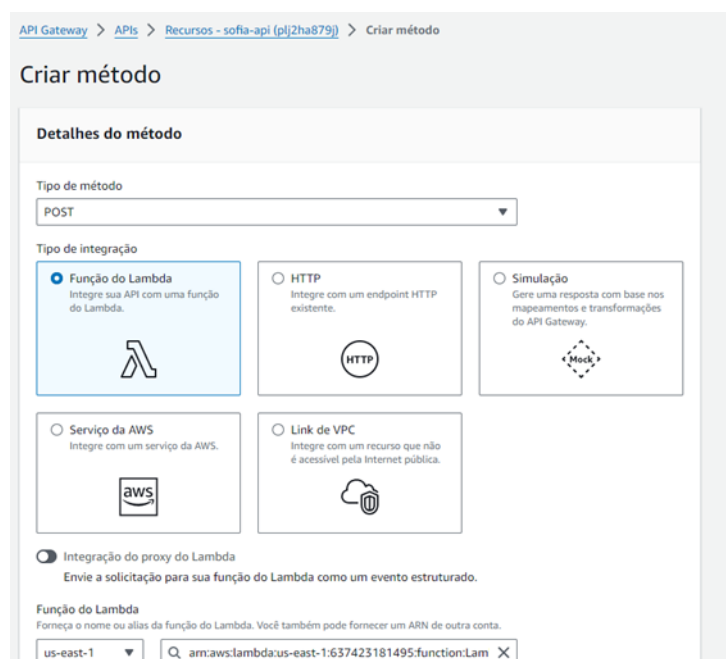
2. Marque a opção **API nova**, insira um nome para a api e clique em **Salvar**.



3. Na página de visualização da API criada, encontre **Métodos** e clique em **Criar método**.



4. Para o **Tipo de método** selecione **POST**. Em **Tipo de integração** marque a opção **Função do Lambda** e no campo **Função do Lambda**, selecione o **ARN** da função lambda criada.



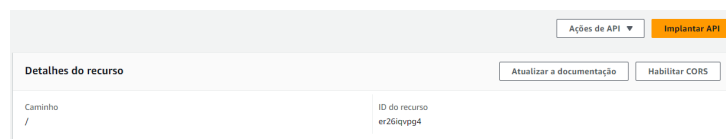
5. Clique em **Criar método**.

5.2 Configuração do CORS

CORS (Cross-Origin Resource Sharing) é um mecanismo que permite a comunicação segura entre diferentes domínios na web, superando as restrições da política de mesma origem dos navegadores. Habilitar o CORS permite que um aplicativo em um domínio acesse recursos de um servidor em outro domínio.

Para habilitar o CORS para que a aplicação hospedada no **EC2** possa consumir a API, siga os seguintes passos:

1. Na página de **Recursos** da API, encontre a área **Detalhes do recurso** e clique em **Habilitar CORS**



2. Em **Access-Control-Allow-Methods** selecione o método **POST** e clique em **Salvar**



3. Na página de **Recursos** da API, clique no método **POST**
4. Navegue para a aba **Resposta do método** e clique em **Editar**



- a. Escolha **Adicionar cabeçalho** e, em seguida, insira **Access-Control-Allow-Headers**.
- b. Selecione novamente **Adicionar cabeçalho** e, em seguida, insira **Access-Control-Allow-Methods**.
- c. Selecione novamente **Adicionar cabeçalho** e, em seguida, insira **Access-Control-Allow-Origin**. Clique em **Salvar**

API Gateway > APIs > Recursos - sofa-api (z18riuuvbk) > Editar resposta do método

Editar resposta do método [Informações](#)

