

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Tadeu Rodrigues dos Santos Braga

**Desenvolvimento de um Sistema de Cadastro
de Atividades do Docente**

Uberlândia, Brasil

2019

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Tadeu Rodrigues dos Santos Braga

**Desenvolvimento de um Sistema de Cadastro de
Atividades do Docente**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Sistemas de Informação.

Orientador: Bruno Augusto Nassif Travençolo

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2019

Tadeu Rodrigues dos Santos Braga

Desenvolvimento de um Sistema de Cadastro de Atividades do Docente

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Bruno Augusto Nassif Travençolo
Orientador

Claudio Douglas Gouveia Linhares

Leiliane Pereira de Rezende

Uberlândia, Brasil
2019

Agradecimentos

À Deus, por ter me dado força e saúde em todos momentos.

À Universidade Federal de Uberlândia e seu corpo docente pelo excelente nível de ensino; especialmente ao meu orientador, por todo auxílio que me deu neste trabalho.

À minha família, minha namorada e à família dela por estarem ao meu lado nos momentos difíceis e pelo apoio que me deram a cada momento.

E, aos meus colegas de curso por terem passado comigo esses últimos anos, principalmente ao Marco Antônio, que me auxiliou com as dificuldades que tive na graduação.

Resumo

Uma tarefa importante aos docentes da Universidade Federal de Uberlândia (UFU) é a criação de relatórios com as atividades realizadas dentro de um período de tempo. Essa tarefa possibilita a evolução acadêmica e a avaliação do desempenho deles. No entanto, a criação desses relatórios é feita manualmente por cada docente. Nesse contexto, este trabalho teve por objetivo o desenvolvimento de um sistema que permite ao professor o cadastro das atividades feitas, geração e avaliação dos relatórios por outros professores, contribuindo para a automatização do processo e proporcionando maior praticidade, comodidade e padronização no processo.

Palavras-chave: Cadastro de Atividades Docentes, Sistema, SpringMVC, AngularJS, PostgreSQL.

Lista de ilustrações

Figura 1 – Relação de Classes Funcionais e Titulações (G: Graduado; E: Especialista; M: Mestre; D: Doutor) dos Docentes (CONDIR, 2017).	9
Figura 2 – Diagrama de casos de uso que representa o sistema.	17
Figura 3 – Diagrama Relacional do SCAD.	18
Figura 4 – Diagrama de arquitetura do SCAD com divisão de camadas e tecnologias	20
Figura 5 – Tela de autenticação do SCAD.	24
Figura 6 – Tela inicial do sistema.	24
Figura 7 – Listagem de professores cadastrados.	25
Figura 8 – Cadastro e edição de professores.	26
Figura 9 – Listagem de versões da tabela de pontuação.	26
Figura 10 – Cadastro de versões da tabela de pontuação.	26
Figura 11 – Listagem dos tipos de atividade.	27
Figura 12 – Cadastro e edição de tipo de atividade.	27
Figura 13 – Listagem de tabelas de pontuação no cadastro de tipos de atividade no SCAD.	28
Figura 14 – Listagem de relatórios.	28
Figura 15 – Cadastro de relatórios.	29
Figura 16 – Listagem de atividades.	29
Figura 17 – Cadastro de atividade.	30
Figura 18 – Resumo de Pontuação por Tabela.	30
Figura 19 – Confirmação de encerramento do relatório.	31
Figura 20 – Relatórios para Avaliação.	32
Figura 21 – Confirmação para avaliar um relatório.	32
Figura 22 – Listagem de atividades do relatório na avaliação.	32
Figura 23 – Correção da atividade na avaliação.	33
Figura 24 – Confirmação de encerramento da avaliação do relatório.	33

Lista de abreviaturas e siglas

FACOM	Faculdade de Computação
UFU	Universidade Federal de Uberlândia
SCAD	Sistema de Cadastro de Atividades do Docente
DAO	<i>Data Access Object</i> (Objeto de Acesso aos Dados)
API	<i>Application Programming Interface</i> (Interface de programação de aplicações)
JPA	<i>Java Persistence API</i> (API de Persistência Java)
MVC	<i>Model-View-Controller</i> (Modelo-Visão-Controladora)
JSP	<i>Java Server Pages</i> (Páginas)
JSTL	JSP Standard Tag Library (Biblioteca Padrão de Tag da JSP)
HTML	<i>Hypertext Markup Language</i> (Linguagem de Marcação de Hipertexto)
CSS	<i>Cascading Style Sheets</i> (Folhas de Estilo em Cascada)
CONDIR	Conselho Diretor
Java EE	Java Enterprise Edition (Edição Empresarial Java)
ORM	Object Relational Mapping (Mapeamento Objeto-Relacional)
MIT	<i>Massachusetts Institute of Technology</i> (Instituto de Tecnologia de Massachusetts)
Licença MIT	Licença de <i>software</i> criada pelo MIT

Sumário

1	INTRODUÇÃO	9
1.1	Objetivos	10
1.1.1	Objetivo Geral	10
1.1.2	Objetivos Específicos	10
1.2	Organização do Trabalho	10
2	FUNDAMENTOS TEÓRICOS	11
2.1	Java EE	11
2.2	Padrão MVC	11
2.3	Hibernate e Camada de Persistência	11
2.4	Visões	12
2.4.1	Expression Language e JSP Standard Tag Library	12
2.4.2	AngularJS	12
3	DESENVOLVIMENTO	14
3.1	Método	14
3.2	Atores	14
3.3	Requisitos	15
3.3.1	Requisitos Funcionais	15
3.3.2	Requisitos Não Funcionais	16
3.4	Diagrama de Casos de Uso	16
3.5	Modelo de Dados	17
3.5.1	Ambiente	19
3.6	Visão da Arquitetura	20
3.6.1	Camadas de Front-End	20
3.6.1.1	Controladoras AngularJS	20
3.6.1.2	Páginas JSP	21
3.6.1.3	Folhas de Estilo CSS	21
3.6.2	Camadas de Back-End	21
3.6.2.1	Controladoras Spring	21
3.6.2.2	DAOs	22
3.6.2.3	Úteis	22
3.6.2.4	Fabrica de Entidades	22
4	RESULTADOS OBTIDOS	23
4.1	Módulo principal	23

4.1.1	Tela de Autenticação	23
4.1.2	Tela Inicial	24
4.2	Módulo Administrativo	25
4.2.1	Gerência de Professores	25
4.2.2	Gerência de Versões da Tabela de Pontuação e Tipos de Atividade	26
4.3	Módulo Pessoal	28
4.3.1	Gerência de Relatórios	28
4.3.2	Gerência de Atividades	29
4.3.3	Demais Funcionalidades do Relatório	30
4.4	Módulo Avaliação	31
4.4.1	Gerência de Relatórios Avaliados	31
5	CONCLUSÃO	34
	REFERÊNCIAS	35

1 Introdução

Nos dias atuais, é necessário evoluir e adaptar processos para que se possa obter um cenário otimizado. Nesse contexto, os sistemas de informação são extremamente importantes, pois automatizam rotinas, possibilitam análises de grandes quantidades de dados, oferecem comunicação em tempo real, etc. De acordo com [Garcia \(2005\)](#) essa importância tornou-se fundamental para qualquer instituição de sucesso, de forma que uma empresa que pense em abrir mão de todos os processos e sistemas informatizados, também se torna altamente suscetível ao fracasso.

Atualmente, a evolução de carreira para professores universitários e de ensino básico em algumas instituições federais é um processo inerente a sua atividade, de forma a exigir que os docentes relatem as atividades desenvolvidas e apresentem respectivas documentações comprobatórias. Essa evolução é subdividida nos conceitos de promoção e progressão. Promoção é quando ocorre a passagem do docente para a classe imediatamente superior a que ele se encontra, e progressão é a mudança de nível dentro da mesma classe. Em relação aos docentes de ensino superior da Universidade Federal de Uberlândia (UFU), as classes e níveis estão organizadas como mostrado na Figura 1.

Classe	Denominação	Titulação
A	Auxiliar	G, A ou E
	Assistente A	M
	Adjunto A	D
B	Assistente	G, A ou E
	Assistente	M
	Assistente	D
C	Adjunto	G, A e E
	Adjunto	M
	Adjunto	D
D	Associado	D
E	Titular	D

Figura 1 – Relação de Classes Funcionais e Titulações (G: Graduado; E: Especialista; M: Mestre; D: Doutor) dos Docentes ([CONDIR, 2017](#)).

Cada universidade regulamenta as regras de progressão e promoção acadêmica, porém, não foram encontrados indícios bibliográficos a respeito de melhoria deste processo de organização de atividades de forma a padronizar e digitalizar o mesmo. Nesse contexto, docentes precisam manualmente fazer a organização das suas atividades, utilizando planilhas ou documentos de texto adaptados ao processo. É possível destacar os seguintes problemas nesse processo:

- Erro de cálculo de pontuação das atividades desenvolvidas e do relatório final;
- Falta de padronização do processo, sendo que não há um modelo específico para criação manual do relatório;
- Grande demanda de tempo do professor para confecção e organização do documento.

Sendo assim, este trabalho tem o intuito de extinguir erros de cálculo e padronizar a forma como os relatórios são cadastrados, gerados e avaliados. Desta forma, reduzindo o tempo demandado pelo processo.

