

Hériclys Sousa Borges
Orientador: Msc. Rayner Gomes Sousa

Sistema de Gerenciamento de Processo Seletivo Para Cursinho Popular Pré-ENEM Paulo Freire

Picos - PI
31 de julho de 2023

Hériclys Sousa Borges
Orientador: Msc. Rayner Gomes Sousa

Sistema de Gerenciamento de Processo Seletivo Para Cursinho Popular Pré-ENEM Paulo Freire

Trabalho de Conclusão de Curso de Bacharelado em Sistemas de Informação do Campus Senador Helvídio Nunes de Barros da Universidade Federal do Piauí como parte dos requisitos para obtenção do Grau de Bacharel em Sistemas de Informação.

Universidade Federal do Piauí
Campus Senador Heuvídio Nunes de Barros
Bacharelado em Sistemas de Informação

Picos - PI
31 de julho de 2023

FICHA CATALOGRÁFICA
Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

B732s Borges, Hériclys Sousa

Sistema de gerenciamento de processo seletivo para cursinho popular Pré
– Enem Paulo Freire [recurso eletrônico] / Hériclys Sousa Borges - 2023.
56 f.

1 Arquivo em PDF

Indexado no catálogo *online* da biblioteca José Albano de Macêdo-CSHNB
Aberto a pesquisadores, com restrições da Biblioteca

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal do
Piauí, Bacharelado em Sistemas de Informação, Picos, 2023.
“Orientador : Prof. Me. Rayner Gomes Sousa”

1. *Web Site*. 2. *Site* - criação. 3. Automotização – processo seletivo. 4.
Cursinho popular. 5. Pré - Enem. I. Sousa, Rayner Gomes. II. Título.

CDD 005.133

SISTEMA DE GERENCIAMENTO DE PROCESSO SELETIVO PARA CURSINHO
POPULAR PRÉ-ENEM PAULO FREIRE

HÉRICLYS SOUSA BORGES

Monografia aprovada como exigência parcial para obtenção do grau de Bacharel em Sistemas
de Informação.

Data de Aprovação

Picos – PI, 14 de agosto de 2023



Prof. Rayner Gomes Sousa

Prof. Juliana Oliveira de Carvalho



Prof. Frank Cesar Lopes Vera

Agradecimentos

Inicio este tópico agradecendo à Deus, por me permitir estar vivo e me dando chances de, todo dia, viver uma vida maravilhosa e feliz ao lado de pessoas que eu amo. Gostaria de agradecer também a minha família, em especial aos meus pais Maria Carleuza e Pedro Hilário, minhas irmãs Maria Clara e Janaína, minha filha Iza, também a minha namorada Rayssa Cabral. O apoio de vocês foi fundamental para eu superar os desafios e conquistar essa realização.

Agradeço à meu orientador Prof. Ms. Rayner Gomes Sousa, por toda a ajuda que me deu nesta caminhada, por todo conhecimento transmitido e pelas sugestões valiosas para aprimorar o trabalho. Além disso, agradeço também a todos os professores do curso de Sistemas de Informação, que contribuíram significativamente para minha formação acadêmica.

Por fim, gostaria de agradecer a todos os amigos que fiz durante o curso, que além de compartilharmos o conhecimento, também vivenciamos momentos de descontração. Agradeço, em especial, aos amigos mais próximos: Thiago Paes, Tiago Oliveira, Francisco das Chagas (Chaguinha), Deusimar Damiano e Lairton Sullivan. Sem vocês, essa jornada não teria sido tão rica e prazerosa.

H. D'Larii

O que importa é o que interessa.

Resumo

Os Cursinhos Populares, também chamados de Pré-Enem, estão presentes na maioria das universidades do Brasil. O objetivo principal desses cursos é oferecer conhecimento preparatório para o Exame Nacional do Ensino Médio, a baixo ou nenhum custo, para estudantes de baixa renda. O maior diferencial dos cursinhos populares é que os alunos que possuem base escolar mais carentes são amparados mais de perto pelos educadores, tentando compreender suas dificuldades e até mesmo acompanhando esses alunos individualmente. Estas iniciativas não se restringem apenas ao estudo pré-vestibular, mas contribuem grandemente incentivando os alunos através da familiaridade com os universitários, professores e demais integrantes que convivem no ambiente das universidades, que têm experiência com provas de vestibulares ou concursos públicos mais concorridos. Além de tudo, a influência dos cursinhos populares não se reflete somente em alunos do pré-vestibular, como também, nos voluntários e educadores que podem dividir com os alunos seus conhecimentos e experiências, bem como aprender com outras realidades distintas. O convívio entre os professores e alunos acaba influenciando no desenvolvimento de um maior senso de responsabilidade por parte dos voluntários. Considerando que os alunos enxergam essas iniciativas como uma inspiração para se tornarem mais responsáveis, para desenvolver mais habilidades de liderança principalmente na tomada de decisões, características que são requisitos fundamentais em processos de seleção para a maioria das vagas de empregos, independente da área. O processo de inscrição desses cursinhos geralmente se dá de forma manual. Os alunos devem se dirigir à comissão responsável com a documentação impressa para realizar a inscrição, gerando um acúmulo de documentos e transtornos com o armazenamento dos mesmos. Outrossim, os concorrentes devem responder a um questionário impresso elaborado pela comissão organizadora. Ao final do período de inscrição a comissão tem que analisar todos os documentos e questionários para gerar uma nota para cada aluno que servirá como aporte para concorrer com os outros participantes. Esse método de inscrição gera diversos problemas, principalmente os relacionados ao processo de análise individual de cada processo, a perda ou acúmulo das documentações dos concorrentes. A proposta deste trabalho foi desenvolver um *web site* para o gerenciamento dos processos de inscrição e seleção dos alunos do Cursinho Popular Pré-Enem Paulo Freire da Universidade Federal do Piauí, Campus Senador Helvidio Nunes de Barros, Picos-PI. O site é de acesso aberto a todos os interessados em participar dos processos seletivos. Sua contribuição para sociedade é facilitar e automatizar o processo de seleção e avaliação dos concorrentes às vagas ofertadas pelo cursinho.

Palavras-chaves: Cursinho, Cursinho Popular, Pré-Enem, *Web Site*, *Site*, Automatizar Processo de Seleção.

Abstract

Popular courses, also called Pre-Enem, are present in most universities in Brazil. The main objective of these courses is to offer preparatory knowledge for the National High School Exam, at low or no cost, to low-income students. The most significant difference between the popular courses is that the students who have a lower school base are supported more closely by the educators, trying to understand their difficulties and even following these students individually. These initiatives are not restricted to pre-university study but contribute significantly by encouraging students through familiarity with university students, professors, and other members who live in the university environment who have experience with more competitive entrance exams or public examinations. Furthermore, the popular courses' influence is reflected not only in pre-university entrance exam students but also in volunteers and educators who can share their knowledge and experiences with students and learn from other different realities. The interaction between teachers and students influences the development of a greater sense of responsibility for volunteers. At the same time, students see these initiatives as an inspiration to become more responsible, and to develop more leadership skills, especially in decision-making, characteristics that are fundamental requirements in selection processes for most job openings, regardless of area. The registration process for these courses is usually done manually. Students must go to the responsible committee with the printed documentation to register, generating an accumulation of documents and inconvenience with their storage. Furthermore, competitors must answer a printed questionnaire prepared by the organizing committee. At the end of the registration period, the committee has to analyze all documents and questionnaires to generate a grade for each student that will contribute to competing with the other participants. This registration method creates several problems, especially those related to the individual analysis of each process and the loss or accumulation of documentation from competitors. This work aimed to develop a website for managing students' enrollment and selection processes from the Popular Pre-Enem Course Paulo Freire at the Federal University of Piauí, Campus Senador Helvidio Nunes de Barros, Picos-PI. The website is open to all interested in participating in the selection process. Its contribution to society is to facilitate and automate competitors' selection and evaluation process for vacancies offered by the course.

