

**FACULDADE FORTIUM**

**Sistemas de Informação**

GABRIEL CABRAL DE ALMEIDA

WAGNER DE ARAÚJO ALVES

SISTEMA DE GERENCIAMENTO DE ESTACIONAMENTO - STACIONE

Brasília - DF

2015

GABRIEL CABRAL DE ALMEIDA

WAGNER DE ARAÚJO ALVES

SISTEMA DE GERENCIAMENTO DE ESTACIONAMENTO - STACIONE

Trabalho de Conclusão de Curso apresentado à Faculdade Fortium como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Sérgio Rodrigues Lima

Brasília – DF

2015

GABRIEL CABRAL DE ALMEIDA

WAGNER DE ARAÚJO ALVES

SISTEMA DE GERENCIAMENTO DE ESTACIONAMENTO - STACIONE

Trabalho de Conclusão de Curso apresentado à Faculdade Fortium como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Sérgio Rodrigues Lima

Sérgio Rodrigues Lima

Professor 1

Professor 2

Brasília – DF

2015

Dedicamos este projeto primeiramente a Deus, pois a ele toda honra e toda glória; as mães, pelo esforço e apoio em todos os momentos; as esposas que foram nossos braços direito, pela compreensão e dedicação com nossos filhos; aos colegas de classe, que sabem o quanto é difícil conciliar estudo e trabalho; aos professores e a todos que de alguma forma contribuíram para este projeto.

**AGRADECIMENTOS**

Agradeço а todos os professores por nos proporcionar о conhecimento não apenas racional, mas а manifestação do caráter е afetividade da educação no processo de formação profissional, por tanto que se dedicaram а nós, não somente por terem mе ensinado, mas por terem mе feito aprender. А palavra mestre, nunca fará justiça аоs professores dedicados аоs quais sem nominar terão оs meus eternos agradecimentos.

*“Você não pode simplesmente perguntar ao usuário o que ele quer e então tentar dar-lhe isso. Quando você conseguir terminar o produto, o usuário estará querendo outra coisa. ” (Steve Jobs)*

**RESUMO**

O trabalho apresenta o Sistema de Gerenciamento de Estacionamento – STACIONE com objetivo de automatizar e agilizar os processos feitos manualmente pelas empresas. Através da análise de requisitos e da comparação com sistemas similares ao STACIONE é possível ter uma visão clara dos requisitos funcionais do sistema. A utilização do IBM Processo Unificado da Rational – IRUP e da Linguagem de Unificada de Modelagem – UML em sua vasta recursividade está presente na modelagem do sistema, nas atividades e nos artefatos produzidos. Para a codificação do STACIONE é utilizado à linguagem de programação PHP 5.6. Os dados são gerenciados pelo Sistema Gerenciador de Banco de Dados – SGDB Mysql 5.6. Por se tratar de um sistema voltado para internet os padrões do Word Web Consortiun - WC3 são essenciais para o bom funcionamento do sistema, para tal é utilizado à linguagem de marcação de hipertexto – HTML5 e sistema de estilo em cascata – CSS3. Ao final o sistema é submetido a vários testes, resultando em um produto de software com qualidade e que atende os requisitos propostos.

Palavras-chaves: sistema. Requisitos. Programação. Banco de dados

**ABSTRACT**

*The work presents the Parking Management System - STACIONE with objective to automate and streamline processes OS Manually Made By Companies. Through the Requirements Analysis and Comparison with SIMILAR systems When STACIONE and possible have a clear view of the Functional System Requirements. Using the IBM Rational Unified Process - IRUP and Unified Modeling Language - UML IN YOUR Wide recursion IS present in the System Modeling, NAS Activities and artifacts produced nos. To make a STACIONE coding E Used to PHP programming language 5.6. The data are managed by the Database Management System - DBMS MySQL 5.6. Because it is geared to- hum System Internet Standards OS to Word Web Consortiun - WC3 Are Essential for Good RUN System, paragraph E Used as the Hypertext Markup Language - HTML5 and cascading style system - CSS3. The final THE VARIOUS system and subjected to tests, resulting in a software product with quality and that meets the proposed requirements.*

*Keywords: system. Requirements. Programming. Database*

**LISTA DE FIGURAS**

Figura 1 – Engenharia de *Software* em Camadas 16

Figura 2 – Modelo Cascata 18

Figura 3 – Modelo de Desenvolvimento Evolucionário 19

Figura 4 – Modelo Espiral 20

Figura 5 – Modelo Incremental 21

Figura 6 – Ciclos de Desenvolvimento do RUP 22

Figura 7 – Visão e contribuição do W3C 31

Figura 8 – Sistema de banco de dados 34

Figura 9 – Sistema Gerenciador de Banco de Dados 35

Figura 10 – Exemplo Tabelas Banco de Dados 36

Figura 11 – Programa *Astah Community* 42

Figura 12 – Programa *NetBeans* 43

Figura 13 - Programa *MySQL Workbench* 44

Figura 14 - Programa do WAMPP 45

**LISTA DE QUADROS**

Quadro 1 – Quadro TIOBE 40

Quadro 2 – Quadro de Análise de Riscos 50

Quadro 3 – Descrição de Tarefas 52

Quadro 4 – Cronograma 53

**LISTA DE SIGLAS**

**CASE –** Computer Aided Software Engineering  
**CSS –** *Cascading Style Sheet* **GNU -** *General Public Licence***HTML -** *HyperText Markup Language* **IRUP** - *IBM Rational Unified Process* **JS –** *JavaScript***MD5-** *Message-Digest algorithm 5* **MVC-** *Model View e Controller* **NCSA –** *National Center for Computer Aplications* **OMG -** *Object Management Group* **OMT -** *Object Modeling Technique***OOSE -** *Object-Oriented Software Engineering* **PHP -** *HyperText Preprocessor***SGBD -** Sistema de Gerenciamento de Banco de Dados**STACIONE –** Sistema de gerenciamento de Estacionamento **SMIL -** Multimídia Sincronizada **SQL –** *Structured Query Language* **TCP –** *Transmission Control Protocol***UML -** *Unified Modeling Language* **XHTML -** *eXtensible Hypertext Markup Language* **WAI -** *Acessibilidade do conteúdo na Web* **W3C** *– World Wide Web Consortium* **XML -** Linguagem de Marcação Extensível

**SUMÁRIO**

[1. INTRODUÇÃO 13](#_Toc436387991)

[1.1. Objetivos 13](#_Toc436387992)

[1.2. Estrutura do Trabalho de Conclusão 14](#_Toc436387993)

[2. REFERENCIAL TEÓRICO 15](#_Toc436387994)

[2.1. Engenharia de *Software* 15](#_Toc436387995)

[2.2. IBM Rational Unified Process (RUP) 22](#_Toc436387996)

[2.3. *Unified Modeling Language* (UML) 26](#_Toc436387997)

[2.4. Padrões de Projeto 28](#_Toc436387998)

[2.5. *World Wide Web Consortium* (W3C) 30](#_Toc436387999)

[2.6. Banco de Dados 33](#_Toc436388000)

[2.6.1 MySQL 37](#_Toc436388001)

[2.7. Linguagem de Programação 38](#_Toc436388002)

[*2.8.* Servidor *Web* 40](#_Toc436388003)

[2.9. Ferramentas 41](#_Toc436388004)

[2.10. Requisitos de Segurança de Software 45](#_Toc436388005)

[3. METODOLOGIA 48](#_Toc436388006)

[3.1. Concepção 48](#_Toc436388007)

[3.2. Elaboração 54](#_Toc436388008)

[3.3. Projeto de Bando de Dados 55](#_Toc436388009)

[3.4. Projeto de Analise e Design 56](#_Toc436388010)

[3.5. Segurança 57](#_Toc436388011)

[3.6. Teste e Implantação 59](#_Toc436388012)

[3.7. Implantação 60](#_Toc436388013)

[4. CONCLUSÃO 64](#_Toc436388014)

[REFERÊNCIAS 65](#_Toc436388015)

[APÊNDICES A - ENTREVISTA 69](#_Toc436388016)

[APÊNDICES B - DOCUMENTO DE VISÃO 70](#_Toc436388017)

[APÊNDICES C – DIAGRAMA E ESPECIFICAÇÃO DE CASO DE USO 74](#_Toc436388028)

[APÊNDICES D - DIAGRAMAS DE SEQUÊNCIA 75](#_Toc436388158)

[APÊNDICES E - DIAGRAMAS DE ATIVIDADES 83](#_Toc436388159)

[APÊNDICES F - DIAGRAMA DE IMPLANTAÇÃO 88](#_Toc436388160)

[APÊNDICES G - DIAGRAMA DE CLASSES 89](#_Toc436388161)

[APÊNDICES H - MER FÍSICO E DICIONÁRIO DE DADOS 89](#_Toc436388162)

[APÊNDICES I - TESTE DE QUALIDADE 94](#_Toc436388164)

[APÊNDICES J – CD/DVD COM IMPLEMENTAÇÃO 109](#_Toc436388165)

[APÊNDICES L – ROTEIRO DE TESTE 110](#_Toc436388166)

[APÊNDICES m – TESTE DE STRESS 110](#_Toc436388166)

[APÊNDICES n – MANUAL DE INSTALAÇAO DO SISTEMA 121](#_Toc436388167)

# INTRODUÇÃO

As organizações ou empresas necessitam de uma gestão adequada e eficiente para que a mesma possa conseguir alcançar o sucesso, ou seja, o gerenciamento e a administração da instituição se tornam indispensáveis.

A gestão surgiu quando após a revolução industrial, os profissionais decidiram buscar solução para problemas que não existiam antes, usando vários métodos de ciências, para administrar os negócios da época o que deu início a ciência da administração, pois é necessário o conhecimento e aplicação de modelos e técnicas administrativas.

A implantação de um sistema de gestão é uma etapa muito importante para o sucesso de um negócio, com o avanço da tecnologia, facilita o controle da empresa e de seu produto que no caso do STACIONE.

Neste projeto serão apresentadas as características do STACIONE, mostrando informações importantes para o entendimento do sistema bem como suas funcionalidades através da análise de requisitos e da comparação com sistemas similares ao STACIONE é possível ter uma visão clara dos requisitos funcionais do sistema. Para maximizar o desenvolvimento e o desempenho proposto, é utilizada a linguagem PHP que possui uma grande quantidade de extensões e bibliotecas disponíveis para criar um sistema seguro e funcional, permitindo uma grande quantidade de mecanismos para gerenciar os dados armazenados no banco de dados MySQL, alterações de dados e controle de entrada e saída de veículos.

## Objetivos

* + 1. **Geral**

Este projeto é desenvolvido para atender empresas que trabalham com estacionamento, o sistema auxilia na gestão da empresa controlando a entrada e saída de veículos, registrando funcionário que permite fazer o controle da entrada. O trabalho dos gerentes e atendentes, para que estes usufruam de um maior desempenho e eficiência em suas atividades.

* + 1. **Específicos**

a) Elaborar a visão do sistema, avaliando os possíveis riscos, estabelecendo as prioridades, levantando e documentando os requisitos e mantendo o acompanhamento das mudanças desses requisitos;

b) Modelar o sistema visualmente utilizado modelos os diversos diagramas da UML para apresentar as visões estáticas e dinâmicas do sistema;

c) Desenvolver o sistema de maneira incremental e iterativa;

e) Testar, implantar e verificar a qualidade do sistema garantindo que atenda aos padrões de qualidade da organização.

## Estrutura do Trabalho de Conclusão

Este documento está dividido em quatro capítulos, os quais estão organizados da seguinte forma:

Capítulo 2 – Referencial Teórico: apresenta conceitos inerentes ao sistema, como também a metodologia e as técnicas a serem utilizadas, a análise das necessidades e problemas diagnosticados;

Capítulo 3 – Metodologia: apresenta a visão geral do sistema, contendo a classificação dos requisitos gerais identificados para solucionar o problema, a lista e as especificações dos casos de uso, os diagramas de atividades, sequência e colaboração de cada caso de uso, o projeto lógico, o diagrama relacional do sistema e o de entidade e relacionamento; mecanismos de segurança da linguagem de programação e SGBD, bem como segurança de privacidade de dados; teste de implantação, testes de qualidade, stress e segurança, bem como a implantação do sistema;

# REFERENCIAL TEÓRICO

São apresentadas as referências utilizadas no sistema, ou seja, todas as tecnologias e ferramentas utilizadas. Neste capítulo consta o referencial teórico para o desenvolvimento do projeto.

## Engenharia de *Software*

O *software*, hoje em dia, está em todos os âmbitos, e em constante crescimento. Cada vez mais pessoas necessitam do produto software. Com esse crescimento, o software ganhou um papel muito importante na sociedade.

Quando o cliente recebe o produto (software) quase nunca é considerado perfeito, há sempre uma correção a ser feita, ou seja, a manutenção é constante. O software passa por vários estágios. Aos poucos se deprecia, muitas vezes, a tal ponto que a empresa percebe que não tem como mais manter o produto no mercado, substituindo-o por um novo produto ou versão.

Segundo Sommerville (2003), engenharia de software “são todos os processos de software desde o seu estado inicial, na produção, até a fase de manutenção do sistema.”

Na engenharia de software, os engenheiros aplicam teorias, métodos de desenvolvimento e ferramentas em um processo seletivo, buscando soluções para os problemas mesmo quando não existem teorias aplicáveis e métodos que apoiam no desenvolvimento.

A engenharia de software adota uma abordagem sistemática para a produção de software de alta qualidade selecionando métodos, ferramentas e teorias para o melhor processo de desenvolvimento do software.

A engenharia de software não se destina somente a processos técnicos de desenvolvimento, dedica-se também, ao gerenciamento de projetos, desenvolvimento de ferramentas, métodos de desenvolvimento e teorias que apoiam a produção do software.

Segundo Pressman (2006), o software e o hardware possuem semelhanças no decorrer do desenvolvimento. Os dois conquistam a qualidade através de um bom projeto.

### O desenvolvimento em camadas

Segundo Pressman (2006), a “engenharia de software é uma tecnologia em camadas” conforme apresentado na Figura 1.

Figura 1 - Engenharia de Software em Camadas.



Fonte: Pressman (2006)

O processo de engenharia de software é que mantém unido as camadas de tecnologia que permitem o desenvolvimento de software de computador. O processo define como deve ser utilizada a tecnologia de engenharia de software, formando uma base de controle gerencial nos projetos em que os métodos são aplicados nos produtos de trabalho, assim, os marcos são estabelecidos, e a qualidade é assegurada no decorrer do desenvolvimento.

Os métodos de engenharia de software definem as técnicas de como desenvolver o software, incluindo as modelagens de desenvolvimento e metodologias a serem aplicadas.

As ferramentas de engenharia de software apoiam o desenvolvimento de software baseadas em métodos. Por exemplo, existem ferramentas que apoiam a modelagem ou a descrição de processos, entre outras. Essa prática é chamada de “Engenharia de Software apoiada”.

### Arcabouço de processo de software

Segundo Pressman (2006), “um arcabouço de processo de software estabelece um alicerce para os processos de software que podem ser aplicados a projetos de qualquer tamanho.” O arcabouço de processo é o desenvolvimento do projeto em blocos em que cada pessoa tem uma função naquele determinado bloco de atividades.

Na maioria dos projetos, existe um arcabouço genérico que é aplicado no desenvolvimento do projeto de software, dividido em cinco fases:

• Comunicação: abrange a atividade que envolve a comunicação entre o cliente e os interessados no levantamento de requisito e atividades relacionadas ao projeto.

• Planejamento: Abrange a descrição das tarefas técnicas a serem conduzidas, os riscos, os recursos necessários, os produtos de trabalho para serem produzidos e o cronograma do projeto.

• Modelagem: é uma atividade que inclui o desenvolvimento de modelos, requisitos de software para o melhor entendimento do cliente junto ao projeto o que será realizado durante as etapas do desenvolvimento.

• Construção: é a atividade que gera a implementação do código e o teste do produto em busca de erros durante o processo de desenvolvimento do aplicativo.

• Implantação: é o produto que o cliente recebe para avaliação. Neste momento, o cliente avalia o produto, e, caso surgirem novas solicitações, retorna para a fase de planejamento, (PRESSMAN, 2006).

Essas cinco atividades apresentadas podem fazer parte de um projeto de software grande ou pequeno, pois os processos se aplicam para qualquer tipo de projeto.

### Modelos de processos de software

No decorrer dos últimos anos, vários modelos foram lançados no mercado. A seguir, são apresentados alguns deles.

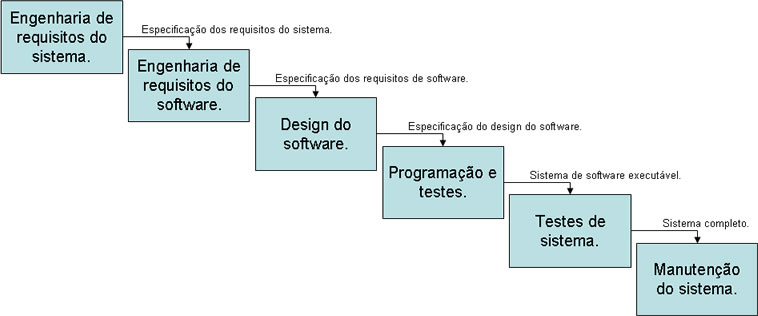
### Modelo cascata

Segundo Pressman (2006, p.38), a modelo cascata “possui um ciclo de vida clássico em que são abordados os principais estágios do modelo de desenvolvimento, abordando, assim, um modelo sistemático e sequencial. ” O modelo é demonstrado conforme a Figura 2.

Raros são os projetos que seguem um fluxo sequencial, como o sugerido pelo modelo, mesmo que o modelo acomode iterações, as modificações podem causar confusão na equipe no decorrer do projeto. Um dos pontos é o fato de que, no início do projeto, é raro o cliente estabelecer todos os requisitos, e a modelo cascata exige esse conhecimento em uma etapa muito inicial.

Nesse modelo, o cliente precisa ter muita paciência, pois o executável do software é gerado no final da iteração.

Figura 2 - Modelo Cascata



Fonte: Pressman (2006)

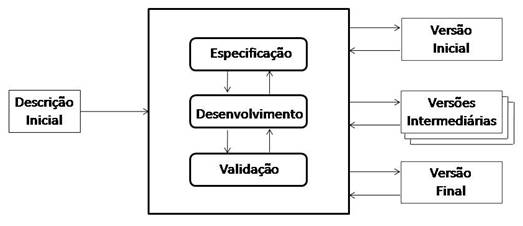
### Modelo Evolucionário

Segundo Sommerville (2003, p.39), o modelo evolucionário “é o desenvolvimento de uma implementação inicial, apresentando o sistema para o cliente para realização de aprimoramentos. ” Esse método desenvolve várias versões até que o sistema esteja adequado às necessidades do cliente. O modelo pode ser classificado em:

• Exploratório: possui o objetivo de trabalhar com o cliente a fim de observar os requisitos e entregar uma versão final ao cliente, o sistema evolui de acordo com as especificações do cliente.

• Protótipos descartáveis: o objetivo é compreender os requisitos passados pelo cliente, criando protótipos do sistema a fim de melhor definir os requisitos que não foram bem compreendidos. Com esse artefato, inicia-se o desenvolvimento conforme a Figura 3.

Figura 3- Modelo de Desenvolvimento Evolucionário



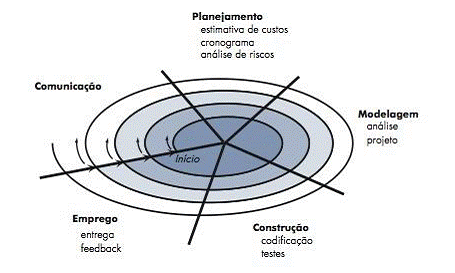
Fonte: Sommerville (2003)

A abordagem, nesse método, é mais eficaz que a modelo cascata por atender melhor as necessidades dos clientes. A grande vantagem desse modelo é que o desenvolvimento é gradativo. Segundo Sommerville (2003), essa abordagem possui três problemas: “O processo não é visível, os sistemas são, frequentemente, mal estruturados e podem ser exigidas ferramentas e técnicas específicas. ”

### Modelo Espiral

Segundo Pressman (2006, p.44), modelo espiral “é um modelo de desenvolvimento de software que utiliza o modelo evolucionário em parceria com o modelo cascata. ” Nesse modelo, as iterações são realizadas de uma forma de espiral, como mostra a figura a seguir.

Figura 4 - Modelo Espiral



Fonte: Pressman (2006)

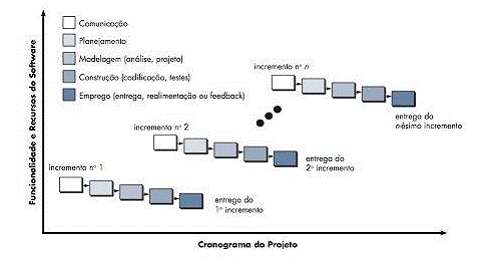
Esse modelo aborda o desenvolvimento de uma forma evolucionária, originando, assim, várias versões: o início se dá com uma nova iteração, assim, o software ganha novas versões a cada iteração, no final das iterações, o software é entregue ao cliente.

O modelo aborda o desenvolvimento de software de forma realista, em que em cada iteração, o software evolui tornando-se mais maduro.

* + - 1. **Modelo Incremental**

Segundo Pressman (2006, p.40), o modelo incremental “utiliza-se do modelo cascata nas suas iterações. Cada iteração do projeto segue o modelo cascata, incrementando-se e surgindo, assim, a cada final de estágio, unido uma nova versão do software. ” A figura a seguir mostra como é o modelo incremental.

Figura 5 - Modelo Incremental



Fonte: Pressman (2006)

No modelo incremental, o primeiro incremento ou a primeira iteração é usada como núcleo do produto. Neste núcleo, os requisitos básicos do software são satisfeitos na primeira iteração. Muitas vezes, o cliente já começa a usar o software. Esse modelo tem por base a melhoria contínua do software, pois o cliente já usa uma versão para avaliação. A utilização do cliente, logo em fases iniciais, faz com que sejam solicitadas novas funcionalidades e melhorias nos processos das atividades do software.

Apesar dos inúmeros modelos existentes, percebeu-se a necessidade de flexibilização no processo de desenvolvimento de software e, a partir dessa percepção, deu-se início ao movimento ágil.

## IBM Rational Unified Process (RUP)

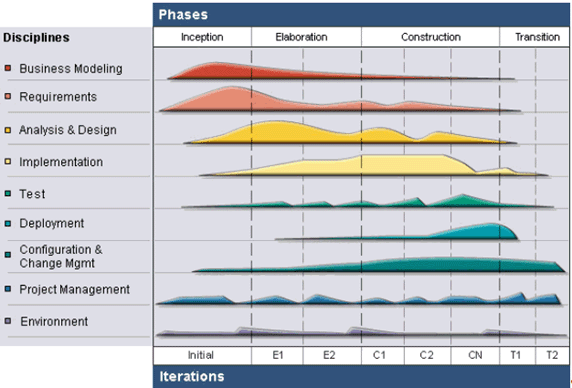
O *Rational Unifield Process* – RUP é um processo de engenharia de software comercializado por *Rational* Software. (KRUCHTEN, 2003)

A maioria das descrições do RUP tenta combinar as perspectivas estática e dinâmica em um único diagrama. Por achar que essa tentativa torna o processo mais difícil de ser compreendido, usa-se descrições separadas de cada perspectiva. (KRUCHTEN, 2003, p. 29)

A visão estática do RUP prioriza as atividades que ocorrem durante o processo de desenvolvimento. Na descrição do RUP, essas são chamadas workflows. Existem seis workflows centrais, identificadas no processo, e três workflows de apoio. O RUP foi projetado em conjunto com a UML, assim, a descrição do workflow é orientada em torno de modelos associados à UML, como um modelo de sequência, modelos de objetos etc. Os workflows centrais de engenharia e de apoio estão descritos na figura 1. (SOMMERVILLE, 2014, p. 29)

Os workflows centrais de engenharia e de apoio estão descritos na figura 6.

Figura 6 – Ciclos de Desenvolvimento do RUP



Fonte: Martins (2007)

Para capturar a dimensão do tempo de um projeto, o RUP divide o projeto em quatro fases, conforme demonstrado na figura 6, são elas:

1. Concepção: O objetivo desta fase é estabelecer um business case para o sistema. Deve-se identificar todas as entidades externas (pessoas e sistemas) que vão interagir com o sistema e definir as interações. Então, usar essas informações para avaliar a contribuição do sistema para o negócio. Se essa contribuição for pequena, então o projeto poderá ser cancelado depois dessa fase. (SOMMERVILLE, 2014, p. 29).
2. Elaboração: As metas da fase de elaboração são desenvolver uma compreensão do problema dominante, estabelecer um framework da arquitetura para o sistema, desenvolver o plano do projeto e identificar os maiores riscos do projeto. No fim dessa fase, deve-se ter o modelo de requisitos para o sistema, que pode ter um conjunto de casos de uso da UML, uma descrição da arquitetura ou um plano de desenvolvimento do software. (SOMMERVILLE, 2014, p. 29)
3. Construção: A fase de construção envolve projeto, programação e teste de sistema. Durante essa fase, as partes do sistema são desenvolvidas em paralelo e integradas. Na conclusão dessa fase deve ter um sistema de software já funcionando, bem como a documentação associada pronta para ser entregue aos usuários. (SOMMERVILLE, 2014, p. 29).
4. Transição: A fase final do RUP implica transferência do sistema da comunidade de desenvolvimento para a comunidade de usuários e em seu funcionamento de um ambiente real. Isso é ignorado na maioria dos modelos de processo de software, mas é, de fato, uma atividade cara e, às vezes, problemática. Na conclusão dessa fase, obrigar-se ter um sistema de software documentado e funcionando corretamente em seu ambiente operacional. (SOMMERVILLE, 2014, p. 29).

As fases indicam a ênfase dada no projeto em um determinado instante.

No RUP o desenvolvimento é feito de forma iterativa e incremental, pois é um dos clássicos modelos de processo de desenvolvimento de software criado em respostas as fraquezas do modelo em cascata, o mais tradicional.

**Modelagem de Negócios**: Visa a atender a estrutura, a dinâmica e o problemas da organização onde o sistema será desenvolvido, identificando potenciais melhorias. Com isso atinge-se um melhor entendimento das necessidades e problemas que devem ser resolvidos pelo sistema. Alguns artefatos associados são: modelos de caso de uso de negócio, modelo 18 de objetos de negócio, glossário de negócios, documentos de arquitetura de negócios etc. (PRIKLADNICK & AUDY, 2007, p.23)

**Requisitos:** Visa estabelecer e manter um acordo com os clientes e outros *stakeholders* com relação ao que o sistema deve fazer e por que motivo, fornecendo aos desenvolvedores um melhor entendimento dos requisitos do sistema. Além disso são definidos o escopo do sistema e as bases para estimativas de tempo e custo de desenvolvimento. Os artefatos utilizados são modelos de casos de uso, especificação dos requisitos do sistema, visão do sistema, glossário e protótipo de interface. (PRIKLADNICK & AUDY, 2007, p.23)

**Análise de Projeto:** Visa a transformar os requisitos em uma especificação que descreve como implementar um sistema. Além disso visa estabelecer uma arquitetura robusta para o sistema e adaptar o projeto para o ambiente de implementação, levando-se em conta critério de desempenho, escalabilidade e testabilidade. Alguns dos artefatos utilizados são: o modelo de objetos (modelo de projeto), documentos de arquitetura de software, modelo de análise e projeto e modelo de dados. (PRIKLADNICK & AUDY, 2007, p.23)

**Implementação:** Visa a desenvolver o sistema no ambiente de implementação definido. Além disso visa organizar o código em termos de subsistemas de implementação organizados em camadas; implementar classes e objetos em termos de componentes (arquivos-fontes, binários, executáveis e outros); testar os componentes desenvolvidos como unidades; interagir em um sistema executável os resultados produzidos por indivíduos ou equipes. Alguns artefatos são: código-fonte, executáveis e arquivos de documentação. O escopo do teste dentro do workflow de implementação é limitado a testes de componentes individuais. Testes de integração e de sistema são descritos no workflow de teste. (PRIKLADNICK & AUDY, 2007, p.24)

**Teste:** Visa a garantir que o sistema final é aquele que foi solicitado pelos *stakeholders* interessados. Nesse workflow são verificadas as interações e a corrente integração entre os componentes, além de identificar e assegurar que todos os defeitos descobertos foram verificados antes da implantação do software. Os artefatos resultantes podem ser certificados de que o sistema está pronto para uso em forma de relatórios de testes, além de plano de teste, caso de teste, entre outros. (PRIKLADNICK & AUDY, 2007, p. 24)

**Implantação:** Visa a disponibilizar para os usuários o produto final de software. Nesse workflow são realizadas as atividades de instalação, distribuição, teste e treinamento do software em seu ambiente final de operação, bem como a migração do software ou as bases de dados existentes, caso necessário. Alguns artefatos são: fatura de materiais, artefatos de instalação, material de treinamento, e manual do usuário. (PRIKLADNICK & AUDY, 2007, p. 24)

**Gerenciamento de Configuração de Mudanças:** Visa o controle das modificações é a manutenção da integridade dos artefatos do projeto. Neste workflow são realizadas atividades de identificação dos itens de configuração, restrição e auditoria das alterações. Alguns artefatos associados são o plano de gerência de configuração e a requisição de mudanças (Change Request). (PRIKLADNICK & AUDY, 2007, p. 24)

**Gerenciamento de Projeto:** Visa a balancear objetivos, riscos e restrições para entrega de um produto de acordo com as necessidades dos *stakeholders*. Tem como propósito fornecer um conjunto de atividades para a gerência de projetos de software, manuais de orientação (guidelines) para o planejamento, gerência de pessoal, execução e monitoramento de projetos, além de práticas de gerência de riscos. Alguns artefatos são: plano de desenvolvimento de software, caso de negócio, plano de iteração, registro de revisões e plano de garantia de qualidade. (PRIKLADNICK & AUDY, 2007, p. 24)

**Ambiente:** Essa fase é similar a abordagem dada a utilização de outros guias de boas práticas ou processos de desenvolvimento – a customização. (SOMMERVILLE, 2011, p. 40).

O RUP é utilizado no sistema STACIONE por se tratar de um modelo interativo e incremental. Essas características permitem definir o problema e construir o software em vários passos, diminuindo assim os riscos associados ao projeto, visto a mutabilidade dos requisitos do software no decorrer do seu desenvolvimento.

## *Unified Modeling Language* (UML)

A Linguagem de Modelagem Unificada (do inglês, UML - Unified Modeling Language) é uma linguagem de modelagem que permite representar um sistema de forma padronizada.

A Unified Modeling Language – UML ou Linguagem de Modelagem Unificada é uma linguagem de forma visual utilizada para modelar software que são elaborados no paradigma de orientação a objetos, sendo uma linguagem de modelagem geral, ou seja, pode ser aplicada a todos os domínios de aplicação. A UML se tornou, nos últimos anos, a linguagem padrão de modelagem sendo usada internacionalmente na Engenharia de Software. (GUEDES, 2011).

A UML não é uma linguagem de programação, é uma linguagem de modelagem, uma notação, cujo objetivo é auxiliar os engenheiros de software a definirem as características do sistema, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e até mesmo suas necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implantado. (GUEDES, 2011, p. 15)

Conforme Engholm (2010, p. 68), a linguagem unificada de modelagem UML fornece diagramas de modelagem que podem ser utilizados nos processos do ciclo de desenvolvimento de software, que são eles:

1. Diagrama de Caso de Uso: Contém elementos modelados para um sistema, ou seja, os atores que irão utiliza-lo e os casos de uso, e mostra os diferentes relacionamentos como também generalizações, associações e dependência entre esses elementos. (REZENDE, 2005)
2. Diagrama de Interação: Na maioria das vezes, utilizado na fase de design para representar o relacionamento entre os objetos do sistema. (ENGHOLM, 2010)
3. Diagrama de Atividades: O diagrama de atividade pretende descrever um comportamento paralelo ou apresentar como interagem comportamentos em vários casos de uso. (REZENDE, 2005)
4. Diagrama de Implantação: Indica como o sistema deverá ser implantado e hospedado, geralmente elaborado na fase de design. (ENGHOLM, 2010)

A UML não é a solução dos problemas de desenvolvimento de software, mas pode ser usada para mostrar descrição de um sistema e suas funcionalidades, utilizando para isso os atores e os casos de uso, mostrando a realização dos casos de uso com diagramas de 21 interação, detalhando a estrutura estática do sistema com os diagramas de classe, detalhar a arquitetura de implementação física com os diagramas de implantação. (REZENDE, 2005)

A UML surgiu da união de três métodos de modelagem: o método de Booch, o método OMT (*Object Modeling Technique*) de Jacobson, e o método OOSE (*Object-Oriented Software Engineering*) de Rumbaugh. Estes eram, até meados da década de 1990, os métodos de modelagem orientada a objetos mais populares entre os profissionais da área de desenvolvimento de software. (GUEDES, 2011, p. 19)

Apesar de a UML oferecer vários tipos de diagramas que descrevem características tanto estruturais quanto comportamentais de um software especifico, não é obrigatório a utilização de todos os diagramas propostos na modelagem de um sistema, pois cada diagrama possui uma função especifica e única e em algumas vezes, alguns destes não é necessário para determinadas situações ou domínios. (GUEDES, 2011)

De acordo com Rezende (2005, p. 202), a UML complementa os conceitos das técnicas de modelagem de dados, que são elas:

1. **Objetos:** Um objeto pode ou não existir em um mundo real, e é alguma coisa que faz sentido dentro de um contexto de uma aplicação. Pode ser definido como um conceito, uma abstração, algo que possui limites claros e significados em relação a um problema. (REZENDE, 2005)
2. **Classes**: É a descrição de um tipo de objeto e todos os objetos devem ser instâncias de uma classe, sendo que a classe descreve as propriedades e os comportamentos de um tipo de objeto. (REZENDE, 2005)
3. **Métodos e Modelos**: Método procura estruturar a ação ou o pensamento de uma pessoa com relação a um determinado problema. E os métodos, possuem modelos que são utilizados para descrever alguma coisa e comunicar o resultado do uso do método. (REZENDE, 2005)
4. **Ferramentas de CASE – UML:** Ferramentas *Computer-Aided Software Engineering* - CASE ou Engenharia de Software Auxiliada por Computador, são softwares que auxiliam para a execução de uma ou mais atividades realizadas durante o processo de Engenharia de Software. Grande parte das ferramentas 22 CASE atuais suportam a UML, sendo em geral sua principal característica. (GUEDES, 2011)

A modelagem UML é utilizada para este projeto, porque permite mostrar a descrição e as funcionalidades de um sistema, auxiliando no desenvolvimento, pois seguindo a ordem dos diagramas propostos por ela facilita o entendimento e consequentemente no desenvolvimento do sistema. A UML determina as regras essências de sequência que deve seguir em um projeto.

## Padrões de Projeto

Um padrão de projeto é uma solução reutilizável para problemas que venham a se repetir no projeto do software. O padrão não é uma solução detalhada que pode ser diretamente transformado em código fonte ou de máquina.

Em vez disso, pode-se pensar nele como uma descrição de conhecimento e experiência, uma solução já aprovada para um problema comum.

Os padrões de projeto são uma ótima ideia, mais para usá-los efetivamente, precisa de experiência em desenvolvimento de software. Tendo a necessidade de reconhecer as situações em que os padrões podem ser utilizados. Programadores inexperientes, mesmo que tenham lido livros, sempre acharam difícil decidir se pode reusar um padrão ou se é necessário desenvolver uma solução especial. (SOMMERVILLE, 2011, p. 20)

Enfim, o conhecimento de padrão de projeto permite decidir o que deve ser feito e o que deve ser evitado, sistemas baseados em padrão têm mais qualidade. Os padrões de projeto utilizados para o desenvolvimento do Locação de Veículos são:

1. Padrões de criação: Eles ajudam a tornar um sistema independente de como seus objetos são criados, compostos e representados. Um padrão de criação de classe usa a herança para variar a classe que é instanciada, enquanto que um padrão de criação de objeto delegará a instanciação para outro objeto. (SOMMERVILLE, 2011, p.27)
2. Padrões Estruturais: Os padrões estruturais se preocupam com a forma como classes e objetos são compostos para formar estruturas maiores. Os de classes utilizam a herança para compor interfaces ou implementações, e os de objeto ao invés de compor 23 interfaces ou implementações, eles descrevem maneiras de compor objetos para obter novas funcionalidades.

A flexibilidade obtida pela composição de objetos provém da capacidade de mudar a composição em tempo de execução o que não é possível com a composição estática (herança de classes). (SOMMERVILLE, 2011, p.27)

1. Padrões de Comportamento: Concentram nos algoritmos e atribuições de responsabilidades entre os objetos. Eles não descrevem apenas padrões de objetos ou de classes, mas também os padrões de comunicação entre os objetos”. (SOMMERVILLE, 2011)

Os padrões comportamentais de classes utilizam a herança para distribuir o comportamento entre classes, e os padrões de comportamento de objeto utilizam a composição de objetos em contrapartida a herança. Alguns descrevem como grupos de objetos cooperam para a execução de uma tarefa que não poderia ser executada por um objeto sozinho. (SOMMERVILLE, 2011, p.27)

1. Padrão MVC: *Model-View-Controller* (MVC) é uma solução para desacoplar as camadas de lógica e regras de negócio da camada de apresentação. Implementa três camadas distintas, sendo elas *Model, View* e *Controller*, cada qual com suas características e atribuições em uma aplicação. O MVC foi originalmente concebido na Xerox PARC entre 1978 e 1979 e mais tarde implementado na biblioteca de classe Samalltalk-80.

Quando MVC é referenciado como um design *pattern* ou padrão de design de software, é porque MVC vem sendo adotado como solução recorrente para um problema conhecido, uma solução reutilizável. Essa possibilidade de utilizar um mesmo modelo de desenvolvimento para diversos problemas distintos acabou fazendo com que muitas pessoas apontassem MVC como um design *pattern*. Um framework de certo modo também tem essa característica por servir de base para vários projetos distintos. Você não precisa escrever uma classe de conexão a dados para cada novo web site ou sistema que cria, porque isso já está escrito no framework. Assim, basta reutilizar a solução já desenvolvida, testada e conhecida. (GABARDO, 2011, p.18)

Os padrões de projeto citados são utilizados porque refletem a experiência, conhecimento e soluções dos desenvolvedores que tiveram sucesso usando esses padrões em seus trabalhos.

## *World Wide Web Consortium* (W3C)

O World Wide Web Consortium – W3C é um consórcio internacional com cerca de 300 membros, que agrega empresas, órgãos governamentais e organizações independentes, e que visa desenvolver padrões para a criação e a interpretação de conteúdos para a Web. Foi fundado por Tim Berners-Lee em 1994 para levar a Web ao seu potencial máximo, por meio do desenvolvimento de protocolos comuns e fóruns abertos que promovam a sua evolução e assegurem a sua interoperabilidade.

Com a padronização, os programas conseguem acessar facilmente os códigos e entender onde deve ser aplicado cada conhecimento expresso no documento. Padrões seus como *HyperText Markup Language* (HTML), *eXtensible Hypertext Markup Language* (XHTML) e *Cascading Style Sheets* (CSS) são muito populares, contudo, em muitos casos são usados de forma errônea devido ao desconhecimento da especificação.

W3C desenvolve especificações técnicas e orientações através de um processo projetado para maximizar a consenso sobre as recomendações, garantindo qualidades técnicas e editoriais, além de transparentemente alcançar apoio da comunidade de desenvolvedores, do consórcio e do público em geral. (W3C, 2014)

Possibilitar que o acesso de qualquer tipo de dispositivo seja tão simples, fácil e conveniente quanto de um computador convencional (W3C, 2011).

