

FACULDADE DE TECNOLOGIA SENAI DE
DESENVOLVIMENTO GERENCIAL
GRADUAÇÃO TECNOLÓGICA

JONATHAN CARVALHO DE DEUS
HELLISON TEODORO DE OLIVEIRA
WALLACE GOMES

Sistema de Acompanhamento e Gerenciamento de Estágio

Goiânia
2014

FACULDADE DE TECNOLOGIA SENAI DE
DESENVOLVIMENTO GERENCIAL

GRADUAÇÃO TECNOLÓGICA

**AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE
CONCLUSÃO DE CURSO EM FORMATO ELETRÔNICO**

Na qualidade de titulares dos direitos de autores, **AUTORIZAMOS** a Graduação Tecnológica da Faculdade de Tecnologia SENAI de Desenvolvimento Gerencial – FATESG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da FATESG, entendendo-se os termos “reproduzir” e “publicar” conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela FATESG, a partir desta data.

Título: Sistema de Acompanhamento e Gerenciamento de Estágio

Autores: Jonathan Carvalho de Deus, Hellison Teodoro de Oliveira e Wallace Gomes

Goiânia, 20 de Novembro de 2014.

Jonathan Carvalho de Deus – Autor

Hellison Teodoro de Oliveira – Autor

Wallace Gomes – Autor

Edjalma Queiroz da Silva – Orientador

JONATHAN CARVALHO DE DEUS
HELLISON TEODORO DE OLIVEIRA
WALLACE GOMES

Sistema de Acompanhamento e Gerenciamento de Estágio

Trabalho de Conclusão de Curso apresentado a Graduação Tecnológica da Faculdade de Tecnologia SENAI de Desenvolvimento Gerencial, como requisito parcial para obtenção do Certificado de Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Edjalma Queiroz da Silva

Goiânia
2014

JONATHAN CARVALHO DE DEUS
HELLISON TEODORO DE OLIVEIRA
WALLACE GOMES

Sistema de Acompanhamento e Gerenciamento de Estágio

Trabalho de Conclusão apresentado à Coordenação do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Graduação Tecnológica da Faculdade de Tecnologia SENAI de Desenvolvimento Gerencial como requisito parcial para obtenção do título de Graduado em Tecnologia em Análise e Desenvolvimento de Sistemas, aprovada em 20 de Novembro de 2014, pela Banca Examinadora constituída pelos professores:

Prof. Edjalma Queiroz da Silva
Graduação Tecnológica - FATESG
Presidente da Banca

Prof. <Nome do membro da banca>
<Unidade acadêmica> – <Sigla da universidade>

Prof. <Nome do membro da banca>
<Unidade acadêmica> – <Sigla da universidade>

Dedico este trabalho primeiramente a Deus.

Aos meus pais, «NomePai» e «Mae», por terem sempre me apoiado em meus projetos.

À minha noiva, «NomeDaEsposaNoiva», pelo companheirismo, compreensão e carinho.

Ao meu irmão, «NomeIrmão», que sempre me ensinou a ver os problemas de maneira simples.

À família de meu amigo «NomeAMigo», que esteve presente em todos os momentos.

Agradecimentos

Algumas pessoas foram essenciais para a realização desse trabalho. Colaborando direta ou indiretamente, muita gente teve um papel fundamental para que os objetivos fossem concretizados. Com o trabalho pronto, chegou a hora de agradecer.

Ao meu orientador, «Nome do Orientador», pela oportunidade, pelos ensinamentos fornecidos para que a pesquisa pudesse ser realizada e por acreditar e me mostrar que a pesquisa científica é um desafio extremamente necessário para se obter bons resultados.

Ao professor Edjalma Queiroz da Silva, pelo empenho, paciência e pela dedicação com que ajudou no desenvolvimento do artigo publicado.

A Faculdade SENAI Universitário, pela excelente estrutura, suporte e ambiente oferecidos aos alunos, e a todos os professores, principalmente aos que eu tive a oportunidade de conhecer e aprender tantas coisas.

Aos amigos que contribuíram diretamente para a realização dessa pesquisa e foram muito importantes durante o andamento do trabalho.

A minha mãe, meu pai, meu irmão, minha cunhada e minha sobrinha que nos felicitam com sua presença. E aos demais familiares, em especial aos meus avós, que mesmo distantes, eu sei que torceram e acreditaram na realização desse trabalho.

A todos, o meu “muito obrigado”!

Dedicação Produz Dedicação; Preguiça produz Preguiça.

João Crisóstomo,
*Foi um teólogo e escritor cristão, arcebispo de Constantinopla no fim do
século IV e início do V.*

Resumo

Deus, Jonathan Carvalho; Oliveira, Hellison Teodoro ; Gomes, Wallace **Sistema de Acompanhamento e Gerenciamento de Estágio** . Goiânia, 2014. 31p. Trabalho de Conclusão de Curso. Graduação Tecnológica, Faculdade de Tecnologia SENAI de Desenvolvimento Gerencial.

