

SENAI – SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL
UNIDADE QUATRO BARRAS
CURSO TÉCNICO EM INFORMÁTICA

GEPE – GESTÃO DE PÁTIO DE ESTACIONAMENTO

QUATRO BARRAS
2013

Alan Rodrigues Moreira
Felipe Sadiovski Parizoto
Fernando da Silva Heleno
Lucas Happel

TCC SENAI 2013

TCC - Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Informática do SENAI – Serviço Nacional de Aprendizagem Industrial, Unidade Quatro Barras como requisito para a composição da segunda nota da referida disciplina, sob orientação do (a) professor (a) Tiago Navarro.

QUATRO BARRAS
2013

RESUMO

O Trabalho de conclusão de curso tem por objetivo apresentar um sistema de gerenciamento de estacionamento, que visa a diminuir o trabalho manual e melhorar o desempenho de controle na gestão de pátios de estacionamentos.

Palavras chave: Sistema. Estacionamento. Gestão. Controle. Pátio. GEPE.

SUMÁRIO

1	INTRODUÇÃO.....	5
2	sistema	5
2.1	estrutura do sistema.....	5
2.1.1	Cadastros	6
	Cadastros realizados pelo usuário para a execução das principais funcionalidades do sistema: 6	
2.2	OBJETIVOS DO SISTEMA.....	6
2.2.1	Melhorias	6
	O sistema visa a melhorar os seguintes pontos de um estacionamento:.....	6
3	ANALISE DO SISTEMA.....	6
3.1	Requisitos (Funcionais/Não Funcionais).....	6
3.2	Diagramas.....	8
3.2.2	Descrição Narrativa	9
3.4	Diagrama de Sequência.....	19
3.3	Pré-requisitos	21
3.3	Códigos	31
4	ELEMENTOS DE APOIO	32
5	CONCLUSÃO	32
	REFERÊNCIAS BIBLIOGRÁFICAS	33
	ANEXO	34

1 INTRODUÇÃO

A partir da análise de outros softwares do mesmo gênero, foi identificada a necessidade de um sistema de gestão de pátio de estacionamento de fácil utilização para empresas de pequeno porte que normalmente têm funcionários sem conhecimento avançado de informática. Então foi proposta a ideia de criação de um sistema intuitivo e simples, mas que atendesse às principais necessidades de um estacionamento.

O Sistema de Gestão de pátio de estacionamento (GEPE) tem por objetivo facilitar o controle dos veículos no pátio do estacionamento e ainda realizar cadastros básicos para reunir em um único local todas as informações que o estabelecimento pode vir a precisar; entre os cadastros disponíveis, pode-se destacar o cadastro de funcionários, usuários, clientes, registro de entrada e saída de veículos (controle de vagas) e fundamentalmente o cálculo do valor.

2 SISTEMA

No momento da instalação, o técnico responsável cadastra um usuário, o proprietário do estacionamento. Esta poderá cadastrar outros usuários que também terão acesso a todas funcionalidades do sistema. Uma vez que os clientes estejam cadastrados, o usuário poderá inserir os veículos do cliente, com todas as informações necessárias (modelo, placa, entre outras.) para que quando o cliente entre no pátio, seja fácil identificar e localizar o seu veículo, assim como a vaga que lhe é reservada. Além de clientes mensalistas, existem clientes avulsos, os quais não serão cadastrados no sistema, apenas as informações do veículo e de entrada e saída serão registradas para que possa ser gerado o valor avulso a ser computado no caixa diário.

2.1 ESTRUTURA DO SISTEMA

O GEPE funciona a partir de cadastros, nos quais são inseridas as informações para o controle do pátio. Existe uma hierarquia de inserção de dados para que o sistema funcione corretamente, ou seja, um cadastro depende do outro para que o sistema possa operar; por exemplo: a cobrança só pode ser efetuada se estiver cadastrado o valor do período de locação de vaga no pátio junto com o valor de cada categoria de veículo (carro, moto).

2.1.1 Cadastros

Cadastros realizados pelo usuário para a execução das principais funcionalidades do sistema:

- Banco (obrigatório)
- Funcionário (obrigatório)
- Usuários (obrigatório)
- Fornecedores (opcional)
- Categorias (obrigatório)
- Preços (obrigatório)
- Vagas (obrigatório)
- Forma de pagamento (obrigatório)
- Clientes (opcional)
- Veículos (obrigatório)

2.2 OBJETIVOS DO SISTEMA

Gerenciar o pátio do estacionamento por meio do controle dos acessos dos clientes mensalistas e avulsos, gerando cobrança por meio do cálculo (tempo x valor/hora(categoria)) apenas para este que só terá seu veículo liberado após o pagamento. Também se propõe a agilizar a localização do veículo no pátio, uma vez que será possível saber a vaga em que o mesmo se localiza por meio do cupom entregue ao cliente na entrada.

2.2.1 Melhorias

O sistema visa a melhorar os seguintes pontos de um estacionamento:

- Localização dos veículos
- Controle de recebimentos
- Entrada de clientes mensalistas (com os dados pré-cadastrados)

3 ANALISE DO SISTEMA

3.1 REQUISITOS (FUNCIONAIS/NÃO FUNCIONAIS)

Requisitos funcionais:

- Cadastro de funcionário: serão armazenados os dados pessoais do funcionário.

- Cadastro de usuários: serão armazenados os nomes dos usuários e senhas, para que seja possível efetuar um *login*.
- Cadastro de clientes: serão armazenados os dados pessoais do cliente.
- Cadastro de fornecedores: serão armazenados os dados dos fornecedores de produtos.
- Registro de quantidades de vagas disponíveis (executado apenas pelo técnico).
- Registro de veículos do cliente, para relacionar o veículo com o cliente.
- Cadastro de categorias de veículos, pois dependendo do tamanho do veículo, maior espaço será ocupado, portanto, maior será o valor cobrado.
- Cadastro de preços: o usuário irá definir a categoria e o tempo de locação de uma vaga.
- Acesso: registra a hora e data da entrada e da saída do veículo no pátio, além da categoria, para que uma nota seja emitida e o cliente seja cobrado, salvo mesalistas.
- Banco: Cadastra instituições bancárias que serão utilizadas no cadastro de contas dos funcionários.

Requisitos Não Funcionais:

- Entrega do veículo ao cliente.

