

**GUTHEMBERG DA SILVA SAMPAIO**

**PROJECT HAUL**

## **TECHNICAL REPORT**

### **HAUL**

Um app de logística que oferece encomendas vindas de e-commerce a passageiros de avião cadastrados e verificados, para transporte expresso até o destinatário em troca de uma quantia.

### **Arquitetura**

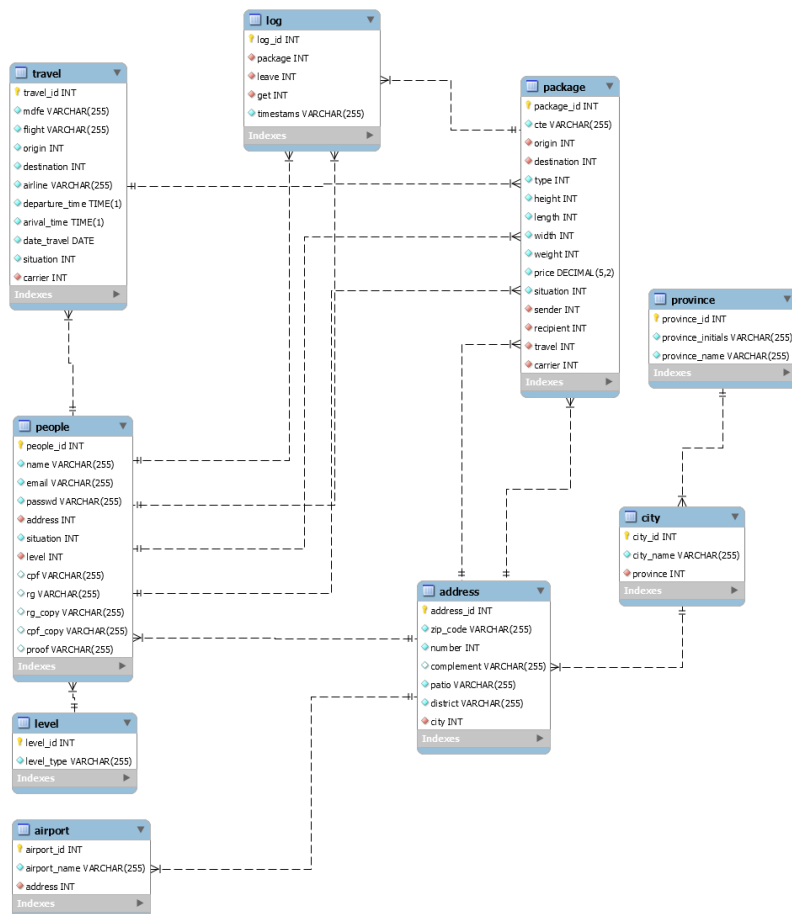
A arquitetura utilizada no seguinte projeto é a Serverless (sem servidor), devido a sua facilidade e rapidez no desenvolvimento e a alta escalabilidade da aplicação.

Sua facilidade é devido à necessidade “zero” de configuração do servidor, sendo necessário apenas o foco nos objetivos de negócios da aplicação.

