

Computação em Nuvem 24/25

Final Report

Grupo 10

Tiago Almeida – 58161

Manuel Campos – 58166

Guilherme Sousa – 58170

Gabriel Henriques – 58182

Motivation, dataset characterization, business capabilities and use cases

Motivation:

Escolhemos o tema “*Stand de Carros Virtual*” porque, dentro do nosso grupo, todos partilhamos um interesse por automóveis e temos planos de adquirir um no futuro. O facto de todos termos o gosto por carros em comum tornou o desenvolvimento do projeto muito mais interessante.

Além disso, considerámos o projeto desafiante do ponto de vista técnico, já que envolve o desenvolvimento de vários microserviços, cada um com responsabilidades distintas.

Acreditamos que este projeto nos permite aplicar conhecimentos adquiridos ao longo do curso, como orquestração de serviços, tolerância a falhas, escalabilidade, e comunicação assíncrona, num contexto que é simultaneamente prático e alinhado com os nossos interesses pessoais.

Dataset Characterization:

O dataset escolhido contém praticamente todas as informações relevantes que o Craigslist fornece sobre vendas de carros, incluindo colunas como preço, estado (condição), fabricante, latitude/longitude e outras 18 categorias.

Para o nosso projeto decidimos retirar grande parte das categorias do dataset, isto porque achámos que não eram úteis para o nosso projeto e estariam só a ocupar espaço, um exemplo óbvio de uma categoria considerada “inútil” foi a descrição de cada anúncio, metade dos anúncios tinham todos a mesma descrição e ocupava muito espaço.

URL do Dataset:

<https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data?resource=download>

Business Capabilities:

1. Gestão de Anúncios de Carros
2. Venda de Carros e Transações
3. Agendamento de Reuniões
4. Manutenção de Carros
5. Inspeção Periódica de Carros
6. Anúncios em Destaque

Inicialmente eram estas as *business capabilities* que tínhamos em mente para aplicar no nosso projeto, conseguimos aplicar todas menos a última.

Use cases:

1. Comprar um carro
2. Alugar um carro
3. Devolver um carro alugado
4. Explorar os anúncios de carros
5. Marcar uma reunião com o staff
6. Ver detalhes dos carros

7. Adicionar um anúncio de um carro
8. Alterar um anúncio
9. Remover um anúncio
10. Adicionar um anúncio aos anúncios em destaque
11. Ver o histórico de anúncios
12. Ver o histórico de vendas e alugueres
13. Ver relatórios de inspeções e manutenções

Tal como as *business capabilities*, estes eram os *use cases* que tínhamos planeado para o nosso projeto, não foi possível fazer os últimos 4 pontos devido à complexidade dos mesmos e do tempo que tínhamos para o projeto.

API Specification

CarStandFCOOL API 1.0.0 OAS 3.0

An API for managing a cloud-based car dealership service.

IMPORTANT: All endpoints in this API must be prefixed with '/api'. For example:

- '/cars' should be accessed as '/api/cars'
- '/users' should be accessed as '/api/users'
- '/meetings' should be accessed as '/api/meetings' And so on for all endpoints.

Servers

/api - Base URL for all endpoints. All paths in this document must be prefixed with this URL.



Car

GET

/cars

Read the list of cars

Try it out

Parameters

No parameters

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | Successfully read cars list <div>Media type application/json</div> <div>Controls Accept header</div> <div>Example Value Schema</div> <pre>{ "carId": 123, "year": 2000, "manufacturer": "Ford", "model": "F-150", "condition": "excellent", "cylinders": "6 cylinders", "fuel": "gas", "odometer": 50000, "transmission": "Automatic", "VIN": "1F0HUL1T1HEAS2362", "drive": "4WD", "size": "compact", "type": "sedan", "paint_color": "white" }</pre> | No links |
| 500 | Server error | No links |

POST

/cars

Create a car

GET

/cars/{carId}

Read one car

PUT

/cars/{carId}

Update a car

DELETE

/cars/{carId}

Delete a car

User

GET

/users

Read the list of users

Try it out

Parameters

No parameters

Responses

| Code | Description | Links |
|------|---|----------|
| 200 | Successfully read user list <div>Media type application/json</div> <div>Controls Accept header</div> <div>Example Value Schema</div> <pre>{ "userId": 123, "firstName": "John", "lastName": "Doe", "email": "john.doe@example.com" }</pre> | No links |
| 500 | Server error | No links |

POST

/users

Create a user

GET

/users/{userId}

Read one user

PUT

/users/{userId}

Update a user

DELETE

/users/{userId}

Delete a user

Maintenance

GET

/maintenances

Read the list of your maintenances

Parameters

No parameters

Try it out

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | Successfully read maintenance list <div><div>Media type</div><div>application/json</div><div>Controls Accept header</div><div>Example Value Schema</div></div> <pre>{ "maintenanceId": 0, "maintenanceCardId": 0, "maintenanceType": "unknown", "maintenanceStatus": "unknown", "maintenanceClientNotes": "string", "maintenanceStaffNotes": "string", "maintenanceCost": 0, "maintenanceStartDate": "2025-06-11T22:11:04.176Z", "maintenanceEndDate": "2025-06-11T22:11:04.176Z"}</pre> | No links |
| 400 | Bad request | No links |
| 401 | Unauthorized | No links |
| 403 | Forbidden | No links |
| 500 | Internal Server Error | No links |

POST

/maintenances

Create a maintenance

GET

/maintenances/{maintenanceId}

Read one maintenance

PUT

/maintenances/{maintenanceId}

Update a maintenance

DELETE

/maintenances/{maintenanceId}

Delete a maintenance

Inspection

GET

/inspections

Read the list of your inspections

Parameters

No parameters

Try it out

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | Successfully read inspection list <div><div>Media type</div><div>application/json</div><div>Controls Accept header</div><div>Example Value Schema</div></div> <pre>{ "inspectionId": 0, "inspectionCardId": 0, "inspectionStatus": "unknown", "inspectionClientNotes": "string", "inspectionStaffNotes": "string", "inspectionCost": 0, "inspectionStartDate": "2025-06-11T22:11:21.846Z", "inspectionEndDate": "2025-06-11T22:11:21.846Z"}</pre> | No links |
| 400 | Bad request | No links |
| 401 | Unauthorized | No links |
| 403 | Forbidden | No links |
| 500 | Internal Server Error | No links |

POST

/inspections

Create an inspection

GET

/inspections/{inspectionId}

Read one inspection

PUT

/inspections/{inspectionId}

Update an inspection

DELETE

/inspections/{inspectionId}

Delete an inspection

Transaction

GET

/transactions

Read the list of transactions

Try it out

Parameters

No parameters

Responses

| Code | Description | Links |
|------|---|----------|
| 200 | Successfully read transactions list <div>Media type application/json</div> <div>Content Accept Header</div> <div>Example Value Schema</div> <pre>[{ "transactionId": 1001, "buyerId": 123, "carId": 456, "type": "Buy", "totalAmount": 25000, "status": "Completed", "transactionDate": "2024-03-08T12:30:00Z", "endDate": "2024-05-08T12:30:00Z" }]</pre> | No links |
| 500 | Server error | No links |

POST /transactions Create a transaction

GET /transactions/{transactionId} Read one transaction

PUT /transactions/{transactionId} Update a transaction

DELETE /transactions/{transactionId} Delete a transaction

Car Listing

GET

/carlistings

Read the list of car listings

Try it out

Parameters

No parameters

Responses

| Code | Description | Links |
|------|---|----------|
| 200 | Successfully read car listing list <div>Media type application/json</div> <div>Content Accept Header</div> <div>Example Value Schema</div> <pre>[{ "listingId": 10, "carId": 10, "userId": 10, "type": "Unknown", "description": "2025-06-11T22:12:16.463Z", "posting_date": "2025-06-11T22:12:16.463Z", "sale_price": 0, "promoted": true, "status": "Unknown" }]</pre> | No links |
| 500 | Server error | No links |

POST /carlistings Create a car listing

GET /carlistings/{listingId} Read one car listing

PUT /carlistings/{listingId} Update a car listing

DELETE /carlistings/{listingId} Delete a car listing

Meeting

GET

/meetings

Read the list of meetings

Try it out

Parameters

No parameters

Responses

| Code | Description | Links |
|--|--------------------------------|----------|
| 200 | Successfully read meeting list | No links |
| Media type: application/json | | |
| Example Value Schema | | |
| <pre>{ "meetingId": 2001, "clientId": 123, "scheduleDate": "2024-04-15T14:00:00Z", "status": "Scheduled" }</pre> | | |
| 500 | Server error | No links |

POST /meetings Create a meeting

GET /meetings/{meetingId} Read one meeting

PUT /meetings/{meetingId} Update a meeting

DELETE /meetings/{meetingId} Delete a meeting

Schemas

Car {

carId > [...]

year > [...]

manufacturer > [...]

model > [...]

condition > [...]

cylinders > [...]

fuel > [...]

odometer > [...]

transmission > [...]

VIN > [...]

drive > [...]

size > [...]

type > [...]

paint_color > [...]

}

User {

userId > [...]

firstName > [...]

lastName > [...]

email > [...]

}

Maintenance {

maintenanceId > [...]

maintenanceCard > [...]

maintenanceType > [...]

maintenanceStatus > [...]

maintenanceClientNotes > [...]

maintenanceStaffNotes > [...]

maintenanceCost > [...]

maintenanceStartDate > [...]

maintenanceEndDate > [...]

}

Inspection {

inspectionId > [...]

inspectionCardId > [...]

inspectionStatus > [...]

inspectionClientNotes > [...]

inspectionStaffNotes > [...]

inspectionCost > [...]

inspectionStartDate > [...]

inspectionEndDate > [...]

}

```
Transaction {
  transactionId* > [...]
  buyerId* > [...]
  carId* > [...]
  type* > [...]
  totalAmount* > [...]
  status* > [...]
  transactionDate* > [...]
  endDate > [...]
}
```

```
CarListing {
  listingId* > [...]
  carId* > [...]
  userId* > [...]
  type* > [...]
  description* > [...]
  posting_date* > [...]
  sale_price* > [...]
  promoted* > [...]
  status* > [...]
}
```

```
Meeting {
  meetingId* > [...]
  clientId* > [...]
  scheduleDate* > [...]
  status* > [...]
}
```

Functional and Non-Functional Requirements

1. Functional Requirements:

1.1. Gestão de anúncios de carros:

1.1.1. Adicionar um anúncio de um carro

- Os utilizadores podem adicionar um anúncio de carro fornecendo os detalhes do carro (fabricante, modelo, ano, estado, etc.) e os detalhes do anúncio (preço de venda, descrição).
- O sistema deverá validar os campos obrigatórios e rejeitar submissões incompletas.
- O anúncio deverá ser atribuído um “listingId” único

1.1.2. Alterar um anúncio de um carro

- Os utilizadores podem alterar os seus anúncios de carro para modificar detalhes como o preço, a descrição ou o estado.

- O sistema deverá validar todas as atualizações antes de aplicar as alterações.

1.1.3. Remover um anúncio de um carro

- Os utilizadores podem remover os seus anúncios de carro quando desejarem.
- Os anúncios removidos deixaram de estar visíveis para os compradores.
- O sistema não deverá reter registos históricos dos anúncios removidos.

1.1.4. Ver anúncios de carros

- Os utilizadores podem navegar pelos carros disponíveis através de uma lista paginada e filtrável.
- O sistema deverá permitir a filtragem por detalhes do carro, como fabricante, modelo, ano, estado, etc.
- Cada item da lista deverá apresentar uma descrição, o preço de venda, a data de publicação e um link para a página completa do anúncio do carro.

1.1.5. Ver um anúncio de um carro específico

- Os utilizadores podem clicar num anúncio para ver todos os detalhes do carro, informações do vendedor (utilizador) e iniciar o processo de compra do carro.

1.2. Venda de Carros e Transações:

1.2.1. Comprar um carro

- Os utilizadores podem iniciar a compra de um carro ao selecionar um anúncio e proceder com a transação.
- O sistema deverá processar os pagamentos e atualizar o estado do anúncio para "vendido" após a conclusão.

1.2.2. Alugar um carro

- Os utilizadores podem alugar um carro ao selecionar um anúncio e definir um período de aluguer.
- O sistema deverá verificar a disponibilidade para aluguer antes de confirmar a transação.
- O sistema deverá processar os pagamentos e atualizar o estado do anúncio para "reservado" após a conclusão.

1.2.3. Devolver um carro alugado

- Os utilizadores que disponibilizarem carros para aluguer podem marcar um carro como devolvido no final do período de aluguer.
- O sistema deverá verificar o estado da devolução e atualizar o anúncio em conformidade.

1.2.4. Ver o histórico de transações

- Os utilizadores podem visualizar as suas transações passadas, incluindo detalhes de compras, alugueres e devoluções.
- O sistema deverá fornecer recibos e atualizações sobre o estado das transações.

1.3. Manutenção de Carros:

1.3.1. Marcar uma manutenção

- Os proprietários de carros podem agendar serviços de manutenção (básica, completa) e adicionar notas prévias ao agendamento.
- O sistema deverá armazenar os pedidos de manutenção juntamente com os detalhes do carro, definindo o estado da manutenção como "Em curso".

1.3.2. Ver o histórico de manutenção

- Os utilizadores podem consultar os registos de manutenção anteriores de um carro.
- O sistema deverá armazenar detalhes como a data do serviço, tipo, custo e notas do mecânico.

1.3.3. Pós-manutenção

- Quando a manutenção do carro termina, um membro do staff adiciona notas, o custo da manutenção e define o estado como "Concluída".
- O sistema deverá armazenar os detalhes adicionados pelo staff e registar a data de conclusão.

1.4. Inspeção Periódica de Carros:

1.4.1. Marcar uma inspeção

- Os utilizadores podem agendar uma inspeção periódica para os seus carros.

1.4.2. Ver o relatório da inspeção

- Os utilizadores podem consultar os relatórios de inspeções anteriores de um carro, incluindo problemas e recomendações.

1.4.3. Pós-Inspeção

- Quando a inspeção do carro termina, um membro do staff adiciona notas, o custo da inspeção e define o estado como "Concluída"
- O sistema deverá armazenar os detalhes adicionados pelo staff e registar a data de conclusão.

1.5. Reunião com Staff:

1.5.1. Marcar uma reunião

- Os utilizadores podem agendar uma reunião com o proprietário do carro no respetivo anúncio.
- O sistema deverá permitir a escolha de um intervalo de tempo disponível.

1.5.2. Gerir reuniões

- Os utilizadores podem alterar a data ou cancelar reuniões.
- O sistema deverá registar o estado da reunião (agendada, concluída, cancelada).

