

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**REINALDO JOÃO BODIN NETTO**

**FERRAMENTA *WEB* PARA GESTÃO DE DOCUMENTOS DA  
PROGRESSÃO E PROMOÇÃO À CLASSE ASSOCIADO DO  
MAGISTÉRIO SUPERIOR DA UTFPR**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PONTA GROSSA**

**2021**

**REINALDO JOÃO BODIN NETTO**

**FERRAMENTA *WEB* PARA GESTÃO DE DOCUMENTOS DA  
PROGRESSÃO E PROMOÇÃO À CLASSE ASSOCIADO DO  
MAGISTÉRIO SUPERIOR DA UTFPR**

Trabalho de Conclusão apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Richard Duarte Ribeiro

Coorientador: Prof. MSc. Giancarlo Rodrigues

**PONTA GROSSA**

**2021**



Ministério da Educação  
**Universidade Tecnológica Federal do Paraná**  
Câmpus Ponta Grossa

Diretoria de Graduação e Educação Profissional  
Departamento Acadêmico de Informática  
Tecnologia em Análise e Desenvolvimento de Sistemas



---

## TERMO DE APROVAÇÃO

### FERRAMENTA WEB PARA GESTÃO DE DOCUMENTOS DA PROGRESSÃO E PROMOÇÃO À CLASSE ASSOCIADO DO MAGISTÉRIO SUPERIOR DA UTFPR

por

REINALDO JOÃO BODIN NETTO

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 04 de maio de 2021 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr. Richard Duarte Ribeiro  
Orientador

---

Prof. MSc. Giancarlo Rodrigues  
Coorientador

---

Prof. Dr. Augusto Foronda  
Membro titular

---

Prof. MSc. Vinícius Camargo Andrade  
Membro titular

---

Prof. MSc. Geraldo Ranthum  
Responsável pelo Trabalho de Conclusão  
de Curso

---

Prof. Dr. André Pinz Borges  
Coordenador do curso

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

## **AGRADECIMENTOS**

Primeiramente, gostaria de agradecer ao Grande Arquiteto do Universo, por nos conceder à vida, experiência tão bela e misteriosa.

Agradeço meu Pai e minha Mãe. Vocês me deram todo o suporte para chegar até aqui. Sou eternamente grato a vocês.

Agradeço o Prof. MSc. Giancarlo Rodrigues e o Prof. Dr. Richard Ribeiro, meus orientadores, por toda a paciência e pelos puxões de orelha. Levarei os aprendizados desta orientação não só para este trabalho, mas também para a vida. Obrigado!

Gostaria de agradecer os amigos do Jiu Jitsu, pela parceria e pelos “treininhos de sábado”, os quais foram essenciais para manter a saúde física e mental.

Um agradecimento especial a todas as mulheres pois vocês tornam os dias mais bonitos.

Enfim, obrigado a todos que estiveram presentes em minha vida neste ciclo. Valeu!

## RESUMO

NETTO, Reinaldo João Bodin. **Ferramenta *Web* para Gestão de Documentos da Progressão e Promoção à Classe Associado do Magistério Superior da UTFPR.** 2021. 94p. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. 2021.

Aos docentes do Magistério Federal é definida a participação em um plano de carreira por meio da Lei Nº 12.772, de 28 de dezembro de 2012, a qual prevê como um dos requisitos para a promoção ou progressão a aprovação em uma avaliação de desempenho. Na UTFPR tal avaliação exige a apresentação de documentos, os quais são utilizados para comprovar a realização de atividades previstas em uma tabela de pontuação. Porém, a gestão manual destes documentos pode apresentar dificuldades como perda de documentos, controle dos pontos por parte do docente e o preenchimento correto do relatório de atividades solicitado pela UTFPR. Para auxiliar nesse processo, este trabalho propõe um sistema *web* para a gestão dos documentos necessários para a progressão ou promoção à Classe Associado do Magistério Superior da UTFPR. Para isso, foi utilizada como estratégia a metodologia de desenvolvimento incremental, a qual tornou possível desenvolver versões intermediárias do *software*, que foram testadas e validadas por docentes que estão realizando ou já realizaram o processo de avaliação. O *software* é organizado em dois módulos: cliente e servidor. O módulo cliente foi desenvolvido utilizando o *framework* Quasar, que utiliza como base o Vue.js. O módulo servidor foi desenvolvido com o *framework* Express, que funciona sobre o ambiente Node.js. Para a persistência dos dados foi utilizado o banco de dados Postgres. Todas as ferramentas utilizadas no processo de desenvolvimento são livres (*open-source*). O resultado final é uma ferramenta que pode ser acessada remotamente por meio de diferentes dispositivos, mantendo a usabilidade das interfaces.

**Palavras-chave:** Desenvolvimento *web*. *Framework* Quasar. Vue.js. *Framework* Express. Carreira do Magistério Superior.

## ABSTRACT

NETTO, Reinaldo João Bodin. **Web Tool for Document Management of Progression and Promotion to the Associated Class of the Higher Education Teaching from the UTFPR**. 2021. 94p. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Federal University of Technology - Paraná. 2021.

Readers and professors of Federal Universities are required to participate in a career plan through the Law No. 12.772, of December 28, 2012, which foresees as one of the requirements for the promotion or progression the approval in a performance evaluation. At UTFPR, such an assessment requires the presentation of documents, which are used to prove the performance of activities provided by a score table. However, the manual management of these documents can present difficulties such as loss of documents, controlling the points on the part of the teacher and the correct completion of the activity report requested by UTFPR. To assist in this process, this work proposes a web system for the management of the documents necessary for the progression or promotion to the Associate Class of the Higher Education of UTFPR. For this, the incremental development methodology was used as a strategy, which made it possible to develop intermediate versions of the software, which were tested and validated by teachers who are carrying out or have already carried out the evaluation process. The software is organized into two modules: client and server. The client module was developed using the Quasar framework, based on Vue.js. The server module was developed with the Express framework, which works on the Node.js environment. For data persistence, the Postgres database was used. All tools used in the development process are free (open-source). The end result is a tool that can be accessed remotely, through different devices, while maintaining the usability of the interfaces.

**Keywords:** Web development. Quasar Framework. Vue.js. Express Framework. Higher Magisterium career.

## LISTA DE FIGURAS

Figura 1 - Fluxo do processo de progressão e promoção à Classe Associado.....	20
Figura 2 - Formato de uma URL.....	25
Figura 3 - Organização das mensagens HTTP.....	27
Figura 4 - Exemplo de tag HTML.....	30
Figura 5 - Exemplo de tag com atributo.....	30
Figura 6 - Estrutura de um documento HTML.....	31
Figura 7 - Sintaxe do CSS.....	32
Figura 8 - Renderização declarativa do Vue.js.....	35
Figura 9 - Componentes do Vue.js.....	36
Figura 10 - Definição de um componente.....	37
Figura 11 - Exemplo de código do Vue.js utilizando componente.....	38
Figura 12 - Exemplo de Single File Component.....	39
Figura 13 - Desenvolvimento Incremental.....	44
Figura 14 - Organização dos diretórios do módulo servidor.....	52
Figura 15 - Organização dos diretórios do módulo cliente.....	53
Figura 16 - Processo de autenticação de usuário.....	55
Figura 17 - Página do módulo de usuários.....	56
Figura 18 - Janela modal para cadastro de usuários.....	57
Figura 19 - Página do módulo de grupos e subgrupos.....	58
Figura 20 - Janela modal para edição de grupos e subgrupos.....	58
Figura 21 - Página do módulo de itens.....	58
Figura 22 - Janela modal para cadastro de itens.....	59
Figura 23 - Passo 1 da primeira etapa para criação de uma tabela de pontuação....	60
Figura 24 - Passo 2 da primeira etapa para criação de uma tabela de pontuação....	61
Figura 25 - Janela modal para adicionar grupo.....	61
Figura 26 - Passo 3 da primeira etapa para criação de uma tabela de pontuação....	62
Figura 27 - Janela modal para adicionar subgrupo.....	62
Figura 28 - Janela modal de gerência dos itens de uma tabela de pontuação.....	63
Figura 29 - Janelas modal para inserção de itens em uma tabela.....	64
Figura 30 - Janela modal para vincular uma tabela de pontuação a um usuário.....	65
Figura 31 - Função para contabilizar a realização de uma atividade.....	66
Figura 32 - Página de registro de atividades.....	67
Figura 33 - Janela modal para cadastrar uma atividade realizada.....	68
Figura 34 - Janela modal para apresentar registros de atividades cadastrados.....	69
Figura 35 - Página de indicadores.....	70
Figura 36 - Página inicial.....	71
Figure 37 - Mensagem do verificador de documentos faltantes quando não há envios pendentess.....	71
Figura 38 - Página de dados e documentos pessoais.....	73
Figura 39 - Página de registro de atividades em diferentes dispositivos.....	76

## LISTA DE QUADROS

Quadro 1 - Organização do plano de carreira para o Magistério Superior.....	17
Quadro 2 - Estrutura da tabela de pontuação para a Classe Associado.....	19
Quadro 3 - Pontuação máxima de subgrupos existentes à classe Associado.....	19
Quadro 4 - Métodos HTTP.....	27
Quadro 5 - Códigos de status.....	28
Quadro 6 - Exemplos de diretivas do framework Vue.js.....	35



## LISTA DE SIGLAS E ACRÔNIMOS

CLI	<i>Command Line Interface</i>
COGERH	Coordenadoria de Gestão de Recursos Humanos
CORS	<i>Cross-Origin Resource Sharing</i>
COUNI	Conselho Universitário da UTFPR
CRUD	<i>Create, Recover, Update and Delete</i>
CSS	<i>Cascading Style Sheets</i>
DNS	<i>Domain Name System</i>
EBTT	Ensino Básico, Técnico e Tecnológico
GED	Gestão Eletrônica de Documentos
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hipertext Transfer Protocol</i>
JWT	<i>JSON Web Token</i>
MVC	<i>Model-View-Controller</i>
NPM	<i>Node Package Manager</i>
SEI	Sistema Eletrônico de Informações
SFC	<i>Single File Component</i>
UI	<i>User Interface</i>
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>
UTFPR	Universidade Tecnológica Federal do Paraná
WEB	<i>World Wide Web</i>

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>11</b>
1.1 Motivação.....	12
1.2 Objetivos.....	13
1.3 Escopo do projeto.....	14
1.4 Estrutura do documento.....	14
<b>2 REFERENCIAL TEÓRICO.....</b>	<b>16</b>
2.1 Plano de carreira docente e os procedimentos.....	16
2.2 Regulamento da avaliação à Classe Associado do Magistério Superior.....	18
2.3 Gestão Eletrônica de Documentos.....	21
2.4 Web.....	24
2.5 Node e Express.....	32
2.5.1 Laravel.....	33
2.6 Vue.js e Quasar <i>Framework</i> .....	34
2.6.1 Angular.....	40
2.6.2 React.....	40
2.7 Banco de Dados.....	41
<b>3 MÉTODO.....</b>	<b>43</b>
3.1 Estratégia de desenvolvimento.....	43
3.2 Ferramentas escolhidas.....	45
3.2.1 Express.....	45
3.2.2 Quasar <i>Framework</i> .....	45
3.2.3 PostgreSQL.....	46
3.2.4 Sequelize.....	46
3.3 Implantação e infraestrutura.....	47
<b>4 DESENVOLVIMENTO.....</b>	<b>48</b>
4.1 Análise dos requisitos.....	48
4.2 Preparação do ambiente de desenvolvimento.....	49
4.2.1 Configuração do módulo servidor.....	50
4.2.2 Configuração do módulo cliente.....	53
4.3 Implementação.....	54
4.3.1 Controle de acesso e módulos base.....	54
4.3.2 Tabelas de pontuação.....	59
4.3.3 Registro de realização atividade.....	66
4.3.4 Indicadores.....	69
4.3.5 Página inicial.....	70
4.3.6 Página Minha conta.....	72
<b>5 CONCLUSÃO.....</b>	<b>74</b>
5.1 Considerações finais.....	74
5.2 Trabalhos futuros.....	76
<b>REFERÊNCIAS.....</b>	<b>77</b>
<b>APÊNDICE A - Tabela de pontuação para Avaliação de Desempenho na progressão e promoção à classe Associado do Magistério Superior.....</b>	<b>82</b>
<b>APÊNDICE B - Diagrama de Caso de Uso.....</b>	<b>94</b>
<b>APÊNDICE C - Diagrama de Entidade-Relacionamento.....</b>	<b>96</b>
<b>APÊNDICE D - Código responsável por preencher o relatório de atividades....</b>	<b>98</b>

## 1 INTRODUÇÃO

Quando um indivíduo decide iniciar sua jornada profissional existem vários caminhos que ele pode escolher trilhar. Tal escolha envolve uma tomada de decisão fundamentada na análise das oportunidades e desafios que cada uma das opções apresentará. Um desses caminhos é a participação em um plano de carreira.

O plano de carreira define uma estrutura hierárquica dos cargos de uma organização, na qual o funcionário é atribuído a uma posição e pode avançar nesta hierarquia uma vez que requisitos (competências) forem satisfeitos (SEBRAE, 2020).

No Magistério Federal existe um plano de carreira para os docentes, o qual é previsto na Lei Nº 12.772, de 28 de dezembro de 2012 (BRASIL, 2012). Para se avançar na carreira do Magistério Federal um dos requisitos, dentre os propostos pela Lei, é a aprovação em uma avaliação de desempenho das diversas atividades realizadas pelo docente durante a sua carreira.

Essas atividades são definidas como itens em uma tabela de pontuação, a qual é organizada em grupos e subgrupos, de forma que a pontuação destes é a soma das pontuações obtidas em cada um de seus itens (UTFPR, 2020). Assim, existem deliberações para definir tabelas de pontuação e regras de contabilização dos pontos para as Classes do plano de carreira no Magistério Federal.

Para solicitar o processo de progressão ou promoção para a Classe Associado do Magistério Superior, são exigidos certos documentos, onde os principais são: relatório de atividades devidamente preenchido e documentos comprobatórios dessas atividades.

A gestão desses documentos pode apresentar dificuldades, pois um documento pode ser perdido, como também danificado. Ainda, como o relatório de atividades é um documento fundamental ao processo, preenchê-lo corretamente é necessário. Entretanto, realizar tal preenchimento manualmente pode induzir a erros, visto que a quantidade de atividades realizadas pelo docente é grande.

Dessa forma, este trabalho propõe o desenvolvimento de uma ferramenta *web* para auxiliar na gestão dos documentos necessários no processo de promoção e progressão à Classe Associado do Magistério Superior da UTFPR.

Tal ferramenta deve também ser multiusuários, permitindo que todos os docentes possam utilizá-la, bastando, para isso, ser cadastrado no sistema. Assim, o *software* possui dois módulos: o módulo cliente e o módulo servidor.

Para implementar o módulo cliente foi escolhido o *framework* Quasar, o qual utiliza como base o Vue.js. O Quasar fornece componentes de interface prontos para serem utilizados, como também outras facilidades ao processo de desenvolvimento. Além disso, com tal *framework* é possível desenvolver páginas que se adaptam a diferentes dispositivos, como *desktop*, *mobile*, entre outros, utilizando o mesmo código-fonte.

Para a implementação do módulo servidor foi escolhido o *framework* Express, que utiliza o ambiente promovido pelo Node.js para funcionar. Tal *framework* utiliza a linguagem de programação JavaScript, assim o desenvolvedor utiliza a mesma linguagem para implementar ambos os módulos. A persistência dos dados na ferramenta é promovida através do Banco de Dados Postgres, de forma que para a interação do módulo servidor com o Banco de Dados utilizou-se do pacote Sequelize.

Para o processo de desenvolvimento do *software* foi escolhida a estratégia de desenvolvimento incremental. Assim, primeiro são desenvolvidas funcionalidades mais básicas, e, incrementalmente, funcionalidades mais complexas. Esta estratégia permite que versões intermediárias do *software* sejam produzidas, as quais podem ser testadas por usuários que tenham experiência com o processo, o que permite corrigir erros durante o processo de desenvolvimento, de forma que o produto final atenda os requisitos levantados.

## 1.1 Motivação

Como a quantidade de itens na tabela de pontuação é grande, conseqüentemente, a quantidade de documentos comprobatórios que deve ser entregue também é. Dessa forma, podem existir dificuldades na gestão desses documentos.

Um ponto no qual podem ser encontradas dificuldades é no armazenamento das cópias físicas dos documentos, bem como sua conservação. Tais cópias podem

ser danificadas ou perdidas, impossibilitando ao docente comprovar a realização da atividade.

Outro ponto importante é o preenchimento manual do relatório de atividades. Como a quantidade de atividades realizadas é grande, o docente pode se perder no cálculo das pontuações que devem ser preenchidas no relatório. Assim, pode ser que tal relatório seja entregue com informações inconsistentes, o que é crítico para o processo de progressão ou promoção.

Existe ainda um outro ponto. Após o envio do portfólio pode ser constatado na avaliação que os pontos contabilizados não são suficientes para que ocorra a progressão ou a promoção requerida. Logo, existe uma dificuldade também no controle dos pontos que se têm em relação com o que é necessário para a aprovação.

Portanto, o desenvolvimento de uma ferramenta que auxilie na gestão dos documentos comprobatórios, bem como documentos pessoais necessários para o processo, no acompanhamento da pontuação obtida e no preenchimento do relatório de atividades, mostra-se como um facilitador para quem precisa realizá-lo.

Ainda, dado o nível de evolução no qual a Internet se encontra, bem como as tecnologias para desenvolvimento de sistemas de *software*, a construção de uma ferramenta *web* para auxiliar os docentes na gestão dos documentos do processo para a progressão e promoção à Classe Associado do Magistério Superior se mostra apropriada.

## 1.2 Objetivos

O objetivo geral desse trabalho é desenvolver uma ferramenta para gestão dos documentos exigidos para progressão ou promoção à Classe Associado do Magistério Superior da UTFPR. Para o cumprimento do objetivo geral deste trabalho, definiu-se os seguintes objetivos específicos:

- Definir uma estratégia para o processo de desenvolvimento;
- Realizar a implementação do *software*

- Validar a ferramenta com docentes que têm experiência no processo de promoção ou progressão à Classe Associado do Magistério Superior.

### 1.3 Escopo do projeto

O escopo do projeto foi delimitado a atender somente à Classe Associado do Magistério Superior. Tal escolha foi realizada pelo fato da carreira de professor de Ensino Básico, Técnico e Tecnológico (EBTT) na UTFPR ter sido descontinuada pelo Governo Federal, isto é, não será permitida a entrada de novos docentes nesta carreira, o que implica que o sistema desenvolvido iria eventualmente deixar de ser usado (BRASIL, 2020, p. 42).

A escolha pela Classe Associado levou em consideração a quantidade de vezes que um mesmo Professor utilizará o sistema e a universalidade da tabela de pontuação. Não foi escolhida a Classe Titular pois esta promoção só ocorre uma única vez, enquanto que a promoção para a Classe Associado, combinada com a progressão para os níveis 2, 3 e 4 desta Classe, somam quatro ocorrências.

Além disso, as regras das tabelas de pontuação da Classe Associado e da Classe Titular são diferentes. Não foram escolhidas as Classes A, B (Assistente) e C (Adjunto) pois a tabela de pontuação para estas é personalizada no acordo entre o Docente e a Chefia, já a tabela de pontuação para a Classe Associado é a mesma para todos os docentes.

### 1.4 Estrutura do documento

A organização deste documento é feita em Capítulos, de forma que, no Capítulo 2, Referencial Teórico, são apresentados os conceitos necessários para o entendimento do trabalho; no Capítulo 3, Método, é apresentada a estratégia escolhida para o processo de desenvolvimento, bem como as ferramentas escolhidas; já, no Capítulo 4, Desenvolvimento, é mostrado como o ambiente de desenvolvimento foi configurado, o processo de implementação do *software* e também decisões tomadas durante este processo; e, no Capítulo 5, Conclusão, são

descritos os resultados do processo de desenvolvimento, como também possibilidades para desenvolvimentos futuros.

## 2 REFERENCIAL TEÓRICO

Aqui serão apresentados e discutidos os conceitos, tecnologias e ferramentas que fundamentam este trabalho. A Seção 2.1 aborda as questões do plano de carreira docente e seus respectivos procedimentos. A Seção 2.2 apresenta o regulamento para a avaliação de desempenho à Classe Associado do Magistério Superior. A Seção 2.3 mostra o processo eletrônico para Gestão de Documentos e os benefícios quando comparado ao processo manual. A Seção 2.4 explana os conceitos da Internet e da *Web*. A Seção 2.5 aborda sobre as ferramentas Express e Node.js. A Seção 2.6 aborda as ferramentas Vue.js e Quasar *framework*. A Seção 2.7 explica conceitos básicos de Bancos de Dados.

### 2.1 Plano de carreira docente e os procedimentos

Na carreira docente do magistério federal é garantido pela Lei Nº 12.772, de 28 de dezembro de 2012 (BRASIL, 2012) a participação em um plano de carreira, o qual é definido como:

Um programa estruturado que estipula o caminho que cada funcionário vai percorrer dentro de uma organização. Ele determina as competências necessárias para cada posição hierárquica e também qual é a expectativa da empresa em relação àquela posição (SEBRAE, 2020).

A Lei Nº 12.772, de 28 de dezembro de 2012 (BRASIL, 2012, p. 2) em conjunto com a Lei Nº 12.863, de 24 de setembro de 2013 (BRASIL, 2013, p. 2) definem os planos de carreira para o Ensino Básico, Técnico e Tecnológico (EBTT) e para o Magistério Superior, bem como definem os cargos existentes nos planos.

Os planos possuem uma divisão por Classes e por níveis, tal que uma Classe é subdividida em níveis, e a evolução na carreira é dada pela ascensão nesta hierarquia. O Quadro 1 ilustra as Classes e níveis existentes no plano de carreira do Magistério Superior.



A evolução na carreira ocorre por meio de progressão ou promoção. A progressão é a “passagem do servidor para o nível de vencimento imediatamente superior dentro de uma mesma Classe (ex: do nível 2 para o nível 3 da Classe C (Professor Adjunto)” (UTFPR, 2020), enquanto que a promoção é a “passagem do servidor de uma Classe para outra subsequente, como por exemplo, do nível 4 da Classe C (Professor Adjunto) para o nível 1 da Classe D (Professor Associado)” (UTFPR, 2020).

Quadro 1 - Organização do plano de carreira para o Magistério Superior

<b>Classe</b>	<b>Denominação</b>	<b>Nível</b>	<b>Regulamento da UTFPR</b>
E	Titular	Único	DELIBERAÇÃO COUNI Nº 14/2014, de 22 de agosto de 2014 <sup>1</sup>
D	Associado	4	DELIBERAÇÃO COUNI Nº 28/18, de 26 de outubro de 2018
		3	
		2	
		1	
C	Adjunto	4	DELIBERAÇÃO COUNI Nº. 05/2017, de 23 de fevereiro de 2017 <sup>2</sup>
		3	
		2	
		1	
B	Assistente	2	
		1	
A	Adjunto-A – se Doutor	2	
	Assistente-A – se Mestre	1	
Auxiliar – se Graduado ou Especialista			

Fonte: Adaptado de UTFPR (2020).