Lista de ilustrações

Figura 1 – Arquitetura Cliente-Servidor	18
Figura 2 – Estrutura dos Sistemas <i>Web</i>	19
Figura 3 – <i>Ranking</i> das Linguagens mais Utilizadas	21
Figura 4 – Modelagem do Banco de Dados	33
Figura 5 – Casos de Uso do Sistema	34
Figura 6 – Representação dos Usuários do Sistema	35
Figura 7 – Tela Inicial do Aluno	36
Figura 8 – Tela de Visualização de Turmas Abertas	36
Figura 9 – Tela de Realizar Inscrição	37
Figura 10 – Tela <i>Upload</i> de Documentação	37
Figura 11 – Tela de Realizar Inscrição	38
Figura 12 – Tela de Responder Questionário Socioeconômico	38
Figura 13 – Tela Inscrição Concluída	38
Figura 14 – Tela do Painel do Gerenciador do Sistema	39
Figura 15 – Tela de Gerenciamento de Turmas	40
Figura 16 – Tela de Criar Nova Turma	40
Figura 17 – Tela de Gerenciamento de Turma Específica	41
Figura 18 – Tela de Avaliar Inscrição	41
Figura 19 – Tela de Visualização de Documentos	42
Figura 20 – Tela de Avaliar Respostas do Formulário	42
Figura 21 – PDF Gerado Pelo Sistema	43
Figura 22 – Tela de Resultados	43
Figura 23 – Tela de Gerenciamento de Formulários	44
Figura 24 – Tela de Criar Formulário	44
Figura 25 – Tela de Gerenciamento de Gerentes	45
Figura 26 – Tela de Criar Novo Gerente	45
Figura 27 – Tela de Gerenciar Alunos	46
Figura 28 – Questão 1	54
Figura 29 – Questão 2	54
Figura 30 – Questão 3	55
Figura 31 – Questão 4	55
Figura 32 – Questão 5	55
Figura 33 – Questão 6	56
Figura 34 – Questão 7	56

Figura 35 – Questão 8	56
Figura 36 – Questão 9	57
Figura 37 – Questão 10	57

Lista de tabelas

Tabela 1 – Trabalhos Relacionados.	29
Tabela 2 – Requisitos Funcionais do Sistema	31
Tabela 3 – Requisitos Não Funcionais do Sistema	32
Tabela 4 – Perguntas do Questionário de Avaliação	47
Tabela 5 – Resultados da Pesquisa de Usabilidade	48

Lista de abreviaturas e siglas

ENEM	Exame Nacional do Ensino Médio
UFPI	Universidade Federal do Piauí
IHC	Interação Humano Computador
CPPPF	Cursinho Pré-ENEM Professor Paulo Freire
OAB	Ordem dos Advogados do Brasil
FTP	<i>File Transfer Protocol</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
HTTP	<i>Hypertext Transfer Protocol</i>
XML	<i>Extensible Markup Language</i>
LP	Linguagem de Programação
BSD	<i>Berkeley Software Distribution</i>
URL	<i>Uniform Resource Locator</i>
HTML	Linguagem de Marcação de Hipertexto
CSS	<i>Cascading Style Sheet</i>
XHTML	<i>EXtensible Hypertext Markup Language</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Sockets Layer</i>
TI	Tecnologia da Informação
PHP	<i>Hypertext Preprocessor</i>
FEL	<i>Finnish Esports League</i>
REST	<i>Representational State Transfer</i>
API	<i>Application Programming Interface</i>
RUP	<i>Rational Unified Process</i>

RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais
BD	Banco de Dados
DER	Diagrama Entidade Relacionamento
UML	Unified Modeling Language
PDF	<i>Portable Document Format</i>

Sumário

1	Introdução	15
1.1	Objetivos	16
1.2	Objetivos Específicos	16
1.3	Organização do Trabalho	16
2	Referencial Teórico	17
2.1	Cursos Populares	17
2.2	Arquitetura Cliente-Servidor	18
2.3	Sistemas <i>Web</i>	19
2.4	Sistemas <i>Web</i> como Auxílio ao Gerenciamento de Informação	20
3	Tecnologias Utilizadas	21
3.1	Linguagem de Programação <i>Python</i>	21
3.2	<i>Framework Django</i>	23
3.3	Linguagens de Construção do <i>site</i>	24
3.3.1	HTML	24
3.3.2	CSS	24
3.3.3	JavaScript	24
3.4	Banco de dados <i>PostgreSQL</i>	25
4	Trabalhos Relacionados	27
4.1	Definição dos Trabalhos	27
4.2	Considerações Finais	28
5	A Ferramenta CPPPF	30
5.1	Descrição do Sistema	30
5.2	Requisitos Funcionais e Não Funcionais	30
5.3	Modelagem do Banco de Dados	32
5.4	Diagrama de Caso de Uso	33
5.5	Usuários do Sistema	35
5.5.1	Usuário Aluno	35
5.5.2	Usuário Gerenciador	39
5.5.2.1	Gerenciamento de Turmas	39
5.5.2.2	Gerenciamento de Formulários	43
5.5.2.3	Gerenciamento de Gerentes	44
5.5.2.4	Gerenciamento de Alunos	45

6 Resultados e Discussões	47
7 Conclusão	49
7.1 Trabalhos Futuros	49
Referências	50
Apêndices	53
APÊNDICE A Resultados da Avaliação.	54

1 Introdução

Assistimos, nos últimos anos, a expansão dos cursos pré-vestibulares dirigidos a segmentos específicos da população com o propósito de promover maior oportunidade de acesso ao ensino superior, em que são historicamente sub-representados. As políticas de educação de caráter universalista, orientadas pela lógica da oferta, não lograram cumprir a promessa de tratar a todos igualmente, o que tem alimentado forte tendência para a adoção de políticas inspiradas na lógica da demanda, as quais tomam o direito à diferença como base do direito à igualdade. Vários estudos alertam para os riscos de fragmentação e dispersão quando políticas focalizadas tornam-se prioritárias (MITRULIS; PENIN, 2006).

Cursinhos populares tem como objetivo preparar os alunos para realizar o Exame Nacional do Ensino Médio (ENEM), ofertando aulas sem nenhum custo ou com contribuições simbólicas a fim de amenizar despesas com o material didático, por exemplo. Geralmente são ofertados a alunos de escolas públicas, pessoas de baixa renda, ou bolsistas de colégios particulares.

Segundo Kato (2015), os espaços alternativos não-formais de ensino tal como os chamados “cursinhos populares” podem representar um campo privilegiado para o acesso a esse processo formativo e as vivências acessíveis apenas à determinada classe hegemônica. Isso porque os professores voluntários podem compartilhar experiências inacessíveis em outros espaços por esses cidadãos. O contato dos estudantes desses núcleos com universitários, pós-graduandos e até mesmo docentes das Universidades constitui um ambiente rico em vivências, e trocas de ideias que aproxima o trabalhador da realidade acadêmica e do sonho de obter um título de ensino superior.

Hoje, porém, já se admite que, limitados os seus riscos e mantidas as responsabilidades públicas com políticas universais, as políticas de ação afirmativa têm um potencial não desprezível de ampliar o leque de opções para a construção de uma sociedade justa e democrática, adotando estratégias de intervenção que privilegiam grupos sociais em desvantagem (MITRULIS; PENIN, 2006).

O processo seletivo de alunos conta geralmente com uma prova de conhecimentos gerais, muitas vezes muito semelhante ao formato de vestibulares renomados, pautados em testes e conceitos específicos; bem como uma análise sócio econômica das rendas familiares e gastos mensais. alguns núcleos fazem visitas às residências para comprovar a condição de vida do candidato; além disso em muitos cursinhos populares há ainda uma entrevista dos candidatos com professores e/ou alunos veteranos que estão no projeto a mais tempo (KATO, 2015).

Nesse contexto, é notável a necessidade de uma ferramenta para ajudar ou auxiliar tanto no processo seletivo, quanto no processo de classificação dos alunos para as vagas ofertadas. Partindo desse problema, surgiu a motivação para o desenvolvimento de uma

plataforma *web* para gerenciar, não só os processos de inscrição e seleção, como também gerenciar os demais processos que envolve o ingresso de candidatos ao cursinho popular ofertado pela Universidade Federal do Piauí.

1.1 Objetivos

O presente trabalho tem como objetivo desenvolver uma aplicação baseada na plataforma *web* para o gerenciamento de todos os processos referentes ao ingresso de candidatos ao Cursinho Popular Pré-Enem Paulo Freire, ofertado pela Universidade Federal do Piauí Campus Senador Helvidio Nunes de Barros, Picos - PI.

1.2 Objetivos Específicos

Este trabalho tem como objetivos específicos:

1. Desenvolver uma aplicação *web* utilizando-se de técnicas de responsividade e conceitos de Interação Humano Computador (IHC);
2. Identificar os fatores que possuem maior impacto sobre a eficiência desse trabalho em relação aos métodos manuais utilizados antes da implantação do CPPPF;
3. Realizar testes de usabilidade de *software* para identificar melhorias e/ou problemas relacionados à interação dos usuários com a aplicação.

1.3 Organização do Trabalho

Esta monografia foi dividida em sete capítulos, organizados da seguinte maneira: Capítulo 2 apresenta o referencial teórico, onde são detalhados conceitos relevantes para este entendimento deste trabalho. No Capítulo 3 serão informadas as principais ferramentas e tecnologias utilizadas para o desenvolvimento do sistema. No Capítulo 4, são exibidos os trabalhos que abordam a mesma temática desta preposição. O Capítulo 5, detalha como foi o desenvolvimento aplicação CPPPF. O Capítulo 6 demonstra como foi realizado o processo de avaliação do *software*. O Capítulo 7 apresenta a conclusão e trabalhos futuros.

2 Referencial Teórico

Neste capítulo são expostos assuntos fundamentais para compreensão do presente trabalho, tais como métodos, técnicas e tecnologias. A Seção 2.2 apresenta conceitos sobre a arquitetura cliente-servidor. Na Seção 2.3 é abordado as principais características dos sistemas *web*. E na seção 2.4 contém algumas informações sobre os sistemas *web* como auxílio ao gerenciamento de informação.

2.1 Cursos Populares

Os cursos populares tiveram origem na década de 1960, devido a um vasto aumento de disputa pelas vagas ofertadas nas universidades. Segundo Zago (2008), As primeiras experiências dos núcleos de pré-vestibulares populares surgem no Brasil na segunda metade dos anos 80, consolidam-se na década de 90 do século XX e têm como principal objetivo a democratização do ensino. Esses cursos surgem em um contexto contraditório do sistema educacional que, conforme observações acima, se por um lado ampliou consideravelmente o número de vagas de outro mantém profundas desigualdades no acesso ao ensino superior.