2. Non-Functional Requirements:

2.1. Autenticação e Autorização:

2.1.1. Integração com o Auth0

- **Objetivo:** Fornece autenticação segura para os utilizadores e gestão da identidade dos mesmos.
- **Implementação:**
 - Utiliza o Auth0 como fornecedor de identidade

- Controla o registo de utilizadores, o início de sessão e a gestão de sessões
- Protege os endpoints da API através de autenticação baseada em tokens
- **Configuração:**
 - Necessita de variáveis de ambiente:
 - AUTH0_DOMAIN
 - AUTH0_CLIENT_ID
 - AUTH0_CLIENT_SECRET
 - AUTH0_AUDIENCE
 - AUTH0_CALLBACK_URL

2.1.2. Role-Based Access Control (RBAC)

- **Objetivo:** Controla o acesso a diferentes funcionalidades do sistema com base nas funções dos utilizadores.
- **Implementação:**
 - Define diferentes funções de utilizador (utilizadores e administradores)
 - Restringe o acesso a operações sensíveis
 - Gere permissões para:
 - Gestão de anúncios de carros
 - Marcação de manutenções
 - Gestão de inspeções
 - Marcação de reuniões

2.2. Continuous Integration and Deployment (CI/CD):

2.2.1. GitHub Actions

- **Objetivo:** Automatiza os processos de teste e de implementação.
- **Componentes:**
 - test.yml: Corre testes automáticos
 - build-push.yml: Gere a criação e o envio das imagens Docker
- **Funcionalidades:**
 - Testes automatizados
 - Construção de imagens Docker
 - Envio das imagens para o repositório de containers

2.3. Monitorização e Observabilidade

2.3.1. Prometheus

- **Objetivo:** Recolha e armazenamento de métricas.
- **Funcionalidades:**
 - Recolhe métricas de todos os microsserviços
 - Armazena dados em séries temporais
 - Disponibiliza capacidades de consulta
- **Acesso:** Disponível via port-fowarding (porta 9090)

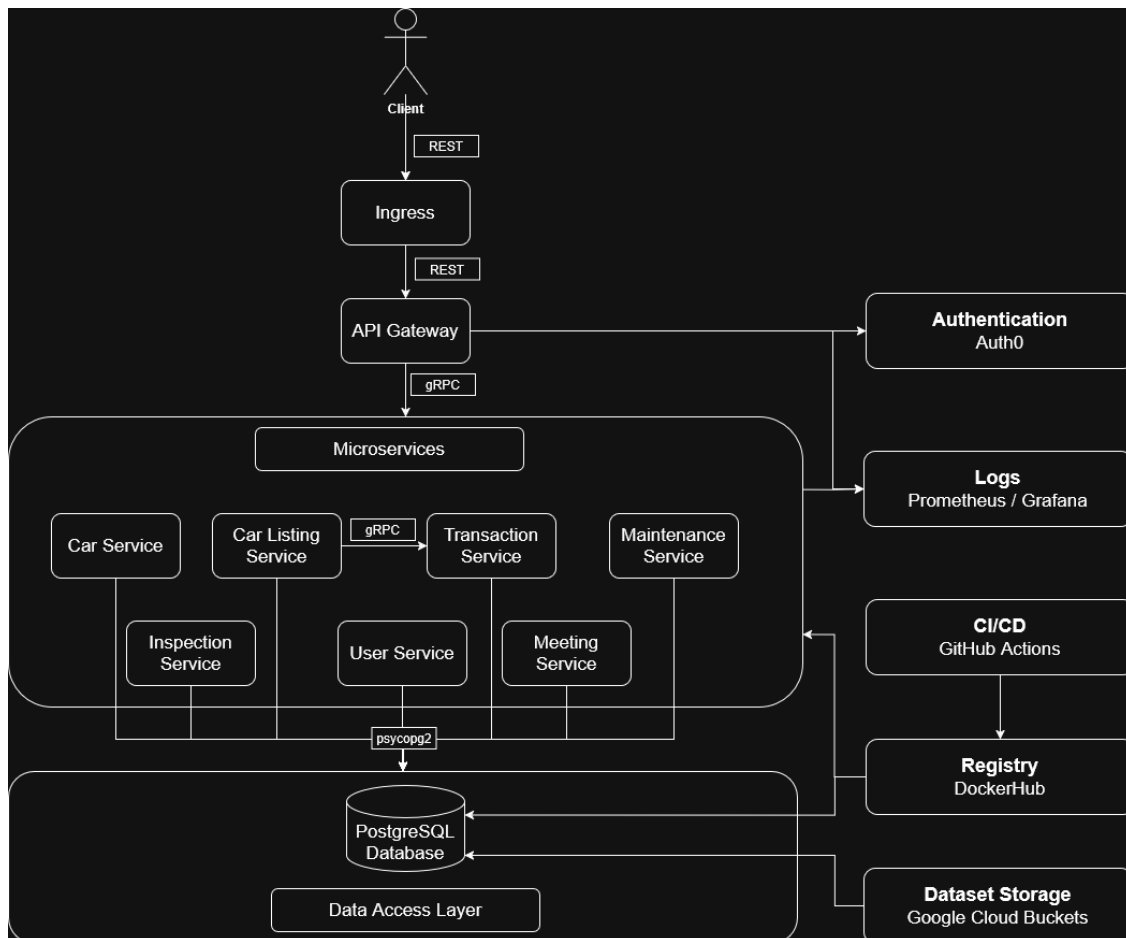
2.3.2. Grafana

- **Objetivo:** Visualização de métricas e criação de dashboards
- **Funcionalidades:**
 - Dashboards personalizadas para diferentes métricas
 - Monitorização em tempo real
 - Configuração de alertas
- **Acesso:** Disponível via port-fowarding (porta 3000)

Architecture (Application and Technical)

Descrição da Arquitetura:

O sistema segue uma arquitetura de microsserviços, sendo que cada serviço gere um domínio específico. Os microsserviços comunicam entre si utilizando o gRPC, enquanto os clientes externos interagem com o sistema através de um API Gateway baseado em REST.



Gateway API:

- Atua como o único ponto de entrada para todos os pedidos externos, expondo uma API REST.
- Recebe pedidos HTTP REST e encaminha-os para os microsserviços apropriados via gRPC.

- Lida com a autenticação e a validação de pedidos utilizando o Auth0.
- Inclui a recolha de métricas do Prometheus para monitorização.
- Fornece uma interface web com templates para autenticação de utilizadores e acesso ao painel.

User Service:

- Gere as informações de Users e armazena os dados na base de dados PostgreSQL partilhada.
- Fornece operações CRUD para entidades Users.

Car Service:

- Gere as informações de Cars e armazena os dados na base de dados PostgreSQL partilhada.
- Fornece operações CRUD para entidades Car.

Car Listing Service:

- Gere informações de Car Listings e armazena os dados na base de dados PostgreSQL partilhada.
- Fornece operações CRUD para entidades de Car Listing.
- Interage com o Transaction Service para atualizações de estado após a conclusão da venda/arrendamento.

Transaction Service:

- Gere Transactions de compra e aluguer e armazena os dados na base de dados PostgreSQL partilhada.
- Fornece operações CRUD para entidades de Transaction.

Maintenance Service:

- Gere agendamentos de Maintenances e armazena os dados na base de dados PostgreSQL partilhada.
- Fornece operações CRUD às entidades de Maintenance.

Inspection Service:

- Gere Inspections periódicas de automóveis e armazena os dados na base de dados PostgreSQL partilhada.
- Fornece operações CRUD para entidades de Inspection.

Meeting Service:

- Gere Meetings entre utilizadores e funcionários do stand e armazena os dados na base de dados PostgreSQL partilhada.
- Fornece operações CRUD para entidades de Meeting.

Camada de Base de Dados:

- Base de dados PostgreSQL única partilhada entre todos os microserviços.
- Contém tabelas para cars, users, car_listings, transactions, maintenance, inspection, and meetings.
- Garante a integridade referencial através de restrições de foreign key.
- Preenchido com dados de amostra para testes e desenvolvimento.
- Os detalhes de ligação à base de dados são configurados através de variáveis de ambiente.

Monitorização e Observabilidade:

- Todos os serviços incluem a recolha de métricas do Prometheus para monitorização.
- Os dashboards do Grafana estão disponíveis para visualização de métricas.
- As métricas incluem contagens de pedidos, histogramas de latência, pedidos ativos e tempo de operação da base de dados.

Autenticação e Autorização:

- Integração com Auth0 para autenticação de utilizadores.
- Controlo de acessos baseado em funções com permissões para diferentes operações.
- Tokens JWT utilizados para autenticação de API.
- Gestão de sessão para interface web.

CI/CD e Delivery:

- Os workflows do GitHub Actions automatizam a pipeline de integração e delivery contínua.
- Compilações, testes e entregas automatizadas são acionadas por commits na main branch
- As Docker images são criadas para cada microsserviço e pushed para o DockerHub.

Implementation (High Level Description)

O sistema foi implementado como uma cloud native application com arquitetura de microsserviços, incluindo uma infraestrutura completa de monitorização, segurança e deployment.

1. Infraestrutura e Deployment

1.1. Containers (Kubernetes Deployment):

- Cada serviço deployed num pod isolado.
- *Service Mesh* para comunicação entre serviços.
- *Persistent Volume Claims* para armazenamento de dados.

1.2. Infraestrutura da Base de Dados:

- *PostgreSQL* Container.
- *Docker* image dedicada.
- *Persistent Volume* para armazenamento de dados.
- Environment variables para configuração.
- Serviço dedicado para acesso.

1.3. Load Balancing e Escalabilidade:

- **NGINX Ingress Controller:**
 - Uso de *Cookies* para gestão de sessões.
 - Health check path com */health*.
 - Suporte de protocolo *HTTP/1.1*.
- **Horizontal Pod Autoscaling:**
 - Scaling dinâmico com 2-3 réplicas por serviço.
 - *CPU* e Memory-Based scaling (80% de threshold).
 - *Rolling update* strategy.
 - Limites de Recursos e Requests.
- **Load Balancing de serviços:**
 - Serviços *ClusterIP* para comunicação interna.
 - *Round-robin* load balancing.

2. Monitorização

2.1. Prometheus Setup:

- *ConfigMap* para configuração do *Prometheus*.
- Armazenamento persistente para metrics.
- Recolha de metrics de todos os microserviços.

2.2. Grafana Integration:

- Uso de secrets para environment variables.
- Configurações de Segurança *CSRF*.
- Dashboards próprias para monitoring.

2.3. Health Monitoring e Logging:

- **Health Checks:**
 - Liveness probes.
 - Readiness checks.
 - Automatic recovery.
- **Monitorização de Serviços:**
 - Health checks automáticos.
 - Recovery automática.
 - Alertas relacionados com health checks.

3. Security Implementation

3.1. Auth0 Integration:

- *OAuth 2.0* implementation.
- *JWT* token validation.
- Session management.
- *Callback URL* handling.

3.2. Role-Based Access Control:

- Uso de *JWT* para verificação de permissões.
- Nomes customizados e adequados para permissões.
- Validação de permissões.
- Error handling para acessos não autorizados.

4. Service Architecture

4.1. Core Services com ports próprias:

- **API Gateway:**
 - **Implementação da REST API:**
 - Uso de *Flask* para endpoints *REST*.
 - *CRUD* operations para todos os serviços.
 - Uso de *RBAC*.
 - Gestão de requests / responses.

- **Endpoints de Serviços:**
 - Cars API (/api/cars).
 - Users API (/api/users).
 - Maintenance API (/api/maintenances).
 - Inspection API (/api/inspections).
 - Transaction API (/api/transactions).
 - Car Listing API (/api/carlistings).
 - Meeting API (/api/meetings).
- **Endpoints de Autenticação:**
 - Login (/login).
 - Callback (/callback).
 - Dashboard (/dashboard).
 - Logout (/logout).
 - Health Check (/health).
- **Microserviços:**
 - Car Service.
 - Car Listing Service.
 - User Service.
 - Transaction Service.
 - Maintenance Service.
 - Inspection Service.
 - Meeting Service.

4.2. Comunicação entre serviços:

- *REST API* para comunicação externa.
- *gRPC* para comunicação entre microserviços.
- *Protocol Buffers* para serialização.

5. Gestão de Dados

5.1. Database Schema:

- **car:** Informação de veículos.
- **users:** Gestão de utilizadores.
- **meeting:** Marcação de reuniões.
- **maintenance:** Gestão de manutenções.
- **inspection:** Registos de inspeções.
- **car_listing:** Gestão de listings.
- **transaction:** Histórico de transações.

5.2. Persistência de Dados:

- *Persistent Volume Claims.*
- Estratégias de Backup da base de dados.

6. Desenvolvimento e Operações

6.1. Container Management (Docker):

- Imagens *Docker*.
- Build e Push de imagens.
- Variáveis de Ambiente para configuração.
- Volume management.

6.2. Configurações Kubernetes:

- **Deployment Strategy:**
 - Rolling updates.
- **Service Configuration:**
 - Comunicação interna entre serviços.
 - Comunicação externa pelo Ingress.

Esta implementação resulta num sistema robusto com:

- Monitorização com *Prometheus* e *Grafana*.
- Autenticação Segura com *Auth0*.
- Deployment escalável com containers.
- Armazenamento de dados persistente.
- Pipeline Automatizada de Deployment.
- Role-based access control.
- Load Balancing e Gestão de Sessões.
- Scaling automático com base na utilização de recursos.

O sistema foi implementado para ser:

- Altamente disponível (*Kubernetes* deployment).
- Seguro (*Auth0* integration).
- Observável (*Prometheus* / *Grafana*).
- Escalável (Containerized microservices).
- Fácil de Manter (Separação de responsabilidades clara).

Deployment Scripts

O processo de deployment é baseado em imagens Docker publicadas no Docker Hub, geradas pela pipeline CI/CD. Estas imagens são utilizadas nos manifestos de Kubernetes para deployment dos microserviços e da base de dados. No entanto, não temos scripts de deployment totalmente automáticos. Em vez disso, utilizamos um script chamado ``apply_kubernetes_manifests.sh``, que, ao ser executado, realiza o deploy para o Google Kubernetes Engine (GKE).

Devido à necessidade de obter o IP do Ingress para configuração no Auth0, optámos por um processo de deployment manual utilizando este script.

CI/CD Pipeline

Para automatizar o ciclo de desenvolvimento contínuo do projeto, foram criados dois workflows no GitHub Actions, localizados na pasta `.github/workflows`:

1. Build and Push Docker Images (`build-push.yml`):

Este workflow é ativado automaticamente em commits no branch `main` quando há alterações nas pastas `databases/` ou `microservices/`. O pipeline executa os seguintes passos:

- Faz checkout do repositório.
- Configura o Docker Buildx.
- Faz autenticação no Docker Hub utilizando `secrets` configurados no GitHub (`DOCKER_USERNAME`, `DOCKER_TOKEN`).
- Constrói e faz push da imagem Docker do serviço de base de dados.
- Identifica dinamicamente os microsserviços definidos no `docker-compose.yml`, constrói as respectivas imagens e faz push no Docker Hub com a tag `latest`.

Com isto, todas as alterações feitas aos principais componentes (bases de dados e microsserviços) são automaticamente integradas e disponibilizadas como imagens no Docker Hub.

2. Test Workflow (`test.yml`):

Este workflow é ativado em commits e pull requests para a branch `main`. Executa testes automáticos para garantir que o código continua funcional após alterações:

- Faz checkout do código.
- Instala todas as dependências necessárias.
- Gera automaticamente o código *gRPC* a partir dos `.proto`.
- Executa os testes com *pytest*, localizados na pasta `tests/`.

Este pipeline permite manter a integridade funcional do sistema ao longo do tempo e facilita a detecção de erros.

Test and Evaluation Techniques (To Group 13)

Avaliação da documentação:

A documentação fornecida pelo outro grupo estava clara, bem estruturada e precisa.

