

# Sistema de Informação *Open-Source* para Auxiliar no Gerenciamento de Processos Seletivos

Bem Hur Ganem Silva  
ORCID: 0009-0001-3455-5755  
Instituto Federal da Bahia (IFBA)  
Vitória da Conquista-BA, Brasil  
bemhur.ganem@gmail.com

Letícia Porto Soares  
ORCID: 0009-0003-5435-6168  
Instituto Federal da Bahia (IFBA)  
Vitória da Conquista-BA, Brasil  
leticiaportosoares@gmail.com

Crescencio Lima  
ORCID: 0000-0002-0286-2056  
Instituto Federal da Bahia (IFBA)  
Vitória da Conquista-BA, Brasil  
crescencio@gmail.com

**Resumo—Contexto e Problema:** A COVID-19 obrigou governos a estabelecerem severas restrições sociais como parte da estratégia para lidar com a situação. Como consequência, a sociedade, organizações e instituições de ensino tiveram que se adaptar rapidamente. Neste cenário pandêmico, realizar as etapas inerentes à inscrição e realização de processos seletivos de pós-graduação de forma presencial tornou-se inviável. Além disso, identificou-se uma lacuna na oferta de soluções *open-source* e sem custo para esse tipo de processo. **Metodologia:** Nesse contexto, este trabalho tem como objetivo a implementação de um Sistema de Informação online *open-source* com a proposta de auxiliar nas etapas supracitadas e com o desafio de não gerar custos. Adotou-se uma abordagem de pesquisa aplicada com desenvolvimento incremental e prova de conceito. Utilizamos a demanda do processo seletivo da Pós-Graduação Lato Sensu em Desenvolvimento Web do Instituto Federal de Educação, Ciência e Tecnologia da Bahia campus Vitória da Conquista como prova de conceito para o desenvolvimento da plataforma. **Resultados:** A plataforma conta com ferramentas para matrícula dos alunos, acompanhamento do processo, suas etapas e resultados e um painel administrativo, no qual professores podem cadastrar e abrir processos seletivos e realizar as avaliações necessárias para sua conclusão. **Impacto:** Desta forma, entende-se que a ferramenta poderá proporcionar a automatização e aceleração de processos tanto para os organizadores da seleção quanto para os participantes. Além de estar apta a ter sua utilização ampliada para outros cursos de pós-graduação.

**Index Terms—**desenvolvimento web, sistemas de informação, processo seletivo, open-source, LGPD

## I. INTRODUÇÃO

O curso de Pós-Graduação Lato Sensu em Desenvolvimento Web (PGDW) do Instituto Federal da Bahia (IFBA), campus Vitória da Conquista, foi criado para atender a demanda por especialização dos profissionais da área de Computação. É um curso gratuito que tem como objetivo atender não somente a uma demanda local, mas também a outras regiões do estado da Bahia.

O primeiro processo seletivo do curso foi realizado em 2019 de forma presencial. O edital e formulários foram disponibilizados no site da Instituição. Os documentos necessários e formulários exigidos para confirmação da inscrição deveriam ser levados ao local onde seriam analisados e autenticados (se necessário) por membros da comissão do processo seletivo que permaneceram de plantão para tal atividade. Posterior-

mente, realizou-se uma prova em sala de aula como parte da pontuação total de cada inscrito.

O processo de inscrição em uma pós-graduação geralmente envolve um grande consumo de tempo por parte dos envolvidos. Para os inscritos existe a necessidade de gerar cópias dos documentos, títulos, comprovantes e atestados, impressão de formulários, deslocamento até a instituição e, por vezes, filas e horários específicos de atendimento. Como parte do público dos inscritos são pessoas envolvidas em atividades profissionais, esses procedimentos podem ser bastante limitantes. No que concerne aos avaliadores do processo, organizar centenas de papéis dos inscritos, fazer a avaliação das fichas de inscrição e manter as tarefas organizadas, atendendo aos prazos do edital, também pode gerar problemas.

Com a pandemia da COVID-19, medidas restritivas foram impostas no campus, o que impediria que o processo acontecesse de forma presencial para uma próxima turma. Esta situação motivou a automatização do processo seletivo do PGDW para proporcionar a realização da seleção de forma segura no contexto de restrições sanitárias. Automatização válida também no cenário pós pandemia, considerando que serão evitados o deslocamento dos inscritos até o campus, a necessidade da comissão se disponibilizar em plantões para receber os inscritos e as análises documentais de forma manual.

O segundo processo seletivo foi realizado em 2024 migrando parte do processo para formato online devido às normas institucionais adotadas após a pandemia. Os formulários foram digitalizados utilizando Google Forms e os documentos foram enviados digitalmente. A avaliação da documentação foi realizada de forma manual pelos membros do colegiado do curso. A prova do processo seletivo foi realizada presencialmente.

A ausência de soluções *open-source*, gratuitas e aderentes às especificidades de processos seletivos de pós-graduação representa uma lacuna relevante na literatura e nos sistemas disponíveis. Após pesquisa nos principais indexadores de conteúdo, utilizando os termos “sistema processo seletivo pós-graduação” e “*software* processo seletivo pós-graduação”, não foram encontrados sistemas web que contemplassem, de forma integrada e aberta, as nuances de uma seleção de pós-graduação, incluindo avaliação curricular por barema,

interposição de recursos, controle de etapas e conformidade com a LGPD. Dessa problemática, e da necessidade de uma solução acessível a instituições públicas com restrições orçamentárias, surgiu o interesse em desenvolver uma plataforma web que atendesse a essas necessidades.

Nesse contexto, este trabalho busca responder à seguinte questão de pesquisa: *É possível desenvolver um sistema de informação web, open-source e sem custos de infraestrutura, capaz de automatizar e gerenciar integralmente as etapas de um processo seletivo de pós-graduação, desde a inscrição até a divulgação do resultado definitivo?*

A proposta do projeto surgiu inicialmente para suprir a demanda do PGDW que serviu como prova de conceito para o desenvolvimento do sistema. Porém, o projeto pode atender a diferentes contextos de processos seletivos, sendo capaz de amparar processos de outros cursos de pós-graduação ou graduação, por exemplo.

Desta forma, os objetivos deste trabalho são: (i) o desenvolvimento de um sistema web *open-source* em que seja possível ter acesso às informações do curso, do processo seletivo, do corpo docente, às notícias e trabalhos de conclusão de cursos e artigos publicados pelos discentes; (ii) uma aplicação para cadastro e gerenciamento do processo seletivo pela comissão, e para inscrição, acompanhamento e interposição de recursos por parte dos inscritos.

O restante do artigo está estruturado da seguinte forma: Seção II apresenta o referencial teórico da pesquisa. A Seção III aborda os trabalhos relacionados. A Seção IV apresenta o desenvolvimento da proposta. A Seção V apresenta os resultados. A Seção VI descreve a metodologia adotada. Seção VII explica as conclusões da pesquisa e fala sobre os trabalhos futuros.

## II. FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta a fundamentação teórica do trabalho, abordando conceitos que caracterizam o sistema desenvolvido e ampararam a sua implementação.

### A. *Sistemas de Informação*

Um sistema é considerado um Sistema de Informação (SI) quando possui como meta primária gerenciar informações, efetuando a coleta, o armazenamento, o tratamento e o fornecimento das mesmas a fim de amparar decisões e processos de uma organização. Um SI consiste na combinação da Tecnologia da Informação (TI) com pessoas, processos, informações e documentos. Desta forma, tanto o aspecto automatizado quando o social devem estar alinhados para que o SI alcance seu objetivo [1].

[1] destaca alguns tipos de SI. Os Sistemas de Informação Transacionais (SPT) armazenam e recuperam dados, além de realizar cálculos sobre eles. O Sistema Especialista automatiza o processo de decisão através de algoritmos e regras que determinarão as ações a serem executadas. Já os Sistemas de Gestão Eletrônica de Documentos (GED) armazenam e recuperam documentos, utilizando bancos de dados.

1) *Sistemas de Informação Acadêmicos e e-Government*: A transformação digital nas instituições de ensino superior tem impulsionado o desenvolvimento de Sistemas de Informação voltados à gestão acadêmica [2], [3]. Esses sistemas abrangem desde matrículas e controle de frequência até processos seletivos e emissão de diplomas. No contexto do *e-government*, a oferta de serviços públicos em ambiente digital reduz custos operacionais, amplia o acesso da população e promove maior transparência nos processos institucionais [4], [5]. A digitalização de processos seletivos em instituições públicas de ensino se insere nesse movimento, representando um avanço na prestação de serviços à comunidade acadêmica.

2) *Sistema para Processo Seletivo*: O processo seletivo é um evento organizado a fim de selecionar candidatos para preencher vagas para um posto de trabalho em uma organização ou para um curso em uma instituição de ensino [6]. Um sistema para gerenciamento do processo seletivo para ingresso em um curso deve amparar o andamento do processo e toda a sua evolução a fim de facilitar a execução de cada etapa tanto para a equipe organizadora quanto para os inscritos. Analisando os processos da seleção de pós-graduação realizada pelo IFBA, foi possível separar as tarefas executadas em dois grupos: as dos alunos e as dos avaliadores.

