

**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 apresentada 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.

Gabriel Cabral de Almeida e Wagner de Araújo Alves

**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 presente trabalho apresenta o Sistema de Gerenciamento de Estacionamento – STACIONE com objetivo de informatizar e agilizar os processos feitos manualmente pelo órgão responsável por gerir o referido programa. 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 5.3. Os dados são gerenciados pelo Sistema Gerenciador de Banco de Dados – SGDB Mysql 5.0. 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 - HTML e sistema de estilo em cascata – CSS. 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**

This paper presents the Parking Management System - STACIONE in order to computerize and streamline processes done manually by the body responsible for managing the program. By requirements analysis and comparison with similar systems to STACIONE it is possible to have a clear view of the functional system requirements. Using the IBM Rational Unified Process - IRUP and Unified Modeling Language - UML in its vast recursion is present in the system modeling, activities and artifacts produced. For coding of STACIONE is used to PHP 5.3 programming language. Data is managed by the Database Management System - DBMS MySQL 5.0. Because it is a system focused on the internet Word Web Consortiun standards - WC3 are essential for the proper functioning of the system, for this is used for hypertext markup language - HTML and cascading style system - CSS. For validation of forms and masking of fields is used JavaScript. At the end the system is subjected to various tests, resulting in a high quality software product 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 24

Figura 7 – Ciclo de Vida do RUP 25

Figura 8 – Sistema de banco de dados 28

Figura 9 – Sistema Gerenciador de Banco de Dados 29

Figura 10 – Página principal do Astah 34

Figura 11 – Página principal do NetBeans 35

Figura 12 - Página do MySQL Workbench 36

Figura 13 - Página do XAMPP 37

**LISTA DE QUADROS**

Quadro 1 – Quadro TIOBE 32

Quadro 2 – Quadro de Análise de Riscos 39

Quadro 2 – Descrição de Tarefas 40

Quadro 3 – Cronograma 41

**LISTA DE SIGLAS**

**CSS** *Cascading Style Sheets* (Folhas de Estilo em Cascata)

**DDL** *Data Definition Language*

**DML** *Data Manipulation Language*

**HTML**  *HyperText Markup Language* (Linguagem de Marcação de Hipertexto)

**IDE** *Integrated Development Environment*

**MVC** *Model-view-controller* (Modelo-Visualização-Controle)

**RUP** *Rational Unified Process* (Processo Unificado Rational)

**SGBD** Sistema Gerenciador de Banco de Dados

**SSL** *Secure Sockets Layer*

**UML** *Unified Modeling Language* (Linguagem de Modelagem Unificada)

**TLS** *Transport Layer Security*

**SUMÁRIO**

[1. INTRODUÇÃO 13](#_Toc434933353)

[1.1. Objetivos 13](#_Toc434933354)

[1.2. Estrutura do Trabalho de Conclusão 13](#_Toc434933355)

[2. REFERENCIAL TEÓRICO 15](#_Toc434933356)

[2.1. Engenharia de *Software* 15](#_Toc434933357)

[2.2. IBM Rational Unified Process (IRUP) 21](#_Toc434933358)

[2.3. *Unified Modeling Language* (UML) 25](#_Toc434933359)

[2.4. Padrões de Projeto 25](#_Toc434933360)

[2.5. *World Wide Web Consortium* (W3C) 26](#_Toc434933361)

[2.6. Banco de Dados 28](#_Toc434933362)

[2.7. Linguagem de Programação PHP 30](#_Toc434933363)

[*2.8.* Servidor *Web* 32](#_Toc434933364)

[2.9. Ferramentas 33](#_Toc434933365)

[2.10. Requisitos de Segurança de Software 37](#_Toc434933366)

[3. METODOLOGIA 40](#_Toc434933367)

[3.1. Concepção 40](#_Toc434933368)

[3.2. Elaboração 45](#_Toc434933369)

[3.3. Construção 46](#_Toc434933370)

[3.4. Implantação 47](#_Toc434933371)

[4. CONCLUSÃO 48](#_Toc434933372)

[REFERÊNCIAS 49](#_Toc434933373)

[APÊNDICES A - ENTREVISTA 51](#_Toc434933374)

[APÊNDICES B - DOCUMENTO DE VISÃO 52](#_Toc434933375)

[APÊNDICES C – DIAGRAMA E ESPECIFICAÇÃO DE CASO DE USO 55](#_Toc434933376)

[APÊNDICES D - DIAGRAMAS DE SEQUÊNCIA 68](#_Toc434933377)

[APÊNDICES E - DIAGRAMAS DE ATIVIDADES 76](#_Toc434933378)

[APÊNDICES F - DIAGRAMA DE IMPLANTAÇÃO 77](#_Toc434933379)

[APÊNDICES G - DIAGRAMA DE CLASSES 78](#_Toc434933380)

[APÊNDICES H - MER FÍSICO E DICIONÁRIO DE DADOS 79](#_Toc434933381)

[APÊNDICES I - TESTE DE QUALIDADE 82](#_Toc434933382)

[APÊNDICES J - CD/DVD COM IMPLEMENTAÇÃO. 83](#_Toc434933383)

[ANEXO A 84](#_Toc434933384)

# INTRODUÇÃO

Junto ao constante aumento da frota de carros nas ruas e a diminuição das áreas para estacionar, esses espaços devem ser geridos de modo a facilitar o fluxo dos automóveis otimizando principalmente tempo e garantindo a satisfação do cliente.

Cria-se o Sistema de Gerenciamento de Estacionamento – STACIONE para auxiliar a gestão de um estacionamento, possibilitando um melhor controle do fluxo de veículos e melhor controle financeiro do mesmo.

Neste projeto serão apresentadas as características do STACIONE, mostrando informações importantes para o entendimento do sistema bem como suas funcionalidades.

## Objetivos

* + 1. **Geral**

Este projeto tem por objetivo apresentar um sistema que fará o gerenciamento de um estacionamento, de modo a automatizar 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 1 - Introdução: este capítulo descreve o problema observado, a importância do tema em um contexto geral, o objetivo do sistema, os resultados específicos e apresenta a forma como o trabalho está organizado;

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;

Capítulo 4 - Conclusão: aborda considerações sobre a solução, a experiência com o desenvolvimento e possíveis evoluções do sistema.

# REFERENCIAL TEÓRICO

Este capítulo apresenta conceitos relacionados ao tema e descreve a metodologia aplicada e o problema diagnosticado.

## 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.

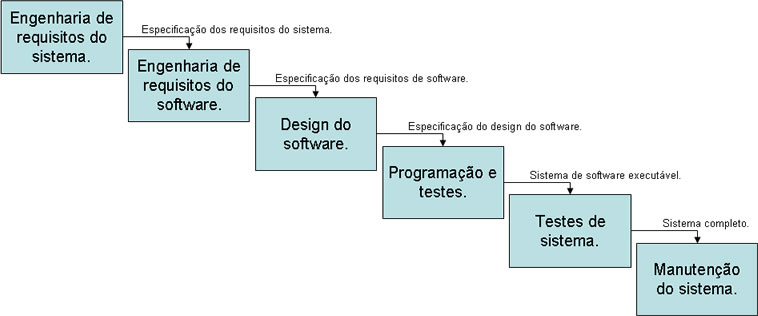
### Modelo cascata

Segundo Pressman (2006, p.38), o 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)

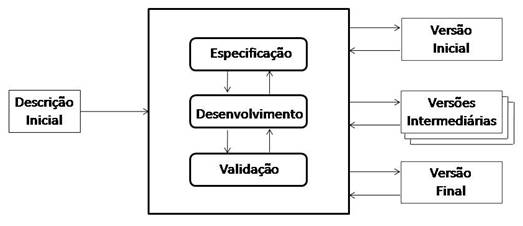
### Modelo Evolucionário

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



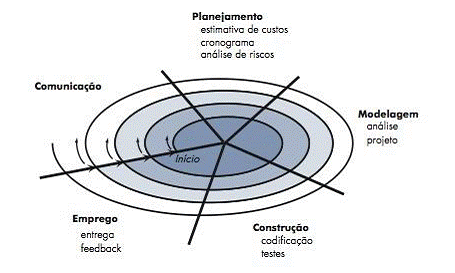
Fonte: Sommerville (2003)

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)

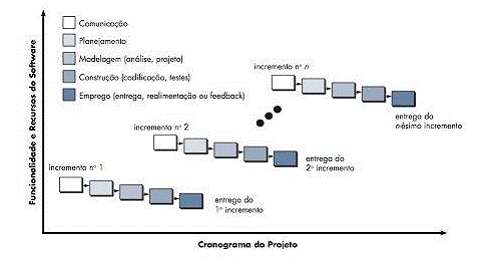
Esse modelo aborda o desenvolvimento de uma forma evolucionária, originando, assim, várias versões: o início se dá com uma nova iteração, assim, o software ganha novas versões a cada iteração, no final das iterações, o software é entregue ao cliente.

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 (IRUP)

De acordo com Kroll e Kruchten (2003), pode-se ter três definições para o Rational Unified Process:

1. O RUP é uma maneira de desenvolvimento de software que é iterativa, centrada à arquitetura e guiada por casos de uso.
2. O RUP é um processo de engenharia de software bem definido e bem   
   estruturado. Ele define claramente quem é responsável pelo que, como as coisas devem ser feitas e quando fazê-las. O RUP também provê uma estrutura bem definida para o ciclo de vida de um projeto, articulando claramente os marcos essenciais e pontos de decisão;
3. O RUP é também um produto de processo que oferece uma estrutura de processo customizável para a engenharia de software.

O RUP utiliza a linguagem de modelagem unificada para especificar, modelar e documentar artefatos. Por ser flexível e configurável, ele pode ser utilizado em projetos de pequeno, médio e grande porte.

Não existe uma fórmula para aplicação do RUP devido ao fato dele poder ser aplicado de várias formas diferentes para cada projeto e organização a ser apresentados. De acordo com Martins (2007), existem alguns princípios que podem caracterizar e diferenciar o RUP de outros métodos iterativos:

a) Atacar os riscos antecipadamente e continuamente;   
b) Certificar-se de entregar algo de valor ao cliente;   
c) Focar no software executável;   
d) Acomodar mudanças antecipadas;   
e) Liberar um executável da arquitetura antecipadamente;   
f) Construir o sistema com componentes;  
g) Trabalhar junto como uma equipe;   
h) Fazer da qualidade um estilo de vida, não algo para depois.

Os princípios do RUP acompanha as premissas referentes a garantia da qualidade do processo visando alcançar a garantia da qualidade do produto de software a ser desenvolvido (Kruchten 2003)

Segundo Kroll e Kruchten (2003), o RUP possui cinco elementos principais: papéis, atividades, artefatos, fluxos de trabalho e disciplinas. Um papel (ou perfil) define o comportamento e as responsabilidades de um determinado indivíduo ou grupo de indivíduos trabalhando como uma equipe. Uma atividade é uma unidade de trabalho que um indivíduo executa quando está exercendo um determinado papel e produz um resultado importante para o contexto do projeto. Um artefato é um pedaço de informação que é produzido, modificado ou utilizado em um processo. Eles então são os produtos de um projeto.