3.2 DIAGRAMAS

3.2.1 UML

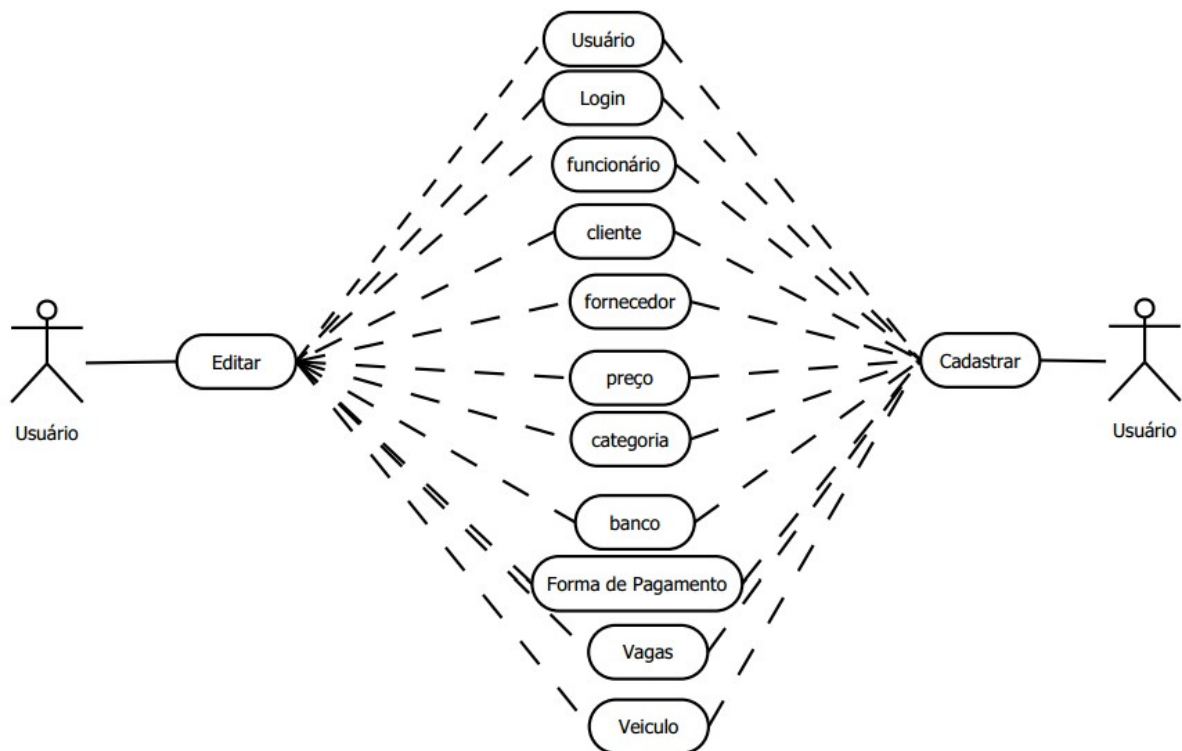


Figura 1.1 - Diagrama de caso de uso: Cadastro e Edição

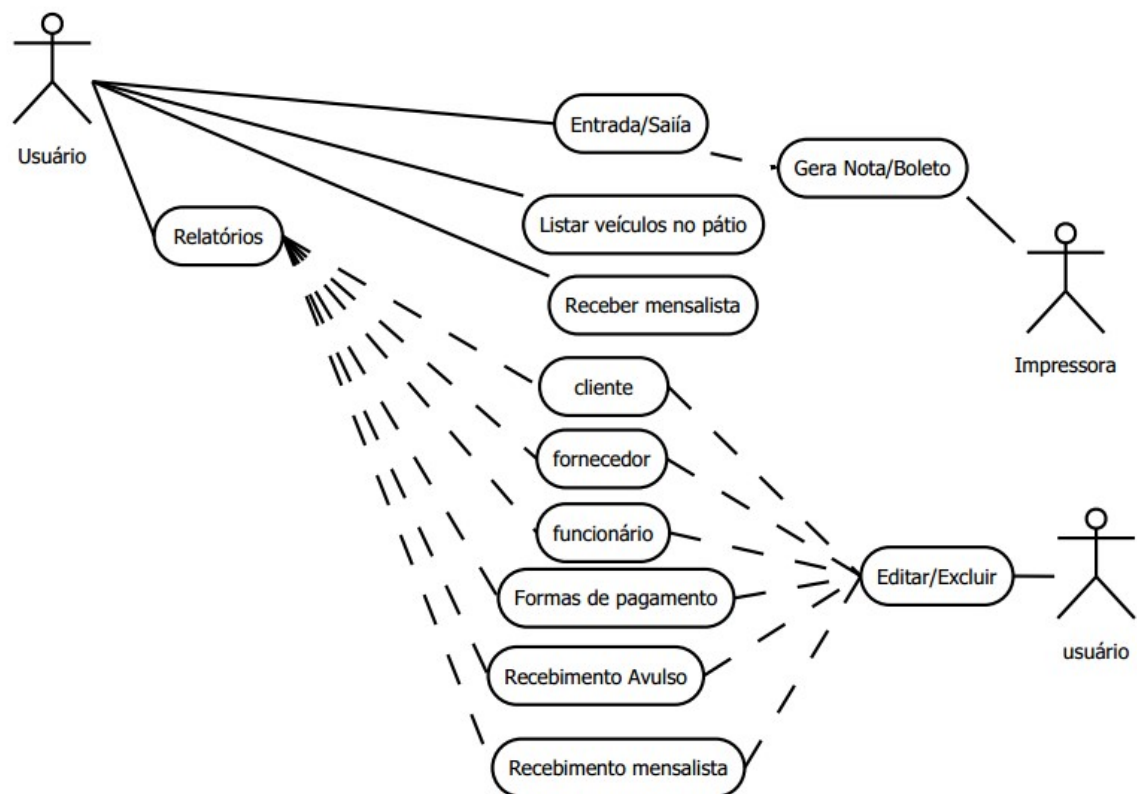


Figura 1.2 - Diagrama de caso de uso: Acesso, recebimento, pátio e relatórios

3.2.2 Descrição Narrativa

As telas referentes aos itens da descrição encontram-se em anexo.

3.2.2.1 *Login*

Nome: *Login*

Descrição: Usuário realiza *login* para ter acesso às funcionalidades do sistema.

Atores: Usuário

Pré-Condições: o usuário deve estar cadastrado para operar o sistema

Pós-Condições: o usuário tem acesso às funcionalidades do sistema

Fluxo básico:

- I.Usuário informa nome de usuário
- II.Usuário informa senha
- III.Usuário envia informações para o sistema
- IV.A página é direcionada para a tela principal

Fluxo alternativo:

- I. a) Caso o usuário informe o nome de usuário incorreto, uma mensagem aparece “Nome de usuário e/ou senha inválidos”.
a.1) Volta ao item I.
- II. b) Caso o usuário informe a senha incorreta, uma mensagem aparece “Nome de usuário e/ou senha inválidos”.
b.1) Volta ao item II.
- III. c) Caso o sistema não consiga estabelecer conexão com o BD, uma mensagem aparece (“Falha ao conectar com o Banco de dados”).

3.2.2.2 Menu

Nome: Menu

Descrição: a partir de botões, o usuário seleciona qual funcionalidade do sistema deseja acessar.

Pré-Condições: O usuário precisa estar logado.

Pós-Condições: após a escolha da funcionalidade, o usuário é redirecionado à outra página.

Fluxo básico:

- I. Usuário escolhe uma aba.
- II. Usuário escolhe uma opção disponibilizada pela aba.
- III. O usuário é redirecionado para a página referente ao item selecionado

Fluxo alternativo:

- I. Caso o usuário clique no botão sair, será feito o *logoff* e ele será redirecionado para a tela de *Login*.

3.2.2.3 Cadastro de Funcionário

Nome: Cadastro de Funcionário

Descrição: Usuário cadastra os funcionários do sistema

Atores: Usuário

Pré-Condições: O usuário deve estar cadastrado e logado para poder cadastrar funcionários

Pós-Condições: As informações do funcionário ficam registradas no sistema

Fluxo básico:

- I. Usuário informa o nome do funcionário
- II. Usuário informa o CPF do usuário
- III. Usuário seleciona um banco para ser vinculado ao funcionário
- IV. Usuário informa a agência bancária do funcionário
- V. Usuário seleciona o tipo da conta bancária do funcionário (conta corrente ou conta poupança)
- VI. Usuário informa o telefone do funcionário
- VII. Usuário informa o celular do funcionário

- VIII. Usuário informa o e-mail do funcionário
- IX. Usuário seleciona o sexo do funcionário
- X. Usuário informa o CEP do funcionário
- XI. Usuário informa o Logradouro do funcionário
- XII. Usuário informa o Bairro do funcionário
- XIII. Usuário informa a cidade do funcionário
- XIV. Usuário seleciona um estado do funcionário
- XV. Usuário envia os dados para o sistema.

Fluxo alternativo:

- I. a) Caso o usuário informe um CPF inválido, o sistema mostra a mensagem ("CPF inválido").
a.1) volta para o item II.
- II. b) Caso o usuário informe um Telefone inválido, o sistema mostra a mensagem ("Telefone inválido").
b.1) Volta para o item VI.
- III. c) Caso o usuário informe um celular inválido, o sistema mostra a mensagem ("Celular inválido").
c.1) Volta para o item VII
- IV. d) Caso o usuário informe um e-mail inválido, o sistema mostra a mensagem ("E-mail inválido").
d.1) Volta para o item VIII.
- V. e) Caso o usuário informar um CEP inválido, o sistema mostra a mensagem ("CEP inválido").
e.1)
- VI. Caso o sistema não consiga estabelecer conexão com o BD, uma mensagem aparece ("Falha ao conectar com o Banco de dados").
- VII. Caso algum campo fique em branco o sistema exibe a mensagem ("Preencha o campo nomedocampo").

3.2.2.4 Cadastro de Usuário

Nome: Cadastro de Usuário

Descrição: Um usuário cadastra outros usuários para que os mesmos operem o sistema.

Pré-Condições: O usuário deve estar *logado* no sistema, o funcionário deve estar cadastrado para se tornar um usuário do sistema.

Pós-Condições: Os usuários cadastrados poderão operar o sistema

- I. Usuário informa o nome do novo usuário
- II. Usuário informa a senha
- III. Usuário confirma a senha
- IV. Usuário confirma o cadastro

Fluxo alternativo:

- I. a.1) Caso as senhas informadas não sejam iguais, o sistema exibe a mensagem “Senhas inválidas/diferentes”.
a.2) volta ao item II.
- II. a.1) Caso o ID do funcionário não esteja cadastrado, o sistema exibe a mensagem “Funcionário não cadastrado”
a.2) Volta ao item IV.

3.2.2.5 Cadastro de Clientes

Nome: Cadastro de Cliente

Descrição: Usuário irá cadastrar os clientes mensalistas para que os mesmos tenham os benefícios concedidos a mensalistas e também possam utilizar os serviços do estacionamento.

Pré-Condições: O usuário deve estar cadastrado e *logado* no sistema

Pós-Condições: O cliente será cadastrado no sistema.

Fluxo Básico:

- I. Usuário insere o nome do cliente
- II. Usuário insere o CPF do cliente
- III. Usuário insere o telefone do cliente
- IV. Usuário insere o celular do cliente
- V. Usuário insere o e-mail do cliente
- VI. Usuário seleciona o sexo do cliente
- VII. Usuário insere o CEP do cliente
- VIII. Usuário insere o endereço do cliente

- IX. Usuário escolhe o número da residência do cliente
- X. Usuário insere o bairro do cliente
- XI. Usuário insere a cidade do cliente
- XII. Usuário seleciona o estado do cliente
- XIII. Usuário clica no botão “enviar” para cadastrar o cliente

Fluxo Alternativo:

- I. a.1) Caso o CPF for inválido, o sistema envia a mensagem “CPF inválido”.
a.2) Volta ao item III.
- II. a.1) Caso o usuário insira um telefone inválido, o sistema envia a mensagem “Telefone inválido”.
a.2) Volta ao item III.
- III. a.1) Caso o usuário insira um Celular inválido, o sistema envia a mensagem “Número do celular inválido”.
a.2) Volta ao item IV.
- IV. a.1) Caso o usuário insira um e-mail inválido, o sistema envia a mensagem “e-mail inválido”.
a.2) Volta ao item V.
- V. a.1) Caso o usuário insira um CEP inválido, o sistema envia a mensagem “CEP inválido”.
a.2) Volta ao item VII.

3.2.2.6 Forma de pagamento

Nome: Cadastro de forma de pagamento

Descrição: Usuário insere as formas de pagamento para serem utilizadas posteriormente.

Pré-Condições: O usuário deve estar *logado*.

Pós-Condições: O sistema reconhecerá as formas de pagamento quando for gerar um caixa.

Fluxo Básico:

- I. Usuário insere a forma de pagamento

Fluxo Alternativo:

N/E

3.2.2.7 Cadastro de Fornecedor

Nome: Cadastro de fornecedor

Descrição: O usuário cadastra o funcionário para que o cadastro do mesmo seja utilizado posteriormente no sistema.

Pré-Condições: O usuário deve estar *logado*.

