

CENTRO UNIVERSITÁRIO UNIOPET
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

EDUARDO DANIEL POLIDORIO
FELIPE GUSO TAVERNA
PABLO BERGERRON TIBURCIO
RODRIGO GABRIEL DARTORA

PROPOSTA PROJETO INTEGRADOR

CURITIBA – PR

2023

EDUARDO DANIEL POLIDORIO
FELIPE GUSSO TAVERNA
PABLO BERGERRON TIBURCIO
RODRIGO GABRIEL DARTORA

PROPOSTA PROJETO INTEGRADOR

Proposta apresentada ao curso de Análise e Desenvolvimento de Sistemas e Gestão de Tecnologia da Informação envolvendo as matérias de Engenharia de Requisitos, Análise e Projeto Orientado a Objetos e Fundamentos de Data Science, como requisito parcial para aprovação nas Disciplinas.

CURITIBA – PR
2023

Sumário

1. INTRODUÇÃO	04
2. OBJETIVO GERAL	05
3. OBJETIVOS ESPECÍFICOS	05
4. JUSTIFICATIVA	07
5. METODOLOGIA E DESENVOLVIMENTO	07
5.1 Diagrama de Casos de Uso	07
5.2 Diagrama de Atividades	08
5.3 Diagrama de Sequência	09
6. FRONTEND	10
7. BACKEND	11
8. ALGORITIMOS	11
9. BANCO DE DADOS	12
9.1 Tabela cliente	14
9.2 Tabela vendedores	14
9.3 Tabela forma de pagamento	14
9.4 Tabela marca	15
9.5 Tabela modelo	15
9.6 Tabela veiculos	16
9.7 Tabela vendas	16
9.8 Comandos MYSQL/PHP	17
10. DESENVOLVIMENTO	17
10.1 Área de Login	17
10.2 Dashboard.....	19
10.3 Barra de Navegação	19
10.4 Clientes	20
10.5 Veículos	22
10.6 Financiamento.....	23

11. REDES	24
11.1 FortGate 500e	24
11.2 FortAnalyzer	25
11.3 Fortmail	27
12. FUNDAMENTOS DE TI.....	28
12.1 Custos da Solução em Nuvem	29
12.2 Requisitos mínimos para implementação do sistema Web	29
12.3 Características do equipamento utilizado para desenvolvimento	29
13. REFERÊNCIAS	30

1 . INTRODUÇÃO

A adoção da tecnologia da informação no ambiente de trabalho aumentou nos últimos anos, não só impulsionada pelo maior acesso a sistemas de gestão e controle, mas também pelos benefícios proporcionados em sua implementação. Através do uso destes sistemas, tarefas repetitivas que antes eram feitas de forma manual hoje podem ser automatizadas, tornando o processo de trabalho mais rápido, confiável e menos custoso. Este trabalho tem como objetivo o desenvolvimento de um sistema capaz de gerenciar vendas, além da gestão de balanços monetário e de estoque de uma concessionária. Após o levantamento do processo de trabalho e necessidades do usuário final, foram criados os requisitos que embasam as funcionalidades presentes no sistema proposto chamado de CarPrime, desenvolvido como um Sistema Web utilizando arquitetura de micro serviços. O resultado final mostra um sistema capaz de atender as necessidades diárias de operações como cadastro de

clientes, veículos e funcionários, além da possibilidade de controle do estoque e balanço monetário.

2 . OBJETIVO GERAL

Tem-se como proposta a criação de um sistema de controle para concessionária que seja capaz de cobrir tarefas de controle de estoque, gerenciamento de clientes, vendas de veículos e custos e geração de relatórios de estatísticas.

3 . OBJETIVOS ESPECÍFICOS

Apresentar um sistema capaz de enviar relatórios automáticos para clientes, com atualizações e novas ofertas, assim como serviços oferecidos dentro do prazo de garantia do veículo adquirido. Para alcançar este objetivo, os seguintes objetivos específicos são definidos:

- Levantamento e análise de requisitos;
- Criação de modelo de banco;
- Pesquisa de arquiteturas para sistemas;
- Definição das tecnologias de desenvolvimento para o projeto;
- Desenvolvimento de versão;
- Realização de revisões e testes;
- Criação de versão protótipo.

Requisitos Funcionais:

RF0001 - Cadastro:	O sistema vai cadastrar; nome, contato, e-mail e CPF/CNPJ do cliente.
RF0002 - Cadastro do administrador:	O sistema deve realizar o cadastro do administrador.
RF0003 - Cadastro do usuário:	O sistema deve realizar o cadastro do usuário.

RF0004 - Cadastro dos produtos:	O sistema deve realizar o cadastro dos produtos.
RF0005 - Editar administrador:	O sistema deve editar o cadastro do administrador.
RF0006 - Editar usuário:	O sistema deve editar o cadastro do usuário.
RF0007 - Editar produtos:	O sistema deve editar o cadastro dos produtos.
RF0008 - Login:	Para efetuar o login o sistema irá pedir e-mail e senha
RF0009 - Fazer venda:	O sistema vai realizar a soma dos produtos por unidade.
RF0010 - Finalizar venda:	O sistema irá finalizar a venda.
RF0011 - Listar estoque:	O sistema vai exibir em tela os produtos que estão no estoque.
RF0012 - Test drive:	O sistema deve permitir agendamento de test - drive.
RF0013 - Relatórios de vendas:	O sistema exibirá relatórios, mensais, trimestrais, semestrais e anuais.
RF0014 - Relatórios de estoque:	O sistema exibirá o estoque disponível para venda.
RF0015 - Relatórios:	O sistema exportará o relatório para uma planilha em Excel.

Requisitos Não Funcionais:

RNF001 - Usabilidade:	O sistema deve ser fácil de usar e entender pelos funcionários da concessionária
RNF002 - Desempenho:	Agilidade, o sistema deve ser capaz de lidar com um grande número de cadastros e transações de forma rápida e eficiente
RNF003 - Segurança:	O sistema deve garantir a proteção das informações dos clientes e da concessionária

RNF004 - Escalabilidade:	O sistema deve ser capaz de suportar um grande número de usuários e transações sem comprometer o desempenho
RNF005 - Disponibilidade:	O sistema deve estar disponível para uso 24 horas por dia, 7 dias por semana, com o mínimo de tempo de inatividade planejado para manutenção e atualizações

4 . JUSTIFICATIVA

Através do sistema, faz-se possível a substituição do uso de planilhas de controle e preenchimentos manuais de contratos novos, contribuindo com a evolução e crescimento do modelo de negócios da empresa, possibilitando através da informatização obter agilidade e maior segurança no armazenamento de informações.

