
PowerMap

Documento de Requisitos

Durval Ferreira Sobrinho Junior
Vanúbia Santos Lima

Versão 1.0

Histórico de Alterações

Data	Versão	Descrição	Autor
06/03/2024	1.0	Versão inicial do documento	Vanúbia
09//03/2024	1.0	Adição de Requisitos Funcionais e não Funcionais	Durval e Vanúbia

Conteúdo

1. INTRODUÇÃO	5
1.1 VISÃO GERAL DO DOCUMENTO	5
1.2 CONVENÇÕES, TERMOS E ABREVIACÕES	5
1.2.1 Identificação dos requisitos	5
1.2.2 Prioridades dos requisitos	5
2. DESCRIÇÃO GERAL DO SISTEMA	5
2.1 ATORES	6
2.2 VISÃO GERAL DO SISTEMA	6
3. REQUISITOS FUNCIONAIS (CASOS DE USO)	7
[RF001] Realizar Cadastro	7
[RF 002] Realizar login	9
[RF 003] Realizar Logout	10
[RF004] Editar Perfil	11
[RF 005] Alterar senha	13
[RF 006] Adicionar admin	14
[RF 007] Deletar Conta	15
[RF008] Visualizar usuários	16
[RF 009] Criar agendamento	17
[RF 010] Marcar agendamento como feito	17
[RF011] Visualizar agendamento	19
[RF 012] Utilizar mapa	20
[RF013] Realizar Calculo de gasto	22
[RF014] Identificar posto mais próximo	22
[RF 015] Adicionar Modelo	23
[RF016] Visualizar Modelo	24
[RF017] Editar Modelos	24
[RF018] Deletar modelo	25
[RF019] Adicionar Carro	26
[RF020] visualizar carro	27
[RF021] Deletar carro	28
[RF022] Notificar agenda	29
4. REQUISITOS NÃO-FUNCIONAIS	32
4.1 USABILIDADE	32
[NF001] Interface Amigável	32
[NF002] Componentes Mobile	33
4.2 SOFTWARE	33
[NF003] Banco de Dados “NoSQL” Firebase(Realtime Database) e SQL Lite e Shared Preferences	33
[NF004] Linguagem Java	33
[NF005] Ferramenta Android Studio	33
4.3 DESEMPENHO	34
[NF005] Agilidade na Execução das Operações	34

5. DIAGRAMAS	35
5.1 DIAGRAMA DE CASO DE USO	35
5.1 Sistema	89
5.2 Gerenciar carro	00
5.3 Utilizar mapa	00
5.4 Gerenciar modelo	00
5.5 Gerenciar agenda	00
5.6 Gerenciar perfil	00
5.7 Gerenciar usuários	00
5.8 Gerir Analytics	00
5.2 DIAGRAMA DE CLASSE	36

1. Introdução

Este documento especifica os requisitos do Sistema PowerMap, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

1.1 Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

- **Seção 2 – Descrição geral do sistema:** apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- **Seção 3 – Requisitos funcionais (casos de uso):** especifica todos os casos de uso do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
- **Seção 4 – Requisitos não-funcionais:** especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho e software.

1.2 Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1.2.1 Identificação dos requisitos

Por convenção, a referência a requisitos é feita através do identificador do requisito, de acordo com a especificação a seguir:

[identificador do requisito]

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] para os requisitos funcionais e [NF001] para os não-funcionais e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

1.2.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, foram adotadas as denominações “essencial”, “importante” e “desejável”.

- **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

-
- **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
 - **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

2. Descrição geral do sistema

Esta seção descreve superficialmente o cliente, os futuros usuários e fornece uma visão geral do *PowerMap*.

2.1 Atores

O sistema contempla quatro tipos de atores, Cliente, Administrador, próprio o Sistema e APIs. O Administrador tem como principal função cadastrar modelos de carros, classificados como híbridos ou elétricos. O Cliente, por sua vez, utiliza o sistema para adicionar carros com base nos modelos previamente cadastrados pelo Administrador, para utilização durante os trajetos no mapa. Além disso, o Cliente pode selecionar um carro para uso no mapa, com a possibilidade do sistema indicar o posto de recarga mais próximo. Adicionalmente, o Cliente pode optar por realizar o cálculo de gastos, determinando a viabilidade de alcançar o posto de recarga com base no total de energia e/ou combustível disponível no veículo. O Sistema fica responsável por realizar ações de retorno ao usuário, notificando os agendamentos nativamente para o mesmo. As API também realizam ações somente de retorno, apresentando o posto mais próximo e retornando a direção entre o ponto de início e ponto de chegada, sendo utilizada para traçar o percurso entre o ponto em que o carro se encontra e o posto mais próximo

2.2 Visão Geral do Sistema

O sistema, concebido como um aplicativo móvel para Android, visa fornecer aos clientes proprietários de veículos elétricos ou híbridos a capacidade de determinar se, com base na quantidade total de energia ou combustível disponível em seus carros, é viável alcançar o posto de recarga mais próximo. Para uma compreensão mais abrangente das

funcionalidades do sistema, os requisitos funcionais e não funcionais serão descritos na próxima seção, oferecendo uma visão detalhada das capacidades e limitações da aplicação.

3. Requisitos funcionais (casos de uso)

Essa seção descreve os requisitos funcionais do sistema de forma detalhada.

3.1.1 [RF001] Realizar Cadastro

Descrição do caso de uso: o deve devem poder realizar o cadastro no sistema informando o nome, e-mail e senha.

Ator: Cliente..

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O cliente entra com os dados solicitados nos campos de cadastro.

Saídas e pós-condição:

O sistema retorna uma mensagem informando o resultado da operação e redireciona para a próxima tela, caso a verificação seja válida.

Fluxo de eventos principal

1. O usuário escolhe a ação de clicar no botão de criar.
2. O cliente é redirecionado para a tela de cadastro.
3. O sistema oferece o formulário de cadastro para o cliente.
4. O cliente entra com os dados e submete o formulário.
5. O sistema insere os dados submetidos nos bancos de dados, com o adicional do campo "isadmin" setado com "false". Os dados também são inseridos no Shared Preferences para iniciar a secção.
6. O sistema retorna para o usuário uma mensagem informando que a operação foi realizada com sucesso.
7. O Cliente é redirecionado para a tela de listar carros.

Fluxos secundários

1. Caso as verificações falhe, uma mensagem é retomada em cada campo de input, informando ao usuário o tipo de erro.

-
-
2. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.2 [RF 002] Realizar Login

Descrição do caso de uso: O usuário deve poder realizar o login no sistema fornecendo seu e-mail e senha que foi fornecido na hora de realizar o cadastro. Ou escolhendo a opção de entrar com Google.

Ator: Cliente e Admin

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O usuário cliente já deve ter concluído o processo de cadastro no sistema, incluindo o fornecimento das informações necessárias para o login, um e-mail e uma senha de acesso. Caso o usuário deseje entrar com o Google, não é necessário informar nenhum campo. Mas é necessário que o usuário tenha internet, assim será realizada a verificação das credenciais de acesso do aplicativo com o Google, e o acesso será liberado.

Saídas e pós-condição:

O acesso ao sistema é concedido, permitindo que o cliente utilize os recursos e serviços disponíveis.

