

UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DESENVOLVIMENTO DE SOFTWARE ORIENTADO A OBJETOS

Mestrandos:

Gabriel Passatuto Gallo

Layon Martins Fonseca

Marcos Tulio Soto De La Vega

Paula Maia de Souza

Professor: Dr. Delano Medeiros Beder

PROJETO

DESENVOLVIMENTO DE UM SISTEMA PARA COMPRA DE PASSAGENS
AÉREAS

SÃO CARLOS / SP

2016

INTRODUÇÃO

Este projeto foi desenvolvido por alunos de pós graduação (stricto sensu) da Universidade Federal de São Carlos, sob a orientação do professor Dr. Delano Medeiros Beder.

A intenção do projeto é desenvolver um sistema web para compra de passagens aéreas, utilizando os conceitos estudados na disciplina de desenvolvimento de software orientado a objetos.

A equipe do projeto apelidou o sistema como “VarinGrails”.

1. Etapas de Desenvolvimento do Projeto

Para desenvolvimento deste projeto foram definidas as seguintes etapas:

- Planejamento
- Definição da Equipe do Projeto
- Levantamento de Requisitos
- Modelagem do Sistema
- Desenvolvimento do Sistema
- Entrega (Apresentação)

2. Planejamento

A primeira reunião da equipe ocorreu em setembro, quando se iniciou o planejamento do projeto. Foram definidas as etapas de desenvolvimento, conforme apresentado no Capítulo 1 e foram definidas também as funções de cada membro da equipe, conforme descrito no Capítulo 3.

No decorrer do mês de setembro foi elaborado o cronograma de atividades, conforme apresentado na Tabela 1 e definidas as ferramentas a serem utilizadas.

Tabela 1 - Cronograma de Atividades

	Setembro	Outubro	Novembro	Dezembro
Planejamento	X	X		
Definição da Equipe do Projeto	X			
Levantamento de Requisitos		X	X	
Modelagem		X	X	
Desenvolvimento		X	X	X
Entrega (Apresentação)				X

2.1. Ferramentas

2.1.1. Grails

Grails é um *framework* para construção de aplicações para web através da linguagem de programação Groovy (uma linguagem dinâmica para a plataforma Java).

Ele utiliza tecnologias consideradas maduras do mundo Java, como os frameworks Hibernate e Spring, através de uma interface que busca ser simples e consistente. O framework isola o desenvolvedor dos detalhes complexos da persistência de dados e incorpora o padrão de desenvolvimento MVC de maneira natural. Ele também fornece templates web para fácil implementação da interface com o usuário e suporte para programação em Ajax.

O Grails foi escolhido para o desenvolvimento desse sistema pois é uma ferramenta que permite agilidade no desenvolvimento.

2.1.2. Astah

Astah é uma ferramenta de modelagem e criação de diagramas UML. Foi projetada para ser leve e intuitiva, fácil o suficiente para um iniciante, mas robusta o suficiente para projetar sistemas completos (ASTAH, 2015).

A linguagem UML “é destinada a visualizar, especificar, construir e documentar os artefatos de um sistema complexo de software” (BOOCH; RUMBAUGH; JACOBSON, 2005).

Neste trabalho foram desenvolvidos os diagramas de Casos de Uso, de Atividades e de Classes, para compreender melhor as especificações e funcionalidades do sistema.

A ferramenta Astah foi escolhida para o desenvolvimento dos diagramas UML do sistema por ser gratuita e por sua compatibilidade entre sistemas operacionais.

3. Definição da Equipe do Projeto

As funções da equipe do projeto foram designadas de acordo a familiaridade de cada membro com as etapas de desenvolvimento de projeto de *software*. Porém foi definido pela equipe que todos os membros devem estar envolvidos em todas as fases (etapas) de desenvolvimento.

- Gabriel Passatuto Gallo ----- Designer e Desenvolvedor
- Layon Martins Fonseca ----- Gerente de Projeto e Desenvolvedor
- Marcos Tulio Soto De La Vega ----- Analista de Banco de Dados e Modelador
- Paula Maia de Souza ----- Analista de Requisitos e Planejamento

4. Levantamento de Requisitos

Para desenvolvimento de um projeto é necessário que seja feito um levantamento de requisitos funcionais e não funcionais do sistema a ser desenvolvido. Partindo desse princípio, foi elaborado o Documento de Requisitos do Sistema, apresentado no Anexo I.