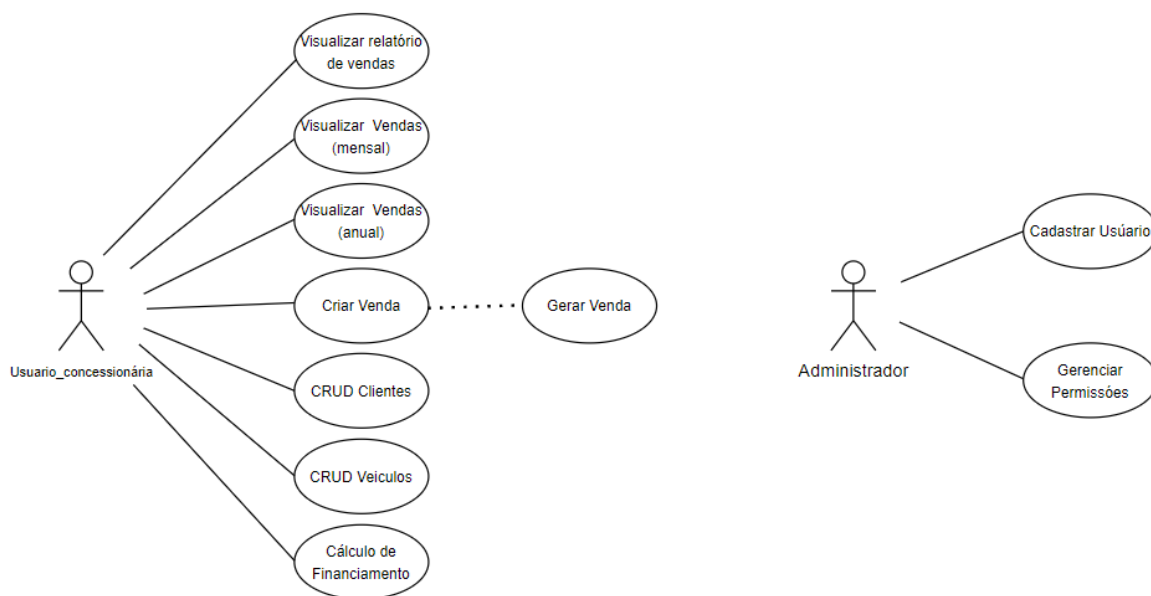
5 . METODOLOGIA E DESENVOLVIMENTO

O método de desenvolvimento utilizado foi o cascata, que visa um desenvolvimento dirigido por planos, na qual se planeja todas as etapas do processo antes de começar de fato a desenvolver. Em primeiro plano será abordado o design e estrutura do sistema, seguido de seu desenvolvimento_utilizando as linguagens HTML5, CSS3 e JavaScript e, por fim os devidos testes para verificação e correção de bugs e falhas.

5.1 Diagrama de Casos de Uso:

Os casos de uso são documentados por um diagrama de casos de uso de alto nível, que é uma característica fundamental da Unified Modeling Language (UML). Em sua forma mais simples, ele identifica os atores envolvidos em uma interação, dando nome a essa interação, suplementando-as com informação adicionais que descrevem a interação com o sistema.

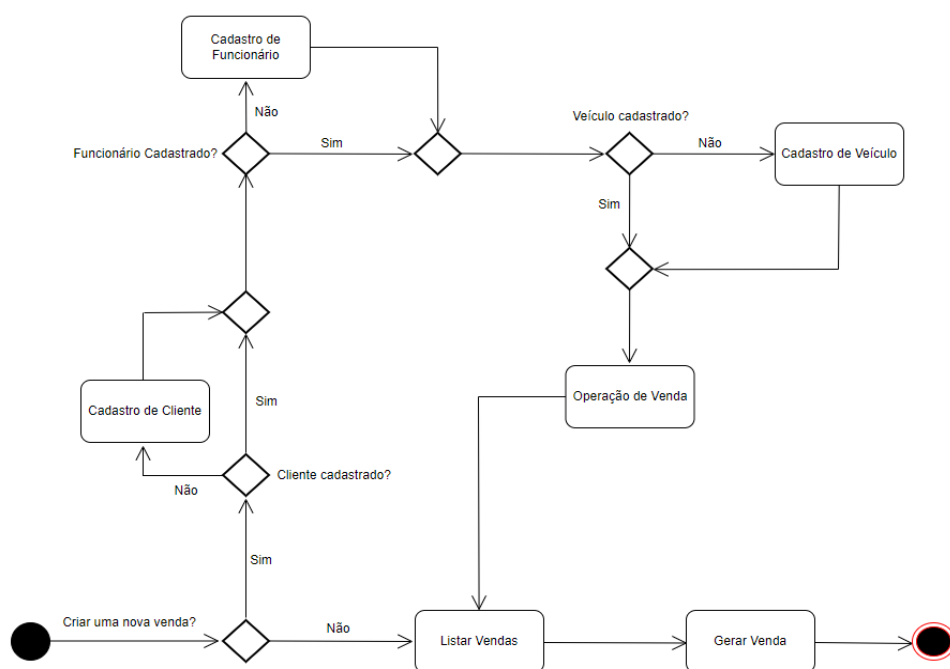
Na Figura são apresentados os casos de uso desenvolvidos no sistema, sendo que existem dois atores: Usuário_concessionária e Administrador. De forma básica, o Usuário_concessionária irá fazer toda a operação do sistema no que diz respeito à realização de cadastros, geração do contrato e visualização dos relatórios. Enquanto que a figura do Administrador é responsável por gerenciar os usuários, seja adicionando ou removendo, quanto distribuindo permissões.



5.2 Diagrama de Atividades:

O diagrama de atividade tem como finalidade representar um processo e o fluxo de controle presente no sistema que o usuário ou o próprio sistema deve executar para chegar a um objetivo, explicitando as atividades e suas dependências. A Figura demonstra o fluxo necessário para a geração de uma venda, sendo que seu caminho mais curto quando a venda já está cadastrada no sistema.

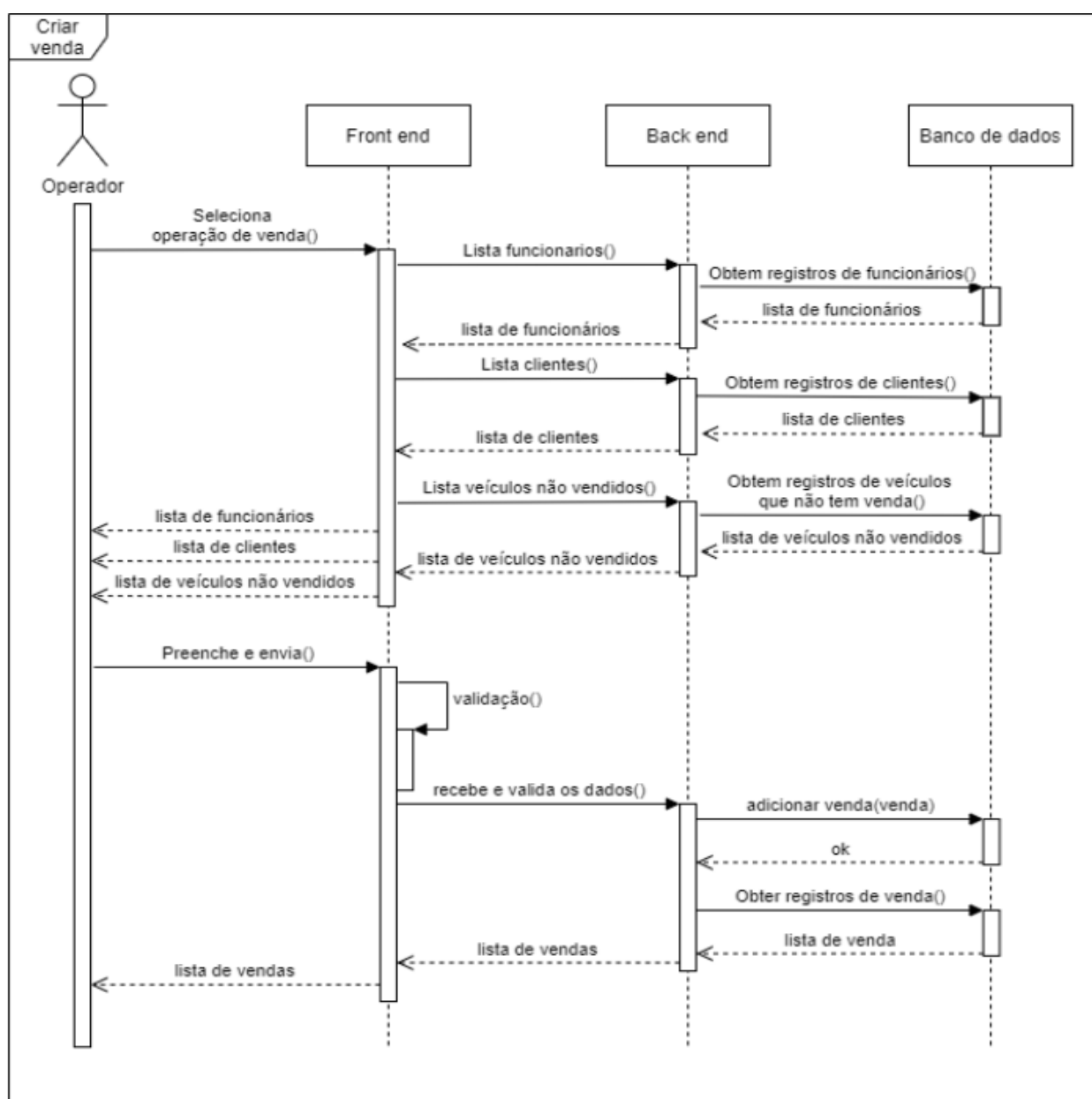
Caso a venda não esteja cadastrada no sistema, o usuário tem como pré-requisito o cadastro de um cliente, um funcionário e um veículo para operar a venda. Seguindo o fluxo de criar uma nova venda, o usuário tem como opção fazer o cadastro de cada um dessas entidades, ou utilizar as já persistidas no sistema. Após a operação, o usuário é redirecionado para a “Lista de vendas”, na qual a partir de uma venda cadastrada poderá gerar seu contrato.



