

TQS: Product specification report

Diogo Gaitas [73259], Giovanni Santos [115637], Rafael Semedo [115665], Vitalie Bologa [107854]

v2025-04-25

1	Introduction.....	1
1.1	Overview of the project.....	1
1.2	Known limitations.....	2
1.3	References and resources.....	2
2	Product concept and requirements.....	2
2.1	Vision statement.....	2
2.2	Personas and scenarios[IO1].....	2
2.3	Project epics and priorities.....	3
3	Domain model.....	3
4	Architecture notebook.....	3
4.1	Key requirements and constrains.....	3
4.2	Architecture view.....	4
4.3	Deployment view.....	4
5	API for developers.....	4

[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.]

1 Introduction

1.1 Overview of the project

<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>

O objetivo deste projeto é propor, conceptualizar e implementar uma aplicação em múltiplas camadas, baseada em frameworks corporativos, além de aplicar uma estratégia de Software Quality Assurance (SQA) e integrar pipelines de CI/CD ao longo de todo o ciclo de desenvolvimento.

Portanto, a **VoltUnity** será desenvolvida para aprimorar a gestão de carregamento de veículos elétricos em ambientes com múltiplos operadores. A plataforma permitirá que motoristas façam buscas e filtros de estações de carregamento próximas em tempo real, reservar vagas com antecedência para evitar filas, desbloqueiem o carregador diretamente pelo aplicativo e acompanhem seu histórico de consumo e a economia de CO₂ gerada.

Já o time de operações poderá cadastrar novas estações e gerenciar status de manutenção, monitorar tendências de uso e gerar relatórios diários, semanais ou mensais exportar dados em CSV ou PDF para análises externas.

1.2 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 >

1.3 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>

2 Product concept and requirements

2.1 Vision statement

<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?>

<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)>

Motoristas dos carros elétricos enfrentam diversas barreiras ao planejar e executar recargas: estações com disponibilidade imprevisível, múltiplos operadores e interfaces díspares, filas inesperadas e ausência de transparência nos custos e impactos ambientais. Do lado dos operadores, há dificuldades para cadastrar estações de forma centralizada, monitorar uso em tempo real e gerar relatórios consolidados para tomada de decisão.

VoltUnity propõe uma plataforma única que unifica toda a jornada de recarga de um E.V., do ponto de vista tanto do motorista quanto do operador, sem expor detalhes de implementação interna

Principais Funcionalidades do Sistema

- **Descoberta de Estações**
 - Busca geoespacial em tempo real, com filtros por localização, tipo de carregador e disponibilidade.
- **Reserva de Slot**
 - Agendamento antecipado de vagas, com prevenção de conflitos (double-booking) e confirmação instantânea.
- **Desbloqueio e Início de Carga**
 - Autenticação via token/mobile, solicitação ao carregador e confirmação de início de sessão diretamente no app.
- **Monitoramento e Relatórios de Consumo**
 - Dashboards que exibem kWh consumidos por sessão, semana e mês; cálculo de economia de CO₂ versus equivalente a combustíveis fósseis.
- **Modelos de Pagamento**
 - Fluxo “pay-per-use” ou planos de assinatura, com gestão de cartões, descontos e renovação automática.
- **Gestão de Estações (Operador)**
 - Cadastro de novas estações, sinalização de manutenção, relatórios de uso com exportação em CSV/PDF.
- **Integração Externa**
 - API pública documentada e sandbox de testes para desenvolvedores terceiros.

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

2.2 Personas and scenarios^[10]

<“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 (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.>

<The scenarios tell the story of the Personas in their lives, doing their daily/professional activities that are relevant to find the points of contact with the system under specification.

Scenarios are somewhat similar to use cases (they have a goal and tell a story), but, unlike use cases, they capture a larger process, with activities that may not use the software. Scenarios don’t required a “template”, like the usual use cases description.>

Sample: secção 4.2 neste artigo (open access)] >

Personas:

1. Emily – Motorista de um E.V. (Viajante Urbana)

- **Idade:** 35
- **Metas:** Localizar, reservar e efetuar carregamentos com eficiência em estações de veículos elétricos durante o trajeto diário.
- **Pontos problemáticos:** Disponibilidade inconsistente de carregadores, longos tempos de espera, preços pouco claros.