Aos alunos cabe encontrar e ler o edital, realizar o processo de inscrição preenchendo todos os dados dos formulários e organizar os documentos do currículo. Além disso, acompanhar a divulgação de resultados, realizar as etapas avaliativas como provas e entrevistas e, em casos pertinentes, interpor recursos desses processos.

Para os avaliadores, fica a responsabilidade de elaborar e divulgar o edital, formulários e possíveis correções, gerenciar as etapas do processo e suas datas, analisar e validar as inscrições dos alunos. Precisam também organizar as avaliações e realizar correções, divulgar resultados e receber pedidos de recurso, avaliar o currículo separando e validando documentos comprobatórios enviados pelos alunos e, de acordo com o barema do edital, pontuar, realizar cortes e classificar as inscrições com base nas vagas ofertadas.

Logo, um sistema web para o processo seletivo teria como objetivo permitir que as tarefas supracitadas sejam realizadas de forma remota, organizando os dados envolvidos para facilitar o bom andamento da seleção. Para isso, deve ser capaz de tornar formulários digitais, armazenar em nuvem os documentos e disponibilizá-los de forma estruturada. Também, divulgar resultados de forma rápida e eficiente, dispor de ferramentas para entrada e análise de recursos, bem como auxiliar as avaliações feitas e sua pontuação e, por fim, realizar a classificação final dos alunos.

No âmbito do projeto, esse tipo de sistema possui características de um SPT, pois deve armazenar os dados de inscrição e do processo seletivo. Por possuir regras de negócio específicas, se encaixa em um Sistema Especialista que utiliza determinados preceitos para classificar os candidatos, resultando na decisão pronta para o usuário. E pode ser considerado um GED, já que armazenará documentos inseridos pelos candidatos e os disponibilizará para a avaliação pela equipe

organizadora.

### B. *Continuous Integration e Continuous Delivery (CI/CD)*

CI (*Continuous Integration* - Integração Contínua) se refere a um conjunto de práticas de desenvolvimento onde todo o código de uma aplicação deve ser mantido em um repositório compartilhado. No momento em que o desenvolvedor solicitar a incorporação do seu código ao código do repositório, *scripts* são acionados a fim de selecionar o código mais recente do repositório, integrá-lo com o novo código e executar testes automatizados predefinidos. Se o código resultante não passar em todos os casos de teste, a fusão é rejeitada [7].

CD pode ser definido como *Continuous Delivery* (Entrega Contínua) ou *Continuous Deployment* (Implantação Contínua). A Entrega Contínua corresponde à compilação das alterações feitas pelo desenvolvedor, garantindo que o código esteja sempre pronto para uma possível implantação em um ambiente de produção [8]. Porém, neste caso, ainda seria necessária uma intervenção manual para a implantação efetiva. No conceito de Implantação Contínua, a publicação das alterações é feita em um ambiente de produção de forma automática após um CI bem-sucedido [7].

Por muitas vezes, o desenvolvimento de *software* requer mudanças rápidas de requisitos dos clientes e restrições quanto ao tempo de lançamento. CI/CD é um aliado nesta esfera para automatizar a implantação sempre que houver uma nova atualização de código e fazer com que os novos recursos cheguem de forma mais rápida, e com menos riscos ao usuário final [9].

### C. *Segurança da Informação e LGPD*

Sistemas que manipulam dados pessoais de candidatos, como documentos de identificação, informações acadêmicas e dados sensíveis relacionados a ações afirmativas e condições de saúde, devem observar os princípios da Lei Geral de Proteção de Dados (LGPD – Lei nº 13.709/2018) [10]. A LGPD estabelece regras sobre coleta, armazenamento, tratamento e compartilhamento de dados pessoais, exigindo das organizações o estabelecimento de políticas de privacidade, controle de acesso, consentimento informado e notificação em caso de incidentes. No contexto de sistemas web para processos seletivos institucionais, a adequação à LGPD é requisito indispensável, especialmente diante do tratamento de dados sensíveis como deficiências, raça/etnia e situação socioeconômica dos candidatos.

## III. TRABALHOS RELACIONADOS

Esta seção apresenta trabalhos e sistemas relacionados ao gerenciamento de processos seletivos, contextualizando a proposta em relação às soluções existentes na literatura e no mercado.

A AIX Sistemas<sup>1</sup> é uma empresa brasileira que desenvolve soluções tecnológicas para a administração de cursos, escolas e faculdades. Entre as soluções está o *software* para processo seletivo que proporciona a organização e gerenciamento de

<sup>1</sup><https://aix.com.br>

vestibular, exame de seleção ou concurso público através do próprio site da instituição, a inscrição manual ou via web, com emissão automática do boleto bancário, e a integração com um módulo acadêmico para aproveitamento de dados dos aprovados [11].

A Universidade Federal de Santa Catarina (UFSC) utiliza um *software* próprio em Java para gerenciar seus processos de ingresso em pós-graduações. O CAPG<sup>2</sup> foi criado em meados de 2010 para atender inscrições no Programa de Especialização em Saúde da Família e posteriormente foi adotado por outros programas. Em 2018 foi feito um novo sistema inspirado no primeiro, tendo como um dos objetivos a melhoria da gestão dos dados dos processos de inscrição [12].

Outras universidades, como as Universidades Federais da Bahia (UFBA) e do Maranhão (UFMA), utilizam o SIGAA<sup>3</sup>, porém, assim como em [12] não se tem acesso a muitas informações sobre a possibilidade de implantá-los. Já o sistema da AIX é uma solução paga, o que não atende à necessidade do trabalho.

Além disso, [13] descreve o desenvolvimento de um sistema web, o sgPIP, para gerenciar o Programa Institucional de Permanência Acadêmica (PIP) da Universidade Federal Rural do Semi-Árido (UFERSA). O sistema visa otimizar o processo de seleção de bolsistas, tradicionalmente presencial e moroso, virtualizando as etapas e reduzindo redundâncias. Tecnologias como ReactJS, NodeJS e MongoDB foram empregadas, utilizando a metodologia ágil Scrum para o gerenciamento do projeto. O sgPIP permite que coordenadores gerenciem processos seletivos e avaliem candidaturas, enquanto os alunos podem submeter documentos online e acompanhar o status de suas inscrições. O objetivo é modernizar a gestão de bolsas, facilitando o acesso e a organização da documentação. A aplicação integra diversas tecnologias de desenvolvimento, incluindo a possibilidade de o coordenador cadastrar um processo como indeferido e descrever o motivo.

No contexto mais amplo de transformação digital em instituições de ensino, [4] discute os fundamentos do *e-government* e seus desdobramentos para sistemas acadêmicos públicos, ressaltando a importância de soluções abertas, acessíveis e aderentes a normas de privacidade. A LGPD representa um marco regulatório relevante nesse cenário, especialmente para sistemas que tratam dados sensíveis de candidatos [10]. A escassez de soluções *open-source* que contemplem as especificidades de processos seletivos de pós-graduação, incluindo avaliação curricular por barema, suporte a ações afirmativas e interposição de recursos, reforça a relevância da proposta apresentada neste trabalho.

Ambos os sistemas visam a digitalização e otimização de processos seletivos, mas o fazem com abordagens e tecnologias diferentes. Nossa proposta utiliza um conjunto de ferramentas mais moderno e integradas (Next.js, Vercel, Firebase) com foco em escalabilidade e facilidade de uso, enquanto o

<sup>2</sup><https://capg.sistemas.ufsc.br/>

<sup>3</sup><https://sigaa.ufba.br/>

sgPIP utiliza uma abordagem mais tradicional (React, Node.js, MongoDB) com foco em funcionalidades específicas para o gerenciamento do programa PIP. A Tabela I apresenta a comparação entre os sistemas.

Tabela I  
COMPARAÇÃO ENTRE A PROPOSTA E O SGPIP

	Proposta	sgPIP
Front-end	Next.js, React	ReactJS, Bootstrap
Back-end	Next.js API	Node.js
Banco de Dados	Firebase (NoSQL)	MongoDB
Publicação	Vercel	Não especificado
Autenticação	JWT	Não especificado
Open-source	Sim (GPL 3.0)	Não especificado
Avaliação Curricular	Sim (barema)	Não
Interposição de Recursos	Sim	Não

#### IV. PROPOSTA

A seguir, são descritas as tecnologias escolhidas para a implementação do projeto e como elas ampararam o desenvolvimento do *frontend* e da API, o armazenamento de dados e arquivos e a autenticação. São também discutidos os *trade-offs* das escolhas tecnológicas. E mostra-se o mapeamento definido para o banco de dados.

##### A. Infraestrutura e Tecnologias

O sistema foi projetado utilizando o Next.js para o desenvolvimento em conjunto com a Vercel para a publicação e o Firebase para armazenamento de dados e arquivos. Levou-se em consideração a qualidade e confiabilidade dessas ferramentas, além do fato de certificar-se de que suas cotas gratuitas de utilização seriam suficientes para o escopo do projeto. A Figura 1 apresenta a infraestrutura utilizada no projeto.