1 <https://nuvem.utfpr.edu.br/index.php/s/iSaxrdZ6IKVsR7j>

2 <https://nuvem.utfpr.edu.br/index.php/s/IPu5ADNwOLPJoQK>

No Quadro 1 observa-se que a carreira dos docentes do magistério superior começa na Classe A nível 1 e termina na Classe E, a qual possui um único nível. Cada Classe possui uma ou mais denominações, utilizadas para descrevê-la. A denominação da Classe A, como pode ser visto no Quadro 1, depende do título do docente, ou seja, se ele é Graduado/Especialista é denominado Auxiliar, se Mestre é denominado Assistente-A ou, então, se Doutor é denominado Adjunto-A.

Os critérios considerados no processo de progressão e de promoção do Magistério Superior estão definidos no parágrafo 2º do artigo 12 da Lei Nº 12.772, de 28 de dezembro de 2012 (BRASIL, 2012, p. 1). Estes critérios são o cumprimento do período de 24 meses de exercício efetivo no nível (para promoção considerar no último nível da classe antecedente) e aprovação em uma avaliação de desempenho individual.

A Lei Nº 12.772, de 28 de dezembro de 2012 (BRASIL, 2012, p. 1) define que uma avaliação de desempenho deve ser realizada e que seu regulamento será definido pela própria instituição de ensino. Na UTFPR essa avaliação e sua regulamentação são definidas pelo Conselho Universitário (COUNI).

No Quadro 1, além da organização do plano de carreira, também são apresentadas quais deliberações do COUNI regulamentam o processo de avaliação para progressão e promoção em cada uma das Classes. Isto implica que tanto as pontuações consideradas na avaliação de desempenho, como o trâmite do processo, podem diferir em função da Classe e nível em que o Docente se encontra.

## 2.2 Regulamento da avaliação à Classe Associado do Magistério Superior

Para a progressão e promoção à Classe Associado do Magistério Superior, as pontuações atribuídas às atividades consideradas na avaliação de desempenho são regulamentadas pela DELIBERAÇÃO Nº. 28/18, de 26 de outubro de 2018 (UTFPR, 2018). A tabela de pontuação que realiza o mapeamento da pontuação para cada uma das atividades existentes está no Apêndice A deste documento.

Tal mapeamento organiza a tabela em uma estrutura de grupos e subgrupos, de forma que um grupo representa uma área específica de atuação. Assim, os

grupos e subgrupos têm itens, os quais definem uma atividade naquela área e atribuem uma pontuação à sua realização.

A tabela de pontuação para a Classe Associado, regulamentada pela DELIBERAÇÃO Nº. 28/18, possui a estrutura de grupos e subgrupos apresentada no Quadro 2.

A pontuação para aprovação na avaliação de desempenho é de 60 pontos, dos quais é requerido um mínimo de 32 pontos no Grupo I, e um mínimo de 12 pontos no Grupo II (UTFPR, 2018). Ainda, alguns subgrupos têm uma pontuação máxima permitida estabelecida, tal como apresenta o Quadro 3.

Quadro 2 - Estrutura da tabela de pontuação para a Classe Associado

<b>Grupo</b>	<b>Subgrupo</b>
Grupo I - Atividades de Ensino, Orientação e Avaliação do Docente pelo Discente	Subgrupo I1 - Ensino
	Subgrupo I2 - Orientação
	Subgrupo I3 - Avaliação do Docente pelo Discente
Grupo II - Produção Intelectual	Não possui subgrupos
Grupo III - Atividades de Pesquisa, de Extensão, Administrativas e de Representação e Outras Atividades	Subgrupo III1 - Atividades de Pesquisa
	Subgrupo III2 - Atividades de Extensão
	Subgrupo III3 - Atividades de Administrativas e de Representação
	Subgrupo III4 - Outras Atividades

Fonte: Adaptado de UTFPR (2018)

Quadro 3 - Pontuação máxima de subgrupos existentes à classe Associado

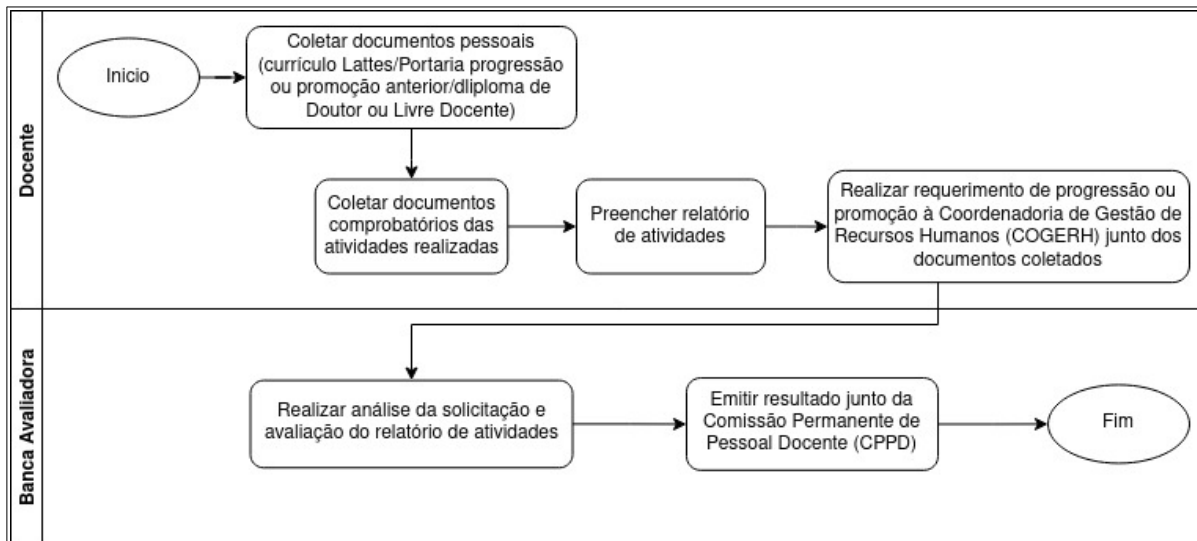
<b>Subgrupo</b>	<b>Pontuação máxima</b>
Subgrupo III1 - Atividades de Pesquisa	15 pontos
Subgrupo III2 - Atividades de Extensão	15 pontos
Subgrupo III3 - Atividades Administrativas e de Representação	10 pontos
Subgrupo III4 - Outras Atividades	10 pontos

Fonte: Adaptado de UTFPR (2018).

Para realizar o processo de progressão ou promoção à Classe Associado, o docente deve requisitar à Coordenadoria de Gestão de Recursos Humanos (COGERH), como estabelece o Art. 3º da DELIBERAÇÃO Nº. 28/18 (UTFPR, 2018). O fluxo do processo é descrito como apresenta a Figura 1. Tal requerimento deve estar acompanhando dos documentos estabelecidos na mesma deliberação (UTFPR, 2018):

- Relatório de atividades realizadas;
- Cópia dos documentos comprobatórios das atividades existentes no relatório de atividades;
- Cópia do currículo Lattes atualizado;
- Cópia da Portaria de promoção ou progressão anterior; e
- Cópia do diploma de Doutor ou Livre Docente para a promoção à Classe Associado.

Figura 1 - Fluxo do processo de progressão e promoção à Classe Associado



Fonte: Adaptado de UTFPR (2018)

De acordo com Figura 1, o primeiro passo ao docente interessado é coletar toda a documentação necessária, conforme estabelece a DELIBERAÇÃO Nº. 28/18. Após isso, realizar a contabilização das atividades realizadas em cada um dos itens da tabela de pontuação e preencher este dado no relatório de atividades. Assim, é

solicitado ao COGERH a progressão ou promoção. Tal solicitação será avaliada por uma banca examinadora, a qual emitirá o resultado junto da Comissão Permanente de Pessoal Docente (CPPD).

Portanto, como os documentos são parte fundamental do processo, deve ser realizada uma gestão adequada destes para que se alcance a aprovação no processo. Porém, o processo manual de gestão desses documentos apresenta algumas falhas.

Um dos problemas que pode ocorrer é no âmbito do armazenamento desses documentos. Cópias físicas podem ser perdidas ou extraviadas. Dessa forma, é interessante que se possa armazenar tais documentos de maneira segura.

Outra dificuldade é quando o professor deseja saber o seu progresso na carreira docente. Como a quantidade de documentos é elevada, realizar a contabilização manual das pontuações, além de trabalhoso, pode ser realizada uma contagem errada.

Ainda, outro ponto relacionado com a contabilização das pontuações é o preenchimento do relatório de atividades, pois caso o relatório seja preenchido com pontuações inconsistentes, o processo de progressão ou promoção do docente pode ser comprometido. Então, uma ferramenta que auxilie o processo de gestão desses documentos se mostra apropriada.

### 2.3 Gestão Eletrônica de Documentos

A gestão de documentos proporciona organização para que a busca destes e de dados, neles contidos, seja otimizado, promovendo um processo eficiente e rápido (CALDERON et al., 2004, p. 101).

O processo de gestão pode se tornar ainda mais fácil e eficiente caso seja utilizada uma ferramenta como auxílio. Tais ferramentas são denominadas Sistemas para Gestão Eletrônica de Documentos (GED), os quais “[...] concentram-se em facilitar a gestão de documentos pertinentes a uma particular empresa, projetos ou grupos de trabalho por meio de redes de computadores” (BJORK, 2002, p. 3, tradução livre).

O principal benefício dos sistemas eletrônicos de gestão é a eliminação do manuseio físico dos documentos. Para utilizá-los é necessário converter o documento físico para o formato eletrônico, o que é feito por meio de um dispositivo de captura como máquinas de fotocópia ou câmeras fotográficas (desde que esta forneça uma imagem com boa resolução) (SILVA, 2001, p. 12).

De posse do documento em formato digital, basta enviá-lo para a plataforma de GED, que atua como um repositório de documentos digitais, os quais podem ser armazenados em uma hierarquia de diretórios para melhor organização (KEEVA, 2020).

Tais ferramentas também permitem que ações sejam realizadas com os documentos, como, por exemplo visualização, *download*, e modificações. Tais ações variam de uma plataforma para outra e são definidas no projeto do sistema.

A lista abaixo aponta alguns dos benefícios obtidos pela utilização de sistemas GED em diversas áreas de aplicação:

- No escopo de projetos de Engenharia Civil, uma área que envolve a gestão de diversos documentos como planta baixa, projeto elétrico, projeto hidráulico, entre outros, uma gestão eficiente se faz necessária. Pois, como afirmam Jadid e Idrees (2005):

[...] na construção de grandes projetos de engenharia civil a gestão de documentos e a disponibilidade imediata de informações são essenciais para um projeto bem-sucedido (JADID; IDREES, 2005, p. 424, tradução livre).

- Em universidades, onde é alto o volume de documentos utilizados diariamente devido à variedade de atividades relacionadas ao ensino, pesquisa e extensão, bem como consequência das atividades de gestão administrativa da instituição, o uso de sistemas para auxiliar na gestão desses documentos pode beneficiar a instituição. Como explicam Costoiu et al. (2012),

Uma vez que um sistema de gestão de documentos é implementado em uma universidade, ocorrerá um aumento no desempenho de atividades da educação e da pesquisa [...] (COSTOIU et al., 2012, p. 1639, tradução livre).

Costoiu et al. (2012) explicam que a Universidade Politécnica de Bucareste utiliza um sistema para gestão eletrônica de documentos, o qual serve como um repositório de documentos para a organização, e no qual os funcionários administrativos e os pesquisadores podem acessar os documentos e realizar tarefas *online* (COSTOIU, 2012). Segundo os autores, tal sistema permite que a gerência da universidade:

“[...] encontre pontos que podem ser melhorados no trabalho e nos procedimentos da universidade definindo, para isso, metas e prazos para as tarefas de cada pessoa” (COSTOIU et al., 2012, p. 1644, tradução livre).

- Ferreira, Azevedo e Fagundes (2008) afirmam que um dos desafios da arquivística é organizar documentos corretamente. Grandes massas de documentos são utilizadas diariamente em instituições e para auxiliar nesse processo propuseram a criação de uma ferramenta para GED, chamada e-DOC. Segundo os autores, as vantagens da utilização de tal ferramenta são “[...] armazenar, controlar, compartilhar e recuperar documentos de forma fácil e segura, agregando valor ao negócio” (FERREIRA; AZEVEDO; FAGUNDES, 2008, p. 2), as quais auxiliam a contornar os desafios da arquivística.

A utilização de ferramentas para gestão eletrônica de documentos é realizada em diversas áreas, dada as vantagens que apresenta quando comparada com o processo de gestão manual. Ainda, os sistemas para GED utilizados em áreas distintas apresentam benefícios em comum, o que demonstra que a abordagem eletrônica é bastante robusta, independente do escopo aplicado.

Para a progressão e promoção na carreira docente não foi encontrado um sistema eletrônico que auxilie a gestão dos documentos utilizados durante o processo. Tal ferramenta facilitaria o processo de gestão dos docentes, por exemplo, quando se deseja saber a pontuação obtida em função das atividades realizadas até

o momento bem como a faltante para alcançar a aprovação na avaliação de desempenho.

Outra vantagem é na localização de documentos, pois uma atividade pode ter sido realizada há muito tempo e o docente ter esquecido o local em que arquivou o documento comprobatório.

Portanto, esses pontos motivaram o desenvolvimento de uma ferramenta de GED do processo de progressão e promoção funcional dos docentes da UTFPR. Tal ferramenta deve suportar o registro de atividades realizadas e permitir anexar os documentos comprobatórios a esse registro, como também possuir a capacidade de processá-los para posterior preenchimento da tabela de pontuações.

Ainda, seria necessário que pudessem ser consultados indicadores, com o objetivo de informar o Docente sobre seu progresso. Também é importante que o sistema desenvolvido seja acessado remotamente, portanto um sistema *web* apresenta-se como a melhor opção para atender às necessidades elencadas.

## 2.4 *Web*

*Web* é a nomenclatura como a Rede Mundial de Computadores (do inglês *World Wide Web*, ou *WWW*) ficou conhecida, a qual foi desenvolvida pelo pesquisador Tim Berners-Lee enquanto trabalhava na Organização Europeia de Pesquisas Nucleares (CERN) (W3, 2020a).

A motivação por trás da criação da *web* foi a necessidade de compartilhar informações de pesquisas científicas entre os computadores do centro de pesquisa, pois até aquele momento não era possível acessá-las remotamente, somente no computador em que a informação se encontrava armazenada (W3, 2020a; W3, 2020b).

Para possibilitar o acesso remoto, Tim Berners-Lee desenvolveu uma solução integrando duas tecnologias existentes: o *hypertext* e a Internet (W3, 2020a). Segundo Mozilla (2020b, tradução livre) “A Internet é basicamente uma grande rede de computadores, os quais se comunicam uns com os outros”. A Internet funciona sobre dois pilares: o Protocolo de Controle de Transmissão/Protocolo de Internet (do



inglês *Transmission Control Protocol/Internet Protocol*, ou TCP/IP) e o Sistema de Nomes de Domínios (do inglês *Domain Name System*, ou DNS) (W3, 2020a).

“O Protocolo de Controle de Transmissão/Protocolo de Internet (TCP/IP) é um conjunto protocolos das camadas de transporte e rede utilizado por máquinas que desejam se comunicar em uma rede” (NTU, 2020, tradução livre). Já o DNS possui a função de traduzir um endereço IP para um nome que é amigável à leitura humana, uma vez que um endereço IP é um conjunto de números decimais e pode ser difícil memorizá-lo (NTU, 2020; MOZILLA, 2020b). Endereço IP nada mais é que o identificador de um computador em uma rede (MOZILLA, 2020b).

*Hypertext*, ou hipertexto, é um termo cunhado por Ted Nelson, e como descreve a W3, “[...] é um texto que contém uma ligação para outro texto” (W3, 2021, tradução livre). Ou seja, ela traz a ideia de uma leitura não-linear, em que, dentro de um texto é possível acessar um outro texto através de um *link*.

No momento da criação da *web* várias formas de hipertexto já existiam, porém funcionavam apenas em textos armazenados do mesmo computador (W3, 2020a). Assim, na *web*, um *link* contém o endereço de uma informação, tal que esta informação pode estar localizada tanto no próprio computador como em outro computador que esteja conectado à Internet.

A arquitetura da *web* é cliente-servidor, isto é, “[...] um cliente, usando um navegador, pode acessar um serviço hospedado em um servidor” (FOROUZAN, 2008, p. 851). Para realizar o acesso, o cliente utiliza o que é chamado de Localizador de Recursos Uniforme (do inglês *Uniform Resource Locator*, ou URL), o qual serve para localizar um recurso na *web*. Uma URL é estruturada conforme apresenta a Figura 2:

Figura 2 - Formato de uma URL

**protocolo://nome-do-servidor[:porta] [/path-no-servidor] [?argumentos]**

Fonte: Adaptado de COULOURIS (2013, p.29)

Isto é, uma URL é construída pela combinação de elementos da Figura 2, tal que:

- **protocolo:** define o protocolo utilizado na comunicação entre o cliente e o servidor;
- **nome-do-servidor:** pode ser tanto um nome a ser resolvido por um DNS como um endereço IP;
- **:porta:** é a porta do servidor que o navegador (*browser*) deverá se conectar;
- **path-no-servidor:** é o caminho do diretório onde está o recurso e ser acessado;
- **?argumentos:** são parâmetros adicionais que podem ser passados para o servidor, o que, no desenvolvimento web, é chamado de *query string*.

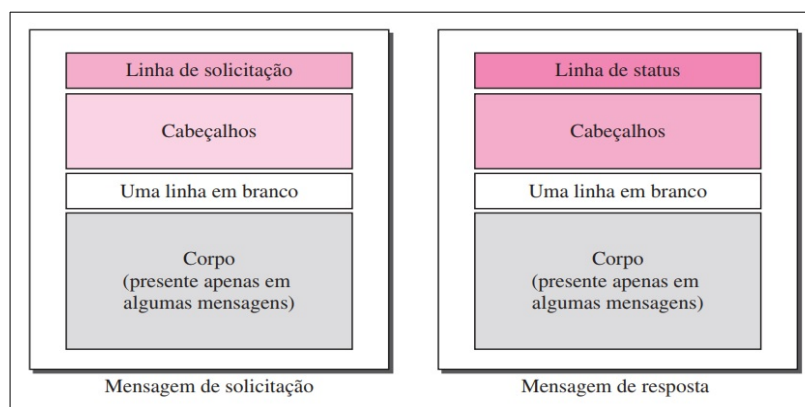
Mesmo sendo possível acessar recursos na *web* utilizando protocolos diferentes do Protocolo de Transferência de Hipertexto (do inglês, *Hypertext Transfer Protocol*, ou HTTP), como o Protocolo de Transferência de Arquivos (do inglês *File Transfer Protocol*, ou FTP) por exemplo, a *World Wide Web Consortium* (W3C), organização que define os padrões da *web* e tem como diretor Tim Berners-Lee, padronizou que o acesso a servidores na *web* é feito utilizando o protocolo HTTP (W3, 2020c).

O HTTP é um protocolo de aplicações cliente-servidor baseado no protocolo de transporte TCP (W3, 2020c). Uma vez que o HTTP é um protocolo para aplicações cliente-servidor, “existem dois tipos de mensagens HTTP, as mensagens de requisição e as mensagens de resposta” (MOZILLA, 2020a, tradução livre). Para acessar um recurso de um servidor *web*, o cliente envia uma mensagem de requisição (HTTP *request*) e aguarda a mensagem de resposta (HTTP *response*).

As mensagens HTTP são compostas pela linha de solicitação/linha de status, pelo cabeçalho (*header*) e pelo corpo (*body*) (FOROUZAN, 2020, p. 861). A Figura 3 mostra como as mensagens são organizadas. A linha de solicitação contém o método da solicitação, a URL do recurso que se deseja acessar e a versão HTTP utilizada na requisição (FOROUZAN, 2008, p. 862).

O método da requisição é uma forma de comunicar que tipo de ação a requisição solicitará e utiliza um verbo para isso (MOZILLA, 2020d). O Quadro 4 apresenta alguns métodos, ou verbos, HTTP e o tipo de ação que ele representa.

Figura 3 - Organização das mensagens HTTP



Fonte: FOROUZAN (2008, p. 862)

Quadro 4 - Métodos HTTP

Método	Ação
GET	Solicita um documento ao servidor
HEAD	Solicita informações sobre um recurso, mas não o recurso em si
POST	Envia informações do cliente para o servidor
PUT	Substituí informações no servidor
DELETE	Remove uma informação do servidor
TRACE	Ecoa uma requisição que chega; utilizado para testes
CONNECT	Reservado
OPTION	Solicita detalhamento sobre opções disponíveis

Fonte: Adaptado de FOROUZAN (2008, p. 863) e MOZILLA (2020d)

A linha de *status*, na mensagem HTTP, contém a versão HTTP utilizada na resposta, o código de *status* e uma frase de *status* (FOROUZAN, 2008, p. 862). O código de *status* é um número, contém três dígitos e representa o resultado a requisição (NTU, 2020).

Os códigos de *status* são divididos em 5 categorias. A frase de *status* é apenas uma descrição do significado do código de *status* para facilitar o entendimento sobre o que o código representa (FOROUZAN, 2008, p. 863). O Quadro 5 mostra a organização dos principais códigos de *status* e suas categorias, das mensagens de *status* e o que significam.

Quadro 5 - Códigos de status

<b>Código</b>	<b>Frase de Status</b>	<b>Descrição</b>
<b>Informativo</b>		
<b>100</b>	Continue	A parte inicial da requisição foi recebida e o cliente, se desejar, pode prosseguir
<b>101</b>	Switching	O servidor está atendendo a requisição de um cliente para alterar os protocolos definidos no cabeçalho de atualização
<b>Sucesso</b>		
<b>200</b>	Ok	A requisição foi bem-sucedida
<b>201</b>	Created	Um novo recurso foi criado
<b>202</b>	Accepted	A requisição foi aceita, mas não pode ser executada imediatamente
<b>204</b>	No content	Conteúdo inexistente no corpo da requisição
<b>Redirecionamento</b>		
<b>301</b>	Moved permanently	A URL solicitada não está mais em uso pelo servidor
<b>302</b>	Moved temporarily	A URL solicitada foi temporariamente movida
<b>304</b>	Not modified	O recurso não foi modificado
<b>Erros no Cliente</b>		
<b>400</b>	Bad request	Erro de sintaxe na requisição
<b>401</b>	Unauthorized	A requisição não tem autorização suficiente para ser executada
<b>403</b>	Forbidden	Serviço negado. O cliente não tem permissão para acessá-lo
<b>404</b>	Not found	O recurso não foi encontrado
<b>405</b>	Method not allowed	O método da requisição não é permitido pelo servidor
<b>406</b>	Not acceptable	O formato solicitado não é aceitável
<b>Erros no Servidor</b>		
<b>500</b>	Internal server error	Há um erro, como um travamento do sistema, por exemplo, no servidor

<b>Código</b>	<b>Frase de Status</b>	<b>Descrição</b>
<b>501</b>	Not implemented	A ação requisitada não pode ser executada
<b>503</b>	Service unavailable	O serviço está temporariamente indisponível mas poderá ser requisitado no futuro

Fonte: Adaptado de FOROUZAN (2008, p. 863); MOZILLA (2020e)

Os cabeçalhos da requisição são uma forma de passar parâmetros adicionais para o servidor com a informação de como deve ser realizada a ação solicitada enquanto que os cabeçalhos da resposta são utilizados para comunicar ao cliente as configurações do servidor e informações sobre o que foi solicitado (FOROUZAN, 2008, p. 865).

