TQS: Product specification report

*Gonçalo Abrantes[104152] João Morais [103730] Nuno Faria[112994] Pedro Rei [107463]*

v2025-05-16

[**1 Introduction 1**](#_heading=h.meda4as4dwr4)

[1.1 Overview of the project 1](#_heading=h.ym6k3jb1na43)

[1.2 Known limitations 2](#_heading=h.d112pyxfi1u1)

[1.3 References and resources 2](#_heading=h.gfu7xo5vz5d5)

[**2 Product concept and requirements 2**](#_heading=h.5d9lvj3mlpyj)

[2.1 Vision statement 2](#_heading=h.6knokaazcpai)

[2.2 Personas and scenarios 2](#_heading=h.wmqdkqewqrp4)

[2.3 Project epics and priorities 3](#_heading=h.36xndxb9g1vi)

[**3 Domain model 3**](#_heading=h.pzprxh3ko0id)

[**4 Architecture notebook 4**](#_heading=h.1x4s81h3qape)

[4.1 Key requirements and constrains 4](#_heading=h.e1y67v6wu2s9)

[4.2 Architecture view 4](#_heading=h.1l2oc93oyl64)

[4.3 Deployment view 4](#_heading=h.z4e0njqbzxrj)

[**5 API for developers 4**](#_heading=h.7yyxu9hs0df8)

[This report is the main source of technical documentation on the project, clarifying the functional scope and architectural choices. Provide concise, but informative content, **allowing other software engineers to understand the product**.

Tips on the expected content placed along the document are meant to be removed!

You may use English or Portuguese; do not mix.]

# Introduction

## Overview of the project

No âmbito da nossa cadeira de TQS, este projecto tem como objectivo aplicar os conhecimentos adquiridos no desenvolvimento de uma solução digital robusta, funcional e centrada no utilizador. O foco esteve na criação de um produto que resolvesse um problema real, recorrendo a boas práticas de engenharia de software, desde o levantamento de requisitos até à validação da solução. A iniciativa focou-se na aplicação de princípios fundamentais da engenharia de software, com especial destaque para as metodologias orientadas à qualidade, como o desenvolvimento orientado por testes (TDD) e os processos de Garantia da Qualidade (QA).

O resultado dos nossos esforços é a nossa aplicação, que tem como seu objetivo facilitar o acesso a postos de carregamento para veículos eléctricos. Através de uma plataforma integrada, o utilizador pode localizar postos disponíveis em tempo real, efectuar reservas e realizar pagamentos de forma simples e eficiente. Com a nossa aplicação, nós pretendemos tornar a experiência de carregamento mais fluida, previsível e adaptada às exigências da mobilidade elétrica moderna.

Durante o desenvolvimento, adotou-se a abordagem TDD, onde os testes são escritos antes da implementação funcional. Esta metodologia permitiu garantir que o código fosse construído de forma incremental, com maior confiança na sua robustez e na detecção precoce de erros. Paralelamente, foram aplicados conceitos de QA como a validação contínua, testes unitários e testes de usabilidade, assegurando que o sistema final não só cumprisse os requisitos técnicos, mas também proporcionasse uma boa experiência ao utilizador.

<contextualize the objectives of this project assignment in the scope of the TQS course>

<introduce your application/product: brief overview of the solution concept. What is it good for? Introduce the name of the product if it has one>

## Known limitations

 <explain the known limitations, especially the features that were planned/expected but not implemented (and why…)

**To be reviewed and completed by the end of the project** >

## References and resources

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>

# Product concept and requirements

## Vision statement

Um dos principais desafios enfrentados por quem possui um veículo elétrico é a dificuldade em encontrar e utilizar estações de carregamento de maneira rápida, confiável e integrada. Em áreas urbanas, essa limitação acaba por comprometer a praticidade e a eficiência da mobilidade elétrica. Pensando nisso, o sistema desenvolvido busca justamente facilitar esse processo. A proposta é oferecer uma aplicação que reúna, numa única plataforma, recursos para localizar, reservar e pagar pelo uso das estações de carregamento, tornando a experiência do utilizador mais simples e eficiente.