Pós- Condições: O funcionário estará cadastrado no sistema.

Fluxo Básico:

- I. Usuário insere o nome da empresa
- II. Usuário insere o CNPJ da empresa
- III. Usuário insere o telefone da empresa
- IV. Usuário insere o celular da empresa
- V. Usuário insere o e-mail da empresa
- VI. Usuário insere o CEP da empresa
- VII. Usuário insere o endereço da empresa
- VIII. Usuário escolhe o número o local da empresa
- IX. Usuário insere o bairro da empresa
- X. Usuário insere a cidade da empresa
- XI. Usuário seleciona o estado da empresa
- XII. Usuário clica no botão “enviar” para cadastra a empresa

Fluxo Alternativo:

- I. a.1) Caso o CNPJ for inválido, o sistema envia a mensagem “CNPJ inválido”.
a.2) Volta ao item II.
- II. a.1) Caso o usuário insira um telefone inválido, o sistema envia a mensagem “Telefone inválido”.
a.2) Volta ao item III.
- III. a.1) Caso o usuário insira um Celular inválido, o sistema envia a mensagem “Número do celular inválido”.
a.2) Volta ao item IV.
- IV. a.1) Caso o usuário insira um e-mail inválido, o sistema envia a mensagem “e-mail inválido”.
a.2) volta ao item V.
- V. a.1) Caso o usuário insira um CEP inválido, o sistema envia a mensagem “CEP inválido”.
a.2) Volta ao item VI.

3.2.2.8 Cadastro de preço

Nome: Cadastro de preço

Descrição: Usuário cadastra valores a serem pagos por período de acordo com a categoria.

Pré-Condições: O usuário deve estar *logado* e alguma categoria deve estar cadastrada no sistema.

Pós-Condições: Os preços estarão cadastrados no sistema para que a cobrança possa ser efetuada posteriormente.

Fluxo Básico:

- I. Usuário seleciona uma categoria
- II. Usuário insere/seleciona um período (em horas)
- III. Usuário insere um valor.

Fluxo Alternativo:

N/E

3.2.2.9 Vagas

Nome: Cadastro de Vagas

Descrição: O técnico cadastra o total de vagas no pátio.

Pré-Condições: O usuário deve estar *logado*

Pós-Condições: O total de vagas no estacionamento será cadastrado no sistema, permitindo que vagas sejam locadas posteriormente.

Fluxo Básico:

- I. Usuário insere/seleciona o número de vagas

Fluxo Alternativo:

N/E

3.2.2.10 Veículos

Nome: Cadastro de veículos

Descrição: O usuário cadastra o veículo referente ao cliente, para que o mesmo possa ser localizado/registrado mais facilmente no pátio.

Pré-Condições: O usuário deve estar *logado* e um ou mais clientes deve estar cadastrado.

Pós-Condições: o veículo do cliente estará cadastrado, facilitando a localização e o registro de sua entrada no estacionamento.

Fluxo Básico:

- I. Usuário seleciona um cliente.
- II. Usuário insere a numeração da placa
- III. Usuário insere o modelo do veículo
- IV. Usuário informa/seleciona um fabricante

Fluxo Alternativo:

N/E

3.2.2.11 Registro

Nome: Registro de entrada/saída de um cliente

Descrição: Usuário informa a entrada e saída de um cliente para efetuar o registro.

Pré-Condições: o usuário deve estar *logado*, um cliente deve estar cadastrado, um veículo deve estar cadastrado, uma categoria deve estar cadastrada.

Pós-Condições: O registro de entrada e saída do cliente será efetuado.

Fluxo Básico:

- I. O usuário insere/seleciona o Cliente
- II. O usuário informa o veículo do cliente.
- III. O sistema informa a hora e data de entrada
- IV. O usuário seleciona a vaga
- V. O usuário seleciona a categoria
- VI. O sistema informa a hora e data da saída
- VII. O usuário fecha o registro

Fluxo Alternativo:

N/E

3.2.2.12 Banco

Nome: Cadastro de banco

Descrição: Usuário cadastra bancos

Pré-Condições: O usuário deve estar *logado*

Pós-Condições: Os bancos estarão cadastrados para serem utilizadas posteriormente em outros cadastros.

Fluxo básico:

- I. Usuário insere o nome do banco.

Fluxo Alternativo: N/E

3.2.2.13 Cadastro de categorias

Nome: Cadastro de categorias

Descrição: Usuário cadastra as categorias de veículos.

Pré-Condições: O usuário deve estar *logado*

Pós-Condições: as categorias de veículos estarão cadastradas, permitindo que sejam utilizadas posteriormente em outros formulários.

Fluxo Básico:

- I. Usuário insere a descrição da categoria.

Fluxo alternativo:

N/E

3.2.2.14 Relatório

Nome: Relatório

Descrição: Usuário gera um relatório a partir das opções disponíveis na aba.

Pré-Condições: Usuário deve estar *logado*.

Pós-Condições: Relatório é exibido na tela de acordo com o filtro realizado pelo usuário

Fluxo básico:

- I. Usuário escolhe uma opção para gerar relatório.

Fluxo alternativo:

Não possui.

3.3 Diagrama de estado

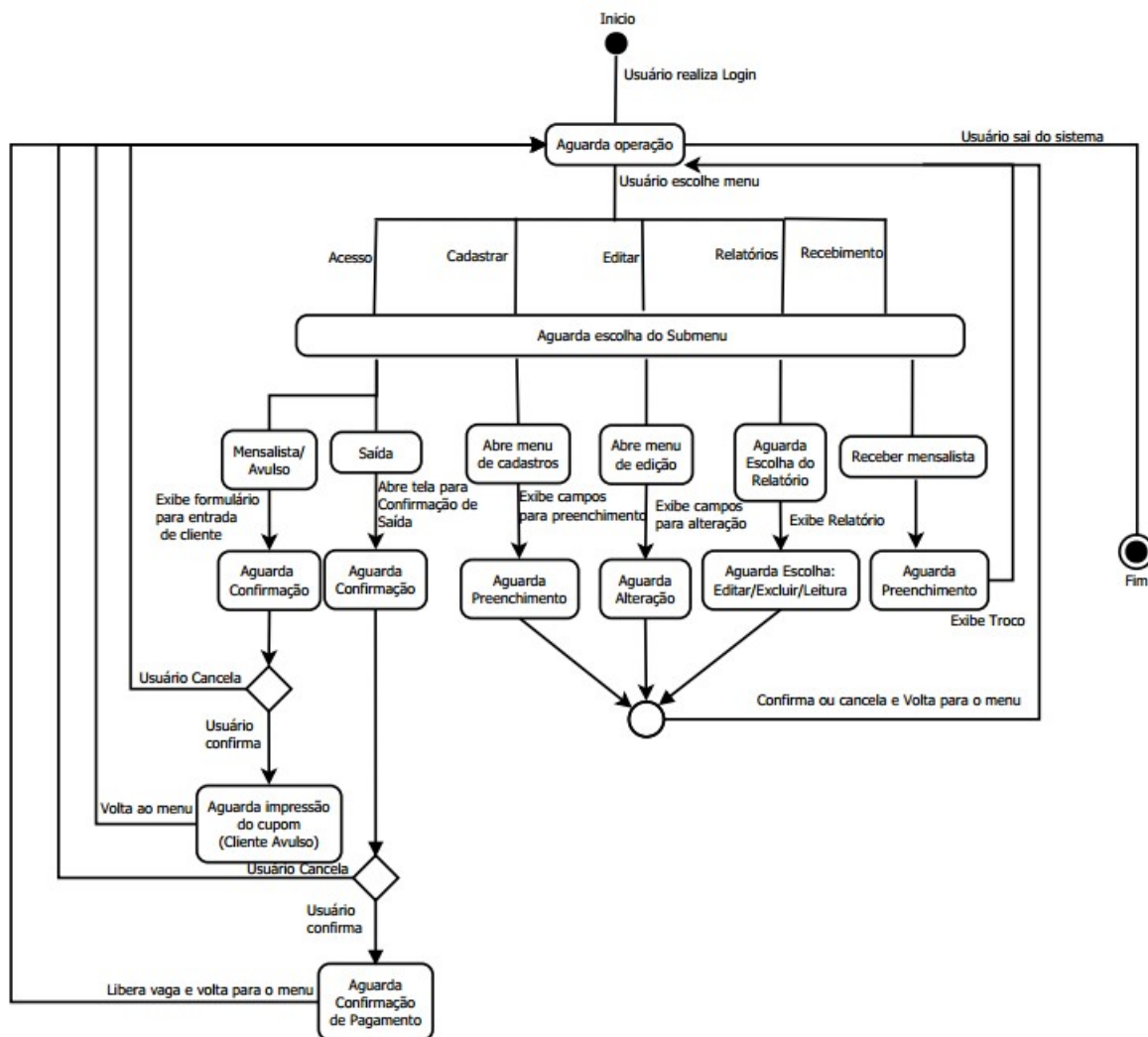


Figura 2 – Diagrama de estado

Observação: Por questão de estética e para tornar o diagrama mais fácil de ser entendido, apesar de que em todas as telas é possível realizar a saída do sistema, o diagrama indica que a saída é possível apenas por meio da tela inicial.

3.4 DIAGRAMA DE SEQUÊNCIA

3.4.1 Registro de entrada/saída

Diagrama referente a rotina de abertura de registro de entrada de cliente no estacionamento.