Entende-se que papéis caracterizam os perfis dos profissionais envolvidos no projeto, como por exemplo, analista de sistemas, projetista etc. Quanto à atividade, ele (método) pode ser dividido nos seguintes passos: planejar uma iteração, encontrar casos de uso e atores, rever o projeto e executar testes de performances. Os artefatos podem ter várias formas como um modelo de caso de uso, um modelo de projeto, uma classe, um caso de negócio, código-fonte etc.

A enumeração de atividades, papéis e artefatos não constituem um processo. É necessário saber a sequência do desenvolvimento das atividades para que possam ser produzidos artefatos de valor para o projeto. Um fluxo de trabalho é uma sequência de atividades que são executadas para a produção de um resultado valioso para o projeto (Kroll e Kruchten 2003). Os fluxos de trabalho podem ser representados por diagramas de sequência, diagramas de colaboração e diagramas de atividades da linguagem de modelagem unificada.

O RUP utiliza três tipos de fluxos de trabalho:

a) Fluxos de trabalho principais, associados com cada disciplina;

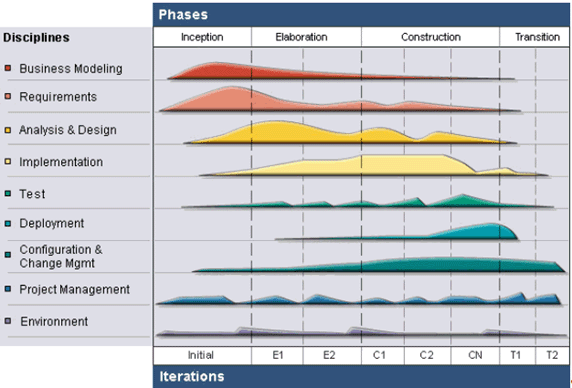
b) Fluxos de trabalho de detalhe, para detalhar cada fluxo de trabalho principal;

c) Planos de iteração, que mostram como a iteração deverá ser executada.

Segundo Martins (2011), uma disciplina "é uma coleção de atividades relacionadas que fazem parte de um contexto comum em um projeto. As disciplinas proporcionam um melhor entendimento do projeto sob o ponto de vista tradicional de um processo cascata.” A separação das atividades em disciplinas torna a compreensão das atividades mais fácil, porém dificulta mais o planejamento das atividades.

O RUP possui nove disciplinas, divididas em disciplinas do processo e de suporte. As disciplinas de processo são: modelagem de negócios, requisitos, análise e projeto, implementação, teste e distribuição. As de suporte são: configuração e gerenciamento de mudanças, gerenciamento de projeto, e ambiente (Kruchten 2003).

Figura 6 – Ciclos de Desenvolvimento do RUP



Fonte: Martins (2007)

Kroll e Kruchten (2003) explicam que o “ciclo de desenvolvimento no RUP possui quatro fases: iniciação, elaboração, construção e transição.” Cada uma delas é concluída por um marco principal conforme mostra a figura 6 (Martins 2007).

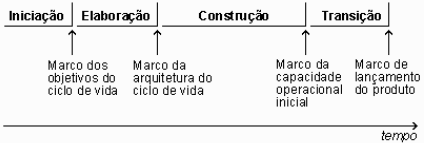
Então, de acordo com a figura 6, constata-se que cada fase é basicamente um intervalo de tempo entre dois marcos principais.

Segundo Martins (2011), o “ciclo de desenvolvimento termina com uma versão completa do produto de software. As fases definem estados do projeto, que são definidos por riscos que estão sendo mitigados ou questões que precisam ser respondidas”.

A fase de iniciação, foca no tratamento de riscos relacionados com o caso de negócio específico. Como implementar RUP é um projeto que visa qualidade do software, acima de tudo, Boente (2003) afirma que “não existe projeto com risco zero”. Sendo assim, a tarefa de quantificação dos riscos de um projeto, passa a ser prioridade essencial, para a implementação de um projeto RUP, pois nos ajuda a verificar se o projeto é viável e se é financeiramente possível de ser implementado.

Na fase elaboração, o foco deve ser nos riscos técnicos e arquiteturais. O escopo deve ser revisado e os requisitos devem estar mais compreendidos. Durante a construção, a atenção será voltada para os riscos lógicos, e a maior parte do trabalho será realizada.

Figura 7 - Ciclo de Vida do RUP



Fonte: Boente (2003)

## *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.

Basicamente, a UML permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados, isto é, semântica. É uma notação independente de processos, embora o RUP tenha sido especificamente desenvolvido utilizando a UML.

## Padrões de Projeto

Segundo Pressman (2006, p. 192), “padrões de projeto incorporam modularidade”.

Em engenharia de software, a modularidade é a decomposição de um software em partes menores, dividindo um problema em vários problemas menores, facilitando seu desenvolvimento e manutenção. Ao se trabalhar com um único software grande, facilmente os desenvolvedores perdem o foco e se preocupam com aspectos não diretamente relacionados com o problema que estão tratando.

Segundo Fowler (apud PRESSMAN 2006, p 203), “Padrões são semi prontos – o que significa que você sempre tem que acabá-los e adaptá-los ao seu próprio ambiente”.

A utilização de padrões como forma de solucionar problemas, busca garantir qualidade e segurança a equipe de trabalho, ao invés de se reinventar, adota-se uma estrutura que já existe e é comprovadamente eficaz, e nesta estrutura pode-se proceder a alterações para adaptá-la a realidade do problema em questão.

## *World Wide Web Consortium* (W3C)

O *Consortium* *World Wide Web (W3C)* é um consórcio internacional no qual organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões para a Web. Liderado pelo inventor da web Tim Berners-Lee e o CEO Jeffrey Jaffe, o W3C tem como missão Conduzir a *World Wide Web* para que atinja todo seu potencial, desenvolvendo protocolos e diretrizes que garantam seu crescimento de longo prazo.

O 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.

### *HyperText Markup Language* (HTML)

HTML - *HyperText Markup Language*: é a linguagem padrão para o acesso e exibição de páginas web. Linguagem que não necessita de compilação, uma vez que as linhas de código são interpretadas e mostradas ao utilizador pelo *browser*. Genericamente, a linguagem HTML é constituída por textos e códigos especiais denominados marcas ou *tags*.

### *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.

A definição mais precisa e simples para folha de estilo encontra-se na homepage das CSS no site do W3C e diz:

“Folha de estilo em cascata é um mecanismo simples para adicionar estilos (por exemplo: fontes, cores, espaçamentos) aos documentos web.”

### *Java Script*

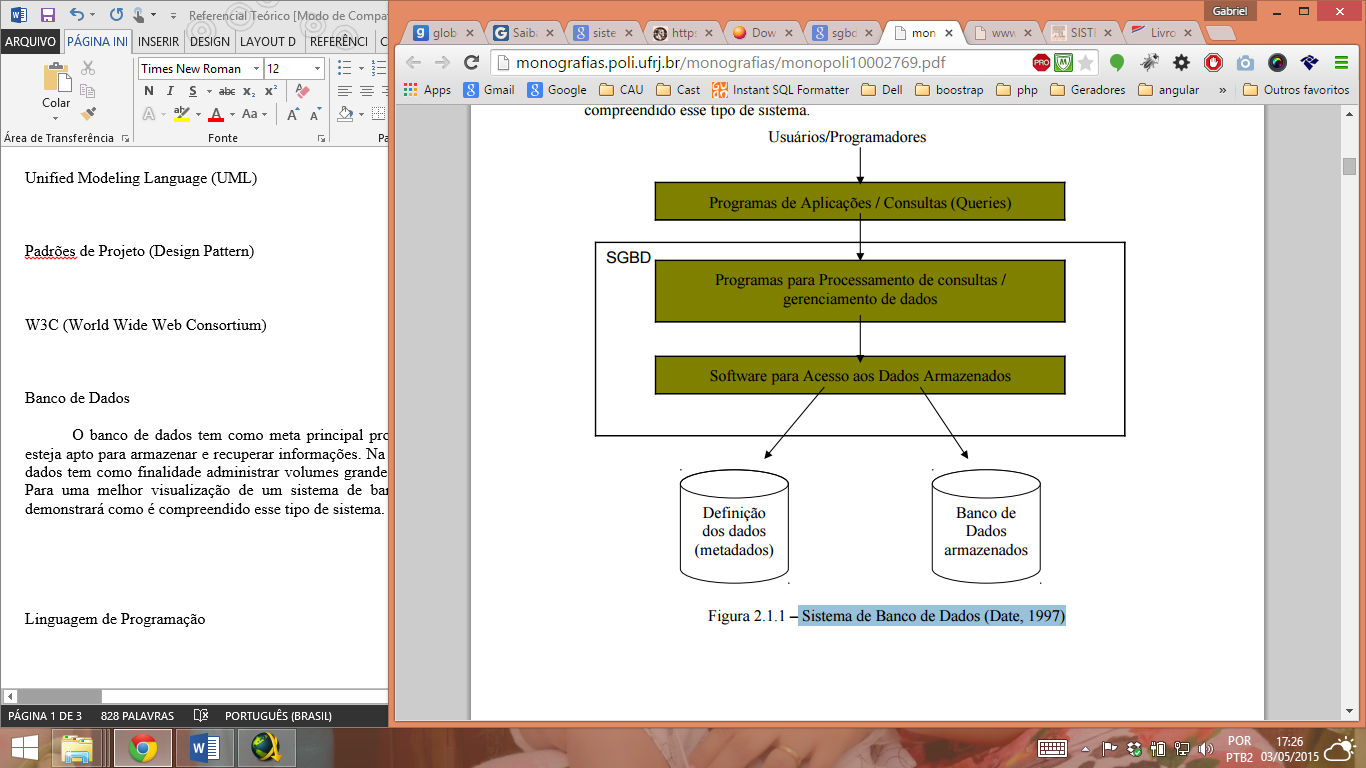
*Java Script* é a linguagem de programação mais popular no desenvolvimento *Web*. Suportada por todos os navegadores, a linguagem é responsável por praticamente qualquer tipo de dinamismo que queiramos em nossas páginas. Se usarmos todo o poder que ela tem para oferecer, podemos chegar a resultados impressionantes. Excelentes exemplos disso são aplicações *Web* complexas como Gmail, Google Maps e Google Docs.

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.

