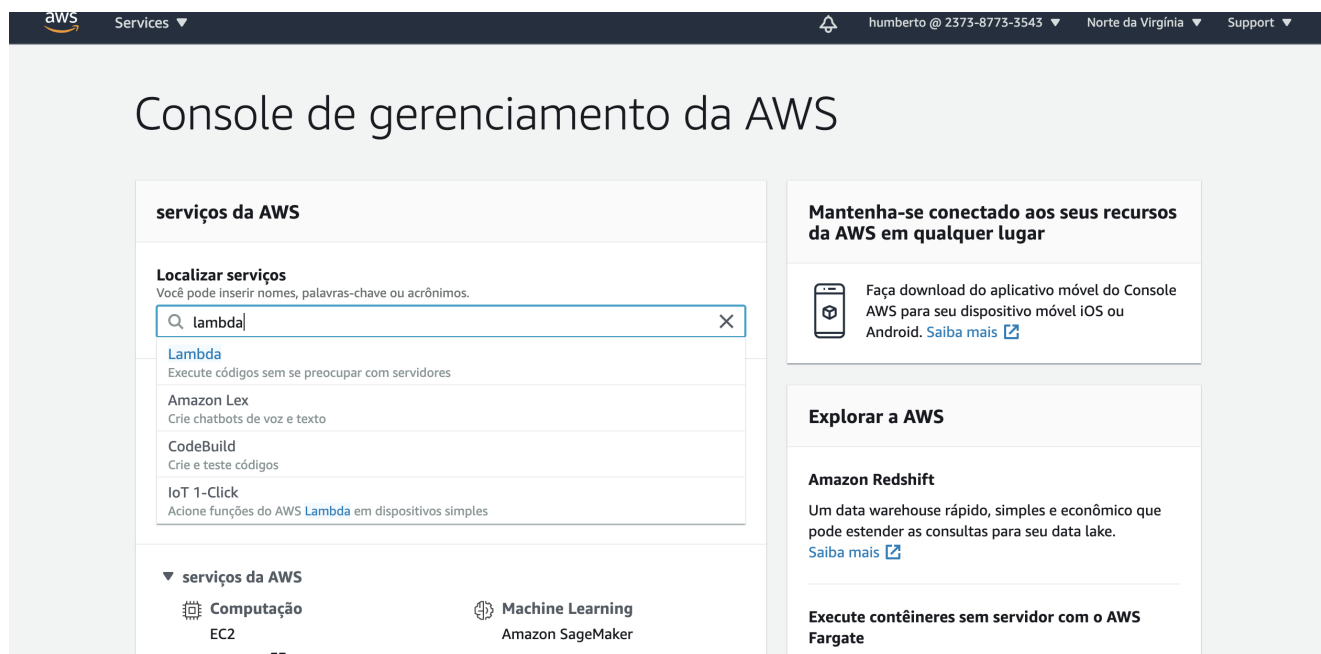
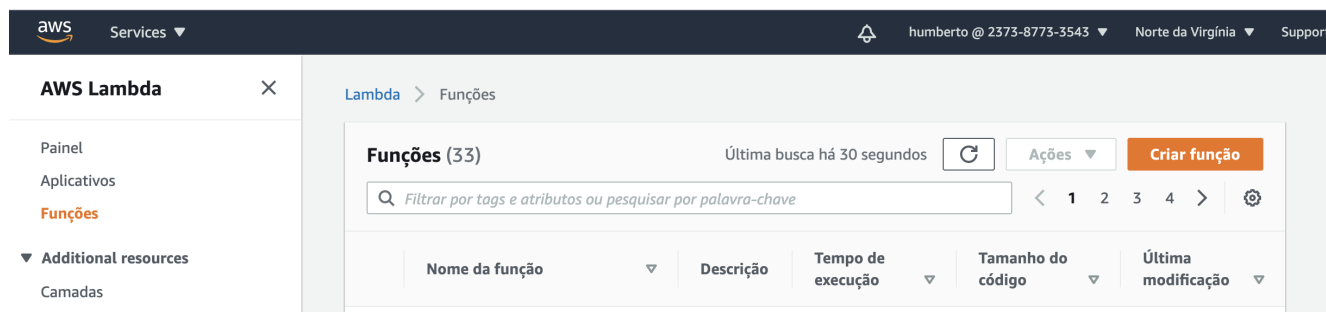


## Faça como eu fiz: Criando uma função lambda

Vamos criar a nossa primeira função lambda na AWS. Acesse o console da AWS utilizando a sua conta, e na página inicial no campo localizador de serviços, digite “lambda” e clique na primeira opção, “Lambda”.



Na próxima tela, clique no botão “Cria função”.



Na tela seguinte, deixe selecionado a opção “Criar do zero”.