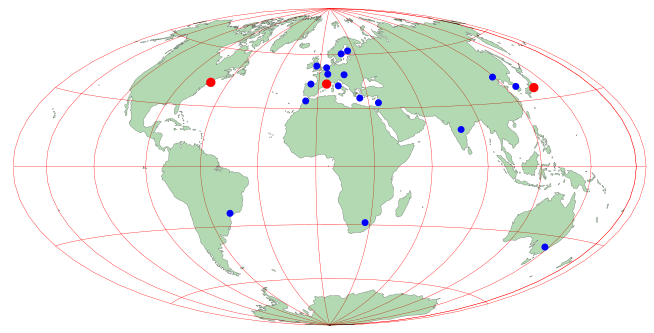
Alguns dos princípios básicos do W3C: Tecnologias web devem ser interoperáveis. A web é baseada em um largo aspecto de tecnologias. Nenhuma tecnologia pode pretender cobrir as necessidades da web. O que torna a interoperabilidade mandatória. Padrões web devem ser abertos, isto é, não apropriados. A web deve ser acessível a todos.

Segue uma lista de atividades do W3C:

*Markup Language* (ou Linguagem de Marcação Extensível) (XML), *Graphics* (Imagens Gráficas), HTML (*HyperText Markup Language*), Internacionalização, *Math* (Matemática), Iniciativa para Web Móvel, Privacidade, Segurança, Web Semântica, Estilo, Vídeo Web, Navegação por voz, Serviços Web, E-Gov, Multimídia Sincronizada (SMIL), Acessibilidade do conteúdo na Web (WAI)

O padrão W3C está inserido em praticamente todas as páginas web, abaixo segue um exemplo da sua visão, demonstrado na figura 7

Figura 7 -Visão e contribuição do W3C



Fonte: http://www.w3c.br

O uso do W3C no projeto se justifica pelo fato de que é necessário que o sistema possa ser usado por navegador, ressaltando que necessita está na rede interna da empresa. E contribuíra na acessibilidade a deficientes físicos, e demonstra que o sistema utiliza as boas práticas que são adotadas no mercado.

### *Cascading Style Sheets* (CSS)

CSS - *Cascading Style Sheets*: em português, pode ser compreendido por Folhas de Estilo em Cascata. É um documento no qual são definidas regras de formatação ou de estilos a serem aplicados aos elementos estruturais de marcação. Tem por finalidade retirar da HTML toda e qualquer declaração que vise à formatação ou à apresentação do documento. De modo geral, pode-se dizer que se usa HTML para estruturar e CSS para apresentar. As principais vantagens do uso das CSS são: saída para diferentes tipos de mídia a partir de uma única versão de HTML, redução do tempo de carga dos documentos *web*, aumento considerável na portabilidade dos documentos *web*, dentre outras.

Com o CSS é possível criar um arquivo com todas as regras de formatação diferente, ou aplicar uma mesma formatação em diversas páginas, será necessário somente referenciar o arquivo CSS que contém as formações, eliminado assim a necessidade de escrever a formatação em todas as páginas individualmente. (SILVA, 2010)

No projeto o CSS é usado para composição das páginas web, por ser um padrão adotado internacionalmente de acordo com os padrões de usabilidade e acessibilidade fornecidos pela W3C.

### *Java Script*

Javascript é uma linguagem de script do lado do cliente que permite que os autores de páginas web para desenvolver páginas web interativas e sites. Embora Javascript é considerada uma linguagem de programação, também é uma parte crítica da página web 32 design e criação. Isso ocorre porque a linguagem Javascript "vive" dentro dos elementos de um web page's. A linguagem é relativamente fácil de aprender, permitindo que nãoprogramadores para incorporar quicky funcionalidade Javascript em uma página web. Na verdade, porque ele é usado extensivamente nas páginas web incontáveis que estão disponíveis na Word Wide Web, Javascript é sem dúvida a linguagem de programação mais usada no mundo. (GOSSELIN, 2010).

A linguagem *JavaScript* foi inventada por Brendan Eich, da Netscape, e a primeira versão da linguagem denominada JavaScript 1.0 foi introduzida no navegador Netscape 2.0 em 1996. Nesse mesmo ano, a Microsoft lançou sua versão com o nome JScript 1.0 e introduziu-a no então Internet Explorer 3.0. Àquela época, em plena guerra dos navegadores, as diferentes implementações das funcionalidades da linguagem nos dois navegadores não seguiam um padrão unificado, causando um verdadeiro martírio para o desenvolvedor implantar scripts para servir ambos os navegadores.

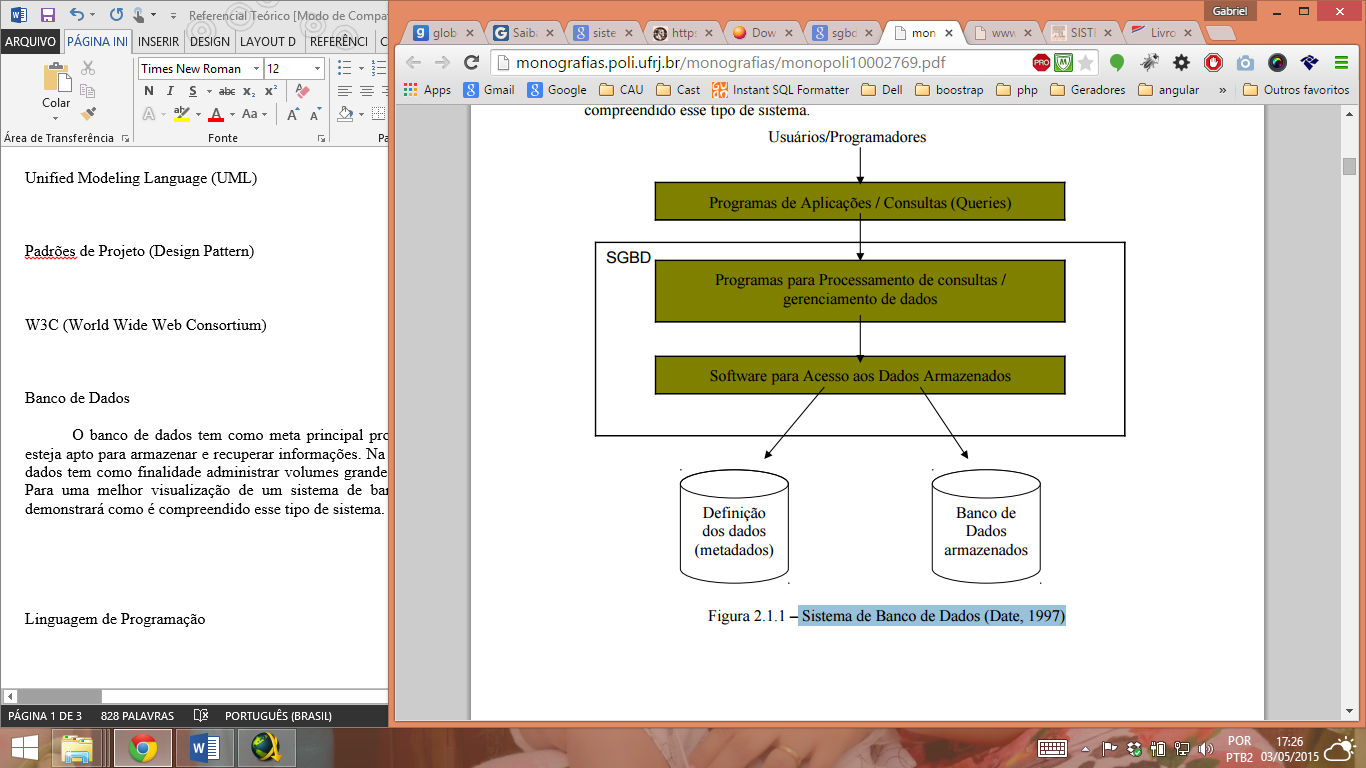
É uma língua importante, pois é o idioma do navegador da web. Sua associação com o navegador faz com que seja uma das linguagens de programação populares do mundo. A API do navegador, *o Document Object Model* (DOM) é bastante horrível, e *Javascript* é injustamente acusada. (CROCKFORD, 2009, p.6)

Javascript é capaz de definir, alterar e controlar de forma dinâmica a apresentação de um documento HTML, como os aspectos relacionados à cor de fundo, de textos e de links, ou mesmo interferir no posicionamento dos elementos HTML de um documento. (SILVA 2011, p. 24)

## Banco de Dados

O banco de dados tem como meta principal prover um ambiente eficiente que esteja apto para armazenar e recuperar informações. Na verdade, o sistema de banco de dados tem como finalidade administrar volumes grandes de informações (DATE, 1997). Para uma melhor visualização de um sistema de banco de dados, a Figura 8 demonstrará como é compreendido esse tipo de sistema.

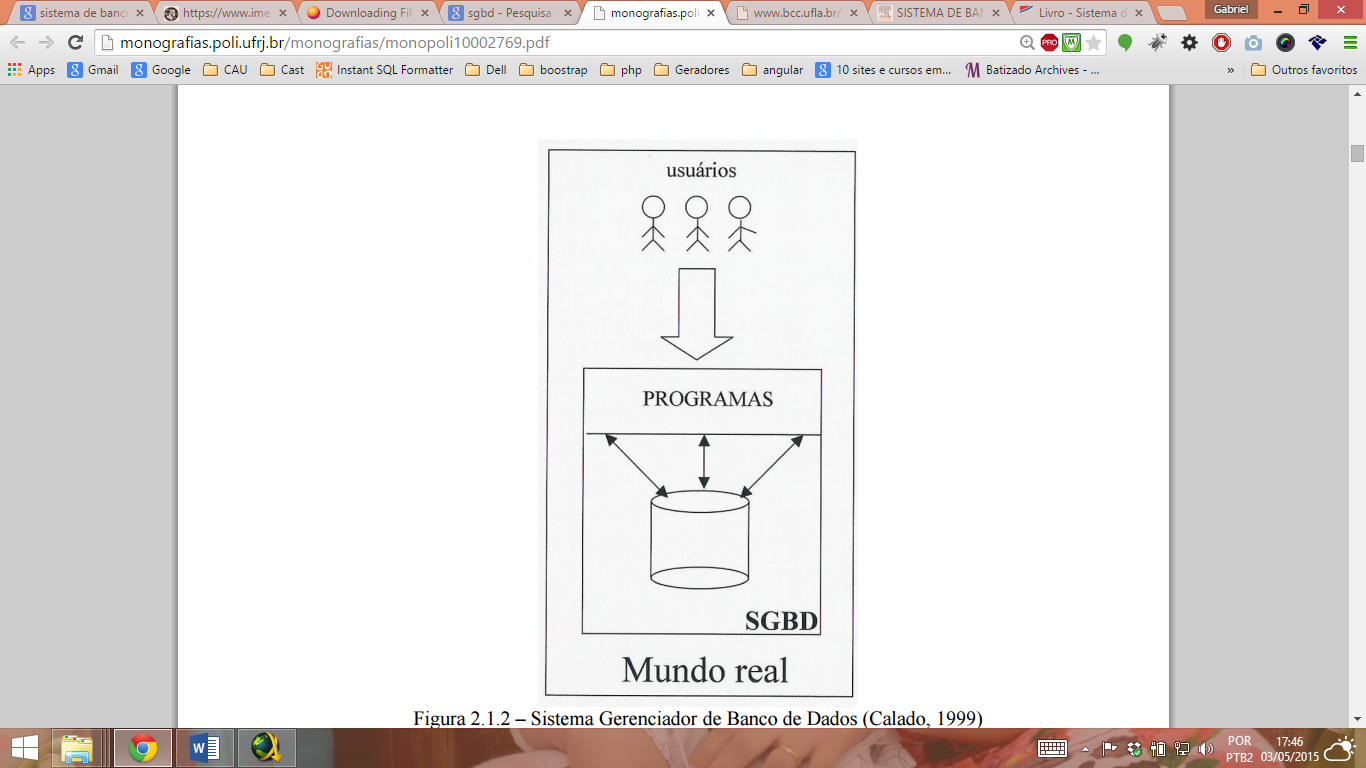
Figura 8 – Sistema de banco de dados



Fonte: DATE (1997)

Para Calado (1999), dentre as diversas vantagens trazidas pelo uso do banco de dados, “salienta-se o fator dos dados armazenados poderem ficar em um único local; os dados poderem ser compartilhados pelas aplicações, além de haver uma independência dos dados e haver uma maior flexibilidade de acesso.” Isso pode ser observado na figura 9.

Figura 9 - Sistema Gerenciador de Banco de Dados



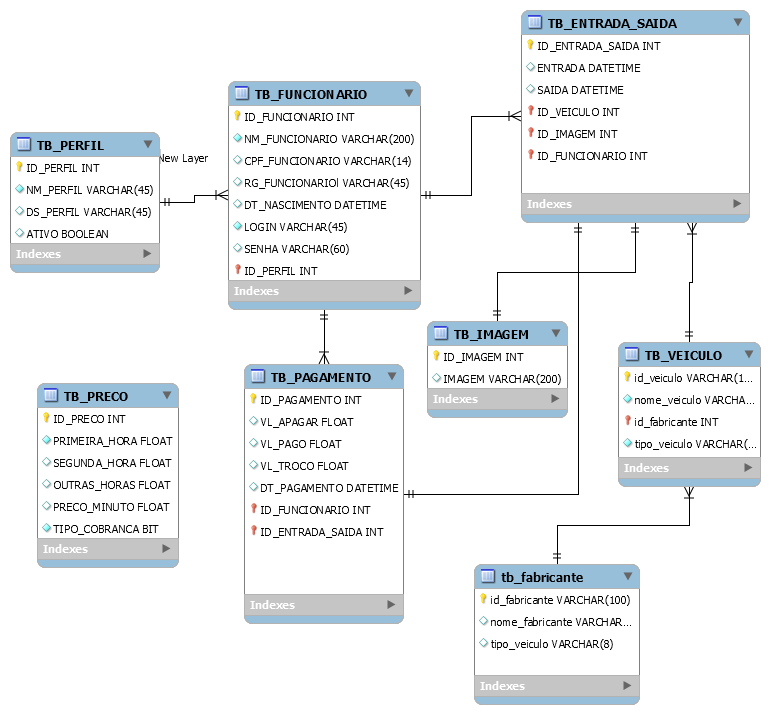
Fonte: Calado (1999)

De acordo com Silberschats (1999), os sistemas gerenciadores de banco de dados, embora ofereçam as vantagens suportadas, precisam ser escolhidos, levando-se conta a sua adequação com plataforma computacional.

O sistema gerenciador de banco de dados objetiva gerenciar o acesso e a devida manutenção dos dados armazenados em um banco de dados (DATE,1997).

As funções encontradas no SGBD são: Métodos de acesso DDL (*Data Definition Language*); DML (*Data Manipulation Language*) e consultas; Restrições de integridade (RIs). Segurança; Autorização de acesso; Recuperação de falhas (*recovery*) e Independência dos dados (DATE, 1997).

De acordo com Soares (2001), a “melhor forma de armazenar e gerenciar dados é através de um gerenciador como o MySQL.” Antigamente, poucas empresas utilizavam gerenciadores por causa do alto custo da despesa, mas isso mudou com a chegada do MySQL por ser uma ferramenta *free* (gratuita). Outro fato é que o MySQL é uma ferramenta muito fácil de ser utilizada e gerenciada e, ainda, disponibiliza de alguns comandos extras para manipulação dos dados.

Figura 10 – Exemplo Tabelas Banco de Dados  
  
fonte: Autor

Conforme demonstrado na figura 10, o banco de dados (SGBD) do sistema STACIONE é bastante simples, facilitando a manutenção e entendimento.

Conforme Date (2003, p. 37), o sistema de gerenciamento de banco de dados (SGBD) é o software que trata de todo acesso ao banco de dados.

O SGBD funciona da seguinte forma: um usuário solicita um pedido de acesso utilizando uma sub-linguagem de dados, esta que na maioria das vezes é o SQL, em seguida o SGBD recebe o pedido e realiza a análise, e então verifica o esquema externo para esse usuário, o mapeamento interno/externo conceitual e a definição do banco de dados armazenado, desta forma o SGBD executa as operações necessárias sobre o banco de dados armazenado. (DATE, 2003)

## 2.6.1 MySQL

## 

O banco de dados MySQL foi criado em 1995, devido à grande lentidão do mSQL (Mini SQL) deu-se origem ao MySQL, originalmente com o intuito de usar o API do mSQL, utilizando rotinas de baixo nível (ISAM), porém atualmente possui versões proprietárias e de código-aberto. (JOBSTRAIBIZER, 2009)

Para Jobstraibizer (2009, p. 14), os principais recursos são:

1. Suporte a *stored* procedures, triggers e *functions* a partir da versão 5x;
2. Implementação de cursores e *views*;
3. Checagem de consistência de erros;
4. Backups on-line em tempo real;
5. *Table space*;
6. Controle de transações;
7. Controle de acesso por domínio;
8. Conexão segura com SSL;
9. Consistência de tipos;

Segundo Jobstraibizer (2009, p. 14), o MySQL é o banco de dados mundialmente mais utilizado para sistemas em PHP. Sua conexão com o PHP pode ser realizada pela extensão .mysql.

No presente projeto, o SGBD utilizado é o MySQL 1.8.3 por ser o banco mais utilizado para sistemas em PHP e o mesmo realiza o armazenamento dos dados pessoais tanto do cliente quanto do funcionário da empresa e gerencia os dados das locações realizadas e dos carros disponíveis para locações.

## Linguagem de Programação

Segundo Sebesta (2003, p.20), programação de sistemas e todas as ferramentas de suporte à programação de um computador são coletivamente conhecidos como seu software básico que é usado continuamente e, portanto, deve ter eficiência na execução.

Conforme Sebesta (2003, p.16), existe alguns motivos e benefícios para estudar os conceitos de linguagens de programação, que são eles:

1. Aumento da capacidade de expressar ideias
2. Maior embasamento para a escolha de linguagens apropriadas.
3. Capacidade aumentada para aprender novas linguagens.
4. Entender melhor a importância da implementação.
5. Aumento da capacidade de projetar novas linguagens.
6. Avanço global da computação.

Uma das grandes pioneiras na criação de linguagens de programação foi a IBM, nas décadas de 60 e 70, que desenvolveram seus próprios códigos de alto nível para serem executados em seus equipamentos, tanto em máquinas de grande porte, quanto em microcomputadores para uso pessoais. (SEBESTA, 2003).

O sistema operacional e todas as ferramentas de suporte à  
programação de um sistema de computação são coletivamente  
conhecidos como seu software de Sistema. Software de sistema  
são aplicativos usados quase continuamente e, dessa forma, devem  
ser eficientes e tais aplicativos devem ter recursos de baixo nível  
que permitam aos aplicativos de software se comunicarem com  
dispositivos externos a serem escritos. (SEBESTA, 2010, p.25)

Necessita-se avaliar alguns critérios e características para realizar a escolha da linguagem que melhor atende em um desenvolvimento de algum sistema ou aplicação, para Sebesta (2003, p.22) deve-se avaliar principalmente; legibilidade, facilidade de escrita e confiabilidade.

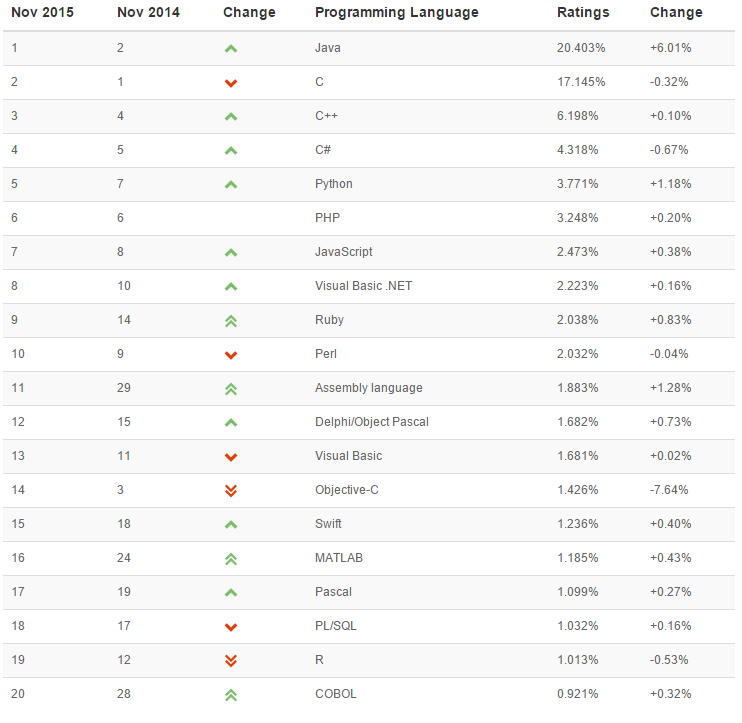
O objetivo do desenvolvimento de um programa, aplicação ou sistema de modo geral é a comunicação com o computador para se obterem as informações desejadas, todos os programas têm algo em comum: são elaborados pelos programadores em uma linguagem  
sob determinadas regras e sintaxe, ou seja, o “código-fonte” do programa, e posteriormente este código tem que ser traduzido para um código que só a CPU interpreta, que é a linguagem da máquina. Por este motivo, as linguagens de programação são fundamentais para que a máquina possa tratar os dados de acordo com a instruções estabelecidas e apresentar as informações desejadas. (LEITE, 2006).

O *Personal Home Page* - PHP foi criado em 1994 por uma única pessoa, Rasmus Lerdorf, e posteriormente foi adotado por outras pessoas e reescrito três vezes para proporcionar o amplo aperfeiçoamento da linguagem. O PHP é uma linguagem de criação de scripts ao lado do servidor que é projetado especificamente para a Web e é um produto de código-fonte aberto, ou seja, é possível utilizá-lo, alterá-lo e redistribuí-lo sem custo. (WELLING THOMSON, 2005)

A orientação a objetos trouxe o PHP para o rol de grandes e  
poderosas linguagens como o JAVA, o que tornou o PHP muito  
mais popular, e consequentemente muito mais utilizado, sendo a  
linguagem adotada até mesmo por empresas multinacionais.  
(JOBSTRAIBIZER, 2009, p.7)

No quadro 1 é apresentado o quadro de TIOBE, onde constam as 20 linguagens de programações mais utilizadas.

Quadro 1 – Quadro TIOBE



Fonte: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Segundo Welling e Thomson (2005, p. 27), alguns dos principais concorrentes do PHP são Perl, Microsoft ASP.NET, JavaServer Pages (JSP) e ColdFusion. E algumas das vantagens do PHP são: Alto desempenho, interfaces para muitos sistemas diferentes de banco de dados, bibliotecas integradas para muitas tarefas comuns da Web, baixo custo, facilidade de aprender e utilizar, Suporte orientado a objetos, portabilidade, disponibilidade de código-fonte e disponibilidade de suporte.

O sistema é desenvolvido em PHP 5.6 por ser uma linguagem fácil de programar e uma das mais utilizadas conforme informado no quadro 1, onde o PHP se encontra entre as seis linguagens mais utilizadas, devido a isso o projeto é em PHP por possibilitar desenvolver sistemas simples e que atende todas as necessidades.

## Servidor *Web*

Servidor Web utiliza o protocolo *HyperText Transfer Protocol Secure* (HTTP) para se comunicar com o cliente, este que permite o tráfego de dados/informações desejadas através de uma conexão TCP (*Transmission Control Protocol*). O HTTP funciona da seguinte forma: um cliente envia um pedido, em forma de mensagem e o servidor envia uma reposta também em forma de mensagem, e então as mensagens de pedido e resposta transportam dados que se formam um documento. Os comandos do cliente e servidor são inseridos numa mensagem de solicitação. (FOROUZAN, 2006).

Servidor web é a ferramenta responsável pela distribuição de documentos  
HTML através do protocolo http, este é essencial e pode determinar o  
sucesso de um sistema na internet. Um servidor WEB é um software que  
controla a recuperação e disponibilização de arquivos no formato de  
hypertexto em um sistema de arquivos. (SOUZA, 2004)

De acordo com o levantamento feito em junho de 2009 pela Netcraft (http://www.netcraftcom/survey/), aproximadamente 50% dos hospedeiros Web em  
33 operação utiliza o servidor Apache e em segundo lugar encontra-se os servidores Microsoft (Microsoft-IIS e Microsoft-PWS) em 25% dos sites e os demais utilizam Iplanet-Enterprise, Netscape-Enterprise, dentre outros. (BRUNO, 2010).

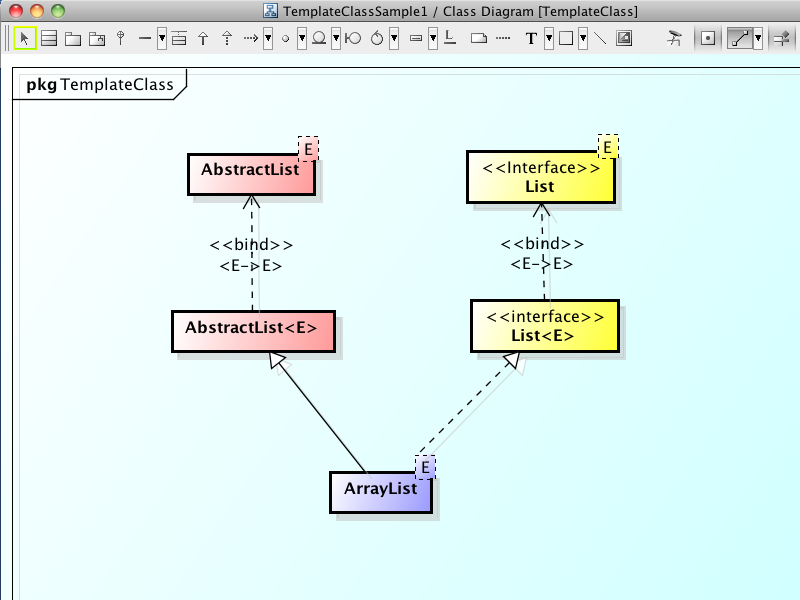
No sistema STACIONE é necessário um servidor WEB para ser armazenado devido ao sistema ser em PHP, ressaltando que não se trata de um sistema que pode ser acessado em qualquer rede, apenas da rede interna (intranet).

## Ferramentas

Para o desenvolvimento do sistema STACIONE são utilizadas ferramentas de apoio, tantas ferramentas de modelagem quanto ferramentas de linguagens, que facilitam o desenvolvimento e implantação do sistema, algumas dessas ferramentas estão citadas nos itens abaixo:

### Astah Community

Astah, uma ferramenta *CASE* de criação de diagramas UML, além de outros diagramas, tais como diagrama de entidade-relacionamento, diagrama de fluxo de dados e outras funcionalidades úteis à fase de especificação e projeto de um sistema.

Figura 11– Programa *Astah Community* 

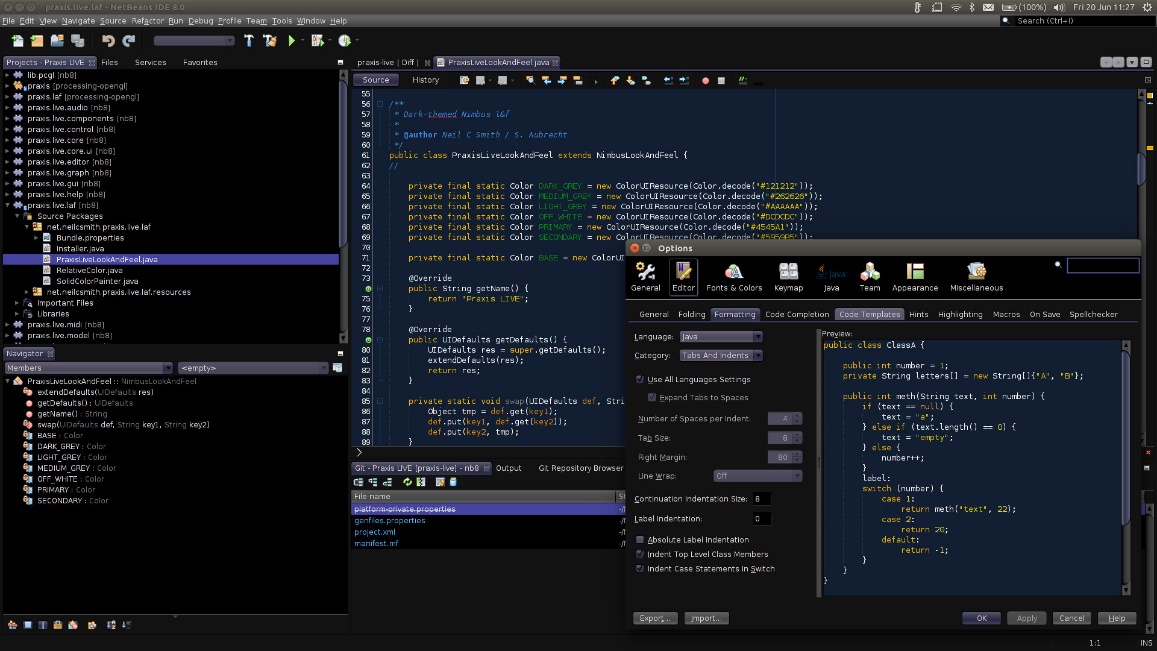
Fonte: http://astah.net/editions/community

### NetBeans Integrated Development Environment (IDE)

O NetBeans é um projeto open source de sucesso com uma grande base de utilizadores, uma crescente comunidade, perto de 100 parceiros mundiais. A Sun  
Microsystems fundou o projeto NetBeans em junho de 2000 e continua a ser o seu principal patrocinador.

O NetBeans IDE é um ambiente de desenvolvimento - uma ferramenta para  
programadores, que permite escrever, compilar, depurar e instalar  
programas. O IDE é completamente escrito em Java, mas pode suportar  
qualquer linguagem de programação. Existe também um grande número de  
módulos para extender as funcionalidades do IDE NetBeans. O NetBeans  
IDE é um produto livre, sem restrições à sua forma de utilização.  
(NETBEANS, 2014 )

Figura 12 - Página principal do *NetBeans*



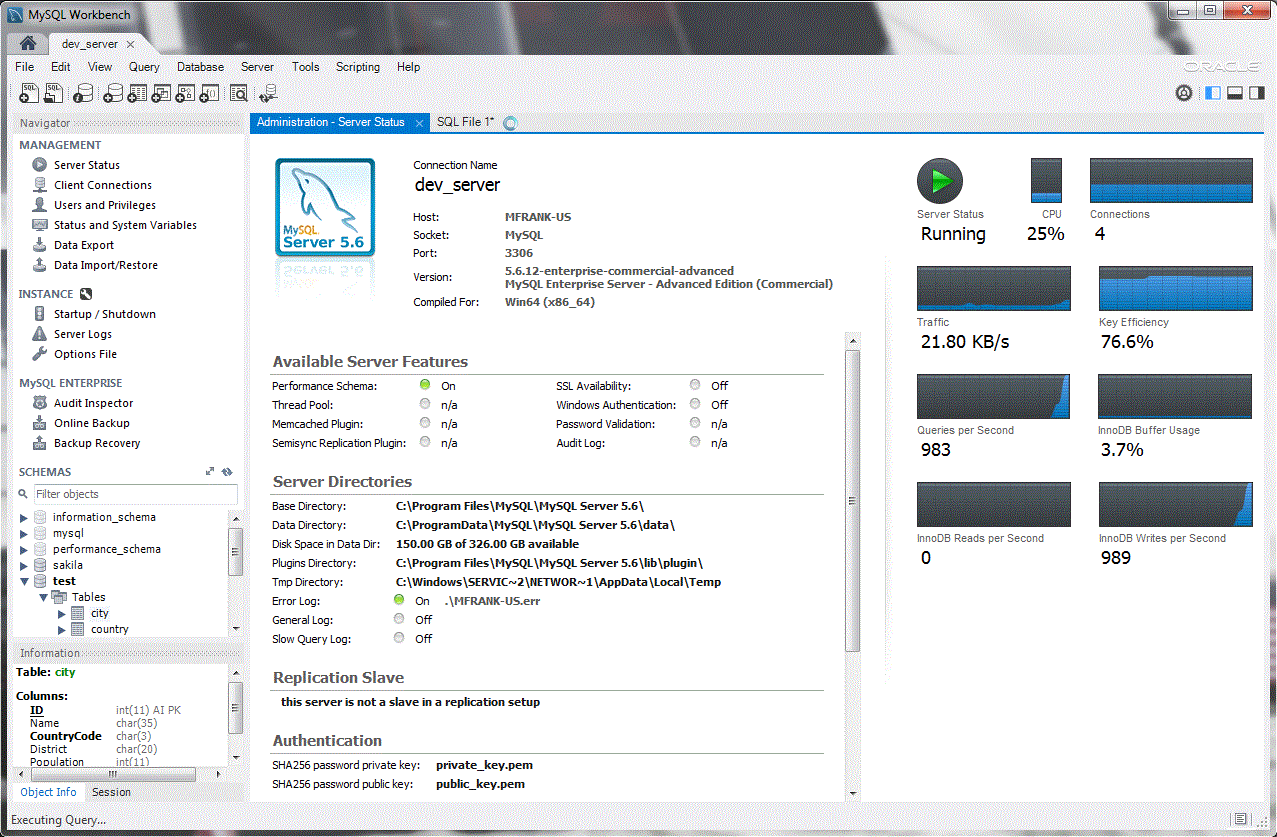
Fonte: http://netbeans.org

Conforme demonstra a figura 11, a interface do *NetBeans* no tema dark é simples e intuitiva.

### MySQL Workbench

O MySQL Workbench é a ferramenta oficial de manipulação de bases de dados no MySQL, que traz um editor visual para diagramas de entidade relacionamento – MER, assim como mecanismos capazes de obter diagramas a partir de um banco já existente, a engenharia reversa. Além de auxiliar nas fases de modelagem e desenvolvimento, a ferramenta também permite administrar o servidor, executando para isto funções como importação, exportação e gerenciamento das contas de usuário. Esses recursos ajudam tanto aos administradores quanto os projetistas. (MySQL, 2014)

Figura 13 – Programa *MySQL Workbench*



Fonte: https://dev.mysql.com/downloads/workbench/

A utilização do MySQL Workbench no projeto é necessária para a execução do modelo de entidade relacional – MER, onde foi definido todas as tabelas do sistema e quais são os seus respectivos relacionamentos, conforme mostra a figura 12.

### WAMP

Muitas pessoas sabem por experiência própria que não é fácil instalar um servidor web Apache e torna-se mais difícil se você quiser acrescentar MySQL, PHP e Perl.O objetivo do WAMP é construir uma distribuição fácil de instalar para desenvolvedores entrarem no mundo do Apache. Para torná-lo conveniente para os desenvolvedores, o WAMP é configurado com todos os recursos ativados. No caso de uso comercial, por favor dê uma olhada nas licenças de produtos; do ponto de vista do WAMPP, o uso comercial também é gratuito. Há atualmente distribuições para Windows, Linux e OS X.

Figura 14 - Programa do WAMPP



Fonte: <http://www.wampserver.com/en/>

A ferramenta WAMP neste projeto foi bastante útil, pois além da facilidade de instalação, o manuseio da ferramenta é simples e de fácil entendimento.

## Requisitos de Segurança de Software

Um sistema só poderá ser considerado capaz de manter informações de forma segura se elas não puderem ser interceptadas; se interceptadas, não puderem ser entendidas e se não puderem ser alteradas ou falsificadas.

A fim de manter os dados de forma segura, os principais objetivos da proteção dos dados são:

Confiabilidade: Evitar a aquisição de informações não autorizada.

Integridade: Evitar a modificação de dados não autorizada.

Disponibilidade: Evitar que dados restritos sejam disponibilizados para pessoas não autorizadas.

Responsabilidade: Estar pronto para identificar a pessoa responsável por cada transação.

Aliada a segurança, a confiabilidade dos dados tem um papel bastante significativo no contexto de sistemas de informação, já que um sistema só poderá ser considerado confiável se for reconhecido como seguro.

### Mecanismos de segurança da Linguagem de Programação e SGDB

Os mecanismos que podem ser usados incluem autenticação do usuário pelo Gerenciador de Aplicação em cooperação com o SGBD, controle do fluxo da navegação pelo Gerenciador de Aplicação com a definição do ponto de início da aplicação e todo o caminho que o usuário deve seguir, página por página, além do controle dos privilégios dos usuários.

### Mecanismos de Segurança e Privacidade de Dados.

A maior parte dos mecanismos de segurança baseia-se em criptografia de chave pública e assinatura confiável e uma implementação dos protocolos SSL e TLS.

Os próximos tópicos abordarão os mecanismos mais utilizados hoje para prover a segurança das informações na Web.

### Encriptação

O processo de criptografia é basicamente feito por algoritmos que fazem o em baralhamento dos bits desses dados a partir de uma determinada chave ou par de chaves, dependendo do sistema criptográfico escolhido.

Os algoritmos de encriptação disponíveis hoje são eficientes e serão bons o suficiente por vários anos, isso torna o mecanismo confiável e recomendado para muitos sistemas que precisam de um sistema se segurança eficaz.

### Protocolos de criptografia

Os principais protocolos utilizados hoje a fim de prover a segurança dos dados na Web são o SSL e o TLS. Ambos utilizados para a comunicação segura em transferências SMTP, HTTP e em outros tipos de transações que envolvam dados de valor.

Ambos, SSL e TLS, são basicamente o mesmo protocolo, porém o TLS contém melhorias em relação ao seu antecessor, o SSL. Por conta disso, o protocolo SSL 3.0 também é conhecido como SSL3, e o TLS 1.0 como TLS1 ou ainda SSL3.1.

Por serem bastante semelhantes, o próximo tópico detalhará apenas o funcionamento do TLS, já que o SSL pode ser analogamente compreendido.

### TLS

Os principais objetivos do TLS são, em ordem de prioridade:

* Segurança com criptografia: O TLS deve ser usado para estabelecer uma conexão segura entre duas partes.
* Interoperabilidade: Programadores independentes devem conseguir desenvolver aplicações utilizando TLS que possam trocar parâmetros criptográficos sem um conhecer o código do outro.
* Extensibilidade: O TLS busca o fornecimento de uma estrutura (framework), em que novos métodos de criptografia simétrica e assimétrica podem ser adicionados, sem a necessidade da implementação de uma nova biblioteca de segurança.
* Eficiência Relativa: Operações de criptografia, principalmente de chave pública, exigem um alto processamento. Sendo assim, o protocolo TLS incorporou um mecanismo de armazenamento para evitar que toda conexão ao ser estabelecida não precise processar operações de criptografia. Com isso, reduz-se também a atividade da rede.

# METODOLOGIA

Do ponto de vista da sua natureza o presente trabalho é uma pesquisa aplicada que objetiva gerar conhecimentos para aplicação prática e dirigida à solução de problemas específicos.

Do ponto de vista da forma de abordagem do problema trata-se de uma pesquisa qualitativa e do ponto de vista de seus objetivos é uma pesquisa exploratória.

Do ponto de vista dos procedimentos técnicos trata-se de um estudo de caso, pois, envolve o estudo que se permita o seu amplo e detalhado conhecimento.

Segundo Ponte (2006) o estudo de caso é:

É uma investigação que se assume como particularística, isto é, que se debruça deliberadamente sobre uma situação específica que se supõe ser única ou especial, pelo menos em certos aspectos, procurando descobrir a que há nela de mais essencial e característico e, desse modo, contribuir para a compreensão global de um certo fenómeno de interesse.” (Ponte, 2006)

Este projeto é voltado a um artigo científico onde relata informações e resultados de pesquisas, permitindo que os contextos existentes sejam interpretados e apresentados de diferentes formas e por autores diversos.

## Concepção

O STACIONE tem o objetivo de informatizar e agilizar os processos feitos manualmente pela empresa responsável por gerir o referido programa.

### Modelagem de Negócio

### Requisitos

O sistema gerencia um estacionamento, tendo como objetivo o controle de fluxo de entrada e saída de veículos, controle financeiro e geração de relatórios estatísticos auxiliando no controle dos fluxos de veículos que fazem uso do estacionamento.

### Entrevista

Segundo requisitos levantados pela Entrevista realizada com a Empresa Estacionamentos LTDA, o estacionamento cobra um valor fixo a cada meia hora. Após três horas de permanência, valor da hora cheia cai 25%. Foi informado que o tempo de tolerância é de vinte minutos. Os valores são estipulados da seguinte forma: Até 15 minutos, gratuito, após esse tempo, cobra-se o valor de 30 minutos. Fração de 30 minutos R$ 1,00.

