
Módulo de Avaliação Semiautomática do Sistema de Candidatura em Processos Seletivos da Pós-Graduação

Ihury Kewin de Oliveira Costa



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Ihury Kewin de Oliveira Costa

**Módulo de Avaliação Semiautomática do
Sistema de Candidatura em Processos Seletivos
da Pós-Graduação**

Trabalho de Conclusão de Curso de Desenvolvimento de Software apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Luiz Gustavo Almeida Martins

Uberlândia - MG

2026

A Deus, à minha família e aos meus amigos.

Agradecimentos

Agradeço, primeiramente, a Deus, por me sustentar e iluminar ao longo de toda esta caminhada.

À minha família, pelo apoio e pelo incentivo constante em cada etapa, e aos meus amigos, pela companhia e pelo encorajamento ao longo de toda a jornada.

Ao meu orientador, Prof. Luiz Gustavo Almeida Martins, pela orientação, pela disponibilidade e pelas valiosas contribuições em todas as etapas deste trabalho.

Aos meus colegas de curso, pelo aprendizado que construímos juntos e pelas oportunidades — em especial as profissionais — que surgiram a partir dessa convivência. Devo a essa trajetória e à formação recebida boa parte da condição de vida que tenho hoje.

À Universidade Federal de Uberlândia e à Faculdade de Computação, pela formação de qualidade, e aos professores do curso de Bacharelado em Ciência da Computação, pelos conhecimentos compartilhados ao longo desta jornada.

“Aquilo que temos de aprender a fazer, aprendemos fazendo.”
(Aristóteles)

Resumo

Os processos seletivos para programas de pós-graduação envolvem etapas críticas de submissão e avaliação documental e curricular, sendo esta última frequentemente realizada de forma manual, o que pode acarretar inconsistências, erros, baixa rastreabilidade e elevado esforço operacional. No contexto da Universidade Federal de Uberlândia (UFU), o sistema SIPPGCO-FACOM informatizou a etapa de submissão de candidaturas, porém a fase de avaliação ainda apresentava limitações relacionadas à padronização e eficiência. Este trabalho apresenta o desenvolvimento de um módulo de avaliação semiautomática integrado ao sistema existente, com o objetivo de apoiar a comissão avaliadora na execução de tarefas repetitivas e sistemáticas, deixando para os avaliadores apenas aquelas que dependem de julgamento de mérito. A solução contempla uma etapa inicial de triagem documental, na qual cada documento obrigatório é conferido individualmente e todo indeferimento é justificado; o cálculo automático da pontuação curricular a partir das regras definidas em edital; etapas avaliativas sem documento associado, como a entrevista, cujas notas são lançadas pela comissão e combinadas por soma ponderada; ajustes manuais com reprocessamento imediato das notas; a geração da classificação dos candidatos, que combina a nota de corte e o número de vagas de cada linha de pesquisa; e a produção de relatórios estruturados, promovendo maior transparência, consistência e auditabilidade do processo. A arquitetura adotada baseia-se em uma abordagem em camadas, composta por uma aplicação web em Angular e uma API REST construída com TypeScript e NestJS, com persistência em banco PostgreSQL compartilhado com o sistema de submissão. O sistema contempla requisitos funcionais e não funcionais relacionados à usabilidade, desempenho, segurança e confiabilidade — incluindo a segregação entre os perfis de coordenação e de comissão avaliadora, trilha de auditoria completa e diretrizes de conformidade com a legislação de proteção de dados — e foi validado por meio de testes automatizados unitários e de integração, complementados por testes funcionais na interface. A solução amplia a rastreabilidade das decisões e padroniza as validações, mantendo a flexibilidade necessária para intervenção da comissão avaliadora, e tem como benefícios esperados a redução do esforço manual e de inconsistências, a serem confir-

mados com o uso em processos seletivos reais. Além disso, o trabalho apresenta uma arquitetura reutilizável e adaptável a outros contextos de avaliação em processos seletivos acadêmicos.

Palavras-chave: avaliação documental. automação de processos. sistemas de informação. processos seletivos. rastreabilidade.

Abstract

Graduate program admission processes involve critical stages of application submission and documentary and curricular evaluation, the latter of which is frequently carried out manually, potentially leading to inconsistencies, human error, low traceability and high operational effort. In the context of the Universidade Federal de Uberlândia (UFU), the SIPPGCO-FACOM system automated the application submission stage; however, the evaluation phase still presented limitations related to standardization and efficiency. This work presents the development of a semi-automatic evaluation module integrated into the existing system, aimed at supporting the evaluation committee in performing repetitive and systematic tasks, leaving to the evaluators only those that depend on judgement of merit. The solution encompasses an initial document screening stage, in which every required document is individually reviewed and each rejection is justified; the automatic calculation of curricular scores based on the rules defined in the selection notice; evaluation stages with no associated document, such as the interview, whose grades are recorded by the committee and combined through a weighted sum; manual adjustments with immediate score reprocessing; the generation of the candidate ranking, which combines the cut-off score and the number of positions available in each research line; and the production of structured reports, thereby promoting greater transparency, consistency, and auditability of the process. The adopted architecture follows a layered approach, composed of an Angular web application and a REST API built with TypeScript and NestJS, persisting data in a PostgreSQL database shared with the submission system. The system addresses functional and non-functional requirements related to usability, performance, security, and reliability — including the segregation between the coordination and evaluation committee profiles, a complete audit trail and data protection compliance guidelines — and was validated through automated unit and integration tests, complemented by functional testing of the user interface. The solution increases decision traceability and standardizes validations, while maintaining the flexibility required for evaluation committee intervention, and its expected benefits — reduced manual effort and fewer inconsistencies — remain to be confirmed through use in real admission processes. Furthermore, this work

presents a reusable and adaptable architecture applicable to other evaluation contexts in academic admission processes.

Keywords: document evaluation. process automation. information systems. selection processes. traceability.

Lista de ilustrações

Figura 1 – Diagrama de Casos de Uso	22
Figura 2 – Modelo de Domínio do Sistema	38
Figura 3 – Diagrama de Sequência - RF005	39
Figura 4 – Diagrama de Sequência - RF008	40
Figura 5 – Diagrama de Estado - Triagem	41
Figura 6 – Diagrama de Classes do Sistema	44
Figura 7 – Ciclo de Desenvolvimento Orientado por Especificação	47
Figura 8 – Arquitetura do Sistema	50
Figura 9 – Tela de Autenticação	62
Figura 10 – Tela de Seleção do Edital	63
Figura 11 – Visão Geral do Edital	63
Figura 12 – Formulário de Configuração dos Critérios	64
Figura 13 – Aba de Configuração do Barema	65
Figura 14 – Diálogo de Cadastro de Seção	66
Figura 15 – Fila de Triagem	68
Figura 16 – Detalhe da Triagem	68
Figura 17 – Reversão de Indeferimento	70
Figura 18 – Fila de Avaliações	70
Figura 19 – Detalhe da Avaliação	71
Figura 20 – Lançamento de Nota de Etapa	72
Figura 21 – Tela de Classificação	74
Figura 22 – Parecer Preliminar	74
Figura 23 – Relatório de Triagem	75
Figura 24 – Relatório de Classificação	76
Figura 25 – Trilha de Auditoria	76

Lista de tabelas

Tabela 1	–	Requisitos de <i>hardware</i> do servidor.	53
Tabela 2	–	Plano de testes funcionais executado na interface.	58
Tabela 3	–	Endpoints da API REST do módulo de avaliação.	87
Tabela 4	–	Suítes de testes unitários.	90
Tabela 5	–	Cenários de teste de integração de ponta a ponta.	91

Lista de siglas

API Interface de Programação de Aplicações - *Application Programming Interface*

CSV Valores Separados por Vírgula - *Comma-Separated Values*

DTO Objeto de Transferência de Dados - *Data Transfer Object*

FACOM Faculdade de Computação

HTTP Protocolo de Transferência de Hipertexto - *Hypertext Transfer Protocol*

HTTPS Protocolo de Transferência de Hipertexto Seguro - *Hypertext Transfer Protocol
Secure*

JWT JSON Web Token

LGPD Lei Geral de Proteção de Dados

ORM Mapeamento Objeto-Relacional - *Object-Relational Mapping*

PDF Formato de Documento Portátil - *Portable Document Format*

PPGCO Programa de Pós-Graduação em Ciência da Computação

REST Transferência de Estado Representacional - *Representational State Transfer*

RF Requisito Funcional

RNF Requisito Não Funcional

SBC Sociedade Brasileira de Computação

SIPPGCO Sistema de Inscrição do Programa de Pós-Graduação em Ciência da Computação

SPA Aplicação de Página Única - *Single-Page Application*

SQL Linguagem de Consulta Estruturada - *Structured Query Language*

UFU Universidade Federal de Uberlândia

UML Linguagem de Modelagem Unificada - *Unified Modeling Language*

Sumário

1	INTRODUÇÃO	16
1.1	Contextualização do Trabalho	16
1.2	Problema e Justificativas	17
1.3	Objetivos (Gerais e Específicos)	18
1.4	Contribuições Esperadas	19
1.5	Organização do Trabalho	19
1.6	Uso de Inteligência Artificial Generativa	20
2	REQUISITOS DO SISTEMA	21
2.1	Requisitos Funcionais – RF	21
2.1.1	Configuração dos Critérios de Avaliação	23
2.1.2	Triagem das Inscrições	25
2.1.3	Avaliação dos Candidatos	26
2.1.4	Classificação e Divulgação dos Resultados	28
2.2	Requisitos Não Funcionais – RNF	30
2.2.1	Usabilidade	30
2.2.2	Confiabilidade	31
2.2.3	Desempenho	32
2.2.4	Segurança	32
2.2.5	Distribuição	33
2.2.6	Padrões	33
2.2.7	Hardware e Software	33
3	ANÁLISE E PROJETO DO SISTEMA	35
3.1	Justificativa Arquitetural	35
3.2	Modelo de Domínio do Sistema	36
3.3	Diagramas de Sequência do Sistema	37
3.3.1	RF005 — Sincronizar e Processar Avaliações	37

3.3.2	RF008 — Ajuste Manual com Reprocessamento Automático	38
3.4	Diagrama de Estado — Ciclo de Vida da Triagem	39
3.5	Modelagem de Dados	41
3.5.1	Modelo Lógico de Dados	41
3.5.2	Modelo Físico de Dados	42
3.6	Diagrama de Classes do Sistema	43
3.7	Rastreabilidade Requisitos–Projeto	43
4	DESENVOLVIMENTO DO SISTEMA	45
4.1	Processo de Desenvolvimento	45
4.1.1	Desenvolvimento Orientado por Especificação com Apoio de Inteligência Artificial	46
4.2	Arquitetura do Sistema	49
4.3	Padrões de Projeto Utilizados	50
4.4	Tecnologias Utilizadas no Desenvolvimento	51
4.4.1	Linguagens de desenvolvimento	51
4.4.2	<i>Frameworks</i>	51
4.4.3	Bibliotecas	51
4.4.4	Ferramentas Auxiliares	52
4.4.5	Sistemas e Componentes Externos Utilizados	52
4.5	Instalação e Configurações	53
4.5.1	Pré-requisitos do Sistema	53
4.5.2	Processo de Instalação/Execução	54
4.5.3	Variáveis de Ambiente	54
4.5.4	Arquivos de Configuração	55
4.6	Compilação e Implantação	55
4.6.1	Processo de Compilação do Sistema	55
4.6.2	Opções de Implantação do Sistema	55
4.6.3	Configurações para Produção	55
4.7	Repositório de Código	56
4.7.1	Link para o Repositório	56
4.7.2	Estrutura de Diretórios	56
4.7.3	Guia de Contribuições	57
4.8	Testes do Sistema	57
4.8.1	Plano de Testes	57
4.8.2	Execução do Plano de Testes	59
4.9	Resultados Alcançados com o Sistema	60

5	UTILIZAÇÃO DO SISTEMA	61
5.1	Acesso ao Sistema	61
5.2	Seleção do Processo Seletivo	62
5.3	Configuração dos Critérios de Avaliação	64
5.3.1	Definição dos Parâmetros de Cálculo	64
5.3.2	Montagem do Barema	65
5.3.3	Cadastro de Seções e Etapas Avaliativas	66
5.4	Triagem Documental	67
5.4.1	Importação das Inscrições	67
5.4.2	Acompanhamento da Fila	67
5.4.3	Conferência dos Documentos	67
5.4.4	Reversão de Indeferimento	69
5.5	Avaliação dos Candidatos	69
5.5.1	Fila de Avaliações	69
5.5.2	Conferência da Demonstrativo do Cálculo	69
5.5.3	Ajuste da Pontuação	71
5.5.4	Lançamento das Notas das Etapas Avaliativas	72
5.6	Classificação e Documentos do Processo	73
5.6.1	Geração da Classificação	73
5.6.2	Emissão do Parecer Preliminar	73
5.6.3	Emissão dos Relatórios	75
5.7	Consulta da Trilha de Auditoria	75
6	CONCLUSÕES	78
6.1	Principais Contribuições	79
6.2	Limitações	80
6.3	Trabalhos Futuros	81
	REFERÊNCIAS	83
	 APÊNDICES	 86
	APÊNDICE A – ENDPOINTS DA API REST	87
	APÊNDICE B – PLANO DE TESTES AUTOMATIZADOS	90
B.1	Testes unitários	90
B.2	Testes de integração (ponta a ponta da API)	91

Introdução

1.1 Contextualização do Trabalho

Processos seletivos para programas de pós-graduação *stricto sensu* envolvem, de maneira geral, duas macroetapas: (i) a submissão da candidatura, com preenchimento de formulário eletrônico e anexação de documentos comprobatórios, e (ii) a avaliação documental e curricular realizada por comissão designada, culminando na classificação e divulgação dos resultados. Em instituições públicas, como a Universidade Federal de Uberlândia (UFU), esses processos devem ainda atender a requisitos de transparência, isonomia, rastreabilidade e conformidade com editais e normas institucionais.

No contexto da Faculdade de Computação (FACOM/UFU), a etapa de submissão foi modernizada com o desenvolvimento do sistema SIPPGCO-FACOM, proposto por Dias (2023), cujo objetivo foi informatizar e organizar o envio de candidaturas ao Programa de Pós-Graduação em Ciência da Computação (PPGCO-UFU). Esse sistema estruturou o fluxo de submissão, centralizou o armazenamento de dados e documentos e reduziu problemas decorrentes do envio manual por e-mail, como perda de arquivos, inconsistências de formato e preenchimento incompleto, introduzindo o conceito da seleção dinâmica dos documentos a serem submetidos no ato da inscrição. Posteriormente, o sistema foi reescrito e evoluído por Junker (2025), incorporando o conceito de formulários dinâmicos de inscrição. O presente trabalho estende essa segunda versão, atualmente em uso como base institucional, operando sobre os dados de inscrição por ela gerados.

Entretanto, apesar da informatização da submissão, a etapa de avaliação permanece predominantemente manual. A comissão avaliadora precisa verificar a presença e a conformidade de documentos obrigatórios, conferir requisitos formais (formato, tipo e tamanho de arquivos), validar pontuações curriculares com base nas regras estabelecidas no edital e consolidar classificações preliminares. Essas atividades, quando realizadas manualmente, demandam elevado esforço operacional, são suscetíveis a erros humanos e podem introduzir divergências de interpretação entre avaliadores.

Nesse contexto, este trabalho apresenta o desenvolvimento de um módulo de avalia-

ção semiautomática integrado ao SIPPGCO-FACOM, com foco na automação de tarefas repetitivas e sistemáticas da etapa de avaliação documental e curricular, deixando para os avaliadores apenas aquelas que dependem de julgamento de mérito. A solução contempla uma etapa inicial de triagem documental, na qual cada documento obrigatório é conferido individualmente e todo indeferimento é justificado; o cálculo automático da pontuação curricular a partir das regras definidas em edital; etapas avaliativas sem documento associado, como a entrevista, cujas notas são lançadas pela comissão e combinadas por soma ponderada; ajustes manuais com reprocessamento imediato das notas; a geração da classificação dos candidatos, que combina a nota de corte e o número de vagas de cada linha de pesquisa; e a produção de relatórios estruturados, promovendo maior transparência, consistência e auditabilidade do processo.

1.2 Problema e Justificativas

O problema central investigado neste trabalho pode ser formulado da seguinte maneira:

Como reduzir esforço humano, inconsistências, subjetividade e falta de rastreabilidade na etapa de avaliação documental e curricular em processos seletivos de pós-graduação, preservando a autonomia e a decisão final da comissão avaliadora?

A etapa de avaliação enfrenta desafios recorrentes, dentre os quais destacam-se:

- ❑ **Erros humanos na conferência:** a verificação manual de múltiplos critérios e documentos aumenta a probabilidade de omissões, cálculos incorretos e retrabalho;
- ❑ **Falta de padronização:** diferentes avaliadores podem interpretar regras do edital de forma distinta, impactando a isonomia do processo;
- ❑ **Tempo excessivo de avaliação:** tarefas operacionais consomem tempo que poderia ser dedicado à análise qualitativa do mérito acadêmico;
- ❑ **Subjetividade e contestação:** ausência de registros estruturados pode dificultar a justificativa de decisões e ampliar o risco de recursos;
- ❑ **Baixa rastreabilidade:** sem registros detalhados de validações e cálculos, torna-se difícil auditar e reproduzir resultados.

A justificativa deste trabalho reside na necessidade institucional de tornar o processo seletivo mais eficiente, padronizado e auditável. Conforme destacam Laudon e Laudon

(2014), sistemas de informação bem projetados são essenciais para apoiar processos decisórios e operacionais em organizações, reduzindo redundâncias e aumentando a confiabilidade das informações. Ao automatizar validações estruturais, cálculo de pontuação e geração de classificações preliminares, o sistema proposto busca reduzir a carga operacional da comissão avaliadora e aumentar a consistência do processo, mantendo a possibilidade de intervenção humana sempre que necessário.

1.3 Objetivos (Gerais e Específicos)

O objetivo geral deste trabalho é projetar, desenvolver e validar um módulo de avaliação semiautomática documental e curricular para processos seletivos de pós-graduação, visando padronizar validações e ampliar a rastreabilidade e a transparência das decisões. A redução do esforço manual da comissão avaliadora é tratada como benefício esperado da automação, e não como resultado a ser mensurado neste trabalho, uma vez que sua quantificação exige o acompanhamento de processos seletivos reais.

Para atingir tal meta, é necessário cumprir os seguintes objetivos específicos:

- ❑ Definir requisitos funcionais e não funcionais para um módulo de avaliação integrável ao SIPP-GCO-FACOM, seguindo práticas consolidadas de engenharia de requisitos (SOMMERVILLE, 2019);
- ❑ Implementar uma etapa estruturada de triagem documental, com conferência individual dos documentos obrigatórios de cada inscrição e registro das justificativas para os indeferimentos;
- ❑ Modelar e implementar o cálculo automático de pontuação curricular com base nas regras do edital, permitindo conferência e ajustes pela comissão avaliadora;
- ❑ Contemplar etapas avaliativas sem documento associado, como a entrevista e a análise de projeto de pesquisa, combinando-as à análise documental por soma ponderada;
- ❑ Segregar as atribuições da coordenação do curso e da comissão avaliadora em perfis de acesso distintos;
- ❑ Disponibilizar mecanismos de apoio à identificação de inconsistências entre informações declaradas e documentação comprobatória;
- ❑ Gerar a classificação dos candidatos combinando a nota de corte e o número de vagas de cada linha de pesquisa;
- ❑ Produzir relatórios estruturados que apoiem auditoria, transparência e rastreabilidade;

- ❑ Projetar e implementar a solução com *NestJS*, *Angular* e *PostgreSQL*, integrando o módulo ao sistema existente por meio de banco de dados compartilhado;
- ❑ Validar o módulo por meio de testes automatizados (unitários e de integração) e testes funcionais de ponta a ponta sobre dados representativos de processos seletivos.

1.4 Contribuições Esperadas

As contribuições esperadas deste trabalho incluem:

- ❑ Um módulo de avaliação semiautomática aplicável a processos seletivos de pós-graduação;
- ❑ Uma arquitetura que separa responsabilidades entre submissão e avaliação, favorecendo manutenção e evolução independentes;
- ❑ Um modelo estruturado de regras de negócio para cálculo de pontuação curricular parametrizado por edital, extensível a etapas avaliativas sem documento associado;
- ❑ Um fluxo de triagem documental com decisões individuais justificadas, derivação automática da situação da inscrição e possibilidade de reversão mediante recurso;
- ❑ Mecanismos de rastreabilidade e geração automática de relatórios.

1.5 Organização do Trabalho

O restante deste documento está estruturado da seguinte forma:

- ❑ O Capítulo 2 descreve os requisitos funcionais e não funcionais do sistema proposto;
- ❑ O Capítulo 3 apresenta a análise e o projeto do sistema, incluindo modelo de domínio, diagramas de sequência, de estados, de classes e modelagem de dados;
- ❑ O Capítulo 4 detalha o desenvolvimento do sistema, abrangendo processo — incluindo o uso de inteligência artificial generativa orientado por especificação —, arquitetura, tecnologias, implementação e testes;
- ❑ O Capítulo 5 funciona como manual do usuário, percorrendo cada funcionalidade do sistema com as respectivas telas e ações;
- ❑ O Capítulo 6 apresenta as conclusões e os trabalhos futuros.

1.6 Uso de Inteligência Artificial Generativa

Em conformidade com o Código de Conduta para pessoas autoras de publicações da Sociedade Brasileira de Computação (SBC), declara-se que foi utilizada a ferramenta de Inteligência Artificial Generativa Claude (Anthropic) como apoio ao desenvolvimento do código do sistema, à revisão textual e gramatical e à estruturação do documento. A forma como a ferramenta foi incorporada ao processo de desenvolvimento — orientada por especificação, com contexto registrado em arquivos de instruções e verificação final independente dos testes automatizados — é detalhada na Seção 4.1.1.

A ferramenta não foi empregada para geração automática de resultados experimentais, dados empíricos ou decisões técnicas não verificadas. Todo o conteúdo apresentado foi revisado e validado pelo autor, que assume integral responsabilidade pelo trabalho.

Requisitos do Sistema

Este capítulo apresenta os requisitos funcionais e não funcionais do módulo de avaliação semiautomática do Sistema de Candidatura em Processos Seletivos da Pós-Graduação. Os requisitos funcionais descrevem os processos e procedimentos que o sistema deve executar, enquanto os requisitos não funcionais definem atributos de qualidade e restrições técnicas.

Cabe registrar uma convenção de nomenclatura adotada ao longo do documento: o termo **edital**, utilizado nas descrições de requisitos, corresponde à entidade *Processo Seletivo* (**SelectiveProcess**) do sistema base de submissão de candidaturas (JUNKER, 2025). O módulo de avaliação não cadastra editais próprios; ele opera sobre os processos seletivos já existentes, acrescentando-lhes a configuração de avaliação, a triagem e o processamento das notas.

A Figura 1 apresenta a visão geral dos casos de uso do módulo e sua relação com os atores envolvidos, dispostos de cima para baixo na mesma sequência em que os requisitos são descritos neste capítulo — que é a ordem lógica de execução dos procedimentos.

2.1 Requisitos Funcionais – RF

Esta seção apresenta os requisitos funcionais do módulo de avaliação semiautomática, organizados na sequência lógica de execução dos procedimentos: primeiro a configuração dos critérios de avaliação do edital, em seguida a triagem documental das inscrições, depois a avaliação dos candidatos e, por fim, a classificação e a divulgação dos resultados. A numeração dos requisitos preserva a ordem em que foram especificados ao longo do projeto, de modo que a identificação de cada um permaneça estável — os requisitos RF009, RF010 e RF011 foram incorporados durante o desenvolvimento.

