

Relatório técnico - **Especificação do produto**

# LogiPack

Unidade curricular: IES - Introdução à Engenharia de Software

Data: Aveiro, 17 de novembro de 2022

Alunos: 102435: Rafael Remígio  
102534: Rafael Gonçalves  
102536: Leonardo Almeida  
102778: Pedro Rodrigues

Sumário do projeto: Sistema de gestão de transportes de mercadorias, destinado às entidades que operam na área da Logística.

## CONTEÚDOS

### [1 Introdução](#)

### [2 Conceito do produto](#)

[Visão](#)

[Personas e suas motivações](#)

[Principais cenários](#)

### [3 Arquitetura](#)

[Requisitos chave e restrições](#)

[Vista geral da arquitetura](#)

[Interações entre os módulos](#)

### [4 Domínio do sistema de informação](#)

### [5 Referências e recursos](#)

# 1 Introdução

No âmbito da unidade curricular “Introdução à Engenharia de Software”, os alunos foram incumbidos de propor, conceptualizar e implementar uma solução de software *multi-layer*, de classe empresarial e com foco na web. Durante a execução do projeto, era expectável que cada grupo empregasse um vasto leque de frameworks e tecnologias, nomeadamente de gestão de projetos e controlo de versões, e que adotasse metodologias ágeis de desenvolvimento de software, o que pressupõe um trabalho colaborativo e iterativo.

Como ponto de partida, foram estipulados alguns requisitos funcionais a cumprir:

- Geração de *data streams*, que simulem o comportamento de sensores reais.
- Publicação de dados, usando um protocolo orientado a mensagens.
- Persistência dos dados.
- Processamento central da informação: além de *fetches* às tabelas da base de dados, esperava-se a criação de notificações de eventos e alguma agregação de dados, embora esse não fosse necessariamente o foco do trabalho.
- REST API.
- Portal web, onde seriam implementadas as *core user stories*.

Ao nível da gestão de projeto\*, aplicaram-se práticas comuns na cultura DevOps:

- **Agile backlog management (plan & track)**, assente nos seguintes conceitos:
  - *Backlog*, para organizar e priorizar o desenvolvimento;
  - *User story*, como unidade de planeamento;
  - Personas, como representantes de utilizadores finais da aplicação;
  - Sprints/iteraões, que produzem novos incrementos de código.

Ferramenta utilizada: Atlassian Jira.

- **Feature-branching workflow:** o desenvolvimento foi orientado a branches, pelo que cada feature-branch reflete uma user story. Além disso, com vista a minimizar conflitos entre segmentos de código desenvolvidos em paralelo e a garantir a sua qualidade, a integração (*merge*) de novas funcionalidades na branch principal (main) foi quase sempre antecedida por um *pull request*, cuja aprovação era da responsabilidade de um outro colega de equipa, habitualmente o DevOps master.

Ferramentas utilizadas: Git, GitHub.

- **Containers-based deployment:** encapsulamento das camadas da aplicação em *containers*, com as suas dependências e configurações, o que facilita a sua distribuição e execução em qualquer ambiente, incluindo o de produção.

Ferramenta utilizada: Docker.

- **Continuous deployment:** com recurso a uma *pipeline*, poucos minutos depois de um *push* no repositório remoto, o código é automaticamente atualizado e compilado no ambiente de produção (máquina virtual fornecida pelos docentes).

Ferramentas utilizadas: Github Actions, IMAP *listener* (Python).

\* Tópico aprofundado na apresentação.

## 2 Conceito do produto

### Visão

A transformação digital dos vários setores da economia é inegável. Com a crescente facilidade de acesso à tecnologia, as empresas tiveram de se adaptar rapidamente a um novo paradigma, tendo-se registado um crescimento de negócios eCommerce. Os sistemas de informação emergentes não se destinam apenas a clientes finais, como também a entidades prestadoras de serviços ou fornecedoras de produtos. Concretamente na área da Logística, as plataformas digitais podem ajudar as empresas a gerir melhor as suas cadeias de abastecimento, aumentando a eficiência e a eficácia das suas operações.