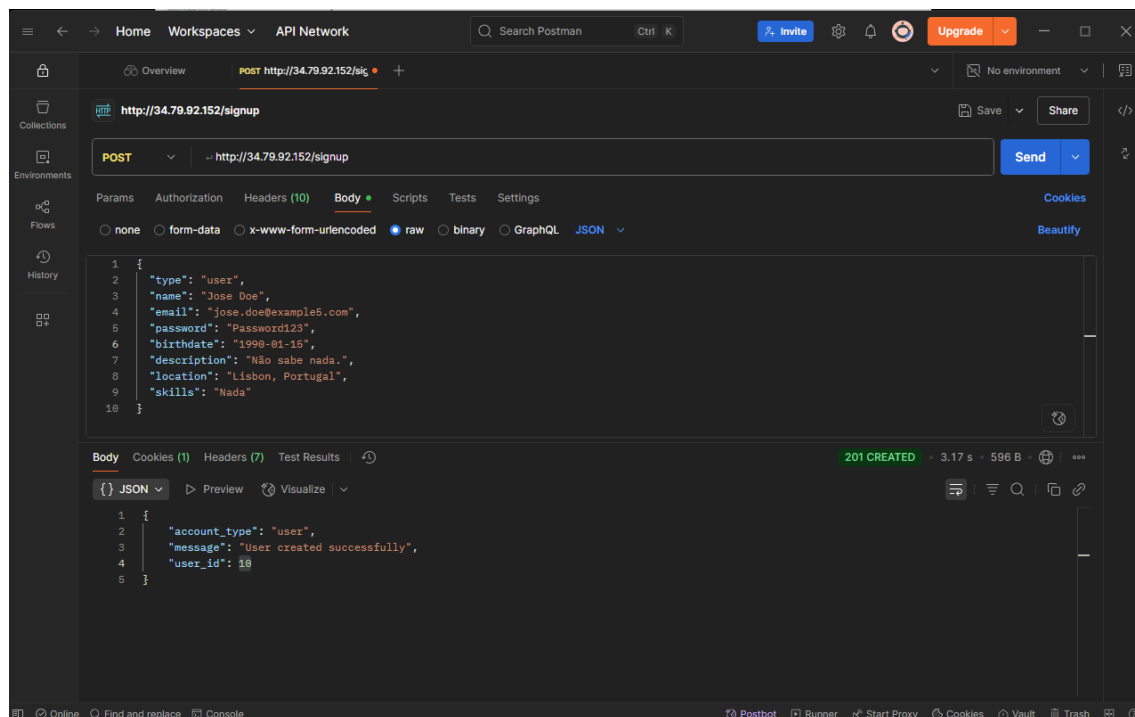
Incluía uma lista abrangente de todos os endpoints disponíveis, cada um com explicações detalhadas sobre a sua funcionalidade.

A inclusão do URL base e de uma especificação OpenAPI tornou fácil compreender como interagir com a API e testar cada endpoint de forma eficiente.

No geral, a documentação facilitou os testes e demonstrou uma forte atenção ao detalhe.

Smoke Tests:

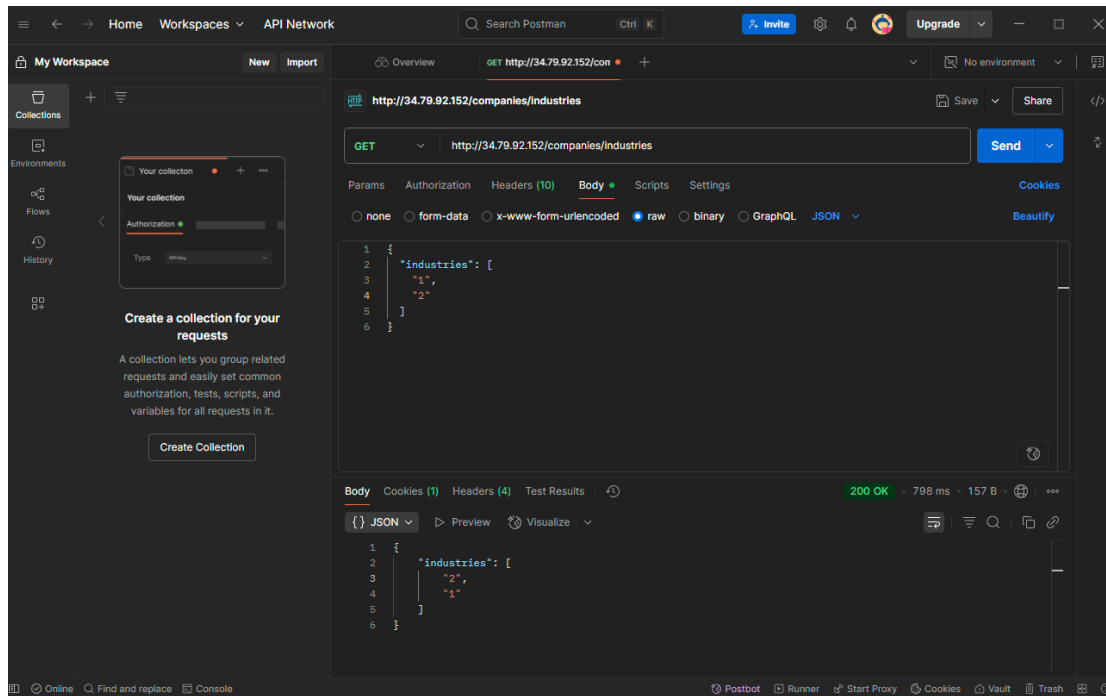
Inicialmente, como pedido pelo grupo 13, foi necessário dar *Sign Up* para utilizar as outras funcionalidades do projeto, para isso, efetuámos um POST com os dados na imagem abaixo:



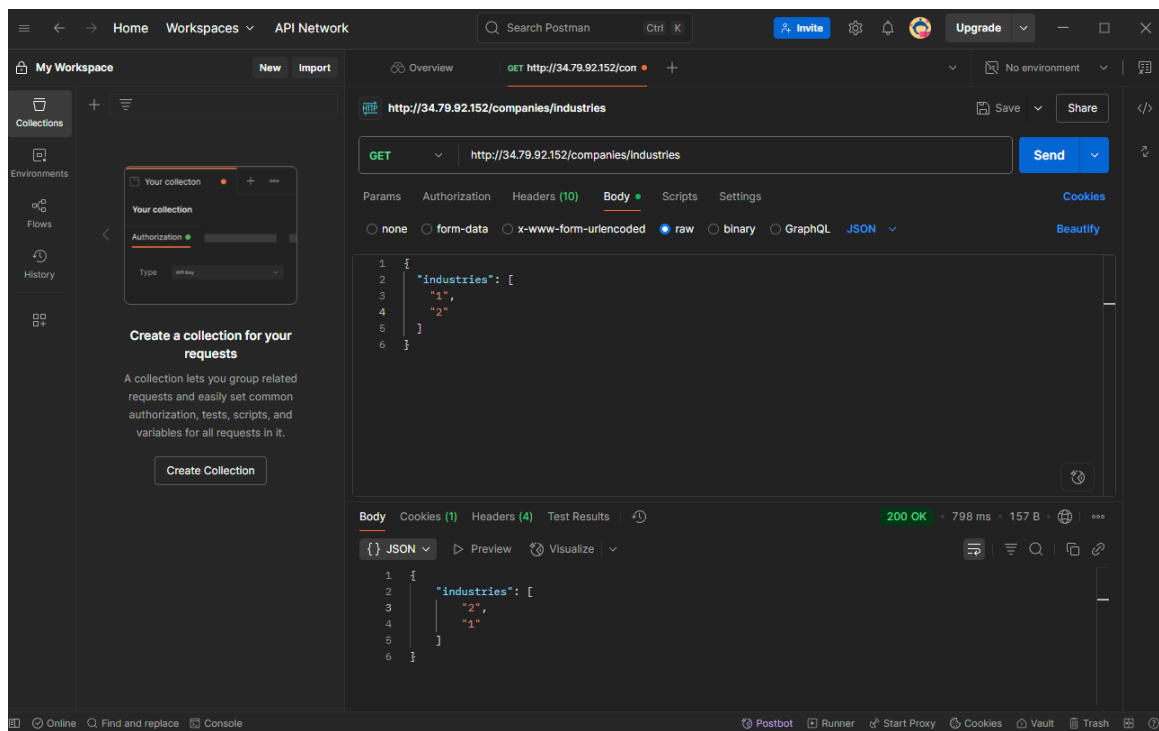
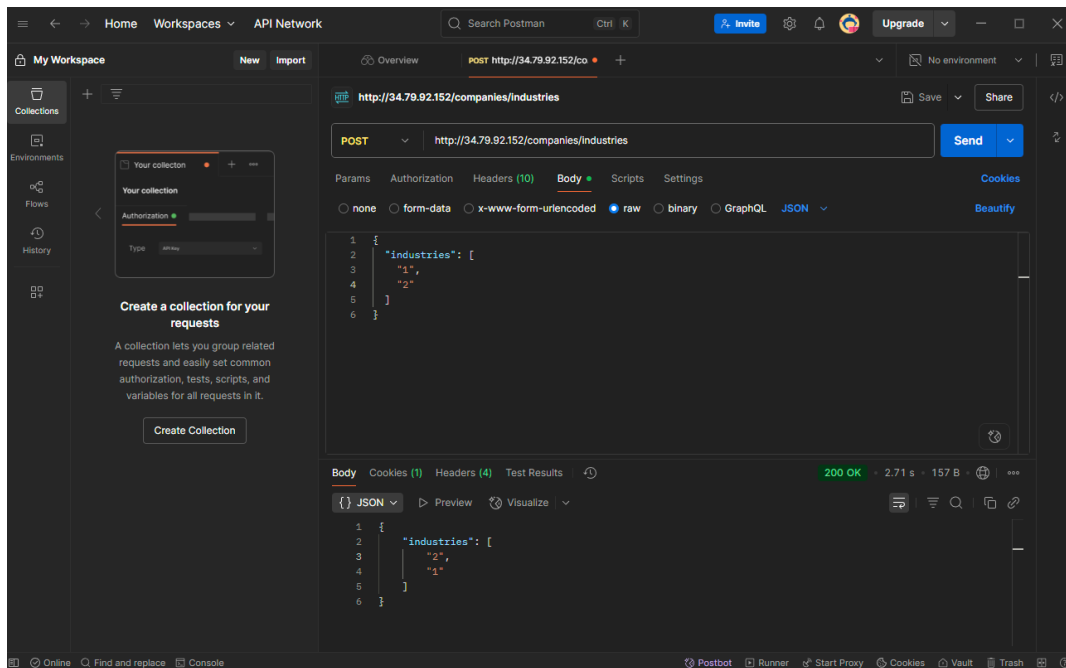
O pedido funcionou como esperado e o utilizador ficou logado.

Endpoint: /companies/industries:

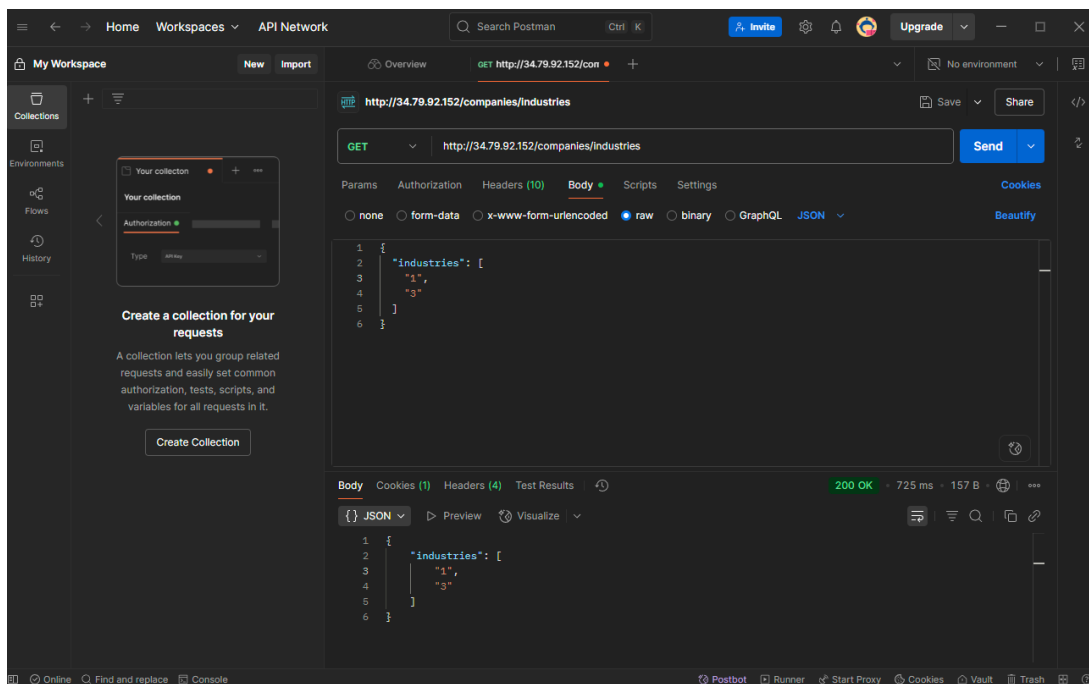
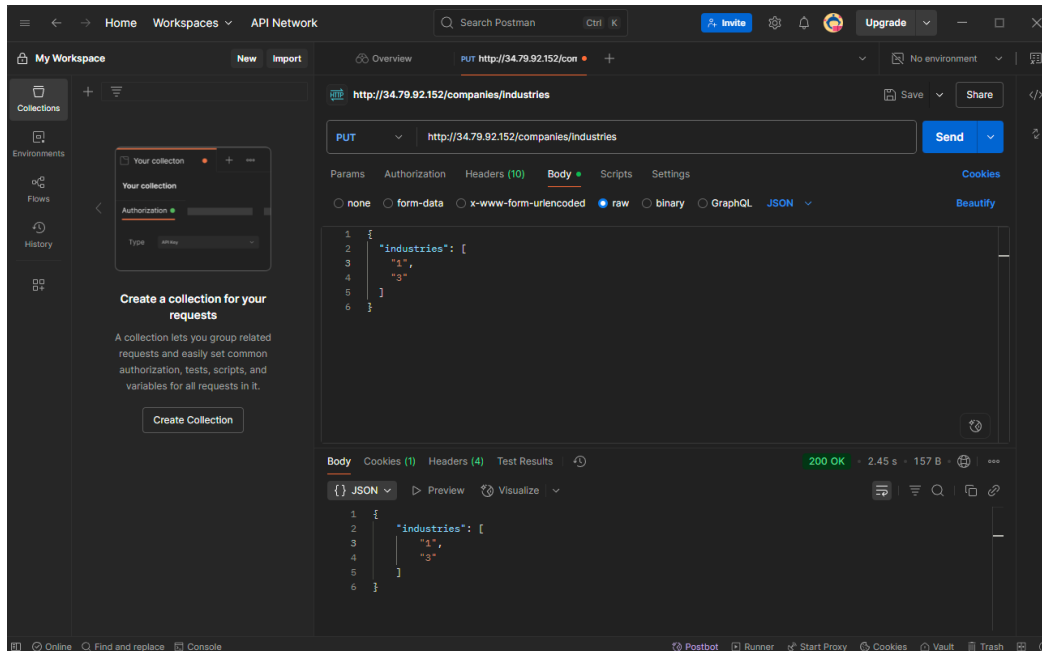
- **GET /companies/industries:**
 - **Resultado:** Retornou a lista de industries com sucesso.
 - **Status Code:** 200 OK.
 - **Observações:** A estrutura da response coincidiu com a documentação.



