Cap 10 - Aplicações em Cloud Computing - Atividade 2

Integrantes

- Abigail Lima - RM96351
- Kaique Bernardo - RM96395

Links:

- Repositorio: https://github.com/abigailmvlima/traveller
- Publicação Site: https://abigailmvlima.github.io/gullivertravell/
- Tripadvisor: https://tripadvisor-content-api.readme.io/reference/searchforlocations

Escolha de API

Lambda AWS + Tripadvisor

APIs UTILIZADAS

Tripadvisor API

A API do Tripadvisor é uma interface de programação de aplicativos (API, na sigla em inglês) fornecida pelo Tripadvisor, que permite que desenvolvedores integrem dados e recursos do site do Tripadvisor em seus próprios aplicativos e sites.

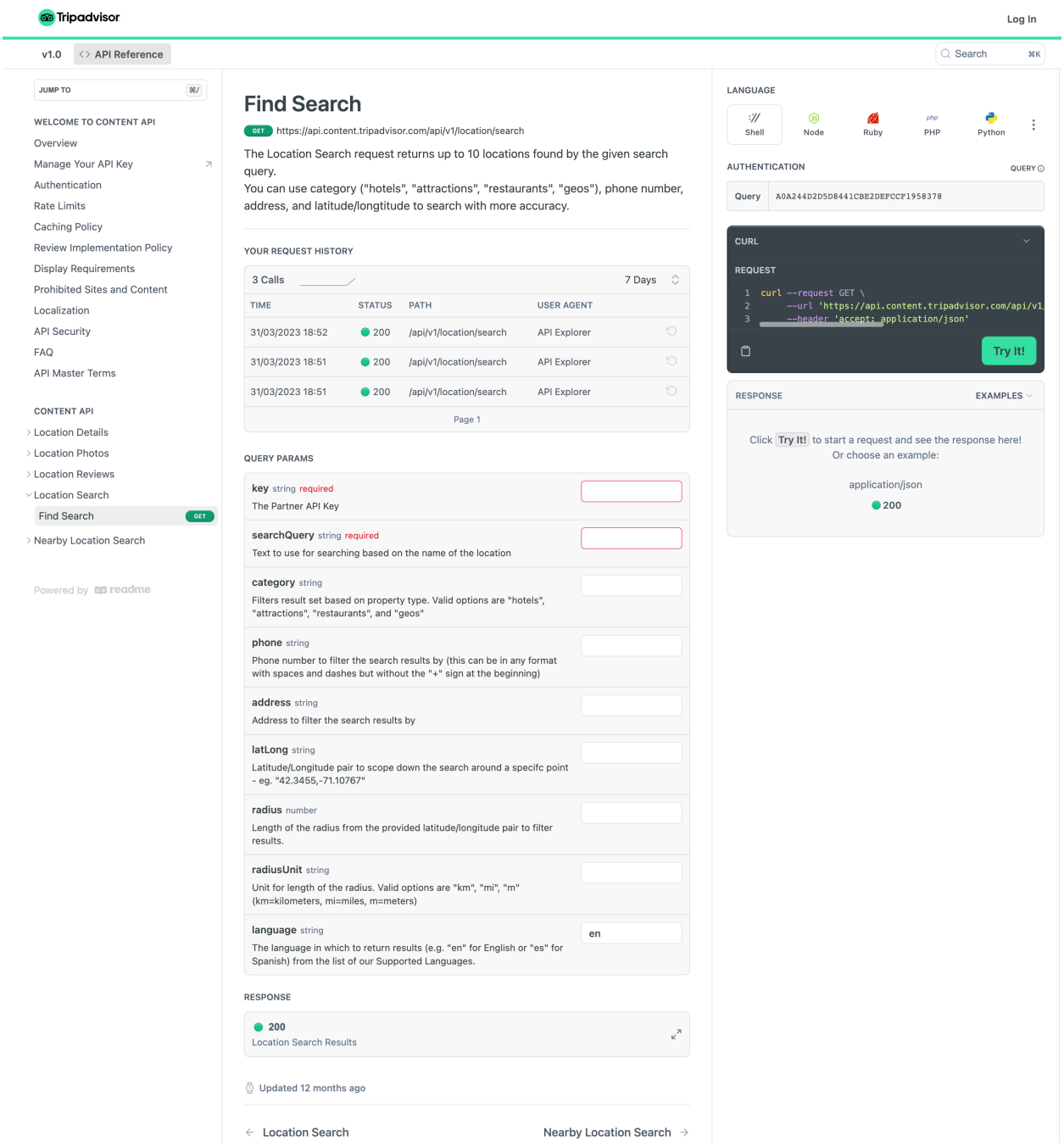
A API fornece acesso aos dados do Tripadvisor, incluindo informações sobre hotéis, restaurantes, atrações turísticas, bem como comentários, avaliações e fotos de

usuários. Com a API, os desenvolvedores podem criar aplicativos e serviços personalizados, como agregadores de avaliações, guias turísticos, aplicativos móveis de reservas de viagens e muito mais.

Documentação Tripadvisor

Tripadvisor

https://www.tripadvisor.com/developers



Utilização dos endpoint e a função de cada um

A utilização dos endpoint do tripadvisor, foi para pesquisa de localidades na apresentação da home no site. Tambem a utilização do endpoint de pesquisa das avaliações da experiencia dos usuarios "Classificação, texto de avaliação e etc …".

Outra para colher os detalhes da localidade podendo ser "Hotel, Restaurante, Ponto turistico e etc …”.

Ja por outro lado a lambda com o objetivo de ocultar os dados mais sensiveis do usuario e credenciais de acesso.

AWS Function Lambda

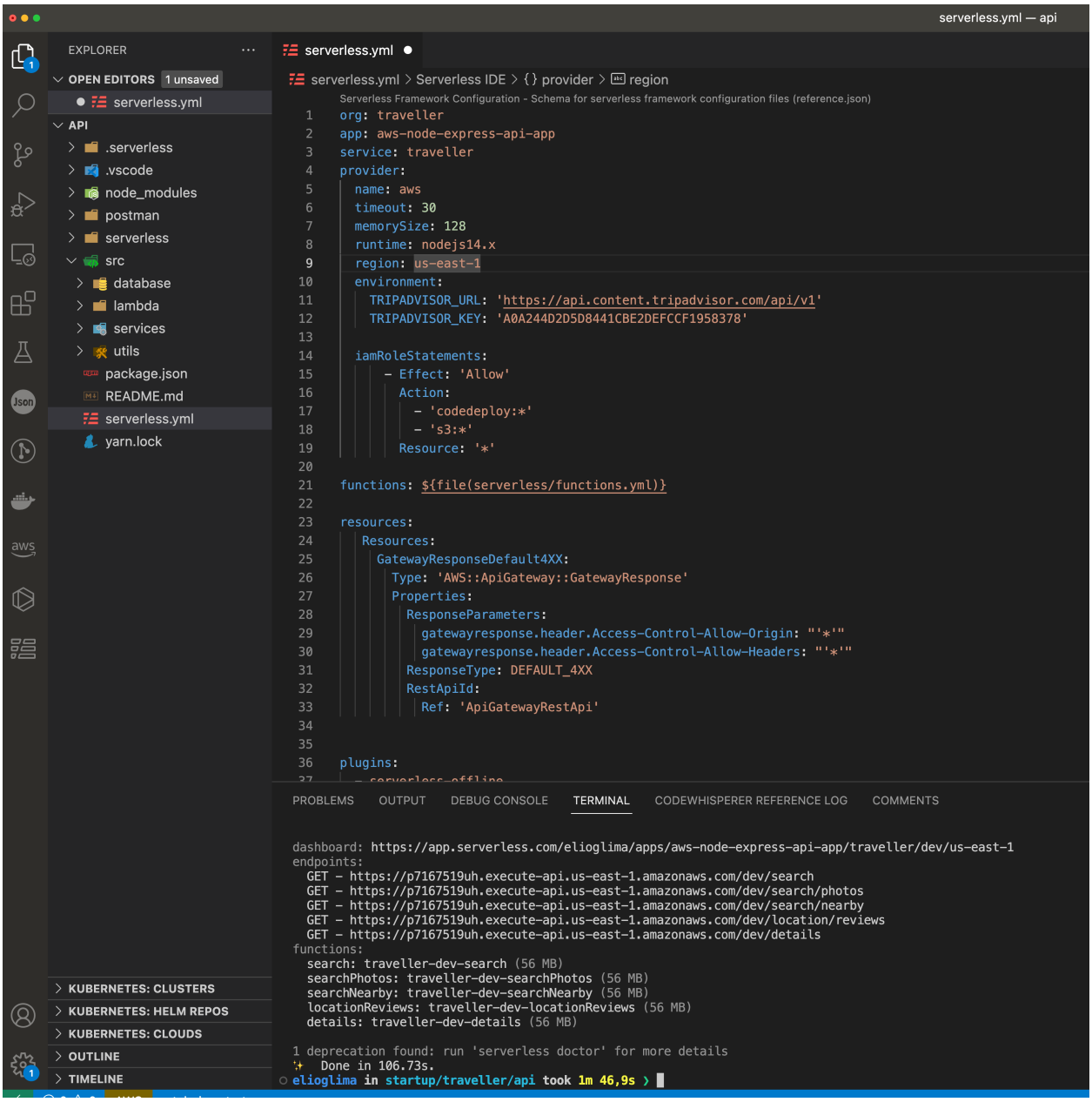
Lambda é um serviço de computação sem servidor da AWS que permite executar código em resposta a eventos sem precisar gerenciar servidores ou infraestrutura. Ao usar o Lambda, não precisa se preocupar com capacidade, escalabilidade, disponibilidade ou manutenção de servidores. Em vez disso, a AWS cuida de tudo isso para você.