Fluxo de eventos principal

1. Usuário escolhe a ação Realizar Login.
2. O sistema apresenta um formulário de login que solicita ao usuário que forneça seu endereço de e-mail e senha registrados durante o cadastro.
3. O usuário insere as informações de login (endereço de e-mail e senha) nos campos apropriados.
4. O sistema verifica as credenciais inseridas pelo usuário.
5. Se as credenciais estiverem corretas e correspondentes às informações registradas nos bancos de dados e no Shared Preferences.
6. O sistema concede o acesso ao usuário e permite que ele utilize os recursos e serviços disponíveis.

Fluxos secundários

1. Caso as verificações falhe, uma mensagem é retomada em cada campo de input, informando ao usuário o tipo de erro.
2. Se ocorrer uma falha de comunicação com o banco de dados durante a verificação das credenciais de login, o sistema deve tratar essa situação.
3. Uma mensagem de erro é gerada e exibida ao usuário, informando-o sobre a falha na comunicação com o banco de dados.
4. A operação de login é cancelada devido à falha.

3.1.3 [RF 003] Realizar Logout

Descrição do caso de uso: os usuários podem efetuar o logout no sistema, encerrando sua sessão ativa.

Ator: Cliente e Admin

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

Antes de efetuar o logout, o Cliente deve estar autenticado no sistema, ou seja, já ter realizado o login ou ter se cadastrado a primeira vez com sucesso.

Saídas e pós-condição:

Após o logout, o Cliente não terá mais acesso ao sistema. Qualquer sessão anterior será encerrada, e ele deverá realizar um novo login para acessar o sistema novamente.

Fluxo de eventos principal

1. Usuário escolhe a ação Realizar Logout.
2. O sistema confirma a ação e encerra a sessão ativa limpando os dados do Shared Preferences.
3. O usuário é redirecionado para tela splash do sistema.

Fluxos secundários

1. Caso as verificações falhe, uma mensagem é retomada em cada campo de input, informando ao usuário o tipo de erro.
2. Se ocorrer uma falha de comunicação com o banco de dados, uma mensagem de erro é gerada e exibida ao usuário, informando-o sobre a falha.
3. A operação é cancelada.

3.1.4 [RF 004] Editar perfil

Descrição do caso de uso: Os usuários sendo Cliente ou Admin, pode editar o perfil. Os dados que podem ser editados são os de nome e do e-mail. Não é obrigatório a edição dos dados.

Ator: Cliente e Admin

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

Para que o Admin ou Cliente possa editar o perfil , o mesmo já deve estar previamente autenticado no sistema, ou seja, já ter realizado o login com sucesso.

Saídas e pós-condição:

Após editar, o usuário ainda permanecerá logado no sistema e poderá tomar ações adicionais.

Fluxo de eventos principal

1. O usuário clica no ícone de perfil no menu.
2. O sistema redireciona para a tela de perfil.
3. A tela é apresentada com os campos de nome e o e-mails previamente preenchidos.
4. O usuário entra com os valores.

-
-
5. O usuário submete os dados ao clicar no botão de editar.
 6. Os dados são atualizados nos bancos de dados e atualizados no Shared Preferences.
 7. O sistema retorna para o usuário uma mensagem informando que a operação foi realizada com sucesso.

Fluxos secundários

1. Caso as verificações falhe, uma mensagem é retomada em cada campo de input, informando ao usuário o tipo de erro.
2. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.5 [RF 005] Alterar senha

Descrição do caso de uso: os usuários, sendo os mesmos, Cliente ou Admin, podem alterar a senha.

Ator: Cliente e Admin

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

Entradas e pré-condições:

Antes de altera a senha, os diários já 'devem estar previamente cadastrados ou logados no sistema.

Saídas e pós-condição:

Após editar, o usuário ainda permanecerá logado no sistema e poderá tomar ações adicionais.

Fluxo de eventos principal

1. O usuário clica no ícone de perfil no menu.
2. O sistema redireciona para a tela de perfil.
3. A tela é a apresentada contendo elementos, entre eles o botão de editar senha.
4. O usuário clica no botão de estiar senha.

-
-
5. O usuário é redirecionado para a tela de editar senha.
 6. O usuário deve inserir os dados referente a senha antiga, a senha nova e a confirmação da senha.
 7. O sistema verifica os dados fornecidos pelo Cliente.
 8. A senha é atualizada nos bancos de dados e no Shared Preferences.
 9. O sistema retorna para o usuário uma mensagem informando que a operação foi realizada com sucesso.

Fluxos secundários

1. Caso as verificações falhe, uma mensagem é retomada em cada campo de input, informando ao usuário o tipo de erro.
2. Caso ocorra uma falha na comunicação com o banco de dados, o usuário recebe uma mensagem informando sobre a falha.

3.1.6 [RF 006] Adicionar admin

Descrição do caso de uso: O Admin poderá editar os dados de seu perfil na plataforma.

Ator: Admin.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

Entradas e pré-condições:

O Admin entra com os dados solicitados nos campos de cadastro. O admin já deve estar previamente cadastrado no sistema, ou seja, está em uma sessão.

Saídas e pós-condição:

O sistema retorna uma mensagem informando o resultado da operação e redireciona para a próxima tela, caso a verificação seja válida.

Fluxo de eventos principal

1. O Admin clica no botão de adicionar admin contido na tela de listar usuários.
2. O Admin é redirecionado para a tela de cadastro.
8. O sistema oferece o formulário de cadastro para o cliente.
9. O Admin entra com os dados e submete o formulário.

-
-
10. O sistema insere os dados submetidos nos bancos de dados, com o adicional do campo "isadmin" setado com "true".
 11. O sistema retorna para o Admin uma mensagem informando que a operação foi realizada com sucesso.
 12. O Admin permanece na mesma tela;

Fluxos secundários

1. Caso as verificações falhe, uma mensagem é retomada em cada campo de input, informando ao usuário o tipo de erro.
2. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.7 [RF 007] Deletar conta

Descrição do caso de uso: Os usuários poderão remover sua conta.

Ator: Cliente e Admin.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

Os usuários precisam estar cadastrado ou logado no sistema para utilizar essa funcionalidade.

Saídas e pós-condição:

O sistema retorna uma mensagem informando o resultado da operação e redireciona para a próxima tela, caso a verificação seja válida.

Fluxo de eventos principal

1. O usuário clica no ícone de perfil no menu.
2. O sistema redireciona para a tela de perfil.
3. A tela é apresentada contendo elementos, entre eles o botão de editar e deletar a conta.
4. O usuário clica no botão de deletar a conta.
5. O usuário é redirecionado para a tela splash.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.8 [RF008] Visualizar usuários
--

Descrição do caso de uso: O Admin pode visualizar os usuários registrados no sistema.

Ator: Admin.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

Para realizar esta ação, o usuário deve estar cadastrado e logado no sistema.

Saídas e pós-condição:

O sistema redireciona o Admin para a página de visualização dos usuários.

Fluxo de eventos principal

1. O Admin clica no ícone de usuários no menu.
2. O sistema redireciona para a tela de listar usuários.
3. A tela é apresentada listando todos os usuários, e contendo três opções de filtro. Para listar todos os usuários, para listar somente os admins, ou para listar somente os clientes.
4. O Admin clica no botão pode clicar selecionar as uma das três opções.
5. O Admin permanece na mesma tela, e a lista é apresentada.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, a tela permanece vazia.

3.1.9 [RF 009] Criar agendamento

Descrição do caso de uso: O cliente pode criar agendamento de avisos.

