



UNIVERSIDADE FEDERAL DO CEARÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

GABRIEL DE SOUZA NOGUEIRA DA SILVA

SISTEMA DE AUTOAVALIAÇÃO - PPGEEC/UFC

SOBRAL

2021

GABRIEL DE SOUZA NOGUEIRA DA SILVA

SISTEMA DE AUTOAVALIAÇÃO - PPGEEC/UFC

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia da Computação.

Orientador: Prof. Dr. Iális Cavalcante de Paula Júnior

SOBRAL

2021

GABRIEL DE SOUZA NOGUEIRA DA SILVA

SISTEMA DE AUTOAVALIAÇÃO - PPGEEC/UFC

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia da Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Iális Cavalcante de Paula
Júnior (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. XXXXXXX XXXXXX XXXXXXX
Universidade do Membro da Banca Dois (SIGLA)

Prof. Dr. XXXXXXX XXXXXX XXXXXXX
Universidade do Membro da Banca Três (SIGLA)

Prof. Dr. XXXXXXX XXXXXX XXXXXXX
Universidade do Membro da Banca Quatro (SIGLA)

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

AGRADECIMENTOS

Ao Prof. Dr. Iális Cavalcante de Paula Júnior por me orientar em meu trabalho de conclusão de curso.

Aos meus amigos e colegas de faculdade, pelo apoio durante os estudos realizados durante a graduação.

Aos meus pais e irmãos, que nos momentos de minha ausência dedicados ao estudo superior, sempre fizeram entender que o futuro é feito a partir da constante dedicação no presente!

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

Ao aluno Thiago Nascimento do curso de ciência da computação da Universidade Estadual do Ceará que elaborou o *template* do qual este trabalho foi adaptado para Universidade Federal do Ceará.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

RESUMO

Ter a oportunidade de colher *feedbacks* dos participantes de um programa de pós-graduação é fundamental para seu sucesso e sobrevivência. A melhor forma conhecida atualmente para isso está na forma de autoavaliação tanto pessoal quanto do programa. E para que isso seja possível, é necessária a existência de uma ferramenta que torne possível que esses *feedbacks* sejam colhidos de forma facilitada, oferecendo ao usuário uma interface clara e simples de se utilizar. O objetivo deste trabalho foi desenvolver uma *Application Programming Interface* (API) bem como um painel administrativo que auxilie na obtenção e um futuro processamento dos dados coletados nos *feedbacks*, tarefa realizada atualmente por uma ferramenta com uma série de limitações e que precisa ser substituída. Para apoiar todas as etapas do projeto, utilizou-se a metodologia ágil do Scrum, que auxiliou em todas as etapas, desde o levantamento de requisitos até as fases de implantação do sistema. Esse novo sistema possui vantagens sobre o sistema atual, tais como: maior facilidade no controle de acesso dos formulários de *feedback*, possui uma interface Web de fácil manipulação, altamente adaptável as necessidades do programa e utiliza ferramentas livres de desenvolvimento, como o *Visual Studio Code*, linguagem e programação Javascript e banco de dados PostgreSQL. Com isso, o principal resultado desse trabalho é a API desenvolvida juntamente com o painel administrativo para manipulação e visualização dos dados cadastrados.

Palavras-chave: API, Linguagem de programação Javascript; Autoavaliação; Banco de dados PostgreSQL

ABSTRACT

Having the opportunity to gather *feedback* from graduate programs participants is critical to its success and survival. The best way currently known for this is in the form of both personal and program self-assessment. And for this to be possible, it is necessary to have a tool that makes it possible for these *feedbacks* to be collected easily, offering the user a clear and simple interface to use. The objective of this work was to develop an Application Programming Interface as well as an dashboard that helps in obtaining and further processing the data collected in *feedbacks*, a task currently performed by a tool with a series of limitations and that needs to be replaced. the agile methodology of Scrum was used, which helped in all stages, from requirements gathering to the system implementation phases, to support all stages of the project. This new system has advantages over the current system, such as greater ease in controlling access to feedback forms, it has an easy-to-handle Web interface, highly adaptable to the program's needs and uses free development tools, such as Visual Studio. Code, Javascript programming language, and PostgreSQL database. With that, the main result of this work is the API developed together with the dashboard for manipulation and visualization of the registered data.

Keywords: API, Javascript language and programming; Self-evaluation; PostgreSQL Database

LISTA DE FIGURAS

Figura 1 – Processo de autoavaliação	16
Figura 2 – Modelagem do banco de dados	26
Figura 3 – Interface da documentação interativa	28
Figura 4 – Interface da documentação interativa, seção de autenticação	29
Figura 5 – Interface da documentação interativa, seção de autenticação	30
Figura 6 – Interface da documentação interativa, seção de autenticação	31
Figura 7 – Interface da documentação interativa, seção de autenticação	32
Figura 8 – Interface do painel administrativo - Login	32
Figura 9 – Interface do painel administrativo	33
Figura 10 – Interface do painel administrativo - tela de criação de admins	33
Figura 11 – Interface do painel administrativo - tela de criação de fichas	34

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
PPGEEC	Programa de Pós-graduação em Engenharia Elétrica e Computação
SQL	<i>Structured Query Language</i>
CSV	<i>Comma-separated-values</i>
SGBD	<i>Sistema de Gerenciamento de Banco de Dados</i>
RDBMS	<i>Relational Database Management System</i>
RDSMS	<i>Relational Data Stream Management System</i>
DQL	<i>Data Query Language</i>
DDL	<i>Data Definition Language</i>
DCL	<i>Data Control Language</i>
DML	<i>Data Manipulation Language</i>
ANSI	<i>American National Standards Institute</i>
ISO	<i>International Organization for Standardization</i>
DOM	<i>Document Object Model</i>
ORM	<i>Object-Relational Mapping</i>
JSON	<i>JavaScript Object Notation</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Justificativa	12
1.2	Objetivo Geral	13
1.3	Objetivos Específicos	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Resolução N° 1/2021/CMEEC/CUFCSOBRAL/REITORIA do Programa de Pós-graduação em Engenharia Elétrica e Computação (PPGEEC)	14
2.2	Metodologia scrum	16
2.2.1	<i>O que é?</i>	16
2.2.2	<i>Pilares do Scrum</i>	16
2.2.3	<i>Papéis do Scrum</i>	17
2.2.4	<i>Como funciona o Scrum</i>	17
2.2.5	<i>Ferramentas para auxiliar o Scrum</i>	18
2.3	Arquitetura <i>Client-server</i> (cliente-servidor)	18
2.4	Padrão de projeto <i>Model-view-controller</i> (MVC)	19
2.5	<i>Backend</i>	19
2.6	API Rest	20
2.6.1	<i>API</i>	20
2.6.2	<i>Rest</i>	20
2.7	Linguagem javascript	22
2.8	Node.js	22
2.9	Linguagem <i>Structured Query Language</i> (SQL)	22
2.10	PostgreSQL	23
2.11	ReactJs	23
2.11.1	<i>React-Admin</i>	24
3	METODOLOGIA	25
3.1	Uso da metodologia Scrum	25
3.2	Levantamento de requisitos com a coordenação do programa	25
3.3	Desenvolvimento do <i>Backend</i>	25
3.3.1	<i>Uso Banco de dados</i>	27

3.4	Desenvolvimento do <i>Frontend</i> de administração	27
4	RESULTADOS	28
4.1	Resultados do desenvolvimento do <i>Backend</i>	28
4.2	Resultados do desenvolvimento do painel administrativo	29
5	CONSIDERAÇÕES FINAIS	35
	REFERÊNCIAS	36
	ANEXOS	37
	ANEXO A–RESOLUÇÃO DO PPGEEC	38

1 INTRODUÇÃO

Para melhor compreensão desse trabalho devemos entender que desde a implantação da pós-graduação no Brasil nos moldes definidos pelo Parecer CFE 977/1965, que constava a instauração do sistema de cursos de pós-graduação e teria o objetivo de formar professorado competente, preparar pesquisadores e treinar técnicos e trabalhadores intelectuais, tendo em vista: (i) atender a expansão do ensino superior e elevar seus níveis de qualidade, (ii) desenvolver a pesquisa científica e (iii) atender ao desenvolvimento nacional em todos setores da economia e da sociedade (BRASIL, 1965), a pós-graduação *stricto sensu* avançou no sentido do seu crescimento numérico (CAPES, 2019).

Passados mais de 50 anos, a pós-graduação avançou, cresceu e se desenvolveu. Entre 1998 e 2017, o número de cursos de pós-graduação *stricto sensu* cresceu mais de 200% e o número de mestres e doutores no país aumentou significativamente (quase 23 mil doutores e mais de 64 mil mestres titulados apenas em 2018) (GEOCAPES, 2018). Com esse crescimento, ter uma forma efetiva de avaliação externa¹ assegura padrões básicos, o que é importante em um país continental como o nosso, mas tem suas limitações. Uma delas é o fato de não ser formativa, em que os que estão no processo se envolvam também na solução dos problemas identificados. Neste sentido, a autoavaliação favorece a construção da identidade, heterogeneidade e envolvimento dos programas avaliados, para além dos padrões mínimos garantidos pela avaliação externa. (CAPES, 2019).

O PPGEEC é o programa de pós-graduação em engenharia elétrica e computação ofertado pela Universidade Federal do Ceara. Com isso, espera-se que o desenvolvimento da autoavaliação no PPGEEC induza um processo de amadurecimento de pesquisadores e discentes no sentido de responsabilização, colaboração e engajamento na melhoria do *stricto sensu*, da qualidade da formação de pesquisadores brasileiros e, principalmente, da prática democrática (LEITE, 2020).

Diante de toda informação apresentada, pode-se perceber que um software capaz de facilitar essa autoavaliação traz benefícios tanto para o programa que o adota, quanto para os docentes e discente que o utilizam, podendo usufruir de *feedbacks* para o melhor funcionamento.

¹ Conhecida como avaliação de desempenho, é realizada por agente externo à instituição, geralmente aplicada em larga escala. É uma ferramenta que fornece elementos para a formulação e o monitoramento de políticas públicas, bem como o redirecionamento de práticas pedagógicas. As avaliações externas têm como característica uma matriz de avaliação e o emprego de provas padronizadas, que permitem o cumprimento do direito à aprendizagem e a interpretação dos resultados para efetuar comparação entre redes e instituições (JOVENSGENIOS, 2020).

Sendo assim, o desenvolvimento de uma aplicação capaz de fazer com que os docentes, discentes e técnicos administrativos possam tanto avaliar o programa como um todo, quanto disciplinas e o desempenho dos docentes e se autoavaliar, se torna de suma importância para a evolução e criação de um programa melhor para todos os envolvidos.

Sendo assim, este trabalho propõe um desenvolvimento de um *software* capaz de suprir as necessidades do programa. Com isso, partindo da premissa dos envolvidos no programa de pós-graduação terem a possibilidade de se autoavaliarem e avaliarem outros aspectos do programa, temos uma aplicação digital capaz de processar e armazenar os dados de todas essas avaliações para que possa ser aplicado uma análise gráfica e/ou uma análise mais profunda visando a correlação de respostas para inferir sobre a satisfação do público alvo, fazendo com que possamos achar pontos de melhoria no programa e transformá-lo em um programa melhor. Portanto, para o desenvolvimento, utilizamos técnicas de metodologias ágeis. Dentre todas as técnicas disponíveis na atualidade decidimos optar pela metodologia do *scrum* pois se adaptava melhor as nossas necessidades. Com relação a modelagem do banco de dados, suportar toda a carga de informações optamos por usar um banco de dados relacional, *postgres* que utiliza a linguagem SQL para a construção do banco de dados, pois para que possamos relacionar os dados por estudante ou entidade avaliada o uso de ferramentas não relacionais traz uma complexidade maior para o projeto. Por fim para dar suporte a todo o software utilizamos o formato de API Rest. Para uma maior facilidade de integração com diversos *frontends* a arquitetura utilizada foi a de *Client-server* (cliente-servidor) em conjunto com o padrão de projeto *Model-view-controller* (MVC), porém a parte de *view* é representada pelo contrato dos dados que são enviados, pois estamos construindo uma API Rest. O sistema possibilita a total manipulação e visualização dos dados criados, bem como a exportação desses dados via *Comma-separated-values* (CSV) para um eventual processamento.

O trabalho está organizado da seguinte forma: na seção dois consta uma fundamentação teórica com o objetivo de fundamentar os assuntos discutidos; na seção três é apresentado a metodologia usada para desenvolver o sistema de autoavaliação proposto; e, por fim, são apresentados alguns resultados, considerações finais e recomendações para trabalhos futuros.

1.1 Justificativa

Para o melhor funcionamento do programa um sistema eficiente e proprietário se torna relevante. Atualmente, utiliza-se os formulários do Google para colher esses *feedbacks*.

Tal sistema possui alguns pontos falhos tais como: é uma solução generalizada com a proposta de cobrir muitos cenários, é de propriedade do Google, ou seja caso um dia o Google não veja sentido na existência da aplicação o PPGEEC perde todos os dados e a possibilidade de circular novos formulários pela plataforma.

Diante de toda informação apresentada, pode-se perceber que a tecnologia proposta traz benefícios tanto para o programa que o adota, quanto para os docentes e discentes que o utilizam, podendo usufruir de *feedbacks* para o melhor funcionamento. Sendo assim, o desenvolvimento de uma aplicação capaz de fazer com que os docente, discentes e técnicos administrativos possam tanto avaliar o programa como um todo, quanto disciplinas e o desempenho dos docentes e se autoavaliar, se torna de suma importância para a evolução e criação de um programa melhor para todos os envolvidos.

1.2 Objetivo Geral

Dessa maneira, o objetivo geral deste trabalho foi iniciar o desenvolvimento de uma aplicação que substitua os formulários criados na plataforma do Google Forms e que centralize todos os dados dentro da própria universidade, evitando depender de programas ou armazenamento de terceiros.