Duas atribuições distintas percorrem esses requisitos e correspondem aos dois perfis de acesso do sistema:

- a **Coordenação do Curso**, que engloba a secretaria da pós-graduação, configura

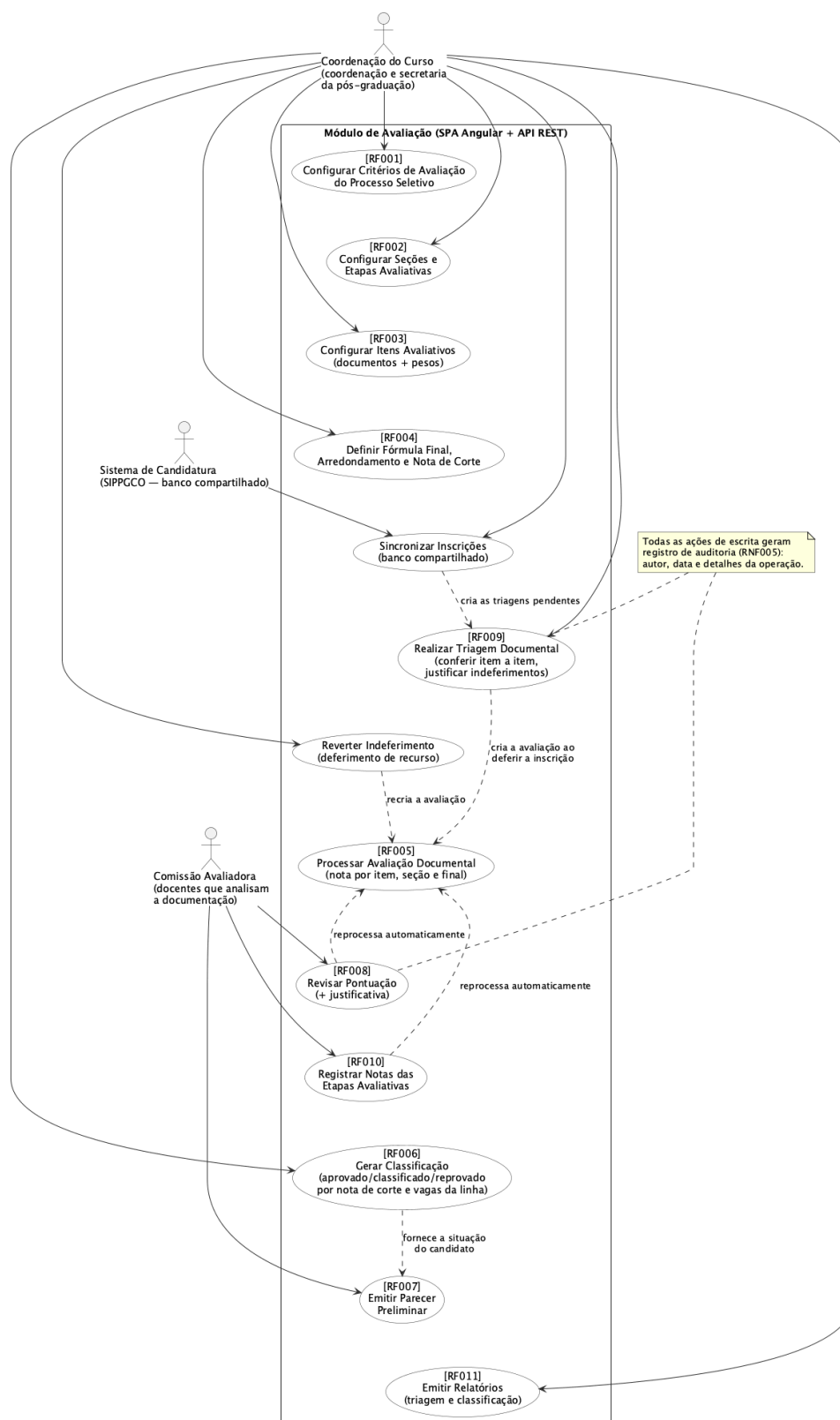


Figura 1 – Casos de uso do módulo de avaliação semiautomática, dispostos na sequência lógica de execução dos procedimentos.

os critérios de avaliação, conduz a triagem documental, gera a classificação e emite os relatórios do processo seletivo;

- ❑ a **Comissão Avaliadora**, composta pelos docentes que analisam a documentação, confere a pontuação apurada, ajusta quantitativos divergentes, registra as notas das etapas avaliativas e emite os pareceres.

A separação é deliberada: a coordenação não altera a pontuação de itens avaliativos, e a comissão não decide deferimentos nem configura o barema.

2.1.1 Configuração dos Critérios de Avaliação

2.1.1.1 [RF001] Configurar Critérios de Avaliação do Edital

- ❑ **Descrição:** Permite à coordenação criar e manter os critérios de avaliação que serão adotados em um processo seletivo existente no sistema base, habilitando-o para o fluxo de avaliação semiautomática.
- ❑ **Ator:** Coordenação do Curso
- ❑ **Prioridade:** Essencial
- ❑ **Entradas e pré-condições:**
 - Usuário autenticado com perfil de coordenação;
 - Processo seletivo previamente cadastrado no sistema base de submissão.
- ❑ **Saídas e pós-condições:**
 - Configuração de avaliação registrada e vinculada ao processo seletivo;
 - Edital disponível para configuração de seções, itens e fórmula.
- ❑ **Fluxo de eventos principal**
 1. A coordenação seleciona um processo seletivo na listagem de editais;
 2. Aciona a criação da configuração de avaliação;
 3. Informa os parâmetros iniciais (tipo de fórmula, regra de arredondamento, casas decimais e, opcionalmente, nota de corte);
 4. O sistema armazena a configuração e registra a ação na trilha de auditoria.

2.1.1.2 [RF002] Configurar Seções e Etapas Avaliativas

- ❑ **Descrição:** Permite definir as seções que estruturam o barema do edital, com peso e ordem de exibição. Uma seção é de natureza **documental**, quando sua nota é apurada automaticamente a partir dos documentos submetidos pelo candidato (por exemplo, Formação Acadêmica e Produção Científica), ou constitui uma **etapa avaliativa** sem documento associado — como a entrevista ou a análise do projeto

de pesquisa —, cuja nota é lançada pela comissão avaliadora. Etapas avaliativas declaram a faixa de notas aceita (valores mínimo e máximo) e as modalidades de curso a que se aplicam, uma vez que nem toda etapa vale para o mestrado e para o doutorado.

❑ **Ator:** Coordenação do Curso

❑ **Prioridade:** Essencial

❑ **Entradas e pré-condições:**

- Configuração de avaliação previamente criada (RF001).

❑ **Saídas e pós-condições:**

- Seções e etapas vinculadas à configuração de avaliação do edital;
- Avaliações existentes reprocessadas automaticamente para refletir a nova estrutura.

2.1.1.3 [RF003] Configurar Itens Avaliativos

❑ **Descrição:** Permite associar a cada seção documental os documentos avaliativos do edital — cadastrados no sistema base de submissão — definindo o peso de cada item na seção. A pontuação por unidade (ou por mês, no caso de itens baseados em período) e o teto de pontuação são herdados do cadastro do documento no sistema base, assim como a modalidade de curso a que o documento se aplica. Etapas avaliativas não recebem itens, por não terem documentação associada.

❑ **Ator:** Coordenação do Curso

❑ **Prioridade:** Essencial

❑ **Entradas e pré-condições:**

- Seção documental previamente cadastrada (RF002);
- Documentos avaliativos disponíveis no processo seletivo.

❑ **Saídas e pós-condições:**

- Item avaliativo registrado e vinculado à seção;
- Avaliações existentes reprocessadas automaticamente.

2.1.1.4 [RF004] Definir Fórmula de Cálculo Final

- ❑ **Descrição:** Permite configurar a fórmula de consolidação da nota final a partir das notas das seções, com três modalidades: soma simples, média ponderada ou expressão personalizada. Como as etapas avaliativas são seções de pleno direito, o peso atribuído a cada uma participa da mesma soma ponderada que combina a análise documental — é por esse mecanismo que a entrevista e a análise de projeto entram na nota final. A configuração inclui ainda a regra de arredondamento (para baixo, para cima ou *half-up*), o número de casas decimais e a nota de corte utilizada na classificação.
- ❑ **Ator:** Coordenação do Curso
- ❑ **Prioridade:** Essencial
- ❑ **Entradas e pré-condições:**
 - Seções e etapas previamente configuradas;
 - Definição de pesos por seção;
 - Regra de arredondamento e, opcionalmente, nota de corte.
- ❑ **Saídas e pós-condições:**
 - Fórmula armazenada e associada ao edital;
 - Todas as avaliações do edital reprocessadas automaticamente com a nova configuração.

2.1.2 Triagem das Inscrições

2.1.2.1 [RF009] Realizar Triagem Documental

- ❑ **Descrição:** Estabelece uma etapa de deferimento documental anterior à avaliação curricular. Cada inscrição recebida do sistema base gera uma triagem com um item de conferência para cada documento obrigatório **da modalidade de curso da inscrição**. A coordenação analisa os anexos submetidos e defere ou indefere cada item individualmente, sendo que os indeferimentos devem ser justificados. A conferência não é interrompida pelo primeiro indeferimento: todos os documentos precisam ser analisados, e a situação da inscrição é **derivada** do conjunto de decisões — enquanto houver item pendente ela permanece pendente; concluída a conferência, fica deferida se todos os itens foram deferidos, ou indeferida caso contrário, reunindo as justificativas de todos os itens indeferidos. Somente inscrições deferidas seguem para a avaliação curricular.
- ❑ **Ator:** Coordenação do Curso

❑ Prioridade: Essencial**❑ Entradas e pré-condições:**

- Inscrições sincronizadas a partir do sistema base;
- Documentos obrigatórios definidos no cadastro do processo seletivo.

❑ Saídas e pós-condições:

- Itens de triagem deferidos ou indeferidos, com autor, data e justificativa registrados;
- Inscrição deferida: avaliação curricular criada automaticamente (RF005);
- Inscrição indeferida: excluída do fluxo de avaliação e da classificação; eventual avaliação previamente gerada é removida;
- Decisão passível de revisão: uma inscrição indeferida pode ser revertida mediante justificativa, no deferimento de recurso, retornando ao fluxo de avaliação com a autoria e a data da alteração registradas em auditoria.

❑ Fluxo de eventos principal

1. A coordenação sincroniza as inscrições do edital, criando as triagens pendentes;
2. Acessa a fila de triagem e seleciona uma inscrição;
3. Confere os anexos de cada documento obrigatório, deferindo ou indeferindo item a item e justificando os indeferimentos;
4. Analisado o último documento, o sistema define a situação da inscrição e, quando todos os itens estão deferidos, cria automaticamente a avaliação curricular;
5. Havendo recurso deferido, a coordenação reverte o indeferimento mediante justificativa, e a avaliação curricular é recriada.

2.1.3 Avaliação dos Candidatos

2.1.3.1 [RF005] Processar Avaliação Documental

- ❑ Descrição:** Executa o cálculo automático das pontuações dos candidatos com base nos critérios definidos. Para cada item, a quantidade alegada nas submissões (unidades ou meses) é convertida em pontos, limitada ao teto do item e ponderada; as notas das seções são agregadas e a nota final é consolidada pela fórmula configurada, com arredondamento e piso zero (a nota final nunca é negativa). O barema é recortado pela modalidade de curso da inscrição: itens documentais só pontuam

quando a modalidade do documento coincide com a do candidato, e etapas avaliativas só participam quando declaradas para aquela modalidade. O processamento é disparado pela sincronização do edital e reexecutado automaticamente sempre que houver alteração nos dados de entrada considerados na avaliação, como a mudança na configuração do barema, um ajuste de quantitativo pela comissão avaliadora ou o lançamento da nota de uma etapa.

❑ **Ator:** Sistema, disparado pela Coordenação do Curso ou pela Comissão Avaliadora

❑ **Prioridade:** Essencial

❑ **Entradas e pré-condições:**

- Edital configurado (RF001–RF004);
- Inscrições com triagem deferida (RF009).

❑ **Saídas e pós-condições:**

- Nota por item calculada e persistida;
- Nota por seção calculada e persistida;
- Nota final calculada, com demonstrativo do cálculo disponível para conferência.

2.1.3.2 [RF008] Revisar Pontuação

❑ **Descrição:** Permite à comissão avaliadora ajustar manualmente a quantidade alegada de um item avaliativo — por exemplo, ao constatar divergência entre o valor declarado pelo candidato no preenchimento da inscrição e a documentação apresentada —, mediante justificativa obrigatória. A avaliação é reprocessada automaticamente no ato do ajuste, sem ação adicional do usuário, e o registro do ajuste (valor anterior, novo valor, autor e justificativa) é preservado na trilha de auditoria mesmo após reprocessamentos subsequentes.

❑ **Ator:** Comissão Avaliadora

❑ **Prioridade:** Importante

❑ **Entradas e pré-condições:**

- Avaliação existente (RF005);
- Nova quantidade alegada e justificativa do ajuste.

❑ **Saídas e pós-condições:**

- Ajuste registrado com valor anterior, novo valor, autor e justificativa;
- Avaliação reprocessada automaticamente, com notas atualizadas.

2.1.3.3 [RF010] Registrar Notas das Etapas Avaliativas

❑ **Descrição:** Permite à comissão avaliadora registrar a nota obtida por um candidato em cada etapa avaliativa sem documento associado, como a entrevista e a análise do projeto de pesquisa. O sistema valida que o valor informado respeita a faixa declarada na etapa e que a etapa se aplica à modalidade de curso da inscrição, e reprocessa a avaliação em seguida, de modo que a nota final já reflita o lançamento. A nota pode ser corrigida posteriormente, com o valor anterior preservado na trilha de auditoria.

❑ **Ator:** Comissão Avaliadora

❑ **Prioridade:** Essencial

❑ **Entradas e pré-condições:**

- Etapa avaliativa cadastrada no barema do edital (RF002);
- Avaliação existente para a inscrição (RF005);
- Nota dentro da faixa declarada na etapa.

❑ **Saídas e pós-condições:**

- Nota da etapa registrada com autor, data e observação opcional;
- Avaliação reprocessada, com a etapa participando da nota final pelo peso da seção;
- Etapas sem nota lançada permanecem sinalizadas na tela da avaliação.

2.1.4 Classificação e Divulgação dos Resultados

2.1.4.1 [RF006] Gerar Classificação

❑ **Descrição:** Ordena automaticamente os candidatos avaliados em ordem decrescente de nota final e atribui a situação de cada um combinando dois critérios configuráveis do edital: a nota de corte e o número de vagas ofertadas. A disputa por vagas ocorre dentro de cada par formado por linha de pesquisa e modalidade de curso, que é o recorte em que as vagas são publicadas. São três as situações possíveis: *aprovado*, quando a nota alcança o corte e a colocação está dentro do número de vagas da linha; *classificado*, quando a nota alcança o corte mas a colocação excede o número de vagas; e *reprovado*, quando a nota é inferior ao corte. A modalidade de cota declarada na inscrição não participa do cálculo — ela serve ao desdobramento e à atribuição manual das vagas pela secretaria da pós-graduação. Cada geração é registrada em auditoria e pode ser exportada em formato CSV.

❑ **Ator:** Coordenação do Curso

❑ **Prioridade:** Essencial

❑ **Entradas e pré-condições:**

- Edital configurado, com avaliações calculadas (RF005);
- Vagas cadastradas por linha de pesquisa e modalidade no sistema base;
- Nota de corte definida na configuração de avaliação, quando aplicável.

❑ **Saídas e pós-condições:**

- Lista classificatória agrupada por linha de pesquisa e modalidade, ordenada por nota final;
- Indicação de situação (aprovado, classificado ou reprovado), com justificativa quando a situação não for de aprovação;
- Arquivo CSV exportável com linha de pesquisa, modalidade, colocação, candidato, nota e situação.

2.1.4.2 [RF007] Emitir Parecer Preliminar

❑ **Descrição:** Gera relatório detalhado por candidato contendo notas por item, por seção e nota final, incluindo a demonstrativo do cálculo (quantidade alegada, pontos por unidade, teto e peso) e os ajustes manuais realizados pela comissão avaliadora, com as respectivas justificativas. O parecer informa também a situação do candidato na classificação, com a colocação obtida dentro da sua linha de pesquisa. É marcado como desatualizado quando a avaliação ou a configuração do edital é alterada após sua emissão, e pode ser exportado em PDF.

❑ **Ator:** Comissão Avaliadora

❑ **Prioridade:** Importante

❑ **Entradas e pré-condições:**

- Avaliação processada, com nota final calculada (RF005);
- Classificação gerada para o edital (RF006), da qual o parecer extrai a situação do candidato.

❑ **Saídas e pós-condições:**

- Parecer persistido e disponível para consulta e exportação;
- Parecer sinalizado como desatualizado quando dados posteriores o invalidarem.

2.1.4.3 [RF011] Emitir Relatórios do Processo Seletivo

- ❑ **Descrição:** Produz as relações oficiais do processo seletivo em dois documentos. O primeiro apresenta as inscrições após a triagem documental, categorizadas em deferidas e indeferidas, com a opção de exibir ou omitir as justificativas — com justificativa, o documento destina-se ao arquivo interno da coordenação; sem justificativa, à divulgação pública dos resultados. O segundo é a relação final de classificação, que apresenta, para cada linha de pesquisa cadastrada, a lista de candidatos ordenada da maior para a menor pontuação e categorizada em aprovado, classificado ou reprovado. Ambos podem ser exportados em PDF e em CSV.
- ❑ **Ator:** Coordenação do Curso
- ❑ **Prioridade:** Importante
- ❑ **Entradas e pré-condições:**
 - Triagem concluída, para a relação de inscrições (RF009);
 - Classificação gerada, para a relação final (RF006).
- ❑ **Saídas e pós-condições:**
 - Relações disponíveis para consulta em tela e exportação em PDF e CSV;
 - Emissão registrada na trilha de auditoria.

2.2 Requisitos Não Funcionais – RNF

Esta seção apresenta os requisitos não funcionais do módulo de avaliação semiautomática, organizados por categoria. A classificação dos atributos de qualidade baseia-se na norma ISO/IEC 25010 (ISO/IEC, 2011), complementada por categorias adicionais (distribuição, padrões, *hardware* e *software*) previstas no modelo de documentação adotado pela FACOM. A numeração dos requisitos é sequencial e global, garantindo rastreabilidade única ao longo do documento.

2.2.1 Usabilidade

2.2.1.1 [RNF001] Interface Intuitiva

- ❑ **Descrição:** A interface deve apresentar organização clara das informações e das ações disponíveis em cada tela, permitindo que a coordenação configure editais e conduza a triagem, e que a comissão avaliadora confira e ajuste pontuações, sem necessidade de conhecimento técnico em programação. O requisito alcança toda funcionalidade que envolva interação com o usuário, e não apenas a configuração dos critérios de avaliação.

- ❑ **Prioridade:** Essencial

- ❑ **Requisito(s) associado(s):** RF001–RF004 e RF006–RF011

2.2.1.2 [RNF002] *Feedback* de Processamento

- ❑ **Descrição:** O sistema deve fornecer mensagens claras de sucesso ou erro em toda operação iniciada pelo usuário — configuração do barema, triagem, processamento, ajuste de pontuação, lançamento de notas de etapa, classificação e emissão de documentos —, informando o resultado da ação e, quando houver falha, o motivo.

- ❑ **Prioridade:** Importante

- ❑ **Requisito(s) associado(s):** RF001–RF011

2.2.2 Confiabilidade

2.2.2.1 [RNF003] Consistência de Cálculo

- ❑ **Descrição:** O motor de regras deve garantir que os cálculos de pontuação sejam determinísticos e reproduzíveis, assegurando que os mesmos dados gerem sempre o mesmo resultado. A exigência alcança também as entradas informadas manualmente — o ajuste de quantitativos e as notas das etapas avaliativas —, que alimentam o mesmo motor.

- ❑ **Prioridade:** Essencial

- ❑ **Requisito(s) associado(s):** RF005, RF006, RF008, RF010

2.2.2.2 [RNF004] Controle Transacional

- ❑ **Descrição:** As operações críticas de processamento devem utilizar controle transacional no banco de dados PostgreSQL, garantindo integridade dos dados em caso de falhas.

- ❑ **Prioridade:** Essencial

- ❑ **Requisito(s) associado(s):** RF005

2.2.2.3 [RNF005] Registro de Auditoria

- ❑ **Descrição:** O sistema deve manter registro detalhado de toda ação que altere o estado da avaliação — configuração dos critérios, seções, etapas e itens; decisões de triagem e sua eventual reversão; processamentos; ajustes de quantitativo; lançamento de notas de etapa; geração de classificações, pareceres e relatórios —, incluindo usuário responsável, data, entidade afetada e justificativa, quando aplicável.

- ❑ **Prioridade:** Essencial
- ❑ **Requisito(s) associado(s):** RF001–RF011

2.2.3 Desempenho

2.2.3.1 [RNF006] Tempo de Processamento

- ❑ **Descrição:** O sistema deve ser capaz de processar automaticamente até 500 candidaturas em tempo inferior a 60 segundos.
- ❑ **Prioridade:** Importante
- ❑ **Requisito(s) associado(s):** RF005

2.2.3.2 [RNF007] Tempo de Resposta da API

- ❑ **Descrição:** As requisições da API REST devem apresentar tempo médio de resposta inferior a 500 ms para operações não relacionadas ao processamento em lote.
- ❑ **Prioridade:** Importante

2.2.4 Segurança

2.2.4.1 [RNF008] Autenticação e Autorização

- ❑ **Descrição:** O sistema deve implementar autenticação baseada em *token* JWT — aceitando os *tokens* emitidos pelo sistema base de submissão (JUNKER, 2025) — e controle de acesso segregado em dois perfis operacionais: *admin*, correspondente à coordenação do curso, que engloba a secretaria da pós-graduação, e *committee*, correspondente à comissão avaliadora, composta pelos docentes que analisam a documentação. A coordenação não altera a pontuação de itens avaliativos, e a comissão não decide deferimentos nem configura o barema. Usuários com perfil de candidato não têm acesso ao módulo de avaliação.
- ❑ **Prioridade:** Essencial
- ❑ **Requisito(s) associado(s):** Todos

2.2.4.2 [RNF009] Proteção de Dados Pessoais

- ❑ **Descrição:** O sistema deve estar em conformidade com a Lei Geral de Proteção de Dados (BRASIL, 2018), garantindo proteção das informações pessoais dos candidatos.
- ❑ **Prioridade:** Essencial

2.2.4.3 [RNF010] Comunicação Segura

❑ **Descrição:** Toda comunicação entre sistemas deve ocorrer por meio de protocolo HTTPS.

❑ **Prioridade:** Essencial

2.2.5 Distribuição

2.2.5.1 [RNF011] Implantação em Servidor Institucional

❑ **Descrição:** O sistema deve ser implantado em servidor institucional da UFU, respeitando as políticas de infraestrutura e segurança da universidade.

❑ **Prioridade:** Essencial

2.2.6 Padrões

2.2.6.1 [RNF012] Arquitetura REST

❑ **Descrição:** A API deve seguir os princípios de arquitetura REST (FIELDING, 2000), utilizando padrões HTTP adequados.

❑ **Prioridade:** Essencial

2.2.6.2 [RNF013] Padrões de Código

❑ **Descrição:** O código-fonte deve seguir boas práticas de desenvolvimento em TypeScript, incluindo tipagem forte, separação de camadas e modularização.

❑ **Prioridade:** Importante

2.2.7 Hardware e Software

2.2.7.1 [RNF014] Ambiente de Execução

❑ **Descrição:** O sistema deverá ser executado em servidor Linux institucional com Node.js na versão 22 ou superior e PostgreSQL na versão 13 ou superior. Recomenda-se distribuição de longo prazo, como Ubuntu Server 22.04 LTS ou Debian 12. Como configuração mínima de *hardware* para o volume típico dos processos seletivos do PPGCO — da ordem de algumas centenas de inscrições por edital —, indicam-se dois núcleos de processamento, 4 GB de memória principal e 20 GB de armazenamento livre, dos quais a maior parte é consumida pelos documentos comprobatórios enviados pelos candidatos. Recomenda-se, para operação institucional, quatro núcleos e 8 GB de memória, de modo a acomodar o processamento em lote

das avaliações sem competir com o sistema de submissão, que compartilha o mesmo banco de dados.

❑ **Prioridade:** Essencial

2.2.7.2 [RNF015] Compatibilidade de Banco

❑ **Descrição:** O banco de dados utilizado deverá ser PostgreSQL versão 13 ou superior, compartilhado com o sistema base de submissão de candidaturas.

❑ **Prioridade:** Importante

Análise e Projeto do Sistema

Este capítulo apresenta a análise e o projeto arquitetural do módulo de avaliação semi-automática do Sistema de Candidatura em Processos Seletivos da Pós-Graduação, fundamentado nos requisitos definidos no Capítulo 2. A modelagem apresentada estabelece a transição entre a especificação de requisitos e a implementação do sistema, assegurando coerência entre domínio do problema, arquitetura e estrutura de dados.

A solução é composta por uma aplicação de página única (SPA) em *Angular* e uma API REST em *NestJS*, organizada internamente em camadas de *Controller* e *Service*, com persistência em PostgreSQL por meio do ORM *Prisma*. O banco de dados é compartilhado com o sistema base de submissão de candidaturas (JUNKER, 2025): o módulo lê as inscrições, os documentos do edital e as submissões diretamente do mesmo esquema relacional, e acrescenta tabelas próprias para a triagem, o barema e as notas. Essa abordagem em camadas, amplamente discutida por Richards e Ford (2020), favorece modularidade, separação de responsabilidades e manutenção evolutiva.

3.1 Justificativa Arquitetural

A arquitetura adotada foi definida com os seguintes objetivos:

- ❑ Separação clara entre regras de negócio e infraestrutura;
- ❑ Redução de acoplamento entre componentes;
- ❑ Facilidade de testes unitários na camada de serviço;
- ❑ Possibilidade de evolução do motor de fórmulas sem impacto estrutural;
- ❑ Integração de baixo atrito com o sistema base, por meio de banco de dados compartilhado e validação dos mesmos *tokens* de autenticação, evitando duplicação de cadastros e sincronizações via rede.