1.1 Objetivos

Nesta seção serão descritos os objetivos deste trabalho de forma geral e específica.

1.1.1 Objetivo Geral

O objetivo deste projeto é o desenvolvimento de um software para auxiliar os professores no processo documental das atividades acadêmicas, visando também mapear e melhorar suas etapas. Este software foi nomeado Sistema de Cadastro de Atividades do Docente (SCAD).

1.1.2 Objetivos Específicos

Esse sistema deve permitir:

- A criação de relatórios para a promoção e progressão funcional.
- A avaliação de relatórios por comissão interna de avaliação.

1.2 Organização do Trabalho

Este trabalho está organizado da seguinte forma: no Capítulo 2, são apresentadas as tecnologias, padrões e conceitos utilizados para o desenvolvimento deste trabalho. Após isso, no Capítulo 3, é apresentado o conjunto de modelos e definições que são utilizados ou desenvolvidos no SCAD. De forma mais específica, são apresentados os atores do sistema, os requisitos, os diagramas de análise e implementação dos dados, uma visão arquitetural distribuída em camadas e a especificação de papéis em cada uma dessas camadas. No Capítulo 4, são demonstrados os resultados obtidos após o desenvolvimento do sistema, sendo apresentado o conjunto de telas e funcionalidades do mesmo. Assim sendo, no Capítulo 5, as considerações finais a respeito do sistema, bem como as expectativas de trabalhos futuros que tenham como objetivo aperfeiçoá-lo são expostas.

2 Fundamentos Teóricos

Para a realização deste trabalho foram utilizados os padrões de desenvolvimento de sistemas Java EE, como a arquitetura Modelo-Visão-Controladora (MVC), o *framework* Spring, para a criação ágil de *controladoras*, juntamente com o *framework* Hibernate, além do banco de dados PostgreSQL. As próximas seções detalham essas ferramentas.

2.1 Java EE

Segundo [Caelum \(2015\)](#), Java Enterprise Edition (Java EE) é um conjunto de especificações que visam facilitar a criação de aplicação empresariais. Esses sistemas apresentam cenários comuns como requisições, transações, persistência de dados, dentre outros.

2.2 Padrão MVC

De acordo com [Massari \(2017\)](#), o *Model View Controller (Modelo Visão Controlador)*, o *MVC*, é um padrão arquitetural de software que faz a separação das camadas da aplicação pelo seu papel na interação com o usuário. Desta forma, o Modelo (*Model*) trata os dados da aplicação, como entidades, constantes e regras de negócio. A Visão (*View*) é normalmente representada por telas do sistema, mas podem ser definidas como a saída do sistema para a representação dos dados, podendo esta ser tabelas e relatórios geradas pela aplicação. O controlador (*Controller*) age como um intermediador que converte as entradas e saídas em operações no modelo ou nas visões. O modelo MVC permite um bom isolamento dos conceitos, além de melhorar a reusabilidade e manutenibilidade dos códigos.

Neste trabalho, o padrão MVC é utilizado para obter uma aplicação com camadas bem definidas e com funções diferenciadas, mas integradas entre si, sendo responsabilidade do modelo conter apenas as definições de dados. A controladora deve realizar o controle de rotas, intermediação e manipulação entre dados e visões do sistema. As visões somente apresentam o conteúdo previamente processado e recebem entradas do usuário.

2.3 Hibernate e Camada de Persistência

JPA ou *Java Persistence API* é um *framework* desenvolvido oficialmente dentro da linguagem JAVA que oferece padrões e funcionalidades para persistência de objetos. As anotações são uma das funcionalidades mais importantes deste *framework*, elas são

marcações inseridas no código da aplicação para facilitar a utilização de um comportamento.

O *Hibernate* é um framework ORM (Mapeamento Objeto Relacional) escrito e utilizado em Java seguindo os padrões da JPA com o objetivo de simplificar a comunicação com o banco de dados e realizar o mapeamento de objetos em tabelas a partir de *anotações da JPA*, definindo assim a parte do modelo de dados. Utiliza da chamada de métodos para persistir dados, sendo que na maior parte dos casos não é necessária a escrita de consultas SQL.

Para realização deste trabalho, o PostgreSQL é utilizado como banco de dados, visando manter a compatibilidade com o Sistema de Distribuição de Disciplinas utilizado pela Faculdade de Computação da Universidade Federal de Uberlândia ([SANTOS, 2018](#)).

2.4 Visões

A *Expression Language* (Linguagem de Expressão) e a *JSP Standard Tag Library* (Biblioteca Padrão de Tag da JSP) para evitar a escrita de códigos JAVA dentro das visões. Também foi utilizado o AngularJS, pela necessidade de atualizar os componentes das visões dependendo da escolha do usuário, sem fazer redirecionamento de página.

2.4.1 Expression Language e JSP Standard Tag Library

A *Expression Language* é uma linguagem simples embutida em várias linguagens de programação que permite acessar e alterar em uma visão o valor de uma propriedade do modelo. Por exemplo, se há um objeto que represente um professor e esse possui o atributo nome, o valor pode ser acessado da seguinte forma: `${professor.nome}`.

A *JSP Standard Tag Library (JSTL)* é uma biblioteca de funções do Java EE que permite de forma prática, executar comandos de repetição e decisão, formatação de dados e outros nas próprias telas, ou visões. Por exemplo, para execução do operador condicional if, há a seguinte sintaxe: `<c:if test="condição">[Código HTML a ser exibido se a condição for verdadeira]</c:if>`.

2.4.2 AngularJS

AngularJS é um *framework* estrutural Javascript mantido pelo Google ([ANGULARJS, 2017](#)), que auxilia na criação de páginas que reagem de acordo com a interação do usuário, sem a necessidade de atualização ou redirecionamento. Foi criado para facilitar principalmente a escrita dos códigos da camada visão.

O *AngularJS* permite adicionar regras e lógicas no *front-end* (lado do cliente), contornando o cenário de várias requisições ao servidor *back-end* (lado do sistema) com pequenas interações do usuário. A utilização de um framework Javascript torna os métodos executados no *front-end* mais simples e claros, abstraindo complexidades desnecessárias, facilitando e agilizando o desenvolvimento de aplicações.

Um importante recurso do AngularJS, nomeado *Two Way Data Binding* (Ligação de Dados Bidirecional) é utilizado. Ele permite manter as visões HTML e as controladoras *AngularJS* (Javascript) atualizadas em tempo real, sem necessidade de criação de longas implementações para que isso aconteça.

A base utilizada é o tema SB Admin. Tal base é um tema de código aberto e de licença MIT gratuita que permite o uso comercial, modificação, distribuição e sublicenciamento. Este foi criado e é mantido pelo site de tecnologia Start Bootstrap. O mesmo possui componentes e ferramentas que facilitam o desenvolvimento de painéis administrativos, aplicações web e *dashboards*.

3 Desenvolvimento

Neste capítulo serão descritos os métodos utilizados para o desenvolvimento do trabalho, e os modelos gráficos e textuais que deram base para a criação do sistema.

3.1 Método

Considerando que não foi utilizado um sistema já existente como base, foi necessário identificar as principais demandas dos professores e as etapas fundamentais para o processo de progressão acadêmica. Desta forma, tornando-o mais fácil de realizar e padronizada. Assim, o sistema implementa em uma interface amigável todas as regras e fluxos que foram mapeados anteriormente.

Primeiramente, foi realizado o entendimento do processo a fim de constatar como o mesmo era realizado manualmente. Este era feito livremente utilizando planilhas e documentos de texto por cada professor, sendo assim, era um processo trabalhoso, difícil de se manter um padrão, bastante suscetível a erros de interpretação dos tipos de atividade e calculo de pontos.

Em um segundo momento, foi preciso definir quais daquelas etapas do processo manual seriam de fato implementadas no sistema e como seriam desenvolvidas ao longo do tempo. Em seguida, foram escolhidas as linguagens, *frameworks* e padrões utilizados para a construção rápida de um sistema WEB. As linguagens escolhidas foram HTML, CSS, Javascript e JAVA, pois de acordo com [Salles \(2018\)](#) algumas delas são as linguagens mais utilizadas no mercado. Quanto aos frameworks, utilizamos AngularJS, Spring MVC e Hibernate.

Por fim, o sistema foi liberado para validação pelos professores da Faculdade de Computação da UFU.

3.2 Atores

Para a implementação correta do sistema, foi necessário definir quais os tipos de usuários finais e o papel desempenhado por cada um no processo. Verificou-se também que todos eles são professores da UFU:

Professor Usuário: É um usuário comum do sistema, ou seja, um professor que cadastra suas atividades, realiza a conferência dos dados e da pontuação calculada e finaliza o relatório para que seja feita a avaliação do mesmo, por outros professores.

Professor Administrador: É o responsável por cadastrar, editar e auditar os usuários. Ele pode atribuir e remover qualquer papel de um usuário.

Professor Avaliador: É o responsável por avaliar as atividades contidas nos relatórios finalizados de outros professores.

