**UNIVERSIDADE CATÓLICA DE SANTOS**

**SISTEMAS DE INFORMAÇÃO**

**Elián Roventini de Andrade**

**Gabriel Souza Recouso**

**Gustavo Lino Barbosa**

**Lara Iasmine Fabiano**

**Leonardo Ribeiro Blume**

**Matheus La Scala Candido da Costa**

**Weslley Richard**

**Projeto interdisciplinar**

**UniDrive**

**SANTOS**

**2023**

**Elián Roventini de Andrade**

**Gabriel Souza Recouso**

**Gustavo Lino Barbosa**

**Lara Iasmine Fabiano**

**Leonardo Ribeiro Blume**

**Matheus La Scala Candido da Costa**

**Weslley Richard**

**Projeto interdisciplinar**

**UniDrive**

Projeto interdisciplinar com as disciplinas de Engenharia de Software e Projeto banco de dados apresentando a banca da Universidade Católica de Santos.

Orientador(a): Prof. Thiago Ferauche.

**SANTOS**

**2023**

**1. Introdução**

**1.1 Objetivo deste documento**

Este documento tem como objetivo auxiliar nas definições das especificações técnicas para o desenvolvimento do Sistema para Gerenciamento de Veículos Automotores para Test Drive.

**1.2 Escopo do produto**

O UniDrive, Sistema para Gerenciamento de Veículos Automotores para Test Drive, tem como foco auxiliar aos usuários na criação de agendamentos para uma avaliação de direção de curto prazo com o veículo automotor escolhido pelo condutor das opções que estejam disponíveis no momento do agendamento. O condutor poderá escolher o veículo de acordo com suas preferências e disponibilidade dos veículos de cada empresa.

As empresas (concessionárias) deverão cadastrar todos seus veículos com as informações necessárias para Test Drive dentro do sistema para que possa ser exibido no catálogo dela.

Além disso, o produto trará uma maior facilidade para conectar clientes em potenciais (Leads) com os vendedores das empresas, visando que as informações dos usuários estarão inseridas no sistema e será repassada para as empresas, caso o cliente em potencial (usuário) permita repassar suas informações para o contato, respeitando a Lei Geral de Proteção de Dados (LGPD). Com isso, o sistema se torna uma “ponte” para possíveis vendas das empresas.

**1.2.1 Componentes Principais**

O produto será chamado “UniDrive” e possuirá os seguintes componentes para garantir o gerenciamento (concessionária) e agendamentos (condutor):

Módulo para gerenciamento de veículos disponíveis para Test Drive: responsável por validar todas as informações referente aos carros cadastrados.

Módulo de Seleção de veículos disponíveis para realização do Test Drive. Os condutores podem selecionar o veículo de sua preferência e agendar para a realização dele.

**1.2.2 Missão do produto**

Facilitar ao usuário realizar um agendamento do carro de seu interesse com todas as concessionárias localizadas em uma só plataforma, tirando o trabalho do usuário de pesquisar várias e entrar em contato ou acessar o site para saber se tem o carro de seu interesse.

**1.2.3 Benefícios do produto**

|  |  |  |
| --- | --- | --- |
| **Número de Ordem** | **Benefício** | **Valor para o condutor** |
| 1 | Consulta online pelo veículo para Test-Drive, disponíveis pelas concessionárias, permitindo-se a reserva do mesmo previamente | Essencial |
| 2 | Aumento nas vendas e captação de condutores por conta das concessionárias | Essencial |
| 3 | Melhoria no processo da execução de Test-Drive | Essencial |

**1.4 Visão geral deste documento**

Este documento está estruturado da seguinte forma:

* **Seção 1**: Escopo;
* **Seção 2:** Diagrama de Caso de Uso;
* **Seção 3:** Detalhamento de Fluxos;
* **Seção 4:** Protótipos das Telas;
* **Seção 5:** Diagrama de Arquitetura;
* **Seção 6:** Banco de Dados;
* **Seção 7:** Tecnologias e Frameworks;
* **Seção 8:** Infraestrutura de Host;
* **Seção 9:** Estrutura Analítica do Projeto;
* **Seção 10:** Links Importantes.

**1.5 Descrição Geral do Produto**

**1.5.1 Interfaces de usuário**

A empresa cadastrará os veículos disponíveis para Test Drive a sua escolha, com os requisitos especificados de cada veículo, que serão usados como filtro pelos usuários na hora da escolha. No UniDrive, o condutor poderá escolher o veículo desejado para efetuar o Test Drive, a partir de uma lista de veículos disponíveis, podendo filtrar por alguns parâmetros, como por exemplo: Raio de distância, Modelo e Marca do Veículo, Cor, Categorias etc.

**1.5.2 Interfaces de comunicação**

Futuramente o sistema, será integrado com as empresas (concessionárias) para que o agendamento fique automatizado e seja mais rápido e prático o controle de horário e disponibilidade de veículos.

**1.5.3 Modos de operação**

O sistema, em seu primeiro módulo “Concessionária”, irá solicitar um cadastro com informações da empresa, também será necessário o cadastro de todos os veículos disponíveis por cada empresa.

Em seu segundo módulo “Condutor”, irá solicitar um cadastro com informações do usuário para controle e contato, depois de efetuado o cadastro, o condutor poderá fazer um agendamento de Test Drive, após escolher o veículo desejado, dentro das opções disponíveis.

Em seu terceiro módulo “Administrador”, o sistema permitirá que o usuário administrador possa visualizar os pedidos de denúncia para retirada de veículos que não estejam disponíveis para utilização e erros de informação. Além disso, o usuário administrador poderá gerenciar todas as empresas e usuários, além das informações oferecidas através do sistema para visualização.

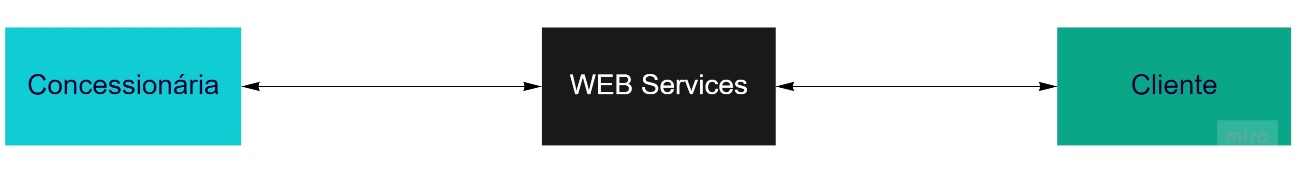
**1.6 Características dos usuários**

Existem três tipos de usuários: as concessionárias, o condutor e o administrador.