No campo “Nome da função”, preencha com `curso-alura-1-3`.

Deixe as demais opções como estão e clique no botão “Criar função”.

Lambda > Funções > Criar função

## Criar função Info

Escolha uma das opções a seguir para criar a função.

**Criar do zero**

Comece com um simples exemplo de Hello World.

**Usar um esquema**

Crie um aplicativo do Lambda a partir do código de exemplo e de predefinições de configuração para casos de uso comum.

**Explorar o repositório de aplicativos sem servidor**

Implante um aplicativo do Lambda de exemplo a partir do AWS Serverless Application Repository.

### Informações básicas

**Nome da função**  
Insira um nome que descreva o propósito da função.

Use somente letras, números, hífens ou sublinhados sem espaços.

**Tempo de execução** Info  
Escolha a linguagem a ser usada para escrever a função.

Após alguns segundos, seremos direcionados para a tela onde teremos à nossa disposição diversos controles: Editor, botão para testar e opções de configuração; nós iremos gastar a maior parte do tempo no editor.

Ao criar a nossa função, ela é gerada com um código inicial, que vamos alterar para o código abaixo:

```
const pacientes = [
  {id: 1, nome: "Maria", idade: 20},
  {id: 2, nome: "Joao", idade: 30},
  {id: 3, nome: "Jose", idade: 45}
];

function buscarPaciente(id) {
  return pacientes.find(paciente => paciente.id == id);
}

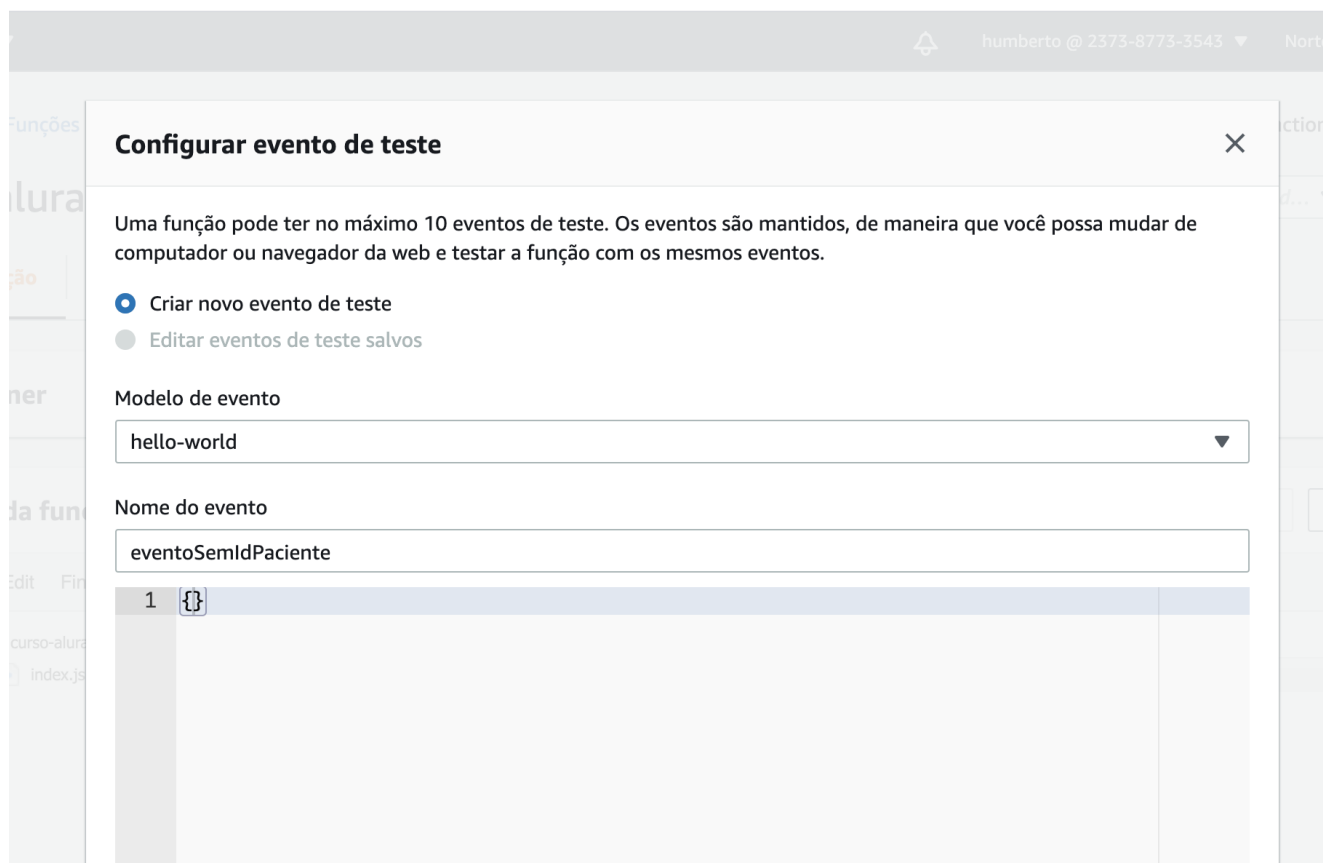
exports.handler = async (event) => {
  console.log("Paciente informado: " + event.pacienteId);
}
```

```
const pacienteEncontrado = event.pacienteId ? buscarPaciente(  
  
const response = {  
  statusCode: 200,  
  body: JSON.stringify(pacienteEncontrado),  
};  
return response;  
};
```