3.3 Requisitos

O conjunto de requisitos de *software* define todas as funcionalidades e características que foram implementadas no sistema. Estes são divididos em Requisitos Funcionais e Não Funcionais. De acordo com [Bezerra \(2007\)](#), o requisito funcional é aquilo que descreve o que o sistema deve fazer a cada ação do usuário ou de outro sistema. Por outro lado, requisitos não funcionais dizem respeito as características e padrões de qualidade que o sistema deve oferecer.

Alguns conceitos foram desenvolvidos no processo de análise de requisitos, estes estão listados a abaixo.

- **Relatório** - Entidade que inclui todas as atividades do professor no período de 2 anos.
- **Atividade** - Representação daquilo que foi exercido pelo professor. Por exemplo, aula ministrada por um professor específico em uma data e soma pontos dentro do relatório.
- **Versão da Tabela de Pontuação** - Conjunto de tipos de atividade que podem estar no relatório.
- **Tipo de Atividade** - Representa qual a natureza da atividade exercida. Por exemplo, aula ministrada, orientação de TCC, publicação de trabalho, entre outras.

3.3.1 Requisitos Funcionais

- **Cadastro de Usuários (RF1):** Permitir ao professor administrador, o cadastro de usuários do sistema, sendo necessário informar os dados pessoais do professor cadastrado.
- **Realização da Autenticação (RF2):** Permitir que todos os usuários cadastrados realizem acesso ao sistema após a inserção de suas credenciais, devendo obrigatoriamente respeitar os papéis de cada usuário (Usuário, Avaliador, Administrador).
- **Gerência de Relatórios (RF3):** Possibilitar ao usuário comum, visualizar seus relatórios em execução e encerrados, criar um novo relatório com os seus dados,

encerrar um relatório quando desejar, e gerar documentações das atividades incluídas no relatório.

- Os dados do relatório definidos na criação são: Data de Início, Pontuação de Referência por Dia e Versão da Tabela de Pontuação.

- **Gerência de Atividades (RF4):** Permitir o cadastro, edição e exclusão de atividades em cada relatório.
 - Os dados de atividades são: tipo de atividade, descrição, o documento da atividade, quantidade de horas, alunos, período de realização (quando se aplica).
- **Avaliação de Relatórios (RF5):** Permitir ao professor avaliador, avaliar relatórios que já foram encerrados pelo professor comum, desta forma este poderá validar as pontuações de acordo com o cálculo realizado pelo sistema ou corrigir informações das atividades e a pontuação final.
- **Gerência Versões da Tabela de Pontuação (RF6):** Permitir ao professor administrador, cadastrar versões da tabela de pontuação.
- **Gerência de Tipos de Atividade (RF7):** Permitir ao professor administrador, dentro de cada Versão da Tabela de Pontuação, cadastrar, editar e remover tipos de atividade com seus dados.
 - Os dados do tipo de atividade são: nome, descrição, unidade, quantidade de pontos, o teto da pontuação por tipo de atividade, a quantidade mínima da unidade de pontuação e a respectiva tabela a qual aquele tipo de atividade pertence.

3.3.2 Requisitos Não Funcionais

- **Compatibilidade com os Principais Navegadores (RNF1):** O sistema deverá ter compatibilidade com as versões mais recentes dos navegadores *Google Chrome*, *Mozilla Firefox* e *Microsoft Edge*, possibilitando a plena utilização de suas funcionalidades em todos estes.
- **Compatibilidade da Base de Dados (RNF2):** O sistema deverá ser compatível com a base de dados já existente, utilizada pela FACOM no sistema online de distribuição de disciplinas (SANTOS, 2018).

3.4 Diagrama de Casos de Uso

O Diagrama de Casos de Uso é uma ferramenta da engenharia de requisitos que permite uma visão mais clara e resumida das funcionalidades do sistema. O diagrama obtido para o SCAD é mostrado na Figura 2.

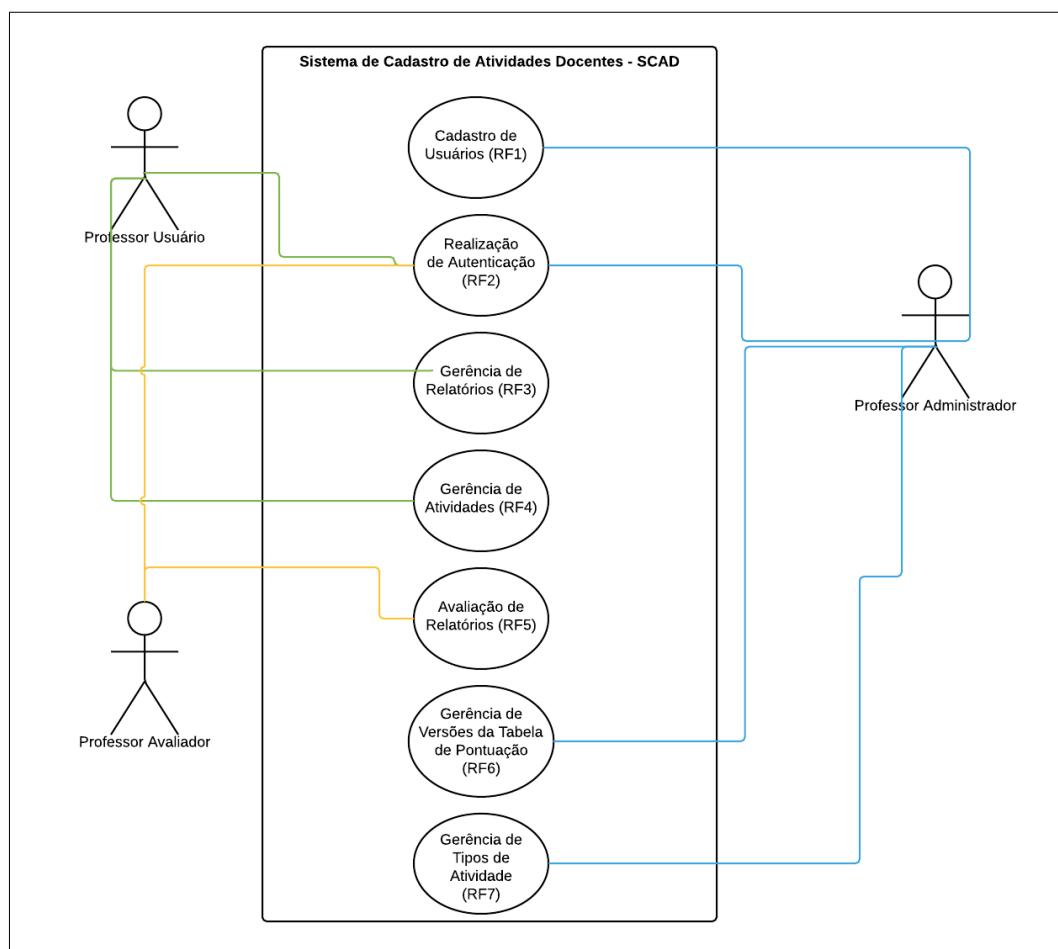


Figura 2 – Diagrama de casos de uso que representa o sistema.

3.5 Modelo de Dados

No desenvolvimento do SCAD, foi necessário utilizar parte do modelo de dados do Sistema de Distribuição de Disciplinas, a fim de aproveitar a base de dados de professores já cadastrados. Dessa forma, o modelo relacional de dados desenvolvido no SCAD está expresso na Figura 3.