**1.7 Requisitos Funcionais**

Para facilitar a compreensão dos requisitos funcionais, foram definidos na tabela abaixo os seguintes termos que serão empregados:

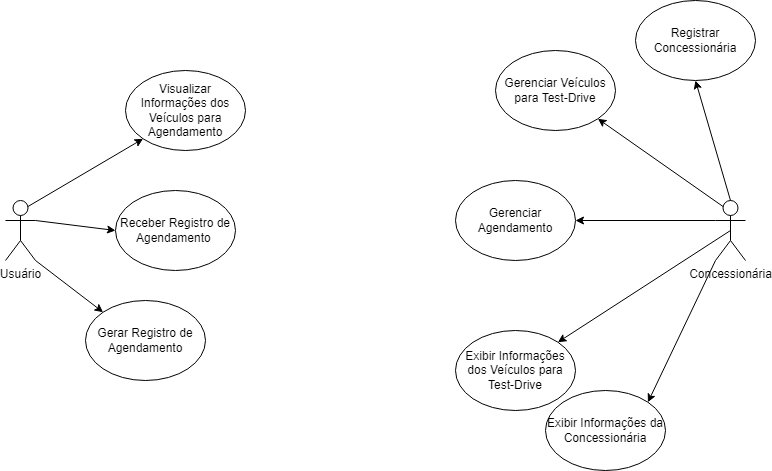
|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição dos Requisitos Funcionais** | **Prioridade** |
| RF-01 | Gerenciamento (Criar, Editar e Excluir) de Agendamento (Usuário/Concessionária) | Essencial |
| RF-02 | Gerar Registro de Agendamento (Usuário) | Essencial |
| RF-03 | Exibir informações dos veículos para Test-Drive (Usuário/Concessionária) | Essencial |
| RF-04 | Receber Registro de Agendamento (Concessionária) | Essencial |
| RF-05 | Gerenciamento (Criar, Editar e Excluir) de Carro para Test-Drive (Concessionária) | Essencial |
| RF-06 | Exibir informações da concessionária (Usuário) | Essencial |

****

**2. Diagrama de caso de uso**

Nessa seção, mostraremos o diagrama de caso de uso para um melhor entendimento dos requisitos do sistema. Além disso, apresentaremos os requisitos não-funcionais.

**2.1 UniDrive**

****

(Imagem 1: UML do sistema)

**2.2 Requisitos Não-Funcionais**

Os requisitos não-funcionais são uma parte importante do sistema, uma vez que demonstram componentes diversos e sua notoriedade em relação ao sistema.

**2.2.1 Requisitos de Portabilidade**

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| RNF/POR-01 | O software deverá funcionar em qualquer navegador em computadores pessoais. |

**2.2.2 Requisitos de Usabilidade**

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| RNF/USA-01 | O software deverá ter uma interface amigável, permitindo aos usuários um rápido entendimento para criar um agendamento. |
| RNF/USA-02 | O software receberá informações de formulários apenas em formato. JSON. |
| RNF/USA-03 | O software deverá gerar registro de agendamentos em formato .PDF e/ou .HTML. |

**2.2.3 Requisitos Legais**

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| RNF/LEG-01 | O software deverá seguir todas as normas e leis impostas pela LGPD. |

**2.2.4 Requisitos de Integração**

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| RNF/INT-01 | O software deverá integrar com a aplicação em Web Service. |

**2.2.5 Requisitos de Implementação**

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| RNF/IMP-01 | O software deverá ser codificado na linguagem PHP. |

**3. Detalhamento de Fluxos**

Nessa seção, mostraremos o detalhamento de cada Requisito Funcional apresentado nesse documento, com foco em exibir sua descrição, o(s) ator(es) relacionados, sua prioridade, os requisitos não-funcionais atrelados, suas pré-condições e sua(s) entrada(s) para realização do requisito.

|  |  |  |
| --- | --- | --- |
| **[RF-01] Gerenciamento de Agendamento (Usuário/Concessionária)** | | |
| **Descrição:** | | Gerenciamento de Agendamento para usuário final e usuário concessionária. |
| **Ator:** | | Usuário e Concessionária |
| **Prioridade:** | | Essencial |
| **Requisitos Não-Funcionais Associados:** | | RNF/LEG-01 |
| **Pré-Condições** | | Conexão com internet. |
| **Entradas (1)** | | Na tela de Gerenciamento de Agendamento para a visão de Concessionária, o sistema deverá realizar as seguintes ações: **I** -Visualizar todos os agendamentos diários para a concessionária específica; **II** - Editar horário e/ou veículo para o agendamento do cliente; **II.I** - Sistema notificará cliente caso a concessionária tenha realizado alguma alteração no agendamento, e o usuário deverá escolher se aceita ou não. Caso não aceite, o agendamento é cancelado; **III** - Excluir agendamento do cliente; **III.I** - O sistema notificará cliente a exclusão do agendamento; **IV** - Concessionária poderá criar um novo agendamento para algum cliente específico ou cliente anônimo (caso esteja presencialmente marcando o Test-Drive). |
| **Entradas (2)** | Na tela de Gerenciamento de Agendamento para a visão de Usuário, o sistema deverá realizar as seguintes ações: **I** - Visualizar todos os agendamentos criados pelo usuário, ordenando entre não realizados e finalizados; **II** - Editar horário e/ou veículo do agendamento baseado no que está disponível pela concessionária; **III** - Excluir agendamento; **III.I** - Sistema deverá notificar a concessionária que o agendamento foi cancelado; **IV** - Usuário poderá criar um novo agendamento, escolhendo entre concessionárias disponíveis em sua região, horários e carros disponíveis. | |
|  |  | |
| **[RF-02] Gerar Registro de Agendamento (Usuário)** | | |
| **Descrição:** | Sistema gera o registro de agendamento para a visualização dele ao usuário final. | |
| **Ator:** | Usuário | |
| **Prioridade:** | Essencial | |
| **Requisitos Não-Funcionais Associados:** | RNF/LEG-01 | |
| **Pré-Condições** | Conexão com internet. | |
| **Entrada** | Na tela após finalizar o registro do agendamento, sistema leva para a tela de finalização do registro, mostrando na tela o registro de agendamento feito. O sistema exibe as seguintes informações: **I** - Informações gerais da concessionária (Nome, Endereço, CNPJ); **II** - Horário definido; **III** - Informações gerais do carro (Nome, Quilometragem); **IV** - Informações gerais do condutor (nome, CNH) | |