[COPIAR CÓDIGO](#)

Este código irá retornar todos os dados disponíveis na variável `pacientes`, caso nenhum parâmetro seja enviado, ou irá retornar um paciente caso um ID de paciente seja enviado como parâmetro.

Vamos fazer uma primeira execução deste código sem informar parâmetro algum. Clique no botão “Testar”, no topo da página, na modal que será exibida, deixe o campo “Modelo de evento” como `hello-world`, o campo “Nome do evento”, preencha como `eventoSemIdPaciente`, e os parâmetros do evento, preencha com um JSON vazio, porque neste cenário não desejamos passar parâmetros. Por fim, clique no botão “Criar”.



**Configurar evento de teste**

Uma função pode ter no máximo 10 eventos de teste. Os eventos são mantidos, de maneira que você possa mudar de computador ou navegador da web e testar a função com os mesmos eventos.

☒ Criar novo evento de teste  
☐ Editar eventos de teste salvos

Modelo de evento  
hello-world ▼

Nome do evento  
eventoSemIdPaciente

1	{ }

Ao clicar em “Testar”, teremos que criar um evento. Como neste primeiro teste não passaremos parâmetros, preencheremos um objeto JSON vazio e clicaremos em “Criar”.

Temos como resposta desta primeira execução todos os pacientes.

Repita o passo anterior para criar um novo evento. Nomeie este evento como `eventoComIdPaciente` e, como dados do evento, preencha o objeto JSON abaixo:

```
{  
  "pacienteId": "1"  
}
```

[COPIAR CÓDIGO](#)

Uma função pode ter no máximo 10 eventos de teste. Os eventos são mantidos, de maneira que você possa mudar de computador ou navegador da web e testar a função com os mesmos eventos.

- ☒ Criar novo evento de teste
- ☐ Editar eventos de teste salvos

Modelo de evento

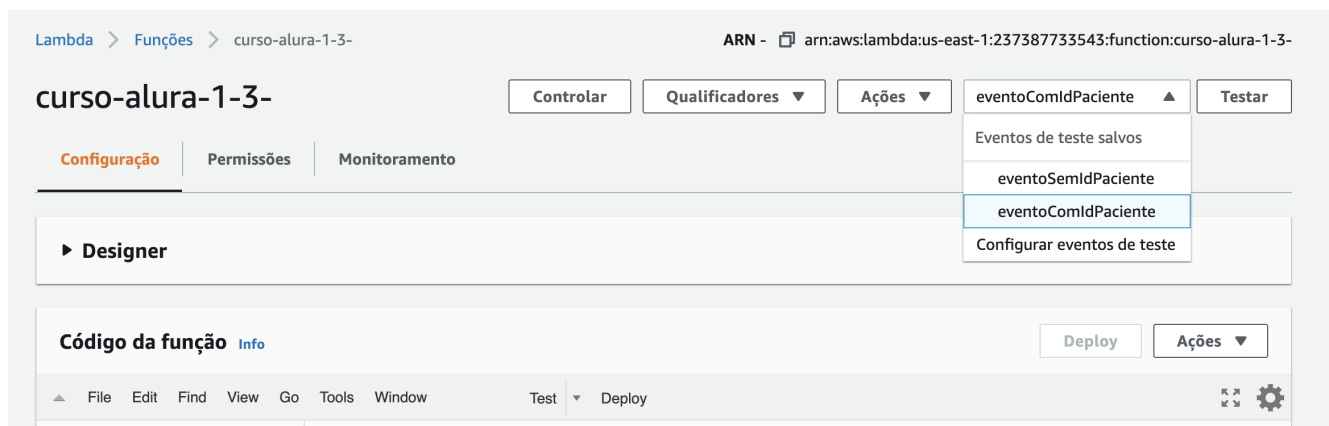
eventoSemIdPaciente

Nome do evento

eventoComIdPaciente

```
1 {  
2   "pacienteId": "1"  
3 }  
4
```

Clique em “Criar”, em seguida selecione o evento que acabamos de criar e clique em “Testar”.



Temos agora como retorno apenas o paciente informado como parâmetro no evento.