O Sistema SAGE será um sistema web com a finalidade de gestão e controle de estágio em uma instituição de ensino. Contará com a parte de cadastro de alunos e empresas e todo o gerenciamento do estágio dentro da instituição. O sistema terá como principal função a gestão do estágio, no qual possibilita o cadastro de empresas, alunos e também todo o acompanhamento do estágio. Armazenando todas as informações em um banco de dados para que a instituição possa realizar consultas sobre os estágios dos alunos. O sistema abrange toda a parte de gestão do estágio, empresa e aluno dentro da instituição.

Palavras-chave

Sistemas de Estágio, Controle de Estágio, estagiário,

Abstract

Deus, Jonathan Carvalho; Oliveira, Hellison Teodoro ; Gomes, Wallace
TÍTULO DO TCC . Goiânia, 2014. 31p. Trabalho de Conclusão de Curso.
Graduação Tecnológica, Faculdade de Tecnologia SENAI de Desenvolvimento
Gerencial.

Recommendation systems work as a counselor, behaving in such a way to guide people in the discovery of products of interest. There are various techniques and approaches in the literature that enable generating recommendations. This is interesting because it emphasizes the diversity of options; on the other hand, it can cause doubt to the system designer about which is the best technique to use. Each of these approaches has particularities and depends on the context to be applied. Thus, the decision to choose among techniques become complex to be done manually. This article proposes an evolutionary approach for combining results of recommendation techniques (Invenire) in order to automate the choice of techniques and get fewer errors in recommendations. To evaluate the proposal, experiments were performed with a dataset from MovieLens and some of Collaborative Filtering techniques. The results show that the combining methodology proposed in this paper performs better than any one of collaborative filtering technique separately in the context addressed. The improvement varies from 3,6% to 118,99% depending on the technique and the experiment executed.

Keywords

Recommender Systems, Collaborative Filtering, combining results, similarity and Invenire

Sumário

Lista de Figuras	11
Lista de Tabelas	12
1 Introdução	14
1.1 Finalidade	14
2 Escopo do Projeto	15
2.1 Situação Atual	15
2.1.1 Motivação	15
2.1.2 Problema	15
2.1.3 Hipótese	15
2.1.4 Justificativa	15
2.2 Objetivos	15
2.2.1 Objetivos Gerais	15
2.2.2 Objetivos Específicos	16
2.3 Cronograma	16
3 Fundamentação Teórica	17
3.1 Engenharia de Software	17
3.2 Gestão de Projetos	17
3.3 Análise e Definição de Requisitos	17
3.3.1 Principais Técnicas de Análise e Levantamento de Requisitos	18
3.3.2 Tipos de Requisitos	18
Requisitos Funcionais	18
Requisitos Não-Funcionais	18
3.3.3 Protótipos	19
3.3.4 Validação de Requisitos	19
3.3.5 Linguagem Unificada de Modelagem - UML	19
3.3.6 Problemas	20
Problemas com Stakeholders	20
Problemas com Engenheiros/Desenvolvedores	20
3.3.7 Rastreabilidade de Requisitos	21
3.3.8 O Uso de Ferramentas para a Análise e Gerenciamento de Requisitos	21
3.3.9 Qualidade na fase de Análise de Requisitos	21
3.4 Arquitetura de Software	22

4	Estudo de Caso	23
4.1	Regras de Negócio e Requisitos Funcionais	23
4.2	Prototipos	23
4.3	Diagramas da UML	25
4.3.1	Diagrama de Fluxo	25
4.3.2	Diagrama de Caso de Uso	25
4.3.3	Diagrama de Atividade	26
4.3.4	Diagrama de Classe	26
4.4	Matriz de Rastreabilidade	27
4.5	Arquitetura do Software	27
4.6	Frameworks e Ferramentas Adotadas	27
4.6.1	Astah Community	27
4.6.2	Eclipse	27
4.6.3	Tomcat	28
4.6.4	Java	28
4.6.5	Java Server Faces	28
4.6.6	Primefaces	28
4.6.7	Spring	29
4.6.8	Hibernate	29
4.6.9	Maven	29
4.6.10	Junit	29
5	Considerações Finais	30
5.1	Conclusão	30
5.1.1	Contribuição	30
5.2	Trabalhos Futuros	30
	Referências Bibliográficas	31

Lista de Figuras

4.1	Protótipo - Tela de empresa.	23
4.2	Protótipo - Tela de estágio.	24
4.3	Protótipo - Tela de orientação de estágio.	24
4.4	Protótipo - Tela de colaborador da empresa.	25
4.5	Diagrama de Caso de Uso.	26
4.6	Diagrama de Classe UML - SAGE.	27

Lista de Tabelas

Lista de Códigos

Introdução

Desenvolvemos uma aplicação Web para facilitar e inovar o gerenciamento das atividades de acompanhamento e gerenciamento de estágio supervisionado para instituições de ensino, possibilitando a automação da autogestão do programa de estágio. Pelo qual esse processo de gestão de estágio é feita manualmente pela coordenação do estágio, no qual perde-se muito tempo no controle e avaliação devido ao processo manual.