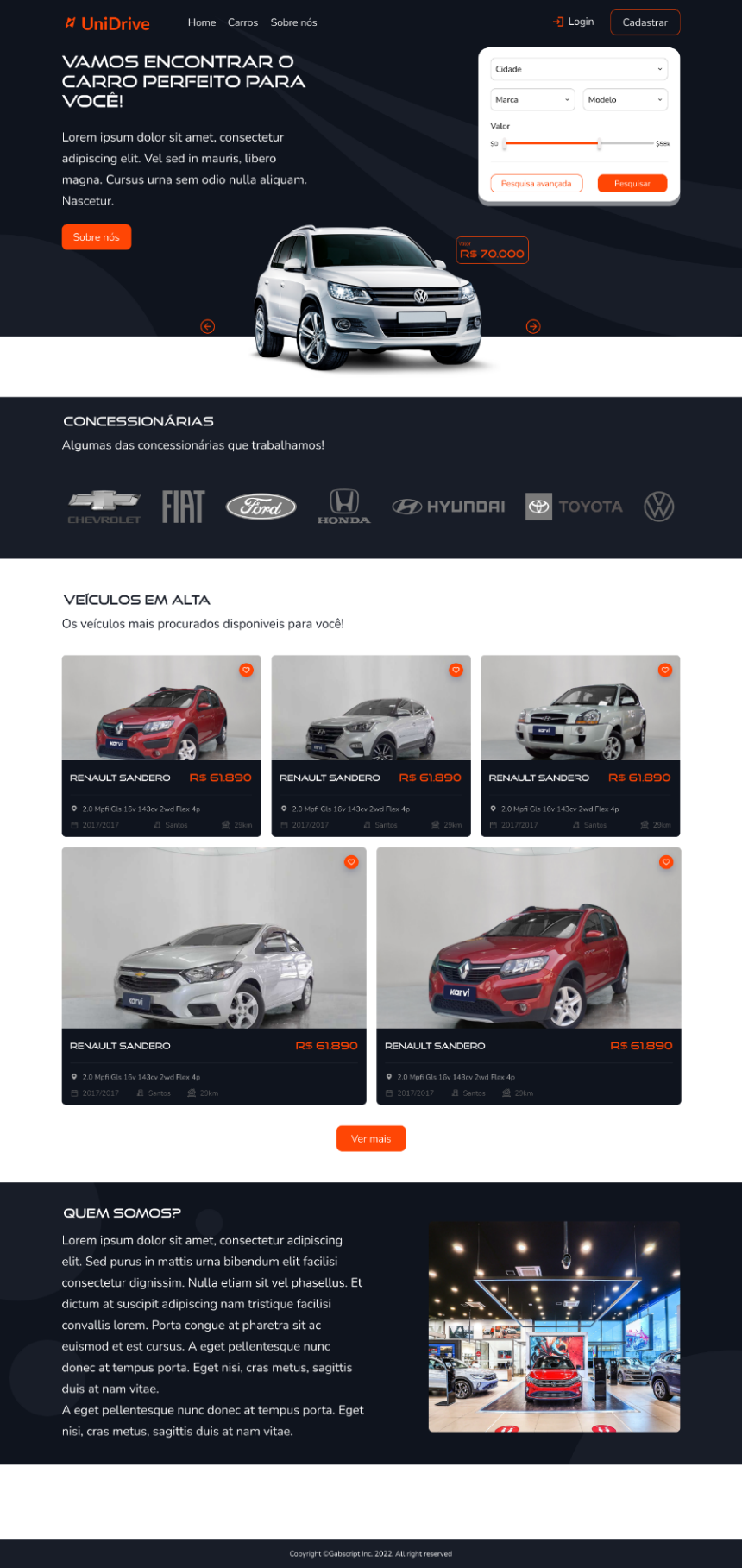
|  |  |
| --- | --- |
| **[RF-03] Exibir informações dos veículos para Test-Drive (Usuário/Concessionária)** | |
| **Descrição:** | Sistema exibe numa página os veículos para realização do Test-Drive para o usuário e concessionária. |
| **Ator:** | Usuário e Concessionária |
| **Prioridade:** | Essencial |
| **Requisitos Não-Funcionais Associados:** | RNF/LEG-01 |
| **Pré-Condições** | Conexão com internet. |
| **Entrada** | Na tela de exibição de informação de veículos, o sistema exibe os veículos disponíveis para Test-Drive filtrados por concessionária escolhida pelo usuário na visão do cliente. |
| Na visão da concessionária, o sistema exibe todos os carros disponíveis para Test-Drive fornecidos pela concessionária. |
|  |  |
| **[RF-04] Receber Registro de Agendamento (Concessionária)** | |
| **Descrição:** | O usuário concessionária recebe o registro de agendamento. |
| **Ator:** | Concessionária |
| **Prioridade:** | Essencial |
| **Requisitos Não-Funcionais Associados:** | RNF/IMP-01; RNF/INT-01 |
| **Pré-Condições** | Conexão com internet. |
| **Entrada** | O sistema recebe o registro de agendamento realização pelo usuário final pelo web service, atrelando a concessionária. |

|  |  |
| --- | --- |
| **[RF-05] Gerenciamento de Carro para Test-Drive (Concessionária)** | |
| **Descrição:** | Gerenciamento de Carro para usuário concessionária. |
| **Ator:** | Concessionária |
| **Prioridade:** | Essencial |
| **Requisitos Não-Funcionais Associados:** | RNF/IMP-01; RNF/INT-01; RNF/LEG-01 |
| **Pré-Condições** | Conexão com internet. |
| **Entrada** | Na tela de Gerenciamento de Carro para a visão de Concessionária, o sistema deverá realizar as seguintes ações: **I** - Na tela de registro de veículos, a concessionária deve registrar seus veículos disponíveis para realização do Test-Drive para visualização e seleção para os clientes. A concessionária deverá informar alguns dados. **I.I** - Nome do veículo; **I.II** - Quilometragem do veículo; **I.III** - Documentação do veículo; **I.IV** - Cor do veículo; **I.V** - Modelo do veículo; **I.VI** - RENAVAM do veículo; **I.VII** - Placa do veículo; **II** -Visualizar todos os carros da concessionária específica; **III** - Editar informações do carro; **IV** - Excluir carro do sistema; **IV.I** - O sistema notificará cliente a exclusão do carro, caso tenha algum agendamento interligado. |
|  |  |
| **[RF-06] Exibir informações da concessionária (Usuário)** | |
| **Descrição:** | O usuário visualiza as informações da concessionária no sistema. |
| **Ator:** | Usuário |
| **Prioridade:** | Essencial |
| **Requisitos Não-Funcionais Associados:** | RNF/IMP-01; RNF/INT-01 |
| **Pré-Condições** | Conexão com internet. |
| **Entrada** | Na tela de informações da concessionária, o sistema exibe informações para o usuário da concessionária pesquisada. |

**4. Protótipos das Telas**

Nessa seção, mostraremos os protótipos das telas do sistema para uma visualização melhor e uma noção mais tangível do sistema.

**4.1 Tela Inicial**

****

(Imagem 2: Tela Inicial)