Os cursos pré-vestibular são popularmente conhecidos como cursinhos, e são uma modalidade de cursos preparatórios voltados para estudantes brasileiros que vão prestar vestibular e/ou Exame Nacional do Ensino Médio (ENEM). O conteúdo desses cursos são o que foi aprendido pelos alunos durante toda a sua vida na escola, a fim de conseguirem uma boa pontuação nas provas dos vestibulares, ingressando, assim, nas instituições de ensino superior.

O termo “cursinho” não remete somente aos cursos pré-vestibulares, os mesmos também podem se referir a outras modalidades de ensino preparatórios, assim como os que são voltados àqueles que pretendem prestar um concurso público. Também se encaixa aos que ensinam os estudantes de direito para o Exame da Ordem dos Advogados do Brasil (OAB).

Esses cursos pré-vestibulares, que aqui denominamos de Cursos Pré-Vestibulares Populares, são iniciativas educacionais de entidades diversas, de trabalhadores em educação e de grupos comunitários, destinados a uma parcela da população que é colocada em situação de desvantagem pela situação de pobreza que lhe é imposta (NASCIMENTO, 2002).

2.2 Arquitetura Cliente-Servidor

O sistema cliente-servidor pode ser definido como uma arquitetura de software composta de cliente e servidor, em que os clientes sempre enviam solicitações enquanto o servidor responde aos pedidos enviados. O cliente-servidor fornece uma comunicação entre processos porque envolve a troca de dados do cliente e do servidor em que cada um deles executa diferentes funções. (OLUWATOSIN, 2014).

Essa arquitetura possibilita que dois processos executados em máquinas distintas enviem mensagens um para o outro, e para isso, um processo deve ser o servidor, que estará escutando em uma determinada porta e o outro processo, o cliente, o qual enviará uma mensagem ao servidor através de algum protocolo (TANENBAUM; FILHO, 1995).

Segundo Oluwatosin (2014), no ambiente computacional, sistemas cliente-servidor tornaram-se populares por serem usados em diferentes aplicações. Alguns protocolos tornaram-se padrões e clientes e servidores os usam para comunicar-se entre si. Dentre eles podemos citar: Protocolo de Transferência de Arquivo (FTP), Protocolo Simples de Transporte de Correio (SMTP) e *Hypertext Transfer Protocol* (HTTP). A Figura 1 demonstra um exemplo da arquitetura cliente-servidor em um sistema *web*

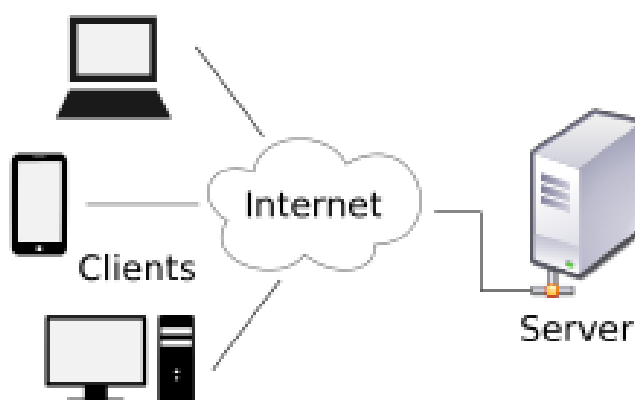


Figura 1 – Arquitetura Cliente-Servidor

Usando a figura 1 como base, os clientes enviam requisições para o servidor, através da internet, e este por sua vez processa as requisições e envia as respostas das requisições. Os servidores estão dispersos em diversos lugares do mundo. A informação que o usuário está acessando de seu terminal pode estar armazenada em qualquer continente, inclusive na mesma cidade em que ele está, dependendo de onde o sistema tiver sido hospedado.

Geralmente é o servidor que armazena os sistemas mais pesados da rede, como por exemplo o banco de dados. Como os clientes não rodam aplicativos tão robustos quanto os servidores, as máquinas dos clientes requerem um poder de processamento menor que as dos servidores.

2.3 Sistemas Web

Os sistemas web, surgiram na década de 1990 com o objetivo de formar um repositório do conhecimento humano (BERNERS-LEE, 1992). Baseando-se em mecanismos de armazenamento, recuperação e visualização de documentos eletrônicos.

De acordo com Gutiérrez (2014), um sistema *Web*, se define com um *software* em que aqueles que são cadastrados podem se conectar de qualquer lugar, bastando apenas ter conexão com a *internet*. Muitas vezes é vantajoso usá-lo como forma de melhorar seu negócio, para isso, é preciso entender seu ramo e como o sistema beneficiará os envolvidos. Podemos definir um *framework* da *Web* como um conjunto de componentes (por exemplo, classes e descritores e arquivos de configuração *Java* em XML) que compõem um *design* reutilizável que facilita e acelera o desenvolvimento de sistemas da *Web*.

Um aplicativo da web é qualquer programa de computador que executa uma função específica usando um navegador da web como seu cliente. O aplicativo pode ser tão simples quanto um quadro de mensagens ou um formulário de contato em um site. Pode ser tão complexo quanto um processador de texto ou um aplicativo de jogos para vários jogadores jogarem de forma concomitante (NATIONS, 2018).

Existem vários tipos de *frameworks Web*: orientados para a interface do usuário, como o *Java Server Faces*, orientados para aplicações de publicação de documentos, como *Coocon*, orientados ao controle de eventos, como *Struts* e alguns que incluem diversos elementos como Tapeçaria (GUTIÉRREZ, 2014).

O usuário interage com aplicativos da *Web* por meio do navegador. Como resultado da atividade do usuário, as solicitações são enviadas ao servidor, onde o aplicativo é hospedado e normalmente faz uso de um banco de dados que armazena todas as informações relacionadas a ele. O servidor processa a solicitação e retorna a resposta ao navegador que a apresenta ao usuário (GARRIDO, 2004). A figura 2 ilustra essas interações.

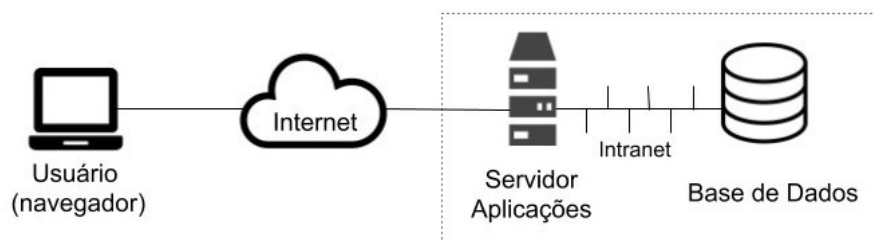


Figura 2 – Estrutura dos Sistemas Web

Segundo Winckler e Pimenta (2002), a arquitetura é basicamente um cliente/servidor instalado sobre uma rede de computadores heterogênea. Do lado do cliente está um programa, chamado *browser* ou navegador, que intermedeia a solicitação de informações ao

servidor e as apresenta para o usuário. O servidor atende os diferentes clientes bem como outros servidores indistintamente.

Winckler e Pimenta (2002) ainda completam afirmando que embora o modelo referencie uma arquitetura cliente/servidor, talvez fosse mais adequado chamá-la de *request/respost* (pedido/resposta) porque, de fato, este é o tipo de comunicação entre as partes; o cliente *browser* solicita um documento ao servidor que processa a chamada, envia o documento ao cliente e encerra a comunicação. Assim, as comunicações entre o cliente e o servidor ocorrem de um modo descontínuo entre as chamadas.

2.4 Sistemas *Web* como Auxílio ao Gerenciamento de Informação

A web tem como característica o fornecimento de uma estrutura de hipertexto entre os documentos que são armazenados, refletindo a necessidade dos usuários de organizar seus conhecimentos. Isso significa que os documentos contêm links ou *hyperlinks* que são referências para outros documentos e recursos que também estão armazenados na *web* (COULOURIS et al., 2013).

A tecnologia web foi adquirindo novos recursos e funções, melhor dizendo, os usuários ao invés de solicitarem páginas apenas com conteúdo estático, podem enviar informações aos servidores, os quais, por sua vez, processavam-na e as enviam de volta ao usuário de forma dinâmica o resultado, além disso, proporciona oportunidades de negócios através da tecnologia (JUNIOR, 2003).

Com o avanço das tecnologias, impulsionadas pela evolução da internet, os sistemas *web* vêm mostrando grande progresso na tomada de decisões, bem como no gerenciamento das informações. Com isso, esses avanços tecnológicos estão possibilitando o surgimento de sistemas *web* para áreas específicas.

Os Sistemas de Informações Gerenciais são importantes ferramentas para otimização da comunicação e da tomada de decisão nas empresas, visto que estes são problemas sérios para a eficiência e a eficácia nos processos produtivos. Assim, os sistemas tendem a melhorar a qualidade das informações e do processo decisório (OLIVEIRA, 2007).