A Tabela II sintetiza os principais *trade-offs* das tecnologias adotadas em relação a alternativas possíveis, evidenciando as razões das escolhas e suas limitações.

1) *SSR com Next.js*: No início do desenvolvimento Web os servidores eram responsáveis por todas as partes de uma aplicação, incluindo banco de dados, páginas (estáticas), *template engine* e *assets* (arquivos CSS e imagens) resultando em um aplicativo monolítico. Por este ser um padrão que não escala de forma satisfatória, atualmente, a arquitetura cliente-servidor é a mais difundida na implementação de aplicações web, onde os aplicativos do lado do servidor lidam com dados e os aplicativos do lado do cliente lidam com a apresentação desses dados [14].

Em uma abordagem tradicional, interações do usuário com a página resultam na transição para uma outra página, isso requer mais largura de banda e mais tempo de espera a cada ação. Já com o desenvolvimento de uma aplicação de página única (*Single Page Application* - SPA) os componentes como arquivos CSS, imagens e *scripts* são carregados de uma só vez no momento que a aplicação é acessada [15] e toda a renderização e navegação da página acontecem

puramente no cliente [14]. Nessa abordagem, o cliente não precisa fazer *download* e renderizar novamente a página inteira para trocar apenas uma parte do conteúdo. Porém, neste caso, o carregamento inicial pode se prolongar e fazer com que usuários desistam do acesso e que mecanismos de busca deem classificações mais baixas à aplicação [14].

Como alternativa existe a Renderização do lado do servidor (*Server Side Rendering* - SSR), onde é possível renderizar a marcação inicial no servidor e em um segundo momento, o aplicativo JavaScript assumir o controle [14] possibilitando atualizar aquela parte ou região específica de uma página em vez de recarregar uma página inteira [15]. Considerando que o renderizador do lado do servidor geralmente está muito mais próximo do servidor de Interface de Programação de Aplicativos (*Application Programming Interface* - API), este terá uma conexão muito melhor e muito mais largura de banda do que os clientes remotos (navegadores) [14].

O Next.js<sup>4</sup> se destaca na abordagem SSR oferecendo uma possibilidade de criação de uma API acoplada, fácil instalação, zero configuração e uma grande comunidade [14]. O Next.js é um *framework* com uma estrutura que permite a construção de aplicativos *frontend* React<sup>5</sup> e lida de forma transparente com a renderização do lado do servidor.

O React é uma biblioteca Javascript<sup>6</sup> para construir interfaces de usuário [16] mantido pelo Facebook, eleita a estrutura da web mais comumente usada em 2021 [17]. Com ele é possível projetar visualizações simples para cada estado, que terão seus componentes atualizados quando os dados forem alterados. Porém, aplicativos que o tem como base, normalmente carregam o conteúdo depois que o Javascript é carregado, resultando em SEO (*Search Engine Optimization* - Otimização para mecanismos de busca) e experiência do cliente desfavoráveis [16]. O Next.js soluciona esses problemas quando fornece suporte à SSR.

Com a plataforma em nuvem Vercel, é possível implantar um site Next.js altamente disponível e pronto para produção, com ativos estáticos sendo servidos por meio do CDN (*Content Delivery Network* - Rede de Distribuição de Conteúdo) e suporte integrado para a otimização estática automática do Next.js e para rotas de API. Com CI/CD integrado, esta ferramenta permite a hospedagem de sites e serviços da Web que são implantados instantaneamente, escalonados automaticamente e não requerem supervisão [18]. O código-fonte do projeto pode ser gerenciado a partir das plataformas de versionamento GitHub, GitLab ou Bitbucket. A cada *push* para a *branch* principal ou as secundárias do projeto, uma nova versão é implantada pela Vercel e pode ser acessada pela URL de produção associada. O acompanhamento da publicação pode ser feito através de um painel do projeto no site da Vercel.

2) *Firebase*: O Firebase<sup>7</sup> é uma plataforma criada pela Google que auxilia no desenvolvimento de aplicações através da infraestrutura que oferece [19]. Entre os seus diversos serviços,

<sup>4</sup><https://nextjs.org/>

<sup>5</sup><https://reactjs.org/>

<sup>6</sup><https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

<sup>7</sup><https://firebase.google.com/>

Tabela II  
Trade-offs DAS PRINCIPAIS TECNOLOGIAS ADOTADAS

Tecnologia	Vantagens	Alternativas	Limitações
Next.js	SSR nativo, API integrada, zero config	NestJS, Django, Laravel	Acoplamento à Vercel; migração complexa
Firebase	BaaS completo, cota gratuita, escalável	PostgreSQL, MongoDB	Vendor lock-in; limites de leitura/escrita
Vercel	CI/CD automático, CDN global	AWS, Railway, Render	Limite de funções serverless no plano free
JWT	Stateless, amplamente suportado	OAuth2, Sessions	Revogação requer infraestrutura adicional

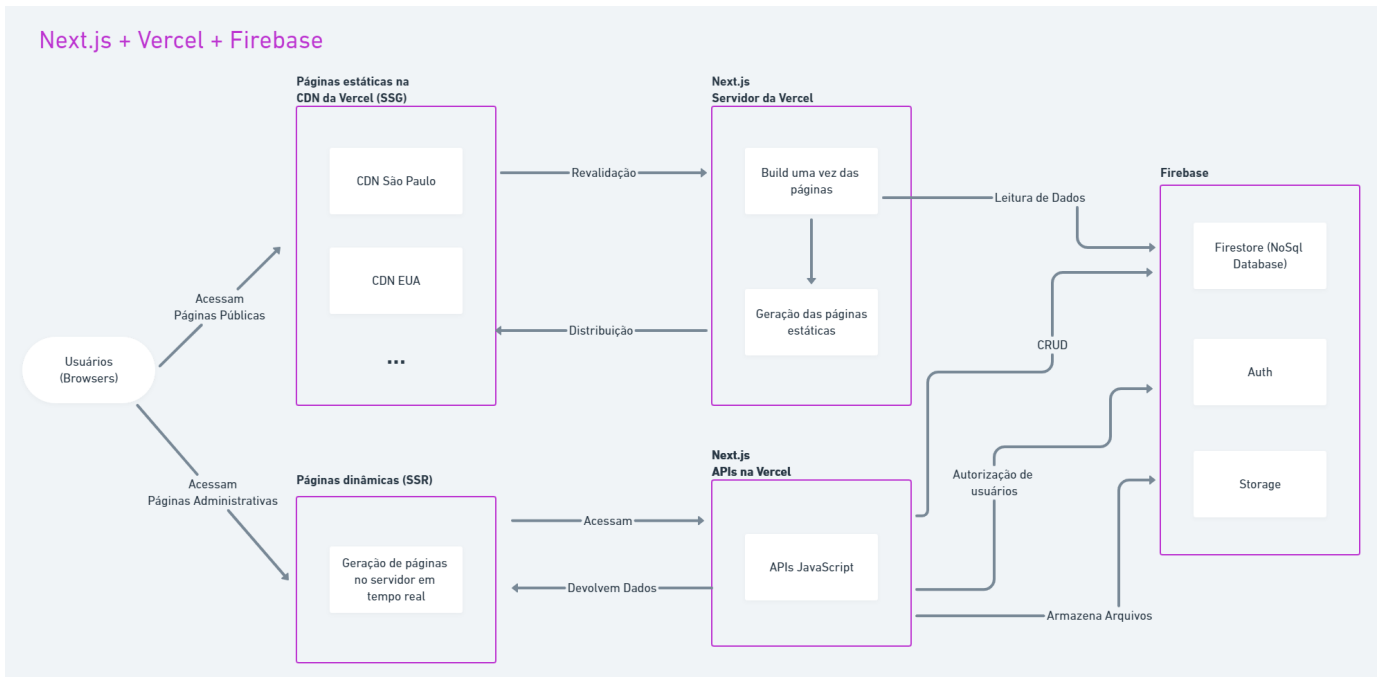


Figura 1. Infraestrutura

o Firebase disponibiliza o *Firebase Authentication*, serviço de autenticação que permite que os usuários se cadastrem, efetuem login de forma segura, sejam identificados e recuperem sua senha [20]. Com isso, passa-se a responsabilidade de desenvolver por conta própria um processo complexo como o de autenticação para uma plataforma consistente como o Firebase.

O *Cloud Firestore* fornece o serviço de banco de dados de nuvem *NoSQL* flexível e escalável a fim de armazenar e sincronizar dados. Seu modelo de dados disponibiliza uma estrutura de armazenamento em documentos, organizados em coleções. É possível realizar consultas para recuperar documentos individuais e específicos ou recuperar todos os documentos em uma coleção que corresponda aos parâmetros da consulta, além de permitir a utilização de filtros. As consultas são indexadas por padrão, desta forma, o desempenho da consulta é proporcional ao tamanho do conjunto de resultados, e não ao conjunto de dados [21].

O *Cloud Storage* oferece o armazenamento de arquivos que permite o envio (*upload*) e *download* de arquivos através de SDK (*Software Development Kit*) de clientes. Este serviço é altamente escalável e pode ser dimensionado para exabytes. O acesso tanto às coleções quanto aos arquivos podem ser

configurados através das regras de segurança do *Firebase Authentication* [20].