Este processo inicia através da empresa divulgando a vaga de estágio, em seguida com os alunos devidamente selecionados para participarem do processo e posteriormente a matrícula no estágio.

Como vimos todo esse processo e feito de forma manual.

1.1 Finalidade

Este documento apresenta a modelagem do sistema SAGE — Sistema de Acompanhamento e Gerenciamento de Estágio.

Apresentaremos neste trabalho, sobre as fases de desenvolvimento de software, que vão o levantamento de requisitos até o produto completo. O público alvo deste documento inclui as pessoas envolvidas no desenvolvimento do sistema (analistas e desenvolvedores) todos os envolvidos no projeto.

Escopo do Projeto

Este documento de modelagem de Sistema prevê uma visão completa dos modelos do sistema SAGE, sendo produzido e utilizado pela equipe de desenvolvimento para documentar os requisitos, modelos, tecnologias e arquitetura do sistema.

2.1 Situação Atual

Atualmente seus processos tem sido feito mediante planilha impressa e de forma manual, relatórios de presença do estagiário e documentos de acompanhamento, entre outros tipos de controle.

2.1.1 Motivação

2.1.2 Problema

2.1.3 Hipótese

2.1.4 Justificativa

2.2 Objetivos

2.2.1 Objetivos Gerais

O projeto tem como objetivo facilitar e inovar o gerenciamento das atividades referentes ao controle de estágio de instituições, retirar o processo manual feito pela coordenação de estágio e pelas empresas. Ganho de tempo, facilidade de acesso a informação, elaboração do programa de estágio de forma mais rápida, possibilitando a automação do processo de estágio.

2.2.2 Objetivos Específicos

Desenvolver um sistema para a melhoria do processo de gestão do estágio, inovar os processos produtivos, fazer controle de todo o processo de cadastro das empresas, divulgação das vagas de estágio, seleção dos alunos aptos a participarem do processo seletivo, matrícula dos alunos no programa de estágio e também possibilitar todo o processo de avaliação do desempenho do aluno no estágio, gerar relatórios. Além das funcionalidades descritas acima, o sistema terá também a construção de um banco de dados com todas as informações referente ao estágio que posteriormente poderá ser consultado.

2.3 Cronograma

Fundamentação Teórica

3.1 Engenharia de Software

A engenharia de software é uma tecnologia em camadas (1 - Ferramentas; 2 - Métodos; 3 - Processo; 4 - Foco na qualidade). Qualquer abordagem de engenharia (inclusive engenharia de software) deve estar fundamentada em um comprometimento organizacional com a qualidade. A gestão de qualidade total Seis Sigma e filosofias similares promovem uma cultura de aperfeiçoamento contínuo de processos, e é esta cultura que, no final das contas, leva ao desenvolvimento de abordagens cada vez mais efetivas na engenharia de software. A pedra fundamental que sustenta a engenharia de software é o foco na qualidade [[Pressman 2011](#)].

3.2 Gestão de Projetos

Abrange uma série de ferramentas e técnicas utilizadas por pessoas para descrever, organizar e monitorar o andamento das atividades do projeto. Os gerentes de projeto são os responsáveis pela administração dos processos envolvidos e pela aplicação das ferramentas e técnicas necessárias ao cumprimento das atividades do projeto. Todo projeto é composto por processos, por mais irrelevante que seja a abordagem empregada.

A Gestão de Projetos consiste na aplicação de conhecimento, competências, ferramentas e técnicas às atividades do projeto, com vista ao cumprimento dos requisitos em pauta. É responsabilidade do gerente de projeto assegurar que tais técnicas sejam utilizadas e seguidas [[Heldman 2006](#)].

3.3 Análise e Definição de Requisitos

A análise e definição dos requisitos de software são atividades no desenvolvimento do sistema onde serão levantados e definidos os elementos que irão compor o software e as restrições associadas a eles. A análise deve estabelecer o

relacionamento entre estes objetivos e restrições e a especificação precisa do software. Nestas atividades, o contato com o cliente é efetivo e constante, uma vez que é necessário extrair as regras do negócio.

É o processo de observação e levantamento dos elementos do domínio no qual o sistema será introduzido. Deve-se identificar as pessoas, as atividades, informações do domínio para que se possa decidir o que deverá ser informatizado.

3.3.1 Principais Técnicas de Análise e Levantamento de Requisitos

Existem algumas técnicas de levantamento de requisitos que são utilizadas, reuniões, questionários, entrevistas individuais, sessões, etc. Os analistas podem empregar uma ou várias técnicas para eliciar os requisitos dos clientes, isto envolve situações tais como organizar entrevistas ou grupos focais (workshops) e a criação de lista de requisitos. Técnicas mais modernas incluem prototipação, e casos de uso, onde o analista irá aplicar uma combinação de métodos para estabelecer os requisitos exatos de seus stakeholders, tal que um sistema que atenda as necessidades do negócio seja produzido.

3.3.2 Tipos de Requisitos

Os requisitos de um sistema são descrições do sistema, ou seja, uma declaração abstrata de alto nível de um serviço que o sistema deve fornecer ou uma restrição do sistema [Sommerville 2008].