Rrazões pelas quais usar o Lambda na AWS:

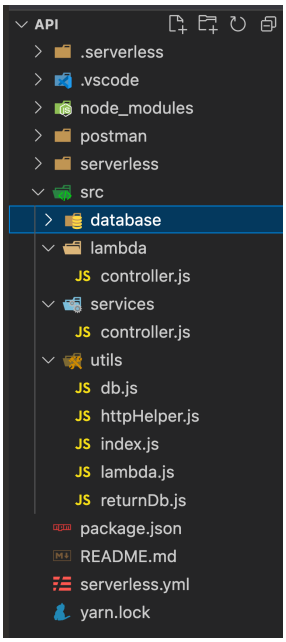
1. Custo: Com o Lambda, você paga apenas pelo tempo que sua função leva para ser executada, sem se preocupar com o tempo de inatividade ou capacidade não utilizada. Isso pode ser muito mais econômico do que manter sua própria infraestrutura ou usar servidores em nuvem tradicionais.
2. Escalabilidade: O Lambda se dimensiona automaticamente para lidar com a carga de trabalho, portanto, você não precisa se preocupar com a capacidade ou escalabilidade de seus servidores.
3. Sem servidor: Ao não precisar gerenciar servidores, você pode se concentrar em escrever e melhorar o código de sua aplicação, em vez de se preocupar com a infraestrutura.
4. Integração com outros serviços da AWS: O Lambda se integra perfeitamente com outros serviços da AWS, permitindo que você construa aplicativos altamente escaláveis e resistentes.

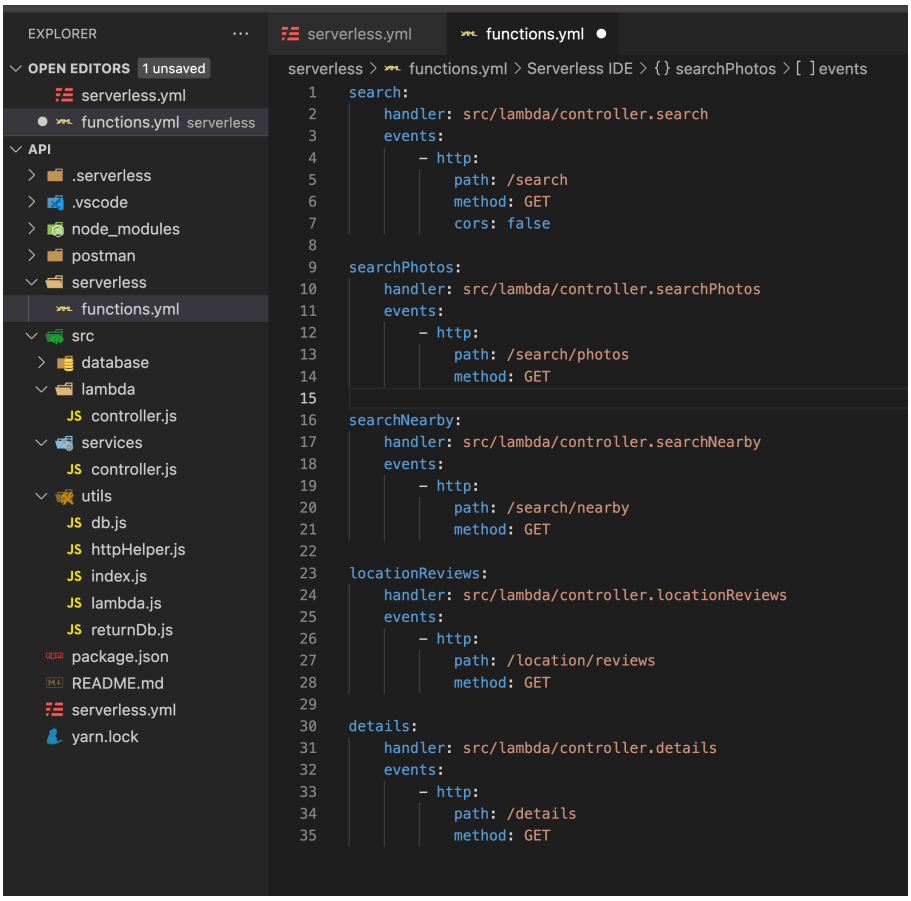
Resumindo, o Lambda é uma excelente opção se você deseja executar código em resposta a eventos de maneira eficiente, econômica e sem se preocupar com a administração de servidores.

Sobre o código fonte



Estrutura de dados dos lambdas





Implementação da Lambda x Tripadvisor

serverless.yml | functions.yml | JS controller.js

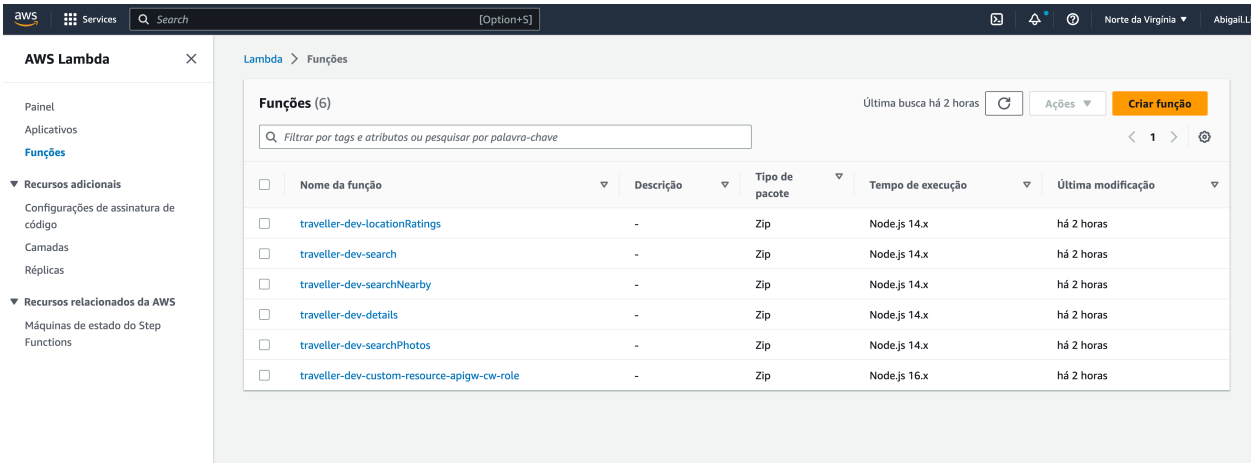
src > services > JS controller.js > ...