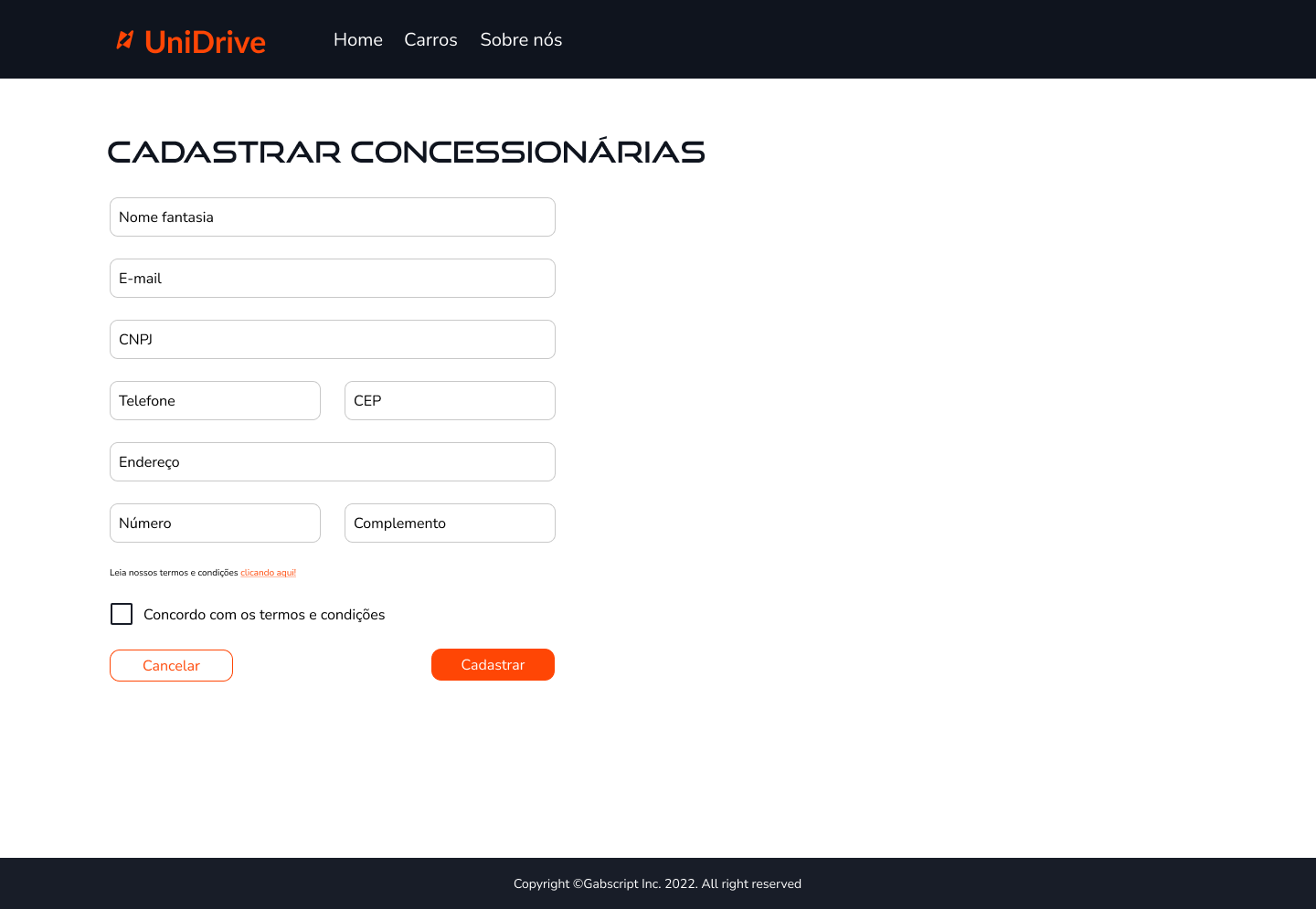
**4.2 Tela de Cadastro de Usuário Condutor e Login**

Interface gráfica do usuário, Site

Descrição gerada automaticamente

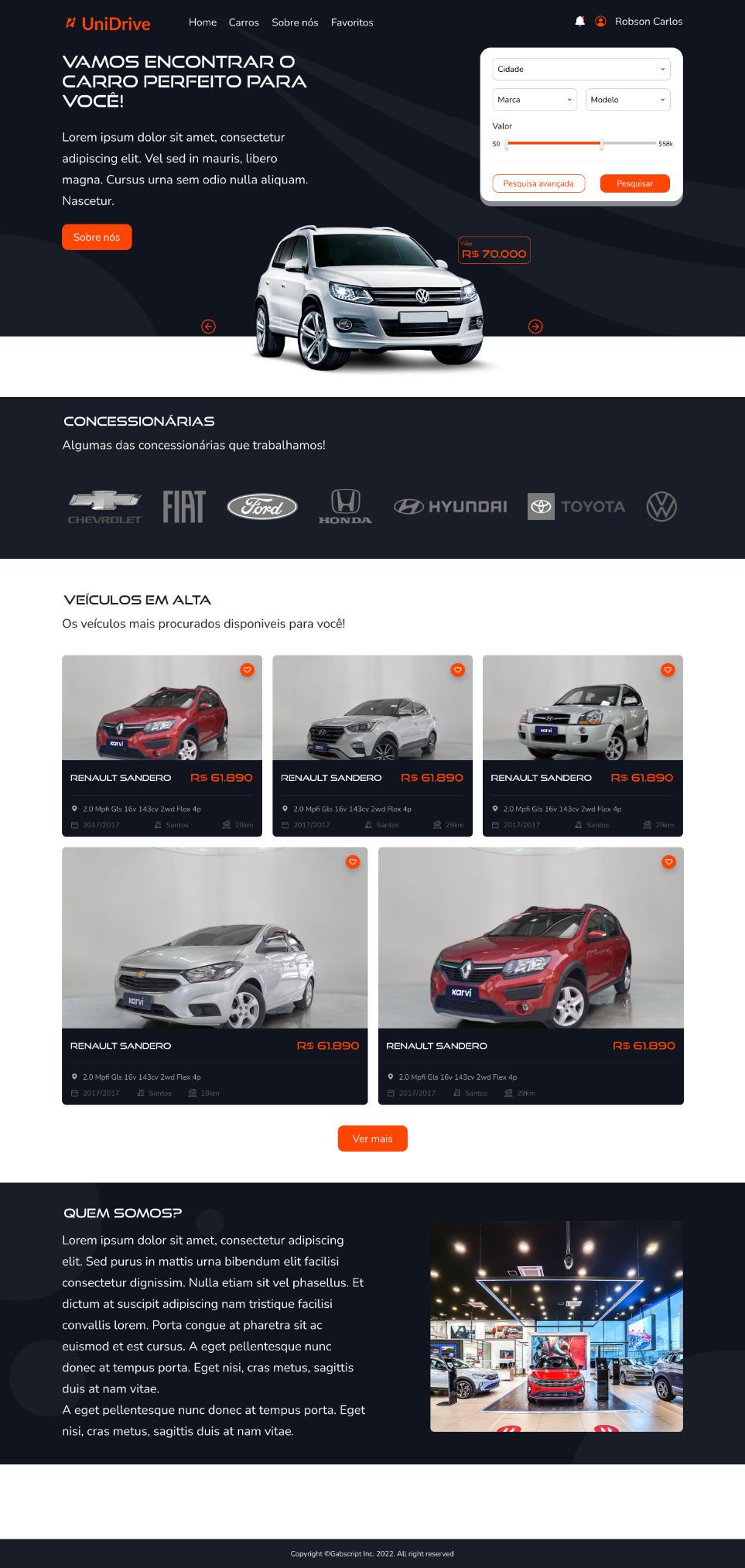
(Imagem 3: Tela de cadastro de usuário condutor e login de usuário condutor/concessionária)