A opção pelo compartilhamento do banco de dados — em vez de integração por chamadas REST entre os dois sistemas — foi uma decisão de projeto tomada em conjunto com a orientação do trabalho. Ela elimina a necessidade de replicar dados de inscrições, garante consistência imediata entre submissão e avaliação e simplifica a implantação institucional, ao custo de acoplar os módulos no nível do esquema de dados. Para mitigar esse acoplamento, o módulo de avaliação trata as tabelas do sistema base como somente leitura e concentra suas escritas em tabelas próprias.

Essa estrutura contribui diretamente para atributos de qualidade definidos na norma ISO/IEC 25010 (ISO/IEC, 2011), especialmente manutenibilidade, confiabilidade e modularidade.

3.2 Modelo de Domínio do Sistema

O modelo de domínio representa a visão conceitual do problema, independente de detalhes tecnológicos. Ele identifica as entidades essenciais envolvidas no processo de avaliação documental e seus relacionamentos estruturais, distinguindo as entidades **reaproveitadas do sistema base** (JUNKER, 2025) daquelas **introduzidas pelo módulo de avaliação**.

Do sistema base, o módulo reaproveita:

- ❑ **Processo Seletivo** — o edital propriamente dito, com vigência e situação;
- ❑ **Inscrição** — a candidatura submetida por um usuário a um processo seletivo;
- ❑ **Documentos do Processo** — cada documento exigido ou pontuável do edital, com a modalidade de curso a que se aplica, regra de contagem (unidades ou período), pontos por unidade e teto de pontuação;
- ❑ **Linha de Pesquisa e Vaga** — as linhas do programa e o número de vagas ofertadas em cada uma, por modalidade de curso, que delimitam a disputa na classificação;
- ❑ **Usuário** — candidatos, coordenação do curso e comissão avaliadora, com seus papéis.

O módulo de avaliação introduz:

- ❑ **Configuração de Avaliação** — parametriza a avaliação de um processo seletivo (fórmula, arredondamento e nota de corte);
- ❑ **Seção e Item da Seção** — estruturam o barema. A seção é documental, e então agrupa itens que referenciam documentos do processo com seus pesos, ou constitui uma etapa avaliativa sem documentação associada — como a entrevista ou a análise

de projeto de pesquisa —, caso em que declara apenas a faixa de notas aceita e as modalidades de curso a que se aplica;

- ❑ **Nota de Etapa** — a nota que a comissão avaliadora atribui a um candidato em cada etapa avaliativa sem documento associado;
- ❑ **Triagem e Item de Triagem** — registram o deferimento documental de cada inscrição, item a item;
- ❑ **Avaliação, Nota por Seção e Nota por Item** — armazenam o resultado do processamento com demonstrativo do cálculo;
- ❑ **Ajuste Manual** — registra intervenções da comissão avaliadora sobre os quantitativos pleiteados pelos candidatos durante o preenchimento das inscrições;
- ❑ **Parecer Preliminar** — relatório estruturado por candidato;
- ❑ **Classificação e Entrada de Classificação** — lista classificatória em que a situação de cada candidato resulta da nota de corte e do número de vagas da sua linha de pesquisa;
- ❑ **Registro de Auditoria** — trilha transversal de todas as ações relevantes.

A Figura 2 apresenta o modelo conceitual do domínio, evidenciando a fronteira entre o sistema base e o módulo proposto.

3.3 Diagramas de Sequência do Sistema

Os diagramas de sequência validam o comportamento do sistema e demonstram a interação entre suas camadas internas e o banco de dados compartilhado.

3.3.1 RF005 — Sincronizar e Processar Avaliações

A Figura 3 representa o fluxo de sincronização e processamento das candidaturas. A coordenação aciona a sincronização do edital; o sistema então identifica as inscrições com triagem deferida (RF009), recorta o barema pela modalidade de curso de cada inscrição, calcula a nota por item (quantidade alegada convertida em pontos, limitada ao teto e ponderada), incorpora as notas já lançadas para as etapas avaliativas (RF010), agrega as notas por seção, aplica a fórmula final configurável com arredondamento e piso zero, e persiste os resultados com registro de auditoria. Inscrições ainda não avaliadas geram novas avaliações; as demais são reprocessadas.

Cabe distinguir o cálculo do reprocessamento, ambos executados por este requisito. O **cálculo** é o mesmo procedimento em qualquer caso: percorrer o barema aplicável à

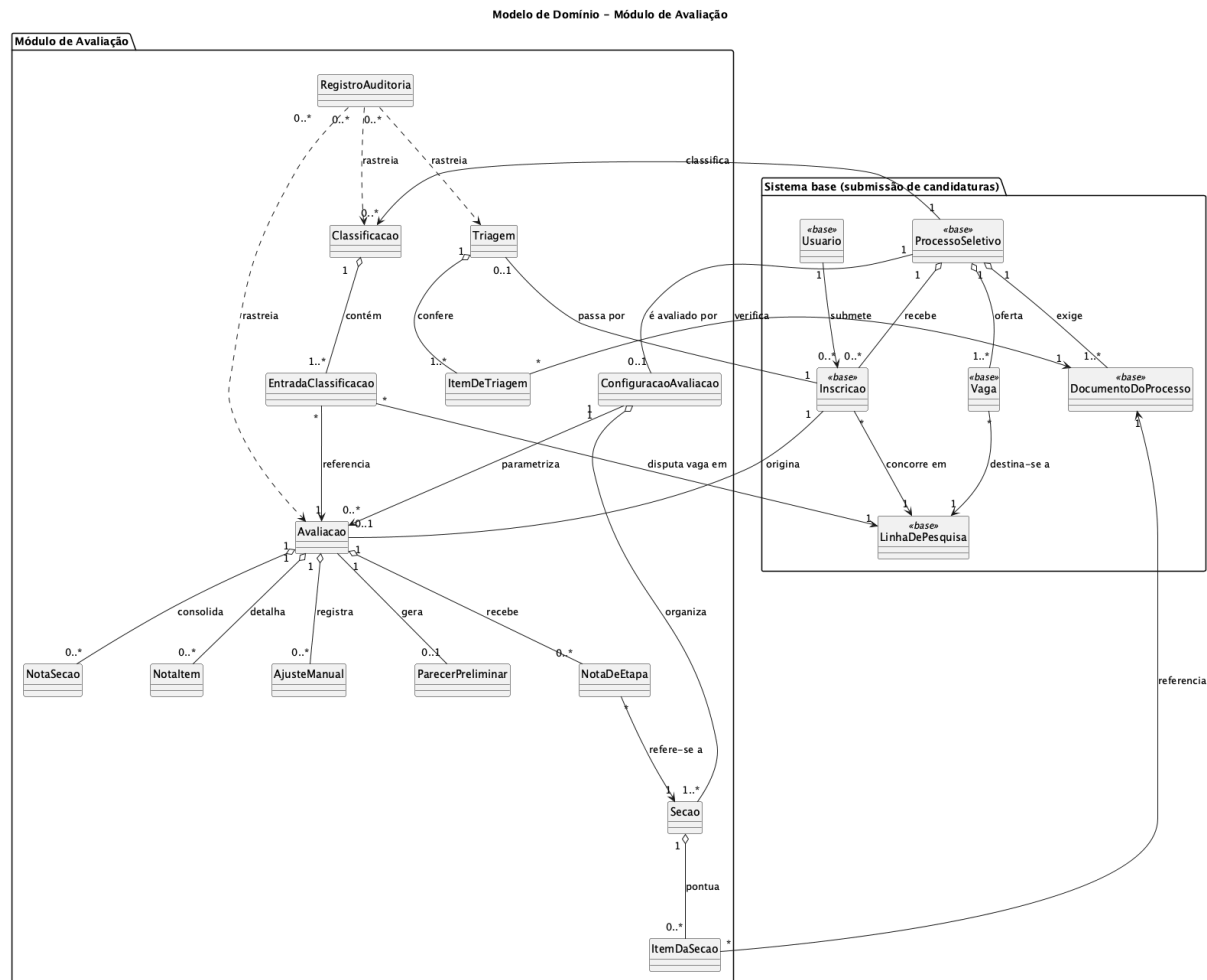


Figura 2 – Modelo conceitual do domínio do módulo de avaliação semiautomática.

inscrição e consolidar a nota final. O que varia é o gatilho. Na primeira execução, a avaliação é criada; nas seguintes, ela é recalculada sobre os mesmos dados de entrada, agora modificados — porque a configuração do barema mudou (RF001–RF004), porque a comissão ajustou um quantitativo (RF008) ou porque lançou a nota de uma etapa (RF010). Não há, portanto, dois procedimentos distintos, e sim um único procedimento reexecutado automaticamente sempre que suas entradas mudam. Essa escolha substitui a antiga etapa manual de reprocessamento e garante que nota, classificação e parecer nunca estejam defasados em relação à configuração vigente.

3.3.2 RF008 — Ajuste Manual com Reprocessamento Automático

A Figura 4 demonstra o fluxo de ajuste manual de pontuação. A comissão avaliadora informa a nova quantidade alegada de um item e a justificativa; o sistema registra o ajuste e a auditoria e, na mesma operação, reprocessa a avaliação — não há etapa manual de reprocessamento. A nota exibida na interface já reflete o ajuste imediatamente, garantindo

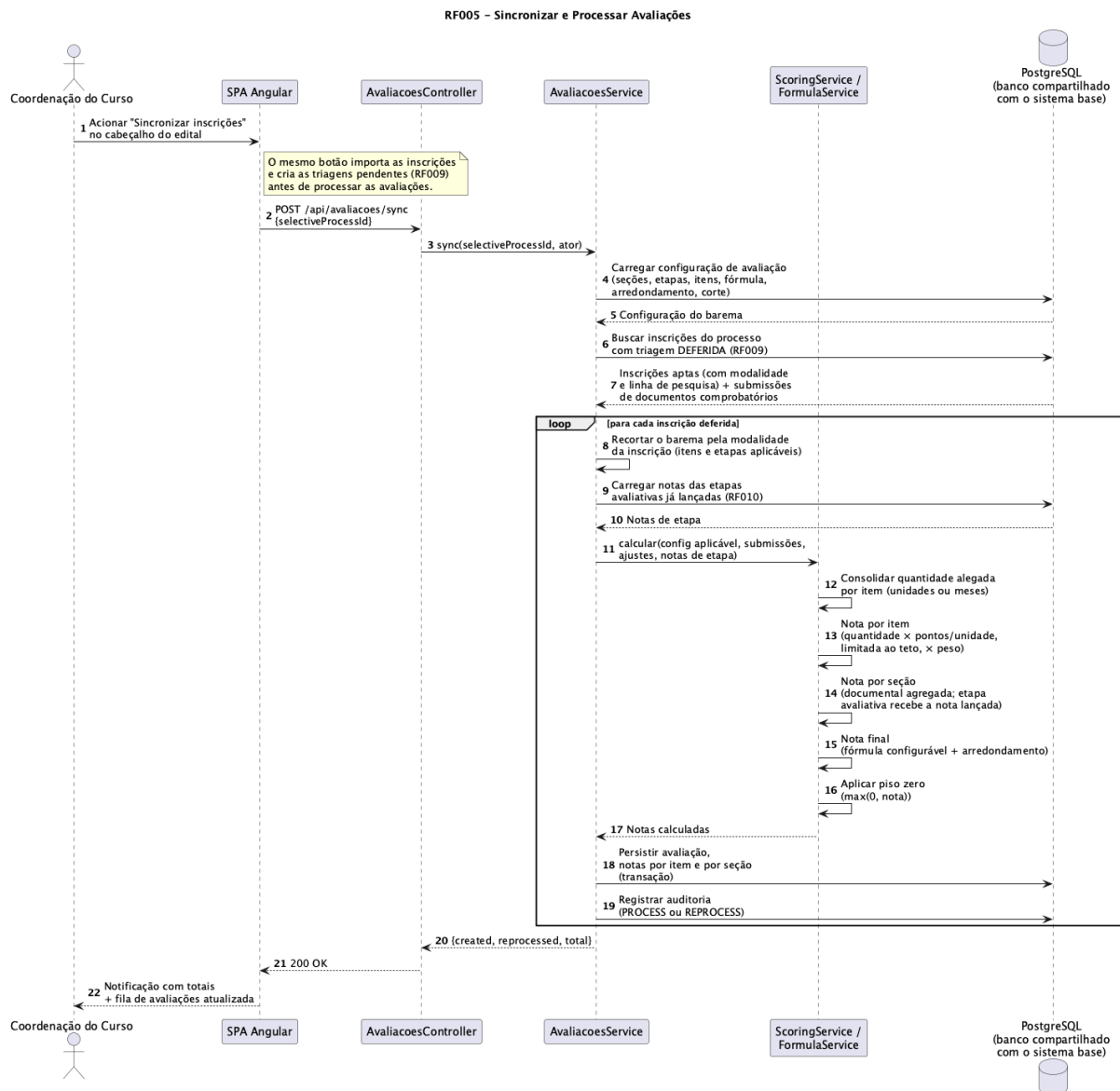


Figura 3 – Sincronização e processamento automático das avaliações.

consistência entre nota final, classificação e parecer preliminar.

3.4 Diagrama de Estado — Ciclo de Vida da Triagem

No projeto inicial do sistema, a entidade Avaliação possuía uma máquina de estados própria (pendente, processada, revisada, reprocessada e homologada), que condicionava as operações permitidas em cada momento. Durante o desenvolvimento, essa máquina foi removida, uma vez que qualquer alteração de configuração, ajuste de quantitativo ou lançamento de nota de etapa reprocessa as notas automaticamente (RF005, RF008, RF010). Dessa forma, a avaliação sempre reflete o estado corrente do cálculo, sem estados intermediários ou necessidade de homologação. A simplificação elimina transições inválidas,

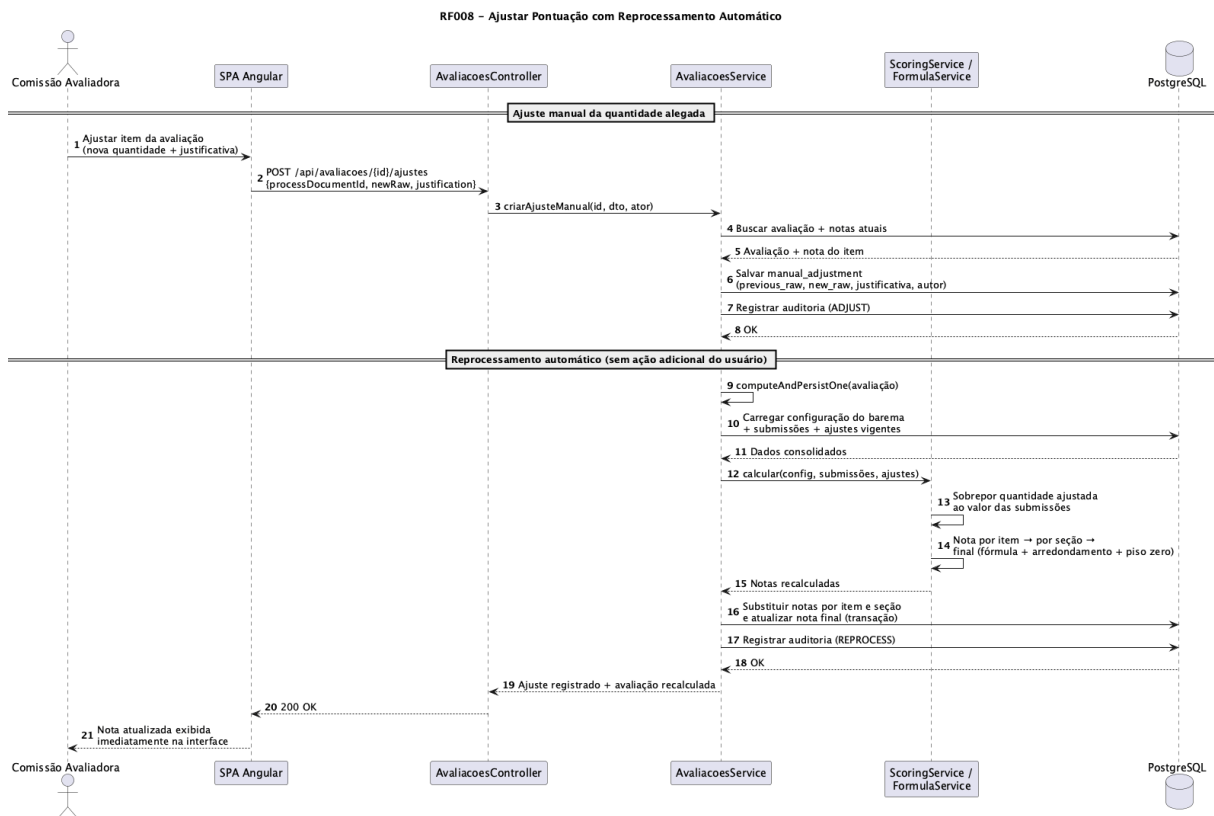


Figura 4 – Ajuste manual com reprocessamento automático da avaliação.

reduz a complexidade da interface e evita resultados divergentes do cenário atual.

O controle de estados do fluxo concentra-se, portanto, na **Triagem** (RF009), cujo ciclo de vida é apresentado na Figura 5. Toda inscrição sincronizada inicia como *pendente* e permanece nesse estado enquanto houver documento sem decisão. Analisado o último item, a situação é derivada do conjunto: *deferida*, se todos os documentos foram deferidos, o que dispara a criação da avaliação curricular; ou *indeferida*, caso contrário, reunindo as justificativas de todos os itens indeferidos.

O indeferimento não é terminal. Toda decisão de triagem é passível de recurso e, sendo ele acolhido, a coordenação reverte o indeferimento mediante justificativa: os documentos indeferidos passam a deferidos, a inscrição retorna ao fluxo de avaliação e a avaliação curricular é recriada. As justificativas originais, a justificativa do recurso, o autor e a data permanecem na trilha de auditoria, de modo que a alteração seja rastreável. Pela mesma razão, a decisão de um item individual também pode ser revista a qualquer momento, com a situação da inscrição recalculada em seguida.

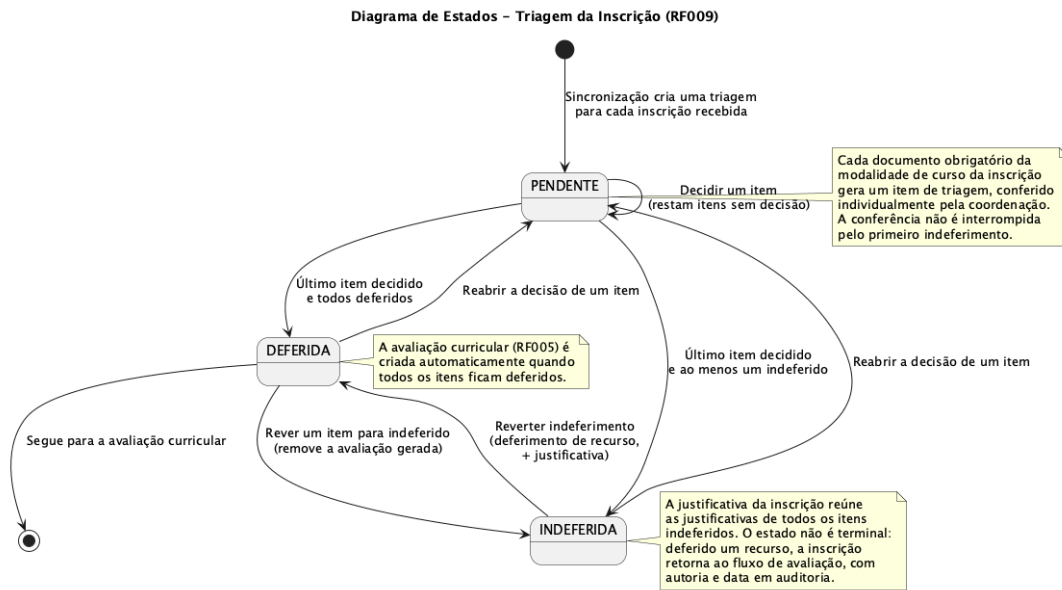


Figura 5 – Ciclo de vida da triagem de uma inscrição.

3.5 Modelagem de Dados

3.5.1 Modelo Lógico de Dados

O modelo lógico foi derivado do modelo de domínio e estruturado segundo o paradigma relacional, sendo normalizado até a Terceira Forma Normal (3FN) (ELMASRI; NAVATHE, 2018), garantindo integridade referencial e eliminação de redundâncias. O esquema convive com as tabelas do sistema base (JUNKER, 2025) no mesmo banco PostgreSQL, sendo que, por convenção, as tabelas do módulo de avaliação utilizam nomenclatura *snake_case*, distinguindo-se visualmente das tabelas herdadas.

As principais estruturas criadas para o novo módulo são:

- ❑ **evaluation_configs** — configuração de avaliação do processo seletivo (fórmula, arredondamento, nota de corte);
- ❑ **sections** e **section_items** — seções do barema e seus itens (referências a documentos do edital, com pesos). A coluna **kind** distingue a seção documental da etapa avaliativa, e as colunas **min_score** e **max_score** delimitam a faixa de notas desta última;
- ❑ **section_modalities** — modalidades de curso a que uma etapa avaliativa se aplica;
- ❑ **stage_scores** — notas lançadas pela comissão avaliadora nas etapas sem documento associado, com autor, data e observação;
- ❑ **trriages** e **trriage_items** — triagem documental de cada inscrição e seus itens de conferência;

- ❑ `evaluations`, `section_scores` e `item_scores` — avaliações com notas por seção e por item;
- ❑ `manual_adjustments` — ajustes manuais com valor anterior, novo valor, autor e justificativa;
- ❑ `rankings` e `ranking_entries` — listas classificatórias e suas entradas. Cada entrada guarda, além da posição geral e da nota, o par (linha de pesquisa, modalidade) em que o candidato disputou vaga, a colocação dentro desse par e o número de vagas ofertadas no momento da geração, de modo que a classificação permaneça um retrato fiel mesmo que a inscrição seja alterada depois;
- ❑ `preliminary_opinions` — pareceres preliminares estruturados;
- ❑ `audit_logs` — trilha de auditoria de todas as ações relevantes.

Dentre as tabelas do sistema base, o módulo de avaliação consome principalmente `selectiveprocesses` (processos seletivos), `selectiveprocesssubscriptions` (inscrições, com a modalidade de curso e a linha de pesquisa disputada), `processdocument` (documentos do edital, com a modalidade a que se aplicam, regra de contagem, pontos por unidade e teto), `researchlines` e `vacancy` (linhas de pesquisa e vagas ofertadas em cada uma) e `users` (usuários e papéis).

3.5.2 Modelo Físico de Dados

A implementação física foi realizada em PostgreSQL com mapeamento via Prisma ORM. Em razão do compartilhamento do banco com o sistema base, que não utiliza ferramenta de migração, adotou-se a estratégia *database-first*, onde as alterações de esquema são aplicadas diretamente no banco e o esquema Prisma é regenerado por introspecção (`prisma db pull`), sem uso de migrações automáticas. O trecho a seguir ilustra o modelo gerado para a entidade de avaliação:

```
model evaluations {
  id                Int           @id @default(autoincrement())
  subscription_id    Int           @unique
  evaluation_config_id Int
  final_score        Decimal?     @db.Decimal(10, 4)
  created_at         DateTime     @default(now())
  updated_at         DateTime     @default(now())

  evaluation_configs evaluation_configs
  selectiveprocesssubscriptions selectiveprocesssubscriptions
}
```

```

item_scores          item_scores[]
section_scores       section_scores[]
manual_adjustments   manual_adjustments[]
preliminary_opinions preliminary_opinions?
}

```

Destaca-se a ausência de coluna de *status*: conforme discutido na seção anterior, a avaliação não possui máquina de estados — a unicidade de `subscription_id` garante uma avaliação por inscrição, sempre refletindo o cálculo mais recente.

