

**ContaCerta: Construindo um produto
fácil, escalável e de baixo custo**



Felipe Valtl de Mello

Desenvolvedor de Software e SRE na Transfeera

Há 8 anos ~~gerando bugs~~ desenvolvendo software

felipe@transfeera.com

github.com/valtlfelipe



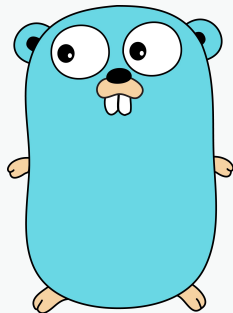
Tiago Bonatti

COO e Desenvolvedor de Software na Transfeera

+10 anos de programação

tiago@transfeera.com

github.com/t-bonatti



Como começou...



Hackathon 🍌

Payment Shift

Em abril de 2018 fomos convidados a participar de um Hackathon na UniSociesc Blumenau: Uma maratona de programação e desenvolvimento de software focada em meios de pagamento.

Como começou...



Como começou...



- Comparação entre documento e nome
- Crawler no Internet Banking da Caixa para validar os dados bancários

E ganhamos o Hackathon

API Conta Certa vence o primeiro Hackathon na UniSociesc Blumenau

11/04/2018



API Conta Certa vence o primeiro Hackathon na UniSociesc Blumenau

Equipe ganhadora da maratona de 24 horas de programação que aconteceu no final de semana criou um software que checa as informações da conta bancária e do CPF/CNPJ do comprador com os dados do pagamento fornecidos pelo cliente

<https://www.unisociesc.com.br/api-conta-certa-vence-o-primeiro-hackathon-na-unisociesc-blumenau/>



A necessidade...

- De todas as transferências que fazíamos, aproximadamente 8% estavam com os dados errados
- Transferências com dados errados eram atrasados
- Necessitava verificação manual do erro



Exemplo: Operação + Número da Conta



Detalhes da transferência



R\$ 1,00

Ciclano Da Silva

368.231.654-05

Criada



Banco

Caixa Econômica

Tipo de operação

013

Tipo da conta

Conta poupança

Agência

3519

Conta poupança

0136231 - 8

Exemplo: Operação + Número da Conta



Sugestão de correção para este favorecido

Conta inválida.
Sugestão: 6231

✓ Aceitar sugestão

Dados bancários

Banco do favorecido

Caixa Econômica

Agência

3519

Dígito (opcional)

Tipo da conta

Conta poupança

Conta poupança

0136231

Dígito

8

Conta inválida. Indicamos o
valor: 6231

Exemplo: Dígito da Conta



Detalhes da transferência



R\$ 1,00

Ciclano Da Silva

368.231.654-05

Criada



Banco

Caixa Econômica

Tipo de operação

013

Tipo da conta

Conta poupança

Agência

3519

Conta poupança

6231-0

Exemplo: Dígito da Conta



Sugestão de correção para este favorecido

Agência, tipo da conta, conta ou dígito verificador da conta inválido.
Sugestão: 8

✓ Aceitar sugestão

Dados bancários

Banco do favorecido

Caixa Econômica

Agência

3519

Dígito (opcional)

Tipo da conta

Conta poupança

Conta poupança

013 6231

Dígito

0

Agência, tipo da conta, conta ou dígito verificador da conta inválido. Nossos cálculos indicam o dígito 8

Exemplo: Razão Social Inválida

Detalhes da transferência



R\$ 1,00

Facebook Ltda

06.990.590/0001-23

Criada



Sugestão de correção para este favorecido

Nome informado não coincide com o nome no Documento

Sugestão: GOOGLE BRASIL INTERNET LTDA.

✓ Aceitar sugestão

Dados do favorecido

Nome do favorecido

Facebook Ltda

CPF / CNPJ

06.990.590/0001-23



Banco

Banco do Brasil

Tipo da conta

Conta corrente

Agência

5214 - 0

Conta corrente

13568 - 2

Porquê serverless?