3) *Aplicação*: Para dar suporte à aplicação, foi criado um projeto no Firebase que será responsável por armazenar o *pool* de usuários, os dados e os arquivos. Já o projeto da Vercel irá amparar a criação da aplicação Next.js com React para abranger o *frontend* e a API. A interação entre a aplicação em Next.js e o Firebase se dá através do SDK fornecido, uma biblioteca que pode ser instalada no projeto React, e da configuração das credenciais disponibilizadas pelo Firebase nas variáveis de ambiente do projeto.

As rotas da aplicação, tanto no âmbito das páginas quanto dos *endpoints* de API são habilitadas pelo próprio sistema de arquivos do projeto React. Desta forma, todos os arquivos Javascript ou Typescript existentes no diretório */pages*, obrigatório na estrutura de um projeto Next.js, são tratados automaticamente como uma rota. No caso das rotas de API, os arquivos devem estar no diretório */pages/api* [22]. Essas rotas poderão receber requisições internas, do próprio projeto, e também de clientes externos se for necessário, possibilitando que serviços de terceiros consultem e integrem com a API, por exemplo.

4) *Banco de Dados*: Utilizando o conceito de banco de dados *NoSQL* foram definidas as coleções *User*, para os dados do usuário, *Course* para os dados do curso, *Teacher* para informações dos docentes, *News* para os posts e *Work* para os trabalhos como pode ser visto na Figura 2.

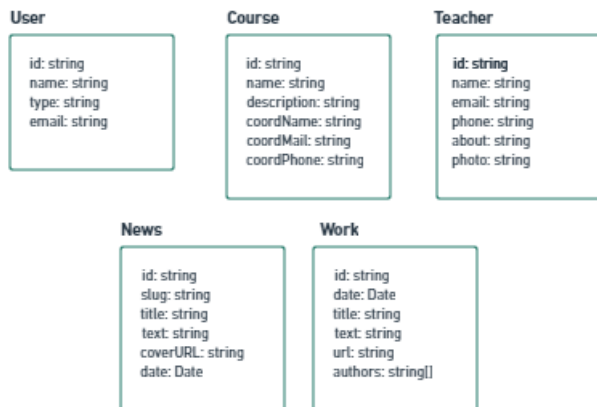


Figura 2. Coleções User, Course, Teacher, News e Work

Na Figura 3 é apresentada a coleção *SelectiveProcess* que representa o processo seletivo e contém os atributos de coleção *ProcessNotice*, para armazenar os editais, *ProcessForm* para armazenar os formulários, *ReservedPlace* para as vagas, *Step* para as etapas e *BaremaCategory* para as categorias do barema. O *BaremaCategory* possui o *Subcategory* para os itens da categoria.

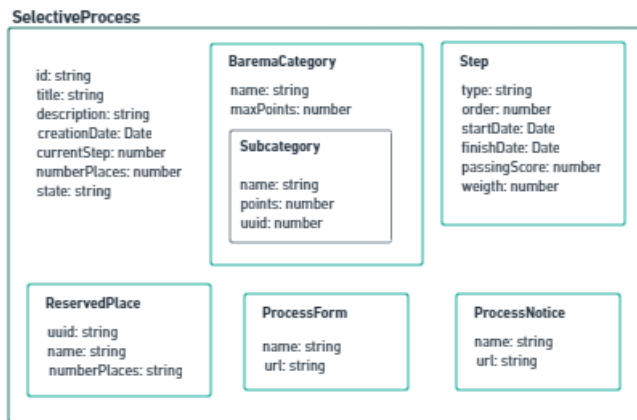


Figura 3. Coleção SelectiveProcess

Na Figura 4 é apresentada a coleção *Subscription* que é o modelo da Inscrição, responsável por armazenar dados pessoais e profissionais do inscrito, dados relacionados à vaga escolhida, além de campos para identificação (ID) do *SelectiveProcess* e do *User*. *Subscription* contém os campos de coleção *ProcessForm* para armazenar os formulários preenchidos, *FileCategory*, para os arqui-

vos relacionados ao *Subcategory* do processo seletivo e *Resource* para os possíveis recursos interpostos.

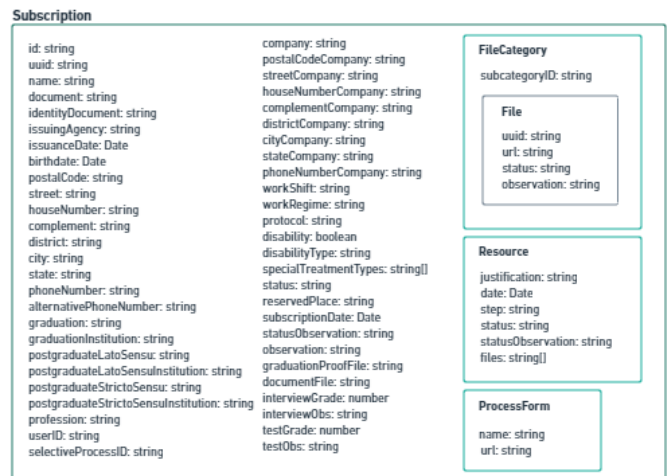


Figura 4. Coleção Subscription

### B. Prova de Conceito

Para atender à demanda do processo seletivo, o sistema foi desenvolvido em três partes. A primeira parte é o **ambiente público** onde se tem acesso às informações sobre o curso e seu corpo docente, às publicações feitas para divulgar notícias referentes ao curso e as etapas do processo seletivo em andamento e aos trabalhos de conclusão de curso e artigos feitos pelos alunos em conjunto com o corpo docente. Os dados da área não logada são inseridos no sistema pelo **ambiente administrativo** (Figura 5). Os usuários que acessam esse ambiente são os que possuem a permissão *Master* ou *Administrador*. A diferença entre eles está no fato do usuário *Master* estar apto a cadastrar outros usuários com permissão *Master* ou *Administrador*. Ambos possuem liberação para cadastrar um processo seletivo e gerenciá-lo, cadastrar os dados do curso, os docentes, as notícias e os trabalhos. O **ambiente de inscrição** é a parte que os usuários que desejam participar do processo seletivo acessam para se inscrever, acompanhar sua inscrição e cadastrar recursos, se necessário.

1) *Autenticação*: Para acessar os ambientes administrativo e de inscrição é necessário ter um cadastro de usuário na plataforma. No caso do ambiente administrativo, o primeiro usuário deve ser cadastrado diretamente no projeto no *Firebase*, onde será definido o e-mail e a senha de acesso no *Firebase Authentication* e será incluído um documento relacionado a esse usuário na coleção *User* no *Firestore Database*. Este documento deve ter o mesmo ID e e-mail gerados para o usuário, o nome e o tipo *Master*. Com este usuário será possível efetuar login na plataforma e cadastrar os novos usuários.

O usuário que pretende participar do processo seletivo deve acessar a página de login e efetuar o seu cadastro. Para isso, deve informar seu nome completo, e-mail e senha; neste momento seus dados são enviados à API. Esta por sua vez

Oi, Leticia Porto Soares! Perfil Sair

**Processo Seletivo 2022.1**

Etapa Atual: **Interposição de Recursos da Prova** (14/10/2021 -- 14/10/2021)

Inscritos que entraram com recurso

Nome	Data de Inscrição	Vaga	Parecer do Recurso
Quantidade de Inscritos: 0			

Candidatos Interposição de Recursos da Prova

Nome	Vaga	Parecer	Pontuação (0-100)	Observações	Ver Dados
Leticia Porto Soares	Servidores do IFBA	✓ Deferida	-		<a href="#">Acessar</a>

Figura 5. Página inicial do ambiente Administrativo

será responsável por se comunicar com o Firebase através do SDK e das configurações efetuadas, para salvar os dados e o usuário poderá ter acesso ao ambiente de inscrição.

Ao efetuar o login na aplicação é gerado junto ao Firebase Authentication um *JSON Web Token* (JWT) com duração de uma hora que será armazenado nos *cookies* do navegador. Este *token* deverá ser enviado nos *headers* de todas as requisições feitas à API na área logada com o nome *Authorization*. Na API verifica-se se o conteúdo enviado no *Authorization* é válido, caso contrário retorna-se um erro com código 401 (não autorizado) e o usuário é redirecionado para a página de login.

No âmbito do *frontend*, para analisar se o usuário tem permissão para acessar determinada página, criou-se uma verificação na função *getServerSideProps* de cada página (Figura 6). É importante destacar que o conteúdo implementado dentro desta função é processado no servidor, o que protege as credenciais do Firebase. Esta verificação acessa um *endpoint* para recuperar o usuário atual enviando o *token* armazenado nos *cookies* e valida se o usuário tem acesso permitido para aquela página (Figura 7), por exemplo, somente usuários do tipo *Master* ou Administrador poderão cadastrar processo seletivo e somente usuário do tipo Aluno poderá se inscrever.