5. Modelagem do Sistema

Para a modelagem do sistema foram utilizados os diagramas UML, mais especificamente, os Diagramas de Casos de Uso, Diagrama de Atividades e Diagrama Classes.

5.1. Diagramas de Casos de Uso

O diagrama de casos de uso procura possibilitar a compreensão do comportamento

externo do sistema por meio de uma linguagem simples. Sua função é representar as interações entre os atores que irão interagir com o sistema e o sistema.

No sistema VarinGrails é possível identificar 2 atores que utilizarão o site, sendo eles o Cliente e o Administrador.

A Figura 1 apresenta o diagrama de Casos de Uso que tem o Cliente como ator.

Já a Figura 2 representa o diagrama de Casos de Uso que tem como ator o Administrador do sistema.

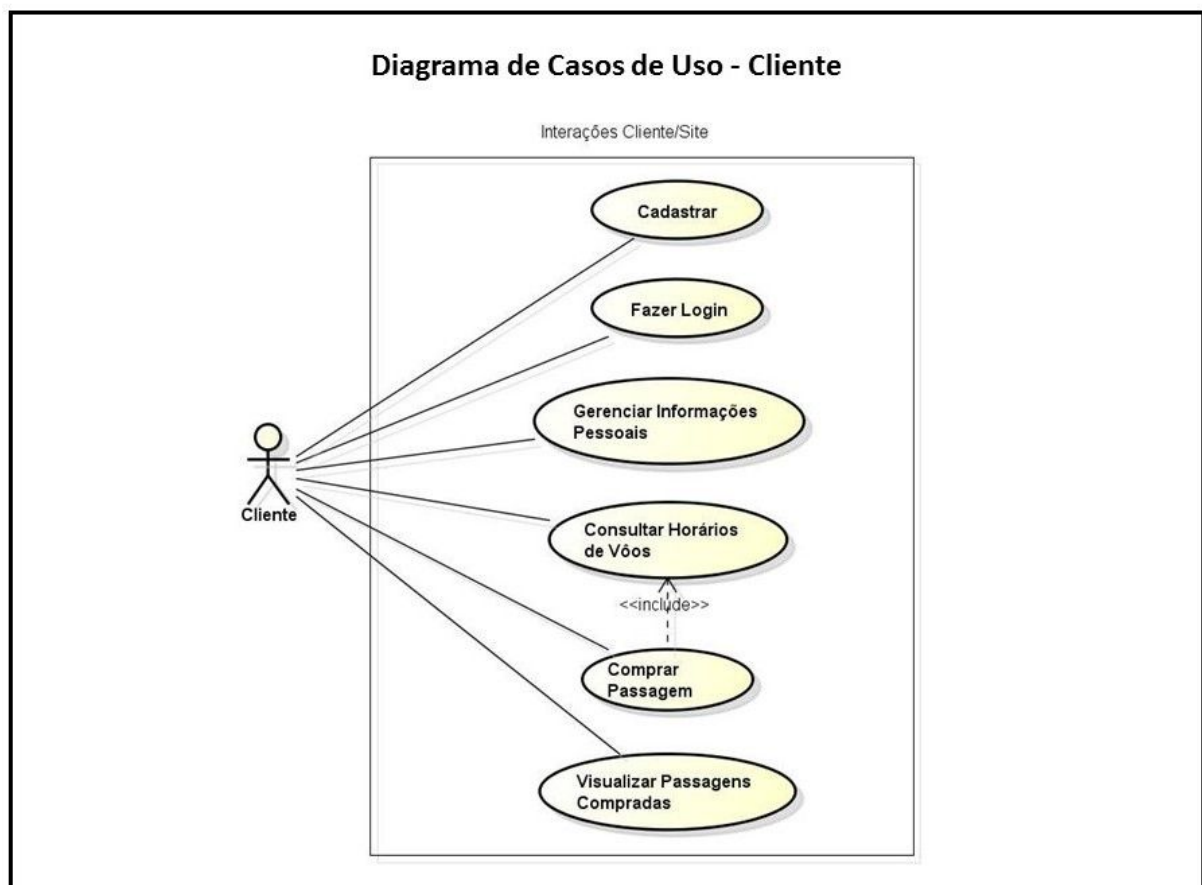


Figura 1 - Diagrama de Casos de Uso - Cliente

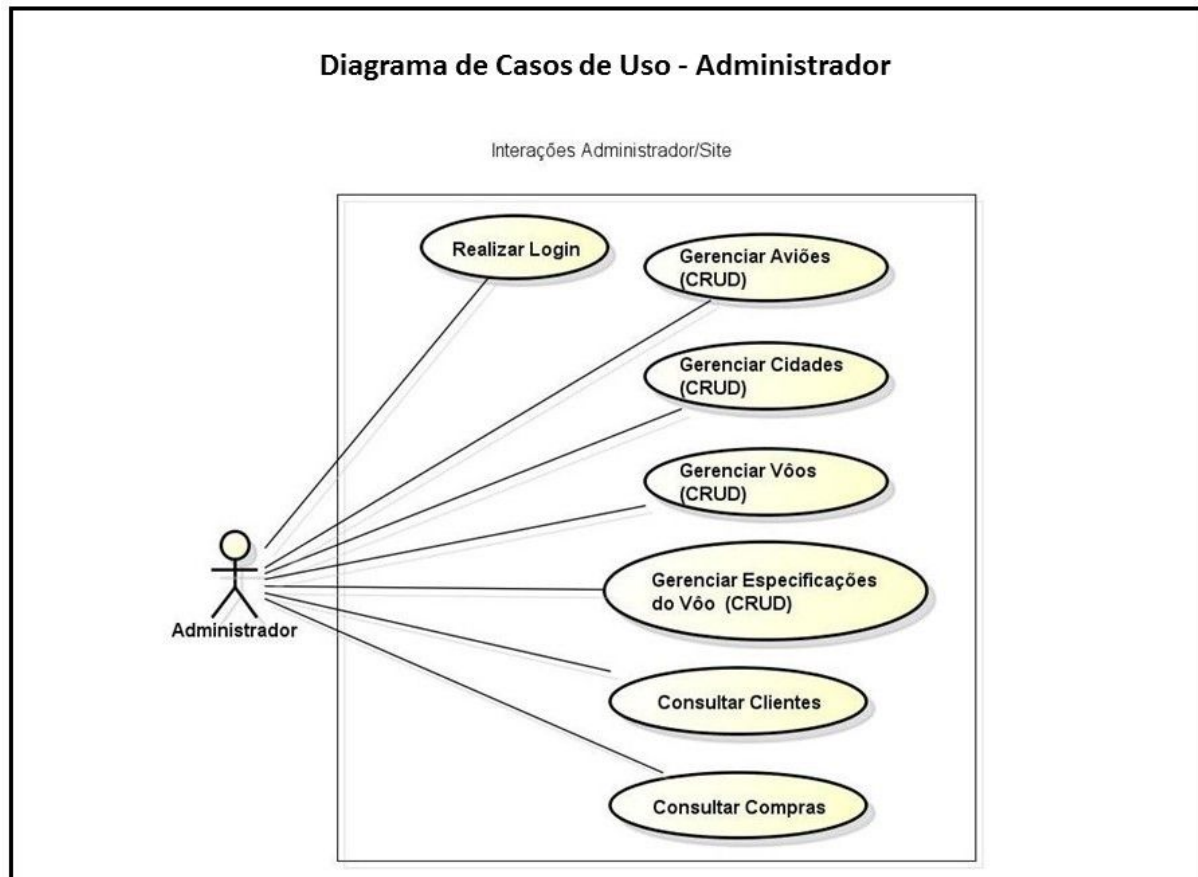


Figura 2 - Diagrama de Casos de Uso - Administrador

5.2. Diagrama de Atividades

O Diagrama de Atividades permite modelar o comportamento do sistema, denotando os caminhos lógicos que um processo pode seguir.

Foi desenvolvido o Diagrama de Atividades de Comprar Passagem, apresentado na Figura 3.