Nossa estrutura na época...



**Amazon
EC2**



Serviço de terceiro...



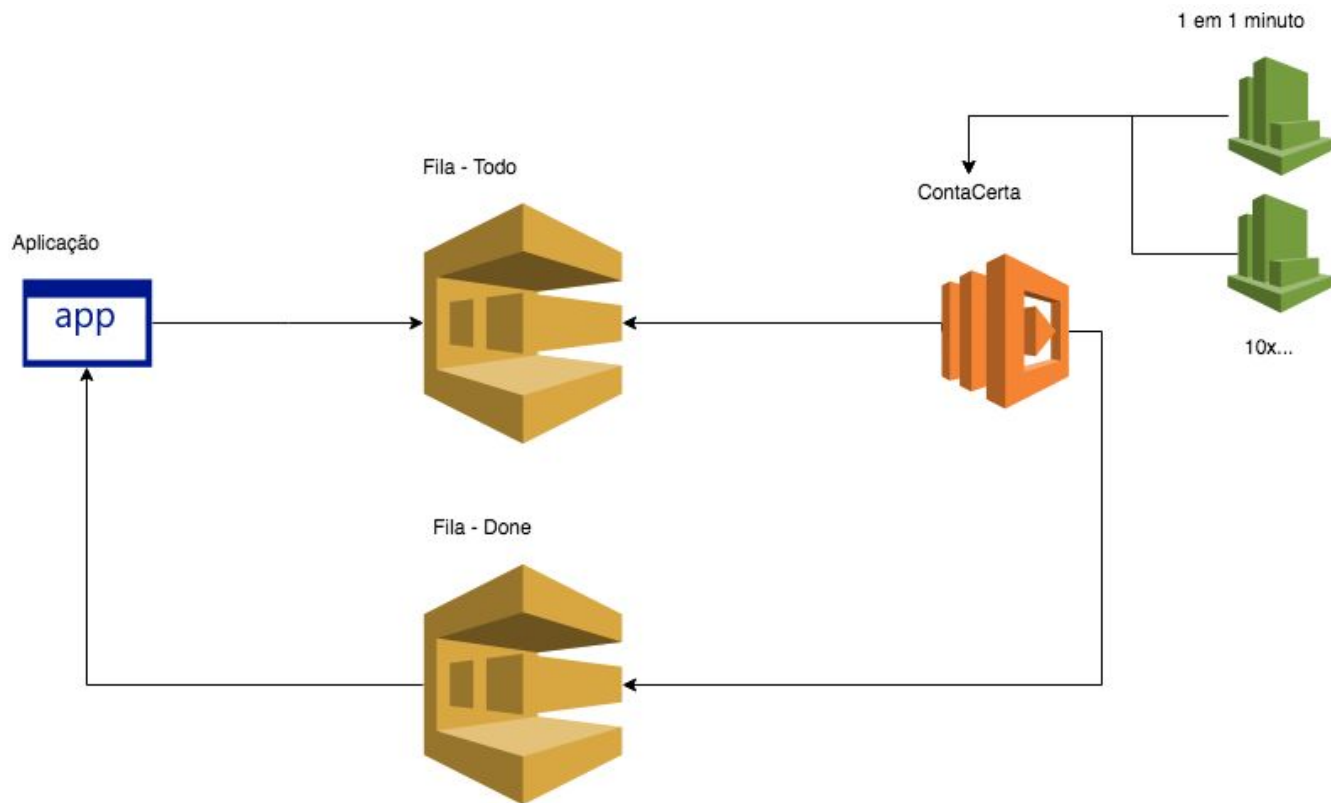
Serverless

- Conceito
 - Sem infra
 - Sob demanda
 - Stateless
 - Paga pelo uso
- Framework
 - CLI
 - Open source
 - AWS / Azure / GCP

```
$ npm install -g serverless
```