1.3 Objetivos Específicos

Para que a aplicação seja bem aceita dentro da universidade, ela deve atender à todas as funcionalidades existentes no Google Forms e ainda trazer outros benefícios, como uma interface mais intuitiva, maior facilidade de uso e maior facilidade de extração dos dados para futuro processamento. Diante disto, um bom levantamento de requisitos é primordial para o sucesso. Sendo assim, para atender todas as partes interessadas no projeto levantaram-se os seguintes objetivos específicos:

- Organizar dados e fichas de avaliação do sistema de autoavaliação do PPGEEC.
- Desenvolver uma API capaz de fornecer uma comunicação eficiente entre os dados e quem o consome.
- Desenvolver um painel administrativo capaz de manipular os dados gerados pelas respostas das fichas avaliativas.

2 FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento do projeto apresentado, foi aplicada a metodologia do *scrum* para nortear o desenvolvimento do projeto. Outrossim, utilizamos o formato de API Rest com a arquitetura de *Client-server* (cliente-servidor) em conjunto com o padrão de projeto *Model-view-controller* (MVC). Ademais, a linguagem de programação utilizada foi o javascript com a biblioteca Node.js para o desenvolvimento do *Backend* da aplicação, bem como a biblioteca ReactJs e do framework React-Admin para a construção de um painel administrativo (*Frontend*). Para a construção do banco de dados foi utilizada a linguagem SQL, aplicada em um *Sistema de Gerenciamento de Banco de Dados* (SGBD) chamado PostgreSQL. Além disso, para solucionar de forma eficiente o problema atual será utilizado as orientações presentes na RESOLUÇÃO Nº 1/2021/CMEEC/CUFCSOBRAL/REITORIA (PPGEEC/UFC, 2021).

2.1 Resolução Nº 1/2021/CMEEC/CUFCSOBRAL/REITORIA do PPGEEC

O presente Regulamento tem por objetivo normalizar o conjunto de atividades relacionadas com a Autoavaliação Periódica do PPGEEC/UFC. Com isso, esta resolução tem os seguintes objetivos: avaliar o PPGEEC de maneira sistemática e periódica promovendo o autoajuste, definir um processo de busca por melhoramentos contínuos, proporcionar sintonia com as políticas de avaliação interna e externa definidas pela CAPES e assegurar canais de comunicação a todos os atores envolvidos com o mestrado.

O Plano de Autoavaliação do PPGEEC/UFC consiste em um conjunto de políticas, diretrizes e instrumentos, inter-relacionados e sistêmicos, com o objetivo de implementar o processo de autoavaliação do curso e permitir a busca por melhoramentos contínuos.

Portanto, a autoavaliação deve ser processo de construção coletiva e contínua, que utiliza diversos recursos metodológicos e instrumentos, objetivando retratar a complexidade e a riqueza do PPGEEC, e efetivar mudanças, a partir das fragilidades e potencialidades detectadas. O processo de autoavaliação compreende as seguintes etapas:

- i Preparação: constituição da Comissão Autoavaliação (CAA) do PPGEEC; elaboração do Plano de Autoavaliação; aprovação do Plano de Autoavaliação pelo colegiado do PPGEEC; e, sensibilização dos diversos atores envolvidos (gestores, docentes, técnicos e discentes);
- ii Desenvolvimento: implementação do Plano de Autoavaliação, com planejamento de ações, coleta de dados e análise de resultados;

iii Consolidação do processo: elaboração de relatórios críticos, divulgação de resultados (para docentes, discentes e técnicos administrativos), acompanhamento à efetivação de mudanças e envio pela Coordenação à CAPES dos procedimentos de autoavaliação.

Dentro desse processo os seguintes aspectos são avaliados:

1. Recursos humanos: docentes e secretaria
2. Estrutura curricular
3. Infraestrutura
4. Interações do Programa: com a graduação, outras instituições e sociedade civil
5. Discentes: disciplinas e orientações
6. Egressos

A Comissão de Autoavaliação (CAA-PPGEEC) será constituída por, no mínimo, 5 (cinco) membros, com total autonomia sobre a matéria, sendo assim composta por:

1. 01 (um) coordenador (membro do corpo docente);
2. 01 (um) coordenador-adjunto (membro do corpo docente);
3. 01 (um) técnico-administrativo;
4. 01 (um) representante do corpo discente;
5. 01 (um) representante egresso;
6. Outros professores quando se julgar necessário.

Todos os membros serão indicados pelo Colegiado do PPGEEC, pelo período de mandato de 01 (um) ano, sendo renovável para até 04 (quatro) anos, correspondente à avaliação quadrienal da CAPES.

Para que essa avaliação seja feita, será usado como instrumento de avaliação questionários (Fichas avaliativas). Essas fichas serão compostas por perguntas que foram elaboradas pelo colegiado e as opções seguem uma escala da seguinte maneira: Ótimo / Bom / Regular / Ruim / Péssimo/ Não conheço / Não se aplica, para as perguntas objetivas. Dessa forma, A Figura 1 demonstra como será executada a autoavaliação. Dado que as fichas foram criadas, demonstrado no canto superior esquerdo da Figura 1, temos o passo 1 que seria as fichas sendo preenchidas pelos docentes, discentes e técnicos administrativos do programa. No passo 2 temos os dados gerados pelas respostas obtidas no passo 1 sendo devidamente armazenados para um futuro processamento. No passo 3 temos o processamento dos dados armazenado gerando relatórios e gráficos. No passo 4, temos esses relatórios e gráficos sendo apreciados pelo público interessado no programa. Para ver em detalhes quais fichas foram abortadas e as perguntas

contidas nessas fichas vá para o anexo A.

Figura 1 – Processo de autoavaliação



Fonte: Elaborado pelo autor.

2.2 Metodologia scrum

2.2.1 O que é?

O método Scrum faz parte de um conjunto de práticas, conhecidas como metodologias ágeis, cujo objetivo é otimizar o tempo de entrega do trabalho e a qualidade do produto ou serviço que está sendo desenvolvido. O desenvolvimento da ferramenta Scrum se baseia na definição de prioridades e em um acompanhamento periódico das etapas do projeto, sempre com foco nas necessidades do cliente. Entre as principais diferenças de uma metodologia tradicional para o Scrum, podemos destacar uma estrutura flexível, aberta às mudanças e uma orientação voltada às pessoas. Dessa forma, ao executar projetos com o auxílio do Scrum, a equipe evita longos ciclos de desenvolvimento, prevê riscos e diminui as chances de entregar um produto ou serviço que não condiz com as expectativas do cliente (EJMC, 2021).

2.2.2 Pilares do Scrum

Para que o método funcione de maneira eficaz há 3 pilares que devem ser aplicados: transparência, inspeção e adaptação.

1. **Transparência:** este pilar é essencial no Scrum, sendo garantido através de reuniões.

Esses eventos ocorrem com uma periodicidade estabelecida pela equipe e trazem uma percepção do projeto como um todo. Portanto, todos os integrantes da equipe conseguem ter um panorama geral do andamento do projeto.

2. **Inspeção:** a inspeção nesse método tem como funcionalidade detectar possíveis problemas antes que eles aconteçam. Alguns meios para promover esse pilar são o estabelecimento de prioridades, através de uma lista de funcionalidades, e a coleta de *feedback*. Ao detectar esses problemas de antemão, a equipe evita a necessidade de realizar um retrabalho devido algum erro ocorrido, economizando tempo e recursos do projeto.
3. **Adaptação:** este terceiro pilar é o que torna o método flexível. Após o processo de inspeção e a identificação de falhas no projeto ou de pontos que possam ser melhorados, a equipe promove mudanças. Essas mudanças podem ser tanto em relação ao produto ou serviço que está sendo elaborado, como no processo de desenvolvimento.

2.2.3 *Papéis do Scrum*

O Scrum define papéis para garantir o funcionamento do método. Primeiramente, é importante conhecer cada um deles para posteriormente entender como aplicá-lo.

1. **Product Owner:** este é um dos papéis principais do Scrum. É o principal responsável por direcionar o projeto, considerando as necessidades e desejos do cliente. O *Product Owner* é a pessoa que irá definir o cronograma, as prioridades e designar as funções para a equipe.
2. **Scrum Master:** o *Scrum Master* atua como um facilitador para a equipe. É responsável por certificar o andamento e o desenvolvimento do projeto. Esse papel é ocupado pela pessoa que tem mais conhecimento sobre a metodologia de trabalho.
3. **Scrum Team:** o terceiro papel é composto por toda a equipe do projeto. O *Scrum Team* é o principal responsável por desenvolver o projeto, seguindo as orientações dadas pelo *Product Owner*.

2.2.4 *Como funciona o Scrum*

O Scrum segue um cronograma e realiza algumas atividades específicas durante o andamento do projeto, sendo elas:

1. **Product Backlog:** é a definição de uma lista de atividades que devem ser feitas durante o processo de desenvolvimento do produto ou serviço, normalmente ordenadas por prioridades. O *Product Backlog* é definido pelo *Product Owner*.

2. **Sprint:** é um ciclo de trabalho. O período de tempo de duração do *sprint* é definido, normalmente, de duas a quatro semanas e a equipe desenvolve tarefas durante essa etapa.
3. **Sprint Planning:** é o planejamento de cada *sprint*. O *Product Owner* é o responsável por definir cada *sprint* e as funcionalidades que devem ser desenvolvidas neles.
4. **Daily Scrum:** são as reuniões diárias que acontecem entre o *Scrum Team* e o *Product Owner*. Isso promove o alinhamento de toda a equipe com o projeto e garante a transparência (um dos pilares do Scrum).

Dessa forma, dentro de um projeto, o Scrum se inicia após a definição dos papéis dos integrantes da equipe. O *Product Owner* irá definir a lista de prioridades e realizará o planejamento dos *sprints*. Posteriormente, a equipe deve iniciar o desenvolvimento do projeto, seguindo o *Product Backlog*, *sprints* definidos e realizando o *Daily Scrum*.

2.2.5 Ferramentas para auxiliar o Scrum

Atualmente, há diversos sites e aplicativos que podem ajudar na implementação do Scrum, principalmente em relação à documentação e organização. Através dessas ferramentas é possível planejar futuros *sprints*, visualizar todas as atividades da equipe no mesmo lugar, saber como está o andamento das tarefas, entre outros. Dentre os diversos softwares que possuem essas funcionalidades dois se destacam que são o Trello e o Jira. Portanto, implementar a metodologia Scrum na equipe é uma maneira de otimizar tempo, melhorar a eficiência do trabalho e entregar um produto que atenda às necessidades e expectativas do seu cliente. Para isso é importante atentar-se não apenas nas etapas da realização das atividades, mas também nos valores e nos princípios ágeis.

2.3 Arquitetura *Client-server* (cliente-servidor)

O modelo cliente-servidor é uma estrutura de aplicativo distribuída que particiona tarefas ou cargas de trabalho entre os provedores de um recurso ou serviço, chamados servidores, e solicitantes de serviços, chamados clientes. Frequentemente, clientes e servidores se comunicam em uma rede de computadores em hardware separado, mas tanto o cliente quanto o servidor podem residir no mesmo sistema. Um *host* de servidor executa um ou mais programas de servidor, que compartilham seus recursos com os clientes. Um cliente geralmente não compartilha nenhum de seus recursos, mas solicita conteúdo ou serviço de um servidor. Os

clientes, portanto, iniciam sessões de comunicação com os servidores, que aguardam solicitações de entrada. Exemplos de aplicativos de computador que usam o modelo cliente-servidor são e-mail, impressão em rede e a World Wide Web (WIKIPEDIA, 2021a).

2.4 Padrão de projeto *Model-view-controller* (MVC)

O padrão MVC separa o projeto do software em três camadas independentes: o modelo (manipulação da lógica de dados), a visão (a interface do usuário) e o controlador (fluxo de aplicação). Esta separação facilita a manutenção do código, que pode ser reutilizado em outros projetos (BALDISSERA, 2021).

2.5 *Backend*

O *Backend* refere-se a qualquer parte de um site ou programa de software que os usuários não veem. Ele contrasta com o *frontend*, que se refere à interface do usuário de um programa ou site. Na terminologia de programação, o *Backend* é a "camada de acesso a dados", enquanto o *frontend* é a "camada de apresentação". A maioria dos sites modernos são dinâmicos, o que significa que o conteúdo da página da Web é gerado dinamicamente. Uma página dinâmica contém um ou mais *scripts* que são executados no servidor web sempre que a página é acessada. Esses *scripts* geram o conteúdo da página, que é enviado ao navegador do usuário. Tudo o que acontece antes da página ser exibida em um navegador da Web faz parte do *Backend* (TECHTERMS, 2020). Exemplos de processos de *Backend* incluem:

1. Processando uma solicitação de página da Web recebida;
2. Executando um *script* (PHP, ASP, JSP, etc.);
3. Acessar dados, como um artigo, de um banco de dados usando consultas SQL;
4. Armazenar ou atualizar registros em um banco de dados;
5. Criptografando e descriptografando dados;
6. Manipulação de *uploads* e *downloads* de arquivos.

2.6 API Rest

2.6.1 API

Uma API é um conjunto de definições e protocolos para construir e integrar software. Às vezes, é referido como um contrato entre um provedor de informações e um usuário de informações - estabelecendo o conteúdo exigido do consumidor (a chamada) e o conteúdo exigido pelo produtor (a resposta). Por exemplo, o design da API para um serviço meteorológico pode especificar que o usuário forneça um código postal e que o produtor responda com uma resposta de 2 partes, a primeira sendo a temperatura alta e a segunda sendo a baixa. Em outras palavras, se você deseja interagir com um computador ou sistema para recuperar informações ou executar uma função, uma API ajuda a comunicar o que você deseja a esse sistema para que ele possa entender e atender à solicitação. Você pode pensar em uma API como um mediador entre os usuários ou clientes e os recursos ou serviços da web que desejam obter. É também uma maneira de uma organização compartilhar recursos e informações enquanto mantém a segurança, o controle e a autenticação - determinando quem tem acesso a quê. Outra vantagem de uma API é que você não precisa saber os detalhes de como seu recurso é recuperado ou de onde ele vem.

2.6.2 Rest

REST é um conjunto de restrições arquitetônicas, não um protocolo ou padrão. Os desenvolvedores de API podem implementar REST de várias maneiras.