Requisitos Funcionais

Os requisitos funcionais de um sistema estão totalmente ligados ao software, o propósito desses requisitos é descrever uma função do sistema, ou seja, descrever detalhadamente seu conjunto de entradas, seu comportamento e as suas saídas [Sommerville 2008]

Requisitos Não-Funcionais

Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema. Esses requisitos surgem devido às necessidades do usuário, tais como: segurança, confiabilidade, usabilidade, desempenho, entre outras necessidades que forem necessárias e que não estão ligadas a funções do sistema [Sommerville 2008].

3.3.3 Protótipos

A prototipação é uma parte importantíssima no processo de desenvolvimento de software, pois é a fase em que o desenvolvedor começa a modelar os elementos do sistema, assim ficando até mais fácil na hora de começar a realmente desenvolver o sistema. Também pode ser útil para o usuário final, já que se trata de elementos do sistema com quais os usuários terão contato [Pressman 2010]. Tornando mais compreensível, para o usuário final, a transformação dos requisitos coletados no sistema propriamente dito. É uma forma de o usuário validar os requisitos coletados.

Além de validar os requisitos já coletados com o cliente, o protótipo pode auxiliar na identificação de novos requisitos que no momento da coleta não foram observados.

É também uma forma de reduzir os custos, já que os erros e omissões durante a fase de levantamento dos requisitos, podem ser muito onerosos quando necessitam de correções em outras fases.

3.3.4 Validação de Requisitos

A validação de requisitos é uma parte muito importante no processo de desenvolvimento de software, pois é nessa parte que o engenheiro de software, clientes, usuários e outros interessados examinam a especificação procurando por erros de conteúdo ou de interpretação, inconsistências, informações omissas, entre outras coisas [Pressman 2010].

3.3.5 Linguagem Unificada de Modelagem - UML

A UML(Unified Modeling Language - linguagem de modelagem unificada) é "uma linguagem-padrão para descrever/documentar projeto de software. A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de software-intensivo". Em outras palavras, assim como os arquitetos criam plantas e projetos para ser usados por uma empresa de construção, os arquitetos de software criam diagramas UML para ajudar os desenvolvedores de software a construir o software [Pressman 2011].

Na UML o significado de “classe” pode ser observada em diferentes perspectivas, como segue:

- Classe Conceitual – coisa ou conceito do mundo real
- Classe de Software – classe que representa uma perspectiva de especificação ou implementação de um elemento de software, independente do processo ou método.
- Classe de Implementação – classe implementada em uma linguagem orientada a objetos específica, como a Java.

3.3.6 Problemas

A atividade de levantamento de requisitos nem sempre traz um cenário perfeito ou amigável para o analista de sistemas. Muitas das vezes as pessoas que serão envolvidas no processo não estão dispostas a colaborar, isso por diversos motivos, tais como o medo de perder o emprego para uma máquina ou medo de que o software leve a tona uma realidade que não é a que eles (o usuário especialista) queriam estar mostrando da empresa. Seja qual for o motivo da não colaboração de nossos stakeholders, ainda assim nos deparamos com alguns problemas críticos, tais como: nem mesmo os “especialistas” daquele departamento ou setor da empresa em que vai ser construído o software sabe ao certo como é o fluxo correto de trabalho, como é a rotina do seu departamento.

Problemas com Stakeholders

No processo de levantamento de requisitos com os stakeholders podem aparecer alguns problemas que impeçam ou diminuam a clareza e obtenção dos requisitos, os principais são:

- Usuários não sabem o que eles querem.
- Usuários que não querem concluir a escrita do conjunto de requisitos.
- Comunicação com o usuário é lenta.
- Os usuários freqüentemente não participam nas revisões ou são incapazes de fazer isto.
- Os usuários são tecnicamente poucos sofisticados.
- Os usuários não entendem do desenvolvimento de processo.

Isto deve levar a situações onde os requisitos do usuário continuam mudando mesmo quando o desenvolvimento do sistema ou produto já se iniciou.

Problemas com Engenheiros/Desenvolvedores

No processo de análise de requisitos podem aparecer alguns problemas como:

- Pessoal técnico e usuários finais têm vocabulários diferentes. Conseqüentemente, eles podem acreditar que estão em perfeito acordo até que o produto final seja entregue.
- Engenheiros e desenvolvedores tentam ajustar os requisitos para um sistema existente ou modelo, em vez de desenvolver um sistema específico que atenda as necessidades do cliente.
- A análise é freqüentemente conduzida por engenheiros ou programadores, ao invés de pessoal com habilidade e domínio do conhecimento para compreender as necessidades dos clientes.

3.3.7 Rastreabilidade de Requisitos

O rastreamento de requisitos é utilizado para prover uma gestão desses requisitos, e para cada requisito é atribuído um identificador. Uma vez identificados os requisitos, tabelas de rastreamento são desenvolvidas. Cada uma dessas tabelas relaciona um ou mais requisitos do sistema ou de seu ambiente. Em muitos casos essas tabelas são mantidas como parte do banco de dados de requisitos, de modo que elas possam ser encontradas com mais facilidade e serem feitas alterações caso seja necessário [Pressman 2010].