Versão 1



```
service: contacerta

provider:
  name: aws
  runtime: nodejs8.10
  stage: ${env:STAGE}
  region: us-east-1
  timeout: 60
  memorySize: 128
  environment:
    NODE_ENV: ${env:NODE_ENV}
    COMPLETE_QUEUE_URL: ${env:COMPLETE_QUEUE_URL}
    READ_QUEUE_URL: ${env:READ_QUEUE_URL}
  iamRoleStatements:
    - Effect: Allow
      Action:
        - sqs:SendMessage
        - sqs:DeleteMessage
        - sqs:ReceiveMessage
      Resource: "*"

functions:
  run:
    handler: index.handler
    events:
      - schedule: rate(1 minute)
      - schedule: rate(1 minute)
      # ... 10x
```

```
const Queue = require('./lib/services/Queue');

const READ_QUEUE_URL = process.env.READ_QUEUE_URL;
const COMPLETE_QUEUE_URL = process.env.COMPLETE_QUEUE_URL;

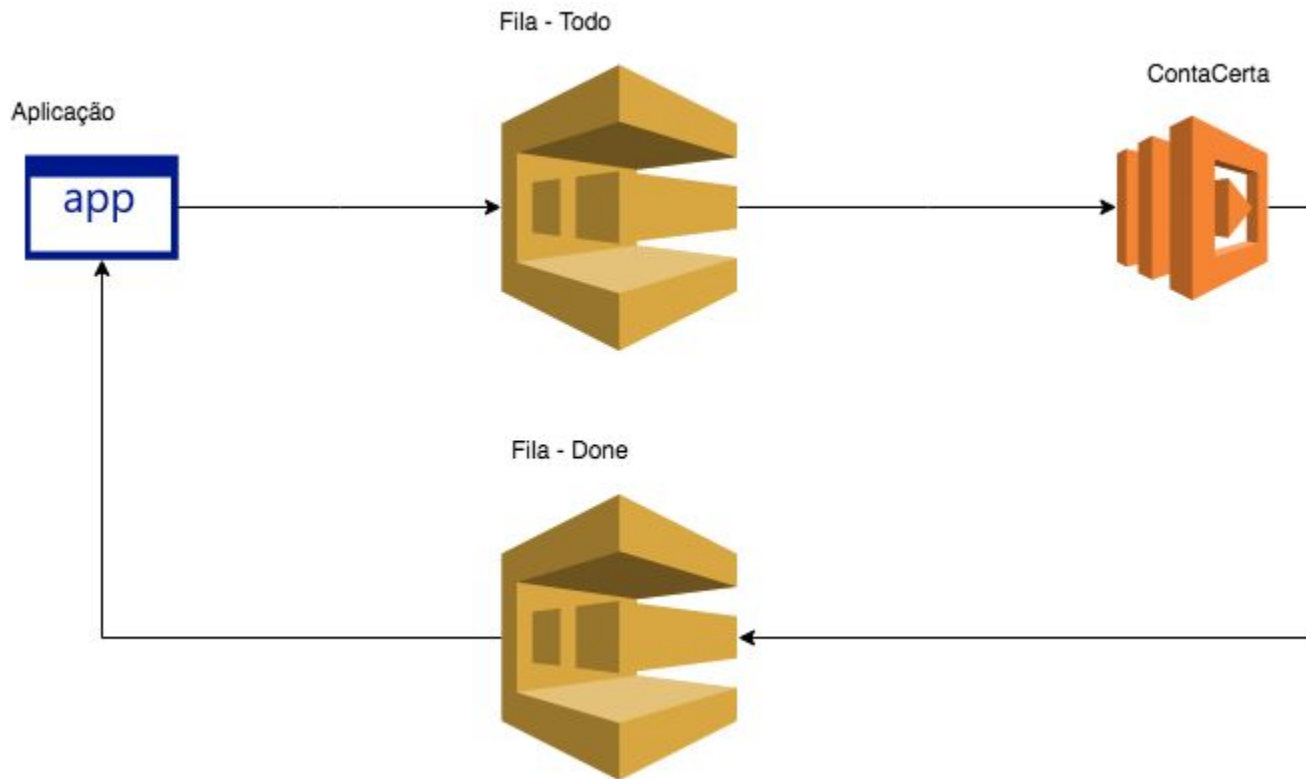
const readQueue = new Queue(READ_QUEUE_URL);
const completeQueue = new Queue(COMPLETE_QUEUE_URL);

module.exports.handler = async (event, context, callback) => {
  const messages = await readQueue.get();

  try {
    await Promise.all(messages.map(message => {
      const result = await new ContaCerta().run(message);
      await completeQueue.send(result);
      await readQueue.delete(message.ReceiptHandle);
    }));
  } catch (e) {
    return callback(e);
  }

  return callback();
};
```