Detalhes da resposta

Código de status HTTP

200

Nome de cabeçalho

Nome

Access-Control-Allow-Headers

Remover

Nome

Access-Control-Allow-Methods

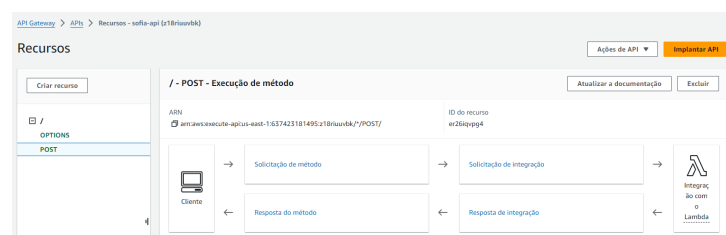
Remover

Nome

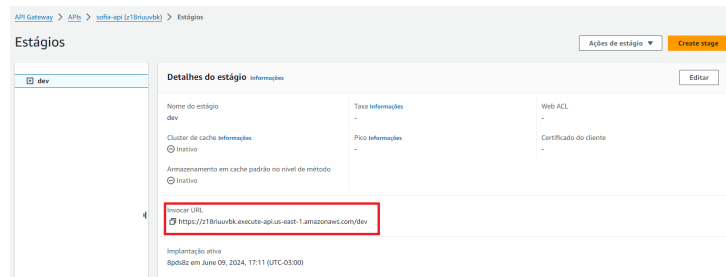
Access-Control-Allow-Origin

Remover

5. Novamente na tela de **Recursos** da API, clique em **Implantar API**.



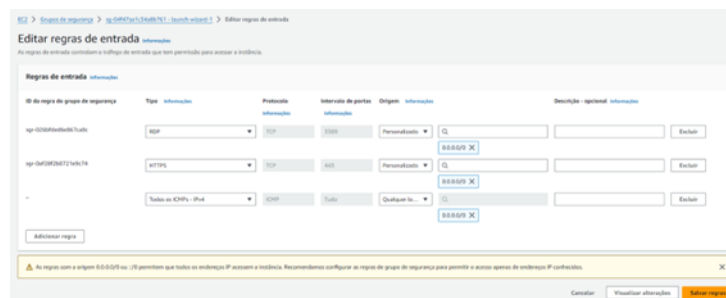
6. De um nome para o estágio e clique em **Implantar**. Isso irá efetuar o **Deploy** da API.
7. Para consumir a API, copie a **URL** criada.



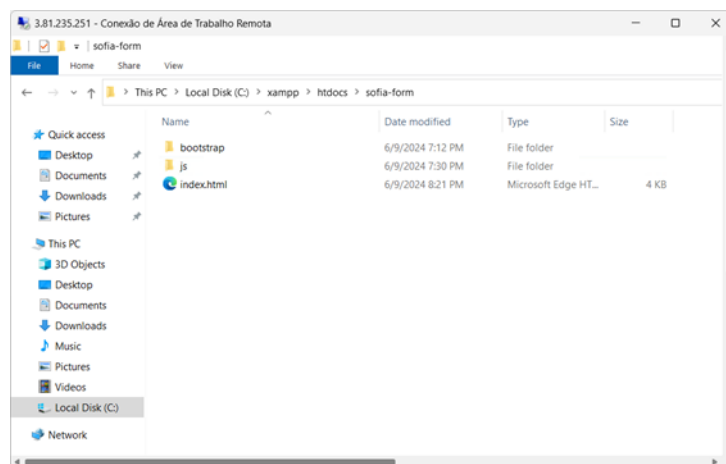
6 CRIAÇÃO DA INSTÂNCIA NO EC2

Para hospedar a aplicação web com o formulário que irá enviar os dados para cadastro no banco criado no RDS (**sofia-db**), iremos criar uma instância do EC2.

1. Na criação da instância, edite as **regras de entrada** no **Grupo de segurança** para permitir **Todos ICMPs - IPV4** com **Qualquer origem IPV4**. Adicione o **HTTPS**.