3.3.8 O Uso de Ferramentas para a Análise e Gerenciamento de Requisitos

As empresas possuem a necessidade de se medir a sua capacidade de gerenciar o processo de Análise de requisitos. Para isto, existem no mercado ferramentas que utilizam modelo CMMI (Capability Maturity Model Integration), que é um modelo gerencial que organiza as melhores práticas existentes, embora os padrões e as práticas que são aplicáveis não sejam completamente definidos.

Dentre essas ferramentas de gerenciamento de requisitos de software destacam-se:

- Borland CaliberRM

É uma ferramenta de software corporativa para o gerenciamento de requisitos que facilita a colaboração, a análise de impacto e a comunicação, permitindo que as equipes de software avancem o projeto com maior precisão e previsibilidade.

A arquitetura aberta do Borland CaliberRM permite que o analista de sistemas conecte os requisitos do software a uma variedade de artefatos utilizados no ciclo de vida, possibilitando assim uma rastreabilidade de requisitos de ponta a ponta.

3.3.9 Qualidade na fase de Análise de Requisitos

Dentro da Engenharia de Software existe uma busca constante no que tange a qualidade no desenvolvimento de software, algumas empresas utilizam o modelo RUP, que é um modelo gerencial que organiza as melhores práticas existentes e que mede a qualidade por meio da maturidade da capacidade dos processos de software.

O que se nota é que se deve focalizar e investir no processo de melhoria contínua visando à qualidade dos requisitos ou da forma de gerência dos requisitos que irão compor o software.

O RUP é um framework genérico e complexo, pois visa atender todos os tipos de projetos de desenvolvimento de software. Toda disciplina do RUP deve ser

analisada e customizada de acordo com as necessidades específicas do projeto antes de sua implantação. Um framework de maneira simples, é um conjunto de objetos extensível para funções relacionadas, que visa fornecer uma implementação para as funções básicas e invariantes e inclui um mecanismo para permitir que o desenvolvedor se conecte às diversas funções ou as estenda.

3.4 Arquitetura de Software

A Arquitetura de Software não esta relacionada a uma fase do desenvolvimento do sistema em si. Essa é a fase em que é feita a representação que permite o engenheiro de software analisar a efetividade do projeto em satisfazer a seus requisitos declarados, considerar alternativas para realizar mudanças arquiteturais e reduzir riscos no processo de construção do software [[Pressman 2010](#)].

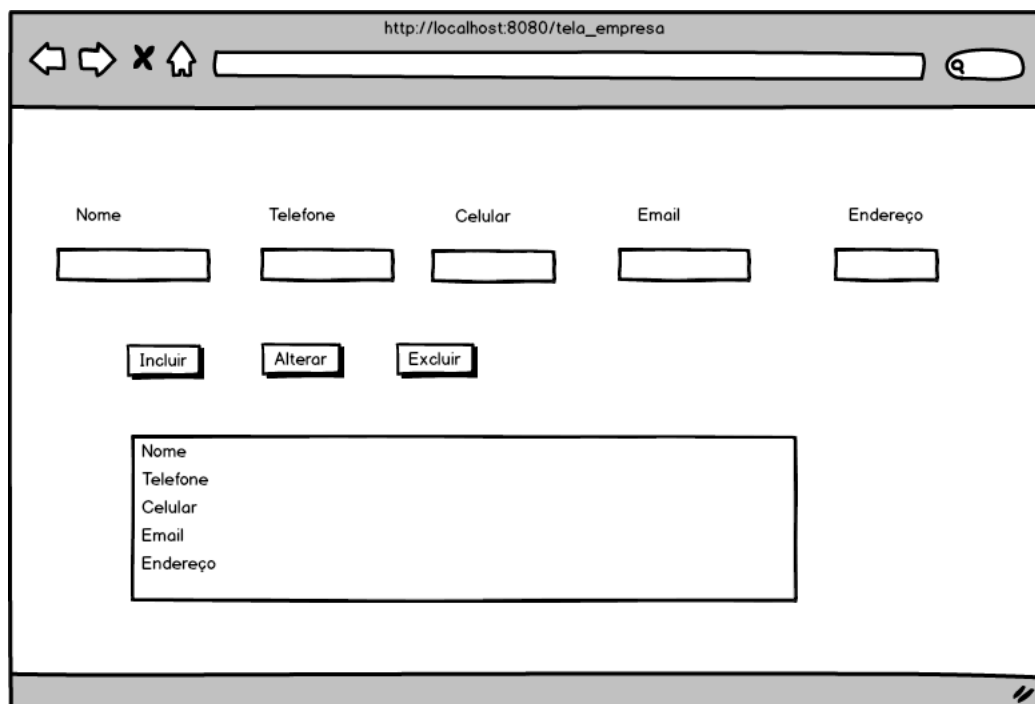
Estudo de Caso

4.1 Regras de Negócio e Requisitos Funcionais

4.2 Protótipos

A prototipação tem como principal objetivo modelar os elementos do sistema como descrito na Seção 3.3.3. Nesta Seção demonstraremos como aplicamos a técnica de prototipação.