Para atender a essa demanda, o sistema oferece diversas funcionalidades integradas. A primeira delas é a exibição em tempo real das estações de carregamento disponíveis, por meio de um mapa interativo que permite a aplicação de filtros como tipo de carregador e velocidade de carga. Outra funcionalidade importante é a possibilidade de reservar uma estação com antecedência, o que ajuda a evitar filas e otimizar o tempo do utilizador. O pagamento também pode ser feito diretamente através da aplicação, com suporte a diferentes métodos, como cartões de crédito e débito, além de carteiras digitais. As estações apresentam ainda uma seção com informações detalhadas, como o tipo de carregador, a potência oferecida, os horários de funcionamento, o status atual (ocupada ou disponível) e as avaliações feitas por outros usuários. Para completar, o sistema envia notificações automáticas, como lembretes sobre o início e o fim do carregamento, horários de reserva e alterações no status das estações, garantindo que o usuário esteja sempre informado.

Esta aplicação também terá acessos exclusivos a funcionários das estações de carregamento, de forma a serem capazes de saber em tempo real o estado das estações, como por exemplo, se precisam de manutenção. Além disso, também poderão ver a quantidade de remuneração que cada estação anda a trazer.

Embora existam outras plataformas populares no setor, como a PlugShare e a Electromaps, que também oferecem mapas com a localização de estações e informações técnicas, a aplicação destaca-se por integrar, de forma contínua, todas as etapas do processo num único ambiente. Enquanto outras soluções exigem que o utilizador recorra a diferentes ferramentas para localizar, reservar e pagar, o sistema proposto concentra tudo isso em um só lugar, proporcionando uma experiência mais prática e fluida.

<functional (black-box) description of the application: Which is the high-level/business problem being solved by your system? Which are the key features you promise to address it?>

<if needed, clarify what was planned/expected to be included but was changed to a different approach/concept >

<optional: how is your system different or similar to other well-known products?>

<optional: additional details on the process for the requirements gathering and selection (how did we developed the concept? Who helped us with the requirements? etc)>

## Personas and scenarios

Persona 1: **Joana Silva (Condutora de Veículos Elétricos)**

Idade: 32 anos

Naturalidade: Lisboa

Joana Silva é uma dona de casa, com dois filhos e um marido que trabalha. Como encarregada de educação, Joana vê-se a utilizar o carro elétrico dela para conduzir as crianças para a escola delas, tal como para as suas atividades extracurriculares e para quaisquer recados que ela tenha. Desta forma, o carro precisa sempre de ser carregado com frequência numa estação que existe em frente ao edifício em que a família vive.

Para garantir que consegue carregar, ela verifica sempre através da aplicação se existe alguma estação livre, e reserva online, de forma a poder recarregar o veículo e também precisa de uma forma de pagamento fácil.

Persona 2: **Carlos Mendes (Funcionário do Posto de carregamento)**

Idade: 45 anos

Naturalidade: Braga

Carlos Mendes é um funcionário dos Postos de carregamento de carros elétricos, tendo como função verificar os estados das estações e verificar se os níveis de consumo estão dentro do normal. Para poder realizar a sua função, ele precisará de um acesso exclusivo para funcionários, de forma que utilizadores normais não sejam capazes de aceder a informação potencialmente sensível. Este acesso para funcionários terá que ser capaz de fornecer informação acerca dos postes, quais estão disponíveis, e quais precisam de manutenção, e obter informação acerca do consumo agregado dos postos.

Persona 3: **Graça Gomes (Condutora)**

Idade: 25 anos

Naturalidade: Porto

Graça é uma estudante universitária, que teve a sorte de ganhar um carro elétrico num sorteio. No entanto, viver na residência de estudantes causa o único problema de ter que procurar por estações de carregamento para o seu carro. Além disso, a quantidade de dinheiro que ela tem para pagar não é sempre muito. Por estas razões, Graça procura juntar o útil ao agradável, e precisa de ser capaz de ver os custos das estações de carregamento perto dela, de forma a encontrar uma que seja perto da residência e que seja barato o suficiente para justificar a distância.