5.3 Diagrama de Sequência:

Os diagramas de sequência são utilizados como ferramenta principal de modelagem das interações entre os atores e objetos, além das interações entre os próprios objetos. A Figura mostra como é feita a criação de uma venda por um operador utilizando-se o sistema, interagindo com três principais componentes: frontend, backend e banco de dados.

A partir da seleção da operação de venda, o frontend envia requisição para o backend de 3 listas: funcionários, clientes e veículos, sendo os veículos aqueles que ainda não foram vendidos. A partir disso, o backend faz requisição no banco de todos esses dados, e os retorna para o frontend, então sendo exibidos para o Operador. Após o preenchimento pelo Operador dos dados da venda, os mesmos são enviados para o backend que ao receber os converte, adicionando-os ao banco de dados. Então, o backend obtém a lista de todas as vendas e as retorna para o backend, que então é exibida ao usuário



6 . FRONTEND

Cada componente é criado com seu template HTML e um arquivo CSS, que contém todos os estilos utilizados exclusivamente por ele. Como configuração, foi criado um arquivo de estilos por meio do uso da Syntactically Awesome Style Sheets (SASS) 16, que é uma linguagem para auxiliar a criação de estilos. Para as interações com o usuário e formatações dinâmicas faz-se uso da linguagem JavaScript, assim como para validações dentro de campos de preenchimento.

7. BACKEND

A integração do Front-end da aplicação com o banco de dados é feita através da linguagem de programação PHP, adotando-se o modelo PDO(PHP Data Object). A versão do PHP utilizada na aplicação é a versão 8.1.5, sendo a versão mais atual da linguagem.

8 . ALGORITIMOS

O sistema apresenta em sua codificação a estrutura de pilha. Pilhas são estruturas de dados que armazenam os elementos em um formato sequencial, empilhando um item acima do outro. Estas estruturas permitem “empilhar” os itens que serão armazenados e “desempilhar” estes elementos da pilha quando precisarmos removê-lo.

```
$explode=explode('.',$_FILES['arquivo']['name']);
$aponta=end($explode);
$extensao=strtolower($aponta);
if(array_search($extensao,$_UP['extensao'])
===false){
    echo "Extensão não aceita";
    exit();
}
```

O código acima pega como referência o último elemento dentro do array contendo todas as extensões aceitas, seguindo o conceito LIFO(Last in First Out) associado a pilhas.

Dentro da codificação PHP, foi-se utilizado a função recursiva para somatório de valores.

```

function jurosComposto($valor, $taxa, $parcelas) {
    $taxa = $taxa / 100;

    $valParcela = $valor * pow((1 + $taxa), $parcelas);
    $valParcela = number_format($valParcela / $parcelas, 2, ",", ".");

    return $valParcela;
}
$valorParcelaComposto = jurosComposto($valor, $taxa, $parcelas);

//Funcao Recursiva para somatorio do valor final do financiamento
function vl_final($valorParcelaComposto, $parcelas){
    if($parcelas==0){
        return 1;
    }else{
        return $valorParcelaComposto + vl_final($valorParcelaComposto, $parcelas-1);
    }
}

@$vl_final = vl_final($valorParcelaComposto, $parcelas);

```

A função `vl_final` soma todos os valores obtidos na função `jurosCompostos`, decrementando a cada interação uma unidade da variável `$parcelas`, sendo chamada a própria função a cada interação. Quando a variável `$parcelas` chega ao seu limite 1, a função se encerra e exporta para o escopo global o somatório da variável `$valorParcelaComposto`, a qual é armazenada na variável `$vl_final`.

9 . BANCO DE DADOS

O SGBD é a sigla que se refere a Sistema Gerenciador Banco de Dados, onde o sistema é usado para armazenar um conjunto de dados que temos o intuito de guardar por um longo prazo. Todo SGBD exige um servidor exclusivo para ele, pois dependendo da aplicação no qual está sendo usada, a entrada e saída de dados é extremamente volumosa para ser compartilhada com outros serviços.

Quando levamos em consideração a performance e segurança dos dados, é recomendado trabalhar com discos físicos, sendo um deles destinado apenas para o banco propriamente dito e o outro apenas para arquivos de log. Mas afinal, o que é log em banco de dados? Log é o local onde se registra todas as transações do banco, seja modificações, aplicações, inserções ou delete. O Log é um ambiente extremamente crítico em situações de perda de dados, pois é nele que é feito o primeiro procedimento para reaver os dados.

O primeiro banco de dados desenvolvido foi em 1960 e houve diversas personalizações e melhorias desde o seu surgimento. O mercado está amplo e com vários

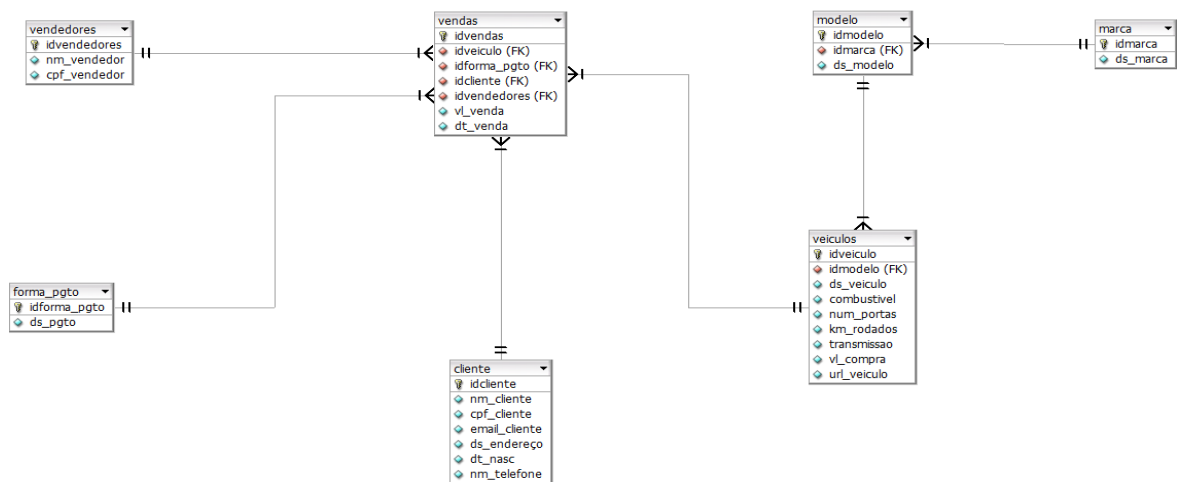
concorrentes disponibilizando prestações de serviços diferenciados. O nosso sistema para concessionárias vamos utilizar o MySQL, pois nele conseguimos elaborar servidor local sem custo para os desenvolvedores. Além dessa facilidade para a criação, o MySQL é indicado para aplicações web, pois o mesmo suporta milhares de consulta e transações simultâneas.