## 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.

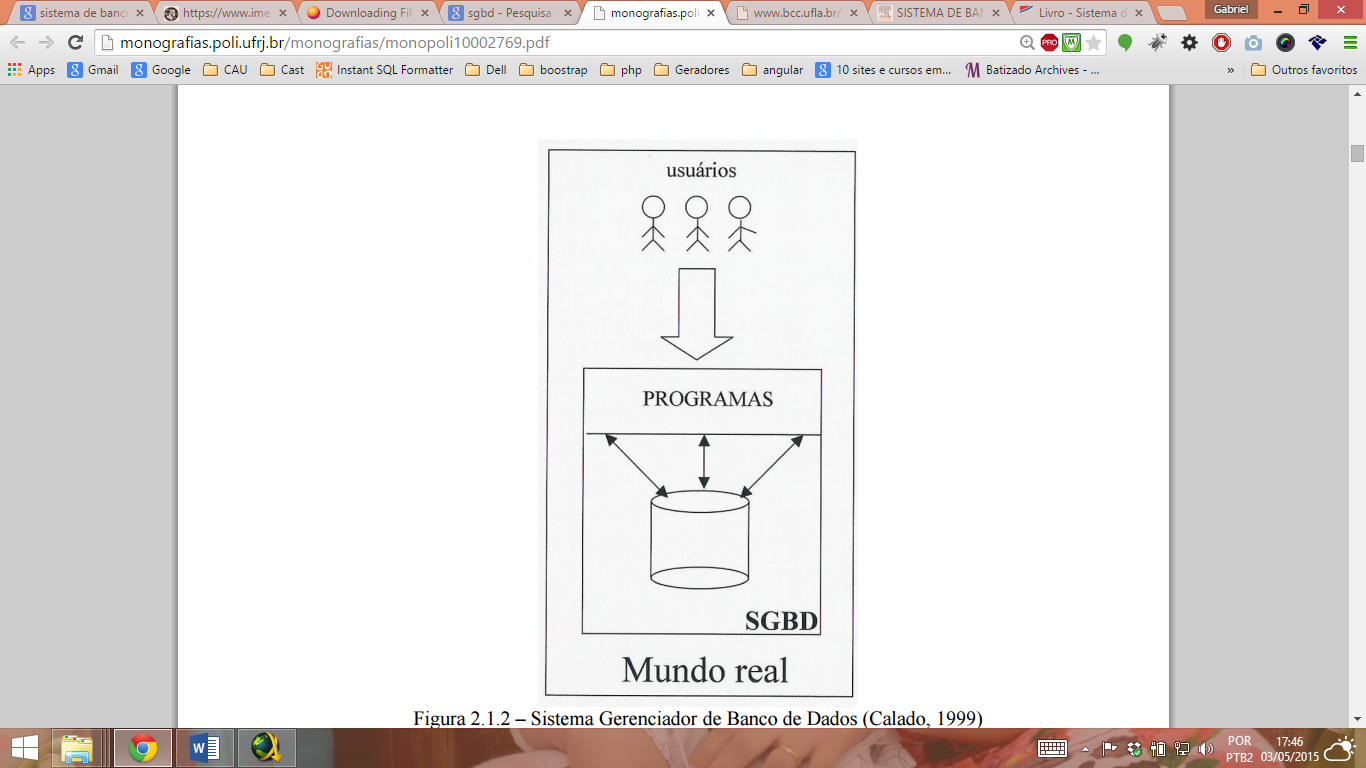
Figura 8 – Sistema de banco de dados



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.

## Linguagem de Programação PHP

Segundo Deitel e Deite (2003, p 929), “PHP é uma tecnologia de código-fonte aberto que é suportada por uma grande comunidade de usuários e desenvolvedores”.

O PHP é focado principalmente nos scripts do lado do servidor, portanto, você pode fazer qualquer coisa que outro programa CGI pode fazer, como: coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies. Mas o PHP pode fazer muito mais.

*Scripts* no lado do servidor (*server-side*). Este é o mais tradicional e principal campo de atuação do PHP. Você precisa de três coisas para isto funcionar. O interpretador do PHP (CGI ou módulo do servidor), um servidor web e um navegador web. Você precisa rodar o servidor web conectado em uma instalação PHP. Você pode acessar os resultados de seu programa PHP com um navegador web, visualizando a página PHP através do servidor web. Tudo isso pode rodar na sua máquina pessoal se você estiver apenas experimentando programar com o PHP. Veja a seção das instruções de instalação para mais informações.

*Scripts* de linha de comando. Você pode fazer um script PHP para executá-lo seu servidor web nem navegador. A única coisa necessária é o interpretador PHP. Esse tipo de uso é ideal para script executados usando o cron (on \*nix or Linux) ou o Agendador de Tarefas (no Windows). Esses scripts podem ser usados também para rotinas de processamento de texto simples. Veja a seção Utilizando o PHP em linha de comando para mais informações.

Escrever aplicações desktop. O PHP provavelmente não é a melhor linguagem para criação de aplicações desktop com interfaces gráficas, mas se você conhece bem o PHP, e gostaria de usar alguns dos seus recursos avançados nas suas aplicações do lado do cliente, você pode usar o PHP-GTK para escrever programas assim. Você também tem a possibilidade de escrever aplicações multi-plataformas desse jeito. O PHP-GTK é uma extensão do PHP, não disponibilizada na distribuição oficial.

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (incluindo HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS e provavelmente outros. O PHP também é suportado pela maioria dos servidores web atualmente. Isso inclui o Apache, o IIS e muitos outros. Também qualquer servidor web que pode utilizar o binário FastCGI do PHP, como o lighttpd e o nginx. O PHP trabalha tanto como módulo quanto como um processador CGI.

Com o PHP, portanto, você tem liberdade de escolha de sistema operacional e de servidor web. Do mesmo modo, você pode escolher entre utilizar programação estruturada ou programação orientada a objeto (OOP), ou ainda uma mistura das duas.

Com PHP você não está limitado a gerar somente HTML. As habilidades do PHP incluem geração de imagens, arquivos PDF e até animações Flash (utilizando libswf ou Ming) criados dinamicamente, on the fly. Você pode facilmente criar qualquer padrão texto, como XHTML e outros arquivos XML. O PHP pode gerar esses padrões e os salvar no sistema de arquivos, em vez de mostrá-los em tela, formando um cache no lado do servidor para seu conteúdo dinâmico.

Uma das características mais fortes e mais significativas do PHP é seu suporte a uma ampla variedade de banco de dados. Escrever uma página web consultando um banco de dados é incrivelmente simples usando uma das extensões específicas de um banco de dados (e.g., mysql), ou usando uma camada de abstração como o PDO ou conectar a qualquer banco de dados que suporte o padrão "Open Database Connection" usando a extensão ODBC. Outros bancos de dados podem utilizar cURL ou sockets, como o CouchDB.

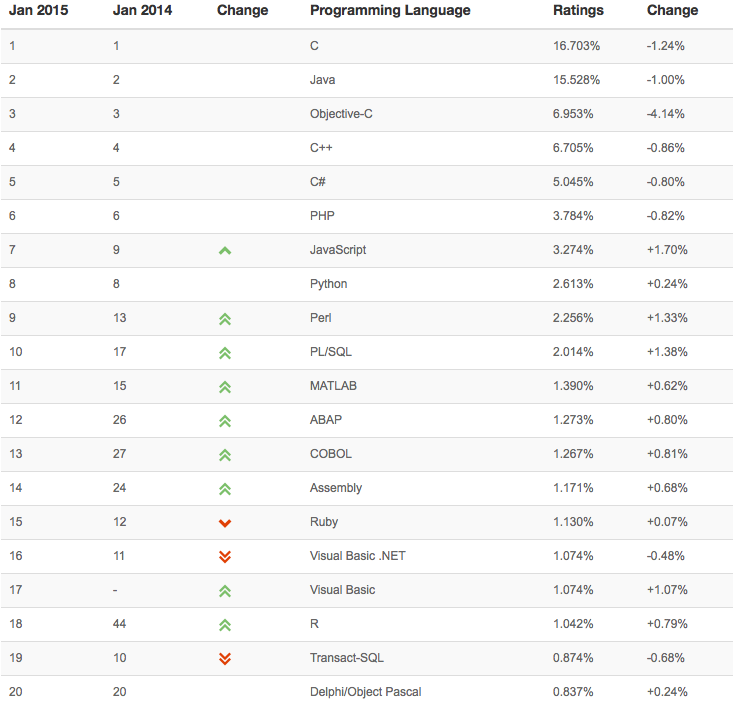
O PHP também tem suporte para comunicação com outros serviços utilizando protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (em Windows) e incontáveis outros. Você também pode abrir sockets de rede e interagir diretamente com qualquer protocolo. O PHP também suporta o intercâmbio de dados complexos WDDX, utilizado em virtualmente todas as linguagens de programação para web. Falando de comunicação, o PHP implementa a instanciação de objetos Java e os utiliza transparentemente como objetos PHP.

O PHP tem funções úteis para processamento de texto, incluindo expressões regulares compatíveis com Perl (PCRE), e muitas outras extensões e ferramentas para interpretar e acessar documentos XML. O PHP padroniza toda a extensão XML a partir da base sólida da libxml2, além de estender os recursos com o acréscimo da SimpleXML, da XMLReader e da XMLWriter.

E existem muitas outras extensões interessantes, que são categorizadas tanto alfabeticamente quanto por categoria. E existem também as extensões PECL que podem, ou não, estar documentadas dentro do próprio manual do PHP, como a » XDebug.

O PHP foi escolhido por ser uma linguagem de grande aceitação, como mostra o quadro TIOBE que informa o *ranking* das 20 linguagens de programação mais utilizadas.

Quadro 1 – Quadro TIOBE



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

## Servidor *Web*

Um programa de computador responsável por aceitar pedidos HTTP de clientes, geralmente os navegadores, e servi-los com respostas HTTP, incluindo opcionalmente dados, que geralmente são páginas web, tais como documentos HTML com objetos embutidos (imagens, etc.); Um computador que executa um programa que provê a funcionalidade descrita acima.

Os pedidos http que se referem habitualmente a páginas HTML são normalmente feitos através de browsers. O processo se inicia com a conexão entre o computador onde está instalado o servidor web e o computador do cliente; como na web não é possível prever a que hora se dará essa conexão, os servidores web precisam estar disponíveis dia e noite. A partir daí é processado o pedido do cliente, e conforme as restrições de segurança e a existência da informação solicitada, o servidor devolve os dados.

Atualmente há cada vez mais programas que fazem pedidos HTTP (leitores de RSS e outros) e quase desde o inicio da web que os ficheiros servidos pelo *webserver* vão para além dos arquivos HTML, incluindo imagens, ficheiros de som, PDF, etc. Genericamente tudo o que se enquadre no conceito de ficheiro pode ser enviado como resultado de um pedido HTTP.

Finalmente, os servidores web também podem executar programas e scripts, interagindo mais com o usuário.

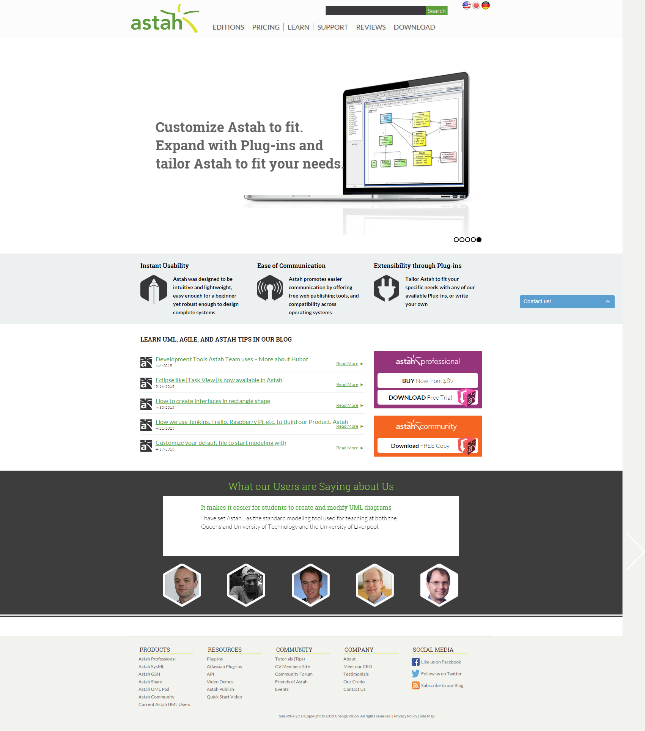
## Ferramentas

Para este projeto, foram utilizadas as ferramentas abaixo descritas:

### 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 10 - Página principal do Astah



Fonte: http://www.astah.net

### NetBeans Integrated Development Environment (IDE)

O NetBeans IDE 8.0 fornece analisadores de código e editores prontos para uso para trabalhar com as tecnologias Java 8 mais recentes -- Java SE 8, Java SE Embedded 8 e Java ME Embedded 8. O IDE também possui uma faixa de novos aprimoramentos que melhoram ainda mais seu suporte para Maven e Java EE com o PrimeFaces; novas ferramentas para HTML5, em particular para AngularJS; e aprimoramentos para o suporte a PHP e C / C++.

