

Semana do Desenvolvedor – Dia 4

Aula 4: Fluxos Adicionais de Pedidos, DLQs e Lambda Layers

Documentação do Laboratório: Arquitetura de Processamento de Pedidos e Arquivos na AWS

(Duração Estimada: 160 minutos)

1. Visão Geral e Objetivos da Aula

Bem-vindos à nossa última aula! Nas sessões anteriores, construímos os fluxos de ingestão de pedidos via API e S3, a validação desses pedidos, a publicação de eventos no EventBridge, e o processamento central com persistência no DynamoDB.

Nesta aula final, vamos expandir a funcionalidade do nosso sistema para lidar com outras operações importantes no ciclo de vida de um pedido: cancelamento e alteração. Utilizaremos novamente o EventBridge para rotear esses novos tipos de eventos para filas SQS dedicadas, que acionarão Lambdas específicas para atualizar o estado do pedido na nossa tabela DynamoDB principal. Além disso, revisaremos a importância das Dead Letter Queues (DLQs) que configuramos ao longo do caminho e faremos um teste prático para ver uma DLQ em ação.

Ao final desta aula, você terá:

- Fluxos funcionais para cancelamento e alteração de pedidos, integrados à arquitetura existente.
- Novas regras no EventBridge, filas SQS e funções Lambda para essas operações.
- Uma compreensão prática do funcionamento e da importância das DLQs.
- Uma visão completa da arquitetura integrada e funcional.

Recursos a Serem Criados/Configurados Nesta Aula:

- IAM Role: `lambda-altera-cancela-role-seu-nome`
- Amazon SQS DLQ (Standard): `cancela-pedido-dlq-seu-nome`
- Amazon SQS Queue (Standard): `cancela-pedido-queue-seu-nome`
- AWS Lambda Function: `cancela-pedido-lambda-seu-nome`
- Amazon SQS DLQ (Standard): `altera-pedido-dlq-seu-nome`

- Amazon SQS Queue (Standard): altera-pedido-queue-seu-nome
- AWS Lambda Function: altera-pedido-lambda-seu-nome
- Amazon EventBridge Rules (2 novas): cancela-pedido-rule-seu-nome, altera-pedido-rule-seu-nome
- Revisão e Teste de DLQs existentes.

Lembrete Importante: Substitua seu-nome em todos os nomes de recursos. Trabalhe consistentemente na mesma região AWS.

2. Configuração de Permissões (IAM Role)

Criaremos uma nova role para as Lambdas que processarão os cancelamentos e alterações.

2.1. Criar Role para Lambdas de Alteração/Cancelamento (lambda-altera-cancela-role-seu-nome)

1. Acesse o serviço **IAM**.
2. Clique em "**Roles**" > "**Create role**".
3. **Trusted entity type:** AWS service.
4. **Use case:** Lambda. Clique em "**Next**".
5. **Permissions policies:** Adicione AWSLambdaBasicExecutionRole. Clique em "**Next**".
6. **Role name:** lambda-altera-cancela-role-seu-nome.
7. Clique em "**Create role**".
 - *Nota: Adicionaremos as permissões específicas após criar as SQS e sabermos o ARN da DynamoDB principal.*

3. Implementação do Fluxo de Cancelamento de Pedido

3.1. Criar Filas SQS para Cancelamento (cancela-pedido-dlq-seu-nome e cancela-pedido-queue-seu-nome)

1. Acesse o **SQS**.
2. Crie a DLQ:
 - Type: Standard.
 - Name: cancela-pedido-dlq-seu-nome.

- Clique em "Create queue".
 - Anote o ARN.
3. Crie a Fila Principal:
- No menu lateral, clique em **"Queues"**.
 - **Create queue.**
 - **Type:** Standard.
 - **Name:** cancela-pedido-queue-seu-nome.
 - **Visibility timeout:** Ajuste para 70 segundos (para ser maior que o timeout da Lambda que criaremos).
 - **Dead-letter queue (DLQ):** Enabled, selecione o ARN da cancela-pedido-dlq-seu-nome.
 - **Max receives:** 3.
 - Clique em **"Create queue"**.
 - Anote o ARN e a URL.