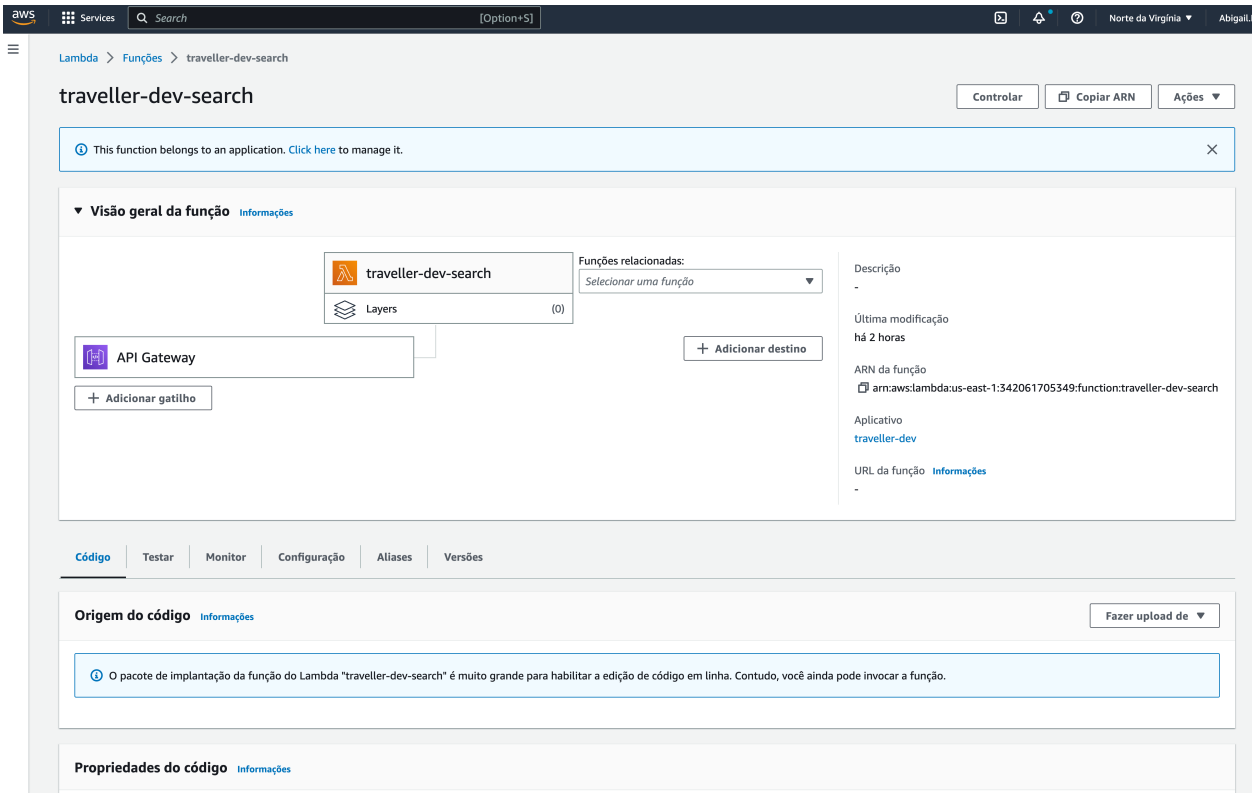
```
const axios = require('axios');
const utils = require('../utils');

const search = async ({ queryStringParameters }) => {
    try {
        const params = utils.httpHelper.objectToUri({ key: process.env.TRIPADVISOR_KEY, language:'pt', ...queryStringParameters})
        const url = `${process.env.TRIPADVISOR_URL}/location/search${params}`;
        const response = await axios.get(url)
        return utils.httpHelper.ok(response.data)
    } catch (error) {
        return utils.httpHelper.badRequest(error.message)
    }
}
const searchPhotos = async ({ queryStringParameters }) => {
    try {
        const location_id = queryStringParameters.location_id
        delete queryStringParameters.location_id

        const params = utils.httpHelper.objectToUri({ key: process.env.TRIPADVISOR_KEY, language:'pt', ...queryStringParameters})
        const url = `${process.env.TRIPADVISOR_URL}/location/${location_id}/photos${params}`;
        console.log(url)
        const response = await axios.get(url)
        return utils.httpHelper.ok(response.data)
    } catch (error) {
        return utils.httpHelper.badRequest(error.message)
    }
}
const searchNearby = async ({ queryStringParameters }) => {
    try {
        const params = utils.httpHelper.objectToUri({ key: process.env.TRIPADVISOR_KEY, language:'pt', ...queryStringParameters})
        const url = `${process.env.TRIPADVISOR_URL}/location/nearby_search${params}`;
        const response = await axios.get(url)
        return utils.httpHelper.ok(response.data)
    } catch (error) {
        return utils.httpHelper.badRequest(error.message)
    }
}
const locationReviews = async ({ queryStringParameters }) => {
    try {
        const location_id = queryStringParameters.location_id
        delete queryStringParameters.location_id
        const params = utils.httpHelper.objectToUri({ key: process.env.TRIPADVISOR_KEY, language:'pt', ...queryStringParameters})
        const url = `${process.env.TRIPADVISOR_URL}/location/${location_id}/reviews${params}`;
        const response = await axios.get(url)
        return utils.httpHelper.ok(response.data)
    } catch (error) {
        return utils.httpHelper.badRequest(error.message)
    }
}
const details = async ({ queryStringParameters }) => {
    try {
        const location_id = queryStringParameters.location_id
        delete queryStringParameters.location_id
```

Efetuando Deploy AWS

```
Deploying traveller to stage dev (us-east-1)

✔ Service deployed to stack traveller-dev (101s)

dashboard: https://app.serverless.com/elioglima/apps/aws-node-express-api-app/traveller/dev/us-east-1
endpoints:
  GET - https://p7167519uh.execute-api.us-east-1.amazonaws.com/dev/search
  GET - https://p7167519uh.execute-api.us-east-1.amazonaws.com/dev/search/photos
  GET - https://p7167519uh.execute-api.us-east-1.amazonaws.com/dev/search/nearby
  GET - https://p7167519uh.execute-api.us-east-1.amazonaws.com/dev/location/reviews
  GET - https://p7167519uh.execute-api.us-east-1.amazonaws.com/dev/details
functions:
  search: traveller-dev-search (56 MB)
  searchPhotos: traveller-dev-searchPhotos (56 MB)
  searchNearby: traveller-dev-searchNearby (56 MB)
  locationReviews: traveller-dev-locationReviews (56 MB)
  details: traveller-dev-details (56 MB)

1 deprecation found: run 'serverless doctor' for more details
✨  Done in 106.73s.
```