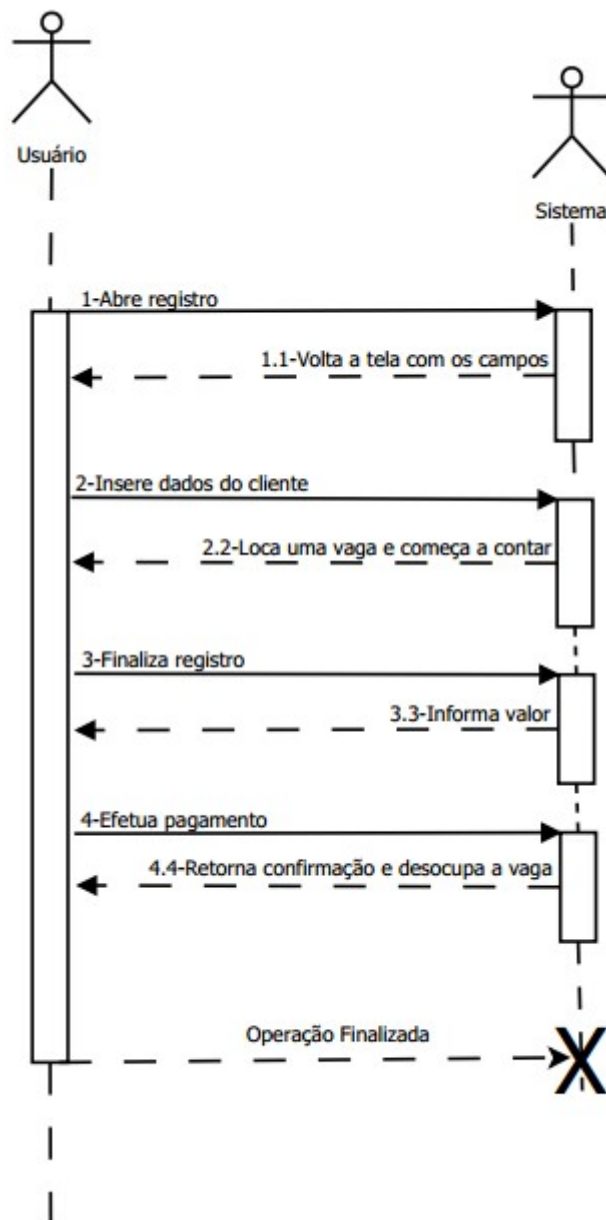


Figura 3.1 – Diagrama de sequência, Acesso avulso

3.4.2 Cadastro de Usuário

Diagrama referente a tela de cadastro de Usuário.

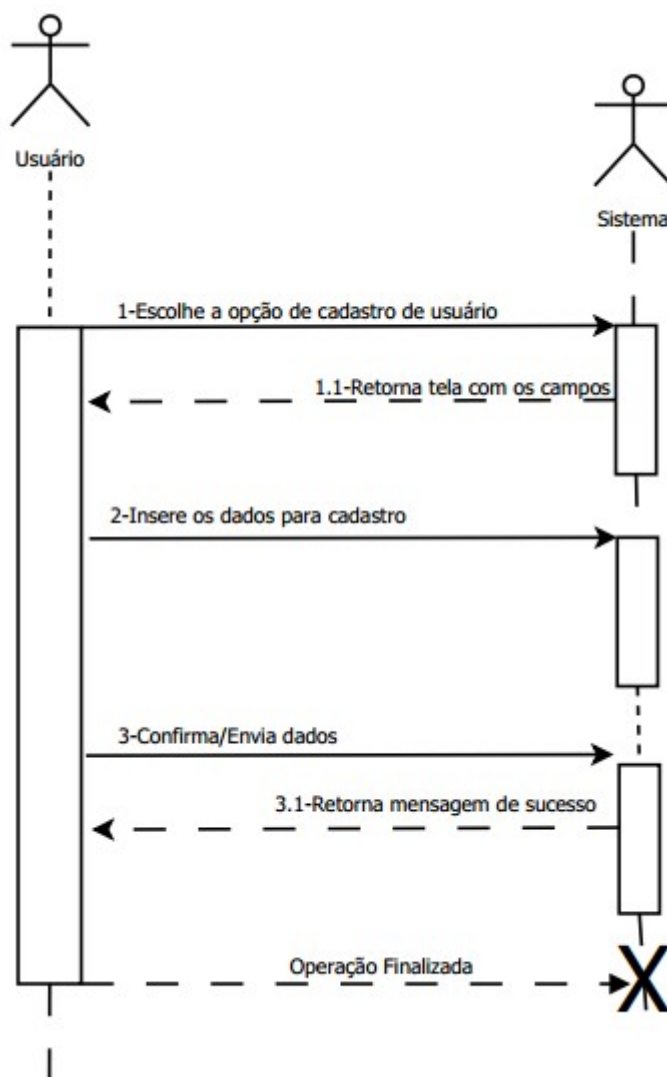


Figura 3.2 – Diagrama de sequência, Cadastro de Usuário

3.5 Diagrama de implantação

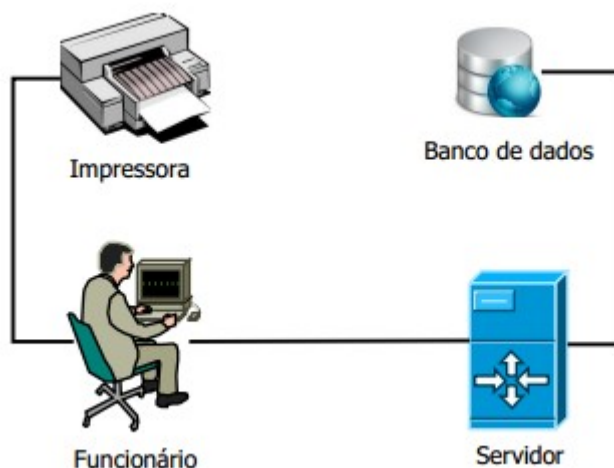


Figura 4 – Diagrama de Implantação

3.3PRÉ-REQUISITOS

Para total funcionamento do sistema, o *hardware* deve atender aos seguintes requisitos.

Hardware:

Processador: 2.5 gigahertz (GHz) ou superior.

Memória RAM: 4GB

Espaço em disco: 500 gigabyte (GB)

Software:

Browser: Google Chrome;

Banco de dados: Postgres 9.2;

Servidor: Apache mais recente disponível na data de implantação.