3 Tecnologias Utilizadas

Nesta seção serão abordados os materiais, ferramentas e tecnologias utilizadas para o desenvolvimento do sistema proposto no presente trabalho. A seção 3.1 conceitua a linguagem de programação *Python* e as vantagens de sua utilização; A seção 3.2 apresenta o *framework* usado no desenvolvimento da aplicação; Na seção 3.3 menciona-se as tecnologias aplicadas na construção do sistema web e A seção 3.4 aborda o banco de dados utilizado como repositório para o armazenamento de informações.

3.1 Linguagem de Programação *Python*

Python é uma linguagem de programação interpretada simples, mas poderosa, que preenche a lacuna entre a programação C e *shell* e, portanto, é ideal para “programação descartável” e prototipagem rápida. Sua sintaxe é construída a partir de construções emprestadas de uma variedade de outras linguagens; mais proeminentes são as influências de ABC, C, *Modula-3* e *Icon*. O interpretador *Python* é facilmente estendido com novas funções e tipos de dados implementados em C. *Python* também é adequado como uma linguagem de extensão para aplicativos C altamente personalizáveis, como editores ou gerenciadores de janela. *Python* está disponível para vários sistemas operacionais, entre os quais vários sabores de *UNIX* (incluindo *Linux*), o *Apple Macintosh O.S.*, *MS-DOS*, *MS-Windows 3.1*, *Windows NT* e *OS / 2* (ROSSUM, 1995).

Python vem ganhando muita popularidade nos últimos anos entre programadores. De acordo com o site *iMasters*¹, a Linguagem de Programação (LP) *Python* é a segunda mais usada no ano de 2022, ficando atrás apenas de *JavaScript*, conforme mostrado na Figura 3.



Figura 3 – *Ranking* das Linguagens mais Utilizadas *iMasters*

¹ <https://imasters.com.br/>

Python é uma linguagem de programação interpretada de alto nível e de uso geral. Criado por Guido van Rossum e lançado pela primeira vez em 1991, a filosofia de design do *Python* enfatiza a legibilidade do código com seu uso notável de espaços em branco significativos. Suas construções de linguagem e abordagem orientada a objetos visam ajudar os programadores a escrever código lógico e claro para projetos de pequena e grande escala (ROSSUM et al., 2007).

Dentre todas as características da LP *Python*, a simplicidade é a que mais chama atenção, além de ser intuitiva, e oferecer uma programação orientada a objetos, com bibliotecas e módulos reutilizáveis. A seguir são mencionadas algumas vantagens ao se escolher a LP *Python* para desenvolver um projeto segundo o site do ICEV².

Simplicidade: *Python* é uma linguagem que requer menos código para concluir tarefas básicas quando comparada a outros padrões de programação, podendo ser de 3 a 5 vezes menor do que Java — e de 5 a 10 vezes mais reduzida em relação ao código em C++. Assim, quanto menos se escreve, menores são as chances de erros, especialmente por parte de iniciantes.

Sintaxe intuitiva: A sintaxe das linguagens de programação costuma ser um emaranhado de regras que, na maior parte dos casos, induzem o programador a erros. Mas isso não acontece com *Python*. Um exemplo clássico sobre essa maneira intuitiva de formatar expressões é que um comando termina exatamente quando a linha acaba. Ou seja, as exigências e regras específicas de sintaxe são mínimas e, assim, a produtividade é maximizada.

Documentação farta: Além da documentação oficial e dos registros feitos pelos próprios programadores durante o desenvolvimento e testes, há uma grande variedade de publicações sobre *Python* — como materiais didáticos e outros conteúdos digitais livres, de acesso gratuito. Outro benefício é a tradução voluntária de membros da comunidade, facilitando a compreensão dos conteúdos em diversos idiomas e democratizando o acesso à linguagem.

Bibliotecas abertas: *Python* possui diversos módulos, em boa parte livres e gratuitos. Dessa forma, aprendizes podem ter uma experiência de imersão em determinada área de aplicação, como criação de jogos, *machine learning* e interação com a *web*. Por ser uma linguagem de *script* de código aberto, há uma base de conhecimento ampla consolidada por programadores, que produzem tutoriais e registram suas experiências em um ambiente colaborativo — o que potencializa o aprendizado.

Reutilização de módulos: A modularização e a capacidade de empacotamento são outras duas vantagens da linguagem. Com isso, estruturas completas podem ser desmontadas e divididas em componentes para reutilização em outros programas. Com a adoção de frameworks — e o aproveitamento de criações anteriores e de funcionalidades já testadas —, há economia de tempo e simplificação do trabalho do desenvolvedor.

² <https://www.somosicev.com/blogs/8-motivos-para-aprender-programar-em-python/>

Multiplataforma: Por ser uma linguagem interpretada —e não compilada para uma linguagem de máquina —, *Python* roda em diferentes plataformas. Isso significa que basta ter um interpretador para que a execução seja produtiva em qualquer sistema ou processador. Além disso, a linguagem é multiparadigma, ou seja, passível de desenvolvimento para qualquer tipo de ambiente e interface, como *web*, móvel ou *desktop*.

3.2 Framework Django

*Django*³ é uma estrutura da *Web Python* de alto nível que incentiva o desenvolvimento rápido e um design limpo e pragmático. Construído por desenvolvedores experientes, ele cuida de grande parte do trabalho de desenvolvimento da *Web*, para que você possa se concentrar em escrever seu aplicativo sem precisar reinventar a roda. É gratuito e de código aberto.

De acordo com o site do *Django*, O nome *Django* foi inspirado no músico de jazz Django Reinhardt, *Django* Foi criado originalmente como sistema para gerenciar um site jornalístico na cidade de Lawrence, no Kansas. Tornou-se um projeto de código aberto e foi publicado sob a licença BSD em 2005. É um *framework* gratuito para a criação de aplicações *web*, escrito em *Python*, além disso, é um conjunto de componentes que ajuda a desenvolver sites de forma mais rápida e mais fácil.

O *Django* é considerado como sendo uma *framework* Model-View-Controller (MVC), embora tenha algumas diferenças em relação a este modelo. No modelo MVC, a parte *Model* refere-se à camada de acesso a dados, *View* à parte do sistema responsável pela escolha e apresentação da informação, e o *Controller* à parte responsável por decidir que *view* escolher, dependendo da situação, e aceder ao *Model* para a recolher (SILVA, 2014).

Este *framework* tem um padrão de três camadas. A primeira são os *Models*, onde classes são criadas e objetos descritos para posteriormente serem guardados no banco. A segunda camada é a de *Templates*, onde pode ser visto toda a parte de html do projeto. A última camada é a *View*, onde classes serão instanciadas, lógicas serão criadas e métodos serão montados para posteriormente serem referenciados nas páginas que montam as URLs. Assim, cada método será desenvolvido para uma página especificamente. Essas camadas são chamadas de MTV: *model*, *template* e *view* (BARROS, 2013).

A utilização do *framework Django* facilita o desenvolvimento e simplifica a criação de aplicações *web*. Sua manutenção, um ponto crucial para quem precisa produzir muito em pouco tempo, também é de fácil alteração por ter a doutrina de não repetir código em diversos arquivos. A manutenção em um bloco de código será facilmente dissipada a outros arquivos do sistema (BARROS, 2013).

³ <https://www.djangoproject.com/>

3.3 Linguagens de Construção do *site*

3.3.1 HTML

HTML (Linguagem de Marcação de HiperTexto) é o bloco de construção mais básico da *web*. Define o significado e a estrutura do conteúdo da *web*. Pode se dizer que o HTML é a linguagem base da internet. Foi criada para ser de fácil e simples entendimento por seres humanos e por máquinas, como por exemplo o *Google* ou outros sistemas que percorrem a *internet* capturando informação.

HTML é uma linguagem de marcação usada para especificar a estrutura de um documento. Um navegador de internet (*web browser*) nada mais é do que um *software* que interpreta estas marcações de estrutura e, então, constrói uma página *web* com recursos de hipermídia com os quais o usuário pode interagir (BORTOLOSSI, 2012).

O principal objetivo da linguagem HTML é criar páginas *web* de maneira simples e eficaz, onde qualquer pessoa que tenha um conhecimento básico na área, seja capaz de criar algo visual. Uma das principais vantagens desta linguagem é que roda em qualquer navegador, resultando numa maior satisfação dos seus usuários e desenvolvedores. (TORRES, 2018).

3.3.2 CSS

CSS (*Cascading Style Sheet*) é usado para estilizar elementos escritos em uma linguagem de marcação como HTML. Responsável por separar conteúdo da representação visual do site. Com ele, você pode modificar praticamente tudo dentro do seu *layout* (como as cores, *background*, características de fontes, margens, preenchimentos, posição, até a própria estrutura do *site* com a propriedade *float*).

De acordo com Grannell (2008) o conceito por trás do CSS era simples, mas revolucionário: remova a apresentação e separe o *design* do conteúdo. Deixe HTML (e mais tarde XHTML) lidar com a estrutura e use um documento CSS separado para a aplicação de apresentação visual.

3.3.3 JavaScript

JavaScript é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma (protótipos, orientado a objeto, imperativo e, funcional), disponível nos navegadores de *internet*. Possibilita páginas web interativas, sendo assim, é uma parte essencial em aplicativos *web*. A maioria dos *sites* usam, e os principais navegadores já possuem um mecanismo *JavaScript* dedicado a executá-lo.