**4.3 Tela Cadastro de Concessionária**



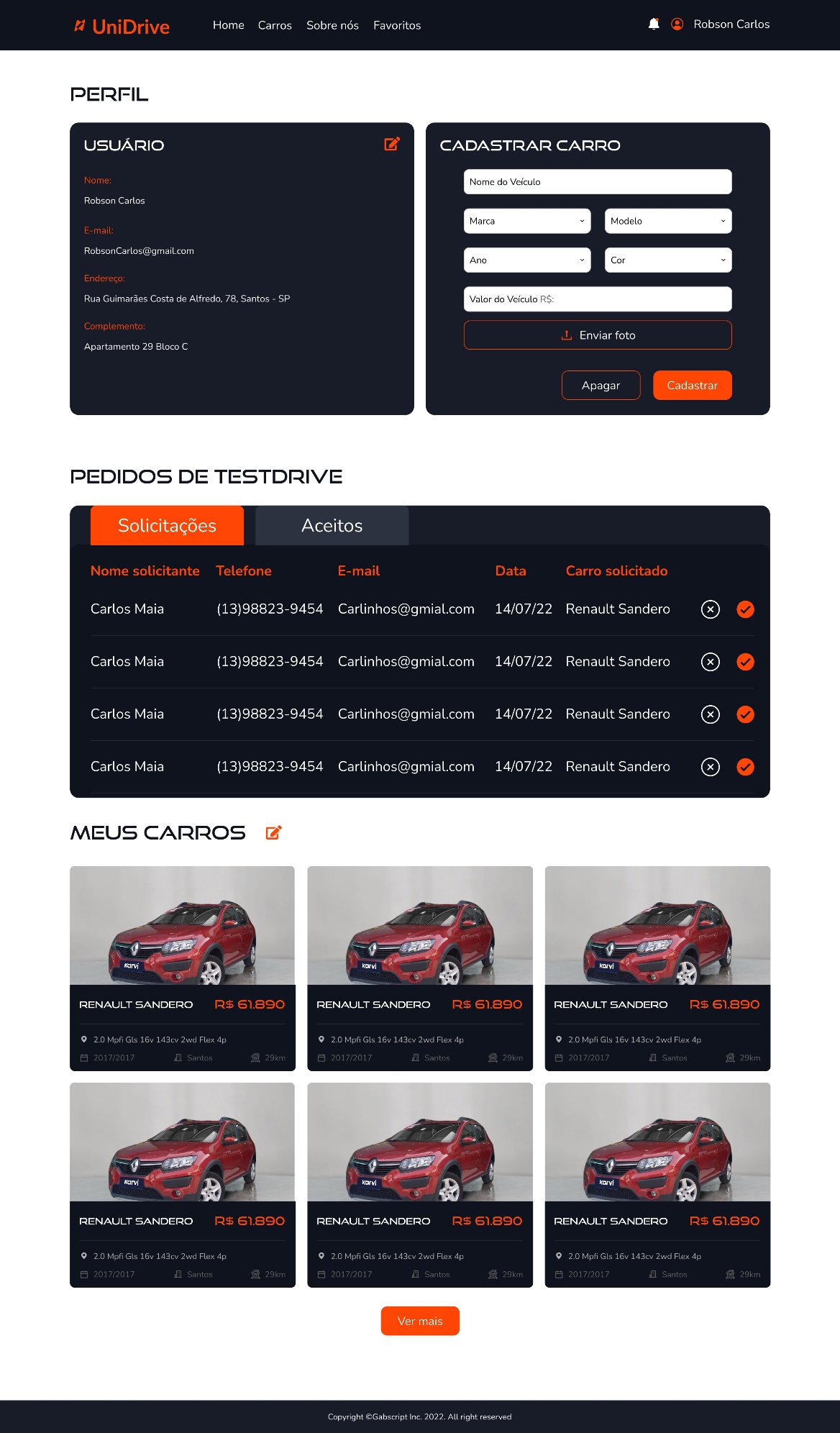
(Imagem 4: Tela de cadastro de concessionária)

**4.4 Tela Inicial do Usuário Condutor**



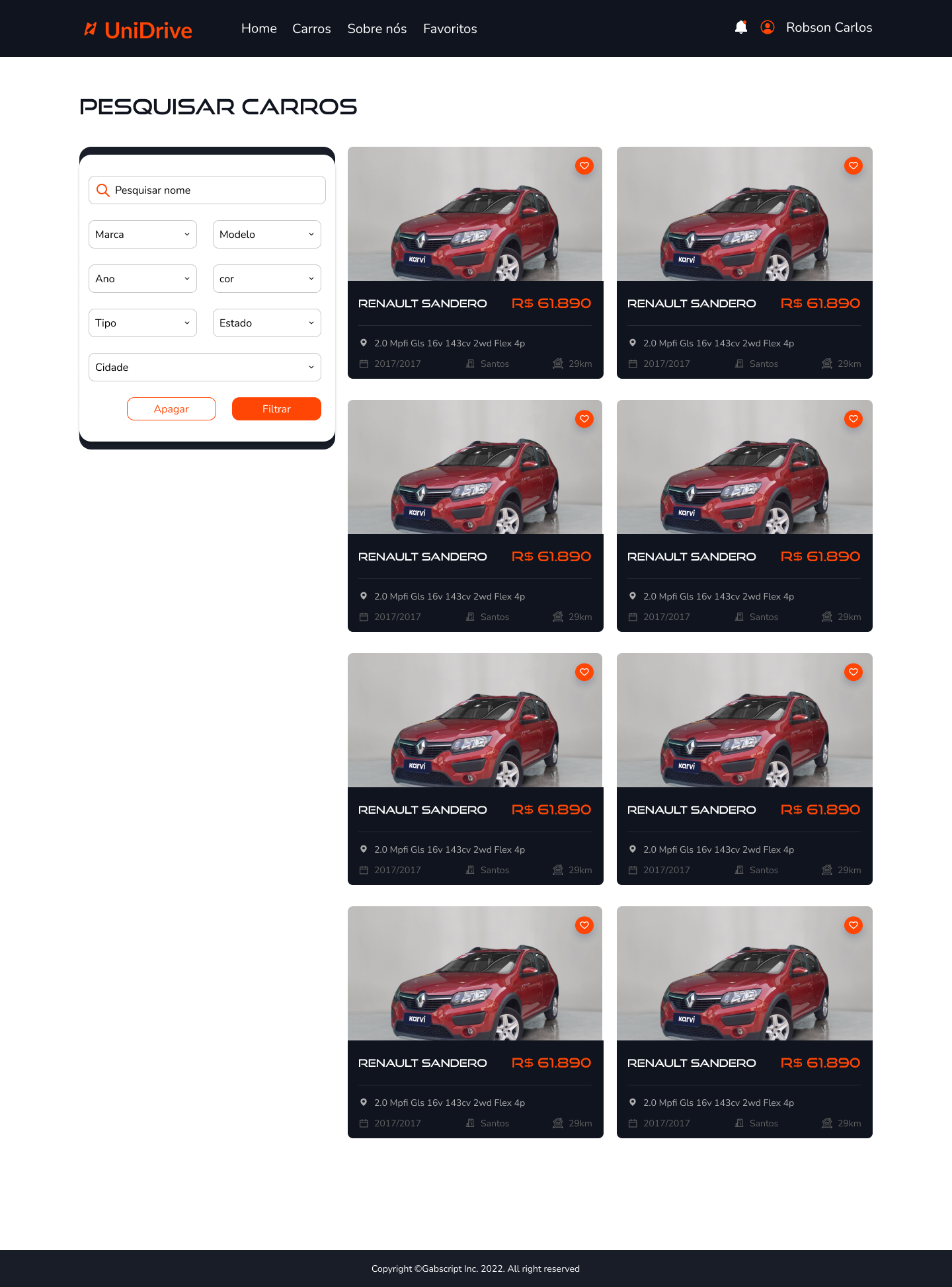
(Imagem 5: Tela Inicial com usuário condutor logado)

**4.5 Tela Cadastro de Veículo**



(Imagem 6: Cadastro de Veículo para usuário concessionária)

**4.6 Tela Pesquisa de Veículo**



(Imagem 7: Pesquisa de veículos para usuário condutor)

