

# TQS: Relatório de especificação do produto

## Conceito do Produto

A **ElectraNet** é uma empresa de tecnologia visionária focada na otimização do ecossistema de veículos elétricos (VE) através da sua plataforma de ponta, que assegura uma interoperabilidade de carregamento perfeita. A empresa aborda um dos principais obstáculos à adoção de VEs — os serviços de carregamento fragmentados — oferecendo uma solução digital unificada que integra funcionalidades em tempo real de pesquisa, reserva e pagamento, cobrindo diversas redes de carregamento.

## Atores principais

- **EV Driver** searches for chargers, books slots, charges the vehicle, makes payments, plan trips,...
- **Station Operator** manages charging station availability and maintenance, monitor daily operations, configure off-peak charging slots and discounts,...
- **Charging station API** (criada por nós): Interagir com os postos de carregamento.
- **Payment services API** (vendedor a definir)
- **\*\*API de mapas geográficos** (\*\*vendedor a definir) (menor prioridade?)

## Limitacoes

### Caso de Uso 5: Efetuar pagamento de carregamento – Estado Atual

Este caso de uso ainda **não está implementado** na plataforma ElectraNet. A funcionalidade prevê o cálculo automático do valor a pagar após o carregamento, seleção do método de pagamento pelo utilizador e integração com uma API de pagamentos.

A funcionalidade será considerada em fases futuras, após definição de requisitos técnicos e de segurança.

## Personas

### Persona 1: João Silva — Condutor de Veículo Elétrico

João Silva tem 38 anos e trabalha como consultor de Tecnologias de Informação em Lisboa. Com um nível avançado de conhecimento tecnológico, ele utiliza um veículo elétrico para as suas deslocações diárias e, para ele, a experiência de condução está diretamente ligada à conveniência e eficiência do processo de carregamento do seu carro.

#### Objetivos:

- Encontrar postos de carregamento disponíveis em qualquer cidade.
- Reservar carregamentos com antecedência.
- Evitar filas ou postos fora de serviço. - Obter comprovativos de pagamento para reembolso empresarial.
- Ver rotas para o ponto que escolheu

#### Frustrações:

- Chegar a postos ocupados ou inativos.
- Ter de usar várias aplicações de diferentes operadores.
- Falta de transparência nos preços.
- Processos de pagamento pouco claros ou inconsistentes.

A **ElectraNet** oferece ao **João Silva** uma plataforma única que integra diferentes redes de carregamento, sistema de reserva e pagamento direto, além de integração com GPS e sugestões de rotas. Isso proporciona uma experiência mais eficiente e sem surpresas.

## **Persona 2: Pedro Gonçalves — Operador de Estação de Carregamento**

Pedro Gonçalves, de 45 anos, é gestor de infraestruturas de carregamento no Porto e tem um nível intermediário de conhecimento tecnológico.

### **Objectivos:**

- Monitorizar o estado dos postos em tempo real
- Maximizar a utilização das estações.
- Monitorizar o tempo de inatividade dos equipamentos.
- Promover carregamentos fora das horas de ponta com descontos.
- Garantir uma boa experiência ao utilizador final.

### **Frustrações:**

- Só tem conhecimento de avarias após reclamações dos clientes.
- Equipamentos permanecem inativos por longos períodos.
- Dificuldade em gerir múltiplos pedidos de assistência.
- Processo de manutenção moroso e manual.

A ElectraNet oferece a Pedro uma solução eficiente com um painel de controlo que permite acompanhar o status dos postos em tempo real, além de uma ferramenta para configurar preços variáveis conforme o horário, promovendo carregamentos fora das horas de pico. A plataforma também automatiza e centraliza os pedidos de manutenção, melhorando a gestão das infraestruturas e a experiência para os utilizadores.

### Persona 3: Ana Ribeiro — Administradora de Sistema

Ana Ribeiro, de 34 anos, é administradora de sistemas e DevOps em Braga, com um elevado nível tecnológico.

#### Objectivos

- Assegurar a disponibilidade e segurança da plataforma.
- Gerir acessos e permissões de diferentes utilizadores.
- Monitorizar a performance e integridade dos serviços em tempo real.
- Coordenar atualizações e manutenção sem interrupção do serviço.

#### Frustrações:

- Problemas de escalabilidade durante picos de uso.
- Gestão complexa de integrações com APIs externas.
- Dificuldade em rastrear erros em tempo útil.
- Falta de visibilidade unificada sobre o estado da infraestrutura.