Versão 2



```
service: contacerta

provider:
  name: aws
  runtime: nodejs10.x
  stage: ${env:STAGE}
  region: us-east-1
  timeout: 60
  memorySize: 128
  environment:
    NODE_ENV: ${env:ENVIRONMENT_TYPE}
    COMPLETE_QUEUE_URL: ${env:COMPLETE_QUEUE_URL}
  iamRoleStatements:
    - Effect: Allow
      Action:
        - sqs:SendMessage
      Resource: "*"

functions:
  run:
    handler: index.handler
    reservedConcurrency: 5
    events:
      - sqs: ${env:READ_QUEUE_URL_ARN}
```

```
const Queue = require('./lib/services/Queue');

const COMPLETE_QUEUE_URL = process.env.COMPLETE_QUEUE_URL;

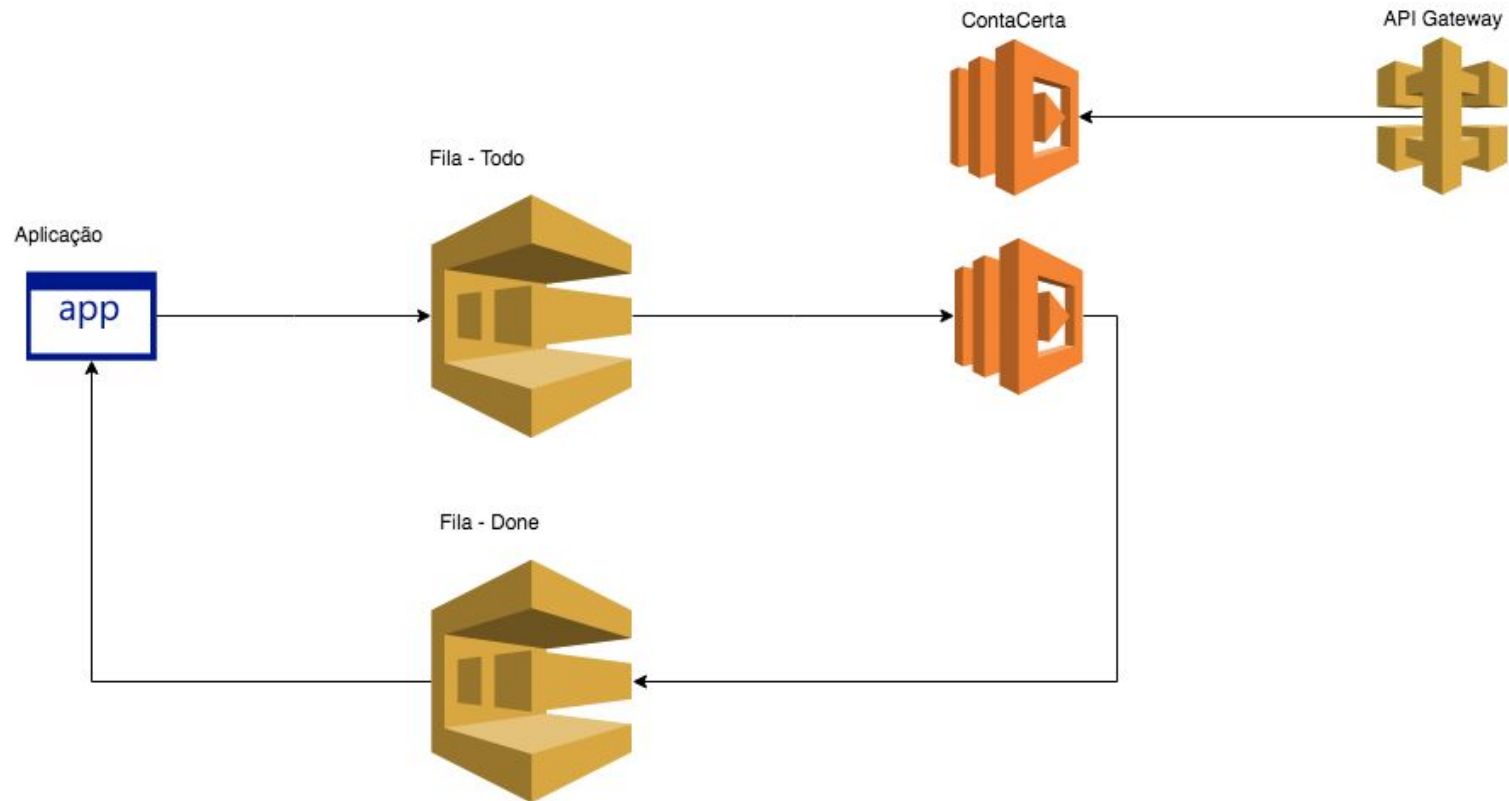
const completeQueue = new Queue(COMPLETE_QUEUE_URL);

module.exports.handler = async (event, context, callback) => {
  const messages = event.Records;

  try {
    await Promise.all(messages.map(message => {
      const result = await new ContaCerta().run(message);
      await completeQueue.send(result);
    }));
  } catch (e) {
    return callback(e);
  }

  return callback();
};
```