O NetBeans IDE 8.0 está disponível em inglês, português do Brasil, japonês, russo e chinês simplificado.

Figura 11 - Página principal do NetBeans



Fonte: http://netbeans.org

### MySQL Workbench

MySql Workbench é uma ferramenta visual unificada para Arquitetos de Banco de Dados, Programadores e DBAs. MySql Workbench fornece modelagem de dados, desenvolvimento de SQL e ferramentas de administração abrangentes para administração e configuração do usuário do servidor, backup e muito mais. MySql Workbench está disponível no Windows, Linux e Mac OS X.

Figura 12 - Página do MySQL Workbench

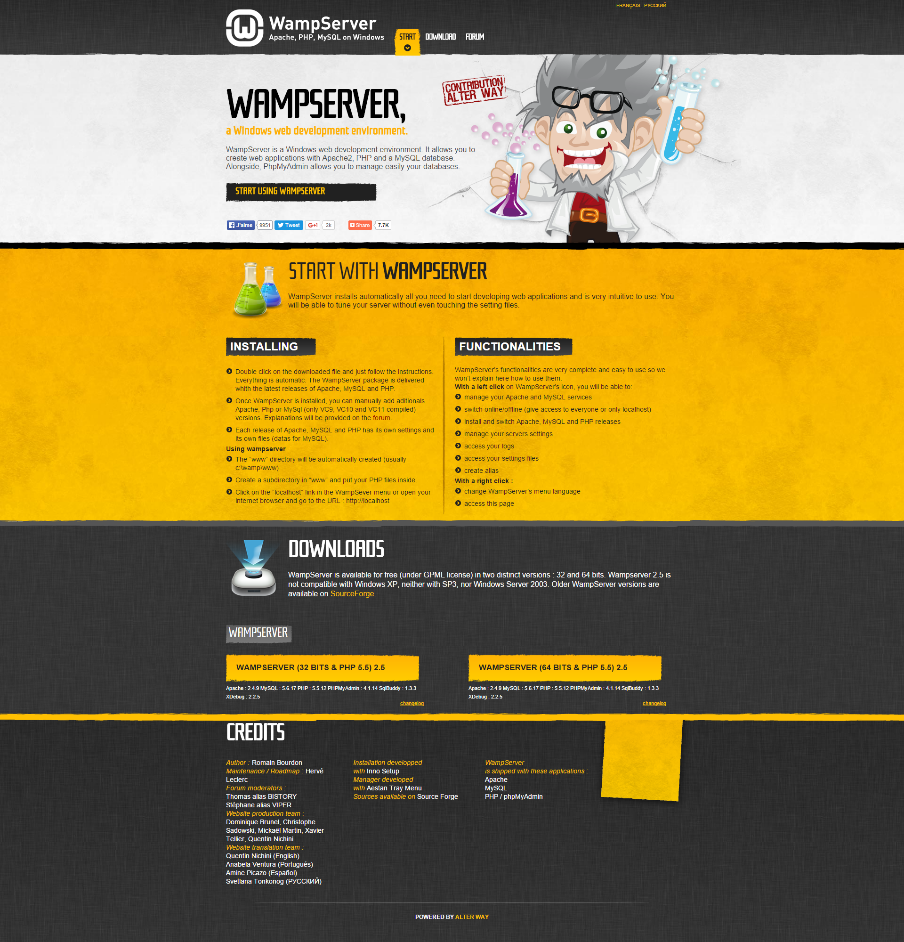


Fonte: https://dev.mysql.com/downloads/workbench/

### 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 XAMPP, o uso comercial também é gratuito. Há atualmente distribuições para Windows, Linux e OS X.

Figura 13 - Página do WAMPP



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

## 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 embaralhamento 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)

Sendo assim, constam na metodologia deste projeto a concepção do sistema, a elaboração do sistema, a construção do sistema e a transição do sistema.

## 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 irá gerenciar o 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 encontra-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



Quadro 1 – 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 |  |

### Plano de Cronograma do Projeto

**Cronograma do Projeto**

| **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 |

Quadro 2 – Descrição de Tarefas

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |  |  |

Quadro 3 - Cronograma

## Elaboração

### Análise e Projeto

O projeto de analise deve mapear as estruturas e funcionalidades identificadas na análise de requerimentos dentro do contexto e das restrições da arquitetura, de forma a tornar possível a construção do software.

### Diagrama de Atividades

Diagrama de atividade mostrando o fluxo de controle de uma atividade para outra e serão empregados para fazer a modelagem de aspectos dinâmicos do sistema.

Apêndice E

### Diagrama de Sequência

O diagrama de sequência representa essa informação de uma forma simples e lógica.

Apêndice D

### Diagrama de Classes

O Diagrama de Classe tem como principal enfoque permitir a visualização das classes que comporão o sistema STACIONE, o sistema com seus respectivos atributos e métodos, bem complementam e transmitem informações entre si. Esse diagrama apresenta uma visão estática como as classes estão organizadas, preocupando-se em como definir a estrutura lógica das mesmas. O diagrama de classe serve ainda como base para construção da maioria dos outros diagramas da linguagem UML.

O Diagrama de Classes do STACIONE encontra-se no Apêndice G deste projeto.

### Projeto de Banco de Dados

No projeto de banco de dados, normalmente são considerados dois níveis de abstração de modelo de dados, o do modelo conceitual e o do modelo lógico.

### Modelo Entidade Relacionamento

O modelo representado no Apêndice H de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Obviamente, o banco de dados contém várias entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais.

### Dicionário de dados

O dicionário de dados também encontrado no Apêndice H permite que os analistas obtenham informações sobre todos os objetos do modelo de forma textual.

## Construção

### Diagrama de Implantação

Diagrama de Implantação encontrado no Apêndice F mostra aspectos da estrutura em termos de tempo de execução.

### Teste

Os testes de implantação, qualidade, stress e segurança bem como o plano e o manual de implantação e o treinamento de usuários serão tratados neste projeto após a realização da programação do sistema.

### Testes de Qualidade

<texto geral>

<< Apêndice I >

### Testes de Stress

<texto geral>

### Testes de Segurança

<texto geral>

## Implantação

<texto geral>

### Plano de Implantação

### Manual de Implantação

### Treinamento de Usuários

# CONCLUSÃO

Por ser um sistema web voltado para a automação de serviços e principalmente quanto à segurança do sistema, será adotada a segurança por ‘perfis’. Ao ser cadastrado um novo usuário no sistema, um *login* e senha será criado limitando este usuário às suas respectivas funcionalidades.

Outra grande vantagem do STACIONE é a geração de relatórios. Estes relatórios facilitarão a visualização de ações importantes realizadas no sistema, como fluxo de caixa, alteração de preços e resultados.

Tendo como base esse documento será possível a implementação do STACIONE solicitado pela empresa Estacionamentos LTDA.

O STACIONE servirá como um diferencial, podendo ser expandido para outros estacionamentos. Futuramente, o sistema passará por diversas melhorias, de acordo com os requisitos dos usuários, dentre elas, o cadastro de mensalistas e leitor de código de barra, e conforme for sendo observado pelo cliente.

# REFERÊNCIAS

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

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

Consórcio *World Wide Web* (W3C). Disponível em: http://www.w3c.br Acesso em: maio 2015.

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

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

PHP. Disponível em: <http://php.net/> . Acesso em: mai/2015.

DBDesigner. Disponível em: http://fabforce.net/dbdesigner4/. Acesso em: jun/2015.

Eclipse. Disponível em: http://www.eclipse.org. Acesso em: mai/2015.

Bootstrap. Disponível em: getbootstrap.com. Acesso em: mai/2015.

Xampp. Disponível em: https://www.apachefriends.org/pt\_br/Acesso em: mai/2015.

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.

Astah. Disponível em: http://www.astah.net Acesso em: mai/2015.

Net Beans. Disponível em: https://netbeans.org Acesso em: mai/2015.

MySQL Workbench. Disponível em: https://dev.mysql.com/downloads/workbench/ Acesso em: mai/2015.

PONTE, João Pedro (2006). **Estudos de caso em educação matemática.** Bolema, 25, 105-132. Este artigo é uma versão revista e atualizada de um artigo anterior: Ponte, J. P. (1994). O estudo de caso na investigação em educação matemática. Quadrante, 3(1), pp3-18. (re-publicado com autorização).

# 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 Estaciomentos 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. |

|  |  |
| --- | --- |
| O problema de | Processo atual de controle de caixa e relatórios não atende os anseios da empresa. |
| afeta | Os proprietários e os gerentes da Empresa Estacionamentos. |
| cujo impacto é | Distorções e erros gerando perdas de receita na empresa. |
| uma boa solução seria | Criar e automatizar o processo de controle de caixa e relatórios, 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.

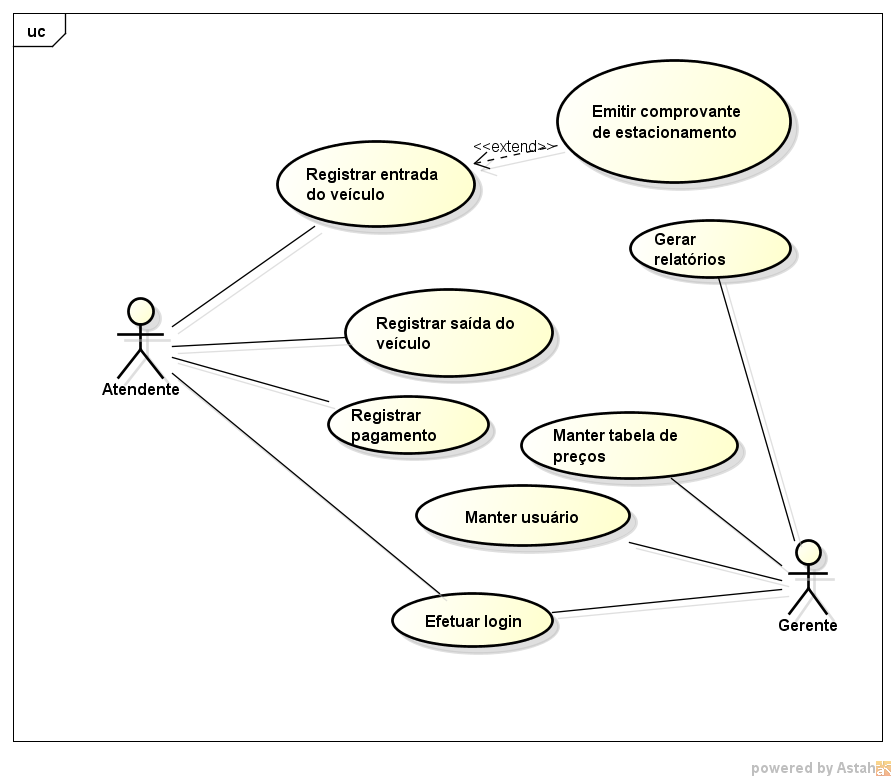
**3.4** **Principais Necessidades dos Usuários ou dos Envolvidos**