- **POST /companies/industries:**
 - **Resultado:** Foram criadas novas industries com sucesso.
 - **Status Code:** 200 OK.
 - **Observações:** O corpo do request foi validado corretamente e os elementos adicionados apareceram corretamente em pedidos GET subsequentes.

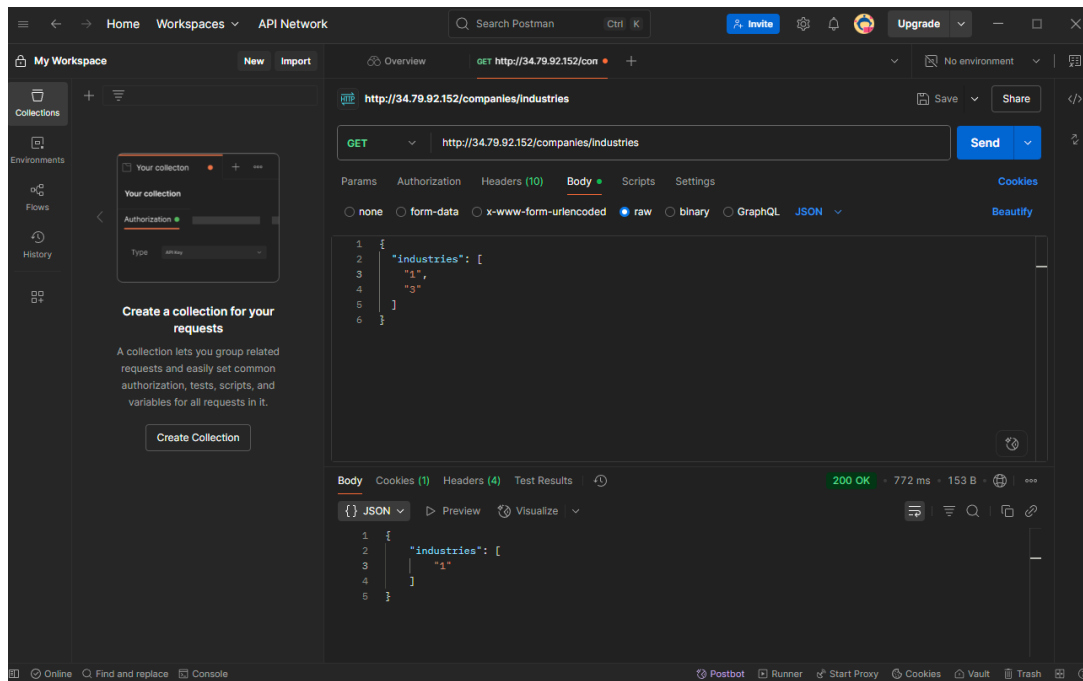


- **PUT /companies/industries:**
 - **Resultado:** As indústrias da company foram modificadas com sucesso.
 - **Status Code:** 200 OK.
 - **Observações:** As mudanças foram alteradas e persistidas com sucesso.



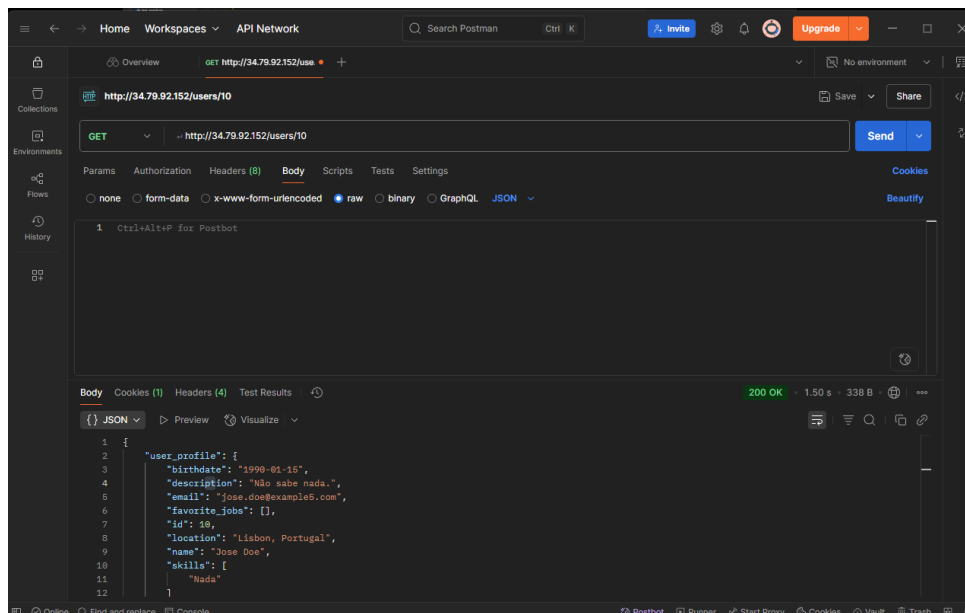
- **DELETE /companies/industries/{name}:**
 - **Resultado:** A industry selecionada foi apagada com sucesso.
 - **Status Code:** 200 OK.

- **Observações:** O recurso deixou de estar visível depois de apagado, como esperado.

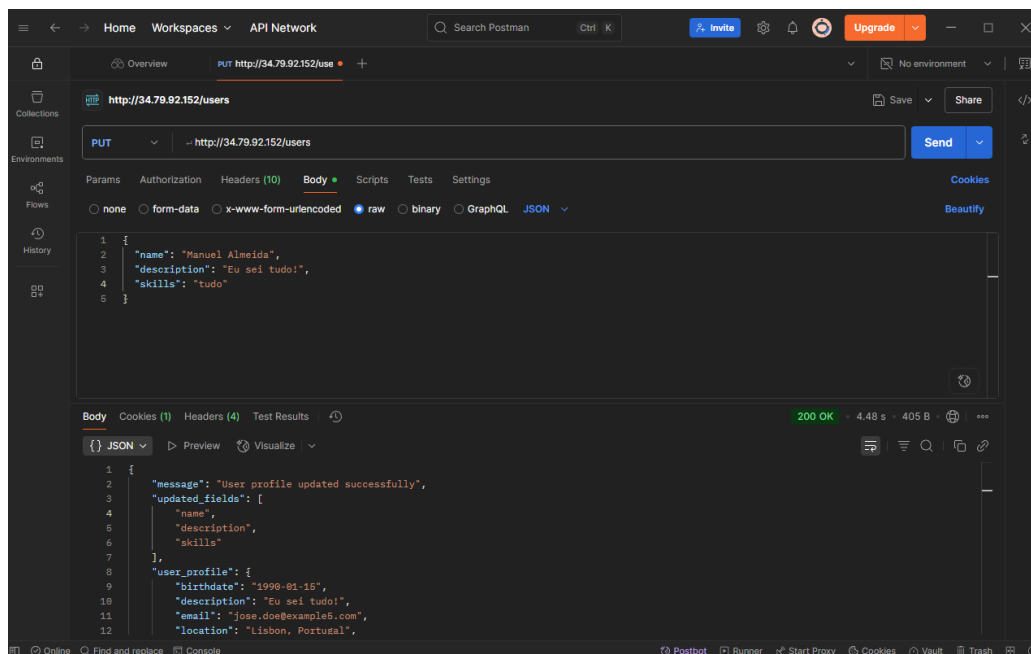


Endpoint: /users:

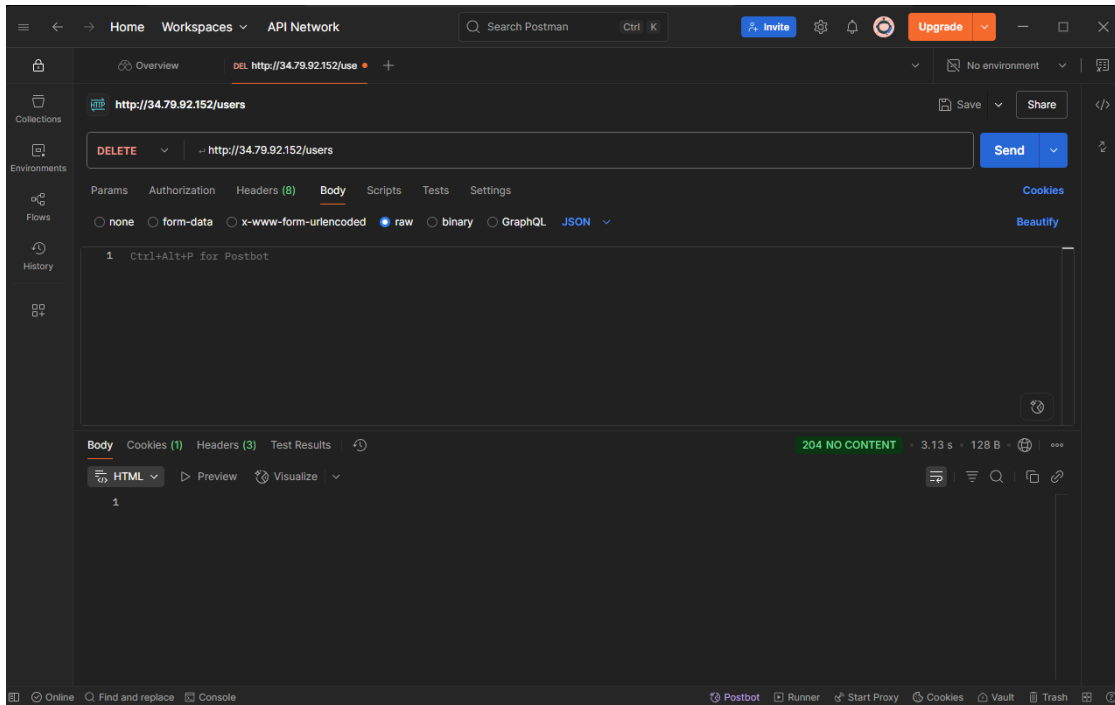
- **GET /users/{id}**: Foi obtido o utilizador que foi criado no sign up acima, funcionou como esperado.



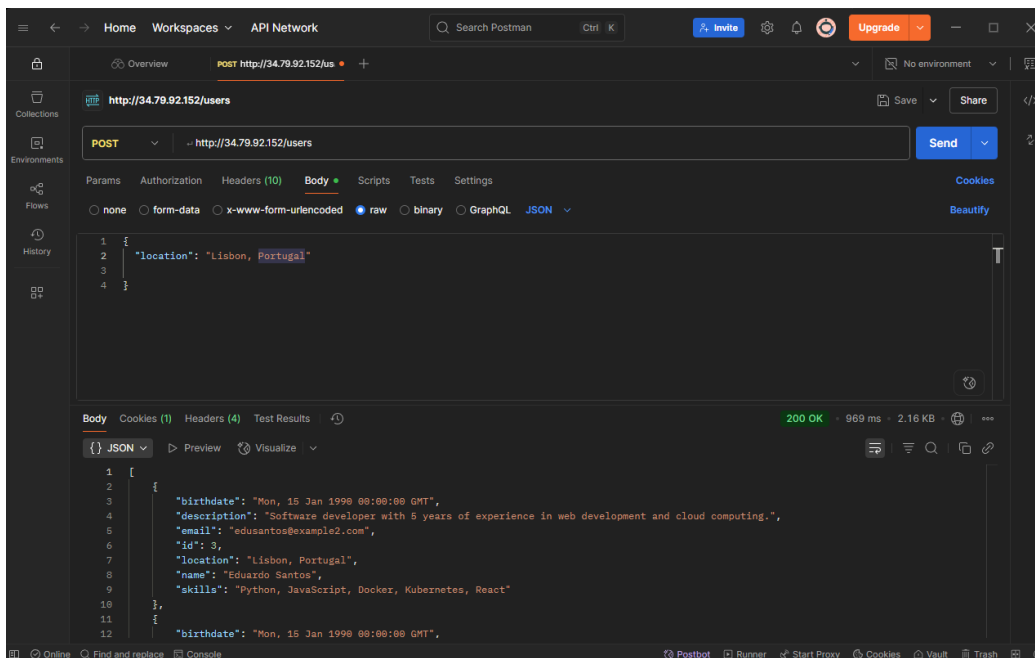
- **PUT /users:** Após tentar alterar o nome, descrição e skills, podemos ver que funcionou e as informações foram atualizadas.



- **DELETE /users:** Após efetuada a operação foi enviado 1 como resposta, indicando que o utilizador foi corretamente apagado.



- **POST /users:** Foi testado com a localização e também funcionou corretamente.



Endpoint: /users/experiences:

- **POST /users/experiences:**

The screenshot shows a REST client interface for a POST request to `http://34.79.92.152/users/experiences`. The request body is a JSON object with the following fields: `job_title` (Developer), `company_name` (Sintavia), `start_date` (2025-06-04), `end_date` (2025-07-04), and `description` (Bom dia Portugal). The response is a 201 status code, indicating successful creation. The response body is a JSON object with the following fields: `experience` (an object with the same fields as the request), `id` (1), `job_title` (Developer), `start_date` (2025-06-04), and `user_id` (7).

```
1 {
2   "job_title": "Developer",
3   "company_name": "Sintavia",
4   "start_date": "2025-06-04",
5   "end_date": "2025-07-04",
6   "description": "Bom dia Portugal"
7 }
8
```

Body Cookies (1) Headers (4) Test Results 201 CREATED

```
1 {
2   "experience": {
3     "company_name": "Sintavia",
4     "description": "Bom dia Portugal",
5     "end_date": "2025-07-04",
6     "id": 1,
7     "job_title": "Developer",
8     "start_date": "2025-06-04",
9     "user_id": 7
10  }
11 }
```

- **GET /users/experiences:**

The screenshot shows a REST client interface for a GET request to `http://34.79.92.152/users/experiences`. The response is a 200 status code, indicating successful retrieval. The response body is a JSON array containing one object with the following fields: `experience` (an object with the same fields as the request), `id` (1), `job_title` (Developer), `start_date` (2025-06-04), and `user_id` (7).