3.1.1. Conceder Permissão SQS para Cancelamento à Role lambda-altera-cancela-role-seu-nome

Antes de criarmos a Lambda de cancelamento e adicionarmos seu trigger SQS, vamos garantir que sua IAM Role (lambda-altera-cancela-role-seu-nome) tenha as permissões para interagir com a fila cancela-pedido-queue-seu-nome.

1. Acesse o serviço **IAM** no Console AWS.
2. No menu à esquerda, clique em **"Roles"** e encontre a role **lambda-altera-cancela-role-seu-nome**. Clique nela.
3. Na aba **"Permissions"**, sob "Permissions policies", clique em **"Add permissions"** e selecione **"Create inline policy"**.
4. Selecione a aba **JSON**.
5. Remova a política existente e cole a política abaixo, modificando as seguintes informações:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ReceiveMessage",
        "sqs:DeleteMessage",
        "sqs:GetQueueAttributes"
      ],
      "Resource": [
        "COLE AQUI O ARN DA SUA FILA cancela-pedido-queue-seu-nome"
      ]
    }
  ]
}
```

Caminho: IAM > Roles > lambda-altera-cancela-role-seu-nome

6. **Next.**
7. Em "**Review policy**":
 - **Name:** SQSReadCancelaTrigger-seu-nome.
8. Clique em "**Create policy**".

3.2. Criar Lambda para Cancelamento (cancela-pedido-lambda-seu-nome)

1. Acesse **AWS Lambda** > "**Create function**" > "Author from scratch".
2. **Function name:** cancela-pedido-lambda-seu-nome.
3. **Runtime:** Python 3.12.
4. **Em Permissions**, expanda o menu **Change default execution role**.
5. Selecione, **Use an existing role** :

- Use a role existente lambda-altera-cancela-role-seu-nome.
- 6. Clique em **"Create function"**.
- 7. Na página da função, rola para baixo, até a aba **"Code"**, substitua lambda_function.py pelo código Python que você encontra, [clikando aqui](#).
- 8. Clique em **"Deploy"**.
- 9. Configure Variáveis de Ambiente:
 - Vá para "Configuration" > "Environment variables" > "Edit" > "Add environment variable".
 - Adicione: Key: DYNAMODB_TABLE_NAME.
 - Value: pedidos-db-seu-nome.
 - Clique em "Save".
- 10. Configure Timeout:
 - Vá para "Configuration" > "General configuration" > "Edit".
 - Defina **"Timeout"** para 60 segundos.
 - Clique em "Save".
- 11. Adicione Trigger SQS:
 - Vá para "Configuration" > "Triggers" > "Add trigger".
 - Em **"Trigger configuration"**, selecione a fonte **SQS**.
 - **SQS queue**: cancela-pedido-queue-seu-nome.
 - Batch size: 1.
 - Clique em "Add".

3.3. Criar Regra no EventBridge para Cancelamento (cancela-pedido-rule-seu-nome)

1. Acesse **Amazon EventBridge** > **"Rules"**.
2. Selecione o Event Bus: pedidos-event-bus-seu-nome.
3. Clique em **"Create rule"**.
4. **Name**: cancela-pedido-rule-seu-nome.
5. **Rule type**: "Rule with an event pattern".
6. Clique em **"Next"**.

7. Role para baixo.
8. Em **Event pattern**: Selecione "Custom patterns (JSON editor)" e cole:

```
{  
  "source": ["lab.aula4.operacoes"],  
  "detail-type": ["CancelarPedido"]  
}
```

(**Nota:** Usaremos *lab.aula4.operacoes* como a origem lógica para esses novos eventos).

9. Clique em "**Next**".
10. **Target types**: AWS service
11. **Select a target**: SQS queue.
12. **Queue**: cancela-pedido-queue-seu-nome.
13. Clique em "**Next**".
14. **Next**.
15. Clique em "**Create rule**".

4. Implementação do Fluxo de Alteração de Pedido

Este fluxo é muito similar ao de cancelamento.

4.1. Criar Filas SQS para Alteração (altera-pedido-dlq-seu-nome e altera-pedido-queue-seu-nome)

Crie a DLQ:

- Acesse **SQS > Create queue**.
- **Type**: Standard.
- **Name**: altera-pedido-dlq-seu-nome.
- **Create queue**.
- Anote o ARN.

Crie a Fila Principal:

- No menu lateral, clique em "**Queues**".
- **Create queue**.

- **Type:** Standard.
- **Name:** altera-pedido-queue-seu-nome.
- **Visibility timeout:** 70 segundos.
- **Dead-letter queue:** Enabled.
- **Choose queue :** Queue ARN da altera-pedido-dlq-seu-nome.
- **Max receives:** 3.
- Clique em "Create queue".
- Anote o ARN e a URL.

4.2.1. Conceder Permissão SQS

Antes de criarmos a Lambda de alteração e adicionarmos seu trigger SQS, vamos garantir que sua IAM Role (lambda-altera-cancela-role-seu-nome) tenha as permissões para interagir com a fila altera-pedido-queue-seu-nome.

1. Acesse o serviço **IAM** no Console AWS.
2. No menu à esquerda, clique em **"Roles"** e encontre a role **lambda-altera-cancela-role-seu-nome**. Clique nela.
3. Na aba **"Permissions"**, sob "Permissions policies", clique em **"Add permissions"** e selecione **"Create inline policy"**.
4. Selecione a aba **JSON**.
5. Remova a política existente e cole a política abaixo, modificando as seguintes informações:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ReceiveMessage",
        "sqs:DeleteMessage",
        "sqs:GetQueueAttributes"
      ],
      "Resource": [
        "COLE AQUI O ARN DA SUA FILA altera-pedido-queue-seu-nome"
      ]
    }
  ]
}
```

Caminho: IAM > Roles > lambda-altera-cancela-role-seu-nome

1. **Next.**
2. Em **"Review policy"**:
 - **Name:** SQSReadAlterarTrigger-seu-nome.
3. Clique em **"Create policy"**.

4.2. Criar Lambda para Alteração (altera-pedido-lambda-seu-nome)

1. Acesse **AWS Lambda** > **"Create function"** > "Author from scratch".
2. **Function name:** altera-pedido-lambda-seu-nome.
3. **Runtime:** Python 3.12.
4. **Em Permissions**, expanda o menu **Change default execution role**.
5. Selecione, **Use an existing role** :
 - Use a role lambda-altera-cancela-role-seu-nome.
6. Clique em **"Create function"**.
7. Na página da função, rola para baixo, até a aba **"Code"**, substitua lambda_function.py pelo código Python que você encontra, [clcando aqui](#).
8. Clique em **"Deploy"**.
1. Configure Variáveis de Ambiente. Vá para a aba **"Configuration"** > **"Environment variables"** > **"Edit"** > **"Add environment variable"**.
 - Adicione: Key: DYNAMODB_TABLE_NAME.
 - Value: pedidos-db-seu-nome.
 - Save.
9. Ainda na aba **"Configuration"**, selecione **"General configuration"** > **"Edit"**.
10. Configure **Timeout**:
 - Defina **Timeout** para 60 segundos.
 - Save.
2. Adicione Trigger SQS. Vá para **"Configuration"** > **"Triggers"** > **"Add trigger"**.
 - **Trigger configuration:** SQS.
 - **SQS queue:** altera-pedido-queue-seu-nome.
 - **Batch size:** 1.
 - **Add.**

4.3. Criar Regra no EventBridge para Alteração (altera-pedido-rule-seu-nome)

1. Acesse **Amazon EventBridge** > **"Rules"**.
2. Selecione o **Event Bus**: pedidos-event-bus-seu-nome.
3. Clique em **"Create rule"**.
4. **Name**: altera-pedido-rule-seu-nome.
5. **Rule type**: "Rule with an event pattern".
6. **Next**.
7. Desça até **Event pattern** e selecione **"Custom patterns (JSON editor)"**.
8. **Cole**:

```
{
  "source": ["lab.aula4.operacoes"],
  "detail-type": ["AlterarPedido"]
}
```

9. **Next**.
10. **Target types**: AWS service.
11. **Select a target**: SQS queue.
12. **Queue**: altera-pedido-queue-seu-nome.
13. **Next**.
14. Pule tags (Next).
15. **"Create rule"**.