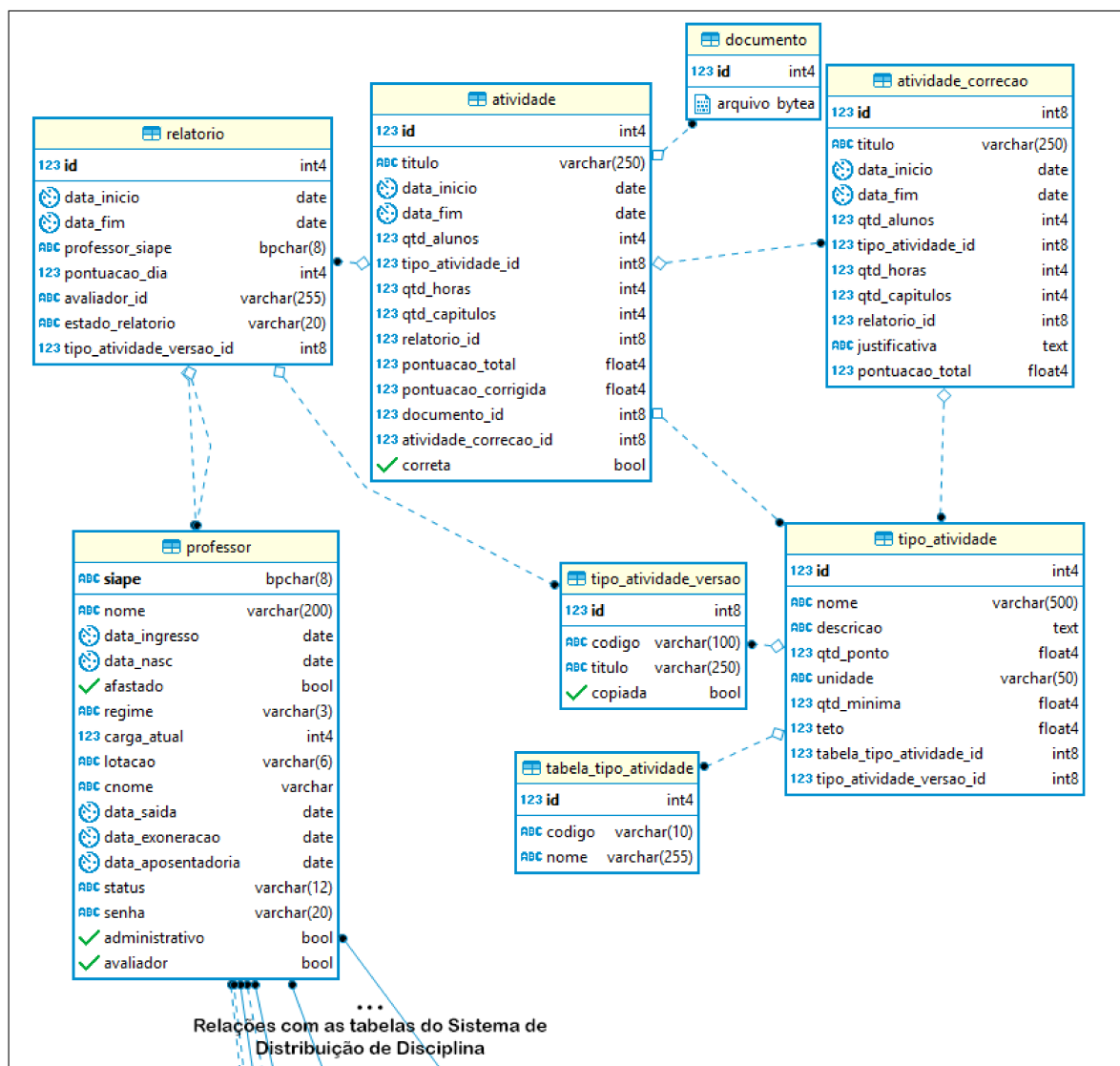


Figura 3 – Diagrama Relacional do SCAD.

As tabelas expressas na Figura 3 representam:

1. **Tabela professor:** A entidade do professor no sistema. Um registro nessa tabela representa o ator Professor Usuário. Quando este registro está com a *flag* (administrativo ou avaliador) marcada como verdadeira, essa entidade também representa o Professor Administrador ou o Professor Avaliador, respectivamente.
 - Os dados de siape, senha e afastado são utilizados para a realização ou bloqueio da autenticação (RF2), quando o professor está afastado.
2. **Tabela relatório:** Pode ser vista como a principal tabela do sistema. Nela é associado o professor responsável pelo relatório e o professor responsável por avaliar o mesmo.
 - São descritas as datas de inicio e fim do relatório, a pontuação de referência por

dia daquele professor no relatório, o estado em que ele se encontra e a versão da tabela de pontuação a ser utilizada.

3. **Tabela atividade:** Esta é a representação para o sistema de uma atividade exercida pelo professor no período do relatório. E ela está associada a um relatório, um documento, um tipo de atividade, e quando este registro está com a *flag* correta marcada como falsa, há também um registro na tabela atividade_correcao.
4. **Tabela atividade_correcao:** Representa uma correção do registro da tabela de atividade. Sendo assim, sempre que existir um registro em atividade_correcao, ela que será considerada no cálculo da pontuação final do relatório. Ela está associada à atividade e ao tipo de atividade.
 - A justificativa é sempre obrigatória quando há uma correção.
5. **Tabela tipo_atividade:** O tipo de atividade é utilizado para especificar uma atividade e definir como será calculada sua pontuação. Por exemplo, uma aula ministrada tem sua pontuação calculada com 1 ponto por hora ministrada, diferentemente de uma orientação para dissertação de mestrado, que tem a pontuação calculada como 2.5 pontos por aluno e mês completo. Essa tabela possui associações para atividade, atividade de correção, a tabela de pontuação (tabela_tipo_de_atividade) e a versão em que esse tipo de atividade é válido.
6. **Tabela documento:** Essa tabela representa o arquivo PDF de uma atividade, que tem o preenchimento opcional na criação.
7. **Tabela tipo_atividade_versão:** Essa tabela serve para dar suporte a várias versões de tipos de atividade. Ela é associada a vários tipos de atividade e também ao relatório. Desta forma, pode-se exibir na criação de uma atividade no relatório somente os tipos de atividade corretos.
8. **Tabela tabela_tipo_atividade:** Pode ser vista como a classificação do tipo de atividade. A finalidade dela é o correto agrupamento dos tipos de atividade ou até mesmo das atividades de um professor dentro do relatório.

3.5.1 Ambiente

Para utilização da base de dados existente, foi utilizado o banco de dados relacional PostgreSQL 9.6. O PostgreSQL é amplamente utilizado no meio acadêmico-científico e corporativo de pequenas e médias empresas, por ter seu código aberto, ser totalmente gratuito e performático. A versão do Hibernate utilizada foi a 5.2.11, sendo o intermediário para a camada de persistência.

3.6 Visão da Arquitetura

Para melhor organização das funcionalidades e melhor isolamento das regras no código, o Modelo MVC e suas extensões, dividem a implementação em diversas camadas, como foi mostrado na Figura 4. Dessa forma, torna-se prático, realizar manutenções e expansões no sistema. Uma vez que as demandas de desenvolvimento comumente podem ser divididas em pequenas partes, essas partes são modificações e melhorias em uma camada específica.

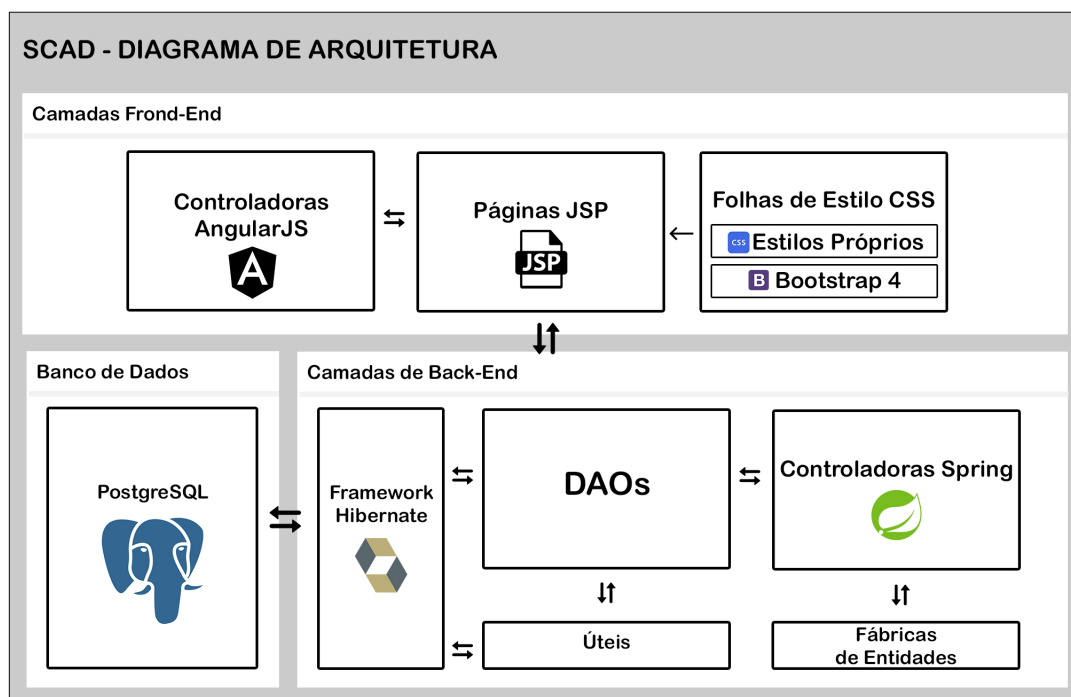


Figura 4 – Diagrama de arquitetura do SCAD com divisão de camadas e tecnologias

3.6.1 Camadas de Front-End

O *front-end* é a parte da aplicação pela qual o usuário interage diretamente, a interface.

3.6.1.1 Controladoras AngularJS

Atualmente, o *front-end* tem se tornado cada vez mais complexo, permitindo novas funcionalidades como reatividade, processamentos complexos e manipulações diretas nos dados. Isso permite que as visões ou telas sejam capazes de alterar seu conteúdo instantaneamente de acordo com as interações do usuário, realizar operações e cálculos em que seria inviável envolver o servidor e lidar diretamente com o modelo de dados, trabalhando com diferentes estruturas de dados e suas respectivas operações.

Nesse aspecto, a camada controladora do AngularJS é ideal, pois o Angular traz uma série de ferramentas completas para comunicação com servidores, manipulações das informações e capacidade de executar lógicas a cada entrada do usuário. Sendo assim, sempre que é necessário adaptar as informações antes ou enquanto são exibidas, define-se funções nessa camada.

3.6.1.2 Páginas JSP

Páginas JSP ou *Java Server Pages* são as visões do Modelo MVC e uma extensão da linguagem web HTML no ambiente JAVA. Elas exibem as informações e permitem ao usuário interagir com o sistema. Nessa camada, houve uma preocupação em refinar as informações a fim de exibi-las para os usuários e melhorar a experiência que estes têm com o sistema.