A linguagem *JavaScript* disponibiliza uma série de recursos de interface gráfica (tais como botões, campos de entrada e seletores), viabilizando assim a construção de páginas

web mais interativas. Mais ainda, a linguagem *JavaScript* permite modificar e integrar, de forma dinâmica, o conteúdo e a aparência dos vários elementos que compõem o documento (BORTOLOSSI, 2012).

3.4 Banco de dados *PostgreSQL*

Um banco de dados é uma coleção de dados relacionados entre si, organizada e armazenada de forma a possibilitar fácil manipulação e controle. Um dado é um campo armazenado, matéria-prima para a informação. Já a informação são dados compilados e processados de acordo com a solicitação de consultas e análises. O banco de dados representa um aspecto do mundo real e é projetado, construído e preenchido com dados para um propósito específico (DATE, 2004).

Para Elmasri et al. (2005), um banco de dados é uma coleção de dados relacionados. Os dados são fatos que podem ser gravados e que possuem um significado implícito. Por exemplo, considere nomes, números telefônicos e endereços de pessoas que você conhece. Esses dados podem ter sido escritos em uma agenda de telefones ou armazenados em um computador, por meio de programas como o *Microsoft Access* ou *Excel*. Essas informações são uma coleção de dados com um significado implícito, conseqüentemente, um banco de dados.

A principal característica ao se utilizar um banco de dados é o fornecimento de algum nível de abstração de dados, por meio da omissão dos detalhes de armazenamento que não são necessários para a maioria dos usuários (YOSHIURA, 2015).

A utilização de um Banco de Dados implica numa melhor organização de suas informações, os dados e registros conseguem serem organizados e recuperados com facilidade. Esta facilidade se dá através de *softwares* especializados denominados Sistemas Gerenciadores de Banco de Dados (SGBDs). Estas ferramentas manipulam as informações de um banco de dados, permitindo ao usuário criar, editar, atualizar, armazenar e recuperar dados em um SGBD. Alguns SGBDs populares atualmente são: *SQL Server*, *MySQL*, *PostgreSQL* e *Oracle*.

A linguagem padrão utilizada pelos SGBD's é a SQL (*Structured Query Language*). É a linguagem de programação utilizada para realizar as consultas no banco de dados. Criada na década de 70, a linguagem tornou-se padrão universal para manipulação de dados em um SGBD. Serve para interagir aos SGBDs e executar várias tarefas referentes à manipulação dos dados.

O *PostgreSQL* é um dos sistemas gerenciadores de banco de dados objeto relacional de código aberto mais avançados que utiliza a linguagem SQL e é muito utilizado em aplicações para a *internet*.

Por muito tempo o *PostgreSQL* foi discriminado no mundo dos bancos de dados, e o seu recente aumento de popularidade veio de usuários de outros bancos de dados em busca

de um sistema com melhores garantias de confiabilidade, melhores recursos de consulta, mais operação previsível, ou simplesmente querendo algo mais fácil de aprender, entender e usar. Essas e outras características são encontradas no *PostgreSQL* (CARVALHO, 2017).

Milani (2008), descreve algumas características e recursos existentes no PostgreSQL.

- SGBD relacional com suporte a ACID;
- Replicação;
- Cluster (alta disponibilidade);
- Multithreads;
- Segurança SSL e criptografia;
- Incorporável em aplicações gratuitamente;
- Capacidade de armazenamento;

O *PostgreSQL* é uma boa alternativa, porque ele é multiplataforma, possui um baixo custo, possui suporte a linguagens de programação, e é recomendado pelo site da *Django*, que é a principal *framework* que será utilizada para a construção do site. Estas e outras vantagens fazem com que o *PostgreSQL* seja adotado por departamentos de TI e desenvolvedores *web*.

4 Trabalhos Relacionados

Neste capítulo são apresentadas pesquisas relevantes que contribuíram para a construção teórica deste trabalho com intuito de desenvolver uma plataforma capaz de gerenciar de maneira eficiente e eficaz todo o processo seletivo do cursinho popular Professor Paulo Freire. Foram selecionados estudos relacionados a plataformas de gerenciamento utilizando ferramentas *web* que propuseram alguma associação com a temática abordada.

O capítulo está dividido em duas partes: na primeira, são descritos os estudos e seus objetivos; na segunda, é apresentada as considerações finais diante desses estudos.

4.1 Definição dos Trabalhos

O trabalho de Zhou (2010) consiste em um sistema de gerenciamento de curso que é um site baseado em dados usado por instrutores e alunos. Este site tem quatro componentes principais: nota, classificação, grupo e envio. A implementação usa uma ferramenta chamada *Django Framework*, que é um excelente *framework* para aplicações *web* de código aberto para desenvolvimento de *sites* baseados em dados complexos.

Em seu trabalho, Caldeira (2015) desenvolveu um sistema *web* utilizando a linguagem de programação *Scripting* PHP, através da criação de um *website* denominado “As minhas músicas preferidas”, que geria os dados da base de dados remota. A gestão era realizada através da inserção, listagem e eliminação dos dados do banco de dados utilizado, caracterizando o *website* como dinâmico e versátil.

No trabalho de Wu (2015) é desenvolvido um *website* para uma plataforma de armazenamento em nuvem privada baseada em *OpenStack*, projetada e implementada com base no *framework* *Swift*. O *framework* *Django* foi adotado para construir o sistema do *site* para completar as tarefas de criação, *upload* e *download* de arquivos de dados corporativos por meio da *web*.

O trabalho de Arafat et al. (2017) é um site projetado em PHP e *Mysql* para o Instituto de Línguas de Empreendedorismo e Informática, que tem como missão realizar atividades de formação em inglês e aplicações informáticas conforme necessidade de mão de obra. O sistema a ser desenvolvido servirá para disponibilizar informação e promover atividades organizadas pelo Instituto.

Já no trabalho de Muslihudin e Larasati (2017) é projetado um sistema de aplicação para admissão de novos estudantes numa instituição educacional em Lampung, usando PHP e *MySQL*. O sistema torna mais fácil o processamento de dados de novos alunos candidatos no processo de inscrição de novos candidatos que estão dentro e fora da região, e apresenta informações sobre admissões de novos alunos que podem ser acessadas diretamente pelo site da mídia.

No trabalho de [Vainikka \(2018\)](#), ele divide o serviço *web* da *Finnish Esports League* (FEL) em dois serviços separados. O objetivo da divisão é facilitar o desenvolvimento do serviço e da manutenção criando um novo serviço construído com o *framework Django REST*. Assim, *REST* foi usado para *back-end* e o aplicativo de *front-end* foi criado com *React*, que se conecta ao *back-end* usando a *API REST*.

O trabalho de [Pietro e Petrucelli \(2019\)](#) trata-se de um *website* profissional de fotografia, que apresenta as principais tecnologias e técnicas mais utilizadas na *web* para criação de *sites* como o PHP trabalhando com Orientação a Objetos e utilizando sua arquitetura para uma maior organização do código e atendendo sua norma de trabalho, o *Bootstrap* que visa o conceito *mobile-first* atendendo a maior demanda de dispositivos móveis, o *MySQL* que trabalha como armazenamento de dados, HTML que possui um esqueleto único para padrões *web* e o *JavaScript* que sua necessidade foi de manipular eventos e elementos de todo *website*.

[Tiwari et al. \(2019\)](#) apresenta a implementação de um *site* de achados e perdidos baseado em *Python* para um campus universitário. O *site* é baseado em *Python*. Este *site* vai ajudar as pessoas a encontrar suas coisas perdidas. O usuário precisará apenas colocar sua coisa perdida no *site* e o mesmo irá encontrá-la o mais rápido possível. O projeto foi desenvolvido com base na *framework Django* como uma estrutura para *Python*.

Em seu trabalho, [Lamsal \(2020\)](#) desenvolveu uma aplicação *web* que consiste num *site* amigável e responsivo usando a linguagem de programação PHP. O *site* pode ser usado para procurar emprego no mercado de trabalho nepalês com um determinado custo do serviço de anúncio de emprego para o provedor de emprego. O aplicativo da *web* foi construído usando a linguagem de núcleo do PHP.

[Nurhasanah Siti e Rusdan \(2020\)](#) aborda em seu trabalho o desenvolvimento de um aplicativo de turismo e viagens baseado em um *website* no *front-end* e fazer reservas de bilhetes. O método de pesquisa utiliza uma abordagem qualitativa com técnicas de coleta de dados por meio da observação e estudo da literatura. Enquanto o método de desenvolvimento de sistema usado é o método *Rational Unified Process* (RUP).

4.2 Considerações Finais

A Tabela 1 apresenta uma comparação sucinta entre o presente estudo e outras pesquisas correlatas. A Tabela foi organizada em ordem cronológica, compreendendo o período de 2010 a 2020, e utilizou três critérios para estabelecer um paralelo entre as pesquisas relacionadas e o trabalho em questão. Os critérios utilizados na análise deste estudo abarcaram uma avaliação minuciosa dos objetivos apresentados. O primeiro critério adotado teve como foco comparar as propostas apresentadas em cada trabalho. O segundo critério consistiu em verificar se o trabalho em questão era de desenvolvimento, ou seja, se houve criação de alguma ferramenta com interface interativa. Por fim, o último critério utilizado

foi a avaliação das Linguagens e Ferramentas utilizadas em cada trabalho, que apresentou quais foram as tecnologias empregadas no desenvolvimento e criação das respectivas aplicações. Esses critérios foram utilizados para uma análise completa e abrangente dos trabalhos relacionados.