4 BANCO DE DADOS

4.1 Modelo Conceitual

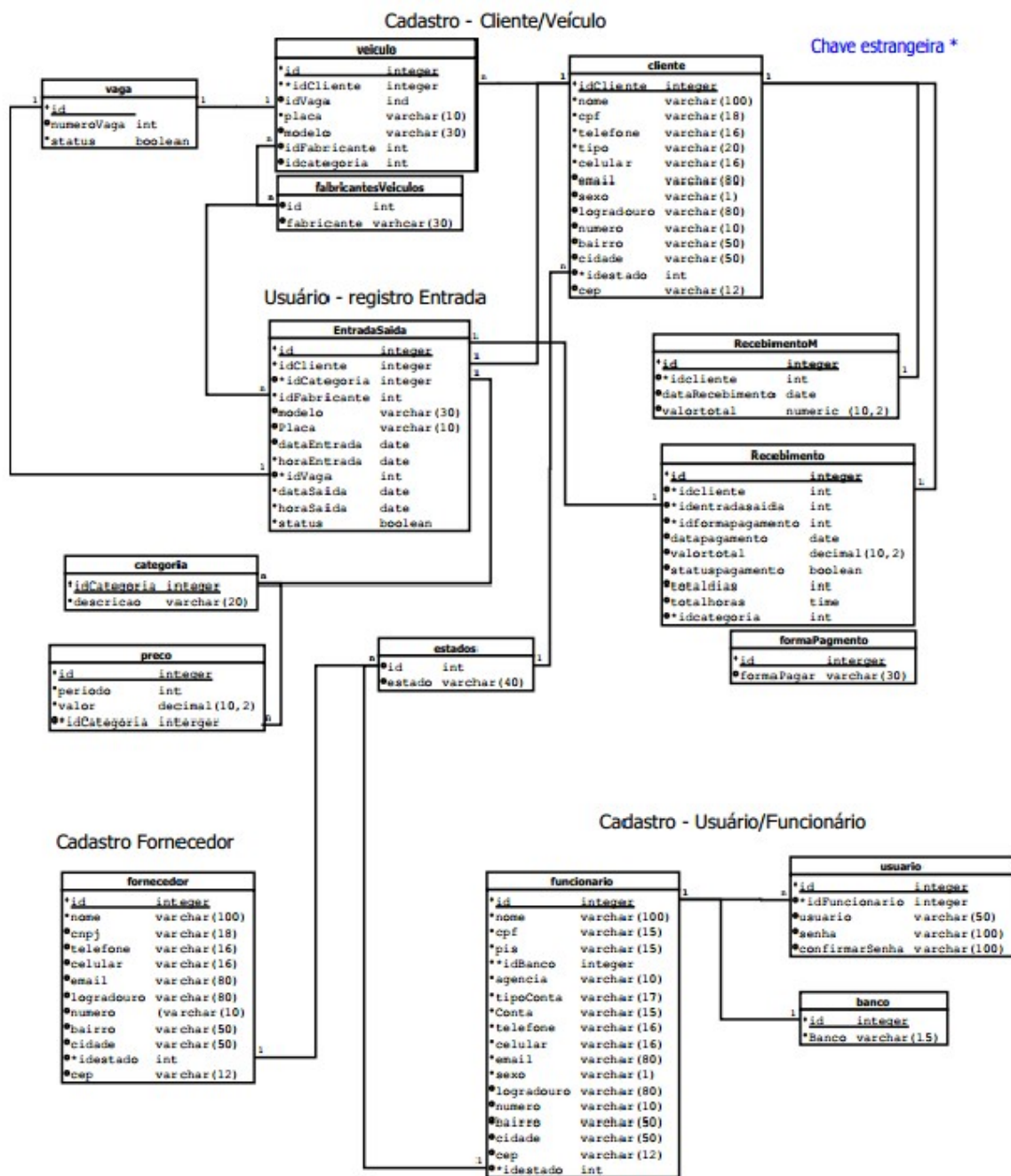


Figura 5 – Diagrama de entidade e relacionamento

4.2 Físico

```
-- Table: banco

-- DROP TABLE banco;

CREATE TABLE banco
(
    id serial NOT NULL,
    banco character varying(15),
    CONSTRAINT banco_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE banco
    OWNER TO postgres;

-- Table: categoria

-- DROP TABLE categoria;

CREATE TABLE categoria
(
    id serial NOT NULL,
    descricao character varying(20),
    CONSTRAINT categoria_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE categoria
    OWNER TO postgres;

-- Table: cliente

-- DROP TABLE cliente;

CREATE TABLE cliente
(
    id serial NOT NULL,
```

```

nome character varying(100),
cpf character varying(18),
telefone character varying(16),
celular character varying(16),
email character varying(80),
sexo character varying(1),
logradouro character varying(100),
numero character varying(10),
bairro character varying(50),
cidade character varying(50),
estado character varying(2),
cep character varying(12),
idpreco integer,
CONSTRAINT cliente_pkey PRIMARY KEY (id),
CONSTRAINT cliente_idpreco_fkey FOREIGN KEY (idpreco)
    REFERENCES preco (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE cliente
    OWNER TO postgres;

-- Table: entradasaida

-- DROP TABLE entradasaida;

CREATE TABLE entradasaida
(
    id serial NOT NULL,
    idcliente integer,
    idcategoria integer,
    modelo character varying(30),
    placa character varying(10),
    dataentrada date,
    horaentrada time without time zone,
    idvaga integer,
    datasaida date,
    horasaida time without time zone,
    status boolean,
    idfabricante integer,
    CONSTRAINT entradasaida_pkey PRIMARY KEY (id),

```



```

CONSTRAINT entradasaida_idcliente_fkey FOREIGN KEY (idcliente)
    REFERENCES cliente (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT entradasaida_idfabricante_fkey FOREIGN KEY (idfabricante)
    REFERENCES fabricantesveiculos (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE entradasaida
    OWNER TO postgres;

-- Table: estados

-- DROP TABLE estados;

CREATE TABLE estados
(
    id character varying(2) NOT NULL,
    estado character varying(40),
    CONSTRAINT estados_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE estados
    OWNER TO postgres;

-- Table: fabricantesveiculos

-- DROP TABLE fabricantesveiculos;

CREATE TABLE fabricantesveiculos
(
    id serial NOT NULL,
    fabricante character varying(50),
    CONSTRAINT fabricantesveiculos_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE fabricantesveiculos

```

```
OWNER TO postgres;
-- Table: formapagamento

-- DROP TABLE formapagamento;

CREATE TABLE formapagamento
(
    id serial NOT NULL,
    formapagar character varying(30),
    CONSTRAINT formapagamento_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE formapagamento
    OWNER TO postgres;

-- Table: fornecedor

-- DROP TABLE fornecedor;

CREATE TABLE fornecedor
(
    id serial NOT NULL,
    nome character varying(100),
    cnpj character varying(18),
    telefone character varying(16),
    celular character varying(16),
    email character varying(80),
    logradouro character varying(100),
    numero character varying(10),
    bairro character varying(50),
    cidade character varying(50),
    estado character varying(2),
    cep character varying(12),
    CONSTRAINT fornecedor_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE fornecedor
    OWNER TO postgres;
```

```

-- Table: funcionario

-- DROP TABLE funcionario;

CREATE TABLE funcionario
(
    id serial NOT NULL,
    nome character varying(100),
    cpf character varying(15),
    pis character varying(15),
    idbanco integer,
    agencia character varying(10),
    tipoconta character varying(17),
    conta character varying(15),
    telefone character varying(16),
    celular character varying(16),
    email character varying(80),
    sexo character varying(1),
    cep character varying(12),
    logradouro character varying(100),
    numero character varying(10),
    bairro character varying(50),
    cidade character varying(50),
    estado character varying(2),
    CONSTRAINT funcionario_pkey PRIMARY KEY (id),
    CONSTRAINT funcionario_idbanco_fkey FOREIGN KEY (idbanco)
        REFERENCES banco (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE funcionario
    OWNER TO postgres;

-- Table: preco

-- DROP TABLE preco;

CREATE TABLE preco
(
    id serial NOT NULL,
    periodo integer,

```

```

    valor numeric(10,2),
    idcategoria integer,
    CONSTRAINT preco_pkey PRIMARY KEY (id),
    CONSTRAINT preco_idcategoria_fkey FOREIGN KEY (idcategoria)
        REFERENCES categoria (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE preco
    OWNER TO postgres;

-- Table: recebimento

-- DROP TABLE recebimento;

CREATE TABLE recebimento
(
    id serial NOT NULL,
    idcliente integer NOT NULL,
    identradasaida integer NOT NULL,
    idformapagamento integer,
    datapagamento date,
    valortotal numeric(10,2),
    statuspagamento boolean,
    totaldias integer,
    totalhoras time without time zone,
    idcategoria integer,
    CONSTRAINT recebimento_pkey PRIMARY KEY (id),
    CONSTRAINT recebimento_idcliente_fkey FOREIGN KEY (idcliente)
        REFERENCES cliente (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT recebimento_identradasaida_fkey FOREIGN KEY (identradasaida)
        REFERENCES entradasaida (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT      recebimento_idformapagamento_fkey      FOREIGN      KEY
(idformapagamento)
        REFERENCES formapagamento (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE

```

```

);
ALTER TABLE recebimento
    OWNER TO postgres;

-- Table: recebimentom

-- DROP TABLE recebimentom;

CREATE TABLE recebimentom
(
    id serial NOT NULL,
    idcliente integer,
    datarecebimento date,
    valortotal numeric(10,2),
    CONSTRAINT recebimentom_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE recebimentom
    OWNER TO postgres;

-- Table: usuario

-- DROP TABLE usuario;

CREATE TABLE usuario
(
    id serial NOT NULL,
    idfuncionario integer,
    usuario character varying(50),
    senha character varying(100),
    confirmarsenha character varying(100),
    CONSTRAINT usuario_pkey PRIMARY KEY (id),
    CONSTRAINT usuario_idfuncionario_fkey FOREIGN KEY (idfuncionario)
        REFERENCES funcionario (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE usuario
    OWNER TO postgres;

```

```
-- Table: vaga
```

```
-- DROP TABLE vaga;
```

```
CREATE TABLE vaga
```

```
(
    id serial NOT NULL,
    numerovaga integer,
    status boolean,
    CONSTRAINT vaga_pkey PRIMARY KEY (id)
)
```

```
WITH (
```

```
    OIDS=FALSE
```

```
);
```

```
ALTER TABLE vaga
```

```
    OWNER TO postgres;
```

```
-- Table: veiculo
```

```
-- DROP TABLE veiculo;
```

```
CREATE TABLE veiculo
```

```
(
    id serial NOT NULL,
    idcliente integer,
    placa character varying(10),
    modelo character varying(30),
    idfabricante integer,
    idvaga integer,
    idcategoria integer,
    CONSTRAINT veiculo_pkey PRIMARY KEY (id),
    CONSTRAINT veiculo_idcategoria_fkey FOREIGN KEY (idcategoria)
        REFERENCES categoria (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT veiculo_idcliente_fkey FOREIGN KEY (idcliente)
        REFERENCES cliente (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT veiculo_idcliente_fkey1 FOREIGN KEY (idcliente)
        REFERENCES cliente (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT veiculo_idfabricante_fkey FOREIGN KEY (idfabricante)
        REFERENCES fabricantesveiculos (id) MATCH SIMPLE
)
```

```

        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT veiculo_idvaga_fkey FOREIGN KEY (idvaga)
        REFERENCES vaga (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
    )
WITH (
    OIDS=FALSE
);
ALTER TABLE veiculo
    OWNER TO postgres;

```

3.3 CÓDIGOS

O código a seguir foi destacado dos demais (em anexo), pois é de extrema importância para o funcionamento do sistema, é a partir dele que é calculado o valor que o cliente deve pagar com base na quantidade de horas e o valor da categoria do veículo dele.

```

public function pesquisaPreco($idCategoria)
{
    include_once '../conexao/PDO.php';

    try
    {
        $conn = conecta::abrir('estacionamento');
        $result = $conn->query("SELECT preco.valor,preco.idcategoria FROM preco
WHERE idcategoria=" . $idCategoria . "");

        while($row = $result->fetch(PDO::FETCH_ASSOC))
        {

            $preco = $row['valor'];

        }

        $conn=null;

    }
    catch (PDOException $e)
    {
        print "Erro: " . $e->getMessage() . "<br/>";
        die();
    }
    return $preco;
}

```

O código a seguir tem a funcionalidade de calcular um troco com base no total (tempo x valorHora(categoria) e retornar uma mensagem para o usuário.

```
public function calculaRecebimento()
{
    if ($this->vetor['valorrecebido'] < $this->vetor['valortotal'])
    {
        throw new Exception("Valor insuficiente");
    }

    else if ($this->vetor['valorrecebido'] > $this->vetor['valortotal'])
    {
        $troco = $this->vetor['valorrecebido'] - $this->vetor['valortotal'];
        echo "<script>alert('TROCO: " . money_format('%i', $troco) . ")</script>";
    }

    else if ($this->vetor['valorrecebido'] == $this->vetor['valortotal'])
    {
        echo "<script>alert('Operação realizada com sucesso!')</script>";
    }

    return;
}
```

4 ELEMENTOS DE APOIO

Dia

Eclipse

Google Chrome

Xampp

Apache

Postgres

Notepad++

5 CONCLUSÃO

Este trabalho cumpre de fato a meta a que se propõe. Foi apresentado o software de gestão de pátio de estacionamento, assim como todas suas funcionalidades e aspectos. Além disto, foi apresentada a análise do sistema, contendo os diagramas e documentações necessárias para o desenvolvimento do projeto, com base nas disciplinas ministradas no curso.