Cloud Rafa / UsersExperiences Save Share

GET http://34.79.92.152/users/experiences Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

Body Cookies (1) Headers (4) Test Results 200 OK 1.28 s 327 B Save Response

```
1 [
2   {
3     "company_name": "Sintavia",
4     "description": "Bom dia Portugal",
5     "end_date": "Fri, 04 Jul 2025 00:00:00 GMT",
6     "id": 1,
7     "job_title": "Developer",
8     "start_date": "Wed, 04 Jun 2025 00:00:00 GMT",
9     "user_id": 7
10  }
11 ]
```

- **DELETE /users/experiences/{id}:**

Cloud Rafa / New Request

DELETE ⌵ http://34.79.92.152/users/experiences/1

Params Authorization Headers (7) Body Scripts Tests Settings

Query Params

| | Key | Value |
|--|-----|-------|
| | Key | Value |

Body Cookies (1) Headers (4) Test Results ↺ **200 OK**

{ } JSON ⌵ ▶ Preview 🔗 Visualize ⌵

```
1 {  
2   "message": "Experience deleted"  
3 }
```

Cloud Rafa / UsersExperiences

GET ⌵ http://34.79.92.152/users/experiences

Params Authorization Headers (7) Body Scripts Tests Set

Query Params

| | Key | Value |
|--|-----|-------|
| | Key | Value |

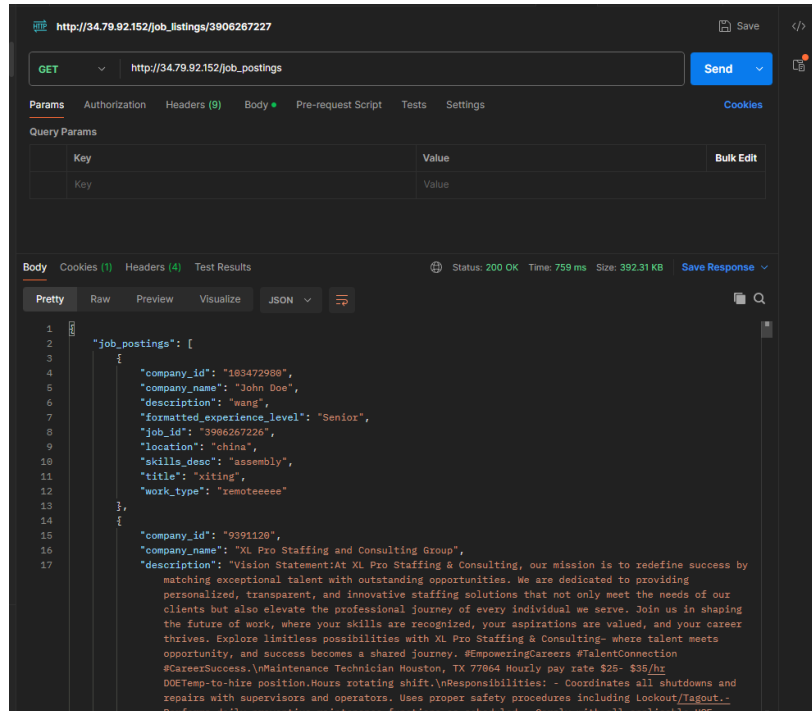
Body Cookies (1) Headers (4) Test Results ↺

{ } JSON ⌵ ▶ Preview 🔗 Visualize ⌵

```
1 []
```


Endpoint: /job_postings:

- **GET /job_postings:**
 - **Resultado:** Retornou a lista de job postings com sucesso.
 - **Status Code:** 200 OK.
 - **Observações:** A estrutura da response coincidiu com a documentação.



- **POST /job_postings:**
 - **Resultado:** Criou e retornou o job posting com a informação dada corretamente.
 - **Status Code:** 201 CREATE.

The screenshot shows a REST client interface with the following details:

- URL:** `http://34.79.92.152/job_postings`
- Method:** `POST`
- Body (Request):**

```
1 {
2   "title": "QA Engineer",
3   "description": "Well paying!",
4   "location": "China",
5   "formatted_experience_level": "Senior",
6   "skills_desc": "GCloud",
7   "work_type": "Full Remote Hybrid"
8 }
```
- Status:** `201 CREATED` | **Time:** `4.01 s` | **Size:** `378 B`
- Body (Response):**

```
1 {
2   "company_id": "193472981",
3   "company_name": "John Doe",
4   "description": "Well paying!",
5   "formatted_experience_level": "Senior",
6   "job_id": "3906267237",
7   "location": "China",
8   "skills_desc": "GCloud",
9   "title": "QA Engineer",
10  "work_type": "Full Remote Hybrid"
11 }
```

- **GET** /job_postings/{job_id}:
 - **Resultado:** Retornou a informação do job posting com o id correspondente.
 - **Status Code:** 200 OK.

The screenshot displays a REST client interface with a GET request to `http://34.79.92.152/job_postings/3906267237`. The response is a JSON object containing job posting details.

Request:

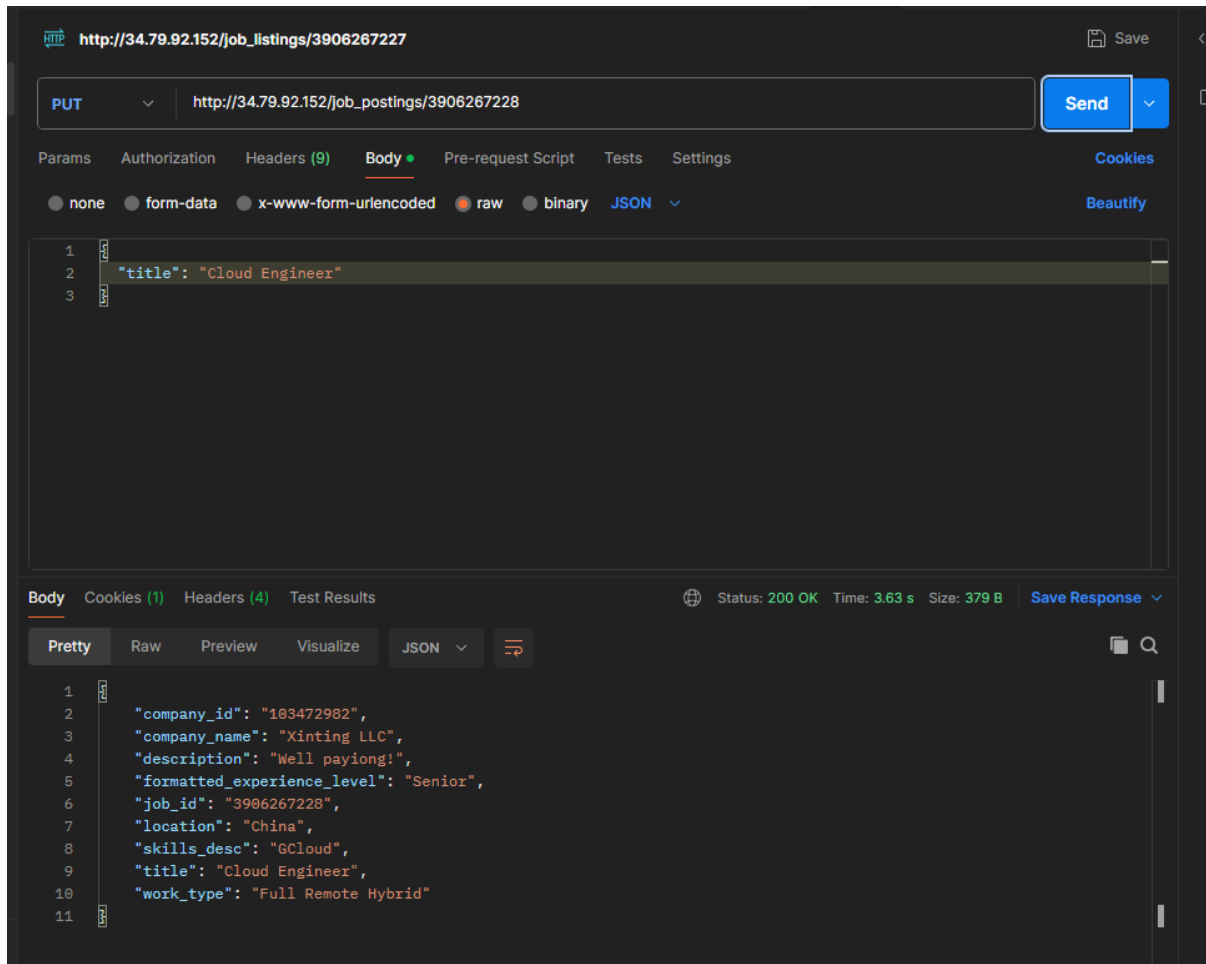
```
1 GET
2 "title": "QA Engineer",
3 "description": "Well payiong!",
4 "location": "China",
5 "formatted_experience_level": "Senior",
6 "skills_desc": "GCloud",
7 "work_type": "Full Remote Hybrid"
8
```

Response:

```
1
2 "company_id": "183472981",
3 "company_name": "John Doe",
4 "description": "Well payiong!",
5 "formatted_experience_level": "Senior",
6 "job_id": "3906267237",
7 "location": "China",
8 "skills_desc": "GCloud",
9 "title": "QA Engineer",
10 "work_type": "Full Remote Hybrid"
11
```

Additional details from the interface: Status: 200 OK, Time: 520 ms, Size: 373 B.

- **PUT** /job_postings/{job_id}:
 - **Resultado:** Atualizou e devolveu a informação o job posting com o id correspondente.
 - **Status Code:** 200 OK.



- **DELETE** /job_postings/{job_id}:
 - **Resultado:** Apagou e devolveu o job posting com o id correspondente.
 - **Status Code:** 200 OK.
 - **Observações:** Após o DELETE, voltou-se a testar os GETs e, tal como esperado, o job posting já não estava presente.

The screenshot shows a REST client interface with a DELETE request to `http://34.79.92.152/job_postings/3906267237`. The request body is a JSON object with the following fields: `"title": "QA Engineer", "description": "Well payiong!", "location": "China", "formatted_experience_level": "Senior", "skills_desc": "GCloud", "work_type": "Full Remote Hybrid"`. The response status is 200 OK, and the response body is a JSON object with the following fields: `"company_id": "103472981", "company_name": "John Doe", "description": "Well payiong!", "formatted_experience_level": "Senior", "job_id": "3906267237", "location": "China", "skills_desc": "GCloud", "title": "QA Engineer", "work_type": "Full Remote Hybrid"`.

```
DELETE http://34.79.92.152/job_postings/3906267237

{"title": "QA Engineer",
"description": "Well payiong!",
"location": "China",
"formatted_experience_level": "Senior",
"skills_desc": "GCloud",
"work_type": "Full Remote Hybrid"}

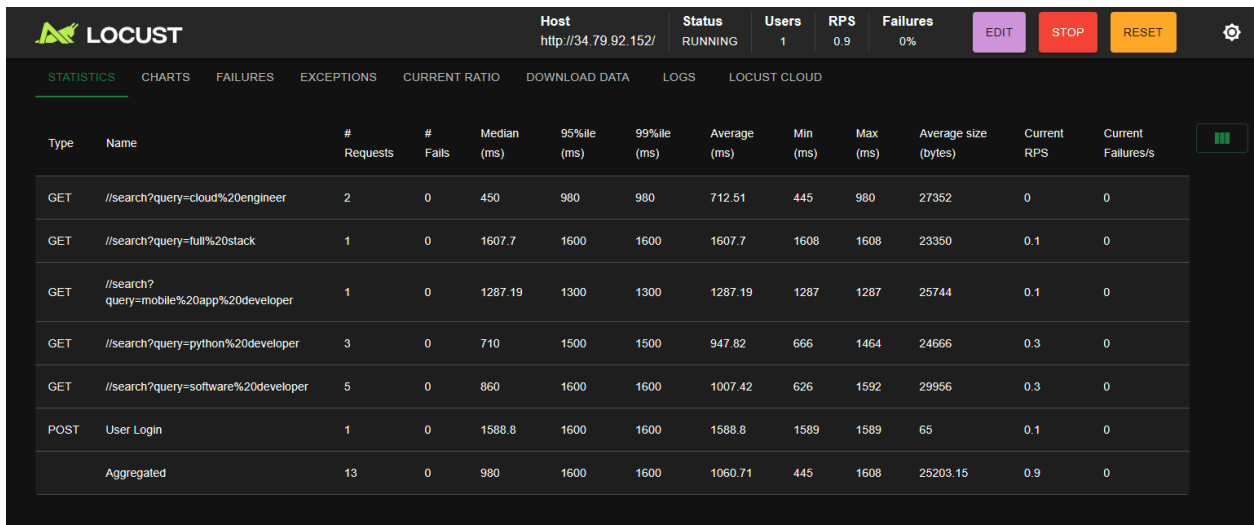
Status: 200 OK Time: 3.43 s Size: 373 B

{"company_id": "103472981",
"company_name": "John Doe",
"description": "Well payiong!",
"formatted_experience_level": "Senior",
"job_id": "3906267237",
"location": "China",
"skills_desc": "GCloud",
"title": "QA Engineer",
"work_type": "Full Remote Hybrid"}
```

Stress Tests:

Endpoint: /search:

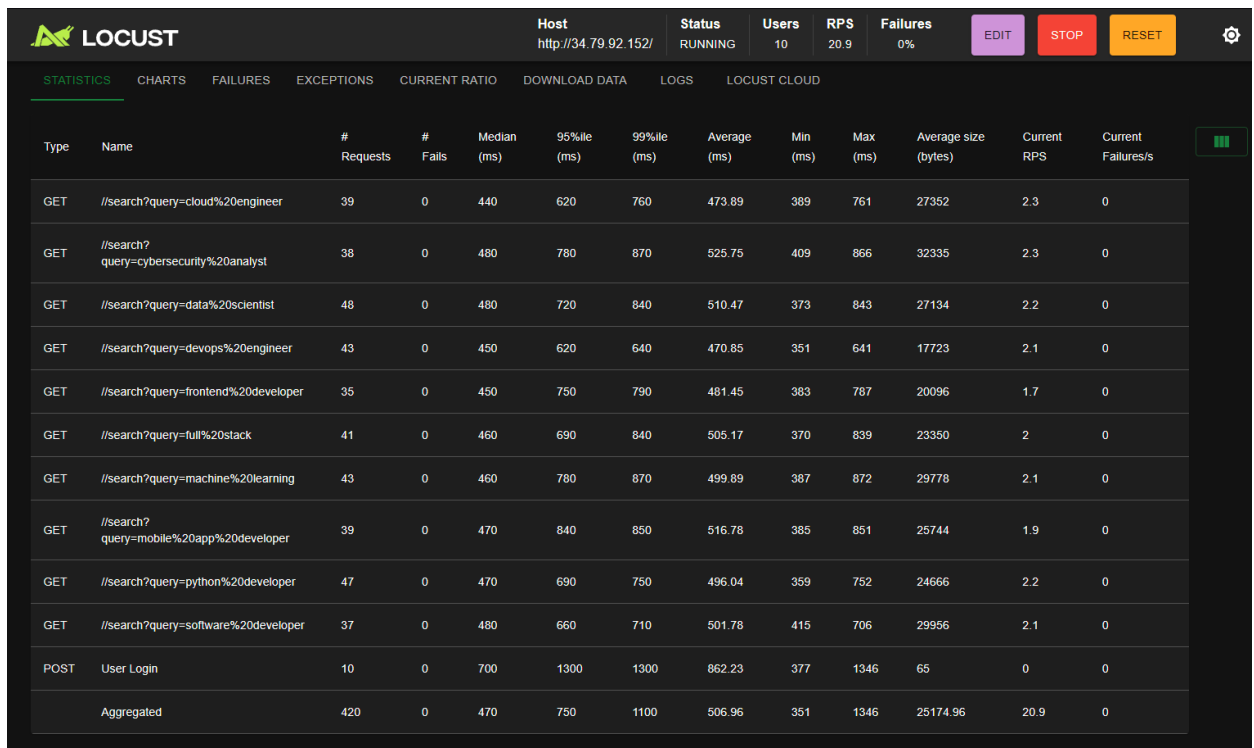
- 1 user: Máximo de 0.9 RPS (1 user criado por segundo).