Ator: Cliente.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O cliente precisa estar cadastrado ou logado no sistema para utilizar essa funcionalidade.

Saídas e pós-condição:

O sistema informa se a ação de iniciar expansão foi bem sucedida.

Fluxo de eventos principal

1. O Cliente clica no ícone de agenda no menu.
2. O sistema redireciona para a tela de listar agendas.
3. A tela é apresentada listando todas as agendas e contendo o botão de adicionar agenda.
4. O Cliente clica no botão de adicionar.
5. O sistema redireciona para a tela de adicionar agenda contendo os campos de entrada de dados, solicitando a data, o horário, a descrição e o tipo de repetição e apresentando o botão de adicionar.
6. O usuário clica no botão de adicionar agenda e os dados são submetidos.
7. O sistema informa para o usuário que a submissão foi bem sucedida.

Fluxos secundários

1. Caso as verificações falhe, uma mensagem é retomada em cada campo de input, informando ao usuário o tipo de erro.
2. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.10 [RF 010] Marcar agendamento como feito
--

Descrição do caso de uso: O cliente pode marcar os agendamentos como feita.

Ator: Cliente.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado ou logado no sistema para utilizar essa funcionalidade. E já deve ter cadastrado ao menos um agendamento.

Saídas e pós-condição:

O sistema apaga o agendamento da tela se a operação for bem sucedida.

Fluxo de eventos principal

1. O Cliente clica no ícone de agenda no menu.
2. O sistema redireciona para a tela de listar agendas.
3. A tela é apresentada listando todas as agendas e contendo o botão de adicionar agenda e um campo de seleção em cada agenda.
4. O Cliente clica no campo de seleção da agenda.
5. O sistema apaga a agenda do banco de dados SQLite e apaga a agenda da tela.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.11 [RF 011] Visualizar agendamento

Descrição do caso de uso: O Cliente pode visualizar agendamentos.

Ator: Cliente.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado ou logado no sistema para utilizar essa funcionalidade. E já deve ter cadastrado ao menos um agendamento.

Saídas e pós-condição:

O sistema apresenta a lista de agendamentos se a operação for bem sucedida.

Fluxo de eventos principal

1. O Cliente clica no ícone de agenda no menu.
2. O sistema redireciona para a tela de listar agendas.
3. A tela é apresentada listando todas as agendas e contendo o botão de adicionar agenda e um campo de seleção em cada agenda.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, a tela permanece sem agendamentos.

3.1.12 [RF 012] Utilizar mapa

Descrição do caso de uso: O Cliente poderá utilizar o mapa.

Ator: Cliente.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado ou logado no sistema, bem como, ter conexão na rede(internet, dados moveis) e ativar a localização do aparelho para utilizar essa funcionalidade.

Saídas e pós-condição:

O sistema retorna a visualização do mapa possibilitando que o Cliente interaja com o mapa.

Fluxo de eventos principal

1. O Cliente clica no ícone de mapa no menu.
2. O sistema redireciona para a tela de apresentação do mapa.
3. A tela é a apresentada mostrando o mapa Possibilitando a interação do usuário com o mapa.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com a Google Maps (sem conexão com a internet ou conexão com a API) o sistema retorna uma mensagem informando sobre a falha.

3.1.13 [RF013] Realizar cálculo de gasto

Descrição do caso de uso: O cliente deve poder realizar o cálculo de gasto. Este cálculo possibilitará que o cliente saiba se perante o total de energia ou combustível que o seu carro possui em relação à distância do posto mais próximo, se é possível chegar até o posto.

Ator: Cliente

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado e logado no sistema, ter conexão na rede(internet, dados móveis) , está com a localização ativada, bem como, o Admin já ter cadastrado ao menos um modelo de carro, para que o Cliente possa adicionar um carro vinculado àquele

modelo, possibilitando que o Cliente escolha ao menos um carro para ser utilizado no mapa, e assim, o mesmo poder realizar o cálculo de gasto vinculado ao carro escolhido.

Saídas e pós-condição:

O sistema retorna um componente do Material Design “dialog” caracterizado por ser uma caixa de diálogo que se assemelha a um card, que será apresentado de forma temporária na tela.

Fluxo de eventos principal

1. O Cliente clica no ícone de mapa no menu.
2. O sistema redireciona para a tela de apresentação do mapa.
3. A tela é apresentada mostrando o mapa, possibilitando a interação do cliente com o mapa e apresentando três botões na tela, o de redirecionamento, o de escolher carro, e o de realizar cálculo de gasto.
4. O usuário deve clicar no botão de realizar cálculo gasto.
5. O sistema deve apresentar o componente “dialog” temporariamente solicitando que o Cliente informe a velocidade e o total de energia do carro atualmente, se o carro for elétrico. Caso o carro seja híbrido, será solicitado também o total de gasolina que o automóvel possui no momento. Também é apresentado o botão de calcular.
6. O cliente deve clicar no botão de calcular.
7. O sistema retornará no “dialog” com a mensagem de se é possível chegar até o posto, a informação da distância até o posto, o total de energia que iria gastar e o total de gasolina que seria gasto durante o percurso.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.14	[RF014] Identificar posto mais próximo
---------------	---

Descrição do caso de uso: A Place API deve retornar os postos mais próximos.

Ator: Api Place e Direction API.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado e logado no sistema, ter conexão na rede(internet, dados móveis) , está com a localização ativada, e o usuário já possuir pelo menos um carro cadastrado. Pois o primeiro carro é selecionado automaticamente para ser utilizado no mapa.

Saídas e pós-condição:

O sistema retorna o mapa com os postos mais próximos ao carro o usuário.

Fluxo de eventos principal

1. O Cliente clica no ícone de mapa no menu.
2. O sistema redireciona para a tela de apresentação do mapa.
3. A tela é apresentada mostrando o mapa com os postos mais próximos retornados pela Place API, e traçando uma linha entre o carro e o posto mais próximo aplicado pela Direction API. Caso o carro for híbrido será apresentado o mapa com os postos de energia e de gasolina.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com a API o mapa retorna a mensagem avisando que não há conexão. E a visualização do mapa fica padrão.

3.1.15 [RF015] Adicionar modelo
--

Descrição do caso de uso: O Admin deve poder adicionar modelos de carros.

Ator: Admin

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado ou logado no sistema para utilizar essa funcionalidade.

Saídas e pós-condição:

O sistema retorna para a tela de listar modelos.

Fluxo de eventos principal

1. O Admin clica no ícone de carro no menu.
2. O sistema redireciona para a tela de listagem de modelos.
3. A tela é apresentada mostrando todos os modelos cadastrados e o botão de adicionar modelo.
4. O Admin deve clicar no botão de adicionar modelo.
5. O sistema redireciona para a tela de adicionar modelo solicitando que o Admin preencha os campos de nome do modelo, ano do modelo, o tipo (híbrido ou elétrico), ao consumo de energia em Kwh/Km e o total de Consumo de Combustível litros/Km. Também é apresentado o botão de adicionar modelo.
6. O Admin deve clicar no botão de adicionar modelo.
7. O sistema salva os dados nos bancos de dados e redireciona para a tela de listar modelos.

Fluxos secundários

-
-
1. Caso as verificações falhe, uma mensagem é retomada em cada campo de input, informando ao usuário o tipo de erro.
 2. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.16 [RF019] Visualizar modelos
--

Descrição do caso de uso: O Admin deve poder visualizar os modelos de carros cadastrados.