Por sua vez, o corpo de uma mensagem HTTP “Pode estar presente em uma mensagem de solicitação ou de resposta. Normalmente contém o documento a ser transmitido ou recebido” (FOROUZAN, 2008, p. 866).

Quando é realizado o acesso a um documento na *web*, este é denominado uma página *web*. As páginas *web* são utilizadas pelo navegador (*browser*) do cliente para visualizar as informações contidas no documento e são construídas com a Linguagem de Marcação de Hipertexto, ou *Hypertext Markup Language* (HTML) (W3SCHOOLS, 2021c).

A HTML define a estrutura de um documento e organiza o texto contido nele. Esta linguagem funciona em conjunto com os *browsers*, que leem este documento, e, a partir da organização definida pela HTML, estruturam o conteúdo que será apresentado ao usuário (MOZILLA, 2020c).

A estruturação do conteúdo em uma página HTML é realizada por meio das chamadas *tags*. Uma *tag* é composta pelos sinais < e >, os quais envolvem uma palavra-chave que a identifica. As *tags* normalmente são utilizadas em pares, sendo uma de abertura e outra de fechamento e envolvem o texto a ser organizado, compondo o que é chamado de elemento (W3SCHOOLS, 2021c). A Figura 4 mostra um exemplo da *tag* que define um parágrafo.

Figura 4 - Exemplo de tag HTML

```
<p>Conteúdo</p>
```

Fonte: A autoria própria (2021)

As *tags* podem ainda receber atributos, os quais são pares chave-valor que alteram o comportamento padrão da *tag* ou passam informações adicionais (W3SCHOOLS, 2021c).

A Figura 5 apresenta um exemplo de *tag* com atributo. O atributo da *tag* na Figura 5 é o **href** que recebe uma URL como valor. A *tag* da Figura 5 define um *link* (ou *hyperlink*), o qual depois de ser *clicado*<sup>3</sup> redireciona o usuário para o documento localizado na URL contida no atributo **href** (W3SCHOOLS, 2021c).

Figura 5 - Exemplo de tag com atributo

```
<a href="URL">Vá para URL</a>
```

Fonte: A autoria própria (2021)

Um documento HTML é estruturado como mostra a Figura 6, tal que a *tag* `<!DOCTYPE>` deve ser definida na primeira linha do documento e serve para informar ao navegador qual o tipo do documento e com qual versão da linguagem HTML este foi desenvolvido (W3SCHOOLS, 2021c). A *tag* `<HTML>` é a raiz do documento e contém os elementos `<HEAD>` e `<BODY>`.

Dentro da *tag* `<HEAD>` são definidas informações a respeito do documento, as quais são chamadas de metadados, ou *metadata* (W3SCHOOLS, 2021c). Todo o conteúdo que será renderizado<sup>4</sup> na tela do dispositivo cliente deve ser definido dentro da *tag* `<BODY>` (W3SCHOOLS, 2021c).

Uma vez que o HTML é utilizado para estruturar uma página *web*, existe um recurso para definir o estilo de apresentação do conteúdo que será renderizado, denominado Folhas de Estilo em Cascata, ou *Cascading Style Sheets* (CSS). O CSS permite definir fatores como tamanho e estilo de fonte, cor do texto, posicionamento,

---

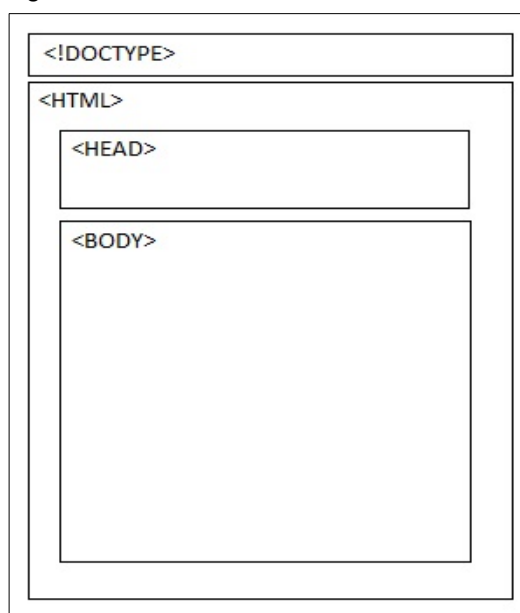
3 Ato de apertar um dos botões do *mouse* enquanto o cursor está sobre o elemento em questão

4 Renderização é a apresentação do conteúdo definido pelo HTML e CSS no navegador

entre outros (W3SCHOOLS, 2021a). Existem três formas possíveis de usar o CSS: *inline*, CSS interno e CSS externo (W3SCHOOLS, 2021b).

Na forma *inline*, o CSS é definido como um atributo da *tag* em que este atuará. Na forma interna, as definições são feitas dentro da *tag* <style>, a qual é definida na seção <HEAD> do documento. Por fim, na forma externa, as definições são realizadas em um documento separado do HTML, tal que este é associado por meio da *tag* <LINK>, a qual recebe como atributo a localização do documento CSS e é definida também no *head* do HTML (W3SCHOOLS, 2021b). A Figura 7 mostra a sintaxe utilizada no CSS interno e externo.

Figura 6 - Estrutura de um documento HTML



Fonte: Adaptado de W3SCHOOLS (2021c)

Na Figura 7 o seletor define o elemento HTML em que a regra atuará. O conjunto propriedade e valor formam o que é chamado de declaração. No CSS *inline* não existe o seletor, uma vez que a declaração é feita como o valor do atributo *style* da *tag* (W3SCHOOLS, 2021b).

Figura 7 - Sintaxe do CSS

```
seletor {  
    propriedade: valor;  
}
```

Fonte: Autoria própria (2021)

O conteúdo das páginas desenvolvidas com apenas HTML não pode ser alterado, isto é, ele é estático, logo, estas são chamadas de páginas estáticas (MOZILLA, 2021e). Portanto, foram desenvolvidas linguagens de *scripting*, as quais atuam nas páginas HTML de forma a personalizar informações que dependem de um contexto, como, por exemplo, nome do usuário ou uma lista de itens. Estas páginas são chamadas de dinâmicas (MOZILLA, 2021e).

Muitas ações realizadas por *softwares* se repetem em diferentes escopos, logo, o mesmo código é desenvolvido em diferentes aplicações. Uma consequência disto é que o desenvolvedor reescreverá o mesmo código toda vez que uma aplicação for desenvolvida.

Para evitar que tarefas repetidas sejam feitas pelo desenvolvedor e assim facilitar a construção de sistemas, são utilizados *frameworks*. Segundo Castro (1999, p. 13) “[...] um framework é um conjunto de classes cooperantes que compõem um projeto reutilizável para uma classe específica de *software*”. Isto é, eles fornecem a estrutura que fundamenta o funcionamento de um *software* de determinado tipo. Os *frameworks* para aplicações do tipo *web* “fornecem as ferramentas e bibliotecas que facilitam o desenvolvimento de funcionalidades comuns na *web*, como roteamento de URL’s, interação com banco de dados [...]” (MOZILLA, 2021d, tradução livre).

## 2.5 Node e Express

A linguagem JavaScript é utilizada nas páginas *web* com o propósito de tornar seu conteúdo dinâmico. Para isto, o navegador possui um interpretador da linguagem, o qual executa os comandos definidos pelo programa (MOZILLA, 2021e).



Para possibilitar o uso do JavaScript fora de páginas *web*, foi desenvolvido o Node.JS, o qual funciona como um ambiente de execução (NODE, 2021). Ele permite criar aplicações com JavaScript em ambientes diferentes do navegador de Internet.

Para gerenciar os pacotes utilizados pela aplicação desenvolvida, o Node conta com um gerenciador de pacotes (*Node Package Manager*, ou NPM) (MOZILLA, 2021b). Tais pacotes contém funções que implementam uma operação ou funcionalidade específica, ou, como descreve W3Schools (2021d, tradução livre), “um pacote no Node.js contém todos os arquivos que um módulo necessita”. Assim, as funcionalidades de um *software* desenvolvido utilizando o Node.js podem ser organizadas em pacotes, ou módulos, e ser reutilizadas em diferentes partes do código.

Portanto, o NPM facilita a implementação do sistema já que fornece soluções prontas para problemas já enfrentados por outros desenvolvedores, bastando o desenvolvedor incluir os pacotes desejados no seu projeto.

Um dos pacotes disponíveis no NPM é o *Express*, o qual é um *framework* para desenvolvimento de servidores de aplicações *Web*. O *express* gerencia requisições HTTP utilizando um sistema de rotas, o qual determina a ação que deve ser realizada com base no verbo HTTP e na URL da requisição (MOZILLA, 2021b). Além disso, o *express* pode ser usado em conjunto de outros pacotes para adicionar funcionalidades à aplicação, como, por exemplo, conexão com banco de dados<sup>5</sup>, envio de arquivos para o servidor<sup>6</sup> e até conexão com planilhas no formato *xlsx*<sup>7</sup>.

Também é possível utilizar pacotes para renderizar páginas no servidor, mas é mais comum deixar esta função para *frameworks* especializados em desenvolvimento do lado cliente, como é o caso do *Vue.js*.

### 2.5.1 Laravel

Existem outros *frameworks* para desenvolvimento do módulo servidor de aplicações *web*, de forma que cada um tem suas vantagens e desvantagens. O

---

5 <https://www.npmjs.com/package/pg>

6 <https://www.npmjs.com/package/multer>

7 <https://www.npmjs.com/package/exceljs>

Laravel é um *framework* que utiliza da linguagem Pré-processador de Hipertexto (do inglês *Hypertext Preprocessor*, ou PHP) (LARAVEL, 2021).

O Laravel possui a vantagem de ter integrado uma vasta quantidade de recursos para desenvolvimento, como filas, motor para renderização de páginas no servidor, módulo de e-mail, entre outros, de forma que nada precisa ser adicionado ao projeto (LARAVEL, 2021). Tal vantagem custa um preço: estes módulos ocuparão espaço em disco, mesmo que não sejam utilizados.

Existe também uma desvantagem para o desenvolvedor, pois como o Laravel utiliza o PHP, será necessário utilizar duas linguagens de programação: o PHP para o servidor e o JavaScript para o cliente. Assim, a velocidade de desenvolvimento do projeto pode ser impactada.

## 2.6 Vue.js e Quasar *Framework*

Como descreve Vue.js (2021b), “Vue é um *framework* progressivo para construir interfaces de usuário”. Tal *framework* é denominado progressivo pois pode ser progressivamente adicionado em uma aplicação, isto é, aos poucos. Assim, tal aplicação não necessita ser totalmente construída com ele.

O Vue.js dispõe de um mecanismo de renderização declarativa que emprega colchetes duplos envolvendo uma variável para exibir seu valor de maneira dinâmica na página (VUEJS, 2021b). Essa sintaxe especial é ilustrada no código-fonte da Figura 8, no qual as variáveis nome e idade são declaradas nas linhas 19 e 20 e exibidas utilizando tal sintaxe como itens de uma lista (linhas 11 e 12) contida na página.

Também podem ser vistos outros detalhes de funcionamento do Vue.js na Figura 8 da página 35. Um deles é a forma como o *framework* é iniciado: utilizando a palavra-chave *new* (linha 16) para criar um objeto do tipo Vue, isto é, criar uma instância do *framework*. Outro, é a forma como ele atua na página *web*: o objeto passado como parâmetro na sua instanciação possui um atributo chamado ‘el’ (linha 17), o qual é o id do elemento em que o *framework* atuará. Ainda, na linha 18, a instância do Vue possui um objeto chamado *data*, o qual contém dados utilizados na página.

Também é possível definir outros objetos, de forma que cada um possui um objetivo específico. Um objeto muito comum é o *methods*, no qual são implementadas funções, de forma a executar uma ação na ocorrência de um determinado evento (VUEJS, 2021b).

Figura 8 - Renderização declarativa do Vue.js

```

1  <!DOCTYPE html>
2  <head>
3    <script
4      | src="https://cdn.jsdelivr.net/npm/vue@2.6.12/dist/vue.js"
5    >
6  </script>
7 </head>
8 <body>
9   <div id="app">
10    <ul>
11      <li>{{ nome }}</li>
12      <li>{{ idade }}</li>
13    </ul>
14    <div>
15      <script>
16        new Vue({
17          el: '#app',
18          data: {
19            nome: 'Fulano',
20            idade: 25
21          }
22        })
23      </script>
24    </div>
25 </body>
</html>

```

→

- Fulano
- 25

Fonte: Autoria própria (2021)

O Vue.js também possui um mecanismo de diretivas, no qual é possível passar atributos especiais para os elementos HTML, os quais possuem como prefixo os algarismos “v-” e implementam alguma ação a ser realizada com o elemento (VUEJS, 2021b). Tais ações incluem condicionais, laços de repetição, ligação com variáveis definidas na aplicação e até definir a execução de uma função na ocorrência de um determinado evento.

O Quadro 6 apresenta algumas diretivas do Vue.js.

Quadro 6 - Exemplos de diretivas do framework Vue.js

Diretiva	Descrição
v-for	Aceita um objeto iterável como

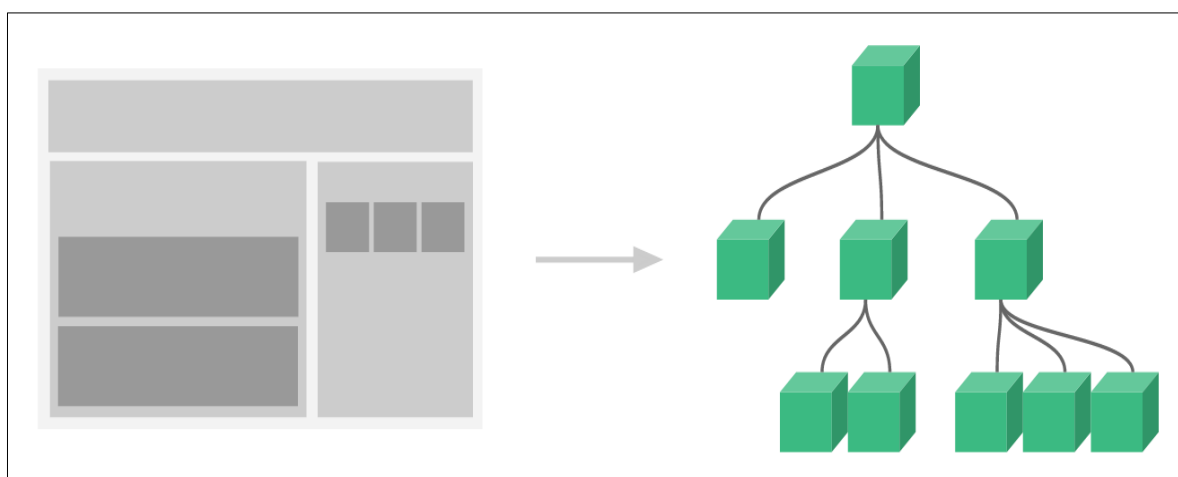
Diretiva	Descrição
	argumento e executa um <i>loop</i> , de forma a renderizar um elemento HTML várias vezes com conteúdo dinâmico
v-if	Realiza ou não a renderização de um elemento HTML com base em uma análise condicional
v-on	Associa ao elemento uma função, a qual é executada na ocorrência de um evento definido
v-bind	Vincula uma variável ao elemento HTML

Fonte: Adaptado de VUEJS (2021b)

Outro mecanismo característico do *framework* é o sistema de componentes. Tal sistema, como explica Vue (2021b, tradução livre), “[...] é uma abstração que nos permite construir aplicações de larga escala compostas por pequenos componentes independentes e, frequentemente, reusáveis”. Cada componente possui seus dados, métodos, HTML e CSS, como também pode ser composto por outros componentes (VUE, 2021a).

A Figura 9 apresenta como interfaces de usuário e seu *layout* são compostas usando este sistema de componentes.

Figura 9 - Componentes do Vue.js



Fonte: VUEJS (2021a)

A criação de um componente é realizada dentro da *tag script* em um documento HTML, da mesma forma que a inicialização do *framework* apresentada na Figura 8 da página 35. Tal criação é apresentada na Figura 10.

Como pode ser visto na Figura 10, o primeiro argumento da função que cria um componente é um texto, o qual será o nome da *tag* HTML para utilizar tal componente. O segundo argumento é um objeto, no qual são definidos os seus dados no campo *data* e o seu código HTML no campo *template*. Utilizando do conceito de componentes, o código fonte do exemplo da Figura 8 (página 35) fica como mostra a Figura 11 (página 38).

Figura 10 - Definição de um componente

```
1 Vue.component('button-counter', {
2   data: function() {
3     return {
4       count: 0
5     }
6   },
7   template: `<button v-on:click="count++">
8     You clicked me {{ count }} times.
9     </button>`
10 })
```

Fonte: Adaptado de VUEJS (2021a)

Entretanto, quando a aplicação desenvolvida possui altos níveis de complexidade e vários componentes, defini-los dessa forma pode tornar o código-fonte de difícil manutenção.

Portanto, para manter o código-fonte organizado, bem como facilitar o desenvolvimento dos componentes, o *framework* Vue.js fornece um mecanismo chamado de Componentes de Arquivo Único (do inglês *Single File Components*, ou SFC) (VUEJS, 2021c).

Estes componentes são definidos como um arquivo separado, com a extensão “.vue”, e utilizados como uma *tag* HTML no arquivo principal do código-fonte. Tais componentes possuem três seções: a seção *template*, a qual contém o HTML; a seção *script*, onde são definidos os dados, métodos e objetos do

*framework* necessários para o componente; e uma seção chamada *style*, no qual podem ser definidas regras de CSS (VUEJS, 2021c). Os exemplos das Figuras 8 e 11, desenvolvidos como um SFC, ficam como apresenta a Figura 12 (página 39).

Figura 11 - Exemplo de código do Vue.js utilizando componente

```
1 <!DOCTYPE html>
2 <head>
3   <script src="https://cdn.jsdelivr.net/npm/vue@2.6.12/dist/vue.js"></script>
4 </head>
5 <body>
6   <div id="app">
7     <my-list></my-list>
8   </div>
9 </body>
10 <script>
11   Vue.component('my-list', {
12     data: function() {
13       return {
14         nome: 'Fulano',
15         idade: 25
16       }
17     },
18     template: `<ul>
19       <li>{{ nome }}</li>
20       <li>{{ idade }}</li>
21     </ul>`
22   })
23
24   new Vue({
25     el: '#app',
26   })
27 </script>
28 </html>
```

Fonte: Autoria própria (2021)

Mesmo que estes componentes são em muitas das vezes reutilizáveis, implementá-los manualmente pode se tornar bastante trabalhoso em grandes aplicações. Particularmente no aspecto da interface de usuário (do inglês, *User Interface*, ou UI), visto que aplicações modernas são compostas por uma variedade de efeitos visuais e componentes complexos.

Portanto, para facilitar e agilizar o processo de desenvolvimento, foram criados *frameworks* que fornecem componentes de UI, os quais possuem a estrutura pronta de HTML e CSS, deixando para o desenvolvedor somente o trabalho de implementar a lógica do fluxo da informação. Um exemplo deste tipo de *framework* é o Quasar (2021), o qual foi utilizado neste trabalho.

Figura 12 - Exemplo de Single File Component

```
1  <template>
2  |   <div>
3  |   |   <ul>
4  |   |   |   <li>{{ nome }}</li>
5  |   |   |   <li>{{ idade }}</li>
6  |   |   </ul>
7  |   </div>
8  </template>
9
10 <script>
11 |   export default {
12 |   |   data: function() {
13 |   |   |   return {
14 |   |   |   |   nome: 'Fulano',
15 |   |   |   |   idade: 25
16 |   |   |   }
17 |   |   }
18 |   }
19 </script>
20
21 <style>
22 </style>
```

Fonte: Autoria própria (2021)

O *Quasar Framework* utiliza o *Vue.js* como motor de funcionamento e conta com uma biblioteca de componentes de interface, como botões, tabelas, barras de navegação, entre outros, prontos para serem usados (QUASAR, 2021). Além disso, o *framework* disponibiliza comandos pré-definidos que podem ser executados por linha de comando (ou CLI, do inglês *Command Line Interface*) para gerar a estrutura de diretórios do código-fonte, recarregar automaticamente a página quando mudanças são efetuadas no código-fonte do projeto (também conhecido como *hot module reload*), entre outros, além de possibilitar o desenvolvimento da aplicação para diferentes plataformas, tais como: *web*, *mobile* e *desktop*, utilizando a mesma base de código (QUASAR, 2021).

Os dados inseridos, ou processados, tanto pelo módulo cliente como pelo servidor, precisam de um meio para serem armazenados na aplicação, de forma que, se algum dos módulos for reiniciado, tais dados não serão perdidos. Para tal persistência são utilizados os Banco de Dados.

### 2.6.1 Angular

O Angular é um *framework* para desenvolvimento no lado cliente de aplicações web, feito pela Google, e que usa o TypeScript como linguagem de programação (ANGULAR, 2021). Este *framework* também utiliza o conceito de componentes para estruturação das páginas, bem como possui bibliotecas integradas que implementam funcionalidades comuns em páginas *web*, para que o desenvolvedor não precise adicionar novos pacotes (ANGULAR, 2021).

Assim, o Angular se mostra como uma opção bastante robusta, que fornece um ambiente de desenvolvimento com várias bibliotecas pré-instaladas. Porém, ele também apresenta a desvantagem de ocupar espaço em disco, pois o desenvolvedor pode não utilizar todas as bibliotecas incluídas no *framework*. O Angular também exige que o desenvolvedor tenha conhecimento do TypeScript, o que pode ser uma desvantagem.

Outro ponto é que o Angular não dispõe de uma biblioteca de componentes de interface incorporada, como o Quasar. Dessa forma, o desenvolvedor precisará integrar o Angular com um *framework* para CSS ou desenvolver o estilo dos componentes manualmente.