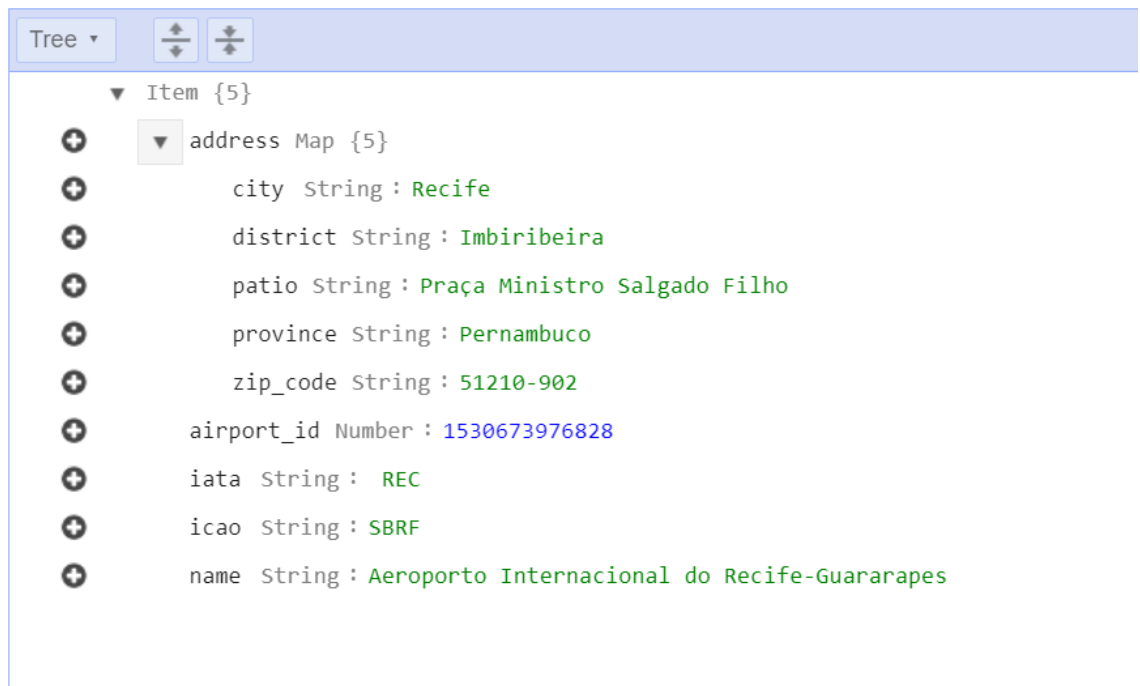
#### **1. Back-End (Em desenvolvimento)**

##### **a. DynamoDB**

Inicialmente pensou-se em utilizar um banco de dados relacional para o armazenamento de dados da aplicação.