O usuário pode recuperar a sua senha através da opção “Esqueci a senha” na tela de login. Ao informar seu endereço de e-mail, o mesmo receberá um link para redefinir a sua senha. Por fim, todos os tipos de usuários têm a possibilidade de acessar uma página de Perfil, onde podem editar a sua senha estando logado na aplicação.

2) *Cadastros Básicos*: Para o Curso são informados o nome, a descrição e os dados do coordenador atual. Para o cadastro de um Docente são inseridos nome, dados de contato e uma foto do mesmo. Na inserção de um Trabalho acadêmico informa-se o seu título, os autores, a data do trabalho e o arquivo correspondente. Uma publicação (*Post*) pode ser criada

manualmente através do acesso ao menu onde informa-se o título, um arquivo de capa e o texto ou no gerenciamento do processo seletivo após a divulgação do resultado definitivo de uma etapa. No último caso o sistema mostra um *Post* pré-preenchido na tela e o usuário pode editar antes de efetuar a publicação.

Um usuário do tipo *Master* tem permissão para cadastrar outros usuários. Desta forma, são inseridos o nome, o e-mail, a senha e o tipo do usuário. Porém, na edição, não é possível atualizar a senha, fazendo-se necessário acessar o Perfil ou a opção Esqueci a senha para isso.

3) *Cadastro do Processo Seletivo*: Para o cadastro do processo seletivo é necessário informar inicialmente uma descrição da página inicial do ambiente administrativo. Outras informações são inseridas na própria página do processo seletivo de forma agrupada. Na parte das vagas é definido o total de vagas do processo e como elas serão distribuídas em seus determinados grupos. O grupo de ampla concorrência deve possuir pelo menos uma vaga, as outras devem ser divididas entre outros grupos a serem definidos pelo usuário, como por exemplo, Servidores públicos, pessoas com deficiência e políticas de ações afirmativas.

Na parte dos documentos deve ser feito o *upload* de arquivos relacionados ao processo seletivo. Os tipos de arquivo são Edital para cadastro das versões que o Edital do processo seletivo venha a ter, e Formulários, arquivos que deverão ser baixados, preenchidos e cadastrados novamente na plataforma no momento da inscrição.

Uma das etapas do processo seletivo é a Avaliação Curricular, por isso existe um barema para que o inscrito forneça documentos comprobatórios de atividades realizadas que podem contar pontos a serem somados à sua nota final no processo. As categorias e subcategorias do barema podem ser dinâmicas e são cadastradas no sistema de forma que uma categoria possui quantidade máxima de pontos aceitos e as suas subcategorias

```

TS index.tsx X
pages > student > subscription > TS index.tsx > SubscriptionLayout
1586 export const getServerSideProps = async (ctx: GetServerSidePropsContext) => {
1587     const permission = Permission();
1588     return await permission.checkPermission(ctx, [UserType.STUDENT]);
1589 };
1590

```

Figura 6. Método getServerSideProps da página de inscrição

```

TS permission.service.ts M X
lib > TS permission.service.ts > Permission > redirectTo
67 async function checkPermission(ctx: GetServerSidePropsContext, typesAllowed: String[]) {
68     try {
69         await checkToken(ctx);
70         const user: User = await getCurrentUser(ctx);
71
72         if(typesAllowed.includes(user.type as UserType)){
73             return await buildReturnMessage(Message.AUTHORIZED);
74         }
75
76         return redirectTo("/error/not-authorized");
77     } catch (err) {
78         return redirectTo("/login");
79     }
80 }

```

Figura 7. Método de checagem da permissão

possuem a informação de quantos pontos cada comprovação de realização da atividade referida vale.

Por fim, são definidas as etapas do processo seletivo com suas datas de início e fim e informações sobre nota de corte e peso para atividades que permitam essas informações. Os grupos de etapa principais disponíveis para cadastro são Inscrição, Entrevista e/ou Prova, Avaliação Curricular e Resultado Definitivo. Cada uma dessas etapas, exceto Resultado Definitivo, possui três etapas subsequentes que são o Resultado Preliminar, que disponibiliza o resultado inicial da avaliação da etapa referida, a Interposição de Recursos, período em que os inscritos podem reclamar o seu resultado e o Resultado Definitivo, resultado após a análise dos recursos. Na Figura 8 é demonstrado um exemplo de fluxo para as etapas do processo.

Uma nota de corte é definida para etapas que são eliminatórias como é o caso da Prova e da Entrevista, e um peso é definido para etapas em que o seu resultado faz parte da avaliação final, como é o caso da Prova, da Entrevista e da Avaliação Curricular.

4) *Inscrição*: No momento da inscrição são informados os dados pessoais do inscrito, como números de documentos, telefone e endereço, e os dados acadêmicos, com curso e instituição onde foi cursada a graduação, e pós-graduações, se houverem. Caso o inscrito esteja atuando profissionalmente no momento, pode informar os dados profissionais, como cargo, nome da empresa e endereço. Em relação às vagas, o

usuário deve informar a qual grupo de vagas deseja concorrer, de acordo com o que foi definido no cadastro do processo seletivo. Além disso, deve informar se é portador de alguma deficiência, tendo opção de descrever o tipo de deficiência e se necessita de um ou mais atendimentos especiais. As opções de atendimento são Prova em Braille, Auxílio de Leitor/Ledor, Intérprete de Libras, Sala de Mais Acesso, Auxílio para Transcrição e Mesa e Cadeiras sem Braço.

Na última etapa da inscrição o usuário deve efetuar o *upload* dos documentos. Os primeiros documentos solicitados são os obrigatórios: documento pessoal com foto e diploma de graduação na área de computação. Posteriormente são listadas as categorias e subcategorias do barema, onde o inscrito pode carregar um ou mais arquivos em cada subcategoria. Estes arquivos serão contabilizados em sua pontuação final. Existe ainda a listagem dos formulários cadastrados no processo seletivo. Estes devem ser baixados, preenchidos e carregados na plataforma. Ao finalizar a inscrição o usuário terá a opção de visualizá-la e verificar o status tanto da inscrição quanto dos arquivos de barema carregados.

5) *Gerenciamento do Processo Seletivo*: O usuário administrativo terá acesso a um painel que se comporta de maneira a facilitar o trabalho dos avaliadores, adaptando-se à etapa em que o processo se encontra. No momento da inscrição, o sistema apresenta a lista de inscritos. Desta forma, o avaliador pode selecionar as inscrições individualmente para ter acesso

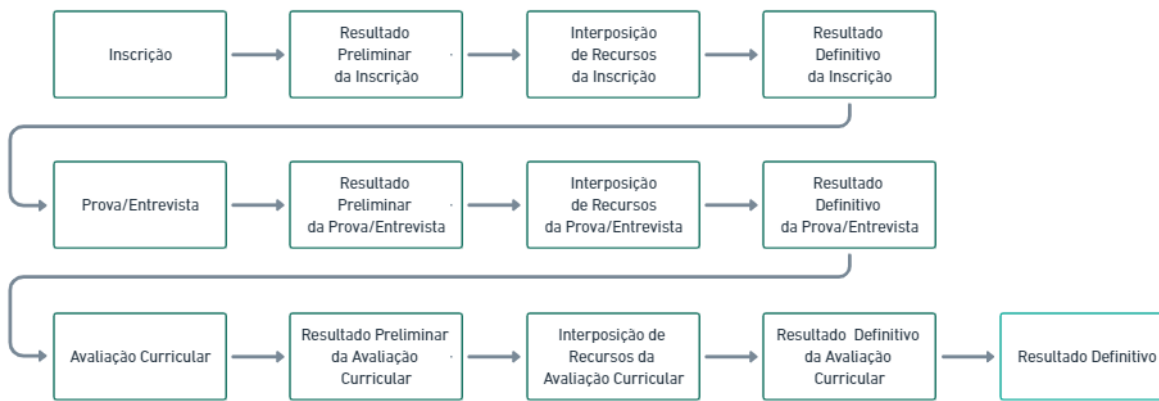


Figura 8. Exemplo de etapas do processo seletivo

ao formulário preenchido e a todos os dados enviados, podendo então, deferir ou indeferir de acordo com os critérios do edital.

Em etapas avaliativas como a de Prova e Entrevista, é definida a nota obtida por cada inscrito. Para a avaliação do barema, o avaliador deve acessar um inscrito e se dirigir à área dos arquivos enviados por ele. O sistema organiza e cataloga os arquivos já nas subcategorias e categorias do barema. O avaliador acessa um arquivo, avalia e faz o deferimento dele. Os arquivos aceitos têm sua pontuação somada à nota final, considerando a pontuação máxima da categoria.

Em etapas de interposição de recursos, são geradas listas com os inscritos que os interpuseram, para que o usuário administrador possa deferi-los ou indeferi-los. Para os resultados finais, o sistema separa os participantes pelas vagas pretendidas e ordena de acordo com as notas das etapas avaliativas e seus pesos, calculando a nota final e sua classificação.