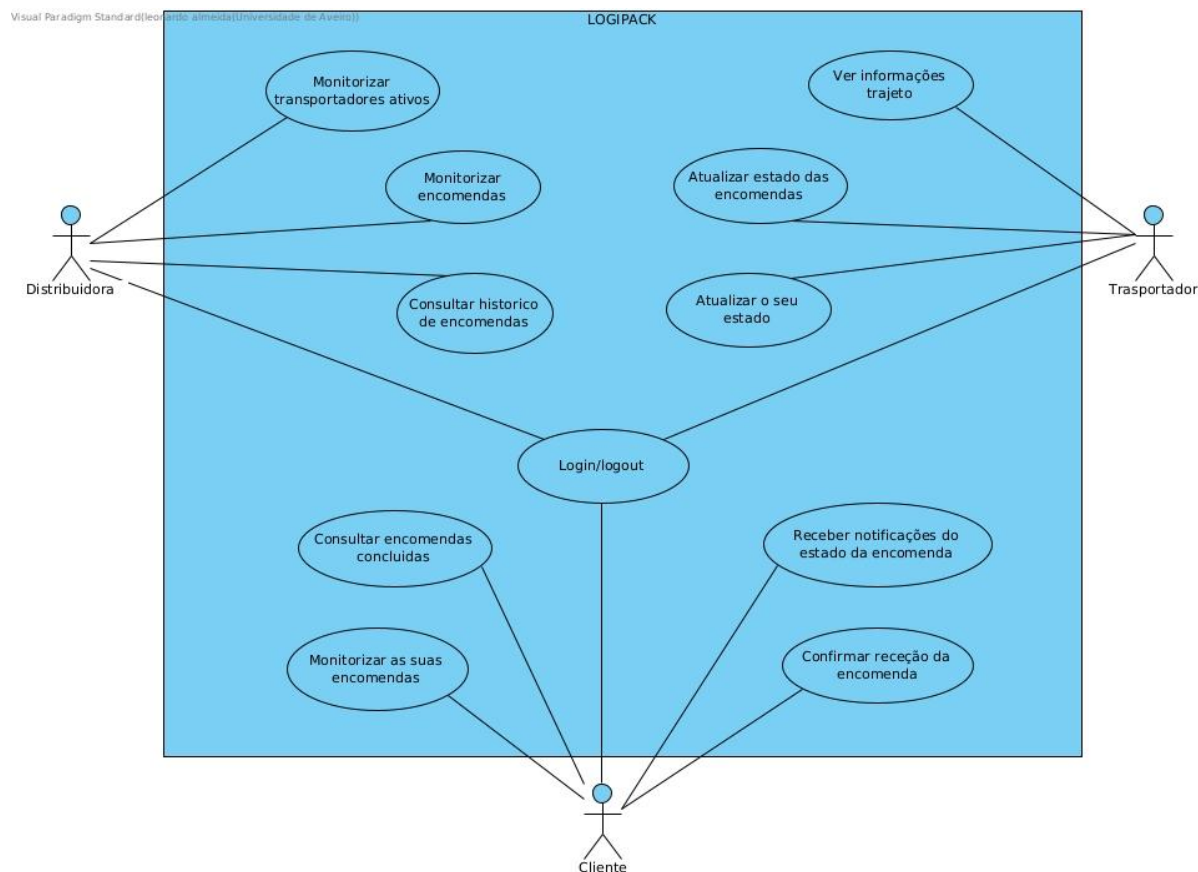
Posto isto, o grupo identificou vários benefícios em trazer este setor para o mundo digital, tais como:

- Automação de tarefas e processos, como rastreio de encomendas, gestão de inventário e registo de eventos em *logs*.
- Acompanhamento em tempo real do movimento das mercadorias, permitindo às empresas tomar decisões mais informadas.
- Melhor comunicação e colaboração entre os diferentes intervenientes na cadeia de abastecimento, como fabricantes, retalhistas e distribuidores.
- Redução de custos, através da otimização de processos e diminuição de riscos.
- Maior competitividade das empresas que utilizam os tais sistemas de informação.

Atualmente, já existem alguns sistemas com objetivos semelhantes. Contudo, são exclusivos das suas redes de distribuição, não representando uma solução genérica. A LogiPack é uma solução geral, passível de ser integrada em qualquer cadeia de abastecimento, melhorando o serviço e gerando valor para as empresas envolvidas.