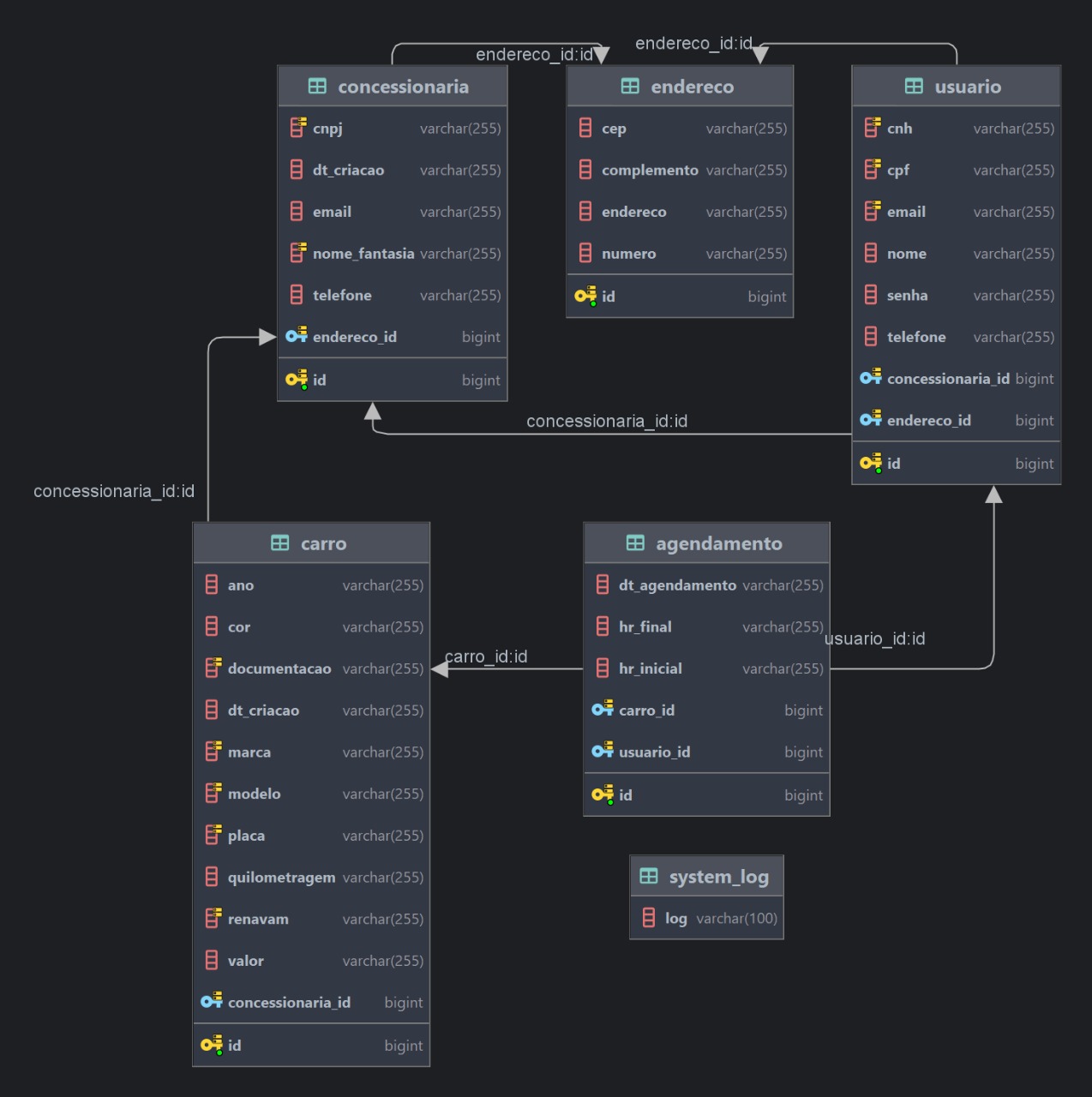
**4.7 Tela Veículo**



(Imagem 8: Tela de visualização de veículo para Test-Drive para usuário condutor)

**5. Banco de Dados**

Nessa seção, mostraremos o modelo de banco de dados definido e estruturado do sistema, apresentando seus devidos relacionamentos com as tabelas. Além disso, será apresentado a explicação de cada tabela e campo do Banco de Dados, seguido com o Modelo Entidade Relacionamento e finalizando com as Triggers, Índices do Banco de Dados e Script do Banco de Dados.

**5.1 Modelo do Banco de Dados** 

(Imagem 9: Modelo de Banco de Dados)

**5.2 Detalhamento das Tabelas e Campos do Banco de Dados**

**5.2.1 Tabela Endereço**

Tabela: Endereco (armazena as informações de endereço do usuário).

Campo 1: endereço\_id (armazena o índice do endereço) [Chave Primária];

Campo 2: cep (armazena o código de endereçamento postal);

Campo 3: complemento (armazena o complemento do endereço);

Campo 4: endereco (armazena a string do endereço);

Campo 5: numero (armazena o número do endereço).

**5.2.2 Tabela Usuário**

Tabela: Usuario (armazena a maior parte das informações do usuário).

Campo 1: usuario\_id (armazena o índice do usuário) [Chave Primária];

Campo 2: cnh (armazena a carteira nacional de habilitação do usuário);

Campo 3: cpf (armazena o cadastro de pessoa física do usuário);

Campo 4: email (armazena o endereço de e-mail do usuário);

Campo 5: nome (armazena o nome do usuário);

Campo 6: senha (armazena a senha em hash do usuário);

Campo 7: telefone (armazena o telefone do usuário).

**5.2.3 Tabela Concessionária**

Tabela: Concessionária (armazena a maior parte das informações da concessionária).

Campo 1: concessionaria\_id (armazena o índice da concessionária) [Chave Primária];

Campo 2: dt\_criacao (armazena a data de criação da concessionária);

Campo 3: email (armazena o endereço de e-mail da concessionária);

Campo 4: nome\_fantasia (armazena o nome fantasia da concessionária);

Campo 5: telefone (armazena o telefone da concessionária).

**5.2.4 Tabela Carro**

Tabela: Carro (armazena todas as informações do carro).

Campo 1: carro\_id (armazena o índice do carro) [Chave Primária];

Campo 2: ano (armazena o ano do carro);

Campo 3: cor (armazena a cor do carro);

Campo 4: documentacao (armazena o certificado de registro e licenciamento do carro);

Campo 5: dt\_criacao (armazena a data de criação do carro);

Campo 6: marca (armazena a marca do carro);

Campo 7: modelo (armazena o modelo do carro);

Campo 8: placa (armazena a placa do carro);

Campo 9: quilometragem (armazena a quilometragem do carro);

Campo 10: renavam (armazena o registro nacional de veículos automotores do carro);

Campo 11: valor (armazena o valor do carro).

**5.2.5 Tabela Agendamento**

Tabela: Agendamento (armazena as informações de agendamento do usuário com o carro).