3.6.1.3 Folhas de Estilo CSS

CSS é uma linguagem específica para melhoria da apresentação visual do conteúdo e sua distribuição pelas páginas HTML e JSP. Nessa camada os estilos próprios para adequar o tema utilizado foram criados visando aperfeiçoar e facilitar a usabilidade dos usuários e criar um padrão estético simples ao longo de todo o sistema. Ainda nesse contexto, utilizou-se o *Bootstrap*, que é uma biblioteca de código aberto com componentes visuais prontos e escritos em CSS.

3.6.2 Camadas de Back-End

O *back-end*, por outro lado, é a camada que implementa todas as regras de negócio cobertas no contexto do sistema, define e manipula formalmente o modelo dos dados, é responsável por persistir e recuperar as informações do banco de dados e realizar todo o processamento prévio dessas informações. Abaixo, estão as camadas que foram importantes para o desenvolvimento do SCAD se tratando de *back-end*:

3.6.2.1 Controladoras Spring

No SCAD foram utilizadas as controladoras para controlar os fluxos que usuários acessam, receber dados a serem salvos ou modificados, retornar os valores corretos para as telas e invocar os métodos específicos nas camadas subsequentes. Essa é a principal camada do *back-end*, uma vez que ela comunica-se com todas as outras camadas desse nível e também com o *front-end*. Ela também é responsável por instanciar o DAOs, Úteis e Fábricas de Entidades e implementar regras de negócio do processo.

3.6.2.2 DAOs

DAO é a camada que representa o padrão *Data Access Object* ou Objeto de Acesso aos Dados. No SCAD foi utilizada para invocar os métodos de persistência e recuperação dos dados do Hibernate, realizar consultas em uma linguagem estruturada do *framework* similar ao SQL, chamada HQL ou até mesmo na própria linguagem SQL. Regras que estão fortemente relacionadas ao modelo de dados foram implementadas nessa camada.

3.6.2.3 Úteis

Este pacote isola comportamentos e funções específicas do sistema, como, por exemplo, funções que realizam cálculos de datas; isolam a lógica para cálculos para pontuação das atividades e do relatório; entre outras.

3.6.2.4 Fabrica de Entidades

Muitas vezes, em aplicações web, partes dos dados são recebidos com o objetivo de criar a entidade a partir dessas partes. A fábrica de entidade é responsável, dentro do sistema, por criar cada objeto e adaptar os dados inicialmente inseridos nas visões e repassados pela controladora.

4 Resultados Obtidos

Nesse capítulo o conjunto de implementações realizadas para a primeira versão do Sistema de Cadastro de Atividades do Docente (SCAD), bem como o processo completo abrangido pelo sistema são apresentados. Com o objetivo de facilitar e isolar conceitos no desenvolvimento do sistema, foi adotada uma divisão modular. Sendo assim, cada módulo atua em um conjunto específico de atividades.

4.1 Módulo principal

O desenvolvimento do módulo principal foi necessário visando redirecionar os professores ao fluxo que desejam realizar e permitir o acesso de acordo com o que for configurado. Este módulo é descrito com maiores detalhes nas subseções [4.1.1](#) e [4.1.2](#).

4.1.1 Tela de Autenticação

Uma tela de acesso foi criada para que os professores possam acessar utilizando o código SIAPE e senha definida anteriormente por outro professor administrador. Essa tela é exibida na [Figura 5](#).

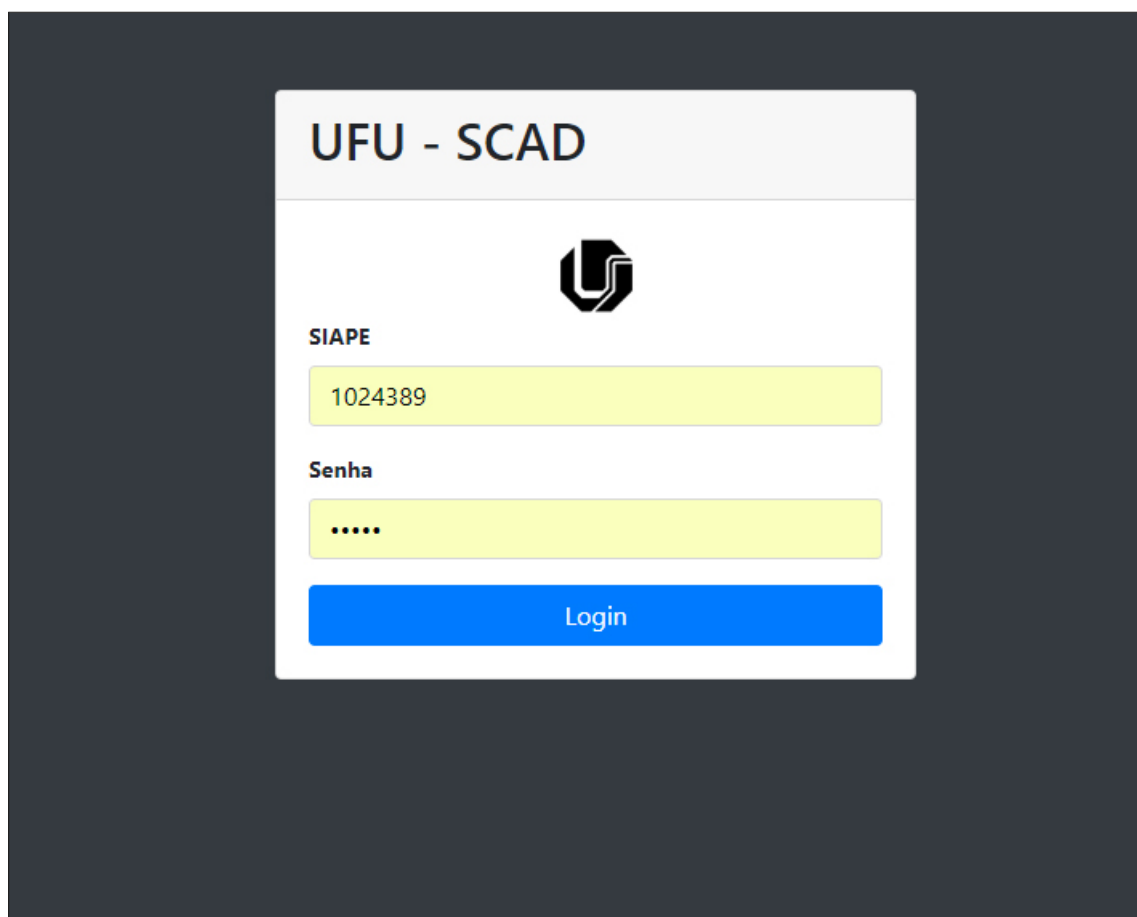


Figura 5 – Tela de autenticação do SCAD.

4.1.2 Tela Inicial

Após a realização do processo de autenticação o usuário é direcionado a tela inicial com a listagem dos módulos que o mesmo tem acesso. Ao escolher um módulo, é aberta a lista de telas para serem acessadas. Essa listagem inicial está explícita na Figura 6. Os módulos acessíveis ao professor podem ser vistos e acessados no lado esquerdo da Figura 6 (Administrativo, Avaliação e Pessoal).



Figura 6 – Tela inicial do sistema.

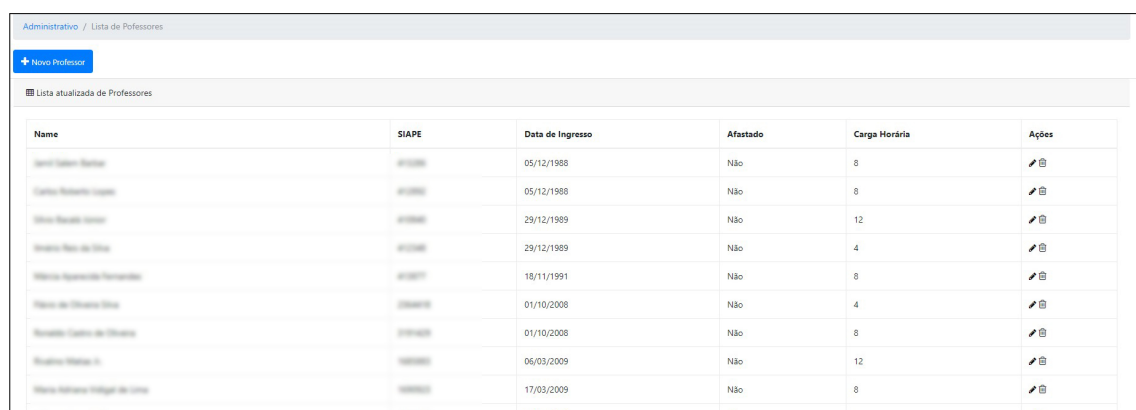
4.2 Módulo Administrativo

O módulo administrativo foi desenvolvido visando manter o controle sobre professores cadastros e parâmetros necessários ao funcionamento do sistema. Foram previstas duas atividades nesse módulo: (I) a gerência de professores e (II) gerência de tabelas de pontuações.

4.2.1 Gerência de Professores

Telas específicas para a manipulação do cadastro de professores no sistema foram criadas, de acordo com o Requisito Funcional (RF1), descrito na subseção 3.3.1.

A listagem de professores cadastrados foi desenvolvida com a exibição de dados chave na tabela, de acordo com a Figura 7.