2. Acesse a instância pela **Conexão de Área de Trabalho Remota** utilizando o endereço ip público.
3. Desative o Firewall.
4. Instale o **Xampp** e ative o **Apache**.
5. No diretório **C:xampp\htdocs**, criado após instalação do **Xampp**, apague os arquivos que vieram com a instalação e crie a pasta do projeto (**sofia-form**).



6. Para estilização do formulário, foi utilizado o **Bootstrap**.

7. Formulário HTML para preenchimento e cadastro na base de dados **sofia-db**.

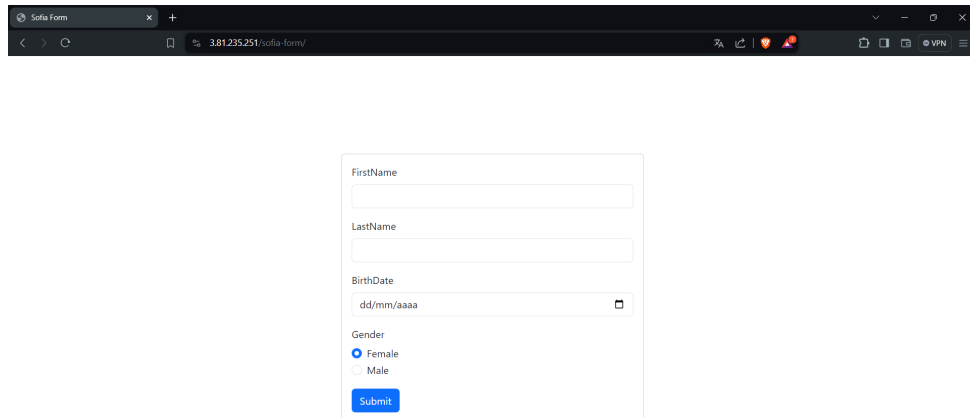
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
7   <title>Sofia Form</title>
8   <style>
9     .card-container {
10       position: absolute;
11       top: 50%;
12       left: 50%;
13       transform: translate(-50%, -50%);
14       width: 30rem;
15     }
16   </style>
17 </head>
18 <body>
19   <div class="card-container">
20     <div class="card">
21       <div class="card-body">
22         <form id="patientForm">
23           <div class="mb-3">
24             <label for="FirstName" class="form-label">FirstName</label>
25             <input type="text" class="form-control" id="FirstName">
26           </div>
27           <div class="mb-3">
28             <label for="LastName" class="form-label">LastName</label>
29             <input type="text" class="form-control" id="LastName">
30           </div>
31           <div class="mb-3">
32             <label for="BirthDate" class="form-label">BirthDate</label>
33             <input type="date" class="form-control" id="BirthDate">
34           </div>
35           <fieldset class="mb-3">
36             <legend class="col-form-label pt-0">Gender</legend>
37             <div class="form-check">
38               <input class="form-check-input" type="radio" name="Gender" id="female" value="Female" checked>
39               <label class="form-check-label" for="female">Female</label>
40             </div>
41             <div class="form-check">
42               <input class="form-check-input" type="radio" name="Gender" id="male" value="Male">
43               <label class="form-check-label" for="male">Male</label>
44             </div>
45           </fieldset>
46           <button type="submit" class="btn btn-primary">Submit</button>
47         </form>
48       </div>
49     </div>
50   </div>
51   <!-- Modal -->
52   <div class="modal fade" id="responseModal" tabindex="-1" aria-labelledby="responseModallabel" aria-hidden="true">
53     <div class="modal-dialog">
54       <div class="modal-content">
55         <div class="modal-header">
56           <h5 class="modal-title" id="responseModallabel">Resposta da API</h5>
57           <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
58         </div>
59         <div class="modal-body">
60           <!-- A mensagem do corpo será inserida aqui -->
61         </div>
62         <div class="modal-footer">
63           <button type="button" class="btn btn-primary" data-bs-dismiss="modal">OK</button>
64         </div>
65       </div>
66     </div>
67   </div>
68   <script src="bootstrap/js/bootstrap.bundle.min.js"></script>
69   <script src="js/main.js"></script>
70 </body>
71 </html>
72
73
```

8. Para consumo da API, implementado o código no arquivo **main.js**, Na primeira linha do arquivo, a constante **URL** recebe o endereço da API Gateway.