<“Personas are fictional people. They have names, likenesses, clothes, occupations, families, friends, pets, possessions, and so forth. They have age, gender, ethnicity, educational achievement, and socioeconomic status. They have life stories, goals and tasks. Scenarios can be constructed around personas, but the personas come first. They are not ‘agents’ or ‘actors’ in a script, they are people. Photographs of the personas and their workplaces are created and displayed. […] It is to obtain a more powerful level of identification and engagement that enable design, development, and testing to move forward more effectively”. Adapted from Grudin, J. and Pruitt, J., 2002, June. Personas, participatory design and product development: An infrastructure for engagement. In Proc. PDC (Vol. 2).

Sample personas: [secção 4.1, neste artigo](http://www.mdpi.com/1424-8220/18/4/1285) (open access)] >

<Develop one or more representative scenarios for each persona. You don’t need to include all possible details. Pick the main scenarios, related to the core value of the system.>

## Project epics and priorities

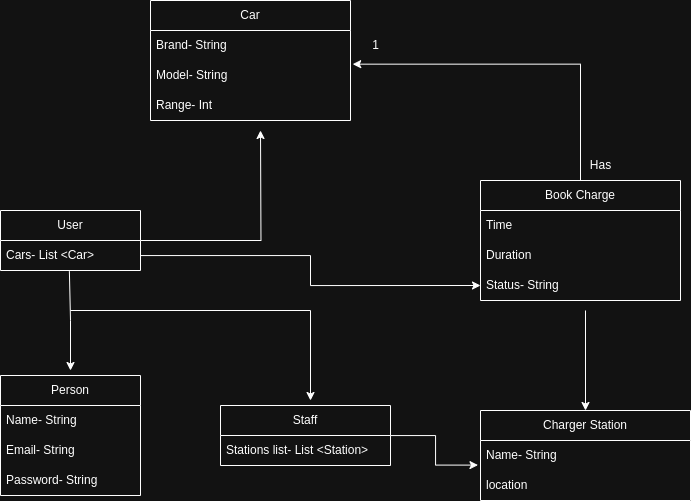
**[**Apresentar um plano indicativo para a implementação incremental da solução ao longo de várias iterações/releases, explicando as funcionalidades a atingir por [*epics*](https://www.atlassian.com/agile/project-management/epics-stories-themes) ]

Neste projeto, adotamos o Scrum com sprints semanais e reuniões de planeamento às sextas-feiras, às 16h. Nessas reuniões, definiremos as *issues* da semana, os respetivos critérios de aceitação e os *reviewers*. O fluxo de desenvolvimento seguirá o GitHub Flow: para cada nova *issue*, criaremos uma *feature branch* isolada a partir da *main branch*. Após a implementação, o programador garantirá a aprovação de todos os testes unitários e uma cobertura de código de pelo menos 80%, além de realizar uma análise com o SonarQube para identificar e corrigir problemas críticos. Uma *pull request* só será aceite na *main branch* após aprovação do *reviewer* designado no planeamento da *sprint*. Caso sejam identificados erros após a integração do código na branch principal, a responsabilidade pela correção recai sobre o programador que criou o Pull Request original.

# Domain model

<which information concepts will be managed in this domain? How are they related?>

<use a logical model (UML classes) to explain the concepts of the domain and their attributes, not a entity-relationship relational database model>



# Architecture notebook

## Key requirements and constraints

A aplicação terá uma interface intuitiva e de fácil utilização, dada a diversidade de perfis de utilizadores. Essencialmente, integrará dois serviços externos principais: um para o processamento de pagamentos e outro para fornecer um mapa atualizado da cidade. Por ser uma aplicação de carregamento, prevemos picos de utilização nas horas de almoço e de saída do trabalho, exigindo que o sistema responda eficientemente a múltiplos utilizadores simultaneamente. Dada a sua operação em múltiplos dispositivos móveis e computadores, o desenvolvimento terá especial atenção à responsividade do *design* para diferentes dimensões de ecrã.

