

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

RAFAEL HENRIQUE ZALESKI ADADA

**SISTEMA DE IDENTIFICAÇÃO DE INDIVÍDUOS COM CAPACIDADES
TÉCNICAS PARA A REALIZAÇÃO DE PROJETOS ATRAVÉS DO
HISTÓRICO ACADÊMICO**

CURITIBA

2017

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

RAFAEL HENRIQUE ZALESKI ADADA

**SISTEMA DE IDENTIFICAÇÃO DE INDIVÍDUOS COM CAPACIDADES
TÉCNICAS PARA A REALIZAÇÃO DE PROJETOS ATRAVÉS DO
HISTÓRICO ACADÊMICO**

Trabalho de Conclusão de Curso
apresentado ao curso de Engenharia de
Computação da Universidade Tecnológica
Federal do Paraná, como requisito parcial
para a obtenção do título de Engenheiro de
Computação.

Orientador:
Prof. Dr. Dario Eduardo Amaral Dergint
UTFPR

CURITIBA

2017

RESUMO

ZALESKI ADADA, Rafael H. Sistema de identificação de indivíduos com capacidades técnicas para a realização de projetos através do histórico acadêmico. 2017. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

Os processos de seleção de indivíduos para tarefas técnicas geralmente envolvem a subjetividade do selecionador e processos demorados de análise de perfis. Além disso, a escolha do perfil mais capacitado esbarra em questões como a habilidade de comunicar suas capacidades por parte do candidato. As instituições de ensino públicas frequentemente encontram-se envolvidas em projetos de pesquisa e esbarram nos mesmos problemas na hora de selecionar os indivíduos do seu corpo docente e discente mais capacitados para participar destes projetos. Mais ainda, esta dificuldade na identificação das habilidades de seus estudantes para os potenciais projetos torna parcerias Universidade-Empresa difíceis de se concretizarem. Trazendo o escopo para a UTFPR, Campus Curitiba, desenvolveu-se o protótipo de um sistema de identificação de indivíduos com potenciais capacidades técnicas para a formação de equipes de trabalho em projetos de escopo tecnológico. Tal identificação é feita através do histórico acadêmico dos discentes da instituição. O projeto foi realizado em seis etapas principais: estudar dados disponíveis dos estudantes, criação de dados de teste, estudo e elaboração de um algoritmo de normalização de desempenho, codificação do núcleo básico do software com telas de visualização, desenvolvimento de algoritmo para seleção de perfis e testes finais. A aplicação é constituída de dois grandes módulos: o módulo de processamento do algoritmo e o módulo de cadastro e visualização de dados. O processamento é realizado através de uma aplicação de busca assíncrona utilizando um algoritmo com matriz de pesos para a triagem de perfis, os quais, por sua vez, são acessados pelo módulo de visualização disponível em um servidor web. A utilização de uma camada de acesso através de um serviço web visa facilitar a centralização de informações para acesso por toda a instituição. Por meio da utilização deste *software*, a UTFPR será capaz de iniciar processos de seleção com o grupo de candidatos disponíveis mais indicados tecnicamente para uma tarefa, trazendo desenvolvimento tecnológico para um processo que, anteriormente, era realizado de forma mecânica: a análise de capacidades.

Palavras-chave: Recursos Humanos. Gestão de Projetos. Equipes de Trabalho. Software.

ABSTRACT

ZALESKI ADADA, Rafael H. Computer system to identify skilled individuals to work on projects using academic records. 2017. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

The available processes to select individuals for technical tasks are influenced by the subjectivity of the interviewer and by slowly procedures of analysis. Moreover, the choice of the most skilled profiles relies on abilities of candidates in communicating their skills. Public universities in Brazil are often involved in research projects and are faced with these obstacles to select the best-fit individuals to take on these projects. Even more, the difficulty to identify the skills of students for potential projects turns it hard to get close partnerships between companies and universities. Bringing the scope to the UTFPR, Curitiba campus, it was developed a prototype of a system to identify potential individuals to compose work teams for technological projects. The proposed project was developed in six main stages: the study of available data, the creation of test data for simulations, the creation of an algorithm to standardize the performance of students, the codification of the software basic core and its interface, the development of an algorithm to select the best-fit profiles and final tests. The application is built in two main modules: the module for algorithm processing and the module to register and view data. The processing is done through an asynchronous search application and an algorithm of a weighted matrix to filter profiles, which can be seen in the visualization module available in a web server. The using of an access layer through a web server intends to favor the data centralization, thus that it can be accessed for all the institution. By using this software, UTFPR will be able to initiate selection processes having the group of available candidates which best matches the requirements of a given task, bringing technological development to a process which used to be made in a mechanical way: the skill analysis.

Keywords: Human Resources. Project Management. Work teams. Software.

LISTA DE FIGURAS

Figura 1 Diagrama de Casos de Uso para o Gestor de Projeto	10
Figura 2 Diagrama de Casos de Uso para o Administrador do Sistema	11
Figura 3 Diagrama de classes do Sistema	12
Figura 4 Arquitetura FIFO de escalonamento de tarefas, a mesma utilizada no processamento de projetos	17
Figura 5 Menu exibido para um usuário com papel Administrador.....	18
Figura 6 Menu exibido para um usuário com papel Gestor de Projeto.....	19
Figura 7 Tela de Busca no Sistema.	19
Figura 8 Tela "Cadastrar Projeto"	21
Figura 9 Tela "Meus Projetos"	22
Figura 10 Tela de resultado do processamento para um projeto	23
Figura 11 Tela de visualização do perfil de um estudante	24
Figura 12 Tela para adicionar novo Gestor de Projeto, acessível apenas para administradores.....	25
Figura 13 Tela para associar capacidades (abstrações de disciplinas)	26

LISTA DE TABELAS

Tabela 1 - Cronograma planejado de execução do projeto.....	30
Tabela 2 - Cronograma real da execução do projeto	31

LISTA DE ABREVIATURAS E SIGLAS

UTFPR	Universidade Tecnológica Federal do Paraná
P&D	Pesquisa e Desenvolvimento
SQL	<i>Structured Query Language</i>
HTML	<i>Hypertext Markup Language</i>
CSS	<i>Cascading Style Sheet</i>
PHP	<i>Hypertext Processor</i>
UML	<i>Unified Modeling Language</i>
JSON	<i>JavaScript Object Notation</i>
CMS	<i>Content Management System</i>
AJAX	<i>Asynchronous JavaScript and XML</i>
XML	<i>Extensible Markup Language</i>
RBAC	<i>Role-based Access Control</i>
GPA	<i>Grade Point Average</i>
REST	<i>Representational State Transfer</i>
API	<i>Application Programming Interface</i>
FIFO	<i>First In, First Out</i>
FCFS	<i>First Come, First Served</i>
CRON	<i>Command Run On</i>
URL	<i>Uniform Resource Locator</i>

SUMÁRIO

1 INTRODUÇÃO	1
1.1 MOTIVAÇÃO	1
1.2 OBJETIVOS.....	2
1.2.1 Objetivo Geral.....	2
1.2.2 Objetivos específicos.....	2
1.3 METODOLOGIA	3
2 FUNDAMENTAÇÃO TÉCNICA.....	5
2.1 TECNOLOGIAS	5
2.1.1 Javascript.....	5
2.1.2 EasyAutocomplete.....	5
2.1.3 ICheck.....	6
2.1.4 Wordpress.....	6
2.1.5 JQuery	7
2.1.6 HTML e CSS.....	7
2.1.7 PHP	7
2.2 ARQUITETURA DO SISTEMA, PATENTES E PROPRIEDADES.....	8
2.2.1 Controle de Acesso.....	8
2.2.2 Diagramas UML	8
2.2.2.1 Diagramas de casos de uso.....	8
2.2.2.2 Diagrama de classes.....	11
2.3 REQUISITOS FUNCIONAIS DO SISTEMA.....	13
2.4 REQUISITOS NÃO FUNCIONAIS	13
2.5 PATENTES E PROPRIEDADES	14
2.6 CONSIDERAÇÕES DO CAPÍTULO	14
3 DESENVOLVIMENTO.....	16
3.1 CARACTERÍSTICAS GERAIS.....	16
3.1.1 Tarefa assíncrona de seleção de candidatos	16

3.1.2 Módulo de Visualização e telas do sistema	17
3.2 ALGORITMO DE CÁLCULO DO SCORE DE UM ESTUDANTE PARA UM PROJETO	26
3.3 CONSIDERAÇÕES DO CAPÍTULO	28
4 GESTÃO DO PROJETO	29
4.1 GESTÃO DE TAREFAS.....	29
4.2 GIT E VERSIONAMENTO DE SOFTWARE	29
4.3 CRONOGRAMA PLANEJADO	29
4.4 CRONOGRAMA EXECUTADO	30
4.5 CONSIDERAÇÕES DO CAPÍTULO	31
5 CONSIDERAÇÕES FINAIS	32
REFERÊNCIAS.....	34

1 INTRODUÇÃO

Este documento apresenta a especificação técnica e o desenvolvimento de um protótipo de sistema de suporte a decisão de gestores de projetos. A aplicação realiza a filtragem de potenciais candidatos para equipes de execução técnica utilizando o histórico acadêmico mantido pela instituição de ensino da qual eles integram o corpo docente. Para o escopo deste projeto a instituição escolhida foi a UTFPR, campus Curitiba.

1.1 MOTIVAÇÃO

A seleção de indivíduos para a realização de tarefas técnicas possui diversas metodologias, as quais geralmente envolvem algum tipo de contato entre selecionador e potencial candidato. O selecionador pode tentar identificar habilidades, mas estas sempre dependerão de algum tipo de capacidade de comunicá-las pelo entrevistado, sejam verbalmente ou de forma escrita. Além disso, tal comunicação depende do autoconhecimento do candidato e de alguma forma de atestar a informação.