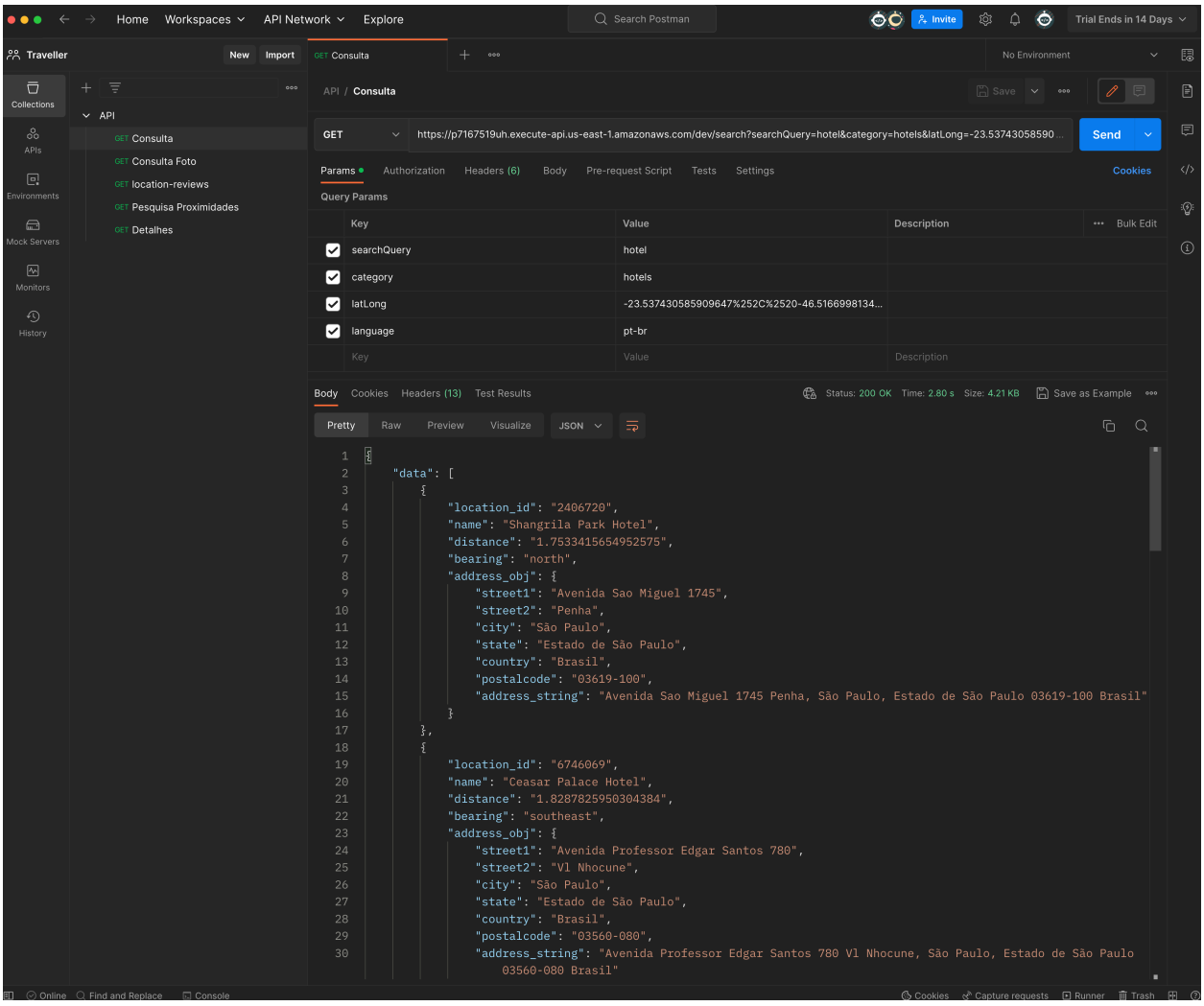
Configurações AWS Lambdas



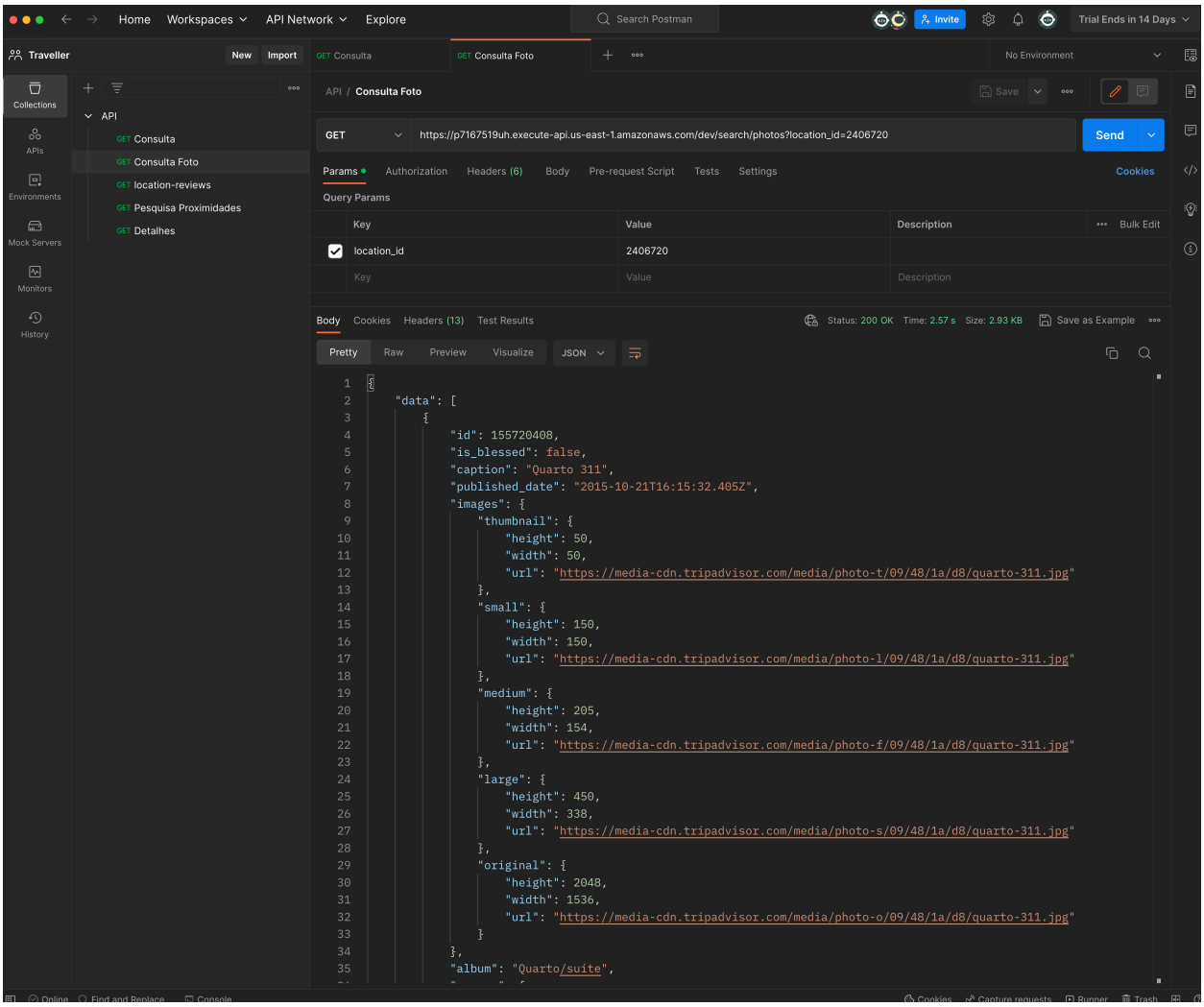


Testando Lambdas e Tripadvisor

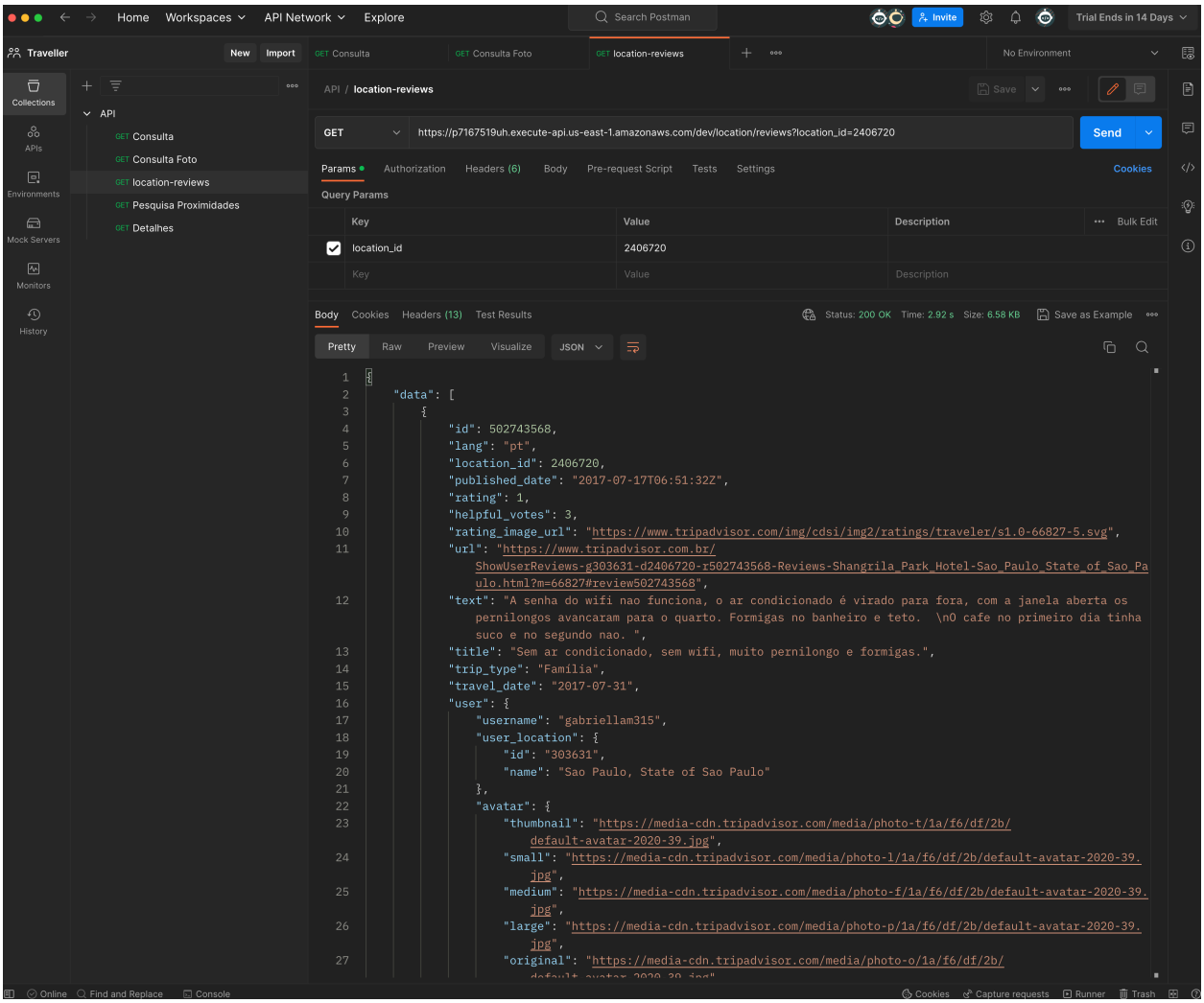
Consulta personalizada



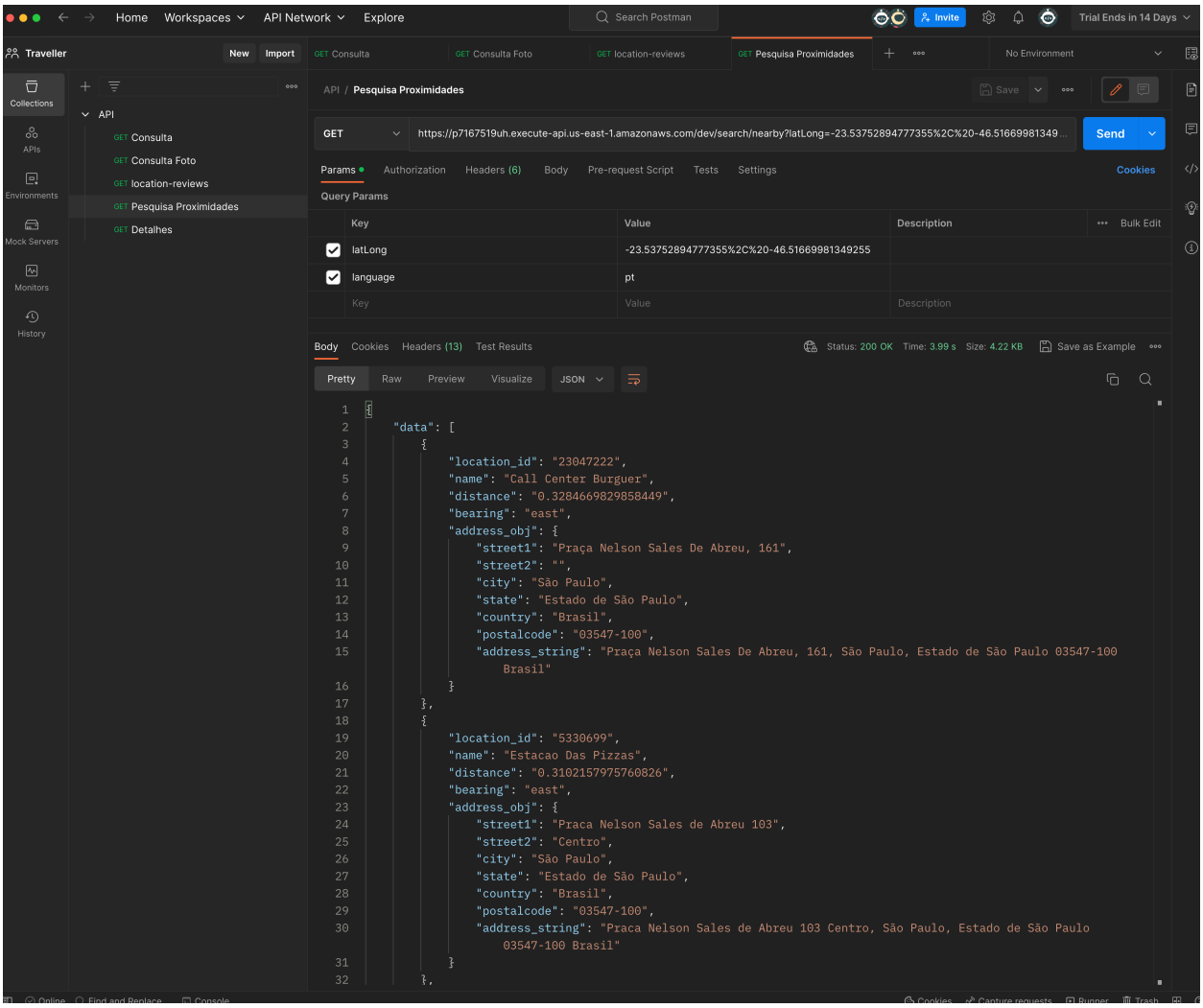
Consulta de fotos por localidade

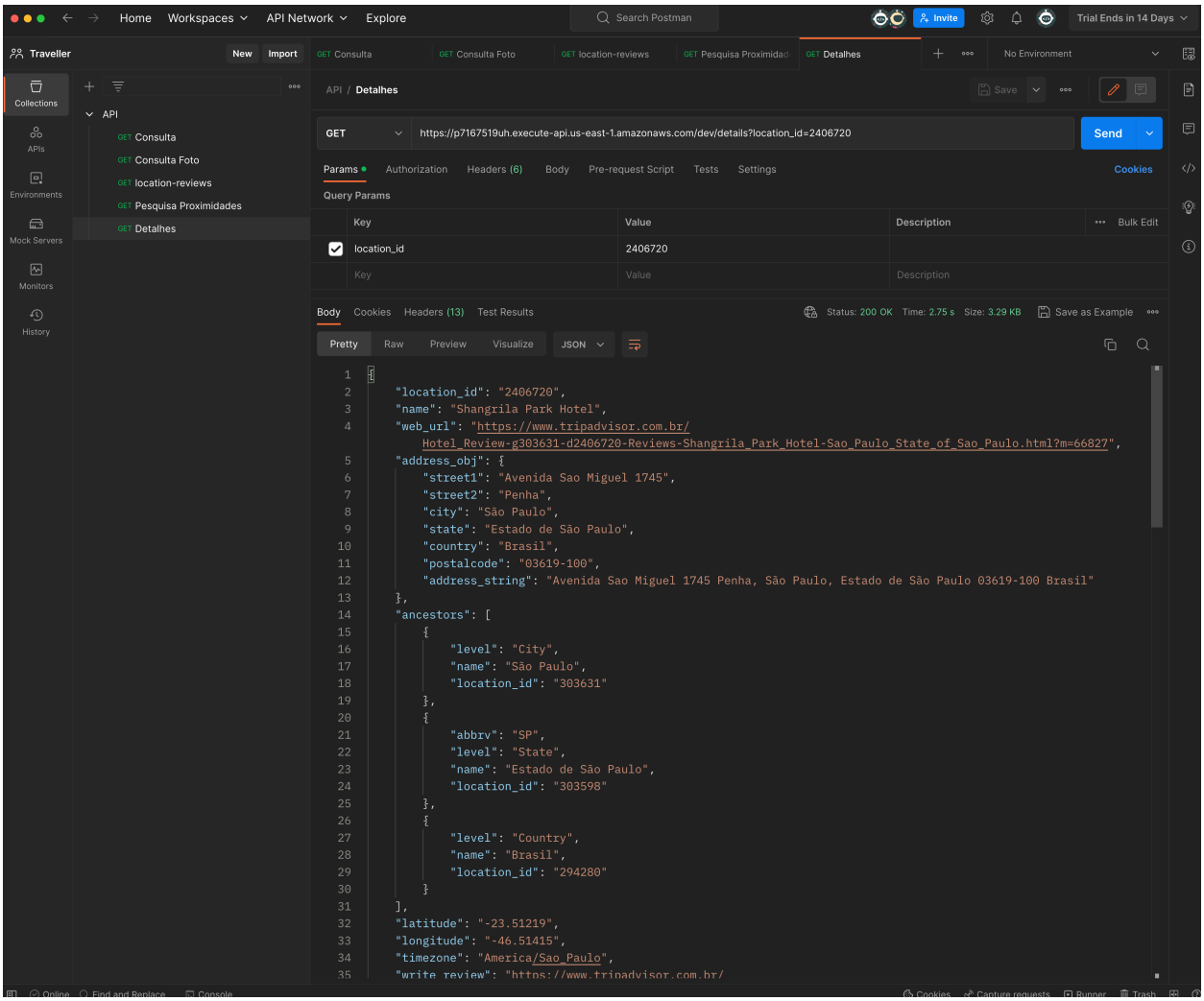


Consulta de avaliações



Pesquisa loclidades nas proximidade de uma determinada geolocalização

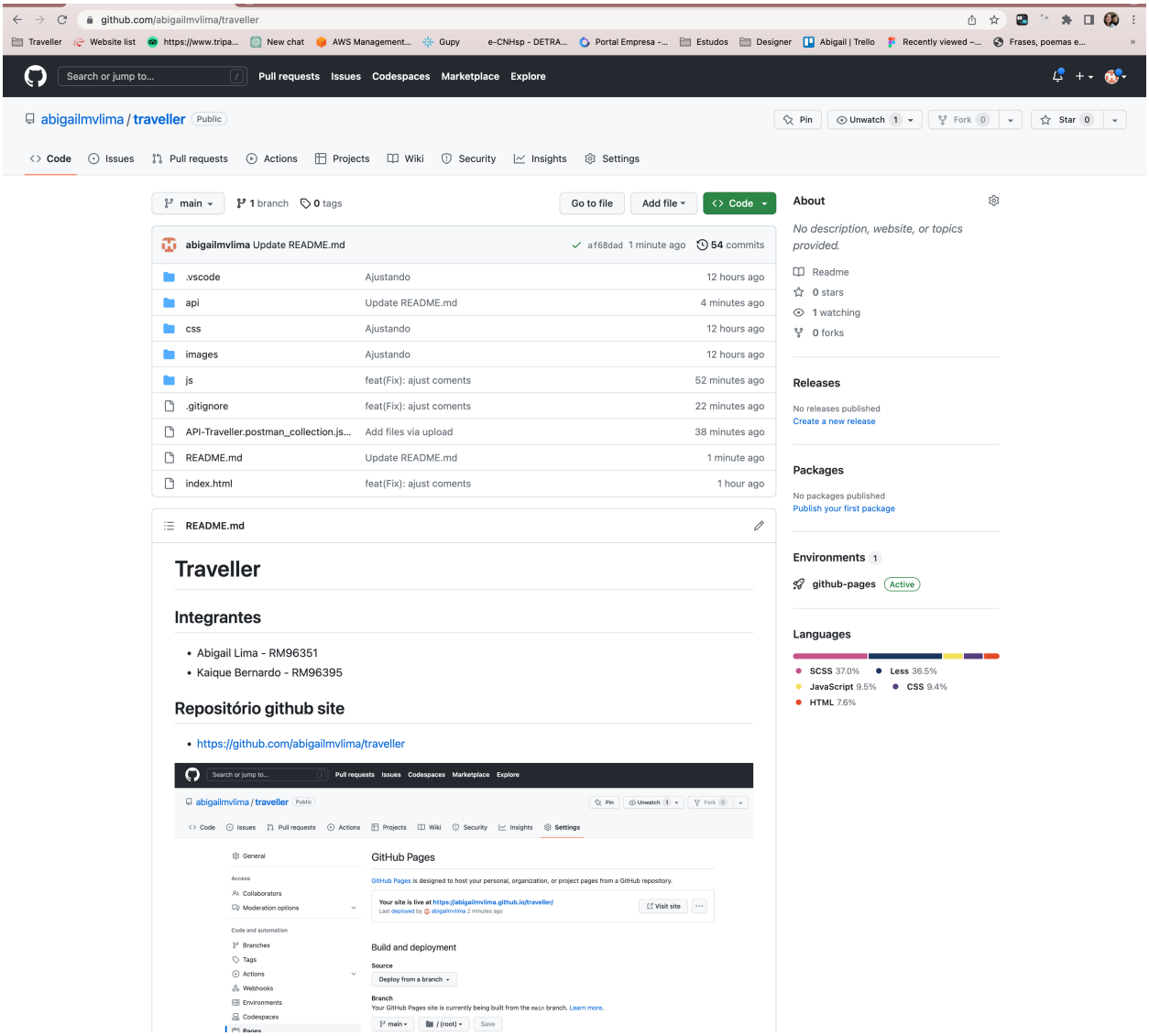

Home
Workspaces
API Network
Explore
Search Postman
Invite
Trial Ends in 14 Days
Traveller
New
Import
GET Consulta
GET Consulta Foto
GET location-reviews
GET Pesquisa Proximidades
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