A entrevista realizada encontra-se no Apêndice A deste documento.

### Diagrama de Caso de Uso

O Diagrama de Casos de Uso descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. As especificações de Caso de Uso do STACIONE encontram-se no Apêndice C.

### Requisitos Funcionais

O sistema abrange as áreas responsáveis pelo funcionamento do estacionamento como a Gerência e o guichê de atendimento.

### Requisitos não funcionais

Os requisitos não funcionais são aqueles que apesar de não descreverem as funções do produto, fornecem as características restantes para se alcançar o objetivo do sistema. Eles delimitam o alcance do produto e fornecem as especificações necessárias para o seu funcionamento.

### Gerenciamento do Projeto

Gerenciamento do Projeto é a aplicação de conhecimentos, habilidades e técnicas para a execução de projetos de forma efetiva e eficaz.

### Análise de Riscos e plano de contingência



Conceitualmente, a Gerência de Risco é um processo composto de dois subprocesso, um na fase de planejamento e outra na fase de execução. Na fase de identificação de risco, o gerente do projeto e sua equipe procuram identificar os fatores de risco a que o projeto está exposto. Na fase de análise de impacto e probabilidade, eles avaliam o impacto de cada fator de risco sobre o projeto, assim como a probabilidade de cada um deles venha a prejudicar o projeto. Na fase de elaboração de planos de contenção e contingência, ele procura estabelecer cursos de ação que diminuam a probabilidade de cada fator de risco vir afetar o projeto (planos de contenção) e, caso a concretização do risco se torne inevitável, cursos de ação que minimizam seu impacto (planos de contingência). (ALENCAR, 2006, p. 37)

As análises de riscos e os planos de contingência de cada risco são apresentados no quadro 1

Quadro 2 – Quadro de Análise de Riscos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nr. | Risco | Categoria | Probabilidade X Impacto | Mitigação | Contingência | Responsável |
| 1 | A falta de conhecimento específico de um ou mais membros da equipe pode acarretar em atraso no desenvolvimento do Projeto. | Escopo | Médio | 1 - Fazer curso de programação na linguagem do projeto;  2 - Iniciar o projeto antes da data definida;  Buscar ajuda especializada para realizar o desenvolvimento do projeto. | 1 - Continuar o desenvolvimento do projeto e apresentar no próximo semestre;  2 - Mudar o projeto para monografia;  3 - Aceitar o risco. |  |
| 2 | Prazo curto para a entrega do projeto. | Organizacional | Alto | 1 - Esforço da equipe na fase de elaboração para evitar problemas futuros, como requisitos incompletos ou mal especificados. | 1 - Realização dos principais pontos do projeto, pois, caso o projeto não seja finalizado, o restante se tornará trabalho futuro;    - Diminuir o período de testes no cronograma do projeto. |  |
| 3 | Falhas na especificação dos requisitos de software. | Tecnológico | Alto | 1 - Repassar o entendimento da equipe para o cliente, para o esclarecimento de dúvidas. | 1 - Elaboração de protótipos; |  |
| 4 | Falta de equipamentos para teste e desenvolvimento | Tecnológico  Financeiro | Médio | 1 - Utilizar hardware (pessoal) reserva, conforme a necessidade. | 1 - Adquirir hardware mais potente. |  |
| 5 | Indisponibilidade do cliente para o levantamento de requisitos | Organizacional | Alta | 1 - Durante o período do levantamento de requisitos algumas reuniões foram substituídas por técnicas como a utilização de questionários, enviados através de e-mail. | 1 - Marcar reuniões com antecedência de 5 dias;  2 - Utilização de mais questionários;  3 - Utilização de bate-papos. |  |
| 6 | Retrabalho em tarefas do projeto | Escopo | Média | 1 – Treinamento da equipe nas ferramentas utilizadas | 1 – Incluir Consultor interno junto a equipe |  |
| 7 | Projeto não atende os requisitos definidos | Escopo | Média |  |  |  |
| 8 | Análise de riscos incompleta | Escopo | Média | 1 – Aplicar medidas imediatas de curto prazo visando a correção do Projeto | 1 – Determinar as causas da falha de planejamento |  |

Fonte: Autor

### Plano de Cronograma do Projeto

O Cronograma do Projeto é utilizado para o gerenciamento e controle dos projetos, permitindo de forma rápida a visualização do seu andamento. Em geral o cronograma refere-se à discriminação das etapas do trabalho com seus respectivos prazos.

No quadro 2, é apresentando as descrições das tarefas.

Quadro 3 – Descrição de Tarefas

| **Descrição de tarefas** | | | |
| --- | --- | --- | --- |
| **Cód.** | **Descrição** | **Marco de Referência** | **Produto** |
| **T1** | Realização da Entrevista | Questionário respondido |  |
| **T2** | Análise dos Casos de uso para identificação das funções associadas a cada subsistema | Descrição dos subsistemas |  |
| **T3** | Especificação abstrata dos subsistemas |  | Diagrama de classe |
| **T4** | Modelo Comportamental do subsistema | Diagrama de sequencia |  |
| **T5** | Projeto de Banco de Dados - Modelo Lógico |  | Tabelas de Representação |
| **T6** | Projeto de Banco de Dados - Modelo Físico |  | Tabelas de Representação |
| **T7** | Projeto de Banco de Dados – Mapeamento de Classes |  | Tabelas de Representação |
| **T8** | Projeto de Interfaces |  | Projeto de Software |
| **T9** | Codificação do Sistema  Acesso a Web | Listagem do código |  |
| **T10** | Teste de Unidade | Avaliação dos Testes de Unidades |  |
| **T11** | Testes de validação |  | Guia de manutenção  Software - programas e documentação |

Fonte: Autor

No quadro 3, apresenta o cronograma das atividades e os respectivos autores de cada atividade.

Quadro 4 – Cronograma

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Cronograma** | | | | | | | | | | | | |
| **Responsável** | **Fev** | **Mar** | **Abr** | **Mai** | **Jun** | **Jul** | **Ago** | **Set** | **Out** | **Nov** | **Dez** |  |
| Wagner de Araújo Alves | T1 |  |  |  |  |  |  |  |  |  |  |  |
| Todos |  | T2 |  |  |  |  |  |  |  |  |  |  |
| Gabriel Cabral de Almeida |  |  | T3 |  |  |  |  |  |  |  |  |  |
| Todos |  |  |  | T4 |  |  |  |  |  |  |  |  |
| Gabriel Cabral de Almeida |  |  |  | T5 |  |  |  |  |  |  |  |  |
| Gabriel Cabral de Almeida |  |  |  |  | T6 |  |  |  |  |  |  |  |
| Gabriel Cabral de Almeida |  |  |  |  |  | T7 |  |  |  |  |  |  |
| Wagner de Araújo Alves |  |  |  |  |  |  | T8 |  |  |  |  |  |
| Todos |  |  |  |  |  |  |  | T9 |  |  |  |  |
| Wagner de Araújo Alves |  |  |  |  |  |  |  |  | T10 |  |  |  |
| Todos |  |  |  |  |  |  |  |  |  | T11 |  |  |

Fonte: Autor

## Elaboração

### Análise de requisitos

Análises de requisitos são características funcionais e não funcionais que o sistema precisa apresentar. O objetivo deste processo é definir as características do sistema conforme observadas pelo cliente, auxiliando o desenvolvimento na direção correta. (MARTINS, 2007).

### Entrevista

A entrevista é uma técnica extremamente útil para obter informações que estão “armazenadas” na memória do cliente, ou seja, como que o cliente deseja que o software em questão seja desenvolvido. (REZENDE, 2005)

A entrevista é disponibilizada no apêndice A.

### Requisitos funcionais

Requisitos funcionais são aqueles que definem o comportamento do sistema, extraídos através dos Casos de Uso, que documentam as entradas, os processos e as saídas geradas. (MARTINS, 2007)

### Requisitos não funcionais

Segundo Martins (2007, p. 208) requisitos não funcionais são compostos por características não necessárias associadas ao comportamento do software. Alguns dos requisitos não funcionais do STACIONE são:

1. Usabilidade: O sistema tem uma interface fácil e intuitiva.
2. Performance: O sistema possui a melhor performance possível, o preenchimento e processamento de ações não ultrapassam 30 segundos.
3. Segurança: O sistema tem criptografias para melhor segurança possível.

### Diagrama de caso de uso

O Diagrama de Caso de Uso contém elementos modelados para um sistema, ou seja, os atores que irão utiliza-lo e os casos de uso, e mostra os diferentes relacionamentos como também generalizações, associações e dependência entre esses elementos. (REZENDE, 2005)

O diagrama de Caso de Uso é representado no apêndice.

## Projeto de Bando de Dados

### Modelo de entidade relacional

Conforme Leite (2008, p. 6) o Modelo Entidade Relacionamento - MER representa os dados agrupados em tabelas, e como se relacionam. O MER mostra o mapeamento entre as diversas tabelas que compõem o banco de dados, e em análise de sistemas essas tabelas são chamadas de relações ou instâncias do banco de dados, uma vez que podem existir fisicamente ou apenas logicamente (LEITE, 2008).

O Modelo Entidade Relacionamento se encontra no apêndice H.

### Dicionário de dados

Dicionário de dados é um documento que complementa o MER, documento este que deve conter maiores detalhes a respeito das tabelas e seus relacionamentos. Além de listar todas as colunas da tabela, o documento deve fornecer também uma descrição do significado desta coluna, os valores possíveis, a quantidade típica de valores armazenados, dentre outras informações do atributo. (PICHILIANI, 2011)

O dicionário de dados é disponibilizado no apêndice Diagrama de Sequência

## Projeto de Analise e Design

Esta seção apresenta os diagramas utilizados para o desenvolvimento do sistema, com uma breve descrição, referenciados para visualização em apêndice.

### Diagrama de Atividades

Os Diagramas de atividades pretendem descrever um comportamento paralelo ou apresentar como interagem comportamentos em vários casos de uso. (REZENDE, 2005)

O diagrama de atividades é disponibilizado no apêndice .

### Diagrama de implantação

Os Diagramas de sequência mostram a iniciação e a execução de uma sequência de funções, ou seja, o diagrama de sequência mostra a ordem em que os métodos de vários objetos são executados. (BRAUDE, 2004)

O diagrama de sequência é disponibilizado no apêndice .

### Diagrama de classe

O diagrama de classes é utilizado para representar a estrutura estática do sistema, composta pelas classes de negócio, classes de interface com o usuário e com outros sistemas, composta pelas classes de negócio, interface com o usuário e com outros sistemas, as classes de controle, estas que são responsáveis pelo controle de transações. (MARTINS, 2007)

O diagrama de classe é disponibilizado no apêndice .

## Segurança

De acordo com Peixoto (2006, p. 37), O termo segurança pode ser designado como uma área do conhecimento que salvaguarda os ativos da informação, contra acessos indevidos, modificados não autorizados ou até mesmo sua não disponibilidade.

Para manter a integridade da informação e proteger os ativos de TI, é  
necessário implementar um processo de gestão de segurança. Esse processo  
inclui o estabelecimento e a manutenção de papéis, responsabilidades,  
políticas, padrões e procedimentos de segurança de TI. A gestão de  
segurança inclui o monitoramento, o teste periódico e a implementação de  
ações corretivas das deficiências ou dos incidentes de segurança. A gestão  
eficaz de segurança protege todos os ativos de TI e minimiza o impacto  
sobre os negócios de vulnerabilidades e incidentes de segurança. (ITGI,  
2007, p. 119)

### Mecanismos de Segurança da Linguagem de Programação e SGBD

O SQL *Injection* ocorre quando um usuário mal-intencionado insere instruções SQL (*Structured Query Language*) num campo de formulário, numa query *string*, ou em um campo oculto, como por exemplo, em um cookie. O ataque tem como objetivo obter informações sigilosas. (LOTAR, 2007)

Qualquer procedimento que trabalha com SQL deve ser submetido a uma revisão de todas as vulnerabilidades que nele pode se encontrar, pois o SQL Server tem como executar várias consultas que sintaticamente foram validadas para ele. E, ainda, mesmo após tudo estar parametrizado, os dados podem ser manipulados por crackers mais qualificados. (MAFRA, 2006).

Quando um cracker coloca um código malicioso em textos que são para serem armazenados em tabelas ou banco de dados, ele fica intacto até ser realizada a sua execução. Também, necessita-se tomar cuidado com as configurações dos usuários que estabelecem a conexão com o banco de dados. O correto seria que as permissões de acesso por esses usuários fossem restritamente limitadas as suas funções exercidas. Por exemplo, para a exibição de um relatório, a conexão com o banco de dados deve ser realizada somente pelos  
 usuários com permissão de leitura e acesso somente às tabelas necessárias para sua operação. (LOTAR, 2007).

Existem várias técnicas de se criar uma criptografia, cada uma com suas características e diferenças como, por exemplo: a criptografia de chave única, MD5 e criptografia de chave pública e chave privada. A criptografia de chave única utiliza a mesma chave tanto para codificar quanto para decodificar mensagens. Apesar de este método ser bastante eficiente em relação ao tempo de processamento, ou seja, o tempo gasto para codificar e decodificar mensagens tem como principal desvantagem a necessidade de utilização de um meio seguro para que a chave possa ser compartilhada entre pessoas ou entidades que desejem trocar informações criptografadas. (BURNETT, 2002)

O MD5 é uma técnica de criptografia unidirecional, um hash MD5 não pode ser transformado novamente no texto que lhe deu origem, normalmente essa criptografia é utilizada para guardar senhas. Para efetuar a comparação entre as senhas é efetuada uma comparação entre o novo hash MD5 da senha informada com o Hash MD5 salvo. A criptografia de chave pública utiliza duas chaves distintas, uma para codificar e outra para decodificar a mensagem. A chave pública pode ser divulgada livremente e a privada é mantida em segredo com seu proprietário. (BURNETT, 2002)

Controle de sessão: A sessão web é bem parecida com a sessão de um PC, que ao iniciar, temos que colocar usuário e senha, assim no STACIONE pode saber quem está usando o sistema e guarda os registros com segurança. No STACIONE, a sessão é importante quando se quer mais segurança na página.

### Mecanismos de Segurança e Privacidade de Dados

Mecanismos de segurança e privacidade de dados são formados pelos seguintes pilares básicos que podem ser definidos da seguinte maneira. (PEIXOTO, 2006): Confiabilidade: É a garantia que as informações transmitidas chegarão ao seu destino sem que se dissipem para outro lugar onde não deveria passar. Várias tecnologias como, por exemplo, criptografia e autenticações podem ser usadas, desde que mantenham a integridade das informações; (PEIXOTO, 2006)

Integridade: É a garantia que as informações não sofreram nenhuma modificação durante o trajeto entre a pessoa que enviou e a pessoa que recebeu a informação, garantindo assim a sua real veracidade após chegarem ao destino. (PEIXOTO, 2006)

Disponibilidade: De nada adianta possuir integridade e confidencialidade, se a informação nunca está disponível. Então, o grande desafio é manter essa estrutura de passagem de informações de forma confiável e integral sem que haja impossibilidade de captar as informações. (PEIXOTO, 2006)

## Teste e Implantação

O processo de testes é caracterizado pela execução das principais etapas da atividade de teste e dos seus desdobramentos (subetapas). (RIOS & MOREIRA, 2013, p. 27).

Processo de executar um programa ou sistema com a intenção de encontrar defeitos (teste negativo). (MYERS, 1979).

Qualquer atividade que a partir da avaliação de atributo ou capacidade de um programa ou sistema seja possível determinar se ele alcança os resultados desejados. (RIOS & MOREIRA, 2013, p. 30).

### Testes de Qualidade

Significa o grau em que um sistema, componente ou processo atinge as necessidades dos usuários. Qualidade é uma característica intrínseca e multifacetada de um produto. (BASILI, 1996).

### Testes de Stress

É um tipo de teste de confiabilidade destinado a avaliar como o sistema responde em condições anormais. O stress no sistema pode abranger cargas de trabalho extremas, memória insuficiente, hardware e serviços indisponíveis ou recursos compartilhados limitados. São executados para compreender como e em quais áreas o sistema é dividido, para que os planos de contingência e a manutenção de atualização possam ser planejados e evitados. (PEIXOTO, 2006)

O teste de stress está disponível no apêndice .

### Testes de Segurança

Qualquer sistema de computador que trabalho com informações sensíveis ou que cause ações que podem inadequadamente prejudicar (ou beneficiar) indivíduos, é um alvo para acesso impróprio ou ilegal. As invasões abrangem uma ampla gama de atividades: hackers que tentam invadir sistemas por diversão, funcionários desgostosos que tentam invadir por vingança, indivíduos desonestos que tentam invadir para obter ganhos pessoais ilícitos. (PRESSMAN, 2011)

O teste de segurança verifica se os mecanismos de proteção incorporados ao sistema vão de fato protegê-lo contra acesso indevido. A segurança de um sistema deve naturalmente ser testada, quanto à invulnerabilidade por um ataque frontal, mas também deve ser testado quanto à invulnerabilidade por ataques laterais ou pela retaguarda. (PRESSMAN, 2011)

Os testes de segurança estão disponíveis no apêndice .

## Implantação

A implantação é a fase do ciclo de vida de um software, no contexto de Sistema de Informação, que corresponde textualmente à passagem do software para a população.

Não basta disponibilizar novos recursos tecnológicos e de sistemas. As  
pessoas, os grupos e os diversos níveis gerenciais que compõem a força de  
trabalho da organização devem estar plenamente comprometidos com os  
resultados almejados, familiarizados com o processo de mudança proposto  
e motiva dos para a assimilação e o uso efetivo da nova tecnologia.  
Gerenciar mudanças, a partir da introdução de novas tecnologias, exige das  
organizações uma habilidade muitas vezes difícil de ser encontrada.  
(ALBANO, 2001, p. 10)

### Plano de Implantação

Após a percepção da necessidade de se implantar o sistema, a organização passa pelo processo de desenvolvimento e implantação deste, que tem toda uma sequência de processos a serem respeitados e seguidos para que se tenha o sucesso em sua implantação.

O desenvolvimento de sistemas é criar novos programas ou reestruturar e  
adaptar as mudanças ao programa já existentes. Para que se de inicio a um  
novo sistema, é necessário identificar os problemas a serem solucionados  
ou oportunidades a serem exploradas até a avaliação e possível refinamento  
da solução escolhida para que se for necessário haja a modificação.  
(STAIR, 1998)

No ato do desenvolvimento do sistema, é importante que todos os profissionais da empresa tomem conhecimento do sistema implantado para suas tomadas de decisões ou ainda podem ajudar no aprimoramento da ideia inicial.

Também afirma que independente do formato ou complexidade do  
software, a maioria das empresas utilizam um planejamento em cinco  
etapas: avaliação, análise, projeto, implementação e manutenção e revisão.  
As empresas utilizam essas etapas porque a medida que o projeto vai sendo  
construído, estipulasse prazos até que o sistema esteja instalado e aceito. E  
se houver grandes necessidades de mudanças no Sistema de Informação,  
além das manutenções e revisões, começa um novo projeto conforme as  
etapas mencionadas abaixo, pois se trata de etapas com pontos de  
verificação para saber se a etapa anterior foi bem aceita. (STAIR, 1998)

Mediante as etapas que serão elaboradas no processo de implantação, mencionaremos o objetivo de cada uma delas:

A etapa de avaliação do sistema: é a fase de identificação do problema a ser resolvido, e elaboração da lista de oportunidades para resolvê-lo.

A etapa de análise de sistemas: são confrontadas as soluções do problema anterior, e as condições que a organização possui para utilizá-las.

Dentre a etapa de projeto de sistemas: é que se escolhe um projeto que tenha condições de solucionar o problema existente e que seja viável para a organização.

No desenvolvimento da etapa de implantação de sistemas: é implantado o sistema, além de se realizar o treinamento dos funcionários e também a obtenção de hardware que serão utilizados no processo.

E por fim a etapa de manutenção e revisão de sistemas: nessa etapa se faz uma avaliação para se perceber se o sistema necessita de mudanças, se está atuando com grande eficácia e se tem condições de auxiliar a empresa no alcance de suas metas. Não satisfazendo esses requisitos às etapas são retroalimentadas ou se implanta um novo sistema, isso é realizado até que a empresa consiga resolver o problema mencionado.

Registram que o sistema de informações gerenciais deve fornecer  
informações básicas de que os gestores necessitam em suas tomadas de  
decisão. Assim, quanto maior for a sintonia entre a informação fornecida e  
as necessidades informativas dos gestores, melhores decisões poderão ser  
tomadas. Isto é, ao projetar um sistema de informações, faz-se necessário a  
alisar cuidados cuidadosamente o processo de decisão e o fluxo de  
informações existente. Esses dois fatores são essenciais e inseparáveis no  
desenho e arquitetura de um sistema de informações gerenciais (MASON,  
JÚNIOR, RICHARD, 1975).

Após o processo de implantação, tendo o sistema sido desenvolvido internamente ou desenvolvido por uma empresa terceirizada, a organização a partir desse sistema sofre algumas mudanças. As mudanças que são mais percebidas nos vários tipos de empresas são a mudança na estrutura organizacional e no número de funções nos cargos.

Com a implantação do sistema de informação as organizações passam por  
uma mudança na estrutura, apresentando uma estrutura horizontal, porque o  
sistema permite uma maior produtividade por parte dos gerentes, pois com  
ela o campo de supervisão do número de funcionários é ampliado,  
supervisando assim um maior número de funcionários e necessitando cada  
vez menos de gerentes especializados. (TURBAM, RAINER JÚNIOR E  
POTTER 2003).

### Manual de Implantação

Com todos os recursos que este arquivo do Sistema do Gerenciamento de estacionamento – STACIONE disponibiliza, é suficiente para implantação do sistema na organização.

### Treinamento de Usuários

O treinamento do usuário ocorrerá no mesmo mês após a contratação deste. A duração do treinamento será de no máximo duas horas, devido à facilidade de manuseio e atualização do sistema STACIONE.

# CONCLUSÃO

O projeto consegue proporcionar aos clientes que tornaram o atendimento e a gestão da empresa mais ágil e eficaz. A aplicação utiliza uma lógica simples e prática. Seu baixo custo e fácil manutenção também são pontos fortes.

O Sistema STACIONE foi desenvolvido visando à inovação e maior facilidade na administração e gestão da empresa. Tem como intenção facilitar o atendimento no que se trata a entrada e saída de veículos. Tendo isso como foco a facilidade dos usuários será algo natural.

O uso das tecnologias PHP, HTML, CSS e MySQL representaram muita importância no desenvolvimento do sistema, pois foram com estas que o sistema foi desenvolvido. A linguagem de modelagem UML também foi fundamental para o desenvolvimento deste projeto, pois permitiu a utilização de métodos utilizados anteriormente por alguns desenvolvedores e consequentemente obteve sucesso.

Com este projeto, foi identificada a possibilidade de comercializar o sistema de forma profissional. Observando a grande procura por serviços tecnológicos que facilitam no dia-a-dia de qualquer empresa, trazendo maior eficiência no atendimento, gestão e administração.

O projeto terá implantações futuras voltadas a atualização das funcionalidades do sistema STACIONE de acordo com as tendências tecnológicas e atendendo as particularidades de cada empresa, tendo como foco também a elaboração de projetos que atendem principalmente na administração, armazenamento de dados e gestão das organizações.

# REFERÊNCIAS

ALBANO, Valter. **Do Caos ao Equilíbrio Fiscal**. 1ª Edição. Local: São Paulo. Editora: Entrelinha, 2001.

ALENCAR, Antônio J./SCHMITZ, Eber A. Análise de Risco em Gerência de Projetos. 1ª Edição. Local: Rio de Janeiro, 2006.

ASTAH COMMUNITY: ASTAH: Mudar a forma como você projeta software e sistemas. Disponível em: <http://www.astah.net/ >. Acesso em: 28/11/2015, 01:30.

BRUNO, Odemir M./ ESTROZI, Leonardo F./ NETO, João E. S. B. Programando para a internet com PHP. 1ª Edição. Local: Rio de Janeiro. Brasport, 2010.

BRAUDE, Eric. Projeto de Software Da programação à arquitetura: uma abordagem baseada em Java. 1ª Edição. Local: São Paulo. Bookman, 2004.

CROCKFORD, Douglas. O melhor do JavaScript. 1ª Edição. Local: São Paulo. Editora Alta Books, 2008.

DATE, C. J. Introdução a Sistemas de BANCOS de DADOS. 8ª Edição. Local: Rio de Janeiro. Elsevier, 2003.

ENGHOLM JR, Hélio. Engenharia de Software na Prática. 1º Edição. - São Paulo: Editora Novatec, 2010.

FARINELLI, Fernanda. Conceitos básicos de programação orientada a objetos. Disponível em: <http://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/1662272077\_POO.pdf>. Acesso em: 10/10/2015, 03:45.

FOROUZAN, Behrouz A. Comunicação de Dados e Redes de Computadores. 3ª Edição. Local: Porto Alegre. Bookman, 2006.

GUEDES T.A., Gilleanes. UML 2 – Guia Prático. 2ª Edição. - São Paulo: Editora Novatec, 2011.

HEUSER, Carlos A. Projeto de Banco de Dados. 4ª Edição. Local: Instituto de Informática da UFRGS, 1998.

JOBSTRAIBIZER, Flávia. Guia Profissional: PHP. 1ª Edição. Local: São Paulo. *Digerati Books*. 2009.

LEITE, Mário. Técnicas de Programação: Uma abordagem moderna. 1ª Edição. Local: Rio de Janeiro. Brasport, 2006.

LEITE, Mário. Acessando Banco de Dados com ferramentas RAD: Aplicações em Delphi. 1ª Edição. Local: Rio de Janeiro. Brasport, 2008.

LOTAR, Alfredo. Como exibir informações com múltiplas instruções SQL. Disponível em: <http://www.linhadecodigo.com.br/> Acesso em: 05/08/2015, 13:28.

KRUCHTEN, Philippe. Introdução ao RUP *Rational Unified Process*. Rio de Janeiro: Ciência Moderna, 2003

MARCELO, Antonio. Apache: Configurando o servidor WEB para Linux. 3ª Edição. Local: Rio de Janeiro. Brasport, 2005.

MARTINS, José C. C. Técnicas Para Gerenciamento de Projetos de Software. 1ª Edição. Local: Rio de Janeiro. Brasport, 2007.

MARTINS, José Carlos Cordeiro. Gerenciando Projeto de Desenvolvimento de Software com PMI, RUP e UML – 5. ed.- Rio de Janeiro: Brasport, 2011.

NETBEANS, 2014. Site Oficial NetBeans. Disponível em: http://netbeans.org. Acesso em 08/11/2014, 20:12.

NIEDERAUER, Juliano. Desenvolvendo Websites com PHP. 2ª Edição. Local: São Paulo. Novatec, 2011.

PEIXOTO, Mário C. P. Engenharia Social e Segurança da Informação na Gestão Corporativa. Local: Rio de Janeiro: Brasport, 2006.

PRESSMAN, R. S. Engenharia de Software. 6ª Edição. São Paulo: McGraw-Hill, 2006.

PICHILIANI, Mauro C. Conversando sobre banco de dados. Coletânea de artigos sobre banco de dados publicados entre 2001 e 2011 no iMaster.

PRIKLADNICK, Rafael. & AUDY, Jorge. Desenvolvimento Distribuído de Software. 1ª Edição. Local: Rio de Janeiro. Elsevier, 2007.

REZENDE A., Denis. Engenharia de Software e Sistemas de Informação. 3ª Edição. Local: Rio de Janeiro. Brasport, 2005.

RIOS, Emerson. MOREIRA, Trayahú. Teste de Software. 3ª Edição. Local: Rio de Janeiro. Alta Books, 2013.

SEBESTA, Robert. Conceitos de Linguagens de Programação. 5ª Edição. Local: *University of Colorado*, Colorado Springs. Bookman, 2003.

SEBESTA, Robert. Conceitos de Linguagens de Programação. 9ª Edição. Local: São Paulo. Bookman, 2009.

SILVA, Maurício Samy CSS3 – Desenvolva aplicações web profissionais com o uso do CSS3. Local: São Paulo, Novatec Editora, 2010.

SOMERVILLE, I. Engenharia de software. 6° ed. Tradução Maurício de Andrade. São Paulo: Ed Addison-Wesley, 2003.

SOMMERVILLE, Ian. Engenharia de Software, Tradução Ivan Bosnic e Kalinka G. de O. Gonçalves. 9ª Edição. Local: São Paulo, Pearson, 2011.

SOUZA, Marcelo S. Análise do Servidor Web Apache em Clusters OpenMosix com Memória Compartilhada Distribuída. Universidade Católica do Salvador (UCSal). Disponível em: http://www.cebacad.net/documentos/artigos/oMosix\_rev\_final.pdf. Acesso em 02/10/2015, 21:00h.

MACHADO, Felipe Nery Rodrigues. Análise e Gestão de Requisitos de Software: São Paulo: Érica,2011.

BOENTE, A. N. P. Gerenciamento & controle de projetos. Rio de Janeiro: Axcel Books, 2003.

DEITEL, H. M. E DEITEL, P.J. Internet & *World Wide Web*, como programar .2ed. Porto Alegre: Bookman, 2003.

SOARES, Walace. MySQL Conceitos e Aplicações. 1 ed. São Paulo: Eda. Érica Ltda, 2001.

WAMP: WAMP Apache + MySQL + PHP + Perl. Disponível em: <http:// www.wampserver.com/ >. Acesso em: 26/11/2015, 17:30.

WORKBENCH, MySQL. Site oficial MySQL. Disponível em: http://www.mysql.com/products/workbench/. Acesso em: 08/11/2014, 21:40.  
64 YIN, K. Robert. Estudo de Caso: Planejamento e Métodos. 5ª Edição. Local: Porto Alegre. Bookman, 2014.

# APÊNDICES A - ENTREVISTA

**Questionário**

**1 - Atualmente, como funciona o gerenciamento de pessoal do estacionamento?**

*O controle do pessoal é feito através de registros em papeis armazenados em armário de pasta suspensa que fica na sala de administração do estacionamento.*

**2 - Atualmente, como funciona o controle de entrada e saída de automóveis que utilizam o estacionamento?**

*O controle é feito através de tickets preenchidos manualmente, no qual são entregues na entrada e recolhidos na saída.*

**3 - Há algum controle desses tickets?**

*Há uma numeração impressa nos tickets.*

**4 - Como é realizado o registro de entrada do veículo no estacionamento?**

*São coletadas algumas informações como marca, cor e placa do carro e horário de entrada e entregue um ticket ao motorista com o horário de entrada.*

**5 - Como é realizado o registro de saída do veículo do estacionamento?**

*O motorista entrega ao atendente o ticket com o horário e calcula o tempo e o valor a ser pago.*

**6 - Como é feito o cálculo do tempo de permanência do veículo no estacionamento?**

*A cada meia hora é cobrado um valor fixo. Após de 3 horas de permanência, o valor da hora cheia cai 25%.*

**7 - Qual é o tempo mínimo de permanência no estacionamento sem haver cobrança?**

*O tempo de tolerância é de 15 minutos.*

**8 - Qual o valor cobrado no estacionamento?**

*Até 15 minutos, gratuito, após esse tempo, cobra-se o valor de 30 minutos.*

*Fração de 30 minutos R$ 1,00.*

*A cada uma hora R$ 2,00.*

*A partir da terceira hora de permanência, a hora cai pra R$ 1,50.*

**9 - Qual a quantidade total de vagas no estacionamento?**

*Dispomos hoje de 300 vagas para automóveis e 50 vagas para motos.*

**10 - Há diferença de valores entre carro e moto?**

*Não, ambos pagam o mesmo valor.*

**11 - Como é feito o fechamento de caixa?**

*São somados todos os tickets e cadastrados no livro de registro.*

**12 - Caso o motorista perca o ticket, como é realizado o controle?**

*O motorista paga uma taxa de R$ 20,00 e no livro é registrado número do ticket que foi perdido.*

**13 - Quantos usuários terão no sistema?**

*Um administrador ou gerente e um atendente.*

# APÊNDICES B - DOCUMENTO DE VISÃO

**Visão**

**1.** **Introdução**

A finalidade deste documento é coletar, analisar e definir necessidades e recursos de nível superior do Sistema de Gerenciamento de Estacionamento - STACIONE. Ele se concentra nos recursos necessários aos envolvidos e aos usuários-alvo e nas razões que levam a essas necessidades. Os detalhes de como o Sistema de Gerenciamento de Estacionamento – STACIONE satisfaz essas necessidades são descritos no caso de uso e nas especificações suplementares.

**1.1** **Referências**

O conteúdo deste documento foi baseado em coleta de informações em Entrevista realizada na Empresa Estacionamentos LTDA, fornecidos pelo cliente, para prover o embasamento legal ao qual o sistema terá sustentação.

**2.** **Posicionamento**

**2.1 Descrição do Problema**

Atualmente a empresa Estacionamentos LTDA trabalha de forma antiquada gerando inconsistências no registro de entrada e saída de veículos no estacionamento. Além disso, o controle de caixa e relatórios é feito manualmente podendo haver distorções e erros no controle financeiro.

|  |  |
| --- | --- |
| O problema de | Processo atual de registro de entrada e saída de carros não atende os anseios da empresa. |
| afeta | Os proprietários e os atendentes da Empresa Estacionamentos. |
| cujo impacto é | Perda de registros e consequentemente perdas de receita na empresa. |
| uma boa solução seria | Criar e automatizar o processo de registro de entrada e saída de automóveis, permitindo melhor controle e atendimento ao cliente. |

**2.2** **Sentença de Posição do Produto**

|  |  |
| --- | --- |
| Para | Empresas que fornecem serviços de estacionamento. |
| Que | Necessitam automatizar os seus serviços. |
| O STACIONE | É um sistema de gerenciamento de estacionamento leve, capaz de automatizar a administração e o estacionamento otimizando o serviço dos funcionários. |
| Que | É uma solução encontrada para realizar o registro de fluxo de automóveis, relatórios e usuários de maneira simples e intuitiva. |
| Ao contrário de | Sistemas de estacionamento convencionais que precisam de vários funcionários, o STACIONE funciona a partir de duas pessoas. |
| Nosso produto | Possui a capacidade de oferecer aos funcionários uma plataforma amigável. |

**3.** **Descrições dos Envolvidos e dos Usuários**

Os envolvidos no presente projeto são os funcionários, os clientes e o dono do estacionamento.

**3.1 Resumo dos Envolvidos**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Descrição** | **Responsabilidades** |
| Dono do estacionamento | Responsável pelo financiamento do projeto. | - Monitora o andamento do projeto  - aprova o financiamento |
| Funcionários | Usuários que utilizarão o sistema. | - Utiliza o sistema.  - atende da melhor forma o cliente. |
| Cliente do estacionamento | Usuário comum que utiliza o serviço de estacionamento. | -realiza o pagamento após uso do estacionamento. |

**3.2 Resumo dos Usuários**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Descrição** | **Responsabilidades** | **Envolvido** |
| Gerente |  | - Pela manutenção dos usuários no sistema;  - Pela geração dos relatórios quando solicitado.  - pelo cadastro e alteração dos preços no estacionamento. | Gerente e Atendente |
| Atendente |  | - Registro de entrada e saída do veículo no estacionamento;  - registro do pagamento. | Gerente e Atendente |

**3.3** **Ambiente do Usuário**

Atualmente, a empresa Estacionamento LTDA é composta por 3 funcionários, sendo um gerente e dois atendentes. Esses que por sua vez registram diariamente a ENTRADA e SAIDA de veículos dos clientes de forma manual, tudo é repassado ao final do dia para o gerente e o mesmo faz o faturamento diário do estacionamento.

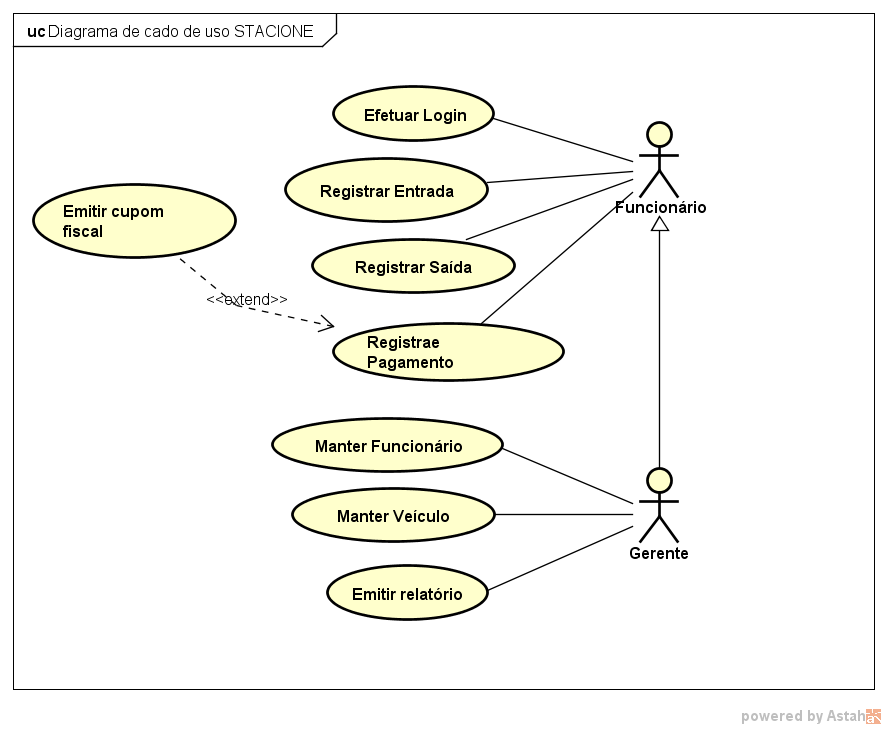
**3.4** **Principais Necessidades dos Usuários ou dos Envolvidos**

No momento de entrada no estacionamento, o cliente recebe um ticket contendo data e hora de entrada, placa do veículo e número de identificação para controle no livro de registro.

Outro problema constatado é que os processos são realizados manualmente, ou seja, relatórios, entrada e saída de veículos são registrados em livro manuscrito ou entrega de ticket, podendo haver erros e discrepâncias.

# APÊNDICES C – DIAGRAMA E ESPECIFICAÇÃO DE CASO DE USO

**Diagrama de Casos de Uso**

****

Login

# Descrição

Este caso uso tem por objetivo permitir e direcionar o acesso de acordo com o perfil do usuário no Sistema de Gerenciamento de Estacionamento – Stacione.

1. Referências

* Manual do Sistema.

1. Atores
   1. Gerente

Responsável por manter os usuários no sistema.

* 1. Atendente

O Atendente é responsável por registrar a entrada e saída dos veículos no estacionamento.

1. Fluxo de eventos
   1. Fluxo principal
   2. O caso de uso é iniciado quando o ator acessa a tela de login do sistema.
   3. O ator informa login e senha.
   4. O sistema valida os dados e direciona ao perfil correspondente. (E1) (E2)
   5. O sistema apresenta o menu principal.
   6. O caso de uso é encerrado.
   7. Fluxos alternativos

Não se aplica.

* 1. Fluxos de Exceção

##### Campos obrigatórios

* 1. O sistema verifica que os campos obrigatórios não foram informados. (RA1)
  2. O sistema apresenta a mensagem: “Campos obrigatórios não informados! ”

##### Dados não encontrados

* 1. O sistema verifica que os dados informados não constam no banco de dados.
  2. O sistema apresenta a mensagem: “Dados informados não encontrados! ”

1. Regras de Negócio

## Regra de Apresentação.

RA1 Todos os campos devem ser preenchidos.

## Regra de Negócio

Não se aplica.

# Requisitos Especiais

## Não se aplica.

1. Pré-condições

## O ator deverá estar devidamente cadastrado no sistema.

1. Pós-condições

## Registro do usuário deverá ser mantido no sistema.

1. Pontos de extensão

## Não se aplica.

Registrar Entrada

# Descrição

Este caso uso tem por objetivo permitir incluir o registro de entrada de veículos no Sistema de Gerenciamento de Estacionamento – Stacione.

1. Referências

* Manual do Sistema.