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

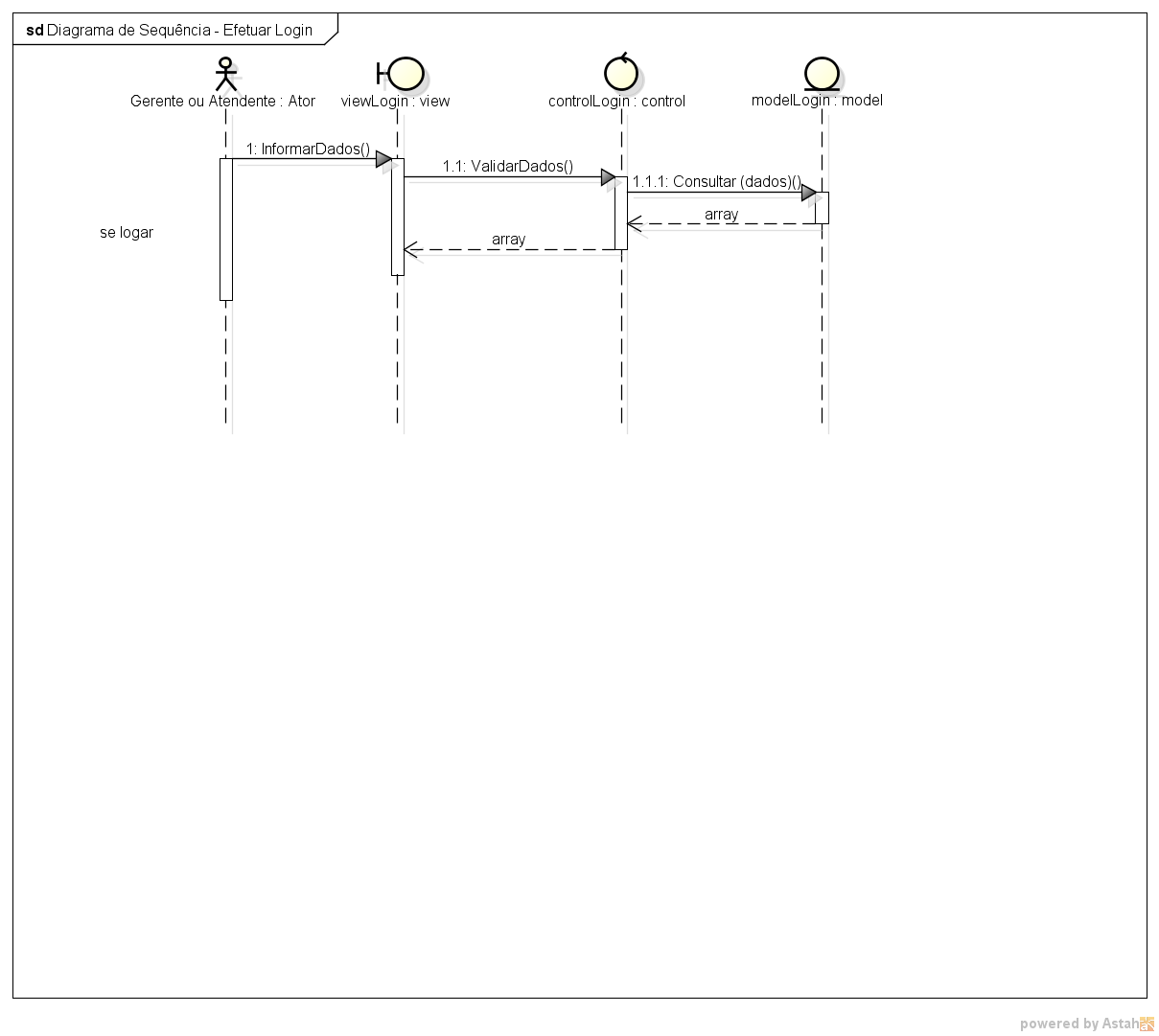


Diagrama de Sequência – Emitir comprovante

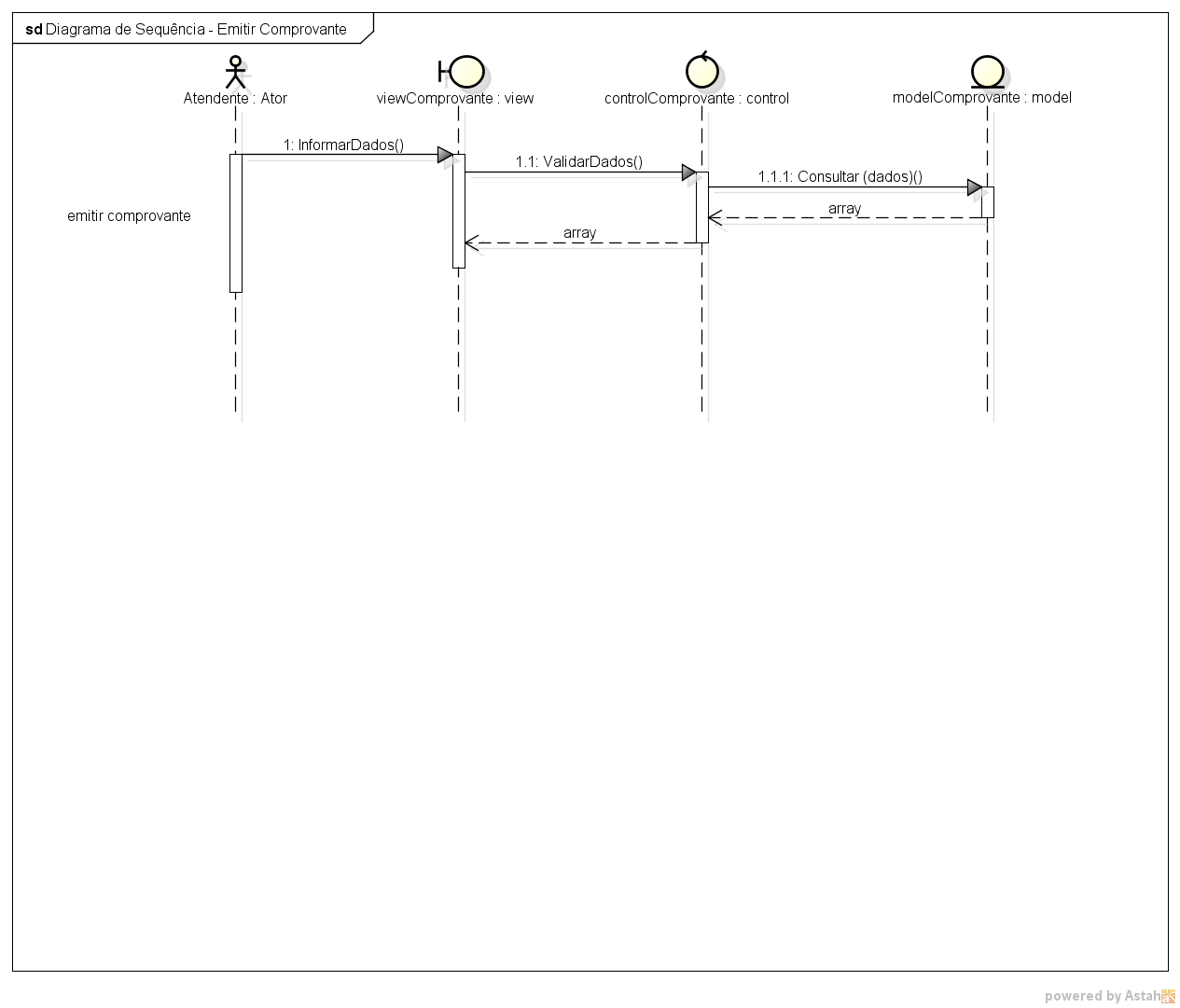


Diagrama de Sequência – Gerar relatórios