Foi uma oportunidade de reunir em um único projeto o conhecimento adquirido no decorrer do curso e também o momento de avaliar o que foi aprendido, as deficiências e destaques de cada um dos integrantes do grupo.

REFERÊNCIAS BIBLIOGRÁFICAS

DALL'OGGIO, Pablo. PHP: programando com orientação a objetos. 2ª ed. São Paulo: Editora Novatec. 2009.

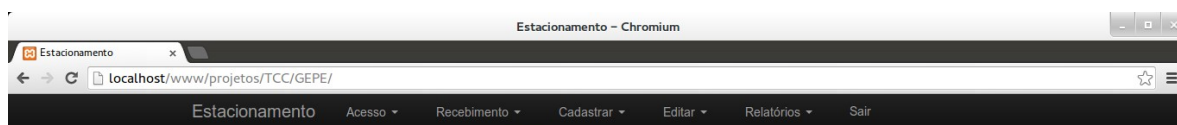
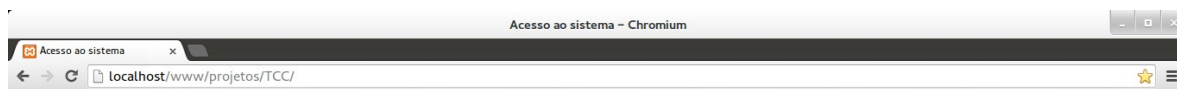
SILVA, Mauricio Samy. jQuery - A biblioteca do programador Java script. 2ª ed. São Paulo: Editora Novatec.

<<http://php.net/>>. Acesso de agosto a novembro de 2013.

<<http://www.w3schools.com/bootstrap>>. Acesso de agosto a novembro de 2013.

ANEXO

1 – Telas



Bem vindo ao GEPE alanmoreira

Contas a receber - Chromium

Contas a receber x

localhost/www/projetos/TCC/GEPE/cadastros/cadastroEntrada.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

00 AVULSO AUTOMÓVEL

AAV ARAB AMERICAN VEHÍ MODELO DO VEÍCULO PLACA DO VEÍCULO

26/11/2013 09:51 2

CONFIRMAR ENTRADA

Contas a receber - Chromium

Contas a receber x

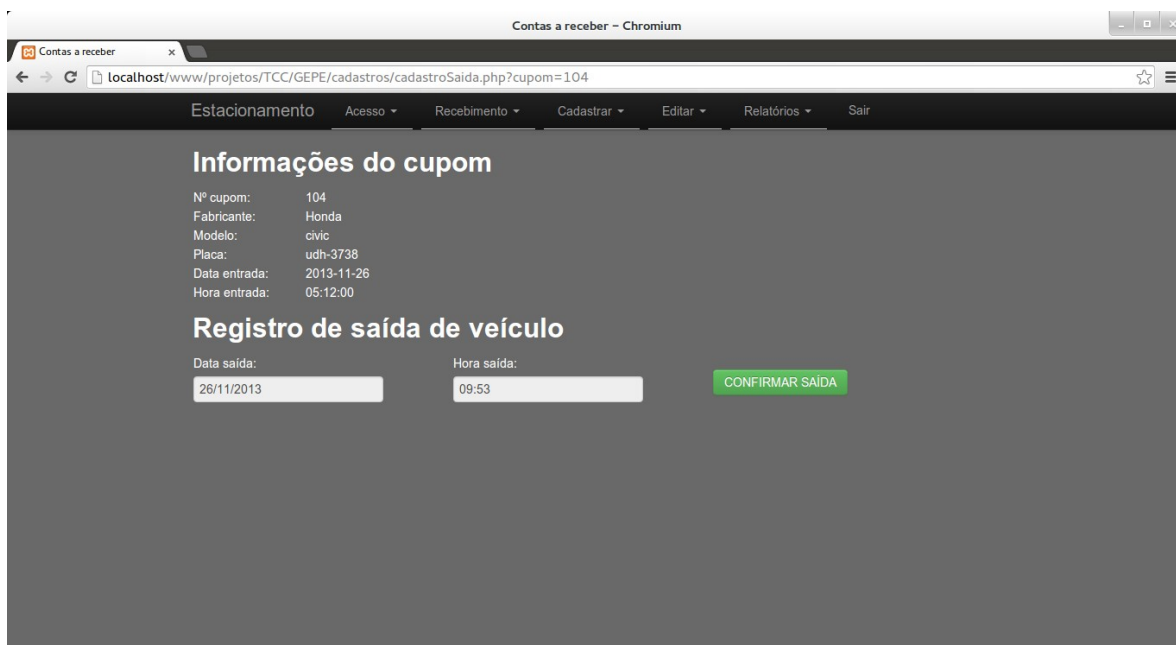
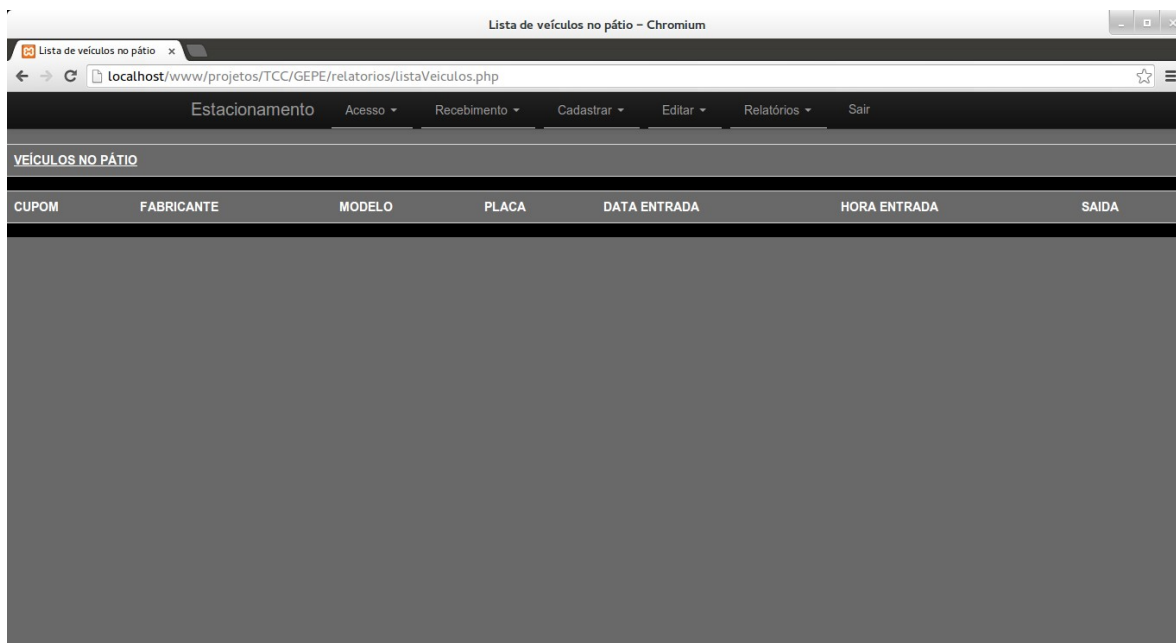
localhost/www/projetos/TCC/GEPE/cadastros/cadastroEntradaCliente.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

ESCOLHA UM CLIENTE

26/11/2013 09:51

CONFIRMAR ENTRADA



Banco - Chromium

localhost/www/projetos/TCC/GEPE/cadastros/cadastroRecebimento.php?cupom=104

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Fechamento de cupom

* Cupom:.....104
* Cliente:.....00 AVULSO
* Categoria:.....AUTOMÓVEL
* Valor/h:.....4.99
* Tempo utilizado:.....04:41:00
* Diárias:.....0

Valor total: 60.00

Forma de pagamento: Valor recebido:

DINHEIRO VALOR RECEBIDO FINALIZAR

Banco - Chromium

localhost/www/projetos/TCC/GEPE/cadastros/cadastroRecebimentoMensalista.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Cliente:
ESCOLHA UM CLIENTE

Data:
26/11/2013

Valor contratado:

Valor recebido:
VALOR RECEBIDO

FINALIZAR

Banco - Chromium

Banco

localhost/www/projetos/TCC/GEPE/cadastros/cadastroBanco.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Banco:

NOME DO BANCO

CADASTRAR

Categoria - Chromium

Categoria

localhost/www/projetos/TCC/GEPE/cadastros/cadastroCategoria.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Categoria:

DESCRIÇÃO DA CATEGORIA

CADASTRAR

Cadastro de clientes - Chromium

Cadastro de clientes

localhost/www/projetos/TCC/GEPE/cadastros/cadastroCliente.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Valor a contratar
350.00

Nome:
NOME

Telefone:
TELEFONE

E-mail:
E-MAIL

CEP:
CEP

Logradouro:
LOGRADOURO

Bairro:
BAIRRO

CPF:
CPF

Celular:
CELULAR

Sexo:
MASCULINO

Estado:
ACRE

Nº:
NÚMERO

Cidade:
CIDADE

CADASTRAR

Forma pagamento - Chromium

Forma pagamento

localhost/www/projetos/TCC/GEPE/cadastros/cadastroFormaPagar.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Forma de pagamento:
FORMA DE PAGAMENTO

CADASTRAR

Cadastro de fornecedores - Chromium

localhost/www/projetos/TCC/GEPE/cadastros/cadastroFornecedor.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Nome da empresa:	CNPJ:
NOME	CNPJ
Telefone:	Celular:
TELEFONE	CELULAR
E-mail:	
E-MAIL	
CEP:	Estado:
CEP	ACRE
Logradouro:	Nº:
LOGRADOURO	NÚMERO
Bairro:	Cidade:
BAIRRO	CIDADE
CADASTRAR	