1. Atores
   1. Atendente

O Atendente é o responsável em registrar a entrada dos veículos no estacionamento utilizando o Sistema de Gerenciamento de Estacionamento - Stacione.

1. Fluxo de eventos
   1. Fluxo principal
   2. O caso de uso é iniciado quando o ator clica a botão Entrada da tela principal.
   3. O sistema apresenta a tela de Registrar Entrada para o ator.
   4. O ator vai tirar uma foto do veículo com a câmera já projetada para placa.
   5. O ator informa os dados do veículo no formulário e aciona o botão registrar. (RA1) (RN1)
   6. O sistema grava os dados. (E1) (PE1)
   7. O sistema apresenta a mensagem: “Registrado com sucesso! ”.
   8. O caso de uso é finalizado.
   9. Fluxos alternativos

Não se aplica.

* 1. Fluxos de Exceção

1. Dados incompletos
   1. O sistema identifica que o ator deixou de preencher algum campo obrigatório;
   2. O sistema apresenta a mensagem: “Campos obrigatórios não informados! ";
   3. O sistema retorna ao P3 do fluxo principal.
2. Regras de Negócio

## Regra de Apresentação

RA1 O campo placa é obrigatório e deverá ser composto de 7 (sete) dígitos numéricos no formato (AAA-9999).

## Regra de Negócio

RN1 Todos os campos devem ser preenchidos.

# Requisitos Especiais

Não se aplica.

1. Pré-condições

## O ator deverá estar devidamente cadastrado no sistema com perfil Atendente.

1. Pós-condições

## Registro do veículo mantido no sistema.

1. Pontos de extensão

PE1 Emitir comprovante de estacionamento.

Registrar Saída

# Descrição

Este caso uso tem por objetivo permitir o registro de saída de veículos no Sistema de Gerenciamento de Estacionamento – Stacione.

1. Referências

* Manual do Sistema.

1. Atores
   1. Atendente

O Atendente é o responsável em registrar a saída dos veículos no estacionamento utilizando o Sistema de Gerenciamento de Estacionamento - Stacione.

1. Fluxo de eventos
   1. Fluxo principal
   2. O caso de uso é iniciado quando o ator acessa a aba início do menu principal.
   3. O sistema apresenta a tela de consulta de entrada para o ator.
   4. O ator consulta a lista dos veículos cadastrados ativos.
   5. O ator seleciona a placa do veículo consultado e aciona o botão registrar saída.
   6. O ator confirma a saída do veículo.
   7. Fluxos alternativos

Não se aplica.

* 1. Fluxos de Exceção

1. Dados incompletos
   1. O sistema identifica que o ator deixou de preencher algum campo obrigatório;
   2. O sistema apresenta a mensagem: “Campos obrigatórios não informados! ";
   3. O sistema retorna ao P3 do fluxo principal.
2. Regras de Negócio

## Regra de Apresentação

RA1 O campo recebido deverá ser em formato em real.

## Regra de Negócio

Não se aplica.

# Requisitos Especiais

Não se aplica.

1. Pré-condições

## O veículo deve estar registrado no sistema.

## O ator deverá estar devidamente cadastrado no sistema com perfil Atendente.

1. Pós-condições

## Registro do veículo mantido no sistema.

1. Pontos de extensão

PE1 Registrar pagamento.

Registrar Pagamento

# Descrição

Este caso uso tem por objetivo permitir o registro do pagamento na saída de veículos no Sistema de Gerenciamento de Estacionamento – Stacione.

1. Referências

* Manual do Sistema.

1. Atores
   1. Atendente

O Atendente é o responsável em registrar o pagamento e a saída dos veículos no estacionamento utilizando o Sistema de Gerenciamento de Estacionamento - Stacione.

1. Fluxo de eventos
   1. Fluxo principal
   2. O caso de uso é iniciado após o Ator executar o P7 do Fluxo principal do Caso de Uso Registrar Saída.
   3. O sistema calcula o tempo de permanência e o preço a ser pago.
   4. O sistema apresenta o valor a ser pago.
   5. O ator aciona a opção registrar pagamento. (PE1) (E4)
   6. O sistema imprime o cupom fiscal.
   7. O caso de uso é finalizado.
   8. O sistema registra o pagamento.
   9. O sistema apresenta a mensagem: “Pagamento efetuado com sucesso!”.
   10. O caso de uso é finalizado.
   11. Fluxos alternativos

Não se aplica.

* 1. Fluxos de Exceção

Não se aplica.

1. Regras de Negócio

## Regra de Apresentação

Não se aplica.

## Regra de Negócio

Não se aplica.

# Requisitos Especiais

Não se aplica.

1. Pré-condições

## O Ator deverá ter inicializado o Caso de Uso Registrar Saída.

1. Pós-condições

## O sistema retorna ao P8 do Fluxo principal do Caso de Uso Registrar Saída.

1. Pontos de extensão

Não se aplica.

Gerar Relatórios

# Descrição

Este caso uso tem por objetivo permitir gerar relatórios dos veículos e pagamentos do Sistema de Gerenciamento de Estacionamento – Stacione.

1. Referências

* Manual do Sistema.

1. Atores
   1. Gerente

O Gerente é o responsável em gerar os relatórios dos veículos no estacionamento utilizando o Sistema de Gerenciamento de Estacionamento - Stacione.

1. Fluxo de eventos
   1. Fluxo principal
   2. O caso de uso é iniciado quando o ator acessa a aba relatórios do menu principal
   3. O sistema apresenta a tela de relatórios para o ator.
   4. O ator adiciona as datas. (RA1) (RN1)
   5. O sistema gera o relatório.
   6. O caso de uso é finalizado.
   7. Fluxos alternativos

Não se aplica.

* 1. Fluxos de Exceção

1. Dados incompletos
   1. O sistema identifica que o ator deixou de preencher algum campo obrigatório;
   2. O sistema retorna ao P3 do fluxo principal.
2. Regras de Negócio

## Regra de Apresentação.

## Regra de Negócio

RN1 Todos os campos devem ser preenchidos.

# Requisitos Especiais

Não se aplica.

1. Pré-condições

## O ator deverá estar devidamente cadastrado no sistema com perfil gerente.

1. Pós-condições

Não se aplica.

1. Pontos de extensão

PE1 Emitir o relatório.

Manter tabela de preços

# Descrição

Este caso uso tem por objetivo permitir cadastrar tabela de preço do Sistema de Gerenciamento de Estacionamento – Stacione.

1. Referências

* Manual do Sistema.

1. Atores
   1. Gerente

O Gerente é o responsável manter a tabela de preços no estacionamento utilizando o Sistema de Gerenciamento de Estacionamento - Stacione.

1. Fluxo de eventos
   1. Fluxo principal
   2. O caso de uso é iniciado quando o ator acessa a aba preços do menu principal
   3. O sistema apresenta a tela de preços para o ator.
   4. O ator seleciona a opção de alterar preço.
   5. O ator adiciona o novo preço . (RA1) (RN1)
   6. O sistema retorna a mensagem que o preço foi cadastrado
   7. O caso de uso é finalizado.
   8. Fluxos alternativos

Não se aplica.

* 1. Fluxos de Exceção

1. Dados incompletos
   1. O sistema identifica que o ator deixou de preencher algum campo obrigatório;
   2. O sistema retorna ao P2 do fluxo principal.
2. Regras de Negócio

## Regra de Apresentação.

## Regra de Negócio

RN1 Todos os campos devem ser preenchidos.

# Requisitos Especiais

Não se aplica.

1. Pré-condições

## O ator deverá estar devidamente cadastrado no sistema com perfil gerente.

1. Pós-condições

Não se aplica.

1. Pontos de extensão

Não se aplica.

Manter Vagas

# Descrição

Este caso uso tem por objetivo permitir cadastrar números de vagas no Sistema de Gerenciamento de Estacionamento – Stacione.

1. Referências

* Manual do Sistema.

1. Atores
   1. Gerente

O Gerente é o responsável manter o número de vagas no estacionamento utilizando o Sistema de Gerenciamento de Estacionamento - Stacione.

1. Fluxo de eventos
   1. Fluxo principal
   2. O caso de uso é iniciado quando o ator acessa a aba Vagas do menu principal
   3. O sistema apresenta a tela de Vagas para o ator.
   4. O ator seleciona a opção de alterar número de vagas.
   5. O ator adiciona o novo número. (RA1) (RN1)
   6. O sistema retorna a mensagem “Números de vagas foi alterado! ”
   7. O caso de uso é finalizado.
   8. Fluxos alternativos

Não se aplica.

* 1. Fluxos de Exceção

1. Dados incompletos
   1. O sistema identifica que o ator deixou de preencher algum campo obrigatório;
   2. O sistema retorna ao P2 do fluxo principal.
2. Regras de Negócio

## Regra de Apresentação.

## Regra de Negócio

RN1 Todos os campos devem ser preenchidos.

# Requisitos Especiais

Não se aplica.

1. Pré-condições

## O ator deverá estar devidamente cadastrado no sistema com perfil gerente.

1. Pós-condições

Não se aplica.

1. Pontos de extensão

Não se aplica.