Para uma visão mais clara da nossa proposta, foi desenvolvido um *UML Use Case Diagram*, onde são apresentadas as funcionalidades do sistema.

Por outro lado, de modo a realçar as motivações do utilizador final para usar o sistema, foram definidas *User Stories*.



### Distribuidora

Use case	User Story
Login	<b>Como</b> administrador, <b>quero</b> iniciar sessão na aplicação, <b>para</b> poder aceder às funcionalidades de administração.
Logout	<b>Como</b> administrador, <b>quero</b> encerrar a sessão da aplicação, <b>para</b> poder sair da mesma.
Monitorizar transportadores ativos	<b>Como</b> administrador, <b>quero</b> ver o estado (movimento, localização, encomendas, etc.) dos transportadores que estão ativos, <b>para</b> ver se a distribuição está a decorrer como esperado.
Monitorizar encomendas	<b>Como</b> administrador, <b>quero</b> ver o estado (veículo, transportador, início, etc.) das encomendas a entregar, <b>para</b> ver se a distribuição está a decorrer como esperado.
Consultar histórico de encomendas	<b>Como</b> administrador, <b>quero</b> ver o histórico de encomendas <b>para</b> efetuar uma gestão inteligente das frotas.

**Transportador**

Use case	User Story
Login	<b>Como</b> transportador, <b>quero</b> iniciar sessão na aplicação, <b>para</b> poder aceder às funcionalidades de transportador.
Logout	<b>Como</b> transportador, <b>quero</b> encerrar a sessão na aplicação, <b>para</b> poder sair da mesma.
Ver informações do trajeto	<b>Como</b> transportador, <b>quero</b> ver os detalhes do trajeto que me foi atribuído (pontos de entrega, tempo restante, etc), <b>para</b> poder gerir da melhor forma o processo de entrega.
Atualizar estado das encomendas	<b>Como</b> transportador, <b>quero</b> atualizar o estado das encomendas, <b>para</b> que os outros intervenientes na cadeia de abastecimento conheçam o estado das encomendas.
Atualizar o seu estado	<b>Como</b> transportador, <b>quero</b> atualizar o meu estado, <b>para</b> dar conhecimento do meu progresso a intervenientes com acesso para o verem.

**Cliente**

Use case	User Story
Login	<b>Como</b> cliente, <b>quero</b> iniciar sessão na aplicação, <b>para</b> poder aceder às funcionalidades do cliente.
Logout	<b>Como</b> cliente, <b>quero</b> encerrar a sessão na aplicação, <b>para</b> poder sair da mesma.
Receber notificações do estado da encomenda	<b>Como</b> cliente, <b>quero</b> receber notificações do estado da encomenda, <b>para</b> ter conhecimento em tempo real da sua expedição e distribuição.
Confirmar a receção de uma encomenda	<b>Como</b> cliente, <b>quero</b> confirmar a receção de uma encomenda, <b>para</b> que os intervenientes na cadeia de distribuição tenham conhecimento do desfecho da entrega.
Monitorizar as suas encomendas	<b>Como</b> cliente, <b>quero</b> monitorizar as minhas encomendas, <b>para</b> poder ter conhecimento dos seus estados atuais, assim como todos os anteriores.
Consultar encomendas concluídas	<b>Como</b> cliente, <b>quero</b> consultar encomendas concluídas <b>para</b> consultar os seus detalhes.

## Personas e suas motivações

Quando se segue uma abordagem ágil no desenvolvimento de software, o levantamento de requisitos acarreta habitualmente a criação de *personas*. Estas consistem em entidades fictícias que representam um determinado grupo de utilizadores finais do sistema, surgindo acompanhadas de características realistas, contexto, motivações e objetivos.