Cadastro de funcionários - Chromium

localhost/www/projetos/TCC/GEPE/cadastros/cadastroFuncionario.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Nome:	PIS:
NOME	PIS
CPF:	Agência:
CPF	NÚMERO DA AGÊNCIA
Banco:	Conta:
BMG	NÚMERO DA CONTA
Tipo da conta:	Celular:
CONTA CORRENTE	CELULAR
Telefone:	Sexo:
TELEFONE	MASCULINO
E-mail:	Estado:
E-MAIL	ACRE
CEP:	Nº:
CEP	NÚMERO
Logradouro:	Cidade:
LOGRADOURO	CIDADE
Bairro:	
BAIRRO	

Preços - Chromium

Preços

localhost/www/projetos/TCC/GEPE/cadastros/cadastroPreco.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Categoria:
AUTOMÓVEL

Período (hora):
PERÍODO EM HORAS

Valor:
VALOR

CADASTRAR

Cadastro de usuário - Chromium

Cadastro de usuário

localhost/www/projetos/TCC/GEPE/cadastros/cadastroUsuario.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Funcionário:
ALAN RODRIGUES MOREIRA

Usuário:
USUÁRIO

Senha:
SENHA

Confirmar senha:
CONFIRMAR SENHA

CADASTRAR

Vagas Estacionamento - Chromium

Vagas Estacionamento x

localhost/www/projetos/TCC/GEPE/cadastros/cadastroVagas.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Total de vagas

TOTAL DE VAGAS

CADASTRAR

Cadastro de veículo - Chromium

Cadastro de veículo x

localhost/www/projetos/TCC/GEPE/cadastros/cadastroVeiculo.php

Estacionamento Acesso Recebimento Cadastrar Editar Relatórios Sair

Cliente:

ALAN RODRIGUES MOREIR

Placa:

PLACA

Modelo:

MODELO

Fabricante:

AAV ARAB AMERICAN VEHIC

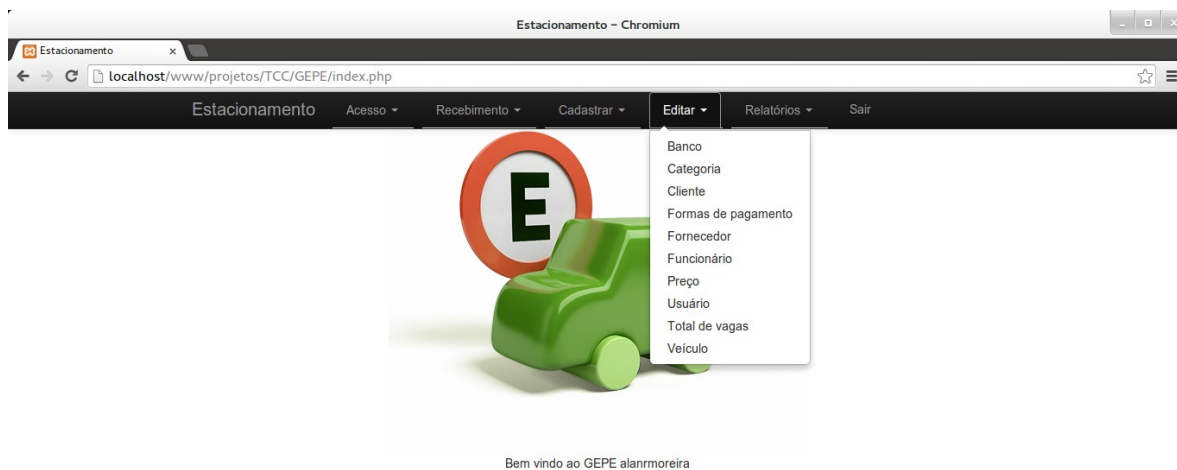
Categoria:

AUTOMÓVEL

Vaga:

2

CADASTRAR



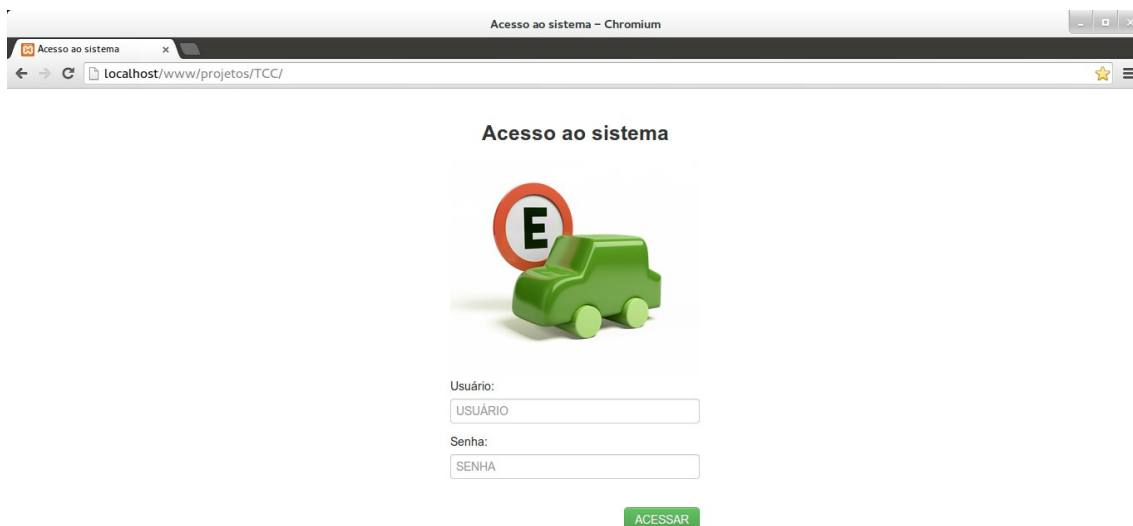
ID	NOME	TELEFONE	CELULAR	E-MAIL	TIPO	BANCO	AGÊNCIA	CONTA	EDITAR	EXCLUIR
4	ALAN RODRIGUES MOREIRA	(041) 3676-2623	(041) 9755-2069	ALANRMOREIRATI@YAHOO.COM.BR	CC	CEF	3511	00778778-0	EDITAR	EXCLUIR
7	LUCAS HAPPEL	(041) 3676-2205	(041) 9767-2984	HAPPEL.LUCAS@GMAIL.COM	CC	SANTANDER	105	171675	EDITAR	EXCLUIR

GEPE – Gestão de Pátio e Estacionamento

Este documento tem como objetivo descrever brevemente as telas do sistema, com o auxílio de imagens para facilitar a explicação de suas funcionalidades.

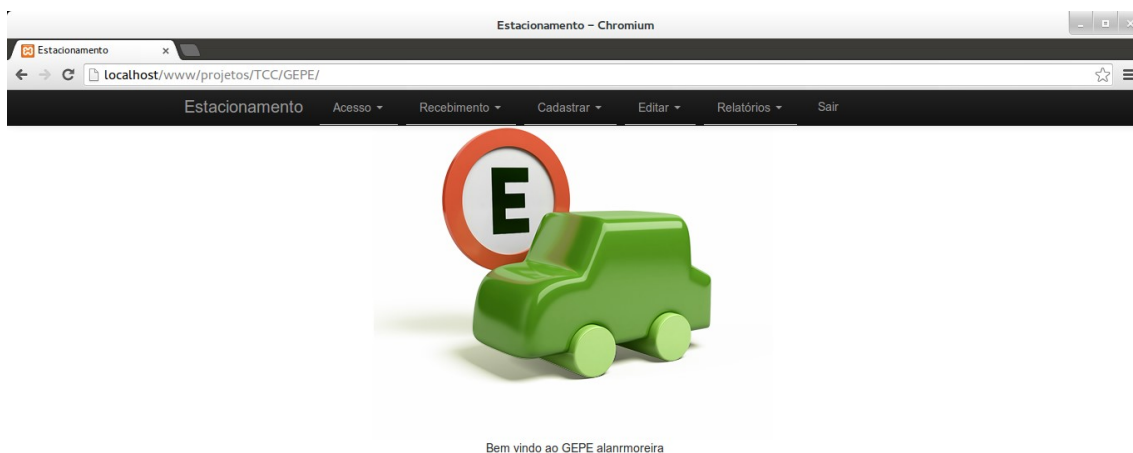
1 Login

Tela inicial do sistema, somente através dela é possível efetuar o *login* e acessar as funcionalidades do sistema.