# APÊNDICES D - DIAGRAMAS DE SEQUÊNCIA

Diagrama de Sequência - Efetuar Login

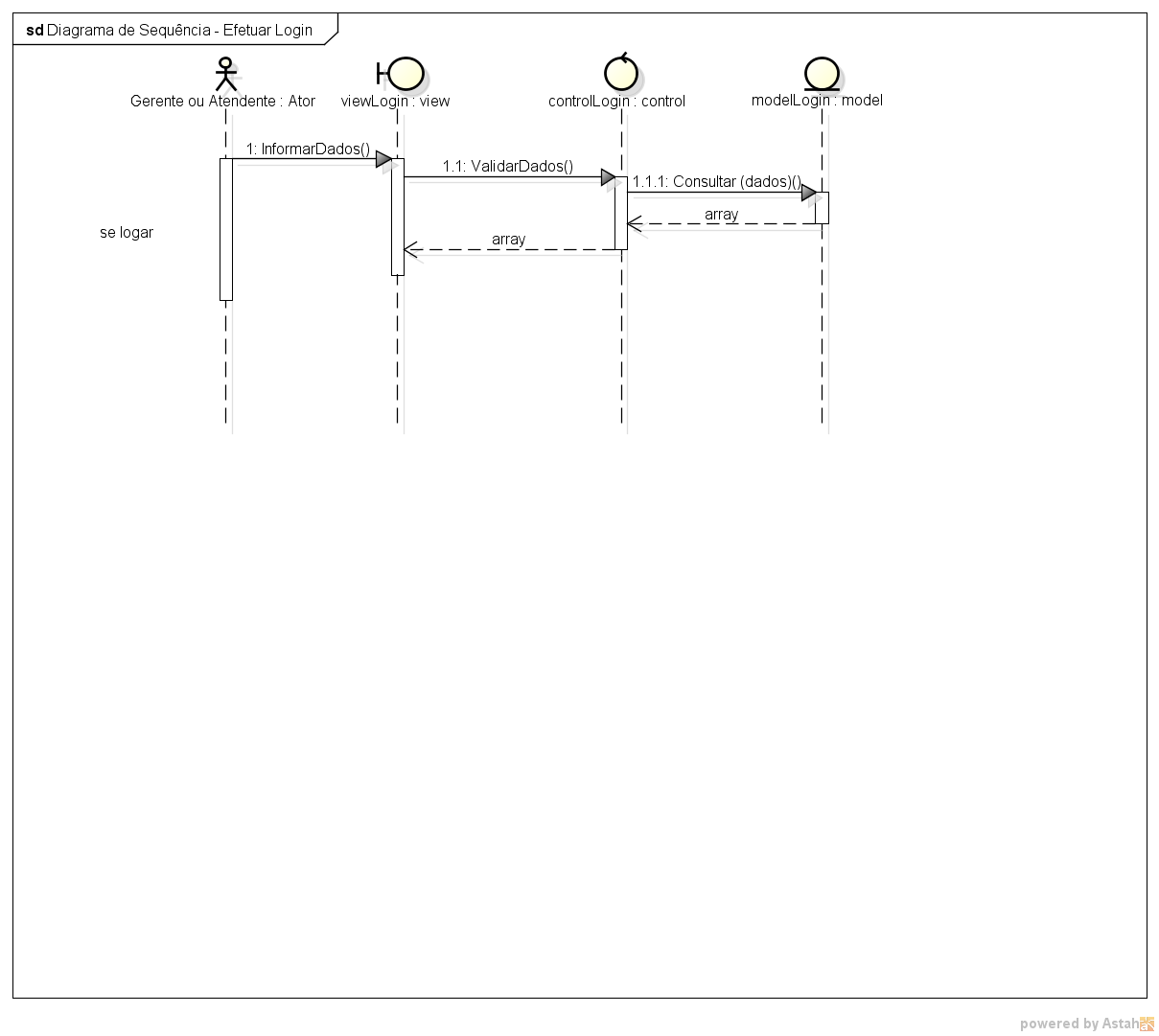


Diagrama de Sequência – Emitir comprovante

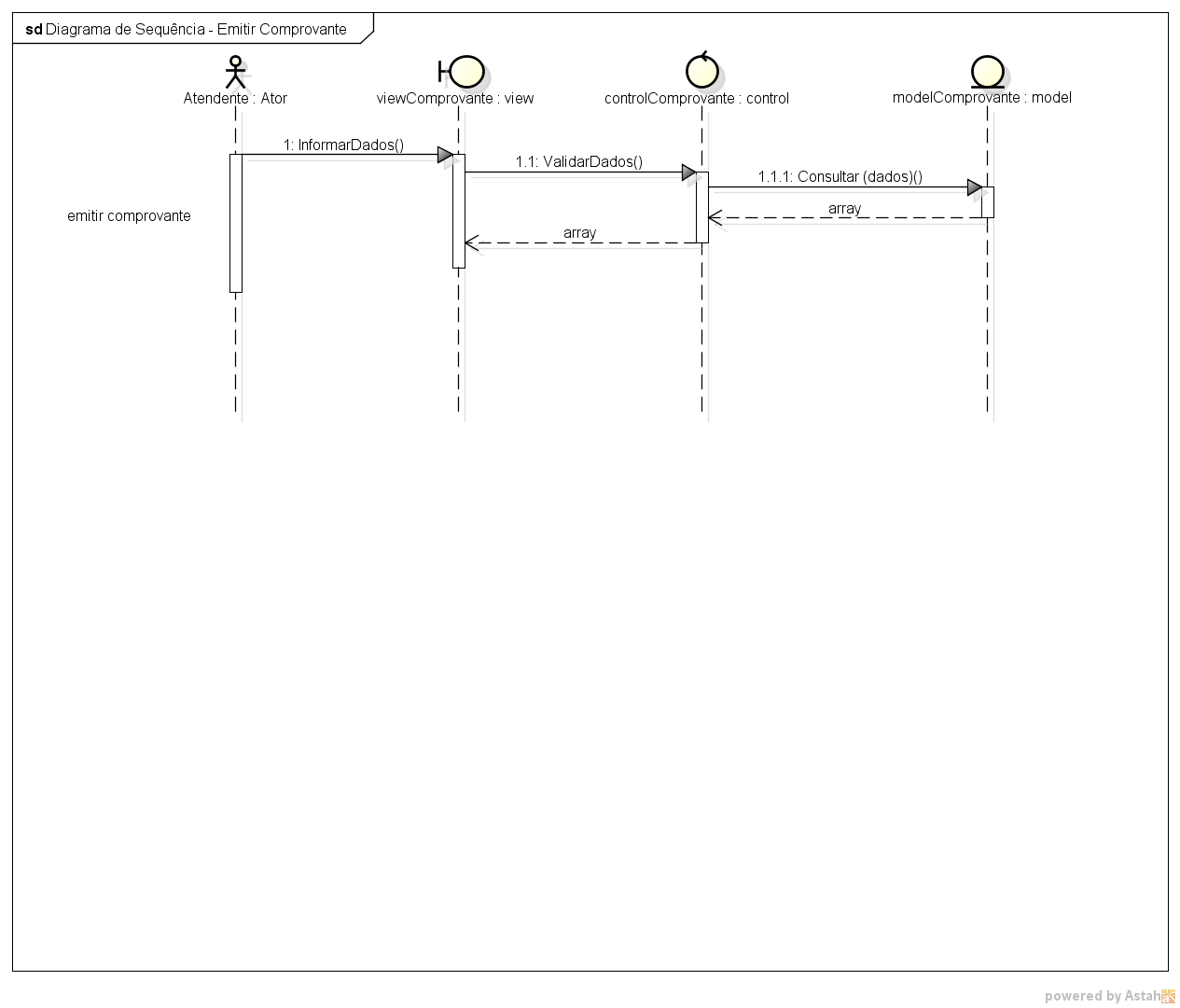


Diagrama de Sequência – Gerar relatórios

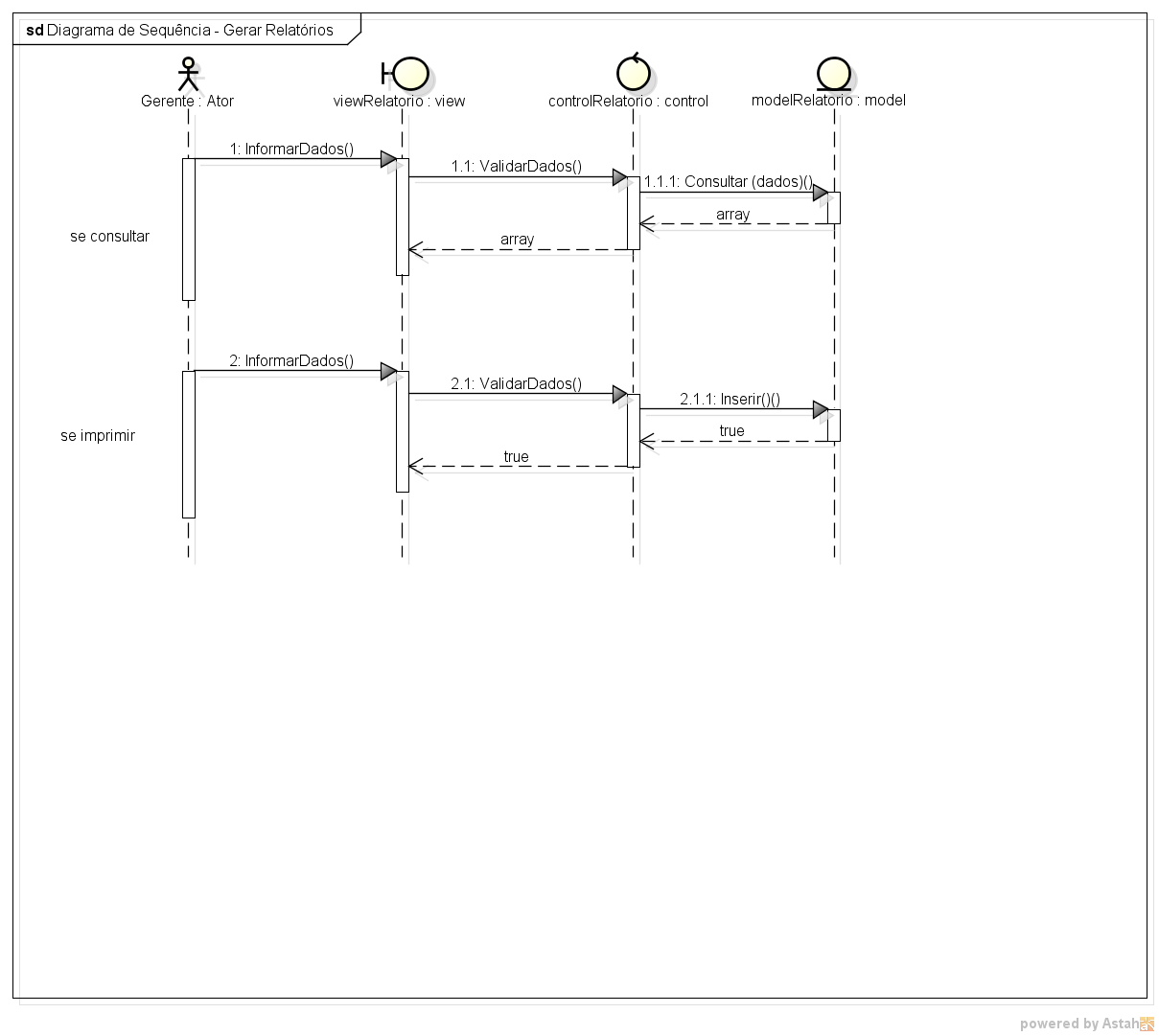


Diagrama de Sequência – Manter Funcionário

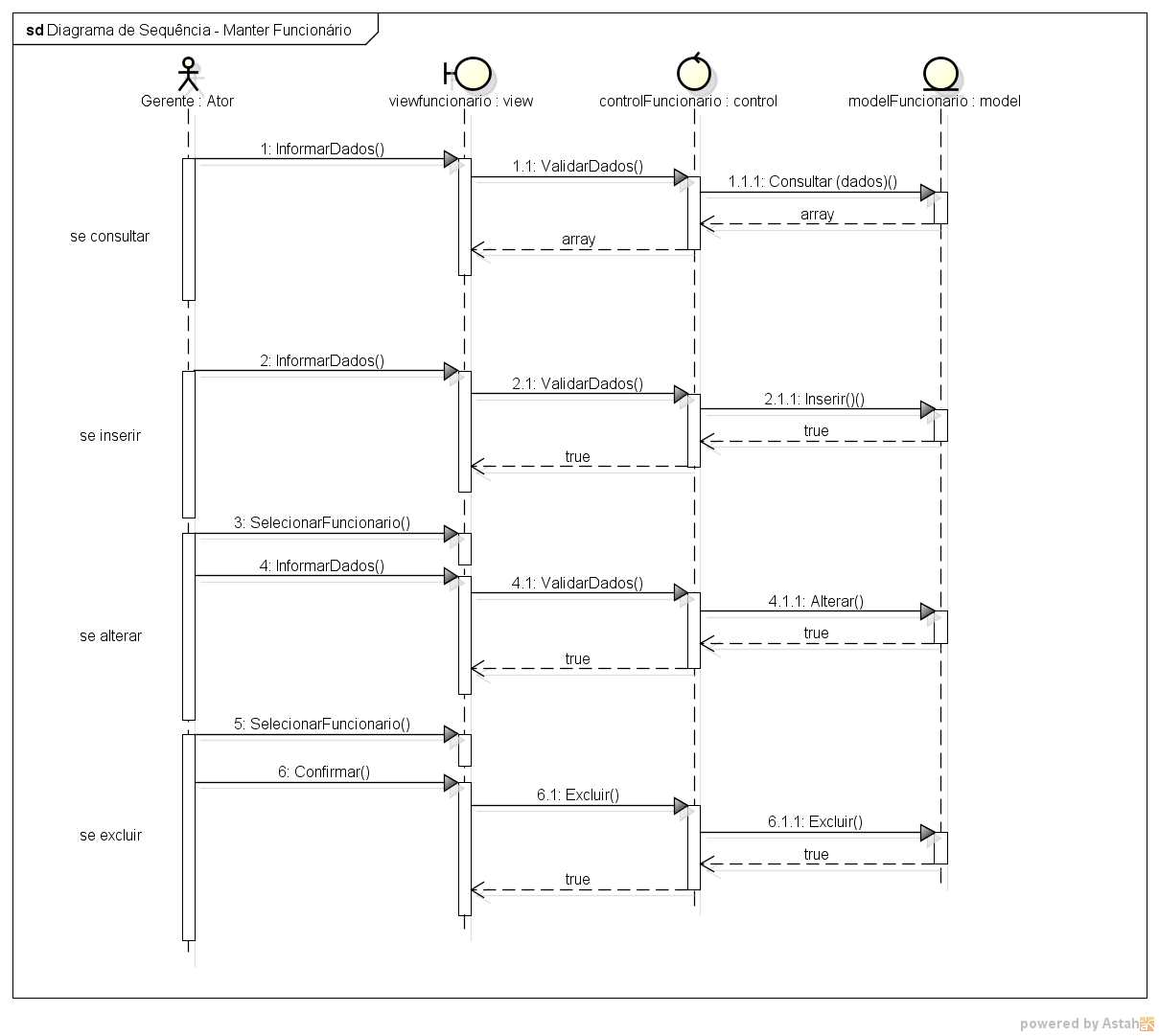


Diagrama de Sequência – Manter Tabela de Preços

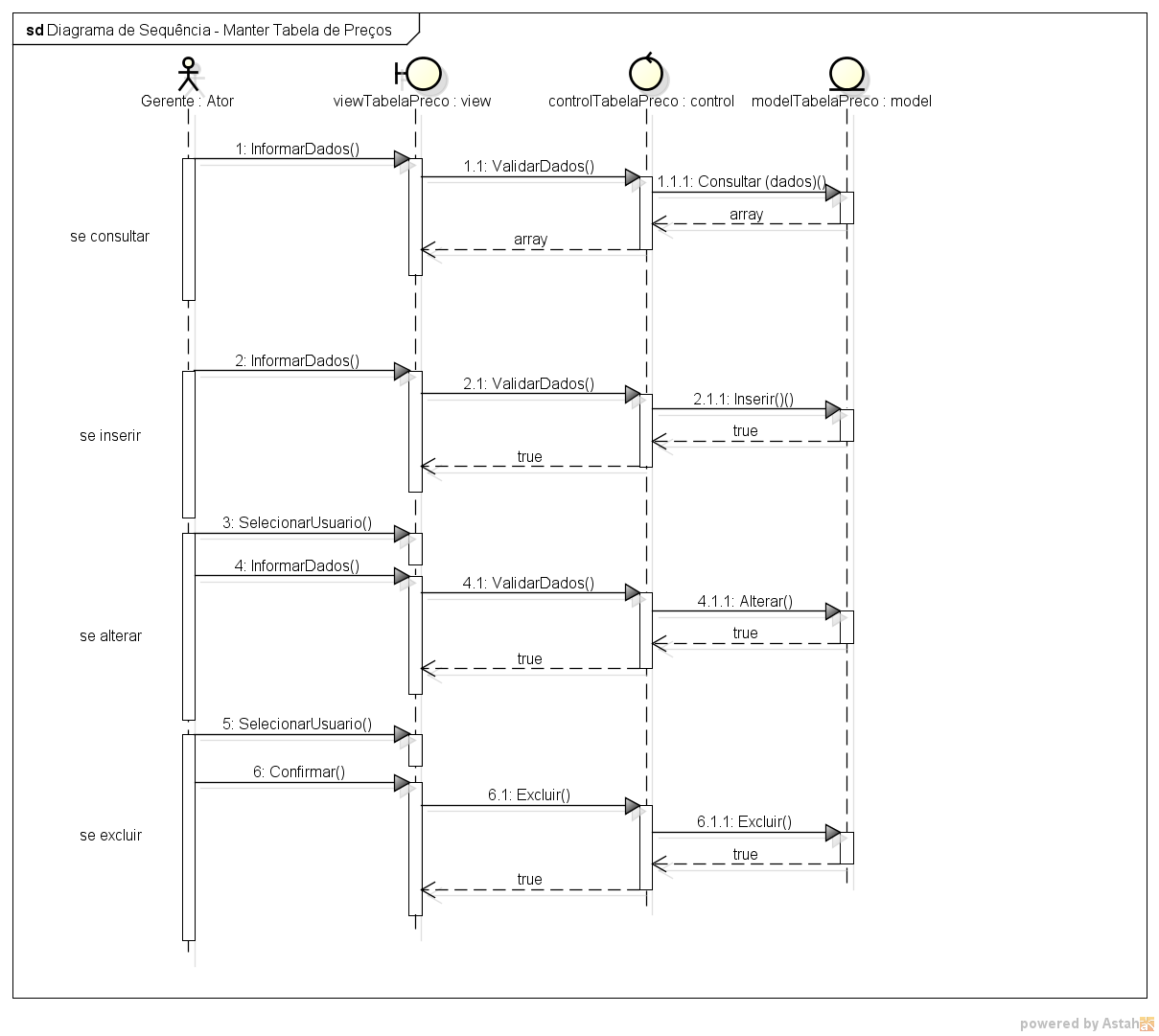


Diagrama de Sequência – Registrar Entrada

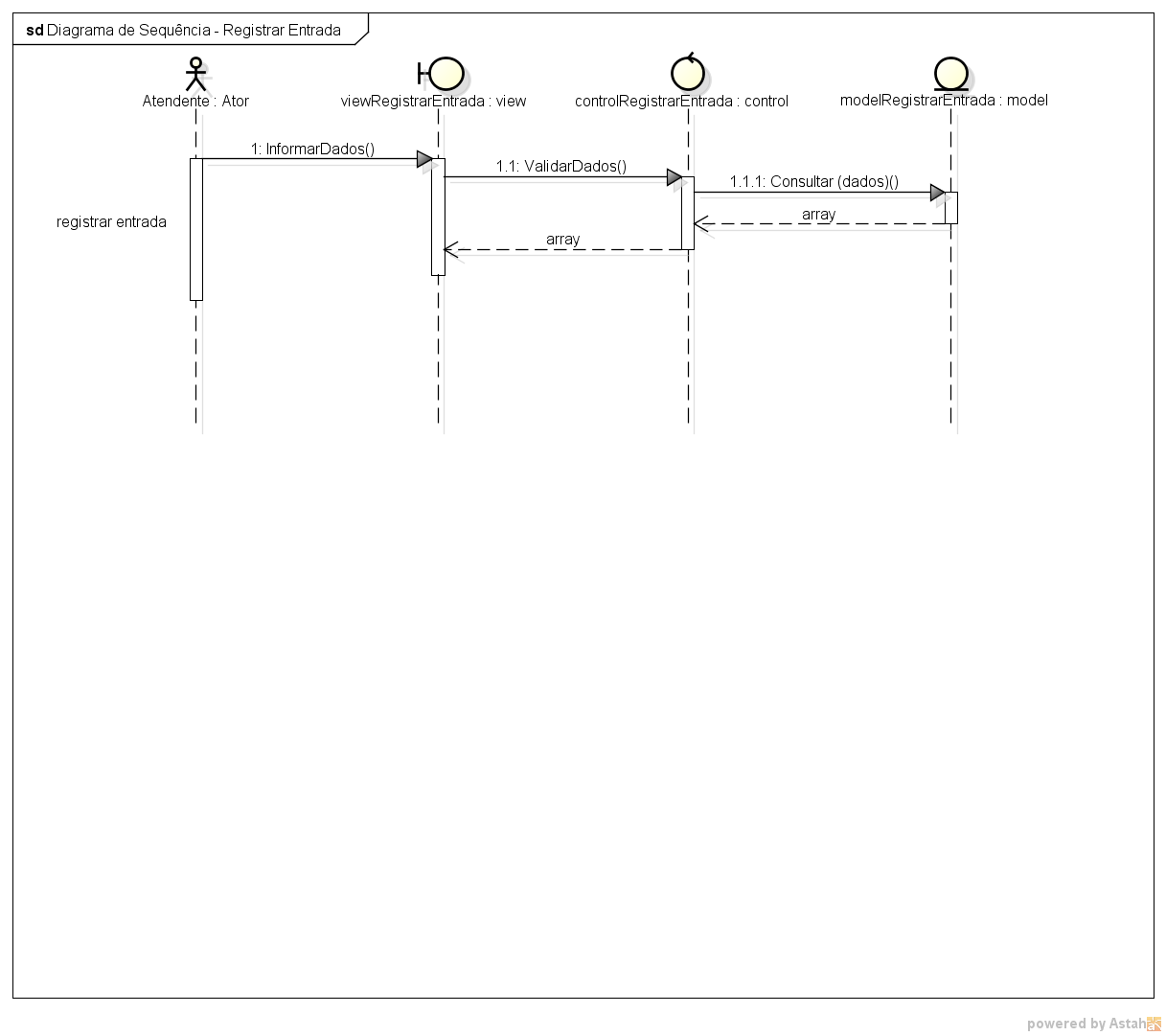


Diagrama de Sequência – Registrar Pagamento

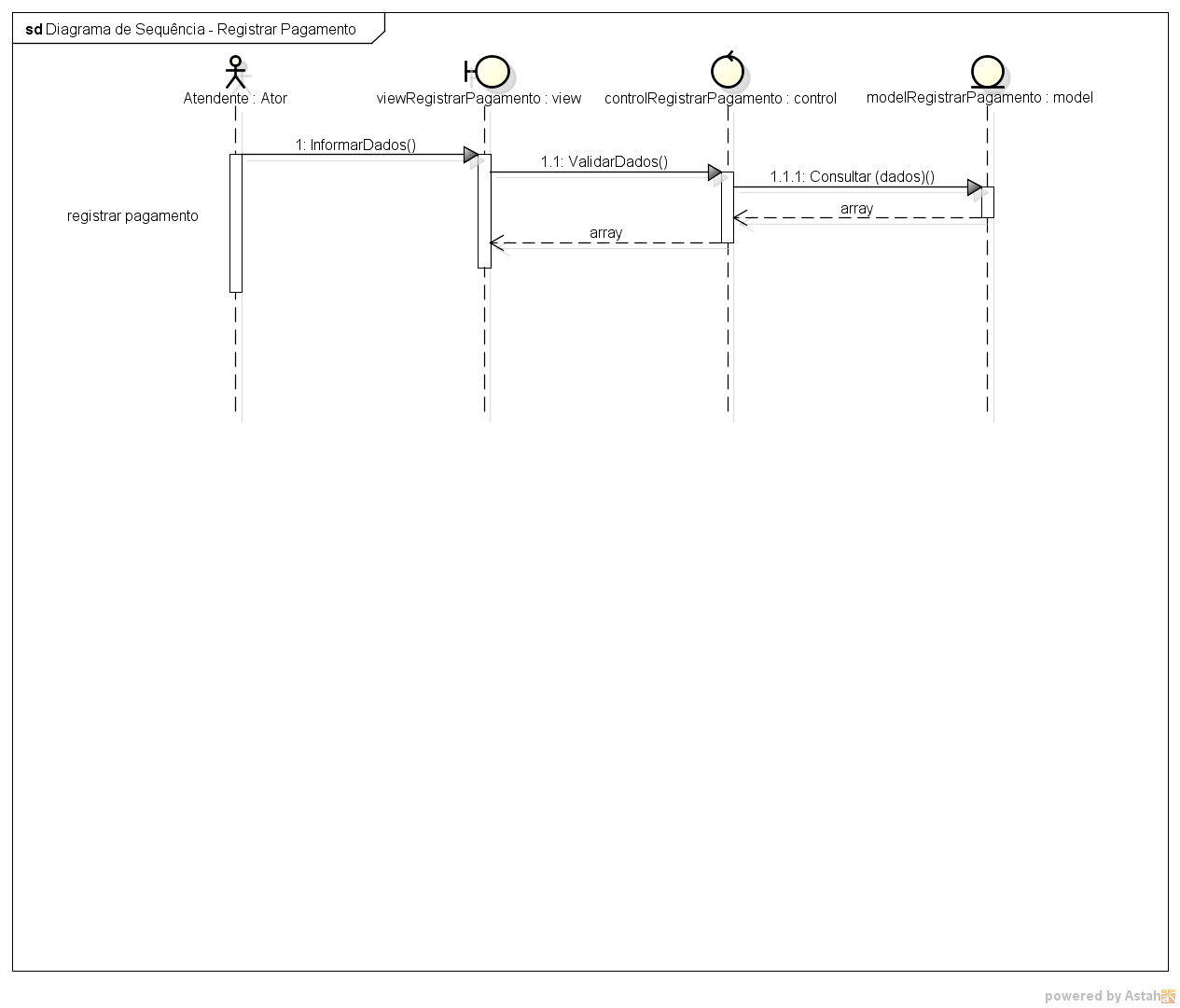
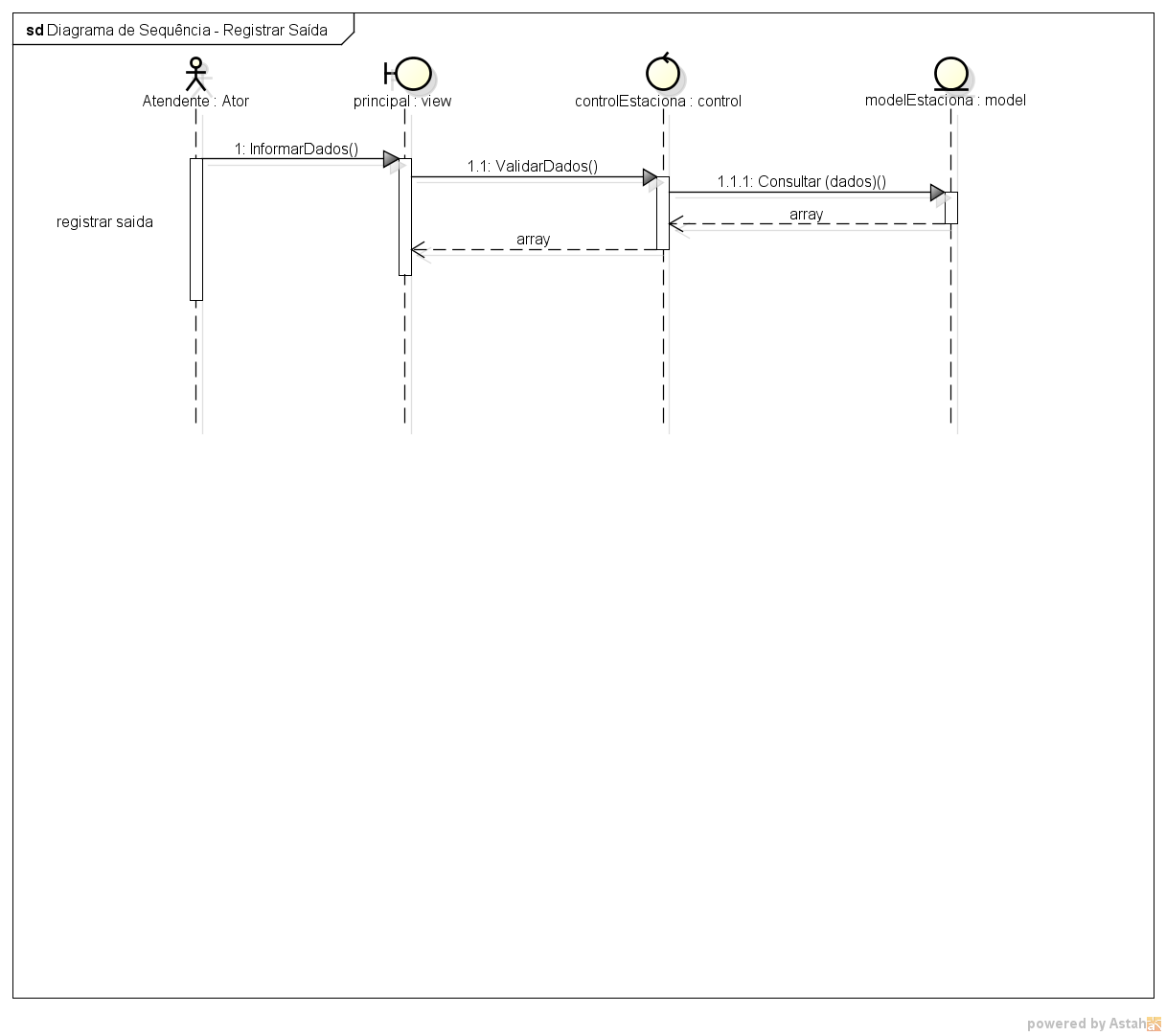
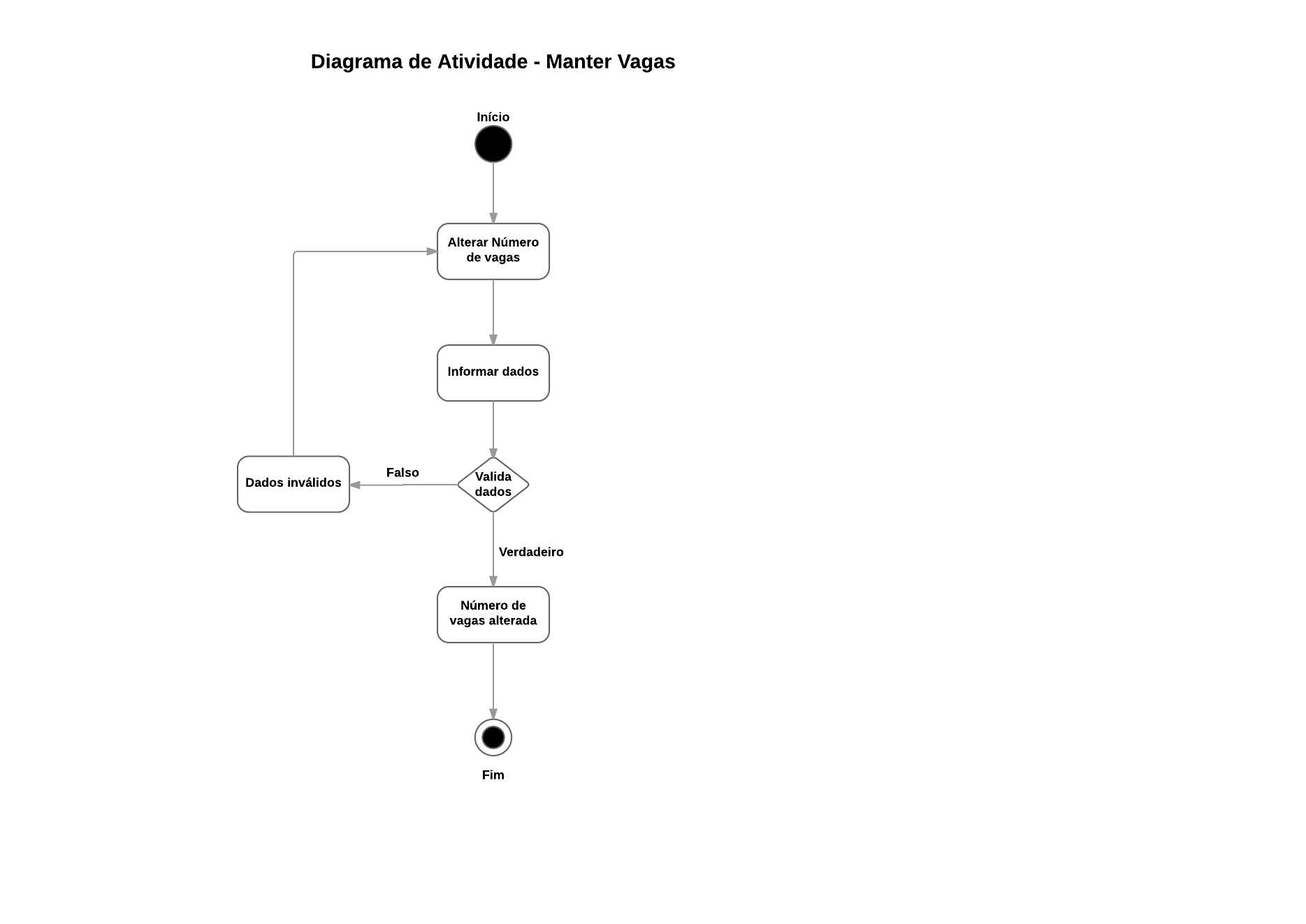
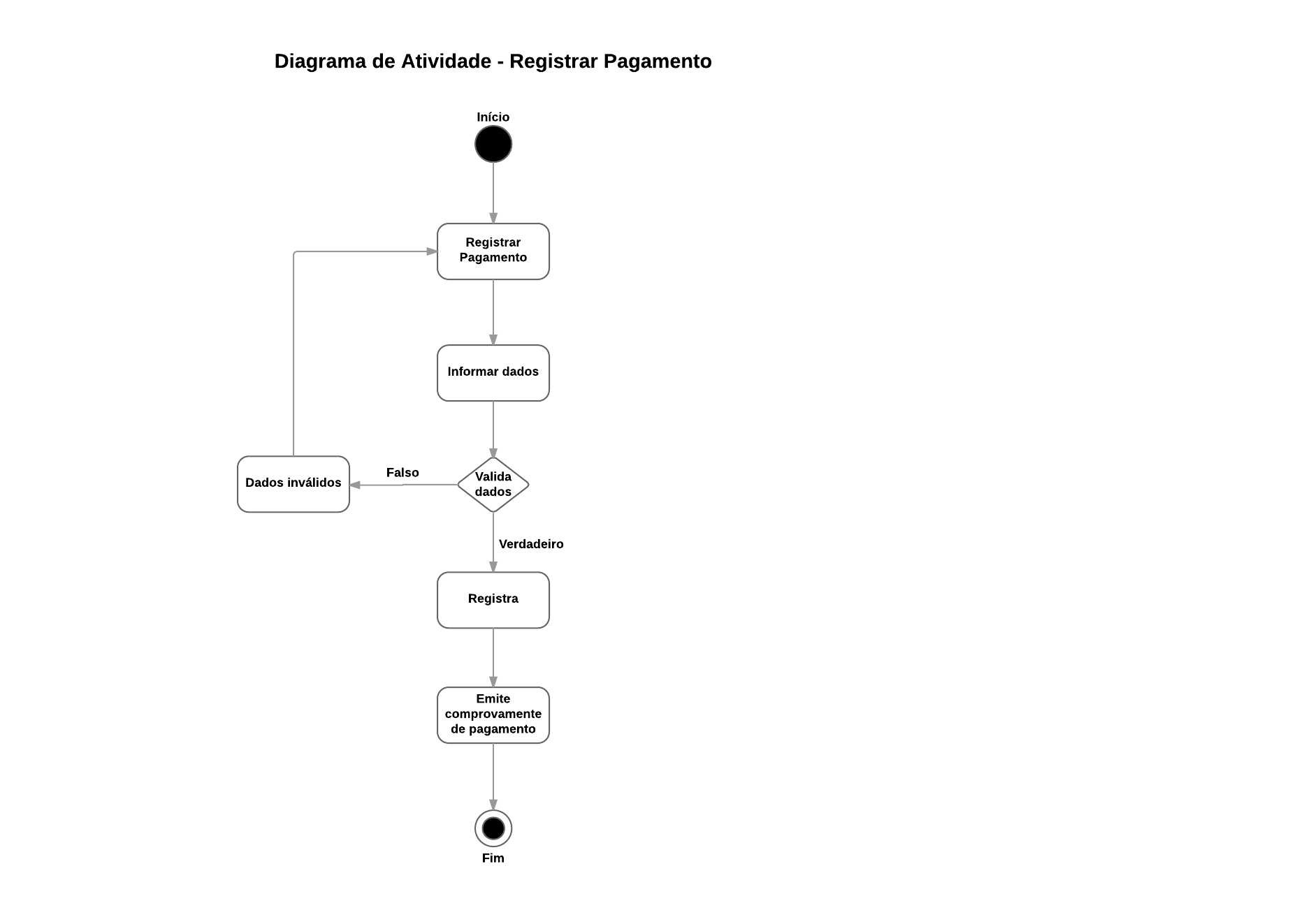
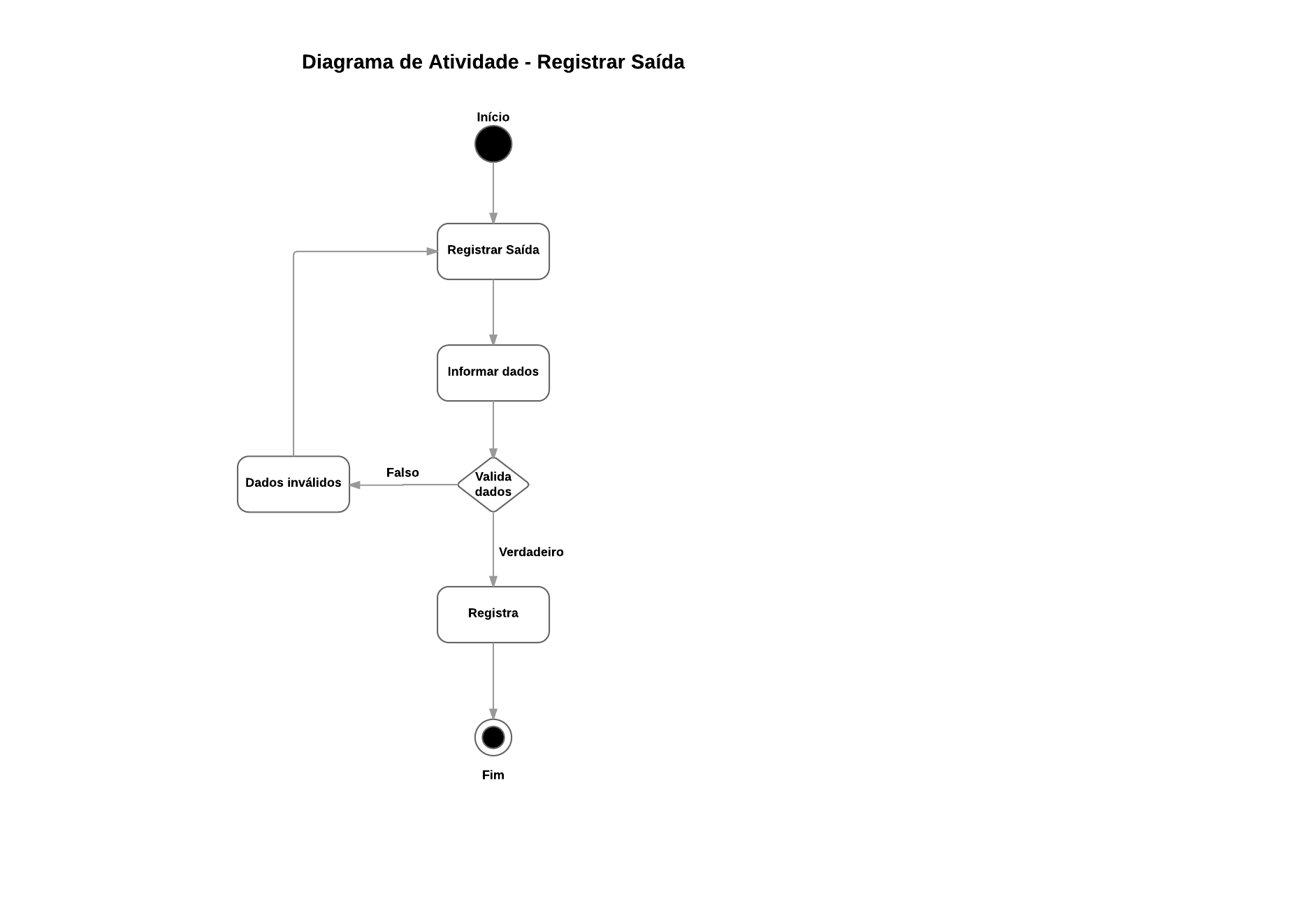
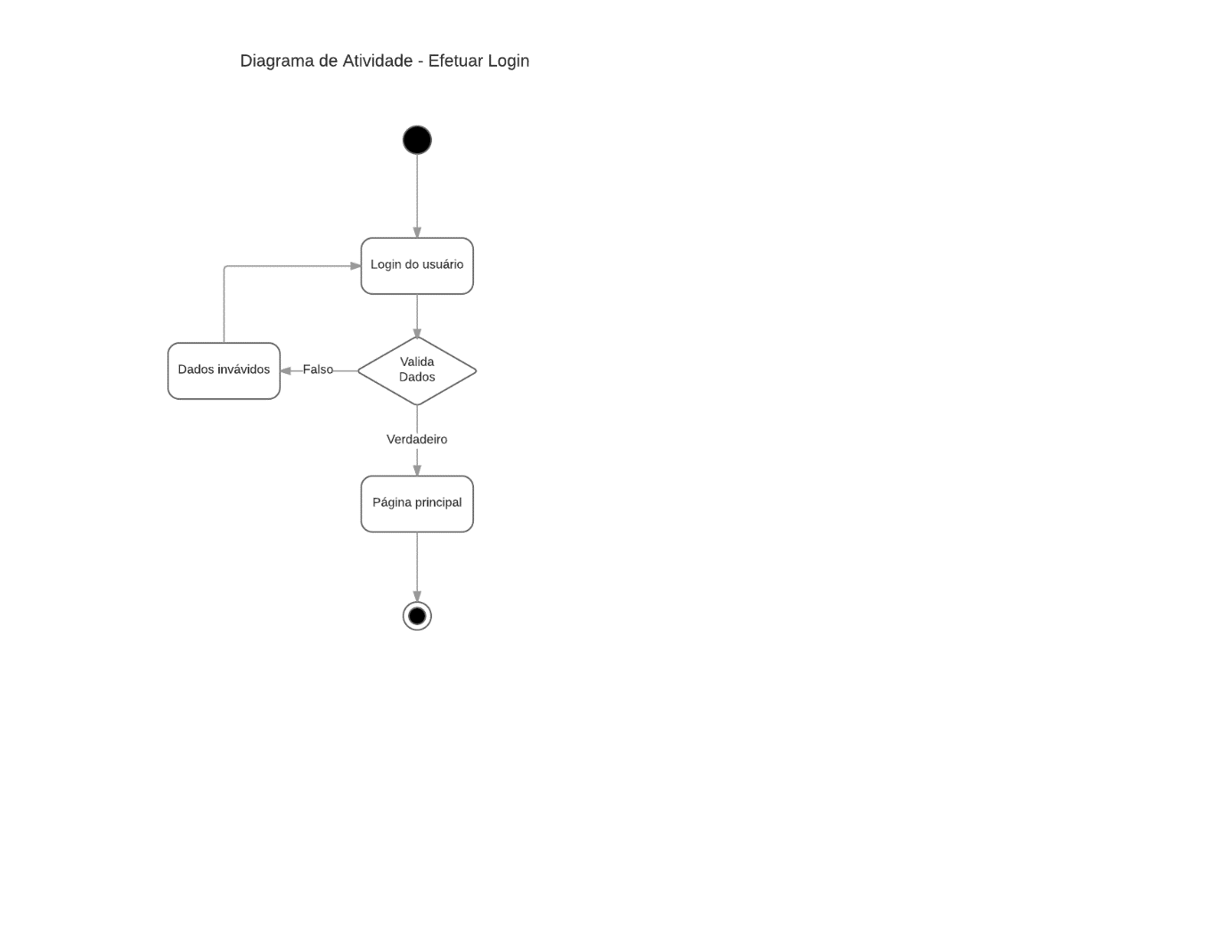
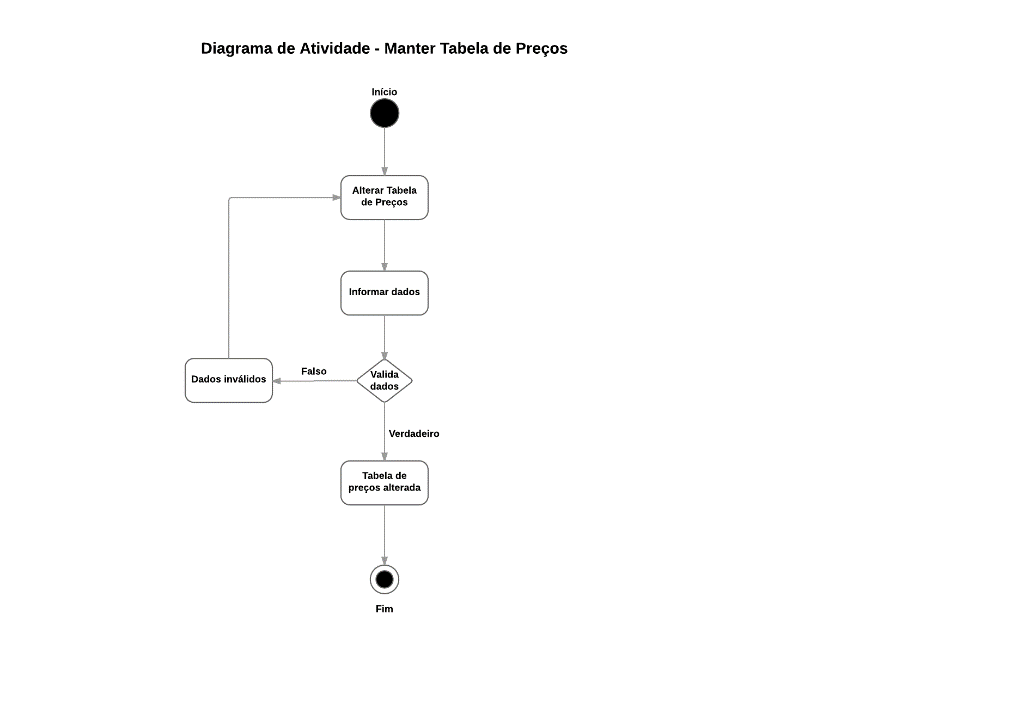


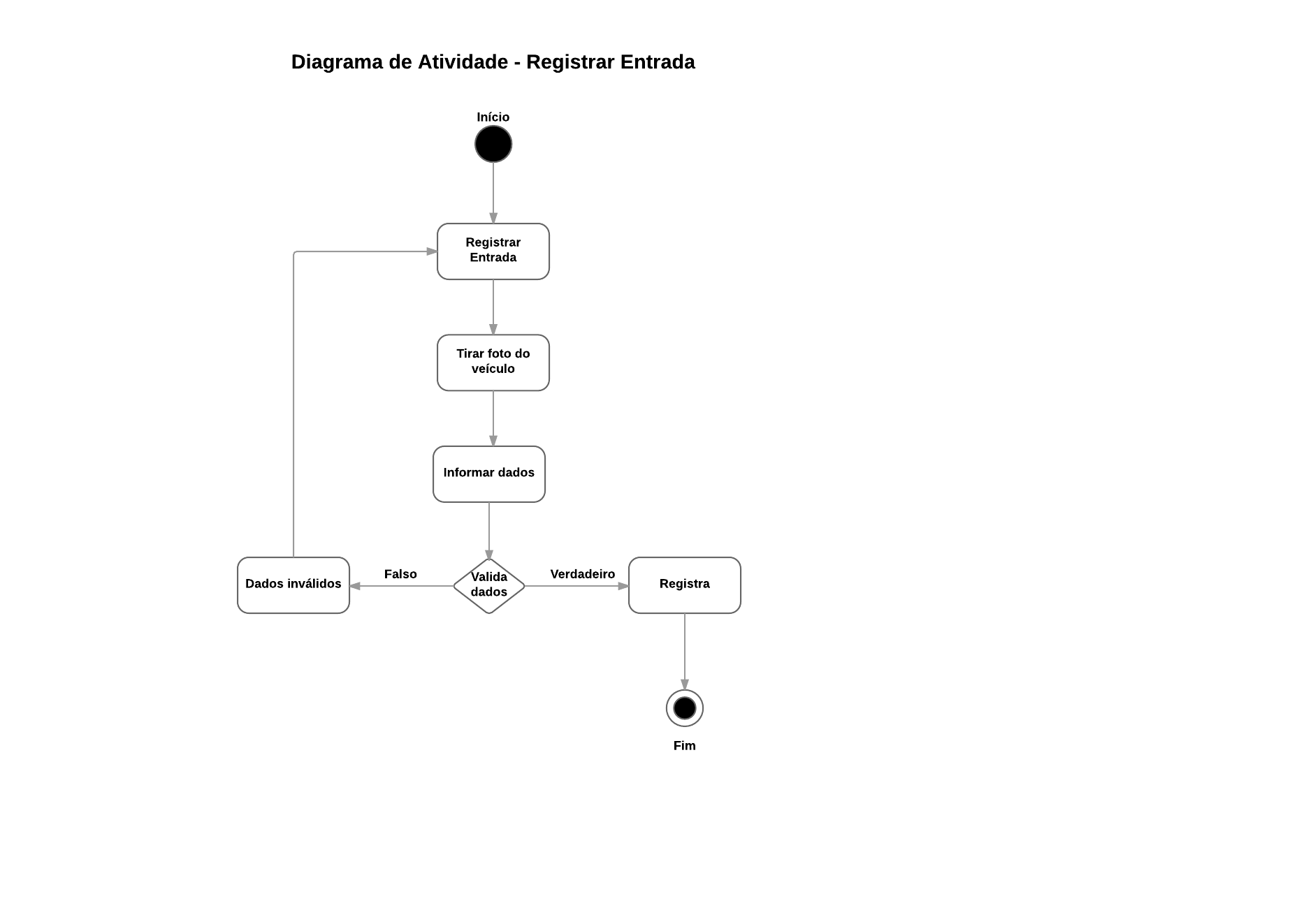
Diagrama de Sequência – Registrar Saída



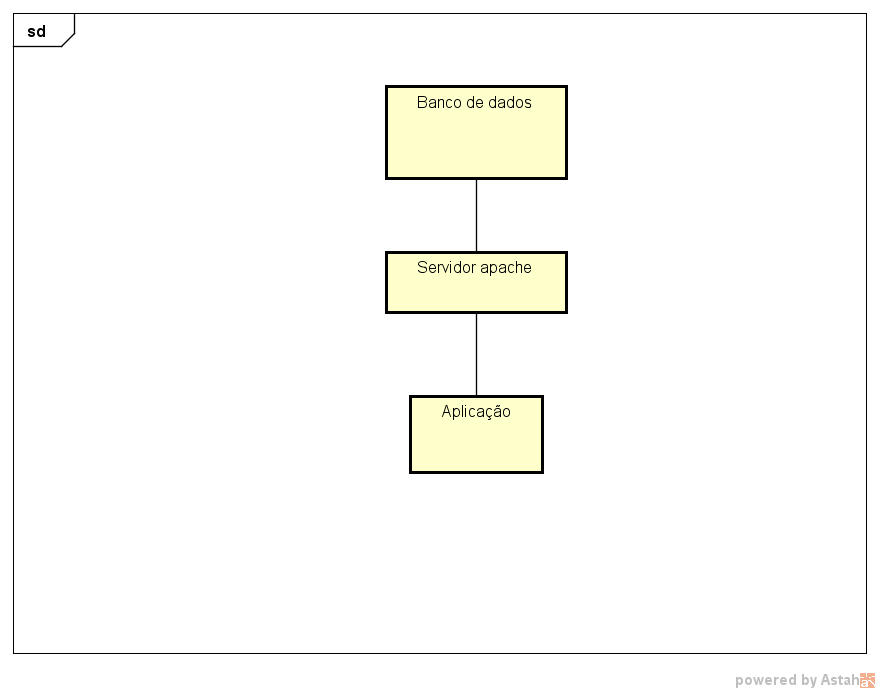
# APÊNDICES E - DIAGRAMAS DE ATIVIDADES







# APÊNDICES F - DIAGRAMA DE IMPLANTAÇÃO



# APÊNDICES G - DIAGRAMA DE CLASSES

**Diagrama de Classe**

# APÊNDICES H - MER FÍSICO E DICIONÁRIO DE DADOS

**Modelo Entidade Relacionamento**

# bancoStacione

### Dicionário de dados

Entidade: Perfil

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TB\_PERFIL** | | | | | | | | | |
| Tabela para guardar os tipos de perfis | | | | | | | | | |
| **Coluna** | **Tipo** | **Tamanho** | **PK** | | **FK** | | **Obrigatório** | | **Descrição** |
| Sim | Não | Sim | Não | Sim | Não |
| ID\_PERFIL | Int |  | S |  |  |  | S |  | Código do perfil |
| NM\_PERFIL | varchar | 60 |  |  |  |  | S |  | Nome do perfil |

Entidade: Funcionário

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TB\_FUNCIONARIO** | | | | | | | | | |
| Tabela que guarda os dados do funcionário | | | | | | | | | |
| **Coluna** | **Tipo** | **Tamanho** | **PK** | | **FK** | | **Obrigatório** | | **Descrição** |
| Sim | Não | Sim | Não | Sim | Não |
| ID\_FUNCIONARIO | int |  | S |  |  |  | S |  | Código do Funcionário |
| ID\_PERFIL | INTEGER |  |  |  | S |  | S |  | Código do perfil |
| NM\_FUNCIONARIO | varchar(255) | 255 |  |  |  |  | S |  | Nome do Funcionário |
| CPF\_FUNCIONARIO | varchar(14) | 14 |  |  |  |  | S |  | Cpf do Funcionário |
| RG\_FUNCIONARIO | varchar(30) | 30 |  |  |  |  |  |  | Rg do do Funcionário |
| DT\_NASCIMENTO | datetime |  |  |  |  |  |  |  | Data de nascimento do Funcionário |
| LOGIN | varchar(60) | 60 |  |  |  |  | S |  | Login do Funcionárioe |
| TELEFONE | varchar(14) | 14 |  |  |  |  |  |  | Telefone do Funcionário |
| SENHA | varchar(255) | 255 |  |  |  |  | S |  | Senha do Funcionário |

Entidade: Entrada/Saída

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TB\_ENTRADA\_SAIDA** | | | | | | | | | |
| Tabela para guardar a entrada e saída do veiculo | | | | | | | | | |
| **Coluna** | **Tipo** | **Tamanho** | **PK** | | **FK** | | **Obrigatório** | | **Descrição** |
| Sim | Não | Sim | Não | Sim | Não |
| ID\_ENTRADA | integer |  | S |  | S |  | S |  | Código de entrada |
| PLACA | Varchar | 9 |  |  |  |  |  |  | Placa do veiculo |
| ENTRADA | timestamp |  |  |  |  |  | S |  | Hora de entrada |
| SAIDA | timestamp |  |  |  |  |  |  |  | Hora de saída |
| ID\_VEICULO | integer |  |  |  |  |  |  |  | Código do Veiculo |
| ID\_IMAGEM | INT |  |  |  |  |  | S |  | Imagem de entrada |
| ID\_FUNCIONARIO | int |  |  |  |  |  | S |  | Código do Funcionário |

Entidade: Veículo

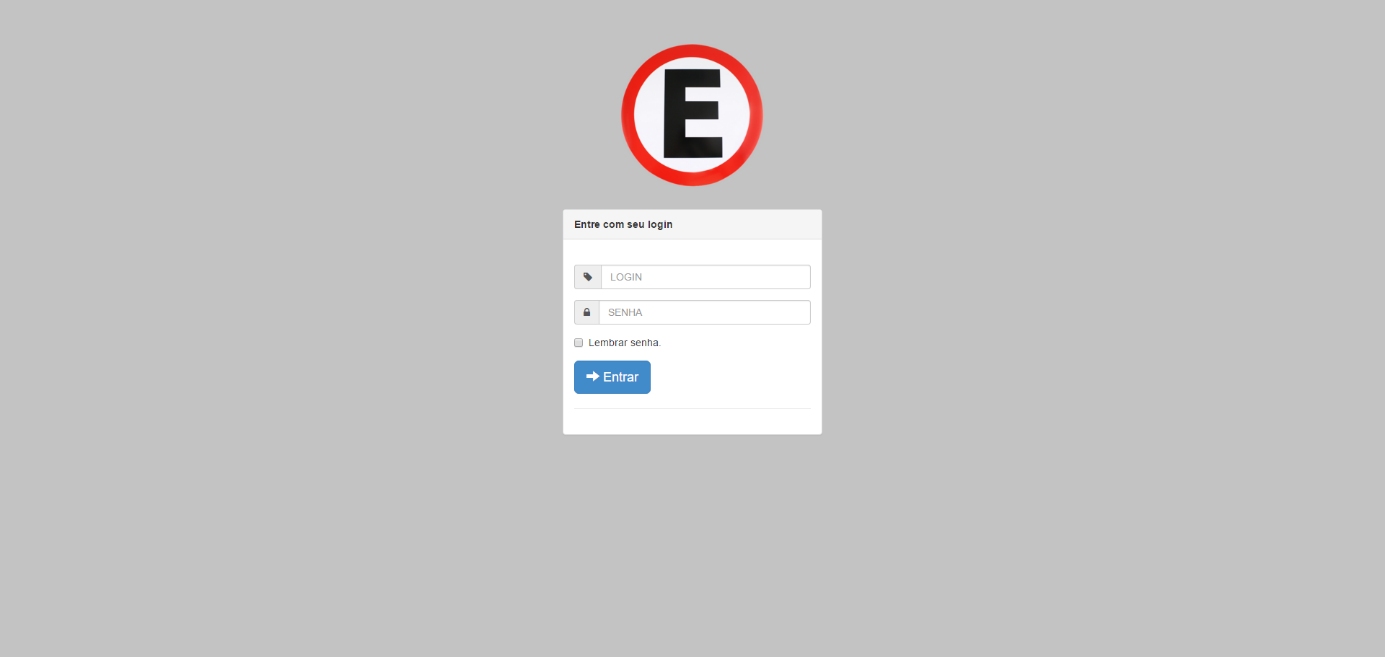
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TB\_VEICULO** | | | | | | | | | |
| Tabela para guardar os dados dos veículos | | | | | | | | | |
| **Coluna** | **Tipo** | **Tamanho** | **PK** | | **FK** | | **Obrigatório** | | **Descrição** |
| Sim | Não | Sim | Não | Sim | Não |
| ID\_VEICULO | varchar | 100 | S |  |  |  | S |  | ID\_VEICULO |
| NOME\_VEICULO | varchar | 255 |  |  | S |  | S |  | Nome do veiculo |
| ID\_FABRICANTE | varchar(100) | 100 |  |  | S |  | S |  | Código do fabricante |
| TIPO\_VEICULO | varchar(8) | 8 |  |  |  |  |  |  | Tipo de veiculo |

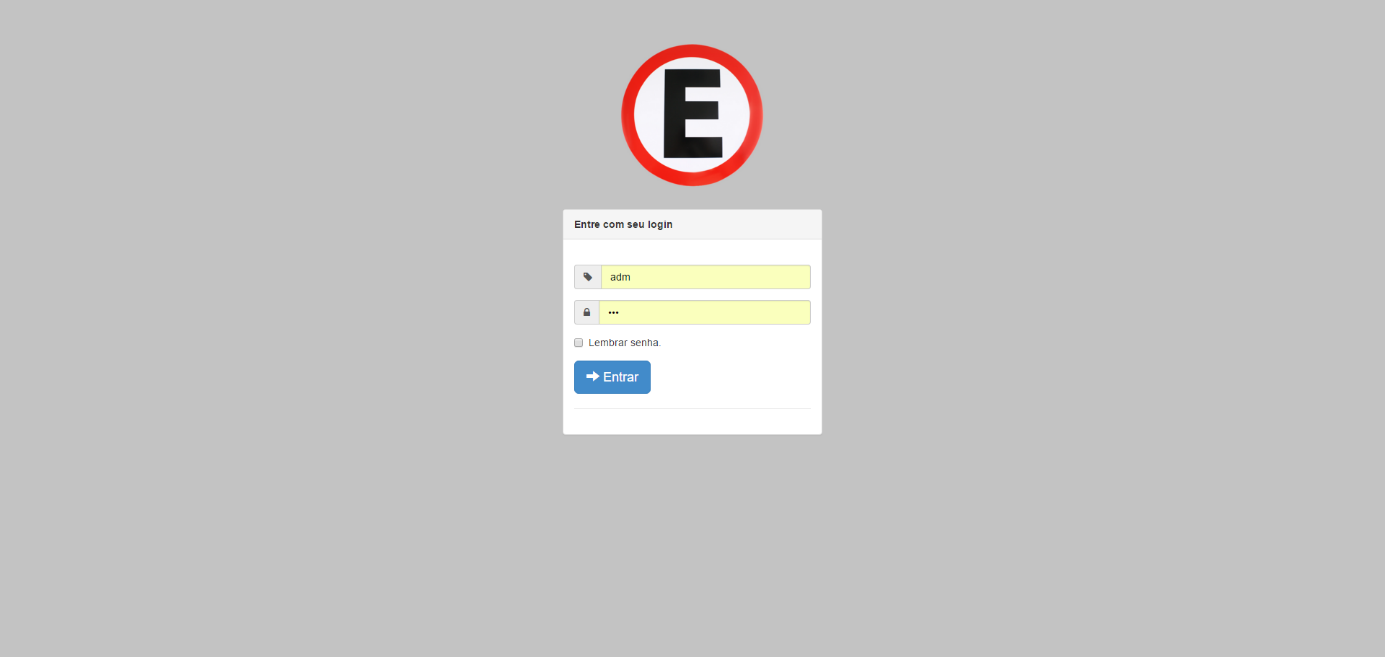
Entidade: TB\_IMAGEM

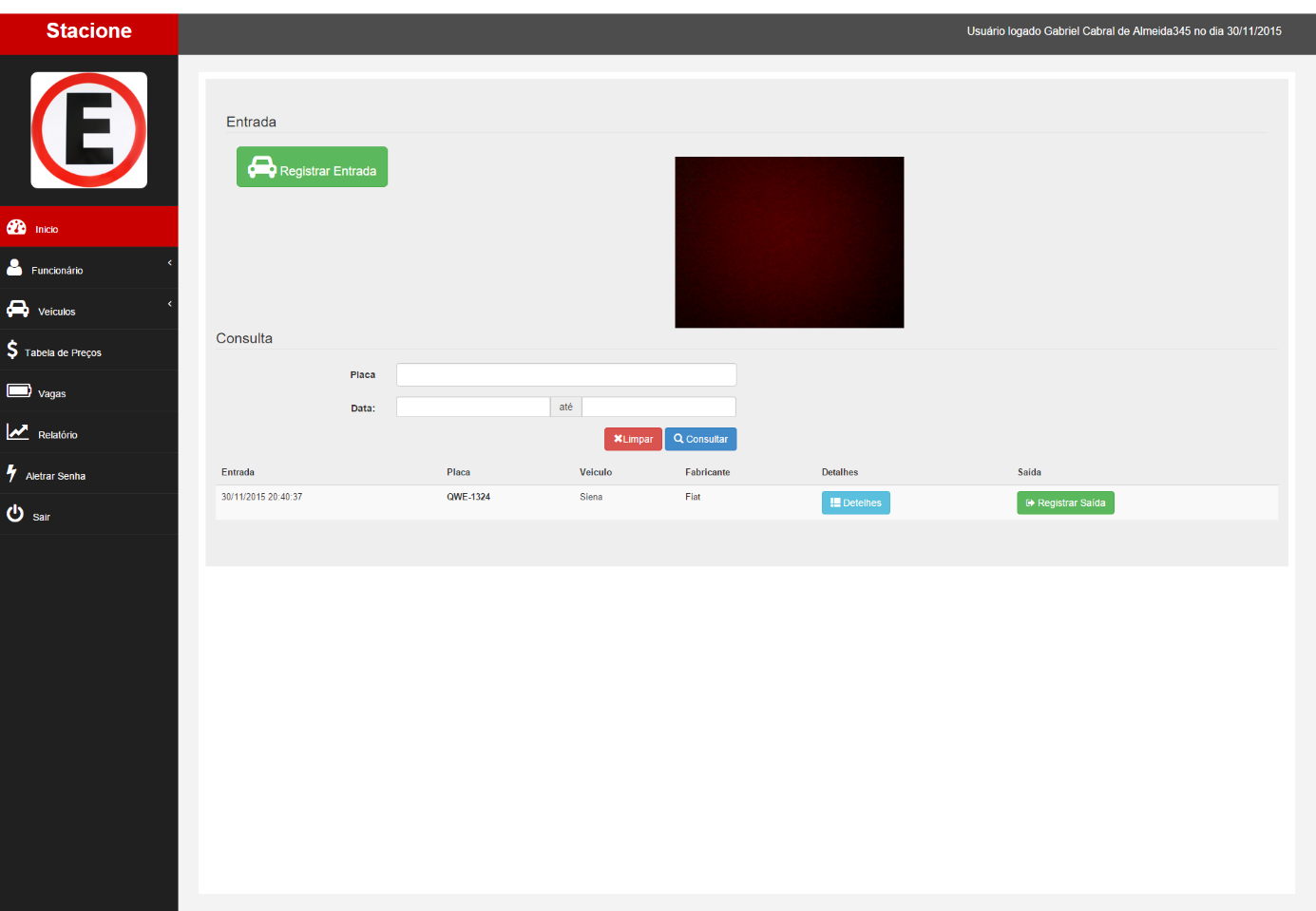
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TB\_IMAGEM** | | | | | | | | | |
| Tabela para guardar as informações da imagem | | | | | | | | | |
| **Coluna** | **Tipo** | **Tamanho** | **PK** | | **FK** | | **Obrigatório** | | **Descrição** |
| Sim | Não | Sim | Não | Sim | Não |
| ID\_IMAGEM | Int | 11 | S |  |  |  | S |  | Código da imagem |
| IMAGEM | varchar | 255 |  |  |  |  | S |  | Nome da imagem |

# APÊNDICES I - TESTE DE QUALIDADE

|  |  |
| --- | --- |
| **Nome: Login** | **Executor de Teste: Gabriel Cabral de Almeida** |

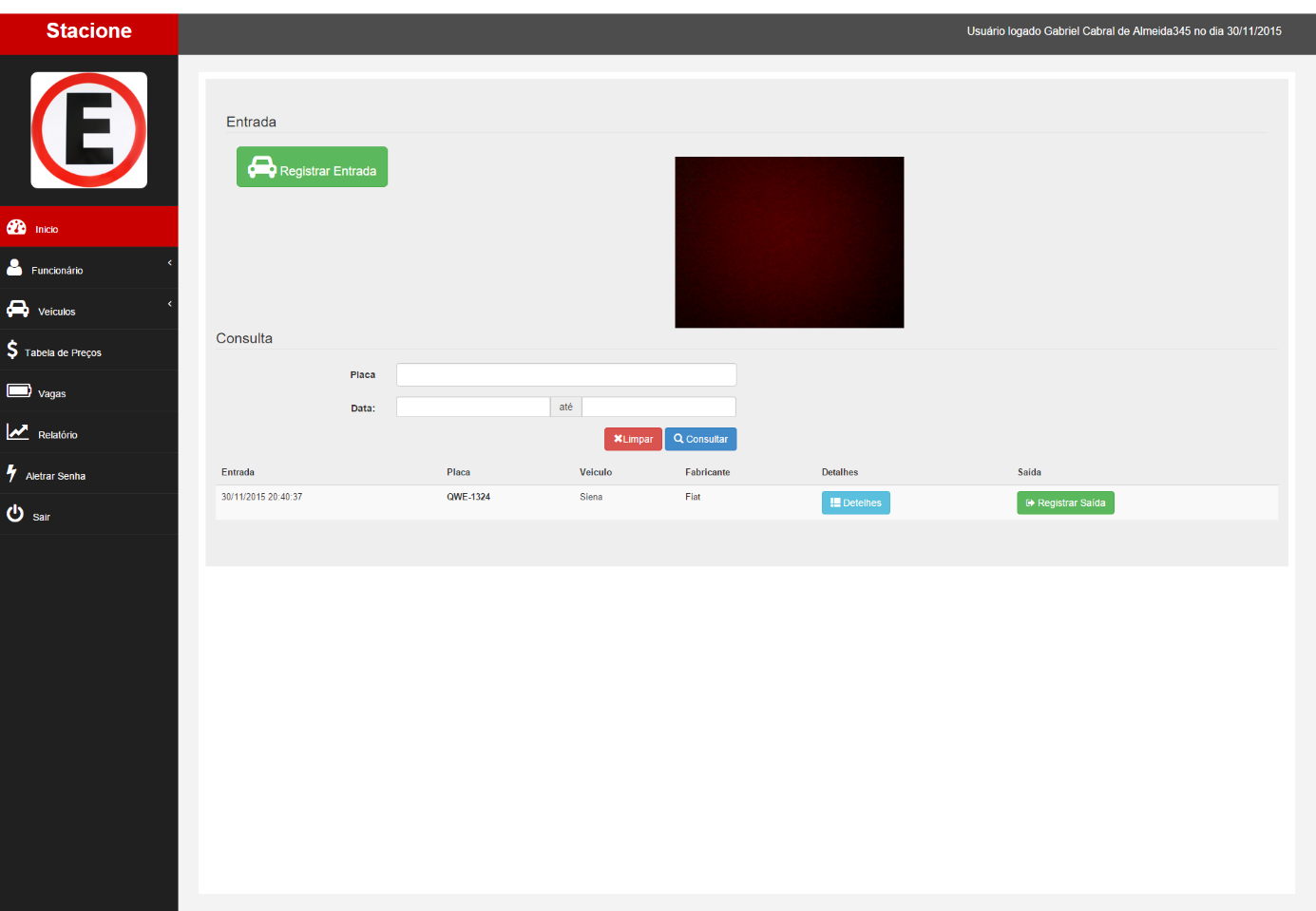
**1o Passo – Acesso ao endereço do sistema**