Como tal, foram descritas 3 personas, associadas a 3 grupos distintos de utilizadores finais do nosso sistema de informação:

1. Administrador da distribuidora
2. Transportador
3. Cliente final

### Administrador da distribuidora

<b>Nome</b>	Daniel Santos
<b>Idade</b>	35 anos
<b>Sexo</b>	Masculino
<b>Profissão</b>	Gestor de operações ( <i>Head of Operations</i> ) <u>Setor:</u> Logística
<b>Estado civil</b>	Casado
<b>Localização</b>	Lisboa



Contexto:

O Daniel Santos é o responsável pela gestão das operações de uma média empresa de distribuição. Tem mais de 10 anos de experiência no setor da Logística e diferencia-se pela organização e atenção aos detalhes. Como valoriza a eficiência e a satisfação dos clientes, está constantemente à procura de melhorar os processos da empresa.

Motivação:

Deseja acompanhar as frotas de entrega, coordenar os transportadores e atender às necessidades dos clientes, com celeridade, a fim de manter a boa reputação dos serviços da distribuidora. Além disso, quer ter acesso a estatísticas em tempo real, de forma a tomar decisões informadas sobre as operações da empresa.

## Transportador

<b>Nome</b>	José Carlos
<b>Idade</b>	51 anos
<b>Sexo</b>	Masculino
<b>Profissão</b>	Camionista
<b>Estado civil</b>	Solteiro
<b>Localização</b>	Viseu



### Contexto:

O José Carlos é condutor profissional de veículos pesados de mercadorias. Iniciou a sua carreira muito novo e esteve em três transportadoras diferentes. Conta com 2 anos de trabalho na sua empresa atual e, apesar da sua vasta experiência, continua a querer provar o seu valor. É um elemento confiável, que se orgulha de ser pontual, no cumprimento das suas funções.

### Motivação:

Deseja obter instantaneamente informações sobre os percursos que lhe são atribuídos, sem precisar de fazer chamadas telefónicas ou de se deslocar até à sede da transportadora. Deste modo, verificará se está a atender às expetativas da sua empresa e dos clientes. Auxiliando-se nos dados do sistema, poderá ainda reportar mais rapidamente qualquer problema ou atraso, minimizando os transtornos.

## Cliente

<b>Nome</b>	Maria Soares
<b>Idade</b>	25 anos
<b>Sexo</b>	Feminino
<b>Profissão</b>	Merceeira
<b>Estado civil</b>	Solteira
<b>Localização</b>	Portimão



### Contexto:

A Maria Soares é licenciada em Gestão e dona de uma pequena mercearia, juntamente com a sua mãe. Desde criança, contacta com tecnologias e tem vindo a alertar a sua família para o impacto positivo das plataformas digitais, nos mais variados negócios.

**Motivação:**

Deseja expandir o negócio da sua família e oferecer aos fregueses produtos frescos e de qualidade. Sabe que as suas aspirações implicam um maior volume de dados para registar e processar, não sendo possível fazê-lo com lápis e papel. Com a Logipack, conseguirá rastrear os produtos encomendados e manter automaticamente o seu histórico.

**Principais cenários**

Os cenários relatam concretamente como um sistema será utilizado, o que pode ajudar os *stakeholders* a compreender melhor as necessidades e expectativas dos utilizadores finais. Adicionalmente, permitem identificar potenciais problemas ou lacunas nas funcionalidades do sistema, servindo como base para validar e priorizar requisitos.

**Daniel Santos (administrador da distribuidora)**

O Daniel está a preparar-se para um dia atarefado, no centro de distribuição. Inicia sessão na LogiPack, para consultar a lista de encomendas, prontas a ser expedidas. Constatou que há uma carga adicional, que não estava prevista no início da semana, e que o sistema atribuiu automaticamente uma nova rota a um transportador inativo.

**José Carlos (transportador)**

José Carlos dirige-se para um ponto de entrega de mercadorias, conforme a rota que lhe foi atribuída. Após 4 horas a conduzir sem paragens, o que contradiz as recomendações para longos trajetos, José Carlos começa a ficar cansado e decide parar. Inicia sessão na LogiPack e atualiza o seu estado.

**Maria Soares (cliente)**

Maria Soares aguarda a entrega de frutas e legumes, para abastecer a sua mercearia. Enquanto organiza as prateleiras, recebe uma notificação de expedição da encomenda. Inicia sessão na LogiPack e conclui que há um atraso, pois os produtos foram expedidos 1 dia depois do registo no sistema.