### 2.6.2 React

O React foi desenvolvido pelo Facebook e é “[...] uma biblioteca JavaScript para construir interfaces de usuário.” (REACT, 2021). Este *framework* também propõe a construção de páginas utilizando a composição por componentes (REACT, 2021). Assim, o desenvolvedor consegue escrever o código-fonte somente uma vez e reutilizá-lo em diferentes partes da aplicação.

O React tem um perfil minimalista, isto é, fornece somente facilidades para a manipulação do HTML e dos dados nele utilizados. Dessa forma, o desenvolvedor precisa incorporar ao projeto novos pacotes para adicionar mecanismos de roteamento das páginas, comunicação com servidor, como também para utilizar componentes de UI prontos para serem utilizados.



## 2.7 Banco de Dados

Existem diferentes tipos de Banco de Dados (BD), uma vez que diferentes autores possuem visões diferentes sobre um determinado problema, como também problemas diferentes podem exigir soluções diferentes (ORACLE, 2021a). Exemplos destes tipos são: orientado a objetos, relacional e, uma forma mais recente, o banco de dados NoSQL. O modelo de banco de dados relacional é o mais popular, pois, como descreve Oracle o modelo relacional é “[...] uma forma direta e intuitiva de representar dados em tabelas” (ORACLE, 2021b, tradução livre).

Um banco de dados relacional possui uma estrutura de relações, ou tabelas, as quais contém atributos para definir os dados daquela relação (ORACLE, 2021b). Assim pode ser definida, por exemplo, uma relação chamada pessoa, de forma que os atributos dessa relação são nome, rg e data de nascimento.

Cada registro de pessoa nesta relação é uma linha da tabela. Para diferenciar cada uma dessas linhas deve ser definido um atributo especial na relação, chamado de chave primária, o qual é frequentemente definido como id (ORACLE, 2021b).

As relações de um banco de dados podem relacionar entre si através de relacionamentos (ORACLE, 2021b). Para definir relacionamentos são utilizados atributos especiais nas relações, chamados de chave estrangeira, os quais referenciam a chave primária da outra relação (IBM, 2021). Usando o exemplo da relação pessoa, uma pessoa pode ter um cargo em uma organização, por exemplo. Então, a relação pessoa e a relação cargo se relacionam entre si. Existem três formas que estes relacionamentos podem acontecer: 1 para 1, 1 para N e N para N (IBM, 2021).

No relacionamento 1 para 1, cada relação pode somente se relacionar com uma instância da outra parte do relacionamento (IBM, 2021). Portanto, nesta forma uma pessoa poderia ter somente um cargo e um cargo poderia ter somente uma pessoa atribuído a ele.

Para o relacionamento 1 para N, um lado da relação pode ser associado a somente uma instância do outro lado, porém, o outro lado pode ser associado a uma ou várias instâncias do primeiro (IBM, 2021). Assim, neste caso, uma pessoa pode ser associada a somente um cargo, mas o cargo pode ter várias pessoas

associadas a ele. Então, a cardinalidade deste relacionamento é 1 para a relação pessoa e N para a relação cargo.

Já no relacionamento N para N, ambos os lados da relação podem ser associados a uma ou várias instâncias do outro lado da relação (IBM, 2021). Neste caso é necessário inserir uma tabela especial entre as relações, a qual é chamada de tabela associativa (IBM, 2021). Assim, uma pessoa poderia ser associada a vários cargos, bem como um cargo pode ter várias pessoas associadas a ele.

Toda esta estrutura de relações, relacionamentos e seus dados não podem ser acessados ou utilizados diretamente, de forma que para realizar tais tarefas deve ser utilizado um Sistema Gerenciador de Banco de Dados (SGBD) (ORACLE, 2021a). Os SGBDs utilizam da Linguagem de Consulta Estruturada (do inglês *Structured Query Language*, ou SQL) para realizar a interação com o banco de dados (ORACLE, 2021a).

A linguagem SQL é “[...] usada por quase todos os bancos de dados relacionais para consultar, manipular e definir dados, como também promover controle de acesso” (ORACLE, 2021a, tradução livre). Então, a criação das relações e relacionamentos, inserção, busca ou remoção dos dados, bem como definições de usuários e seu acesso ao banco de dados são feitos utilizando-se da linguagem SQL.

### 3 MÉTODO

Neste capítulo será apresentada a estratégia utilizada para o desenvolvimento, como também as ferramentas escolhidas. Na Seção 3.1 é apresentada a estratégia adotada para o processo de desenvolvimento. A Seção 3.2 fala sobre as ferramentas escolhidas para a implementação do *software*. A Seção 3.3 discute a infraestrutura e a forma como o *software* pode ser implantado.

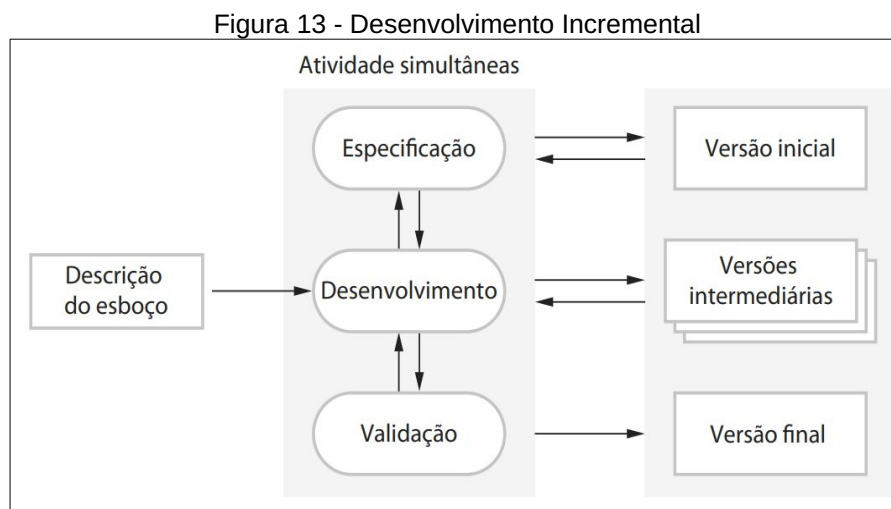
#### 3.1 Estratégia de desenvolvimento

A escolha de uma estratégia contribui com o estabelecimento de passos bem definidos para alcançar um objetivo. Para implementar o sistema proposto a estratégia utilizada foi o Desenvolvimento Incremental (SOMMERVILLE, 2011, p. 20), o qual é definido como uma estratégia que “[...] intercala as atividades de especificação, desenvolvimento e validação. O sistema é desenvolvido como uma série de versões (incrementos), de maneira que cada versão adiciona funcionalidade à anterior” (SOMMERVILLE, 2011, p. 20). Um benefício desse modelo é que o usuário não precisa aguardar até a versão final do produto para utilizá-lo. Segundo Sommerville:

“O desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido. Atividades de especificação, desenvolvimento e validação são intercaladas, e não separadas [...]” (SOMMERVILLE, 2011, p. 21).

A Figura 13 apresenta um esquema descrevendo funcionamento do desenvolvimento incremental. O processo começa na descrição do esboço, estágio em que é feita uma análise inicial dos requisitos bem como a criação de protótipos. Então, com estes dados em mãos, se inicia o desenvolvimento do *software*, produzindo sua versão inicial. Esta versão passa por processos de validação com o

usuário, os quais geram insumo para refinar as especificações e assim aprimorar os requisitos e necessidades do sistema. A partir disso, novas implementações são realizadas gerando versões intermediárias do *software*. Este ciclo é repetido até a obtenção de um *software* que atenda às expectativas do usuário, sendo este a versão final.



Fonte: SOMMERVILLE (2011, p. 22)

Esta estratégia possibilita obter a opinião dos usuários sobre o produto durante o processo de desenvolvimento. Isto representa um grande benefício uma vez que permite ao desenvolvedor entender o que o usuário espera do sistema, e, a partir desse entendimento, refinar o *software* a fim de atender os propósitos do projeto.

Além do benefício no aspecto do produto desenvolvido, esta estratégia também promove vantagem no custo de desenvolvimento. Como afirma Sommerville (2011, p. 22), “[...] é mais barato e mais fácil fazer mudanças no software durante seu desenvolvimento”. Logo, o uso de recursos importantes, como o tempo, por exemplo, são otimizados.

## 3.2 Ferramentas escolhidas

Nesta seção serão apresentadas as ferramentas utilizadas no desenvolvimento do presente projeto, bem como a implantação da ferramenta em ambiente de produção. A Seção 3.2.1 mostra o *framework* Express, e alguns de seus benefícios. A Seção 3.2.2 apresenta o *framework* Quasar e as facilidades proporcionadas por ele. A Seção 3.2.3 aponta o Sistema Gerenciador de Banco de Dados escolhido. A Seção 3.2.4 fala sobre o Mapeamento Objeto-Relacional e qual a ferramenta foi escolhida para este trabalho.

### 3.2.1 Express

Embora existam diversos *frameworks* para desenvolvimento *web* que diferem conforme a linguagem de programação e forma de organização do código-fonte, nesse projeto o desenvolvimento do servidor da aplicação foi realizado com o *framework* Express<sup>8</sup>, o qual utiliza a linguagem de programação JavaScript. Essa decisão foi influenciada por dois fatores: utilizar a mesma linguagem de programação tanto no cliente quanto no servidor visando otimizar a velocidade de desenvolvimento e tirar proveito da grande quantidade de pacotes *open-source* disponível no repositório NPM<sup>9</sup> (MOZILLA, 2021b).

### 3.2.2 Quasar Framework

O Quasar *Framework* foi escolhido para o desenvolvimento do módulo cliente pelas facilidades que este proporciona ao processo, uma vez que contém componentes de interface prontos para serem utilizados, eliminando assim a necessidade de implementá-los manualmente. Além disso, o *framework* possui mecanismos internos para definição das cores das páginas como também para organização de *layout*.

---

8 <https://expressjs.com/>

9 <https://www.npmjs.com/>

### 3.2.3 PostgreSQL

Para a persistência dos dados na aplicação, foi utilizado o Sistema Gerenciador de Banco de Dados (SGBD) PostgreSQL<sup>10</sup> versão 13.0. Este é um *software open-source* bastante robusto, uma vez que existe há mais de 30 anos. Ele possui suporte para os módulos comuns da linguagem SQL (DDL, DML e DCL) e também para linguagens procedurais, como faz o módulo PL/pgSQL utilizado para inserir regras de negócio no banco de dados.

### 3.2.4 Sequelize

Mesmo sendo possível definir as consultas (*queries*) ao banco de dados utilizando a linguagem SQL, existem pacotes que realizam um mapeamento das relações existentes no banco e seus relacionamentos para classes da linguagem de programação utilizada. Tais pacotes realizam o que é chamado de Mapeamento Objeto-Relacional (do inglês Object-Relational Mapping, ou ORM), portanto são assim nomeados.

Os ORMs permitem que a interação com o banco de dados seja realizada acessando propriedades de objetos instanciados no código, onde cada relação no BD é representada por uma classe com seu nome, da qual os objetos são instanciados (USP, 2021). Assim, existe maior praticidade na implementação do *software*. O Sequelize<sup>11</sup> é um ORM para a linguagem JavaScript e permite a utilização em conjunto de diferentes SGBDs como também possui a vantagem de ser um *software open-source*.

---

10 <https://www.postgresql.org/>

11 <https://sequelize.org/>

### 3.3 Implantação e infraestrutura

A implantação deste sistema pode ser realizada pelos departamentos da UTFPR e então fornecer o acesso aos docentes do departamento, ou pela própria instituição, possibilitando o uso a todos os docentes da UTFPR.

Para realizar tal implantação é necessário um servidor *web* Linux, com os *softwares* Node.js na versão 12.19.0 ou superior, NPM versão 6.14.8 ou superior e PostgreSQL versão 13.1. Também é possível utilizar de soluções que auxiliam a configuração da infraestrutura para implantação do *software* na nuvem (*cloud*), como o Heroku<sup>12</sup>, porém, pode ser necessário investimento financeiro.

---

12 <https://www.heroku.com/>

## 4 DESENVOLVIMENTO

Esta seção apresenta a análise de requisitos, o processo de implementação do sistema como também as decisões tomadas a partir das dificuldades encontradas e o resultado do processo de desenvolvimento. Na Seção 4.1 são apontados os requisitos definidos para o *software*. Na Seção 4.2 é exibido como foi configurado o ambiente de desenvolvimento. Na Seção 4.3 é descrito o processo de desenvolvimento, assim como as suas etapas.

### 4.1 Análise dos requisitos

Para entender quais funcionalidades o sistema deve ter, foi analisada a documentação referente ao processo de promoção e progressão para a Classe Associado do Magistério Superior (UTFPR, 2021).

Foi verificado que para solicitar promoção ou progressão à Classe Associado o Docente deve incluir no processo de solicitação os seguintes documentos (UTFPR, 2021, p. 2):

- Relatório de Atividades, o qual é um arquivo no formato *xlsx*<sup>13</sup> e deve estar preenchido de acordo com a quantidade de atividades realizadas pelo Docente para o cálculo da pontuação;
- Cópia dos documentos comprobatórios das atividades utilizadas para contabilizar a pontuação no Relatório de Atividades, organizados em uma estrutura de diretórios de acordo com os itens do Relatório de Atividades/Tabela de Pontuação;
- Cópia da portaria de progressão/promoção anterior;
- Cópia do currículo Lattes; e
- Cópia do diploma de Doutor ou Livre Docente.

---

13 Formato de arquivo gerado pelo *software* de manipulação de planilhas Microsoft Excel.



Assim, é necessário que o sistema permita armazenar toda esta documentação bem como preencher o relatório de atividades. Também é conveniente que ele permita realizar o *download* da documentação comprobatória organizada em uma estrutura de diretórios, a qual é modelada de acordo com os grupos, subgrupos e seus itens, definidos na tabela de pontuação.

Como as atividades realizadas pelo docente definem a nota na avaliação de desempenho, o sistema deve realizar o cálculo das pontuações a cada registro de atividade inserido, permitindo a visualização dos indicadores das pontuações obtidas nos grupos, subgrupos e pontuação total, bem como os valores mínimos de pontuação exigidos, além dos valores máximos permitidos.

Uma vez que a pontuação por atividade realizada só é efetivada se apresentado o documento comprobatório de sua realização, também é necessário incluir um verificador de documentação comprobatória faltante. Sua função é simples: para cada registro de atividade cadastrado é verificado se este possui os documentos comprobatórios anexados, e, caso não os tenha, o usuário é informado.

Com os requisitos do sistema especificados, foi criado um Diagrama de Caso de Uso UML<sup>14</sup> para estabelecer as funcionalidades do sistema. A criação deste diagrama, além de documentar o sistema, ajuda o desenvolvedor a definir a ordem de implementação dos módulos, como também auxilia a modelagem do banco de dados. Tal diagrama é apresentado no Apêndice B deste documento.

De posse do Diagrama de Caso de Uso, foi criado o Diagrama de Entidade-Relacionamento. Este diagrama é fundamental para modelar o formato dos dados presentes no sistema bem como a forma como eles se relacionam. O Diagrama de Entidade-Relacionamento é apresentado no Apêndice C deste documento.

Criados estes dois diagramas, foi iniciada a etapa de implementação.

## 4.2 Preparação do ambiente de desenvolvimento

Para que seja possível iniciar a programação do *software*, é necessário inicializar e configurar as ferramentas utilizadas. Primeiramente foi instalado o editor

---

14 *Unified Modeling Language*: <https://www.uml.org/>

de texto *Visual Studio Code*<sup>15</sup> versão 1.50.1, ferramenta *open-source* mantida pela empresa Microsoft. Após isso, o Node<sup>16</sup> versão 12.19.0, o NPM<sup>17</sup> versão 6.14.8, o QuasarCLI<sup>18</sup> versão 1.1.2 e o Sistema Gerenciador de Banco de Dados (SGBD) PostgreSQL<sup>19</sup> versão 13.1 foram instalados. O sistema operacional utilizado durante a implementação do *software* foi o Slackware Linux<sup>20</sup> versão 14.2 com o *kernel* do Linux versão 4.4.240.

Com o ambiente devidamente configurado é necessário inicializar os módulos cliente e servidor do *software* com a devida estrutura de diretórios e arquivos de configuração. O primeiro módulo configurado foi o módulo servidor, como é descrito na Subseção 4.2.1. Posteriormente, o módulo cliente foi configurado, como apresenta a Subseção 4.2.2.

#### 4.2.1 Configuração do módulo servidor

O *Framework* Express possui a característica de ser bastante flexível, deixando ao desenvolvedor a tarefa de organizar a estrutura de diretórios de acordo com os interesses de seu projeto (EXPRESS, 2021). A escolha da estrutura dos diretórios teve como base a arquitetura em camadas modelo-visão-controlador (do inglês *model-view-controller*, ou MVC) (MOZILLA, 2021c).

Na arquitetura MVC, o *software* é dividido em camadas, de forma que cada camada possui uma responsabilidade bem definida. A camada modelo é responsável pela interação com o banco de dados, a camada controlador é responsável pelas regras de negócio da aplicação e a camada visão responsável por apresentar os dados e informações ao usuário (MOZILLA, 2021c).

Como descreve Mozilla (2021c, tradução livre), “Esta divisão de responsabilidades proporciona uma melhor divisão do trabalho bem como uma melhor manutenção”. Portanto, como esta forma de desenvolvimento apresenta tais

---

15 <https://code.visualstudio.com/>

16 <https://nodejs.org/en/>

17 <https://www.npmjs.com/>

18 <https://quasar.dev/>

19 <https://www.postgresql.org/>

20 <http://www.slackware.com/>

vantagens, a arquitetura em camadas MVC foi escolhida para organizar o código-fonte.

No caso de um sistema cliente-servidor, o módulo servidor implementa as camadas controlador e modelo, deixando o módulo cliente responsável pela visão. A organização dos diretórios e arquivos é apresentada na Figura 14 da página 52.

O primeiro arquivo criado é o *package.json*. Este arquivo armazena a referência de todas as dependências do projeto, seus *scripts* e metadados. O diretório *node\_modules* contém as bibliotecas e pacotes das dependências do projeto, referenciadas no *package.json*. O *download* dessas bibliotecas é realizado pelo NPM quando executado o comando “npm install”.

O arquivo *www* do diretório *bin* é o responsável por inicializar o servidor. Ele cria um servidor Node, o qual recebe a aplicação *Express* contida no arquivo *app.js* (este na raiz do projeto) como parâmetro, e redireciona as requisições recebidas em uma determinada porta (definida no arquivo *.env*) para a aplicação.

O arquivo *database.js*, no diretório *config*, define os parâmetros para realizar a conexão com o banco de dados. Os diretórios *migration* e *seeders* contidos no diretório *database* contém, respectivamente, os arquivos que criam automaticamente as relações, funções e *triggers* no banco de dados e os arquivos que populam o banco de dados com os dados padrão. Um dos dados inseridos pelos *seeders*, por exemplo, é o usuário inicial do sistema, o qual é utilizado para realizar o primeiro acesso à ferramenta.

O diretório *models* representa a camada modelo da arquitetura MVC e contém os arquivos para cada relação no banco de dados. Nestes arquivos estão contidos os atributos da relação bem como seus relacionamentos.

O diretório *controllers*, por sua vez, representa a camada controlador do MVC e seus arquivos implementam as regras de negócio da aplicação. Cada um desses arquivos contém várias funções, as quais manipulam as requisições de acordo com seu propósito, retornando uma resposta ao cliente.

O arquivo *index.js* no diretório *routes* funciona como um orquestrador. Ele processa a URL de uma requisição e redireciona esta requisição para a função correspondente de um dos *controllers*.

O arquivo *cors.js* no diretório *middleware* manipula as requisições antes de serem redirecionadas à função do controlador correspondente, e é necessário para promover maior segurança em ambientes cliente-servidor. O objetivo do Compartilhamento de Recursos de Origem Cruzada (do inglês Cross-Origin Resource Sharing, ou CORS) é definir uma lista de origens que podem acessar um recurso em um servidor (MOZILLA, 2021a). Assim, somente requisições de clientes cujo endereço está na lista serão aceitas pelo servidor.

Por fim, o diretório *documentosStorage* foi criado para armazenar os documentos salvos pelos usuários do sistema.

Figura 14 - Organização dos diretórios do módulo servidor

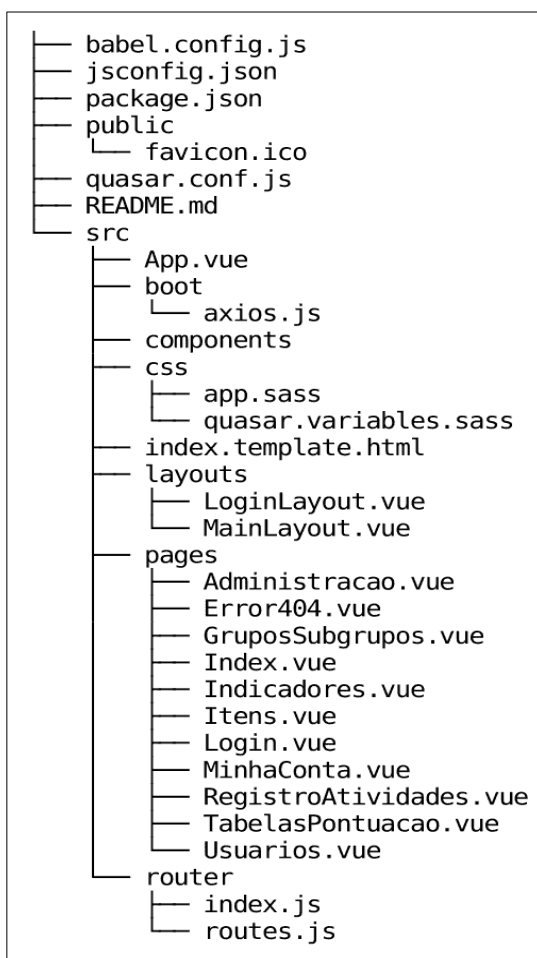


Fonte: Autoria própria (2021)

#### 4.2.2 Configuração do módulo cliente

A estrutura de diretórios do módulo cliente foi gerada automaticamente pelo *Quasar* CLI. A organização do módulo cliente ficou definida como apresenta a Figura 15.

Figura 15 - Organização dos diretórios do módulo cliente



Fonte: Autoria própria (2021)

O principal arquivo deste módulo é o *quasar.conf.js*. Ele contém toda a configuração utilizada pelo *framework* *Quasar*. O arquivo *package.json* e o diretório *node\_modules* possuem a mesma função que no módulo servidor. O diretório *public* armazena ícones e imagens utilizadas nas interfaces.

É no diretório *src* que ocorre o desenvolvimento das interfaces. O arquivo *App.vue* é o componente principal do sistema. O arquivo *index.template.html* trabalha em conjunto com o *App.vue* e funciona como modelo para as páginas. É nele que as páginas são inseridas.