3.6 Diagrama de Classes do Sistema

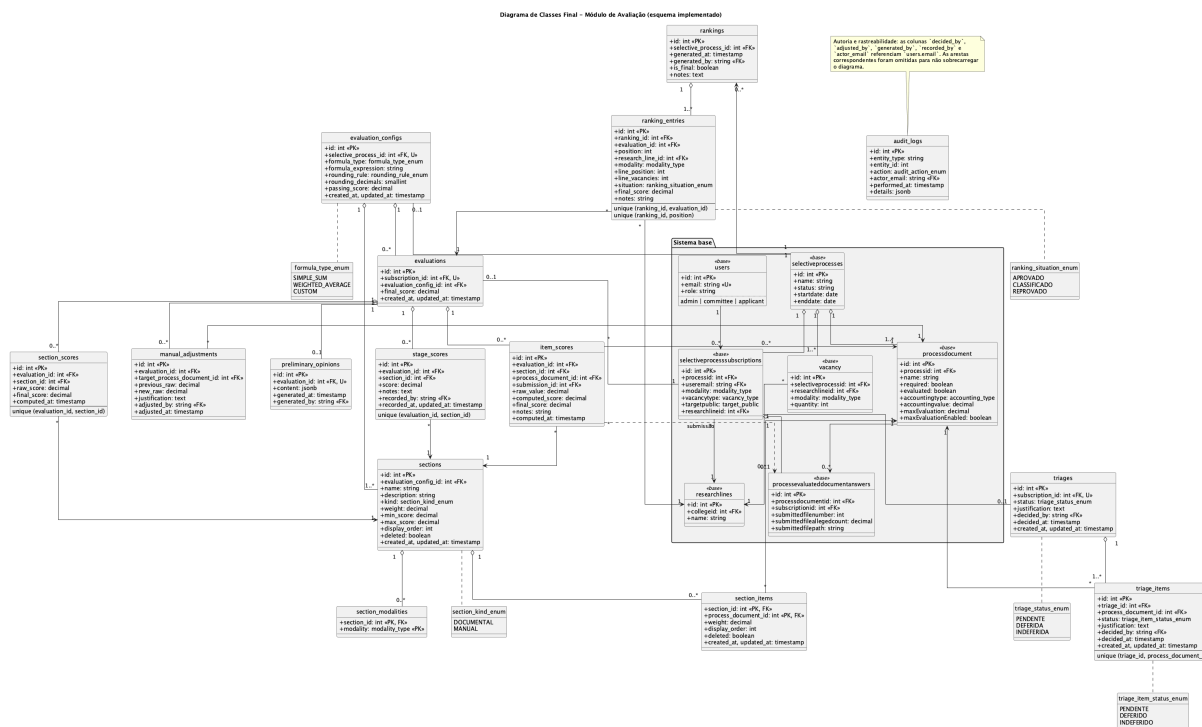
O diagrama de classes representa o refinamento técnico do modelo de domínio, incorporando atributos, tipos e relacionamentos do esquema efetivamente implementado, incluindo a enumeração de natureza da seção (`DOCUMENTAL`, `MANUAL`), as de situação da triagem (`PENDENTE`, `DEFERIDA`, `INDEFERIDA`) e a de situação na classificação (`APROVADO`, `CLASSIFICADO`, `REPROVADO`).

A Figura 6 apresenta o diagrama de classes do sistema, com as entidades do sistema base destacadas em pacote próprio. Para preservar a legibilidade das cardinalidades, as arestas de autoria — as colunas `decided_by`, `adjusted_by`, `generated_by`, `recorded_by` e `actor_email`, todas referenciando `users.email` — foram substituídas por uma nota no próprio diagrama: desenhadas uma a uma, elas atravessavam a fronteira do sistema base e sobrecarregavam a leitura.

3.7 Rastreabilidade Requisitos–Projeto

A modelagem apresentada assegura alinhamento entre requisitos e elementos arquiteturais:

- ❑ RF001–RF004 (configuração dos critérios de avaliação) → entidades `evaluation_configs`, `sections`, `section_items` e `section_modalities`;
- ❑ RF005 (processamento) → diagrama de sequência de sincronização e processamento e entidades `evaluations`, `section_scores` e `item_scores`;
- ❑ RF006 (classificação) → entidades `rankings` e `ranking_entries`, que registram a situação apurada por nota de corte e vagas da linha de pesquisa;
- ❑ RF007 (parecer) → entidade `preliminary_opinions`;
- ❑ RF008 (ajuste de pontuação) → diagrama de sequência de ajuste manual com reprocessamento automático e entidade `manual_adjustments`;



Desenvolvimento do Sistema

Este capítulo descreve o desenvolvimento do módulo de avaliação semiautomática, abrangendo o processo de desenvolvimento adotado — inclusive o uso de inteligência artificial generativa orientado por especificação —, a arquitetura implementada, os padrões de projeto e as tecnologias utilizadas, os procedimentos de instalação, configuração e implantação, o repositório de código resultante, a estratégia de testes e os resultados alcançados. A operação do sistema pelo usuário final é tratada separadamente, no Capítulo 5.

4.1 Processo de Desenvolvimento

O desenvolvimento seguiu um processo **iterativo e incremental**, organizado em ciclos curtos de implementação seguidos de validação com o orientador do trabalho, que atuou como representante do cliente. Cada ciclo entregava um incremento funcional verificável, e o retorno obtido nas validações realimentava os requisitos — prática alinhada às recomendações de Sommerville (2019) para sistemas cujo entendimento do domínio evolui ao longo do projeto.

As principais etapas do processo foram:

1. **Análise do sistema base:** leitura estruturada do código e do esquema de dados do sistema de submissão (JUNKER, 2025), identificando as entidades reaproveitáveis (processos seletivos, inscrições, documentos e usuários) e os pontos de extensão;
2. **Modelagem e projeto:** elaboração do modelo de domínio, dos diagramas comportamentais e do esquema relacional do módulo, apresentados no Capítulo 3;
3. **Implementação do *backend*:** construção da API REST com os módulos de configuração de edital, seções, itens, triagens, avaliações, classificação, pareceres, relatórios, auditoria e autenticação;
4. **Implementação do *frontend*:** construção da SPA com as telas de configuração do barema, fila de triagem, fila de avaliações, classificação, relatórios e auditoria;

5. **Refinamento orientado pelo domínio:** ciclos de validação que resultaram em mudanças de requisito relevantes — a introdução da etapa de triagem documental (RF009), a remoção da máquina de estados da avaliação em favor do reprocessamento automático, a incorporação de etapas avaliativas sem documento associado (RF010), a segregação entre os perfis de coordenação e de comissão avaliadora e a classificação por nota de corte combinada ao número de vagas de cada linha de pesquisa;
6. **Testes e verificação final:** consolidação das suítes de testes automatizados e execução de um plano de testes funcionais de ponta a ponta.

O controle de versão foi realizado com Git, em repositórios independentes para *backend* e *frontend*, com mensagens de *commit* descritivas que documentam a evolução das decisões de projeto.

4.1.1 Desenvolvimento Orientado por Especificação com Apoio de Inteligência Artificial

O código do sistema foi desenvolvido com o apoio do Claude Code (Anthropic, 2026), ferramenta de programação assistida por inteligência artificial generativa da Anthropic que opera de forma agêntica: em uma sessão de trabalho, ela lê o repositório, edita arquivos e executa comandos — compilação, testes e consultas ao banco — sob supervisão do desenvolvedor. Essa autonomia acelera a produção de código, mas traz riscos conhecidos: a partir de instruções vagas, a ferramenta tende a produzir código plausível que se afasta dos requisitos, e as decisões tomadas em uma sessão não são lembradas na seguinte.

Para controlar esses riscos, adotou-se o desenvolvimento orientado por especificação (*spec-driven development*), abordagem em que se define o que construir e por quê antes de decidir como: os requisitos são convertidos em especificação, plano técnico e tarefas, e a implementação é conduzida e conferida contra esses artefatos (GitHub, 2025). A especificação deixa, assim, de ser documentação produzida depois do código e passa a ser a entrada da ferramenta.

4.1.1.1 Ciclo de desenvolvimento

O ciclo adotado, ilustrado na Figura 7, compreende quatro etapas, cada uma materializada em um artefato desta monografia:

1. **Especificar:** os requisitos funcionais RF001 a RF011 e os quinze requisitos não funcionais (Capítulo 2), cada um com descrição, ator, prioridade, entradas, pré-condições, saídas e pós-condições. A especificação descreve o comportamento esperado em termos verificáveis e funciona como o contrato contra o qual o código gerado é julgado;

2. **Planejar:** o modelo de domínio, os diagramas de sequência, de estados e de classes e o esquema relacional (Capítulo 3). O plano fixa as decisões estruturais antes da codificação — quais tabelas do sistema base são somente leitura, quais pertencem ao módulo e como a autenticação é compartilhada —, para que a ferramenta não as improvise a cada tarefa;
3. **Decompor em tarefas:** a implementação foi dividida em um módulo coeso por funcionalidade — configuração de editais, seções, triagens, avaliações, classificação, pareceres, relatórios e auditoria —, cada qual vinculado aos requisitos que atende e delimitado de modo a caber em uma sessão de trabalho e a ser revisado isoladamente;
4. **Implementar:** a ferramenta gera o código da tarefa e os testes correspondentes, executa a compilação, a análise estática e as suítes automatizadas e corrige as falhas encontradas. Os testes funcionam como porta de saída: a tarefa só é dada por concluída quando todos passam e depois da revisão do autor, que registra o incremento em *commit* de sua autoria. Ao final do projeto, as suítes somavam 54 testes unitários e 25 de integração (Seção 4.8).

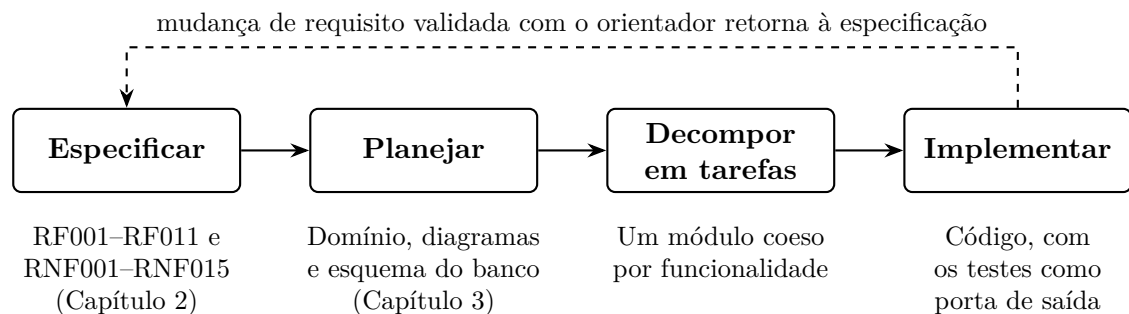


Figura 7 – Ciclo de desenvolvimento orientado por especificação adotado no projeto, com o artefato da monografia que corresponde a cada etapa.

O ciclo não é linear. As validações com o orientador produziram as mudanças de requisito descritas no início desta seção — a triagem documental, o fim da máquina de estados da avaliação, as etapas sem documento associado, a segregação de perfis e a classificação por vagas. Cada uma percorreu o ciclo desde o começo: foi registrada primeiro na especificação e no plano, e só então implementada. Essa ordem impediu que o código passasse a divergir do documento e manteve a especificação como referência única do projeto.

4.1.1.2 Contexto persistente e limites da ferramenta

Como a ferramenta não retém memória entre sessões, o contexto do projeto foi registrado em arquivos de instruções, denominados `CLAUDE.md`, lidos automaticamente no início de cada sessão (Anthropic, 2026). Cada repositório mantém o seu: o da raiz do

projeto descreve o posicionamento do trabalho, o sistema base e as decisões tomadas com o orientador; os do *backend*, do *frontend* e da monografia registram a pilha tecnológica, as convenções e as decisões arquiteturais duradouras, sempre acompanhadas de sua justificativa. São exemplos a gestão do esquema sem migrações automáticas, que disputariam o banco com o sistema base; a situação da triagem derivada das decisões sobre os documentos, nunca informada diretamente; e a segregação entre coordenação e comissão avaliadora. Com a justificativa registrada, uma sessão posterior não desfaz a decisão por desconhecê-la.

Foram definidos ainda dois subagentes — instâncias da ferramenta com instruções e permissões próprias — e um conjunto de comandos reutilizáveis:

- ❑ **Analista do sistema legado:** subagente sem permissão de escrita, encarregado de examinar o código e o esquema do sistema de submissão (JUNKER, 2025). Apoiou a primeira etapa do processo de desenvolvimento, e a análise revelou, por exemplo, que o sistema base utiliza Express com acesso direto ao banco, sem ORM, e que não possuía os papéis de coordenação e de comissão avaliadora;
- ❑ **Revisor de texto:** subagente dedicado à revisão da monografia quanto às normas da ABNT, à classe institucional da FACOM e à consistência de referências cruzadas e citações;
- ❑ **Comandos reutilizáveis:** rotinas para compilar a monografia e relatar os avisos relevantes, verificar referências quebradas, resumir o estado do trabalho e reanalisar o sistema base em busca de mudanças.

Por fim, as permissões da ferramenta foram restringidas por configuração: a leitura e a consulta ao histórico de versões são livres; as ações de escrita dependem de aprovação; e as operações destrutivas ou externas são bloqueadas — exclusão recursiva de arquivos, descarte de alterações no Git, publicação nos repositórios remotos e qualquer alteração no código dos trabalhos anteriores, mantido apenas como referência.

4.1.1.3 Verificação e responsabilidade

O desenvolvimento orientado por especificação torna a verificação objetiva, mas não a dispensa. Testes escritos pela mesma ferramenta que escreveu o código podem compartilhar o mesmo mal-entendido sobre o requisito; por isso, a conferência final não se apoiou apenas nas suítes automatizadas. Ao término do desenvolvimento, o sistema foi percorrido pela interface, a partir da tela de autenticação e com o banco recriado do zero, contra o plano de testes funcionais (Tabela 2) e contra cada afirmação do manual do usuário (Capítulo 5). Essa validação encontrou treze defeitos que haviam passado pelas suítes automatizadas, todos corrigidos, e seis divergências entre o texto e o sistema —

entre elas a regra de arredondamento, descrita como *half-even* quando o sistema aplica *half-up*.

O episódio ilustra a divisão de papéis adotada: a ferramenta acelera a produção de código, testes e texto, mas as decisões de domínio — as regras do barema, as situações da classificação, a segregação de perfis — vieram das validações com o orientador, e a aceitação de cada incremento coube ao autor. A ferramenta foi empregada também na revisão textual desta monografia, sob o mesmo princípio: nenhum trecho foi incorporado sem leitura e conferência com o sistema. Conforme declarado na Seção 1.6, o autor assume integral responsabilidade pelo trabalho.

4.2 Arquitetura do Sistema

O sistema adota arquitetura cliente-servidor em três camadas, como ilustrado na Figura 8 e descrito a seguir:

- ❑ **Camada de apresentação:** SPA desenvolvida em Angular (Google, 2026), executada no navegador do usuário, que consome a API por requisições HTTP autenticadas;
- ❑ **Camada de aplicação:** API REST desenvolvida em NestJS (NestJS, 2026a) sobre Node.js (OpenJS Foundation, 2026b), organizada internamente em *controllers* (entrada HTTP, validação e autorização) e *services* (regras de negócio e orquestração transacional);
- ❑ **Camada de dados:** banco PostgreSQL (PostgreSQL Global Development Group, 2026) compartilhado com o sistema base de submissão, acessado via Prisma ORM (Prisma Data, Inc., 2026).

O compartilhamento do banco de dados é o elemento central da estratégia de integração: o módulo lê inscrições, documentos e submissões diretamente das tabelas do sistema base (tratadas como somente leitura) e escreve exclusivamente em tabelas próprias. A autenticação é igualmente compartilhada — o módulo valida, com a chave pública correspondente, os *tokens* JWT assinados pelo sistema base, e obtém o papel do usuário diretamente do banco, dispensando novo cadastro de credenciais.

Em ambiente de desenvolvimento, a infraestrutura é provida por contêiner Docker (Docker, Inc., 2026) executando o PostgreSQL, com a API e a SPA executadas localmente (portas 3000 e 4200, respectivamente, com *proxy* reverso de desenvolvimento para o caminho `/api`). Para produção, prevê-se a implantação em servidor institucional da UFU (RNF011), servindo a SPA como conteúdo estático e a API atrás de *proxy* reverso com HTTPS (RNF010).

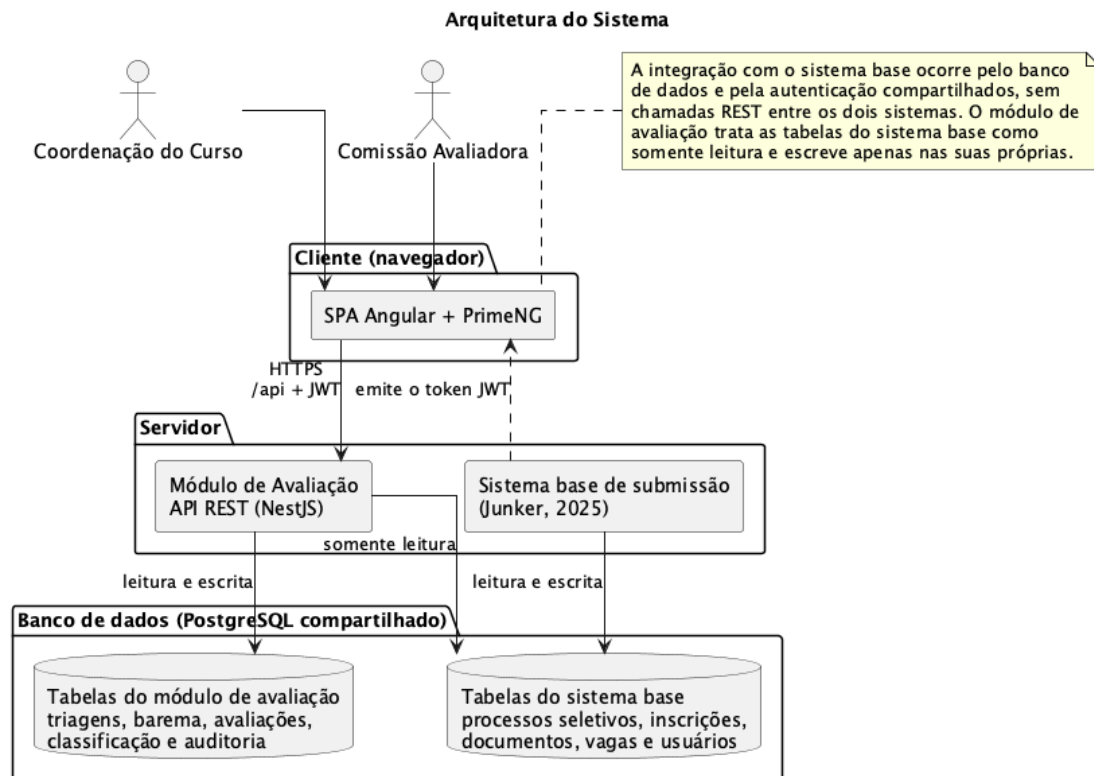


Figura 8 – Arquitetura do sistema: camadas do módulo e integração com o sistema base por banco de dados e autenticação compartilhados.

4.3 Padrões de Projeto Utilizados

Os principais padrões de projeto e princípios arquiteturais empregados foram:

- ❑ **Arquitetura em camadas** (RICHARDS; FORD, 2020): separação entre apresentação, aplicação e dados, e, internamente à API, entre *controllers* e *services*;
- ❑ **Injeção de dependência:** mecanismo nativo do NestJS, utilizado para compor serviços, facilitar testes unitários com substituição de dependências e evitar acoplamento direto entre módulos;
- ❑ **DTO (*Data Transfer Object*) com validação declarativa:** todos os dados de entrada da API são descritos por esquemas Zod (MCDONNELL, 2026) convertidos em DTOs, garantindo validação automática e tipagem estática inferida;
- ❑ ***Guard* e *Interceptor* (responsabilidades transversais):** autenticação (validação do JWT), autorização por papéis e limitação de taxa de requisições são implementadas como *guards* globais, mantendo os *controllers* livres de lógica de segurança repetida;

- ❑ **Estratégias no motor de fórmulas:** as três modalidades de consolidação da nota final (soma simples, média ponderada e expressão personalizada) são estratégias intercambiáveis selecionadas pela configuração do edital;
- ❑ **Auditoria centralizada:** um serviço único de auditoria, disponível globalmente, registra todas as ações relevantes, evitando duplicação do código de rastreabilidade;
- ❑ **Monólito modular:** a API é um único processo organizado em módulos coesos por funcionalidade (editais, seções, itens, triagens, avaliações, classificação, pareceres, relatórios e auditoria), o que simplifica a implantação sem sacrificar a organização interna.

4.4 Tecnologias Utilizadas no Desenvolvimento

4.4.1 Linguagens de desenvolvimento

- ❑ **TypeScript** (Microsoft, 2026): linguagem principal de *backend* e *frontend*, em modo estrito, atendendo ao RNF013 (tipagem forte e modularização);
- ❑ **SQL:** definição do esquema relacional do módulo e consultas de introspecção, sobre PostgreSQL (PostgreSQL Global Development Group, 2026).

4.4.2 Frameworks

- ❑ **NestJS 11** (NestJS, 2026a): *framework* de *backend* sobre Node.js (OpenJS Foundation, 2026b), com sistema de módulos, injeção de dependência e suporte nativo a *guards*, *pipes* e *interceptores*;
- ❑ **Angular 19** (Google, 2026): *framework* de *frontend*, utilizado com componentes *standalone*, sinais (*signals*) para estado reativo e carregamento preguiçoso de rotas.

4.4.3 Bibliotecas

- ❑ **Prisma ORM 6** (Prisma Data, Inc., 2026): mapeamento objeto-relacional com cliente tipado gerado por introspecção do banco (*database-first*);
- ❑ **Zod** (MCDONNELL, 2026): validação de esquemas com inferência de tipos, integrada ao NestJS via `nestjs-zod`;
- ❑ **math.js** (JONG, 2026): avaliação segura das fórmulas personalizadas de nota final, com *sandbox* contra funções perigosas;
- ❑ **PrimeNG 19** (PrimeTek, 2026): biblioteca de componentes de interface para Angular (tabelas, diálogos, notificações);

- ❑ **nestjs-pino** (nestjs-pino, 2026): registro estruturado de logs HTTP com identificador de requisição e redação de cabeçalhos sensíveis;
- ❑ **csv-stringify** (CSV project, 2026): geração dos arquivos CSV de exportação da classificação e dos relatórios, com proteção contra injeção de fórmulas em planilhas;
- ❑ **@nestjs/swagger** (NestJS, 2026b): documentação interativa da API, disponível em ambiente de desenvolvimento;
- ❑ **@nestjs/throttler** (NestJS, 2026c): limitação de taxa de requisições por endereço IP.

4.4.4 Ferramentas Auxiliares

- ❑ **Docker e Docker Compose** (Docker, Inc., 2026): provisionamento do PostgreSQL em desenvolvimento, com aplicação automática dos *scripts* de esquema;
- ❑ **Jest** (OpenJS Foundation, 2026a): execução das suítes de testes unitários e de integração, com *mocks* tipados do Prisma (**jest-mock-extended**);
- ❑ **ESLint e Prettier**: padronização e verificação estática do código;
- ❑ **PlantUML** (PlantUML, 2026): geração dos diagramas UML apresentados neste documento a partir de descrições textuais versionadas;
- ❑ **Git e GitHub**: controle de versão e hospedagem dos repositórios.

4.4.5 Sistemas e Componentes Externos Utilizados

O único sistema externo utilizado é o sistema base de submissão de candidaturas (JUNKER, 2025), com o qual o módulo se integra em dois pontos:

- ❑ **Banco de dados compartilhado**: as tabelas de processos seletivos, inscrições, documentos do edital, submissões de comprovantes e usuários são lidas diretamente, sem replicação de dados;
- ❑ **Autenticação compartilhada**: os *tokens* JWT emitidos pelo sistema base no *login* são aceitos pelo módulo, que os valida com a chave pública correspondente e resolve o papel do usuário no banco.

Durante o desenvolvimento identificaram-se, ainda, duas evoluções pontuais necessárias no sistema base, ambas realizadas em bifurcação (*fork*) própria dos seus repositórios, preservando o código original. A primeira, no cadastro de vagas: por decisão da coordenação, a quantidade de vagas passou a ser única por linha de pesquisa, deixando de ser subdividida por cotas — é essa quantidade que delimita a disputa na classificação

(RF006). A modalidade de cota permanece no sistema, declarada pelo candidato no ato da inscrição; o que a bifurcação retirou foi a sua subdivisão no cadastro de vagas, coerentemente com o critério de classificação adotado. A segunda, no perfil dos usuários: como as contas nascem no sistema de candidatura, foi ali que se introduziu o perfil de comissão avaliadora, acompanhado de uma tela em que a coordenação atribui esse perfil a contas administrativas já existentes no sistema.

4.5 Instalação e Configurações

4.5.1 Pré-requisitos do Sistema

Quanto aos componentes de *software*, são necessários:

- ❑ Node.js versão 22 ou superior e gerenciador de pacotes **npm**;
- ❑ Docker e Docker Compose (para o banco de dados em desenvolvimento);
- ❑ PostgreSQL 16 (provido pelo contêiner em desenvolvimento; instância institucional em produção);
- ❑ Navegador moderno compatível com Angular 19.

Quanto ao *hardware* e ao sistema operacional do servidor, recomenda-se distribuição Linux de longo prazo — como Ubuntu Server 22.04 LTS ou Debian 12 —, ambiente em que o sistema foi desenvolvido e verificado. A Tabela 1 apresenta a configuração mínima e a recomendada, dimensionadas para o volume típico dos processos seletivos do PPGCO, da ordem de algumas centenas de inscrições por edital (RNF014). O espaço em disco é dominado pelos documentos comprobatórios enviados pelos candidatos, armazenados pelo sistema de submissão; a configuração recomendada acomoda o processamento em lote das avaliações sem competir com esse sistema, que compartilha o mesmo banco de dados.

Tabela 1 – Requisitos de *hardware* do servidor.

Recurso	Mínimo	Recomendado
Processador	2 núcleos	4 núcleos
Memória principal	4 GB	8 GB
Armazenamento livre	20 GB	100 GB
Sistema operacional	Linux (Ubuntu 22.04 LTS ou Debian 12)	

Na estação de trabalho do usuário não há exigência particular além de um navegador atualizado, uma vez que toda a computação ocorre no servidor.