O usuário pode avançar manualmente para as etapas seguintes e disponibilizar seus resultados conforme elas são concluídas. Para isso o sistema gera PDFs com informações sobre o deferimento das inscrições e interposição de recursos e pontuação das avaliações, além do resultado final. Além disso, cria-se um post automaticamente com um texto padrão que pode ser editado e com o arquivo PDF gerado em anexo. O resultado ficará disponível na parte pública e no ambiente do inscrito. Para visualizar os resultados de etapas anteriores acessa-se o menu Resultados.

6) *Recurso*: A partir das etapas de Interposição de Recurso, o inscrito tem acesso ao menu de Recursos onde tem a possibilidade de reivindicar o resultado de alguma etapa que não esteja em concordância. Nesta funcionalidade é possível inserir um texto, onde o inscrito irá justificar o seu recurso, e um ou mais arquivos para agregar a justificativa. Após o cadastro, o usuário pode visualizar o(s) seu(s) recurso(s) e acompanhar o status dele(s).

### C. Lições Aprendidas

Durante o desenvolvimento do projeto identificou-se as seguintes lições aprendidas:

- **Necessidade de Flexibilidade na Modelagem do Processo Seletivo**: Durante o desenvolvimento, percebeu-se que cada programa de pós-graduação pode ter regras específicas para seu processo seletivo. Isso exigiu a criação de um sistema parametrizável, permitindo que novas regras fossem facilmente configuradas sem necessidade de mudanças no código.
- **Desafios da Implementação de um Banco de Dados**: A escolha do Firestore trouxe vantagens como escalabilidade e facilidade de integração, mas também desafios, como a modelagem de dados em coleções e subcoleções para garantir eficiência na recuperação e processamento das informações.
- **Importância da Experiência do Usuário (UX) na Automação de Processos**: A digitalização do processo seletivo reduziu burocracias, mas evidenciou a necessidade de uma interface intuitiva. Inscritos e avaliadores tinham dificuldades iniciais em encontrar funcionalidades específicas, o que levou a refinamentos na navegação e layout.
- **Benefícios da Arquitetura SSR com Next.js**: O uso de SSR melhorou o tempo de carregamento e a performance, mas exigiu otimizações para balancear desempenho e interatividade na interface.
- **Integração Contínua (CI/CD) Como Facilitador da Evolução do Sistema**: A adoção de práticas de CI/CD acelerou o ciclo de desenvolvimento, permitindo testes contínuos e garantindo estabilidade do sistema ao longo de suas atualizações.
- **Gestão de Recursos e Sustentabilidade da Solução**: A escolha de tecnologias com planos gratuitos foi fundamental para garantir a viabilidade do projeto. No entanto, a limitação de cotas gratuitas do Firebase exigiu estratégias para otimizar armazenamento e minimizar custos futuros.
- **Desafios na Validação de Documentos Digitais**: Com a digitalização da inscrição, tornou-se necessário definir procedimentos claros para verificação da autenticidade dos documentos enviados, garantindo segurança e con-

fiabilidade nas avaliações.

- Expansão Potencial para Outros Processos Seletivos: Embora o sistema tenha sido desenvolvido inicialmente para um curso específico, identificou-se que sua estrutura permite adaptações para outros cursos e até para processos seletivos de graduação e concursos internos da instituição.
- Adoção de Código Aberto e Engajamento da Comunidade: Tornar o sistema *open-source* ampliou a possibilidade de colaboração, mas também exigiu uma documentação mais detalhada para facilitar a participação de outros desenvolvedores na evolução do projeto. O código-fonte está disponível sob a licença GPL 3.0, e a documentação de instalação e dependências está descrita no repositório do projeto.
- Importância da Segurança e Proteção de Dados: A manipulação de dados pessoais exigiu uma atenção especial à segurança da informação, incluindo autenticação, controle de acesso e conformidade com normas como a Lei Geral de Proteção de Dados (LGPD — Lei nº 13.709/2018). Dados sensíveis, como deficiências e pertencimento a grupos de ações afirmativas, demandam tratamento cuidadoso conforme o Art. 11 da LGPD, que exige base legal específica para dados sensíveis.

## V. RESULTADOS

O sistema desenvolvido foi avaliado em ambiente de produção por meio de uma auditoria automatizada com a ferramenta *Google Lighthouse* (versão 13.0.2), executada a URL de produção hospedada na plataforma Vercel (<https://pos-web-ifba.vercel.app/>). O *Lighthouse* é uma ferramenta de código aberto, mantida pelo Google, amplamente utilizada para auditar a qualidade de páginas web nos quesitos de desempenho (*performance*), acessibilidade, boas práticas e *Search Engine Optimization* (SEO) [23].

### A. Pontuações Gerais

A Tabela III apresenta as quatro pontuações categóricas geradas pelo *Lighthouse*, em uma escala de 0 a 100.

Tabela III  
PONTUAÇÕES CATEGORIAIS DO RELATÓRIO LIGHTHOUSE

Categoria	Pontuação	Classificação
Desempenho ( <i>Performance</i> )	89	Boa
Acessibilidade ( <i>Accessibility</i> )	88	Boa
Boas Práticas ( <i>Best Practices</i> )	100	Excelente
SEO	91	Excelente

Os resultados evidenciam que o sistema atingiu pontuações elevadas nas quatro dimensões, com destaque para a categoria *Best Practices*, que obteve pontuação máxima, refletindo a adoção de boas práticas de desenvolvimento web como o uso de HTTPS, ausência de erros no console do navegador, utilização de DOCTYPE HTML válido e codificação de caracteres corretamente declarada.

### B. Métricas de Desempenho

As métricas de desempenho mensuradas pelo *Lighthouse* são baseadas nas *Core Web Vitals* e em indicadores complementares que capturam diferentes aspectos da experiência de carregamento percebida pelo usuário. A Tabela IV detalha os valores obtidos.

A maior parte das métricas de desempenho apresentou resultados satisfatórios. O *First Contentful Paint* (FCP) de 0,9 s indica que o primeiro conteúdo visível é exibido rapidamente ao usuário, enquanto o *Largest Contentful Paint* (LCP) de 0,9 s demonstra que o maior elemento da página é renderizado em tempo igualmente rápido. O *Total Blocking Time* (TBT) de 0 ms é positivo, indicando ausência de tarefas longas que bloqueiem a *thread* principal durante o carregamento, o que se reflete em alta capacidade de resposta à interação do usuário. O *Cumulative Layout Shift* (CLS) de 0,01 confirma a estabilidade visual da página, com deslocamento de layout mínimo.

O único indicador com resultado insatisfatório foi o *Speed Index* (SpI) de 12,9 s, consideravelmente acima do valor de referência de 3,4 s. Essa métrica captura a velocidade com que o conteúdo visível é progressivamente preenchido durante o carregamento. A análise do relatório revelou que o valor elevado do SpI deve-se ao tempo de hidratação do Next.js (*before-hydration*: 21,4 s), inerente à arquitetura SSR com *hydration* do lado do cliente. Durante esse período, o conteúdo HTML inicial já está visível, mas a aplicação JavaScript ainda não assumiu o controle da página. Esse comportamento é uma característica da abordagem SSR adotada, conforme discutido na Seção IV, e não representa falha de usabilidade, uma vez que o FCP e o LCP permanecem baixos.

### C. Acessibilidade

A pontuação de acessibilidade de 88/100 indica boa conformidade com as diretrizes WCAG 2.1, avaliadas automaticamente pela biblioteca *axe-core* (versão 4.11.0). A Tabela V lista os problemas identificados automaticamente.

O problema mais grave é a utilização incorreta do atributo *aria-expanded* em um elemento `<label>`, que não permite esse atributo ARIA. Esse problema afeta tecnologias assistivas, como leitores de tela. Os problemas de contraste afetam a legibilidade para usuários com baixa visão, pois os textos de primeiro plano sobre o fundo verde claro não atingem a razão de contraste mínima de 3:1 (para texto grande em negrito) e 4,5:1 (para texto normal) conforme WCAG 2.1 nível AA. A hierarquia de cabeçalhos incorreta dificulta a navegação por teclado para usuários de tecnologia assistiva. Esses problemas são tecnicamente corrigíveis e foram mapeados como melhorias prioritárias para versões futuras da plataforma.verde-claro

### D. SEO e Boas Práticas

A pontuação de SEO de 91/100 indica boa configuração para indexação pelos mecanismos de busca. O único item não atendido foi a ausência da *meta description*, que permite que os buscadores exibam uma descrição relevante nos resultados de pesquisa. Todos os demais critérios foram atendidos: a

Tabela IV  
MÉTRICAS DE DESEMPENHO OBTIDAS NO RELATÓRIO LIGHTHOUSE

Métrica	Valor obtido	Referência (bom)	Classificação
<i>First Contentful Paint</i> (FCP)	0,9 s	≤ 1,8 s	Bom
<i>Largest Contentful Paint</i> (LCP)	0,9 s	≤ 2,5 s	Bom
<i>Total Blocking Time</i> (TBT)	0 ms	≤ 200 ms	Excelente
<i>Cumulative Layout Shift</i> (CLS)	0,01	≤ 0,1	Bom
<i>Speed Index</i> (SpI)	12,9 s	≤ 3,4 s	Fraco
<i>Time to Interactive</i> (TTI)	0,9 s	≤ 3,8 s	Bom

Tabela V  
PROBLEMAS DE ACESSIBILIDADE IDENTIFICADOS PELO LIGHTHOUSE

Problema	Elemento afetado	Severidade
Atributo ARIA não permitido ( <i>aria-expanded</i> em <code>&lt;label&gt;</code> )	Botão de navegação responsivo	Crítica
Contraste insuficiente de cores (razão 2,95 e 3,05)	Texto “IFBA” e botão “Saiba” Mais”	Séria
Hierarquia de cabeçalhos fora de ordem ( <code>h1</code> → <code>h3</code> → <code>h5</code> )	Seções de processos seletivos e notícias	Moderada

página possui título HTML, está habilitada para indexação, utiliza texto descritivo nos links, e os *anchors* são rastreáveis.