Por mais desenvolvido que seja o método utilizado de seleção, o caráter subjetivo sempre estará presente, conforme exposto por (ROBBINS, DECENZO e VERHULST, 2015).

Seria ingênuo supor, porém, que todos os avaliadores interpretam e padronizam com imparcialidade os critérios segundo os quais seus avaliados serão avaliados. Isso é particularmente verdadeiro para trabalhos para os quais é difícil, se não impossível, estabelecer padrões de desempenho. Alguns desses trabalhos incluem, mas certamente não se limitam, aos cargos de pesquisador, professor, engenheiro e consultor. (ROBBINS, DECENZO e VERHULST, 2015).

Sendo assim, para tarefas de natureza técnica, onde as habilidades de comunicação e marketing pessoal do indivíduo podem não ter foco primordial, métodos usuais de seleção são lentos, custosos e até ineficazes.

No contexto acadêmico, a universidade não mantém um registro de potenciais habilidades de seu corpo docente e, menos ainda, do seu corpo docente, para auxiliar na seleção de equipes para tarefas de P&D, se limitando a lançar editais e esperar os candidatos aparecerem.

Desta maneira, identificou-se a necessidade de sistemas que reduzam a subjetividade do primeiro contato entre selecionador e entrevistado, fornecendo uma poderosa ferramenta de filtragem para o início de um processo de seleção baseando-se em aspectos puramente técnicos. Tal sistema auxiliaria o gestor de projetos na tarefa complexa de busca dos melhores candidatos com base em suas habilidades potenciais.

1.2 OBJETIVOS

Nesta seção serão apresentados o objetivo geral e os objetivos específicos do projeto.

1.2.1 Objetivo Geral

Desenvolver um sistema de identificação de indivíduos com potenciais capacidades técnicas para a formação de equipes de trabalho em projetos de escopo tecnológico utilizando-se de uma base de dados de histórico acadêmico da UTFPR, campus Curitiba.

1.2.2 Objetivos específicos

- Desenvolver uma abstração para a representação de projetos que possa ser inserida no sistema pelo gestor de projetos.
- Criar um método de caracterização das competências dos estudantes.
- Desenvolver interface que permita a associação de capacidades a disciplinas.
- Realizar a filtragem de potenciais candidatos por sua adequabilidade ao projeto e apresentar através de um serviço *web*.
- Desenvolver interface para a apresentação do perfil técnico de todos os estudantes.

1.3 METODOLOGIA

Para o desenvolvimento do sistema proposto, foi necessário estudar os dados disponíveis dos estudantes e criar uma base de dados em um servidor MySQL com entradas fictícias seguindo o modelo armazenado pela UTFPR. Tal modelo foi possível através da cessão dos diagramas de banco de dados do aplicativo UTFAPP (UTFAPP, 2017), e, a partir destes, foi possível originar o registro de cem estudantes fictícios de Engenharia de Computação, curso disponibilizado pela UTFPR, campus Curitiba. Embora os dados pessoais fossem fictícios, os dados de disciplinas, turmas, matrizes e outros não pessoais eram verdadeiros. Estes estudantes receberam um nome selecionado de uma lista de nomes brasileiros aleatórios, a qual foi disponibilizada pelo serviço web Fake Name Generator (CORBAN WORKS LLC, 2017). Além disso, estes estudantes receberam um coeficiente de rendimento, um período no curso, uma lista de disciplinas cursadas e seus respectivos desempenhos.

De posse dos dados fictícios e do esquema de Banco de Dados que a instituição possui, foi possível elaborar um método de comparação de desempenho para os estudantes, o qual utiliza um somatório de multiplicação de matrizes com pesos.

Em seguida, foi estruturado e codificado o núcleo do software, o qual fornece todas as funcionalidades do sistema para a interface gráfica, que foi construída com as tecnologias HTML, CSS e JQuery. Através da interface gráfica, a etapa de filtragem de perfis foi testada de forma mais rápida e eficaz, não sendo necessário trabalhar com grande quantidade de dados sendo apresentada através de um console do sistema.

Após o desenvolvimento da camada de visualização, foi implementado e testado o algoritmo de filtragem de candidatos, cuja proposta é o cerne do projeto. A linguagem de programação utilizada foi a PHP e o processamento do algoritmo foi codificado em uma fila de tarefas não bloqueante para não prejudicar o desempenho de execução.

Depois de finalizadas as etapas anteriores, foi possível criar telas de detalhamento técnico de indivíduos para consulta pelos gestores quando recebem os resultados de suas buscas.

Finalizada a implementação do algoritmo, este foi testado contra uma série de projetos fictícios e apresentou os resultados esperados conforme a especificação do *software*.

Seguiu-se a metodologia apresentada e foi elaborado o presente documento para a apresentação do projeto, o qual é dividido em cinco capítulos. O capítulo 1 trata da introdução e apresentação do projeto, objetivos pretendidos e a metodologia utilizada. O capítulo 2 apresenta a fundamentação técnica, com as tecnologias utilizadas, a arquitetura do sistema, os diagramas UML, os requisitos funcionais e não funcionais e a pesquisa de patentes. No capítulo 3 é demonstrado o desenvolvimento do *software*, suas telas e o funcionamento dos algoritmos. O capítulo 4 apresenta a metodologia de gestão utilizada, o cronograma planejado e o cronograma executado. No último capítulo são apresentados os resultados obtidos, as sugestões para projetos futuros e as considerações finais.

2 FUNDAMENTAÇÃO TÉCNICA

Nesta seção serão apresentados os fundamentos técnicos para o desenvolvimento do sistema.

2.1 TECNOLOGIAS

Nesta seção serão apresentadas as tecnologias utilizadas para a concretização do projeto.

2.1.1 Javascript

O JavaScript é uma linguagem de programação orientada a objetos usada amplamente no desenvolvimento de páginas web (FLANAGAN, 2006). Sua execução se dá no browser do usuário acessando a aplicação, o que pode ser utilizado para reduzir o processamento do lado do servidor. Com o passar dos anos, inúmeros frameworks foram criados para a linguagem, sendo que alguns permitem criar aplicações inteiras apenas utilizando JavaScript e HTML.

2.1.2 EasyAutocomplete

A biblioteca EasyAutocomplete (PAWEŁCZAK, 2017) foi desenvolvida para fornecer o recurso de auto completar em campos de entrada de texto baseando-se em uma lista fornecida no formato JSON. Esta funcionalidade permite que o usuário receba uma lista de sugestões antes de terminar de digitar o que estava procurando.

No escopo do projeto, a biblioteca é utilizada na busca geral, que permite buscar estudantes e disciplinas, e também na tela de adicionar projeto, para auto completar as capacidades.

2.1.3 ICheck

ICheck (SULTANOV, 2017) é uma biblioteca jQuery para melhorar a visualização de campos do tipo botão de rádio e campos do tipo caixa de seleção em uma página web, aumentando-os e padronizando a exibição em diferentes plataformas.

A biblioteca foi utilizada para dar mais visibilidade nas opções de relevância para uma capacidade na tela de adicionar projeto.

2.1.4 Wordpress

O Wordpress é um sistema gerenciador de conteúdo *open source* e gratuito que utiliza a linguagem de programação PHP e o sistema gerenciador de banco de dados MySQL. Em maio de 2017, cerca de 27.9% dos dez milhões de websites mais acessados no mundo utilizavam Wordpress (W3TECHS, 2017).

Sua arquitetura é baseada em um sistema de plugins e temas. Através de plugins os usuários podem estender as funcionalidades de um website de acordo com suas necessidades, inserindo código próprio em praticamente qualquer ponto de execução do sistema sem necessidade de modificar o código do *core* do Wordpress (WORDPRESS).

Os temas permitem modificar a aparência e funcionalidade sem alterar o código do *core* e o conteúdo armazenado no banco de dados. Existe um conjunto de padrões que devem ser seguidos para a elaboração de temas (WORDPRESS, 2017).

Ao longo de seus treze anos de existência, o Wordpress evoluiu de uma plataforma de blogs para um CMS completo e com baixa curva de aprendizado. Atualmente, com a evolução dos aplicativos em nuvem e para dispositivos móveis, o time de desenvolvimento entregou junto com a versão 4.7 o Wordpress REST API (WORDPRESS, 2017), com a intenção de tornar a ferramenta um framework de aplicação completo para qualquer linguagem de programação.

2.1.5 JQuery

JQuery é uma biblioteca JavaScript de código livre para simplificação de sintaxe, além de fornecer ferramentas poderosas para a criação de animações, manipular páginas web e executar rotinas comuns em JavaScript, como chamadas AJAX. Seu lema “*write less, do more*” define a missão da biblioteca e, os mais de dez anos de mercado e milhões de utilizadores, a consolidam como uma ferramenta sólida para utilização em aplicações escaláveis.

2.1.6 HTML e CSS

A HTML é uma linguagem de marcação e a base de qualquer página web (MOZILLA DEVELOPER NETWORK, 2016). Através de sua sintaxe hierárquica similar ao XML ela dá instruções ao browser de como o conteúdo deve ser exibido e do que o conteúdo se trata, como imagens, vídeos, endereços, campos de entrada de dados, etc.

O CSS é uma linguagem utilizada para especificar como os documentos serão apresentados aos usuários em relação ao aspecto visual (MOZILLA DEVELOPER NETWORK, 2016), como cores, espaçamento, posicionamento, etc.