O diretório *pages* contém os arquivos das páginas do sistema, as quais são desenvolvidas utilizando os componentes fornecidos pelo *framework*. O diretório *router* possui arquivos de definição das rotas do sistema, que relacionam uma URL com a página que deve ser apresentada.

O diretório *boot* possui arquivos de configuração utilizados na inicialização do app. O *axios.js* inicializa a biblioteca utilizada para realizar requisições ao módulo servidor com os parâmetros (endereço IP e porta) correspondentes.

### 4.3 Implementação

A primeira etapa na implementação do *software* foi programar as funcionalidades base da aplicação. Isto é, as funcionalidades de gerência de dados e elementos fundamentais, os quais são consumidos pela aplicação para compor funcionalidades de maior complexidade. Ainda, como o sistema é multiusuários, foi implementada a interface e a funcionalidade de autenticação do sistema.

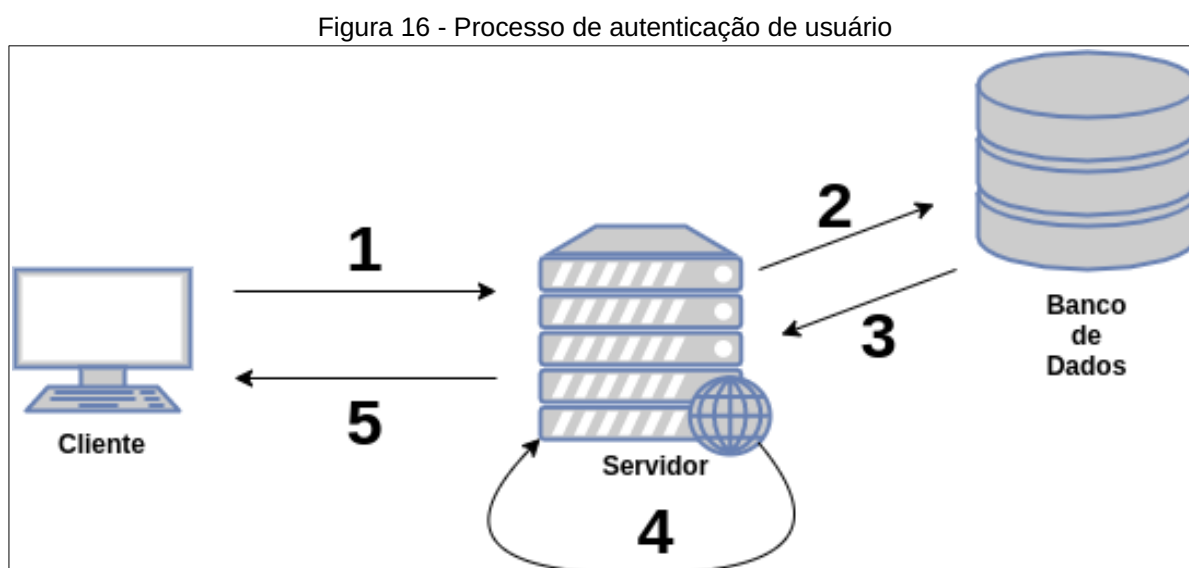
#### 4.3.1 Controle de acesso e módulos base

A primeira página desenvolvida foi a de acesso/autenticação dos usuários, uma vez que esta é a porta de entrada do sistema. Para a implementação desta funcionalidade são necessários dois campos para entrada de dados (registro do usuário e sua senha) e um botão para enviar ao servidor a solicitação de autenticação.

No momento que o servidor recebe a solicitação, é realizada a comparação da senha enviada pelo cliente com a que está armazenada no banco de dados (BD) para o usuário com o registro informado. É importante lembrar que a senha

armazenada no BD está criptografada pela biblioteca *bcrypt*<sup>21</sup>. Então, caso a credencial seja validada é retornado um *token* de autenticação *JSON Web Token*<sup>22</sup> (JWT) para o cliente. Tal *token* será armazenado no navegador e anexado ao cabeçalho das requisições posteriores, permitindo, assim, que nos próximos acessos não seja necessário realizar novamente a autenticação do usuário. Porém, caso a credencial não passe no teste de validação, é retornado um erro informando que ou o registro ou a senha estava incorreta, assim, solicitando outra tentativa de autenticação.

O processo de autenticação dos usuários é ilustrado na Figura 16.



Fonte: Autoria própria (2021)

Na etapa 1 o cliente envia uma solicitação de autenticação com os dados do usuário. Então, no passo 2, o servidor busca no BD os dados do usuário com o registro informado na requisição e no passo 3 o BD retorna ao servidor o resultado da busca. Posteriormente, no passo 4, o servidor interpreta a resposta e executa uma ação: caso exista usuário com o registro informado, o servidor utiliza a biblioteca *bcrypt* para comparar a senha informada no passo 1 com a *hash* armazenada e, se a senha estiver correta, retorna, na etapa 5, um *token* de

21 Biblioteca utilizada para criptografar senhas: <https://www.npmjs.com/package/bcrypt>

22 <https://jwt.io/>

autenticação para o cliente; caso contrário, o servidor retorna ao cliente uma mensagem que não existe usuário cadastrado com aquele registro.

Construída a página de acesso da aplicação, foi iniciado o desenvolvimento dos módulos para gerenciar os dados base do *software*. Existem três módulos para a gestão destes dados: de usuários, de grupos e subgrupos e de itens.

A primeira página a ser desenvolvida foi para o módulo de usuários, o qual está intimamente ligada ao sistema de autenticação. Na página do módulo de usuários existe um botão (ver item 1 da Figura 17) cuja função é inserir novos usuários. Também é apresentada a lista dos usuários já cadastrados junto de duas opções: atualizar os dados do usuário ou removê-lo do sistema. A Figura 17 mostra a interface do módulo de usuários.

Figura 17 - Página do módulo de usuários

Nome	Registro	Email	Administrador	Vincular Tabela de Pontuação	Editar Dados	Remover Usuário
Admin	12345	admin@email.com	Sim	3	2	
Teste	6789	teste@email.com	Não		2	

Fonte: Autoria própria (2021)

Junto aos dados pessoais do usuário existe um atributo nomeado ADMINISTRADOR, que recebe os valores verdadeiro ou falso e define o nível de permissão deste usuário no sistema. Caso o usuário tenha permissões de administrador ele possui acesso ao menu de administração do sistema, o qual é composto pelos módulos base e pela funcionalidade de manter tabelas de pontuação. Caso o usuário tenha somente permissões comuns o menu de administração não é exibido.

Para realizar o cadastro de tais usuários, bem como a atualização de seus registros, foram implementadas duas janelas modais<sup>23</sup>, as quais são exibidas quando os botões CADASTRAR USUÁRIO (indicado pelo número 1 em vermelho na Figura 17) e EDITAR USUÁRIO (indicado pelo número 2 na Figura 17) são pressionados,

<sup>23</sup> Elemento gráfico apresentado sobre a interface gráfica principal



respectivamente. Tais janelas são apresentadas ao usuário da forma como mostra o a Figura 18.

Figura 18 - Janela modal para cadastro de usuários

The image shows a web application interface with a modal form for user registration. The modal is titled "Cadastrar Usuário" and contains the following fields and options:

- Dados Gerais**
- Nome: [input field]
- Registro: [input field]
- Email: [input field]
- Senha: [input field]
- Administrador
- CANCELAR button
- CADASTRAR button

The background shows a dashboard with a table of users and navigation menus. The table has columns for Nome, Registro, and Email. The navigation menu includes HOME, INDICADORES, REGISTRO DE ATIVIDADES, PAINEL DE ADMINISTRAÇÃO, MINHA CONTA, and LOGOUT. The table has a header row with a "+ CADASTRAR USUÁRIO" button and a body with two rows of user data.

Nome	Registro	Email
Admin	12345	admin@email.com
Teste	6789	teste@email.com

Fonte: Autoria própria (2021)

Finalizada esta etapa, foi implementada a página do módulo de grupos e subgrupos como também o de itens. Os grupos e subgrupos possuem a descrição, que é um texto, e um indicador se a descrição é para um grupo ou subgrupo. Por sua vez, os itens possuem somente a descrição.

Foi decidido implementar o sistema desta forma para permitir flexibilidade e agilidade no cadastro de tabelas de pontuação, pois como as descrições já estão armazenadas no sistema é necessário somente adicioná-las e organizá-las de acordo com a tabela de pontuação.

A página para os grupos e subgrupos é apresentada na Figura 19. As janelas modais para cadastro e edição dos grupos e subgrupos é ilustrada na Figura 20 e a página de administração de itens é mostrada na Figura 21 (página 58).

Figura 19 - Página do módulo de grupos e subgrupos

Nome	Grupo ou Subgrupo	Editar Grupo/Subgrupo	Remover Grupo/Subgrupo
Grupo I - Atividades de Ensino, Orientação e Avaliação do Docente pelo Discente	Grupo		
Grupo II - Produção Intelectual	Grupo		
Grupo III - Atividades de Pesquisa, de Extensão, Administrativas e de Representação e Outras Atividades	Grupo		
Subgrupo I1 - Ensino	Subgrupo		
Subgrupo I2 - Orientação	Subgrupo		

Fonte: Autoria própria (2021)

Figura 20 - Janela modal para edição de grupos e subgrupos

Fonte: Autoria própria (2021)

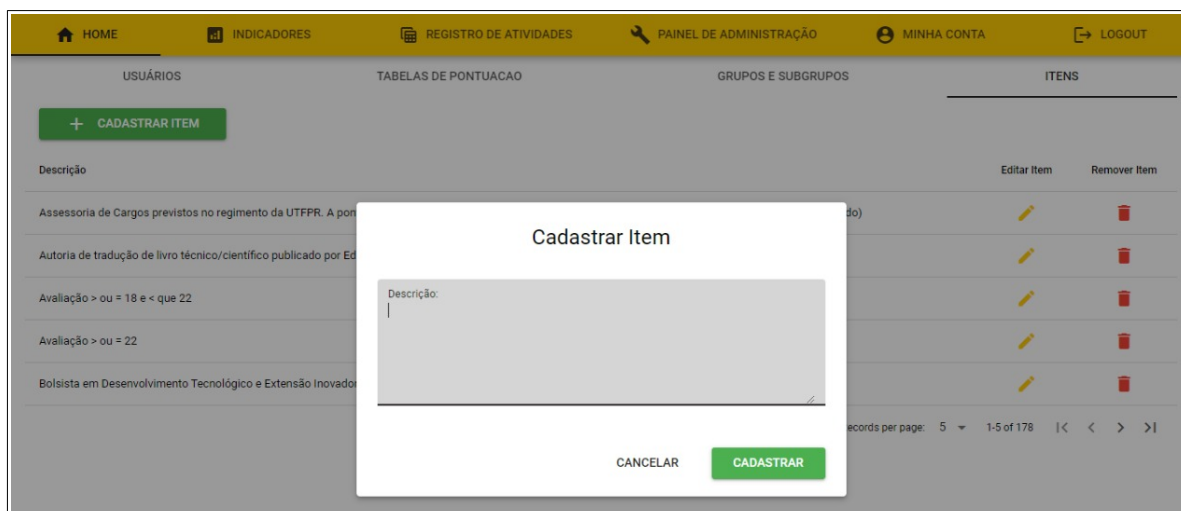
Figura 21 - Página do módulo de itens

Descrição	Editar Item	Remover Item
Assessoria de Cargos previstos no regimento da UTFPR. A pontuação será equivalente a 60 % da pontuação anual do cargo assessorado. (0,6* Cargo Assessorado)		
Autoria de tradução de livro técnico/científico publicado por Editora. Pontuação por livro.		
Avaliação > ou = 18 e < que 22		
Avaliação > ou = 22		
Bolsista em Desenvolvimento Tecnológico e Extensão Inovadora (DT) do CNPQ e similares de outras agências oficiais de fomento.		

Fonte: Autoria própria (2021)

As janelas modais para cadastro e edição de itens seguem a estrutura do exemplo da Figura 22.

Figura 22 - Janela modal para cadastro de itens



Fonte: Autoria própria (2021)

#### 4.3.2 Tabelas de pontuação

A implementação da gerência das tabelas de pontuação precisou passar por refinamentos em etapas posteriores do desenvolvimento. Tal mudança se mostrou necessária pelo fato do sistema ser multiusuários, assim, era necessário que a tabela de pontuação estivesse desacoplada do usuário que a estivesse utilizando. Ainda, o sistema deve ser flexível o suficiente para permitir que mais de uma tabela de pontuação seja cadastrada, pois, caso contrário, em uma eventual mudança nos valores das pontuações ou adição de novos itens, grupos ou subgrupos, o sistema se tornaria obsoleto.

Portanto, foi necessário que a tabela de pontuação possuísse uma descrição indicando a que ela se refere, como, por exemplo, Tabela para Associado 2018, como também a pontuação mínima de aprovação exigida na avaliação de desempenho, permitindo então que possam ser cadastradas diferentes versões da tabela de pontuação no sistema.

A criação das tabelas de pontuação ocorrem em duas etapas. Na primeira é definida uma descrição e a pontuação mínima para aprovação como também a organização dos grupos, subgrupos e suas pontuações mínimas exigidas bem como pontuações máximas permitidas. Na segunda são inseridos os itens na estrutura criada anteriormente, com sua respectiva regra de pontuação.

A sequência de passos para realizar a primeira etapa é apresentada nas Figuras 23, 24 e 26. No passo 1 são definidos a descrição da tabela e a pontuação mínima exigida para aprovação.

Figura 23 - Passo 1 da primeira etapa para criação de uma tabela de pontuação

HOME INDICADORES REGISTRO DE ATIVIDADES PAINEL DE ADMINISTRAÇÃO MINHA CONTA LOGOUT

USUÁRIOS TABELAS DE PONTUACAO GRUPOS E SUBGRUPOS ITENS

+ CADASTRAR TABELA DE PONTUAÇÃO

Cadastro de Tabela

Escreva uma descrição para a tabela e a pontuação mínima exigida na avaliação de desempenho

Descrição:

Pontuação Mínima para Aprovação:

PRÓXIMO

Organize a estrutura da tabela com seus devidos grupos e subgrupos  
Os grupos e subgrupos podem ter pontuações mínimas exigidas e valores de saturação definindo uma pontuação máxima possível

Insira subgrupos nos grupos correspondentes

Fonte: Autoria própria (2021)

No passo 2, apresentado na Figura 24, são adicionados os grupos da tabela. Para adicionar um grupo é pressionado o botão ADICIONAR GRUPO, de forma que é apresentada a janela modal da Figura 25.

Figura 24 - Passo 2 da primeira etapa para criação de uma tabela de pontuação

**Cadastro de Tabela**

- ✓ Escreva uma descrição para a tabela e a pontuação mínima exigida na avaliação de desempenho
- ✓ Organize a estrutura da tabela com seus devidos grupos e subgrupos  
Os grupos e subgrupos podem ter pontuações mínimas exigidas e valores de saturação definindo uma pontuação máxima possível

**+ ADICIONAR GRUPO**

**Grupos**

Grupo	Pontuação Mínima Exigida	Pontuação Máxima Permitida	Leva Excedente para Próxima Avaliação	Remover Grupo
Grupo 1 - Atividades de Ensino, Orientação e Avaliação do Docente pelo Discente	12 pontos	Não tem	Sim	<span style="color: red;">✖</span>

**VOLTAR** **PRÓXIMO**

Insira subgrupos nos grupos correspondentes

Fonte: Autoria própria (2021)

Figura 25 - Janela modal para adicionar grupo

**Adicionar Grupo**

Grupo

Pontuação Mínima Exigida:

Pontuação Máxima Permitida:

Leva excedente para próxima avaliação?  
 Não  Sim

\* Caso o grupo não tenha pontuação mínima necessária para aprovação deixe o campo vazio  
 \*\* Caso o grupo não tenha um limite de pontuação máxima permitida deixe o campo vazio

**CANCELAR** **ADICIONAR**

Fonte: Autoria própria (2021)

No passo 3, apresentado na Figura 26, são adicionados os subgrupos nos grupos que foram adicionados no passo 2, caso algum deles contenha subgrupos, de forma que a janela modal para inserir um subgrupo é ilustrada na Figura 27 e é exibida ao usuário quando pressionado o botão ADICIONAR SUBGRUPO, da Figura 26.

Figura 26 - Passo 3 da primeira etapa para criação de uma tabela de pontuação

Fonte: Autoria própria (2021)

Figura 27 - Janela modal para adicionar subgrupo

Fonte: Autoria própria (2021)

Uma vez que a estrutura da tabela está organizada, é pressionado o botão PRÓXIMO, da Figura 26 (indicado como 1), o qual armazena a estrutura criada e

redireciona o usuário à janela modal da Figura 28, na qual é possível administrar os itens da tabela criada. Primeiramente deve ser escolhido o Grupo em que se deseja inserir o item, e, caso o Grupo contenha subgrupos, escolher o Subgrupo.

Figura 28 - Janela modal de gerência dos itens de uma tabela de pontuação

Célula do Relatório	Nome	Descrição	Valor	Forma de Contabilização	Horas Necessárias	Pontuação Máxima Permitida Neste Item	Editar Item	Remover Item
C6	1.1.1	Curso técnico, graduação, lato sensu sem remuneração e stricto sensu. Pontuação por 15 horas de carga horária total da disciplina por semestre.	1 pontos	Por Horas Realizadas	15 horas	n/a		
C7	1.1.2	Curso técnico, graduação, lato sensu (para programas institucionais com financiamento oficial. Ex.: ETEC, FNDE, PARFOR, UAB). Pontuação por 15 horas de carga horária total da disciplina por semestre.	0.5 pontos	Por Horas Realizadas	15 horas	n/a		

Fonte: Autoria própria (2021)

Quando é pressionado o botão ADICIONAR ITEM (indicado pelo número 1 na Figura 28), ou o botão para editar (indicado pelo número 2 na Figura 28) um item que já está inserido na tabela, a janela modal responsável pela ação é apresentada ao usuário, como apresenta a Figura 29.

O primeiro componente é referente à descrição do item, a qual é uma das cadastradas no módulo de itens. O segundo campo é o identificador do item na tabela de pontuação. O terceiro campo, “Endereço da célula na planilha”, é utilizado para mapear a célula do relatório de atividades que está associada ao item, de forma que este mapeamento é utilizado na funcionalidade para preencher o relatório. Nos próximos campos devem ser inseridas as regras de contabilização, em que é definido o valor do item bem como se este valor é recebido por unidade de atividade realizada ou por horas realizadas, como também a pontuação máxima permitida, caso exista.

Figura 29 - Janelas modal para inserção de itens em uma tabela

The image shows a modal window titled "Adicionar Item" overlaid on a web application interface. The modal contains the following fields and controls:

- A dropdown menu for "Descrição".
- A text input field for "Item: (ex: I.2.1)".
- A text input field for "Endereço na célula na planilha".
- A text input field for "Valor:".
- A text input field for "Pontuação Limite neste Item:".
- Two radio buttons: "Valor Por Atividade Realizada" (unselected) and "Valor Por Horas Realizadas" (selected).
- A text input field for "Horas necessárias:".
- Buttons for "CANCELAR" and "ADICIONAR".

The background interface shows a table with columns "Célula do Relatório", "Nome", and "De". The table has two rows: one with "C6", "I.1.1", and "C...", and another with "C7", "I.1.2", and "C...".

Fonte: Autoria própria (2021)

O campo de Endereço da Célula na Planilha é associado a uma das funcionalidades principais da ferramenta: preenchimento automático do relatório de atividades. O relatório de atividades é uma planilha no formato xlsx, em que para contabilizar a pontuação obtida nos itens, devem ser inseridos os valores referentes à quantidade de atividades realizadas para cada item da tabela de pontuação.

Para preencher os valores no relatório, são definidas células (campos) específicas em que tal valor deve ser inserido, assim, foi necessário adicionar este atributo no item de forma que fosse possível ao *software* realizar o preenchimento das células. Então, a funcionalidade percorre cada um dos itens da tabela e utiliza o atributo Endereço da Célula na Planilha para encontrar a célula em que deverá inserir o valor. O código que realiza o preenchimento automático da planilha é apresentado no Apêndice D deste documento.

Uma vez que a gerência das tabelas de pontuação foi implementada, aliado ao fato de o sistema já incorporar acesso multiusuários, tais usuários precisam então ter acesso às tabelas de pontuação, bem como poder registrar atividades realizadas nos respectivos itens.



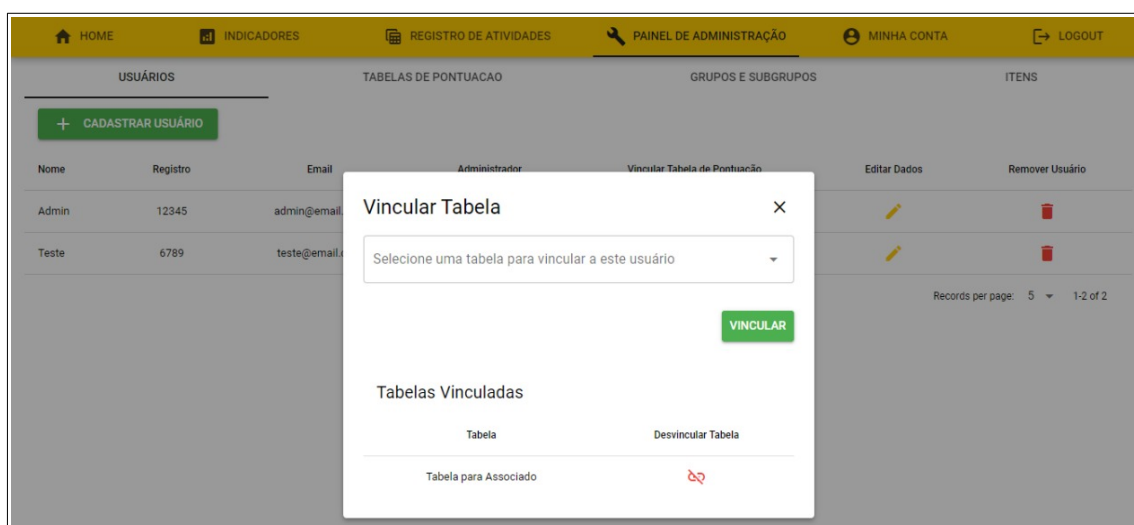
Em um primeiro momento, foi definido que os usuários criariam as suas próprias tabelas de pontuação, utilizando dos grupos, subgrupos e itens cadastrados no sistema. Tal forma permitiria bastante flexibilidade, pois o usuário poderia cadastrar quantas tabelas desejasse da forma como bem entendesse.

Porém, como todos os docentes da Classe Associado utilizam a mesma tabela de pontuação, seria desvantajoso deixar que cada usuário do sistema realizasse o mesmo processo de cadastro de tabela pois o processo de cadastro de cada um dos grupos, subgrupos e seus itens seria feito repetidamente por todos os usuários.