Ator: Admin

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Admin precisa estar previamente cadastrado ou logado no sistema e deve já ter cadastrado ao menos um modelo.

Saídas e pós-condição:

O sistema retorna a tela listando os modelos cadastrados.

Fluxo de eventos principal

8. O Cliente clica no ícone de carro no menu.
9. O sistema redireciona para a tela de listagem de modelos.
10. A tela é apresentada mostrando todos os modelos cadastrados e o botão de adicionar modelo.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é apresentada na tela informando da falta.

3.1.17 [RF020] Editar modelo

Descrição do caso de uso: O Admin poderá editar o modelo do carro.

Ator: Admin.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

Entradas e pré-condições:

O Admin precisa estar previamente cadastrado e logado no sistema para utilizar essa funcionalidade. E já deve ter cadastrado ao menos um modelo.

Saídas e pós-condição:

O sistema retorna a tela listando os modelos cadastrados.

Fluxo de eventos principal

1. O Admin clica no ícone de carro do menu.
2. O sistema redireciona para a tela de listagem de modelos.
3. A tela é apresentada com os carros cadastrados, contendo os botões de editar e de deletar o modelo.
4. O Admin deve clicar no botão de editar.
5. O sistema deve redirecionar para tela de editar o modelo, contendo os campos de nome, ano do modelo e o total de energia, previamente preenchidos, se o carro for elétrico. Caso o modelo seja híbrido o campo de gasolina é apresentado. A tela também apresenta o botão de editar e de adicionar foto, além de não permitir que o modelo seja editado.
6. O Admin preenche os dados e submete os dados ao clicar no botão de editar.
7. Os dados são atualizados nos bancos de dados.
8. O sistema retorna para o Admin uma mensagem informando que a operação foi realizada com sucesso.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.18 [RF021] Deletar modelo

Descrição do caso de uso: O usuário Admin poderá deletar um modelo previamente cadastrado.

Ator: Admin

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado ou logado no sistema para utilizar essa funcionalidade. E já deve ter cadastrado ao menos um modelo de carro.

Saídas e pós-condição:

O sistema apaga o modelo da tela.

Fluxo de eventos principal

1. O Admin clica no ícone de carro do menu.
2. O sistema redireciona para a tela de listagem de modelos.
3. A tela é apresentada com os carros cadastrados, contendo os botões de editar e de deletar o modelo.
4. O Admin deve clicar no botão de deletar.
5. O sistema deve apagar o modelo dos bancos de dados.
6. O sistema deve apagar o modelo da tela de listagem.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.19 [RF019] Adicionar carro

Descrição do caso de uso: O Cliente deve poder adicionar um carro vinculado a um modelo.

Ator: Cliente

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado ou logado no sistema, e o Admin já deve ter cadastrado ao menos um modelo de carro.

Saídas e pós-condição:

O sistema retorna para a tela de listar carros.

Fluxo de eventos principal

1. O Cliente clica no botão de carro no menu
2. O sistema redireciona para a tela de listar carros, que contem o botão de adicionar carro.
3. O Cliente é redirecionado para a tela de adicionar carro.
13. O sistema oferece o formulário de cadastro para o cliente, solicitando o nome de um carro e a seleção de um modelo e apresenta o botão de adicionar
14. O Admin submete o formulário clicando no botão de adicionar.
15. O sistema insere os dados submetidos nos bancos de dados.
16. O sistema retorna para o Admin uma mensagem informando que a operação foi realizada com sucesso.
17. O Admin é redirecionado para a tela de listar carros;

Fluxos secundários

1. Caso as verificações falhe, uma mensagem é retomada em cada campo de input, informando ao usuário o tipo de erro.
2. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.20 [RF020] Visualizar carro
--

Descrição do caso de uso: O Cliente deve poder visualizar os carros cadastrados.

Ator: Cliente

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado ou logado no sistema para utilizar essa funcionalidade. E já deve ter cadastrado ao menos um carro.

Saídas e pós-condição:

O sistema apresenta a tela de listar carros.

Fluxo de eventos principal

1. O Cliente clica no botão de carro no menu
2. O sistema redireciona para a tela de listar carros

Fluxos secundários

1. Caso ocorra uma falha na apresentação dos dados ou não existam carros cadastrados, é apresentado uma mensagem explanando que não existem carros cadastrados.

3.1.21 [RF022] Deletar carro

Descrição do caso de uso: O Cliente deve poder deletar um carro já cadastrado.

Ator: Cliente.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado ou logado no sistema para utilizar essa funcionalidade. E já deve ter cadastrado ao menos um carro.

Saídas e pós-condição:

O sistema apaga o carro da tela.

Fluxo de eventos principal

7. O Cliente clica no ícone de carro do menu.
8. O sistema redireciona para a tela de listagem de carros.
9. A tela é apresentada com os carros cadastrados.
10. O Admin deve clicar no botão de deletar.
11. O sistema deve apagar o modelo dos bancos de dados.
12. O sistema deve apagar o modelo da tela de listagem.

Fluxos secundários

2. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.22 [RF023] Notificar agenda
--

Descrição do caso de uso: O sistema deve realizar a ação de notificar um agendamento. Seguindo a regar de cair 3 horas antes do horário delimitado no agendamento.

Ator: Sistema.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O Cliente precisa estar previamente cadastrado ou logado no sistema para utilizar essa funcionalidade. E ter previamente já ter cadastrado um agendamento.

Saídas e pós-condição:

O sistema irá notificar o cliente, mesmo que o cliente esteja com o aplicativo em segundo plano.

Fluxo de eventos principal

1. abrir o aplicativo
2. Deixar o aplicativo em segundo plano ou aberto;
3. Receber a notificação.

Fluxos secundários

1. Caso ocorra uma falha na execução o aplicativo não notifica.

4. Requisitos não-funcionais

4.1 Usabilidade

Esta seção descreve os requisitos não funcionais associados à facilidade de uso da interface com o usuário.

4.1.1 [NF001] Interface Amigável

O sistema terá uma interface amigável ao usuário primário sem se tornar cansativa aos usuários mais experientes.

Prioridade: ■ Essencial □ Importante □ Desejável

4.1.2 [NF002] Componentes Mobile

A interface deverá utilizar elementos comuns a usuários de sistemas mobile, como campos de entrada de texto, botões, *ícones* e *redirecionamento*. A ideia é focar nos aspectos operacionais e na apresentação limpa e usual dos elementos visuais, de modo que o sistema fique intuitivo e de fácil uso.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

4.2 Software

Esta seção descreve os requisitos não-funcionais associados aos softwares que devem ser utilizados para o desenvolvimento do sistema.

4.2.1 [NF003] Bancos de Dados "NoSQL" Firebase(Realtime Database) e SQL Lite e Shared Preferences

O sistema deve utilizar dois bancos de dados. O banco de dados SQL Lite servirá para permanência de dados, de forma local. Já o banco de dados Realtime Database servirá para permanência de dados em nuvem. O recurso Shared Preferences será utilizado para armazenar dados de forma temporária, para disponibilizar dados do usuário durante as sessões.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

4.2.2 [NF004] Linguagem Java

Visando criar um produto com segurança, deve-se adotar a linguagem de programação Java como linguagem principal de desenvolvimento, seguindo cuidadosamente as técnicas de orientação a objetos.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

4.2.3 [NF005] Ferramenta Android Studio

Visando o desenvolvimento rápido e uma boa qualidade do sistema, será utilizado para o desenvolvimento da aplicação a ferramenta de desenvolvimento mobile Android Studio.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

4.3 Desempenho

Esta seção descreve os requisitos não-funcionais associados à eficiência, uso de recursos e tempo de resposta do sistema.