Enquanto o HTML é utilizado para estruturar o conteúdo, o CSS é utilizado para trabalhar com sua formatação gráfica.

2.1.7 PHP

Segundo (PHP GROUP, 2016), “PHP, que significa ‘PHP: Hypertext Preprocessor’, é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a web e pode ser mesclada dentro do código HTML.”.

O PHP é processado do lado do servidor, o que permite gerar páginas com conteúdo dinâmico, coletar dados do browser e trabalhar com *cookies*, arquivos e bancos de dados. O interpretador PHP é compatível com a maioria dos sistemas operacionais modernos e suportado pela maioria dos servidores web, como o Apache, o IIS e o NodeJS (PHP GROUP, 2016).

2.2 ARQUITETURA DO SISTEMA, PATENTES E PROPRIEDADES

Nesta seção será apresentado o padrão de arquitetura escolhido para o desenvolvimento do sistema e a consulta de patentes e propriedades.

2.2.1 Controle de Acesso

A política de controle de acesso é baseada em papéis, também conhecida como RBAC e foi herdada do CMS utilizado, ou seja, o Wordpress. Uma política de controle de acesso baseada em papéis define um conjunto de papéis no sistema e um conjunto de autorizações para cada um deles. Esta abordagem permite desacoplar os usuários das permissões (MAZIERO, 2013).

As autorizações e os papéis podem ser editados, adicionados e removidos no painel administrativo do CMS. No projeto foram configurados dois papéis, Administrador e Gestor de Projeto. O administrador tem todas as permissões e o gestor possui somente aquelas relevantes para cadastrar projetos e realizar consultas.

2.2.2 Diagramas UML

Nesta seção serão apresentados os diagramas UML do sistema utilizados na especificação do software.

2.2.2.1 Diagramas de casos de uso

Nos diagramas de casos de uso são apresentados os dois atores do sistema: administrador e gestor de projetos; bem como as funcionalidades e tarefas que estes podem executar.

O gestor de projetos pode buscar uma disciplina (Figura 1), sendo o resultado desta apresentado junto de uma lista das turmas para a disciplina. Ao acessar os dados de uma turma, o gestor tem acesso a lista dos estudantes da turma e seus desempenhos, podendo também acessar o perfil individual destes alunos e visualizá-los por completo.

Um estudante pode ser encontrado através de uma busca pelo seu registro acadêmico ou nome (Figura 1), sendo o resultado da busca o perfil do estudante para visualização, ou seja, não há uma lista de resultados e ao escolher um estudante entre os sugeridos pela função auto completar da busca, o gestor é direcionado ao perfil deste estudante.

A funcionalidade de cadastrar projeto (Figura 1) permite ao gestor criar requisições dentro do sistema para encontrar perfis adequados a uma proposta de trabalho. Uma vez que o projeto seja processado e tenha recebido o estado “pronto”, o gestor pode consultar os resultados (Figura 1), os quais são apresentados através de uma lista ordenada de perfis, que podem ser acessados individualmente, similar a lista de estudantes para uma turma.

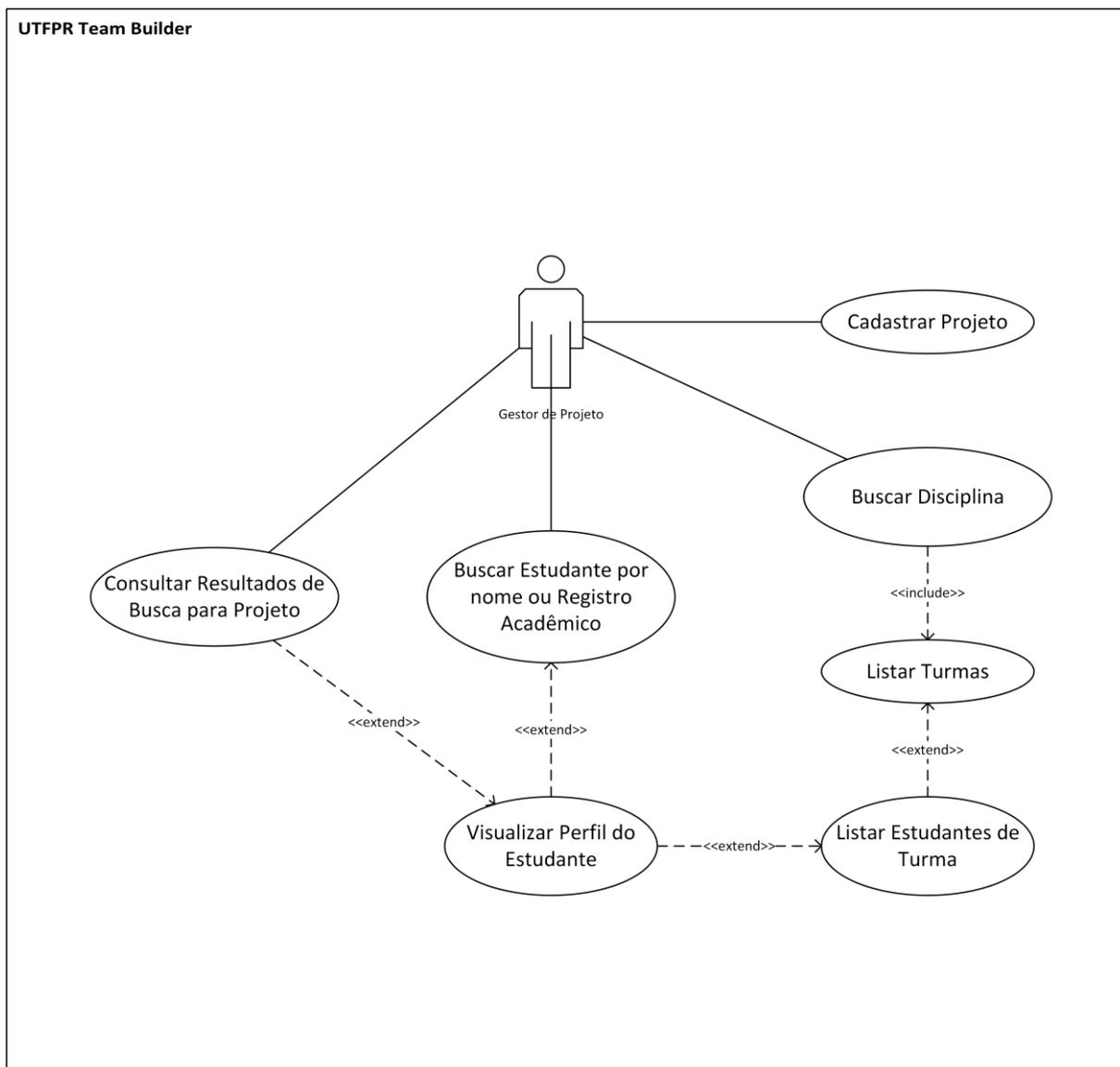


Figura 1 Diagrama de Casos de Uso para o Gestor de Projeto
Fonte: autoria própria

O administrador do sistema pode realizar todas as tarefas acessíveis ao gestor de projetos (Figura 2). Como funcionalidades adicionais, é possível acessar o painel administrativo do CMS utilizado como base do sistema (Figura 2), associar capacidades a disciplinas, bem como cadastrar novos usuários, sejam gestores de projeto ou administradores.