A Figura 4.1 demonstra a tela do sistema onde os usuários poderam cadastrar, consultar, alterar e excluir *Empresas*. O procedimento onde cadastramos, consultamos, alteramos e excluimos algum objeto no sistema é conhecido como C.R.U.D.



O protótipo de tela de empresa é exibido em um navegador web com o endereço `http://localhost:8080/tela_empresa`. A interface contém cinco campos de entrada para o cadastro de uma empresa: Nome, Telefone, Celular, Email e Endereço. Abaixo desses campos, há três botões de ação: Incluir, Alterar e Excluir. Na parte inferior da tela, há uma caixa de texto que lista os campos: Nome, Telefone, Celular, Email e Endereço.

Figura 4.1: Protótipo - Tela de empresa.

A figura 4.1 demonstra a tela do sistema onde os usuários poderam realizar um cadastro de um estágio.

Protótipo da tela de estágio (tela_estagio). A interface é dividida em duas seções principais: 'Aluno' e 'Empresa'. A seção 'Aluno' contém campos para Nome, Telefone, CPF e Email. A seção 'Empresa' contém campos para Nome, Telefone, Celular, Email e Endereço. Abaixo da seção 'Empresa', há um campo para 'Tipo Estágio' com uma seta para baixo, e dois campos para 'Data Inicial' e 'Data Final'. Um botão 'Incluir' está localizado na parte inferior esquerda da seção de formulário. O navegador mostra a URL 'http://localhost:8080/tela_estagio'.

Figura 4.2: Protótipo - Tela de estágio.

A figura 4.3 demonstra a tela do sistema onde os usuários podem realizar um cadastro de orientação referente a um estágio.

Protótipo da tela de orientação de estágio (tela_orientacao). A interface inclui um campo de busca para 'Nome' com o placeholder 'Nome Aluno'. Abaixo, há uma tabela com cabeçalhos 'Professor Orientador', 'Empresa' e 'Status'. O campo 'Professor Orientador' possui um botão 'Arquivo' e o texto '< Selecciona um arquivo'. Um botão 'Cadastrar' está na parte inferior. O navegador mostra a URL 'http://localhost:8080/tela_orientacao'.

Figura 4.3: Protótipo - Tela de orientação de estágio.

A figura 4.4 demonstra a tela do sistema onde os usuários poderam realizar um C.R.U.D.

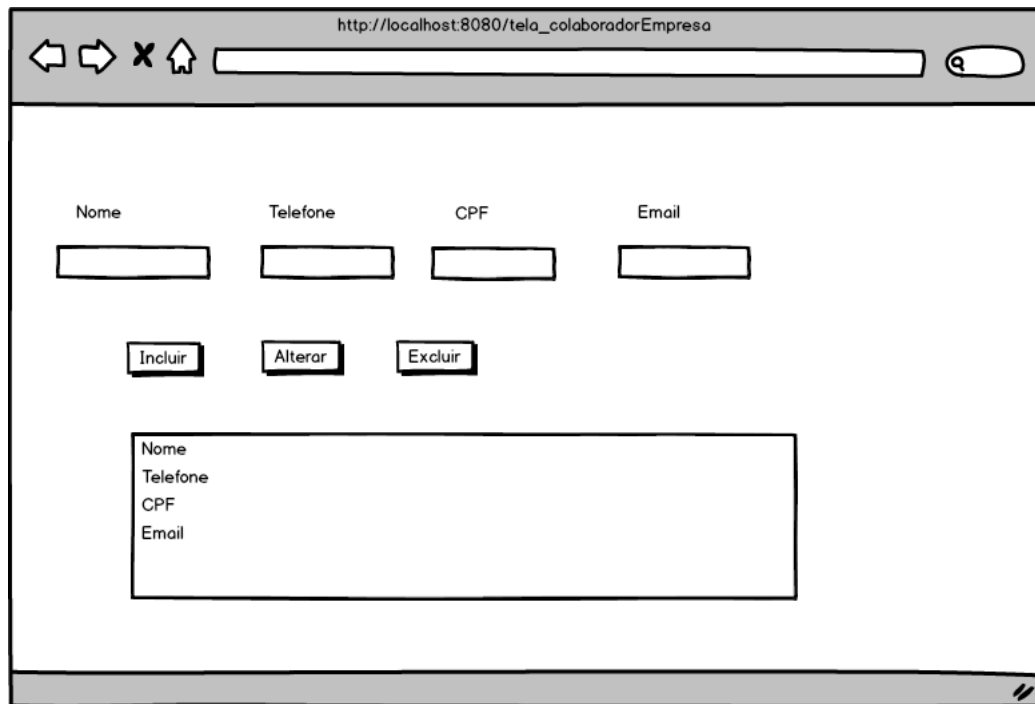


Figura 4.4: Protótipo - Tela de colaborador da empresa.