4.3.1 [NF005] Agilidade na Execução das Operações

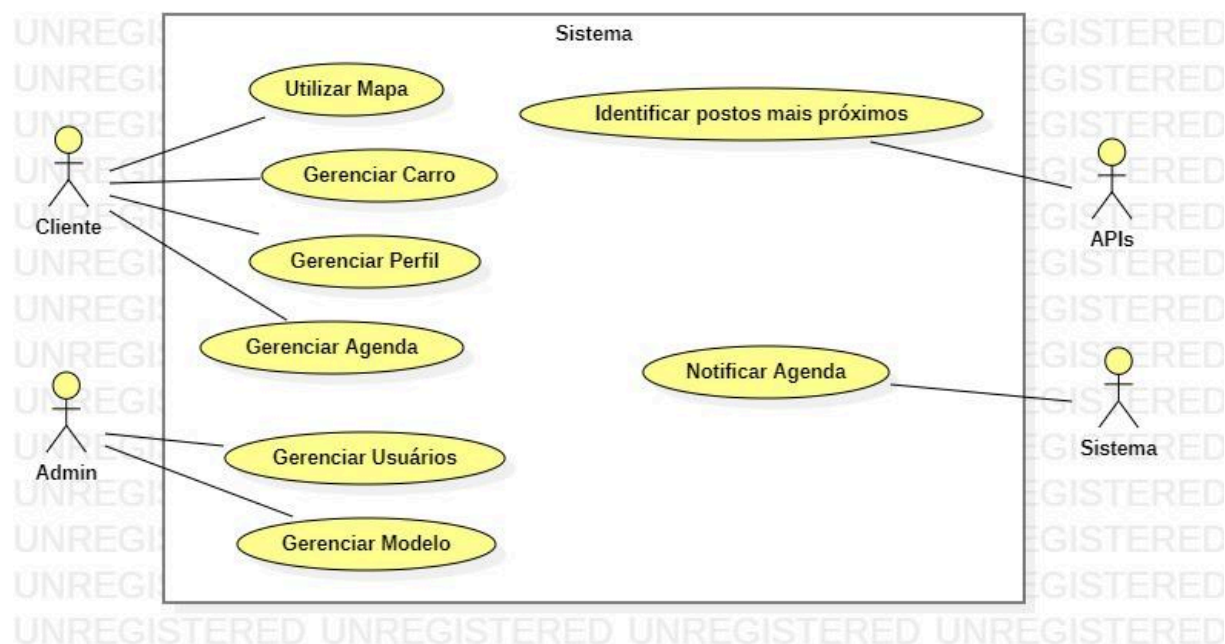
O sistema deve executar as operações no menor tempo possível, visando dar uma maior agilidade ao processo.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