4.5.2 Processo de Instalação/Execução

A execução em ambiente de desenvolvimento segue os passos:

1. Clonar os repositórios do *backend* e do *frontend*;
2. No *backend*, instalar as dependências (`npm install`) e iniciar o banco com `docker compose up -d postgres` — na primeira execução, os *scripts* SQL do esquema (sistema base e módulo de avaliação) são aplicados automaticamente;
3. Copiar o arquivo `.env.example` para `.env` e preencher as variáveis de ambiente;
4. Sincronizar o cliente do ORM com o banco (`npm run db:sync`) e, opcionalmente, popular dados de demonstração (`npm run db:seed`);
5. Iniciar a API em modo de desenvolvimento (`npm run start:dev`), disponível em `http://localhost:3000`;
6. No *frontend*, instalar as dependências e iniciar o servidor de desenvolvimento (`npm start`), disponível em `http://localhost:4200`, com *proxy* configurado para a API.

4.5.3 Variáveis de Ambiente

As principais variáveis de ambiente do *backend* são:

- ❑ `DATABASE_URL` — cadeia de conexão com o banco PostgreSQL compartilhado;
- ❑ `JWT_PUBLIC_KEY` — chave pública utilizada para validar os *tokens* emitidos pelo sistema base;
- ❑ `JWT_PRIVATE_KEY` — chave privada utilizada apenas em desenvolvimento, para emissão local de *tokens* de teste;
- ❑ `PORT` — porta de escuta da API (padrão 3000);
- ❑ `CORS_ORIGIN` — origem autorizada para requisições do *frontend*;
- ❑ `NODE_ENV` — ambiente de execução (`development` ou `production`).

As variáveis são validadas na inicialização por um esquema Zod: valores ausentes ou malformados impedem a subida da aplicação com mensagem explicativa, prevenindo configurações silenciosamente incorretas.

4.5.4 Arquivos de Configuração

- ❑ `docker-compose.yml` — definição do contêiner PostgreSQL e da aplicação dos *scripts* de esquema;
- ❑ `db/01_inscricao_schema.sql` e `db/02_avaliacao_schema.sql` — recriação do esquema de inscrição do sistema base para desenvolvimento e esquema próprio do módulo, respectivamente;
- ❑ `prisma/schema.prisma` — esquema do ORM, gerado por introspecção do banco (não editado manualmente);
- ❑ `proxy.conf.json` — redirecionamento das chamadas `/api` do servidor de desenvolvimento do Angular para a API;
- ❑ `src/environments/` — configurações de ambiente do *frontend* (desenvolvimento e produção).

4.6 Compilação e Implantação

4.6.1 Processo de Compilação do Sistema

O *backend* é compilado com `npm run build`, que transpila o código TypeScript para JavaScript no diretório `dist/`. O *frontend* é compilado com `npm run build`, que gera o *bundle* otimizado de produção. Ambas as compilações são verificadas pelas suítes de teste e pela análise estática antes da integração de mudanças.

4.6.2 Opções de Implantação do Sistema

A implantação prevista em servidor institucional (RNF011) compreende: execução da API como serviço Node.js (diretamente ou em contêiner), publicação do *bundle* do *frontend* como conteúdo estático em servidor web e *proxy* reverso com terminação HTTPS encaminhando o caminho `/api` para a API. O banco de dados utilizado é a instância PostgreSQL institucional já empregada pelo sistema base.

4.6.3 Configurações para Produção

Em produção, a documentação interativa da API é desabilitada; o bloco de autenticação alternativa de desenvolvimento da tela de *login* é removido da compilação; os registros de log são emitidos em formato estruturado, sem dados sensíveis; e a emissão local de *tokens* é desativada, de modo que a autenticação ocorra exclusivamente pelo sistema base.

4.7 Repositório de Código

4.7.1 Link para o Repositório

O código-fonte está disponível publicamente nos repositórios:

❑ *Backend* (API REST): <<https://github.com/Ihury/sippgco-avaliacao-back>>

❑ *Frontend* (SPA): <<https://github.com/Ihury/sippgco-avaliacao-front>>

A evolução pontual do cadastro de vagas do sistema base foi realizada em bifurcações próprias:

❑ <<https://github.com/Ihury/programa-pos-graduacao-ufu-back>>

❑ <<https://github.com/Ihury/programa-pos-graduacao-ufu-front>>

4.7.2 Estrutura de Diretórios

O *backend* está organizado por módulos de domínio, como segue:

```
sippgco-avaliacao-back/
|-- db/           scripts SQL do esquema e seed de demonstracao
|-- prisma/       schema do ORM (gerado por introspeccao)
|-- src/
|   |-- auth/     autenticacao JWT e autorizacao por papeis
|   |-- editais/  configuracao de avaliacao (RF001, RF004)
|   |-- secoes/   secoes do barema (RF002)
|   |-- secao-itens/ itens avaliativos (RF003)
|   |-- triagens/ triagem documental (RF009)
|   |-- avaliacoes/ processamento e ajustes (RF005, RF008)
|   |   '-- scoring/ motor de pontuacao e formulas
|   |-- rankings/ classificacao e exportacao CSV (RF006)
|   |-- pareceres/ parecer preliminar (RF007)
|   |-- relatorios/ relatorios de triagem e classificacao (RF011)
|   |-- auditoria/ trilha de auditoria (RNF005)
|   '-- common/   guards, filtros e utilitarios
'-- test/         testes de integracao ponta a ponta
```

O *frontend* está estruturado por funcionalidades, como segue:

```
sippgco-avaliacao-front/
'-- src/app/
```

```

|-- core/          modelos de API, servicos HTTP, guards
|                  e interceptores
|-- layout/        estrutura visual (menu e cabecalho)
'-- features/
    |-- auth/       tela de login
    '-- editais/    abas do edital: visao geral, triagem,
                    avaliacoes, classificacao, formula,
                    relatorios e auditoria

```

4.7.3 Guia de Contribuições

Os repositórios incluem documentação de desenvolvimento com as convenções adotadas (organização de módulos, fluxo *database-first* de evolução do esquema e comandos de execução e teste), suficiente para que novos colaboradores reproduzam o ambiente e contribuam com o projeto.

4.8 Testes do Sistema

4.8.1 Plano de Testes

A estratégia de verificação combinou três níveis complementares:

- ❑ **Testes unitários** — exercitam isoladamente as regras de negócio críticas, com dependências simuladas: arredondamento (regras *floor*, *ceil* e *half-up*), motor de pontuação (conversão de quantidades em pontos, teto e pesos), avaliação de fórmulas personalizadas (incluindo proteção do *sandbox*), classificação dos candidatos por nota de corte e vagas da linha de pesquisa, guardas de autenticação e autorização — inclusive a segregação entre coordenação e comissão avaliadora nos dois sentidos — e regras de edição da configuração;
- ❑ **Testes de integração (ponta a ponta da API)** — exercitam o fluxo completo contra banco PostgreSQL real: criação da configuração do barema, sincronização e conferência da triagem sensível à modalidade, derivação da situação da inscrição, acúmulo de justificativas, reversão de recurso, criação automática de avaliações, ajuste manual com reprocessamento, lançamento de nota de etapa, geração da classificação, do parecer e dos relatórios e consulta da trilha de auditoria;
- ❑ **Testes funcionais** — plano de cenários executados na interface, cobrindo o caminho completo do usuário.

Os cenários do plano de testes funcionais, executados na interface, são apresentados na Tabela 2.

Tabela 2 – Plano de testes funcionais executado na interface.

Número	Descrição	Resultados esperados
TM01	Autenticação com credenciais válidas	Sessão iniciada e redirecionamento para a listagem de editais
TM02	Listagem de editais	Editais exibidos com indicadores de inscritos e de configuração
TM03	Configuração dos critérios de avaliação (fórmula, arredondamento e nota de corte)	Configuração persistida e associada ao edital
TM04	Cadastro de seção documental e associação de itens avaliativos	Seção e itens registrados; avaliações existentes reprocessadas
TM05	Cadastro de etapa avaliativa com faixa de notas e modalidade	Etapa registrada, sem aceitar documentos associados
TM06	Sincronização das inscrições do edital	Uma triagem pendente por inscrição, com um item por documento obrigatório da modalidade da inscrição
TM07	Listagem de triagens	Contadores por situação (pendente, deferida, indeferida) corretos
TM08	Indeferimento de um documento com justificativa	Conferência prossegue; inscrição permanece pendente enquanto restar item sem decisão
TM09	Conclusão da conferência com dois documentos indeferidos	Inscrição indeferida reunindo as justificativas de ambos
TM10	Conclusão da conferência com todos os documentos deferidos	Inscrição deferida e avaliação curricular criada automaticamente
TM11	Reversão de indeferimento mediante justificativa	Inscrição devolvida ao fluxo, avaliação recriada e alteração registrada em auditoria
TM12	Listagem de avaliações	Inscrições deferidas presentes; inscrição indeferida ausente
TM13	Conferência da demonstrativo do cálculo	Itens pontuados apenas na modalidade da inscrição, com quantidade alegada, teto, peso e nota resultante
TM14	Ajuste manual da quantidade alegada	Nota recalculada imediatamente, sem ação adicional, e ajuste registrado em auditoria
TM15	Lançamento da nota de uma etapa avaliativa	Nota aceita dentro da faixa, recusada fora dela, com reprocessamento imediato da nota final
TM16	Geração da classificação	Candidatos agrupados por linha de pesquisa e modalidade, com situação aprovado, classificado ou reprovado e justificativa quando aplicável

continua na próxima página

Tabela 2 – continuação

Número	Descrição	Resultados esperados
TM17	Emissão do parecer preliminar	Parecer com demonstrativo do cálculo, ajustes e situação do candidato na classificação
TM18	Emissão dos relatórios de triagem e de classificação	Relações exibidas em tela e exportadas em PDF e CSV, com e sem justificativas
TM19	Segregação de perfis	Coordenação impedida de ajustar pontuação; comissão impedida de conduzir a triagem e de gerar a classificação
TM20	Consulta da trilha de auditoria	Registro de todas as ações executadas, com autor, data e detalhes

4.8.2 Execução do Plano de Testes

Ao final do desenvolvimento, a suíte automatizada avaliou 54 testes unitários e 25 testes de integração, todos com execução bem-sucedida. O plano de testes funcionais foi executado integralmente na interface, com aprovação em todos os cenários. A execução validou, em particular, o encadeamento triagem → avaliação → ajuste → classificação → auditoria, o comportamento do reprocessamento automático diante de mudanças de configuração e o recorte do barema pela modalidade de curso da inscrição.

Cabe distinguir o alcance dessa verificação. Os testes funcionais confirmam que cada funcionalidade se comporta como especificado, mas não avaliam a usabilidade da interface: verificar que uma tela produz o resultado esperado não equivale a verificar que usuários da coordenação e da comissão avaliadora conseguem operá-la com facilidade, eficiência e satisfação. O requisito de interface intuitiva (RNF001) orientou o projeto das telas — organização uniforme das ações, componentes visuais consistentes e mensagens explicativas —, mas não foi validado, pois isso exigiria uma avaliação de usabilidade com usuários representativos, não conduzida neste trabalho.

O requisito de desempenho RNF006 — processamento de até 500 candidaturas em menos de 60 segundos — foi verificado sobre um processo seletivo sintético com exatamente 500 inscrições e 4.000 submissões de documentos avaliativos, distribuídas em duas seções documentais e quatro itens. A criação e o processamento das 500 avaliações consumiram 1,4 segundo; o reprocessamento integral das mesmas 500, entre 1,6 e 1,7 segundo. A margem em relação ao limite de 60 segundos é, portanto, superior a uma ordem de grandeza. Na mesma carga, a geração da classificação consumiu 0,1 segundo e as consultas de relatório permaneceram abaixo de 0,1 segundo, dentro do previsto no RNF007.

A medição também confirmou o RNF003: três execuções consecutivas do processamento sobre os mesmos dados produziram as 500 notas finais idênticas, sem divergência. Cabe registrar que os tempos foram obtidos em máquina de desenvolvimento, com o

banco em contêiner local; recomenda-se repetir a medição no servidor institucional antes da implantação, uma vez que ali o banco é compartilhado com o sistema de submissão.

4.9 Resultados Alcançados com o Sistema

O desenvolvimento resultou em um módulo funcional que cobre integralmente os requisitos funcionais especificados (RF001 a RF011), verificado por testes automatizados e funcionais. Destacam-se como resultados:

- ❑ **Fluxo completo de avaliação informatizado:** da chegada da inscrição (via banco compartilhado) à lista classificatória, passando por triagem justificada, cálculo automático de notas com demonstrativo do cálculo e parecer estruturado;
- ❑ **Rastreabilidade integral:** toda decisão — deferimento, indeferimento, reversão de recurso, ajuste de quantitativo, lançamento de nota de etapa, configuração, processamento e classificação — é registrada com autor, data e justificativa, atendendo ao RNF005 e respondendo diretamente ao problema de baixa auditabilidade identificado no Capítulo 1;
- ❑ **Consistência automática:** a eliminação da máquina de estados em favor do processamento imediato garante que notas, classificações e pareceres nunca estejam defasados em relação à configuração vigente;
- ❑ **Segregação de atribuições:** a coordenação conduz a triagem, configura os critérios e gera a classificação, enquanto a comissão avaliadora confere a pontuação, lança as notas das etapas e emite os pareceres, sem que um perfil possa executar as atribuições do outro;
- ❑ **Integração de baixo atrito:** o módulo opera sobre os dados reais do sistema base sem duplicação de cadastros, sem sincronizações por rede e sem alterações no código do sistema existente;
- ❑ **Potencial de redução do esforço operacional:** tarefas antes manuais — conferência de presença de documentos, cálculo de pontuação, ordenação de candidatos e produção de relatórios — passam a ser executadas ou apoiadas pelo sistema, restando à comissão as decisões de mérito. Trata-se de benefício esperado, cuja magnitude não foi medida neste trabalho.

Como limitação, o sistema ainda não foi submetido a uso real em um processo seletivo completo. Essa etapa, juntamente com a avaliação empírica comparativa entre o fluxo manual e o assistido, constitui trabalho futuro discutido no Capítulo 6.

Utilização do Sistema

Este capítulo funciona como manual do usuário do módulo de avaliação semiautomática. Enquanto o Capítulo 4 tratou de como o sistema foi construído, aqui se descreve como ele é operado: cada funcionalidade é percorrida na ordem em que aparece no processo seletivo real, com a tela correspondente e o detalhamento das ações disponíveis.

As telas apresentadas foram obtidas sobre o banco de demonstração distribuído com o sistema, que reproduz um processo seletivo do PPGCO com 18 inscrições, quatro linhas de pesquisa e candidatos de mestrado e de doutorado.

O capítulo está organizado segundo os dois perfis de acesso descritos no RNF008. À **coordenação do curso** cabem o acesso ao sistema, a configuração dos critérios de avaliação, a triagem documental, a geração da classificação e a emissão dos relatórios; à **comissão avaliadora**, a conferência da pontuação, o ajuste de quantitativos, o lançamento das notas das etapas avaliativas e a emissão dos pareceres. A trilha de auditoria é consultável por ambos.

5.1 Acesso ao Sistema

O acesso é feito com as mesmas credenciais institucionais já utilizadas no sistema de submissão de candidaturas, sem novo cadastro. A tela de autenticação, apresentada na Figura 9, solicita o e-mail institucional e a senha; o botão *Entrar* valida as credenciais e inicia a sessão.

Concluída a autenticação, o sistema identifica o perfil do usuário e ajusta a interface: as abas e os botões que não pertencem às suas atribuições simplesmente não são exibidos. O perfil em uso aparece no canto superior direito de todas as telas, abaixo do nome — *Coordenação* ou *Comissão Avaliadora* —, de modo que o usuário sempre saiba em que papel está operando. Um candidato que tente acessar o módulo é recusado, pois o perfil de candidato não tem permissão sobre a avaliação.

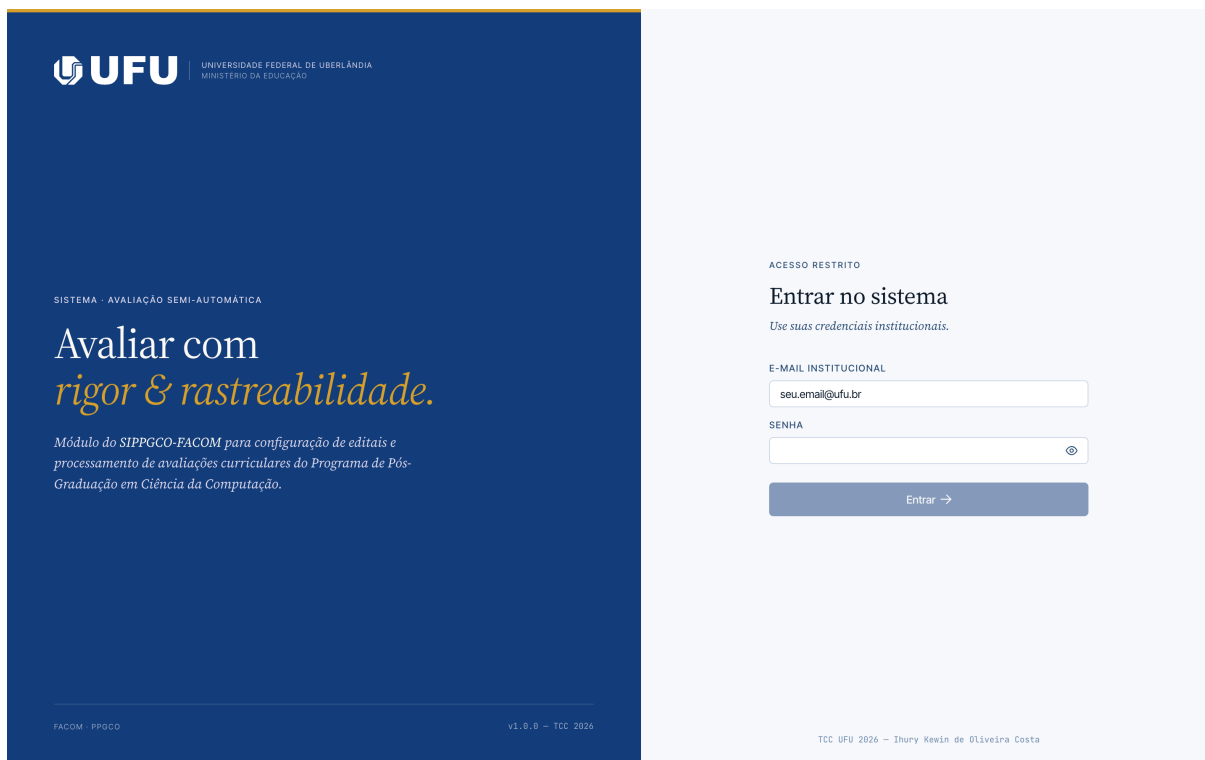


Figura 9 – Tela de autenticação do módulo de avaliação.

5.2 Seleção do Processo Seletivo

A tela inicial, apresentada na Figura 10, lista os processos seletivos existentes no sistema de submissão. Cada linha traz o número e o nome do edital, a faculdade, a vigência, o número de inscritos, a situação de configuração — se o barema já foi definido ou está pendente — e a situação do próprio edital. Os filtros no topo da listagem restringem a exibição por situação (publicado, rascunho ou arquivado) e por configuração, e o campo de busca localiza um edital por código ou nome.

Ao selecionar um edital, abre-se a sua área de trabalho, cujo cabeçalho fixo — visível em todas as telas seguintes — reúne a identificação do edital, o número de inscritos, a situação e a indicação de barema configurado. Logo abaixo ficam as abas que concentram o fluxo de avaliação: *Visão geral*, *Triagem*, *Avaliações*, *Classificação*, *Fórmula*, *Relatórios* e *Auditoria*. As abas *Triagem* e *Relatórios* são exibidas apenas para a coordenação.

À direita do cabeçalho ficam as duas ações de escopo do edital, ambas restritas à coordenação: *Editar fórmula*, que leva ao formulário de configuração dos critérios de avaliação, e *Sincronizar inscrições*, que importa do sistema de submissão as inscrições recebidas e monta a fila de triagem. A aba *Visão geral*, apresentada na Figura 11, resume o estado do edital: número de inscrições, avaliações geradas, avaliações com ajuste manual, vigência e os parâmetros de cálculo em vigor.

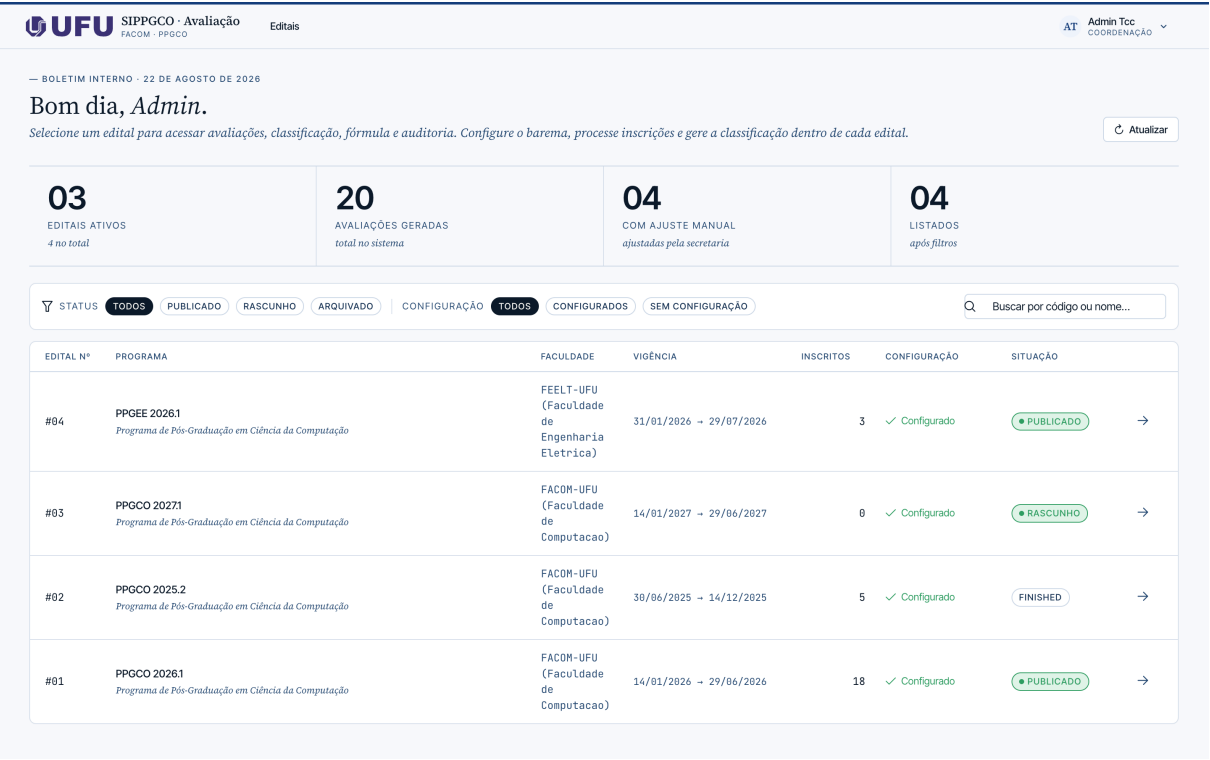


Figura 10 – Listagem dos processos seletivos disponíveis para avaliação.

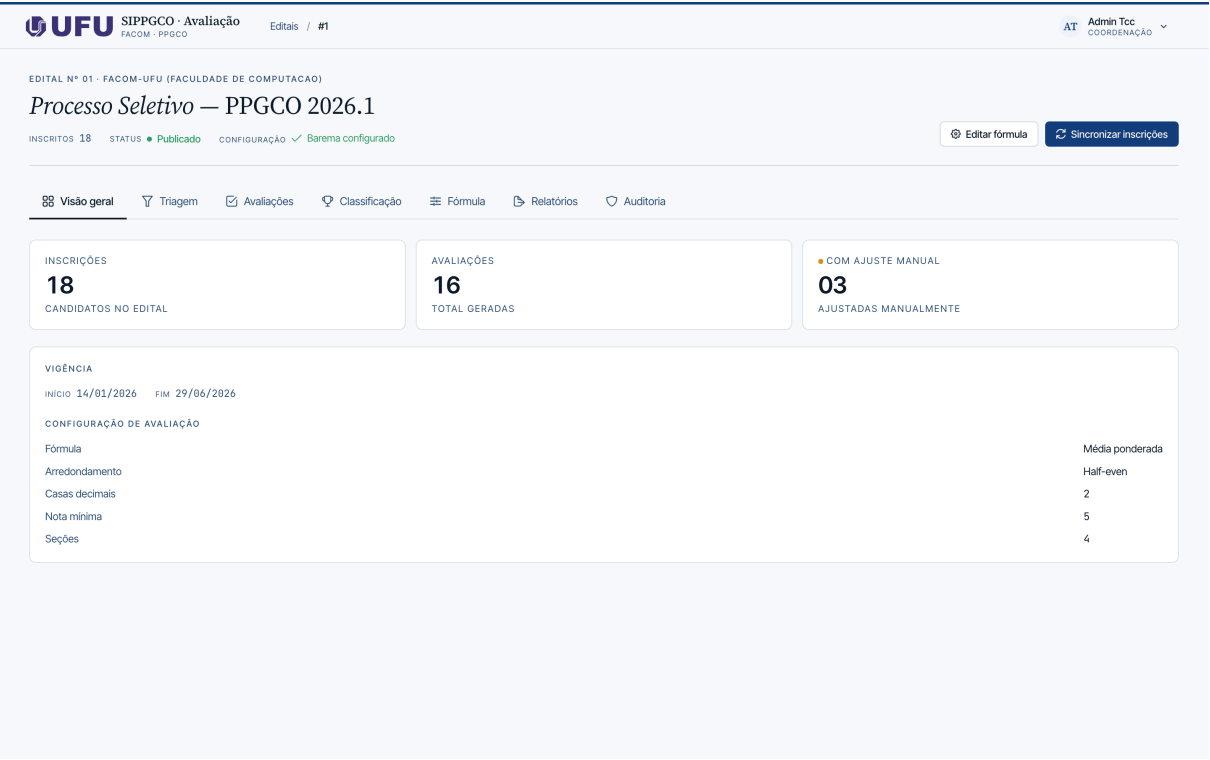


Figura 11 – Visão geral do edital, com os indicadores do processo e os parâmetros de cálculo vigentes.