Para o Carprime será necessário desenvolver diversas tabelas de cadastro, consultas e relatórios. O sistema será de manipulação interna das concessionárias e clientes, por esse motivo, aplicaremos tabela cadastro, sendo designada para clientes, endereço, telefone de contato, CPF, RG e email.

A concessionária utilizará a tabela com nomenclatura Concessionária também com ligações as tabelas veículos, tipo do veículo, marca, modelo, motorização, tipo do câmbio, cor, seminovo ou zero km. As principais tabelas Cadastro e Concessionárias serão interligadas, pois com os dados de ambas conterão informações salientes para ambos usuários, concessionária e clientes.

9.1 Modelo Conceitual:

O modelo conceitual do sistema leva consigo a regra de negócio da aplicação, onde a tabela central se da pela tabela Vendas. Cada tabela filho se liga a tabela pai(Vendas) por meio de sua relação específica, sendo ela um pra muitos, muitos pra um ou um pra um. O sistema conta com 7 tabelas, sendo elas: Vendas, Cliente, Vendedores, forma de pagamento, marca, modelo e veiculos.



9.2 Tabela cliente

```
SQL> create table cliente
2  (idcliente number(4),
3   cpf_cliente number(11),
4   email_cliente varchar(50),
5   ds_endereco varchar(100),
6   dt_nasc date,
7   nm_telefone number(11),
8   primary key (idcliente));
```

Tabela criada.

```
SQL> desc cliente
Nome                               Nulo?   Tipo
-----
IDCLIENTE                         NOT NULL NUMBER(4)
CPF_CLIENTE                       NUMBER(11)
EMAIL_CLIENTE                     VARCHAR2(50)
DS_ENDERECO                       VARCHAR2(100)
DT_NASC                           DATE
NM_TELEFONE                       NUMBER(11)
```

9.3 Tabela vendedores

```
SQL> create table vendedores
2   (idvendedores number(4),
3   nm_vendedor varchar(80),
4   cpf_vendedor number(11),
5   primary key (idvendedores));
```

Tabela criada.

```
SQL> desc vendedores
```

Nome	Nulo?	Tipo
IDVENDEDORES	NOT NULL	NUMBER(4)
NM_VENDEDOR		VARCHAR2(80)
CPF_VENDEDOR		NUMBER(11)

9.4 Tabela forma de pagamento

```
SQL> create table forma_pgto
2   (idforma_pgto number(4),
3   ds_pgto varchar (10),
4   primary key (idforma_pgto));
```

Tabela criada.

```
SQL> desc forma_pgto
```

Nome	Nulo?	Tipo
IDFORMA_PGTO	NOT NULL	NUMBER(4)
DS_PGTO		VARCHAR2(10)

9.5 Tabela marca

```
SQL> create table marca
2   (idmarca number(4),
3   ds_marca varchar (30),
4   primary key (idmarca));
```

Tabela criada.

```
SQL> desc marca
```

Nome	Nulo?	Tipo
IDMARCA	NOT NULL	NUMBER(4)
DS_MARCA		VARCHAR2(30)

9.6 Tabela modelo

```
SQL> create table modelo
2   (idmodelo number(4),
3   idmarca number(4),
4   ds_modelo varchar (10),
5   primary key (idmodelo),
6   foreign key (idmarca) references marca (idmarca));
```

Tabela criada.

```
SQL> desc modelo
Nome                               Nulo?   Tipo
-----
IDMODELO                          NOT NULL NUMBER(4)
IDMARCA                           NUMBER(4)
DS_MODELLO                         VARCHAR2(10)
```

9.7 Tabela veículos

```
SQL> create table veiculos
2   (idveiculo number(4),
3   idmodelo number(4),
4   ds_veiculo varchar(30),
5   combustivel varchar(10),
6   num_portas number (1),
7   km_rodados number(6),
8   transmissao varchar (30),
9   vl_compra float,
10  url_veiculo varchar(80),
11  primary key (idveiculo),
12  foreign key (idmodelo) references modelo (idmodelo));
```

Tabela criada.

```
SQL> desc veiculos
Nome                               Nulo?   Tipo
-----
IDVEICULO                          NOT NULL NUMBER(4)
IDMODELO                           NUMBER(4)
DS_VEICULO                         VARCHAR2(30)
COMBUSTIVEL                        VARCHAR2(10)
NUM_PORTAS                         NUMBER(1)
KM_RODADOS                         NUMBER(6)
TRANSMISSAO                       VARCHAR2(30)
VL_COMPRA                          FLOAT(126)
URL_VEICULO                        VARCHAR2(80)
```

9.8 Tabela vendas


```

SQL> create table vendas
  2      (idvendas number(4),
  3      idcliente number(4),
  4      idforma_pgto number(4),
  5      idveiculo number(4),
  6      idvendedores number(4),
  7      vl_venda float,
  8      dt_venda date,
  9      primary key (idvendas),
 10      foreign key (idcliente) references cliente (idcliente),
 11      foreign key (idveiculo) references veiculos (idveiculo),
 12      foreign key (idforma_pgto) references forma_pgto (idforma_pgto),
 13      foreign key (idvendedores) references vendedores (idvendedores));

```

Tabela criada.

```
SQL> desc vendas
```

Nome	Nulo?	Tipo
IDVENDAS	NOT NULL	NUMBER(4)
IDCLIENTE		NUMBER(4)
IDFORMA_PGTO		NUMBER(4)
IDVEICULO		NUMBER(4)
IDVENDEDORES		NUMBER(4)
VL_VENDA		FLOAT(126)
DT_VENDA		DATE

9.8 Comandos MYSQL/PHP

Dashboard

```

SELECT Funcionario as Vendedor, DATE_FORMAT(dt_venda, '%d/%m/%Y') as 'Data da venda', vl_venda as 'Valor da Venda' FROM vendas WHERE dt_venda BETWEEN '2022-11-01' AND '2022-11-31';

("SELECT count(*) as total_vendas from `vendas` WHERE `dt_venda` BETWEEN '2022-11-01' AND '2022-11-30';");

("SELECT count(vl_venda) as total_vendas_ano from `vendas` WHERE `dt_venda` BETWEEN '2022-01-01' AND '2022-12-31';");

SELECT SUM(vendas.vl_venda-veiculos.vl_compra) as Lucro_mensal from veiculos, vendas WHERE vendas.dt_venda BETWEEN '2022-11-01' AND '2022-11-30' AND vendas.id_venda = veiculos.id_veiculo;

SELECT SUM(vendas.vl_venda-veiculos.vl_compra) as Lucro_anual from veiculos, vendas WHERE vendas.dt_venda BETWEEN '2022-01-01' AND '2022-12-31' AND vendas.id_venda = veiculos.id_veiculo;

("SELECT count(*) as vendas FROM `vendas` WHERE Funcionario = 'Roberto';");

("SELECT count(*) as vendas FROM `vendas` WHERE Funcionario = 'Fernanda';");

("SELECT count(*) as vendas FROM `vendas` WHERE Funcionario = 'Rafael';");

("SELECT count(*) as vendas FROM `vendas` WHERE Funcionario = 'Letícia';");

("SELECT count(*) as vendas FROM `vendas` WHERE Funcionario = 'Daniel';");

("SELECT count(*) as vendas FROM `vendas` WHERE Funcionario = 'Renato';");

("SELECT count(*) as vendas FROM `vendas` WHERE Funcionario = 'Aline';");

("SELECT count(*) as vendas FROM `vendas` WHERE Funcionario = 'Fernando';");

```

Clientes

```

SELECT * FROM `clientes`

("SELECT * FROM `clientes` WHERE `id_cliente` = ?;")

```

Login

```

("SELECT * FROM `cadastro` WHERE `user_cad` = ? AND `senha_cad` = ?;")