2. Raj – Operador de estação (gerente com conhecimento técnico)

- **Idade:** 42
- **Metas:** Monitorar o uso da estação, gerenciar a manutenção e garantir alto tempo de atividade.
- **Pontos problemáticos:** Falta de insights em tempo real, monitoramento manual, horários de menor movimento subutilizados.

3. Lea – Integrador terceirizado (Desenvolvedor de aplicativos)

- **Idade:** 29
- **Metas:** Criar serviços focados em E.V. através de APIs e dados da **VoltUnity**.
- **Pontos problemáticos:** Ambientes de teste inacessíveis, documentação de API deficiente.

Scenarios:

US 1.1 – Buscar estações próximas

Cenário Principal

1. Motorista abre a tela “Buscar Estações”.
2. App obtém localização do GPS e mostra mapa centrado.
3. Motorista aplica filtros (tipo de carregador “CCS”, disponibilidade “sim”).
4. Sistema retorna lista de estações que atendem aos critérios.
5. Motorista seleciona uma estação no mapa; app exibe detalhes (nome, distância, status).

US 1.2 – Reservar slot de carregamento

Cenário Principal

1. Motorista visualiza horários disponíveis na estação selecionada.
2. Escolhe um horário de início (ex.: 15:00–15:30).
3. O sistema bloqueia esse slot e retorna a confirmação imediata.
4. Agenda aparece como “Reservado” na UI.

US 2.1 – Desbloquear ponto de carregamento

Cenário Principal

1. Motorista acessa “Minhas Reservas” e seleciona o slot ativo.
2. Clica em “Desbloquear” no app.
3. App envia requisição ao backend e à API da estação.
4. Carregador libera fisicamente o conector; app mostra “Carregamento iniciado” com contador de kWh.

US 3.1 – Escolher pagamento ou assinatura

Cenário Principal

1. Antes de confirmar a reserva, a app apresenta opções “Pay-per-use” e “Assinatura Mensal”.
2. Motorista escolhe “Assinatura” e revisa valor mensal e benefícios.
3. Clica em “Confirmar” e é redirecionado ao checkout.

US 4.1 – Marcar estação em manutenção

Cenário Principal

1. Operador/Admin faz login no “Portal de Operações”.
2. Em “Gerenciamento de Estações”, ele ativa o toggle “Em Manutenção” na estação desejada.

3. O Status atualiza instantaneamente na API.
4. Motoristas veem a estação como “Indisponível” em buscas e reservas.

US 4.2 – Relatórios de tendência de uso

Cenário Principal

1. Operador acessa “Relatórios de Uso”.
2. Seleciona período (diário, semanal, mensal).
3. Dashboard renderiza gráfico de linha/trend.
4. Clica em “Exportar” e baixa CSV ou PDF.

US 5.1 – Visualizar economia de CO₂ (motorista)

Cenário Principal

1. Motorista abre “Impacto Ambiental” no app.
2. O sistema calcula CO₂ evitado com base no total de kWh e fator de emissão (e.g. 0,233 kg CO₂/kWh).
3. Exibe indicadores diários, semanais, mensais e gráfico comparativo com gasolina.

US 5.2 – Agregação de dados de CO₂ (admin)

Cenário Principal

1. Admin acessa o painel “Sustentabilidade”.
2. Selecionar filtros de localização e período.
3. O painel mostra gráficos de barras/linhas agregados (todas as estações).

2.3 Project epics and priorities

Epic 1: Descoberta de estações e reserva de horários

User Story 1.1

Como motorista de veículo elétrico, quero procurar estações de carregamento disponíveis perto de mim para poder planejar minha rota de carregamento com eficiência.

- **Critérios de aceitação:**
 - O utilizador pode filtrar estações por localização, disponibilidade e tipo de carregador.
 - Resultados atualizados em tempo real.