History
API
GET Consulta
GET Consulta Foto
GET location-reviews
GET Pesquisa Proximidades
GET Detalhes
API / Pesquisa Proximidades
Save
GET
https://p7167519uh.execute-api.us-east-1.amazonaws.com/dev/search/nearby?latLong=-23.53752894777355%2C%20-46.51669981349...
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
latLong
-23.53752894777355%2C%20-46.51669981349255
language
pt
Key
Value
Description
Body
Cookies
Headers (13)
Test Results
Status: 200 OK
Time: 3.99 s
Size: 4.22 KB
Save as Example
Pretty
Raw
Preview
Visualize
JSON
{
"data": [
{
"location_id": "23047222",
"name": "Call Center Burguer",
"distance": "0.3284669829858449",
"bearing": "east",
"address_obj": {
"street1": "Praça Nelson Sales De Abreu, 161",
"street2": "",
"city": "São Paulo",
"state": "Estado de São Paulo",
"country": "Brasil",
"postalcode": "03547-100",
"address_string": "Praça Nelson Sales De Abreu, 161, São Paulo, Estado de São Paulo 03547-100 Brasil"
}
},
{
"location_id": "5330699",
"name": "Estacao Das Pizzas",
"distance": "0.3102157975760826",
"bearing": "east",
"address_obj": {
"street1": "Praca Nelson Sales de Abreu 103",
"street2": "Centro",
"city": "São Paulo",
"state": "Estado de São Paulo",
"country": "Brasil",
"postalcode": "03547-100",
"address_string": "Praca Nelson Sales de Abreu 103 Centro, São Paulo, Estado de São Paulo 03547-100 Brasil"
}
},