Nome	SIAPE	Data de Ingresso	Afastado	Carga Horária	Ações
David Cabral Santos	411288	05/12/1988	Não	8	 
Carlos Roberto Lopes	411288	05/12/1988	Não	8	 
Oliver Ricardo Costa	411288	29/12/1989	Não	12	 
Marcelo Reis da Silva	411288	29/12/1989	Não	4	 
Mônica Aparecida Fernandes	411287	18/11/1991	Não	8	 
Flávia do Oliveira Silva	224408	01/10/2008	Não	4	 
Renata Castro do Oliveira	224408	01/10/2008	Não	8	 
Paulo Roberto S.	148802	06/03/2009	Não	12	 
Mônica Mariana Világio de Lima	148802	17/03/2009	Não	8	 

Figura 7 – Listagem de professores cadastrados.

Além disso, implementou-se a tela para criação e edição dos professores. Ela permite que sejam inseridos dados de acesso, informações de recursos humanos, registros burocráticos e papéis que sistema utiliza para permitir ou negar o acesso do professor à determinadas partes do sistema, como demonstra a Figura 8.

Administrativo / Cadastro de Professores

Cadastro de Professores

SIAPE: 123456 Senha: Digite a senha

Nome Completo: Teste Antonio Carlos 5

Regime: 20 Horas Carga Horária: 10 Lotação: Monte Carmelo Status: Ativo

Data de Nascimento: 01/01/2000 Data de Ingresso: 01/01/2001 Data de Saída: 01/01/2002 Data de Aposentadoria: 01/12/2004

Data de Exoneração: 01/11/2004 Apellido: Antonio

☐ Afastado ☐ Administrador ☐ Avaliador

[Voltar](#) [Salvar](#)

Figura 8 – Cadastro e edição de professores.

A operação de remoção do professor foi criada. Essa operação gera um alerta de confirmação quando o professor clica no ícone de lixeira na listagem de professores.

4.2.2 Gerência de Versões da Tabela de Pontuação e Tipos de Atividade

Visando garantir que o SCAD seja capaz de comportar as mudanças dos tipos de atividade de acordo com o especificado pela UFU ao longo do tempo, foram desenvolvidas telas de gerência de versões da tabela de pontuação e dos tipos de atividade. As versões de tabela de pontuação do sistema podem ser visualizadas ou cadastradas, como mostra a Figura 9 e Figura 10, respectivamente.

Administrativo / Versões de Tabela de Pontuação

+ Nova Versão

1
1.00 - SIAG - VERSÃO INICIAL
Visualizar Tipos de Atividade

2
3.00 - teste
Visualizar Tipos de Atividade

Figura 9 – Listagem de versões da tabela de pontuação.

Administrativo / Versão de Tabela de Pontuação

Versão de Tabela de Pontuação

Código: 2.00 Título: Teste

[Salvar](#)

Figura 10 – Cadastro de versões da tabela de pontuação.

O Tipo de Atividade é a entidade que define as regras de pontuação de cada atividade que o professor usuário lançar em seus relatórios. Para cada versão de tabela de pontuação é possível associar diversos tipos de atividade. Ao tipo de atividade, são permitidas inserção, leitura, edição e remoção. As operações de leitura e inserção são demonstradas na Figura 11 e na Figura 12, respectivamente.

Código	Nome e Descrição	Regras de Pontuação
1	Aula teórica ou prática de componentes curriculares ministrados na modalidade presencial ou a distância, na Educação Básica, na Educação Profissional, em cursos de graduação ou pós-graduação stricto e lato sensu da UFU, aprovadas pelo Conselho da Unidade. Obs: 1 - Comprovação por cópia do diário de classe dos componentes curriculares ministrados ou declaração do coordenador do curso ou diretor de unidade especial, discriminando a quantidade de horas-aula ministradas no interstício e o número de discentes matriculados. 2 - Para componentes curriculares ministrados por mais de um docente, a pontuação deverá ser atribuída ao docente de acordo com a carga horária ministrada pelo mesmo. 3 - Componentes curriculares ministrados num mesmo horário, concomitantemente para diferentes turmas e/ou cursos e/ou projetos pedagógicos, independentemente do número de diários, devem ser contabilizados uma única vez. 4 - Nos casos das atividades desenvolvidas no formato de estágios, deverão ser consideradas neste item as atividades realizadas com a presença do docente, quando previstas pelo projeto pedagógico do curso, sendo as demais atividades pontuadas nos itens 12 ou 13 da Tabela A1.2.	1 ponto(s) por hora-aula ministrada
2	Oficina Psicoeducacional ministrada em modalidade de aula para alunos com deficiência e demais discentes da educação básica, profissional ou em cursos de graduação, aprovada pelo Conselho da Unidade. Obs: Comprovação conforme especificações 1 e 2 do Item 1.	1 ponto(s) por hora-aula ministrada
3	Atendimento educacional especializado oferecido em modalidade de aula para alunos da Educação Especial (Decreto 5671/2008 e 7611/2011) em decorrência de quadro de deficiência e que estejam regularmente matriculados na educação básica, na educação profissional, em cursos de graduação ou pós-graduação stricto e lato sensu da UFU, aprovado pelo Conselho da Unidade. Obs: Comprovação conforme especificações 1 e 2 do Item 1.	1 ponto(s) por hora-aula ministrada

Figura 11 – Listagem dos tipos de atividade.

A Figura 12 exibe os campos necessários para o cadastro de um tipo de atividade. Desta forma, os relatórios que forem da mesma versão passarão a exibir este tipo de atividade.

Nome (opcional)

Descrição *

Unidade *

Quantidade de Pontos *

Teto de Pontos *

Qtde. Mínima (de alunos, hora...) *

Código CODIR *

Tabela de Pontuação *

Voltar Salvar

Figura 12 – Cadastro e edição de tipo de atividade.

A Figura 13 exibe a listagem dos registros da tabela de pontuação dentro do cadastro do tipo de atividade.

Descrição *

Escolha...

- A1.1 - ATIVIDADES DE ENSINO
- A1.2 - ATIVIDADES DE ORIENTAÇÃO
- A1.3 - PRODUÇÃO INTELECTUAL
- A1.4 - ATIVIDADES DE ENSINO, EXTENSÃO E PESQUISA - SEM REMUNERAÇÃO SUPLEMENTAR
- A1.5 - ATIVIDADES DE ENSINO, EXTENSÃO E PESQUISA - COM REMUNERAÇÃO SUPLEMENTAR
- A1.6 - ATIVIDADES DE GESTÃO
- A1.7 - LICENÇAS, SITUAÇÕES ESPECIAIS E AFASTAMENTOS

Escolha...

Pontos *

Qtd. Mínima (de alunos, hora ou etc.) *

0

Voltar Salvar

Figura 13 – Listagem de tabelas de pontuação no cadastro de tipos de atividade no SCAD.

4.3 Módulo Pessoal

Este módulo foi desenvolvido para que o professor usuário possa realizar o lançamento dos seus relatórios e atividades, conferência de pontuação, envio para avaliação, dentre outras operações.

4.3.1 Gerência de Relatórios

Inicialmente, foi necessário implementar a funcionalidade que permita ao professor usuário manipular seus relatórios no sistema. A listagem dos relatórios é apresentada na Figura 14. Todas as operações a serem realizadas no relatório surgem a partir dessa tela.

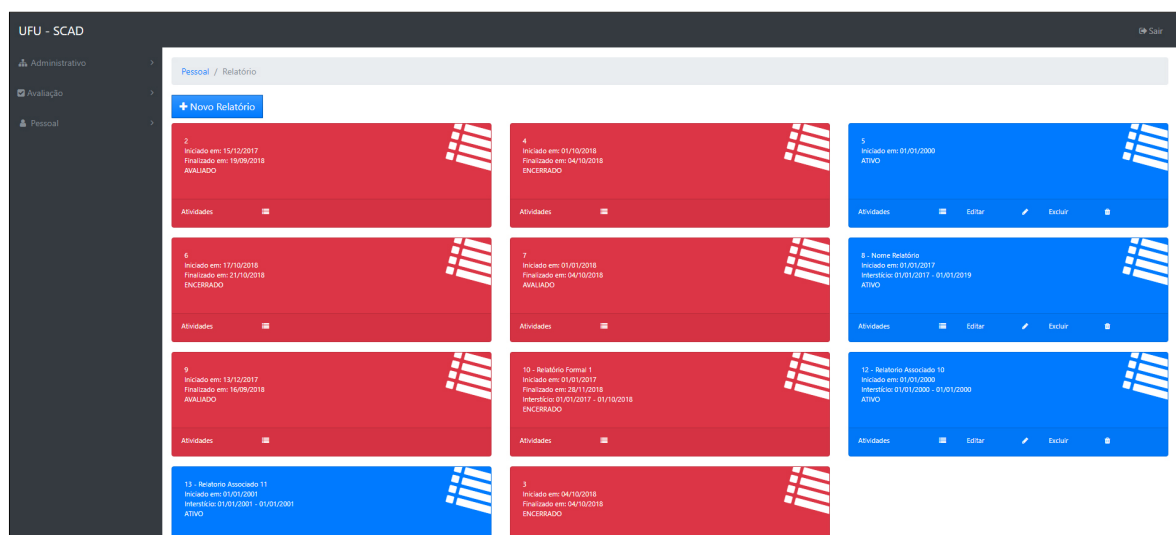


Figura 14 – Listagem de relatórios.

