**Especificação do Sistema Teste-Back-End**

**Desenvolvido por:**

**Áquilla Odlanier Faria Nascimento**

**Januária/MG**

**Maio 2019**

**Especificação do Sistema Teste-Back-End**

Este documento apresenta a especificação do Sistema Teste-Back-End, que tem como principal função rastrear possíveis clientes do site icasei. Para a construção do sistema foi feita a análise de requisitos, aplicação de técnicas de modelagem do esquema das classes.

**Januária/MG**

**Maio 2019**

**Índice**

[**1.** **Introdução** 4](#_Toc436299616)

[**1.1** **Introdução à situação problema** 4](#_Toc436299617)

[**1.2** **Mini Mundo** 5](#_Toc436299618)

[**1.3** **Requisitos** 5](#_Toc436299619)

[**1.3.1** **Requisitos funcionais e não funcionais** 5](#_Toc436299620)

[**1.4** **Objetivos** 6](#_Toc436299621)

[**1.4.1** **Objetivos Gerais** 6](#_Toc436299622)

[**1.4.2 Objetivos Específicos** 7](#_Toc436299623)

[**2.** **Análise** 7](#_Toc436299624)

[**2.1** **Diagrama Conceitual** 7](#_Toc436299625)

[**2.2** **Diagrama Lógico do Banco de dados.** 8](#_Toc436299626)

[**2.3** **Diagrama de Pacotes** 8](#_Toc436299627)

[**2.4** **Diagrama de Classes** 9](#_Toc436299628)

[**2.5** **Diagrama de casos de uso** 10](#_Toc436299629)

[**2.6** **Especificação de casos de uso** 10](#_Toc436299630)

[**2.7** **Diagrama de estados** 13](#_Toc436299631)

[**2.7.1 Diagrama de estados – Login** 13](#_Toc436299632)

[**2.7.2 Diagrama de estados – Venda** 14](#_Toc436299633)

[**2.7.3 Diagrama de estados – Entrega da venda** 14](#_Toc436299634)

[**2.8** **Diagrama de sequencia** 15](#_Toc436299635)

[**2.8.1 Caso de uso – Login** 15](#_Toc436299636)

[**2.8.2 Caso de uso – Cadastrar produto** 15](#_Toc436299637)

[**2.8.3 Caso de uso – Atualizar Estoque** 16](#_Toc436299638)

[**2.8.4 Caso de uso – Venda** 16](#_Toc436299639)

[**2.9** **Diagrama de Implantação e componentes** 17](#_Toc436299640)

[**3.** **Desenvolvimento** 17](#_Toc436299641)

[**3.1** **Ambiente ALM** 17](#_Toc436299642)

[**3.1.1 Linguagem** 17](#_Toc436299643)

[**3.1.2 IDE**  17](#_Toc436299644)

[**3.1.3 Controle de Versão** 18](#_Toc436299645)

[**3.1.4 SGBD** ...................................................................................................................18](#_Toc436299646)

[**3.1.5 Ferramenta para Issue/Bug Tracking** 18](#_Toc436299647)

[**3.1.6 Criação de Diagramas UML** 19](#_Toc436299648)

[**3.2** **Arquitetura de Software** 19](#_Toc436299649)

[**3.3** **API's**............................................................................................................................20](#_Toc436299650)

[**3.4** **Protótipo de telas** 22](#_Toc436299651)

[**3.4.1** **Tela Inicial** 22](#_Toc436299652)

[**3.4.2 Tela de cadastro de produto** 23](#_Toc436299653)