Assim, foi decidido centralizar a tabela de pontuação, tal que os usuários administradores possuem permissão para acessar a gerência das tabelas. Com isso, quando um usuário desejar utilizar uma tabela de pontuação, ele pode ser associado à tabela de forma que seus registros e pontuações obtidas estarão desacoplados da tabela em si.

Para realizar a vinculação de um usuário a uma tabela de pontuação é utilizada a janela modal apresentada na Figura 30. Para que tal janela seja exibida, deve ser pressionado o botão “Vincular Tabela de Pontuação” (indicado como 3) na Figura 17 da página 56. Também foi decidido limitar a vinculação de apenas uma tabela por usuário, dado que um Docente só pode pertencer a uma Classe e nível por vez.

Figura 30 - Janela modal para vincular uma tabela de pontuação a um usuário



Fonte: Autoria própria (2021)

### 4.3.3 Registro de realização atividade

A implementação do módulo de registro de atividades exigiu um cuidado de como o progresso e a pontuação do docente seriam armazenados e calculados. Para solucionar tal questão duas ações foram desenvolvidas: (i) criação de gatilhos (ou *triggers*)<sup>24</sup> no banco de dados para incrementar ou decrementar a quantidade de atividades cadastradas em um item como também as horas realizadas, caso a pontuação no item seja contabilizada por horas; (ii) criação de funções no servidor para aumentar ou diminuir a pontuação total obtida nos grupos e subgrupos.

As *triggers* executam funções (ou *functions*) armazenadas no banco, tal que o código da função executada quando um registro de atividade é armazenado pode ser observado na Figura 31. A página para registrar as atividades é apresentada na Figura 32.

Figura 31 - Função para contabilizar a realização de uma atividade

```

1 CREATE OR REPLACE FUNCTION processaincreasecounter() RETURNS TRIGGER AS $$
2 BEGIN
3   UPDATE "pontuacaoBind" SET
4     "qtAtividadesRealizadas" = "qtAtividadesRealizadas" + 1,
5     "qtHorasRealizadas" = "qtHorasRealizadas" + NEW."horasConsideradas"
6 WHERE
7   "pontuacaoldPontuacao" = NEW."pontuacaoldPontuacao"
8 AND
9   "usuarioldUsuario" = NEW."usuarioldUsuario";
10  RETURN NULL;
11 END

```

Fonte: Autoria própria (2021)

Com a implementação deste módulo foi possível obter o primeiro *feedback* de usuário, no caso, um Docente do Magistério Superior da UTFPR que já passou pelo processo de promoção à Classe Associado. Essa avaliação foi extremamente

<sup>24</sup> Uma *trigger* é armazenada dentro do banco de dados e define uma ação a ser realizada na ocorrência de inserções, atualizações ou remoções em uma relação (MYSQL, 2021)

importante pois apontou erros no processo de cadastro da realização de uma atividade não observados até o momento.

Na sua primeira versão, a funcionalidade de registro de atividade contabilizava a pontuação obtida no item a cada envio de documento comprobatório. Dessa forma, a pontuação estava atrelada ao envio do documento.

Porém, no *feedback* do Docente, foi relatado que poderiam haver atividades nas quais a sua comprovação é feita por mais de um documento. Assim, a contabilização da pontuação foi abstraída em um registro de atividade, no qual podem ser anexados documentos comprobatórios.

Figura 32 - Página de registro de atividades

Pontuação Contabilizada	Registrar Realização de Atividade	Item	Descrição	Valor	Forma de Contabilização	Horas Necessárias	Pontuação Máxima Permitida	Atividades Registradas
3.1333333333333333 pontos	<span style="color: red; font-weight: bold;">1</span>	I.1.1	Curso técnico, graduação, lato sensu sem remuneração e stricto sensu. Pontuação por 15 horas de carga horária total da disciplina por semestre.	1 pontos	Por Horas Realizadas	15 horas	n/a	
0 pontos		I.1.2	Curso técnico, graduação, lato sensu (para programas institucionais com financiamento oficial. Ex.: ETEC, FNDE, PARFOR, UAB). Pontuação por 15 horas de carga horária total da disciplina por semestre.	0.5 pontos	Por Horas Realizadas	15 horas	n/a	

Fonte: Autoria própria (2021)

Para realizar o registro de uma atividade deve ser pressionado o botão REGISTRAR REALIZAÇÃO DE ATIVIDADE (indicado como 1 na Figura 32), o qual apresenta a janela modal da Figura 33.

Nesta janela deve ser inserida uma descrição, a qual é um texto para descrever a atividade realizada e facilitar a organização dos registros. Caso o item contabilize a pontuação por horas, também deve ser inserida a quantidade de horas realizadas.

O botão LIMPAR DADOS, da Figura 32, remove todos os registros de atividades cadastrados, bem como os documentos comprobatórios associados. Tal funcionalidade deve ser utilizada cuidadosamente uma vez que não é possível reverter a ação executada.

Figura 33 - Janela modal para cadastrar uma atividade realizada

The screenshot displays a web interface with a modal window in the foreground. The modal is titled "Registrar Atividade Realizada" and contains the following elements:

- A text input field: "Escreva uma breve descrição para identificar a atividade realizada"
- A text input field: "Quantidade de horas realizadas:"
- A section titled "Anexar Documento(s) Comprobatório(s)" with a file selection button "Selecione..." and a note: "Escolha o(s) documento(s) que comprova(m) a realização desta atividade (seleção múltipla permitida). Caso não o(s) tenha em posse, é possível anexá-los depois." followed by "0 (0.00B)".
- Buttons: "CANCEL" and "REGISTRAR"

The background interface includes a top navigation bar with links for HOME, INDICADORES, REGISTRO DE ATIVIDADES, PAINEL DE ADMINISTRAÇÃO, MINHA CONTA, and LOGOUT. A sidebar on the left shows a tree structure for "Grupo I - Atividades de Ensino, Orientação e Avaliação do Docente pelo Discente" and "Subgrupo I1 - Ensino". A table on the right lists activities with columns for "Horas Necessárias", "Pontuação Máxima Permitida", and "Atividades Registradas".

Fonte: Autoria própria (2021)

Para consultar os registros de atividade inseridos em um item é utilizada a janela modal da Figura 34.

No *feedback* docente também foi mencionado sobre a possibilidade de realizar o cadastro de um registro de atividade sem anexar os documentos comprobatórios. Dessa forma, seria possível cadastrar a atividade e simular a pontuação obtida, mesmo sem a posse de tais documentos, e, no momento em que obtê-los, anexá-los no registro.

Assim, na janela modal da Figura 34, foi permitido realizar o registro sem anexar documentos. Para isso, foi inserido um botão (indicado como 1) para permitir posteriormente anexar os documentos no registro em questão.

Figura 34 - Janela modal para apresentar registros de atividades cadastrados



Fonte: Autoria própria (2021)

Porém, como isto implica que podem existir registros de atividades sem os respectivos documentos comprobatórios anexados, foi desenvolvida uma funcionalidade de verificação de documentos comprobatórios faltantes. Tal funcionalidade é apresentada na Figura 36 da página 71.

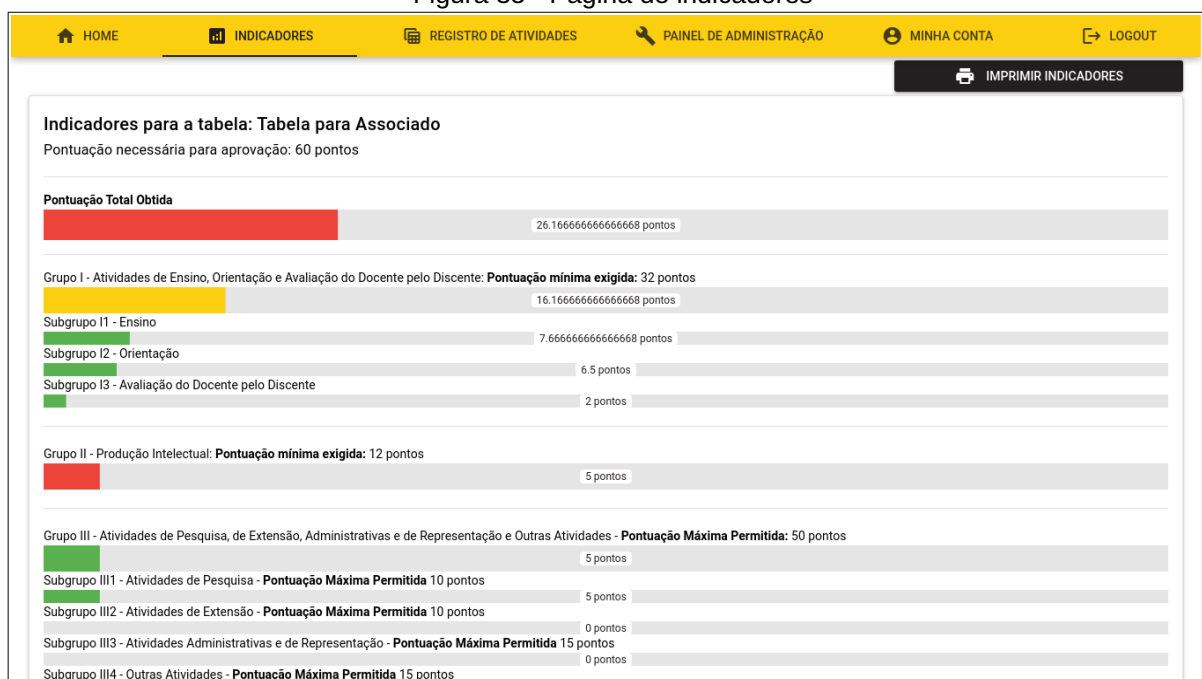
#### 4.3.4 Indicadores

Para o desenvolvimento deste módulo utilizou-se componentes prontos do *framework* Quasar. A página de indicadores ficou conforme apresenta a Figura 35.

Em tal página é apresentado uma barra de progresso com a pontuação total obtida, a qual é a soma simples das pontuações obtidas nos grupos. Também são apresentadas barras de progresso para cada um dos grupos e subgrupos.

A cor das barras de progresso é alternada entre vermelho, amarelo e verde, dependendo do quanto falta para alcançar a pontuação para aprovação, bem como as pontuações mínimas exigidas. Para os grupos que não tem pontuação mínima exigida a cor será sempre verde.

Figura 35 - Página de indicadores



Fonte: A autoria própria (2021)

Na página de indicadores também foi inserido um botão para imprimi-los. Esta funcionalidade é conveniente pois permite ao docente guardar uma cópia de seu desempenho em Progressões ou Promoções passadas.

#### 4.3.5 Página inicial

A página inicial foi desenvolvida de forma a agrupar funcionalidades finais da ferramenta, como o *download* dos documentos comprobatórios e o preenchimento automático do relatório de atividades. Assim, a página inicial é útil para que o docente tenha acesso a essas funcionalidades de forma centralizada e sem precisar navegar no sistema para encontrá-las.

Nesta página também foi agrupado junto das funcionalidades o verificador de documentos comprobatórios faltante, uma vez que tais documentos são parte fundamental do processo de progressão e promoção. A página inicial é apresentada a Figura 36.

Figura 36 - Página inicial

Bem Vindo à Ferramenta para Gestão de Documentos da Progressão e Promoção à Classe Associado do Magistério Superior

Gestão de Documentos

DOWNLOAD DE DOCUMENTOS COMPROBATÓRIOS

PREENCHER RELATÓRIO DE ATIVIDADES COM AS PONTUAÇÕES

Checklist de Documentos Comprobatórios Pendentes

VERIFICAR

LIMPAR

Registro(s) de Atividade com documentação comprobatória pendente na tabela: Tabela para Associado

Registro de Atividade	Grupo/Subgrupo	Item	Descrição	Ir para
Registro 2	Subgrupo I2 - Orientação	II.1.1	Orientação em andamento de aluno de mestrado stricto sensu. Pontuação por aluno.	<a href="#">↻</a>
Aulas para graduação ADS	Subgrupo I1 - Ensino	I.1.1	Curso técnico, graduação, lato sensu sem remuneração e stricto sensu. Pontuação por 15 horas de carga horária total da disciplina por semestre.	<a href="#">↻</a>
Aulas para Graduação BCC	Subgrupo I1 - Ensino	I.1.1	Curso técnico, graduação, lato sensu sem remuneração e stricto sensu. Pontuação por 15 horas de carga horária total da disciplina por semestre.	<a href="#">↻</a>

Records per page: 5 1-3 of 3

Fonte: Autoria própria (2021)

Para preencher o relatório de atividades existe um pré-requisito: todos os registros de atividade devem ter documentos comprobatórios anexados. Tal dependência existe pois toda a pontuação obtida deve ser comprovada. Assim, foi implementado o verificador de documentos comprobatórios faltante na página inicial do sistema.

Tal funcionalidade apresenta uma tabela com todos os registros de atividades sem documentos comprobatórios anexados, como ilustra a Figura 36. Para cada registro existe um *link* que leva o usuário para a página de registro de atividades, de forma que o respectivo grupo e subgrupo do item no qual o registro foi inserido estarão selecionados. Caso não exista documentos faltando é apresentado um texto informativo, como na Figura 37.

Figure 37 - Mensagem do verificador de documentos faltantes quando não há envios pendentes

Checklist de Documentos Comprobatórios Pendentes

VERIFICAR

LIMPAR

Não existem envios pendentes!

Fonte: Autoria própria (2021)

Após a implementação do verificador, o preenchimento automático do relatório de atividades foi implementado. Para preencher o relatório deve ser pressionado o botão PREENCHER RELATÓRIO DE ATIVIDADES. Assim, é apresentado uma janela modal, na qual deve ser inserido o relatório fornecido pela UTFPR, e então enviá-lo. Após isso, o sistema retornará o documento preenchido para o usuário salvar localmente.

Como na avaliação de desempenho é solicitada toda a documentação comprobatória, foi decidido implementar uma funcionalidade para permitir ao docente realizar o *download* dos documentos anexados aos registros. Para usar esta funcionalidade deve ser pressionado o botão DOWNLOAD DE DOCUMENTOS COMPROBATÓRIOS.

Tal funcionalidade organiza os documentos em uma estrutura de diretórios de acordo com os itens, grupos e subgrupos da tabela de pontuação. Para cada registro em um item é criado um diretório com os documentos que comprovam aquele registro. No final é retornado ao usuário um arquivo compactado no formato zip.

#### 4.3.6 Página Minha conta

A página de dados pessoais consiste nas ações que permitem ao docente atualizar seus dados cadastrais, como registro, nome e e-mail. Em tal página também é possível realizar a redefinição da senha.

Como o valor do sistema está em ajudar o docente no processo de promoção e progressão, foi decidido implementar a página de dados pessoais de forma a permitir armazenar outros documentos pessoais exigidos no processo de progressão e promoção à Classe Associado: cópia do currículo Lattes, diploma de Doutor ou Livre Docente e portaria da progressão ou promoção anterior.

Assim, a ferramenta também serve como um repositório, no qual todos os documentos necessários para o processo estarão armazenados. A Figura 38 ilustra o *layout* da página de dados e documentos pessoais.



Figura 38 - Página de dados e documentos pessoais

The image shows a web interface for a user's profile. At the top is a yellow navigation bar with links: HOME, INDICADORES, REGISTRO DE ATIVIDADES, PAINEL DE ADMINISTRAÇÃO, MINHA CONTA, and LOGOUT. Below this is a section titled 'Meus Dados' containing a form with fields for 'Nome: Admin', 'Registro: 12345', and 'Email: admin@email.com'. A black button labeled 'ALTERAR DADOS' is positioned below the form. The next section is 'Documentos Pessoais', which features three green buttons with plus signs and labels: '+ LATTES', '+ PORTARIA DE PROGRESSÃO ANTERIOR', and '+ DIPLOMA DE DOUTOR OU LIVRE DOCENTE'. Below these buttons are three links: 'Ver Documento', 'Documento', and 'Remover Documento'. The final section is 'Alterar Senha', which contains a red button labeled 'ALTERAR SENHA'.

Fonte: Autoria própria (2021)

## 5 CONCLUSÃO

Nesta seção serão apresentadas as considerações finais em que se chegou a partir do desenvolvimento deste trabalho, assim como ideias de trabalhos futuros.

### 5.1 Considerações finais

O desenvolvimento deste trabalho teve como propósito apresentar uma ferramenta para auxiliar os docentes do Magistério Superior da UTFPR na gestão dos documentos comprobatórios necessários para o processo de progressão e promoção na carreira docente.

A ferramenta desenvolvida atua como um repositório, no qual os documentos são anexados a registros de atividades realizadas pelo docente, os quais contabilizam pontos nos itens da tabela de pontuação usada para a avaliação de desempenho. Também foram desenvolvidas funcionalidades que mapeiam o valor desses itens, assim, a cada registro de atividade é possível apresentar indicadores de progresso, os quais ajudam o docente a gerenciar as atividades que desempenhará em seu cotidiano a fim de alcançar a aprovação na avaliação.

Como no processo é solicitada a entrega dos documentos comprobatórios, bem como o relatório de atividades devidamente preenchido, foram desenvolvidos módulos na ferramenta para preencher o relatório com a pontuação contabilizada pelos registros de atividades cadastrados e para permitir ao docente realizar o *download* dos documentos comprobatórios devidamente organizados em uma estrutura de diretórios, a qual é definida de acordo com os grupos, subgrupos e seus itens da tabela de pontuação.

Ainda, como o valor da ferramenta está no que ela tem a oferecer de forma a ajudar o docente no processo de progressão e promoção, foram desenvolvidas funcionalidades para verificar se existem documentos comprobatórios faltando nos registros de atividade cadastrados. Também foi permitido ao docente salvar outros documentos pessoais na ferramenta, como: currículo Lattes, portaria da última progressão ou promoção e diploma de Doutor ou Livre Docente.

Desse modo, o docente poderá armazenar todos os documentos necessários ao processo na plataforma, e no momento em que iniciar o pedido de progressão/promoção, terá fácil acesso a todos os seus documentos.

Para a implementação foram usadas tecnologias de desenvolvimento *web*, uma vez que é necessário o acesso remoto à ferramenta, bem como o acesso via diferentes plataformas. Portanto, como as ferramentas *web* utilizam como base de funcionamento a Internet, elas se mostraram ideais para o projeto.

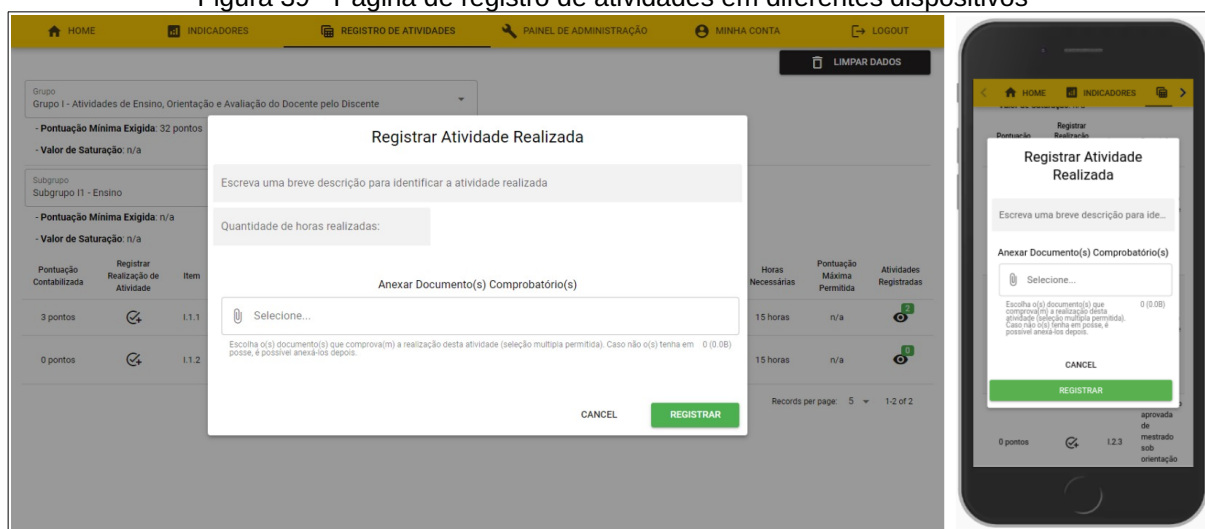
Para o desenvolvimento do módulo cliente foi utilizado o *framework* Quasar, o qual possui componentes de UI prontos para serem utilizados, proporcionando um desenvolvimento mais ágil quando comparado ao desenvolvimento manual destes componentes.

O desenvolvimento do módulo servidor utilizou do *framework* Express, o qual utiliza o ambiente Node.js para funcionamento, bem como a ferramenta NPM para a gerência dos pacotes utilizados, sendo, dessa forma, uma ferramenta robusta para o desenvolvimento de aplicações.

A estratégia utilizada no processo de implementação do *software* seguiu a metodologia de desenvolvimento incremental. Portanto, módulos mais simples e que funcionam como base para o sistema foram implementados primeiro, gerando insumo para o desenvolvimento de módulos mais complexos. Tal estratégia permitiu apresentar o sistema em versões intermediárias para docentes que possuem experiência com o processo de promoção, os quais contribuíram com *feedback* importante para ajustar o sistema ao processo de progressão e promoção.

O resultado do desenvolvimento se apresentou como um sistema que pode ser acessado remotamente e de diferentes meios, uma vez que a interface se adéqua ao dispositivo do usuário (*desktop, mobile*), como mostra a Figura 39. Além disso, o sistema obtido também suporta algumas mudanças nas tabelas de pontuação, sendo assim um sistema flexível.

Figura 39 - Página de registro de atividades em diferentes dispositivos



Fonte: Autoria própria (2021)

## 5.2 Trabalhos futuros

Este projeto foi desenvolvido unicamente para a Classe Associado devido ao fato da avaliação de desempenho para as outras Classes do Magistério Superior possuírem regras distintas para o cálculo das pontuações. Mas ainda existe a possibilidade de desenvolvimento de novos módulos como forma de ampliar o escopo do sistema às outras Classes da carreira docente.

Também existem possibilidades para projetos de integração, como, por exemplo, com a plataforma Lattes. Tal integração poderia consumir os dados da plataforma e então inserir automaticamente registros quando novas atividades forem realizadas.

Outra possibilidade para desenvolvimentos futuros seria a integração com o Sistema Acadêmico, permitindo que informações como, carga horária das disciplinas ministradas, por exemplo, fossem automaticamente coletadas e incluídas na plataforma.

## REFERÊNCIAS

ANGULAR. **What is Angular?**, 2021. Disponível em: <https://angular.io/guide/what-is-angular>. Acesso em: 13 mai. 2021.

BJÖRK, B.-C. The Impact of Electronic Document Management on Construction Information Management. **Distributing Knowledge In Building**, Denmark, p. 1–9, jun. 2002. Disponível em: <https://itc.scix.net/paper/w78-2002-3>. Acesso em: 28 abr. 2020.