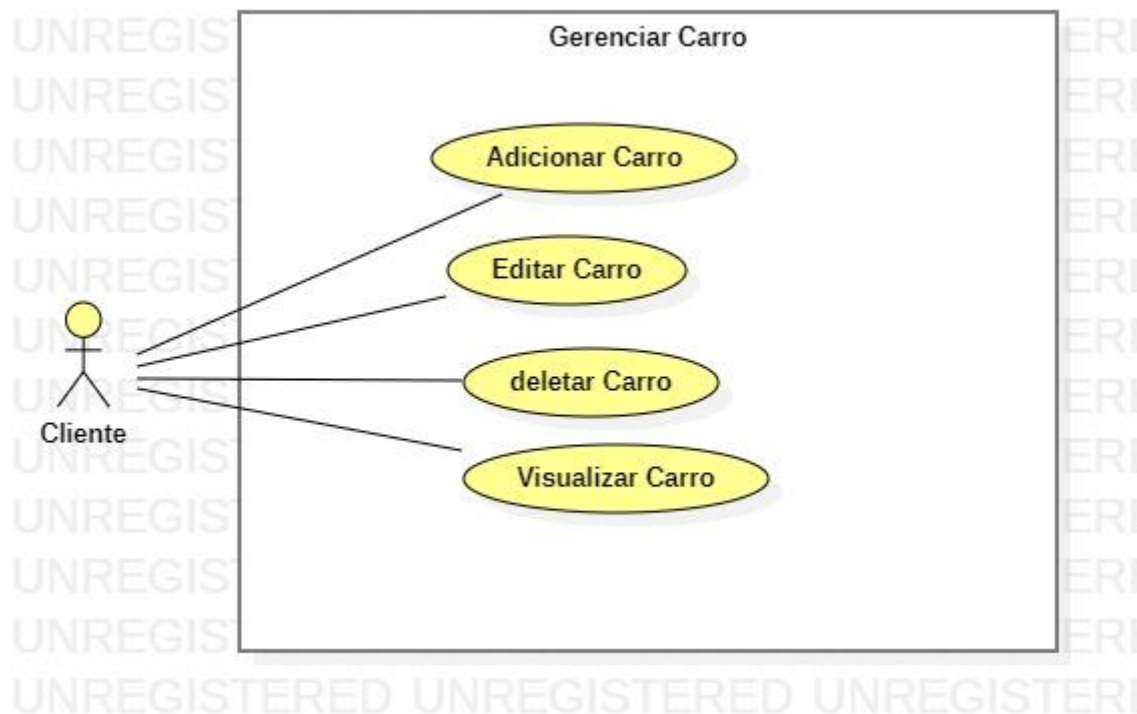
5. Diagramas

5.1 Diagramas de Caso de Uso

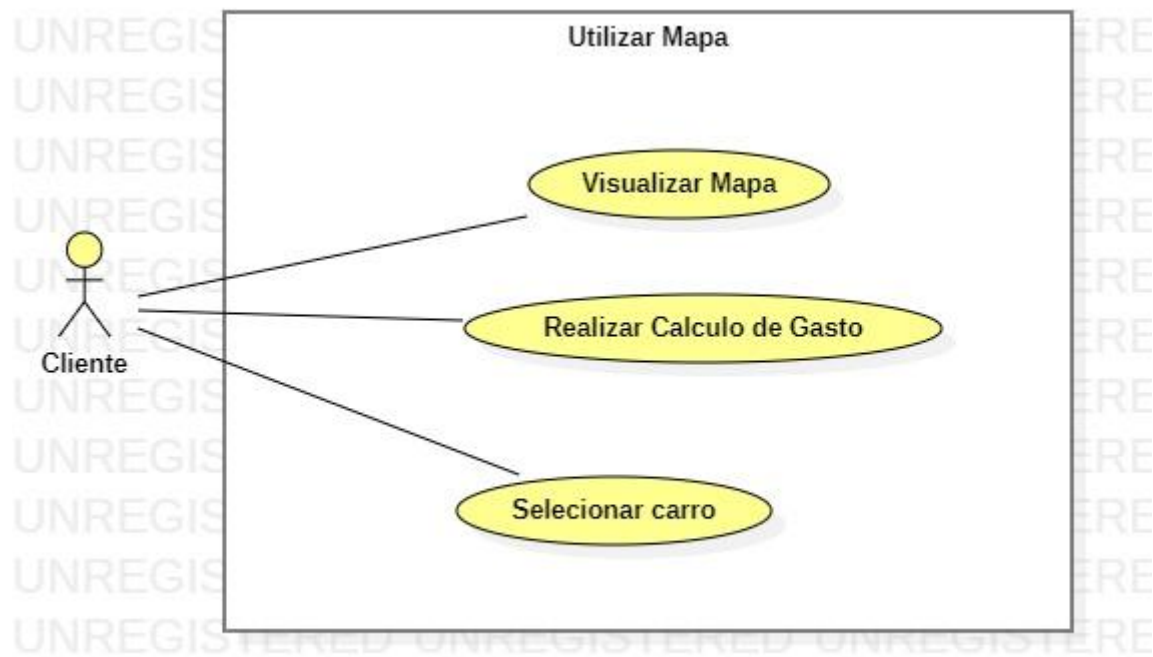
5.1 Sistema



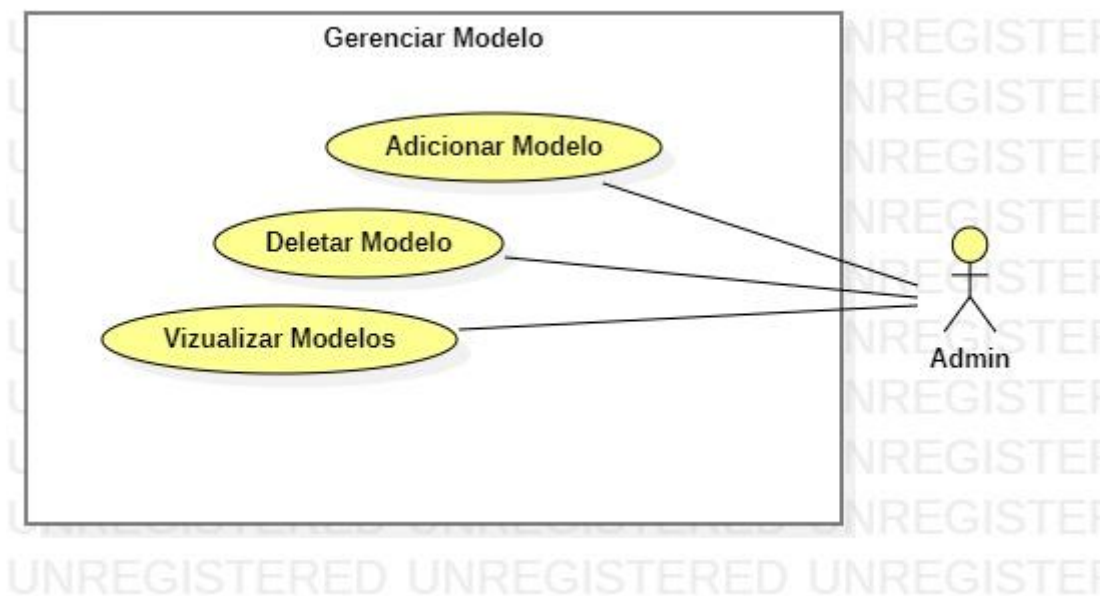
5.2 Gerenciar carro