5. Atualizar Permissões da Role de Alteração/Cancelamento

Agora concederemos as permissões SQS e DynamoDB necessárias à role lambda-altera-cancela-role-seu-nome.

- Vá para **IAM** > **Roles** > lambda-altera-cancela-role-seu-nome.
- Na aba **"Permissions"** > **"Add permissions"** > **"Create inline policy"**.
- Selecione a aba **JSON**.

- Remova a política existente e cole a política abaixo, modificando as seguintes informações:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:UpdateItem",
        "dynamodb:GetItem"
      ],
      "Resource": "COLE AQUI O ARN DA SUA TABELA pedidos-db-seu-nome"
    }
  ]
}
```

- **Next.**
- Em "**Review policy**":
 - **Policy name:** ReadCancelAlterDynamoUpdate-seu-nome.
- Clique em "**Create policy**".

6. Teste dos Novos Fluxos de Cancelamento e Alteração

Vamos simular a chegada de eventos de cancelamento e alteração no nosso Event Bus. A forma mais fácil é usar o próprio console do EventBridge.

6.1. Preparar um Pedido Existente:

- Certifique-se de que você tem um pedido na tabela pedidos-db-seu-nome que foi processado anteriormente (ex: apiP001-seu-nome ou s3P003-seu-nome).
- **Caminho:** DynamoDB > Explore items > pedidos-db-seu-nome > pedidos-db-seu-nome
- Anote o `pedidoId` de um deles.

- **Exemplo:** apiP001-seu-nome

6.2. Testar o Fluxo de Cancelamento:

1. Vá para **Amazon EventBridge** > **"Event buses"**.
2. Selecione seu bus **pedidos-event-bus-seu-nome**.
3. Clique em **"Send events"**.

pedidos-event-bus-seu-nome

[Edit](#)[Delete](#)[Send events](#)[Start discovery](#)[CloudFormation Template ▾](#)

Event bus details

Event bus name pedidos-event-bus-seu-nome	Schema discovery ⊖ Not Initiated	Created on May 7, 2025, 03:53
Description	Discoverer ID	Last modified

4. **Event source:** lab.aula4.operacoes

5. **Detail type:** CancelarPedido

6. **Event detail (JSON):** (Substitua <ID_DO_SEU_PEDIDO> pelo ID anotado no passo 6.1)

```
{  
  "pedidoId": "<ID_DO_SEU_PEDIDO>"  
}
```

Exemplo:

```
{  
  "pedidoId": "apiP001-seu-nome"  
}
```

7. Clique em **"Send"**.

Verificações:

8. **Fila SQS cancela-pedido-queue-seu-nome:** A mensagem deve passar por ela rapidamente.
9. **Logs da cancela-pedido-lambda-seu-nome:** Verifique no CloudWatch se a Lambda foi invocada, processou o pedidoId e logou a atualização no DynamoDB.

Processando cancelamento para pedido: apiP001-seu-nome

Pedido apiP001-seu-nome atualizado para CANCELADO. Resposta **DynamoDB**: {'Attributes': {'timestampAtualizacao': '2025-05-13T19:46:17.248662Z', 'statusPedido': 'CANCELADO'}, 'ResponseMetadata': {'RequestId': 'SBGVGCBF6295KFRMFV3F6RG50FVV4KQNS05AEMVJF66Q9ASUAAJG', 'HTTPStatusCode': 200, 'HTTPHeaders': {'server': 'Server', 'date': 'Tue, 13 May 2025 19:46:17 GMT', 'content-type': 'application/x-amz-json-1.0', 'content-length': '108', 'connection': 'keep-alive', 'x-amzn-requestid': 'SBGVGCBF6295KFRMFV3F6RG50FVV4KQNS05AEMVJF66Q9ASUAAJG', 'x-amz-crc32': '3088614704'}, 'RetryAttempts': 0}}

10. **DynamoDB pedidos-db-seu-nome:** Explore os itens da tabela. O status do pedido que você cancelou deve agora ser "CANCELADO" e o timestampAtualizacao deve ter sido atualizado.

Antes

<input type="checkbox"/>	pedidoId (String)	statusPedido	timestampCriacaoEvento
<input type="checkbox"/>	apiP001-seu-nome	PEDIDO_PROCESSADO	2025-05-13T00:03:01
<input type="checkbox"/>	s3P003-seu-nome	PEDIDO_PROCESSADO	2025-05-13T00:07:56

Depois

<input type="checkbox"/>	pedidoId (String)	statusPedido	timestampAtualizacao
<input type="checkbox"/>	apiP001-seu-nome	CANCELADO	2025-05-13T19:46:17
<input type="checkbox"/>	s3P003-seu-nome	PEDIDO_PROCESSADO	

6.3. Testar o Fluxo de Alteração:

1. Use o mesmo pedidoId (ou crie/processe um novo pedido se preferir).
2. Vá para **Amazon EventBridge** > **"Event buses"** > pedidos-event-bus-seu-nome > **"Send events"**.
3. **Event source:** lab.aula4.operacoes
4. **Detail type:** AlterarPedido
5. **Event detail (JSON):** (Substitua <ID_DO_SEU_PEDIDO>)