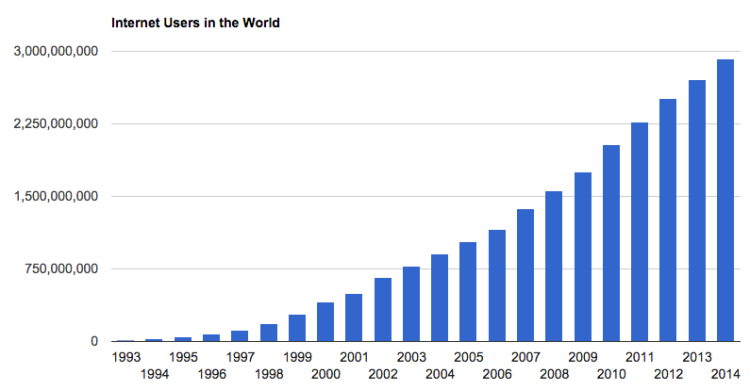
[**3.4.3 Tela de compra de um jogo** 24](#_Toc436299654)

[**4.** **Referencias** 24](#_Toc436299655)

# 

# **Introdução**

## **Introdução à situação problema**

O uso da internet no mundo está crescendo rapidamente, só nos últimos 10 anos ela cresceu cerca de 44.4%, chegando a aproximadamente 2 bilhões de usuários, na figura 1 pode ser visto a quantidade em milhões de usuários da internet por região. Assim pode-se constatar que com o advento da internet, o rastreio de seus usuários a procura de um novo produto ofertado por uma empresa especifica, mudou consideravelmente, vendo isso o mercado das empresas de casamento, que tem por principal função ofertar ao cliente uma série de serviços nesse setor, aumentou bastante. Assim se faz necessário o uso de um sistema para melhor rastrear os dados, visto que a quantidade de possíveis clientes também cresceu. O sistema deve ser capaz de agilizar os processos da empresa, visando um melhor atendimento aos clientes e maior rapidez nas pesquisas de seus produtos atuais e vindouros.

**Gráfico 01 – Interent Users in the World**

**Fonte:[3]**

## **Mini Mundo**

A empresa de **Icasei**, especializada em casamentos, vai colocar um novo produto no mercado chamado CaseMais.

Com o novo produto CaseMais da Icasei o cliente acessa um sistema que mantem registrado seus passos dentro dele.

## **Requisitos**

## **Requisitos funcionais e não funcionais**

**Responsável**

|  |  |
| --- | --- |
| **Nome** | **Assinatura** |
| Áquilla Odlanier Faria Nascimento | Áquilla Odlanier Faria Nascimento |

**Requisitos funcionais e não funcionais**

|  |  |
| --- | --- |
| F1 Cadastrar contatos | **Oculto ()** |
| **Descrição:** Cadastra os contatos. |  |
| **Requisitos não funcionais** |  |
| **Nome** | **Restrição** |
| NF1.1 Nome | O Nome deve ser obrigatório. |
| NF1.1 Email | O Email deve ser obrigatório |
| NF1.1 Telefone | O Telefone deve contar no mínimo 8(oito) caracteres no máximo 9(nove). |

|  |  |
| --- | --- |
| F2 Rastrear visitante | **Oculto ()** |
| **Descrição:** Rastrear os visitantes e possíveis clientes. |  |
| **Requisitos não funcionais** |  |
| **Nome** | **Restrição** |
| NF1.1 GUID | O GUID deve ser obrigatório. |
| NF1.1 URL | O URL deve ser obrigatório |
| NF1.1 Data | O Data deve ser obrigatório. |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## **Objetivos**

### **Objetivos Gerais**

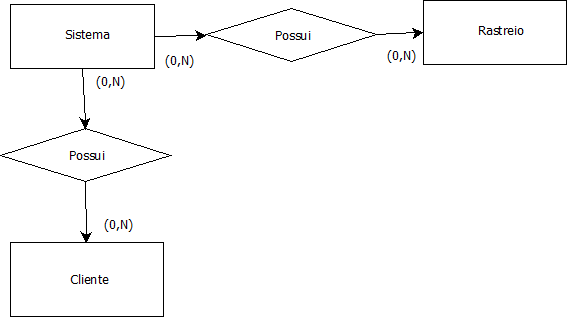
O Sistema CaseMais tem por objetivo ser uma solução rápida e eficaz para a empresa Icasei. Funcionando como um sistema online de rastreamento de futuros clientes. O objetivo geral do sistema é ser uma solução para rastreamento.