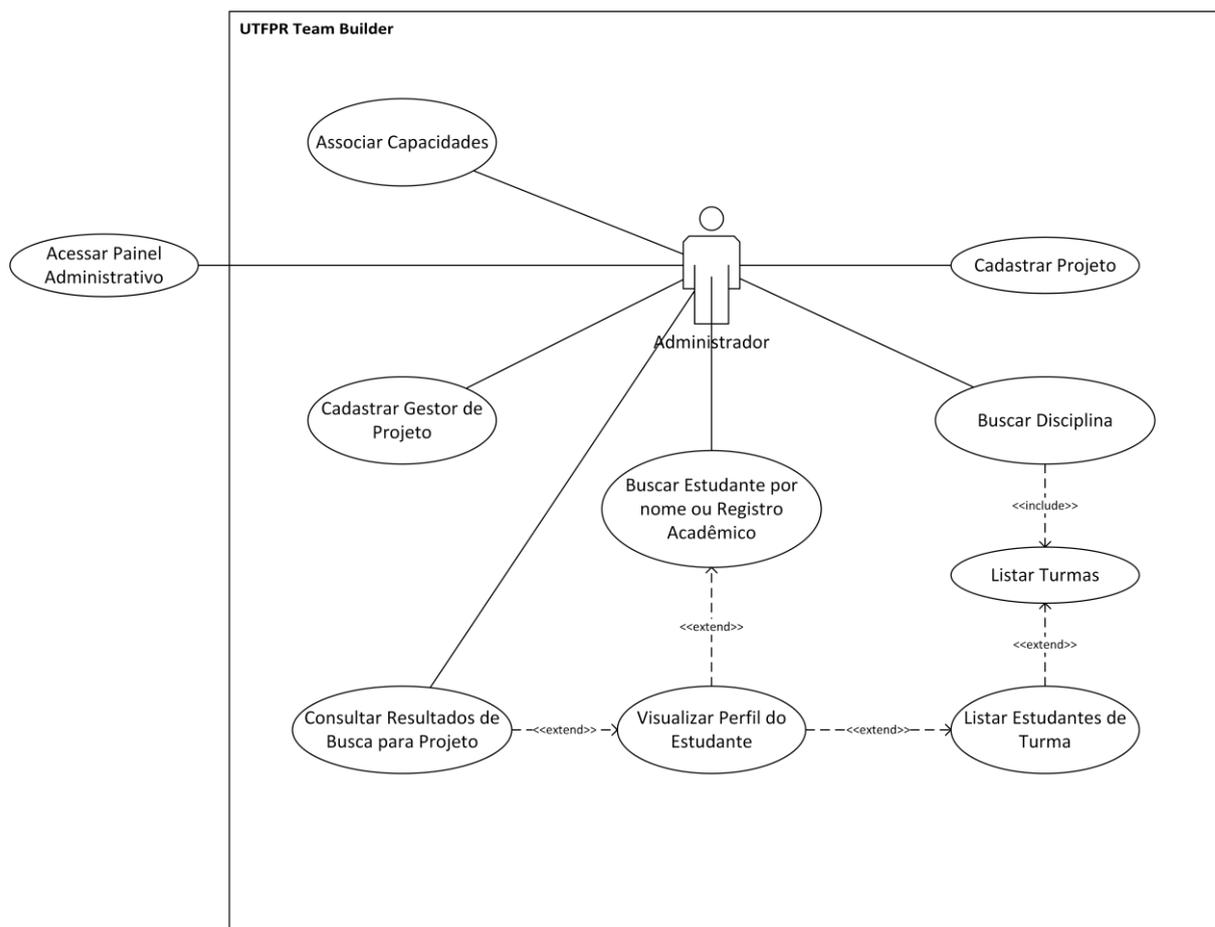


Figura 2 Diagrama de Casos de Uso para o Administrador do Sistema
Fonte: autoria própria

2.2.2.2 Diagrama de classes

O diagrama de classes é apresentado na Figura 3, sendo as classes utilizadas: Turma, Estudante, Disciplina, Capacidade e Projeto. Matriz e SituacaoDisciplina foram representadas como classes, mas se tratam de matrizes associativas do PHP, os quais são representados por conjuntos de chave e valor. Tal escolha de representação tem por finalidade aumentar a clareza do diagrama.

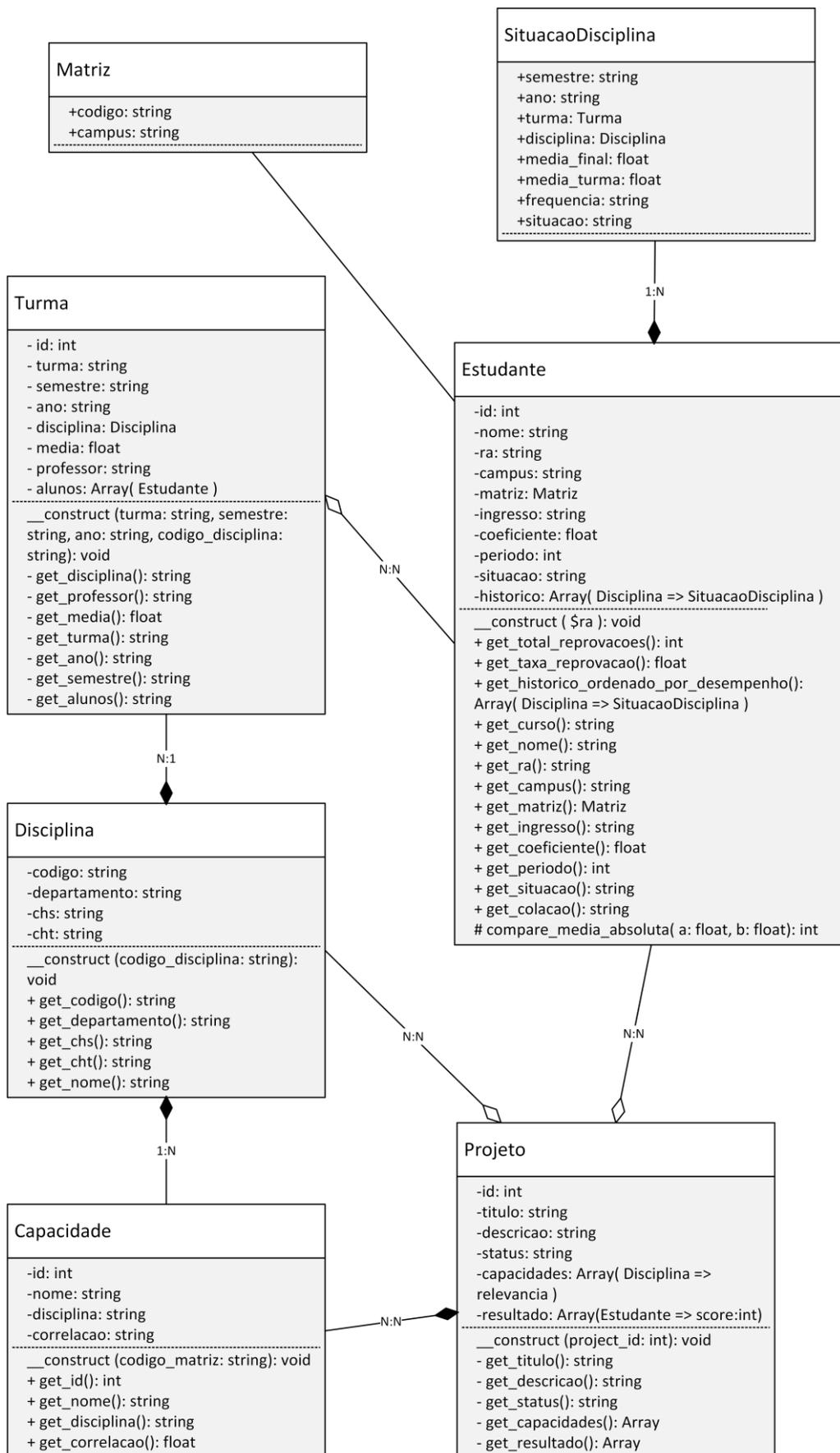


Figura 3 Diagrama de classes do Sistema
 Fonte: autoria própria

2.3 REQUISITOS FUNCIONAIS DO SISTEMA

- RF001 Os usuários deverão autenticar no sistema utilizando usuário e senha
- RF002 O sistema deverá possuir dois níveis de usuário: administrador e gestor de projeto.
- RF003 O usuário com nível administrador poderá adicionar, remover e editar qualquer usuário.
- RF004 Um usuário administrador autenticado pode cadastrar uma área de conhecimento
- RF005 Um usuário autenticado como administrador pode sugerir uma correlação entre área de conhecimento e capacidade técnica e indicar a força da correlação
- RF006 Cada correlação entre capacidade técnica e área de conhecimento possui 4 níveis de intensidade: baixo, médio, alto e representam a mesma coisa.
- RF007 O gestor de projetos pode cadastrar novos projetos e indicar capacidades técnicas necessárias, junto com as suas relevâncias ao projeto.
- RF010 O sistema deve encontrar os discentes mais capacitados em seu banco de dados após um projeto ser cadastrado
- RF011 O sistema deve permitir a visualização do perfil de um discente presente nos resultados de busca por projeto, apresentando suas capacidades técnicas proeminentes e também as de menor desempenho.
- RF012 O sistema deve balancear a relevância da média final com base na média da turma para normalizar o desempenho entre estudantes que cursaram uma mesma disciplina em diferentes turmas.
- RF013 O resultado de uma consulta de indivíduos deve ser apresentado em forma de lista e ordenado por ordem de adequabilidade ao projeto.
- RF014 O sistema deve ser capaz de listar todos os estudantes e permitir a busca por nome para a visualização do perfil de competências.

2.4 REQUISITOS NÃO FUNCIONAIS

- RNF001 O sistema deverá ser acessado via navegador web.

RNF002 O sistema deverá estar acessível para todos os usuários que estejam na mesma rede que ele.

RNF003 O sistema deverá ser independente de sistema operacional do cliente.

RNF004 O sistema deverá utilizar o modelo de dados armazenados pelo sistema acadêmico da UTFPR.

2.5 PATENTES E PROPRIEDADES

Não foram encontradas patentes que utilizassem o histórico acadêmico de uma instituição para pesquisa e filtragem de possíveis candidatos para formar equipes de execução de projetos. As pesquisas foram realizadas em bases de dados brasileiras, dos Estados Unidos e Europeias.

Os termos procurados foram “*GPA assessment*”, “*GPA abilities*”, “*GPA projects*”, “*GPA project management*”, “*academic score*”, “histórico acadêmico”, “capacidades técnicas” e “*academic record*”.

2.6 CONSIDERAÇÕES DO CAPÍTULO

As tecnologias utilizadas foram escolhidas pela sua adequabilidade ao projeto, pela produtividade que apresentariam e pela solidez no mercado, uma vez que é preciso permitir que o sistema evolua para atender mais necessidades dentro do seu escopo no futuro. O Wordpress é uma escolha assertiva neste propósito, uma vez que sua REST API pode servir para dividir dados com uma aplicação para dispositivos móveis no futuro, por exemplo. Além disso, a estrutura do seu sistema com abstrações de dados permite uma expansão ilimitada de funcionalidades, ficando apenas a limitação de desempenho, uma vez que sua alta abstração acaba refletindo em alta necessidade de processamento para lidar com as abstrações e a infinidade de possibilidades de customização. Por fim, a grande quantidade de documentação disponível, reflexo da larga utilização comercial, torna mais fácil que outra equipe dê continuidade ao projeto no futuro e torna este menos sensível a erros, visto que há uma equipe de desenvolvimento comprometida com a estabilidade e atualizações da plataforma base.

Os diagramas UML serviram para dar direção ao projeto e documentar a sua estrutura. Além disso, permitem que outros engenheiros de software compreendam o funcionamento da aplicação e a interconexão dos componentes caso tenham de dar prosseguimento ou suporte ao sistema.