A **ElectraNet** responde a estas necessidades com um **painel de monitorização centralizado e alertas proativos**, integração de logs e métricas com ferramentas de observabilidade, **gestão avançada de utilizadores** e **ferramentas CI/CD** que permitem atualizações contínuas sem causar paragens.

## Use cases

### Caso de Uso 1: Procurar e reservar posto de carregamento

**Ator principal:** Condutor de VE (João Silva)

**Atores secundários:** API de mapas, Sistema de reservas

**Descrição:** O condutor localiza postos de carregamento disponíveis, filtra por critérios, e reserva um posto.

**Pré-condições:**

- O condutor tem conta ativa e sessão iniciada na app.
- A localização do utilizador ou destino é conhecida.
- A plataforma está conectada às APIs de mapas e postos.

**Fluxo principal:**

1. O condutor abre a funcionalidade de pesquisa.
2. A app mostra os postos mais próximos com base na localização.
3. O condutor aplica filtros (tipo de conector, disponibilidade).
4. Seleciona um posto disponível.
5. Reserva o posto para o horário desejado.
6. Recebe confirmação e código de reserva.

**Exceções:**

- Posto fica indisponível antes da reserva ser concluída → mensagem de erro e sugestão de alternativa.
- Falha na comunicação com a API de mapas → tentativa de reenvio ou modo offline limitado.

**Pós-condições:**

- Reserva é criada com validade limitada.
- Condutor pode visualizar a reserva e navegar até o posto.

## Caso de Uso 2: Rotas Posto

**Ator principal:** Condutor de VE

**Atores secundários:** API de mapas, Sistema de rotas

**Descrição:** O condutor concede a utilização do serviço de localização para o cálculo de rotas para o posto mais próximo

**Pré-condições:**

- Sessão iniciada.
- Conectividade com API de mapas.

**Fluxo principal:**

1. O condutor autoriza a partilha da sua localização.
2. A ElectraNet calcula a rota para o posto mais próximo.

## Caso de Uso 3: Monitorizar e configurar uma estação

**Ator principal:** Operador da estação

**Atores secundários:** Sistema de monitorização da ElectraNet

**Descrição:** O operador acede ao painel para gerir o estado, configurar preços e gerir operações envolvidas com as estações.

**Pré-condições:**

- Acesso autorizado ao painel do operador.

**Fluxo principal:**

1. O operador inicia sessão e acede à lista de estações.
2. Controla as operações relacionadas com as estações, como mudar preço por hora , mudar estado , ver, editar reservas das estações

**Pós-condições:**

- Configurações aplicadas.

## **Caso de Uso 5: Efetuar pagamento de carregamento**

**Ator principal:** Condutor de VE

**Atores secundários:** API de pagamentos, Sistema ElectraNet

**Descrição:** O condutor realiza o pagamento por um carregamento.

**Pré-condições:**

- Carregamento concluído.
- API de pagamentos está funcional.

**Fluxo principal:**

1. A app calcula o total com base no tempo e kWh utilizados.
2. O condutor seleciona o método de pagamento.
3. A app envia os dados à API de pagamentos.
4. A transação é processada e confirmada.
5. Recibo é emitido e associado à conta do utilizador.

**Exceções:**

- Pagamento recusado → notificação ao utilizador e pedido de outro método.

- API indisponível → opção de repetir tentativa ou pagar no local (se aplicável).

**Pós-condições:**

- Transação registada.
- Recibo disponível para consulta ou exportação.

**Epics****Pesquisa e Reserva de Postos de Carregamento**

**Origem:** Casos de uso 1 e 2

**Objetivo:** Permitir que o condutor encontre postos disponíveis, aplique filtros relevantes e reserve sessões de carregamento.

**Funcionalidades associadas:**

- Localização e visualização de postos num mapa.
- Filtros por tipo de conector, potência, rede, etc.
- Reserva de sessões de carregamento com confirmação.
- Sugestão de alternativas em caso de falha.

**Rotas de Viagem**

**Origem:** Caso de uso 2

**Objetivo:** Oferecer ao condutor um serviço de rotas personalizado sugeridos com base na proximidade com a estação mais próxima.

**Funcionalidades associadas:**

- Cálculo de rota otimizada.
- Sugestão de postos mais próximo, e disponibilização no mapa.



## Gestão e Pagamento de Sessões de Carregamento

**Origem:** Caso de uso 5

**Objetivo:** Calcular e processar pagamentos de sessões, com integração em APIs de pagamento seguras.

### Funcionalidades associadas:

- Cálculo automático do custo de carregamento.
- Seleção de método de pagamento
- Processamento e confirmação da transação.
- Emissão de recibos e armazenamento histórico.
- Gestão de falhas e pagamentos alternativos.