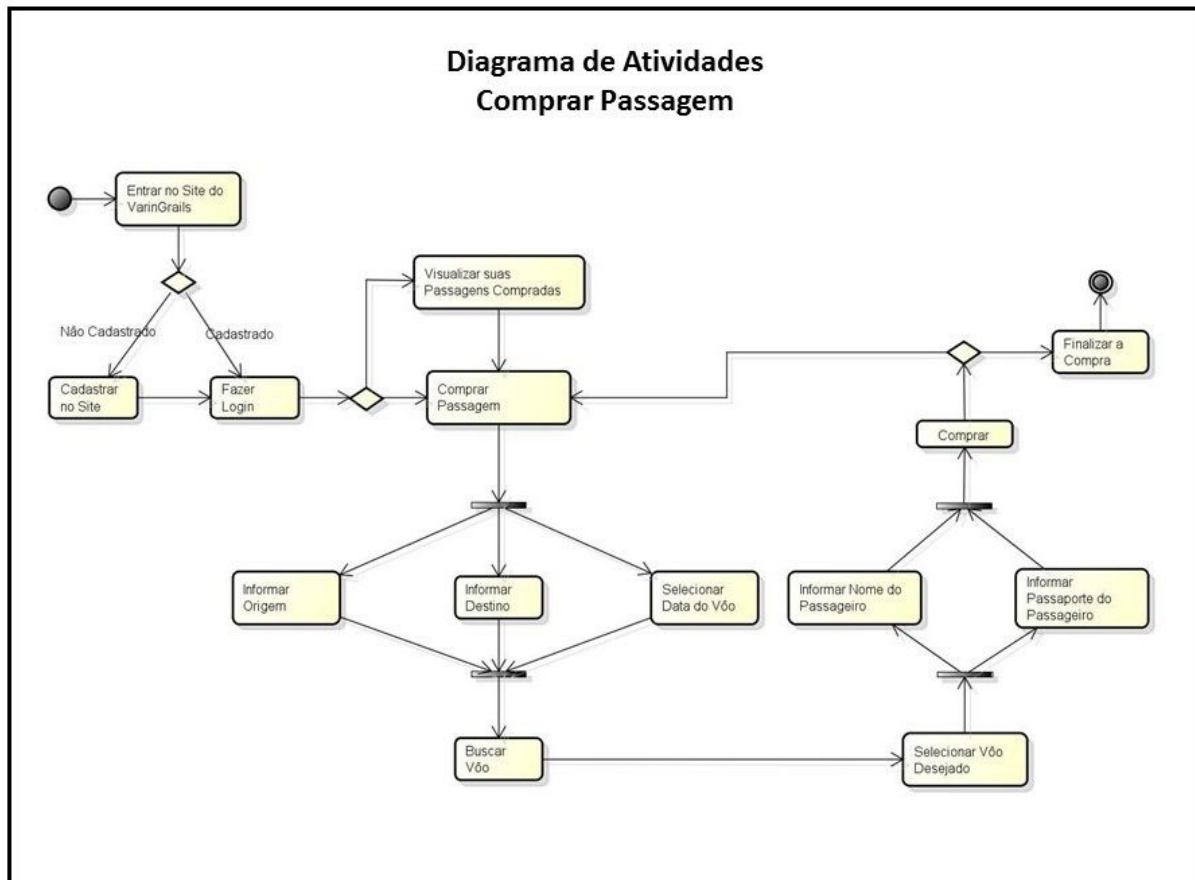


Figura 3 - Diagrama de Atividades - Comprar Passagem

5.3. Diagrama de Classes

O diagrama de classe tem a função de modelar os elementos de um programa orientado a objetos em tempo de desenvolvimento, é um dos diagramas mais utilizados da UML e serve de apoio para a maioria dos outros diagramas.

É composto por:

- Classes –Cada classe com seus atributos e métodos
- Associações –Relacionamento entre as classes

A Figura 4 apresenta o Diagrama de Classe do sistema.