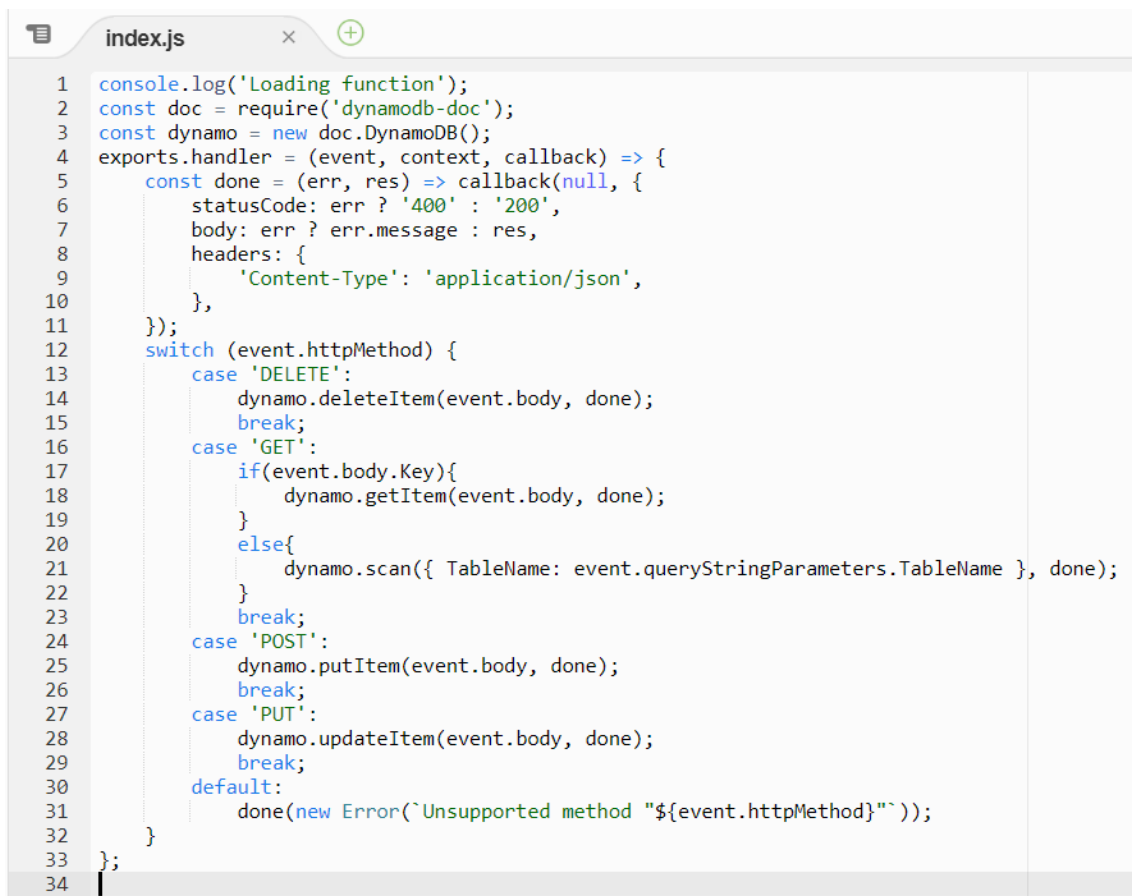


Devido a integração nativa do banco NoSQL DynamoDB com o AWS Lambda foi feita a mudança para agilizar o processo de desenvolvimento. Como o banco escolhido não é relacional foi feita uma adaptação, como por exemplo, retirar a normalização, colocando as colunas com mais de uma palavra em forma de árvore.



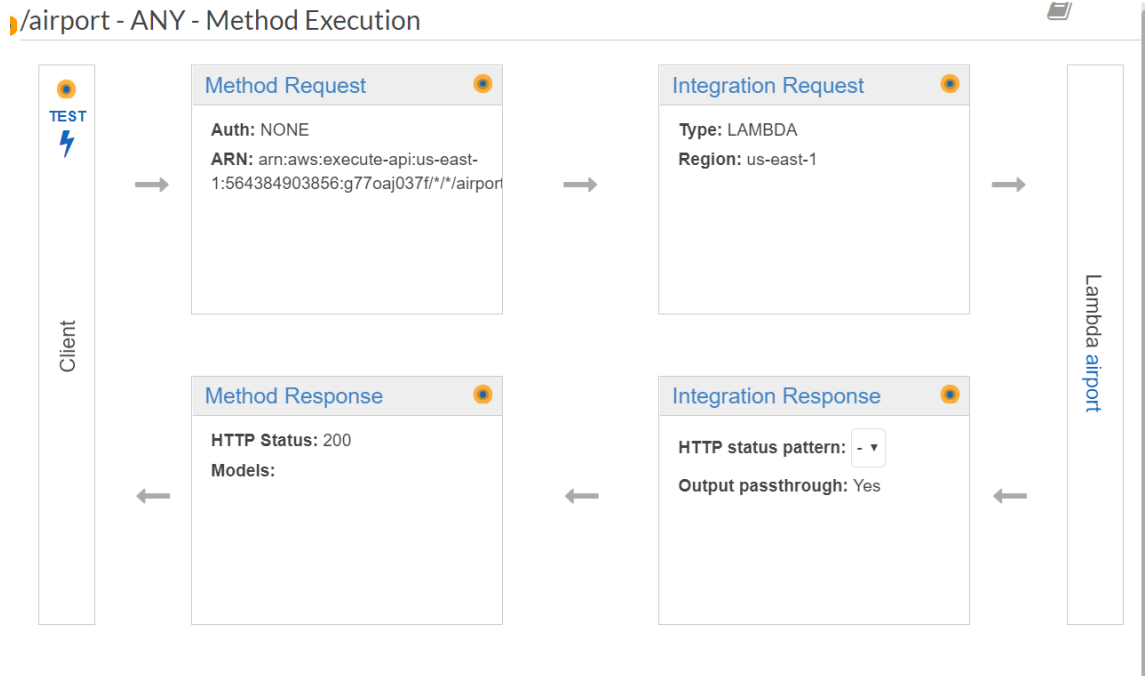
## b. AWS Lambda

AWS Lambda é onde o código ou função é implementada, no caso de exemplo temos o código de um Endpoint completo que é utilizada na nossa aplicação.