### **Objetivos Específicos**

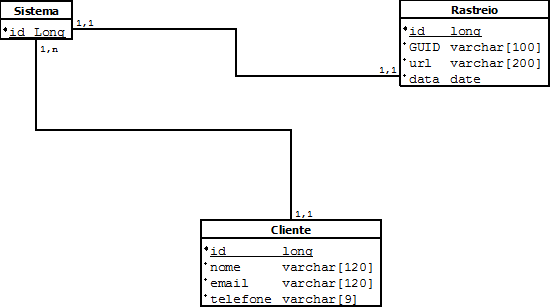
* Cadastrar clientes
* Rastrear futuros clientes

# **Análise**

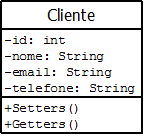
## **Diagrama Conceitual**



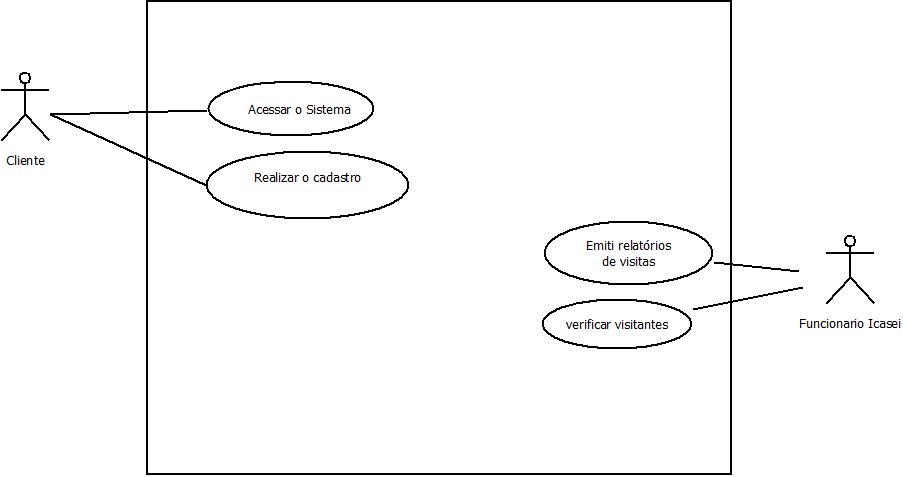
## **Diagrama Lógico do Banco de dados.**



## **Diagrama de Classes**



## **Diagrama de casos de uso**



## **Especificação de casos de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Ator(es)** | **Descrição** | **Referência Cruzada** |
| CDU1 : Acessar o sistema | Cliente | O cliente acessa o sistema via web. |  |
| CDU2 : Realizar Cadastro | Cliente | O cliente realiza o cadastro no sistema com seu email e nome. |  |
| CDU3 : Emitir relatório de visitas | Funcionário Icasei | O funcionário emite um relatório com as datas decrescentes |  |
| CDU4 : Verificar visitantes | Funcionário Icasei | O funcionário verifica quem acessou o sistema e se cadastrou |  |

**Especificação**

|  |  |
| --- | --- |
| **Caso de uso** | **1 – Acessar o sistema** |
| **Pré-requisito** | PR1 – O usuário deve acessar a área do sistema referente a tela inicial, clicando no link especifico para isso. |
| **Fluxo Principal** | 1 – O usuário deverá acessar a página inicial do sistema. |