## 3 Arquitetura

### Requisitos chave e restrições

- Validação dos utilizadores e posterior autorização/negação de acesso à API (configurações de segurança).
  - Solução: autenticação por *token*, com Spring Security. Quando um utilizador inicia sessão na aplicação, o *Authorization Controller* devolve um *token*, que deve ser incluído nos cabeçalhos dos pedidos à API subsequentes.
- Distinção de três grupos de utilizadores, na camada de serviços.
- Armazenamento persistente de dados correlacionados.
  - Solução: base de dados relacional (PostgreSQL); Spring Data JPA.
- Receção de novos estados das encomendas/transportadores em tempo real.
- Sistema *Append-Only Log* com o histórico dos estados das encomendas.
  - Solução: base de dados orientada ao documento (MongoDB). É *schemaless*, o que a torna mais adequada para funções analíticas em tempo real, e permite operações de escrita muito rápidas.
- Interface de utilizador na forma de aplicação web (e também aplicação móvel).

### Vista geral da arquitetura

#### Frontend

A arquitetura do sistema pretende responder aos requisitos chave supracitados. A **WebApp** (ou a *MobileApp*) comunica com uma RESTful API (com REST *controllers*), para consultar ou manipular recursos, que são armazenados na base de dados.

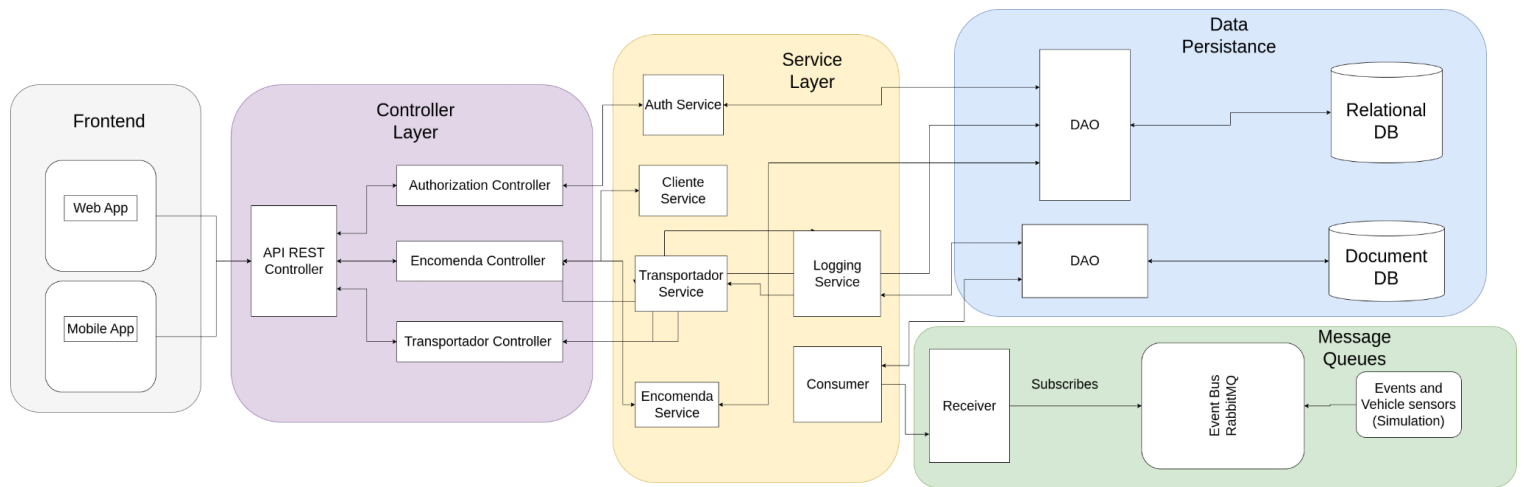
#### Backend

- Camada de controlo (*boundary*) - a "fronteira" com a camada de apresentação, isto é, com o frontend. É responsável por receber pedidos e enviar respostas, no protocolo HTTP.
- Camada de serviços (*business logic*): é responsável por processar as solicitações e gerar as respostas. É aqui onde se encontra a lógica de negócio da aplicação e onde são injetadas as suas dependências.
- Camada de persistência de dados (*persistence layer*): responsável por armazenar e descarregar conteúdo da base de dados.