User Story 1.2

Como motorista de veículo elétrico, quero reservar um horário de carregamento com antecedência para evitar tempos de espera.

- **Critérios de aceitação:**
 - A reserva mostra os horários disponíveis.
 - Reservas duplicadas são evitadas.

Epic 2: Rastreamento de carregamento e uso

User Story 2.1

Como motorista de veículo elétrico, quero desbloquear um ponto de carregamento com meu telefone para poder começar a carregar rapidamente.

- **Critérios de aceitação:**
 - Somente reservas válidas ou horários disponíveis podem ser usados para desbloquear.
 - Mensagem de confirmação após o desbloqueio.

User Story 2.2

Como motorista de veículo elétrico, quero visualizar meu histórico de consumo de energia para poder monitorar o uso e o custo.

- **Critérios de aceitação:**
 - O painel exibe kWh por sessão/semana/mês.
 - As economias de CO2 são mostradas em comparação com alternativas à gasolina.

Epic 3: Pagamentos e Assinaturas

User Story 3.1

Como motorista de veículo elétrico, quero escolher entre pagamento por uso e assinaturas para poder controlar meus gastos.

- **Critérios de aceitação:**
 - O modelo de pagamento pode ser selecionado antes da reserva.
 - As assinaturas são refletidas na fatura e permitem descontos/acesso ilimitado.

Epic 4: Gestão da Estação (Operador)

User Story 4.1

Como operador de estação, quero marcar uma estação como em manutenção para que os motoristas não reservem horários indisponíveis.

- **Critérios de aceitação:**
 - O painel do operador inclui botões de alternância de status.
 - As atualizações de status são refletidas instantaneamente em toda a plataforma.

User Story 4.2

Como operador de estação, quero relatórios de tendências de uso para poder planejar a manutenção e os preços de forma eficaz.

- **Critérios de aceitação:**
 - Estatísticas diárias/semanais/mensais visualizadas.
 - Relatórios CSV ou PDF exportáveis.

História do Usuário 4.3

Como operador de estação, desejo registrar novas estações de carregamento na plataforma para que elas fiquem disponíveis para uso dos motoristas.

- **Critérios de aceitação:**
 - As estações recém-cadastradas aparecem na busca por motoristas após aprovação ou sincronização.
 - Formulário de inscrição com local, número de vagas e tipo de carregador.
 - Validação da entrada.

Epic 5: Impacto e Análise de CO2

User Story 5.1

Como motorista de veículo elétrico, quero ver quanto CO2 economizei ao longo do tempo para entender o impacto ambiental.

- **Critérios de aceitação:**
 - O painel mostra a economia de CO2 por dia, semana e mês.
 - Comparação com emissões equivalentes de gasolina.

User Story 5.2

*Como administrador do **VoltUnity**, quero dados agregados de CO2 e uso para mostrar o impacto da plataforma na sustentabilidade.*

- **Critérios de aceitação:**
 - Visuais e estatísticas sobre uso e benefícios ambientais.
 - Filtrar por local e período de tempo.

Epic 6: Integração com APIs externas

User Story 6.1

Como integrador de serviços, quero ter acesso a uma API bem documentada com um ambiente de teste para poder desenvolver aplicativos que se integrem à plataforma VoltUnity.

- **Critérios de aceitação:**
 - Sandbox acessível com dados simulados.
 - Documentação clara da API com exemplos de chamadas.

3 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>

4 Architecture notebook

4.1 Key requirements and constrains

<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](#) the software architect should be aware
- [Identify architectural characteristics](#) that are relevant for your project (will drive the key design decisions). Note the [case study](#) and the explicit characteristics related to users and extensibility. This will support later non-functional tests.

4.2 Architecture view

→ Discuss architecture planned for the software solution: what are the main building blocks? [include a diagram](#) (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](#) applied, if any

à 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.>

4.3 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 símbolo do PostgreSQL na Base de dados,...]. Indicar a existência de containers (Docker), endereços IP e portos,...

Esta parte será completada quando houver efetivamente deployments

5 API for developers

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

à Be sure to use [best practices for REST Api design](#). 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)

[10]or, in alternative, actors and use cases