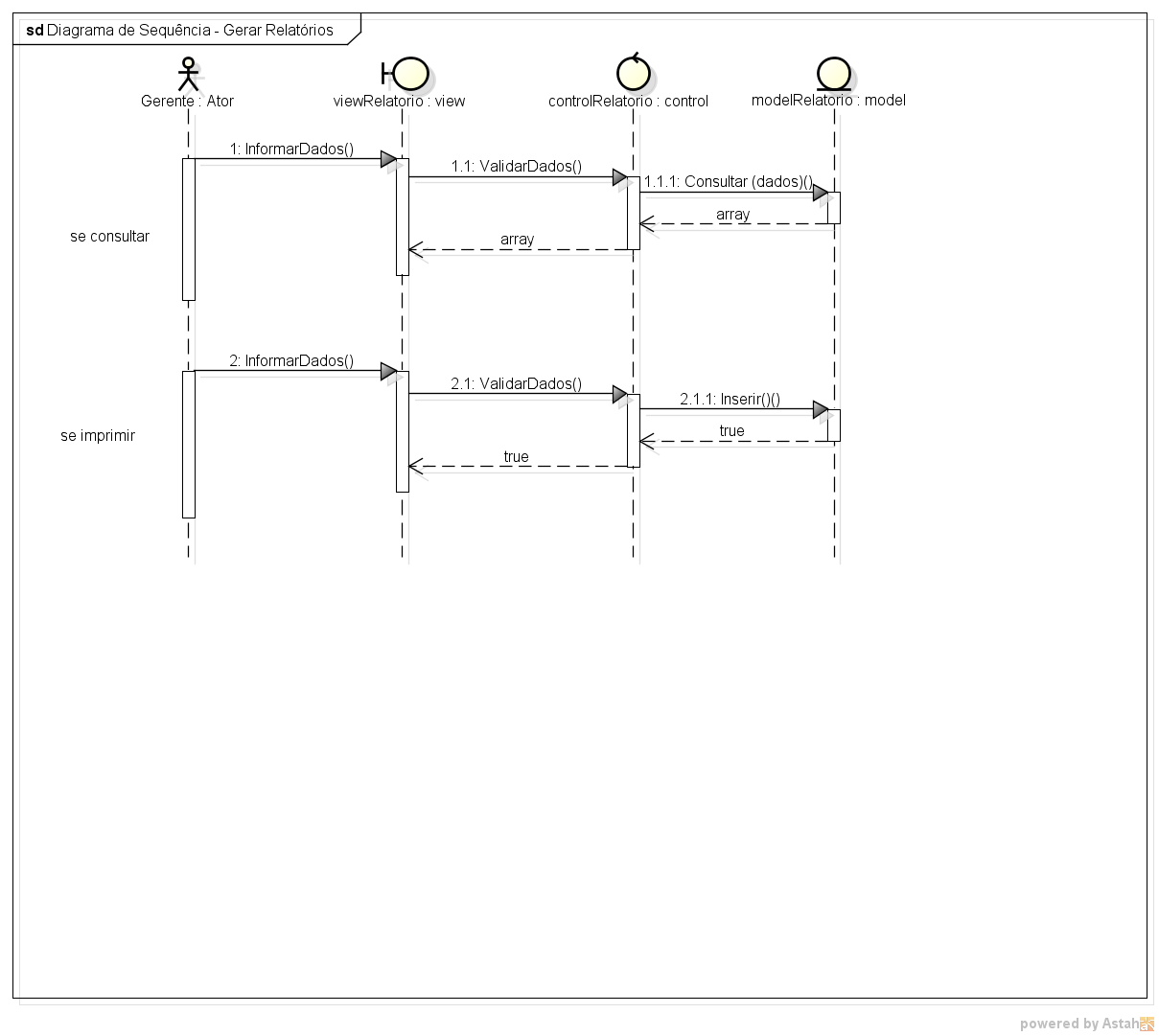


Diagrama de Sequência – Manter Funcionário

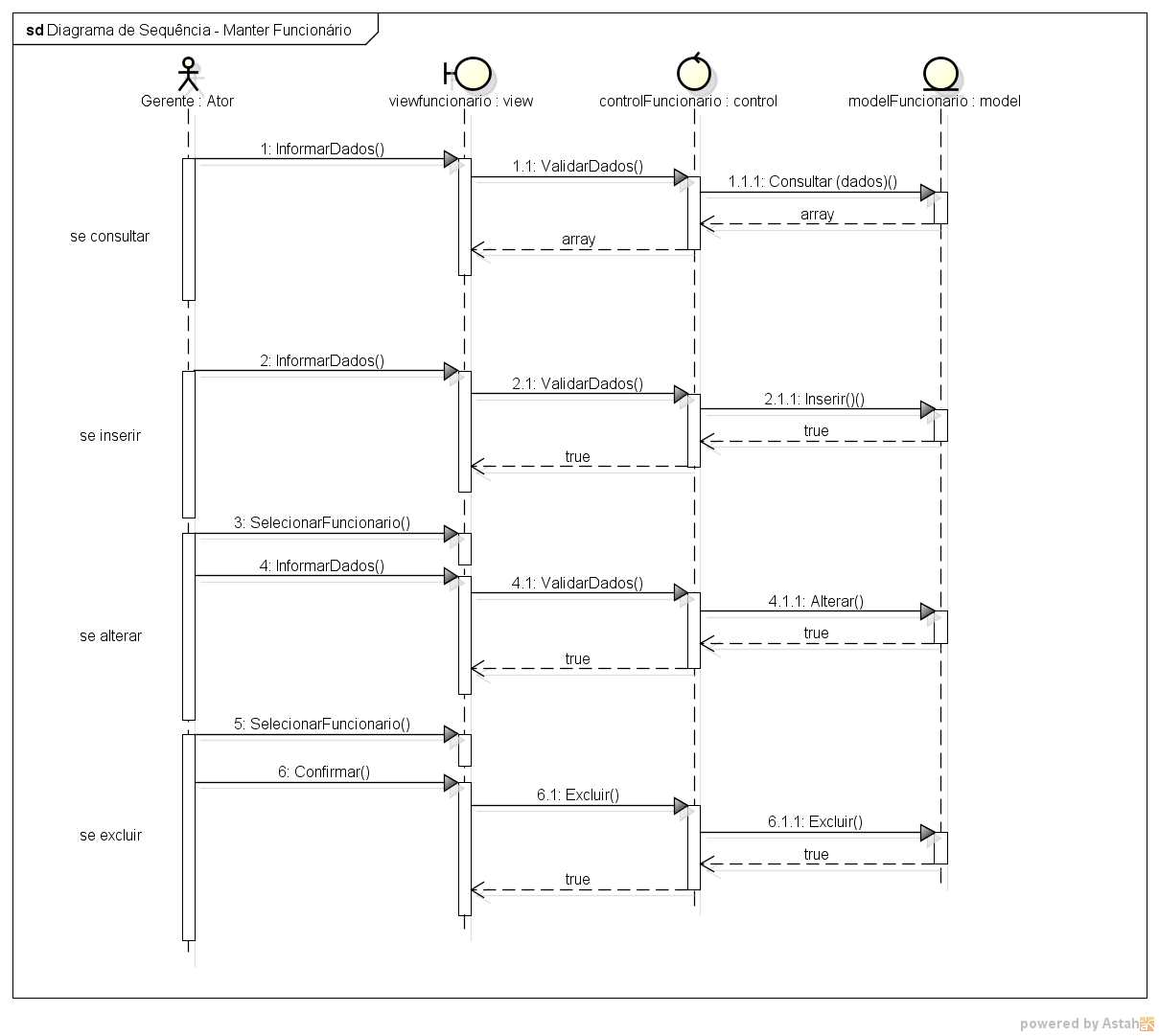


Diagrama de Sequência – Manter Tabela de Preços

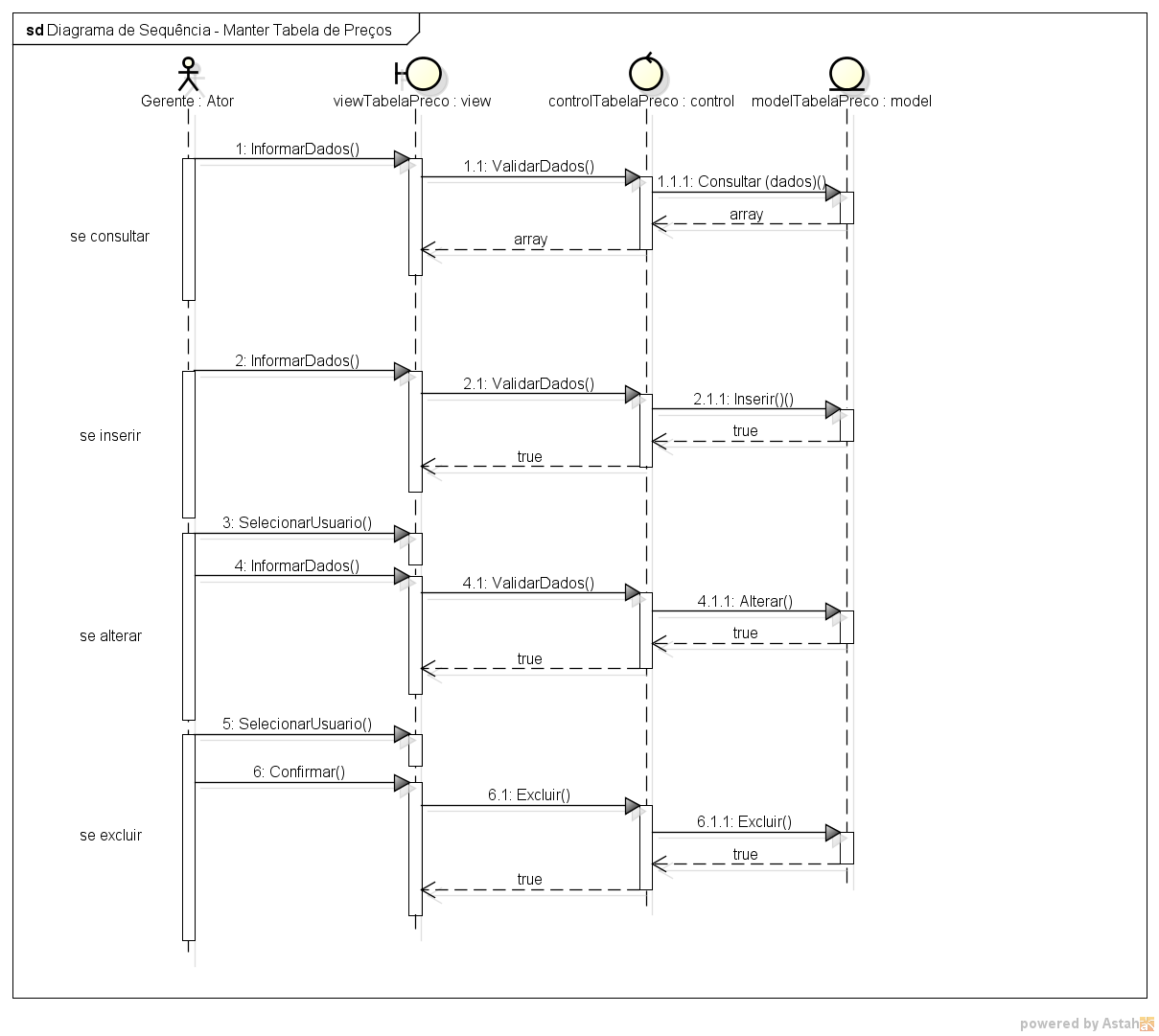