```

Veiculos

```

SELECT * FROM veiculos;

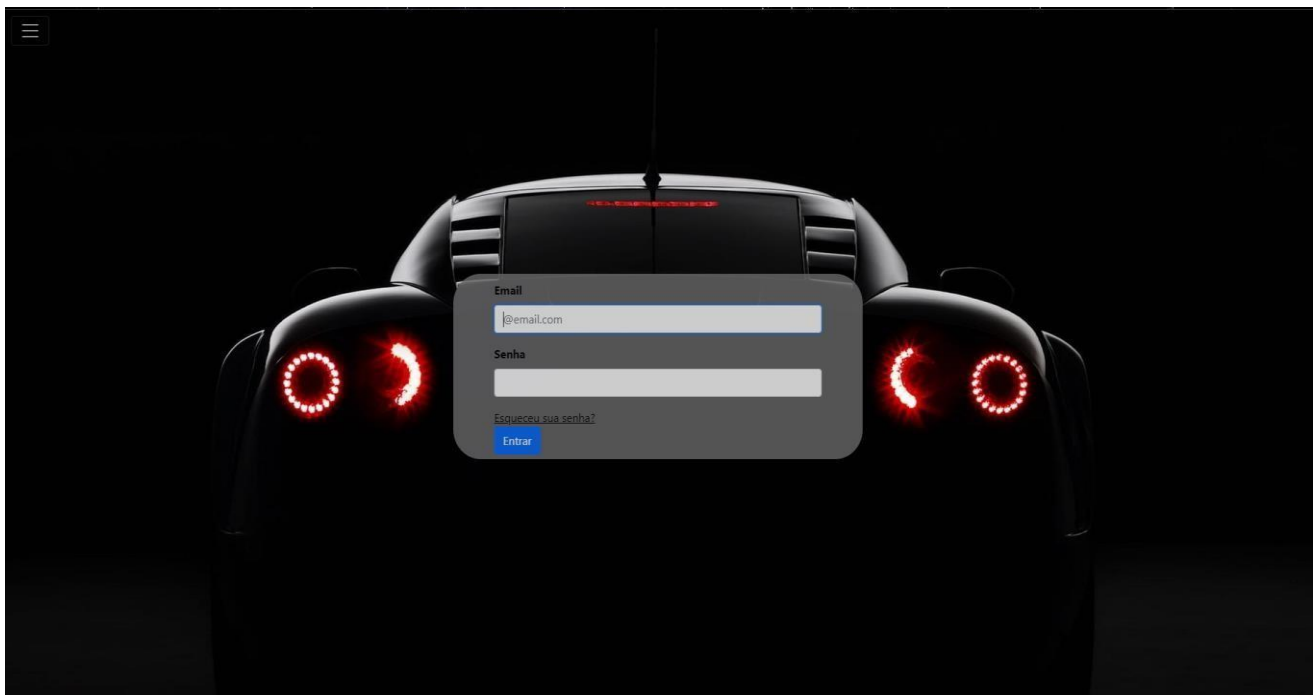
```

10 . DESENVOLVIMENTO

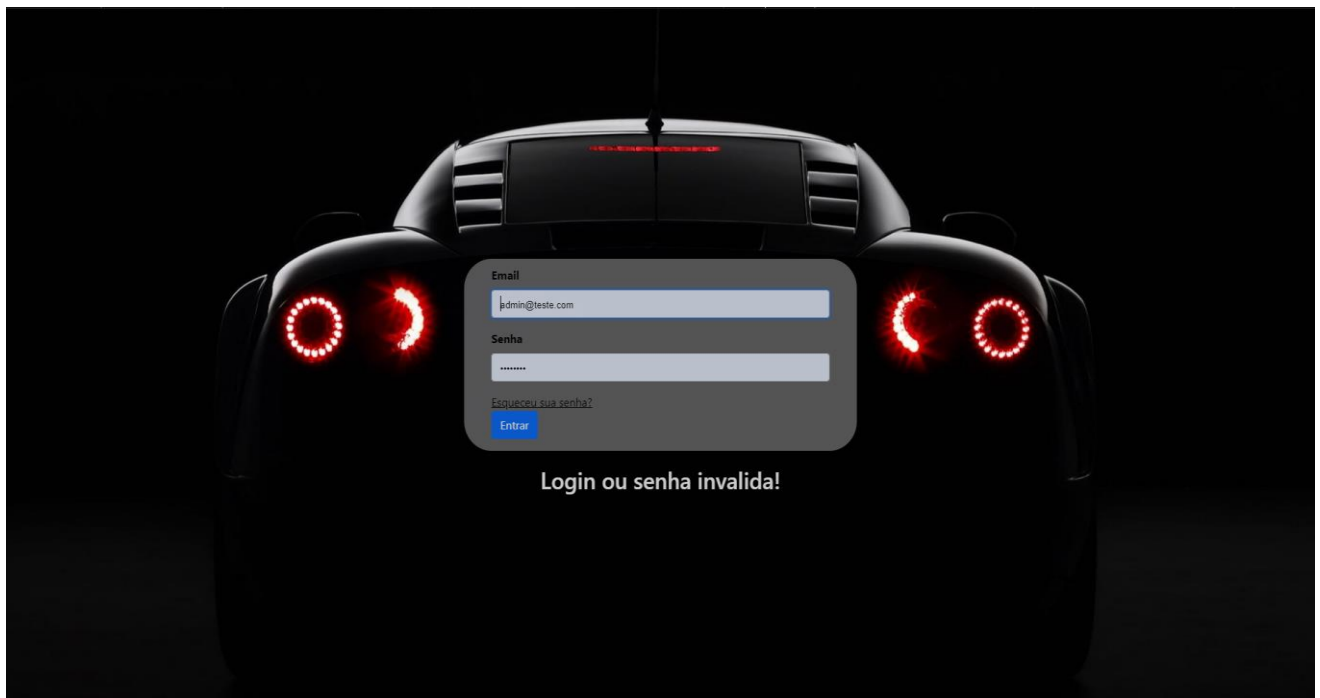
A aplicação foi dividida em módulos disponíveis no menu: Dashboard, Clientes, Veículos e Cadastro.

10.1 Área de Login:

Para acessar o sistema, se o usuário ainda não estiver logado, será exibido a tela de login. Caso o usuário já tenha feito o login, a sessão é salva e o usuário seria redirecionado para a tela principal da aplicação.