The Locust interface displays the following statistics for 1 user:

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
|------|---|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|
| GET | //search?query=cloud%20engineer | 2 | 0 | 450 | 980 | 980 | 712.51 | 445 | 980 | 27352 | 0 | 0 |
| GET | //search?query=full%20stack | 1 | 0 | 1607.7 | 1600 | 1600 | 1607.7 | 1608 | 1608 | 23350 | 0.1 | 0 |
| GET | //search?query=mobile%20app%20developer | 1 | 0 | 1287.19 | 1300 | 1300 | 1287.19 | 1287 | 1287 | 25744 | 0.1 | 0 |
| GET | //search?query=python%20developer | 3 | 0 | 710 | 1500 | 1500 | 947.82 | 666 | 1464 | 24666 | 0.3 | 0 |
| GET | //search?query=software%20developer | 5 | 0 | 860 | 1600 | 1600 | 1007.42 | 626 | 1592 | 29956 | 0.3 | 0 |
| POST | User Login | 1 | 0 | 1588.8 | 1600 | 1600 | 1588.8 | 1589 | 1589 | 65 | 0.1 | 0 |
| | Aggregated | 13 | 0 | 980 | 1600 | 1600 | 1060.71 | 445 | 1608 | 25203.15 | 0.9 | 0 |


- 10 users: Máximo de 20.9 RPS (1 user criado por segundo).



The Locust interface displays the following statistics for 10 users:

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
|------|---|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|
| GET | //search?query=cloud%20engineer | 39 | 0 | 440 | 620 | 760 | 473.89 | 389 | 761 | 27352 | 2.3 | 0 |
| GET | //search?query=cybersecurity%20analyst | 38 | 0 | 480 | 780 | 870 | 525.75 | 409 | 866 | 32335 | 2.3 | 0 |
| GET | //search?query=data%20scientist | 48 | 0 | 480 | 720 | 840 | 510.47 | 373 | 843 | 27134 | 2.2 | 0 |
| GET | //search?query=devops%20engineer | 43 | 0 | 450 | 620 | 640 | 470.85 | 351 | 641 | 17723 | 2.1 | 0 |
| GET | //search?query=frontend%20developer | 35 | 0 | 450 | 750 | 790 | 481.45 | 383 | 787 | 20096 | 1.7 | 0 |
| GET | //search?query=full%20stack | 41 | 0 | 460 | 690 | 840 | 505.17 | 370 | 839 | 23350 | 2 | 0 |
| GET | //search?query=machine%20learning | 43 | 0 | 460 | 780 | 870 | 499.89 | 387 | 872 | 29778 | 2.1 | 0 |
| GET | //search?query=mobile%20app%20developer | 39 | 0 | 470 | 840 | 850 | 516.78 | 385 | 851 | 25744 | 1.9 | 0 |
| GET | //search?query=python%20developer | 47 | 0 | 470 | 690 | 750 | 496.04 | 359 | 752 | 24666 | 2.2 | 0 |
| GET | //search?query=software%20developer | 37 | 0 | 480 | 660 | 710 | 501.78 | 415 | 706 | 29956 | 2.1 | 0 |
| POST | User Login | 10 | 0 | 700 | 1300 | 1300 | 862.23 | 377 | 1346 | 65 | 0 | 0 |
| | Aggregated | 420 | 0 | 470 | 750 | 1100 | 506.96 | 351 | 1346 | 25174.96 | 20.9 | 0 |

- **100 users:** Máximo de 30.2 RPS (1 user criado por segundo).



LOCUST

Host

http://34.79.92.152/

Status

RUNNING

Users

100

RPS

30.2


Failures

0%

EDIT

STOP

RESET



STATISTICS

CHARTS

FAILURES


EXCEPTIONS

CURRENT RATIO


DOWNLOAD DATA

LOGS

LOCUST CLOUD

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |  |
|------------|---|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|---|
| GET | //search?query=cloud%20engineer | 339 | 0 | 790 | 6200 | 9400 | 1731.28 | 359 | 10992 | 27352 | 3.3 | 0 | |
| GET | //search?query=cybersecurity%20analyst | 328 | 0 | 870 | 6200 | 8400 | 1752.42 | 366 | 9580 | 32335 | 2.9 | 0 | |
| GET | //search?query=data%20scientist | 342 | 0 | 870 | 6100 | 7800 | 1700.89 | 381 | 8803 | 27134 | 2.8 | 0 | |
| GET | //search?query=devops%20engineer | 353 | 0 | 780 | 6800 | 11000 | 1788.79 | 332 | 14241 | 17723 | 3.5 | 0 | |
| GET | //search?query=frontend%20developer | 332 | 0 | 830 | 5900 | 8900 | 1733.41 | 358 | 11206 | 20096 | 2.7 | 0 | |
| GET | //search?query=full%20stack | 346 | 0 | 850 | 6700 | 9700 | 1885.9 | 360 | 10722 | 23350 | 2.4 | 0 | |
| GET | //search?query=machine%20learning | 363 | 0 | 900 | 7000 | 8600 | 1998 | 376 | 12144 | 29778 | 3.7 | 0 | |
| GET | //search?query=mobile%20app%20developer | 336 | 0 | 880 | 6600 | 10000 | 1955.14 | 376 | 12348 | 25744 | 2.7 | 0 | |
| GET | //search?query=python%20developer | 338 | 0 | 900 | 6600 | 9300 | 1905.81 | 355 | 12352 | 24666 | 2.6 | 0 | |
| GET | //search?query=software%20developer | 355 | 0 | 890 | 5700 | 7900 | 1723.03 | 379 | 9562 | 29956 | 3.6 | 0 | |
| POST | User Login | 100 | 0 | 390 | 1100 | 1200 | 428.31 | 334 | 1176 | 65 | 0 | 0 | |
| Aggregated | | 3532 | 0 | 830 | 6500 | 9000 | 1779.09 | 332 | 14241 | 25085.68 | 30.2 | 0 | |

- **1000 users:** Máximo de 523.9 RPS (100 users criados por segundo).

 LOCUST

Host
http://34.79.92.152/

Status
RUNNING

Users
1000


RPS
523.9

Failures
1%

EDIT

STOP

RESET



STATISTICS

CHARTS

FAILURES


EXCEPTIONS

CURRENT RATIO

DOWNLOAD DATA

LOGS

LOCUST CLOUD

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |  |
|------|---|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|---|
| GET | //search?query=cloud%20engineer | 7249 | 1 | 840 | 9300 | 29000 | 1925.3 | 65 | 87189 | 1921.31 | 50.1 | 0 | |
| GET | //search?query=cybersecurity%20analyst | 7403 | 1 | 840 | 10000 | 25000 | 1884.02 | 65 | 84416 | 2306.38 | 53 | 0 | |
| GET | //search?query=data%20scientist | 7311 | 1 | 840 | 8400 | 30000 | 1941.98 | 36 | 106905 | 1943.5 | 51.4 | 0 | |
| GET | //search?query=devops%20engineer | 7294 | 0 | 840 | 9300 | 28000 | 1931.74 | 65 | 84827 | 1328.54 | 53.3 | 0 | |
| GET | //search?query=frontend%20developer | 7188 | 1 | 840 | 8800 | 27000 | 1867.99 | 64 | 87189 | 1400.25 | 52.5 | 0 | |
| GET | //search?query=full%20stack | 7321 | 0 | 840 | 10000 | 28000 | 1949.45 | 64 | 80150 | 1714.74 | 54.7 | 0 | |
| GET | //search?query=machine%20learning | 7326 | 1 | 840 | 7500 | 26000 | 1804.64 | 64 | 79126 | 2052.93 | 51.4 | 0 | |
| GET | //search?query=mobile%20app%20developer | 7381 | 1 | 840 | 8600 | 30000 | 1910.77 | 65 | 71897 | 1829.91 | 52.6 | 0 | |
| GET | //search?query=python%20developer | 7248 | 2 | 840 | 9500 | 27000 | 1878.06 | 61 | 71307 | 1784.69 | 51.1 | 0 | |
| GET | //search?query=software%20developer | 7252 | 0 | 840 | 9200 | 30000 | 1947.18 | 64 | 87307 | 2088.44 | 53.8 | 0 | |
| POST | User Login | 1000 | 393 | 7500 | 10000 | 11000 | 7109.5 | 584 | 10719 | 55.57 | 0 | 0 | |
| | Aggregated | 73973 | 401 | 840 | 10000 | 28000 | 1974.47 | 36 | 106905 | 1814.21 | 523.9 | 0 | |

- Erros encontrados: Todos relacionados com o serviço de autenticação.

[illegible]

2025-06-04 17:42:57,907 DESKTOP-7378P4BE/INFO/locust.main: Shutting down (exit code 1)

| Type | Name | # reqs | # fails | Avg | Min | Max | Med | req/s | failures/s |
|------------|-------------------------------------|--------|-------------|------|-----|--------|------|--------|------------|
| GET | /searchquery-cloud20engineer | 11245 | 2(0.02%) | 1895 | 64 | 96135 | 828 | 47.33 | 0.01 |
| GET | /searchquery-cybersecurity20analyst | 11374 | 2(0.02%) | 1869 | 64 | 89993 | 810 | 47.38 | 0.01 |
| GET | /searchquery-data20scientist | 11192 | 2(0.02%) | 1858 | 35 | 186984 | 820 | 47.11 | 0.01 |
| GET | /searchquery-devops20engineer | 11251 | 0(0.00%) | 1829 | 64 | 85648 | 820 | 47.36 | 0.00 |
| GET | /searchquery-fronten20developer | 11163 | 1(0.01%) | 1873 | 63 | 144611 | 828 | 46.99 | 0.00 |
| GET | /searchquery-full20stack | 11417 | 1(0.01%) | 1829 | 63 | 149992 | 828 | 48.06 | 0.00 |
| GET | /searchquery-machine20learning | 11433 | 1(0.01%) | 1912 | 63 | 88435 | 828 | 48.13 | 0.00 |
| GET | /searchquery-mobile20app20developer | 11428 | 1(0.01%) | 1907 | 64 | 87077 | 828 | 48.10 | 0.00 |
| GET | /searchquery-python20developer | 11403 | 3(0.03%) | 1884 | 61 | 86669 | 820 | 48.00 | 0.01 |
| GET | /searchquery-software20developer | 11209 | 0(0.00%) | 1898 | 63 | 90448 | 820 | 47.18 | 0.00 |
| POST | User Login | 1000 | 393(39.30%) | 7109 | 584 | 10719 | 7500 | 4.21 | 1.65 |
| Aggregated | | 114115 | 406(0.36%) | 1912 | 35 | 149992 | 828 | 480.35 | 1.71 |

Response time percentiles (approximated)

| Type | Name | 50% | 66% | 75% | 80% | 90% | 95% | 98% | 99% | 99.9% | 99.99% | 100% # reqs |
|------------|-------------------------------------|------|------|-------|-------|-------|-------|-------|-------|-------|--------|-------------|
| GET | /searchquery-cloud20engineer | 820 | 940 | 1000 | 1100 | 1300 | 7900 | 20000 | 20000 | 74000 | 87000 | 11245 |
| GET | /searchquery-cybersecurity20analyst | 810 | 930 | 1000 | 1100 | 1300 | 7000 | 15000 | 26000 | 75000 | 88000 | 11374 |
| GET | /searchquery-data20scientist | 820 | 940 | 1000 | 1100 | 1300 | 5800 | 19000 | 20000 | 70000 | 102000 | 11192 |
| GET | /searchquery-devops20engineer | 820 | 940 | 1000 | 1100 | 1300 | 6200 | 19000 | 27000 | 61000 | 85000 | 11251 |
| GET | /searchquery-fronten20developer | 820 | 940 | 1000 | 1100 | 1300 | 5900 | 19000 | 27000 | 74000 | 127000 | 11163 |
| GET | /searchquery-full20stack | 820 | 940 | 1000 | 1100 | 1300 | 8500 | 20000 | 26000 | 76000 | 146000 | 11417 |
| GET | /searchquery-machine20learning | 820 | 940 | 1000 | 1100 | 1300 | 4900 | 19000 | 26000 | 73000 | 85000 | 11433 |
| GET | /searchquery-mobile20app20developer | 820 | 940 | 1000 | 1100 | 1300 | 5000 | 21000 | 30000 | 72000 | 84000 | 11428 |
| GET | /searchquery-python20developer | 820 | 920 | 1000 | 1100 | 1300 | 6200 | 19000 | 26000 | 64000 | 83000 | 11403 |
| GET | /searchquery-software20developer | 820 | 940 | 1000 | 1100 | 1300 | 6100 | 19000 | 29000 | 78000 | 90000 | 11209 |
| POST | User Login | 7500 | 9800 | 10000 | 10000 | 10000 | 10000 | 10000 | 11000 | 11000 | 11000 | 1000 |
| Aggregated | | 820 | 940 | 1000 | 1100 | 1400 | 8600 | 19000 | 20000 | 74000 | 101000 | 150000 |

Error report

| # occurrences | Error |
|---------------|--|
| 393 | POST User Login: HTTPError('500 Server Error: INTERNAL SERVER ERROR for url: User Login') |
| 2 | GET /searchquery-cloud20engineer: HTTPError('502 Server Error: Bad Gateway for url: /searchquery-cloud20engineer') |
| 2 | GET /searchquery-cybersecurity20analyst: HTTPError('502 Server Error: Bad Gateway for url: /searchquery-cybersecurity20analyst') |
| 2 | GET /searchquery-machine20learning: HTTPError('502 Server Error: Internal Server Error for url: /searchquery-machine20learning') |
| 2 | GET /searchquery-python20developer: HTTPError('502 Server Error: Bad Gateway for url: /searchquery-python20developer') |
| 1 | GET /searchquery-data20scientist: HTTPError('500 Server Error: Internal Server Error for url: /searchquery-data20scientist') |
| 1 | GET /searchquery-mobile20app20developer: HTTPError('500 Server Error: Internal Server Error for url: /searchquery-mobile20app20developer') |
| 1 | GET /searchquery-python20developer: HTTPError('500 Server Error: Internal Server Error for url: /searchquery-python20developer') |
| 1 | GET /searchquery-fronten20developer: HTTPError('502 Server Error: Bad Gateway for url: /searchquery-fronten20developer') |
| 1 | GET /searchquery-full20stack: HTTPError('502 Server Error: Bad Gateway for url: /searchquery-full20stack') |
| 1 | GET /searchquery-data20scientist: HTTPError('502 Server Error: Bad Gateway for url: /searchquery-data20scientist') |

Endpoint: /companies/specialities:

- 1 user:

×

Start new load test

Number of users (peak concurrency) *

1

Ramp up (users started/second) *

1


Host

http://34.79.92.152/

Advanced options

▼

START



Host

http://34.79.92.152/

Status

RUNNING

Users

1

RPS

2.2

Failures

0%

EDIT

STOP

RESET

STATISTICS

CHARTS

FAILURES

EXCEPTIONS

CURRENT RATIO

DOWNLOAD DATA

LOGS

LOCUST CLOUD

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
|------|-------------------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|
| GET | /companies/specialities | 111 | 0 | 390 | 630 | 820 | 439.97 | 316 | 1895 | 20 | 2.2 | 0 |
| POST | Company Login | 1 | 0 | 1151.69 | 1200 | 1200 | 1151.69 | 1152 | 1152 | 79 | 0 | 0 |
| | Aggregated | 112 | 0 | 390 | 680 | 1200 | 446.33 | 316 | 1895 | 20.53 | 2.2 | 0 |

Não houve erros e havia cerca de 2 pedidos por segundo. Tudo a funcionar normalmente.