Quando uma solicitação do cliente é feita por meio de uma API RESTful, ela transfere uma representação do estado do recurso para o solicitante ou terminal. Essas informações, ou representação, são fornecidas em um dos vários formatos via HTTP: JSON (Javascript Object Notation), HTML, XML, Python, PHP ou texto simples. JSON é o formato de arquivo mais popular para usar porque, apesar do nome, é independente de linguagem, bem como legível por humanos e máquinas.

Outra coisa a ter em mente: cabeçalhos e parâmetros também são importantes nos métodos HTTP de uma solicitação HTTP da API RESTful, pois contêm informações de identificador importantes quanto aos metadados da solicitação, autorização, identificador uniforme de recursos (URI), armazenamento em cache, cookies e mais. Existem cabeçalhos de solicitação e cabeçalhos de resposta, cada um com suas próprias informações de conexão HTTP e códigos de status.

Para que uma API seja considerada RESTful, ela deve estar em conformidade com estes critérios:

1. Uma arquitetura cliente-servidor composta de clientes, servidores e recursos, com solicitações gerenciadas por meio de HTTP.
2. Comunicação cliente-servidor sem estado, o que significa que nenhuma informação do cliente é armazenada entre as solicitações de obtenção e cada solicitação é separada e desconectada.
3. Dados armazenáveis em cache que otimizam as interações cliente-servidor.
4. Uma interface uniforme entre os componentes para que as informações sejam transferidas em um formato padrão. Isso requer que:
 - a) os recursos solicitados são identificáveis e separados das representações enviadas ao cliente.
 - b) os recursos podem ser manipulados pelo cliente por meio da representação que eles recebem, pois a representação contém informações suficientes para isso.
 - c) mensagens autodescritivas retornadas ao cliente têm informações suficientes para descrever como o cliente deve processá-las.
 - d) hipertexto / hipermídia está disponível, o que significa que após acessar um recurso o cliente deve ser capaz de usar hiperlinks para encontrar todas as outras ações disponíveis atualmente que ele possa realizar.
5. Um sistema em camadas que organiza cada tipo de servidor (os responsáveis pela segurança, balanceamento de carga, etc.) envolvia a recuperação das informações solicitadas em hierarquias, invisíveis para o cliente.
6. Code-on-demand (opcional): a capacidade de enviar código executável do servidor para o cliente quando solicitado, estendendo a funcionalidade do cliente.

Embora a API REST tenha esses critérios para estar em conformidade, ela ainda é considerada mais fácil de usar do que um protocolo prescrito como SOAP (Simple Object Access Protocol), que tem requisitos específicos como mensagens XML e segurança integrada e conformidade de transações que o tornam mais lento e mais pesado.

Em contraste, REST é um conjunto de diretrizes que pode ser implementado conforme necessário, tornando as APIs REST mais rápidas e leves, com maior escalabilidade - perfeito para Internet das Coisas (IoT) e desenvolvimento de aplicativos móveis (HAT, 2020).

2.7 Linguagem javascript

JavaScript é uma linguagem de programação que permite a você implementar itens complexos em páginas web — toda vez que uma página da web faz mais do que simplesmente mostrar a você informação estática — mostrando conteúdo que se atualiza em um intervalo de tempo, mapas interativos ou gráficos 2D/3D animados, etc. — você pode apostar que o JavaScript provavelmente está envolvido (MDN, 2022). Com o passar dos anos e a evolução da linguagem, o javascript transcendeu do seu papel inicial como apresentado acima para também funcionar no *Backend*, graças ao motor V8 do chrome que pôde ser abstraído dando origem ao Node.js que será detalhado na Seção 2.8.

2.8 Node.js

Node.js é um ambiente de tempo de execução JavaScript de código aberto, multi-plataforma e *Backend* que roda no mecanismo V8 e executa o código JavaScript fora de um navegador da web. O Node.js permite que os desenvolvedores usem JavaScript para escrever ferramentas de linha de comando e para scripts do lado do servidor - executando scripts do lado do servidor para produzir conteúdo dinâmico de página da web antes que a página seja enviada ao navegador do usuário. Conseqüentemente, o Node.js representa um paradigma "JavaScript em todos os lugares", unificando o desenvolvimento de aplicativos da web em torno de uma única linguagem de programação, em vez de diferentes linguagens para scripts do lado do servidor e do lado do cliente. O Node.js tem uma arquitetura orientada a eventos capaz de entrada / saída assíncrona. Essas opções de design visam otimizar o rendimento e a escalabilidade em aplicativos da web com muitas operações de entrada / saída, bem como para aplicativos da Web em tempo real (por exemplo, programas de comunicação em tempo real e jogos de navegador). O projeto de desenvolvimento distribuído Node.js era anteriormente administrado pela Fundação Node.js, e agora se fundiu com a Fundação JS para formar a Fundação OpenJS, que é facilitada pelo programa Projetos Colaborativos da Fundação Linux (WIKIPEDIA, 2020a).

2.9 Linguagem SQL

SQL é uma linguagem específica de domínio usada em programação e projetada para gerenciar dados mantidos em um *Relational Database Management System* (RDBMS) ou para processamento de fluxo em um *Relational Data Stream Management System* (RDSMS).

É particularmente útil no tratamento de dados estruturados, ou seja, dados que incorporam relações entre entidades e variáveis. Originalmente baseado em álgebra relacional e cálculo relacional de tuplas, o SQL consiste em muitos tipos de instruções, que podem ser informalmente classificadas como sublinguagens, que são: *Data Query Language* (DQL), uma *Data Definition Language* (DDL), uma *Data Control Language* (DCL) e *Data Manipulation Language* (DML). O escopo do SQL inclui consulta de dados, manipulação de dados (inserir, atualizar e excluir), definição de dados (criação de esquema e modificação) e controle de acesso a dados. Embora SQL seja essencialmente uma linguagem declarativa, ela também inclui elementos procedurais. SQL tornou-se um padrão do *American National Standards Institute* (ANSI) em 1986, e da *International Organization for Standardization* (ISO) em 1987. Desde então, o padrão foi revisado para incluir um conjunto maior de recursos (WIKIPEDIA, 2021d).

2.10 PostgreSQL

O PostgreSQL, também conhecido como Postgres, é um RDSMS gratuito e de código aberto que enfatiza a extensibilidade e a conformidade com o SQL. Foi originalmente chamado POSTGRES, referindo-se às suas origens como um sucessor do banco de dados Ingres desenvolvido na Universidade da Califórnia, Berkeley. Em 1996, o projeto foi renomeado para PostgreSQL para refletir seu suporte ao SQL. O PostgreSQL apresenta transações com propriedades de Atomicidade, Consistência, Isolamento, Durabilidade (ACID), visualizações automaticamente atualizáveis, visualizações materializadas, gatilhos, chaves estrangeiras e procedimentos armazenados. Ele é projetado para lidar com uma variedade de cargas de trabalho, de máquinas individuais a *data warehouses* ou serviços da Web com muitos usuários simultâneos. É o banco de dados padrão do macOS Server e também está disponível para Windows, Linux, FreeBSD e OpenBSD (WIKIPEDIA, 2021b).

2.11 ReactJs

React (também conhecido como React.js ou ReactJS) é uma biblioteca JavaScript *Frontend* gratuita e de código aberto para a construção de interfaces de usuário baseadas em componentes de interface de usuário. É mantido pela Meta (antigo Facebook) e por uma comunidade de desenvolvedores individuais e empresas. O React pode ser usado como base no desenvolvimento de aplicativos de página única ou móveis. No entanto, o React está preocupado

apenas com o gerenciamento de estado e renderização desse estado para o *Document Object Model* (DOM), portanto, a criação de aplicativos React geralmente requer o uso de bibliotecas adicionais para roteamento, bem como certas funcionalidades do lado do cliente (WIKIPEDIA, 2020b).

2.11.1 React-Admin

React Admin (react-admin) é a estrutura de *Frontend* para construir um painel de administração com React e se adapta a qualquer RESTful, GraphQL ou APIs personalizadas. O React-admin é mais do que apenas um modelo de painel de administração em HTML/CSS. É uma estrutura completa para construir qualquer painel em apenas algumas horas. Ele fornece adaptadores de dados, componentes CRUD, autenticação e autorização, temas, tradução e um ecossistema de pacotes de suporte incríveis. A melhor parte é que o react-admin pode ser usado de forma independente ou integrado ao aplicativo React existente (GEEK, 2021).

3 METODOLOGIA

Ao longo desta seção serão descritos e detalhados as etapas para o desenvolvimento da aplicação proposta.

3.1 Uso da metodologia Scrum

Todo o projeto de desenvolvimento foi pautado pelo uso da metodologia ágil Scrum. Com isso, o time era composto pelo Prof. Dr. Iális e mais dois alunos o Gabriel de Souza e o Fabrício Moura. Dito isto, podemos perceber que o time é pequeno logo algumas adaptações ao modelo descrito da metodologia foram necessários como as *daily*s setem feitas em um grupo do Whatsapp, os papéis de *Scrum master* e *Product Owner* eram feitos pelo Prof. Dr. Iális e os outros participantes faziam o papel de *Scrum team*. Inicialmente, fizemos uma lista de atividades a serem seguidas para o desenvolvimento da aplicação (*Product Backlog*) que seriam revisadas a cada fim/início de uma *sprint*. Além disso, as *sprints* tinham um período de duas semanas onde realizávamos uma reunião online para discutir as entregas da semana anterior (*Sprint Review*) e planejar as entregas da próxima *sprint* (*Sprint Planning*).

3.2 Levantamento de requisitos com a coordenação do programa

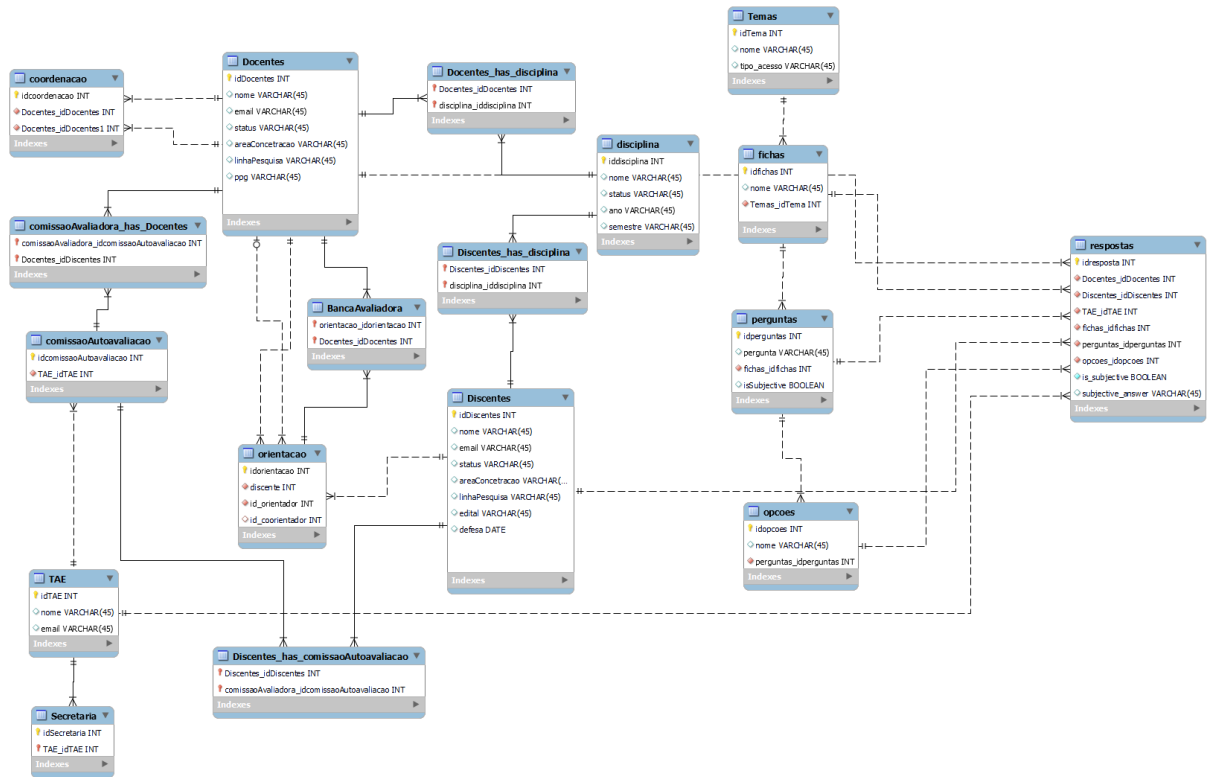
Inicialmente, foi realizado um levantamento de quais seriam os pontos relevantes para que a aplicação tivesse um impacto positivo e cumprisse seu papel. Dito isso, ocorreu uma sequencia de reuniões onlines com a coordenação e responsáveis, onde foi discutido quais os pontos seriam importantes para uma primeira versão utilizável do programa, foi discutido como seria a forma ideal de fichas de avaliações em termos de layout, como seriam o formato das opções de respostas em termos de layout e quais seriam as primeiras fichas implementadas. Sendo assim, todas as reuniões foram pautadas pela resolução RESOLUÇÃO Nº 1/2021/CMEEC/CUFCSOBRA/REITORIA (PPGEEC/UFC, 2021) para melhor atender a demanda.