BRASIL. Lei nº 12.772, de 28 de dezembro de 2012. **Diário Oficial da União**, seção 1, Brasília, DF, ano 149, n. 251, p. 1, 31 dez. 2012. Disponível em: <http://pesquisa.in.gov.br/imprensa/jsp/visualiza/index.jsp?data=31/12/2012&jornal=1&pagina=1&totalArquivos=320>. Acesso em: 28 abril 2020.

BRASIL. Lei nº 12.863, de 24 de setembro de 2013. **Diário Oficial da União**, seção 1, Brasília, DF, ano 150, n. 186, p. 1, 25 set. 2013. Disponível em: <http://pesquisa.in.gov.br/imprensa/jsp/visualiza/index.jsp?data=25/09/2013&jornal=1&pagina=1&totalArquivos=192>. Acesso em: 28 abril 2020.

BRASIL. Portaria Nº 920, de 4 de Novembro de 2020. **Diário Oficial da União**, seção 1, Brasília, DF, ano 158, n. 212, p. 42, 06 nov. 2020. Disponível em: <https://pesquisa.in.gov.br/imprensa/jsp/visualiza/index.jsp?data=06/11/2020&jornal=515&pagina=42&totalArquivos=174>. Acesso em: 21 jan. 2021.

CALDERON, W. R. *et al.*. O processo de gestão documental e da informação arquivística no ambiente universitário. **Ciência da Informação**, v. 33, n. 3, p. 97–104, 2004.

CASTRO, Ceferino Castro. **Uma proposta baseada em padrões de design para o desenvolvimento de sistemas cooperativos em ambiente aberto**. 1999. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 1999.

COULOURIS, G. *et al.*. **J. Sistemas distribuídos : conceitos e projeto**. Porto Alegre: Bookman, 2013. ISBN 9788582600542.

COSTOIU, M. *et al.*. Electronic document management information system for universities. **Chemical Engineering Transactions**, v. 29, p. 1639–1644, 2012.

EXPRESS. **FAQ**, 2021. Disponível em: <https://expressjs.com/en/starter/faq.html>. Acesso em: 22 jan. 2021.

FERREIRA, D. C.; AZEVEDO, D. J. P.; FAGUNDES, C. e-DOC : Sistema de Gestão Eletrônica de Documentos. **Anais SULCOMP**, Criciúma, 24 a 27 de setembro de

2014. Disponível em <http://periodicos.unesc.net/sulcomp/article/view/1806>. Acesso em: 15 abr. 2021.

FOROUZAN, B. A. **Comunicação de dados e redes de computadores**. Porto Alegre: ArtMed, 2010. ISBN 9788563308474.

IBM. **Database relationships**, 2021. Disponível em: <https://www.ibm.com/docs/en/mam/7.6.0?topic=structure-database-relationships>. Acesso em: 13 mai. 2021.

JADID, M. N.; IDREES, M. M.. Electronic document management system (EDMS) in civil engineering projects. **33rd Annual Conference of the Canadian Society for Civil Engineering**, 6th Construction Specialty Conference, 2005, v. 1, p. 417-425, 2005.

KEEVA. **O que é GED? Confira nosso guia completo e descubra**, 2020. Disponível em: <https://keeva.io/especiais/ged>. Acesso em: 9 mai. 2020.

LARAVEL. **Instalation**, 2021. Disponível em: <https://laravel.com/docs/8.x#why-laravel>. Acesso em: 13 mai. 2021.

MOZILLA. **An overview of HTTP**, 2020a. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>. Acesso em: 23 jun. 2020.

MOZILLA. **Cross-Origin Resource Sharing (CORS)**, 2021a. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>. Acesso em: 15 fev. 2021.

MOZILLA. **Express/Node introduction**, 2021b. Disponível em: [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction). Acesso em: 25 jan. 2020.

MOZILLA. **How does the Internet work?**, 2020b. Disponível em: [https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/How\\_does\\_the\\_internet\\_work](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_internet_work). Acesso em: 23 jun. 2020.

MOZILLA. **HTML: HyperText Markup Language**, 2020c. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/HTML>. Acesso em: 11 jun. 2020.

MOZILLA. **HTTP Methods**, 2020d. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>. Acesso em: 11 jun. 2020.

MOZILLA. **HTTP response status codes**, 2020e. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>. Acesso em: 11 jun. 2020.

MOZILLA. **MVC**, 2021c. Disponível em: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>. Acesso em: 11 jan. 2021.

MOZILLA. **Server-side web frameworks**, 2021d. Disponível em: [https://developer.mozilla.org/en-US/docs/Learn/Server-side/First\\_steps/Web\\_frameworks](https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks). Acesso em: 13 jan. 2021.

MOZILLA. **What is JavaScript?**, 2021e. Disponível em: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript). Acesso em: 13 jan. 2021.

MYSQL. **Using Triggers**, 2021. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/triggers.html>. Acesso em: 30 mar. 2021.

NODE. **About Node.js**, 2021. Disponível em: <https://nodejs.org/en/about/>. Acesso em: 25 jan. 2021.

NTU. **HTTP (HyperText Transfer Protocol)**, 2020. Disponível em: [https://personal.ntu.edu.sg/ehchua/programming/webprogramming/http\\_basics.html](https://personal.ntu.edu.sg/ehchua/programming/webprogramming/http_basics.html). Acesso em: 23 jun. 2020.

ORACLE. **What Is a Database**, 2021a. Disponível em: <https://www.oracle.com/database/what-is-database/>. Acesso em: 13 mai. 2021.

ORACLE. **What Is a Relational Database**, 2021b. Disponível em: <https://www.oracle.com/database/what-is-a-relational-database/>. Acesso em: 13 mai. 2021.

QUASAR. **Why Quasar?**, 2021. Disponível em: <https://quasar.dev/introduction-to-quasar>. Acesso em: 8 jan. 2021.

REACT. **React**, A JavaScript library for building user interfaces, 2021. Disponível em: <https://reactjs.org/>. Acesso em: 13 mai. 2021.

SEBRAE. **Portal Sebrae**, 2020. Plano de carreira: o que é e como criar. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/artigos/plano-de-carreira-o-que-e-e-como-criar,66d839f5192ed510VgnVCM1000004c00210aRCRD>. Acesso em: 14 mar. 2020.

SILVA, Flávio Luiz D'Oliveira e.. **Gerenciamento Eletrônico de Documentos (GED): Natureza, Princípios e Aplicações**. 2001. Projeto Supervisionado (Bacharelado em Ciência da Computação) – Faculdade de Ciência e Tecnologia, Universidade Federal de Mato Grosso, Cuiabá, 2001.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011. ISBN 978-85-7936-108-1.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ. Conselho Universitário. **Deliberação nº 28, de 26 de outubro de 2018**. Regulamento de avaliação de desempenho acadêmico para fins de promoção e progressão funcional para a Classe de professor associado de docentes da carreira de Magistério Superior da Universidade Tecnológica Federal do Paraná: Conselho Universitário, 2018. Disponível em: [https://sei.utfpr.edu.br/sei/publicacoes/controlador\\_publicacoes.php?acao=publicacao\\_visualizar&id\\_documento=663762&id\\_orgao\\_publicacao=0](https://sei.utfpr.edu.br/sei/publicacoes/controlador_publicacoes.php?acao=publicacao_visualizar&id_documento=663762&id_orgao_publicacao=0). Acesso em: 16 mar. 2020.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ. **Instruções para o Preenchimento do Relatório de Atividades Docentes para a Promoção para a Classe de Professor Associado e para a Progressão Funcional na Classe de Professor Associado da UTFPR**. Curitiba: Ed. UTFPR, 2021. Disponível em: <http://portal.utfpr.edu.br/servidores/site/documentos/carreira-e-remuneracao/magisterio-superior/associado/instrucoes-de-preenchimento-de-formulario-para-fins-de-promocao-a-classe-associado>. Acesso em: 8 jan. 2021.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ. **Portal UTFPR**, 2020. Promoção e Progressão. Disponível em: <http://portal.utfpr.edu.br/servidores/site/carreira-e-remuneracao/docentes-magisterio-superior/promocao-e-progressao/promocao-e-progressao>. Acesso em: 16 mar. 2020.

USP. Instituto de Matemática e Estatística da Universidade de São Paulo. **Object-Relational Mapping**, 2021. Disponível em: [https://www.ime.usp.br/~reverbel/object\\_relational\\_mapping.html](https://www.ime.usp.br/~reverbel/object_relational_mapping.html). Acesso em: 02 fev. 2021.

VUEJS. **Components Basics**, 2021a. Disponível em: <https://vuejs.org/v2/guide/components.html>. Acesso em: 01 fev. 2021.

VUEJS. **Introduction**, 2021b. Disponível em: <https://vuejs.org/v2/guide/>. Acesso em: 25 jan. 2021.

VUEJS. **Single File Components**, 2021c. Disponível em: <https://vuejs.org/v2/guide/single-file-components.html>. Acesso em: 26 mar. 2021.

W3. **Answers for Young People**, 2020a. Disponível em: <https://www.w3.org/People/Berners-Lee/Kids.html#invent>. Acesso em: 23 jun. 2020.

W3. **Frequently asked questions**, 2020b. Disponível em: <https://www.w3.org/People/Berners-Lee/FAQ.html#InternetWeb>. Acesso em: 23 jun. 2020.

W3. **The World-Wide Web**, 2020c. Disponível em: <https://www.w3.org/People/Frystyk/thesis/WWW.html>. Acesso em: 23 jun. 2020.

W3. **What is HyperText**, 2021. Disponível em: <https://www.w3.org/WhatIs.html>. Acesso em: 23 jan. 2021.



W3SCHOOLS. **CSS Introduction**, 2021a. Disponível em:  
[https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp). Acesso em: 13 jan. 2021.

W3SCHOOLS. **How To Add CSS**, 2021b. Disponível em:  
[https://www.w3schools.com/css/css\\_howto.asp](https://www.w3schools.com/css/css_howto.asp). Acesso em: 13 jan. 2021.

W3SCHOOLS. **HTML Introduction**, 2021c. Disponível em:  
[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp). Acesso em: 13 jan. 2021.

W3SCHOOLS. **Node.js NPM**, 2021d. Disponível em:  
[https://www.w3schools.com/nodejs/nodejs\\_npm.asp](https://www.w3schools.com/nodejs/nodejs_npm.asp). Acesso em: 14 fev. 2021.

**APÊNDICE A** - Tabela de pontuação para Avaliação de Desempenho na  
progressão e promoção à classe Associado do Magistério Superior

<b>Grupo I - Atividades de Ensino, Orientação e Avaliação do Docente pelo Discente</b>			
<b>Atividade</b>	<b>Descrição</b>	<b>Ponto por ano</b>	<b>Ponto por evento</b>
<b>Subgrupo I1 - Ensino</b>			
I.1.1	Curso técnico, graduação, <i>lato sensu</i> sem remuneração e <i>stricto sensu</i> . Pontuação por 15 horas de carga horária total da disciplina por semestre		1
I.1.2	Curso técnico, graduação, <i>lato sensu</i> (para programas institucionais com financiamento oficial. Ex.: ETEC, FNDE, PARFOR, UAB). Pontuação por 15 horas de carga horária total da disciplina por semestre		0,5
<b>Subgrupo I2 - Orientação</b>			
I.2.1	Orientação em andamento de aluno de mestrado <i>stricto sensu</i> . Pontuação por aluno		0,5
I.2.2	Coorientação em andamento de aluno de mestrado <i>stricto sensu</i> . Pontuação por aluno		0,3
I.2.3	Dissertação aprovada de mestrado sob orientação do postulante		6
I.2.4	Dissertação aprovada de mestrado sob coorientação do postulante		3
I.2.5	Orientação em andamento de aluno de doutorado <i>stricto sensu</i> . Pontuação por aluno		0,5
I.2.6	Coorientação em andamento de aluno de doutorado <i>stricto sensu</i> . Pontuação por aluno		0,3
I.2.7	Tese aprovada de doutorado sob orientação do postulante		10
I.2.8	Tese aprovada de doutorado sob coorientação do postulante		5
I.2.9	Orientação concluída de alunos inscritos nos programas institucionais de iniciação científica, extensão, inovação e docência da UTFPR (e.g., PIBIC, PIBITI, PIBIC-JR, PIBIC-EM, PIBEXT, PIBIN, PIBID, PET, RONDON) e/ou voluntário sem bolsa de projeto registrado na diretoria responsável pelo acompanhamento.		1,5
I.2.10	Orientação - Coordenador de área de projetos PIBID. Pontuação por orientação de grupo com pelo menos 12 bolsistas.		5
I.2.11	Orientação de estágio não obrigatório ou monitoria. Pontuação por 60 horas de orientação realizada.		0,5
I.2.12	Orientação concluída de estágio obrigatório em cursos técnico, e graduação (Bacharelados e Tecnologias). Pontuação por orientação individual		0,5
I.2.13	Orientação concluída de estágio obrigatório em cursos de Licenciatura. Pontuação por orientação individual		1,5
I.2.14	Orientação concluída de estágio em cursos de Licenciatura para uma turma. Pontuação por orientação de turma		5

I.2.15	Orientação de bolsista de programa de desenvolvimento científico regional, de recém-doutor e de pós-doutoramento. Pontuação por bolsista por semestre concluído de orientação ou supervisão		1
I.2.16	Orientação de bolsistas com titulação de mestre ou doutor, participante de projetos P&D&I (ex.: PADCT, CNPq, FINEP, FA, Empresas). Pontuação por bolsista por semestre concluído de orientação ou supervisão		1
I.2.17	Orientação concluída de trabalho de conclusão de curso técnico e de graduação. Pontuação por trabalho orientado		1,5
I.2.18	Orientação em andamento de trabalho de conclusão de curso técnico e de graduação. Pontuação por trabalho com mais de 6 meses de orientação		0,5
I.2.19	Orientação de trabalho de conclusão de curso de especialização sem remuneração ou com remuneração para programas institucionais com financiamento (ex.: Etec, FNDE, PARFOR, UAB). Pontuação por trabalho orientado		1
I.2.20	Orientação de trabalho no programa de desenvolvimento educacional – PDE ou equivalente. Pontuação por trabalho orientado, limitado a 20 pontos		2
I.2.21	Coorientação concluída de trabalho de conclusão de curso técnico e de graduação. Pontuação por trabalho		1
I.2.22	Coorientação em andamento de trabalho de conclusão de curso técnico e de graduação. Pontuação por trabalho com mais de 6 meses de orientação		0,2
I.2.23	Coorientação de trabalho de final de curso de especialização sem remuneração ou com remuneração para programas institucionais com financiamento (ex.: eTEC, FNDE, PARFOR, UAB). Pontuação por trabalho		0,6
<b>Subgrupo I3 – Avaliação do Docente pelo Discente</b>			
I.3.1	Avaliação > ou = 18 e < que 22	4	
I.3.2	Avaliação > ou = 22	6	
Observação Grupo I – No Subgrupo I.1 quando uma disciplina é dividida entre 2 ou mais professores a pontuação referente a carga horária daquela disciplina deverá ser dividida entre os professores. Para disciplinas compartilhadas por dois ou mais professores, que efetivamente atuam em conjunto, a pontuação será concedida integralmente para os envolvidos, desde que acompanhada de declaração do coordenador de curso. No Subgrupo I.3a pontuação será concedida com base na média das duas avaliações realizadas no ano.			
<b>Grupo II – Produção Intelectual</b>			
<b>Atividade</b>	<b>Descrição</b>	<b>Ponto por ano</b>	<b>Ponto por evento</b>
	Artigo científico publicado em revistas científicas especializadas nacionais ou internacionais. Será considerado o maior Qualis do periódico dentre as áreas de conhecimento da CAPES. Anexar a classificação atualizada Qualis da CAPES no processo. Unidade: pontos por artigo documentado		

II.1	Qualis A ou JCR acima de 1.2. Pontuação por artigo	30
	Qualis B1 e B2 ou JCR acima de 0.5 e abaixo de 1,2. Pontuação por artigo	15
	Qualis B3, B4, B5 e Qualis C ou JCR < 0,5. Pontuação por artigo	6
	Sem classificação ou local. Pontuação por artigo	2
II.2	Publicação integral de artigo em Anais de Congressos, Simpósios, Seminários e similares <sup>1</sup> , em eventos de abrangência nacional <sup>2,3</sup> . Pontuação por artigo documentado	4
II.3	Publicação integral de artigo em Anais de Congressos, Simpósios, Seminários e similares, em eventos de abrangência internacional <sup>4</sup> . Pontuação por artigo documentado	5
II.4	Publicação de artigo em Seminário de Iniciação Científica e Tecnológica ou seminários específicos (ex SEI, Sulpet, etc), de abrangência regional ou nacional. Pontuação por artigo documentado	2
II.5	Trabalho de Conclusão de Curso (TCC) concluído e disponibilizado no Portal Institucional em Acesso Aberto da UTFPR (PIAA)	2
II.6	Publicação de resumo de artigo em Anais de Congresso, Simpósios, Seminários e similares, em eventos de abrangência nacional. Pontuação por artigo documentado	2
II.7	Publicação de resumo de artigo em Anais de Congresso, Simpósios, Seminários e similares, em eventos de abrangência internacional. Pontuação por artigo documentado	3
II.8	Participação como moderador, debatedor ou coordenador de sessão em Congressos, Simpósios, Seminários, e demais eventos técnico-científicos nacionais. Pontuação por artigo documentado	2
II.9	Participação como moderador, debatedor ou coordenador de sessão em Congressos, Simpósios, Seminários, e demais eventos técnico-científicos internacionais. Pontuação por artigo documentado	5
II.10	Palestra ou minicurso proferida em Congressos, Simpósios, Seminários e Workshop de Sociedades Científicas como convidado da Organização/Comitê Técnico do evento ( <i>Invited Talk</i> ), em eventos de abrangência nacional. Pontuação por palestra documentada	2
II.11	Palestra ou minicurso proferida em Congressos, Simpósios, Seminários e Workshop de Sociedades Científicas como convidado da Organização/Comitê Técnico do evento ( <i>Invited Talk</i> ), em eventos de abrangência internacional. Pontuação por palestra documentada	5
II.12	Livro técnico/científico em língua portuguesa com ISBN. Livro(s) relevante (s) na área de avaliação do postulante podem ser pontuados pela comissão. Pontuação limitada a 20 pontos por livro	20
II.13	Livro técnico/científico em língua estrangeira com ISBN. Livro(s) relevante (s) na área de avaliação do postulante podem ser pontuados pela comissão. Pontuação limitada a 30 pontos por livro	30
II.14	Capítulo de livro técnico/científico em língua portuguesa publicado após ser submetido ao Conselho Editorial em Editora. Pontuação por capítulo	10
II.15	Capítulo de livro técnico/científico em língua estrangeira publicado após	15

	ser submetido ao Conselho Editorial em Editora. Pontuação por capítulo		
II.16	Edição ou organização de livro técnico/científico com ISBN ou Anais de Congressos de Sociedades Científicas. Pontuação por livro		10
II.17	Autoria de tradução de livro técnico/científico publicado por Editora. Pontuação por livro		10
II.18	Citações em Artigos de Periódicos indexados, anexando ao processo apenas as citações da base de dados <i>Web of Knowledge</i> (ISI) e Scopus de quaisquer artigos de sua autoria somente no interstício da avaliação. Este item está limitado a um máximo de 10 pontos. A cada 2 anos de período de avaliação considerado. Pontuação por citação por artigo		2
II.19	Inventos e demais produtos de pesquisa, com patente ou registro de software concedido. Pontuação por depósito ou registro		10
II.20	Inventos e demais produtos de pesquisa, com patente ou registro de software depositado. Pontuação por concessão		30
II.21	Produção de manual técnico e/ou didático de autoria própria ou coautoria. Pontuação por manual		10
II.22	Produção de Apostila para Disciplina da Matriz Curricular de Curso do ensino médio, técnico, graduação e pós-graduação stricto sensu da UTFPR de autoria ou coautoria própria. Pontuação por apostila		10
II.23	Relatório Técnico Demandado a UTFPR sob a forma de consultoria. Pontuação por relatório		10
II.24	Editor-Chefe de periódico indexado nas bases Web of Knowledge (ISI) e Scopus, possuidor de Fator de Impacto com base nos indicadores JCR e/ou SNIP por pelo menos 1 ano. Pontuação por periódico na função de editor-chefe		20
II.25	Editor-chefe de periódico indexado com abrangência nacional por pelo menos 1 ano. Pontuação por periódico na função de editor-chefe		10
II.26	Editor associado ou de área de periódico indexado com abrangência internacional por pelo menos 1 ano. Pontuação por periódico na função		10
II.27	Editor associado ou de área de periódico indexado com abrangência nacional por pelo menos 1 ano. Pontuação por período na função		5
II.28	Revisor de artigo científico para periódico indexado com abrangência internacional. Pontuação por revisão limitado a 10 pontos a cada 2 anos		2
II.29	Revisor de artigo científico para periódico indexado com abrangência nacional. Pontuação por consultoria limitado a 5 pontos a cada 2 anos		1
II.30	Consultor científico “ <i>ad hoc</i> ” de projeto individual de pesquisa (e.g., CNPq, CAPES, FAPs). Pontuação por consultoria limitado a 5 pontos a cada 2 anos.		1
II.31	Consultor científico “ <i>ad hoc</i> ” de projeto institucional de pesquisa (e.g., CNPq, CAPES, FAPs). Pontuação por consultoria limitado a 5 pontos a cada 2 anos		1
II.32	Membro de Conselho Científico e Editorial de revista científica indexada. Pontuação por conselho limitado a 3 pontos a cada 2 anos		1
II.33	Recebimento de comendas e premiações advindas do exercício de		20

	atividades de ensino, pesquisa e extensão concedidas por entidade distinta da UTFPR. A comissão poderá atribuir pontuações maiores para casos de excelência nacional e internacional (ex.: Comenda de Mérito Científico Nacional, Prêmio de Inovação FINEP, Prêmio Jovem Cientista, Pesquisador Emérito CNPq, Medalha da Ordem de Mérito Militar no exercício de atividades na UTFPR)		
<b>Observações para o Grupo II</b>			
<p>1) Para todos os Congressos, Simpósios, Seminários e similares listados nas Atividades de Pesquisa serão somente considerados eventos abertos com chamada de trabalhos (ou seja, que não são compostos apenas por comunicações convidadas), comitê de programa e processo de arbitragem na escolha dos artigos.</p> <p>2) Entende-se como evento de abrangência nacional aquele cuja chamada de trabalhos é restrita a comunicações/artigos de pesquisadores de um mesmo país, mesmo que realizado fora do Brasil.</p> <p>3) Não serão consideradas neste item publicações em Congressos e/ou Seminários de Iniciação Científica.</p> <p>4) Entende-se como evento de abrangência internacional aquele cuja chamada de trabalhos é dirigida à participação de pesquisadores de vários países, normalmente apoiado por Associações, Entidades e Organizações de caráter internacional, ainda que realizado no Brasil.</p>			
<b>Grupo III – Atividades de Pesquisa, de Extensão, Administrativas e de Representação e Outras Atividades</b>			
<b>Atividade</b>	<b>Descrição</b>	<b>Ponto por ano</b>	<b>Ponto por evento</b>
<b>Subgrupo III1 – Atividades de Pesquisa</b>			
III.1.1	Bolsista Produtividade em Pesquisa (PQ) do CNPQ e similares de outras agências oficiais de fomento		15
III.1.2	Coordenação de projeto com instituição de excelência internacional com produção intelectual relevante e /ou formação em dupla diplomação de doutor (ex.: PROBRAL). Pontuação por projeto por ano	15	
III.1.3	Membro do projeto com instituição de excelência internacional com produção intelectual relevante e /ou formação em dupla diplomação de doutor sob orientação do postulante. Pontuação por projeto por ano	5	
III.1.4	Liderança de projeto com instituição de excelência internacional com formação de dupla diplomação ou intercâmbio acadêmico de alunos no nível graduação. Pontuação por projeto por ano	3	
III.1.5	Coordenação de projeto com instituição de excelência internacional para formação de dupla diplomação de recursos humanos no nível de Mestrado. Pontuação por mestre orientado, comprovado por declaração da Pró-Reitoria responsável do docente líder da ação ou comprovante da agência de fomento		15
III.1.6	Coordenação de projetos de (PD&I) em parceria com outros institutos, universidades e centros de pesquisa de excelência internacional. Pontuação por projeto por ano	10	