A pontuação máxima em *Best Practices* (100/100) comprova a adoção de práticas seguras e modernas: HTTPS ativo com certificado válido, ausência de cookies de terceiros, ausência de APIs depreciadas, sem erros no console do navegador, ausência de problemas de aspecto ou resolução de imagens, e utilização dos protocolos HTTP/2 para todos os recursos de primeira parte, o que melhora o desempenho de carregamento paralelo.

#### E. Discussão dos Resultados

Os resultados do relatório Lighthouse demonstram que o sistema atinge um nível de qualidade técnica adequado para uso em produção, especialmente considerando que foi desenvolvido com infraestrutura de custo zero. As pontuações obtidas (Desempenho: 89, Acessibilidade: 88, Boas Práticas: 100, SEO: 91) indicam que a arquitetura escolhida — Next.js com SSR, Firebase e Vercel — é capaz de entregar uma aplicação web com boa experiência de usuário.

A ausência de bloqueio da *thread* principal (TBT = 0 ms) e os tempos de FCP e LCP inferiores a 1 segundo são indicadores relevantes para um sistema de processo seletivo, cuja base de usuários inclui candidatos que podem estar acessando a plataforma por conexões mais lentas ou dispositivos mais limitados. O SpI elevado, embora tecnicamente desfavorável, não implica má experiência de uso real, dado que o conteúdo já está visível ao usuário bem antes do término da hidratação do JavaScript.

Os problemas de acessibilidade identificados são corrigíveis com ajustes pontuais no código e devem ser priorizados em versões futuras, especialmente em conformidade com a LGPD e com as diretrizes de acessibilidade digital previstas pelo Decreto nº 5.296/2004 e pela Lei Brasileira de Inclusão (Lei

nº 13.146/2015), que estabelecem requisitos de acessibilidade para sistemas de informação de instituições públicas.

#### F. Ameaças à Validade

Dentre as principais limitações do projeto, a necessidade de não gerar custos foi uma das mais desafiadoras. Embora as tecnologias usadas estejam entre o que existe de mais atual, a equipe ficou restrita a ferramentas com cotas e hospedagem gratuitas, além do domínio do projeto não ser personalizado. Em um cenário onde a quantidade de inscritos seja muito alta, podem ocorrer problemas com as cotas de leitura e escrita de documentos e principalmente com o servidor de arquivos.

A Vercel oferece um plano gratuito que atendeu bem à fase inicial do projeto. Entretanto, há limitações de uso, como restrição de tempo de execução de funções de API, armazenamento de logs e desempenho em alto tráfego. Em caso de necessidade de maior escalabilidade ou personalização, pode ser preciso migrar para um plano pago ou buscar outra infraestrutura, o que implicaria custos adicionais e ajustes técnicos.

A escolha do Next.js trouxe benefícios: SSR, otimização automática e facilidade na construção de APIs internas. No entanto, essa dependência limita a flexibilidade do projeto em cenários onde outras tecnologias ou arquiteturas possam ser mais adequadas. Caso seja necessário migrar para outra tecnologia, como um *framework* diferente ou uma arquitetura baseada em microsserviços, a reestruturação do código pode ser complexa e demandar grande esforço.

*Portabilidade e Alternativas de Hospedagem:* O sistema foi desenvolvido para funcionar de forma integrada à infraestrutura da Vercel, o que simplifica o processo de *deploy* e manutenção. No entanto, essa forte vinculação pode dificultar a migração para outras plataformas de hospedagem, como AWS, DigitalOcean ou servidores locais, sem ajustes significativos na configuração e na otimização do projeto.

*Atualizações e Mudanças nas Políticas das Ferramentas:* Tanto o Next.js quanto a Vercel são ferramentas mantidas por terceiros e estão sujeitas a atualizações, mudanças de políticas de preços e até mesmo descontinuação de funcionalidades gratuitas. Isso pode impactar o funcionamento da aplicação no futuro, exigindo adaptações não previstas inicialmente.

*Avaliação e Validação Empírica:* A ausência de uma avaliação empírica formal constitui uma ameaça à validade interna deste trabalho. Não foram conduzidos análises de desempenho com métricas quantitativas ou estudos comparativos controlados entre o processo manual e o digital. As conclusões

sobre automatização e aceleração do processo baseiam-se na observação qualitativa da prova de conceito e nos relatos dos membros da comissão organizadora. Trabalhos futuros devem incluir avaliações com instrumentos validados, como questionários SUS (*System Usability Scale*) ou similares.

Outra limitação foi a dificuldade de encontrar sistemas abertos e trabalhos publicados semelhantes para fins de comparação. Entender funcionalidades usadas pelo mercado, pontos fracos que podem ser melhorados e pontos fortes que devem ser replicados são importantes para o desenvolvimento do projeto. Ter múltiplas referências agregaria mais valor para a qualidade do sistema.

## VI. METODOLOGIA

Esta seção descreve a metodologia adotada no desenvolvimento do sistema, incluindo o levantamento de requisitos, as etapas do projeto e os critérios de validação utilizados.

### A. Classificação da Pesquisa

Este trabalho classifica-se como uma pesquisa aplicada de natureza tecnológica, com abordagem quali-quantitativa. Do ponto de vista dos objetivos, caracteriza-se como exploratório-descritiva, uma vez que investiga a viabilidade de desenvolvimento de um sistema de informação para gerenciamento de processos seletivos de pós-graduação. Em termos de procedimentos técnicos, constitui um estudo de caso com desenvolvimento de sistema, tendo o processo seletivo do PGDW como prova de conceito.

### B. Levantamento de Requisitos

O levantamento de requisitos foi conduzido por meio da análise documental dos editais e formulários dos processos seletivos realizados pelo PGDW em 2019 e 2024, em conjunto com entrevistas informais com membros da comissão organizadora. A partir dessa análise, os requisitos funcionais e não funcionais foram categorizados em dois grupos de atores: (i) **inscritos**, responsáveis pelo cadastro, envio de documentos, acompanhamento do processo e interposição de recursos; e (ii) **avaliadores/administradores**, responsáveis pela configuração do processo seletivo, gestão de etapas, análise de inscrições e divulgação de resultados. A Tabela VI resume os principais requisitos funcionais levantados.

Tabela VI  
PRINCIPAIS REQUISITOS FUNCIONAIS DO SISTEMA

Inscritos	Administradores
Realizar inscrição online	Cadastrar e configurar processo seletivo
Enviar documentos digitais	Gerenciar etapas e datas
Acompanhar status da inscrição	Analisar e deferir/indeferir inscrições
Interpor recursos	Avaliar documentos do barema
Visualizar resultados	Divulgar resultados e gerar PDFs

### C. Processo de Desenvolvimento

O desenvolvimento do sistema seguiu uma abordagem iterativa e incremental, com sprints quinzenais definidos de acordo com a disponibilidade da equipe. Cada iteração compreendeu as etapas de planejamento, implementação, revisão e testes. A adoção de práticas CI/CD via Vercel garantiu integração e implantação contínuas ao longo do desenvolvimento, possibilitando que novas funcionalidades fossem validadas em ambiente de produção de forma controlada.

### D. Validação e Prova de Conceito

A validação do sistema foi conduzida por meio de sua aplicação no processo seletivo do PGDW, funcionando como prova de conceito para avaliação da plataforma em um cenário real de uso. A análise contemplou aspectos funcionais e técnicos do sistema, incluindo a execução das etapas previstas no edital, a experiência de navegação e o desempenho da aplicação.

Os resultados indicaram que a plataforma executou integralmente as funcionalidades previstas para o processo seletivo, permitindo o gerenciamento das inscrições, submissão de documentos e acompanhamento das etapas pelos candidatos e pela comissão organizadora. Além disso, a análise técnica realizada com a ferramenta Google Lighthouse evidenciou desempenho satisfatório da aplicação, apresentando baixo tempo de bloqueio total (TBT de 0 ms), reduzido atraso potencial de entrada (*Max Potential First Input Delay* de 20 ms) e estabilidade visual adequada durante a navegação (CLS dentro dos parâmetros recomendados). Esses resultados demonstram boa responsividade e estabilidade da interface web.

Também foram observadas melhorias necessárias relacionadas à organização de alguns formulários e ajustes pontuais de navegação, especialmente em dispositivos com diferentes resoluções de tela. Ainda assim, os resultados obtidos indicam que a arquitetura proposta apresentou viabilidade técnica e operacional para utilização em processos seletivos acadêmicos.