Este trabalho tem como objetivo descrever o desenvolvimento de um sistema *web* capaz de gerenciar todo o processo de inscrições de um cursinho popular da Universidade Federal do Piauí. O sistema tem como propósito o auxílio no gerenciamento do processo de seleção em geral, desde as inscrições, até a divulgação dos resultados da seleção. O principal diferencial deste trabalho em relação aos demais analisados é que o sistema conta com o gerenciamento de formulários diretamente no *website*. Estes formulários gerenciados de forma manual estão entre os maiores causadores de transtornos no decorrer dos processos seletivos dos cursinhos. A implementação do sistema poderá proporcionar uma maior eficiência e eficácia no processo de seleção dos alunos inscritos, bem como uma maior exatidão na divulgação dos resultados.

Tabela 1 – Trabalhos Relacionados.

Trabalho	Objetivo	Característica	Ferramentas/Técnicas
Zhou (2010)	<i>Website</i> para gerenciamento de curso.	Gerencia notas, classificação, grupos e envios dos alunos do curso.	Python, Django e SQLite.
Caldeira (2015)	<i>Website</i> para gerenciamento de dados de uma base de dados remota.	Insere, lista e elimina dados do banco de dados.	PHP e MySQL.
Wu (2015)	<i>Website</i> para uma plataforma de armazenamento em nuvem.	Permite criação, <i>upload</i> e <i>download</i> de arquivos por meio da <i>web</i> .	OpenStack, Python, Django e Swift.
Arafat et al. (2017)	<i>Website</i> para uma instituição que realiza atividades de formação em inglês e aplicações informáticas.	Gerencia as inscrições nas atividades da instituição.	PHP e MySQL.
Muslihudin e Larasati (2017)	<i>Website</i> para admissão de novos alunos numa instituição educacional	Gerencia e fornece dados dos alunos.	PHP e MySQL.
Vainikka (2018)	<i>Website</i> de torneio para jogadores de esporte.	Publica notícias e gerencia torneios.	Python, Django, React, RESTful e WordPress.
Pietro e Petrucci (2019)	<i>Website</i> profissional de fotografia.	Disponibiliza informações sobre os serviços disponíveis.	PHP, JavaScript, HTML e Bootstrap.
Tiwari et al. (2019)	<i>Website</i> de achados e perdidos para um campus universitário.	Armazena e busca informações sobre objetos perdidos.	HTML, CSS, JavaScript, Bootstrap, jQuery, Python, Django e SQLite.
Lamsal (2020)	<i>Website</i> para procurar vagas de emprego.	Fornece informações sobre vagas de emprego disponíveis.	PHP, MySQL, Javascript e Bootstrap.
Nurhasanah Siti e Rusdan (2020)	<i>Website</i> de turismo e viagens.	Realiza a reserva de bilhetes.	Python, Django e MySQL.
Este Trabalho	<i>Website</i> para o gerenciamento do processo seletivo de um cursinho popular Pré-ENEM.	Gerencia as inscrições, alunos, gerentes, formulários e o ranking de selecionados do processo seletivo.	Python, Django e PostgreSQL.

5 A Ferramenta CPPPF

Este capítulo apresenta informações sobre a ferramenta CPPPF, abordando a metodologia seguida, seu funcionamento, os testes realizados e os resultados alcançados.

5.1 Descrição do Sistema

A aplicação *web* desenvolvida no presente trabalho é um sistema de gestão que visa oferecer, de forma ágil e eficiente, o gerenciamento do processo seletivo do Cursinho Popular Pré-ENEM Paulo Freire da Universidade Federal do Piauí campus Picos. O *site* fornecerá informações sobre o cursinho tais como datas e horários para inscrições, permitindo também que o usuário que concorre às vagas acompanhe os resultados dos processos seletivos na plataforma.

O sistema foi desenvolvido utilizando a LP *Python* (versão 3.6) e o *framework Django* (versão 2.0). Por ser uma aplicação *web*, para a construção, estilização e comportamento das páginas, foram usadas as tecnologias HTML, CSS e *JavaScript*. Como repositório para os dados vindos do sistema, foi utilizado o SGBD *PostgreSQL*.

O sistema possibilita o gerenciamento das inscrições dos candidatos às vagas, das documentações dos candidatos, do questionário socioeconômico, do *ranking* de classificados, postagem de notícias sobre os processos seletivos, entre outras funcionalidades.

Ao se conectar no sistema com nome de usuário e senha (*login*) previamente cadastrados no sistema, o utilizador tem acesso total a todas as funcionalidades da aplicação tais como efetuar inscrição para processo seletivo (desde que tenha processo seletivo em aberto), buscar turmas abertas, buscar processos seletivos abertos, dentre outras. Usuários sem *login* podem visualizar notícias sobre o curso, tais como datas e horários sobre processos seletivos, divulgação de resultados.

5.2 Requisitos Funcionais e Não Funcionais

Os requisitos são as variadas funcionalidades que um sistema pode fornecer ao seu usuário, já os requisitos não funcionais são as regras de negócio e restrições que o sistema possui. Durante a levantamento e análise de requisitos, foram considerados todos os aspectos que o *software* deveria contemplar para atender às necessidades e expectativas dos usuários. Isso envolveu uma avaliação completa das funcionalidades que o sistema deveria ter, bem como as limitações técnicas e restrições que precisariam ser levadas em conta.

Além disso, foi importante estabelecer uma lista clara e concisa de requisitos funcionais e não funcionais que deveriam ser atendidos pelo *software*. Os requisitos funcionais se

referem às funções específicas que o sistema deve realizar, enquanto os requisitos não funcionais dizem respeito às características do sistema, como desempenho, segurança e usabilidade. As Tabelas 2 e 3 mostram os requisitos funcionais e não funcionais do sistema, respectivamente.

Tabela 2 – Requisitos Funcionais do Sistema

Identificador	Descrição
RF01: Cadastrar Usuário	O usuário sem login se cadastra para ter acesso a algumas das funcionalidades.
RF02: Modificar Cadastro	O usuário com cadastro poderá modificar dados do seu cadastro.
RF03: Excluir Cadastro	O usuário com cadastro poderá excluir dados do seu cadastro.
RF04: Realizar Matrícula	O usuário com cadastro poderá se matricular em uma turma.
RF05: Inserir Documentos	O usuário com cadastro deverá inserir seus documentos no ato da matrícula.
RF06: Modificar Documentos	O usuário com cadastro poderá modificar seus documentos após sua matrícula.
RF07: Consultar Documentos	O usuário com cadastro poderá consultar seus documentos após sua matrícula.
RF08: Excluir Documentos	O usuário com cadastro poderá excluir seus documentos após sua matrícula.
RF09: Cadastrar Turmas	O usuário administrador cadastrado poderá cadastrar uma turma.
RF10: Modificar Turmas	O usuário administrador cadastrado poderá modificar dados de uma turma.
RF11: Excluir Turmas	O usuário administrador cadastrado poderá excluir uma turma.
RF12: Gerenciar Alunos	O usuário administrador cadastrado poderá gerenciar alunos em uma turma.
RF13: Gerenciar <i>Ranking</i>	O sistema fará a seleção dos alunos que serão contemplados com as vagas ofertadas.

Tabela 3 – Requisitos Não Funcionais do Sistema

Identificador	Descrição	Categoria
RNF01	O sistema possui um <i>design</i> responsivo para que o usuário possa utilizar a aplicação de forma agradável tanto em <i>desktop</i> como em dispositivos móveis.	Portabilidade
RNF02	O sistema possui métricas de autenticação possibilitando segurança dos dados armazenados no BD.	Segurança
RNF03	Por se tratar de uma aplicação <i>web</i> , o sistema deve permanecer disponível a todo o momento.	Disponibilidade
RNF04	Os dados armazenados permanecem íntegros e seguros no BD.	Confiabilidade
RNF05	A aplicação deverá ter uma interface simples e intuitiva para motivar e facilitar a sua utilização.	Usabilidade
RNF06	O sistema deverá ser acessado por meio de um <i>browser</i> (navegador).	Portabilidade
RNF07	O sistema deverá ser desenvolvido na linguagem de programação Python, também utilizando o <i>framework</i> Django.	Arquitetura
RNF08	O sistema deverá apresentar um bom desempenho evitando constrangimento por parte dos usuários.	Desempenho
RNF09	O sistema deve persistir seus dados sobre o SGBD PostgreSQL.	Arquitetura

5.3 Modelagem do Banco de Dados

A modelagem do Banco de Dados é uma das etapas mais importantes no processo de desenvolvimento do sistema, é ela quem permite uma organização eficiente dos dados para um melhor desempenho da aplicação. Para isso, foi realizado um processo de levantamento e análise dos requisitos para criar um modelo conceitual que representasse as entidades, atributos e relacionamentos dos dados.

O modelo escolhido para a aplicação desenvolvida foi o DER (Diagrama Entidade-Relacionamento), que possibilitou uma visualização clara e precisa da estrutura do banco de dados. O diagrama DER do sistema pode ser visualizado na Figura 4.