3.3 Desenvolvimento do *Backend*

Após os requisitos estarem claros dei inicio a modelagem do banco de dados que irá armazenar todas as respostas dos usuários, bem como suas informações, a Figura 2 mostra

como a modelagem foi definida para melhor atender os requisitos. Por conseguinte, começamos

Figura 2 – Modelagem do banco de dados



Fonte: Elaborado pelo autor.

o desenvolvimento da API, a aplicação foi separada por "camadas" onde cada pasta representava uma camada diferente. Com isso, desenvolvemos uma camada de banco de dados que lida com a parte de conexão, configuração e construção do banco de dados. A segunda camada seria responsável pelo modelo das tabelas traduzido para código javascript onde será possível manipular os dados recebidos e salva-los no banco, para isso utilizei o *Object-Relational Mapping* (ORM) sequelize que traz uma gama de funções abstraindo o uso de SQL puro e adicionando uma camada de segurança as operações feitas no banco. A terceira camada seria a de roteamento onde foi definido todas as rotas que podem ser chamadas na API, esse roteamento foi dividido por entidades, cada tabela da Figura 2 se transformou em um grupo de rotas responsáveis por fazer as manipulações necessárias nos dados dessa entidade como criar, ler, atualizar e deletar. A quarta camada seria a logica que cada rota requisitada passa para permitir a manipulação dos dados, nela adicionei mais passos de segurança validando os dados enviados na requisição para evitar que algum usuário sem permissão possa manipular dados que não deveria e validações para os dados enviados, principalmente sobre dados obrigatórios de cada entidade. Além disso, construí uma documentação interativa da API, onde os usuários podem testar as rotas diretamente na

documentação e para que tenham conhecimento do que poderá ou não fazer e quais rotas devem ser chamadas com quais parâmetros para que obtenha um sucesso, a qual podemos ver na Figura 3. Por fim, fiz as devidas configurações para que esse *Backend* pudesse ser acessado via internet através de um servidor pessoal para todos do time testarem e verificarem as funcionalidades desenvolvidas.

3.3.1 *Uso Banco de dados*

Na Figura 2 temos o modelo entidade-relacionamento da modelagem do banco de dados. Dito isto, foi utilizado um banco de dados relacional, pois foram definidas várias relações entre as entidades como podemos observar na Figura 2, caso fosse optado um banco não relacional seria muito complexo gerenciar todas essas relações. Além disso, a distribuição dos campos de cada tabela e a existência de cada tabela veio do levantamento de requisitos, onde discutimos o que precisaria ser salvo no banco de dados de modo a prover relatórios mais precisos e coesos.

3.4 *Desenvolvimento do Frontend de administração*

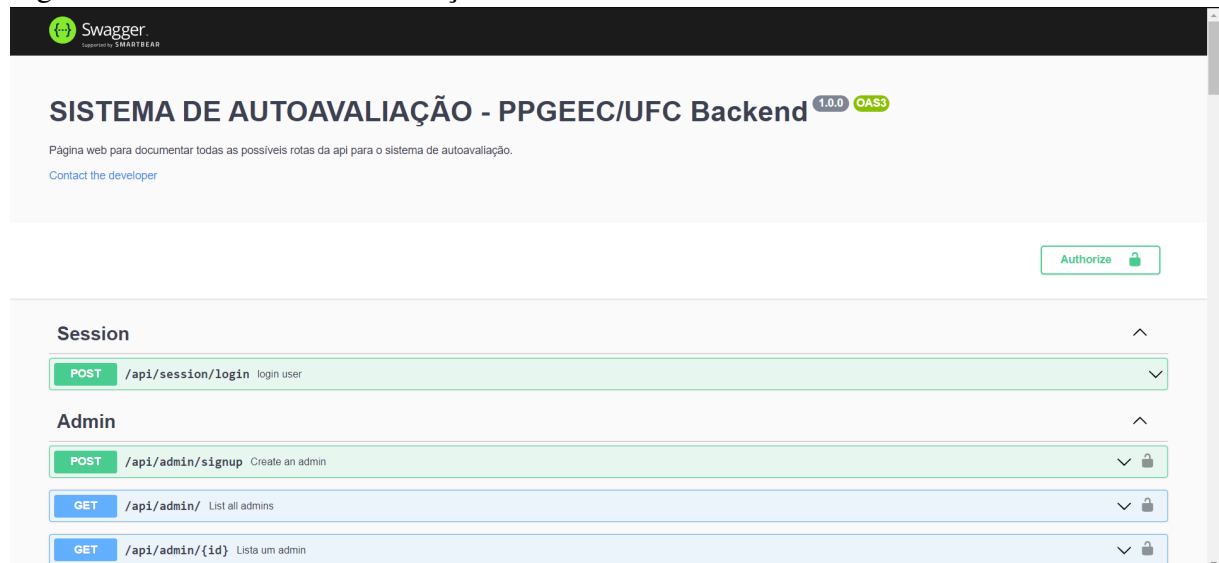
Em paralelo com a construção do *Backend*, iniciei a construção do painel de administração, configurando o *framework* react-admin para se adaptar melhor a API que estava desenvolvendo. Com isso, ao fim do desenvolvimento de cada entidade no *Backend* desenvolvia essa funcionalidade no painel administrativo para poder testar as rotas de uma forma que o usuário final utilizaria. Após iniciar o desenvolvimento do painel pude perceber a falta de algumas opções como a de filtrar registros das tabelas para maior facilidade de encontrar dados, isso levou um tempo maior de desenvolvimento no *Backend* para implementar tais filtros. Um dos requisitos do painel era a exportação do conteúdo das tabelas como um csv, um dos motivos de escolha desse *framework* especializado em painéis administrativos foi essa opção, pois era um dos requisitos iniciais e já estaria implementado com o uso do *framework*. Por fim, fiz as devidas configurações para que esse *Frontend* pudesse ser acessado via internet através de um servidor pessoal para todos do time testarem e verificarem as funcionalidades desenvolvidas. A Figura 9 mostra como ficou a interface desenvolvida.

4 RESULTADOS

4.1 Resultados do desenvolvimento do *Backend*

Como resultado do desenvolvimento do *Backend* as Figuras 3, 4, 5, 6 e 7 são referente a documentação interativa que foi desenvolvida.

Figura 3 – Interface da documentação interativa



Fonte: Elaborado pelo autor.

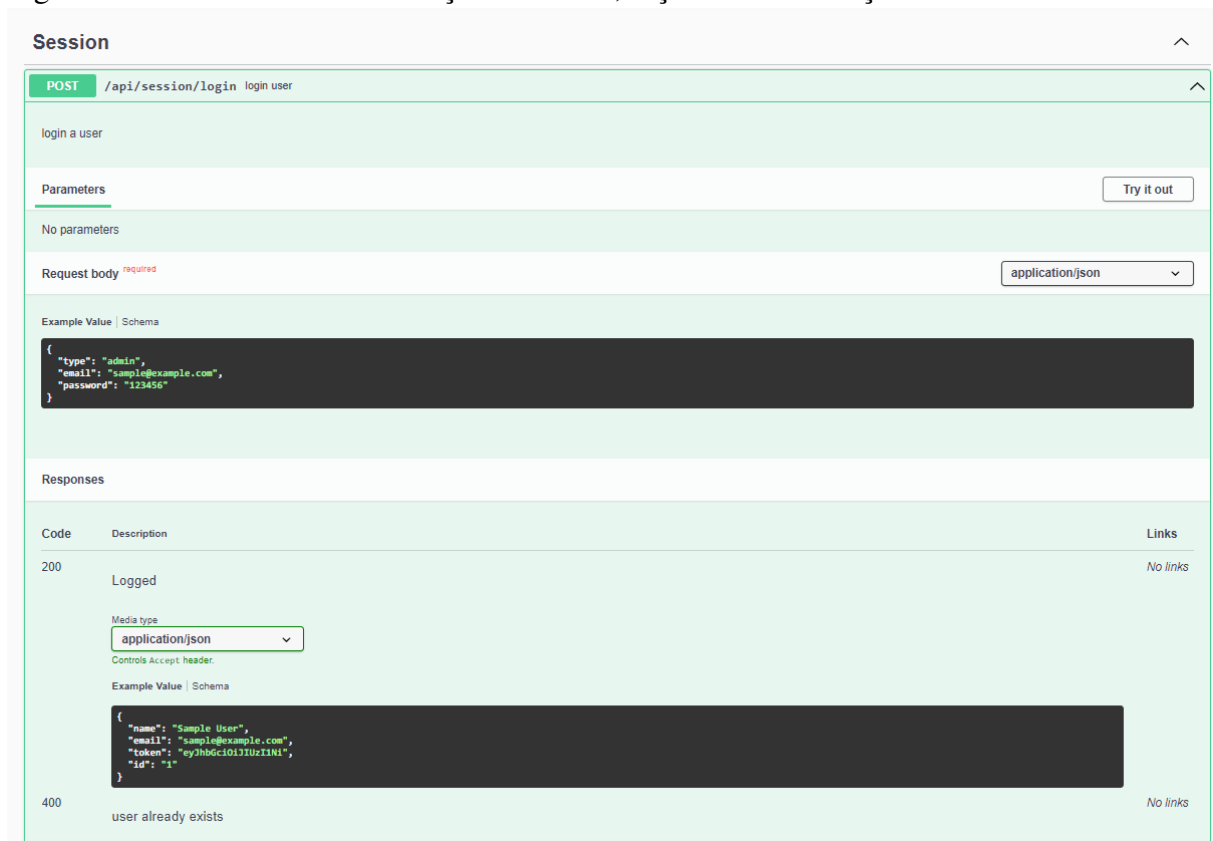
A Figura 3 mostra como é essa interface, onde cada seção tem uma cor baseada no método http que é usado, ao clicar abre-se um menu onde o usuário pode verificar quais parâmetros são recebido para essa requisição e pode testá-lo para validar. Na Figura 4 podemos observar os parâmetros que essa requisição pede e um exemplo de como será a resposta da mesma em caso de sucesso. Ao clicar no botão *Try it out*, uma nova seção é aberta contendo campos editáveis para o usuário editar e executar uma requisição testando em tempo real.

Na Figura 5 podemos observar um campo branco com um objeto em *JavaScript Object Notation* (JSON) que é editável, após preencher com os dados que deseja testar basta clicar no botão azul com o nome *Execute* e ele irá executar a requisição e mostrará o resultado em baixo em caso de sucesso ou erro.

Na Figura 6 podemos observar um exemplo de sucesso da requisição de autenticação, que por motivos de segurança alguma informações foram ocultadas.

O exemplo ilustrado foi da seção de autenticação, para as outras seções o conceito continua o mesmo, porém cada requisição tem suas particularidades de parâmetros e afins. Na Figura 3 podemos observar que algumas requisições tem um ícone de cadeado ao lado, isso

Figura 4 – Interface da documentação interativa, seção de autenticação



Fonte: Elaborado pelo autor.

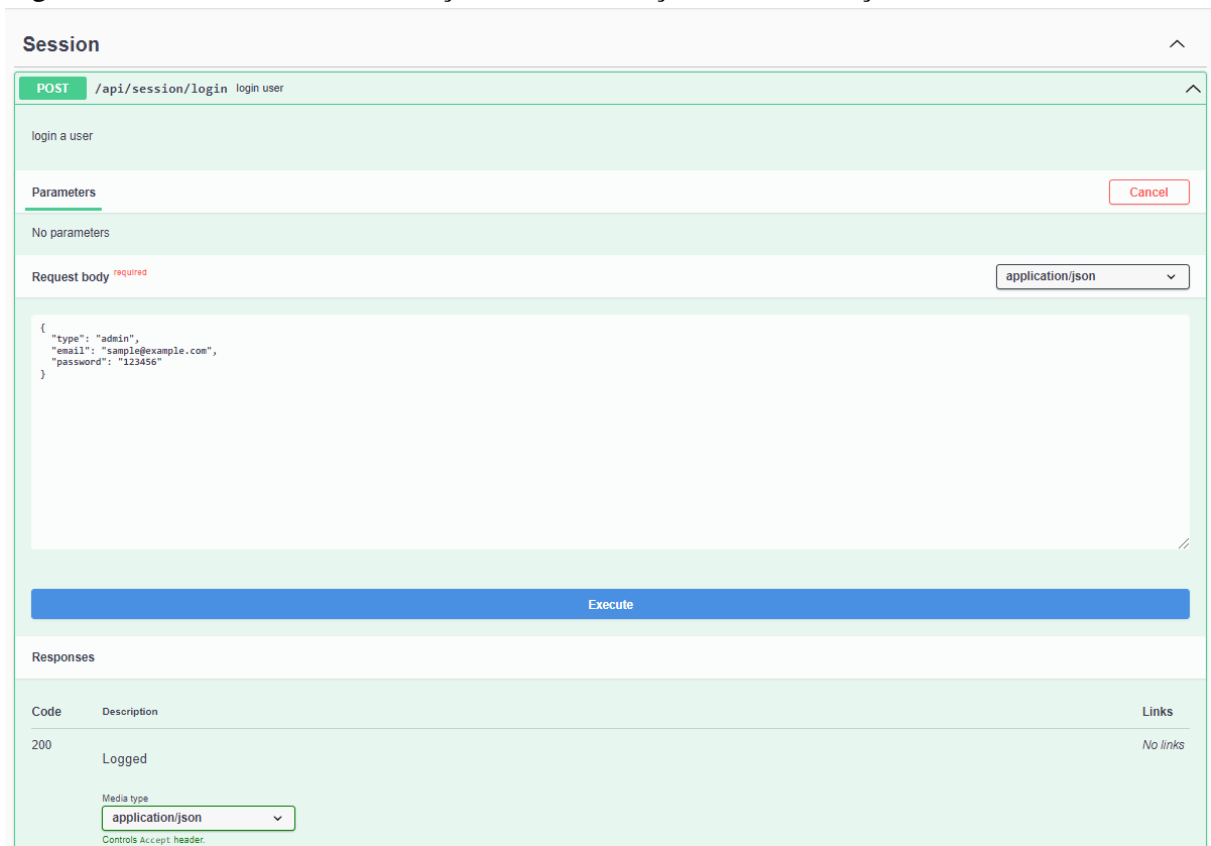
significa que para o uso desse tipo de requisição é necessário que haja autenticação ativada para efetuar o teste de requisição. Ainda na Figura 3 podemos ver no topo um botão escrito *Authorize*, ao clicar nele um modal aparece na tela com dois campos para que o usuário digite o seu token de autenticação como ilustrado na Figura 7. Após obter o token usando a requisição de login (*Session* na documentação), caso as credenciais estejam corretas no objeto de resposta temos um campo escrito token, copiamos o valor dele e colamos no campo correto do modal baseado no tipo de usuário que efetuou o login. Com isso, após colar o token no local correto basta clicar no botão *Authorize* que as requisições já estarão autenticadas.