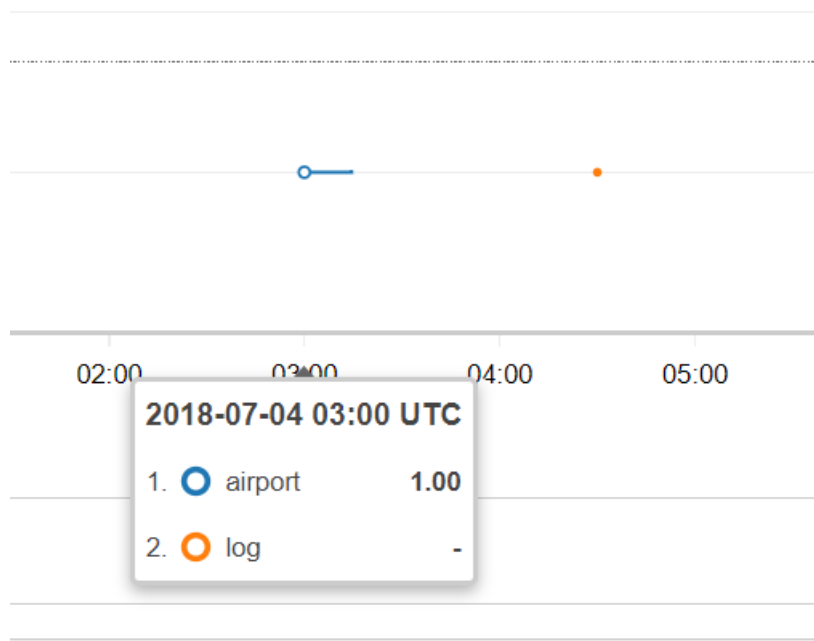
### c. Amazon API Gateway

O Gateway da Amazon é responsável por interligar as requisições do front-end com a função lambda. A imagem mostra a tela de teste da API Gateway.



### d. CloudWatch

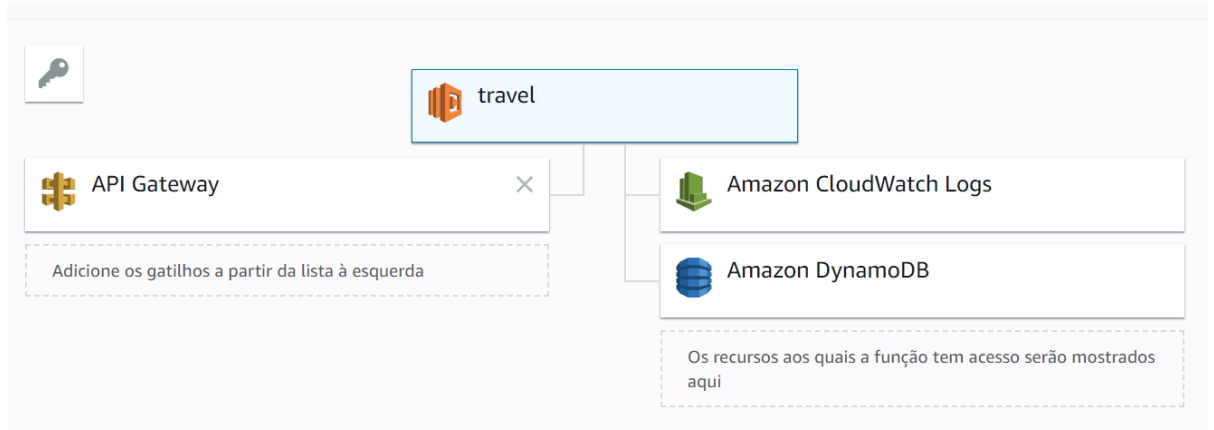
O CloudWatch é responsável por salvar os logs das chamadas das funções Lambdas, por monitorar o uso de recursos da aplicação e por acionar eventos caso aconteça algum imprevisto.



## e. IAM

Identity and Access Management é responsável pelo controle de acesso de toda aplicação, onde é definido as regras e as políticas de uso, por exemplo, a comunicação do API Gateway com a função Lambda, autorizar a comunicação externa com a API.

Aqui é onde termina o que é necessário para se ter uma aplicação Serverless na AWS.



## f. Amazon Cognito(Em desenvolvimento)

O Cognito é responsável pelo acesso do usuário, devido à necessidade de um front-end esta parte ainda está em desenvolvimento.

## 2. Front-End(Em planejamento)

### ReactJS ou Angular 6?



## **Referências Bibliográficas**

<https://serverless-stack.com/>

<http://viniciusgarcia.me/development/o-que-eh-arquitetura-serverless/>

<https://aws.amazon.com/pt/documentation/>

<https://trends.google.com.br/trends/explore?q=Angular,%2Fm%2F012l1vxv>