Nesse contexto, o cadastro e edição do relatório permitem ao professor a inserção de dados que são fundamentais no processo controle da atividade docente, como é apresentado na Figura 15.

Pessoal / Relatório

Relatório

Nome *

Relatorio Associado 10

Pontuação de Referência Por Dia *

0

Versão da Tabela de Pontuação *

1.00 - SIAG - VERSÃO INICIAL

Data de Início *

dd/mm/aaaa

Data de Fim

dd/mm/aaaa

Data de Início do Interstício *

dd/mm/aaaa

Data de Fim do Interstício *

dd/mm/aaaa

Salvar

Figura 15 – Cadastro de relatórios.

4.3.2 Gerência de Atividades

A atividade é o principal objeto de cadastro do professor usuário, pois é por meio dela que os professores são capazes de mensurar, resumir e avaliar seu desempenho dentro de um relatório. Essa subseção está diretamente relacionada com a anterior, uma vez que todas as atividades cadastradas estão associadas a um relatório. A listagem e cadastro dessas atividades são mostrados na Figura 16 e Figura 17, respectivamente.

UFU - SCAD

Pessoal / Atividades por Relatório

+ Nova Atividade | Gerar Relatório | Gerar Relatório Por Tabelas | Resumo da Pontuação | Encerrar Relatório

Atividades para o Relatório 8 (Iniciado em: 01/01/2017) - Interstício: 01/01/2017 - 01/01/2019
Versão da Tabela de Pontuação: 1.00 - SIAG - VERSÃO INICIAL

10 resultados por página

Código	Tipo de Atividade	Referência Para o Cálculo	Pontos			
2	Teste Oficina Psicoeducacional ministrada em modalidade de aula para alunos com deficiência e demais discentes da educação básica, profissional ou em cursos de graduação, aprovada pelo Conselho da Unidade. 1 ponto(s) por hora-aula ministrada ⚠ Atividade sem documentação inserida.	Quantidade de Horas 120 horas	120			
3	Atividade diferenciada Atendimento educacional especializado oferecido em modalidade de aula para alunos da Educação Especial (Decreto 5671/2008 e 7611/2011) em decorrência de quadro de deficiência e que estejam regularmente matriculados na educação básica, na educação profissional, em cursos de graduação ou pós-graduação stricto e lato sensu da UFU, aprovado pelo Conselho da Unidade. 1 ponto(s) por hora-aula ministrada ⚠ Atividade sem documentação inserida.	Quantidade de Horas 3 horas	3			

Mostrando de 1 até 2 de 2 registros

Anterior 1 Próximo

Figura 16 – Listagem de atividades.

UFU - SCAD

Pessoal / Atividades por Relatório

Cadastro de Atividades -

Tipo Atividade
3 - Atendimento educacional especializado oferecido em modalidade de aula para alunos da Educação Especial (Decreto 5671/2000 e 7611/2011) em decorrência de quadro de deficiência e que estejam regularmente matriculados na educação básica, na edu...

Descrição
Descrição(opcional)

Quantidade de Horas
130

Obs.: Comprovação conforme especificações 1 e 2 do item 1.
1 ponto(s) por hora-aula ministrada

Documento
Escolher arquivo | Nenhum arquivo selecionado

[Voltar](#) [Salvar](#)

Figura 17 – Cadastro de atividade.

4.3.3 Demais Funcionalidades do Relatório

Foi necessário realizar a implementação de outras funcionalidades para permitir uma melhor visualização dos relatórios e a completa execução do fluxo de cadastro de atividades. O desenvolvimento do resumo de pontuação do relatório tem como objetivo sumarizar a pontuação dos relatórios por tabela de pontuação, o professor usuário é capaz verificar seus lançamentos em cada divisão conceitual das atividades, conforme demonstrado na Figura 18.

UFU - SCAD

Pessoal / Atividades por Relatório

Resumo de Pontuação do Relatório 8

Pontuação de Referência: 720 Pontos

Título da Atividade	Pontuação
Tabela A1.1 - ATIVIDADES DE ENSINO	123
Tabela A1.2 - ATIVIDADES DE ORIENTAÇÃO	0
Tabela A1.3 - PRODUÇÃO INTELLECTUAL	0
Tabela A1.4 - ATIVIDADES DE ENSINO, EXTENSÃO E PESQUISA - SEM REMUNERAÇÃO SUPLEMENTAR	0
Tabela A1.5 - ATIVIDADES DE ENSINO, EXTENSÃO E PESQUISA - COM REMUNERAÇÃO SUPLEMENTAR	0
Tabela A1.6 - ATIVIDADES DE GESTÃO	0
Tabela A1.7 - LICENÇAS, SITUAÇÕES ESPECIAIS E AFASTAMENTOS	0
Total	123

[OK](#)

Figura 18 – Resumo de Pontuação por Tabela.

Outro ponto desenvolvido foi o encerramento do relatório pelo professor usuário, que é o momento em que o professor finaliza o cadastro e modificação de suas atividades

e realiza o envio deste para a avaliação. Esse é demonstrado na Figura 19. A partir deste momento o sistema deixa de permitir a edição do relatório.

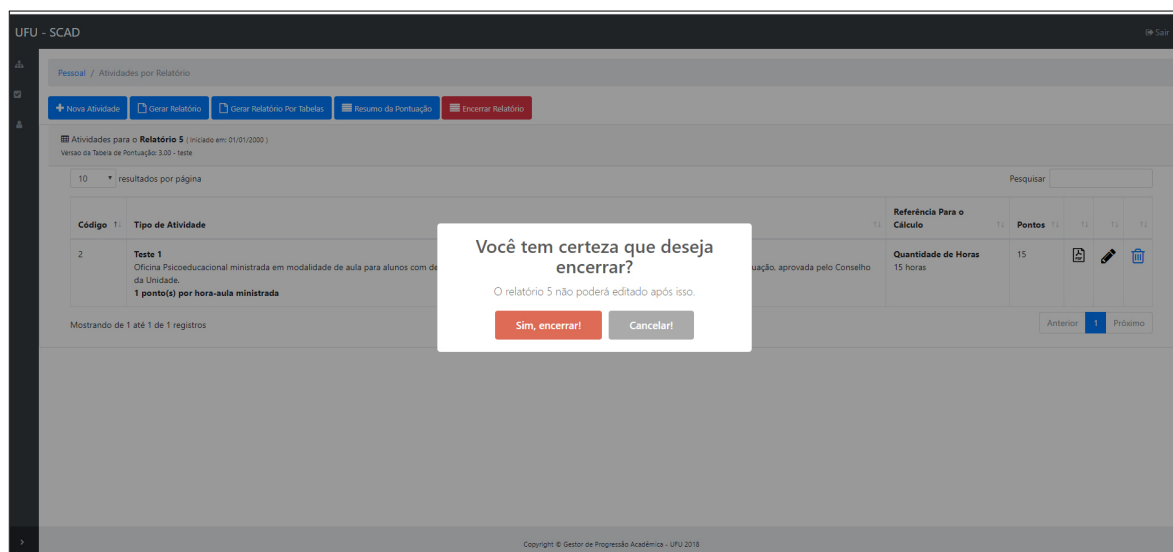


Figura 19 – Confirmação de encerramento do relatório.

4.4 Módulo Avaliação

Este módulo foi desenvolvido visando a finalização do fluxo de um relatório, em que o professor avaliador pode avaliar os relatórios encerrados, podendo alterar a pontuação final do relatório corrigindo cada atividade.

4.4.1 Gerência de Relatórios Avaliados

Primeiramente, foi criada a tela de listagem de relatórios, que está dividida em três partes (conforme mostrado na Figura 20):

1. **Relatórios em avaliação:** Com a listagem de relatórios que o professor escolheu avaliar, porém que ainda não concluiu tal tarefa.
2. **Relatórios a serem avaliados:** Lista todos os relatórios encerrados no sistema que ainda não foram avaliados por nenhum professor.
3. **Relatórios avaliados anteriormente:** São listados aqueles em que o professor já aplicou as devidas correções e finalizou a avaliação.

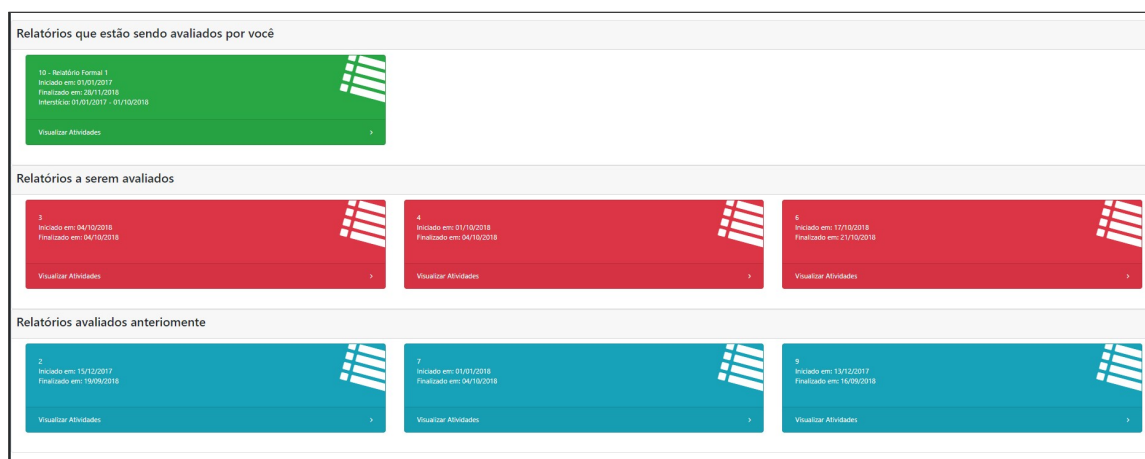


Figura 20 – Relatórios para Avaliação.