Os requisitos funcionais e não funcionais permitiram delimitar o escopo do desenvolvimento para atender os objetivos propostos, além de facilitar o mapeamento de tarefas e servir de métrica de desempenho do produto final.

A pesquisa de patentes mostrou que não existem barreiras legais no âmbito da proteção intelectual que pudessem significar entraves ao desenvolvimento.

3 DESENVOLVIMENTO

Nesta seção são apresentadas as características gerais do sistema, sua arquitetura e os requisitos funcionais e não funcionais.

3.1 CARACTERÍSTICAS GERAIS

O projeto compreende o desenvolvimento de um sistema para identificação de indivíduos com habilidades técnicas utilizando a base de dados de histórico acadêmico da UTFPR, campus Curitiba. O *software* proporciona aos gestores de projetos uma ferramenta para auxílio de tomada de decisões quando estiverem formando suas equipes.

A aplicação é acessada através de uma página *web* por qualquer navegador conectado a mesma rede do servidor em que ela está instalada. Isto permite o funcionamento com dados centralizados e o acesso por qualquer sistema operacional moderno. O sistema possui dois módulos, sendo um deles a tarefa assíncrona de seleção de candidatos e o outro a camada de visualização e controle da aplicação e seu fluxo de execução.

A seguir será explicado o funcionamento dos dois grandes módulos do sistema.

3.1.1 Tarefa assíncrona de seleção de candidatos

No projeto inicial, a arquitetura foi projetada para que uma camada de visualização interagisse com o usuário enquanto uma camada de processamento realizasse os cálculos para seleção dos potenciais candidatos para um projeto. O principal motivador desta decisão foi evitar a sobrecarga do sistema de visualização, que ficaria lento pela quantidade de dados e entregaria uma experiência de usuário ruim.

A tarefa mais lenta e custosa do sistema é a consulta ao banco de dados de histórico dos estudantes e o cálculo de score com base nas especificações do projeto. Para contornar este desafio técnico, tal tarefa foi projetada para ser executada de forma assíncrona e seguindo a ordem de chegada ao banco de dados (Figura 4), ou

seja, uma arquitetura FIFO, que também é chamada comumente de FCFS (KRZYZANOWSKI, 2015).

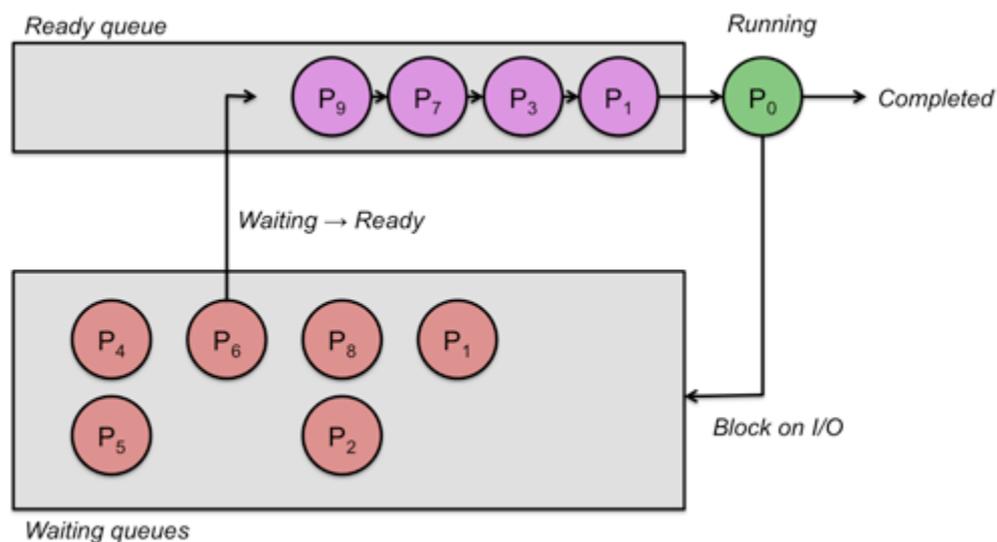


Figura 4 Arquitetura FIFO de escalonamento de tarefas, a mesma utilizada no processamento de projetos
Fonte: (KRZYZANOWSKI, 2015)

Foi desenvolvida uma tarefa agendada que é executada a cada 1 minuto e seleciona o projeto mais antigo aguardando processamento para encontrar os potenciais candidatos para uma demanda. Uma vez terminado o cálculo, o projeto recebe o status de "pronto" e já pode ter seus resultados conferidos na tela "Meus Projetos".

A implementação do assincronismo utilizou como base a classe *WP_Async_Task* proposta por (DELICIOUS BRAINS, 2016). Esta classe utiliza o escalonador CRON nativo do Wordpress para controlar execuções intervaladas de tarefas com alta sobrecarga de processamento.

3.1.2 Módulo de Visualização e telas do sistema

A apresentação gráfica foi programada com HTML, CSS, Javascript e frameworks de larga utilização atual no mercado como JQuery e Twitter Bootstrap. A interface permite ao gestor inserir os dados de um novo projeto e habilidades necessárias dos candidatos. Após realizar a busca com base nos dados solicitados pelo gestor, o sistema envia o projeto para processamento, o qual, uma vez concluído,

apresenta uma lista de candidatos e o seu score de adequabilidade, ordenando-os pela métrica de score. Caso o gestor se interesse por algum indivíduo poderá visualizar mais detalhes do perfil técnico deste, através de representações gráficas que permitem um entendimento mais imediato das habilidades do perfil. Os resultados são armazenados no banco de dados e podem ser consultados a qualquer momento.

A camada de visualização para Wordpress utilizou o tema Understrap (KOENEMANN, 2017) como framework, o que permitiu ganho significativo de tempo nesta etapa.

A primeira tela visível ao usuário após o login é o menu principal, onde as opções em cinza são as acessíveis a todos os níveis de usuários e, as em azul, somente para administradores (Figura 5).



Figura 5 Menu exibido para um usuário com papel Administrador
Fonte: autoria própria

Na Figura 6, é apresentado o menu para usuários com papel Gestor de Projetos, sendo as opções em azul escondidas para não gerar confusão no usuário, visto que ele não tem as permissões necessárias para acessá-las. Mesmo que o acesso seja forçado através da URL no *browser*, o sistema irá redirecioná-lo novamente ao menu principal.



Figura 6 Menu exibido para um usuário com papel Gestor de Projeto
Fonte: autoria própria

A opção “Busca no Sistema” (Figura 7) oferece um campo do tipo texto de linha única com a função auto completar. As opções acessíveis são a lista de estudantes e as disciplinas cadastradas no sistema. Se um estudante for selecionado, o usuário é direcionado até a página de visualização de perfil. Caso uma disciplina seja selecionada, é apresentada a lista de turmas que já cursaram esta disciplina. Ao abrir a turma na lista de turmas, uma lista de estudantes é apresentada e o perfil destes pode ser visualizado. Ou seja, o caminho a partir da busca no sistema sempre terminará em uma visualização de perfil.

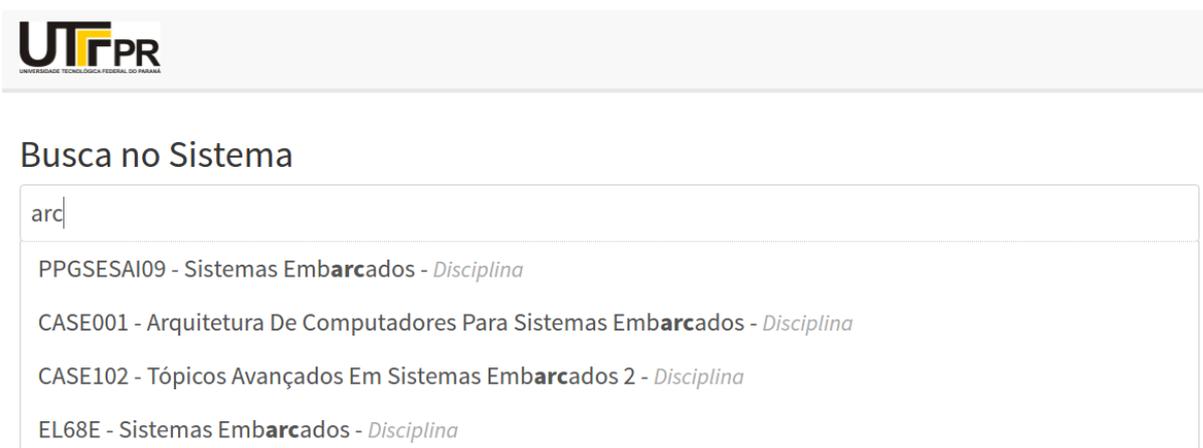


Figura 7 Tela de Busca no Sistema.
Fonte: autoria própria

Na tela “Cadastrar Projeto” (Figura 8), o usuário pode inserir um título para o projeto, que permitirá identifica-lo rapidamente no futuro na tela “Meus Projetos”. Em

seguida, uma breve descrição do projeto permite adicionar mais detalhes sobre o mesmo, entretanto este campo só é acessível ao próprio gestor e ao administrador, não sendo relevante fazer uma descrição muito detalhada.

Um pouco abaixo, na área de capacidades, é possível descrever as capacidades desejáveis ao projeto. O usuário deve escolher uma capacidade ou disciplina na lista que é oferecida quando o preenchimento do campo “Habilidade/Competência” é iniciado. Em seguida, deverá escolher uma das relevâncias para a capacidade entre as opções: interessante, importante, muito importante e crucial. Tais opções irão definir o fator multiplicativo para a capacidade no cálculo do score de um estudante para um projeto.