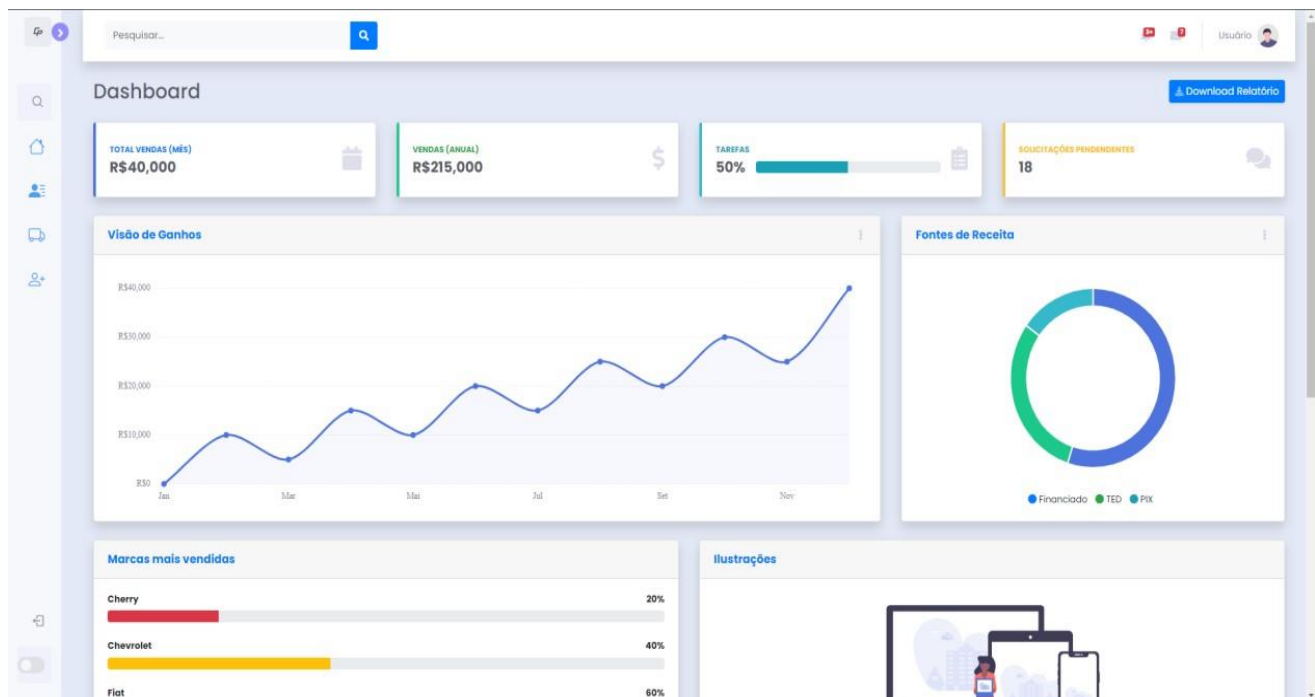


O sistema utiliza a linguagem PHP para realizar a validação de email e senha do usuário. Se o email e a senha digitados pelo usuário estiverem contidas no banco de dados, o sistema irá validar a sessão e redirecionar o mesmo para a aplicação. Em caso de as informações fornecidas não forem compatíveis com as presentes no banco de dados o sistema irá bloquear o acesso do usuário a aplicação e irá apresentar a seguinte mensagem: "Email ou senha inválidos!".



10.2 Dashboard

No menu Dashboard o sistema apresenta informações gerais sobre vendas, funcionários e metas de vendas.

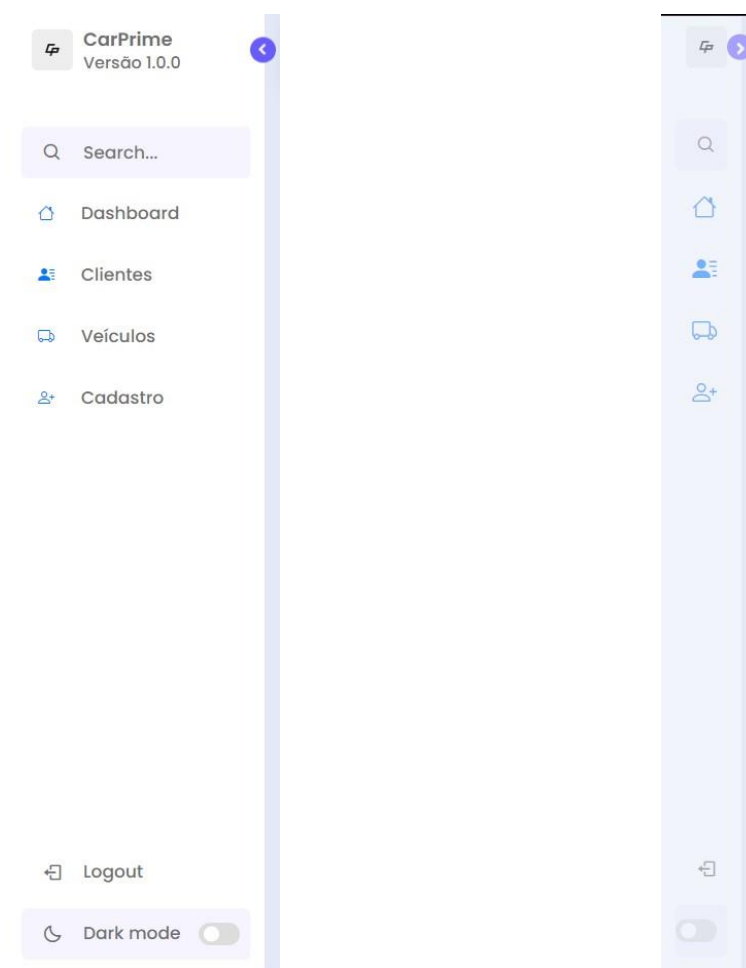


O Sistema disponibiliza para o usuário o relatório mensal de vendas, detalhando o vendedor que realizou a venda, o valor e a data em que a mesma foi realizada.

Vendedor	Data da venda	Valor da Venda
Roberto	01/11/2022	35000.00
Rafael	17/11/2022	80500.00
Renato	17/11/2022	93500.00
Roberto	01/11/2022	35000.00
Rafael	17/11/2022	80500.00
Renato	17/11/2022	93500.00

10.3 Barra de navegação

O sistema possui uma barra de navegação recolhível que ao acessar uma nova função minimiza automaticamente para um maior campo de trabalho.



10.4 Clientes

Na aba clientes o sistema disponibiliza uma lista com informações como Nome, CPF e email dos clientes da rede, sendo possível a adição alteração e exclusão de informações.

Nome: CPF: email:

Endereço: dd/mm/aaaa Telefone:

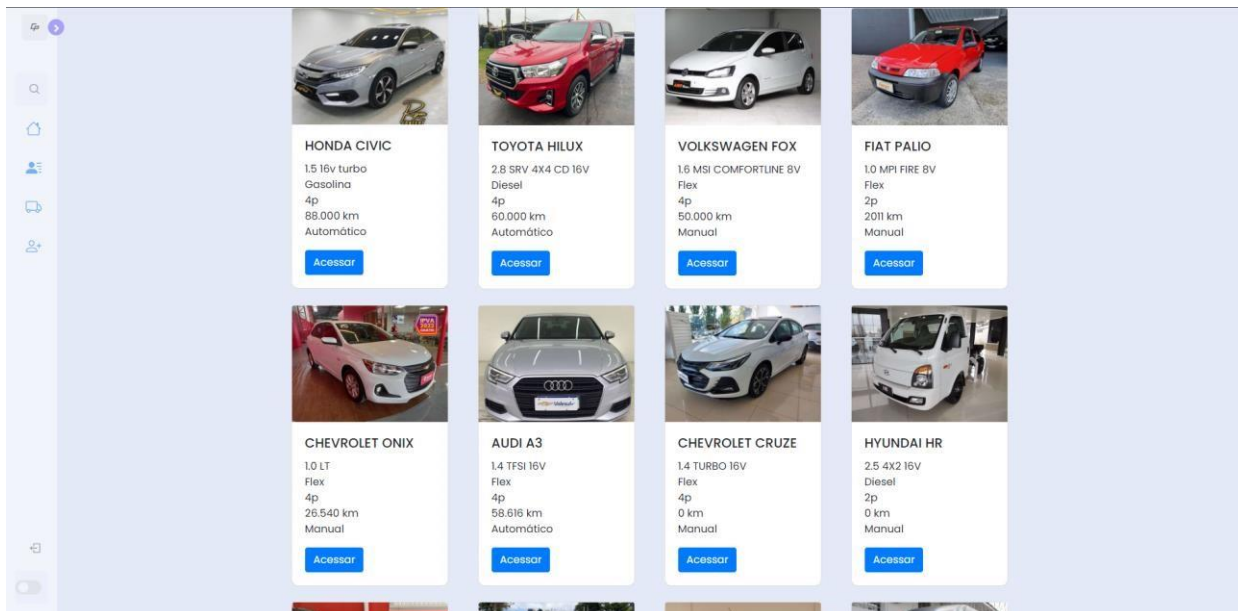
Nome	CPF	Email		
Eduardo	09741870035	eduardo@gmail.com	Excluir	Alterar
Daniel	02730016040	daniel@gmail.com	Excluir	Alterar
Vanessa	21310373094	vanessa@hotmail.comm	Excluir	Alterar
Marcelo	02730016040	marcelo@gmail.com	Excluir	Alterar

Para realizar a inclusão de um novo cliente, é necessário preencher as informações requisitadas pelo sistema. Após o preenchimento o usuário irá clicar no botão “Cadastrar”, fazendo assim a inclusão dos dados preenchidos na base de dados, sendo gerado uma mensagem de sucesso.