4.3 Diagramas da UML

A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de software-intensivo como descrito na Seção 3.3.5.

4.3.1 Diagrama de Caso de Uso

Este documento serve para documentar o que o sistema faz do ponto de vista do usuário, descrevendo as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema.

A figura 4.5 demonstra o Diagrama de Caso de Uso do sistema *SAGE*.

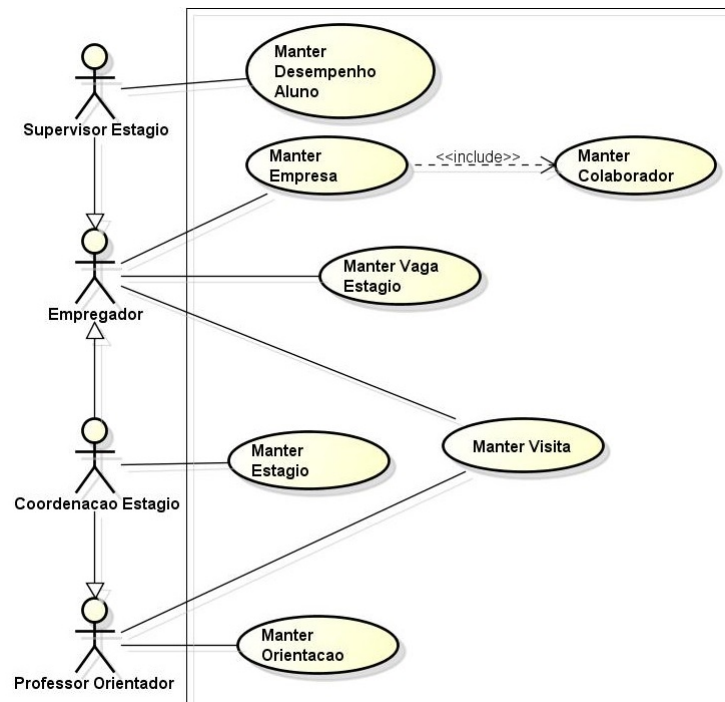


Figura 4.5: *Diagrama de Caso de Uso.*

4.3.2 Diagrama de Atividade

4.3.3 Diagrama de Classe

A figura 4.6 demonstra o diagrama de classes UML do sistema *SAGE*.

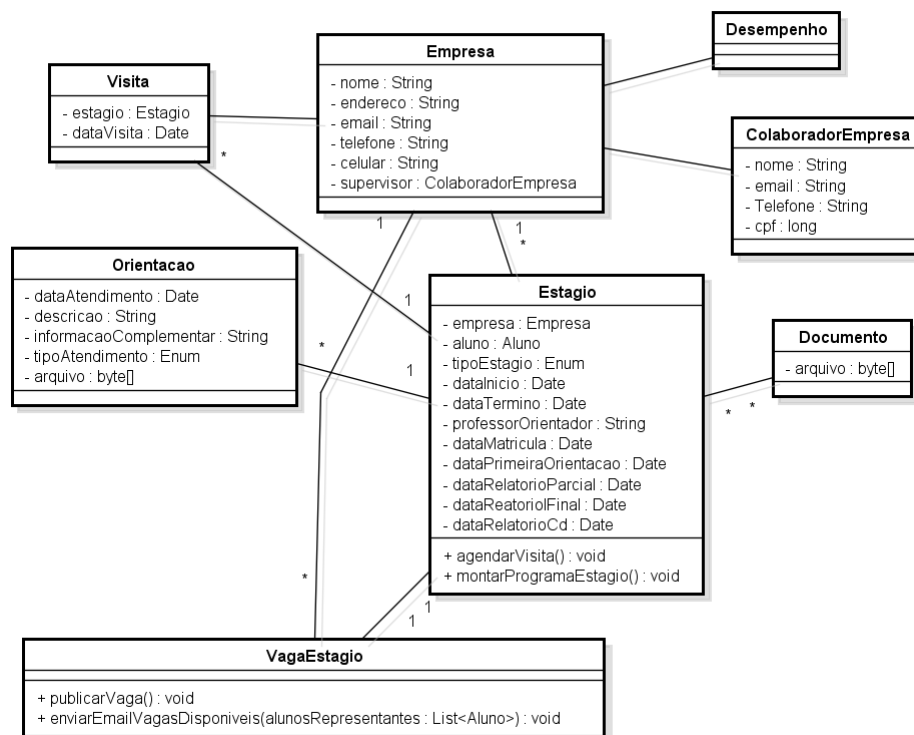


Figura 4.6: Diagrama de Classe UML - SAGE.

4.4 Matriz de Rastreabilidade

4.5 Arquitetura do Software

4.6 Frameworks e Ferramentas Adotadas

4.6.1 Astah Community

Astah Community é um software para modelagem UML. É desenvolvido na plataforma Java, o que garante sua portabilidade para qualquer plataforma que possui uma máquina virtual Java. Foi utilizado para a modelagem UML por ser uma ferramenta gratuita, e atender totalmente as necessidades do projeto ao desenvolver os diagramas UML.