Caso deseje adicionar mais capacidades, o usuário deverá clicar em “Adicionar mais” no canto inferior direito da área de capacidades. Ao terminar o preenchimento do projeto, o usuário deverá clicar em “Enviar para processamento”, que fica no canto inferior esquerdo da tela “Cadastrar Projeto”.

O projeto ficará acessível na tela “Meus projetos” após o seu cadastramento conforme os passos acima mencionados.

Cadastrar Projeto

Campos marcados com * são requeridos

Dados do Projeto

Título do Projeto *

Descrição do Projeto

Capacidades

Habilidade/Competência	Interessante	Importante	Muito Importante	Crucial
Programação em Java [227746]	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
EBBS05 - Fundamentos De Processan	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
trabalho	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

PPGA13 - **Trabalho**, Sujeito E Mediações [78667] - *Disciplina*

PPGA11 - História Do **Trabalho**, Da Técnica E Da Tecnologia Nas Organizações [78610] - *Disciplina*

PPGA12 - Comportamento Humano No **Trabalho** [78611] - *Disciplina*

PPGSENC04 - **Trabalho** Individual [78600] - *Disciplina*

PGT2040MD - Tópicos Especiais Em Tecnologia E Sociedade: Discursos Sobre **Trabalho** E Tecnologia Em Textos Da Cultura Ocidental E Nacional [78515] - *Disciplina*

Figura 8 Tela "Cadastrar Projeto"
Fonte: autoria própria

Na página "Meus Projetos" (Figura 9), o usuário encontra uma lista dos projetos cadastrados por ele, os quais são apresentados em forma de blocos. As informações disponíveis são: o título do projeto, a lista de capacidades com seus respectivos pesos e um botão que indica o estado do projeto.

O botão é laranja e desabilitado se o projeto estiver na fila de processamento. Caso o projeto já tenha sido processado, então o botão é apresentado em verde e contém a mensagem “Ver Resultados”.



The screenshot displays the 'Meus Projetos' interface. At the top left is the UTFPR logo (Universidade Tecnológica Federal do Paraná). Below it, the heading 'Meus Projetos' is centered. Two project cards are shown side-by-side. The left card, titled 'Análise financeira por software', is in a 'PROCESSANDO' state, shown with an orange button. It lists three courses: 'Gestão Financeira' (0.75), 'Fundamentos De Programação 1' (0.50), and 'Estrutura De Dados 1' (0.75). The right card, titled 'Projeto de Robô autônomo', is in a 'VER RESULTADOS' state, shown with a green button. It lists two courses: 'Programação em Java' (0.75) and 'Sistemas Inteligentes 1' (1.0).

Projeto	Curso	Nota
Análise financeira por software	Gestão Financeira	0.75
	Fundamentos De Programação 1	0.50
	Estrutura De Dados 1	0.75
Projeto de Robô autônomo	Programação em Java	0.75
	Sistemas Inteligentes 1	1.0

Figura 9 Tela "Meus Projetos"
Fonte: autoria própria

Para um projeto já processado, após clicar em “Ver Resultados”, o usuário é levado até a página individual de resultado do projeto (Figura 10). Nesta página estão disponíveis as informações do projeto e a lista de candidatos selecionados em forma de tabela. Os estudantes são ordenados por score para o projeto e uma coluna de score normalizado em escala percentual é apresentada para permitir uma comparação mais imediata da distância de score entre os candidatos.

Esta lista de resultados sempre estará disponível para consulta posterior na tela “Meus Projetos”. Da lista de estudantes é possível acessar os perfis individuais clicando em “Ver Estudante” na coluna “Ações”.

Projeto de Robô autônomo

Capacidades

Programação em Java: 0.75

Sistemas Inteligentes 1: 1.0

Candidatos Selecionados

Estudante	Score	Score Normalizado	Ações
518689 - Andre Matheus Fedalto	1,16	100,00 %	VER ESTUDANTE
908010 - Rafael Henrique Zaleski Adada	1,10	94,62 %	VER ESTUDANTE
1244167 - Arthur Floriani Martins	1,01	87,10 %	VER ESTUDANTE

Figura 10 Tela de resultado do processamento para um projeto
Fonte: autoria própria

No perfil do estudante (Figura 11), é possível visualizar uma série de dados sobre este, os quais são divididos em: dados de registro, dados de desempenho e áreas de destaque positivo e negativo. Sendo os mais relevantes ao gestor de projeto os dois últimos conjuntos de dados: desempenho e destaques.

Na seção de dados de desempenho, é possível consultar o coeficiente acadêmico, o período no sistema, o índice percentual de Reprovação e as disciplinas de melhor e pior desempenho em relação à média da turma. O índice percentual de reprovação é calculado conforme a equação abaixo:

$$\text{Índice de Reprovação} = \frac{\text{Total de Disciplinas Reprovadas}}{\text{Total de Disciplinas Cursadas}} \times 100$$

Nas seções de áreas de destaque positivo e negativo é possível identificar as cinco disciplinas com maiores discrepâncias positivas ou negativas da média do estudante em relação à média da turma. O formato escolhido para representação foi um gráfico de barras horizontal, onde a média da turma é apresentada em uma barra cinza e a do discente em verde, no caso de destaque positivo, ou vermelho, em caso de destaque negativo. Sobrepondo o cursor do mouse às barras, é possível ver o valor

exato representado por estas, mas também é possível estimar usando a escala inferior do gráfico.

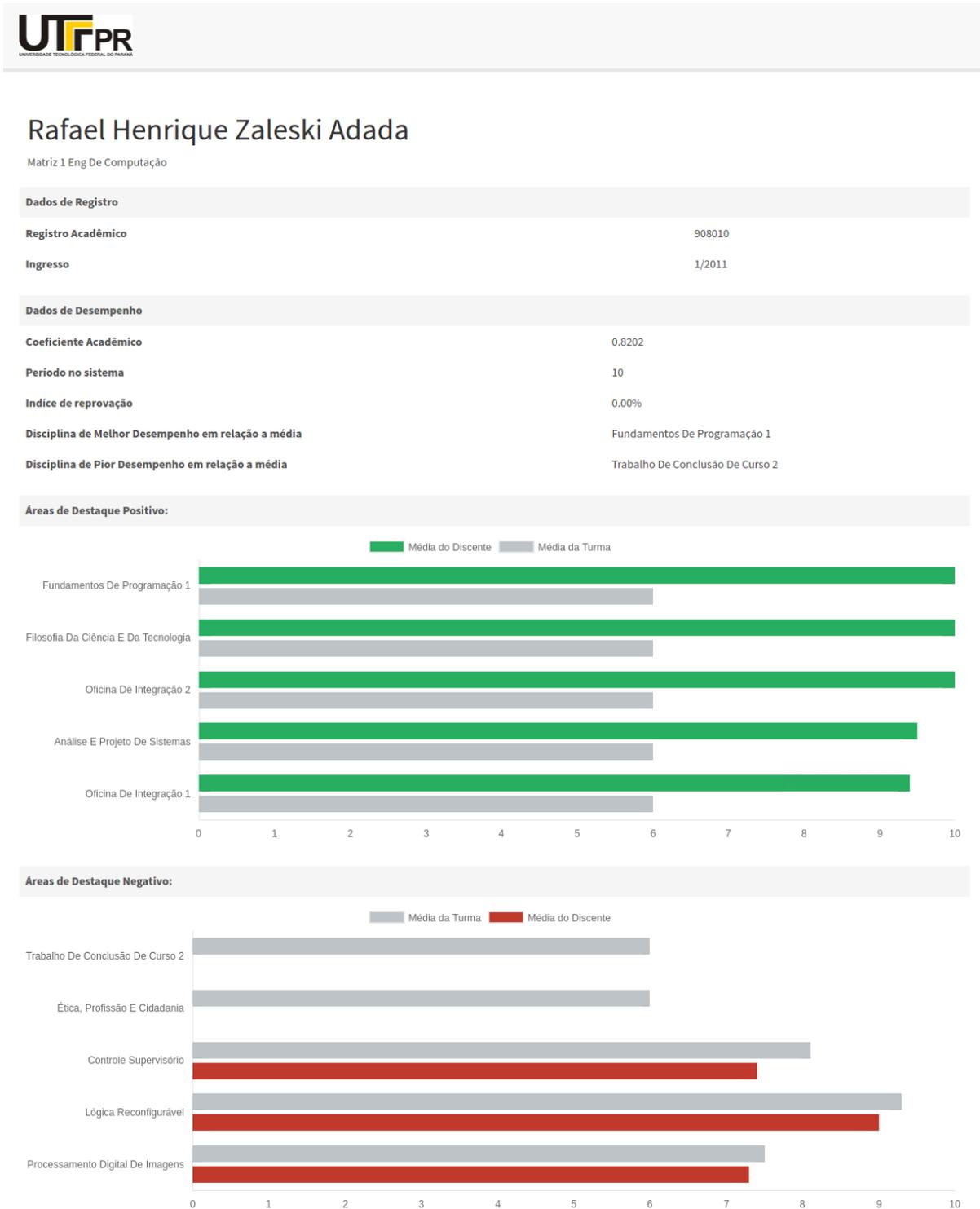


Figura 11 Tela de visualização do perfil de um estudante
Fonte: autoria própria

As telas seguintes são apenas para administradores, tendo acesso bloqueado aos usuários que somente tenham o papel de Gestor de Projetos.