Nome: CPF: email:

Endereço: dd/mm/aaaa Telefone:

Nome	CPF	Email		
Eduardo	02730016040	eduardo@email.com	Excluir	Alterar
Daniel	02730016040	daniel@gmail.com	Excluir	Alterar
Vanessa	21310373094	vanessa@hotmail.comm	Excluir	Alterar
Marcelo	02730016040	marcelo@gmail.com	Excluir	Alterar



10.6 Financiamento

Na aba de Financiamento, o usuário pode simular o valor das parcelas mensais do veículo, de acordo com seu valor e a taxa mensal de juros. O sistema também mostra o valor total do veículo financiado.

VALOR TOTAL DO VEICULO

TAXA DE JUROS MES

QUANTIDADE DE PARCELAS

Calcular

A screenshot of a web-based calculator interface for vehicle financing. The interface has a light blue background and a vertical sidebar on the left with icons for search, home, users, chat, and profile. The main content area contains a central form with the following fields and labels:

- VALOR TOTAL DO VEICULO (input field)
- TAXA DE JUROS MES (input field)
- QUANTIDADE DE PARCELAS (input field)
- Calcular (button)

Below the form, the results are displayed:

- Valor da parcela R\$ 3.405,80
- Valor final do Veículo R\$ 163.44

11. REDES

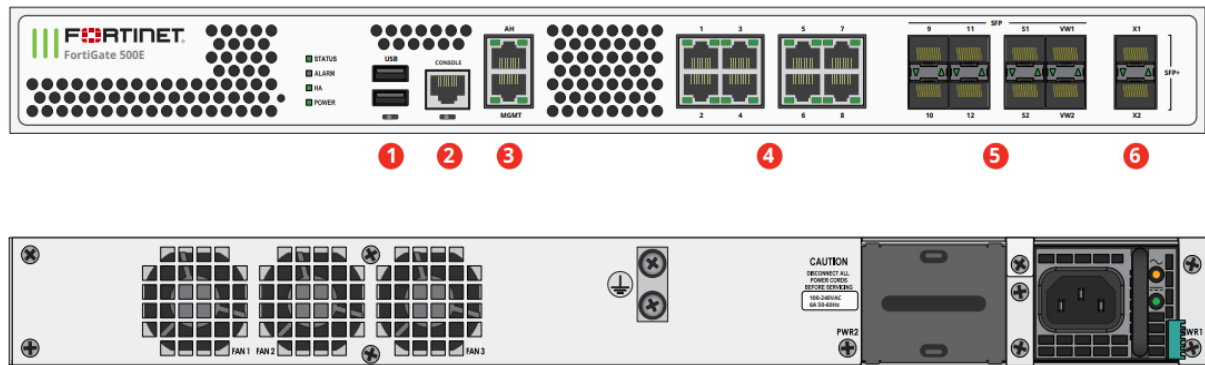
Para mais segurança adequamos um sistema de firewall NGFW da Fortinet, uma solução mais robusta e segura para os dados. Assim adicionado o Fortigate 500E, FortiAnalyzer e FortiMail.

11.1 FortiGate 500E

A série FortiGate 500E fornece uma solução SD-WAN centrada em aplicativos, escalável e segura com recursos de firewall de próxima geração (NGFW). Protege contra ameaças cibernéticas com aceleração de sistema em um chip e SD-WAN segura líder do setor em uma solução simples, acessível e fácil de implantar. A abordagem Security-Driven Networking da Fortinet fornece integração total da rede com a nova geração de segurança.

Com a adoção do equipamento, busca-se a melhoria da segurança dos dados internos de clientes e fornecedores, visto que o firewall irá fazer toda a filtragem de dados de entrada e saída da rede, além de inspeção dentro da camada de aplicação da rede.

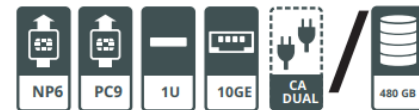
FortiGate 500E



Interfaces

1. Porta USB
2. Porta do console
3. 2 portas GE RJ45 MGMT/HA
4. 8 portas GE RJ45
5. 8xSlots GE SFP
6. 2x10 slots GE SFP+

Recursos de hardware



11.2

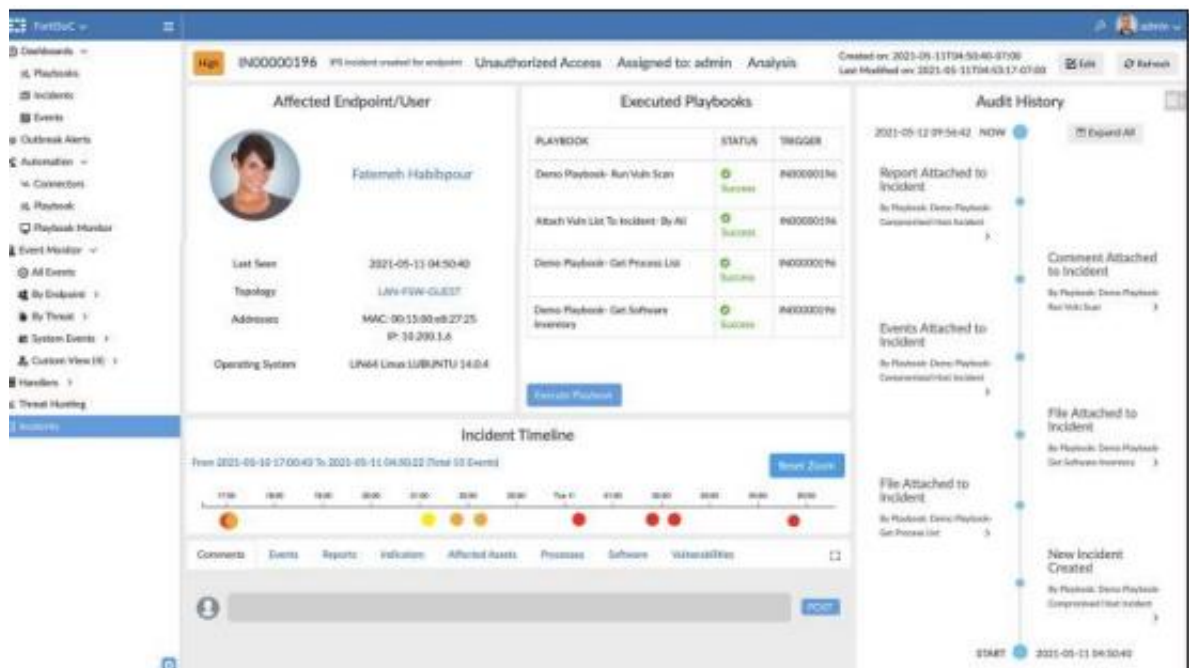
FortiAnalyzer

Arquitetura de segurança integrada com capacidades de gerenciamento de logs e segurança impulsionada por análises podem resolver essa falta de visibilidade.

Como parte do Fortinet Security Fabric, o FortiAnalyzer suporta casos de uso baseados em análises para fornecer uma melhor detecção contra violações.



FortiSoc integrado com o Fortianalyzer ajuda as equipes de segurança e operações de rede a proteger os ativos de rede com dados de registro e ameaças correlacionados e insight através de visões acionáveis com capacidades de perfuração. Notificação em tempo real, relatório e dashboards pré-definidos ou personalizados proporcionam visibilidade.