Campo 1: agendamento\_id (armazena o índice do agendamento) [Chave Primária];

Campo 2: dt\_agendamento (armazena a data do agendamento);

Campo 3: hr\_final (armazena a hora que finaliza o agendamento);

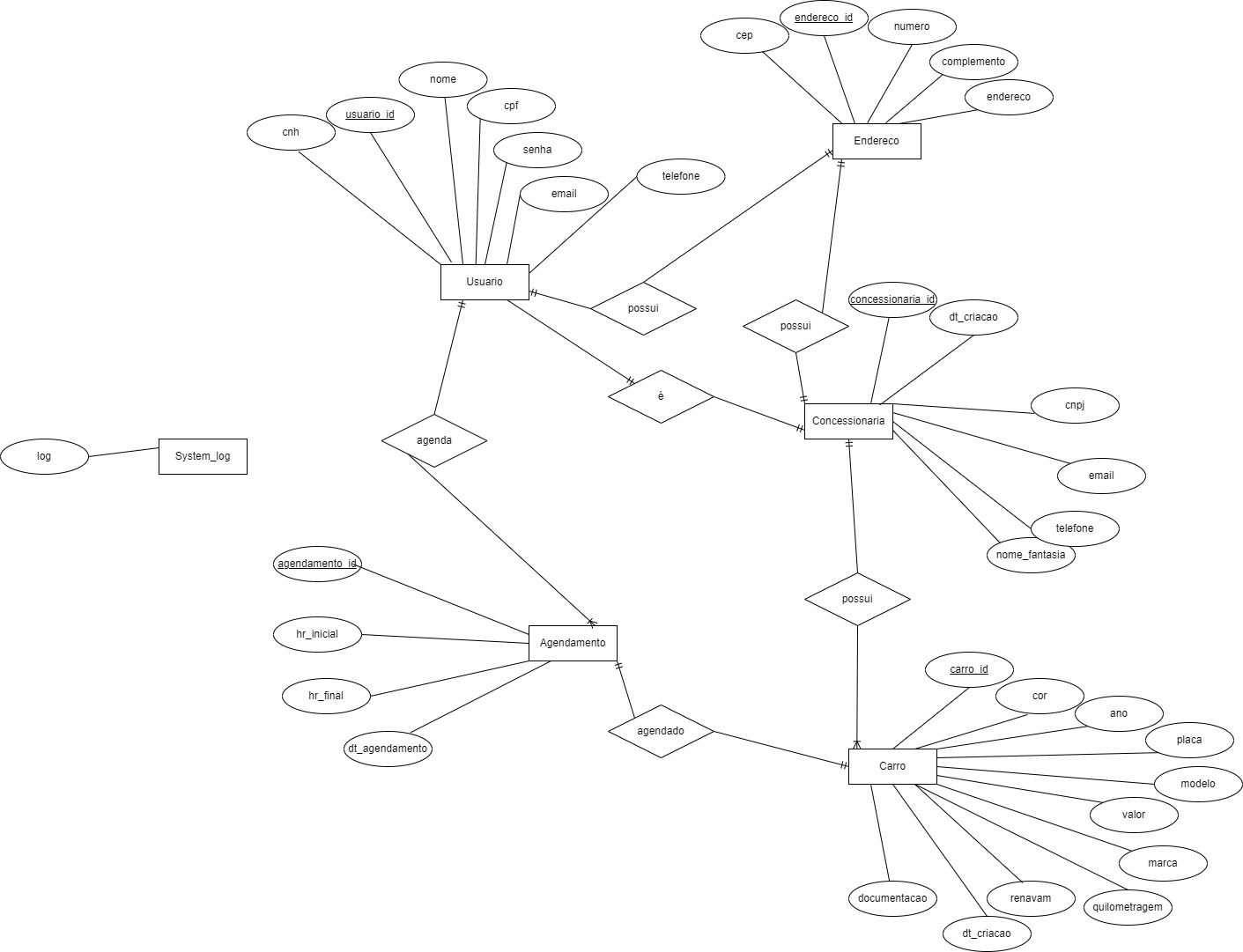
Campo 4: hr\_inicial (armazena a hora que inicia o agendamento).

**5.2.6 Tabela Log do Sistema**

Tabela: System\_log (armazena o log de dados do sistema).

Campo 1: log (armazena os processos de registro de eventos relevantes pelo sistema).

**5.3 Modelo Entidade-Relacionamento do Banco de Dados**



(Imagem 10: Modelo Entidade-Relacionamento do Banco de Dados)

**5.4 Triggers do Banco de Dados**

**5.4.1 Trigger 1**

Trigger criada para realizar um registro da data e hora que um agendamento for excluído:

CREATE TRIGGER delete\_agendamento\_log AFTER DELETE

ON agendamento

FOR EACH ROW

BEGIN

INSERT INTO unidrive.system\_log (log)

values (concat('O Agendamento do ID ', old.id, ' foi deletado ', 'na data de ', NOW()));

end;

**5.4.2 Trigger 2**

Trigger criada para realizar um registro da data e hora que um carro for excluído:

CREATE TRIGGER delete\_carro\_log AFTER DELETE

ON carro

FOR EACH ROW

BEGIN

INSERT INTO unidrive.system\_log (log)

values (concat('O Carro do ID ', old.id, ' foi deletado ', 'na data de ', NOW()));

end;

**5.4.3 Trigger 3**

Trigger criada para realizar um registro da data e hora que um carro for editado:

CREATE TRIGGER update\_carro\_log AFTER update

ON carro

FOR EACH ROW

BEGIN

INSERT INTO unidrive.system\_log (log)

values (concat('O Carro do ID ', new.id, ' foi alterado ', 'na data de ', NOW()));

end;

**5.5 Índices do Banco de Dados**

**5.5.1 Índice 1**

Índice criado para otimizar a busca quando for feita a chamada da funcionalidade buscar carros por um modelo. Ex.: Gol, Onix, HB20.

CREATE INDEX idx\_CARRO\_MODELO ON carro(modelo);

**5.5.2 Índice 2**

Índice criado para otimizar a busca quando for feita a consulta nos carros de uma marca. Ex.: KIA, Volkswagen.

CREATE INDEX idx\_CARRO\_MARCA ON carro(marca);

**5.5.3 Índice 3**

Índice criado para otimizar a busca quando for feita a consulta nos usuários usando o e-mail.

CREATE INDEX idx\_USUARIO\_EMAIL ON usuario(email);

**5.6 Script do Banco de Dados**

DROP TABLE IF EXISTS `agendamento`;

CREATE TABLE `agendamento` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`dt\_agendamento` varchar(255) DEFAULT NULL,

`hr\_final` varchar(255) DEFAULT NULL,

`hr\_inicial` varchar(255) DEFAULT NULL,

`carro\_id` bigint DEFAULT NULL,

`usuario\_id` bigint DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `FK4hbdv4vq84ncod446xql38r8q` (`carro\_id`),

KEY `FKfbyksmvbd2yav8b059npbeif` (`usuario\_id`),

CONSTRAINT `FK4hbdv4vq84ncod446xql38r8q` FOREIGN KEY (`carro\_id`) REFERENCES `carro` (`id`),

CONSTRAINT `FKfbyksmvbd2yav8b059npbeif` FOREIGN KEY (`usuario\_id`) REFERENCES `usuario` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

DROP TABLE IF EXISTS `carro`;

CREATE TABLE `carro` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`ano` varchar(255) DEFAULT NULL,

`cor` varchar(255) DEFAULT NULL,

`documentacao` varchar(255) DEFAULT NULL,

`dt\_criacao` varchar(255) DEFAULT NULL,

`marca` varchar(255) DEFAULT NULL,

`modelo` varchar(255) DEFAULT NULL,

`placa` varchar(255) DEFAULT NULL,

`quilometragem` varchar(255) DEFAULT NULL,

`renavam` varchar(255) DEFAULT NULL,

`valor` varchar(255) DEFAULT NULL,

`concessionaria\_id` bigint DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_od3tctt9ma2xgjnn9gubgbgbd` (`documentacao`),