- **10 users:**

LOCUST

Host

http://34.79.92.152/

Status

RUNNING

Users

10

RPS

25.1

Failures

0%

EDIT

STOP

RESET

STATISTICS

CHARTS

FAILURES

EXCEPTIONS

CURRENT RATIO

DOWNLOAD DATA

LOGS

LOCUST CLOUD

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
|------|--------------------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|
| GET | //companies/specialities | 590 | 0 | 360 | 530 | 710 | 380.33 | 293 | 778 | 20 | 25.1 | 0 |
| POST | Company Login | 10 | 0 | 530 | 1100 | 1100 | 752 | 408 | 1112 | 79 | 0 | 0 |
| | Aggregated | 600 | 0 | 360 | 550 | 780 | 386.52 | 293 | 1112 | 20.98 | 25.1 | 0 |

Também não existiram erros e os pedidos por segundo multiplicaram por 10 como era de esperar. Tudo a funcionar normalmente também.

- **100 users:**

×

Start new load test

Number of users (peak concurrency) *

100

Ramp up (users started/second) *

1

Host

http://34.79.92.152/

Advanced options

▼

START

| LOCUST | | Host | | Status | Users | RPS | Failures | EDIT | | STOP | RESET | |
|------------|-------------------------|----------------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|
| | | http://34.79.92.152/ | | RUNNING | 100 | 26.8 | 0% | | | | | |
| STATISTICS | | | | | | | | | | | | |
| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
| GET | //companies/specialties | 1695 | 1 | 700 | 9600 | 11000 | 2544.09 | 95 | 16660 | 31.03 | 26 | 0.1 |
| POST | Company Login | 97 | 2 | 1100 | 19000 | 26000 | 4177.53 | 110 | 25950 | 78.22 | 0.8 | 0.2 |
| | Aggregated | 1792 | 3 | 730 | 9800 | 13000 | 2632.51 | 95 | 25950 | 33.58 | 26.8 | 0.3 |

```
[2025-06-04 17:47:47,713] Desktop_GFSS/INFO/locust.runners: Ramping to 100 users at a rate of 1.00 per second
Failed to login: {"error": "Authentication service error"}

Failed to login: {"error": "Authentication service error"}

[2025-06-04 17:49:27,226] Desktop_GFSS/INFO/locust.runners: All users spawned: {"TestCompaniesSpecialties": 100} (100 total users)
```

| LOCUST | | | | | | | | | | Host | Status | RPS | Failures | NEW | RESET |
|---|--------------------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|----------|-----|-------|
| | | | | | | | | | | http://34.79.92.152/ | STOPPED | 235.2 | 7% | | |
| STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS LOCUST CLOUD | | | | | | | | | | | | | | | |
| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s | | | |
| GET | //companies/specialities | 7363 | 9 | 3100 | 6200 | 7300 | 3266.23 | 82 | 41833 | 120.93 | 235.2 | 0.2 | | | |
| POST | Company Login | 588 | 555 | 9800 | 17000 | 20000 | 9842.51 | 376 | 20894 | 43.13 | 0 | 0 | | | |
| Aggregated | | 7951 | 564 | 3400 | 8500 | 16000 | 3752.57 | 82 | 41833 | 115.17 | 235.2 | 0.2 | | | |

Com 100 utilizadores houve uma pequena percentagem de erros. Mais especificamente ao dar login, os pedidos multiplicaram-se por 10 novamente. Os erros ocorridos estão na imagem acima e nas imagens abaixo.

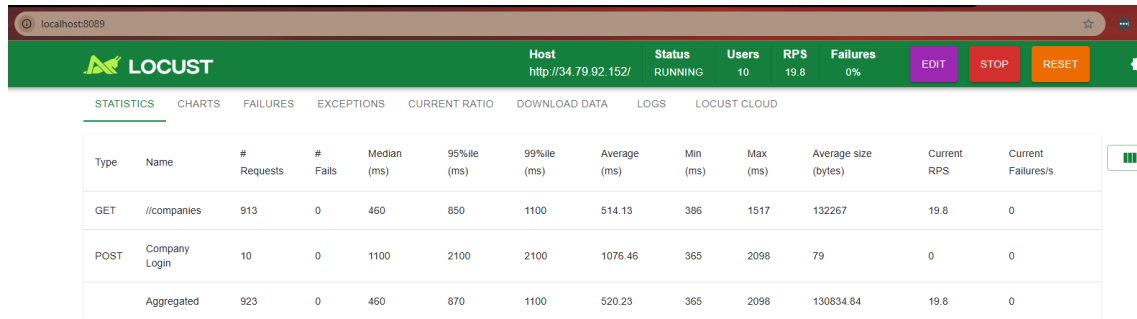
| | | | | | | | | | | | | | |
|--|--|------|-------|-------|-------|-------|-------|-------|-------|-------|--------|-------|--------|
| POST | Company Login | 9800 | 12000 | 13000 | 14000 | 16000 | 17000 | 19000 | 20000 | 21000 | 21000 | 21000 | 588 |
| Aggregated | | 3400 | 4200 | 4700 | 5100 | 6300 | 8500 | 14000 | 16000 | 28000 | 42000 | 42000 | 7951 |
| Error report | | | | | | | | | | | | | |
| # occurrences | Error | | | | | | | | | | | | |
| 555 | POST Company Login: HTTPError('500 Server Error: INTERNAL SERVER ERROR for url: Company Login') | | | | | | | | | | | | |
| 9 | GET //companies/specialities: HTTPError('502 Server Error: Bad Gateway for url: //companies/specialities') | | | | | | | | | | | | |
| Response time percentiles (approximated) | | | | | | | | | | | | | |
| Type | Name | 50% | 66% | 75% | 80% | 90% | 95% | 98% | 99% | 99.9% | 99.99% | 100% | # reqs |
| GET | //companies/specialities | 3100 | 4000 | 4500 | 4800 | 5500 | 6200 | 7000 | 7300 | 28000 | 42000 | 42000 | 7363 |
| POST | Company Login | 9800 | 12000 | 13000 | 14000 | 16000 | 17000 | 19000 | 20000 | 21000 | 21000 | 21000 | 588 |
| Aggregated | | 3400 | 4200 | 4700 | 5100 | 6300 | 8500 | 14000 | 16000 | 28000 | 42000 | 42000 | 7951 |

Endpoint: /companies:

- **1 User:** Máximo de 1.5 RPS.

| LOCUST | | | | | | | | | | Host | Status | Users | RPS | Failures | EDIT | STOP | RESET | |
|---|---------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|-----|----------|------|------|-------|--|
| | | | | | | | | | | http://34.79.92.152/ | RUNNING | 1 | 1.5 | 0% | | | | |
| STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS LOCUST CLOUD | | | | | | | | | | | | | | | | | | |
| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s | | | | | | |
| GET | //companies | 70 | 0 | 460 | 780 | 1500 | 520.5 | 401 | 1517 | 132267 | 1.5 | 0 | | | | | | |
| POST | Company Login | 1 | 0 | 1130.3 | 1100 | 1100 | 1130.3 | 1130 | 1130 | 79 | 0 | 0 | | | | | | |
| Aggregated | | 71 | 0 | 460 | 870 | 1500 | 529.09 | 401 | 1517 | 130405.2 | 1.5 | 0 | | | | | | |

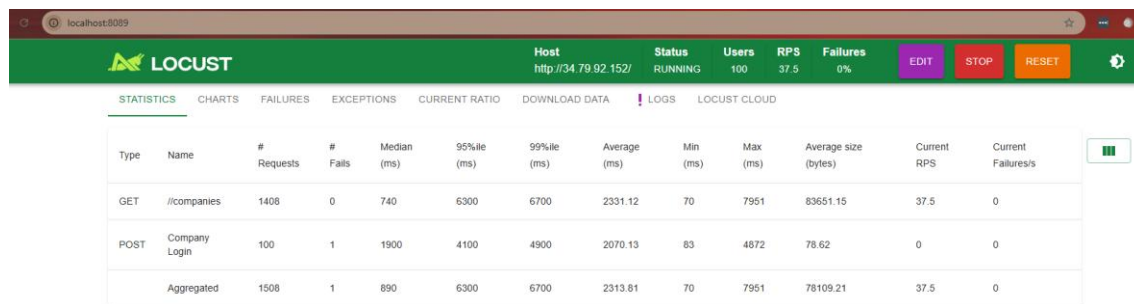
- **10 Users:** Máximo de 19.8 RPS.



The screenshot shows the Locust web interface with the following data:

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
|------------|---------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|
| GET | /companies | 913 | 0 | 460 | 850 | 1100 | 514.13 | 386 | 1517 | 132267 | 19.8 | 0 |
| POST | Company Login | 10 | 0 | 1100 | 2100 | 2100 | 1076.46 | 365 | 2098 | 79 | 0 | 0 |
| Aggregated | | 923 | 0 | 460 | 870 | 1100 | 520.23 | 365 | 2098 | 130834.84 | 19.8 | 0 |

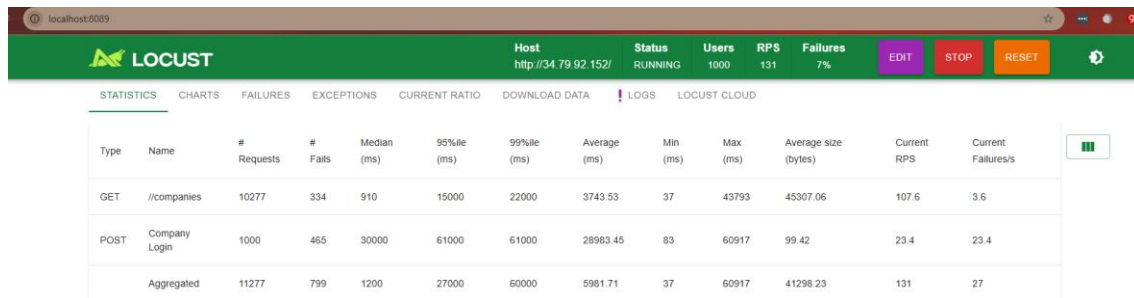
- **100 Users:** Máximo de 37.5 RPS.



The screenshot shows the Locust web interface with the following data:

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
|------------|---------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|
| GET | /companies | 1408 | 0 | 740 | 6300 | 6700 | 2331.12 | 70 | 7951 | 83651.15 | 37.5 | 0 |
| POST | Company Login | 100 | 1 | 1900 | 4100 | 4900 | 2070.13 | 83 | 4872 | 78.62 | 0 | 0 |
| Aggregated | | 1508 | 1 | 890 | 6300 | 6700 | 2313.81 | 70 | 7951 | 78109.21 | 37.5 | 0 |

- **1000 Users:** Máximo de 131 RPS.



The screenshot shows the Locust web interface with the following data:

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
|------------|---------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|
| GET | /companies | 10277 | 334 | 910 | 15000 | 22000 | 3743.53 | 37 | 43793 | 45307.06 | 107.6 | 3.6 |
| POST | Company Login | 1000 | 465 | 30000 | 61000 | 61000 | 28983.45 | 83 | 60917 | 99.42 | 23.4 | 23.4 |
| Aggregated | | 11277 | 799 | 1200 | 27000 | 60000 | 5981.71 | 37 | 60917 | 41298.23 | 131 | 27 |

Erros ao usar **1000 Users**:

| Error report | |
|---------------|---|
| # occurrences | Error |
| 133 | POST Company Login: HTTPError('500 Server Error: INTERNAL SERVER ERROR for url: Company Login') |
| 334 | GET //companies: HTTPError('502 Server Error: Bad Gateway for url: //companies') |
| 142 | POST Company Login: HTTPError('502 Server Error: Bad Gateway for url: Company Login') |
| 190 | POST Company Login: HTTPError('504 Server Error: Gateway Time-out for url: Company Login') |

Endpoint: job_postings/company:

O endpoint em si aguentou com a carga de stress causada, no entanto, a autenticação começou a falhar com o aumento dos users, que por sua vez, fazia com que não fosse possível testar o endpoint que se queria realmente testar.


- 1 User:

| LOCUST | | | | | | | | | | | |
|---------------------|---------------------------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|
| Host | | | | | | Status | Users | RPS | Failures | | |
| http://34.79.92.152 | | | | | | RUNNING | 1 | 1.4 | 0% | EDIT | STOP |
| | | | | | | | | | | RESET | |
| STATISTICS | | | | | | | | | | | |
| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS |
| GET | /job_postings/company/103472981 | 33 | 0 | 690 | 870 | 920 | 699.01 | 638 | 917 | 1755 | 1.4 |
| POST | /login | 1 | 0 | 1084.55 | 1100 | 1100 | 1084.55 | 1085 | 1085 | 79 | 0 |
| Aggregated | | 34 | 0 | 690 | 920 | 1100 | 710.35 | 638 | 1085 | 1705.71 | 1.4 |

- 10 Users:

| LOCUST | | | | | | | | | | | |
|---------------------|---------------------------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|
| Host | | | | | | Status | Users | RPS | Failures | | |
| http://34.79.92.152 | | | | | | RUNNING | 10 | 110.6 | 0% | EDIT | STOP |
| | | | | | | | | | | RESET | |
| STATISTICS | | | | | | | | | | | |
| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS |
| GET | /job_postings/company/103472981 | 2984 | 0 | 90 | 99 | 100 | 89.6 | 70 | 122 | 122 | 110.6 |
| POST | Company Login | 10 | 10 | 5100 | 5100 | 5100 | 5107.01 | 5100 | 5117 | 41 | 0 |
| Aggregated | | 2994 | 10 | 90 | 99 | 110 | 106.36 | 70 | 5117 | 121.73 | 110.6 |

- 100 Users:

LOCUST

Host

http://34.79.92.152

Status

RUNNING

Users

100

RPS

70.1


Failures

0%

EDIT

STOP

RESET



STATISTICS

CHARTS

FAILURES


EXCEPTIONS

CURRENT RATIO


DOWNLOAD DATA

LOGS

LOCUST CLOUD

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |  |
|------|---------------------------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|---|
| GET | /job_postings/company/103472981 | 1820 | 0 | 690 | 3300 | 4200 | 1131.37 | 75 | 5965 | 1213.06 | 69.4 | 0 | |
| POST | Company Login | 100 | 2 | 450 | 2500 | 5300 | 736.31 | 345 | 5312 | 66.7 | 0.7 | 0 | |
| | Aggregated | 1920 | 2 | 680 | 3300 | 4200 | 1110.8 | 75 | 5965 | 1153.35 | 70.1 | 0 | |

- 1000 Users (início):

 LOCUST

Host
http://34.79.92.152

Status
RUNNING

Users
1000


RPS
33.4

Failures
0%

EDIT

STOP

RESET



STATISTICS

CHARTS

FAILURES

EXCEPTIONS


CURRENT RATIO

DOWNLOAD DATA


LOGS

LOCUST CLOUD

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
|------|---------------------------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|
| GET | /job_postings/company/103472981 | 580 | 0 | 5000 | 15000 | 16000 | 6064.53 | 583 | 15907 | 1755 | 28.4 | 0 |
| POST | /login | 232 | 0 | 4900 | 27000 | 29000 | 11106.21 | 1781 | 29797 | 79 | 5 | 0 |
| | Aggregated | 812 | 0 | 4900 | 24000 | 27000 | 7505.01 | 583 | 29797 | 1276.14 | 33.4 | 0 |



- 1000 Users (começou a falhar):

 LOCUST

Host

http://34.79.92.152

Status

RUNNING

Users

1000

RPS

164.1


Failures

25%

EDIT

STOP

RESET



STATISTICS

CHARTS

FAILURES


EXCEPTIONS

CURRENT RATIO


DOWNLOAD DATA

LOGS

LOCUST CLOUD

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |  |
|------------|---------------------------------|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|---|
| GET | /job_postings/company/103472981 | 2838 | 229 | 3900 | 17000 | 21000 | 5572.29 | 54 | 28248 | 925.3 | 119 | 15.5 | |
| POST | /login | 1000 | 718 | 53000 | 60000 | 61000 | 45051.66 | 1781 | 60734 | 131.77 | 45.1 | 44.4 | |
| Aggregated | | 3838 | 947 | 5900 | 60000 | 60000 | 15858.73 | 54 | 60734 | 718.54 | 164.1 | 59.9 | |

Todos os endpoints combinados: Testámos todos os endpoints em conjunto, com **100 users totais, 1 user criado por segundo**.