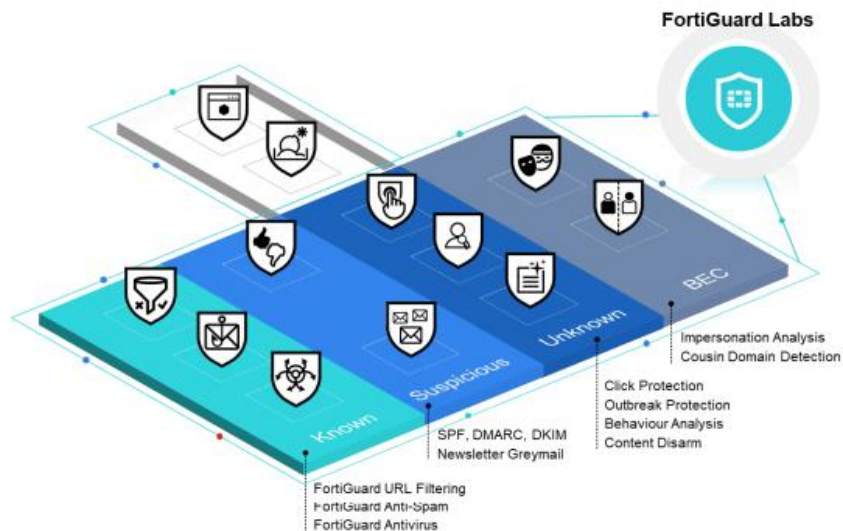
11.3 FortiMail

Com melhor desempenho da categoria validado por empresas de testes independentes, o FortiMail oferece proteção avançada em várias camadas contra todo espectro de ameaças transportadas por e-mail. Com o FortiGuard Labs e integrado com o Fortinet Security Fabric, o fortimail ajuda sua organização a prevenir, detectar e responder ameaças por e-mail, incluindo spam, phishing, malware, ameaças de dia zero.

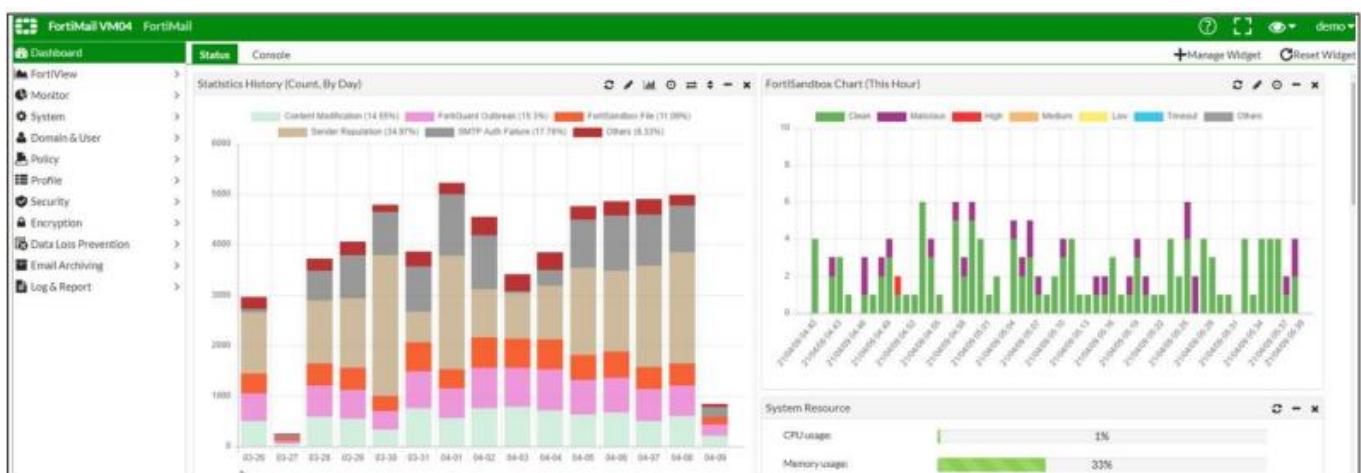
O FortiMail aborda todo o espectro de riscos que o e-mail representa para as organizações, fortalecido pela visibilidade global do FortiGuard Labs sobre as

últimas

ameaças.



Gestão intuitiva de e-mail, painéis de controle em tempo real, relatórios de riscos, quarentenas centralizadas e controle ao usuário final.



12. FUNDAMENTOS DE TI

A proposta de solução se traduz no desenvolvimento de um sistema web utilizando arquitetura de microserviços, visto que, no mercado digital os sites são grandes aliados dos ramos que tem envolvimento direto com clientes. Dito isso, o desenvolvimento preza por dar possibilidade de gerenciamento, controle de estoque e cadastros de clientes, otimizando o tempo de busca de informações e possibilitando

melhores análises do negócio automotivo, pois tem como nicho principal concessionárias de veículos.

Considerando que no modelo convencional de hospedagem, o software precisava ser instalado no computador da empresa contratante, a equipe buscou a implementação do software em modelo SaaS, do inglês Software as a Service, que é a oferta de software como serviço e pode ser acessado através de um navegador na internet, pois os dados ficam salvos em nuvem, tornando-se vantajoso tanto para a empresa quanto para o contratante

Como a arquitetura de microsserviços divide o software em funcionalidades independentes, a atividade de desenvolvimento da aplicação passa a ter um nível menor de complexidade, pois ela busca simplificar e facilitar a implementação dos serviços, atendendo as crescentes necessidades de criar soluções cada vez mais ágeis. A principal vantagem desse formato de arquitetura, está no fato de construir funções separadamente tornando-as capazes de promover serviços diversificados e sem relação de dependência, agregando maior agilidade e eficiência para a equipe de desenvolvimento.

O banco de dados necessário para atender a proposta é o não relacional, visto que, uma parcela de dados não pode ser inserida e acessada por tabelas, como exemplos as imagens dos veículos. Com um banco de dados não relacional, os dados poderão ser manipulados diretamente, permitindo a colaboração em tempo real e a edição simultânea que não seriam possíveis de outra forma.

12.1 Custos da Solução em Nuvem

Visando melhor custo-benefício de hospedagem em nuvem, a solução escolhida foi a AWS Amplify, que é disponibilizada em classificação gratuita por 12 meses*. A solução trata-se de um conjunto de ferramentas e recursos, voltados para criação de aplicações de forma fácil e rápida, principalmente desenvolvidas para público de desenvolvedores frontend para plataformas móveis.

O console do AWS Amplify cobra por dois recursos: compilação/implantação e hospedagem. Para o recurso de compilação e implantação, o preço por minuto de compilação é de 0,01 USD. Para o recurso de hospedagem, o preço por GB oferecido é

de 0,15 USD e o preço por GB hospedado é de 0,023 USD. Com o nível de uso gratuito da AWS, você pode começar a usar gratuitamente. Após se cadastrarem, os novos clientes da AWS recebem 1.000 minutos de compilação por mês para o recurso de compilação e implantação, e 15 GB fornecidos por mês e 5 GB de armazenamento de dados por mês para o recurso de hospedagem.

*Respeitando os níveis da AWS.

12.2 Requisitos mínimos para implementação do sistema Web (CarPrime):

- Sistema operacional: Windows >= 8
- Memória RAM: 4GB
- Navegador: versão mais recente do navegador Chrome, Firefox, Safari ou Edge
- Conexão de internet: velocidade de 5Mbit ou 800kbps

12.3 Características do equipamento utilizado para desenvolvimento

Desenvolvimento web é um dos mais leves, pois você vai precisar basicamente de um editor de código e um navegador, no desenvolvimento do software foi utilizado o equipamento com as seguintes características:

- Notebook-Lenovo
- Processador - Intel Core i5 - Modelo i5-7200U
- Memória RAM - 8GB
- Armazenamento - HD 500GB
- Sistema Operacional - Windows 10 Pro
- Telas - 14"
- Placa de Vídeo - Intel HD Graphics
- Placa de Rede - Gigabit Ethernet

**Imagem meramente ilustrativa



Custo \cong R\$ 3.317,11

13. REFERÊNCIAS

<https://aws.amazon.com/pt/amplify/pricing/>

<https://www.premiumstore.com.br/produto/notebook-lenovo-b320-14-ikbn-81cc0004br-i5-7200u-4gb-500gb-14-full-hd-windows-10-pro/>

<https://getbootstrap.com/>

<https://github.com/>

<https://www.jotform.com/>