Na opção de adicionar Gestor de Projeto (Figura 12), o administrador deve adicionar o nome e sobrenome do novo usuário, seu departamento na UTFPR e o e-mail institucional. O sistema irá criar uma senha aleatória automaticamente, guardá-la em forma de *hash* no banco de dados e enviá-la ao usuário por e-mail. Os usuários gestores de projeto não podem editar seus perfis e informações, mas o administrador pode através do painel do Wordpress, inclusive para alteração de senhas. Não é possível consultar senhas em texto plano em nenhum local da aplicação, nem mesmo no banco de dados através de acesso direto.

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Adicionar Gestor de Projeto

Campos marcados com * são requeridos

Primeiro Nome do Gestor *

Sobrenome do Gestor *

Departamento UTFPR *

Eletrônica ▼

Email Institucional *

Um link será enviado ao e-mail institucional do gestor para completar seu cadastro e escolher uma senha.

CADASTRAR

Figura 12 Tela para adicionar novo Gestor de Projeto, acessível apenas para administradores
Fonte: autoria própria

Por fim, a tela de associar capacidades (Figura 13) permite adicionar uma abstração de capacidade à uma disciplina, tal funcionalidade é útil quando uma disciplina não compreende um conteúdo em sua totalidade, mas em parte.

O administrador deve selecionar a disciplina da lista com autopreenchimento que é apresentada quando este começa a digitar o nome da disciplina. Em seguida,

deverá nomear a capacidade que está abstraindo e delegar um fator de correspondência (grau de correlação) que é apresentado de forma textual com as opções: baixo, médio, alto e representam a mesma coisa. O mapeamento do fator de correlação é feito dentro do conjunto {0.25, 0.50, 0.75, 1.0} na mesma ordem em que foram citados acima.

Assim que são inseridas as informações, o administrador pode clicar em “Cadastrar Associação” e a capacidade estará disponível no sistema.

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Associar Capacidades

Campos marcados com * são requeridos

Disciplina *

Capacidade *

Grau de correlação

Baixo

Médio

Alto

Representam a mesma coisa

CADASTRAR ASSOCIAÇÃO

Figura 13 Tela para associar capacidades (abstrações de disciplinas)
Fonte: autoria própria

Todo o restante dos dados do sistema, além de funcionalidades de controle, exclusão e edição dos mesmos está acessível pelo painel de controle do Wordpress e não cabe ser explicado neste documento, pois não é de autoria própria e, sim, intrínseco ao sistema gerenciador de conteúdo escolhido.

3.2 ALGORITMO DE CÁLCULO DO SCORE DE UM ESTUDANTE PARA UM PROJETO

O algoritmo de cálculo de score utiliza uma multiplicação de duas matrizes com pesos, onde a matriz A possui as competências solicitadas pelo gestor de projetos e a matriz B corresponde ao desempenho do aluno para a competência solicitada. Se a

competência não for encontrada no histórico do estudante, é atribuído o valor zero no elemento correspondente. O resultado da multiplicação é chamado de score para o projeto.

$$[C_1 \cdot R_1 \quad \dots \quad C_n \cdot R_n] \times \begin{bmatrix} DE1_{C1} & \dots & DE m_{C1} \\ \dots & \dots & \dots \\ DE1_{Cn} & \dots & DE m_{Cn} \end{bmatrix} = [SE_1 \quad \dots \quad SE_m]$$

Na equação acima, a primeira matriz representa o projeto e suas n capacidades, enquanto C_n representa o fator multiplicativo da capacidade n . Se a capacidade é uma disciplina, então o fator é 1, se ela for uma abstração criada a partir de uma disciplina, então ela é um valor no conjunto $\{0.25, 0.50, 0.75, 1.0\}$. O valor em questão foi escolhido pelo administrador que criou a capacidade indicando a força de correlação da disciplina com a capacidade. Em seguida, o fator multiplicativo da capacidade é multiplicado pela relevância da capacidade dentro do projeto, a qual é escolhida pelo gestor de projetos no momento do cadastro da demanda. R_n também pertence ao conjunto $\{0.25, 0.50, 0.75, 1.0\}$.

Na segunda matriz, cada coluna m representa um estudante, enquanto cada linha n , representa a disciplina correspondente à capacidade na coluna n da primeira matriz. Sendo assim, $DE m_{Cn}$ representa o desempenho normalizado do estudante m na disciplina da capacidade n .

A multiplicação gera um somatório e fornece a terceira matriz, onde cada elemento SE_m representa o score do estudante m para o projeto

A seleção inicial de perfis que serão avaliados é realizada através de uma consulta ao banco de dados com cláusula condicional do tipo união de todos os estudantes que possuam ao menos uma das competências em seu histórico. Tal processo é chamado de pré-seleção e visa somente buscar candidatos cujo potencial score para o projeto seja maior que zero.

Após o cálculo do score para um estudante da pré-seleção, este é adicionado na última posição da terceira matriz, a qual, ao final do processo de cálculo para todos os perfis pré-selecionados, é ordenada segundo o score obtido pelos potenciais candidatos, sendo a primeira posição ocupada pelo estudante mais adequado e, a

última, pelo menos adequado segundo os critérios do sistema – soma das variações em relação à média da turma para todas as capacidades.

Por fim, o resultado pode ser consultado na tela de resultado de processamento de projeto, onde os estudantes são ordenados em uma lista e o score absoluto e normalizado é apresentado ao gestor que solicitou o processamento.

3.3 CONSIDERAÇÕES DO CAPÍTULO

Neste capítulo foi apresentado o sistema de controle de acessos da aplicação, o qual utiliza a arquitetura RBAC e possui dois papéis: Administrador e Gestor de Projetos. Além disso, a arquitetura da tarefa assíncrona de processamento de projetos foi apresentada e justificada através do levantamento do problema de desempenho que tratá-la de forma síncrona poderia trazer ao sistema como um todo.

Cada permissão acessível para os papéis foi demonstrada com as imagens das telas e a descrição de funcionamento, desta forma ficam claros os procedimentos de navegação e a forma como a estrutura dos diagramas UML está correlacionada com o produto final obtido.

Por fim, o algoritmo de cálculo de score para um projeto foi explicado e sua equação foi apresentada, junto com a técnica de normalização de desempenho utilizada.

4 GESTÃO DO PROJETO

Nesta seção são apresentados os temas de relevância para a gestão do projeto e sua execução.

4.1 GESTÃO DE TAREFAS

Para a gestão de tarefas e organização do projeto foi utilizada a ferramenta online gratuita Freedcamp (FREEDCAMP INC, 2017). Foi configurado um novo projeto com o nome TCC e o fórum de discussões foi utilizado para anotações pertinentes que poderiam ser esquecidas. Além disso um quadro Kanban foi utilizado para gerenciar a lista de atividades intermediárias, suas prioridades e ordem em que deveriam ser executadas. A metodologia Kanban divide as tarefas em três grupos: não iniciadas, em progresso e finalizadas.

4.2 GIT E VERSIONAMENTO DE SOFTWARE

Para permitir o controle da evolução do código ao longo do tempo, alinhar o desenvolvimento com as tarefas configuradas no quadro Kanban e manter um backup do trabalho realizado, foi utilizada a ferramenta de versionamento de software GIT. O serviço utilizado foi o Bitbucket (ATLASSIAN, 2017), o qual fornece repositórios privados ilimitados para usuários gratuitos. A escolha também foi motivada por familiaridade do desenvolvedor.

4.3 CRONOGRAMA PLANEJADO

O cronograma a seguir representava uma simples estimativa de execução e foi utilizado para fins de planejamento, não tendo qualquer compromisso com a real execução no período informado.

Atividade	Período	Horas
Estudar viabilidade de projeto	31/10/2016 a 15/11/2016	12h

Estudar Técnicas de Busca utilizando Lógica Fuzzy	31/10/2016 a 15/11/2016	20h
Estudar linguagem escolhida para a camada inferior	31/10/2016 a 01/12/2016	30h
Identificar dados armazenados no sistema da UTFPR e gerar dados de teste	15/11/2016 a 01/12/2016	15h
Implementar as interfaces para facilitar testes	01/12/2016 a 23/12/2016	60h
Implementar cadastro de correlações	10/12/2016 a 01/02/2017	40h
Criar e implementar estratégia de normalização de desempenho	10/01/2017 a 01/02/2017	60h
Implementar o cadastro de projetos	01/02/2017 a 01/03/2017	50h
Implementar camada inferior	01/03/2017 a 01/04/2017	80h
Implementar comunicação entre camadas	01/03/2017 a 01/04/2017	80h
Testar e calibrar o Sistema	01/02/2017 a 01/04/2017	100h
Redigir Monografia	01/04/2017 a 01/05/2017	100h
Total de horas		647h

Tabela 1 - Cronograma planejado de execução do projeto

4.4 CRONOGRAMA EXECUTADO

O cronograma a seguir reflete a real execução do projeto e todas as etapas acessórias envolvidas.

Atividade	Período	Horas
Estudo de viabilidade e planejamento	31/10/2016 a 05/01/2017	60h
Estudar ferramentas técnicas disponíveis e seu grau de produtividade para o projeto	15/12/2016 a 01/01/2017	40h
Identificar estrutura de dados armazenada pela UTFPR e UTFAPP	05/01/2017 a 10/01/2017	20h
Importação dos dados e estruturação dentro do sistema	10/01/2017 a 24/01/2017	100h

Criação de estudantes falsos para testes	25/01/2017 a 07/02/2017	60h
Criação de interfaces para as estruturas de dados e telas de entrada de informações	01/02/2017 a 01/05/2017	150h
Implementação de algoritmo do processador de projetos	01/03/2017 a 15/05/2017	120h
Testes do sistema	15/05/2017 a 01/06/2017	40h
Resolução de erros encontrados e novos testes	01/06/2017 a 15/06/2017	40h
Redigir Monografia	01/04/2017 a 15/06/2017	100h
Total de horas		730h

Tabela 2 - Cronograma real da execução do projeto

4.5 CONSIDERAÇÕES DO CAPÍTULO

As ferramentas utilizadas no gerenciamento do projeto auxiliaram na organização do tempo disponível e das tarefas que deveriam ser realizadas. O Freedcamp forneceu um mural Kanban que permitiu um maior controle temporal e definição de metas de trabalho. O Bitbucket auxiliou no versionamento do software, controle da evolução do mesmo e backup para o caso de algum imprevisto.