Diagrama de Sequência – Registrar Entrada

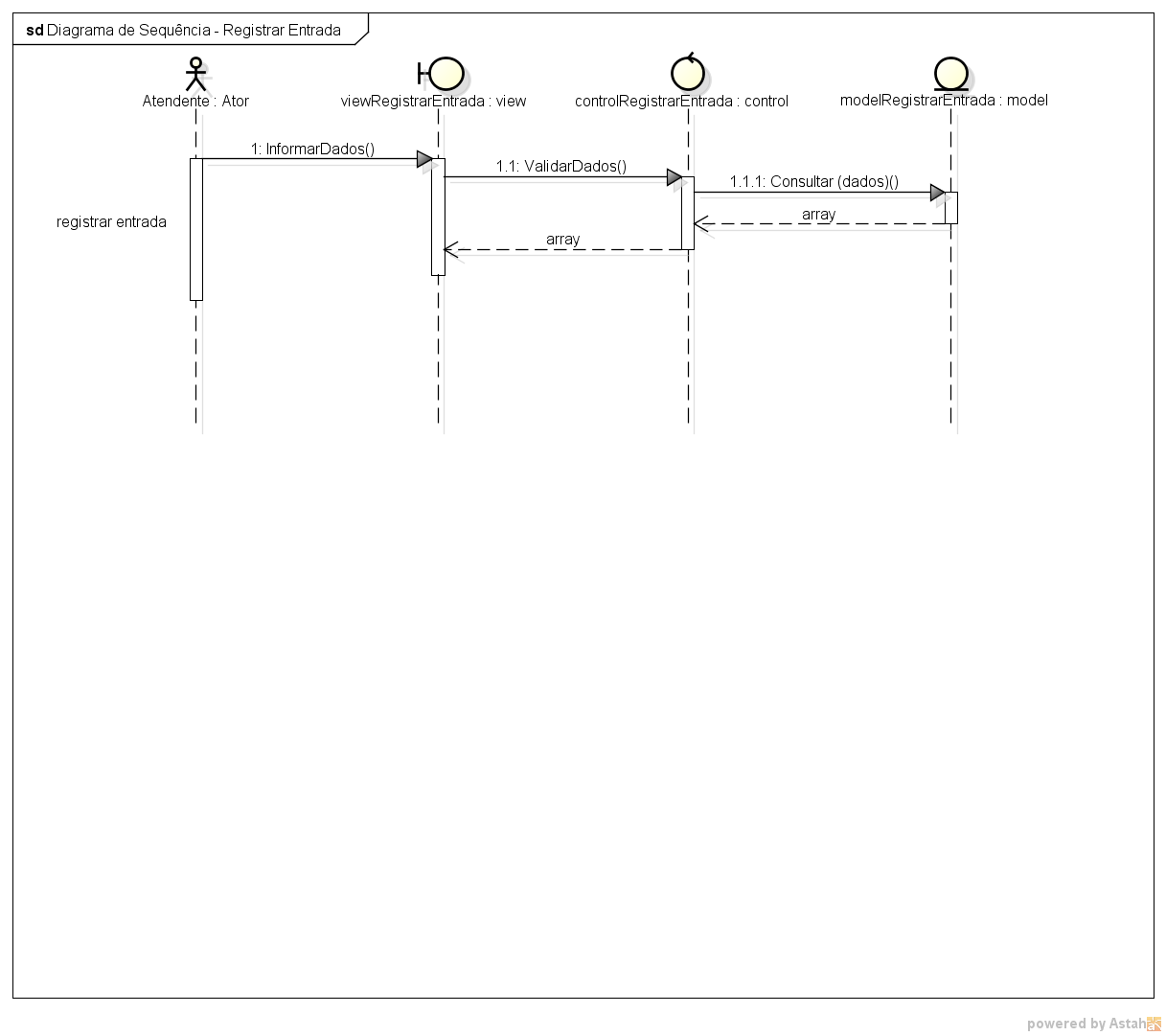


Diagrama de Sequência – Registrar Pagamento

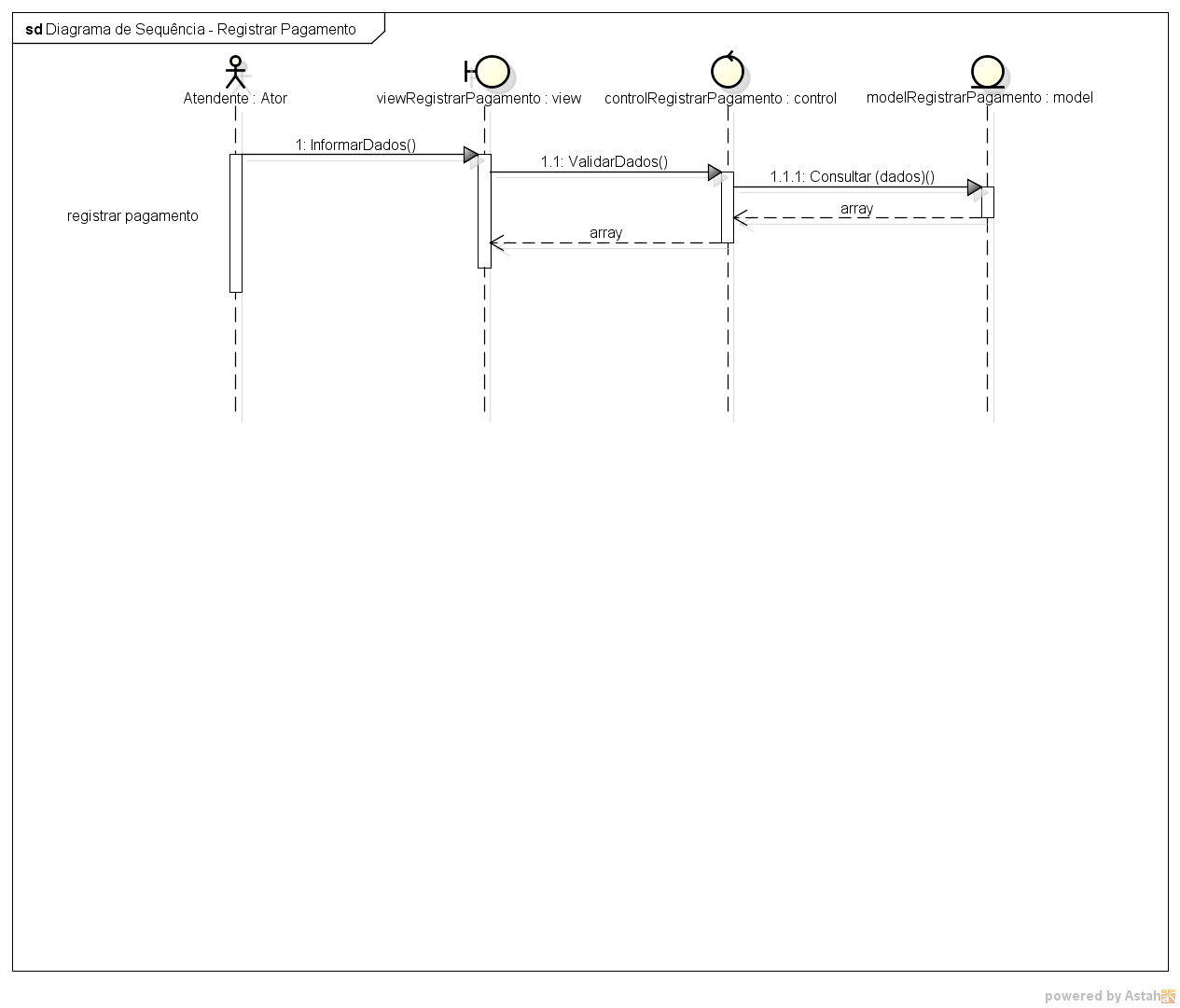
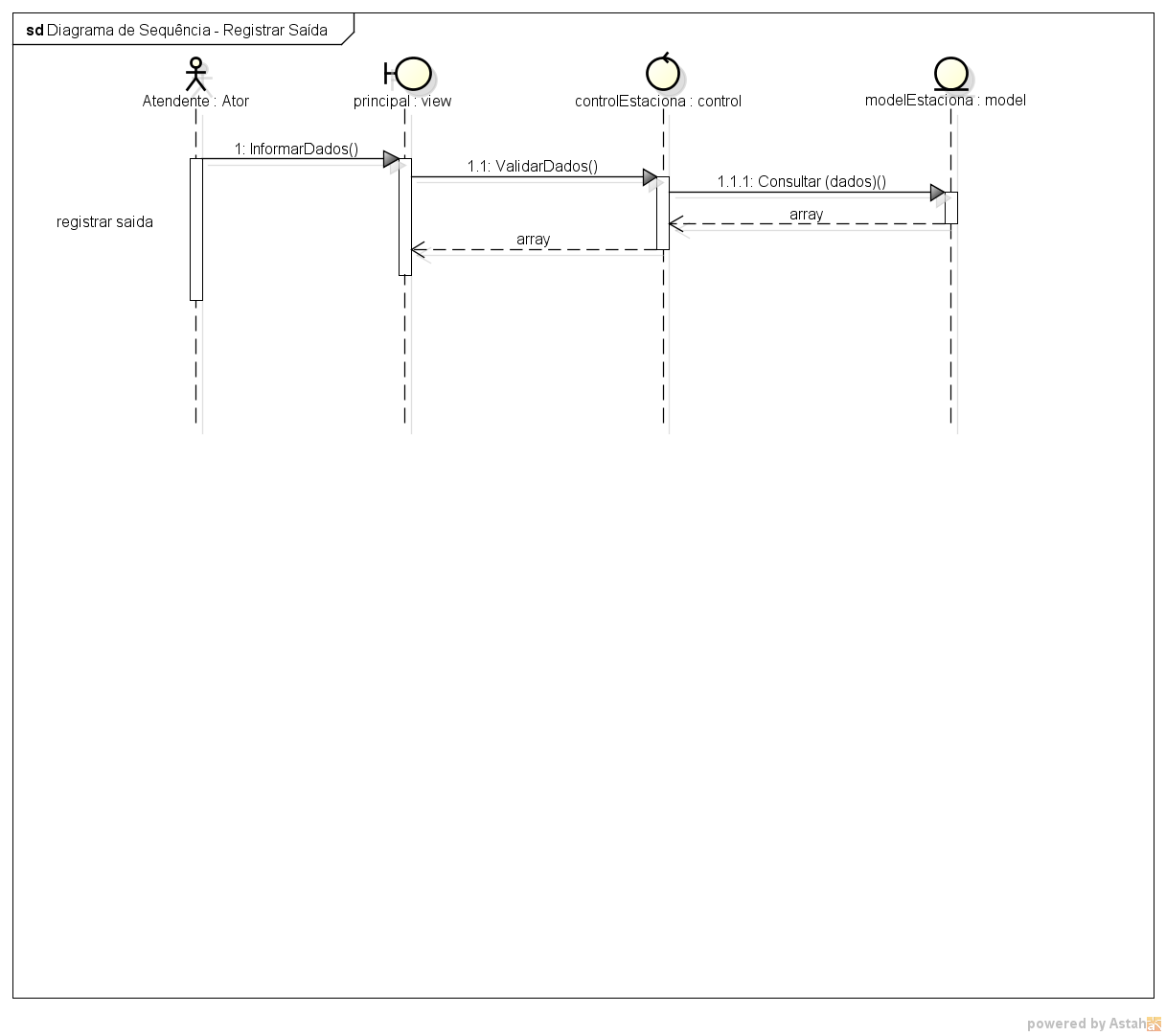
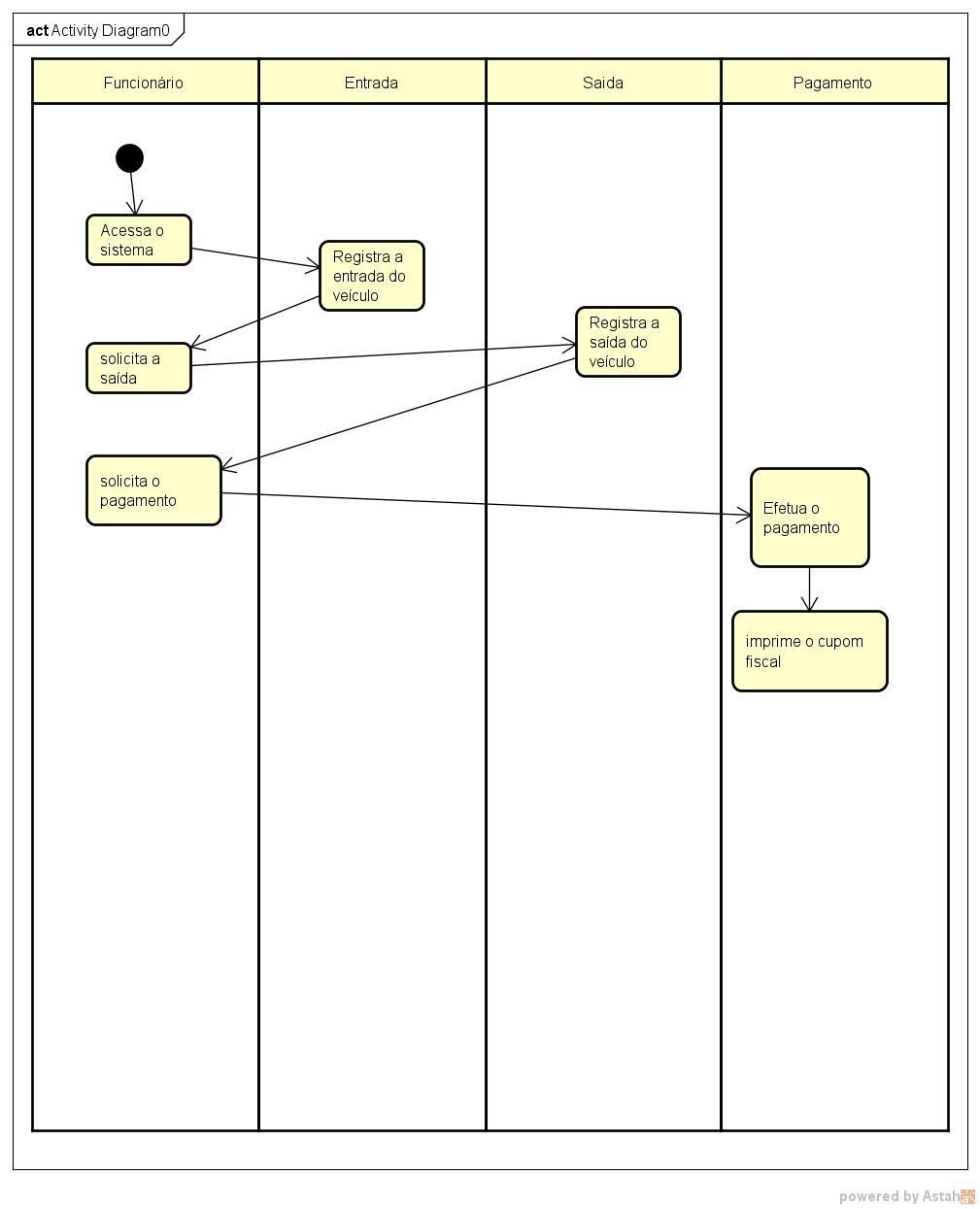


