

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e Informática

Curso de Engenharia de Software

Roteiro 3: Aplicações Serverless com LocalStack

Laboratório de Desenvolvimento de Aplicações Móveis e Distribuídas

Professores: Artur Mol, Cleiton Tavares e Cristiano Neto

1. Introdução

Nesta terceira etapa do trabalho, o aluno deverá escolher **UMA** das duas opções apresentadas a seguir. Ambas utilizam o LocalStack para simular serviços AWS em ambiente local.

⚠ ATENÇÃO: O aluno deve escolher APENAS UMA das opções abaixo.

OPÇÃO A: CRUD Serverless com Notificações SNS

 **Valor: 31 pontos**

A.1 Objetivo

Desenvolver uma aplicação CRUD (Create, Read, Update, Delete) utilizando arquitetura serverless com o Serverless Framework e LocalStack, integrando notificações via Amazon SNS para eventos do sistema.

A.2 Descrição

Implementar um sistema CRUD completo com as seguintes características:

- **API REST com operações CRUD** para gerenciamento de recursos
- **Funções Lambda** para cada operação (Create, Read, Update, Delete)
- **Persistência de dados** utilizando DynamoDB
- **Notificação via SNS** em pelo menos um evento do CRUD
- **Ambiente local** simulado com LocalStack

A.3 Stack Tecnológica

Tecnologia	Descrição
Serverless Framework	Framework para deploy de aplicações serverless
LocalStack	Emulador local dos serviços AWS
AWS Lambda	Funções serverless para lógica de negócios

Tecnologia	Descrição
API Gateway	Exposição dos endpoints REST
DynamoDB	Banco de dados NoSQL para persistência
Amazon SNS	Serviço de notificações em tópico

A.4 Funcionalidades Obrigatórias

- CRUD Completo:** Implementar as 4 operações básicas via endpoints REST
- Notificação SNS:** Publicar mensagem em um tópico SNS quando um recurso for criado ou atualizado
- Subscriber:** Implementar pelo menos um subscriber que receba as notificações do tópico
- Validação:** Validar dados de entrada nas operações de criação e atualização

A.5 Endpoints da API

Método	Endpoint	Descrição
POST	/items	Criar novo item + notificação SNS
GET	/items	Listar todos os itens
GET	/items/{id}	Buscar item por ID
PUT	/items/{id}	Atualizar item existente
DELETE	/items/{id}	Remover item

A.6 Entregáveis

- Código-fonte** do projeto no repositório Git
- Arquivo serverless.yml** com configuração completa
- Funções Lambda** implementadas para cada operação CRUD
- Configuração do tópico SNS** e subscriber
- README.md** com instruções de execução
- Evidências de testes** (screenshots ou logs) demonstrando o funcionamento

OPÇÃO B: Simulação de Cloud com LocalStack (S3)

● Valor: 31 pontos

B.1 Contexto

Introdução a Cloud AWS em ambiente local. Esta opção foca na substituição do armazenamento de arquivos locais, introduzindo armazenamento de objetos (S3) para as fotos tiradas no aplicativo móvel.

B.2 Objetivo

Configurar o LocalStack para simular um bucket S3 da AWS localmente, permitindo que as fotos tiradas no App Mobile sejam armazenadas "na nuvem" em vez de ficarem apenas no dispositivo.

B.3 Especificação

💡 **Situação Atual:** As fotos tiradas no App Mobile ficam apenas no celular. O aluno deve configurar o LocalStack para simular um bucket S3 da AWS localmente.

B.4 Requisitos Técnicos

1. **Docker Compose:** Configurar um container do LocalStack no *docker-compose.yml* expondo as portas necessárias.
2. **Serviço de Upload (Backend):** Criar um endpoint no Backend (pode ser no API Gateway ou um novo Media Service) que recebe a imagem em Base64 ou Multipart, e utiliza o SDK da AWS (*aws-sdk*) para salvar no bucket S3 do LocalStack.
3. **Integração Mobile:** Quando o usuário tirar uma foto e salvar a tarefa (online), o app deve enviar a foto para o backend, que a salvará no "S3 Local".

B.5 Roteiro da Demonstração (Sala de Aula)

🎥 **Roteiro obrigatório para apresentação em sala**

1. **Infraestrutura:** Rodar *docker-compose up* e mostrar o LocalStack subindo.
2. **Configuração:** Executar comando via terminal (AWS CLI apontando para local) para listar os buckets e mostrar que o bucket *shopping-images* existe.
3. **Ação:** No app mobile, tirar uma foto de um produto e salvar.
4. **Validação:** Via terminal ou navegador de S3 local, listar os objetos do bucket e provar que a imagem foi salva lá "na nuvem local".

B.6 Entregáveis

1. **Código-fonte** do projeto no repositório Git

2. **docker-compose.yml** com configuração do LocalStack
3. **Endpoint de upload** implementado no backend
4. **Integração no app mobile** para envio de fotos
5. **README.md** com instruções de execução
6. **Evidências** (screenshots ou logs) demonstrando o funcionamento

2. Critérios de Avaliação (Ambas as Opções)

Critério	Peso
Implementação correta das funcionalidades principais	40%
Integração com serviços AWS (SNS ou S3)	30%
Organização do código e boas práticas	15%
Documentação (README e comentários)	15%

3. Observações Gerais

- O trabalho pode ser realizado individualmente ou em dupla
- **Escolha apenas UMA das opções (A ou B)**
- Para a Opção A, o domínio do CRUD (tarefas, produtos, usuários, etc.) fica a critério do aluno
- Utilizar LocalStack para simular os serviços AWS localmente em ambas as opções
- A apresentação/demonstração será realizada em sala de aula
- Em caso de dúvidas, consulte os professores durante as aulas de laboratório

4. Comparativo das Opções

Aspecto	Opção A (CRUD + SNS)	Opção B (S3)
Foco Principal	API REST + Mensageria	Armazenamento de arquivos
Serviços AWS	Lambda, DynamoDB, SNS	S3
Integração Mobile	Opcional	Obrigatória
Complexidade	Mais funções, menos integração	Menos funções, mais integração

Bom trabalho!