```
{
  "pedidoId": "<ID_DO_SEU_PEDIDO>",
  "novosItens": [
    {"sku": "PROD-ALTERADO-1", "qtd": 1},
    {"sku": "PROD-ALTERADO-2", "qtd": 99}
  ]
}
```

Exemplo:

```
{
  "pedidoId": "apiP001-seu-nome",
  "novosItens": [
    {"sku": "PROD-ALTERADO-1", "qtd": 1},
    {"sku": "PROD-ALTERADO-2", "qtd": 99}
  ]
}
```

6. Clique em **"Send"**.

Verificações:

7. **Fila SQS altera-pedido-queue-seu-nome:** A mensagem deve passar rapidamente.
8. **Logs da altera-pedido-lambda-seu-nome:** Verifique no CloudWatch a invocação, o processamento e a atualização no DynamoDB.

```
Processando alteração para pedido: apiP001-seu-nome com novos itens: [{"sku": "PROD-ALTERADO-1", "qtd": 1}, {"sku": "PROD-ALTERADO-2", "qtd": 99}]
```

```
Pedido apiP001-seu-nome atualizado para ALTERADO. Resposta DynamoDB: {'Attributes': {'itens': [{'sku': 'PROD-ALTERADO-1', 'qtd': Decimal('1')}, {'sku': 'PROD-ALTERADO-2', 'qtd': Decimal('99')}]}, 'timestampAtualizacao': '2025-05-13T20:09:57.871676Z', 'statusPedido': 'ALTERADO'}, 'ResponseMetadata': {'RequestId': '7UE0BL101RQ6VQSODFMHU8H7MJVV4KQNS05AEMVJF66Q9ASUAAJG', 'HTTPStatusCode': 200, 'HTTPHeaders': {'server': 'Server', 'date': 'Tue, 13 May 2025 20:09:58 GMT', 'content-type': 'application/x-amz-json-1.0', 'content-length': '232', 'connection': 'keep-alive', 'x-amzn-requestid': '7UE0BL101RQ6VQSODFMHU8H7MJVV4KQNS05AEMVJF66Q9ASUAAJG', 'x-amz-crc32': '3651666611'}, 'RetryAttempts': 0}}
```

9. **DynamoDB pedidos-db-seu-nome:** O status do pedido deve ser "ALTERADO", o timestampAtualizacao atualizado, e o atributo itens deve refletir a nova lista de itens enviada no evento.

Antes

<input type="checkbox"/>	pedidoId (String)	statusPedido	timestampAtualizacao
<input type="checkbox"/>	apiP001-seu-nome	CANCELADO	2025-05-13T19:46:17
<input type="checkbox"/>	s3P003-seu-nome	PEDIDO_PROCESSADO	

Depois

<input type="checkbox"/>	pedidoId (String)	statusPedido	timestampAtualizacao
<input type="checkbox"/>	apiP001-seu-nome	ALTERADO	2025-05-13T20:09:57.8
<input type="checkbox"/>	s3P003-seu-nome	PEDIDO_PROCESSADO	

7. Revisão e Teste de Dead Letter Queues (DLQs)

Configuramos DLQs para todas as nossas filas SQS. Elas são essenciais para capturar mensagens que falharam repetidamente no processamento por uma Lambda. Vamos testar uma delas.

1. Escolher uma Lambda e Forçar um Erro:

- Vamos usar a processa-pedidos-lambda-seu-nome (da Aula 3) como exemplo.
- Vá para a função Lambda processa-pedidos-lambda-seu-nome.
- Na aba "**Code**", edite lambda_function.py.
- Logo no início da função lambda_handler, adicione a linha: `raise ValueError("Teste Forçado para DLQ!")`
- Clique em "**Deploy**".

2. Disparar o Fluxo:

3. Envie um novo pedido pela API Gateway (como no Passo 8.1 da Aula 3).

4. Prepare a Requisição

Você enviará uma requisição POST para a URL da sua API com um payload em formato JSON. Utilize o comando que você encontrará, [clikando aqui](#), e altere os **valores abaixo**:

- **<INVOKE_URL>: URL de Invocação (Invoke URL) da sua API** que você anotou anteriormente.
 - **Caminho:** API Gateway > APIs > pedidos-api-seu-nome > Stages
- **pedidoId e clienteId:** personalize com os dados:
 - **"pedidoId":** "apiP0043-TESTE-FLUXO-AULA-4"
 - **"clienteId":** "clienteAPI-TRE-seu-nome",
 - **"itens":** [{"item": "Produto X API", "qtd": 1}]

Exemplo com URL fictício de como deverá ficar:

```
curl -X POST https://rtc9oh0au6.execute-api.us-east-1.amazonaws.com/dev/pedidos \
-H "Content-Type: application/json" \
-d '{
  "pedidoId": "apiP0043-TESTE-FLUXO-ERRO-AULA-4",
  "clienteId": "clienteAPI-TRE-seu-nome",
  "itens": [
    {
      "Item": "Produto X API",
      "qtd": 1
    }
  ]
}'
```

5. Observar o Comportamento:

- **Logs da processa-pedidos-lambda-seu-nome:** Você verá a Lambda sendo invocada, logando o evento recebido e imediatamente falhando com o `ValueError("Teste Forçado para DLQ!")`.
- **Fila SQS pedidos-pendentes-queue-seu-nome:** A mensagem aparecerá, desaparecerá (enquanto a Lambda tenta processar e falha), e reaparecerá algumas vezes (conforme o Visibility Timeout e as tentativas).
- **Você não localiza a mensagem?** Vamos investigar por que a mensagem pode não estar visível na fila de origem.

Causas Possíveis e Pontos de Verificação:

- Lambda Falhando Rápido Demais e SQS Movendo para DLQ Muito Rapidamente:
- Se a Lambda processa-pedidos-lambda-seu-nome está falhando extremamente rápido (o que é o caso com um `raise ValueError` logo no início) e o `maxReceiveCount` na fila

pedidos-pendentes-queue-seu-nome está configurado para um número baixo (**como 1, ou o padrão de 3 que usamos**), a SQS pode mover a mensagem para a DLQ (**pedidos-pendentes-dlq-seu-nome**) quase que imediatamente após as tentativas falharem.

- Você pode não ter tempo de "vê-la" na fila principal usando "Poll for messages" no console, pois a transição para a DLQ é rápida.
- **Achamos a causa possível!** Vamos seguir em frente.
- **DLQ pedidos-pendentes-dlq-seu-nome:** Após o número máximo de tentativas (configurado como 3), a mensagem será movida para esta DLQ.
 - Vá para a fila DLQ no console SQS.
 - Observe a coluna "Messages available". Ela deverá aumentar para 1.
 - Use "Send and receive messages" > "Poll for messages" para visualizar a mensagem na DLQ.

6. Limpeza:

- **IMPORTANTE:** Volte para a Lambda processa-pedidos-lambda-seu-nome, remova a linha raise ValueError("Teste Forçado para DLQ!") do código e clique em "**Deploy**" novamente para restaurar o funcionamento normal.
- Você pode deixar a mensagem na DLQ ou deletá-la. Em um cenário real, você analisaria a mensagem na DLQ para entender o erro e decidir como reprocessá-la (talvez usando "Start message redrive").