## Gestão de Estações por Operadores

**Origem:** Caso de uso 3

**Objetivo:** Permitir aos operadores configurarem, monitorizarem e manterem as estações de carregamento.

### Funcionalidades associadas:

- Painel de gestão para operadores.
- Visualização do estado operacional das estações.
- Agendamento de manutenção.
- Configuração de tarifas e promoções.
- Geração de relatórios operacionais.

## Administração do Sistema

**Origem:** Caso de uso 4

**Objetivo:** Fornecer ao administrador ferramentas para gerir utilizadores, permissões, logs e integrações.

**Funcionalidades associadas:**

- Gestão de contas e permissões.
- Monitorização de logs do sistema.
- Integração de novas APIs (mapas, pagamentos, etc.).
- Gestão de atualizações e manutenção programada.
- Políticas de segurança e registo de auditoria.

# Arquitetura do Sistema

## Principais requisitos e restrições

Sendo o ElectraNet uma aplicação nova desenvolvida do zero, não existem restrições em termos de arquitetura. Isso permite-nos projetar o sistema utilizando ferramentas modernas, frameworks atuais e linguagens de programação adequadas.

Para simplificar a experiência de desenvolvimento e garantir que o sistema possa ser executado em qualquer máquina, serão utilizados **Docker Containers**. Estes permitem iniciar ambientes isolados prontos a executar o código da aplicação em qualquer sistema compatível.

Dado que o ElectraNet envolve diferentes entidades com papéis distintos na aplicação, é essencial aplicar mecanismos de autenticação, garantindo a segurança e integridade das operações.

Além disso, o sistema deverá assegurar um bom nível de desempenho, mesmo sob situações de carga elevada, considerando o seu domínio de utilização.

## Visão da arquitetura

O **ElectraNet** segue uma arquitetura clássica de **frontend-backend**, sem necessidade de middleware adicional.

Para gerir os pedidos HTTP e os caminhos de acesso, vamos utilizar o **NGINX** como serviço de proxy reverso. Este componente é responsável por receber as requisições, analisar os seus metadados e redirecioná-las para o módulo correto da aplicação.

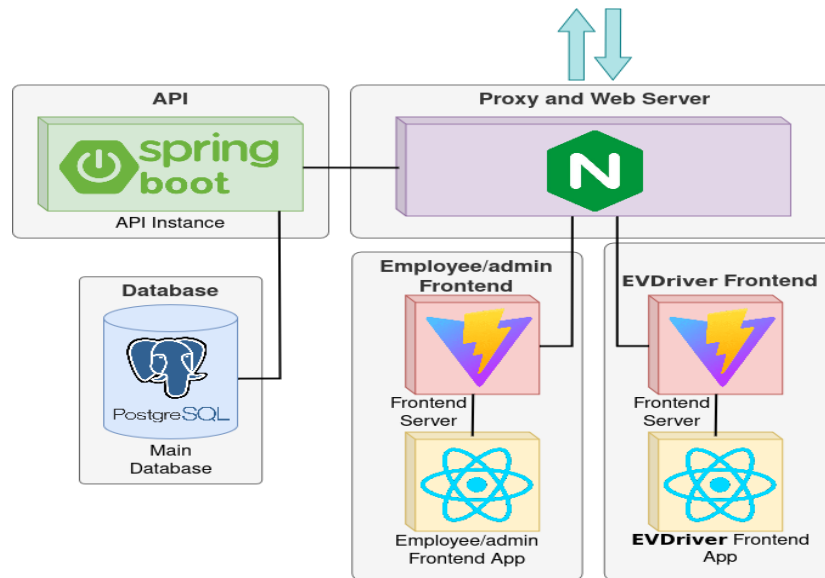
O **backend** é desenvolvido com **Spring Boot**, o que permite um desenvolvimento rápido, estruturado e com fácil integração de ferramentas de teste, bibliotecas externas (como sistemas de logging e inspeção de código), bem como bases de dados em memória para caching. Este módulo trata da lógica de negócio, autenticação e comunicação com a base de dados.

A base de dados escolhida é o **PostgreSQL**, devido à sua integração eficiente com o JPA do Spring Boot, baixo custo computacional, natureza open source e facilidade de execução via **Docker**, o que simplifica a configuração e acelera o processo de desenvolvimento.

O **frontend** é dividido em duas interfaces principais, ambas desenvolvidas em **React**:

- **Interface do Utilizador Final** (clientes normais do sistema).
- **Interface de Gestão** (para trabalhadores e administradores da plataforma).