4.2 Resultados do desenvolvimento do painel administrativo

Como resultado do desenvolvimento do painel administrativo segue as Figuras 8, 9, 10 e 11 que são da interface que foi desenvolvida utilizando o *Backend* apresentado no tópico anterior.

A Figura 8 mostra a interface de login, onde somente usuários administradores tem acesso. Além disso, foi desenvolvido um painel de manipulação dos dados salvos no banco como

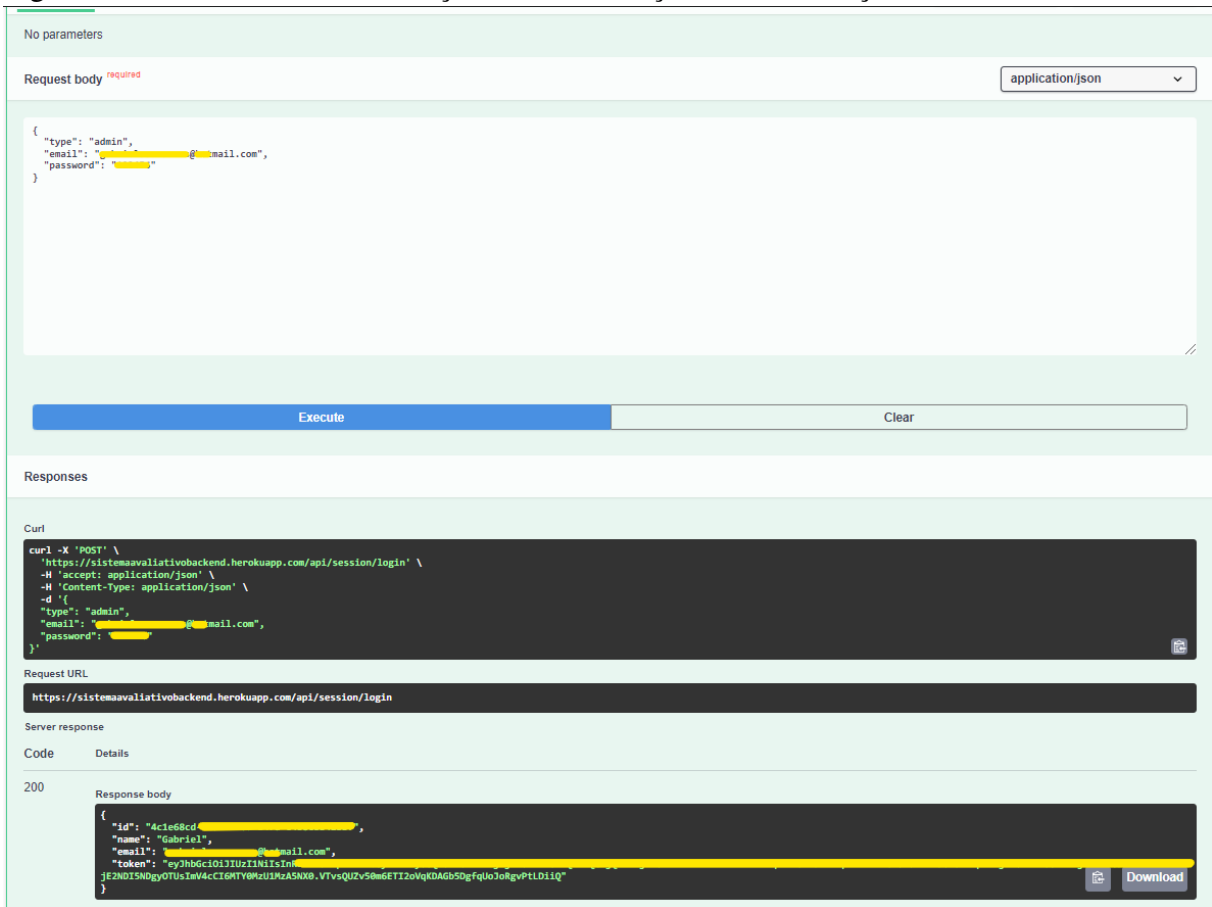
Figura 5 – Interface da documentação interativa, seção de autenticação



Fonte: Elaborado pelo autor.

na Figura 9, que ilustra um exemplo da tabela de administradores cadastrados, nela podemos observar vários botões, na esquerda temos o menu principal, onde podemos navegar pela interface de manipulação das outras entidades do banco de dados. No topo da tela temos o cabeçalho da página, que identifica o recurso que o usuário está na esquerda, na Figura 9 é onde temos escrito "Admins", o botão com 3 traços na horizontal serve para ocultar parte do menu visando melhorar a visualização das tabelas. No lado direito do cabeçalho temos um botão de recarregamento das informações e um botão para interações do usuário logado, no momento só temos a opção de desconectar da conta, ação que leva o usuário de volta para a tela de login Figura 8. No topo da interface, abaixo do cabeçalho temos um campo onde pode ser consultado os administradores por e-mail. No mesmo nível a direita podemos ver 2 botões um escrito "CRIAR" e outro escrito "EXPORTAR". Ao clicar no botão exportar é feito um *download* em formato de CSV dos dados contidos nessa tabela para ser usado para processamento no futuro. Ao interagir com o botão de criação, o usuário é redirecionado para outra tela como na Figura 10, onde os campos possíveis para cada entidade estão expostos. Depois de preencher todos os campos, basta clicar no botão de salvar e o usuário é redirecionado a página padrão de visualização da tabela da sua entidade. No exemplo, estamos utilizando a entidade de administradores então voltamos para a tela da Figura

Figura 6 – Interface da documentação interativa, seção de autenticação

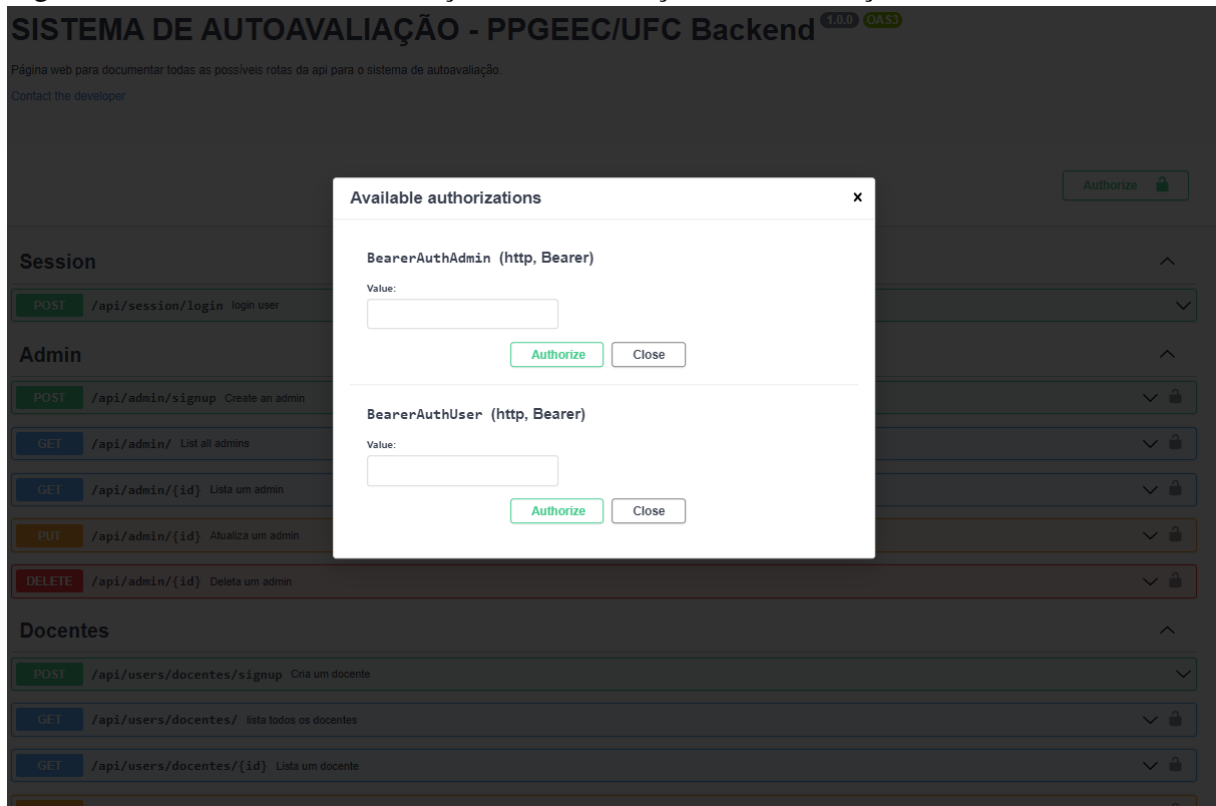


Fonte: Elaborado pelo autor.

9. Dentro da Figura 9 podemos observar um botão escrito "REMOVER", como o próprio nome já diz, ele deleta esse registro do banco de dados. Abaixo da tabela temos botões que alteram a quantidade de itens por página e controlam qual a página atual, as próximas e as anteriores, facilitando ao usuário navegar por elas.

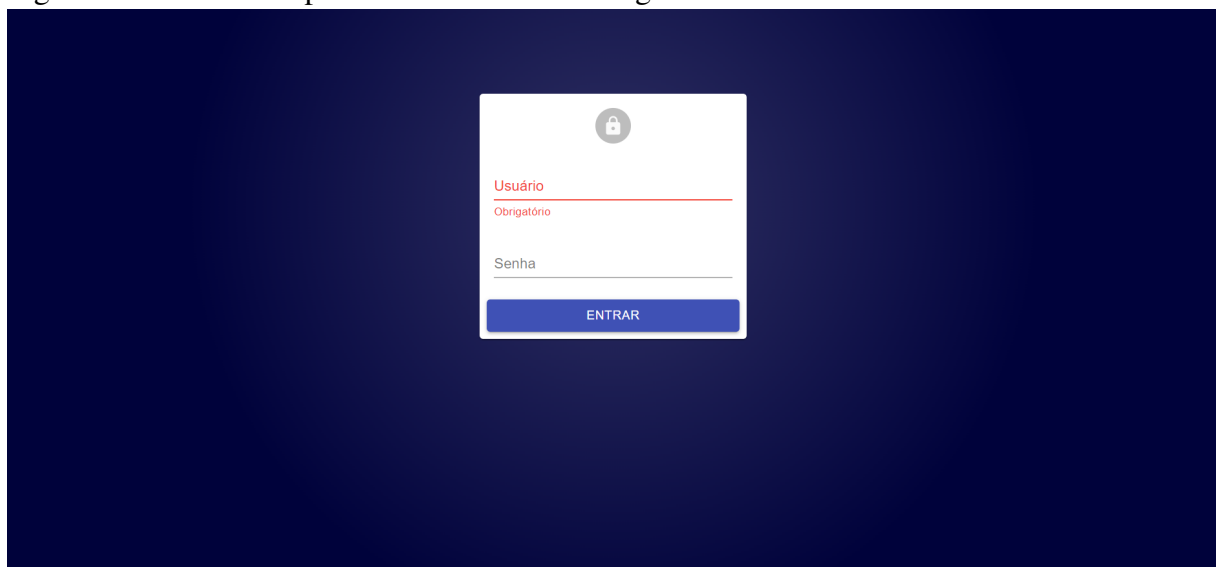
Todas as outras telas do painel seguem a mesma lógica apresentada acima, contendo os mesmo elementos, diferindo somente na quantidade de campos para criação e exibição dos mesmos e o tipo de filtro que pode ser aplicado para a busca na visualização. Além disso, o método para criação de fichas avaliativas é um pouco mais complexo, como mostra a Figura 11. Após escolher o tema da ficha, partimos para o nome da mesma e as perguntas, atualmente não existe limite para a quantidade de perguntas definido em código. Entretanto, a criação das fichas seguirá o que foi escrito na resolução do PPGEEC. Com isso, caso a pergunta seja subjetiva podemos deixar a opção em branco, caso contrario temos que preencher com as opções para a determinada pergunta, aqui também não existe um limite em código, e durante a criação será utilizado as opções passadas na resolução do PPGEEC. Por fim, terminando de preencher toda a ficha temos que clicar no botão de salvar para criá-la.

Figura 7 – Interface da documentação interativa, seção de autenticação



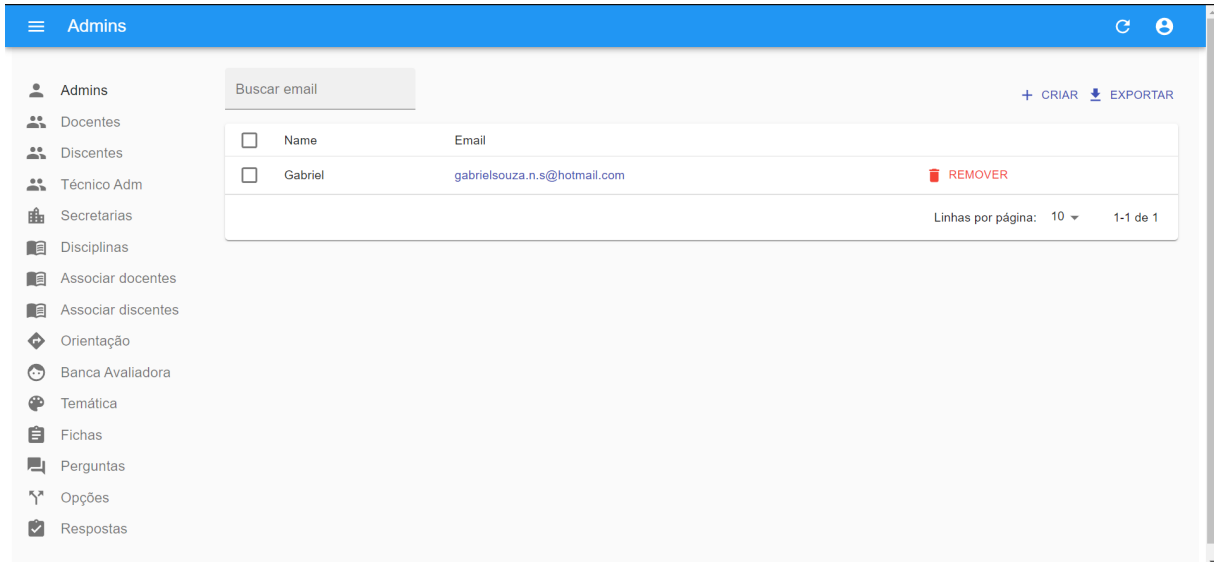
Fonte: Elaborado pelo autor.