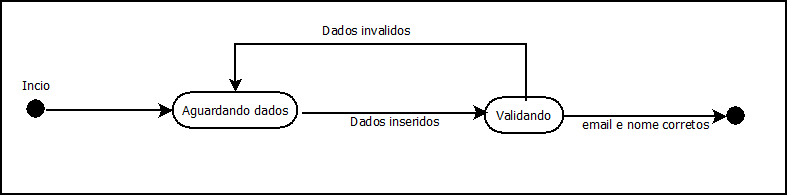
|  |  |
| --- | --- |
| **Caso de uso** | **2 – Realizar cadastro** |
| **Pré-requisito** | PR1 – O usuário acessa a área de cadastro do sistema. |
| **Fluxo Principal** | FP1 – O usuário deve inserir os dados, nome, email e telefone.  FP2 – O usuário deve apertar o botão de cadastrar. |
| **Fluxo Alternativo** | FA1 – Caso algum dado seja invalido, demonstrar um erro na tela para o usuário. |

|  |  |
| --- | --- |
| **Caso de uso** | **3 – Emitir Relatório de visitas** |
| **Pré-requisito** | PR1 – O funcionário acessa a área de relatórios do sistema. |
| **Fluxo Principal** | FP1 – O funcionário deve clicar na opção de visualizar relatório. |
| **Fluxo Alternativo** | FA1 – Caso não haja nenhuma visita o botão exibe uma mensagem de que não há relatório. |

|  |  |
| --- | --- |
| **Caso de uso** | **4 – Verificar visitantes** |
| **Pré-requisito** | PR1 – O funcionário acessa a área de relatórios do sistema. |
| **Fluxo Principal** | FP1 – O funcionário deve clicar na opção de verificar visitantes. |
| **Fluxo Alternativo** | FA1 – Caso não haja nenhuma visita o botão exibe uma mensagem de que não há relatório. |

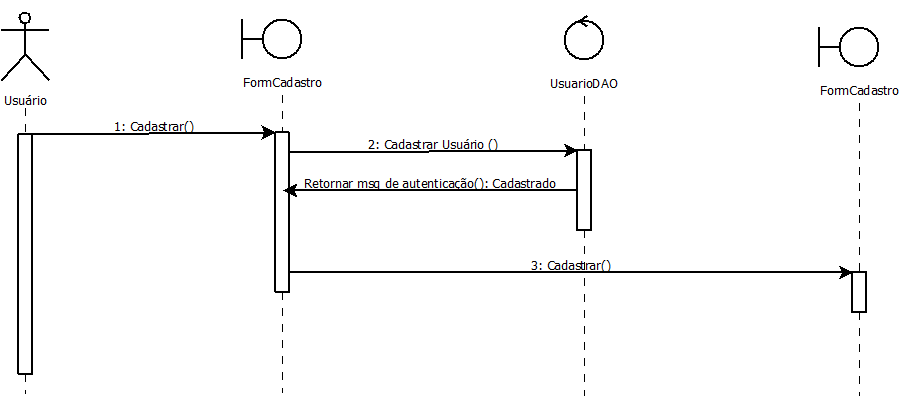
## **Diagrama de estados**

### **Diagrama de estados – Cadastrar no sistema CaseMais**

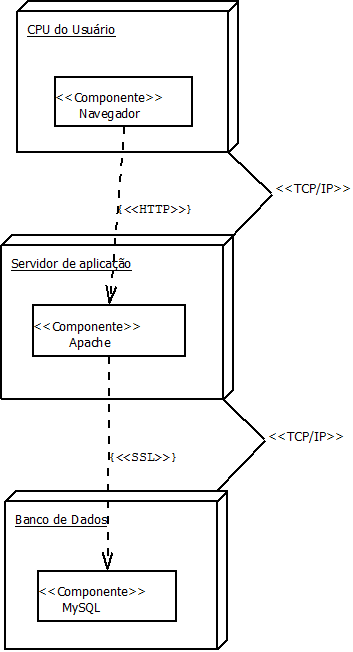


## **Diagrama de sequencia**

### **Caso de uso – Cadastro no sistema CaseMais**



## **Diagrama de Implantação e componentes**



# **Desenvolvimento**

## **Ambiente ALM**

### **Linguagem**

A linguagem utilizada na programação será JavaScript por ser uma linguagem de programação orientada a objetos. Segundo [4] e [5] ela possui portabilidade que facilita muito a distribuição do sistema. Além de ser uma linguagem, é uma plataforma de desenvolvimento, pois com ele é possível desenvolver aplicações para desktop, celular, cartão, web e etc.