5.3 Configuração dos Critérios de Avaliação

A configuração é a primeira tarefa do processo e responde aos requisitos RF001 a RF004. Ela é feita em duas telas: o formulário de parâmetros de cálculo e a aba *Fórmula*, onde o barema é montado seção a seção.

5.3.1 Definição dos Parâmetros de Cálculo

O botão *Criar fórmula* — ou *Editar fórmula*, quando a configuração já existe — abre o formulário da Figura 12. Nele a coordenação informa:

- ☐ o **tipo de fórmula** que consolida a nota final a partir das notas das seções: soma simples, média ponderada ou expressão personalizada;
- ☐ a **expressão personalizada**, quando esse tipo é escolhido, escrita em função dos identificadores das seções e validada no próprio formulário;
- ☐ a **regra de arredondamento** — para baixo, para cima ou *half-up* — e o **número de casas decimais**;
- ☐ a **nota de corte**, opcional, abaixo da qual o candidato é reprovado na classificação.

UFU SIPPGeo - Avaliação FACOM - PPGCO Editais / #1 / Fórmula / Configuração AT Admin Tcc COORDENAÇÃO

EDITAL Nº 01 - FACOM-UFU (FACULDADE DE COMPUTAÇÃO)

Processo Seletivo — PPGCO 2026.1

INSCRITOS 18 STATUS Publicado CONFIGURAÇÃO Barema configurado [Editar fórmula](#) [Sincronizar inscrições](#)

Visão geral Triagem Avaliações Classificação **Fórmula** Relatórios Auditoria

EDITAL Nº 01 - CONFIGURAÇÃO

Editar configuração de avaliação

Esta fórmula define como as notas das seções são combinadas em uma nota final por candidato.

TIPO DE FÓRMULA

- Soma simples**
 $\sum \text{score_seção}$
Soma direta das notas das seções, sem ponderação.
- Média ponderada** ☒
 $\sum (\text{score} \times \text{peso}) / \sum \text{peso}$
Cada seção contribui proporcionalmente ao seu peso.
- Personalizada**
sua expressão mathjs
Defina manualmente como combinar as seções.

ARREDONDAMENTO E NOTA DE CORTE

Regra de arredondamento
Arredondar (half-up)

Casas decimais
2

PREVIEW 7,8543 → 7,85

NOTA DE CORTE opcional
5
Candidatos com nota inferior são marcados como NÃO_CLASSIFICADO na classificação. Deixe vazio para sem corte.

[Cancelar](#) [Salvar alterações](#)

SEÇÕES DISPONÍVEIS

- s42** Formação Acadêmica peso 3.0 - máx — +
- s43** Produção Científica peso 4.0 - máx — +
- s44** Experiência Profissional e Academi... peso 2.0 - máx — +
- s45** Entrevista peso 3.0 - máx 10 +

MODELOS PRONTOS

Mínimo entre seções
 $\min(s42, s43)$
Pega a menor nota entre as duas

Média 60/40
 $(s42 \times 0.6) + (s43 \times 0.4)$
60% da primeira + 40% da segunda

Raiz da soma
 $\sqrt{s42 + s43}$

Figura 12 – Formulário de definição da fórmula, do arredondamento e da nota de corte.

O botão *Salvar* persiste a configuração e retorna à aba *Fórmula*. Toda alteração posterior nesses parâmetros dispara o reprocessamento automático das avaliações já existentes, e o sistema informa quantas foram recalculadas.

5.3.2 Montagem do Barema

A aba *Fórmula*, apresentada na Figura 13, é onde o barema é construído. À esquerda ficam as seções, numeradas em algarismos romanos, cada uma exibindo o peso, a pontuação máxima e os itens associados; à direita, um painel resume a fórmula em vigor, o arredondamento, a nota de corte e a composição do barema por peso.

Cada seção oferece três ações, identificadas por ícones à direita do seu cabeçalho: o **lápiz** abre a edição da seção, o **signal de mais** adiciona um documento avaliativo a ela e a **lixeira** a remove. Nos itens já associados, o **lápiz** edita o peso do item na seção e a **lixeira** desfaz a associação. O botão *Adicionar seção*, no topo, cria uma nova seção.

The screenshot shows the 'Fórmula' tab in the UFU SIPPGCO Avaliação system. The main area displays the 'Estrutura de pontuação' for the 'Processo Seletivo — PPGCO 2026.1' edital. It lists two sections: 'I Formacao Academica' (5 items) and 'II Producao Cientifica' (10 items). Each item is associated with a document type (e.g., 'Diploma de Graduação', 'Proficiência em Inglês') and a weight. The right sidebar shows the 'FÓRMULA DE CÁLCULO' (MÉDIA PONDERADA) and the 'RESUMO DO BAREMA'.

SEÇÃO	ITEM	DOCUMENTO	MODALIDADE	POR QUANTIDADE	PONTOS	MÁX	PESO
I Formacao Academica	I.1	Diploma de Graduação	MESTRADO	POR QUANTIDADE	10,00	MÁX 10	PESO 1.0
	I.2	Proficiência em Inglês	MESTRADO	POR QUANTIDADE	5,00	MÁX 5	PESO 1.0
	I.3	Diploma de Graduação	DOCTORADO	POR QUANTIDADE	10,00	MÁX 10	PESO 1.0
	I.4	Diploma de Mestrado	DOCTORADO	POR QUANTIDADE	15,00	MÁX 15	PESO 1.0
	I.5	Proficiência em Inglês	DOCTORADO	POR QUANTIDADE	5,00	MÁX 5	PESO 1.0
II Producao Cientifica							

FÓRMULA DE CÁLCULO
MÉDIA PONDERADA
$$\frac{\sum (\text{sec_score} \times \text{peso_sec})}{\sum \text{peso_sec}}$$

ARREDONDAMENTO: Half-even, DECIMAIS: 2
NOTA MÍN.: 5, ID CONFIG: #29

RESUMO DO BAREMA

SEÇÃO	ITEM	DOCUMENTO	MODALIDADE	POR QUANTIDADE	PONTOS	MÁX	PESO
I	Formacao Academica						peso 3.0
II	Producao Cientifica						peso 4.0
III	Experiencia Profissional e Academica						peso 2.0
IV	Entrevista						peso 3.0
Total							12.0

Figura 13 – Barema do edital, com seções documentais, itens avaliativos e etapa avaliativa.

Cada item exibe a modalidade de curso a que o documento se aplica, a regra de contagem — por quantidade ou por período —, os pontos por unidade, o teto e o peso dentro da seção. A indicação da modalidade é relevante porque o mesmo documento costuma ser cadastrado duas vezes no edital, uma para o mestrado e outra para o doutorado, com pontuações que podem diferir; o sistema pontua cada candidato apenas pelos itens da sua modalidade.

5.3.3 Cadastro de Seções e Etapas Avaliativas

O diálogo de cadastro, apresentado na Figura 14, atende aos dois tipos de seção previstos no RF002. O campo **Natureza da seção** alterna entre *Documental* e *Etapa avaliativa*, e o formulário se ajusta à escolha:

- ❑ em uma seção **documental**, a nota é apurada automaticamente dos documentos submetidos pelo candidato. Informam-se o nome, a descrição, o peso da seção na fórmula final, a pontuação máxima (opcional, que limita a soma dos itens) e a ordem de exibição;
- ❑ em uma **etapa avaliativa** — entrevista, análise de projeto de pesquisa ou qualquer outra etapa prevista no edital sem documentação associada —, o formulário passa a exigir a **nota mínima** e a **nota máxima** que delimitam o lançamento, e oferece o campo **Modalidades aplicáveis**. Deixá-lo vazio significa que a etapa vale para todas as modalidades; selecionar *Doutorado*, por exemplo, restringe a entrevista aos candidatos de doutorado, situação atual do programa.

The image shows a web interface for 'UFU SIPPGCO - Avaliação'. A modal window titled 'EDITAL - BAREMA' is open, showing the 'Novo seção avaliativa' form. The form has the following fields and values:

- NOME DA SEÇÃO**: Entrevista
- DESCRIÇÃO**: Texto exibido aos avaliadores
- NATUREZA DA SEÇÃO**: Documental (selected), Etapa avaliativa (available)
- NOTA MÍNIMA**: menor valor aceito (empty)
- MODALIDADES APLICÁVEIS**: Todas as modalidades (selected)
- PESO DA SEÇÃO**: 1
- NOTA MÁXIMA**: maior valor aceito (empty)
- ORDEM DE EXIBIÇÃO**: 0

Buttons at the bottom of the modal are 'Cancelar' and 'Salvar seção'. The background shows the 'Estrutura de pontuação' table with the following data:

Seção	Nome	Modalidade
I	Formacao Academica	5 ITENS AVALIATIVOS
I.1	Diploma de Graduação	MESTRADO
I.2	Proficiência em Inglês	MESTRADO
I.3	Diploma de Graduação	DOUTORADO
I.4	Diploma de Mestrado	DOUTORADO
I.5	Proficiência em Inglês	DOUTORADO
II	Producao Cientifica	10 ITENS AVALIATIVOS

Figura 14 – Cadastro de uma etapa avaliativa, com faixa de notas e modalidades aplicáveis.

Uma etapa avaliativa não aceita documentos associados, e por isso o botão de adicionar item não é oferecido nesse tipo de seção. Sua nota é lançada pela comissão avaliadora na tela de cada avaliação, como descrito na Seção 5.5.4. Do ponto de vista do cálculo,

contudo, a etapa é uma seção como qualquer outra: o peso atribuído a ela participa da mesma soma ponderada que combina a análise documental, que é o mecanismo pelo qual a entrevista entra na nota final.

5.4 Triagem Documental

A triagem é a conferência da documentação obrigatória, anterior à avaliação curricular, e atende ao RF009. É atribuição da coordenação do curso.

5.4.1 Importação das Inscrições

O botão *Sincronizar inscrições*, no cabeçalho do edital, importa do sistema de submissão as inscrições ainda não conhecidas pelo módulo. Para cada uma, cria-se uma triagem pendente com um item de conferência por documento obrigatório **da modalidade da inscrição** — um candidato de doutorado, portanto, não é cobrado pelos documentos exigidos apenas no mestrado. Ao final, o sistema informa quantas inscrições foram importadas, quantos documentos há a conferir e quantas avaliações foram reprocessadas. A ação é idempotente: acionada novamente, importa apenas o que houver de novo.

5.4.2 Acompanhamento da Fila

A aba *Triagem*, apresentada na Figura 15, reúne todas as inscrições importadas. Os quatro contadores no topo — total, pendentes, deferidas e indeferidas — resumem o andamento da conferência e funcionam como filtros: clicar em um deles restringe a listagem à situação correspondente. A tabela traz, por inscrição, o número da triagem, o candidato, a modalidade de curso, a modalidade de cota declarada, a situação, o progresso da conferência e a data da última alteração. A seta à direita de cada linha abre o detalhe daquela inscrição.

5.4.3 Conferência dos Documentos

O detalhe da triagem, apresentado na Figura 16, lista os documentos obrigatórios da inscrição. Para cada um, exibem-se a descrição do documento, os arquivos anexados pelo candidato e a decisão registrada. O ícone de link ao lado de cada anexo abre o arquivo em nova aba, para conferência visual.

Cada documento oferece dois botões: *Deferir*, que o aprova sem exigir justificativa, e *Indeferir*, que abre um diálogo em que a justificativa é obrigatória e fica registrada em auditoria.

O indeferimento de um documento **não encerra** a conferência: os demais seguem pendentes e precisam ser analisados, de modo que o candidato receba o parecer completo

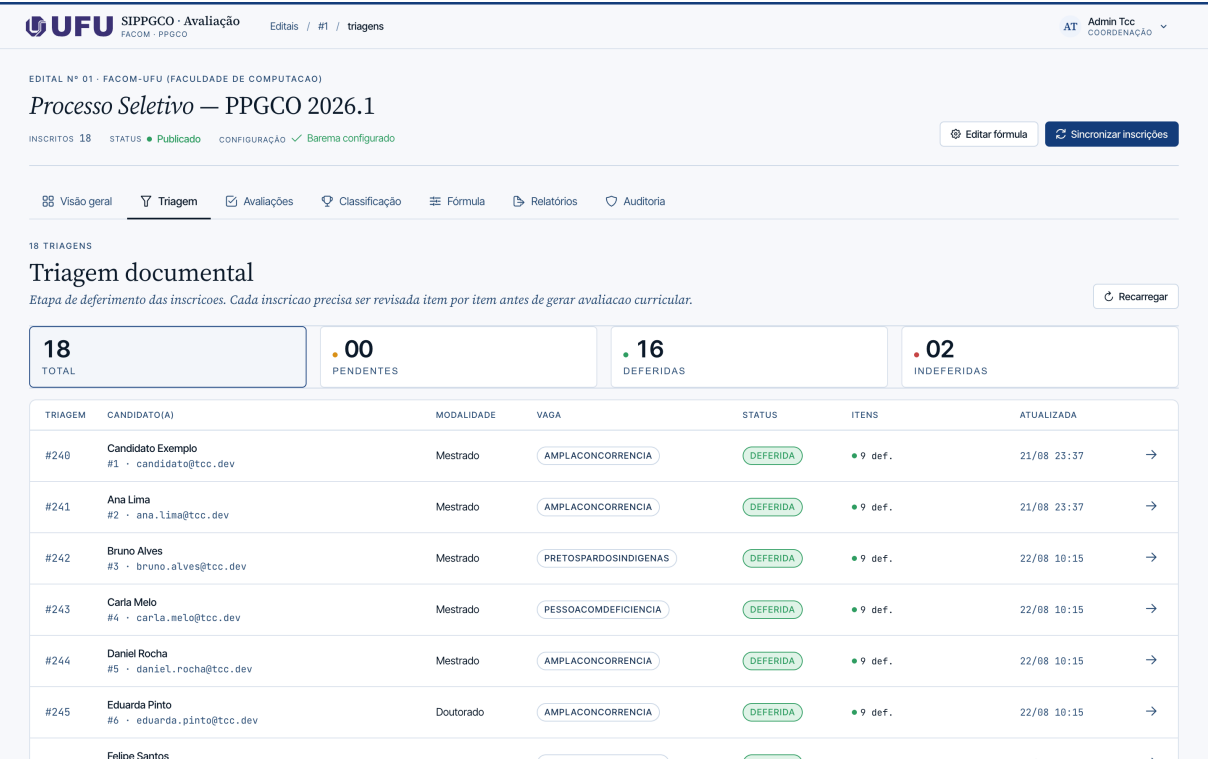


Figura 15 – Fila de triagem do edital, com os contadores por situação e o progresso da conferência.

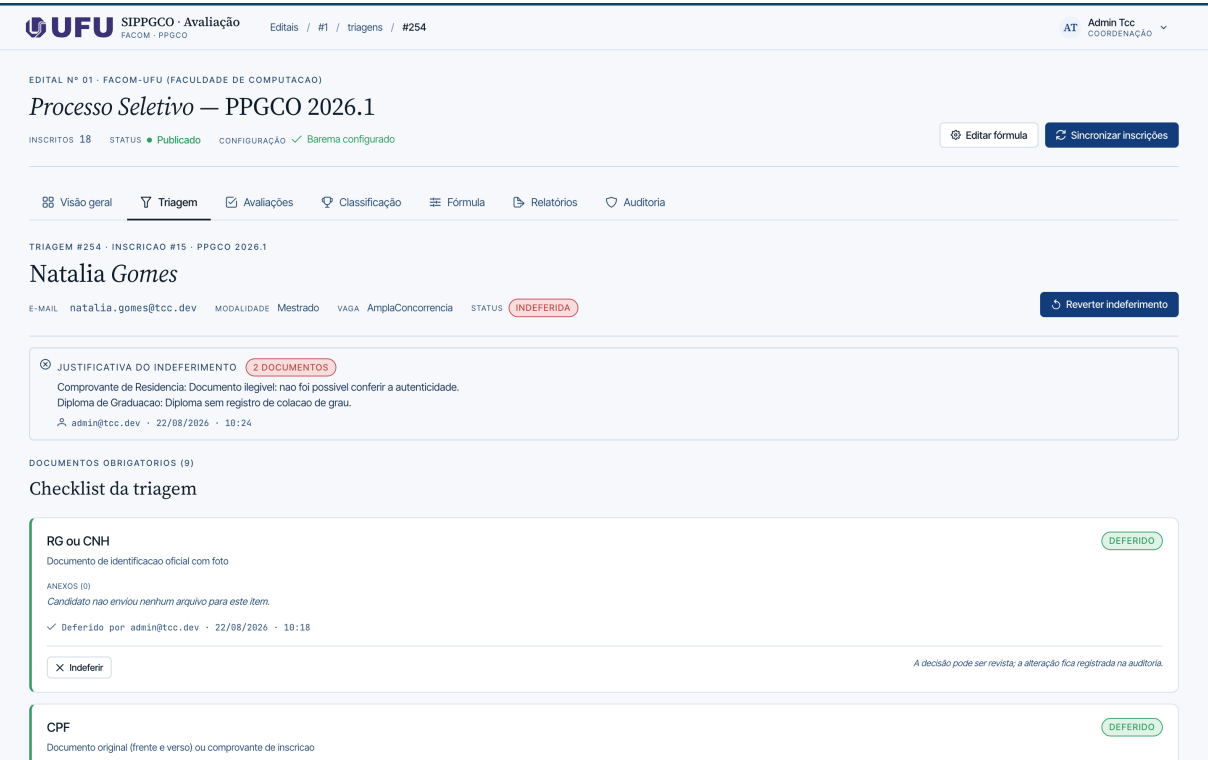


Figura 16 – Conferência documento a documento, com a justificativa reunida dos itens indeferidos.

sobre a sua documentação. Um aviso no topo da tela indica quantos documentos ainda estão sem decisão. A situação da inscrição é **derivada** do conjunto de decisões, e não informada diretamente — por isso não existe um botão de deferir ou indeferir a inscrição. Analisado o último documento, o sistema conclui:

- ❑ se todos os documentos foram deferidos, a inscrição fica **deferida** e a avaliação curricular é criada automaticamente;
- ❑ se ao menos um foi indeferido, a inscrição fica **indeferida**, e a justificativa apresentada reúne as justificativas de todos os documentos indeferidos, identificadas pelo nome de cada documento. Havendo avaliação previamente gerada, ela é removida.

A decisão de um documento pode ser revista a qualquer momento, e a situação da inscrição é recalculada em seguida; a alteração fica registrada na trilha de auditoria, com autor e data.

5.4.4 Reversão de Indeferimento

Toda decisão de triagem é passível de recurso. Quando o recurso é acolhido, o botão *Reverter indeferimento*, disponível no cabeçalho das inscrições indeferidas, abre o diálogo da Figura 17. A justificativa do recurso é obrigatória; ao confirmar, todos os documentos indeferidos passam a deferidos, a inscrição retorna ao fluxo de avaliação e a avaliação curricular é recriada. As justificativas originais, a justificativa do recurso, o autor e a data permanecem na trilha de auditoria.

5.5 Avaliação dos Candidatos

Concluída a triagem, a avaliação curricular passa a ser atribuição da comissão avaliadora, e atende aos requisitos RF005, RF008 e RF010.

5.5.1 Fila de Avaliações

A aba *Avaliações*, apresentada na Figura 18, lista as avaliações geradas. Cada linha traz o candidato, a modalidade, a nota final apurada, a indicação de que há ajuste manual aplicado e a data do último processamento. A seta à direita abre o detalhe. Inscrições indeferidas na triagem não aparecem aqui, pois não geram avaliação.

5.5.2 Conferência da Demonstrativo do Cálculo

O detalhe da avaliação, apresentado na Figura 19, é a tela central do trabalho da comissão. Ela expõe integralmente como a nota foi obtida, seção por seção:

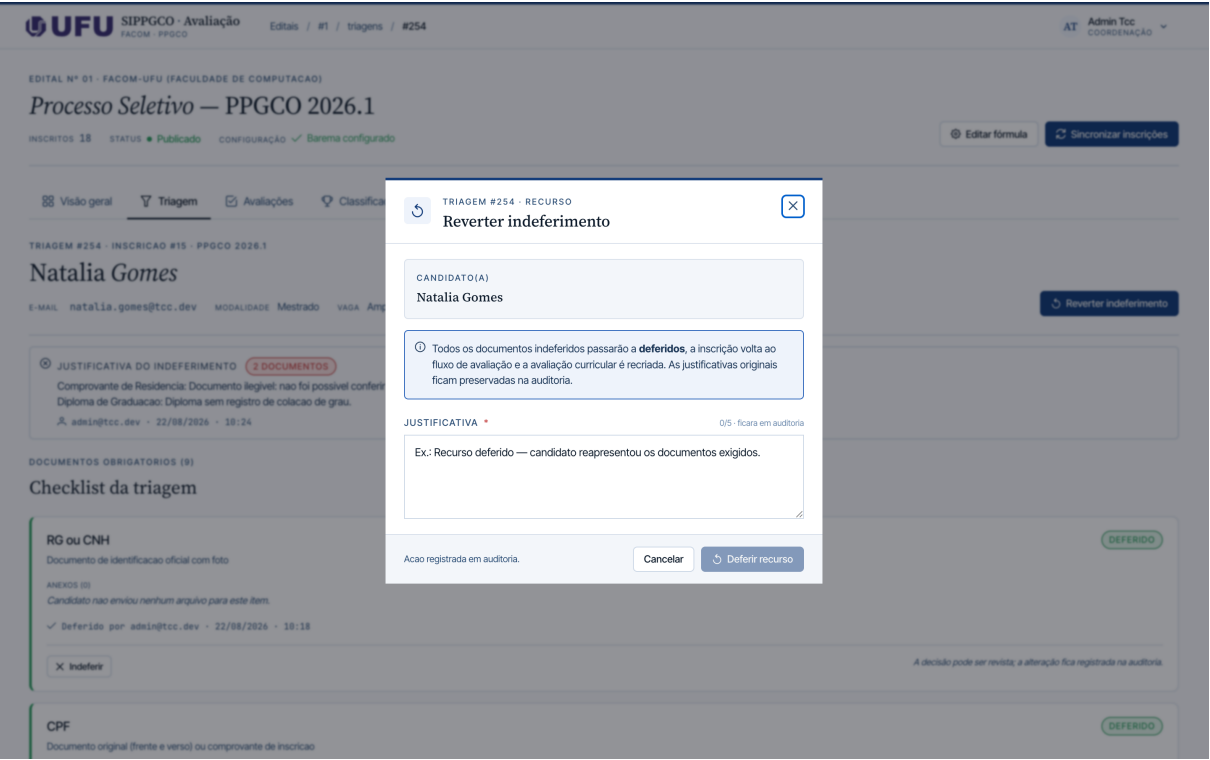


Figura 17 – Deferimento de recurso: reversão do indeferimento mediante justificativa.

UFU SIPPGCO - Avaliação

EDITAIS / #1 / Avaliações

Admin Tcc COORDENAÇÃO

EDITAL Nº 01 - FACOM-UFU (FACULDADE DE COMPUTAÇÃO)

Processo Seletivo — PPGCO 2026.1

INSCRITOS 18 STATUS Publicado CONFIGURAÇÃO Barema configurado

Editar fórmula Sincronizar inscrições

Visão geral Triagem Avaliações Classificação Fórmula Relatórios Auditoria

16 AVALIAÇÕES

Avaliações curriculares

Notas calculadas pela fórmula configurada no barema. Ajustes manuais recalculam automaticamente.