III.1.7	Coordenação de projetos de (PD&I) em parceria com outros institutos, universidades e centros de pesquisa nacional. Pontuação por projeto por ano.	4	
III.1.8	Participação como membro de projeto de PD&I, contemplado em editais de PD&I cooperativos com instituições parceiras. Pontuação por projeto por ano, limitado a 8 pontos a cada 2 anos	2	
III.1.9	Liderança de grupo de pesquisa homologado pela UTFPR e Ativo. Pontuação por grupo de pesquisa por ano	1	
III.1.10	Liderança na captação de recursos financeiros em projetos de PD&I com instituições ou empresas parceiras. Pontuação por projeto		15
III.1.11	Contratos de transferência de tecnologia e licenciamento chancelado pela Agência de Inovação da UTFPR. Pontuação por transferência de tecnologia ou licenciamento		30
III.1.12	Desenvolvimento de produtos ou processos (produtos, protótipos e processos implantados e não patenteados, software não registrados). Não somente conceitual, deve demonstrar o aspecto de utilidade prática. Pontuação por produto ou processo		5
III.1.13	Cargo de assessoria em órgãos de fomento à pesquisa, ao ensino ou à extensão. Pontuação por cargo por ano	15	
III.1.14	Consultoria ad hoc para órgãos de fomento à pesquisa, ao ensino ou à extensão. Pontuação por consultoria		2
III.1.15	Participação em Conselhos, comitês e afins em órgãos de fomento à pesquisa, ao ensino e à extensão. Com comprovação de participação efetiva. Pontuação por conselho por ano	2,5	
III.1.16	Coordenação de projeto institucional de pesquisa financiado por PADCT, FNDCT, RHAECNPq, FINEP ou similares, executado na UTFPR. Pontuação por projeto		15
III.1.17	Participação de projeto institucional de pesquisa financiado por PADCT, FNDCT, RHAECNPq, FINEP ou similares, executado na UTFPR. Pontuação por projeto		10
III.1.18	Participação em projeto individual de pesquisa, de estudos e/ou de diagnósticos financiado por órgãos públicos e/ou privados executado na UTFPR. Pontuação por projeto		5
III.1.19	Coordenação de projeto de pesquisa financiado CNPq, CAPES, FINEP, Fundação Araucária ou outros órgãos de fomento oficiais, executado na UTFPR. Pontuação por projeto		10
III.1.20	Participação de projeto de pesquisa financiado CNPq, CAPES, FINEP, Fundação Araucária ou outros órgãos de fomento oficiais, executado na UTFPR. Pontuação por projeto		5
III.1.21	Participação em projetos de desenvolvimento institucional (ex.:INSERIR). Pontuação por projeto em que participou com mais de 1 anos de atuação		10
III.1.22	Líder na captação e execução de recursos para projetos de desenvolvimento institucional (infraestrutura CT-INFRA/Pro-Equipamentos, Capacitação: MCTI/MEC e assemelhados). Pontuação por projeto aprovado e executado		30
III.1.23	Coordenação de projeto individual de pesquisa, de estudos e/ou de diagnósticos financiado por órgãos públicos e/ou privados, executado na UTFPR. Pontuação por coordenação de projeto		10



<b>Subgrupo III2 - Atividades de Extensão</b>			
III.2.1	Bolsista em Desenvolvimento Tecnológico e Extensão Inovadora (DT) do CNPQ e similares de outras agências oficiais de fomento		15
III.2.2	Coordenação não remunerada de Curso de Extensão. Com comprovação de atividade não remunerada. Pontuação por coordenação		5
III.2.3	Hora-aula não remunerada de Curso de Extensão (Até 30 horas). Com comprovação de atividade não remunerada. Pontuação por capacitação.		2
III.2.4	Hora-aula não remunerada de Curso de Extensão (Mais de 30 horas). Com comprovação de atividade não remunerada. Pontuação por participação		3
III.2.5	Palestras e conferências, não científicas, a convite. <i>Palestras e conferências de caráter científico devem ser relacionadas no item II.</i> Pontuação por palestra		2
III.2.6	Divulgação técnica em televisão, rádio, revista ou jornal (os pontos serão divididos entre os autores). Pontuação por participação		2
III.2.7	Coordenação de programa de extensão institucional apoiado pela UTFPR (comunitário, esportivo ou similar). Pontuação por ano	4	
III.2.8	Coordenação de projeto de extensão institucional apoiado pela UTFPR (comunitário, esportivo ou similar). Pontuação por ano	3	
III.2.9	Membro de programa de extensão institucional apoiado pela UTFPR (comunitário, esportivo ou similar). Pontuação por ano, limitado a 8 pontos a cada 2 anos	2	
III.2.10	Coordenação de projeto institucional de extensão financiado por órgão público ou privado (ex.: MCTI, MDS, CNPq, FA). Pontuação por projeto		15
III.2.11	Membro de projeto institucional de extensão financiado por órgão público ou privado. Pontuação por projeto		10
III.2.12	Participação em projeto individual de extensão financiado por órgão público ou privado. Pontuação por projeto		5
III.2.13	Orientação de estagiário/bolsista de projeto de extensão registrado na DIREC. Pontuação por ano de orientação	1	
III.2.14	Organização e coordenação de dia de campo, exposição, visita/reunião técnica, registrados na DIREC. Pontuação por atividade		1
III.2.15	Membro de comissão organizadora de dia de campo, exposição, visita/reunião técnica, mesa redonda, feiras, fóruns, atividades culturais, dias da indústria, jogos regionais da UTFPR, entre outras, registrados na DIREC. Pontuação por atividade		1
III.2.16	Orientação de bolsista e/ou voluntário de projeto de extensão não registrado na DIREC. Pontuação por ano de orientação	0,5	
III.2.17	Organização e coordenação de dia de campo, exposição, visita/reunião técnica, não registrados na DIREC. Pontuação por atividade		0,5
III.2.18	Membro de comissão organizadora de dia de campo, exposição, visita/reunião técnica, mesa redonda, feiras, fóruns, atividades culturais, dias da indústria, jogos regionais da UTFPR, entre outras, não registrados na DIREC. Pontuação por atividade		1
III.2.19	Membro de Comissão Técnica ou de Programas de eventos técnicos ou científicos, devidamente comprovados pela Diretoria competente ou		3

	Associação de Classe		
III.2.2 0	Participação em Organismos Internacionais reconhecidos (UNESCO, ONU, FAO, IEEE, IEE, e similares) como coordenador de Comissão ou de Área		5
III.2.2 1	Trabalhos técnicos e consultorias com parecer ou laudo técnico emitido e devidamente registrado no DEPET (curto prazo)		10
III.2.2 2	Trabalhos técnicos e consultorias com parecer ou laudo técnico emitido e não registrado no DEPET mas documentados em órgãos institucionais (ex.: departamento) (curto prazo)		8
III.2.2 3	Coordenação de projetos de extensão tecnológica com instituições parceiras. Pontuação por projeto		10
III.2.2 4	Membro da comissão organizadora de congressos, <i>workshops</i> , seminários, mostras em âmbito internacional. Pontuação por evento		15
III.2.2 5	Membro da comissão organizadora de congressos, <i>workshops</i> , seminários, mostras em âmbito nacional. Pontuação por evento		10
III.2.2 6	Membro da comissão organizadora de congressos, <i>workshops</i> , seminários, mostras em âmbito regional ou local. Pontuação por evento		4
III.2.2 7	Membro de comissão de avaliação e premiação (projetos, trabalhos científicos, trabalhos artísticos e culturais) em nível nacional e internacional. Unidades: Pontuação por evento		10
III.2.2 8	Coordenação de projeto com empresa, credenciado pela DIREC. Pontuação por programa		10
III.2.2 9	Coordenação de programa de extensão individual registrado na DIREC. Pontuação por programa		10
III.2.3 0	Membro de programa de extensão individual registrado na DIREC. Pontuação por programa		5
III.2.3 1	Coordenador de projeto de extensão individual registrado na DIREC. Pontuação por projeto		10
III.2.3 2	Participação em projeto individual de extensão registrado na DIREC. Pontuação por projeto		2
III.2.3 3	Projeto de Curso de Extensão aprovado pelo Conselho de Ensino da UTFPR. Pontuação por coordenação de projeto		5
III.2.3 4	Orientação de alunos inscritos em Projetos de Extensão registrados na DIREC. Pontuação por aluno por ano	1	
III.2.3 5	Coorientação de alunos inscritos em Projetos de Extensão registrados na DIREC. Pontuação por aluno por ano	0,5	
III.2.3 6	Orientação de alunos em empresas pré-incubadas no Hotel Tecnológico. Pontuação por aluno por ano	1	
<b>Subgrupo III3–Atividades de Administrativas e de Representação</b>			
III.3.1	Membro do Conselho Universitário	5	
III.3.2	Membro do Conselho de Pesquisa e Pós-Graduação	4	
III.3.3	Membro do Conselho de Relações Empresariais e Comunitárias	4	
III.3.4	Membro do Conselho de Graduação e Educação Profissional	4	
III.3.5	Membro do Conselho de Planejamento e Administração	4	

III.3.6	Cargo de Direção – Reitor e Vice-Reitor	30	
III.3.7	Cargo de Direção – Pró-Reitor e Diretores de Gestão da Reitoria	25	
III.3.8	Cargos de Direção – Diretores Gerais de Câmpus	25	
III.3.9	Cargo de Direção – Adjuntos de Pró-Reitores, Diretores de Área nas Pró-Reitorias e nos Câmpus	20	
III.3.10	Assessoria de Cargos previstos no regimento da UTFPR. A pontuação será equivalente a 60 % da pontuação anual do cargo assessorado. (0,6*Cargo Assessorado)	0,6*C	
III.3.11	Cargo de Ouvidoria Institucional e do Câmpus	20	
III.3.12	Chefe de Departamento Acadêmico	20	
III.3.13	Coordenador de Curso de Graduação, Técnico e Pós-Graduação <i>Stricto Sensu</i>	20	
III.3.14	Coordenador ou Chefe de órgãos vinculados a gestão de Câmpus e Pró-Reitorias previstos no regimento	20	
III.3.15	Chefe de gabinete de Reitoria	20	
III.3.16	Chefe de Auditor da UTFPR	20	
III.3.17	Relator de Processos do CEDA da UTFPR ou processo equivalente de outras IFES		4
III.3.18	Presidente de Órgãos ou Comissões Permanentes de Apoio (ex.: CPPD, CPA, NPPD, Comissão Professor Associado, Comissão Professor Titular ou similares)	7	
III.3.19	Membro de Órgãos ou Comissões Permanentes de Apoio (ex.:CPPD, CPA, NPPD, Comissão Professor Titular ou similares)	5	
III.3.20	Responsável pelas atividades de Estágio, Responsável pelas Atividades Complementares e Responsável pelo Trabalho de Conclusão de Curso dos cursos técnicos e de graduação	10	
III.3.21	Responsável pelos laboratórios, Chefes de grupo de disciplinas e Assessor de Coordenação de curso	10	
III.3.22	Membro do Colegiado de Curso, do Conselho Departamental ou do Núcleo Docente Estruturante	2	
III.3.23	Presidência de Comissão designada por ato da Administração da UTFPR. Pontuação por portaria		3
III.3.24	Membros de Comissão ou atribuição de função designada por ato da Administração da UTFPR.Pontuação por portaria		2
III.3.25	Participação, como representante da UTFPR, designado por portaria, em comitês/equipes de trabalho em órgãos como MEC, CAPES, FINEP, CNPq, CREA e outros conselhos profissionais.Pontuação por portaria		5
III.3.26	Cargo de Direção. Diretor Executivo da FUNTEF	25	
III.3.27	Membro do Conselho Deliberativo ou Fiscal da FUNTEF	15	
III.3.28	Presidência da Associação de Servidores da UTFPR	15	

III.3.2 9	Membro eletivo de diretoria de associação de classe e/ou esportiva vinculada à UTFPR	4	
III.3.3 0	Presidência do sindicato docente (local ou nacional)	15	
III.3.3 1	Membro da direção do sindicato docente (local ou nacional)	3	
III.3.3 2	Coordenação de núcleo de inovação tecnológica	15	
III.3.3 3	Exercer função administrativa, não prevista no Regimento da UTFPR comprovado mediante portaria	5	
III.3.3 4	Exercer função administrativa, prevista no Regimento da UTFPR, com função ou cargo diferente dos listados neste Subgrupo	8	
<b>Subgrupo III4–Outras Atividades</b>			
III.4.1	Conclusão de Curso de Licenciatura na área de atuação durante a carreira docente na UTFPR		10
III.4.2	Conclusão de Curso de Aperfeiçoamento na área de atuação durante a carreira docente na UTFPR		5
III.4.3	Conclusão de Curso de curta duração ( <i>workshops</i> , seminários, mostras, jornadas, treinamentos)		1
III.4.4	Formação: participação em eventos técnicos científicos com certificado		1
III.4.5	Participação em missão de trabalho internacional		2
III.4.6	Participação em missão de trabalho nacional		2
III.4.7	Pós-doutorado em programa de excelência internacional no exterior e no país (nível 6 e 7 CAPES)		25
III.4.8	Pós-doutorado em programa de excelência nacional		20
III.4.9	Pós-doutorado em outros programas		15
III.4.1 0	Membro de banca de tese de doutorado (exceto orientador e coorientador)		5
III.4.11	Membro de banca de qualificação de doutorado (exceto orientador e coorientador)		2,5
III.4.1 2	Membro de banca de dissertação de mestrado (exceto orientador e coorientador)		3
III.4.1 3	Membro de banca de qualificação da dissertação de mestrado (exceto orientador e coorientador)		1,5
III.4.1 4	Membro de banca de concurso público para provimento de cargo de Professor efetivo		5
III.4.1 5	Presidência de Associação Científica de abrangência nacional ou internacional		1,5
III.4.1 6	Membro de diretoria (exceto Presidência) de Associação Científica de abrangência nacional ou internacional		1
III.4.1 7	Membro de banca de concurso público para provimento de cargo de Professor Substituto		2
III.4.1 8	Membro de Comissão designada pelo MEC para avaliação das condições de criação, credenciamento e recredenciamento de cursos		3

III.4.1 9	Membro de Comissão designada pelo MEC para avaliação institucional		3
III.4.2 0	Membro de Banca (exceto orientador e coorientador) de Avaliação de Estágio ou de trabalho de final de curso de graduação ou de especialização (não remunerado)		1
III.4.2 1	Implementação ou execução de projeto didático-pedagógico no curso técnico, na graduação e na pós-graduação stricto sensu para melhoria e desenvolvimento de curso		3
III.4.2 2	Consultoria para Organismos Internacionais reconhecidos(UNESCO, ONU, FAO, IEEEE, IEE, e similares). Pontuação por consultoria		2

Fonte: Adaptado de UTFPR (2018)

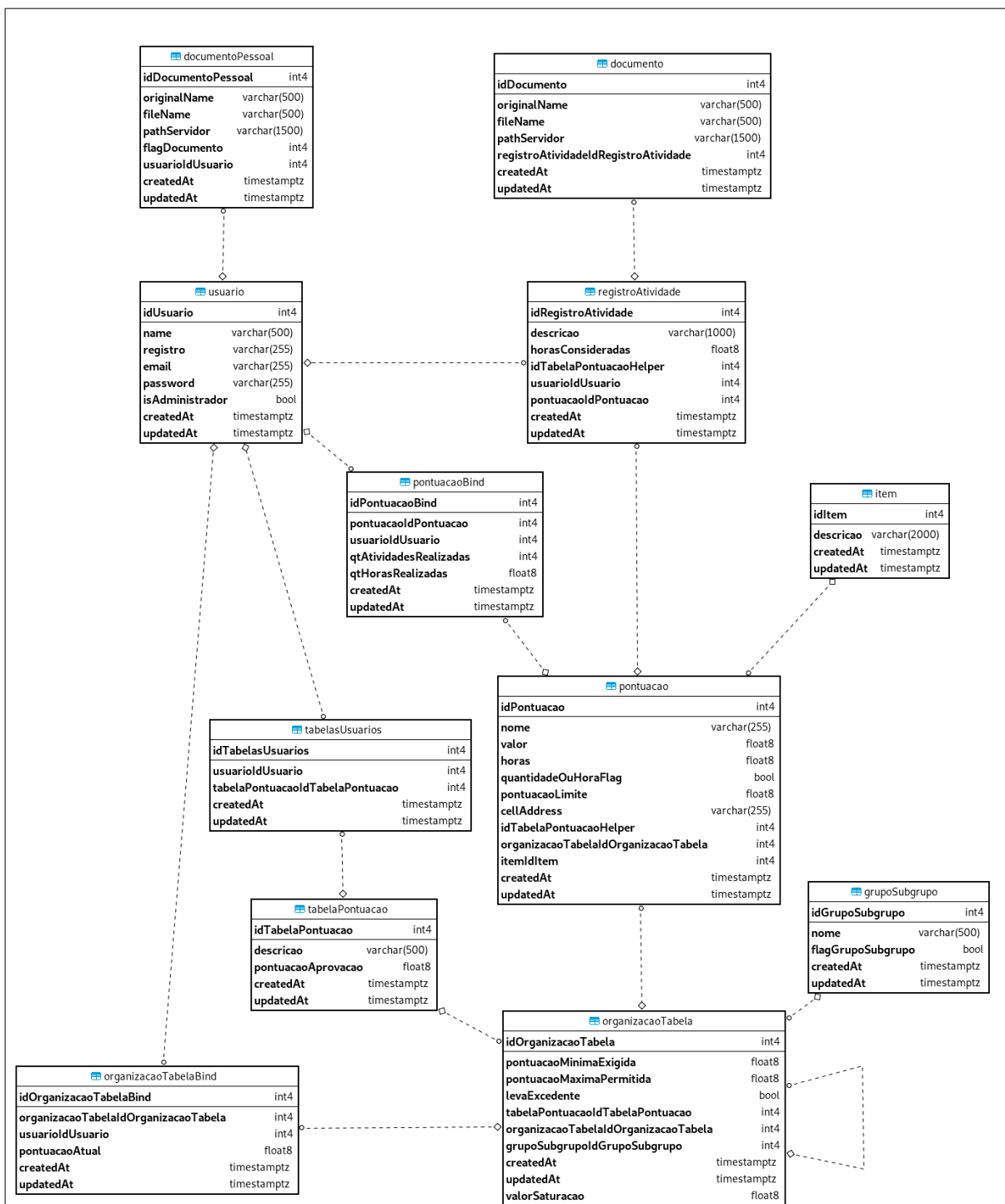
## **APÊNDICE B - Diagrama de Caso de Uso**



Fonte: Autoria própria (2021)

## **APÊNDICE C - Diagrama de Entidade-Relacionamento**





Fonte: Autoria própria (2021)

**APÊNDICE D** - Código responsável por preencher o relatório de atividades

```

1 const originalName = documento.originalname;
2 const fileName = Date.now() + '-' + documento.originalname;
3 const directoryPath = path.join(__dirname, '../documentosStorage');
4 const pathServidor = directoryPath + '/' + fileName;
5
6 fs.appendFile(pathServidor, documento.buffer, async function(err) {
7   if (err) throw err;
8   console.log('Saved!');
9
10  await pontuacao.findAll({
11    where: {
12      idTabelaPontuacaoHelper: idTabelaPontuacao
13    },
14    include: [{
15      model: pontuacaoBind,
16      where: {
17        usuarioldUsuario: idUsuario
18      }},
19    {
20      model: item
21    }
22  ]
23 })
24 .then(async result => {
25   const workbook = new ExcelJS.Workbook();
26   await workbook.xlsx.readFile(pathServidor);
27   const worksheet = workbook.worksheets[0];
28
29   result.forEach(pontuacaoData => {
30     const cellAddress = pontuacaoData.dataValues.cellAddress;
31     const cell = worksheet.getCell(cellAddress);
32
33     if(pontuacaoData.dataValues.quantidadeOuHoraFlag === true) {
34       if(
35         (pontuacaoData.dataValues.valor *
36
37         (pontuacaoData.dataValues.pontuacaoBinds[0].dataValues.qtHorasRealizadas/
38         pontuacaoData.dataValues.horas)
39         ) > 0)
40         {
41           var newValue =
42           (pontuacaoData.dataValues.pontuacaoBinds[0].dataValues.qtHorasRealizadas/
43           pontuacaoData.dataValues.horas);
44           cell.value = parseFloat(newValue);
45         }
46       }
47     }
48   });
49 }

```

```
46     else {
47         if(
48             (pontuacaoData.dataValues.valor *
49             pontuacaoData.dataValues.pontuacaoBinds[0].dataValues.qtAtividadesRealizadas)
50             > 0)
51             {
52                 var newValue =
53                 pontuacaoData.dataValues.pontuacaoBinds[0].dataValues.qtAtividadesRealizadas;
54                 cell.value = parseFloat(newValue);
55             }
56         }
57     })
58
59     await workbook.xlsx.writeFile(pathServidor);
60     res.download(pathServidor, function (err) {
61         if (err) {
62             console.log('---ERR CALLBACK---');
63             console.log(err);
64         }
65         else {
66             fs.unlink(pathServidor, function() {
67                 if (err) throw err;
68                 console.log('File deleted!');
69             })
70         }
71     });
72 })
73 .catch(e => {
74     console.log(e);
75     res.status(500).send(e);
76 })
77 })
```

Fonte: Autoria própria (2021)