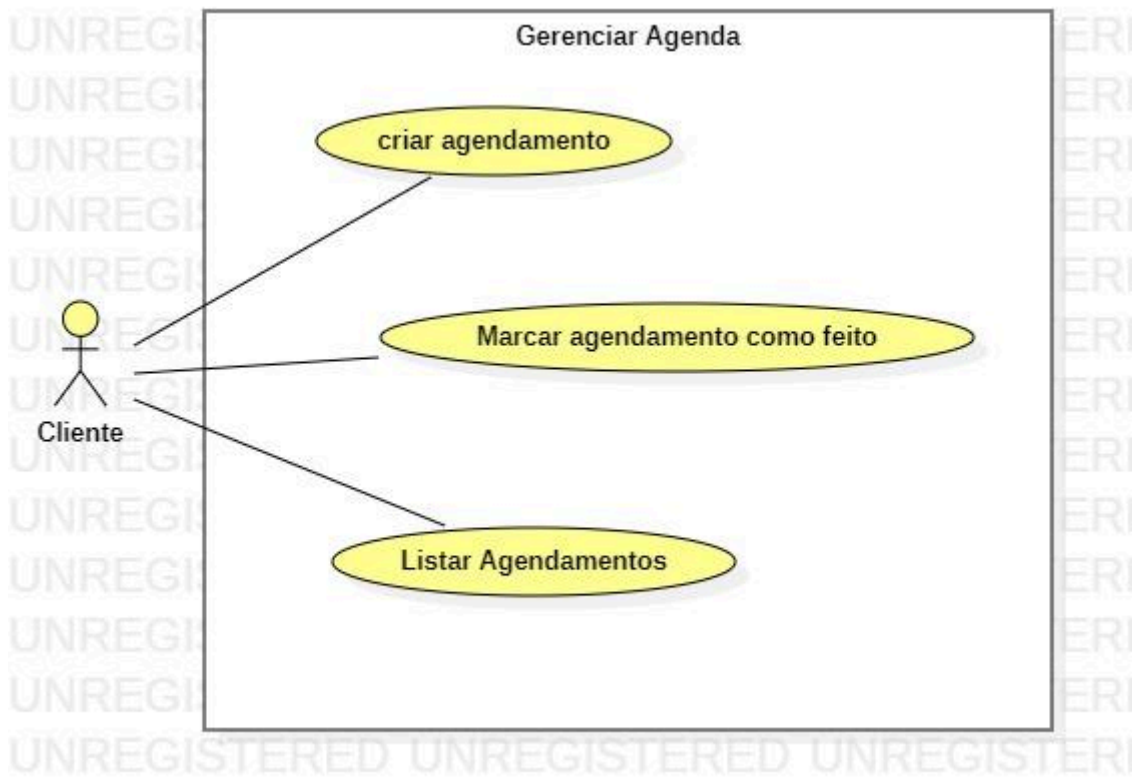
5.3 Utilizar mapa



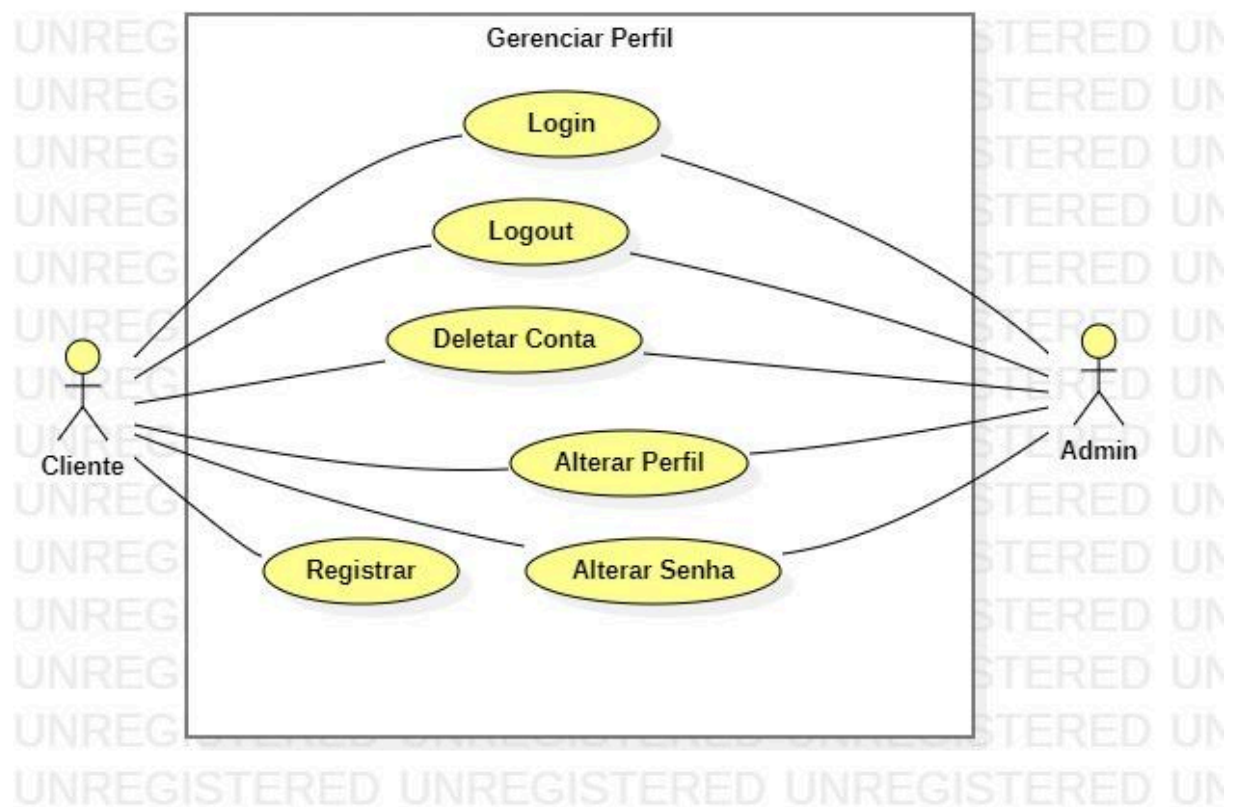
5.4 Gerenciar modelo



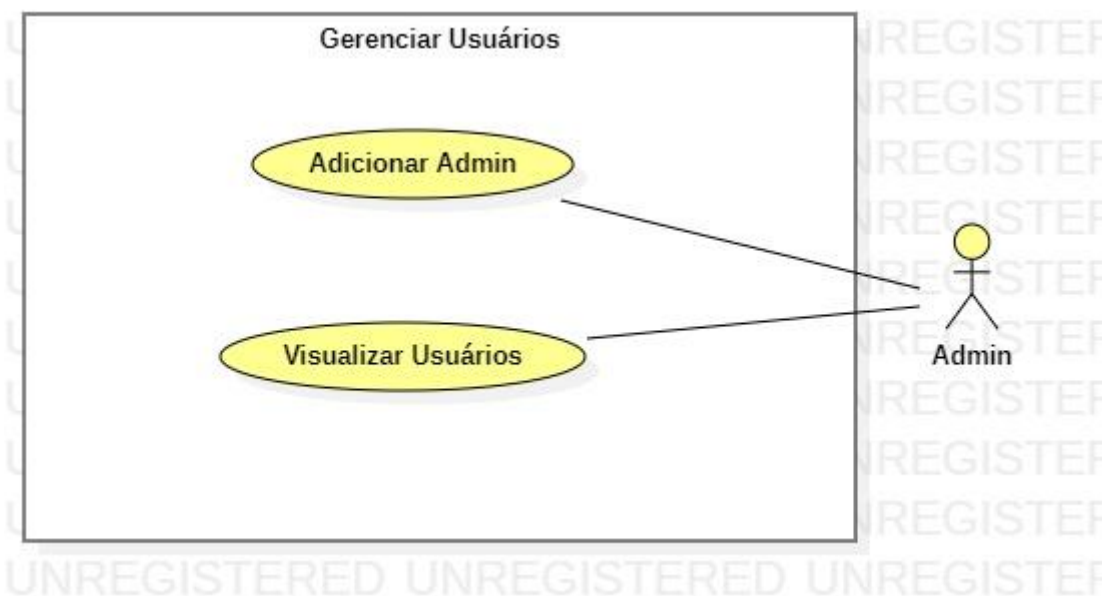
5.5 Gerenciar agenda



5.6 Gerenciar perfil

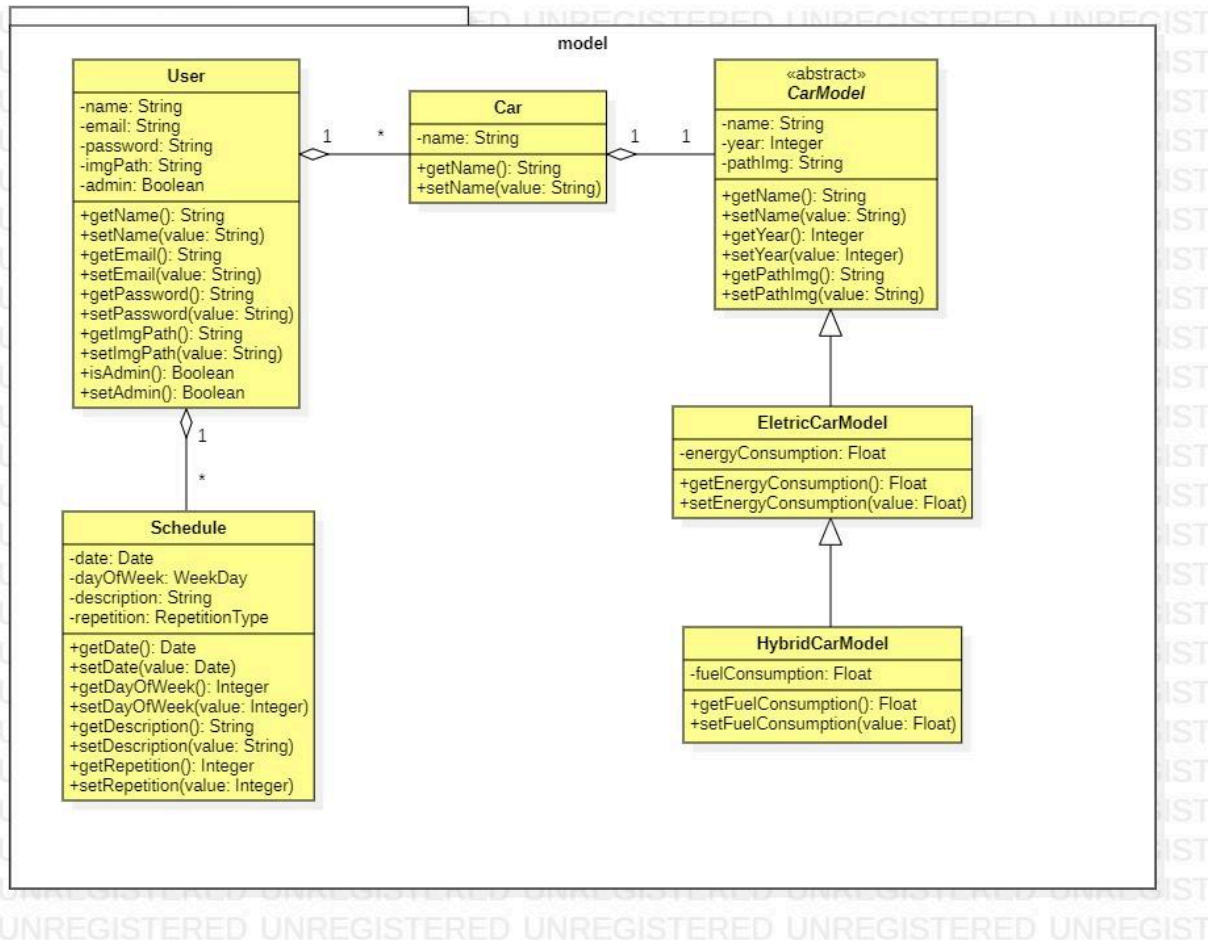