A modelagem do banco de dados foi projetada para atender a todos os requisitos no início do desenvolvimento. Contudo, durante o processo foram feitos incrementos no escopo inicial do projeto, realizados de maneira iterativa. Esses acréscimos afetaram o modelo inicial do banco de dados. Contudo, essas alterações não causaram atrasos nas entregas de *sprints*. Com esse modelo, foi possível implementar um banco de dados eficiente e seguro, garantindo a integridade e consistência dos dados.

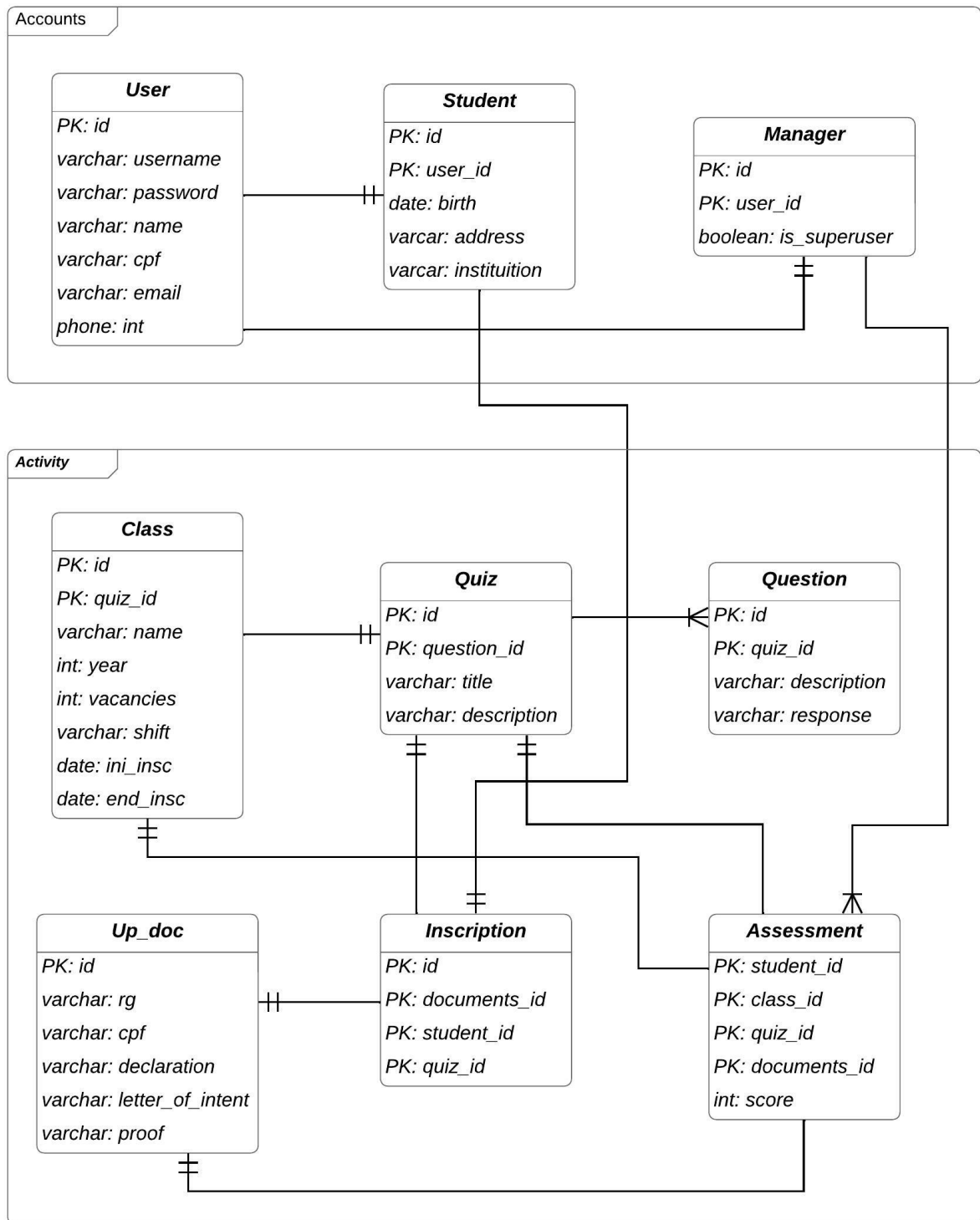


Figura 4 – Modelagem do Banco de Dados

5.4 Diagrama de Caso de Uso

O diagrama de casos de uso resume os detalhes dos usuários do sistema e as interações deles com o sistema, definido na *Unified Modeling Language* (UML), que tem como objetivo a elaboração da estrutura de projetos de *software*.

A Figura 5 demonstra os atores, definido pela UML como o usuários do sistema, interagindo com a ferramenta CPPPF, além de mostrar os casos de uso, que são tarefas ou funcionalidades realizadas pelo ator.

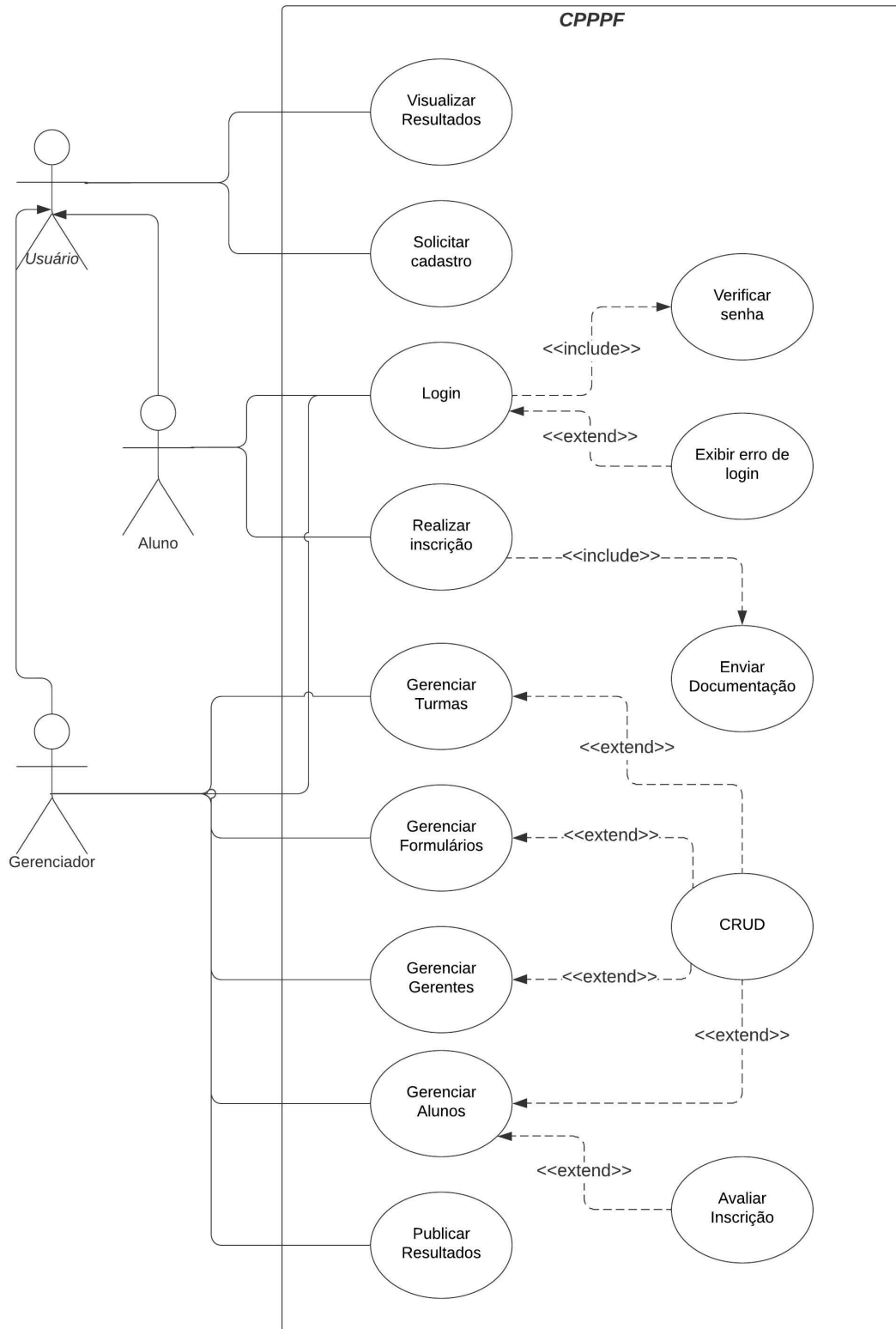


Figura 5 – Casos de Uso do Sistema

5.5 Usuários do Sistema

A plataforma foi projetada para atender às necessidades de dois tipos de usuários: o administrador do sistema e o usuário Aluno. O administrador tem acesso total ao sistema e pode gerenciar todo o processo seletivo em si, como turmas, formulários, alunos, gerentes e resultados. O gerente também tem acesso a todos os dados das tabelas por meio da visualização administrativa da *framework Django*. Já o usuário Aluno poderá efetuar seu cadastro no sistema, efetuar *login* e se inscrever em uma turma com inscrições abertas, bem como consultar os resultados das seleções.