Ambas as interfaces são servidas com **ViteJS**, que garante uma compilação rápida e eficiente, e facilita a integração com React. A escolha do React baseia-se na sua popularidade, ecossistema maduro e suporte extensivo da comunidade.



A aplicação comunica principalmente através de pedidos HTTP, o que simplifica a análise e o tratamento de erros. As interfaces do condutor de veículo elétrico (EVdriver) e do funcionário/administrador enviam pedidos HTTP para o servidor NGINX, que os encaminha para o backend. O backend trata e persiste os dados recebidos, respondendo de seguida ao NGINX, que os reencaminha para o cliente correspondente.

## API para Desenvolvedores

Método	Endpoint	Descrição	Headers / Body / Query Params
POST	<code>https://deti-tqs-17.ua.pt:8080/auth/login</code>	Autenticação de utilizador	<ul style="list-style-type: none"> <li><b>Query Params:</b> <ul style="list-style-type: none"> <li><code>username</code>: <code>string</code></li> <li><code>password</code>: <code>string</code></li> <li><b>Resposta:</b> JWT Token</li> </ul> </li> </ul>
POST	<code>https://deti-tqs-17.ua.pt:8080/auth/register</code>	Registo de novo utilizador	<ul style="list-style-type: none"> <li><b>Body (JSON):</b> <pre>json&lt;br&gt;{&lt;br&gt;  "firstName": "string",&lt;br&gt;  "lastName": "string",&lt;br&gt;  "email": "string",&lt;br&gt;  "phone": number,&lt;br&gt;  "password": "string"&lt;br&gt;}</pre> </li> </ul>
GET	<code>https://deti-tqs-17.ua.pt:8080/api/getAllStations</code>	Obter todas as estações disponíveis	<ul style="list-style-type: none"> <li><b>Headers:</b> <ul style="list-style-type: none"> <li><code>Authorization</code>: Bearer Token</li> </ul> </li> </ul>
GET	<code>https://deti-tqs-17.ua.pt:8080/reserva/getAll</code>	Obter todas as reservas no sistema	<ul style="list-style-type: none"> <li><b>Headers:</b> <ul style="list-style-type: none"> <li><code>Authorization</code>: Bearer Token</li> </ul> </li> </ul>
POST	<code>https://deti-tqs-17.ua.pt:8080/reserva/fazerReserva</code>	Criar uma nova reserva de carregamento	<ul style="list-style-type: none"> <li><b>Headers:</b> <ul style="list-style-type: none"> <li><code>Authorization</code>: Bearer Token</li> </ul> </li> <li><b>Query Params:</b> <ul style="list-style-type: none"> <li><code>stationID</code>: <code>long</code></li> <li><code>dataReserva</code>: <code>dd/MM/yyyy</code></li> <li><code>horaReserva</code>: <code>HH:mm</code></li> <li><code>duracaoReserva</code>: <code>int</code> (horas)</li> <li><code>tipoCarregamento</code>: <code>string</code></li> </ul> </li> </ul>
PUT	<code>https://deti-tqs-17.ua.pt:8080/reserva/manageReserva</code>	Gerir estado de uma reserva (ex: cancelar, concluir)	<ul style="list-style-type: none"> <li><b>Headers:</b> <ul style="list-style-type: none"> <li><code>Authorization</code>: Bearer Token</li> </ul> </li> <li><b>Query Params:</b> <ul style="list-style-type: none"> <li><code>idReserva</code>: <code>long</code></li> <li><code>operation</code>: <code>string</code> (<code>cancelar</code>, <code>concluir</code>, etc.)</li> </ul> </li> </ul>

A nossa plataforma disponibiliza uma API REST completa para que os desenvolvedores possam integrar e expandir funcionalidades de forma fácil e segura. Com os nossos endpoints, podem gerir utilizadores, autenticar sessões, consultar estações de carregamento e criar ou gerir reservas de forma programática.

Todos os pedidos que envolvem dados sensíveis ou ações restritas requerem autenticação via token JWT, garantindo segurança e controlo de acesso.

Se estão a desenvolver aplicações móveis, web ou sistemas de gestão ligados a carregamento de veículos elétricos, a nossa API é uma solução robusta para automatizar processos e criar experiências personalizadas para os utilizadores.

Sintam-se à vontade para explorar os endpoints, testar as funcionalidades e integrar a API nas vossas aplicações. Estamos disponíveis para ajudar com dúvidas ou suporte técnico para garantir uma integração suave e eficiente.