2 Menu

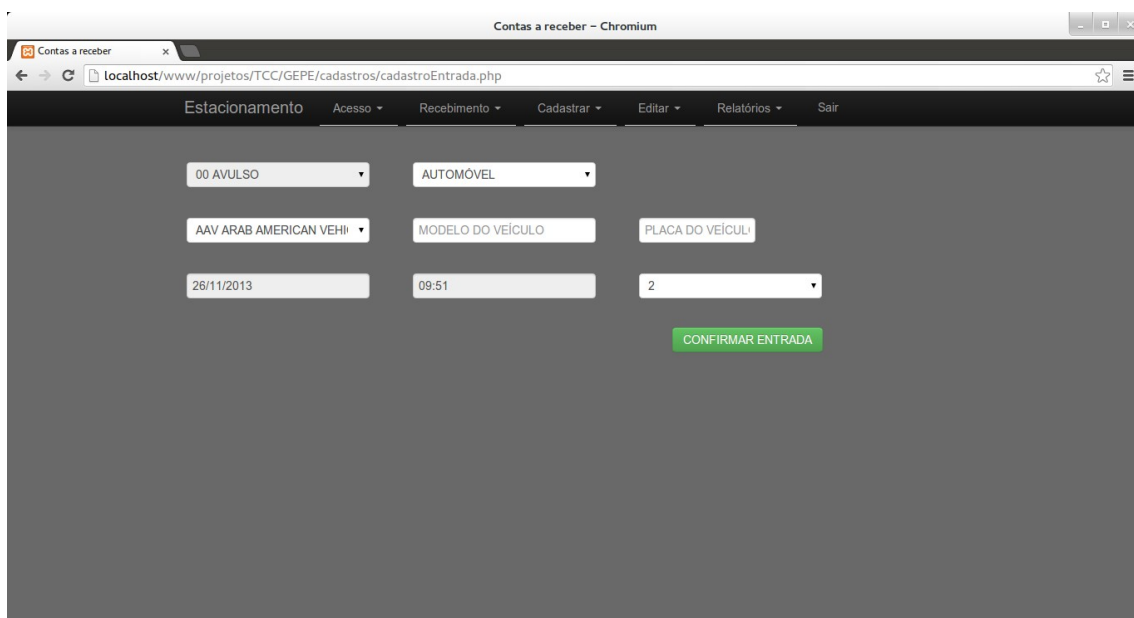
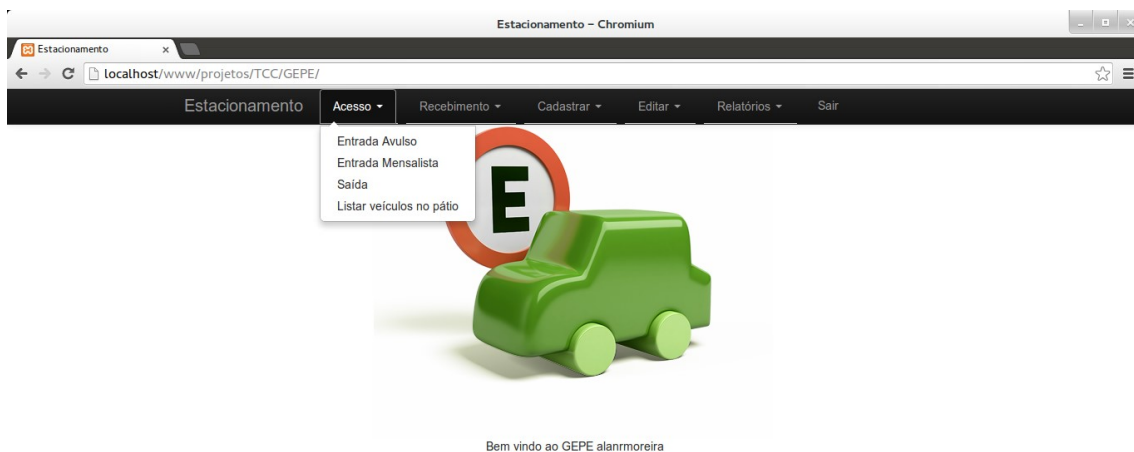
Tela principal do sistema, através dela o usuário pode escolher entre as diversas funcionalidades do sistema clicando nos botões/Abas para escolher um submenu. Após o clique, o usuário será redirecionado para a página correspondente a opção escolhida.



3 Acesso

3.1 Entrada avulso

É através da tela de entrada avulso que o usuário pode registrar uma nova entrada de cliente, preenchendo os campos do formulário com as informações necessárias para mais tarde, poder efetuar a saída do cliente

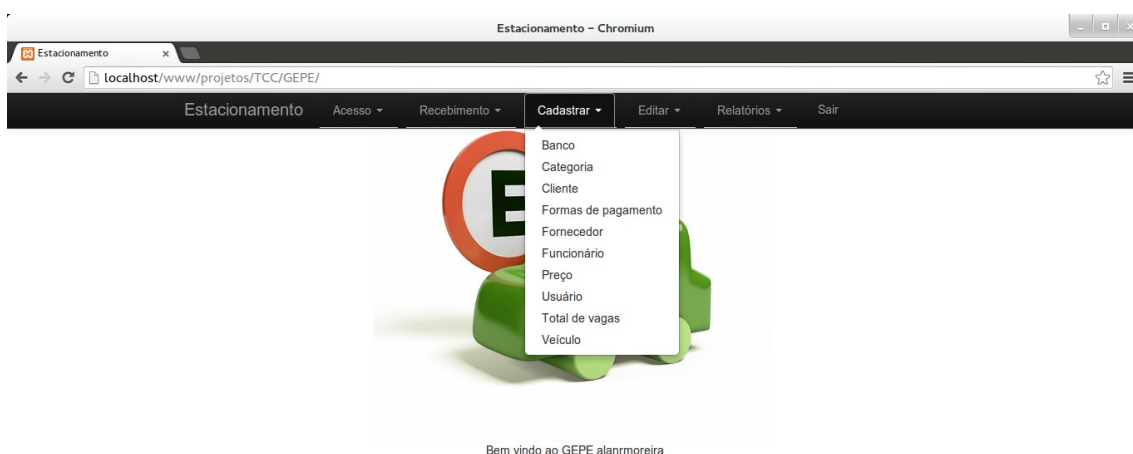


3.2 Entrada mensalista

É através da tela de entrada mensalista que o usuário pode selecionar um cliente cadastrado, para que o sistema preencha automaticamente os formulários com as opções disponíveis, facilitando a entrada do cliente e não cobrando na hora da saída, pois o pagamento é efetuado mensalmente.

4 Cadastrar

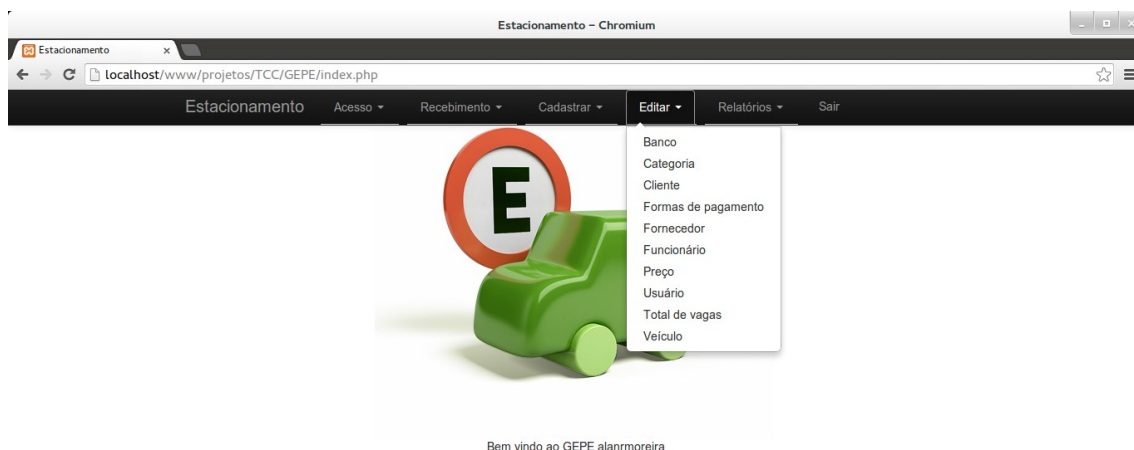
É através das telas de cadastros que o usuário pode cadastrar banco, categoria, cliente, formas de pagamento, fornecedores, funcionários, preço, usuário, total de vagas e veículos, para que os mesmos possam utilizados posteriormente no sistema.



Bem vindo ao GEPE alanmoreira

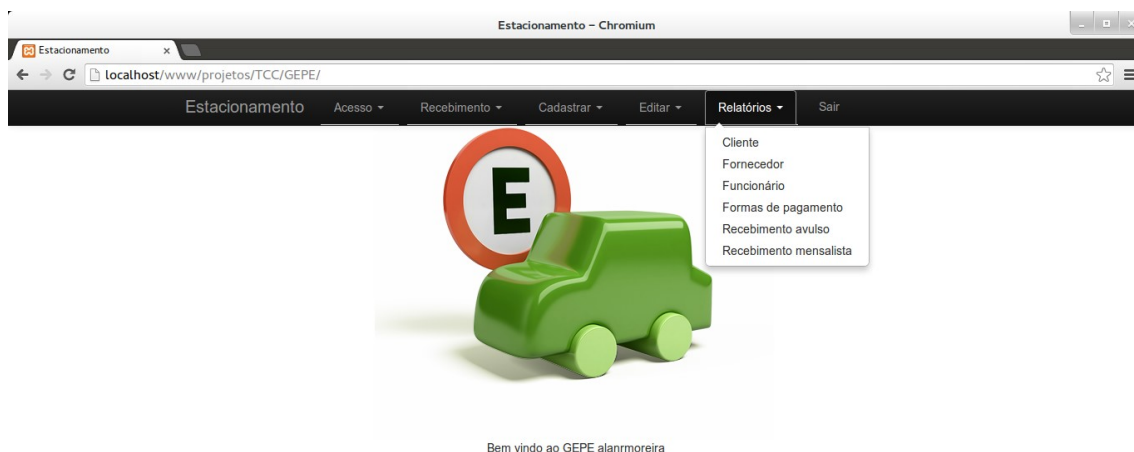
5 Editar

Qualquer cadastro pode ser editado através da tela de edição do sistema, para cada submenu escolhido, há uma tela de edição correspondente como banco, categoria, cliente, formas de pagamento, fornecedores, funcionários, preço, usuário, total de vagas e veículos.



6 Relatórios

A partir da aba de relatórios, o usuário pode escolher um submenu para gerar um relatório que irá exibir informações de acordo com o item escolhido. Após a exibição do relatório contendo as informações, o usuário também poderá excluir ou editar os cadastros listados.



7 Sair

Após clicar em sair, o usuário será redirecionado a página de *Login*, onde deve efetuar o mesmo caso queira operar o sistema.