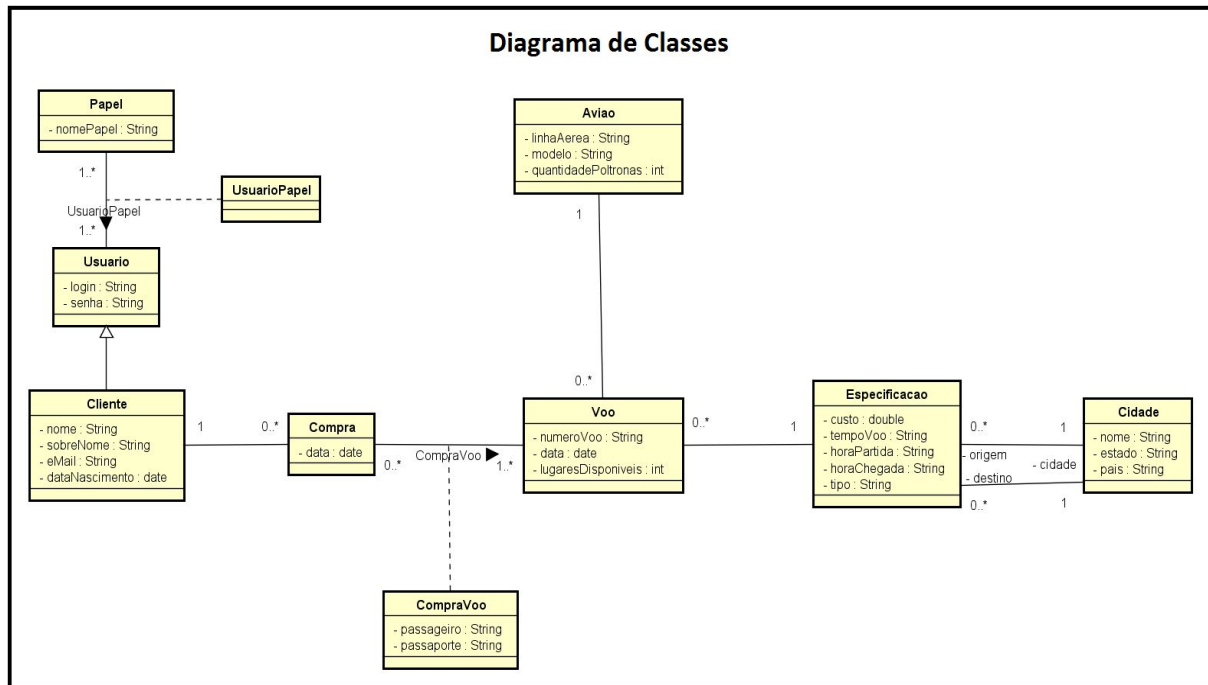


Figura 4 - Diagrama de Classes

6. Desenvolvimento do Sistema

O sistema denominado VarinGrails foi desenvolvido com o Grails e está disponível no GitHub pelo link: <<https://github.com/layonmartins/SistemaVendaPassagemAerea>>.

ANEXO I

DOCUMENTO DE REQUISITOS DO SISTEMA *VARINGRAILS*

1. Introdução

1.1. Propósito

O presente documento apresenta e descreve os requisitos necessários para o software VarinGrails. O documento destina-se ao analista de banco de dados, desenvolvedores e demais membros da equipe do projeto.

1.2. Escopo

O objetivo do VarinGrails é auxiliar na realização de compra de passagens aéreas através de um sistema web.

1.3. Visão Geral

O restante deste anexo está organizado da seguinte forma. Na Seção 1 são definidos os termos necessários ao entendimento dos requisitos do sistema. A Seção 2 contém uma descrição geral do VarinGrails. A Seção 3 identifica os requisitos funcionais e não funcionais do VarinGrails.

1.4. Definições

- **Administrador do sistema:** Funcionário da empresa de aviação responsável pela administração do site da empresa.
- **Cliente:** Cliente da empresa aérea, que compra uma passagem, podendo ser para ele mesmo ou para outra pessoa (passageiro);
- **Passageiro:** Aquele a quem pertence uma passagem e irá viajar em um voo;
- **Voo:** Uma viagem aérea de uma origem determinada para um destino determinado, realizado em um dia e hora específicos;
- **Origem:** Aeroporto de onde parte o voo;
- **Destino:** Aeroporto onde chega o voo;
- **Aeronave (ou avião):** Veículo utilizado na realização do voo;

2. Descrição Geral

2.1. Perspectiva do Produto

O VarinGrails deve ser desenvolvido para a plataforma WEB.

2.2. Funções do Produto

O software deve permitir que um cliente adquira uma ou mais passagens para viajar de avião de um local para outro. O sistema deve cuidar da armazenagem das transações e medidas de segurança, além de impedir a venda de passagens caso o voo estiver lotado.

2.3. Características do Usuário

- O **cliente** interage com o VarinGrails a partir de um navegador Web.
- O **administrador** (mantenedor) é a pessoa responsável por manter o sistema, realizando diferentes atividades, tais como: incluir novos voos, origens, destinos, cidades, aviões, etc.

3. Requisitos

3.1. Requisitos Funcionais

A seguir são apresentados os Requisitos Funcionais que o VarinGrails deverá atender.

F1 - Permitir cadastro de novos clientes

O sistema deve permitir que o usuário crie uma conta no VarinGrails. Para isso o usuário deve entrar no site do VarinGrails, ir em “Cadastrar” e informar os seguintes dados: nome (obrigatório), sobrenome (obrigatório), email (obrigatório), data de nascimento (obrigatório), login (obrigatório) e uma senha de acesso (obrigatório).