Quanto ao cronograma, mesmo com uma redefinição de escopo para atingir as metas de entrega, foi necessário cerca de 11% mais tempo do que o previsto inicialmente. Não houveram problemas de organização ou grandes impedimentos para aumentar a definição de tempo, apenas houve dificuldade de estimar o esforço necessário para a realização de um software com escopo amplo. A parte mais trabalhosa foi a importação de dados do UTFAPP, onde foi preciso criar um algoritmo para redefinir os dados em uma maneira que funcionasse no Wordpress. Devido à grande quantidade de dados a importação era lenta e, conseqüentemente, erros tinham um custo temporal muito alto.

Não houveram custos significativos para serem contabilizados, pois se trata de uma aplicação sem hardware ou necessidade de equipamentos especiais para o desenvolvimento e todo o material necessário já estava disponível.

5 CONSIDERAÇÕES FINAIS

O presente projeto apresentou o protótipo funcional de um sistema de seleção de estudantes para projetos utilizando históricos acadêmicos. Os objetivos propostos foram plenamente atingidos sem ressalvas, mas novas funcionalidades para o aperfeiçoamento da filtragem de candidatos podem ser implementadas.

A opção de ferramentas foi assertiva e é centrada na utilização do CMS Wordpress, o qual é líder de mercado e possui um ecossistema de desenvolvimento e atualizações forte, além de uma documentação de fácil acesso e vasta em conteúdo. Baseando-se em uma ferramenta de alta produtividade como esta, é possível expandir a aplicação e compartilhar dados com outros sistemas através da arquitetura de serviços REST nativa oferecida pela plataforma. Além disso, ao basear toda a arquitetura em um serviço web, foram desacoplados os requisitos de hardware e instalação do cliente, facilitando o acesso, visto que todos os sistemas operacionais modernos possuem navegadores de internet pré-instalados e com suporte a todas as tecnologias utilizadas. Sendo assim, há um ganho de sobrevida no projeto e atualizações de segurança serão fornecidas automaticamente e gratuitamente pelos desenvolvedores do CMS.

A etapa de planejamento consistiu em levantamento de requisitos, desenvolvimento de diagramas UML para a arquitetura e documentação da aplicação e escolha e estudo de ferramentas. Tais procedimentos foram considerados como um investimento valioso que permitiu ganho de tempo posterior na etapa de implementação, uma vez que guiaram o desenvolvimento e trouxeram produtividade. Ainda nesta etapa, decidiu-se por utilizar estudantes simulados devido aos aspectos legais envolvidos no fornecimento de dados de terceiros sem autorização, além da questão ética e da manutenção da privacidade individual. Os únicos dados reais do sistema são os do autor da aplicação e dos autores do UTFAPP, que gentilmente cederam seus dados para testes.

Os testes realizados com o sistema obtiveram os resultados esperados, os estudantes foram corretamente selecionados com base no algoritmo proposto de *score* para projetos. Além disso, todos os cadastros de dados apresentados nas telas tinham o funcionamento desejado. A aplicação foi exaustivamente testada e todas as falhas encontradas foram corrigidas. Os gestores de projetos são capazes de cadastrar projetos e consultar resultados de seleção, buscar disciplinas e estudantes

cadastrados e visualizar os dados do sistema, como disciplinas, turmas e estudantes. Os administradores podem realizar todas as tarefas citadas e associar capacidades a disciplinas, criando uma abstração de conhecimento para os que cursam a mesma, além de cadastrar gestores de projetos.

O gerenciamento do projeto permitiu a organização do tempo disponível e das tarefas que deveriam ser realizadas. O Freedcamp forneceu um mural Kanban e ferramentas de gerenciamento de tarefas. O Bitbucket foi utilizado para o controle de versão do *software*. Não houveram custos significativos que devessem ser explicitados, devido ao projeto se tratar de desenvolvimento de *software* sem integração com *hardware* e todas as ferramentas de trabalho já estarem disponíveis. Em relação ao inicialmente planejado, foi gasto cerca de 11% mais tempo que a estimativa inicial, mesmo com a redução do escopo do projeto. A preocupação em entregar um protótipo final plenamente utilizável consumiu grande parte do tempo com ajustes, além de subestimar o esforço para a importação de dados externos. Entretanto, mesmo frente a estes fatores, não se considera a discrepância temporal muito grande, visto a dificuldade de estimar o tempo de desenvolvimento de um software com escopo amplo mantendo precisão absoluta.

Neste protótipo, a abstração de capacidades é feita com base em um nome de disciplina, o que exige profundo conhecimento das disciplinas pelo gestor e adiciona subjetividade ao sistema. Para desenvolvimentos futuros, a partir da estrutura oferecida pela aplicação apresentada, com toda sua funcionalidade de interação com o usuário e processamento básico já projetada, é possível trabalhar em algoritmos mais complexos de mineração de dados e inteligência artificial utilizando o texto descritivo das ementas das disciplinas.

Além disso, pode ser implementado um sistema de votação para as autoridades ou professores de determinada área arbitrarem o grau de relevância de uma disciplina para uma capacidade. Por exemplo, para a capacidade de programação em Java pode ser arbitrado, através de votação, que a disciplina IF62C do Departamento de Informática da UTFPR Campus Curitiba possui relevância 0.72, valor o qual seria utilizado na multiplicação da matriz de pesos para gerar um resultado mais confiável.

REFERÊNCIAS

ATLASSIAN. Bitbucket - a solução GIT para equipes profissionais, 2017. Disponível em: <<https://bitbucket.org/product>>. Acesso em: 02 jun. 2017.

CORBAN WORKS LLC. Order Bulk Identities, 2017. Disponível em: <<http://www.fakenamegenerator.com/order.php>>. Acesso em: 16 abr. 2017.

DELICIOUS BRAINS. How to do Background Processing in WordPress Plugins and Themes, 2016. Disponível em: <<https://deliciousbrains.com/background-processing-wordpress/>>. Acesso em: 17 abr. 2017.

FLANAGAN, D. **JavaScript: The Definitive Guide**. [S.l.]: O'Reilly & Associates, 2006. 16 p. Acesso em: 3 jun. 2017.

FREEDCAMP INC. Freedcamp - Free project Management, 2017. Disponível em: <<https://freedcamp.com/>>. Acesso em: 02 jun. 2017.

KOENEMANN, H. Understrap: The Bootstrap 4 + WordPress Theme Framework, 2017. Disponível em: <<https://understrap.com/>>. Acesso em: 03 jun. 2017.

KRZYŻANOWSKI, P. Process Scheduling: Who gets to run next?, 2015. Disponível em: <<https://www.cs.rutgers.edu/~pxk/416/notes/07-scheduling.html>>. Acesso em: 3 jun. 2017.

MAZIERO, C. A. **Aspectos de Segurança**. [S.l.]: [s.n.], 2013.

MOZILLA DEVELOPER NETWORK. Introdução ao HTML, 2016. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/HTML/Introduction>>. Acesso em: 07 dez. 2016.

MOZILLA DEVELOPER NETWORK. O que é CSS, 2016. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS/Getting_Started/Oque_é_CSS>. Acesso em: 07 dez. 2016.

PAWEŁCZAK, Ł. EasyAutocomplete: JQuery autocomplete plugin., 2017. Disponível em: <<http://easyautocomplete.com/>>. Acesso em: 05 fev. 2017.

PHP GROUP. Manual do PHP: O que o PHP pode fazer?, 2016. Disponível em: <https://secure.php.net/manual/pt_BR/intro-whatcando.php>. Acesso em: 07 dez. 2016.

PHP GROUP. Manual do PHP: Prefácio, 2016. Disponível em: <https://secure.php.net/manual/pt_BR/preface.php>. Acesso em: 07 dez. 2016.

ROBBINS, S. P.; DECENZO, D. A.; VERHULST, S. L. **Fundamentos da administração de recursos humanos**. 11^a. ed. [S.l.]: Elsevier Brasil, 2015.

SULTANOV, D. Checkboxes and radio button customization (jQuery and Zepto) plugin, 2017. Disponível em: <<http://icheck.fronteed.com/>>. Acesso em: 05 fev. 2017.

UTFAPP. UTFAPP: Aplicativo completo da UTFPR., 2017. Disponível em: <<http://utfapp.com/>>. Acesso em: 20 maio 2017.

W3TECHS. Usage of content management systems for websites, 2017. Disponível em: <https://w3techs.com/technologies/overview/content_management/all/>. Acesso em: 19 maio 2017.

WORDPRESS. Access your WordPress site's data through an easy-to-use HTTP REST API, 2017. Disponível em: <<http://v2.wp-api.org/>>. Acesso em: 19 maio 2017.

WORDPRESS. Wordpress Codex: Theme Development, 2017. Disponível em: <https://codex.wordpress.org/Theme_Development>. Acesso em: 19 maio 2017.

WORDPRESS. Wordpress Codex: Plugins. Disponível em: <<https://codex.wordpress.org/Plugins>>. Acesso em: 19 maio 2017.