```
1  const URL = 'https://z18riuuvbk.execute-api.us-east-1.amazonaws.com/dev'
2
3  document.getElementById('patientForm').addEventListener('submit', function(e) {
4    e.preventDefault();
5
6    var data = {
7      FirstName: document.getElementById('FirstName').value,
8      LastName: document.getElementById('LastName').value,
9      BirthDate: document.getElementById('BirthDate').value,
10     Gender: document.querySelector('input[name="Gender"]:checked').value
11   };
12
13   fetch(URL, {
14     method: 'POST',
15     headers: {
16       'Content-Type': 'application/json'
17     },
18     body: JSON.stringify(data)
19   })
20   .then(response => response.json())
21   .then(data => {
22     if(data.statusCode === 200) {
23       // Define o texto do corpo do modal
24       document.querySelector('#responseModal .modal-body').textContent = data.body;
25       // Exibe o modal
26       var responseModal = new bootstrap.Modal(document.getElementById('responseModal'));
27       responseModal.show();
28     }
29   })
30   .catch((error) => {
31     console.error('Erro:', error);
32   });
33
34   // Limpa o formulário quando o modal é fechado
35   document.getElementById('responseModal').addEventListener('hidden.bs.modal', function () {
36     document.getElementById('patientForm').reset();
37   });
38 });
39
```

7 EXECUÇÃO DA APLICAÇÃO WEB

Com a aplicação hospedada no **EC2**, é possível acessar pelo navegador através do endereço **"https://3.81.235.251"** (Ip público gerado pela instância).



FirstName

LastName

BirthDate

dd/mm/aaaa

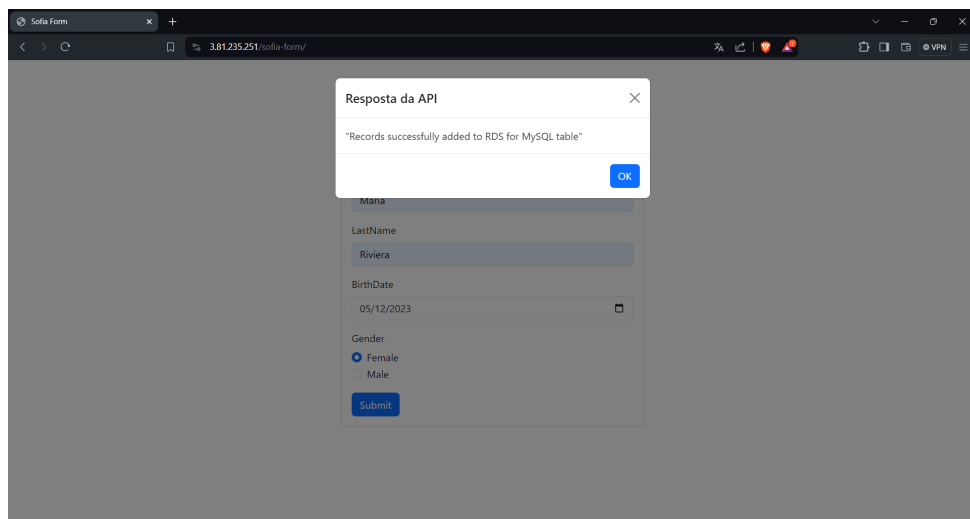
Gender

☒ Female

☐ Male

Submit

Ao preencher e submeter os dados do formulário, a API retorna a mensagem de resposta da API



Resposta da API

Records successfully added to RDS for MySQL table

OK

Maria

LastName

Riviera

BirthDate

05/12/2023

Gender

☒ Female

☐ Male

Submit