Recarregar

AValiação	CANDIDATO(A)	MODALIDADE	VAGA	NOTA FINAL	AJUSTES	ATUALIZADA
#305	Otávio Fonseca #16 - otavio.fonseca@tcc.dev	Mestrado	AMPLA CONCORRÊNCIA	57,49	—	22/08 10:19 →
#293	Bruno Alves #3 - bruno.alves@tcc.dev	Mestrado	PRETOS/PARDOS/INDÍGENAS	33,09	—	22/08 10:19 →
#295	Daniel Rocha #5 - daniel.l.rocha@tcc.dev	Mestrado	AMPLA CONCORRÊNCIA	31,40	—	22/08 10:19 →
#296	Eduarda Pinto #6 - eduarda.pinto@tcc.dev	Doutorado	AMPLA CONCORRÊNCIA	30,44	21	22/08 10:19 →
#298	Gabriela Cunha #8 - gabriela.cunha@tcc.dev	Mestrado	PESSOAS DO POLÍGLO/HUMANITÁRIA	28,04	—	22/08 10:19 →
#292	Ana Lima #2 - ana.lima@tcc.dev	Mestrado	AMPLA CONCORRÊNCIA	25,91	—	22/08 10:19 →

Figura 18 – Avaliações geradas a partir das inscrições deferidas na triagem.

□ no cabeçalho de cada seção, a **nota bruta** — a soma dos itens antes do teto da seção — e a **nota final ponderada**;

- ❑ em cada item, a **quantidade alegada** pelo candidato (em unidades ou meses), os **pontos por unidade**, o **teto** do item, o **peso** na seção e a **nota resultante**;
- ❑ no painel à direita, a **nota final**, a nota mínima de aprovação, o número de ajustes aplicados e os parâmetros de cálculo em vigor.

The screenshot displays the UFU SIPP/PGCO evaluation system interface. At the top, it shows the UFU logo and navigation links for 'Editais / #1 / Avaliações / #296'. The main header identifies the process as 'Processo Seletivo — PPGCO 2026.1' with 18 applicants and a published status. Below this, a navigation bar includes 'Visão geral', 'Avaliações' (selected), 'Classificação', 'Fórmula', and 'Auditoria'.

The main content area shows the candidate 'Eduarda Pinto' with their email and exam details. A section titled 'Demonstrativo do cálculo' provides a detailed breakdown of scores by section:

Seção	Item	Quantidade	Pontos por Unidade	Teto	Peso	Nota Bruta	Nota Final Ponderada
I - Formação Acadêmica	I.1 - Diploma de Graduação	8,00 unidades	10,00	10,00	1,00	0,00	0,00
	I.2 - Diploma de Mestrado	1,00 unidades	15,00	15,00	1,00	15,00	15,00
	I.3 - Proficiência em Inglês	1,00 unidades	5,00	5,00	1,00	5,00	5,00
	Total I					20,00	20,00
II - Produção Científica	II.1 - Publicações Qualis A1-A4	2,00 unidades	10,00	30,00	1,50	30,00	30,00
	Total II					52,20	52,20

On the right side, a dark box displays the 'NOTA FINAL' as 30,44, along with the minimum approval score (5), number of adjustments (1), and the date of the last update (22/08/2026). Below this, a 'CONFIGURAÇÃO' section shows the formula used (Média ponderada), the number of decimal places (2), the rounding method (ARRED: Half-even), and the number of sections (4).

Figura 19 – Demonstrativo do cálculo da avaliação, com as notas por item e por seção e o registro do ajuste manual.

Cada item pode ser expandido, revelando as submissões de comprovantes correspondentes e o cálculo detalhado; os arquivos podem ser abertos para conferência. Itens marcados com a etiqueta *no teto* tiveram a pontuação limitada pelo teto do item, e itens marcados como *ajustado* tiveram a quantidade alegada revista pela comissão. O botão *Exibir parecer*, no canto superior direito, leva ao parecer preliminar do candidato.

5.5.3 Ajuste da Pontuação

O ícone de lápis à direita de cada item abre o diálogo de ajuste, em que a comissão informa a nova quantidade alegada e a justificativa obrigatória. O ajuste altera apenas o quantitativo pleiteado pelo candidato — não a nota diretamente —, e a pontuação é recalculada pelo motor aplicando os pontos por unidade, o teto e o peso sobre o novo valor. Ao confirmar, a avaliação é reprocessada imediatamente, sem ação adicional, e a nota final exibida já reflete a mudança. O valor anterior, o novo valor, o autor e a justificativa ficam permanentemente registrados, e sobrevivem a reprocessamentos posteriores.

5.5.4 Lançamento das Notas das Etapas Avaliativas

As seções cadastradas como etapa avaliativa aparecem no demonstrativo do cálculo identificadas pela etiqueta *etapa avaliativa*. Enquanto não houver nota registrada, a etapa exibe a etiqueta *sem nota lançada* e vale zero no cálculo, situação sinalizada na própria tela para que a comissão não conclua a avaliação por engano.

O botão *Lançar nota* — ou *Corrigir nota*, quando já existe registro — abre o diálogo da Figura 20. Ele exibe a faixa de notas aceita e o peso da etapa, e solicita a nota e uma observação opcional. O campo de nota não aceita valor fora da faixa declarada: ao receber um número acima do máximo ou abaixo do mínimo, ajusta-o ao limite correspondente; a mesma faixa é validada no servidor, de modo que nenhum valor fora dela chega a ser registrado. O lançamento também não é oferecido em candidato de modalidade a que a etapa não se aplica, pois nesse caso a etapa sequer compõe o barema da inscrição. Ao confirmar, a avaliação é reprocessada e a nota final passa a incorporar a etapa pelo peso da seção.

AVALIAÇÃO #296 - ETAPA
Corrigir nota da etapa

ETAPA AVALIATIVA
Entrevista
Arguição sobre o projeto de pesquisa e a trajetória acadêmica
NOTA DE 0 A 10 PESO 3.0

Nota atual 9,50, lançada por comissao@tcc.dev. A correção fica registrada na auditoria e reprocessa a nota final.

NOTA * entre 0 e 10
9,5

OBSERVAÇÃO opcional - fica em auditoria
Boa articulação do projeto com a linha de pesquisa.

A nota final é recalculada na hora. Cancelar Salvar correção

EDITAL Nº 01 - FACOM-UFU (FACULDADE DE COMPUTAÇÃO)
Processo Seletivo — PPGCO 2026.1
INSCRITOS 18 STATUS Publicado CONFIGURAÇÃO Barema configurado

AValiação #296 - INSCRIÇÃO #6 - EDITAL Nº 1
Eduarda Pinto
E-MAIL: eduarda.pinto@tcc.dev MODALIDADE Doutorado VAGA Ampla

DETALHAMENTO POR SEÇÃO
Demonstrativo do cálculo

I Formação Acadêmica	
I.1 > Diploma de Graduação	AJUSTADO
I.2 > Diploma de Mestrado	
I.3 > Proficiência em Inglês	

II Produção Científica	
II.1 > Publicações Qualis A1-A4	# 2,00 unidades × 10,00 TETO 30,00 PESO 1,50 30,00

BRUTA 52,20 FINAL PONDERADA 52,20

NOTA FINAL
30,44
Mín. aprovação 5
Ajustes 1
Atualizada 22/08/2026 - 10:19

CONFIGURAÇÃO
FÓRMULA Média ponderada 2
ARRED. Half-even SEÇÕES 4

Figura 20 – Lançamento da nota de uma etapa avaliativa sem documento associado.

5.6 Classificação e Documentos do Processo

5.6.1 Geração da Classificação

A aba *Classificação* reúne as versões geradas para o edital e é consultável pelos dois perfis, uma vez que o parecer preliminar depende da situação apurada nela. A **geração**, contudo, é atribuição exclusiva da coordenação: apenas a esse perfil se oferece o botão *Gerar classificação*, que produz uma nova versão a partir das notas vigentes, permitindo registrar uma observação que a identifique.

A tela de uma classificação, apresentada na Figura 21, começa por quatro indicadores: total de candidatos, aprovados dentro das vagas, classificados além das vagas e reprovados abaixo do corte. Em seguida, a lista é apresentada **agrupada por linha de pesquisa e modalidade de curso**, que é o recorte em que as vagas são publicadas no edital. O cabeçalho de cada grupo informa a linha, a modalidade e o número de vagas ofertadas; dentro dele, os candidatos aparecem ordenados da maior para a menor pontuação, com a colocação obtida naquele grupo e a situação apurada:

- ❑ **aprovado** — nota igual ou superior ao corte e colocação dentro do número de vagas da linha;
- ❑ **classificado** — nota igual ou superior ao corte, porém além do número de vagas;
- ❑ **reprovado** — nota inferior à nota de corte.

Quando a situação não é de aprovação, o sistema apresenta a justificativa correspondente, indicando se a nota ficou abaixo do corte ou se a colocação excedeu as vagas da linha. Os botões *Exportar CSV* e *Imprimir*, no topo da tela, geram respectivamente o arquivo para planilha e a versão para impressão.

5.6.2 Emissão do Parecer Preliminar

O parecer preliminar, apresentado na Figura 22, é o documento que a comissão avaliadora emite sobre um candidato. Ele é alcançado pelo botão *Exibir parecer*, no detalhe da avaliação, e consolida em formato de documento oficial a identificação do candidato, a fórmula e o arredondamento aplicados, as notas por seção e por item, os ajustes manuais com suas justificativas e a situação do candidato na classificação, com a colocação obtida na sua linha de pesquisa.

O botão *Baixar PDF* salva o documento no formato adequado ao arquivo do processo, e *Voltar à avaliação* retorna ao demonstrativo do cálculo. Quando a avaliação ou a configuração do edital é alterada após a emissão, o parecer é automaticamente sinalizado como desatualizado e o botão *Atualizar parecer* passa a ser oferecido, de modo que o documento nunca seja arquivado divergente do cálculo vigente.

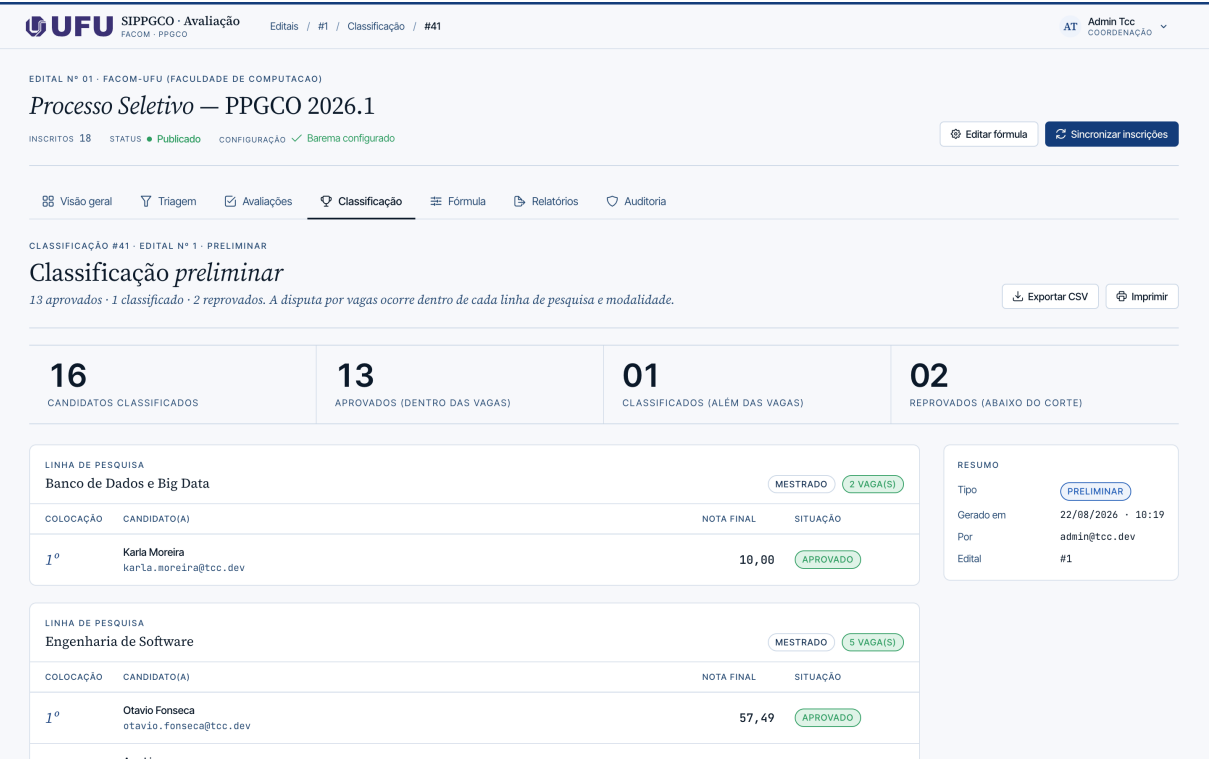


Figura 21 – Classificação dos candidatos agrupada por linha de pesquisa e modalidade.

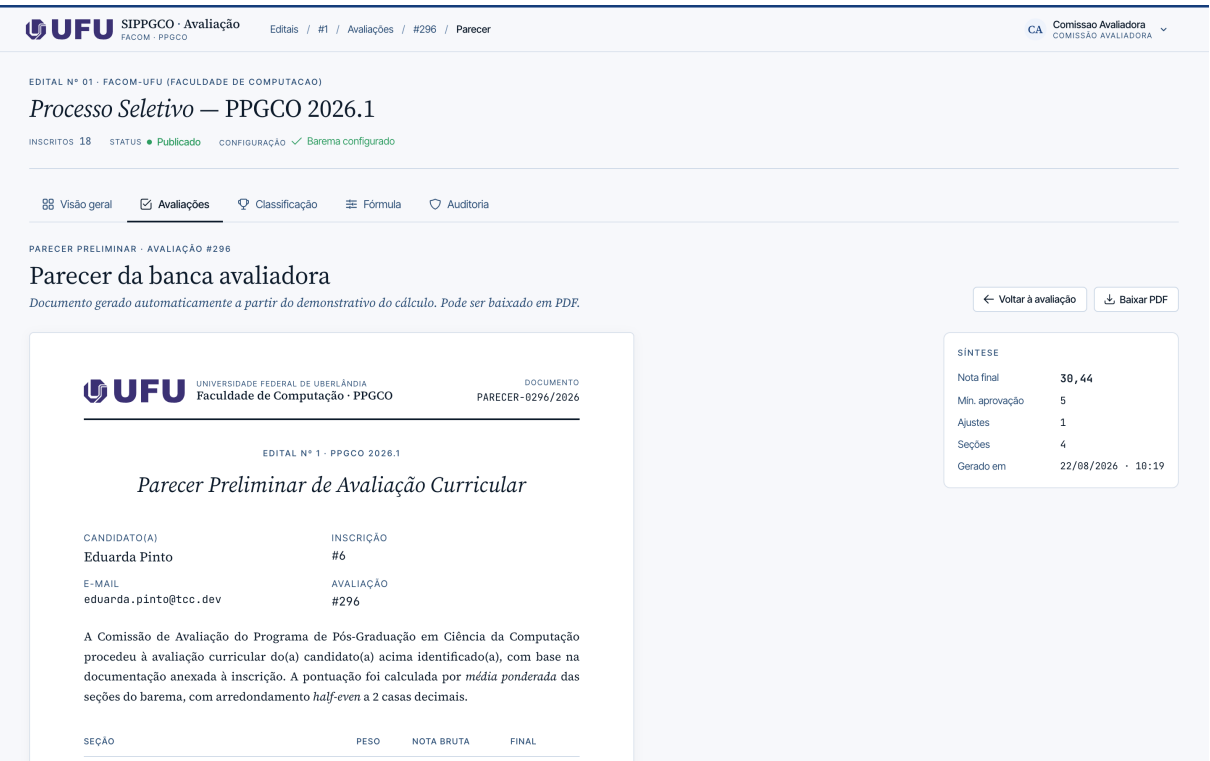


Figura 22 – Parecer preliminar de avaliação curricular, pronto para exportação em PDF.

5.6.3 Emissão dos Relatórios

A aba *Relatórios*, restrita à coordenação, produz as relações oficiais do processo seletivo. O seletor no topo alterna entre os dois documentos, e os botões *Exportar PDF* e *Exportar CSV* salvam a relação exibida.

O **relatório de triagem**, apresentado na Figura 23, apresenta as inscrições separadas em deferidas, indeferidas e ainda em conferência, com o candidato, a modalidade e a linha de pesquisa. A chave *Exibir justificativas* distingue os dois usos previstos pela coordenação: ativada, o documento traz o motivo de cada indeferimento e destina-se ao arquivo interno; desativada, produz a relação sem justificativas, adequada à divulgação pública dos resultados.

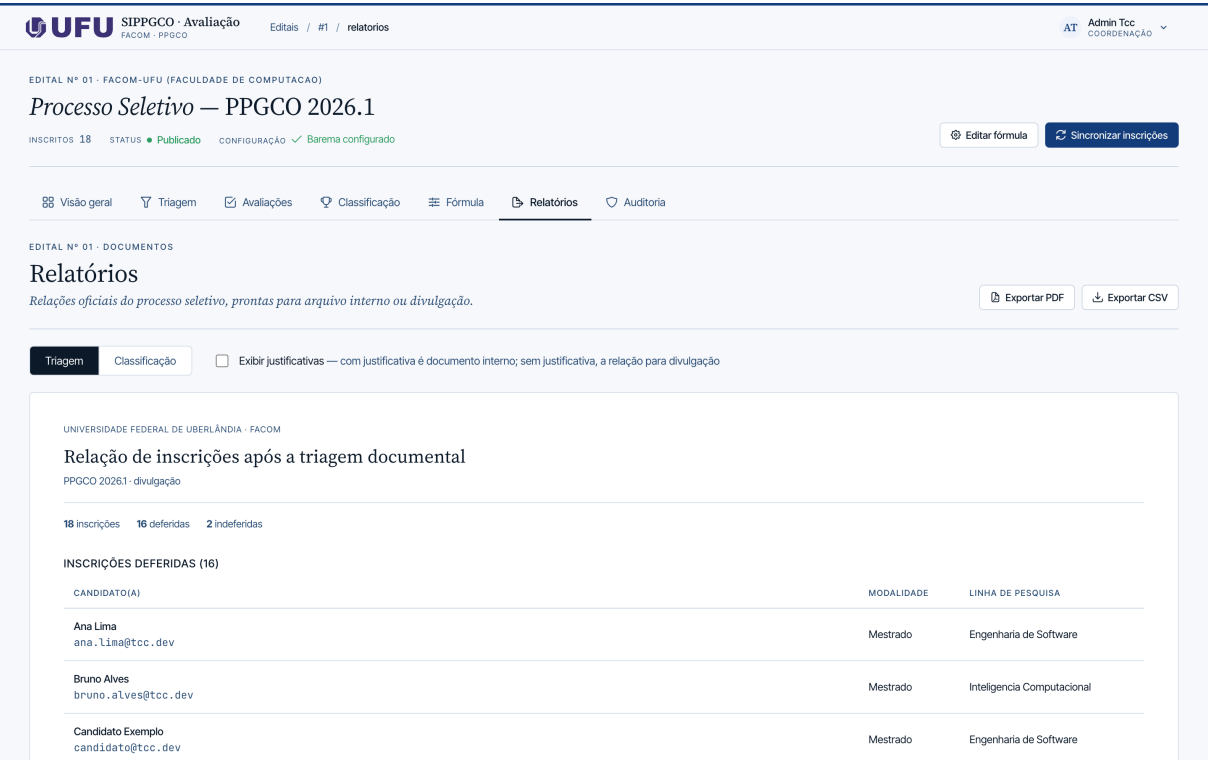


Figura 23 – Relação das inscrições após a triagem documental, com a chave de exibição das justificativas.

O **relatório de classificação**, apresentado na Figura 24, é a relação final: para cada linha de pesquisa cadastrada, apresenta a lista de candidatos ordenada da maior para a menor pontuação, categorizada em aprovado, classificado ou reprovado, com o número de vagas ofertadas no grupo.

5.7 Consulta da Trilha de Auditoria

A aba *Auditoria*, apresentada na Figura 25, reúne o histórico completo de ações do edital e atende ao RNF005. Cada registro traz a ação executada, a entidade afetada, o

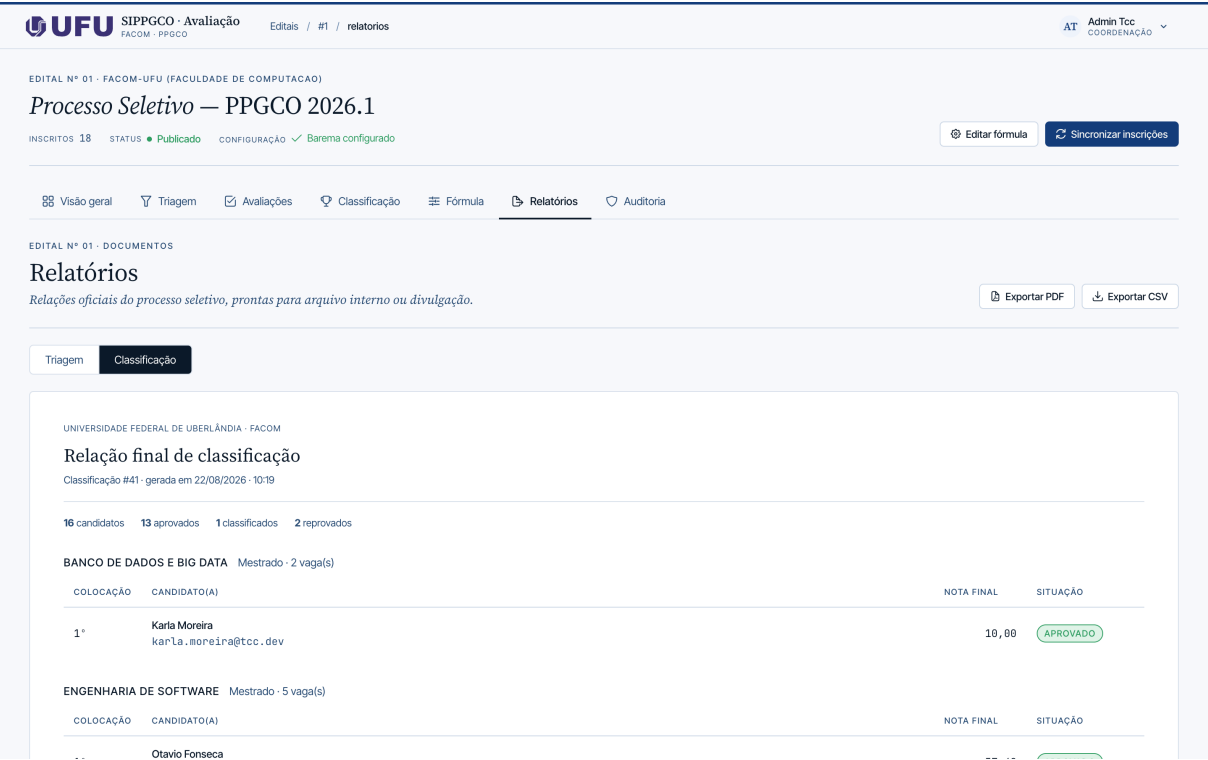


Figura 24 – Relação final de classificação, agrupada por linha de pesquisa e modalidade.

autor, a data e a hora.

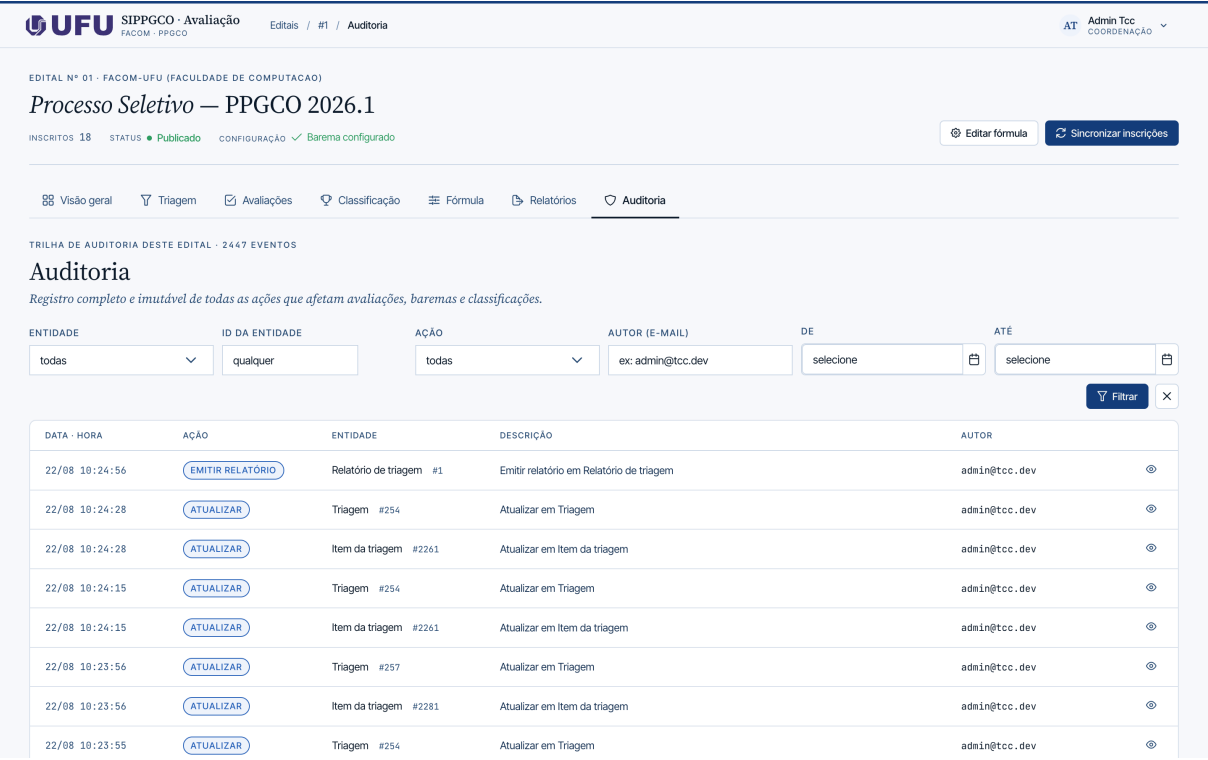


Figura 25 – Trilha de auditoria do edital, com filtros por ação, entidade, autor e período.