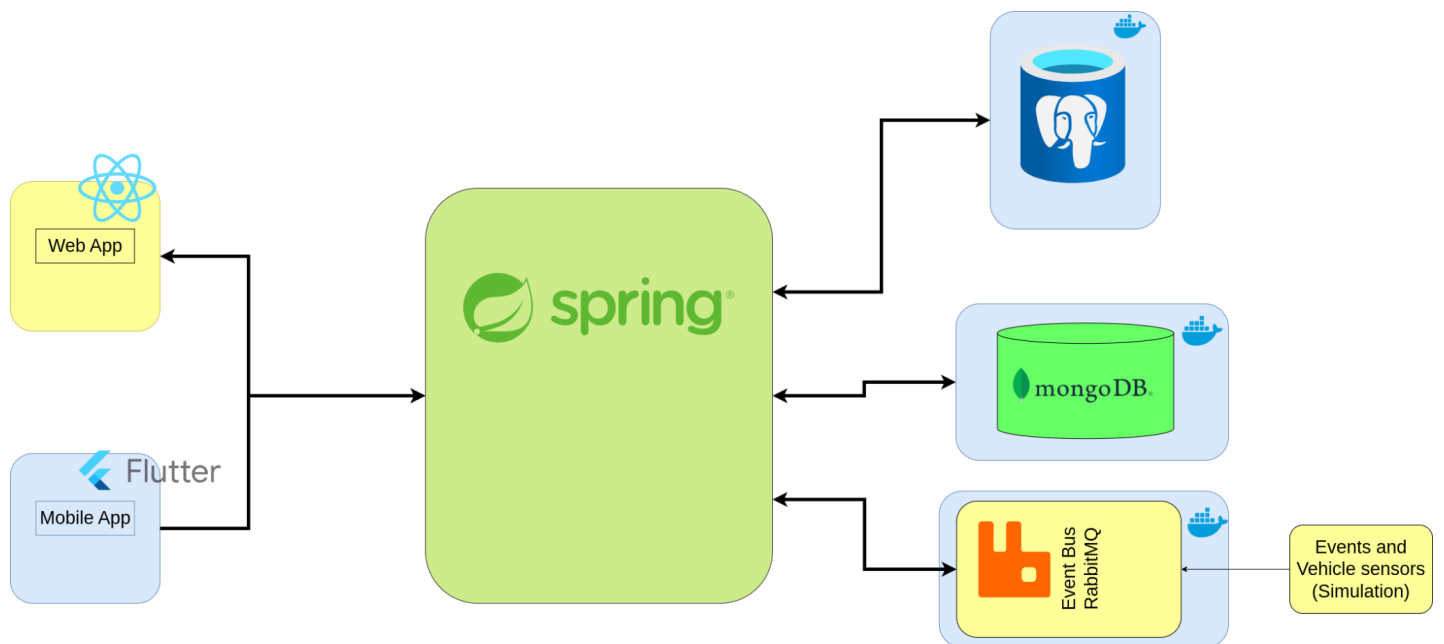
#### Gerador de dados

Para receber e processar eventos gerados por um simulador, é adequada a utilização de um **message broker** (RabbitMQ), que implemente um protocolo de filas de mensagens. Com este módulo, assente no padrão *publish-subscribe*, a arquitetura do sistema é claramente orientada a eventos. O Spring assume também um papel relevante nesta camada da aplicação, pois contém um componente responsável por ler e registar o conteúdo da **event bus** na base de dados.

## Arquitetura



## Tecnologias Usadas



## Interações entre os módulos

- WebApp/MobileApp (frontend) ↔ SpringBoot (backend): pedidos e respostas HTTP, com métodos GET, POST, PUT e DELETE.
- SpringBoot (backend) ↔ bases de dados
  - Especificação (conjunto de interfaces): Jakarta Persistence API (JPA).
  - Implementação: Hibernate (por *default*, no SpringBoot).
- SpringBoot (backend) ↔ simulador

Os dados são publicados num event bus do RabbitMQ, ao qual o serviço de *logging* (associado ao MongoDB) está subscrito.