Reconhece-se, entretanto, que a avaliação realizada possui caráter preliminar e exploratório. O estudo não contemplou testes extensivos de carga, avaliações quantitativas de usabilidade com instrumentos validados. Dessa forma, trabalhos futuros poderão ampliar a avaliação experimental, incorporando métricas quantitativas, estudos com maior número de usuários e testes de escalabilidade em cenários com maior volume de acessos.

## VII. CONCLUSÃO

A implementação de um sistema web para auxiliar o processo seletivo do PGDW proporciona a automatização e aceleração de processos tanto para a comissão quanto para os inscritos. A prova de conceito realizada demonstrou que o sistema é capaz de apoiar as etapas previstas no edital, desde a inscrição e envio de documentos até a divulgação do resultado definitivo, com suporte a interposição de recursos e avaliação curricular por barema. Além disso, traz a possibilidade de colaborar com seleções de outros cursos da instituição, visto que podem ser criadas outras instâncias da aplicação com o

mesmo código-fonte junto à Vercel e serem feitos os ajustes necessários, como a definição das credenciais do projeto Firebase relacionado.

O projeto é *open-source* e está disponível para colaboração de terceiros, visto que as contribuições são de grande relevância para a evolução do projeto e para tornar seu propósito mais amplo e eficaz. O código-fonte está disponível no repositório do projeto <https://github.com/crescenciolima/pos-web>, hospedado no GitHub sob a licença GPL 3.0, com instruções de instalação, dependências e configuração documentadas no README do repositório. O sistema em produção pode ser acessado através do link gerado pela Vercel pela URL <https://pos-web-ifba.vercel.app/>.

É importante ressaltar as ferramentas utilizadas. Tecnologias modernas, como o Next.js, porém consistente no conceito de SSR e baseado no *framework* React, já consolidado no mercado de desenvolvimento. E o Firebase, mantido pelo Google, que fornece a confiabilidade em relação ao armazenamento de dados. Destaca-se ainda a contribuição acadêmica do trabalho, visto que ainda existem poucos trabalhos no meio que relatem a utilização do Next.js junto ao Firebase, e a abordagem utilizada pode ser exemplo para a comunidade.

As principais contribuições deste trabalho são: (i) **contribuição técnica**, uma arquitetura de referência para sistemas de processo seletivo baseada em Next.js, Firebase e Vercel, com zero custo de infraestrutura; (ii) **contribuição institucional**, uma solução funcional e open-source adotada como prova de conceito pelo PGDW; e (iii) **contribuição acadêmica**, documentação de uma experiência de desenvolvimento com tecnologias modernas aplicadas à gestão acadêmica pública, incluindo lições aprendidas e ameaças à validade.

Para trabalhos futuros, sugere-se a ampliação dos testes realizados na aplicação a fim de verificar sua qualidade em diversos níveis, como código, funcionalidade, usabilidade e segurança. Em especial, recomenda-se a condução de estudos de usabilidade formais, utilizando instrumentos como o *System Usability Scale* (SUS), e a realização de testes de carga para avaliar o comportamento do sistema em cenários com alto volume de inscrições. Também é importante verificar a conformidade plena do sistema com a LGPD, especialmente no tocante ao tratamento de dados sensíveis. Junto a isso, seria interessante evoluir a funcionalidade de Relatórios para trazer estatísticas relevantes à medida que os processos forem sendo realizados através da plataforma.

Além disso, seria interessante reduzir a dependência das tecnologias Next.js e Vercel. Uma possível estratégia seria aumentar a modularização da aplicação, tornando-a menos acoplada à Vercel e mais portátil para outras plataformas. Outra abordagem seria testar alternativas como Docker e Kubernetes, permitindo maior controle sobre a infraestrutura e evitando custos inesperados. Monitorar mudanças nas políticas da Vercel e do Next.js também é essencial para garantir a viabilidade do projeto a longo prazo.

## REFERÊNCIAS

- [1] S. Loh, *31 Tipos de Sistemas de Informação - 31 Maneiras de a Tecnologia da Informação ajudar as Organizações*. Porto Alegre, RS, 2014.
- [2] R. G. Maciel, P. G. Fonseca, F. R. Duarte, and E. M. d. Santos, "Sistema eletrônico do serviço de informação ao cidadão (e-sic) e sua contribuição para a transparência: uma experiência gerencial em uma universidade federal," *Perspectivas em Ciência da Informação*, vol. 24, p. 143–164, Apr 2019.
- [3] F. D. S. LUNA and V. J. BRETERNITZ, "Digital transformation in private Brazilian higher education institutions: Pre-coronavirus baseline," *RAM. Revista de Administração Mackenzie*, vol. 22, no. 6, p. eRAMD210127, 2021.
- [4] T. Janowski, "Digital government evolution: From transformation to contextualization," *Government Information Quarterly*, vol. 32, no. 3, pp. 221–236, 2015.
- [5] F. Neves and P. Silva, "E-government em portais públicos de municípios: do visível para o invisível," *Revista Catarinense da Ciência Contábil*, vol. 20, p. e3160, 2021.
- [6] E. G. D. Silva, F. F. Hatori, and T. Gonçalves, "Sistema de gerenciamento de processo seletivo - sps." <https://acervodigital.ufpr.br/handle/1884/35016>, 2013. Acesso em 02 Fev 2025.
- [7] C. Singh, N. S. Gaba, M. Kaur, and B. Kaur, "Comparison of different ci/cd tools integrated with cloud platform," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 7–12, IEEE, 2019.
- [8] R. Hat, "Integração e entrega contínuas: Ci/cd." <https://www.redhat.com/pt-br/topics/devops/what-is-ci-cd>, 2028. Acesso em 07 Fev 2025.
- [9] G. G. Claps, R. Berntsson Svensson, and A. Aurum, "On the journey to continuous deployment: Technical and social challenges along the way," *Information and Software Technology*, vol. 57, pp. 21–31, 2015.
- [10] N. P. C. D'Oliveira and F. J. A. P. Cunha, "Lei geral de proteção de dados (lgpd): a relação entre as políticas e os regimes de informação," *RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação*, vol. 22, no. 00, p. e024015, 2024.
- [11] A. Sistemas, "Processo seletivo." <https://aix.com.br/software-processo-seletivo/>, 2021. Acesso em 07 Fev 2025.
- [12] M. R. Lange, "Atualização tecnológica do formulário de inscrição do processo seletivo da pós-graduação da ufsc." [https://repositorio.ufsc.br/bitstream/handle/123456789/187874/Monografia\\_Makhles.pdf?sequence=1&isAllowed=y](https://repositorio.ufsc.br/bitstream/handle/123456789/187874/Monografia_Makhles.pdf?sequence=1&isAllowed=y), 2018. Acesso em 07 Fev 2025.
- [13] S. Negreiros, M. D. A. Mesquita, and C. V. S. Lacerda, "sgpip: Sistema de gerenciamento do programa institucional de permanência acadêmica," *Revista de Ciência da Computação*, vol. 4, pp. 7 – 12, maio 2022.
- [14] K. Konshin, *Next.js Quick Start Guide: Server-side rendering done right*. Packt Publishing Ltd, 2018.
- [15] M. A. Jadhav, B. R. Sawant, and A. Deshmukh, "Single page application using angularjs," *International Journal of Computer Science and Information Technologies*, vol. 6, no. 3, pp. 2876–2879, 2015.
- [16] S. C. Hoang, "Shopify upsell app: Using next.js, react.js to boost sale." [https://www.theseus.fi/bitstream/handle/10024/354290/Chu\\_Son.pdf?sequence=2](https://www.theseus.fi/bitstream/handle/10024/354290/Chu_Son.pdf?sequence=2), 2020. Acesso em 07 Fev 2025.
- [17] I. Stack Exchange, "Stack overflow developer survey 2021." <https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-web-frameworks>, 2021. Acesso em 07 Fev 2025.
- [18] Vercel, "Unlock the full potential of next.js." [https://vercel.com/solutions/nextjs?utm\\_source=next-site&utm\\_medium=banner&utm\\_campaign=next-website](https://vercel.com/solutions/nextjs?utm_source=next-site&utm_medium=banner&utm_campaign=next-website), 2021. Acesso em 02 Fev 2025.
- [19] C. Khawas and P. Shah, "Application of firebase in android app development-a study," *International Journal of Computer Applications*, vol. 179, no. 46, pp. 49–53, 2018.
- [20] D. Stevenson, "What is firebase? the complete story, abridged." <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>, 2021. Acesso em 07 Fev 2025.
- [21] I. Firebase, "Cloud firestore." <https://firebase.google.com/docs/firestore>, 2021. Acesso em 07 Fev 2025.
- [22] Next.js, "The react framework for production." <https://nextjs.org/>, 2021. Acesso em 07 Fev 2025.
- [23] G. Developers, "Lighthouse." Documentação Oficial do Google Chrome Developers, 2025. Disponível em: <https://developer.chrome.com/docs/lighthouse/>.