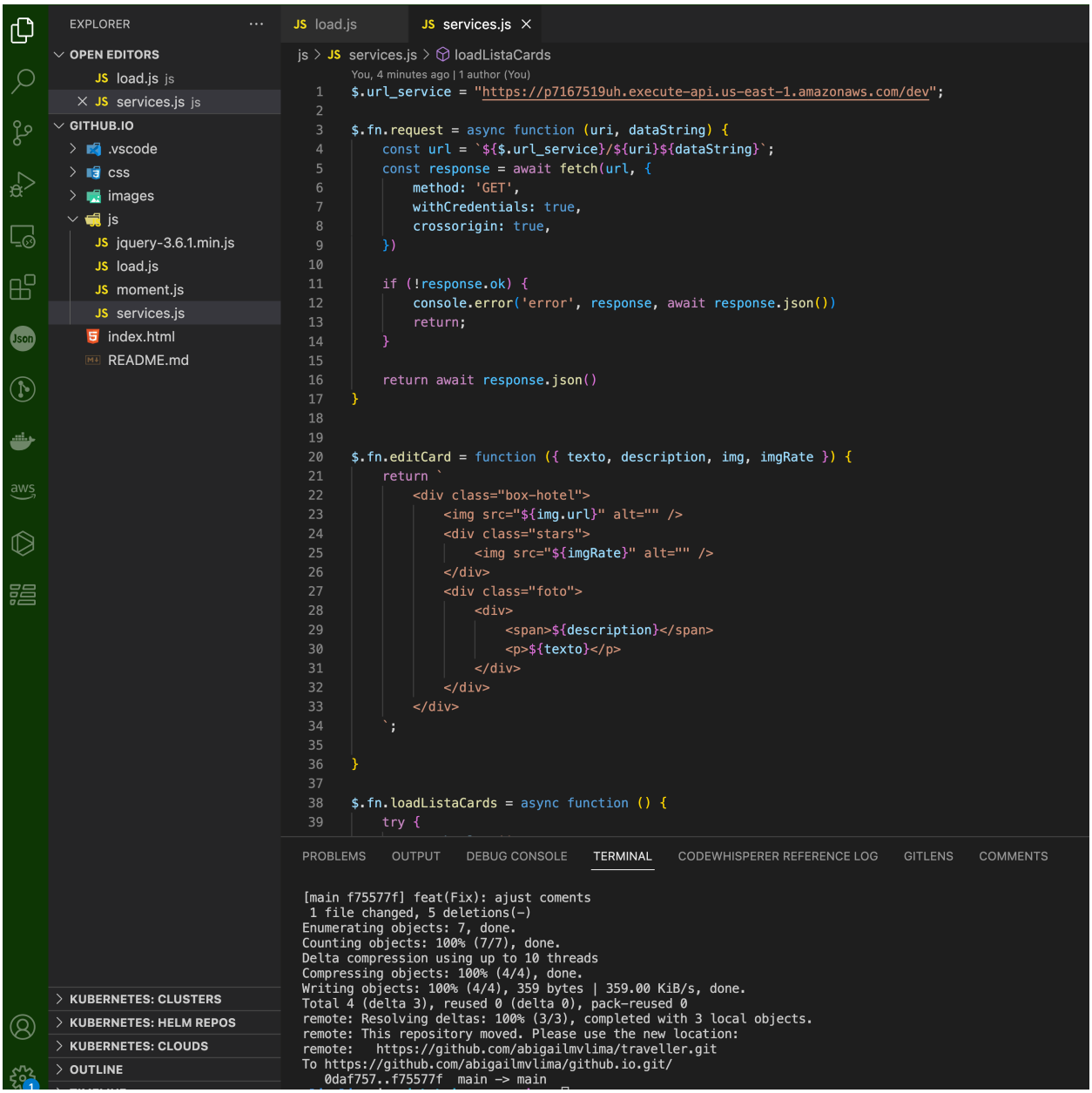


Sobre o codigo do website

https://github.com/abigailmvlima/traveller

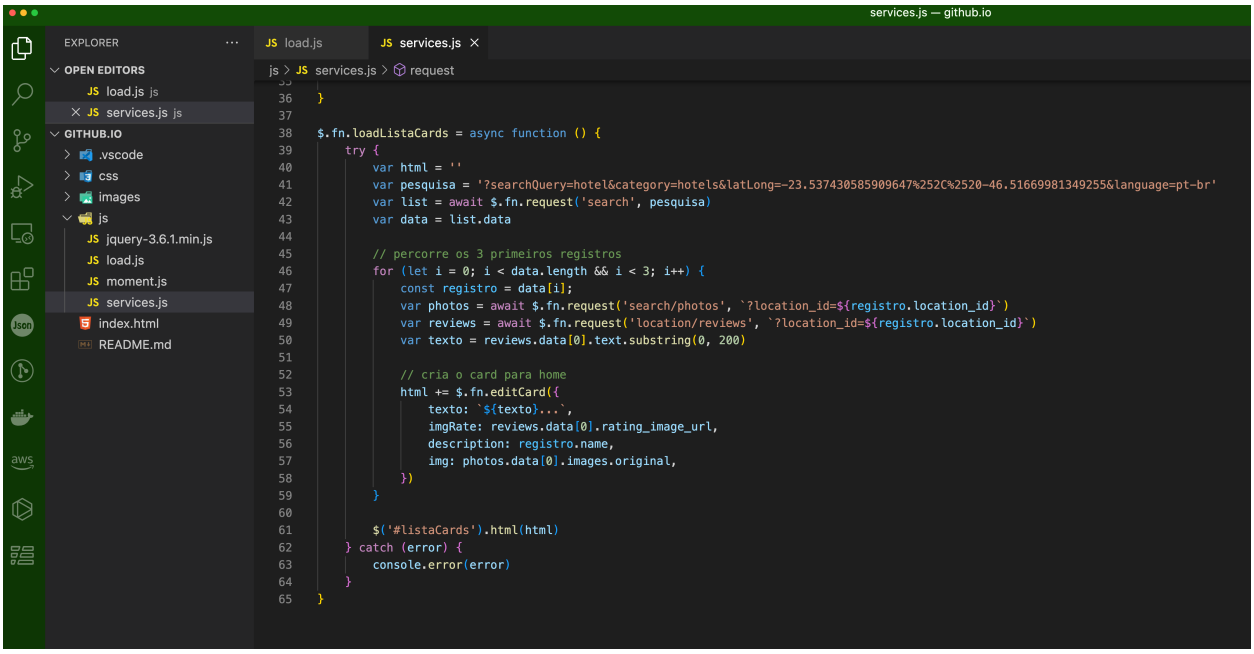
https://github.com/abigailmvlima/traveller


github.com/abigailmvlima/traveller
abigailmvlima / traveller
Public
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
abigailmvlima Update README.md
af68dad 1 minute ago
54 commits
.vscode Ajustando 12 hours ago
api Update README.md 4 minutes ago
css Ajustando 12 hours ago
images Ajustando 12 hours ago
js feat(Fix): ajust coments 52 minutes ago
.gitignore feat(Fix): ajust coments 22 minutes ago
API-Traveller.postman_collection.js... Add files via upload 38 minutes ago
README.md Update README.md 1 minute ago
index.html feat(Fix): ajust coments 1 hour ago
README.md
Traveller
Integrantes
Abigail Lima - RM96351
Kaique Bernardo - RM96395
Repositório github site
https://github.com/abigailmvlima/traveller
GitHub Pages
Build and deployment
About
No description, website, or topics provided.
Readme
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Environments 1
github-pages Active
Languages
SCSS 37.0%
Less 36.5%
JavaScript 9.5%
CSS 9.4%
HTML 7.6%


EXPLORER
OPEN EDITORS
load.js js
services.js js
GITHUB.IO
.vscode
css
images
js
jquery-3.6.1.min.js
load.js
moment.js
services.js
index.html
README.md
KUBERNETES: CLUSTERS
KUBERNETES: HELM REPOS
KUBERNETES: CLOUDS
OUTLINE
load.js
services.js
js > services.js > loadListaCards
You, 4 minutes ago | 1 author (You)
$.url_service = "https://p7167519uh.execute-api.us-east-1.amazonaws.com/dev";
$.fn.request = async function (uri, dataString) {
const url = `${$.url_service}/${uri}${dataString}`;
const response = await fetch(url, {
method: 'GET',
withCredentials: true,
crossorigin: true,
})
if (!response.ok) {
console.error('error', response, await response.json())
return;
}
return await response.json()
}
$.fn.editCard = function ({ texto, description, img, imgRate }) {
return `

${description}
${texto}

`;
}
$.fn.loadListaCards = async function () {
try {
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
CODEWHISPERER REFERENCE LOG
GITLENS
COMMENTS
[main f75577f] feat(Fix): ajust coments
1 file changed, 5 deletions(-)
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 10 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 359 bytes | 359.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
remote: This repository moved. Please use the new location:
remote: https://github.com/abigailmvlima/traveller.git
To https://github.com/abigailmvlima/github.io.git/
0daf757..f75577f main -> main