<**Identify issues that will drive the choices for the architecture** such as: Are there hardware dependencies that should be isolated from the rest of the system? Does the system need to function efficiently under unusual conditions? Are there integrations with external systems? Is the system to be offered in different user-interfacing platforms (web, mobile devices, big screens,…)?

For a more systematical approach:

* Note the collection of [Architectural Characteristics](https://learning.oreilly.com/library/view/fundamentals-of-software/9781098175504/ch04.html) the software architect should be aware
* [Identify architectural characteristics](https://learning.oreilly.com/library/view/fundamentals-of-software/9781098175504/ch05.html) that are relevant for your project (will drive the key design decisions). Note the [case study](https://learning.oreilly.com/library/view/fundamentals-of-software/9781098175504/ch05.html#ch-identifying-ss) and the explicit characteristics related to users and extensibility. This will support later non-functional tests.

The application will be a web application and however it will is supposed to be used on a mobile so we will need to have some consideration regarding screen size and adjustment.

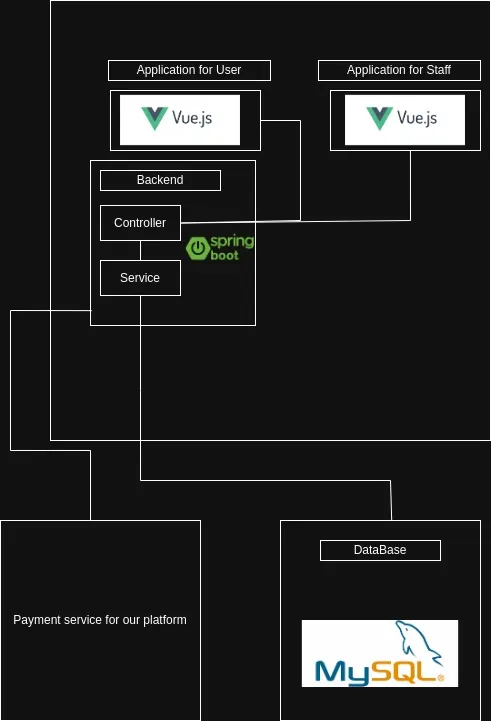
## Architecture view

→ Discuss architecture planned for the software solution: what are the main building blocks? [include a diagram](https://learning.oreilly.com/library/view/fundamentals-of-software/9781098175504/ch23.html) (a logical view, such as a package or block diagram). Avoid implementation technology or deployment references, but protocols/standards can be included.

→ refer to the [architecture style](https://learning.oreilly.com/library/view/fundamentals-of-software/9781098175504/part02.html) applied, if any

🡪 explain how the identified modules will interact. Use a sequence diagram to clarify the interactions along time, when needed

→ discuss more advanced app design issues: integration with Internet-based external services, data synchronization strategy, distributed workflows, push notifications mechanism, distribution of updates to distributed devices, etc.>



## Deployment view

[Explicar a organização prevista da solução em termos configuração de produção (*deployment*). Anotar, no diagrama, as tecnologias de implementação, e.g.: colocar o simbolo do PostgreSQL na Base de dados,…]. Indicar a existência de containers (Docker), endereços IP e portos,...

Esta parte será completada quando houver efetivamente deployments

# API for developers

[Explicar genericamente a organização da API e coleções principaus. Os detalhes/documentação dos métodos devem ficar numa solução *hosted* de documentação de APIs, como o [Swagger](https://swagger.io), Postman documentation, ou incluída no próprio desenvolvimento (e.g.: maven site)

🡪 Be sure to use [best practices for REST Api design](https://learning.oreilly.com/library/view/mastering-rest-apis/9798868803093/html/617304_1_En_7_Chapter.xhtml). Keep minda REST API applies a resource-oriented design (APIs should be designed around resources, which are the key entities your application exposes, not actions)