Figura 8 – Interface do painel administrativo - Login



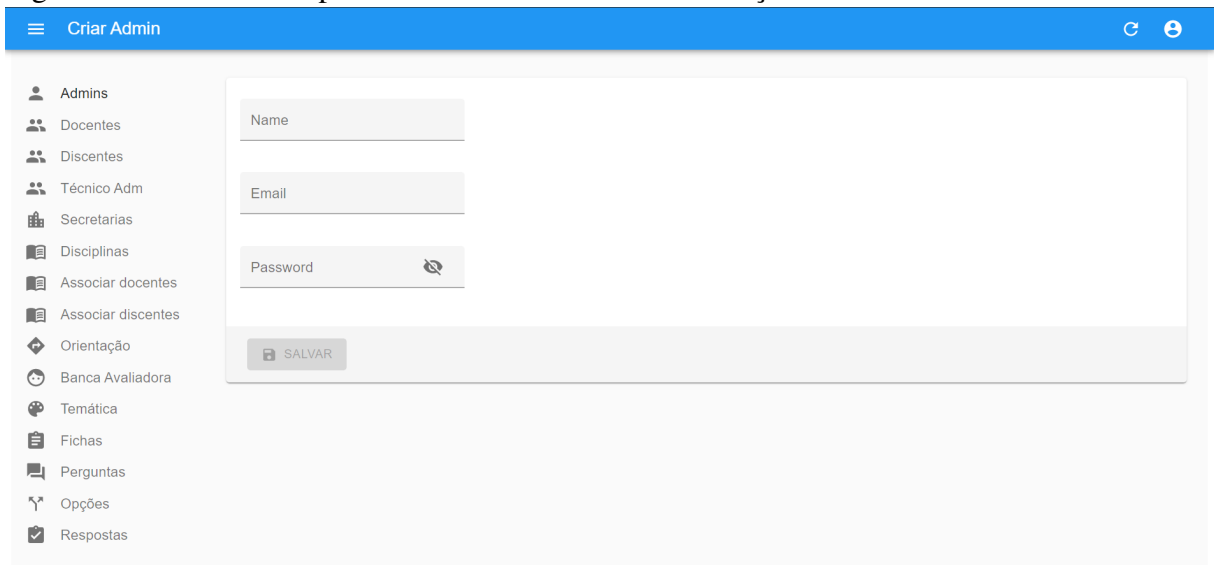
Fonte: Elaborado pelo autor.

Figura 9 – Interface do painel administrativo



Fonte: Elaborado pelo autor.

Figura 10 – Interface do painel administrativo - tela de criação de admins



Fonte: Elaborado pelo autor.

Figura 11 – Interface do painel administrativo - tela de criação de fichas

The screenshot displays the 'Fluxo criar Fichas' (Flow create Cards) interface. On the left, a sidebar lists various administrative functions: Admins, Docentes, Discentes, Técnico Adm, Secretárias, Disciplinas, Associar docentes, Associar discentes, Orientação, Banca Avaliadora, Temática, Fichas, Perguntas, Opções, and Respostas. The main area features a dropdown menu for selecting a theme, currently set to 'Inteligencia Artificial'. Below this, there is a section for 'Fichas' (Cards) with a list of items. The first item is '1 Qual o nome da Ficha? *' with a 'REMOVER' button. The second item is '1 É uma pergunta subjetiva?' with a toggle switch and a 'REMOVER' button. Below the toggle is a 'Pergunta *' input field and a note: 'Caso seja uma pergunta subjetiva deixe as opções em branco.' The third item is '1 Opção *' with an 'Opção *' input field and a 'REMOVER' button. At the bottom of the list are three 'ADICIONAR' (Add) buttons and a 'SALVAR' (Save) button.

Fonte: Elaborado pelo autor.

5 CONSIDERAÇÕES FINAIS

Diante do trabalho apresentado pode-se concluir que o desenvolvimento de um software de qualidade não é uma coisa simples, ele envolve vários processos, metodologias e boas práticas de escrita para ser desenvolvido. Dito isto, a aplicação da metodologia scrum se mostrou bastante eficiente embora não tenhamos seguido fielmente a literatura, com certas adaptações por limitação no número de pessoas se mostrou bastante eficiente, pois tanto a equipe de desenvolvimento como o cliente estão sempre mantendo uma boa comunicação e corrigindo eventuais erros rapidamente, antes que se torne um problema maior. Além disso, a escolha da arquitetura e do formato de API, foi uma escolha bastante sábia, pois o mesmo *Backend*, poderá ser usado para a construção de um aplicativo mobile sem ter alterações, ou seja é bastante versátil. Entretanto a escolha do padrão de projeto *Model-view-controller* (MVC), é uma boa escolha para projeto simples e menores, porém com o crescimento do projeto talvez não seja o melhor, pois algumas funcionalidades de regras de negócio são acopladas aos *controllers* o que pode gerar uma manutenção um pouco mais complexa, uma alternativa para isso seria a separação dessa lógica em *services* especializados nas regras de negócios. Com relação a colocar a aplicação disponível na internet foi um grande acerto, pois os *feedback* eram rápido e precisos, quando tinha algum erro nas funcionalidades rapidamente era reportado e corrigido, aumentando a produtividade.

Para trabalhos futuros temos o aprimoramento do software como um todo, refatorando para deixar mais sucinto ou refatorando em uma arquitetura mais escalável para melhor manutenibilidade do projeto. Como sugestão de melhoria deixo a implementação de uma versão mobile e um *Frontend* de interação para o usuário, que está em desenvolvimento por outros colegas. Além disso, deixo como sugestão uma nova rota ou um novo backend para consumir os dados gerados pelas fichas e gerar análises para que possa achar pontos de melhoria no programa e transformá-lo em um programa melhor.

REFERÊNCIAS

BALDISSERA, O. **Quais são os tipos de arquitetura de software e como escolher o melhor para seu projeto.** 2021. Disponível em: <https://posdigital.pucpr.br/blog/tipos-de-arquitetura-de-software>. Acesso em: 20 jan. 2022.

BRASIL. **Parecer nº 977/1965, aprovado em 3 dez. 1965.** 1965. Disponível em: <http://www.scielo.br/pdf/rbedu/n30/a14n30.pdf>. Acesso em: 21 dez. 2021.

CAPES. Autoavaliação de programas de pós-graduação. **Capex**, Capex, v. 1, n. 1, 2019.

EJMC. **Vantagens do Scrum e como aplicá-lo em sua empresa – Empresa Júnior Mackenzie Consultoria.** 2021. Disponível em: <https://jrmack.com.br/vantagens-do-scrum-e-como-aplica-lo-em-sua-empresa/?gclid=CjwKCAiAz--OBhBIEiwAG1rIOl3QI3PCneIT2wWeM6IGjrGnyJqJXNr5gvBTS1pYwYRT40XKCZeC8xoBwE>. Acesso em: 20 jan. 2022.

GEEK. **React Admin - The Best React Framework to Build Dashboard.** 2021. Disponível em: <https://ageek.dev/react-admin>. Acesso em: 20 jan. 2022.

GEOCAPES. **GEOCAPES - Sistema de Informações Georreferenciadas.** 2018. Disponível em: <https://geocapes.capes.gov.br/geocapes/>. Acesso em: 21 dez. 2021.

HAT, R. **What is a REST API?** 2020. Disponível em: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>. Acesso em: 20 jan. 2022.

JOVENSGENIOS. **Avaliação Interna e Avaliação externa.** 2020. Disponível em: <https://blog.jovensgenios.com/avaliacao-interna-e-externa/#:~:text=Conhecida%20como%20avalia%C3%A7%C3%A3o%20de%20desempenho,o%20redirecionamento%20de%20pr%C3%A1ticas%20pedag%C3%B3gicas>. Acesso em: 20 jan. 2022.

LEITE, R. V. D. **A autoavaliação na Pós-Graduação (PG) como componente do processo avaliativo CAPES.** 2020. Disponível em: <https://ppgeec.ufc.br/wp-content/uploads/2021/02/sei-ppgeec-resolucao-autoavaliacao.pdf>. Acesso em: 20 jan. 2022.

MDN. **O que é JavaScript? - Aprendendo desenvolvimento web | MDN.** 2022. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript. Acesso em: 20 jan. 2022.

PPGEEC/UFUC. **Regulamento da Autoavaliação Periódica e o Plano de Autoavaliação.** 2021. Disponível em: <https://ppgeec.ufc.br/wp-content/uploads/2021/02/sei-ppgeec-resolucao-autoavaliacao.pdf>. Acesso em: 20 jan. 2022.

SCRUM.ORG. **Scrum.** 2022. Disponível em: <https://www.scrum.org/>. Acesso em: 20 jan. 2022.

TECHTERMS. **Backend Definition.** 2020. Disponível em: <https://techterms.com/definition/backend>. Acesso em: 05 mar. 2022.

WIKIPEDIA. **Node.js.** 2020. Disponível em: <https://en.wikipedia.org/wiki/Node.js>. Acesso em: 20 jan. 2022.

WIKIPEDIA. **React (JavaScript library).** 2020. Disponível em: [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)). Acesso em: 20 jan. 2022.

WIKIPEDIA. **Client–server model**. 2021. Disponível em: https://en.wikipedia.org/wiki/Client%E2%80%93server_model. Acesso em: 20 jan. 2022.

WIKIPEDIA. **PostgreSQL**. 2021. Disponível em: <https://en.wikipedia.org/wiki/PostgreSQL>. Acesso em: 20 jan. 2022.

WIKIPEDIA. **Software architecture**. 2021. Disponível em: https://en.wikipedia.org/wiki/Software_architecture. Acesso em: 20 jan. 2022.

WIKIPEDIA. **SQL**. 2021. Disponível em: <https://en.wikipedia.org/wiki/SQL>. Acesso em: 20 jan. 2022.

ANEXO A – RESOLUÇÃO DO PPGEEC



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO CEARÁ
REITORIA
CAMPUS DA UFC EM SOBRAL/DIRETORIA
CURSO DE MESTRADO EM ENGENHARIA ELÉTRICA E DA COMPUTAÇÃO

RESOLUÇÃO Nº 1/2021/CMEEC/CUFCSOBRAL/REITORIA, de 27 de janeiro de 2021.

Dispõe sobre regulamento e plano de ação do processo de autoavaliação do Programa de Pós-graduação em Engenharia Elétrica e Computação da Universidade Federal do Ceará - Campus de Sobral (PPGEEC/UFC).

O COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E COMPUTAÇÃO, DA UNIVERSIDADE FEDERAL DO CEARÁ - CAMPUS DE SOBRAL, no uso das atribuições legais e estatutárias,

CONSIDERANDO a Proposta de Aprimoramento do Modelo de Avaliação da Pós- Graduação da CAPES;

CONSIDERANDO ainda aprovação dos membros do colegiado, por unanimidade, na reunião ordinária do dia 15 de janeiro de 2021;

CONSIDERANDO deliberação tomada em sua Reunião nº xxx, de y de mmmmmmm de aaaa;

CONSIDERANDO o constante dos autos do processo nº 23067.003490/2021-60,

R E S O L V E :

Art. 1º Aprovar o Regulamento da Autoavaliação Periódica e o Plano de Autoavaliação do Programa de Pós-graduação em Engenharia Elétrica e Computação da Universidade Federal do Ceará - Campus de Sobral (PPGEEC/UFC), conforme anexos I e II desta Resolução.

Art. 2º Esta Resolução entra em vigor na data de sua publicação.



Documento assinado eletronicamente por **IALIS CAVALCANTE DE PAULA JUNIOR, Coordenador de Pós-Graduação**, em 27/01/2021, às 20:14, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site



https://sei.ufc.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1767544** e o código CRC **391538C5**.

Prof. Iális Cavalcante de Paula Júnior
Coordenador PPGEEC/UFC

Referência: Processo nº 23067.003490/2021-60

SEI nº 1767544

ANEXO I DA RESOLUÇÃO No 01/2021

REGULAMENTO DO PLANO DE AUTOAVALIAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E COMPUTAÇÃO – PPGEEC

CAPÍTULO I DAS DISPOSIÇÕES PRELIMINARES

Art. 1o. O presente Regulamento tem por objetivo normalizar o conjunto de atividades relacionadas com a Autoavaliação Periódica do PPGEEC/UFC.

Art. 2o. O Plano de Autoavaliação do PPGEEC tem como objetivos:

- I. Avaliar de maneira sistemática e periódica o funcionamento do PPGEEC e dotar o curso de mecanismos de autoajuste, sempre que se mostrarem necessários;
- II. Estimular a busca por padrões de excelência operacional, através da identificação de oportunidades de melhoramentos, dentro de um processo de busca por melhoramentos contínuos;
- III. Proporcionar mecanismos de sintonia com as políticas de avaliação interna e externa da pós-graduação da instituição, definidas pela CAPES;
- V. Assegurar canais de comunicação, abertos em permanência, destinado ao corpo discente, técnico-administrativo e docente do PPGEEC/UFC, para receber queixas, reclamações, sugestões, etc.