Os filtros no topo restringem a consulta por tipo de ação, entidade, autor e período,

e a paginação percorre o histórico. Clicando em um registro, abre-se o detalhamento da operação: no lançamento de uma nota de etapa, por exemplo, mostram-se a etapa, a nota anterior, a nova nota e a observação; na reversão de um indeferimento, a justificativa original de cada documento e a justificativa do recurso. É por essa tela que se reconstitui integralmente o histórico de decisões do processo seletivo, inclusive as que foram revistas.

Conclusões

Este trabalho apresentou o projeto e o desenvolvimento de um módulo de avaliação semiautomática para processos seletivos de pós-graduação, integrado ao sistema de submissão de candidaturas SIPPGCO-FACOM (DIAS, 2023; JUNKER, 2025). O objetivo geral — projetar, desenvolver e validar um módulo capaz de padronizar validações e ampliar a rastreabilidade e a transparência das decisões — foi atingido, conforme evidenciado pelo sistema funcional desenvolvido e pela verificação realizada por testes automatizados e funcionais. A redução do esforço manual, por sua vez, permanece como benefício potencial: o módulo passa a executar tarefas antes manuais, mas a magnitude dessa redução não foi medida, como se discute nas limitações do trabalho.

Quanto aos objetivos específicos definidos inicialmente, observa-se que:

- ❑ Os requisitos funcionais e não funcionais do módulo foram definidos segundo práticas consolidadas de engenharia de requisitos e estão devidamente documentados, incluindo os requisitos de triagem documental (RF009), de etapas avaliativas (RF010) e de relatórios (RF011), incorporados durante o desenvolvimento;
- ❑ A etapa estruturada de triagem documental foi implementada com conferência individual dos documentos obrigatórios da modalidade da inscrição, justificativa obrigatória nos indeferimentos, derivação automática da situação da inscrição a partir do conjunto de decisões e possibilidade de reversão mediante recurso;
- ❑ O cálculo automático de pontuação curricular foi modelado e implementado com base nas regras do edital — pontos por unidade, tetos, pesos por item e seção, fórmula final configurável, arredondamento parametrizado e piso zero — com demonstrativo do cálculo integral para conferência;
- ❑ As etapas avaliativas sem documento associado foram contempladas como seções de pleno direito do barema, com faixa de notas, modalidades aplicáveis e lançamento pela comissão avaliadora, combinando-se à análise documental pela mesma soma ponderada;

- ❑ As atribuições da coordenação do curso e da comissão avaliadora foram segregadas em perfis distintos, de modo que nenhum dos dois execute as ações do outro;
- ❑ Os mecanismos de apoio à identificação de inconsistências materializaram-se na exibição lado a lado das quantidades alegadas e dos comprovantes submetidos, no ajuste manual com justificativa e no reprocessamento imediato das notas;
- ❑ A classificação dos candidatos foi implementada combinando a nota de corte e o número de vagas da linha de pesquisa, ambos critérios configuráveis do edital, com as situações de aprovado, classificado e reprovado, e exportação em CSV;
- ❑ Os relatórios estruturados foram contemplados pelo parecer preliminar por candidato — com notas, demonstrativo do cálculo, ajustes e situação na classificação —, pelas relações de triagem e de classificação e pela trilha de auditoria completa de todas as ações;
- ❑ A solução foi projetada e implementada com *NestJS*, *Angular* e *PostgreSQL*, integrada ao sistema existente por banco de dados e autenticação compartilhados;
- ❑ O módulo foi validado por 54 testes unitários, 25 testes de integração de ponta a ponta e um plano de testes funcionais executado na interface, todos com resultado positivo.

O desenvolvimento também produziu aprendizados de projeto relevantes. A máquina de estados originalmente prevista para a avaliação revelou-se desnecessária diante do reprocessamento automático, e a simplificação resultante eliminou estados inconsistentes e reduziu a complexidade da interface. De modo análogo, a classificação por nota de corte e número de vagas por linha de pesquisa, sem alocação por cotas no módulo de avaliação, manteve o escopo da classificação aderente ao procedimento adotado atualmente pelas comissões avaliadoras para a elaboração das listas enviadas ao PPGCO. Quanto ao processo, o uso de inteligência artificial generativa mostrou-se produtivo na medida em que foi conduzido pela especificação: requisitos e modelos precederam o código, as decisões ficaram registradas para as sessões seguintes e a validação pela interface detectou defeitos que haviam passado pelos testes automatizados, o que confirma que a ferramenta não substitui a verificação humana.

6.1 Principais Contribuições

As principais contribuições deste trabalho são:

- ❑ **Um módulo de avaliação semiautomática funcional e integrado**, que completa o ciclo de informatização dos processos seletivos do PPGCO/FACOM/UFU

iniciado pelo sistema de submissão (DIAS, 2023; JUNKER, 2025): a etapa de avaliação, antes manual, passa a contar com triagem estruturada, cálculo automático de notas, classificação e relatórios;

- ❑ **Um modelo de regras de negócio parametrizado por edital**, em que seções, itens, pesos, tetos, fórmula final, arredondamento e nota de corte são configuráveis sem alteração de código, tornando o módulo reutilizável em diferentes editais e adaptável a outros programas;
- ❑ **Um fluxo de triagem documental com decisões justificadas e revisáveis**, que formaliza a etapa de deferimento das inscrições — cada documento obrigatório é conferido individualmente, os indeferimentos exigem justificativa registrada e a decisão pode ser revertida no acolhimento de recurso, sem perda do histórico — fortalecendo a isonomia e a transparência do processo;
- ❑ **Rastreabilidade integral das decisões**, por meio de trilha de auditoria que registra autor, data e detalhes de toda ação relevante, respondendo diretamente aos problemas de baixa auditabilidade e risco de contestação identificados na motivação do trabalho;
- ❑ **Uma estratégia de integração de baixo atrito com sistemas legados**, baseada em banco de dados e autenticação compartilhados com tabelas próprias para o novo domínio, documentada e reproduzível em contextos semelhantes de extensão de sistemas institucionais.

O sistema desenvolvido indica, assim, um caminho para reduzir o esforço operacional e a subjetividade da avaliação documental preservando a autonomia da comissão avaliadora — benefício esperado, cuja confirmação depende do uso em processos seletivos reais. Em outras palavras, as decisões de mérito permanecem humanas, apoiadas por cálculos determinísticos, conferência facilitada e registro completo.

6.2 Limitações

Apesar de cobrir integralmente os requisitos funcionais especificados, o trabalho apresenta limitações que delimitam o alcance dos resultados:

- ❑ **Ausência de uso em processo seletivo real**: o módulo foi implantado no servidor institucional da FACOM (RNF011), mas ainda não acompanhou um processo seletivo do início ao fim, com inscrições reais e prazos de recurso;
- ❑ **Ausência de estudo empírico comparativo**: não foi conduzida a avaliação quantitativa entre o fluxo manual e o assistido — originalmente prevista — capaz de mensurar objetivamente a redução de esforço e de inconsistências;

- ❑ **Usabilidade não validada:** o requisito de interface intuitiva (RNF001) não foi validado. Os testes funcionais executados na interface confirmam que as funcionalidades operam conforme especificado, mas não constituem avaliação de usabilidade, que exigiria a observação de usuários representativos da coordenação e da comissão avaliadora na execução de tarefas reais;
- ❑ **Verificação parcial de outros requisitos não funcionais:** o desempenho (RNF006) foi medido em máquina de desenvolvimento, e não no servidor institucional, onde o banco é compartilhado com o sistema de submissão; a conformidade com a LGPD (RNF009) foi tratada por diretrizes de projeto, sem auditoria formal;
- ❑ **Acoplamento ao sistema base:** a integração por banco de dados compartilhado, embora vantajosa, vincula o módulo ao esquema do sistema de submissão, exigindo coordenação em evoluções futuras desse esquema.

6.3 Trabalhos Futuros

Como desdobramentos deste trabalho, destacam-se:

- ❑ **Estudo empírico comparativo:** conduzir a avaliação comparativa entre o fluxo manual e o assistido em um processo seletivo real, medindo tempo de execução e incidência de inconsistências — etapa originalmente prevista que não pôde ser realizada no período do trabalho;
- ❑ **Avaliação de usabilidade:** validar o RNF001 com membros da coordenação e da comissão avaliadora, por meio de testes baseados em tarefas representativas do processo seletivo e da aplicação de um instrumento padronizado de percepção, como a escala SUS (*System Usability Scale*) (BROOKE, 1996);
- ❑ **Geração de PDF no servidor:** mover a exportação do parecer preliminar, hoje realizada no navegador, para o *backend*, garantindo documentos com formatação institucional padronizada e assinatura digital;
- ❑ **Processamento assíncrono em lote:** para editais com volume muito superior à demanda atual do PPGCO, é recomendado processar as avaliações em fila, com acompanhamento de progresso, em substituição ao processamento síncrono atual;
- ❑ **Controle de concorrência:** adicionar bloqueio de aplicação para impedir processamentos simultâneos do mesmo edital disparados por usuários distintos;
- ❑ **Etapa de recursos:** estender o fluxo para registrar e tramitar os recursos dos candidatos contra indeferimentos de triagem e notas atribuídas, com prazos e pareceres de resposta. O sistema já permite reverter uma decisão de triagem mediante

justificativa, com rastreabilidade integral, mas o recurso em si — sua interposição pelo candidato e sua tramitação — permanece fora do escopo;

- ❑ **Desdobramento das vagas por cota:** apoiar a atribuição das vagas por modalidade de cota, hoje realizada manualmente pela secretaria a partir da declaração feita pelo candidato na inscrição e da classificação produzida pelo sistema;
- ❑ **Notificações:** comunicar automaticamente candidatos e comissão sobre eventos relevantes (indeferimento, publicação da classificação e dos pareceres), por correio eletrônico ou pelo próprio sistema.

Referências

- Anthropic. **Claude Code overview — Claude Code Docs**. 2026. Disponível em: <<https://code.claude.com/docs/en/overview>>. Acesso em: 5 out. 2026. Citado 2 vezes nas páginas 46 e 47.
- BRASIL. **Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)**. Brasília, DF: [s.n.], 2018. Diário Oficial da União. Citado na página 32.
- BROOKE, J. SUS: a “quick and dirty” usability scale. In: JORDAN, P. W. et al. (Ed.). **Usability Evaluation in Industry**. London: Taylor & Francis, 1996. p. 189–194. Citado na página 81.
- CSV project. **csv-stringify — CSV stringifier for Node.js**. 2026. Disponível em: <<https://csv.js.org/stringify/>>. Acesso em: 10 jun. 2026. Citado na página 52.
- DIAS, V. L. **Desenvolvimento de um Sistema para Submissão de Candidaturas ao Processo Seletivo do Programa de Pós-graduação em Computação da UFU**. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal de Uberlândia, Uberlândia, Brasil, 2023. Bacharelado em Sistemas de Informação, Faculdade de Computação. Citado 3 vezes nas páginas 16, 78 e 80.
- Docker, Inc. **Docker — Accelerated Container Application Development**. 2026. Disponível em: <<https://www.docker.com>>. Acesso em: 10 jun. 2026. Citado 2 vezes nas páginas 49 e 52.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 7. ed. São Paulo: Pearson, 2018. Citado na página 41.
- FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. Tese (Doutorado) — University of California, Irvine, 2000. Citado na página 33.
- GitHub. **Spec Kit — Toolkit to help you get started with Spec-Driven Development**. 2025. Disponível em: <<https://github.com/github/spec-kit>>. Acesso em: 5 out. 2026. Citado na página 46.
- Google. **Angular — The web development framework for building modern apps**. 2026. Disponível em: <<https://angular.dev>>. Acesso em: 10 jun. 2026. Citado 2 vezes nas páginas 49 e 51.

ISO/IEC. **ISO/IEC 25010:2011 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models**. Geneva: [s.n.], 2011. International Organization for Standardization. Citado 2 vezes nas páginas 30 e 36.

JONG, J. d. **math.js — An extensive math library for JavaScript and Node.js**. 2026. Disponível em: <<https://mathjs.org>>. Acesso em: 10 jun. 2026. Citado na página 51.

JUNKER, G. B. N. **Desenvolvimento de um Sistema de Candidaturas para Processos Seletivos de Pós Graduação**. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal de Uberlândia, Uberlândia, Brasil, 2025. Bacharelado em Sistemas de Informação, Faculdade de Computação. Citado 11 vezes nas páginas 16, 21, 32, 35, 36, 41, 45, 48, 52, 78 e 80.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação Gerenciais**. 11. ed. São Paulo: Pearson, 2014. Citado na página 18.

MCDONNELL, C. **Zod — TypeScript-first schema validation with static type inference**. 2026. Disponível em: <<https://zod.dev>>. Acesso em: 10 jun. 2026. Citado 2 vezes nas páginas 50 e 51.

Microsoft. **TypeScript: JavaScript with syntax for types**. 2026. Disponível em: <<https://www.typescriptlang.org>>. Acesso em: 10 jun. 2026. Citado na página 51.

NestJS. **NestJS — A progressive Node.js framework**. 2026. Disponível em: <<https://nestjs.com>>. Acesso em: 10 jun. 2026. Citado 2 vezes nas páginas 49 e 51.

_____. **OpenAPI (Swagger) — NestJS Documentation**. 2026. Disponível em: <<https://docs.nestjs.com/openapi/introduction>>. Acesso em: 10 jun. 2026. Citado na página 52.

_____. **Rate Limiting — NestJS Documentation**. 2026. Disponível em: <<https://docs.nestjs.com/security/rate-limiting>>. Acesso em: 10 jun. 2026. Citado na página 52.

nestjs-pino. **nestjs-pino — Platform agnostic logger for NestJS based on Pino**. 2026. Disponível em: <<https://github.com/iamolegga/nestjs-pino>>. Acesso em: 10 jun. 2026. Citado na página 52.

OpenJS Foundation. **Jest — Delightful JavaScript Testing**. 2026. Disponível em: <<https://jestjs.io>>. Acesso em: 10 jun. 2026. Citado na página 52.

_____. **Node.js — Run JavaScript Everywhere**. 2026. Disponível em: <<https://nodejs.org>>. Acesso em: 10 jun. 2026. Citado 2 vezes nas páginas 49 e 51.

PlantUML. **PlantUML — Diagramming tool with simple textual descriptions**. 2026. Disponível em: <<https://plantuml.com>>. Acesso em: 10 jun. 2026. Citado na página 52.

PostgreSQL Global Development Group. **PostgreSQL: The World's Most Advanced Open Source Relational Database**. 2026. Disponível em: <<https://www.postgresql.org>>. Acesso em: 10 jun. 2026. Citado 2 vezes nas páginas 49 e 51.

PrimeTek. **PrimeNG — The Most Complete Angular UI Component Library**. 2026. Disponível em: <<https://primeng.org>>. Acesso em: 10 jun. 2026. Citado na página 51.

Prisma Data, Inc. **Prisma ORM — Next-generation Node.js and TypeScript ORM**. 2026. Disponível em: <<https://www.prisma.io>>. Acesso em: 10 jun. 2026. Citado 2 vezes nas páginas 49 e 51.

RICHARDS, M.; FORD, N. **Fundamentals of Software Architecture: An Engineering Approach**. Sebastopol: O'Reilly Media, 2020. Citado 2 vezes nas páginas 35 e 50.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2019. Citado 2 vezes nas páginas 18 e 45.

Apêndices

APÊNDICE **A**

Endpoints da API REST

Este apêndice lista os *endpoints* expostos pela API REST do módulo de avaliação. Todas as rotas são prefixadas por `/api` e, salvo a verificação de saúde (`/health`) e o *login* de desenvolvimento, exigem autenticação por *token* JWT. As operações de escrita são restritas conforme o RNF008: a configuração dos critérios, a triagem, a classificação e os relatórios cabem ao perfil de coordenação (*admin*); o ajuste de pontuação, o lançamento de notas de etapa e a emissão de pareceres, ao perfil de comissão avaliadora (*committee*). As consultas são acessíveis a ambos.

Tabela 3 – Endpoints da API REST do módulo de avaliação.

Método	Rota	Descrição
Autenticação e saúde		
GET	<code>/health</code>	Verificação de saúde da aplicação e do banco
POST	<code>/auth/login</code>	<i>Login</i> de desenvolvimento (e-mail e senha)
Editais / configuração (RF001–RF004)		
GET	<code>/editais</code>	Lista os processos seletivos
GET	<code>/editais/:id</code>	Detalha um edital e sua configuração
GET	<code>/editais/:id/documentos-avaliativos</code>	Lista os documentos avaliativos do edital
POST	<code>/editais/config</code>	Cria a configuração de avaliação (RF001)
PUT	<code>/editais/:id/config</code>	Atualiza a configuração (RF004) e reprocessa
DELETE	<code>/editais/:id/config</code>	Remove a configuração de avaliação
Seções e itens (RF002–RF003)		
GET	<code>/secoes</code>	Lista as seções de uma configuração

continua na próxima página

Tabela 3 – continuação

Método	Rota	Descrição
GET	/secoes/:id	Detalha uma seção
POST	/secoes	Cria uma seção (RF002)
PUT	/secoes/:id	Atualiza uma seção e repro- cessa
DELETE	/secoes/:id	Remove uma seção e repro- cessa
GET	/secao-itens	Lista os itens de uma seção
POST	/secao-itens	Associa um documento à se- ção (RF003)
PUT	/secao-itens/:sectionId/ :processDocumentId	Atualiza o peso/ordem de um item
DELETE	/secao-itens/:sectionId/ :processDocumentId	Desassocia um item da seção
Triagem (RF009)		
POST	/triagens/sync	Sincroniza inscrições e cria tri- agens pendentes
GET	/triagens	Lista triagens (com filtro por situação)
GET	/triagens/stats	Contadores por situação
GET	/triagens/:id	Detalha uma triagem e seus itens
POST	/triagens/:id/itens/ :processDocumentId/deferir	Defere um documento e rede- riva a situação
POST	/triagens/:id/itens/ :processDocumentId/indeferir	Indefere um documento (com justificativa)
POST	/triagens/:id/reverter	Reverte o indeferimento (defe- rimento de recurso)
Avaliações (RF005, RF008, RF010)		
POST	/avaliacoes/sync	Sincroniza e processa as avali- ações (RF005)
GET	/avaliacoes	Lista as avaliações de um edi- tal
GET	/avaliacoes/stats	Total e quantidade com ajuste manual
GET	/avaliacoes/:id	Detalha uma avaliação (de- monstrativo do cálculo)
POST	/avaliacoes/:id/ajustes	Registra ajuste manual e re- processa (RF008)
GET	/avaliacoes/:id/ajustes	Lista os ajustes manuais da avaliação

continua na próxima página

Tabela 3 – continuação

Método	Rota	Descrição
PUT	/avaliacoes/:id/etapas/:sectionId	Lança a nota de uma etapa avaliativa e reprocessa (RF010)
Classificação (RF006)		
POST	/rankings	Gera uma classificação
GET	/rankings	Lista as classificações de um edital
GET	/rankings/:id	Detalha uma classificação e suas entradas
GET	/rankings/:id/export.csv	Exporta a classificação em CSV
Pareceres (RF007)		
POST	/pareceres/:evaluationId	Gera o parecer preliminar
GET	/pareceres/:evaluationId	Consulta o parecer de uma avaliação
Relatórios (RF011)		
GET	/relatorios/triagem	Relação das inscrições após a triagem
GET	/relatorios/triagem/export.csv	Exporta a relação de triagem em CSV
GET	/relatorios/classificacao	Relação final agrupada por linha de pesquisa
Auditoria e arquivos (RNF005)		
GET	/auditoria	Consulta a trilha de auditoria (com filtros)
GET	/arquivos/submissoes-avaliativas/:id	Baixa o arquivo de uma submissão avaliativa
GET	/arquivos/documentos/:id	Baixa o arquivo de um documento

Plano de Testes Automatizados

Este apêndice detalha as suítes de testes automatizados do *backend*, complementando o plano de testes funcionais manuais apresentado na Tabela 2 (Capítulo 4). Ao final do desenvolvimento, a suíte reunia 54 testes unitários e 25 testes de integração de ponta a ponta, todos com execução bem-sucedida.

B.1 Testes unitários

Os testes unitários exercitam isoladamente as regras de negócio críticas, com as dependências de banco substituídas por *mocks* tipados. A Tabela 4 resume as suítes.

Tabela 4 – Suítes de testes unitários.

Suíte	Casos	Foco da verificação
<code>rounding.spec.ts</code>	7	Regras de arredondamento (<i>floor</i> , <i>ceil</i> , <i>half-up</i>) e casas decimais
<code>scoring.service.spec.ts</code>	9	Motor de pontuação: conversão de quantidade em pontos, aplicação de teto e peso, agregação por seção
<code>formula.service.spec.ts</code>	10	Fórmulas (soma simples, média ponderada e personalizada), <i>sandbox</i> de expressões e piso zero
<code>rankings.service.spec.ts</code>	8	Classificação por nota de corte e vagas da linha de pesquisa (aprovado, classificado e reprovado), disputa por par (linha, modalidade) e ordenação
<code>editais.service.spec.ts</code>	5	Criação e edição da configuração de avaliação
<code>jwt-auth.guard.spec.ts</code>	7	Validação do <i>token</i> JWT e resolução do usuário

Tabela 4 – continuação

Suíte	Casos	Foco da verificação
<code>roles.guard.spec.ts</code>	8	Autorização por papéis e segregação entre coordenação e comissão avaliadora nos dois sentidos
Total	54	

B.2 Testes de integração (ponta a ponta da API)

Os testes de integração exercitam o fluxo completo contra uma instância real do PostgreSQL, do *endpoint* HTTP até a persistência. A Tabela 5 lista os cenários, na ordem de execução, cobrindo o encadeamento triagem → avaliação → ajuste → classificação → parecer → relatórios → auditoria. O *token* de teste carrega o perfil desejado, de modo que a segregação entre coordenação e comissão avaliadora seja exercitada nos dois sentidos.

Tabela 5 – Cenários de teste de integração de ponta a ponta.

#	Cenário
1	GET <code>/health</code> retorna estado de saúde da aplicação e do banco
2	GET <code>/api/editais</code> lista o edital semeado
3	POST <code>/api/triagens/sync</code> cria triagens pendentes e os itens de conferência
4	Os itens de triagem criados respeitam a modalidade de curso de cada inscrição
5	GET <code>/api/triagens</code> lista as triagens do edital
6	GET <code>/api/triagens/:id</code> retorna o detalhe com os itens
7	A inscrição permanece PENDENTE enquanto restar item sem decisão
8	Decidido o último item, a inscrição fica DEFERIDA e a avaliação é criada
9	O indeferimento não interrompe a conferência e as justificativas são reunidas na inscrição
10	POST <code>/api/triagens/:id/reverter</code> defere o recurso e recria a avaliação
11	POST <code>/api/avaliacoes/sync</code> reprocessa as avaliações deferidas
12	GET <code>/api/avaliacoes/:id</code> retorna a avaliação com as notas calculadas
13	GET <code>/api/avaliacoes</code> traz a modalidade e a cota declarada de cada inscrição
14	Os itens pontuados são apenas os da modalidade da inscrição
15	POST <code>/api/avaliacoes/:id/ajustes</code> recalcula a nota automaticamente
16	Etapa restrita ao doutorado é recusada em inscrição de mestrado
17	PUT <code>/api/avaliacoes/:id/etapas/:secao</code> lança a nota e reprocessa a avaliação
18	Nota fora da faixa declarada na etapa é recusada
19	A coordenação é impedida de ajustar a pontuação de item avaliativo
20	A comissão é impedida de conduzir a triagem e de gerar a classificação
21	POST <code>/api/rankings</code> classifica por nota de corte e vagas da linha de pesquisa
22	POST <code>/api/pareceres/:id</code> gera o parecer estruturado
23	GET <code>/api/relatorios/triagem</code> separa deferidas e indeferidas, com e sem justificativa

Tabela 5 – continuação

#	Cenário
24	GET /api/relatorios/classificacao agrupa a relação por linha de pesquisa
25	GET /api/auditoria registra os eventos de triagem e avaliação