5.8 Gerenciar usuários

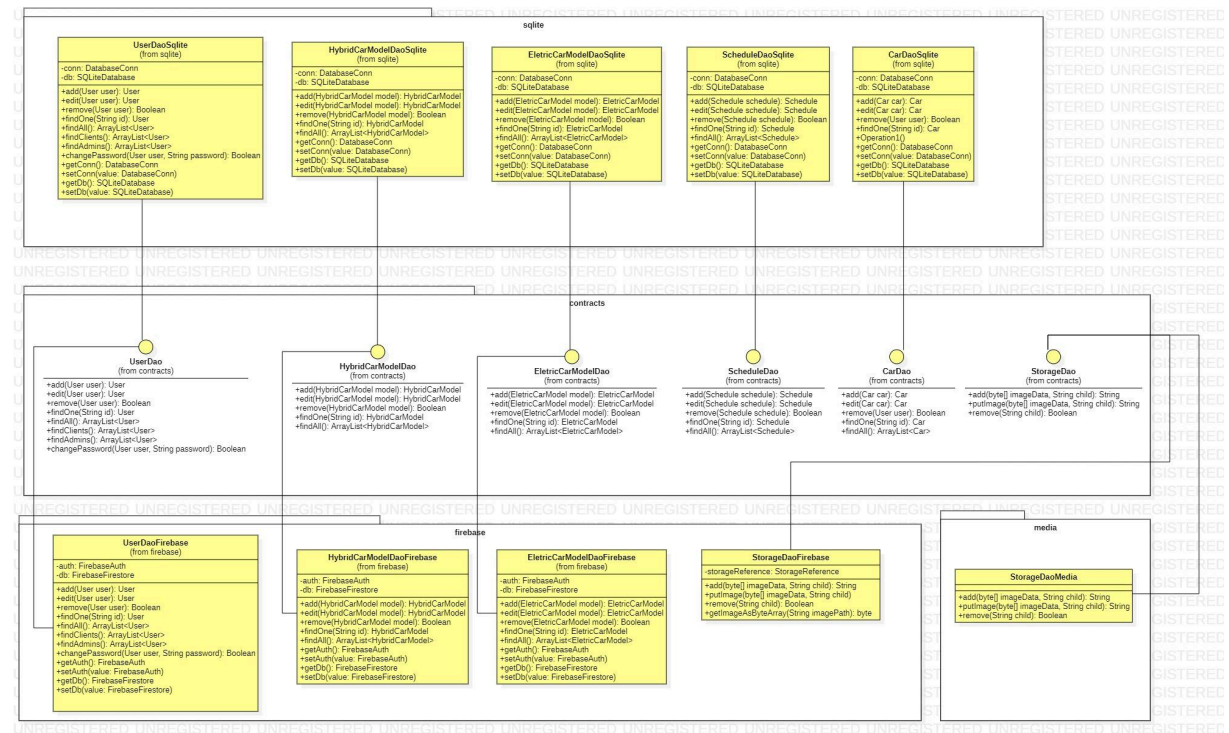


5.2 Diagrama de Classe

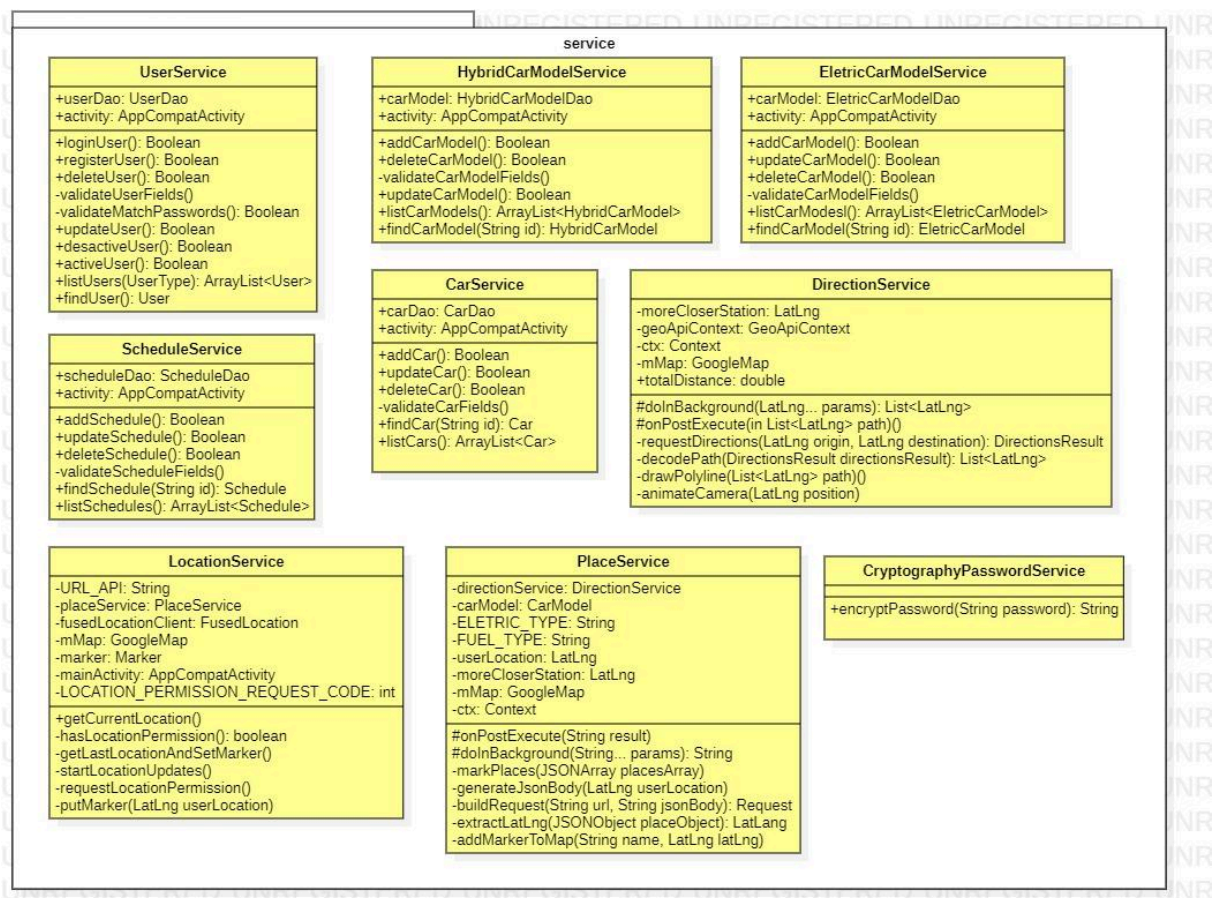
5.2.1 Model



5.2.2 DAO



5.2.3 Service



5.3 Diagrama de pacotes