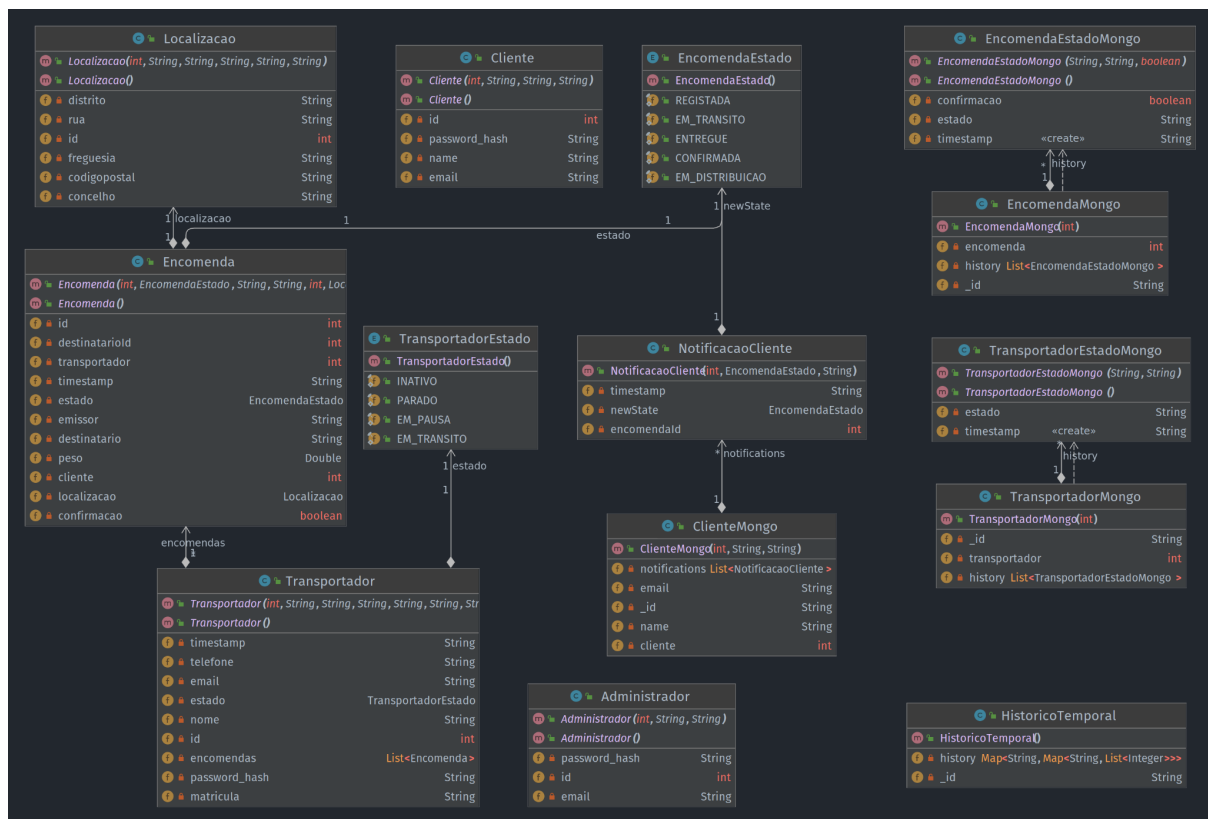
## 4 Perspetiva de informação

O sistema desenvolvido apresenta dois perfis distintos com necessidades distintas:

1. Perfil onde os dados apenas necessitam de ser armazenados, de modo a serem acedidos posteriormente através da REST API.
2. Perfil que exige uma inserção contínua de dados de um mesmo tipo, isto é, pertencentes a uma mesma entidade.

Com o objetivo de responder a estes dois perfis, houve a necessidade de recorrer a duas bases de dados distintas, uma relacional (PostgreSQL) e outra não relacional (MongoDB).

As diferentes entidades modeladas encontram-se representadas a seguir:



## 5 Referências e recursos

<https://evincedev.com/blog/transport-logistics-digital-transformation/>

<https://lvivivity.com/digital-transformation-in-logistics>

<https://javatodev.com/spring-boot-jwt-authentication/>

<https://github.com/features/actions>

<https://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api>