UNIQUE KEY `UK\_5uv3fisdwp2pmj1357rivswea` (`placa`),

UNIQUE KEY `UK\_r16ccovgniyke4xfb3yj7rbt7` (`renavam`),

KEY `FK4g61a0hsgngyya53uyuove4j9` (`concessionaria\_id`),

KEY `idx\_CARRO\_MODELO` (`modelo`),

KEY `idx\_CARRO\_MARCA` (`marca`),

CONSTRAINT `FK4g61a0hsgngyya53uyuove4j9` FOREIGN KEY (`concessionaria\_id`) REFERENCES `concessionaria` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

DROP TABLE IF EXISTS `concessionaria`;

CREATE TABLE `concessionaria` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`cnpj` varchar(255) DEFAULT NULL,

`dt\_criacao` varchar(255) DEFAULT NULL,

`email` varchar(255) DEFAULT NULL,

`nome\_fantasia` varchar(255) DEFAULT NULL,

`telefone` varchar(255) DEFAULT NULL,

`endereco\_id` bigint DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_kf5x9k6dpnqipc42q22diekyp` (`cnpj`),

UNIQUE KEY `UK\_fuvi7fv94why041fau1l4mmbv` (`nome\_fantasia`),

KEY `FKklv2xg45b44wcimecvhivkffx` (`endereco\_id`),

CONSTRAINT `FKklv2xg45b44wcimecvhivkffx` FOREIGN KEY (`endereco\_id`) REFERENCES `endereco` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

DROP TABLE IF EXISTS `endereco`;

CREATE TABLE `endereco` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`cep` varchar(255) DEFAULT NULL,

`complemento` varchar(255) DEFAULT NULL,

`endereco` varchar(255) DEFAULT NULL,

`numero` varchar(255) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

DROP TABLE IF EXISTS `system\_log`;

CREATE TABLE `system\_log` (

`log` varchar(100) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

DROP TABLE IF EXISTS `usuario`;

CREATE TABLE `usuario` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`cnh` varchar(255) DEFAULT NULL,

`cpf` varchar(255) DEFAULT NULL,

`email` varchar(255) DEFAULT NULL,

`nome` varchar(255) DEFAULT NULL,

`senha` varchar(255) DEFAULT NULL,

`telefone` varchar(255) DEFAULT NULL,

`concessionaria\_id` bigint DEFAULT NULL,

`endereco\_id` bigint DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_bno4gmt60moxf0miou6pvas3f` (`cnh`),

UNIQUE KEY `UK\_692bsnqxa8m9fmx7m1yc6hsui` (`cpf`),

UNIQUE KEY `UK\_5171l57faosmj8myawaucatdw` (`email`),

KEY `FKrolquucnw3e97rp0tnsjbiux` (`concessionaria\_id`),

KEY `FK8fl5dxscva53gw12f19q6qxf8` (`endereco\_id`),

CONSTRAINT `FK8fl5dxscva53gw12f19q6qxf8` FOREIGN KEY (`endereco\_id`) REFERENCES `endereco` (`id`),

CONSTRAINT `FKrolquucnw3e97rp0tnsjbiux` FOREIGN KEY (`concessionaria\_id`) REFERENCES `concessionaria` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

DROP TABLE IF EXISTS `usuario\_perfis`;

CREATE TABLE `usuario\_perfis` (

`usuario\_id` bigint NOT NULL,

`perfis\_id` bigint NOT NULL,

KEY `FK7bhs80brgvo80vhme3u8m6ive` (`perfis\_id`),

KEY `FKs91tgiyagbilt959wbufiphgc` (`usuario\_id`),

CONSTRAINT `FK7bhs80brgvo80vhme3u8m6ive` FOREIGN KEY (`perfis\_id`) REFERENCES `perfil` (`id`),

CONSTRAINT `FKs91tgiyagbilt959wbufiphgc` FOREIGN KEY (`usuario\_id`) REFERENCES `usuario` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

**6. Diagrama de Arquitetura**

Nessa seção, mostraremos o diagrama de arquitetura para ser possível compreender melhor a ideia do sistema. A arquitetura é baseada no modelo MVC (Model-View-Controller), utilizando-se validação a partir de token para realizar as ações do sistema.



(Imagem 11: Arquitetura do Sistema em modelo MVC)

**7. Tecnologias e Frameworks**

Nessa seção, apresentaremos as tecnologias utilizadas na criação do sistema e os frameworks utilizados.

Para realização do *Front-End* da aplicação, utilizaremos o framework *Laravel*, sua base é feita na linguagem de programação PHP e visamos com ele uma maior facilidade para criar e atualizar quando necessário.

Em relação ao *Back-End* da aplicação, utilizaremos a linguagem de programação Java, juntamente com seu framework *Springboot*, devido ao fato de ter maiores recursos no mercado e a melhor escolha para realização dessa aplicação.

Além disso, visando a segurança do sistema, juntamente com sua integração a partir de APIs, utilizaremos o framework *Spring Security*, e para acesso e validação de *Token* utilizaremos o *JSON Web Token* (JWT).

No mais, para armazenamento de dados, utilizaremos o banco de dados *MySQL*, e para manipulação de dados, utilizaremos o framework *Spring Data*.

**8. Infraestrutura de Host**

Nessa seção, apresentaremos a infraestrutura escolhida para hospedagem em nuvem do sistema. Criamos a melhor escolha para armazenar o sistema e suportá-lo, uma vez que o sistema visa atendimento em nível nacional e atender uma quantidade não muito grande de requisições simultâneas.

O plano escolhido para a infraestrutura é formado pelos seguintes componentes:

* Localização: *North America, Canada, Beauharnois (BHS)*
* Processador: 8 vCore
* Memória: 8 GB
* Armazenamento: 160 GB SSD NVMe
* Banda Larga Pública: 2 Gbps

A infraestrutura da aplicação será um VPS, hospedada pela empresa OVH ([www.ovhcloud.com](http://www.ovhcloud.com)) inicialmente com o plano de 2 anos dentro do planejamento de negócio, sendo possível um escalonamento caso necessário.

A VPS inclui 8 vCores (Virtual Cores), com 8 GB de memória RAM sendo expansível até 32 GB, armazenamento de 160 GB SSD NVMe sendo expansível até 640 GB (Imagem I01), imagem Ubuntu como base de sistema operacional e com localização “*North America, Canada, Beauharnois (BHS)*”.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

(Imagem 12: Configuração da instância de VPS)

Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente

(Imagem 13: Seleção do sistema operacional)

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

(Imagem 14: Seleção da localização do datacenter)

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

(Imagem 15: Teste de MS)

**9. Estrutura Analítica do Projeto (EAP)**

Nessa seção, apresentaremos a EAP, utilizado principalmente para facilitar uma visibilidade geral do projeto para uma análise melhor do conteúdo e entregáveis que o projeto possui.



(Imagem 16: Estrutura Analítica do Projeto)

**10. Links Importantes**

Nessa seção, apresentaremos alguns links que são importantes para o projeto. Abaixo estará o link do site hospedado e o link da documentação do Backend.

**10.1 URL do site hospedado**

Esta URL é voltada ao sistema completo funcionando, os fluxos totais do sistema que está hospedada online:

https://unidrive.works/

**10.2 URL da Documentação do Backend**

Esta URL é voltada a documentação dos *endpoints* do sistema, das entidades, mostrando a entrada de dados no *body* nas requisições via JSON e respostas esperadas pelo sistema:

https://api.unidrive.works/