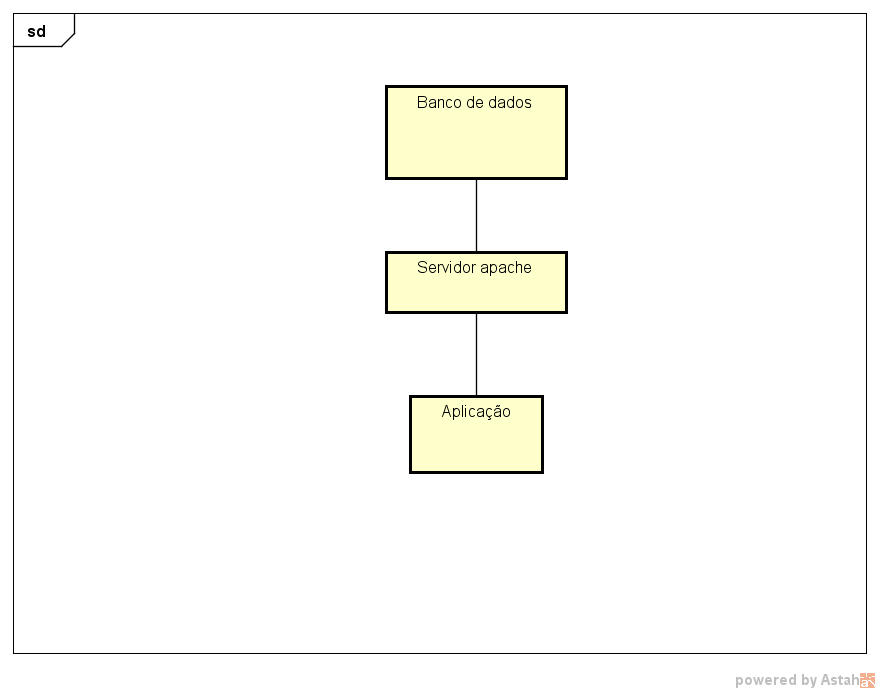
Diagrama de Sequência – Registrar Saída



# 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

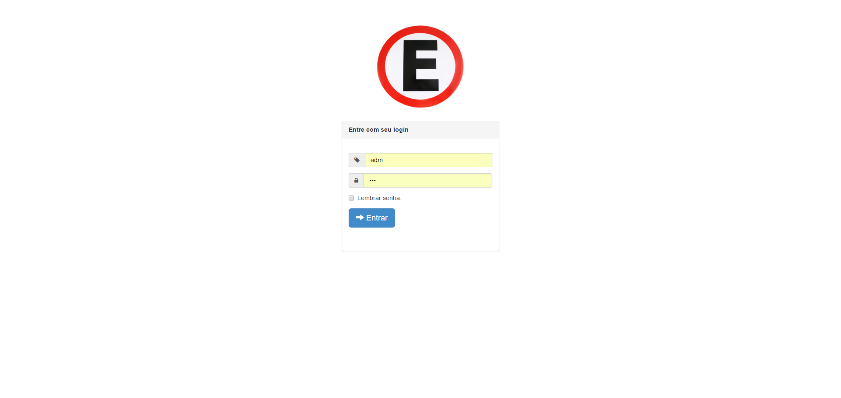
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TB\_IMAGEM** | | | | | | | | | |
| Tabela para guardar as informações da imagem | | | | | | | | | |
| **Coluna** | **Tipo** | **Tamanho** | **PK** | | **FK** | | **Obrigatório** | | **Descrição** |
| Sim | 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

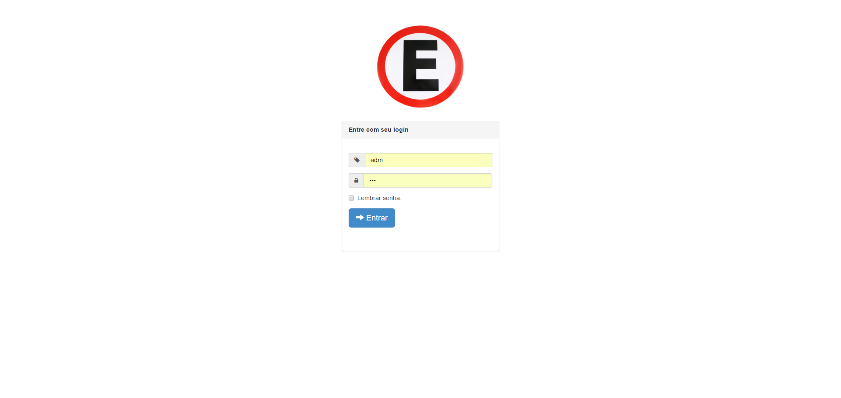
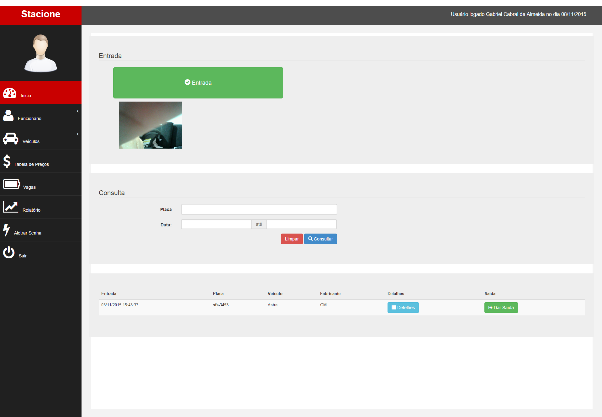
|  |  |
| --- | --- |
| **Nome: Login** | **Executor de Teste: Gabriel Cabral de Almeida** |

# ID 1

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



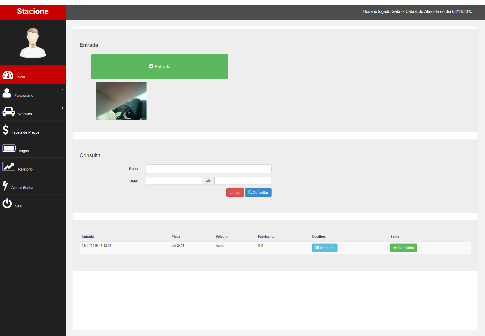
**2o Passo – Login e Senha (Clicar em Entrar)**

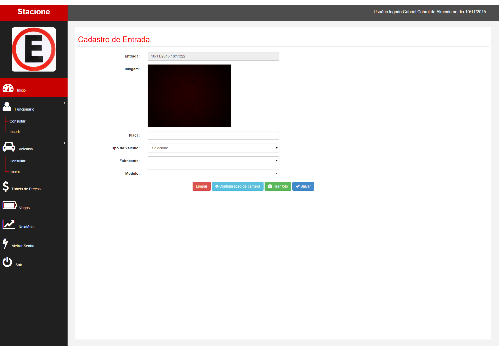
 **3o Passo – Acesso ao sistema  
**

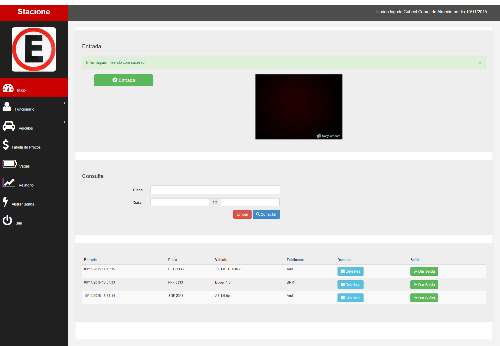
|  |  |
| --- | --- |
| **Nome: Entrada no Estacionamento** | **Executor de Teste: Gabriel Cabral de Almeida** |

# ID 1

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

****

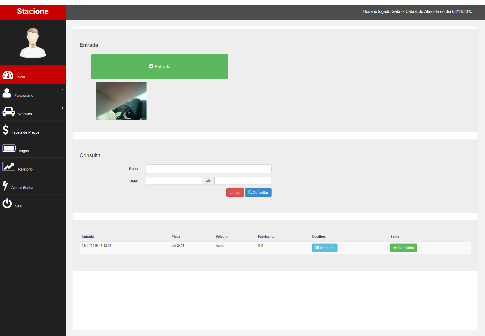
**2o Passo – inserir os dados do veículo (Clicar em salvar)  
**

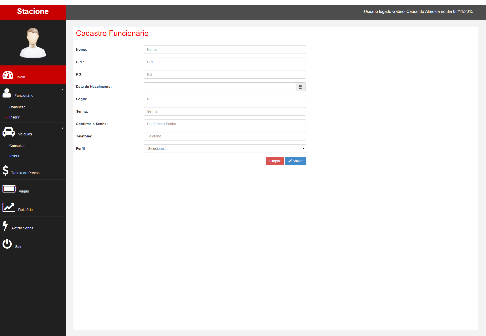
**3o Passo – Acesso ao sistema  
**

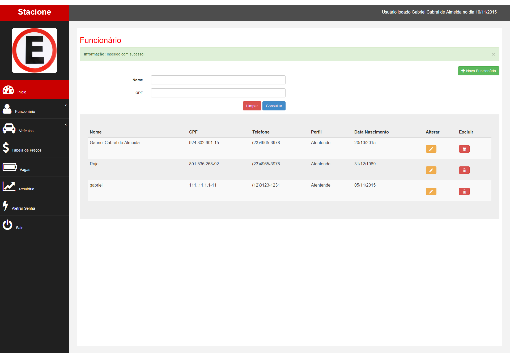
|  |  |
| --- | --- |
| **Nome: cadastro funcionário** | **Executor de Teste: Gabriel Cabral de Almeida** |

# ID 1

**1o Passo – Acesso a cadatro de funcionario (Clicar em Funcionário e inserir)**

****

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

**3o Passo – Acesso ao sistema  
**

# ANEXO A