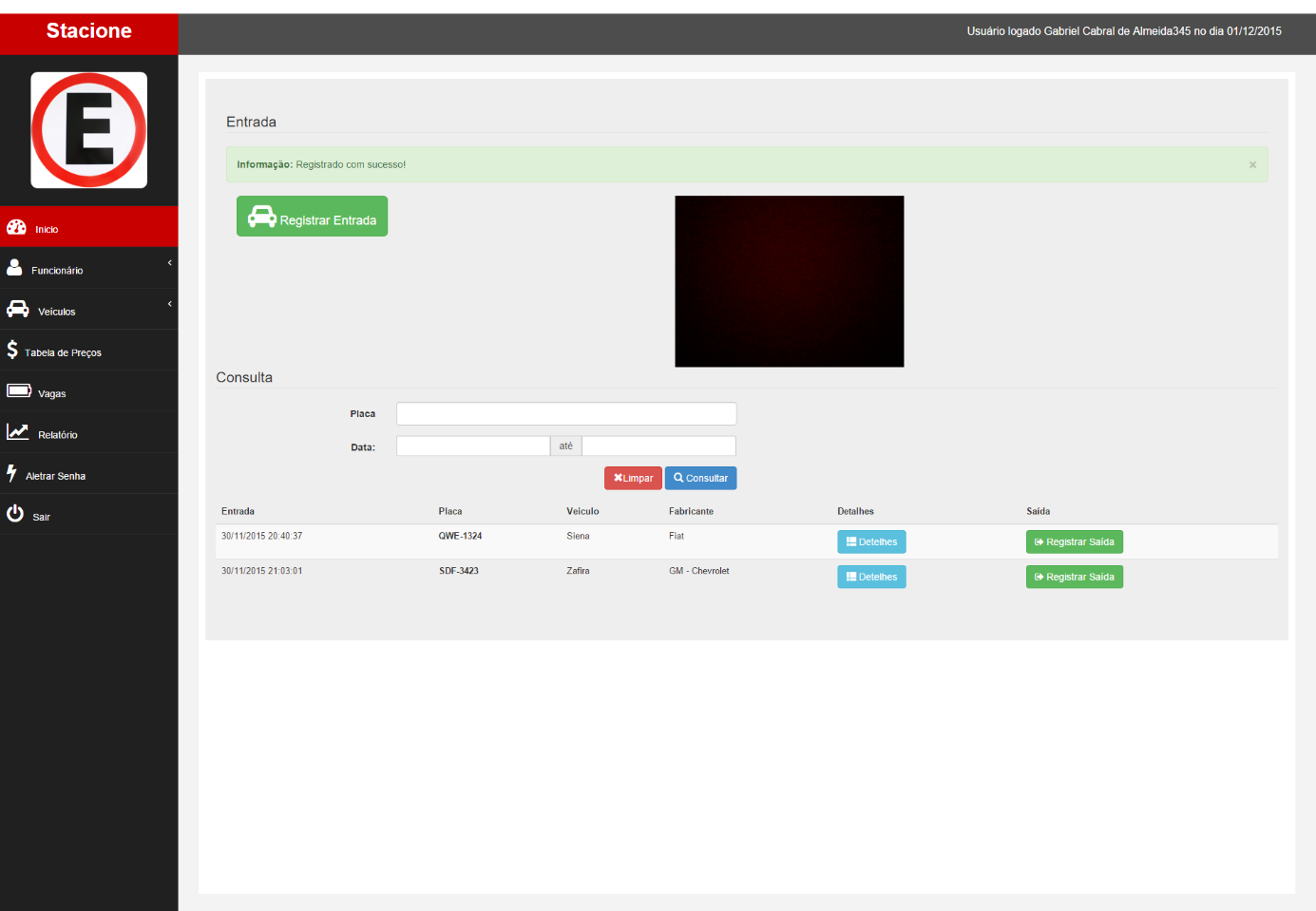
**2o Passo – Login e Senha (Clicar em Entrar)  
3o Passo – Acesso ao sistema  
**

|  |  |
| --- | --- |
| **Nome: Entrada no Estacionamento** | **Executor de Teste: Gabriel Cabral de Almeida** |

**1o Passo – Acesso a dar entrada no veículo (Clicar em Entrada)**

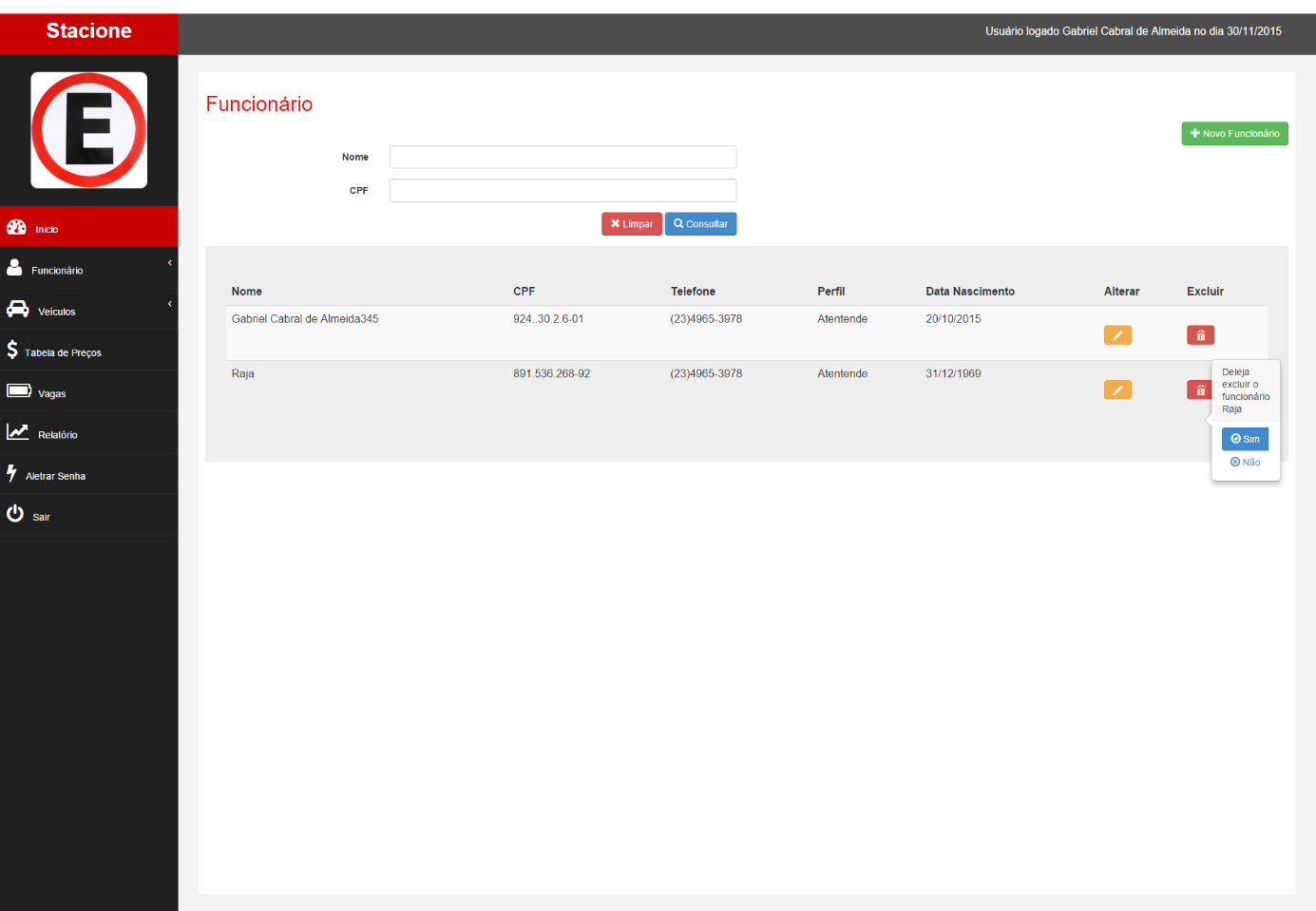
****

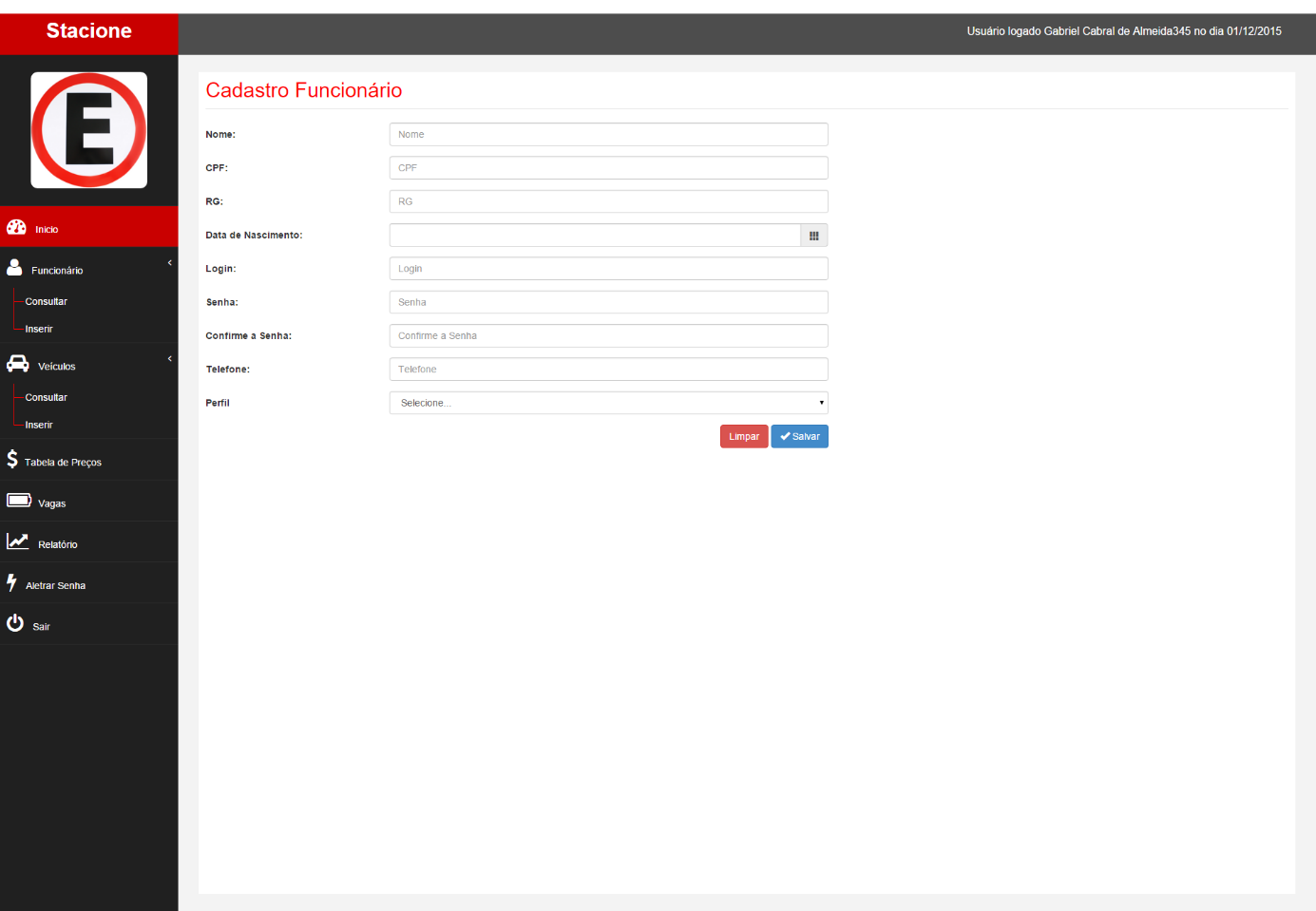
**2o Passo – inserir os dados do veículo (Clicar em salvar)  
**

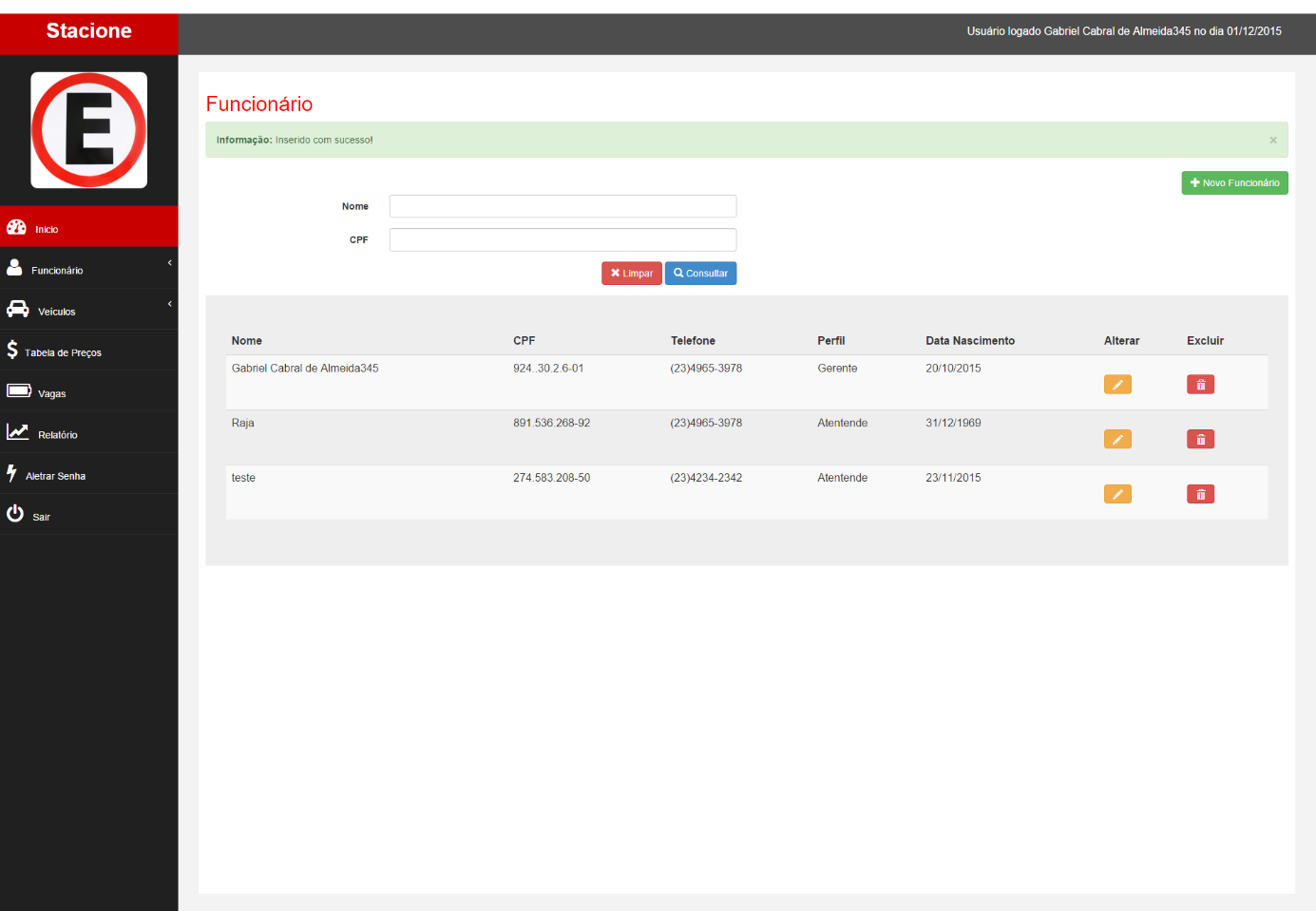
**3o Passo – retorna para o fluxo de entrada  
**

|  |  |
| --- | --- |
| **Nome: Manter funcionário** | **Executor de Teste: Gabriel Cabral de Almeida** |

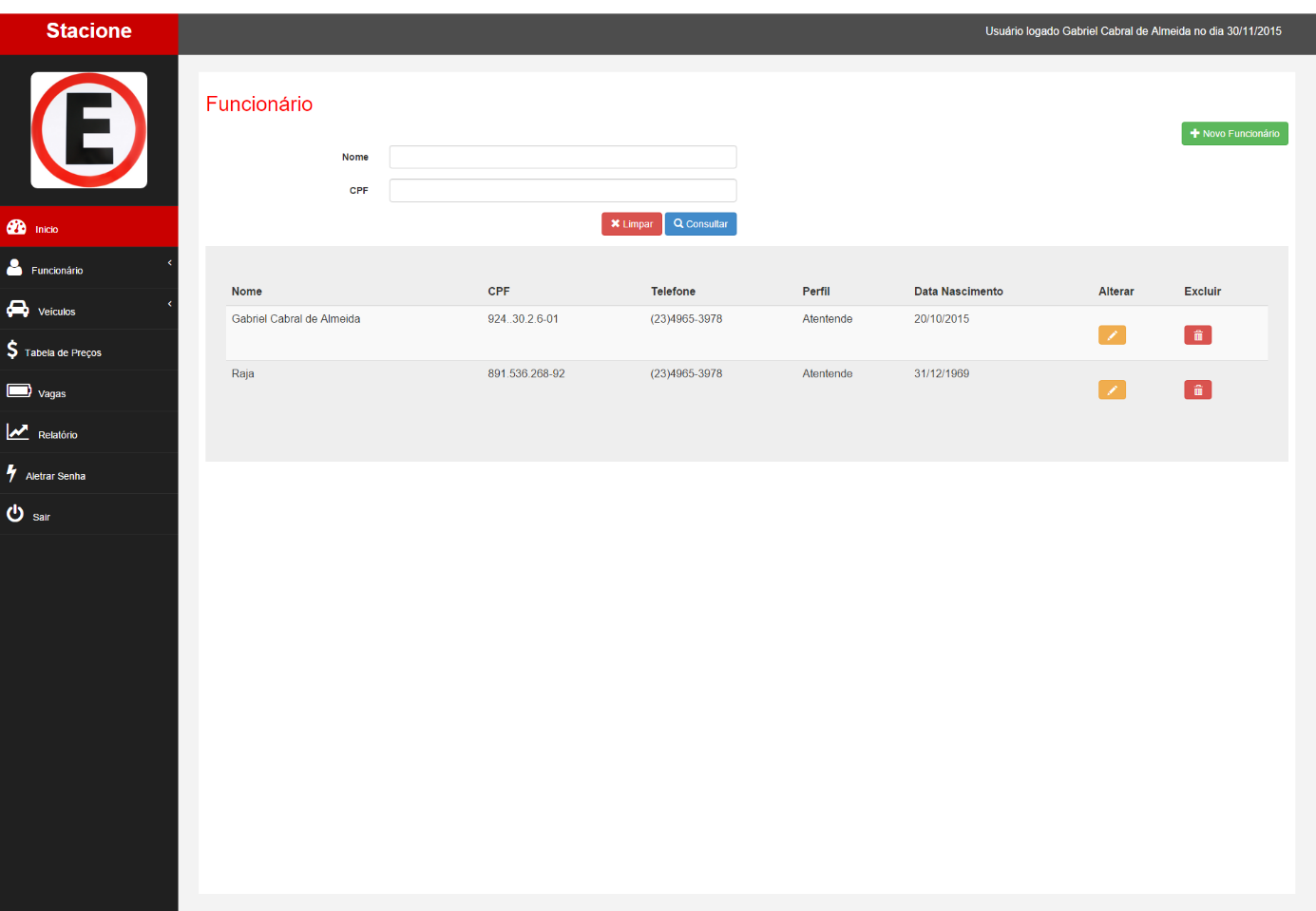
**1o Passo – Acesso a cadastro de funcionário (Clicar em Funcionário e inserir)**

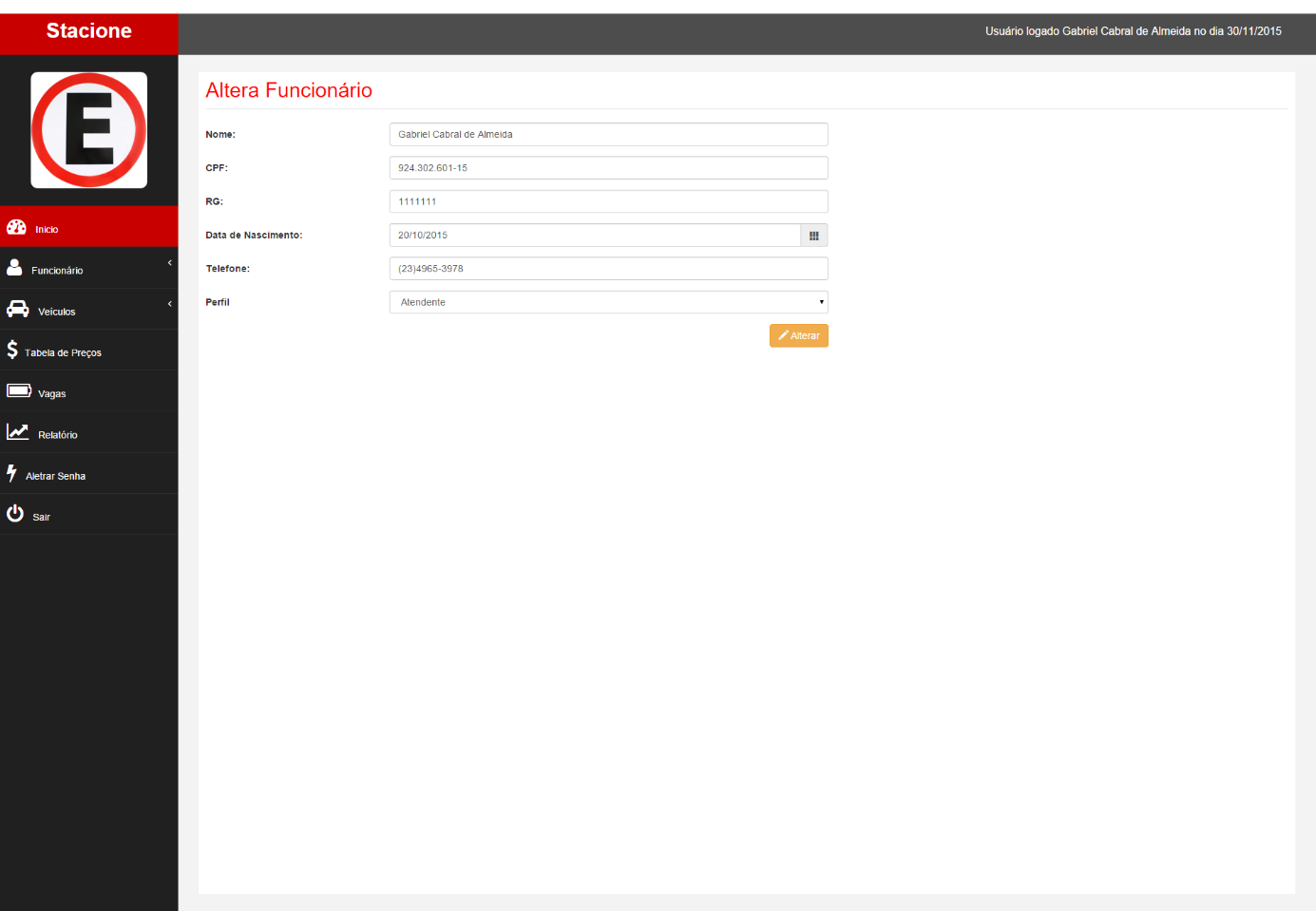
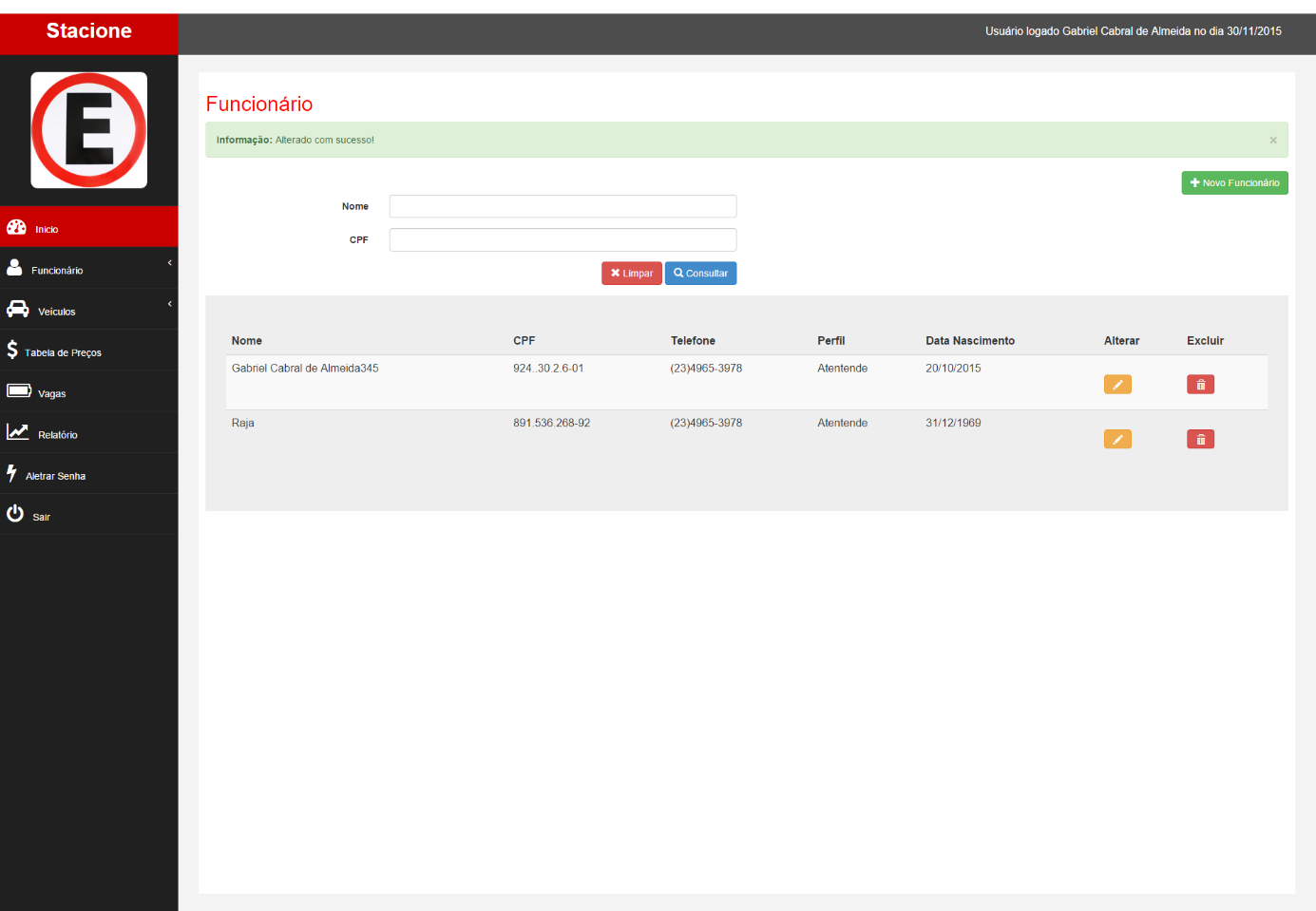


**2o Passo – inserir os dados do Funcionário (Clicar em salvar)  
**

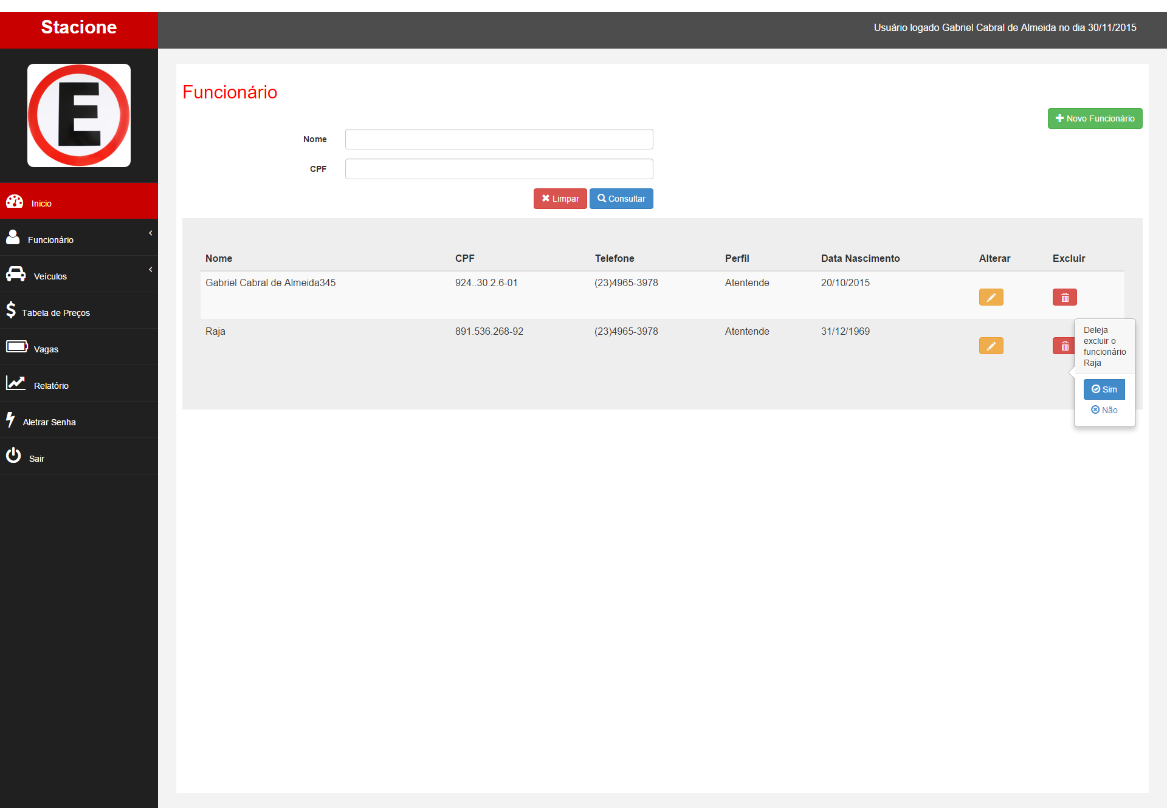
**3o Passo – Retorna pata o fluxo pesquisar Funcionário  
**

**4o Passo – Acesso a alterar funcionário (Clicar em Funcionário e na lista clicar em alterar)**



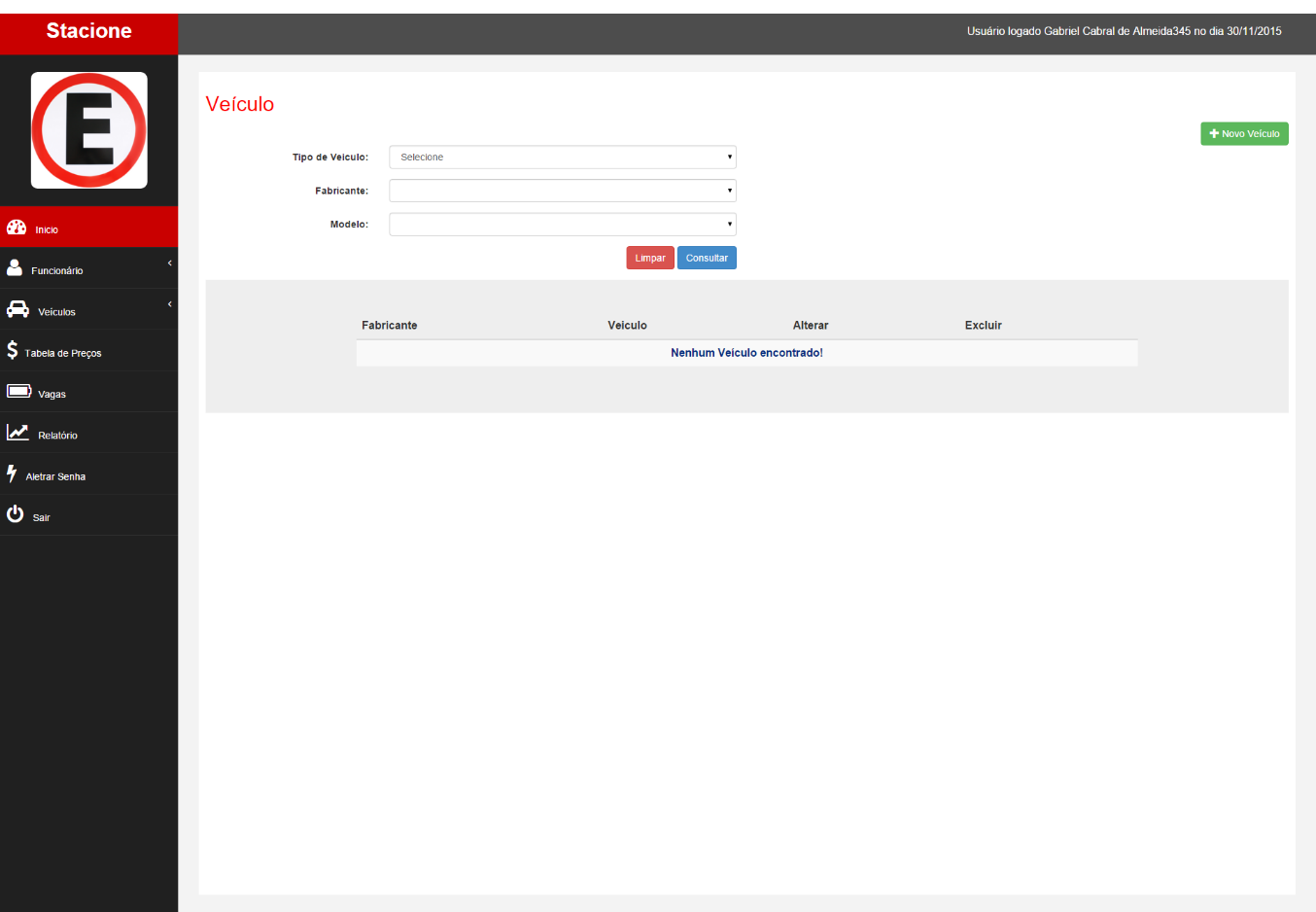
**5o Passo – Alterar os dados do Funcionário (Clicar em alterar)  
  
6o Passo – Retorna pata o fluxo pesquisar Funcionário  
**

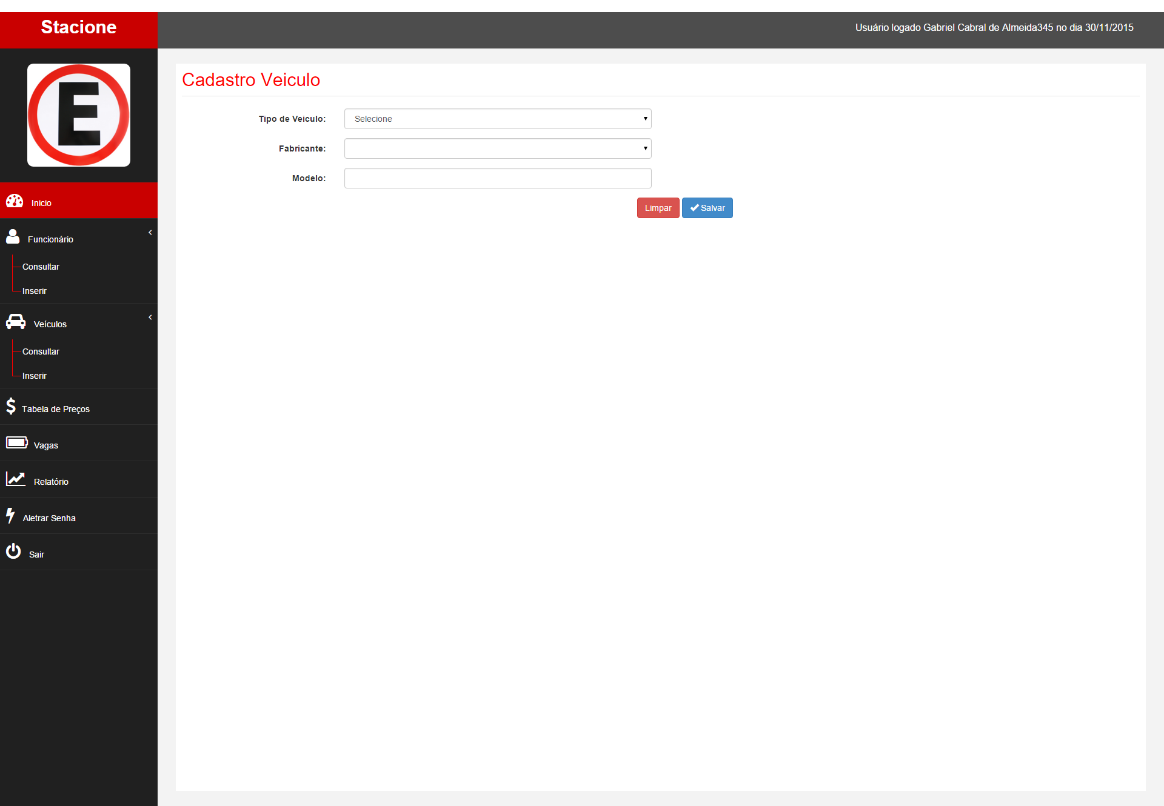
**7o Passo – Confirmar a Exclusão funcionário (Clicar em Funcionário e na lista clicar em SIM)**

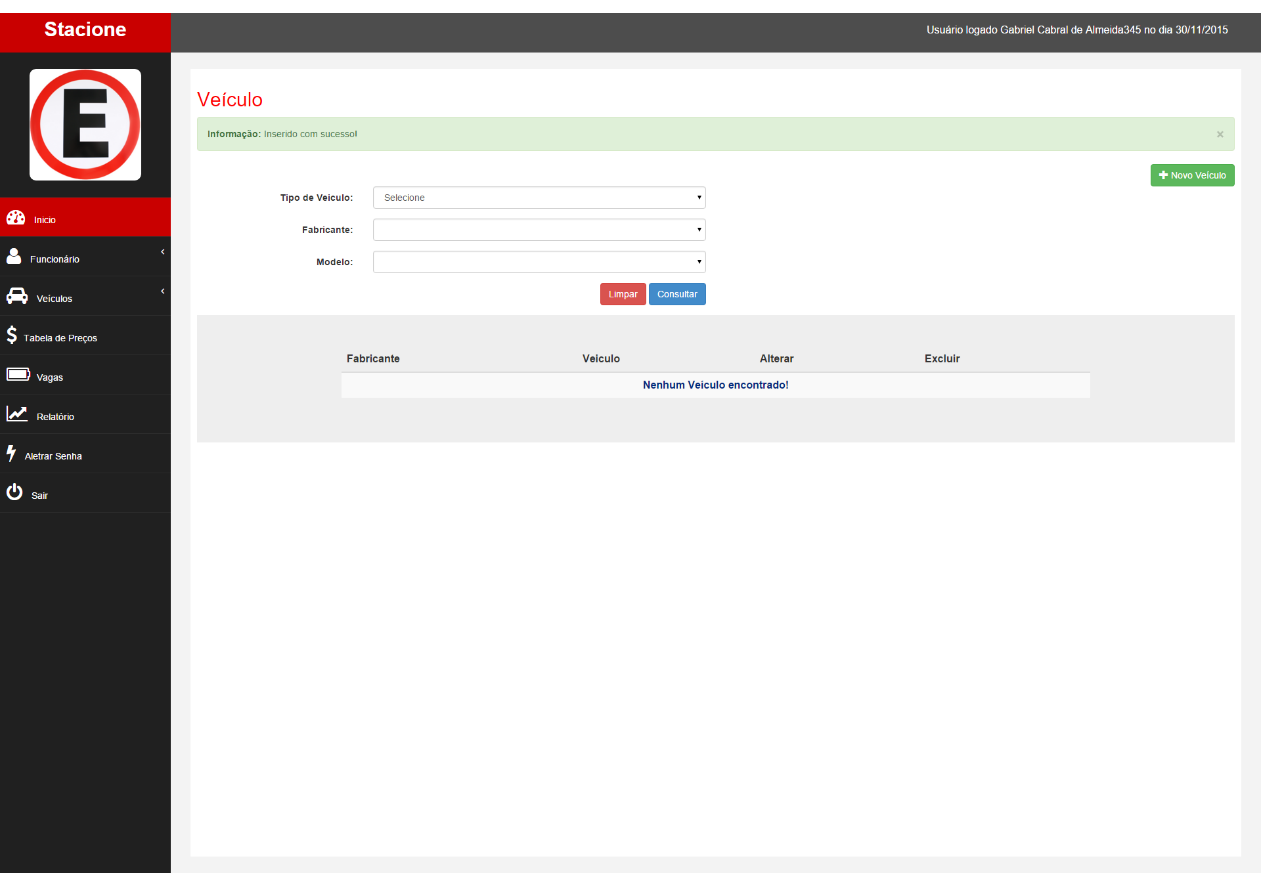
****

|  |  |
| --- | --- |
| **Nome: Manter veiculo** | **Executor de Teste: Gabriel Cabral de Almeida** |