4.6.2 Eclipse

Eclipse é uma IDE gratuita open source feita em Java e atualmente a mais utilizada no mundo. Um dos principais fatores que influenciaram o seu uso hoje na versão (Kleper, 4.3.1) uso foi a sua enorme quantidade de plug-ins, que dão um grande suporte aos desenvolvedores durante o desenvolvimento.

4.6.3 Tomcat

Tomcat é um servidor WEB Java, distribuído como software livre bastante consolidado na comunidade, mantido pela Apache Software Foundation (Uma organização sem fins lucrativos, que inclusive contribui ativamente com diversos projetos para o Java). Esse servidor foi escolhido por ser um servidor gratuito e de fácil instalação e por oferecer suporte a todos os recursos utilizados no projeto.

4.6.4 Java

A plataforma Java foi utilizada por hoje ser uma plataforma bastante consolidada no mercado, um de seus grandes benefícios está em sua máquina virtual Java (JVM), um grande conjunto de API e também a linguagem Java em si. Por ser uma máquina virtual a JVM tem importante característica nas aplicações feitas em Java que é a portabilidade, outro recurso bastante interessante que a JVM nos proporciona é a capacidade de executarmos outras linguagens já que a ela é também multi-linguagem. Devido a sua popularização, o Java conta com um grande apoio não só da comunidade como também de diversas empresas, o que resultou em uma enorme variedade de ferramentas e bibliotecas o que a torna uma poderosa ferramenta de desenvolvimento.

4.6.5 Java Server Faces

O Java Server Faces (JSF) é uma especificação Java para arquitetura MVC, sua arquitetura baseada em componentes o torna uma poderosa ferramenta para construção de Ricas Interfaces com o Usuário (RIA), seu funcionamento orientado a eventos faz com o que os desenvolvedores se dediquem menos em conhecer o funcionamento do protocolo HTTP e dediquem-se mais na lógica da aplicação, o que torna o seu funcionamento bem parecido com aplicações desktop. Sua arquitetura o permite que desenvolvedores construam novos componentes baseado em componentes já existentes, promovendo assim o reuso e a criação de componentes mais complexo.

4.6.6 Primefaces

Primefaces é uma poderosa biblioteca Java para a criação de Interfaces Gráficas com o Usuário (GUI) para o JSF com uma série de componentes prontos. Este foi utilizado devido ser hoje a mais recomendada no mercado, a frente de seus concorrentes como RichFaces e IceFaces, e por ter o seu uso bastante simplificado.

4.6.7 Spring

Spring é um framework Java open source. Este foi utilizado visando realizar a injeção de dependência e a inversão de controle, diminuindo o acoplamento entre classes e melhorando a qualidade do código desenvolvido no projeto.

4.6.8 Hibernate

Hibernate é um software livre, open source para o mapeamento objeto-relacional (ORM). Hoje ele é o ORM mais utilizado no mercado em aplicações Java. Ele foi utilizado devido uma disparidade de paradigmas entre a programação orientada a objetos e o banco de dados relacional, além de tirar do desenvolvedor o trabalho de escreverem complexos scripts SQL, aumentando a produtividade da equipe. Um recurso interessante disponibilizado por esta ferramenta é a possibilidade de torna-lo independente de banco de dados, por realizar a intermediação entre a aplicação e o banco de dados ele pode facilmente se adaptar a diversos bancos.

4.6.9 Maven

Apache Maven ou simplesmente Maven como é conhecido, é um projeto desenvolvido pela Apache Software Foundation, essa ferramenta nos permite realizar a construção, compilação e teste de aplicações Java de maneira simples e automáticas. Essa ferramenta foi utilizada principalmente em nossos projetos por ela realizar o gerenciamento das APIs de forma automatizada, padronizada, facilitando e principalmente evitando problemas como conflitos de bibliotecas, incompatibilidade de versões e a fácil localização de novas bibliotecas.

4.6.10 Junit

Junit é um framework open source para a realização de testes unitários em Java. Ele facilita a criação de testes unitários como também a prática do TDD. Por ser um framework bastante maduro e consolidado no mercado utilizando esse framework por também se integrar ao Maven o que permite criar testes e automatiza-los, esses testes seriam feitos sempre que realizarmos um deploy no projeto evitando que falhas fossem disponibilizadas, mantendo a entrega de produtos mais confiável.

Considerações Finais

5.1 Conclusão

5.1.1 Contribuição

5.2 Trabalhos Futuros

Referências Bibliográficas

[Heldman 2006]HELDMAN, K. *Gerência de Projetos guia para o exame oficial do PMI*. Rio de Janeiro: Elsevier, 2006.

[Pressman 2010]PRESSMAN, R. S. *Engenharia de Software*. [S.l.]: AMGH, 2010.

[Pressman 2011]PRESSMAN, R. S. *Engenharia de Software - Uma Abordagem Profissional*. São Paulo: Bookman, 2011.

[Sommerville 2008]SOMMERVILLE, I. *Engenharia de software*. [S.l.]: Pearson Prentice Hall, 2008.