Após a visualização dos relatórios, o professor avaliador tem a possibilidade de exibir as atividades de cada relatório se esse não estiver sendo avaliado por ele. Também é possível avaliá-lo como mostrado na Figura 21.

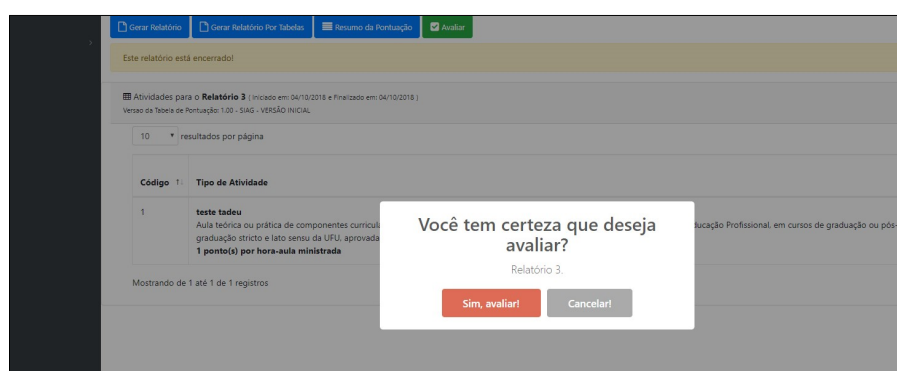


Figura 21 – Confirmação para avaliar um relatório.

Implementou-se também uma listagem das atividades, quando o professor está avaliando o relatório. Ele pode marcar a atividade como correta ou corrigi-la, como exibido na Figura 22. Essa listagem é similar a visão do professor usuário ao abrir um relatório.

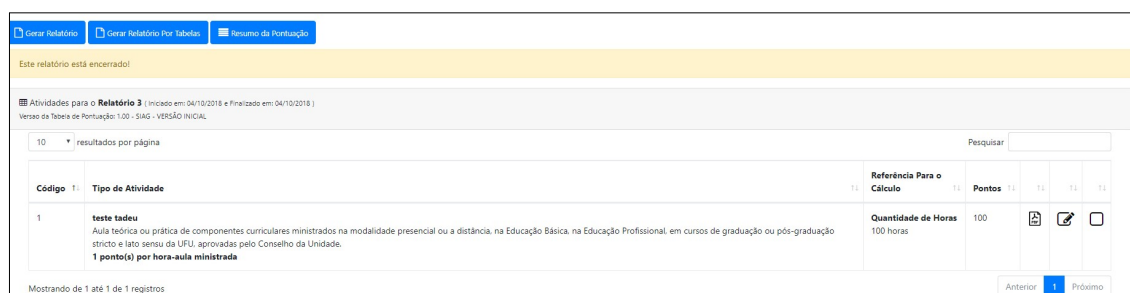


Figura 22 – Listagem de atividades do relatório na avaliação.

A correção de uma atividade do relatório que foi desenvolvida também exibe uma tela similar a do professor usuário ao criar ou editar uma atividade, porém, nesse contexto é exigida uma justificativa para a correção e o professor avaliador pode alterar a nota calculada pelo sistema para a atividade, demonstrado na Figura 23.

A interface de correção de atividade, intitulada "Cadastro de Atividades - 1", apresenta os seguintes campos e elementos:

- Tipo Atividade:** Um menu suspenso com a opção selecionada: "1 - Aula teórica ou prática de componentes curriculares ministrados na modalidade presencial ou a distância, na Educação Básica, na Educação Profissional, em cursos de graduação ou pós-graduação stricto e lato sensu da UFU, aprovadas pelo Conselho da ...".
- Descrição:** Um campo de texto contendo "Aula dada".
- Quantidade de Horas:** Um campo de texto contendo "65", acompanhado de um botão "Visualizar PDF".
- Obs.:** Um bloco de texto amarelo com orientações sobre a pontuação e a justificativa.
- Pontuação Total:** Um campo de texto contendo "64".
- Justificativa:** Um campo de texto contendo "só teste".
- Botões:** "Voltar" (azul) e "Congr" (verde).

Figura 23 – Correção da atividade na avaliação.

O fluxo completo de um relatório no sistema é dado quando o professor avaliador encerra a correção do mesmo, como demonstra a Figura 24. Após este momento, não é possível realizar nenhuma modificação nesse relatório.

A interface de encerramento de avaliação do relatório, intitulada "UFU - SCAD", apresenta os seguintes elementos:

- Menu de Navegação:** "Pesquisa / Atividades por Relatório" com botões "Gerar Relatório", "Gerar Relatório por Tabelas", "Resumo da Pontuação" e "Encerrar Avaliação".
- Alerta:** "Este relatório está encerrado!".
- Informações do Relatório:** "Atividades para o Relatório 10 (Iniciado em: 01/01/2017 e Finalizado em: 28/11/2018) - Interstício: 01/01/2017 - 01/10/2018".
- Filtros:** "Exibir Somente Atividades Corrigidas" e "10 resultados por página".
- Modal de Confirmação:** "Você tem certeza que deseja encerrar a avaliação? Relatório 10." com botões "Sim, encerrar!" e "Cancelar!".
- Tabela de Atividades:** Uma tabela com colunas: Referência Para o Cálculo, Pontos, Quantidade de Horas, e botões de ação. A primeira linha mostra "Aula dada" com 65 horas e 64 pontos.
- Controles:** "Mostrando de 1 até 1 de 1 registros", "Anterior", "1", "Próximo".

Figura 24 – Confirmação de encerramento da avaliação do relatório.

5 Conclusão

O desenvolvimento do SCAD irá otimizar e promover uma melhoria do cadastro, monitoramento e avaliação das tarefas cumpridas pelos professores. O sistema desenvolvido tornou prática e padronizada a realização de processo de cadastro de atividades, que anteriormente eram feitas de forma manual.

Nota-se também que é extremamente importante continuar realizando manutenções e atualizações no SCAD para que este possa continuar adequado as demandas dos docentes e da UFU. Devendo assim, haver um esforço na análise de novas demandas de implementação que por ventura apareçam e sejam proveitosas à evolução sistema.

Este Trabalho de Conclusão de Curso foi extremamente proveitoso tanto para o aprofundamento nas tecnologias adotadas, quanto como contribuição para comunidade docente. Sendo assim, finalizadas as implementações da versão inicial do SCAD, os objetivos propostos para este TCC foram alcançados.

Referências

- ANGULARJS. *AngularJS: Developer Guide: Introduction*. 2017. <<https://docs.angularjs.org/guide/introduction>>. Acesso em 25 de Nov. 2018. Citado na página 12.
- BEZERRA, E. *Princípios de Análise e Projeto de Sistema com UML*. 2. ed. Rio de Janeiro: ELSEVIER, 2007. v. 6. Citado na página 15.
- CAELUM. *O que é Java EE?* 2015. <<https://www.caelum.com.br/apostila-java-web/o-que-e-java-ee/>>. Acesso em 09 de Jan. 2019. Citado na página 11.
- CONDIR. *Resolução nº 03/2017-CONDIR - Progressão e promoção docente*. 2017. <<http://www.progep.ufu.br/legislacoes/resolucao-no-032017-condir-progressao-e-promocao-docente>>. Acesso em 09 de Jan. 2019. Citado 2 vezes nas páginas 5 e 9.
- GARCIA, M. *Informática aplicada a negócios*. 1. ed. Rio de Janeiro: Brasport Livros e Multimídia Ltda., 2005. Citado na página 9.
- MASSARI, J. *Padrão MVC / Arquitetura Model-View-Controller*. 2017. <<https://www.portalgsti.com.br/2017/08/padrao-mvc-arquitetura-model-view-controller.html>>. Acesso em 30 de Nov. 2018. Citado na página 11.
- SALLES, F. *Top 10 linguagens de programação mais usadas no mercado*. 2018. <<https://www.devmedia.com.br/top-10-linguagens-de-programacao-mais-usadas-no-mercado/39635>>. Acesso em 16 de Jan. 2019. Citado na página 14.
- SANTOS, A. V. dos. Trabalho de Conclusão de Curso, *Sistema Online de Distribuição de Disciplinas: Manutenção e Implementação de Novos Requisitos*. 2018. <<https://repositorio.ufu.br/bitstream/123456789/22411/3/SistemaOnlineDistribui%C3%A7%C3%A3o.pdf>>. Acesso em 25 de Dez. 2018. Citado 2 vezes nas páginas 12 e 16.