F2 - Permitir login de clientes

Clientes (usuários já cadastrados no site) poderão realizar login no VarinGrails. Para isso, eles devem entrar no site VarinGrails e, na área de login, informar seu Login e sua Senha.

F3 - Permitir uma consulta detalhada dos vôos disponíveis

A opção de fazer consulta deve estar disponível para clientes, e terá as seguintes definições para busca:

- O cliente poderá fazer uma consulta dos vôos disponíveis de uma cidade origem para uma cidade destino. Para isso o cliente deverá informar uma cidade (origem) e uma cidade (destino);
- O cliente deverá selecionar a data da viagem.

F4 - Realizar compra de passagens aéreas

Primeiramente o sistema deve permitir que o cliente realize a busca do vôo (F3). Após feita a busca e escolhido o vôo desejado, o cliente poderá realizar a compra da passagem.

Na compra deverá ser informado o nome completo e o número do passaporte do passageiro a quem se destina a passagem. O cliente poderá consultar outros vôos e comprar mais passagens. A compra só deverá ser finalizada quando o cliente “Finalizar a Compra”.

F5 - Visualizar passagens compradas

O sistema deve permitir que o cliente, devidamente logado, consiga visualizar as passagens que já comprou.

F6 - Gerenciar Aviões

O sistema deve permitir que seu administrador realize o gerenciamento dos Aviões (CRUD - Create, Read, Update e Delete), ou seja, o administrador poderá Cadastrar, Consultar, Editar e Deletar aviões.

F7 - Gerenciar Cidades

O sistema deve permitir que seu administrador realize o gerenciamento das Cidades (CRUD - Create, Read, Update e Delete), ou seja, o administrador poderá Cadastrar, Consultar, Editar e Deletar cidades .

F8 - Gerenciar Especificações de Vôos

O sistema deve permitir que seu administrador realize o gerenciamento das Especificações de Vôos (CRUD - Create, Read, Update e Delete), ou seja, o administrador poderá Cadastrar, Consultar, Editar e Deletar as especificações de vôos.

F9 - Gerenciar Vôos

O sistema deve permitir que seu administrador realize o gerenciamento dos Vôos (CRUD - Create, Read, Update e Delete), ou seja, o administrador poderá Cadastrar, Consultar, Editar e Deletar vôos.

F10 - Consultar Clientes

O sistema deve permitir que seu administrador realize a consulta para verificar os clientes cadastrados.

F11 - Consultar Compras

O sistema deve permitir que seu administrador realize a consulta para verificar as compras realizadas.

3.2. Requisitos Não Funcionais

A seguir são apresentados os Requisitos Não Funcionais que o VarinGrails deverá atender.

N1 - O sistema deve ficar disponível durante 24 horas por dia, todos os dias da semana.

N2 - O sistema deve prover segurança máxima. Não deve ser possível que clientes tenham acesso a compras de outros clientes. A única forma de acesso à conta deve ser por meio do nome de usuário (login) e senha corretos.

N3 - O cliente deverá conseguir realizar a compra de uma passagem com um tempo aproximado de 2 minutos e realizar em média de 15 cliques.

N4 - O sistema não pode permitir que o administrador realize, edite, nem exclua compras.

N5 - O sistema não pode permitir que o administrador cadastre, edite, nem exclua clientes.

N6 - O sistema deve ser executado na plataforma WEB.

REFERÊNCIAS

ASTAH. **UML and Modeling Tools**. Disponível em: <<http://astah.net/>>. Acessado em 29 de Abril. 2015.

BOOCH, G; RUMBAUGH, J e JACOBSON, I. **UML – Guia do Usuário**. 2ª edição. Rio de Janeiro, Campus, 2005.

FOWLER, M; SCOTT, K. **UML Essencial**. 2ª edição. Porto Alegre, Bookman, 2000.

LARMAN, Craig. **Utilizando UML e Padrões**. 1ª edição. Porto Alegre, Bookman, 2000.

MARTINS, José Carlos Cordeiro; Colaboração de RAMIREZ, Fabrício. **Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML**. 5ª edição. Rio de Janeiro, Brasport, 2010.

ROCHER, G.; LEDBROOK, P.; PALMER, M.; BROWN, J.; DALEY, L.; BECKWITH, B.; HOTARI, L. **The Grails Framework - Reference Documentation**. Versão: 3.1.10. Disponível em: <<http://docs.grails.org/3.1.10/>>. Acessado em novembro de 2016.