CAPÍTULO II DA DEFINIÇÃO

Art. 3o. O Plano de Autoavaliação do PPGEEC/UFC consiste em um conjunto de políticas, diretrizes e instrumentos, inter-relacionados e sistêmicos, com o objetivo de implementar o processo de autoavaliação do curso e permitir a busca por melhoramentos contínuos:

I. Políticas: definidas pela CAPES;

II. Diretrizes: conjunto de normas e recomendações elaboradas pela Pró-Reitoria de Pesquisa e Pós-Graduação da UFC, responsável pela coordenação dos processos internos desta instituição, de sistematização e de prestação das informações solicitadas pelos órgãos de fiscalização;

III. Instrumentos: seminários de informação e sensibilização destinados aos diferentes públicos-alvo (gestores/coordenadores do Programa, docentes e funcionários técnico- administrativos participantes deste Programa, e discentes), questionários (impressos ou online) e entrevistas a serem aplicadas. Além de analisar seus resultados analisados periodicamente e, em caso de necessidade, realizar pesquisas de campo para identificar tendências ou obter maiores informações sobre situações detectadas.

Art. 4o. A autoavaliação deve ser processo de construção coletiva e contínua, que utiliza diversos recursos metodológicos e instrumentos, objetivando retratar a complexidade e a riqueza do PPGEEC, e efetivar mudanças, a partir das fragilidades e potencialidades detectadas.

Parágrafo único. O processo de autoavaliação compreende as seguintes etapas:

- a) Preparação: constituição da Comissão Autoavaliação (CAA) do PPGEEC; elaboração do Plano de Autoavaliação; aprovação do Plano de Autoavaliação pelo colegiado do PPGEEC; e, sensibilização dos diversos atores envolvidos (gestores, docentes, técnicos e discentes);
- b) Desenvolvimento: implementação do Plano de Autoavaliação, com planejamento de ações, coleta de dados e análise de resultados;
- c) Consolidação do processo: elaboração de relatórios críticos, divulgação de resultados (para docentes, discentes e técnicos administrativos), acompanhamento à efetivação de mudanças e envio pela Coordenação à CAPES dos procedimentos de autoavaliação.

CAPÍTULO III DA OBRIGATORIEDADE

Art. 5o. O Plano de Autoavaliação do Programa detalha a operacionalização do processo de autoavaliação do Programa, componente obrigatória das atividades operacionais do Programa, conforme definido pela CAPES, em seu Relatório do Grupo de Trabalho de Autoavaliação de Programas de Pós-Graduação, 2019.

Art. 6o. A avaliação do PPGEEC compreenderá o processo de formação acadêmica do corpo discente (avaliação dos parâmetros internos) e o produto do processo de formação (avaliação do nível e da qualidade dos conteúdos apreendidos pelos diplomados egressos do Programa).

Parágrafo Único. A avaliação terá como referência as informações fornecidas ao Sistema Nacional de Pós-Graduação (SNPG) por meio da plataforma SUCUPIRA e seguirá as diretrizes da Portaria 14/2018 que dispõe sobre processos avaliativos das propostas de cursos e dos programas de pós-graduação stricto sensu em funcionamento.

Art. 7o. A avaliação permitirá confrontar o planejado versus o realizado e fornecerá observações e idéias importantes para identificar os aspectos a serem corrigidos, melhorados ou ainda otimizados, no processo de formação acadêmica do PPGEEC. O processo de avaliação interna será realizado anualmente (ou a cada período letivo, em determinadas avaliações) e envolverá os seguintes aspectos:

§ 1o. Avaliação dos recursos humanos do curso, sob os aspectos da qualificação, adequação ao perfil de trabalho e produção:

- a) Corpo docente;
- b) Corpo técnico-administrativo.

§ 2o. Avaliação da proposta curricular:

- a) Estrutura curricular do curso, a fim de assegurar a aderência às diretrizes da CAPES (Portaria n. 182/2018);
- b) Conteúdos das disciplinas a fim de assegurar a aderência à área de concentração e linhas de pesquisa do programa;
- c) Ementas e programas das disciplinas;
- d) Referências bibliográficas adotadas nas disciplinas, editais e similares;
- e) Orientação acadêmica das dissertações;
- f) Avaliação contínua da proposta do programa.

§ 3o. Avaliação da infraestrutura:

- a) Física: salas de aula, laboratórios e biblioteca (se atendem à quantidade de alunos admitidos no curso e aos propósitos acadêmicos do programa);
- b) Equipamentos: hardware (se atendem às exigências tecnológicas, em função dos propósitos técnicos e acadêmicos do curso);
- c) Equipamentos: softwares (se atendem às exigências tecnológicas, em função dos propósitos técnicos e acadêmicos do curso).

§ 4o. Implementação das atividades do Programa:

- a) Metodologia de funcionamento e frequência das reuniões pedagógicas;
- b) Estruturação e metodologia de funcionamento do colegiado do Programa.

§ 5o. Avaliação das interações do Programa:

- a) Integração do curso com a graduação (via projetos e atividades acadêmicas);
- b) Integração com outras instituições públicas ou privadas, a nível municipal, estadual e federal, via projetos de pesquisa ou de extensão e acordos de cooperação técnica;
- c) Integração com a sociedade civil, através de projetos de pesquisa ou extensão e políticas de inclusão.

§ 6o. Avaliação dos discentes:

- a) Desempenho nas disciplinas;
- b) Desempenho nas dissertações;

c) Produção intelectual discente.

§ 7o. Acompanhamento dos egressos:

- a) Percepção dos egressos em relação ao programa;
- b) Desempenho dos egressos em suas atividades após o mestrado.

CAPÍTULO IV DA COMISSÃO INTERNA DE AVALIAÇÃO

Art. 8o. A Comissão de Autoavaliação (CAA-PPGEEC) será constituída por, no mínimo, 5 (cinco) membros, com total autonomia sobre a matéria, sendo assim composta:

- a) 01 (um) coordenador (membro do corpo docente);
- b) 01 (um) coordenador-adjunto (membro do corpo docente);
- c) 01 (um) técnico-administrativo;
- d) 01 (um) representante do corpo discente;
- e) 01 (um) representante egresso;
- f) Outros professores quando se julgar necessário.

§1o. Todos os membros serão indicados pelo Colegiado do PPGEEC, pelo período de mandato de 01 (um) ano, sendo renovável para até 04 (quatro) anos, correspondente à avaliação quadrienal da CAPES.

§2o. O mesmo Colegiado indicará um dos membros para coordenar esta Comissão.

§3o. Com exceção do técnico administrativo, do representante do corpo discente e do representante egresso, somente professores permanentes e colaboradores no PPGEEC poderão compor a Comissão.

CAPÍTULO V DA IMPLEMENTAÇÃO DA AUTOAVALIAÇÃO

Art. 9o. As ações de autoavaliação julgadas pertinentes incluem:

- a) Avaliar as políticas de inclusão de estudantes em situação econômica desfavorecida e dos portadores de necessidades especiais;
- b) Avaliar a participação dos discentes em programas acadêmicos: estágios, avaliação institucional e atividades de intercâmbio estudantil;
- c) Aprimorar os meios de comunicação, interna e externa, relativos ao processo de autoavaliação no âmbito do curso;
- d) Implementar ações efetivas e contínuas de promoção de qualificação acadêmica de docentes e discentes;
- e) Avaliar o acervo (diversidade e quantidade de títulos) na área do PPGEEC da Biblioteca do Campus de Sobral;
- f) Estabelecer, e manter ao longo do tempo, a coerência entre bibliotecas, laboratórios, equipamentos de informática e as práticas pedagógicas dos docentes;
- g) Consolidar a cultura de avaliação;

- h) Consolidar a CAA-PPGEEC conforme sua legitimidade e propiciar maior reconhecimento no âmbito do Programa;
- i) Fortalecer a relação entre a autoavaliação e o planejamento;
- j) Estimular a participação/acompanhamento dos docentes e discentes das regras de avaliação da pós-graduação pela CAPES;
- k) Avaliar os mecanismos de acompanhamento dos egressos.

Art. 10o. Um calendário de ações de autoavaliação deverá ser elaborado antes do início do ano letivo pela Comissão de Autoavaliação (CAA) detalhando:

- a) Escolha das ações que serão implementadas nos dois períodos letivos seguintes: seminários de informação e sensibilização e/ou questionários e/ou entrevistas estruturadas e/ou pesquisa de campo, etc.;
- b) Identificação dos responsáveis pela execução, supervisão e informação às partes interessadas das ações a serem implementadas, bem como dos públicos-alvo;
- c) Definição das datas de realização das ações de autoavaliação, levando-se em conta outras atividades acadêmicas do PPGEEC, bem como feriados (nacionais e municipais).

Art. 11o. As ações de autoavaliação serão levadas a termo nas datas definidas no calendário proposto, operacionalizadas pelos professores e técnicos administrativos do Programa, com a participação e sob a supervisão da Comissão de Autoavaliação.

CAPÍTULO VI DA AVALIAÇÃO DOS RESULTADOS

Art. 12o. Durante o período-letivo, os dados relativos às ações de autoavaliação realizadas naquele período serão coletados, verificados, validados e consolidados.

Art. 13o. Os dados consolidados serão objeto de tabulação pelos membros da Comissão de Autoavaliação, e subsequente produção de relatório integrado relativo ao período letivo analisado.

Art. 14o. A análise dos dados tabulados e dos gráficos procurar-se-á identificar:

- a) Médias, medianas, modas, normais, desvios padrão das distribuições de dados;
- b) Identificação de tendências ou de afastamentos da média, quando comparados com dados obtidos pelo Programa em períodos anteriores, ou com dados de outros cursos similares (na região ou no país);
- c) Identificação e quantificação dos impactos proporcionados pelas medidas corretivas adotadas no(s) período(s) anterior(es).

CAPÍTULO VII DAS ATRIBUIÇÕES E DEVERES

Art. 15o. Compete à Comissão de Autoavaliação do PPGEEC:

- a) Articular-se com o Colegiado do Programa para comunicar sobre as diretrizes, organização e desenvolvimento dos trabalhos de autoavaliação;
- b) Elaborar o calendário de todas as ações de autoavaliação e submetê-lo à apreciação (e aprovação) do Colegiado do Programa, conforme as normas estabelecidas neste regulamento;
- c) Sensibilizar o corpo docente e discente para a importância do processo de autoavaliação do Programa, de maneira sistemática e contínua;
- d) Elaborar, atualizar e aplicar junto ao corpo docente e discente do curso os questionários de avaliação interna, segundo o calendário de ações aprovado;
- e) Tirar dúvidas do corpo docente e discente, quanto ao preenchimento dos questionários de autoavaliação do Programa;
- f) Coletar os dados e analisar resultados das ações de avaliação do Programa;
- g) Gerar relatórios críticos, divulgar resultados junto ao Colegiado do Programa e acompanhar a efetivação de mudanças;
- h) Convocar, sempre que necessário, reuniões com o Colegiado do Programa, para discutir questões relativas à implementação de ações de melhoramentos no Programa;
- i) Arquivar, junto à Coordenação do Programa, os documentos referentes às ações da Comissão de Autoavaliação;
- j) Elaborar e emitir a cada período letivo e, ao final do ano letivo, os resultados das avaliações realizadas;
- k) Encaminhar à Coordenação do Programa, ao final de cada período letivo e de cada ano letivo, os resultados das avaliações das disciplinas e professores;
- l) Reunir-se ordinariamente uma vez por mês, ou extraordinariamente quando necessário;
- m) Tomar as medidas necessárias para o cumprimento do presente regulamento.

Art. 16o. Compete à Coordenação do PPGEEC:

- a) Encaminhar ao colegiado do PPGEEC as portarias de participação dos membros da Comissão de Autoavaliação (CAA-PPGEEC);
- b) Informar sobre o calendário das ações de avaliação elaborado pela Comissão de Autoavaliação;
- c) Divulgar os dados gerais do relatório e discutir no âmbito do Colegiado do Programa os resultados das ações de autoavaliação, analisando as medidas corretivas propostas, bem como as modalidades de implementação das ações corretivas;
- d) Requisitar a colaboração proativa dos professores do Programa para a implementação das ações corretivas que se fizerem necessárias;
- e) Acompanhar, juntamente com a Comissão de Autoavaliação, o processo de implementação das ações corretivas decididas, bem como dos resultados alcançados;
- f) Cumprir o que está previsto neste regulamento.

Art. 17o. Compete aos professores e técnicos-administrativos do PPGEEC:

- a) Participar das reuniões programadas pela Comissão de Autoavaliação e pela Coordenação do Programa, para discutir os problemas encontrados e a implementação de ações visando a melhoria do funcionamento operacional do curso;
- b) Participar de maneira proativa das ações visando a melhoria do funcionamento do curso;
- c) Cumprir o que está previsto neste regulamento.

Art. 18o. Compete aos estudantes do PPGEEC:

- a) Participar nos prazos determinados das ações relacionadas ao processo de avaliação do curso;
- b) Tirar todas as dúvidas e responder de maneira proativa aos questionários de avaliação ao final de cada semestre letivo;
- c) Procurar se inteirar dos resultados das ações de autoavaliação e das modificações propostas objetivando o processo de melhoria operacional contínua do Programa;
- d) Cumprir o que está previsto neste regulamento.

Art. 19o. Compete ao Coordenador da Comissão de Autoavaliação:

- a) Assegurar canal de comunicação permanente, junto ao corpo discente, técnico-administrativo e docente do PPGEEC, para queixas, recursos, sugestões, etc.
- b) Participar das reuniões da Comissão de Autoavaliação, assegurando que o processo de autoavaliação em sua globalidade seja pautado pela impessoalidade e o respeito às normas e diretrizes de autoavaliação vigentes;
- c) Assegurar a adequada implementação pelo CAA-PPGEEC das atividades de autoavaliação do curso;
- d) Cumprir o que está previsto neste regulamento.

CAPÍTULO VIII DAS DISPOSIÇÕES FINAIS

Art. 20o. Os casos omissos serão resolvidos pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Federal do Ceará – Campus de Sobral (PPGEEC/UFC).

Art. 21o. O presente Regulamento entra em vigor na data de sua aprovação.

Art. 22o. Revogam-se as disposições em contrário.

Sobral/CE, 15 de janeiro de 2021.