Existem uma variedade de frameworks para JavaScript, que facilitam e muito o trabalho do desenvolvedor.

### **IDE**

A IDE a ser usada será o ATOM, porque segundo [6] e [7] ela oferece suporte abrangente e de primeira classe para as tecnologias e aprimoramentos Java mais recentes, antes de outros IDEs.

O ATOM oferece ainda analisadores de código e editores para trabalhar com as últimas tecnologias.

Com seu Editor JavaScript em constante aprimoramento, muitas funcionalidades avançadas e uma extensa linha de ferramentas, modelos e exemplos, o ATOM define o padrão de desenvolvimento com suas tecnologias inovadoras.

### **Controle de Versão**

A ferramenta para controle de versão irá ser o Git, pois desta [8] e [9] possui de forma gratuita todas as vantagens de controle de versão,como: Backup automático de todos os arquivos, controle de histórico, possibilidade de se trabalhar em equipe, possuir a funcionalidade de marcação e resgate de versões estáveis e permitir a ramificação do projeto.

### **SGBD**

O Sistema Gerenciador de Banco de Dados (SGBD) a ser implementado será o MySQL, porque conforme [10] é um banco de dados conhecido por sua facilidade de uso, sendo ele usado pela NASA, HP, Bradesco, Sony, e muitas outras empresas. Sua interface simples, e também sua capacidade de rodar em vários sistemas operacionais, são alguns dos motivos para ser tão usado atualmente, e seu uso estar crescendo cada vez mais.

Segundo [11] o MySQL é protegido por uma licença de software livre, desenvolvida pela GNU. É também um dos programas que vem geralmente instalado com o GNU/Linux. Uma das vantagens dele em relação a outros bancos de dados do mesmo porte é possuir uma facilidade maior para programação, ter funções mais simples, poder ser totalmente modificado, entre outras.

### **Ferramenta para Issue/Bug Tracking**

Para [12] o Github é o maior anfitrião de código do planeta, com mais de 12,6 milhões de repositórios. Sendo grande ou pequeno, cada repositório vem com as mesmas ferramentas. A ferramenta issue tracker é flexível e por isso permite que você fique em cima dos erros e se concentre nos recursos. Ainda segundo [13] estas ferramentas são abertas à comunidade para projetos públicos e seguras para projetos privados.

### **Criação de Diagramas UML**

O programa a ser usado para a criação dos diagramas UML será o DIA, ele é inspirado no programa comercial do Windows "Visio", embora seja mais voltado para diagramas informais para uso casual. Ele pode ser utilizado para desenhar diversos tipos diferentes de diagramas. Atualmente tem objetos especiais para ajudar a desenhar diagramas de relacionamento de entidade, diagramas UML, fluxogramas, diagramas de rede, e muitos outros diagramas. Também é possível adicionar suporte para novos formatos de gravação de arquivos XML simples, usando um subconjunto de SVG para desenhar [14].

Para [15] Ele pode carregar e salvar os diagramas para um formato XML personalizado (compactado por padrão, para economizar espaço), pode exportar diagramas para uma série de formatos, incluindo EPS, SVG, xfig, WMF e PNG, e pode imprimir diagramas (incluindo os que abrangem várias páginas).

## **Arquitetura de Software**

A arquitetura a ser utilizada para o desenvolvimento do software será o padrão de arquitetura MVC. Onde aplicação é dividida em 3 camadas (Model-View-Controller), de modo que haja uma total separação da lógica de negócios da camada de apresentação. Para [16] esta apresenta algumas nas quais são:

* Gerenciar múltiplos visualizadores usando o mesmo modelo e de modo fácil de manter, testar e atualizar sistemas múltiplos.
* Ser muito simples para incluir novos clientes apenas incluindo seus visualizadores e controles.
* Tornar a aplicação escalável.
* Ser possível de desenvolver em paralelo para o Model , View e Controller pois são independentes.
* Separar dados ou lógica de negócios (Model) da interface do usuário (View) e do fluxo da aplicação (Controller).
* Permitir que uma mesma lógica de negócios possa ser acessada e visualizada através de várias interfaces.
* A lógica de negócios (Model) não saber de quantas nem quais interfaces existem ou estão exibindo seus dados.