LOCUST

Host

http://34.79.92.152/

Status

STOPPED

RPS


118.4

Failures

0%

NEW

RESET



STATISTICS

CHARTS

FAILURES

EXCEPTIONS

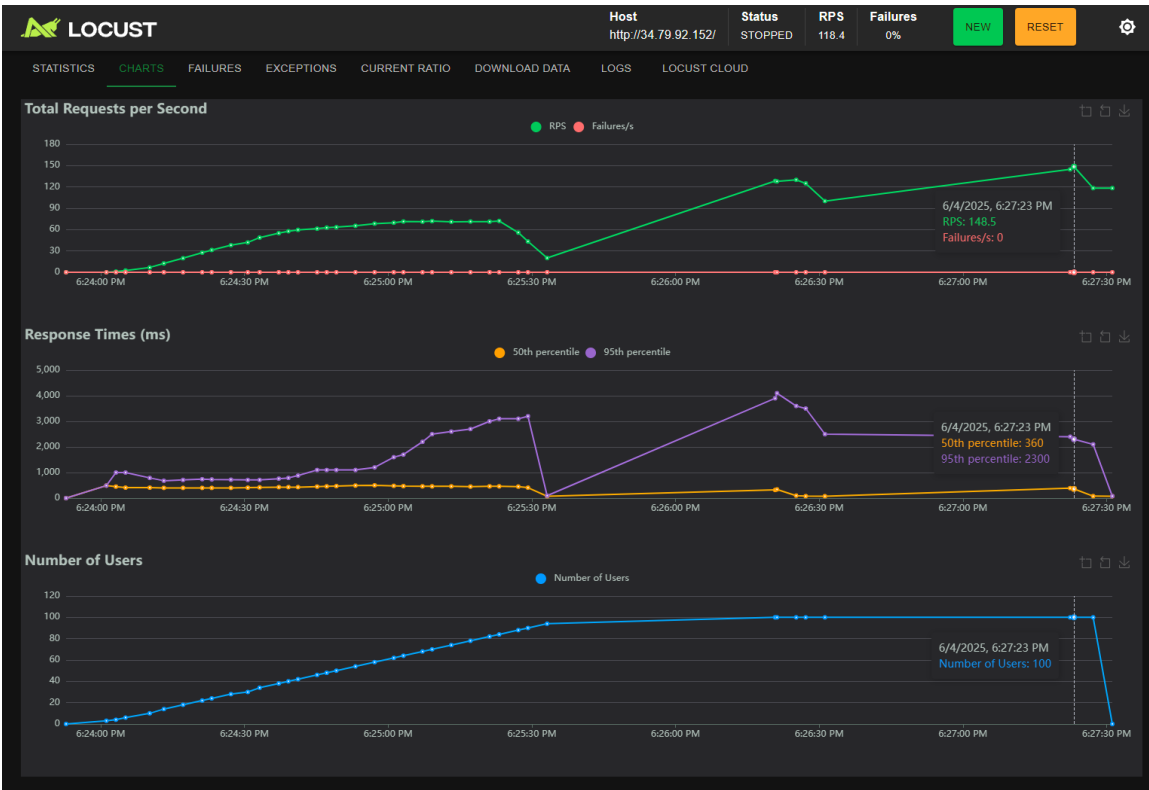
CURRENT RATIO

DOWNLOAD DATA

LOGS

LOCUST CLOUD

| Type | Name | # Requests | # Fails | Median (ms) | 95%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s | |
|------|--|------------|---------|-------------|-------------|-------------|--------------|----------|----------|----------------------|-------------|--------------------|--|
| GET | /companies | 2325 | 0 | 450 | 3700 | 22000 | 1279.3 | 62 | 57687 | 108907.17 | 15.7 | 0 | |
| GET | /companies/specialties | 2783 | 0 | 380 | 2900 | 23000 | 1067.41 | 63 | 57517 | 33.74 | 17.9 | 0 | |
| GET | /job_postings/company/103472981 | 1894 | 0 | 700 | 3600 | 28000 | 1560.14 | 63 | 49194 | 1442.88 | 11.3 | 0 | |
| GET | /search?query=cloud%20engineer | 955 | 0 | 77 | 560 | 1200 | 335.24 | 62 | 46276 | 9046.6 | 6.8 | 0 | |
| GET | /search?query=cybersecurity%20analyst | 944 | 0 | 78 | 550 | 690 | 264.32 | 62 | 40893 | 11314.65 | 8 | 0 | |
| GET | /search?query=data%20scientist | 1005 | 0 | 77 | 570 | 1200 | 299.56 | 62 | 26042 | 9663.55 | 7.4 | 0 | |
| GET | /search?query=devops%20engineer | 975 | 0 | 78 | 580 | 1200 | 334.48 | 61 | 41408 | 6205.63 | 7.4 | 0 | |
| GET | /search?query=frontend%20developer | 954 | 0 | 77 | 570 | 840 | 346.48 | 62 | 44004 | 6800.94 | 6.6 | 0 | |
| GET | /search?query=full%20stack | 932 | 0 | 77 | 570 | 900 | 293.29 | 62 | 26867 | 7947.74 | 5.8 | 0 | |
| GET | /search?query=machine%20learning | 937 | 0 | 78 | 540 | 760 | 239.15 | 62 | 19545 | 10091.73 | 8.3 | 0 | |
| GET | /search?query=mobile%20app%20developer | 977 | 0 | 77 | 580 | 1300 | 349.54 | 62 | 47744 | 8356.71 | 8.1 | 0 | |
| GET | /search?query=python%20developer | 999 | 0 | 77 | 560 | 1200 | 336.98 | 62 | 37087 | 8499.88 | 7.1 | 0 | |
| GET | /search?query=software%20developer | 961 | 0 | 78 | 560 | 830 | 338.96 | 62 | 38991 | 10273.63 | 8 | 0 | |
| POST | Company Login | 60 | 0 | 480 | 21000 | 24000 | 3326.06 | 348 | 23716 | 79 | 0 | 0 | |
| POST | User Login | 20 | 4 | 1100 | 5100 | 5100 | 1782.84 | 371 | 5074 | 60.2 | 0 | 0 | |
| | Aggregated | 16721 | 4 | 360 | 2300 | 5300 | 727.46 | 61 | 57687 | 20393.81 | 118.4 | 0 | |



Feedback Final:

O sistema está funcional e bem documentado, demonstrando um bom nível de maturidade e organização no seu desenvolvimento. A documentação é clara, o que facilita a compreensão da arquitetura e a utilização dos diferentes serviços disponíveis. No entanto, existe uma área com potencial para melhoria: o serviço de autenticação.

Atualmente, o serviço de autenticação funciona corretamente, mas poderia ser otimizado para garantir um desempenho mais eficiente, especialmente em cenários de carga elevada ou picos de utilização.

Investir nesta melhoria não só aumentaria a robustez do sistema como também garantiria uma melhor experiência para os utilizadores em contextos de elevada concorrência, contribuindo para a estabilidade geral da aplicação.

Cost Analysis (FinOps)

Utilização e Custos Reais:

Durante a fase de desenvolvimento e testes do projeto, o custo total na Google Cloud Platform (GCP) foi de apenas \$10,30 nos meses de abril e maio de 2025:

- **Abril 2025: \$4,14**

| Service description | SKU description | Usage amount | Usage unit | Cost (\$) |
|---------------------|--|--------------|----------------|-----------|
| | | | | 4.14 |
| Kubernetes Engine | Autopilot Pod mCPU Requests (europe-west4) | 26,868.042 | hour | 1.32 |
| Kubernetes Engine | Autopilot Kubernetes Clusters | 11.918 | hour | 1.19 |
| Cloud Monitoring | Prometheus Samples Ingested | 7,761,562 | Samples | 0.47 |
| Compute Engine | Balanced PD Capacity in Netherlands | 2.25 | gibibyte month | 0.25 |
| Networking | Cloud Load Balancer Forwarding Rule Minimum for Netherlands (europe-west4) | 8.848 | hour | 0.24 |
| Kubernetes Engine | Autopilot Pod Memory Requests (europe-west4) | 37.334 | gibibyte hour | 0.20 |
| Networking | Networking Private Service Connect Partner Select End Point | 11.359 | hour | 0.11 |
| Networking | Network Intelligence Center Network Analyzer Resource Hours | 86 | count | 0.09 |
| Networking | Network Intelligence Center Topology and Google Cloud Performance Resource Hours | 86 | count | 0.09 |
| Networking | Network Intelligence Center Internet to Google Cloud Performance Resource Hours | 86 | count | 0.07 |
| Networking | Cloud Load Balancer Forwarding Rule Minimum Global | 1.895 | hour | 0.05 |
| Kubernetes Engine | Autopilot SSD Pod Ephemeral Storage Requests (europe-west4) | 150.14 | gibibyte hour | 0.02 |
| Compute Engine | Network Internet Data Transfer Out from Netherlands to EMEA | 0.161 | gibibyte | 0.02 |

- **Maio 2025:** \$6,16

| Service description | SKU description | Usage amount | Usage unit | Cost (\$) |
|---------------------|--|--------------|----------------|-----------|
| | | | | 6.16 |
| Kubernetes Engine | Autopilot Pod mCPU Requests (europe-west4) | 42,661.097 | hour | 2.09 |
| Kubernetes Engine | Autopilot Kubernetes Clusters | 19.189 | hour | 1.92 |
| Networking | Cloud Load Balancer Forwarding Rule Minimum for Netherlands (europe-west4) | 16.095 | hour | 0.44 |
| Compute Engine | Balanced PD Capacity in Netherlands | 4.014 | gibibyte month | 0.44 |
| Cloud Monitoring | Prometheus Samples Ingested | 6,387,996 | Samples | 0.38 |
| Kubernetes Engine | Autopilot Pod Memory Requests (europe-west4) | 70.077 | gibibyte hour | 0.38 |
| Networking | Networking Private Service Connect Partner Select End Point | 18.364 | hour | 0.18 |
| Networking | Network Intelligence Center Network Analyzer Resource Hours | 87 | count | 0.10 |
| Networking | Network Intelligence Center Topology and Google Cloud Performance Resource Hours | 87 | count | 0.10 |
| Networking | Network Intelligence Center Internet to Google Cloud Performance Resource Hours | 87 | count | 0.07 |
| Kubernetes Engine | Autopilot SSD Pod Ephemeral Storage Requests (europe-west4) | 184.741 | gibibyte hour | 0.03 |
| Compute Engine | Network Internet Data Transfer Out from Netherlands to EMEA | 0.1 | gibibyte | 0.01 |

Este custo reduzido deveu-se não só à natureza leve do ambiente de desenvolvimento, mas também a uma estratégia consciente de contenção de custos, em que se privilegiou o desenvolvimento local em Minikube, utilizando o GKE apenas para testes pontuais e validações finais:

- GKE Autopilot (CPU, memória, horas de cluster): ~70% do custo total
- Monitorização e logging
- Redes e Armazenamento Básico (Load Balancer, Persistent Disk)

Custo Estimado em Operação Total:

Foi realizada uma estimativa de custos com base na configuração de recursos utilizada durante o desenvolvimento e nos resultados de um teste de stress com 200 utilizadores simultâneos, número esse que coincide com a quantidade de utilizadores que esperamos atingir numa fase inicial após o lançamento público do sistema.

Durante este teste, foram identificadas limitações em dois microserviços, causadas por falta de recursos. Como medida preventiva, decidiu-se aumentar a alocação atual de recursos na estimativa para ambiente de produção, permitindo uma distribuição mais adequada entre os microserviços com maior exigência.

O custo mensal estimado para suportar essa carga é de aproximadamente 1627,07€, com os seguintes destaques:

- CPU, RAM em instâncias do GKE: ~1590 €
- Serviços de Rede e Load Balancing: ~20 €
- Armazenamento Persistente e Hyperdisk: ~6 €

| service_display_name | name | quantity | region | total_price, EUR |
|-----------------------------------|--|----------|--------------|------------------|
| GKE (Kubernetes Engine) | Zonal Kubernetes Clusters | 0.0 | global | 0 |
| GKE (Kubernetes Engine) | Regional Kubernetes Clusters | 0.0 | global | 0 |
| GKE (Kubernetes Engine) | Balanced PD Capacity in Netherlands | 50.0 | europa-west4 | 4.86282 |
| GKE (Kubernetes Engine) | Custom Instance Ram running in Netherlands | 39420.0 | europa-west4 | 170.68304 |
| GKE (Kubernetes Engine) | Custom Instance Core running in Netherlands | 43800.0 | europa-west4 | 1415.12093 |
| Instances (Compute Engine) | Spot Preemptible N4 Instance Core running in Netherlands | 1460.0 | europa-west4 | 10.61086 |
| Instances (Compute Engine) | Spot Preemptible N4 Instance Ram running in Netherlands | 5840.0 | europa-west4 | 4.82265 |
| Instances (Compute Engine) | Hyperdisk Balanced Capacity in Netherlands | 10.0 | europa-west4 | 0.74269 |
| Cloud Load Balancing (Networking) | Regional External Application Load Balancer Inbound Data F | 100.0 | | 0.79574 |
| Cloud Load Balancing (Networking) | Regional External Application Load Balancer Outbound Data | 100.0 | | 0.79574 |
| Cloud Load Balancing (Networking) | Regional External Proxy Network Load Balancer Forwarding | 1.0 | | 17.74931 |
| Cloud Logging (Cloud Operations) | Log Storage cost | 50.0 | global | 0 |
| Cloud Logging (Cloud Operations) | Log Retention cost | 100.0 | global | 0.88415 |
| | | | | 1627.06792 |

Conclusions

Conseguimos atingir os objetivos principais que tínhamos planejado para este projeto. Escolhemos um tema que gostamos e conseguimos contribuir para a cloud, aplicando conceitos e práticas que fomos aprendendo ao longo do semestre. Durante a segunda fase do projeto, identificámos áreas que poderiam ter sido aperfeiçoadas, mas devido à gestão de múltiplos projetos de várias cadeiras em simultâneo, não conseguimos dedicar o tempo necessário para as melhorias desejadas.

Apesar disso, estamos satisfeitos com os resultados alcançados e com o conhecimento técnico adquirido sobre a cloud e os seus componentes relacionados. Este projeto foi uma ótima oportunidade para crescer as nossas competências e explorar uma "nova" tecnologia.