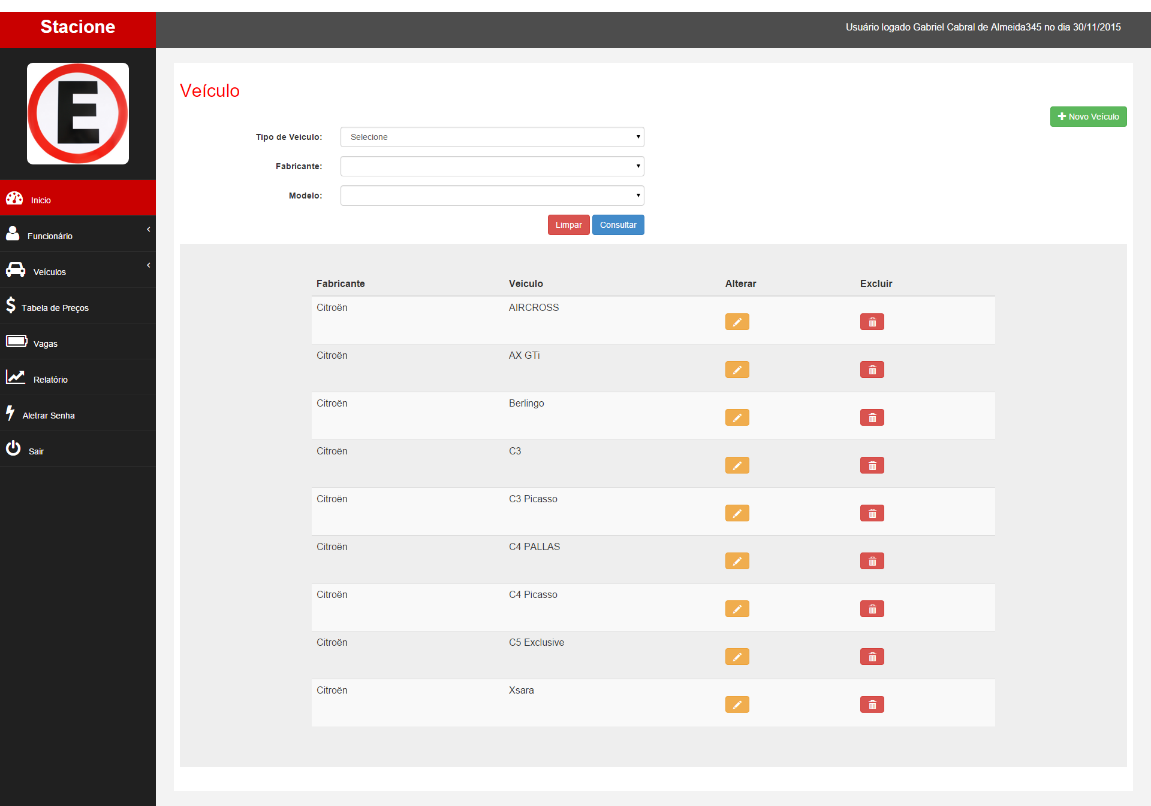
**1o Passo – Acesso a cadastro de veículos (Clicar em veículos e inserir)**

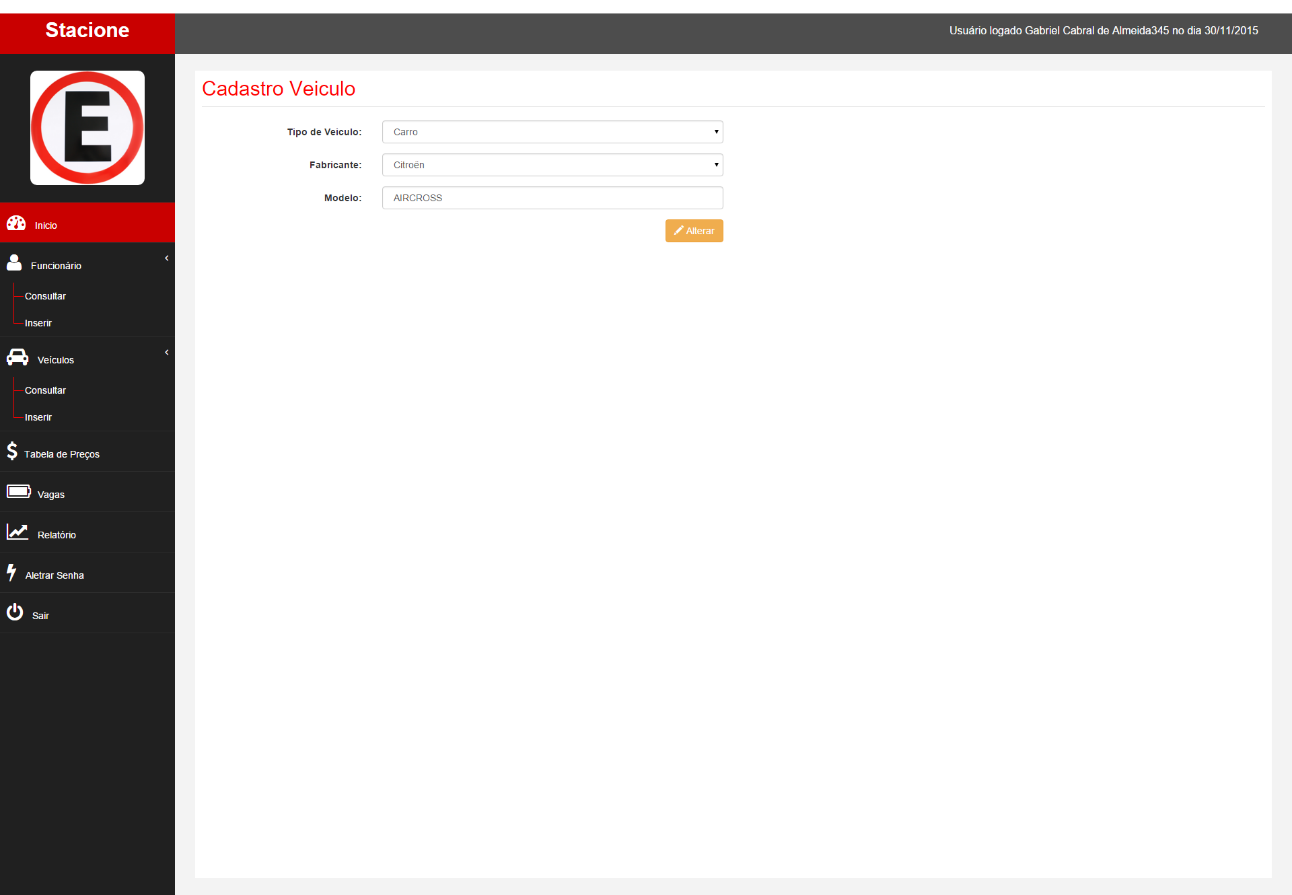


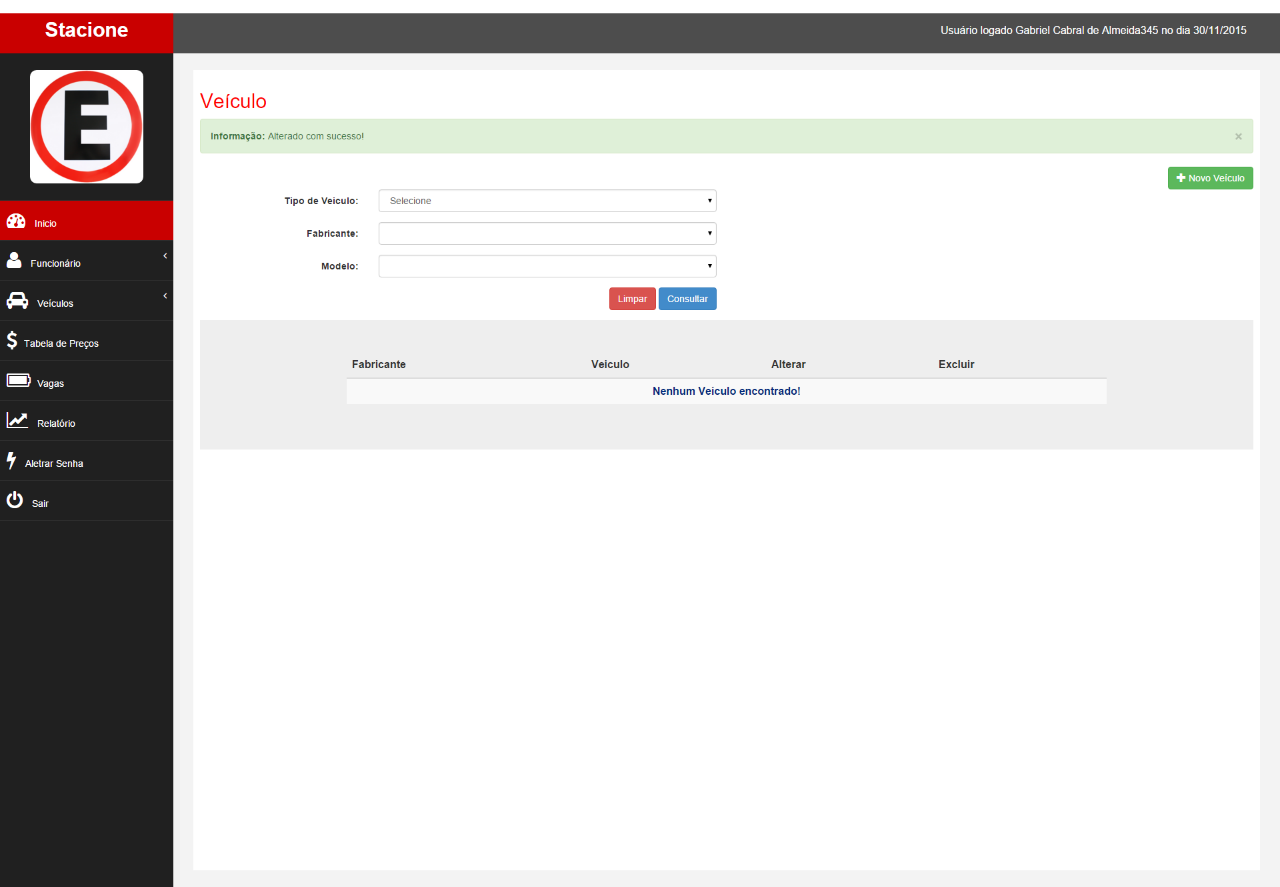
**2o Passo – inserir os dados do veículos (Clicar em salvar)  
**

**3o Passo – Retorna pata o fluxo pesquisar veículos  
**

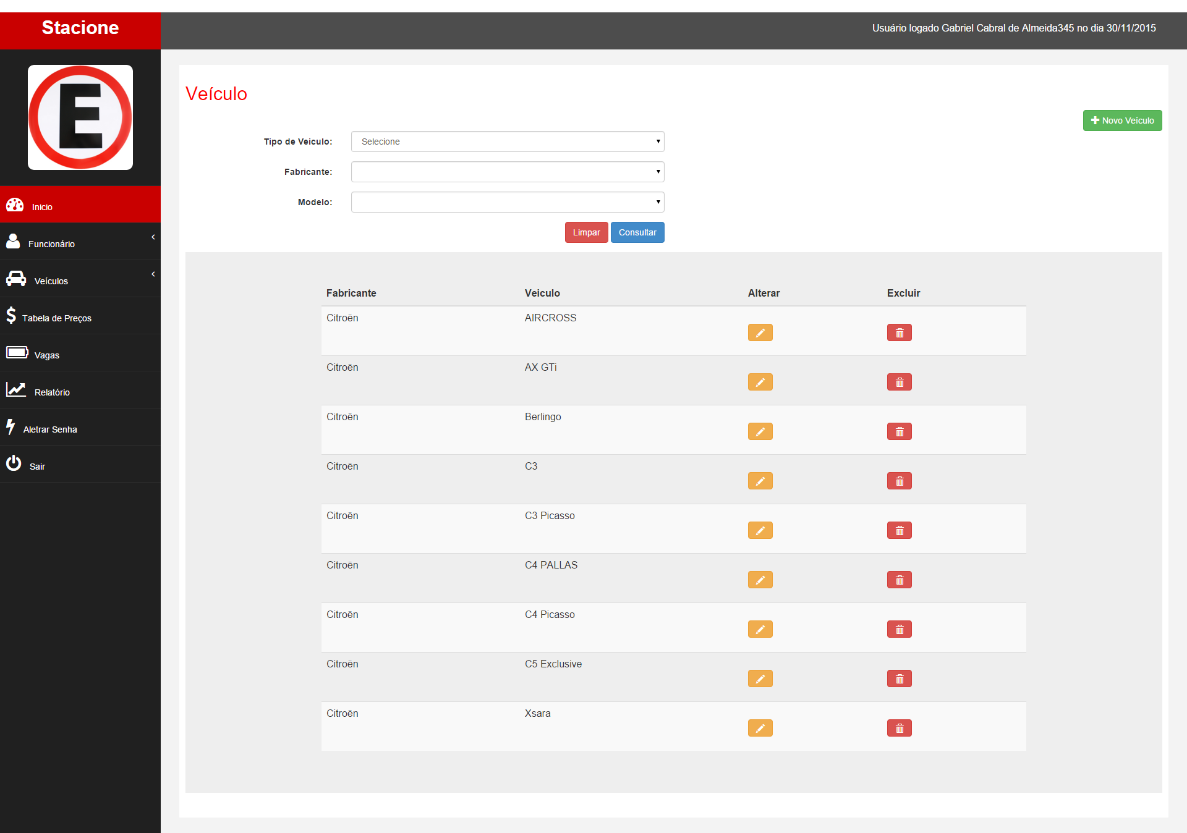
**4o Passo – Acesso a alterar veículos (Clicar em veículos e na lista clicar em alterar)**



**5o Passo – Alterar os dados do veículos (Clicar em alterar)  
**

**6o Passo – Retorna pata o fluxo pesquisar veículos  
**

**7o Passo – Acesso a Excluir veículos (Clicar em veículos e na lista clicar em Excluir)**



**8o Passo – Confirmar a Exclusão veículos (Clicar em veículos e na lista clicar em SIM)**

# 

# APÊNDICES J – CD/DVD COM IMPLEMENTAÇÃO

# APÊNDICES L – ROTEIRO DE TESTE

***Roteiro de Teste – Efetuar Login***

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Teste:** Funcionalidade | | **Nome da Funcionalidade:** Efetuar Login | |
| **Data Início:** 27/11/2015 | **Data Fim:**  27/11/2015 | | **Tempo Despendido (h):**  1 Horas |
| **Pré-Condição:** O usuário deve acessar o sistema com perfis de funcionário ou administrador | | | |

|  |
| --- |
| **Localização:** Acessar o sistema |

|  |  |  |
| --- | --- | --- |
| **Requisito: caso de uso** | | |
| **Objeto de Teste:**   1. Validar funcionalidade. 2. Validar interface. 3. Validar mensagem da funcionalidade. | | |
| **Descrição do Caso de Teste:**   1. O sistema deverá controlar a inserção de dados na interface. 2. O sistema deverá informar uma mensagem quando dados estiveram incompletos ou incorretos 3. O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for logado | | |
| **Pré-Condição**:  ID 1 - Funcionário, deve ter o perfil atendente;  ID 2 – Gerente deve ter o perfil de gerente; | | |
| **Dados de Entrada: Dados do funcionário** | | |
| **ID** | **Passo** | **Procedimento** |
| 1/2 | P1 | Acessa do sistema. |
| V1 | Deverá exibir a tela principal do sistema |
| 1/2 | P2 | Deverá exibir formulário para inserir os dados |
| 1/2 | P3 | Após informar, selecionar a opção. Entrar. |
| 1/2 | V2 | Deverá validar as informações inseridas e conceder acesso a tela principal do sistema. |
| **Resultado Esperado:** Devem ser cumpridas todas as regras licitadas acima. | | |

***Roteiro de Teste – Manter funcionário***

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Teste:** Funcionalidade | | **Nome da Funcionalidade:** Funcionalidade | |
| **Data Início:** 27/11/2015 | **Data Fim:**  27/11/2015 | | **Tempo Despendido (h):**  1 Horas |
| **Pré-Condição:** O usuário deve acessar o sistema com perfil de administrador | | | |

|  |
| --- |
| **Localização:** Menu |

|  |  |  |
| --- | --- | --- |
| **Requisito: caso de uso** | | |
| **Objeto de Teste:**   1. Validar funcionalidade. 2. Validar interface. 3. Validar mensagem da funcionalidade. | | |
| **Descrição do Caso de Teste:**  **1-** O sistema deverá controlar a inserção de dados do cliente na interface. **2-** O sistema deverá informar uma mensagem quando os dados estiverem incorretos. **3-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for inserido um funcionário. **4-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for alterado um funcionário. **5-** O sistema deverá limpar da base de dados e gerar a mensagem informativa quando for excluído um funcionário. **6-** O sistema deverá informar uma mensagem quando não forem informados os dados obrigatórios. | | |
| **Pré-Condição**:  ID 2 – Gerente deve ter o perfil de gerente; | | |
| **Dados de Entrada: Dados do funcionário** | | |
| **ID** | **Passo** | **Procedimento** |
| 2 | P1 | Acessa do sistema. |
| V1 | Deverá exibir a tela principal do sistema |
| P2 | Acessar o menu funcionário |
| V2 | Deverá exibir a tela de Funcionário |
| P3 | Para realizar pesquisas informe os dados necessários e clique consultar |
| V3 | Deverá aparecer a tabela de resultados. |
| P4 | Para inserir cliente deverá clicar no Novo Funcionário |
| V4 | Deverá exibir formulário para inserir os dados |
| P5 | Após informar dados do cliente, selecionar a opção inserir. |
| V5 | Deverá exibir mensagens de cliente inserido com sucesso, salvar dados e exibir tela de funcionário |
| P6 | Para excluir ou alterar cliente, deve-se pesquisar o mesmo e selecionar o respectivo ícone na última coluna (alterar/excluir) |
| V6 | Deverá exibir mensagens de cliente alterado ou excluído com sucesso e salvar ou excluir dados. |
| **Resultado Esperado:** Devem ser cumpridas todas as regras licitadas acima. | | |

***Roteiro de Teste – Manter veículo***

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Teste:** Funcionalidade | | **Nome da Funcionalidade:** Funcionalidade | |
| **Data Início:** 27/11/2015 | **Data Fim:**  27/11/2015 | | **Tempo Despendido (h):**  1 Horas |
| **Pré-Condição:** O usuário deve acessar o sistema com perfil de administrador | | | |

|  |
| --- |
| **Localização:** Menu |

|  |  |  |
| --- | --- | --- |
| **Requisito: caso de uso** | | |
| **Objeto de Teste:**   1. Validar funcionalidade. 2. Validar interface. 3. Validar mensagem da funcionalidade. | | |
| **Descrição do Caso de Teste:**  **1-** O sistema deverá controlar a inserção de dados do cliente na interface. **2-** O sistema deverá informar uma mensagem quando os dados estiverem incorretos. **3-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for inserido um veículo. **4-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for alterado um veículo. **5-** O sistema deverá limpar da base de dados e gerar a mensagem informativa quando for excluído um veículo. **6-** O sistema deverá informar uma mensagem quando não forem informados os dados obrigatórios. | | |
| **Pré-Condição**:  ID 2 – Gerente deve ter o perfil de gerente; | | |
| **Dados de Entrada: Dados do funcionário** | | |
| **ID** | **Passo** | **Procedimento** |
| 2 | P1 | Acessa do sistema. |
| V1 | Deverá exibir a tela principal do sistema |
| P2 | Acessar o menu veículo |
| V2 | Deverá exibir a tela de veículo |
| P3 | Para realizar pesquisas informe os dados necessários e clique consultar |
| V3 | Deverá aparecer a tabela de resultados. |
| P4 | Para inserir veículo deverá clicar no Novo veículo |
| V4 | Deverá exibir formulário para inserir os dados |
| P5 | Após informar dados do veículo, selecionar a opção inserir. |
| V5 | Deverá exibir mensagens de veículo inserido com sucesso, salvar dados e exibir tela de veículo |
| P6 | Para excluir ou alterar veículo, deve-se pesquisar o mesmo e selecionar o respectivo ícone na última coluna (alterar/excluir) |
| V6 | Deverá exibir mensagens de veículo alterado ou excluído com sucesso e salvar ou excluir dados. |
| **Resultado Esperado:** Devem ser cumpridas todas as regras licitadas acima. | | |

***Roteiro de Teste – Manter Vagas***

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Teste:** Funcionalidade | | **Nome da Funcionalidade:** Funcionalidade | |
| **Data Início:** 27/11/2015 | **Data Fim:**  27/11/2015 | | **Tempo Despendido (h):**  1 Horas |
| **Pré-Condição:** O usuário deve acessar o sistema com perfil de administrador | | | |

|  |
| --- |
| **Localização:** Menu |

|  |  |  |
| --- | --- | --- |
| **Requisito: caso de uso** | | |
| **Objeto de Teste:**   1. Validar funcionalidade. 2. Validar interface. 3. Validar mensagem da funcionalidade. | | |
| **Descrição do Caso de Teste:**  **1-** O sistema deverá controlar a inserção de dados do cliente na interface. **2-** O sistema deverá informar uma mensagem quando os dados estiverem incorretos. **3-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for inserido um veículo. **4-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for alterado um veículo. **5-** O sistema deverá limpar da base de dados e gerar a mensagem informativa quando for excluído um veículo. **6-** O sistema deverá informar uma mensagem quando não forem informados os dados obrigatórios. | | |
| **Pré-Condição**:  ID 2 – Gerente deve ter o perfil de gerente; | | |
| **Dados de Entrada: Dados do funcionário** | | |
| **ID** | **Passo** | **Procedimento** |
| 2 | P1 | Acessa do sistema. |
| V1 | Deverá exibir a tela principal do sistema |
| P2 | Acessar o menu Vagas |
| V2 | Deverá exibir a tela de vagas |
| P3 | Deverá exibir formulário para alterar a vagas |
| P4 | Após informar dados, selecionar a opção alterar. |
| V4 | Deverá exibir mensagens de vagas alterado com sucesso |
| **Resultado Esperado:** Devem ser cumpridas todas as regras licitadas acima. | | |

***Roteiro de Teste – Manter preço***

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Teste:** Funcionalidade | | **Nome da Funcionalidade:** Funcionalidade | |
| **Data Início:** 27/11/2015 | **Data Fim:**  27/11/2015 | | **Tempo Despendido (h):**  1 Horas |
| **Pré-Condição:** O usuário deve acessar o sistema com perfil de administrador | | | |

|  |
| --- |
| **Localização:** Menu |

|  |  |  |
| --- | --- | --- |
| **Requisito: caso de uso** | | |
| **Objeto de Teste:**   1. Validar funcionalidade. 2. Validar interface. 3. Validar mensagem da funcionalidade. | | |
| **Descrição do Caso de Teste:**  **1-** O sistema deverá controlar a inserção de dados do cliente na interface. **2-** O sistema deverá informar uma mensagem quando os dados estiverem incorretos. **3-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for inserido um veículo. **4-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for alterado um veículo. **5-** O sistema deverá limpar da base de dados e gerar a mensagem informativa quando for excluído um veículo. **6-** O sistema deverá informar uma mensagem quando não forem informados os dados obrigatórios. | | |
| **Pré-Condição**:  ID 2 – Gerente deve ter o perfil de gerente; | | |
| **Dados de Entrada: Dados do funcionário** | | |
| **ID** | **Passo** | **Procedimento** |
| 2 | P1 | Acessa do sistema. |
| V1 | Deverá exibir a tela principal do sistema |
| P2 | Acessar o menu tabela de preço |
| V2 | Deverá exibir a tela de tabela de preço |
| P3 | Deverá exibir formulário para alterar a preço |
| P4 | Após informar dados, selecionar a opção alterar. |
| V4 | Deverá exibir mensagens de preço alterado com sucesso |
| **Resultado Esperado:** Devem ser cumpridas todas as regras licitadas acima. | | |

***Roteiro de Teste – Manter entrada***

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Teste:** Funcionalidade | | **Nome da Funcionalidade:** Funcionalidade | |
| **Data Início:** 27/11/2015 | **Data Fim:**  27/11/2015 | | **Tempo Despendido (h):**  1 Horas |
| **Pré-Condição:** O usuário deve acessar o sistema com perfil de administrador | | | |

|  |
| --- |
| **Localização:** Menu |

|  |  |  |
| --- | --- | --- |
| **Requisito: caso de uso** | | |
| **Objeto de Teste:**   1. Validar funcionalidade. 2. Validar interface. 3. Validar mensagem da funcionalidade. | | |
| **Descrição do Caso de Teste:**  **1-** O sistema deverá controlar a inserção de dados do cliente na interface. **2-** O sistema deverá informar uma mensagem quando os dados estiverem incorretos. **3-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for inserido um veículo. **4-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for alterado um veículo. **5-** O sistema deverá limpar da base de dados e gerar a mensagem informativa quando for excluído um veículo. **6-** O sistema deverá informar uma mensagem quando não forem informados os dados obrigatórios. | | |
| **Pré-Condição**:  ID 1 – Gerente deve ter o perfil de Atendente;  ID 2 – Gerente deve ter o perfil de gerente; | | |
| **Dados de Entrada: Dados do funcionário** | | |
| **ID** | **Passo** | **Procedimento** |
| 1/2 | P1 | Acessa do sistema. |
| V1 | Deverá exibir a tela principal do sistema |
| P2 | Selecionar a opção Registrar Entrada. |
| V2 | Deverá exibir a tela de cadastro de entrada |
| P3 | Selecionar a opção tirar foto. |
| P4 | Após informar dados, selecionar a opção salvar. |
| V4 | Deverá exibir mensagens de entrada com sucesso |
| **Resultado Esperado:** Devem ser cumpridas todas as regras licitadas acima. | | |

***Roteiro de Teste – Manter Saída***

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Teste:** Funcionalidade | | **Nome da Funcionalidade:** Funcionalidade | |
| **Data Início:** 27/11/2015 | **Data Fim:**  27/11/2015 | | **Tempo Despendido (h):**  1 Horas |
| **Pré-Condição:** O usuário deve acessar o sistema com perfil de administrador/Atendente | | | |

|  |
| --- |
| **Localização:** Menu |

|  |  |  |
| --- | --- | --- |
| **Requisito: caso de uso** | | |
| **Objeto de Teste:**   1. Validar funcionalidade. 2. Validar interface. 3. Validar mensagem da funcionalidade. | | |
| **Descrição do Caso de Teste:**  **1-** O sistema deverá controlar a inserção de dados do cliente na interface. **2-** O sistema deverá informar uma mensagem quando os dados estiverem incorretos. **3-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for inserido um veículo. **4-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for alterado um veículo. **5-** O sistema deverá limpar da base de dados e gerar a mensagem informativa quando for excluído um veículo. **6-** O sistema deverá informar uma mensagem quando não forem informados os dados obrigatórios. | | |
| **Pré-Condição**:  ID 1 – Gerente deve ter o perfil de Atendente;  ID 2 – Gerente deve ter o perfil de gerente; | | |
| **Dados de Entrada: Dados do funcionário** | | |
| **ID** | **Passo** | **Procedimento** |
| 1/2 | P1 | Acessa do sistema. |
| V1 | Deverá exibir a tela principal do sistema |
| P2 | Deverá fazer consulta. |
| V2 | Deverá exibir a tabela de entrada |
| P3 | Selecionar a opção e registrar saída. |
| P4 | Após informar dados, selecionar a opção e confirmar saída. |
| V4 | Deverá exibir mensagens de saída com sucesso |
| **Resultado Esperado:** Devem ser cumpridas todas as regras licitadas acima. | | |

***Roteiro de Teste – Efetuar Pagamento***

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Teste:** Funcionalidade | | **Nome da Funcionalidade:** Funcionalidade | |
| **Data Início:** 27/11/2015 | **Data Fim:**  27/11/2015 | | **Tempo Despendido (h):**  1 Horas |
| **Pré-Condição:** O usuário deve acessar o sistema com perfil de administrador/Atendente | | | |

|  |
| --- |
| **Localização:** Menu |

|  |  |  |
| --- | --- | --- |
| **Requisito: caso de uso** | | |
| **Objeto de Teste:**   1. Validar funcionalidade. 2. Validar interface. 3. Validar mensagem da funcionalidade. | | |
| **Descrição do Caso de Teste:**  **1-** O sistema deverá controlar a inserção de dados do cliente na interface. **2-** O sistema deverá informar uma mensagem quando os dados estiverem incorretos. **3-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for inserido um veículo. **4-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for alterado um veículo. **5-** O sistema deverá limpar da base de dados e gerar a mensagem informativa quando for excluído um veículo. **6-** O sistema deverá informar uma mensagem quando não forem informados os dados obrigatórios. | | |
| **Pré-Condição**:  ID 1 – Gerente deve ter o perfil de Atendente;  ID 2 – Gerente deve ter o perfil de gerente; | | |
| **Dados de Entrada: Dados do funcionário** | | |
| **ID** | **Passo** | **Procedimento** |
| 1/2 | P1 | Deverá fazer ter dado saída em um veículo. |
| V1 | Deverá exibir a tela pagamento |
| P2 | Após informar dados, selecionar a opção registrar . |
| V2 | Deverá exibir mensagens de pagamento com sucesso |
| **Resultado Esperado:** Devem ser cumpridas todas as regras licitadas acima. | | |

***Roteiro de Teste – Gerar Relatório***

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Teste:** Funcionalidade | | **Nome da Funcionalidade:** Funcionalidade | |
| **Data Início:** 27/11/2015 | **Data Fim:**  27/11/2015 | | **Tempo Despendido (h):**  1 Horas |
| **Pré-Condição:** O usuário deve acessar o sistema com perfil de administrador | | | |

|  |
| --- |
| **Localização:** Menu |

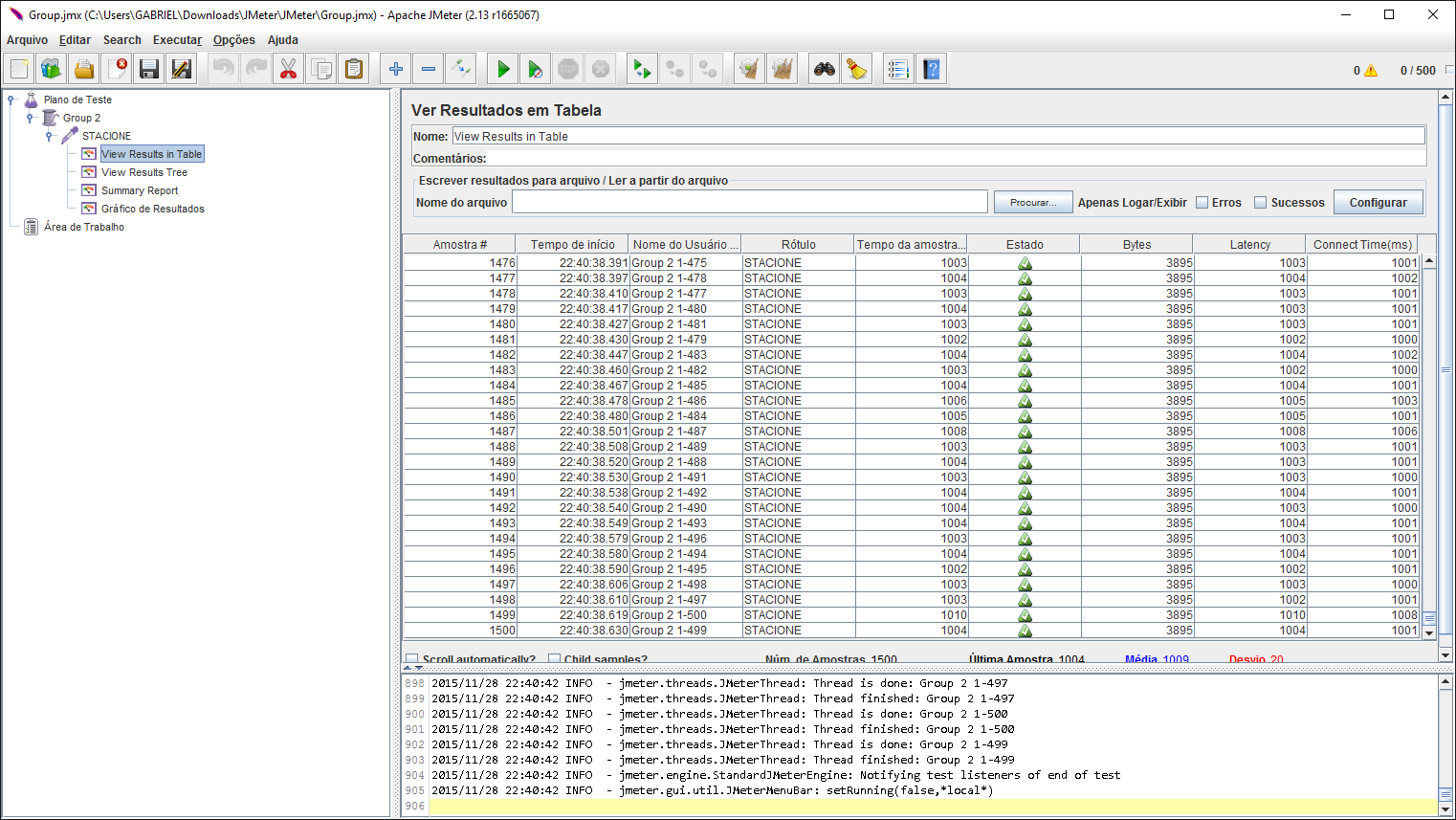
|  |  |  |
| --- | --- | --- |
| **Requisito: caso de uso** | | |
| **Objeto de Teste:**   1. Validar funcionalidade. 2. Validar interface. 3. Validar mensagem da funcionalidade. | | |
| **Descrição do Caso de Teste:**  **1-** O sistema deverá controlar a inserção de dados do cliente na interface. **2-** O sistema deverá informar uma mensagem quando os dados estiverem incorretos. **3-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for inserido um veículo. **4-** O sistema deverá salvar na base de dados e gerar a mensagem informativa quando for alterado um veículo. **5-** O sistema deverá limpar da base de dados e gerar a mensagem informativa quando for excluído um veículo. **6-** O sistema deverá informar uma mensagem quando não forem informados os dados obrigatórios. | | |
| **Pré-Condição**:  ID 2 – Gerente deve ter o perfil de gerente; | | |
| **Dados de Entrada: Dados do funcionário** | | |
| **ID** | **Passo** | **Procedimento** |
| 2 | P1 | Acessa do sistema. |
| V1 | Deverá exibir a tela principal do sistema |
| P2 | Selecionar a opção relatório. |
| V2 | Deverá exibir a tela de relatório |
| P3 | Após informar dados, selecionar a opção gerar. |
| V3 | Deverá baixar o relatório em pdf |
| **Resultado Esperado:** Devem ser cumpridas todas as regras licitadas acima. | | |

# APÊNDICES M – TESTE DE *STRESS*

A execução manual de um caso de teste é rápida e efetiva, mas a execução e repetição de um vasto conjunto de testes executados manualmente é uma tarefa cara e entediante. Assim, foi usada a ferramenta JMeter para realizar o teste de stress no sistema STACIONE.

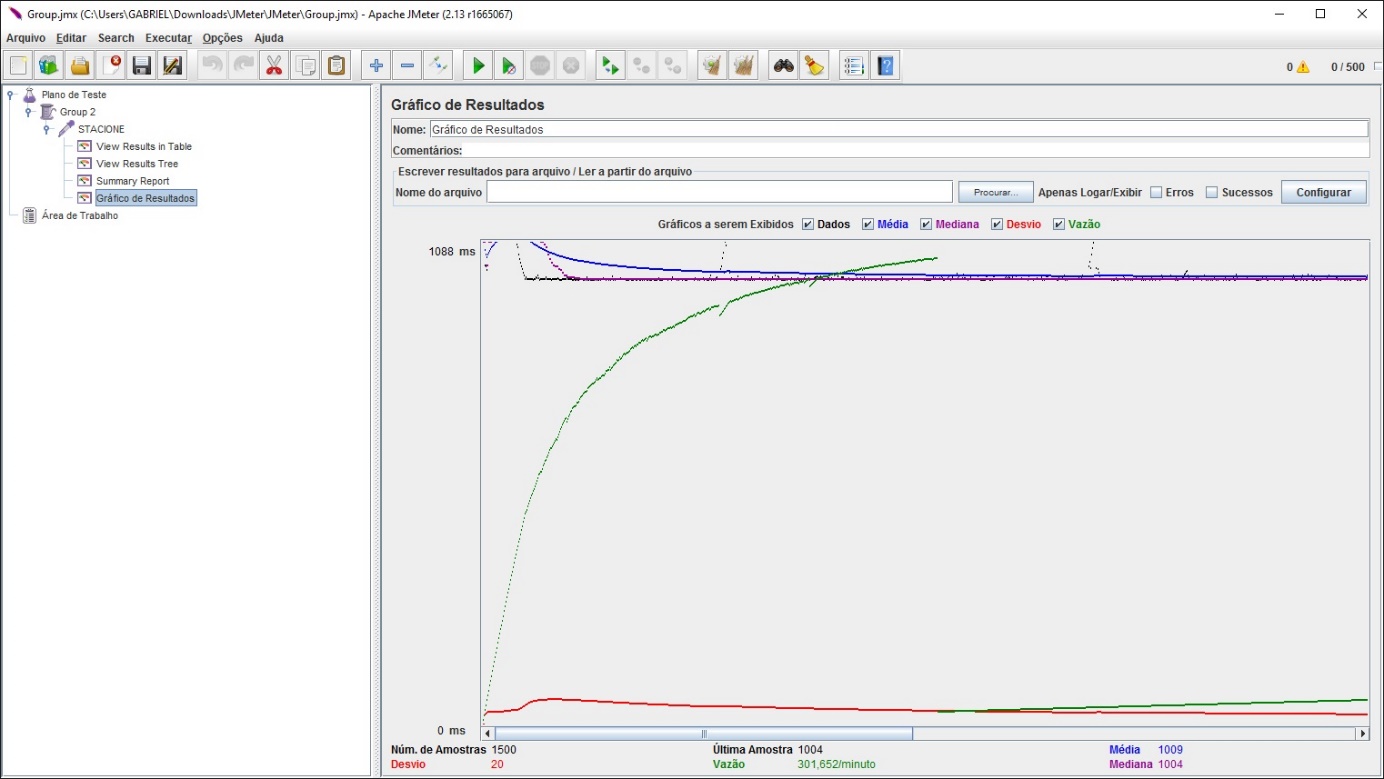
Os usuários do sistema são representados como *threads*. Dentro de um script, *asthreads* executam a sequência de passos determinada. Cada thread simula um usuário, como mostra na figura 1 abaixo

**Figura 1 – Tela de *thread***



Foi avaliado o comportamento da aplicação sob condição crítica, verificando o volume de transações que a aplicação pode suportar. A aplicação rodou normalmente com ótimo tempo de resposta para 500 usuários virtuais, acessando ao mesmo tempo (número de  
*threads* usado no teste).