## **FrameWork**

As tecnologias a serem utilizadas para o desenvolvimento do software serão:

**Ruby On Rails:** Ruby on Rails é um [framework](https://pt.wikipedia.org/wiki/Framework) [livre](https://pt.wikipedia.org/wiki/Software_livre) que promete aumentar velocidade e facilidade no desenvolvimento de sites orientados a banco de dados (database-driven web sites), uma vez que é possível criar aplicações com base em estruturas pré-definidas. Frequentemente referenciado como Rails ou RoR, o Ruby on Rails é um projeto de código aberto escrito na linguagem de programação [Ruby](https://pt.wikipedia.org/wiki/Ruby_(linguagem_de_programa%C3%A7%C3%A3o)" \o "Ruby (linguagem de programação)). As aplicações criadas utilizando o framework Rails são desenvolvidas com base no padrão de arquitetura [MVC](https://pt.wikipedia.org/wiki/MVC) (Model-View-Controller).

## **Protótipo de telas**

### **Tela Inicial**

### **Tela de cadastro CaseMais**

# **Referencias**

[1] Sommerville, Ian. ***Engenharia de software***. Vol. 6. São Paulo: Addison Wesley, 2003.

[2] Rocha, Ana Regina Cavalcante da, José Carlos Maldonado, and Kival Chaves Weber. ***Qualidade de software***. São Paulo: Prentice Hall, 2001.

[3] INTERNET USAGE STATISTICS. Jun 2012. Disponível em: <http://www.internetworldstats.com/stats.htm >. Acesso em: 25 Nov. 2015.

[4] DE SANTANA, Otávio Gonçalves. Por que java ? Disponível em: < http://www.devmedia.com.br/por-que-java/20384> Acesso em: 24 Nov. 2015.

[8] ZAPPAROLI, Arthur. Git controle de versões do jeito certo. Out 2009. Disponível em: <http://pt.slideshare.net/arthurgeek/git-controle-de-verses-do-jeito-certo>. Acesso em: 24 Nov. 2015.

[9] TOFFOLO, Túlio. Controle de Versão com GIT. Disponível em: < http://www.decom.ufop.br/toffolo/site\_media/uploads/2011-1/git.pdf>. Acesso em: 24 Nov. 2015.

[10] PACIEVITCH, Yuri. MySQL. Jan 2011. Disponível em: <http://www.infoescola.com/informatica/mysql/>. Acesso em: 25 Nov. 2015.

[11] SANTOS, Adriano .MySQL: Quem é você? Disponível em: <http://www.devmedia.com.br/mysql-quem-e-voce/1752>. Acesso em: 25 Nov. 2015.

[12] GitHub – Features. Disponível em: <https://github.com/features> . Acesso em: 25 Nov. 2015.

[13] GitHub, você está por dentro das vantagens? Fev 2013. Disponível em: <http://www.ilegra.com/unilegra/2013/02/07/github-voce-esta-por-dentro-das-vantagens-deste-servico/>. Acesso em: 25 Nov. 2015.

[14] WILLIAM, Jon Mc Cann. DIA. Nov 2013. Disponível em: <https://wiki.gnome.org/action/show/Apps/Dia?action=show&redirect=Dia>. Acesso em: 25 Nov. 2015.

[15] VIEIRA, Marcio Junior. Modelagem UML com Software Livre. 2004. Disponível em:< http://webcache.googleusercontent.com/search?>. Acesso em: 25 Nov. 2015.

[16] DE ALMEIDA, Rodrigo Rebouças. Model-View-Controller (MVC). Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm>. Acesso em: 25 Nov. 2015.