Prof. Iális Cavalcante de Paula Júnior
Coordenador PPGEEC/UFC



ANEXO II DA RESOLUÇÃO No 01/2021

PLANO DE AUTOAVALIAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E COMPUTAÇÃO – PPGEEC

1) Introdução

O Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Federal do Ceará – Campus de Sobral (PPGEEC/UFC) oferece o curso de Mestrado desde 2013, e mantém um perfil de qualificação acadêmica atestado pela CAPES. Alinhado à sua proposta de planejamento estratégico para o quadriênio 2021-2024, sua missão é: Criar e disseminar o conhecimento para formar profissionais qualificados nas áreas relacionadas à engenharia elétrica e de computação por meio da pesquisa científica, do desenvolvimento tecnológico e da inovação em prol das demandas da sociedade. Em sua visão, encontra-se: Ser um programa de pós-graduação de excelência, com impacto regional, nacional e internacional, através do desenvolvimento de pesquisas e parcerias com instituições públicas e privadas. E dentre seus valores estão: Ética, Excelência, Inovação, Compromisso, Qualidade, Produção Internacional, Interdisciplinaridade.

As duas áreas de concentração do PPGEEC envolvem: Sistemas de Informação, que contempla duas linhas de pesquisa: Sistemas de comunicação e Algoritmos e Computação Distribuída; e a área de Sistemas de Energia Elétrica, que possui a linha de pesquisa Eletrônica de potência.

Tomando como base as diretrizes da CAPES para o aprimoramento do processo de avaliação da pós-graduação no Brasil, este documento traz instrumentos que serão utilizados para a autoavaliação do PPGEEC, levando-se em conta o Regulamento do Plano de Autoavaliação do Programa.

2) Objetivos específicos

O processo de autoavaliação do PPGEEC se pauta e busca contribuir para o planejamento estratégico do Programa a curto, médio e longo prazos. O plano de autoavaliação tem como objetivos específicos alcançar o seguinte:

a) Elevar o impacto acadêmico e social do programa;

- b) Focar na formação discente pós-graduada na perspectiva da inserção social, científica e profissional;
- c) Reduzir o número de evasão discente;
- d) Contribuir para a maior empregabilidade dos egressos;
- e) Aumentar a produção de docentes inserida nos estratos A1, A2, A3 e A4, do Qualis periódicos da área de Engenharia IV na CAPES, preferencialmente de forma simétrica entre os docentes;
- f) Aumentar a produção de discentes e de egressos em periódicos do Qualis da área de Engenharia IV na CAPES, especialmente nos estratos A1, A2, A3 e A4;
- g) Aumentar o percentual de produção discente em relação à produção total do programa, especialmente no que se refere a artigos de docente com participação de discentes do PPGEEC;
- j) Elevar o número de docentes bolsistas de produtividade em pesquisa do CNPq e Funcap;
- h) Incrementar a internacionalização, principalmente no desenvolvimento de projetos conjuntos de cooperação com financiamento bilateral;
- i) Alcançar as metas exigidas pela CAPES para abertura de um curso de doutorado;
- j) Possibilitar que a estrutura curricular e o conjunto de atividades realizadas no programa sejam consistentes e coerentes com a área de concentração, como as linhas de pesquisa e com os projetos desenvolvidos no PPGEEC;
- k) Criar condições favoráveis para que o programa obtenha conceito “Muito Bom” em todos os indicadores numéricos da área de avaliação;
- l) Elevar a nota do PPGEEC de 3 para 4 na próxima avaliação da CAPES.

3. Instrumentos de autoavaliação e cronograma

3.1. Instrumentos de autoavaliação

- a) Ficha de avaliação da secretaria
- b) Ficha de avaliação da coordenação
- c) Ficha de avaliação dos discentes em relação aos docentes (disciplinas)
- d) Ficha de avaliação dos discentes em relação aos docentes (orientações)
- e) Ficha de avaliação dos docentes em relação ao desempenho dos discentes das disciplinas ministradas
- f) Ficha de avaliação dos docentes em relação ao desempenho dos discentes nas orientações
- g) Ficha de avaliação das disciplinas
- h) Ficha de avaliação da banca de dissertação
- i) Ficha de avaliação de eventos (quando ocorrerem)
- j) Ficha de avaliação do egresso sobre o desempenho do PPGEEC
- k) Ficha de avaliação da infraestrutura
- l) Tabela de Produção docente
- m) Tabela de Produção discente

3.2. Cronograma

Pretende-se que o cronograma siga a avaliação quadrienal da CAPES, de 2021 a 2024.

Atividades/Avaliações	2021		2022		2023		2024	
	1º	2º	1º	2º	1º	2º	1º	2º
Da secretaria		X		X		X		X
Da coordenação		X		X		X		X
Dos discentes em relação aos docentes (disciplinas)	X	X	X	X	X	X	X	X
Dos discentes em relação aos docentes (orientações)	X		X		X		X	
Dos docentes em relação ao desempenho dos discentes nas disciplinas	X	X	X	X	X	X	X	X
Dos docentes em relação ao desempenho dos discentes nas orientações	X		X		X		X	
Das disciplinas	X	X	X	X	X	X	X	X
Da banca de dissertação	X	X	X	X	X	X	X	X
Dos eventos (quando ocorrerem)	X	X	X	X	X	X	X	X
Do egresso sobre o desempenho do PPGEEC	X	X	X	X	X	X	X	X
Da infraestrutura		X		X		X		X
Produção Docente		X		X		X		X
Produção Discente		X		X		X		X

Nota: o cronograma proposto é uma previsão.

3.3. Detalhamento dos instrumentos de autoavaliação

Opções de preenchimento dos instrumentos: Ótimo / Bom / Regular / Ruim / Péssimo / Não conheço / Não se aplica

a) Ficha de avaliação da secretaria

1. A secretaria funciona dentro dos horários previstos pela UFC?
2. A secretaria atende o público com cortesia e respeito?
3. O contato com a secretaria por meio eletrônico é satisfatório?
4. A secretaria disponibiliza os documentos de acordo com o Regimento do PPGEEC?
5. Os procedimentos da secretaria são céleres e adequados?
6. Indique no espaço abaixo os serviços que a secretaria pode melhorar: (resposta subjetiva)

b) Ficha de avaliação da coordenação

1. A coordenação disponibiliza horário de atendimento presencial?
2. A coordenação atende o público com cortesia e respeito?
3. O contato com a coordenação por meio eletrônico é satisfatório?
4. A coordenação orienta academicamente de forma satisfatória?
5. Os procedimentos da coordenação são céleres e adequados?
6. Indique no espaço abaixo os serviços que a coordenação pode melhorar: (resposta subjetiva)

c) Ficha de avaliação dos discentes em relação aos docentes (disciplinas)

Nome da disciplina: (descrição)

Ano/semestre: (descrição)

1. O professor responsável pela disciplina demonstra domínio do conteúdo ministrado?
2. O professor responsável pela disciplina demonstra domínio da bibliografia indicada?
3. Considera que a metodologia adotada pelo professor responsável possibilitou o aprofundamento e desenvolvimento de análise crítica em relação aos conteúdos ministrados?
4. O professor responsável incentiva a participação dos alunos nos debates que ocorrem em sala de aula?
5. O professor responsável incentiva o desenvolvimento de pensamento crítico a respeito dos assuntos ministrados?
6. O professor responsável está aberto à discussão de ideias distintas daquelas por ele defendidas?

d) Ficha de avaliação dos discentes em relação aos docentes (orientações)

1. O orientador disponibiliza horário de atendimento presencial/remoto?
2. O contato com o orientador por meio eletrônico é satisfatório?
3. O orientador orienta academicamente de forma satisfatória?
4. Os procedimentos do orientador são céleres e adequados?

5. O orientando colaborou para a orientação do professor?
6. Indique no espaço abaixo os aspectos que podem ser melhorados pelo orientador:
(resposta subjetiva)

e) Ficha de avaliação dos docentes em relação ao desempenho dos discentes das disciplinas ministradas

Nome da disciplina: (descrição)

Ano/semestre: (descrição)

1. Os alunos dedicaram-se às atividades propostas?
2. Os alunos demonstraram interesse nas discussões propostas?
3. Os alunos demonstraram domínio de conteúdos básicos necessários à compreensão dos temas sugeridos para discussão?
4. Os alunos procuraram realizar o estudo da bibliografia indicada?
5. Os alunos procuraram ir além da bibliografia indicada?
6. Considera satisfatório o desempenho dos alunos nas atividades avaliativas?
7. Os alunos procuraram participar ativamente das discussões que ocorreram em sala de aula?

f) Ficha de avaliação dos docentes em relação ao desempenho dos discentes nas orientações

1. O orientando dedicou-se às atividades propostas?
2. O orientando demonstrou interesse nas discussões propostas?
3. O orientando demonstrou domínio de conteúdos básicos necessários aos temas sugeridos para discussão?
4. O orientando procurou realizar o estudo da bibliografia indicada?
5. O orientando procurou ir além da bibliografia indicada?
6. O orientando compareceu às reuniões nos horários e dias previstos?
7. Considera satisfatório o desempenho do orientando nas atividades relativas à dissertação?
8. Em termos gerais, como você classifica o trabalho final (dissertação)?

g) Ficha de avaliação das disciplinas

Nome da disciplina: (descrição)

Ano/semestre: (descrição)

1. Considera a disciplina adequada à área de concentração do programa?
2. A disciplina foi ministrada regularmente nos horários e dias previstos?
3. A disciplina foi desenvolvida adequadamente pelo docente responsável?
4. O docente seguiu o conteúdo proposto na ementa da disciplina?
5. O docente abordou todo o conteúdo proposto na ementa da disciplina?

6. A bibliografia adotada é pertinente à ementa da disciplina?
7. O acesso à bibliografia ocorreu satisfatoriamente?
8. O conteúdo programático da disciplina é atual?
9. Os discentes colaboraram (dedicaram-se) para o bom desempenho da disciplina?

h) Ficha de avaliação da banca de dissertação

1. A banca ocorreu dentro dos horários previstos pelo PPGEEC?
2. A banca avaliou criteriosamente o candidato?
3. O presidente da banca conduziu satisfatoriamente a banca de dissertação?
4. A defesa do mestrando/doutorando foi adequada?
5. O candidato demonstrou domínio sobre o tema da pesquisa?
6. O candidato demonstrou domínio sobre a(s) metodologia(s) adotada(s) na pesquisa?
7. A dissertação tem afinidade com a área de concentração do PPGEEC?
8. A dissertação tem afinidade com as linhas de pesquisa do PPGEEC?
9. A dissertação tem contribuição científica relevante para a sociedade?

i) Ficha de avaliação de eventos (quando ocorrerem)

1. Considera o evento adequado à área de concentração do programa?
2. Considera o evento adequado às linhas de pesquisa do programa?
3. A apresentação foi ministrada nos horários e dias previstos?
4. A apresentação foi desenvolvida adequadamente pelo responsável?
5. O local, dia e horário da apresentação foram adequadamente divulgados?
6. O acesso ao local da apresentação ocorreu satisfatoriamente?
7. A audiência colaborou para o bom desempenho do apresentador?

j) Ficha de avaliação do egresso sobre o desempenho do PPGEEC

1. O conteúdo das disciplinas oferecidas contribuiu para o aprofundamento na área de concentração do PPGEEC?
2. O conteúdo das disciplinas oferecidas contribuiu para o aprofundamento nas linhas de pesquisa do PPGEEC?
3. A metodologia empregada nas disciplinas permite o aprofundamento e análise crítica dos conteúdos ministrados?
4. A orientação proporcionada pelo orientador auxiliou no desenvolvimento da dissertação?
5. O ciclo acadêmico proposto pelo PPGEEC contribuiu para o andamento do curso?
6. De forma geral, considera que o Curso foi relevante para seu crescimento acadêmico e profissional?
7. O que você julga serem as principais razões para a evasão discente, quando ocorrem? (resposta subjetiva)

8. Indique no espaço abaixo sugestões para melhoria do PPGEEC: (resposta subjetiva)

k) Ficha de avaliação da infraestrutura

1. Salas de aula: quantidade
2. Salas de aula: qualidade das instalações (piso, ventilação, iluminação)
3. Salas de aula: acessibilidade para deficientes e ou mobilidade reduzida
4. Salas de aula: limpeza e conservação
5. Salas de estudo
6. Acesso a equipamentos de informática
7. Disponibilidade de internet e serviços web da instituição
8. Laboratórios: quantidade
9. Laboratórios: qualidade
10. Instalações administrativas (secretaria)
11. Biblioteca (área de concentração e linhas de pesquisa do PPGEEC): acervo quantidade
12. Biblioteca (área de concentração e linhas de pesquisa do PPGEEC): acervo qualidade
13. Indique no espaço abaixo sugestões para melhoria da infraestrutura do PPGEEC: (resposta subjetiva)

l) Produção docente

Ano	
Nome	
Disciplinas ministradas	
Graduação	
Pós-graduação	
Artigos publicados	
Internacionais	
A1; A2; A3; A4	
B1; B2; B3; B4	
Sem qualis	
Nacionais	

A1; A2; A3; A4	
B1; B2; B3; B4	
Sem qualis	
Livros	
Livros publicados	
Capítulos de livros	
Participação em eventos	
Eventos internacionais	
Eventos nacionais	
Anais de eventos	
Eventos internacionais	
Eventos nacionais	
Produção Técnica	
Trabalhos técnicos	
Editoria de periódicos	
Revisor de periódicos	
Bancas	
Doutorado	
Mestrado	
Orientações	
Iniciação científica	
Graduação	
Pós-graduação	
Processos seletivos	
Organização de eventos	

Nacionais	
Internacionais	
Financiamento de pesquisa	
Agência de fomento	
Setor privado	
Extensão	
Capacitações	
Grupos de pesquisa	

m) Produção discente

Ano	
Nome	
Artigos publicados	
Internacionais	
A1; A2; A3; A4	
B1; B2; B3; B4	
Nacionais	
A1; A2; A3; A4	
B1; B2; B3; B4	
Livros	
Livros publicados	
Capítulos de livros	
Participação em eventos	
Eventos internacionais	
Eventos nacionais	
Anais de eventos	
Eventos internacionais	

Eventos nacionais	
Produção Técnica	
Trabalhos técnicos	
Revisor de periódicos	
Organização de eventos	
Nacionais	
Internacionais	
Extensão	