**Figura 2 – Gráfico dos testes**

****

# APÊNDICES N – MANUAL DE INSTALAÇAO DO SISTEMA

Modelo de Manual de Instalação, Administração e Uso do Stacione

Versão 1.0

Autores

Gabriel Cabral de Almeida

Wagner de Araújo Alves

Brasília, novembro de 2015

# **Licença deste Documento**

**Para a utilização deste documento é necessário seguir as regras da licença Creative Commons pela mesma Licença 2.5 Brasil (**[**http://creativecommons.org/licenses/by-nc-sa/2.5/br/deed.pt\_BR**](http://creativecommons.org/licenses/by-nc-sa/2.5/br/deed.pt_BR#_blank)**).**

**Você tem a liberdade de:**

**Compartilhar** — copiar, distribuir e transmitir a obra.

**Remixar** — criar obras derivadas.

Sob as seguintes condições:

**Atribuição** — Você deve creditar a obra da forma especificada pelo autor ou licenciante (mas não de maneira que sugira que estes concedem qualquer aval a você ou ao seu uso da obra).

**Compartilhamento pela mesma licença** — Se você alterar, transformar ou criar em cima desta obra, você poderá distribuir a obra resultante apenas sob a mesma licença, ou sob uma licença similar à presente.

Ficando claro que:

**Renúncia** — Qualquer das condições acima pode ser [renunciada](http://creativecommons.org/licenses/by-nc-sa/2.5/br/deed.pt_BR) se você obtiver permissão do titular dos direitos autorais.

**Domínio Público** — Onde a obra ou qualquer de seus elementos estiver em [domínio público](http://wiki.creativecommons.org/Public_domain) sob o direito aplicável, esta condição não é, de maneira alguma, afetada pela licença.

**Outros Direitos** — Os seguintes direitos não são, de maneira alguma, afetados pela licença:

* 1. Limitações e exceções aos direitos autorais ou quaisquer [usos livres](http://wiki.creativecommons.org/Frequently_Asked_Questions" \l "Do_Creative_Commons_licenses_affect_fair_use.2C_fair_dealing_or_other_exceptions_to_copyright.3F) aplicáveis;
  2. Os [direitos morais](http://wiki.creativecommons.org/Frequently_Asked_Questions" \l "I_don.E2.80.99t_like_the_way_a_person_has_used_my_work_in_a_derivative_work_or_included_it_in_a_collective_work.3B_what_can_I_do.3F) do autor;
  3. Direitos que outras pessoas podem ter sobre a obra ou sobre a utilização da obra, tais como [direitos de imagem](http://wiki.creativecommons.org/Frequently_Asked_Questions" \l "When_are_publicity_rights_relevant.3F) ou privacidade.

**Aviso** — Para qualquer reutilização ou distribuição, você deve deixar claro a terceiros os termos da licença a que se encontra submetida esta obra. A melhor maneira de fazer isso é com um link para esta página.

# Introdução ao STACIONE

O STACIONE é um sistema de gerenciamento de estacionamento que tem os programadores Gabriel Cabral de Almeida e Wagner de Araújo Alves como a mantenedores. O sistema é baseado no modelo MVC sendo capaz de registrar a entrada e saída de veículos, cadastro de usuários, criação de relatórios e gerenciamento de vagas e preços.

## *Características e principais funcionalidades*

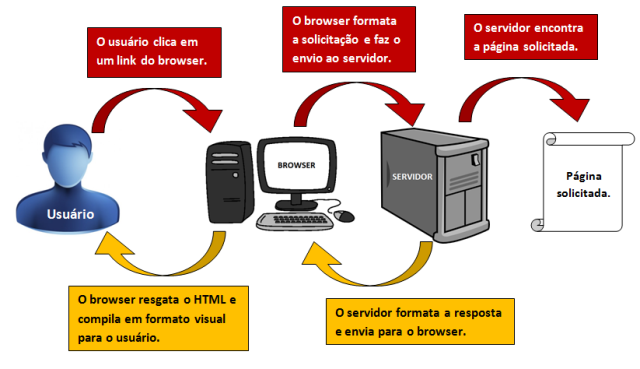
Sua implementação é em PHPTM, HTMLTM e CSSTM tendo como camada de armazenamento o sistema gerenciador de banco de dados MySQLTM.

O sistema possui variadas funcionalidades entre as quais destacam-se:

1. Suporte em plataforma WindowsTM e a LinuxTM ;
2. Integração com câmeras de capturas de imagens que registra a placa do automóvel na entrada do estacionamento;
3. Facilita o controle financeiro através de relatórios gerenciais;
4. Controle de usuários;

# Arquitetura

Deverá ser explicitado a arquitetura do sistema em questão contendo textos, figuras, fluxos, etc.



Falta a arquitetura.

# Instalação do Sistema

## Requisitos gerais para instalação

1. As estações (computadores) em rede para que possa ser instalado um navegador web (browser). As estações podem ser tanto Windows como Linux.

## Hardware mínimo necessário

* Computador com processador Intel ou Amd.
* Memória RAM: 512MB RAM (2GB RAM recomendável);
* HD: 10GB para instalar o GNU/Linux, o STACIONE e a base de dados (50GB SCSI recomendável)

## Pré-requisitos

Servidor web: Apache 2;

Banco de Dados: MySQL;

Linguagem de programação: PHP5.

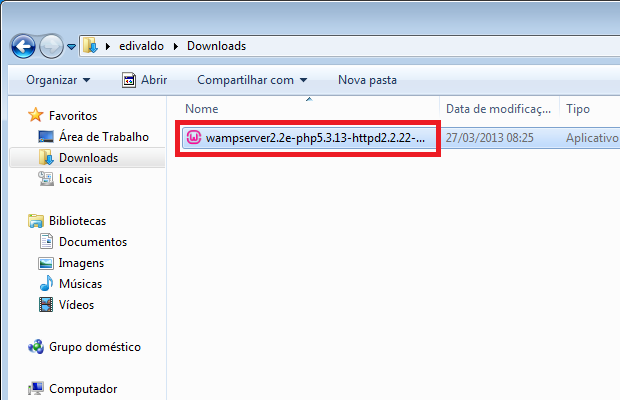
Para que o STACIONE funcione, é necessária a instalação de um servidor web. Existem vários aplicativos que fazem a simulação de um servidor web como o XAMPP, EASYPHP ou WAMP.

Utilizaremos a instalação do WAMP SERVER como exemplo.

**Instalação do WampServer no Windows**

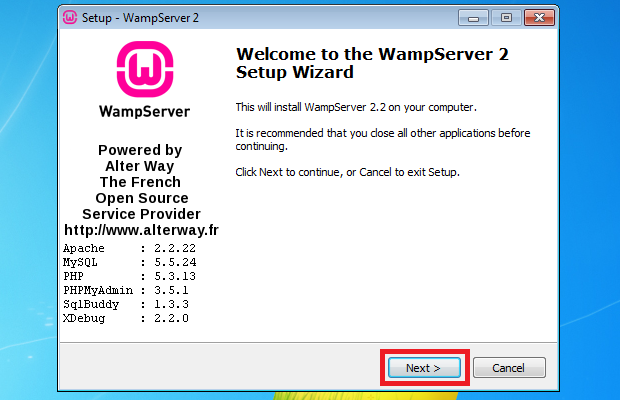
Passo 1. Insira o CD de instalação e identifique o aplicativo **WampServer.**

Passo 2. Dê um duplo clique sobre ele. Se for solicitada a senha do administrador, digite-a ou apenas confirme a autorização;



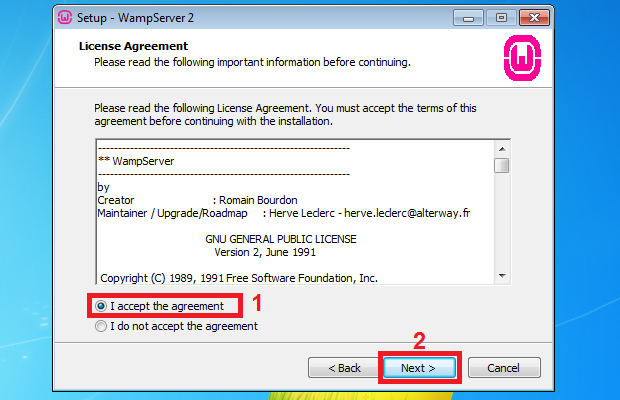
**Executando o instalador baixado**

Passo 3. Na primeira tela do instalador, apenas clique no botão “Next”;



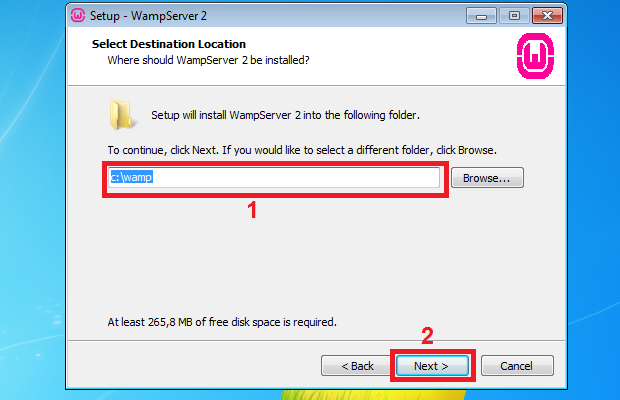
**Iniciando a instalação**

Passo 4. Na tela seguinte, aceite os “Termos de Uso” (é bom ler) e clique novamente em “Next”;



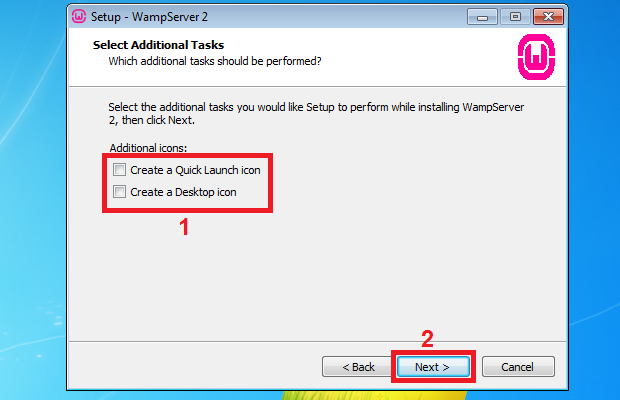
**Concordando com os "termos de uso" do programa**

Passo 5. Na próxima tela, escolha a pasta onde será instalado o software. De preferência, deixe o endereço padrão. Clique no botão “Next”;



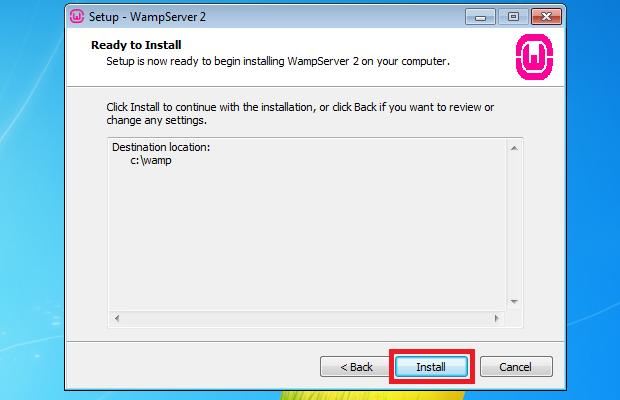
**Escolhendo a pasta onde será instalado o programa**

Passo 6. Em seguida, é possível marcar ou desmarcar opções referentes à criação do ícone do programa. Depois, novamente, clique em “Next”;



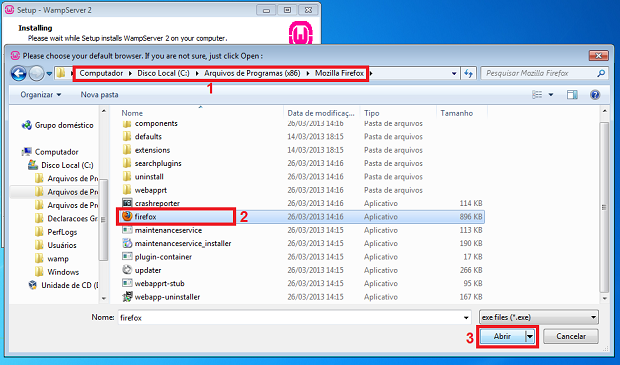
**Escolhendo onde serão instalados ícones do programa**

Passo 7. Finalmente, clique no botão “Install”;



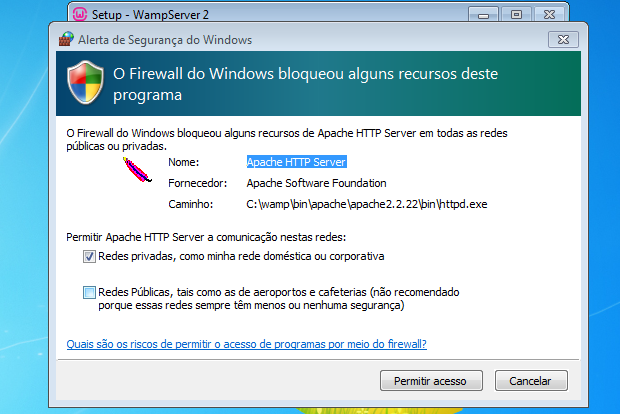
**Confirmando e autorizando a instalação**

Passo 8. Agora é necessário informar para o instalador qual navegador você usa. Na janela que aparece, vá até a pasta onde ele fica e clique no executável dele. Depois, clique no botão “Abrir”. No exemplo, foi utilizado o Mozilla Firefox, mas você pode escolher qualquer um;



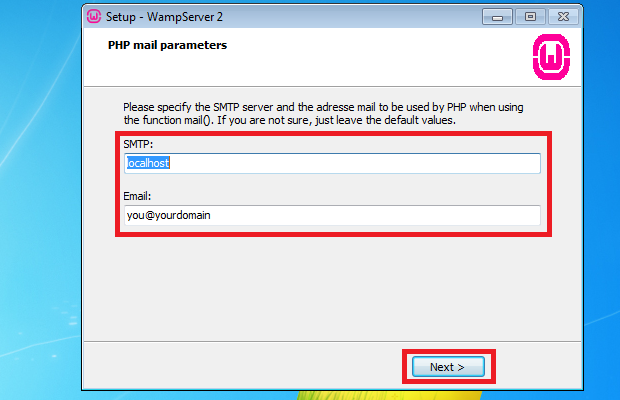
**Informando qual navegador será usado pelo programa**

Passo 9. Se aparecer uma tela do Firewall do Windows, clique no botão “Permitir acesso” para autorizar a execução dos serviços do servidor;



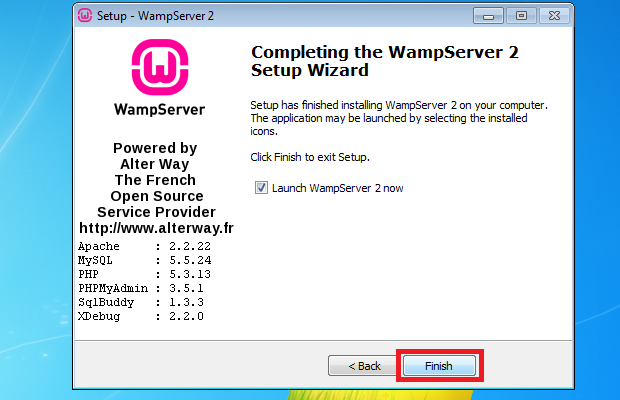
**Autorizando a execução dos serviços**

Passo 10. Nessa etapa é preciso informe o servidor SMTP (servidor de e-mail) para o instalador, entretanto, como ele está sendo instalando localmente, apenas clique no botão “Next”.



**Colocando informações do servidor de e-mail**

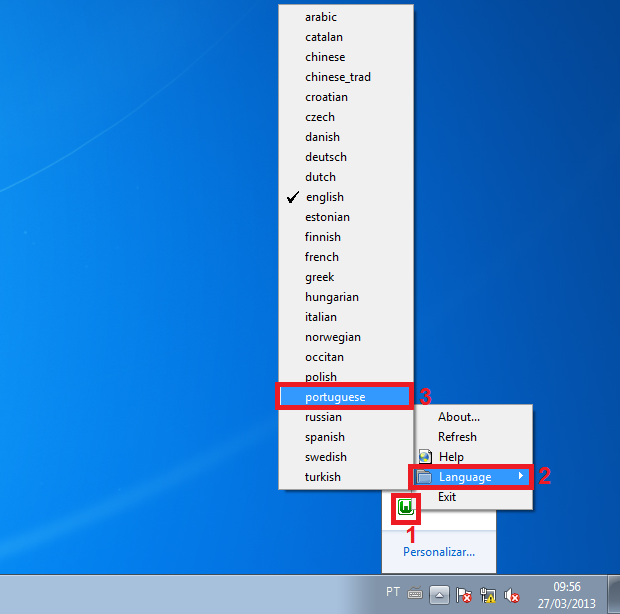
Passo 11. Nessa última tela, clique no botão “Finish”, para finalizar a instalação. O WampServer será iniciado e estará pronto para ser utilizado!



**Finalizando a instalação**

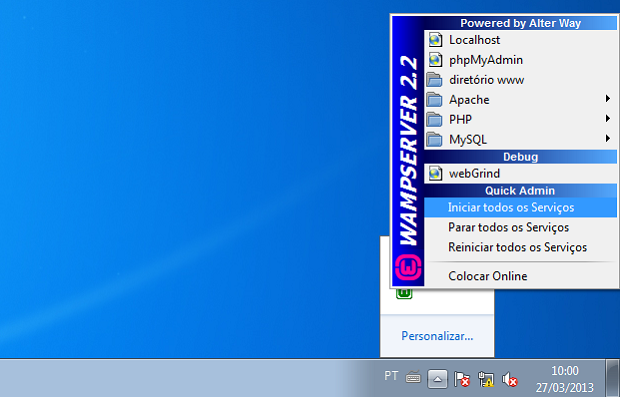
**Acessando as opções do servidor**

Passo 1. Clique com o botão direito no ícone do WampServer, ele fica ao lado do relógio do Windows na “Área de Notificação”. No menu que aparece, clique na opção “Language” e em seguida no item “Português”;



**Colocando o WampServer em português**

Passo 2. Clique com o botão esquerdo sobre o mesmo ícone e serão exibidas todas as opções para o servidor do programa, como parar, reiniciar e iniciar os serviços (php, mysql, apache).



**Acessando o menu com as opções do WampServer**

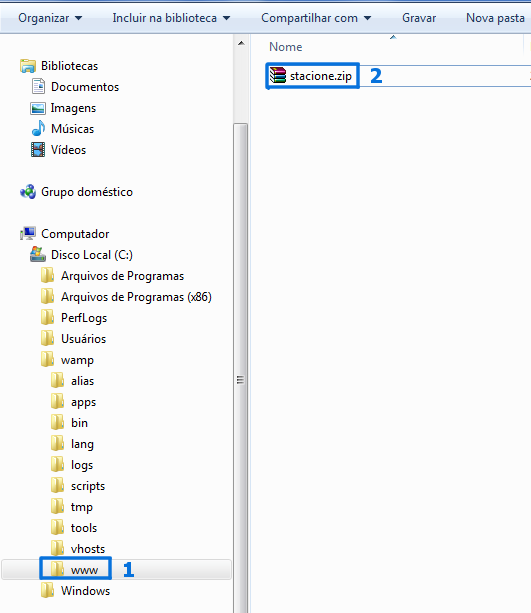
Passo 3. Para acessar ao servidor web local no endereço, clique na opção “Localhost” do menu;

Passo 4. Para acessar o gerenciador do banco de dados Mysql, clique na opção “phpMyAdmin”;

Passo 5. Para acessar o diretório dos arquivos públicos do site, clique na opção “diretório www”.

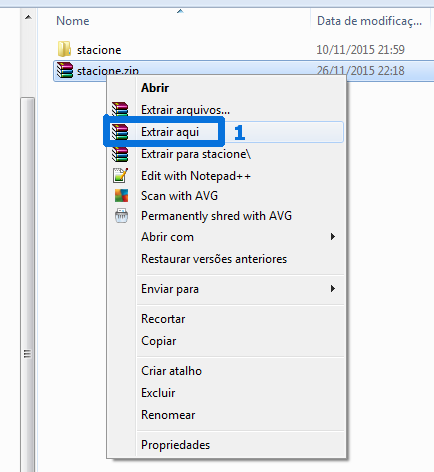
**Instalação do Sistema STACIONE**

Para iniciar a instalação do Stacione, copie o arquivo “stacione.zip” e cole na pasta padrão do WampServer, no caso C://wamp/www conforme exemplo:



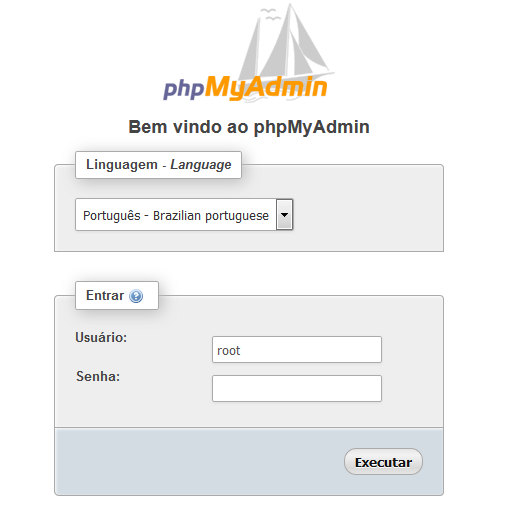
**Copiando o arquivo stacione.zip na pasta www**

Descompacte o arquivo stacione.zip



**Descompactando o arquivo stacione.zip**

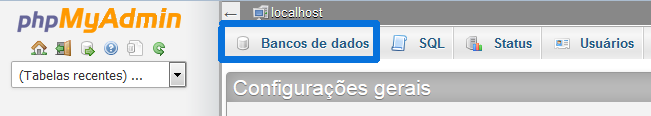
Para preparar o banco de dados do Stacione, entre no navegador web e digite localhost/phpmyadmin (o aplicativo wamp tem que está ativo).



**Tela de acesso ao phpMyAdmin**

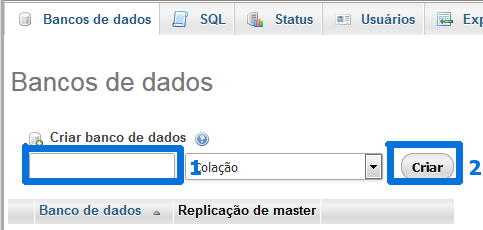
Acesse o phpMyAdmin com o usuário e senha cadastrados (geralmente o usuário é root e a não tem senha) e clique em executar.

Na tela principal do phpMyAdmin, clique em banco de dados:



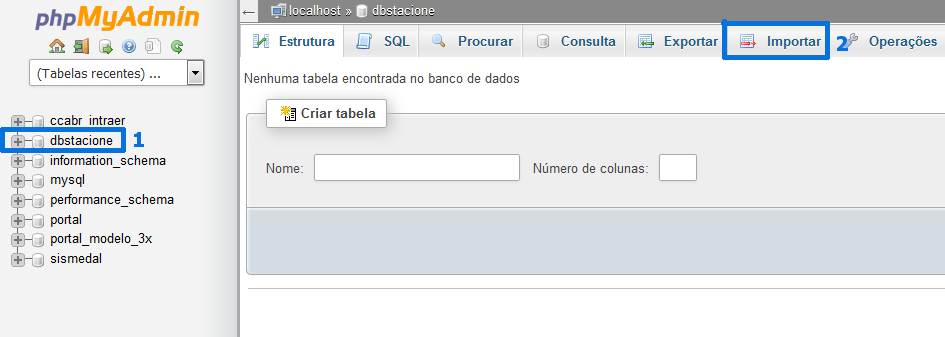
**Clicando no banco de dados**

Clique na área demarcada, digite o nome do banco de dados "dbstacione" sem as aspas e clique no botão "criar".



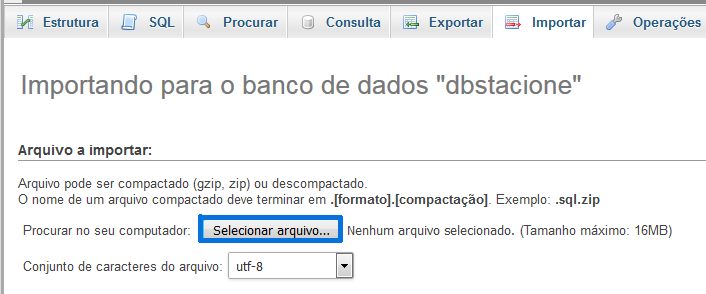
**Criando banco de dados**

Após a criação do banco "dbstacione", acesse o banco criado e escolha a opção "Importar".



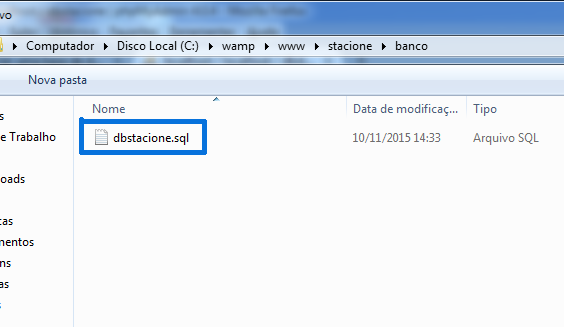
**Acessando a opção Importar**

Para selecionar o arquivo para importar, clique no botão "Selecionar arquivo..."



**Selecionando o arquivo para importar**

Navegue até a pasta "banco" e selecione o arquivo "dbstacione.sql" (caminho até o arquivo: C://wamp/www/stacione/banco).



**Selecionando o arquivo dbstacione.sql**

Após selecionar o arquivo "stacione.sql", clique em "Executar".

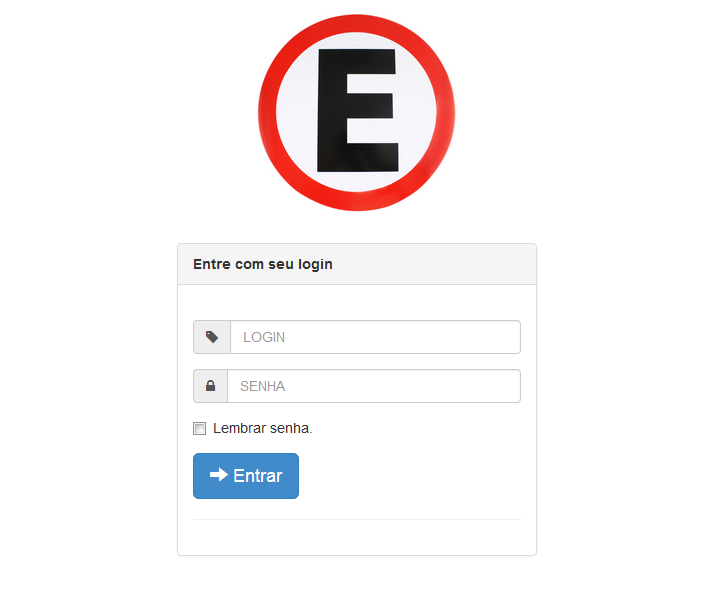


**Executando o arquivo dbstacione.sql**

Pronto! O banco de dados foi importado e o sistema já pode ser utilizado.

# Administração e Uso do Sistema

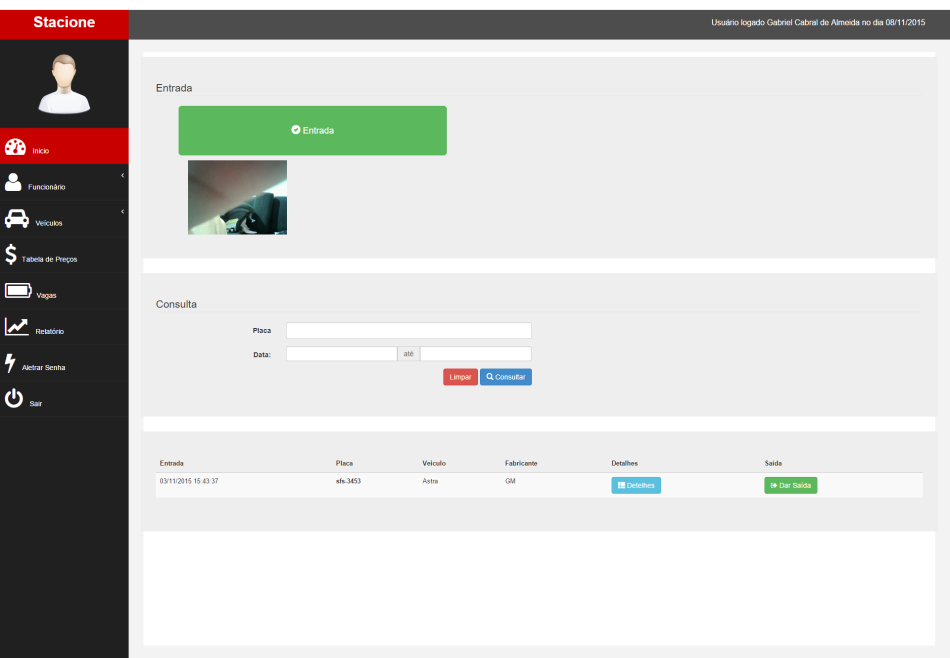
Para acessar o Stacione, digite no navegador web (browser) "localhost/stacione".



**Tela de login do Stacione**

Utilize o usuário e senha padrão para o primeiro acesso ao sistema e acesso a tela principal.

**Usuário: admin e Senha: admin**

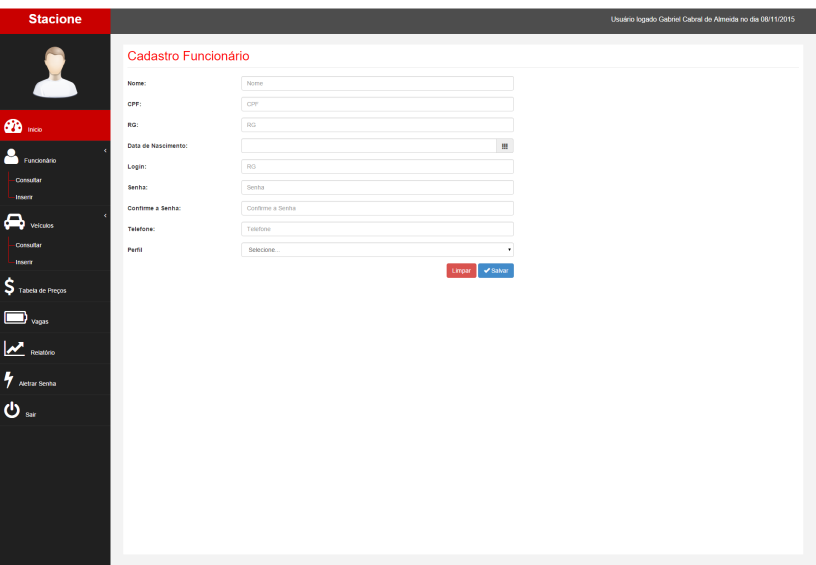
****

**Tela principal do sistema**

**Funcionalidades**

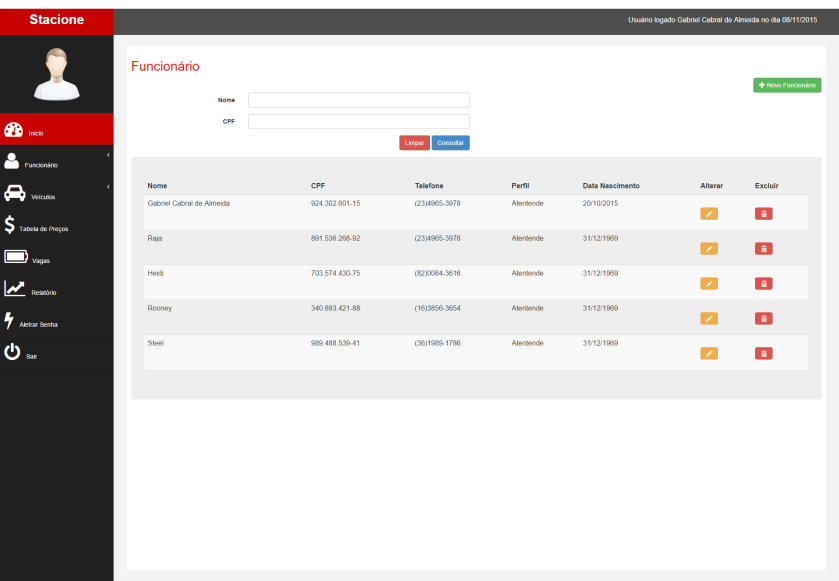
**Cadastro de funcionários**

Para cadastrar um funcionário, clique em **Funcionário: Inserir**. O nível de acesso ao sistema será de acordo com o perfil do funcionário. Selecione o nível de acesso do funcionário (Gerente ou Atendente), cadastre os campos solicitados e clique em salvar.

****

**Tela de cadastro de Funcionário**

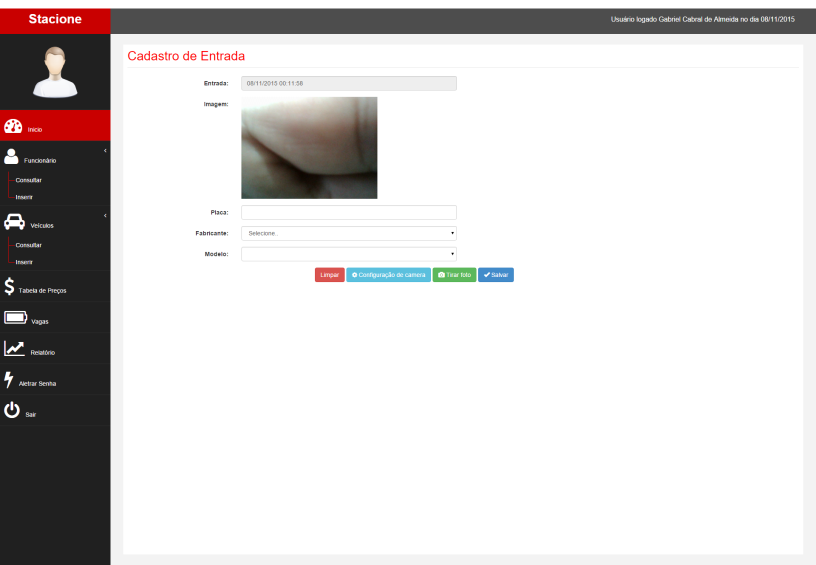
Para alterar qualquer informação referente aos usuários acesse **Funcionário:: Consultar**. Aparecerá uma lista com todos os funcionários cadastrados. Selecione o funcionário e faça as alterações desejadas.

****

**Tela de consulta de Funcionários**

**Registrar entrada**

Para registrar a entrada de um carro no estacionamento, clique no botão registrar entrada.

****

**Tela de Registrar entrada de veículos**

**Registrar saída**

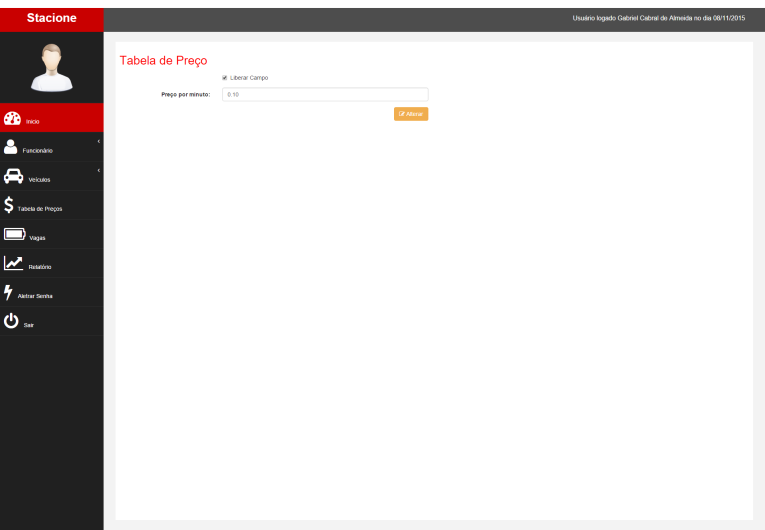
Para registrar a saída de veículos, clique em **Veículos: Consultar**, selecione o veículo da lista e registre a saída do veículo.



**Tela de Registrar saída de veículos**

**Alterar tabela de preços**

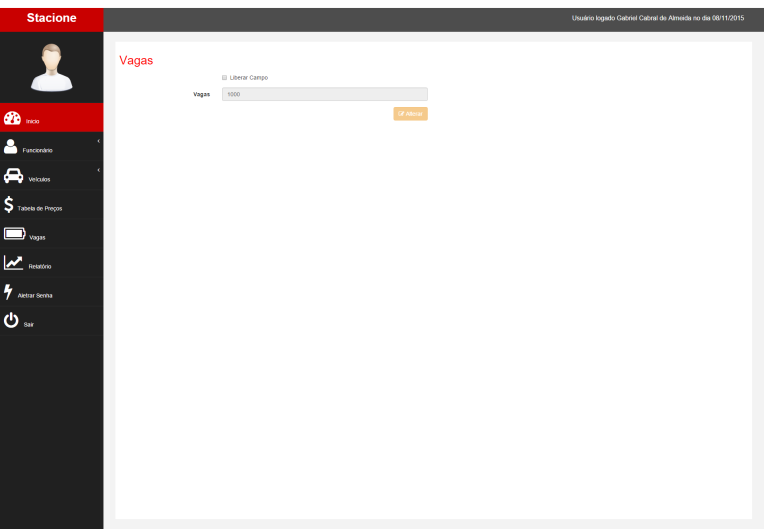
A cobrança no sistema é feita por minuto. Para modificar o preço, clique em **Tabela de preço** e altere o valor conforme desejado. Após modificação, clique em **Alterar.**

****

**Tela de Tabela de Preço**

**Alterar quantidade de vagas no estacionamento**

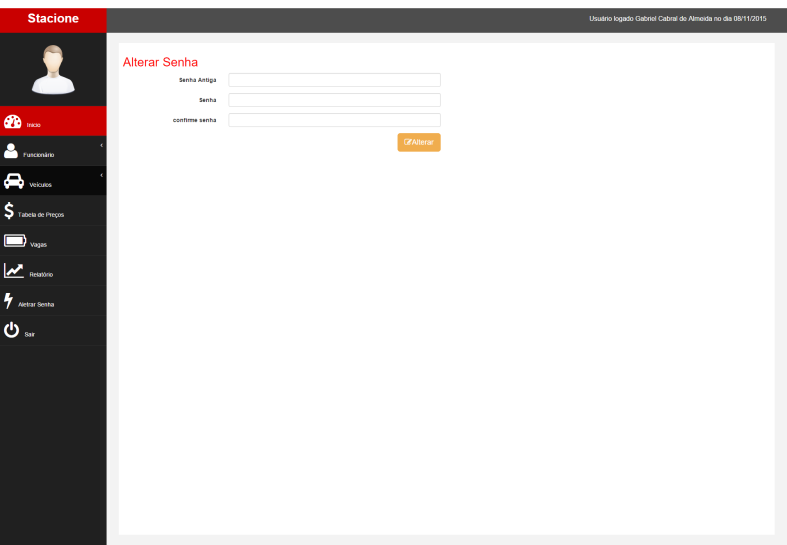
É possível definir a quantidade de vagas disponíveis no estacionamento. Para fazer qualquer alteração no número de vagas, clique em Vagas e altere conforme desejado.

****

**Tela de alteração do número de vagas**

**Alterar senha**

Para manter a segurança na utilização do sistema, é desejável que periodicamente a senha de acesso seja alterada. Para realizar essa ação, clique em **Alterar Senha.**

****

**Tela de alteração de senha**

**Gerar relatórios**

No perfil de Gerente, há a opção de geração de relatórios. Para acessá-lo, clique em Relatórios e imprima.

A utilização do Stacione é fácil e intuitiva. É necessário pouco tempo para que o usuário aprenda as funcionalidades e se familiarize com o sistema.

# Link da Licença Júridica Creative Commons

*http://creativecommons.org/licenses/by-sa/2.5/br/legalcode*