Versão 3



```
service: contacerta

provider:
  name: aws
  runtime: nodejs10.x
  stage: ${env:STAGE}
  region: us-east-1
  timeout: 60
  memorySize: 128
  environment:
    NODE_ENV: ${env:ENVIRONMENT_TYPE}
    COMPLETE_QUEUE_URL: ${env:COMPLETE_QUEUE_URL}
  iamRoleStatements:
    - Effect: Allow
      Action:
        - sqs:SendMessage
      Resource: "*"

functions:
  run:
    handler: index.handler
    reservedConcurrency: 5
    events:
      - sqs: ${env:READ_QUEUE_URL_ARN}
  validateAPI:
    handler: index.APIHandler
    events:
      - http:
          path: validate
          method: post
          private: true
```

```
module.exports.APIhandler = async (event, context, callback) => {  
  const body = JSON.parse(event.body);  
  const result = await new ContaCerta().run(body);  
  
  const response = {  
    statusCode: 200,  
    headers: {  
      'Content-Type': 'application/json',  
      'Access-Control-Allow-Origin': '*'  
    },  
    body: JSON.stringify(result)  
  };  
  
  return callback(null, response);  
};
```



```
$ serverless deploy
```

Service Information

service: contacerta

stage: devteste

region: us-east-1

stack: contacerta-devteste

resources: 10

api keys:

None

endpoints:

POST - <https://tgm6xhjz3h.execute-api.us-east-1.amazonaws.com/devteste/validate>

functions:

validateAPI: contacerta-devteste-validateAPI

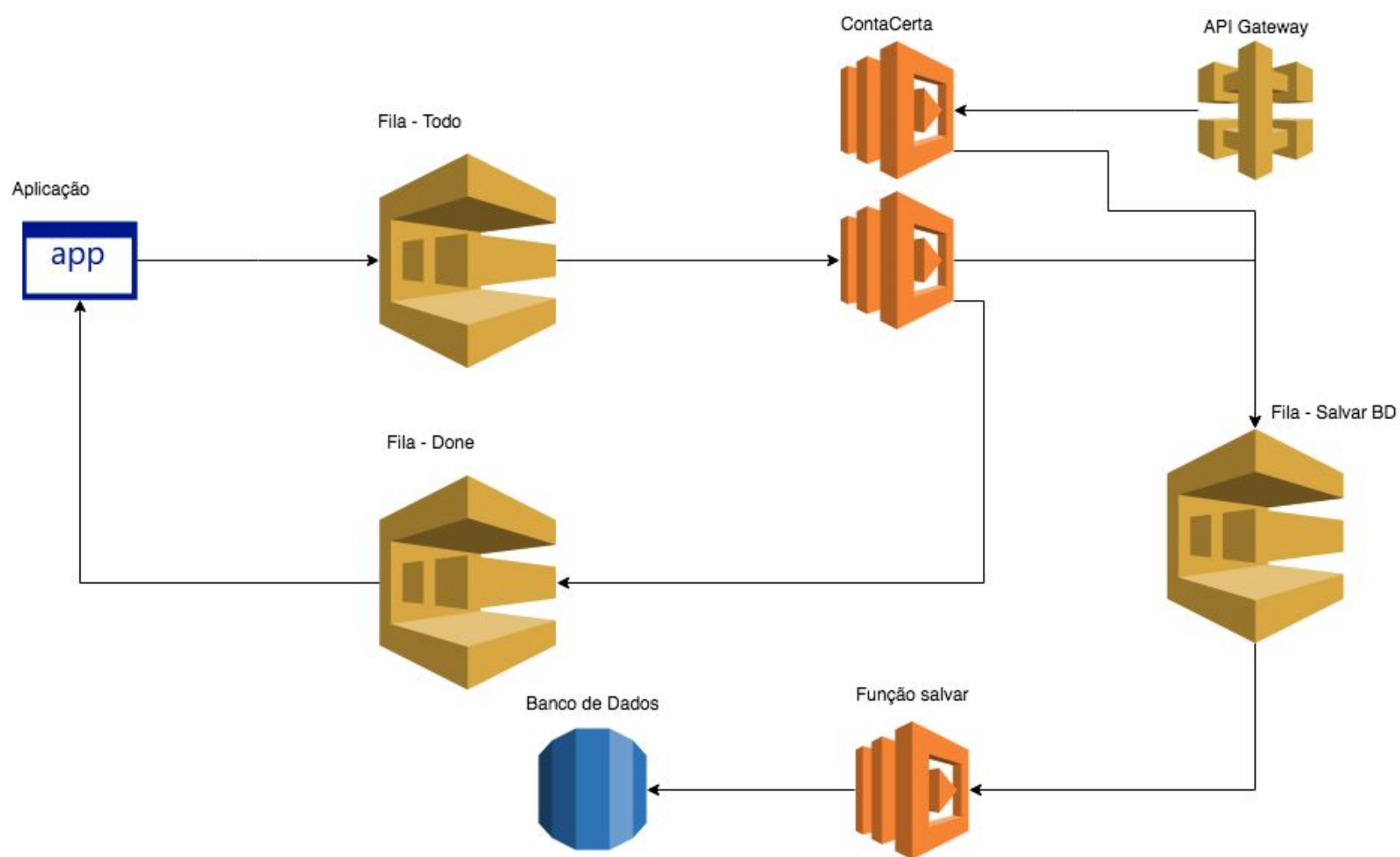
layers:

None

Melhorias

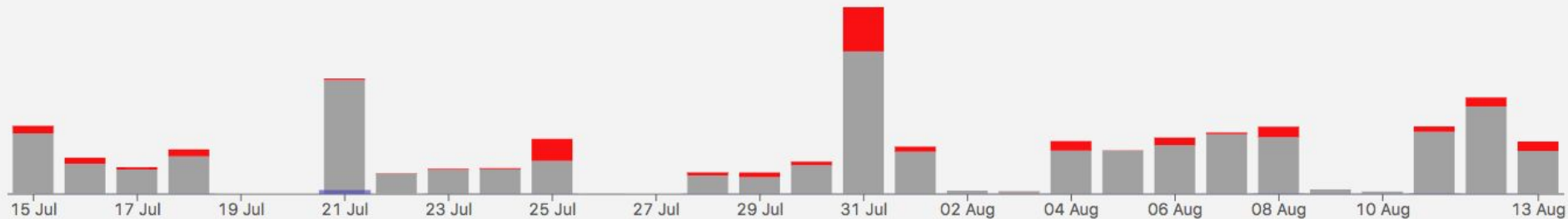
- Logs no Banco de Dados
- Operação síncrona
- Abrir e Fechar conexão
- Quantidade de conexão não é escalável
- Solução: Fila e Lambda










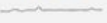






```
functions:
  insertAPIDataToDB:
    handler: index.insertAPIDataToDB
    reservedConcurrency: 1
    events:
      - sqs:
          arn: ${env:DB_QUEUE_ARN}
          batchSize: 10
```

```
module.exports.insertAPIDataToDB = async (event, context, callback) => {  
  const messages = event && event.Records;  
  const repo = new DataRepository();  
  
  try {  
    await Promise.all(messages.map(m => {  
      const body = JSON.parse(m.body);  
      return repo.create(body.data, body.complement);  
    }));  
  
    await repo.close();  
  
    return callback();  
  } catch (e) {  
    return callback(e);  
  }  
};
```



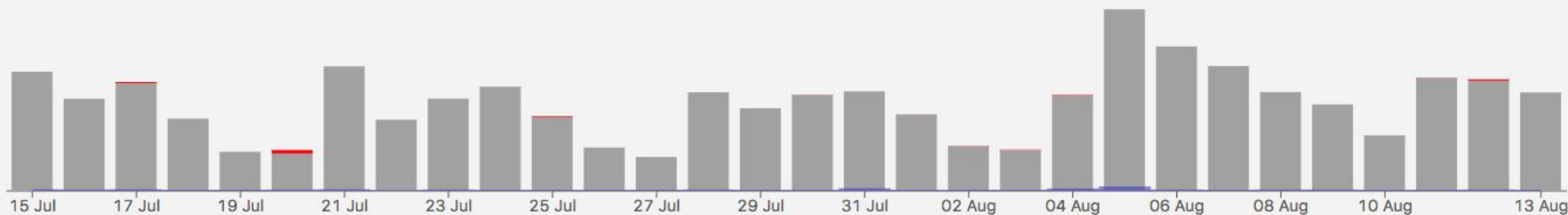
					
39.2k	5.1k	1094ms	58%	0.17\$	us-east-1
 INVOCATIONS	 ERRORS	 DURATION	 MEMORY	 COST	 REGION

- 40k execuções
- US\$ 0.17

- 28k execuções
- US\$ 0.04
- 756ms

contacerta-prod-validateAPI

1 hour 1 day 7 days 30 days 



Σ
27.7k
INVOCATIONS

🏠
102
ERRORS

🕒
756ms
DURATION

📊
59%
MEMORY

💰
0.04\$
COST

🌐
us-east-1
REGION

O que deu de errado?

- Loop SQS com Lambda
- Lambda Timeout
- Cold start
- Limite da AWS



<https://transfeera.com/lp/conta-certa/>

<https://transfeera.com/lp/conta-certa/>

Obrigado!
Dúvidas?