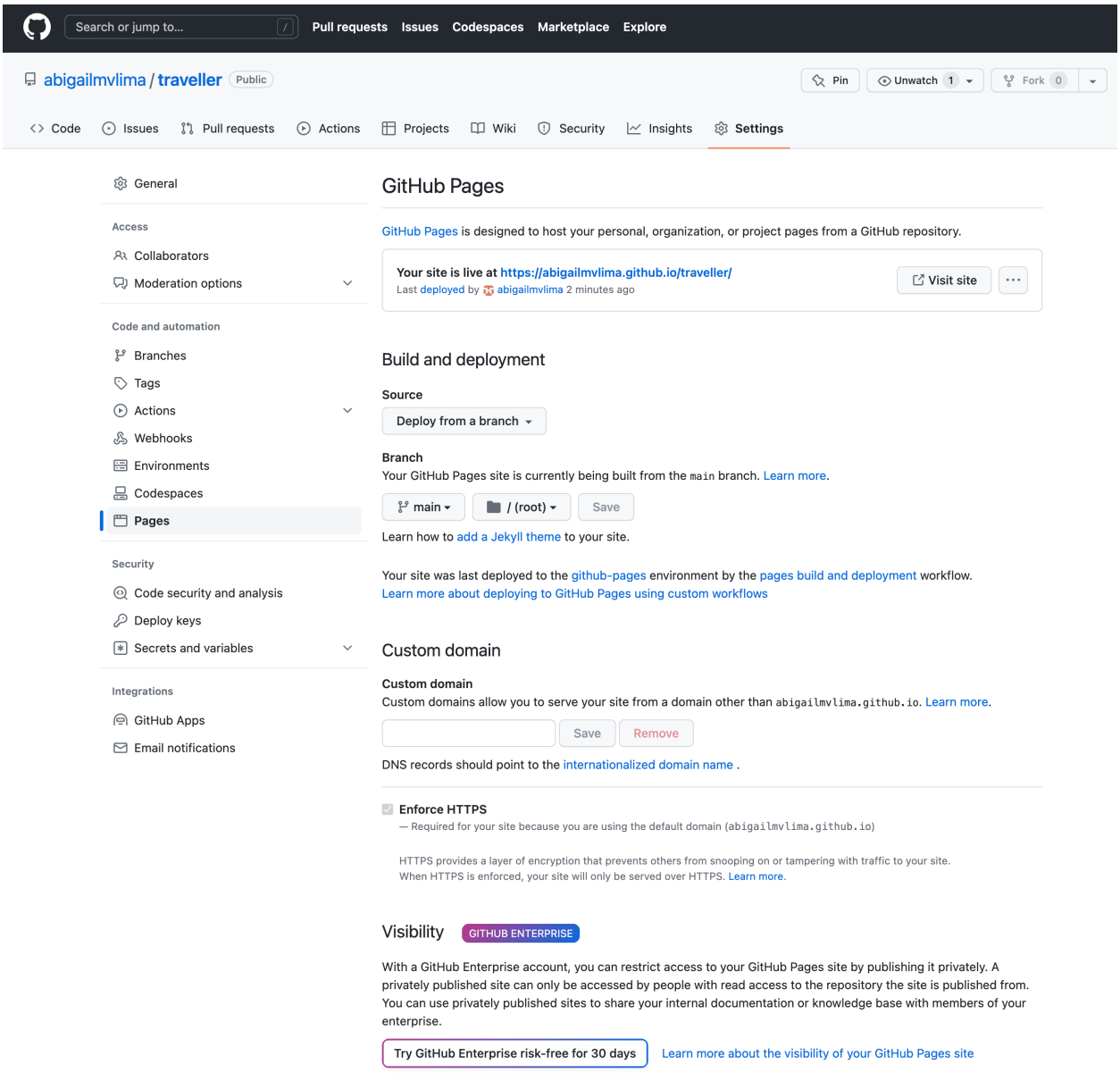


Repositório github site

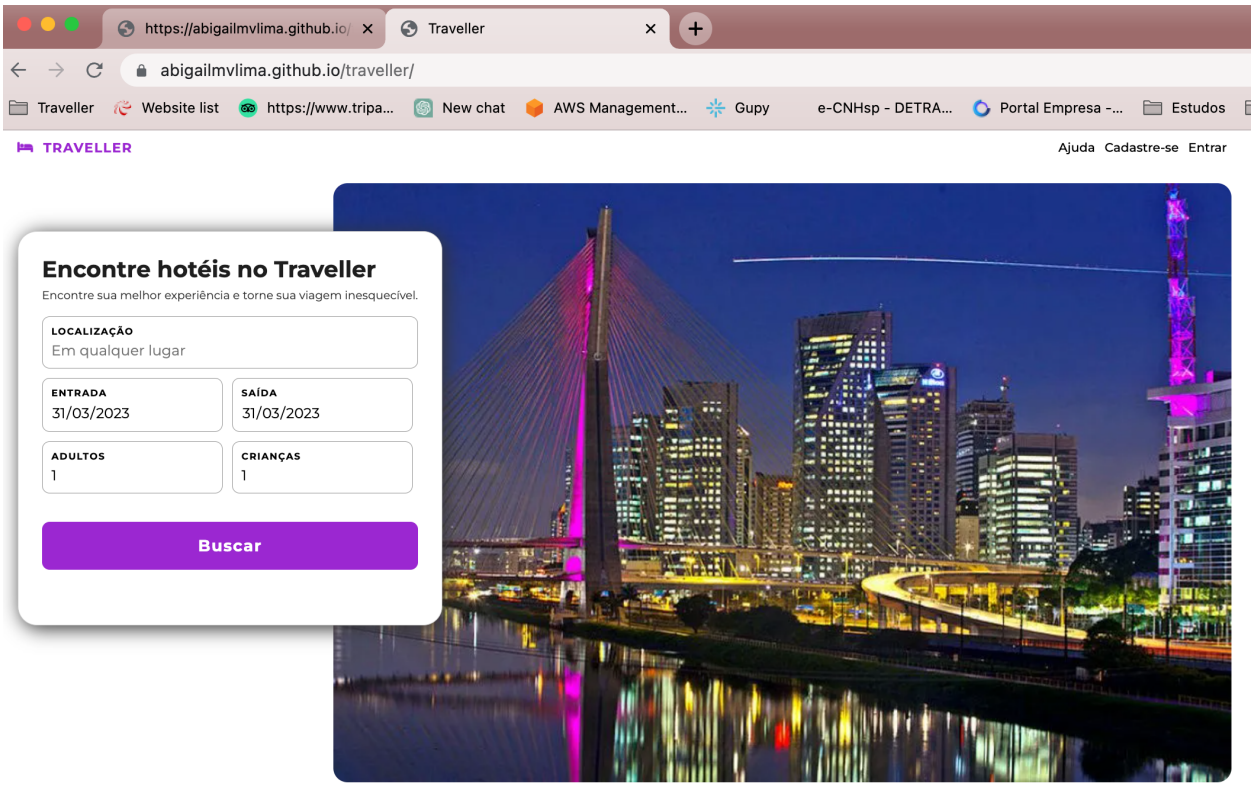
- https://github.com/abigailmvlima/traveller

Publicação site no github.io

- https://abigailmvlima.github.io/gullivertravell/



Site Publicado



O que os hóspedes estão falando sobre as acomodações

Mais de **490.000 avaliações** com uma média de **4.8** de **5 estrelas**

●○○○○

Shangrila Park Hotel

A senha do wifi nao funciona, o ar condicionado é virado para fora, com a janela aberta os pernilongos avancaram para o quarto. Formigas no banheiro e teto. O cafe no primeiro dia tinha suco e no se...

●●●●●

Ceasar Palace Hotel

Eu classifico como ótimo clássico de Vegas, Quarto com espaço para casal, com ante-sala e banheiro espaço... indico com certeza para família, ótimos restaurantes no complexo do hotel de fácil acesso, ...

●○○○○

OYO Hotel Park Leste

Hotel é ruim, não da para família hospedar. Fizemos reserva, quando chegamos, tivemos que ir para outro local, ainda tivemos que pagar taxa de cancelamento. Infelizmente consta no site como hotel.....

Reserve no Traveller e viaje com tranquilidade