9. Conclusão da Aula 4 e do Laboratório

Parabéns por concluir a Aula 4! Você expandiu significativamente a funcionalidade do nosso sistema de processamento de pedidos, adicionando fluxos de cancelamento e alteração, e validou o funcionamento crucial das Dead Letter Queues.

10. Conclusão Geral do Laboratório

A Jornada:

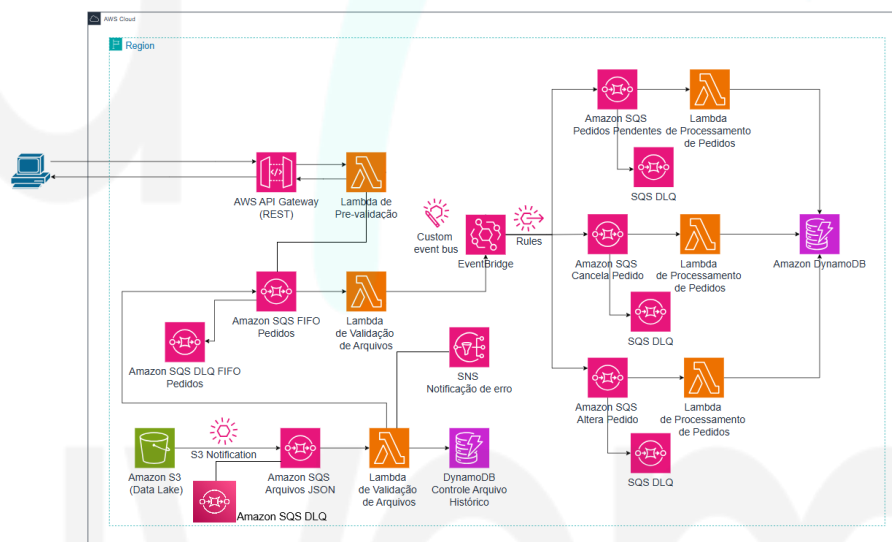
Ao longo destas quatro aulas, embarcamos juntos na construção de uma

arquitetura AWS completa e funcional. Começamos na Aula 1 estabelecendo a porta de entrada principal via API Gateway e o início do pipeline com SQS e EventBridge. Na Aula 2, adicionamos um fluxo robusto para ingestão de arquivos via S3, integrando-o inteligentemente ao pipeline principal. Na Aula 3, conectamos as pontas, implementando o processamento central que consome os eventos do EventBridge e persiste os pedidos no DynamoDB. Finalmente, na Aula 4, adicionamos complexidade e realismo com fluxos de cancelamento e alteração e solidificamos o tratamento de erros com DLQs.

A Arquitetura Construída:

Vocês agora têm diante de si uma arquitetura orientada a eventos, serverless e desacoplada, capaz de lidar com múltiplas fontes de dados e diferentes operações de negócio. Ela utiliza serviços essenciais da AWS:

- **Compute:** AWS Lambda
- **Integração:** Amazon API Gateway, Amazon SQS (Standard e FIFO), Amazon EventBridge, Amazon SNS
- **Armazenamento:** Amazon S3, Amazon DynamoDB
- **Gerenciamento e Governança:** AWS IAM, AWS CloudWatch (Logs e Métricas)



Vocês Conseguiram!

Estes quatro dias foram intensos, cheios de práticas, configurações e, inevitavelmente, alguns desafios. Cada erro encontrado e solucionado é uma oportunidade de aprendizado valiosa e faz parte intrínseca do processo de desenvolvimento na nuvem. A persistência e a curiosidade que demonstraram são admiráveis.

Vocês não apenas seguiram passos, mas construíram algo complexo e funcional, peça por peça. Parabéns por chegarem até aqui, por todo o esforço e pela dedicação! O conhecimento que vocês consolidaram aqui é uma base sólida para continuar explorando o vasto universo da AWS.

Desejo a cada um de vocês todo o sucesso do mundo em suas carreiras e projetos futuros na nuvem. Continuem aprendendo, experimentando e construindo coisas incríveis.

Voem!