Cada perfil de usuário tem uma interface de visualização específica, que será discutida em detalhes nas seções 5.5.1 e 5.5.2 do artigo.

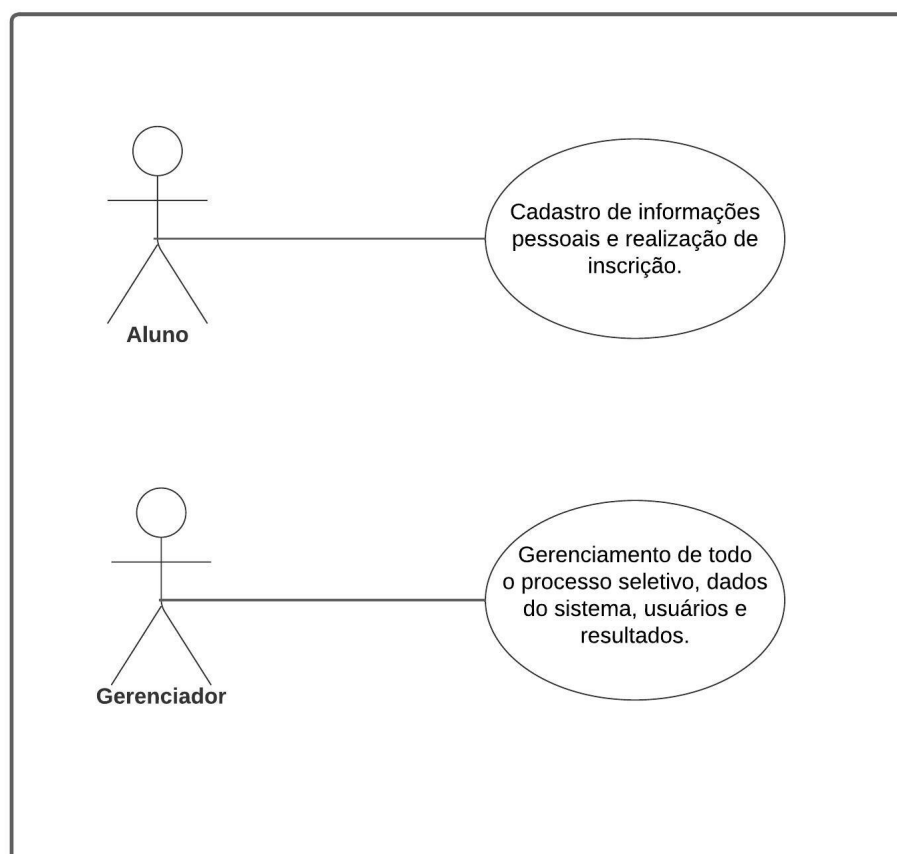


Figura 6 – Representação dos Usuários do Sistema

5.5.1 Usuário Aluno

O usuário Aluno ou Estudante terá poucas funcionalidades dentro da plataforma. Após efetuar seu cadastro e *login* no sistema, o aluno poderá efetuar sua inscrição na turma onde as datas de inscrições ainda estejam abertas. A Figura 7 ilustra a tela inicial de visualização do estudante.

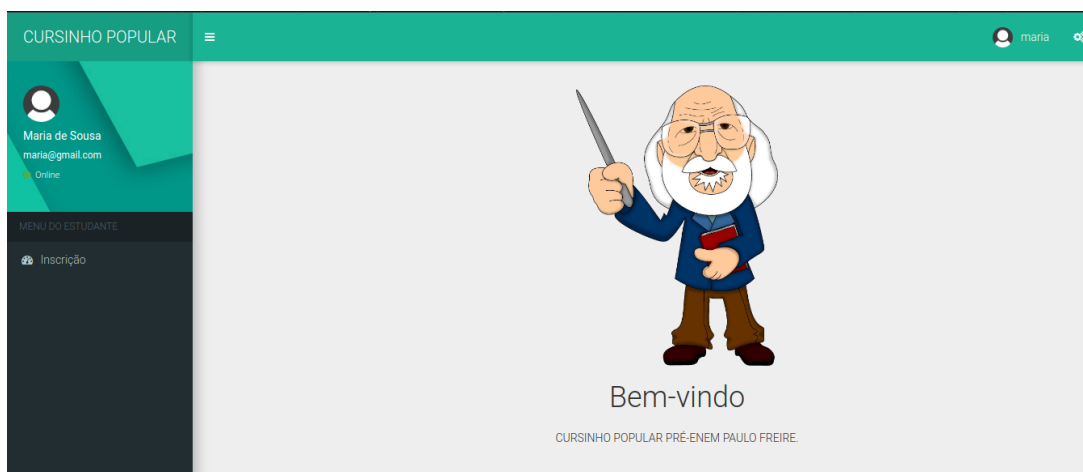


Figura 7 – Tela Inicial do Aluno

Ao clicar em “Inscrição”, no canto esquerdo da tela, o estudante poderá acessar as informações das turmas abertas, bem como efetuar sua inscrição em qualquer uma das turmas, como visto na Figura 8. O aluno só poderá efetuar uma inscrição em apenas uma turma. Após efetuar uma inscrição em uma determinada turma, o sistema não permitirá que o aluno faça inscrição em outra turma.

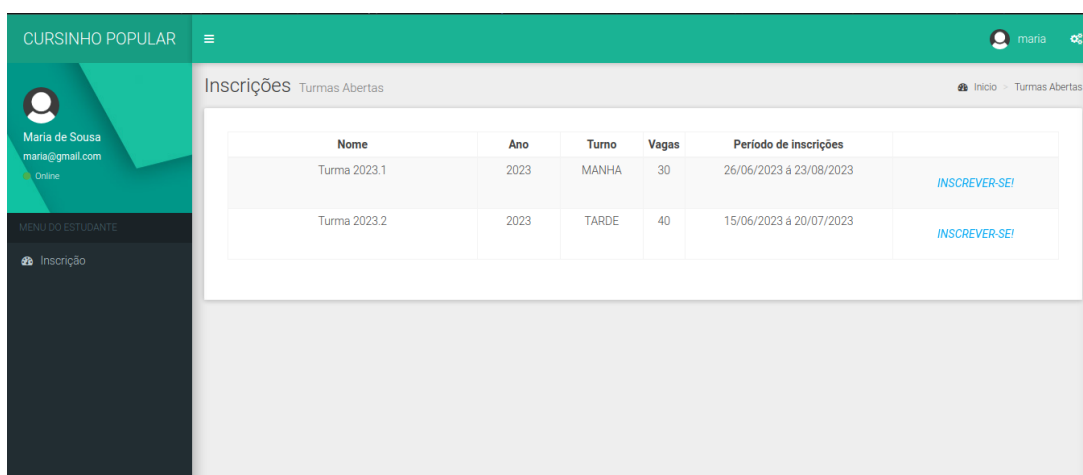


Figura 8 – Tela de Visualização de Turmas Abertas

Como visto anteriormente na Figura 8, clicando em “INSCREVER-SE!”, no canto direito, o aluno poderá efetuar sua inscrição na turma selecionada. O processo de inscrição é constituído por 3 etapas: cadastro de informações pessoais, envio de documentação e resposta ao questionário socioeconômico.

Qualquer usuário pode completar a primeira etapa ao se cadastrar no sistema. Os dados inseridos para cadastrar um usuário e efetuar o *login* já servem como dados pessoais da inscrição. Ao clicar em “INSCREVER-SE”, o usuário será redirecionado para a tela onde é descrito as etapas da inscrição. A Figura 9 demonstra a tela de inscrição do aluno.

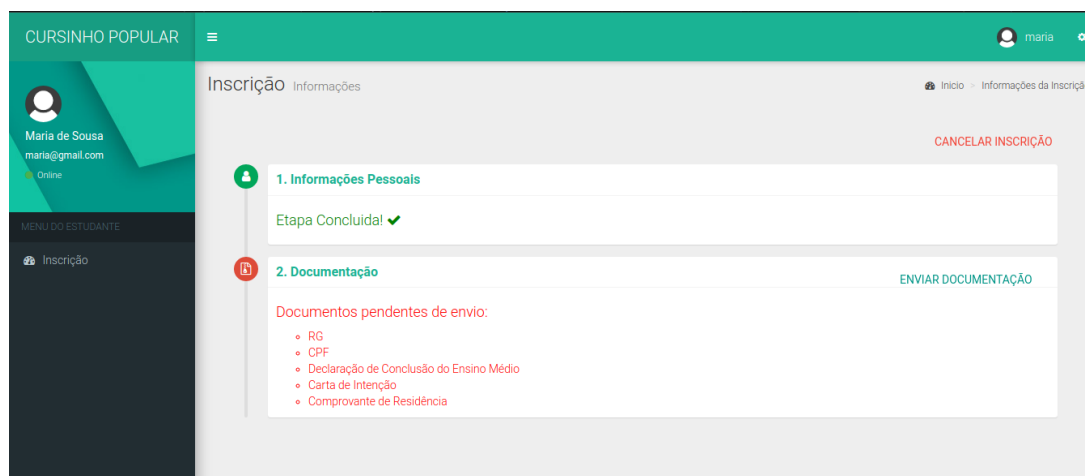
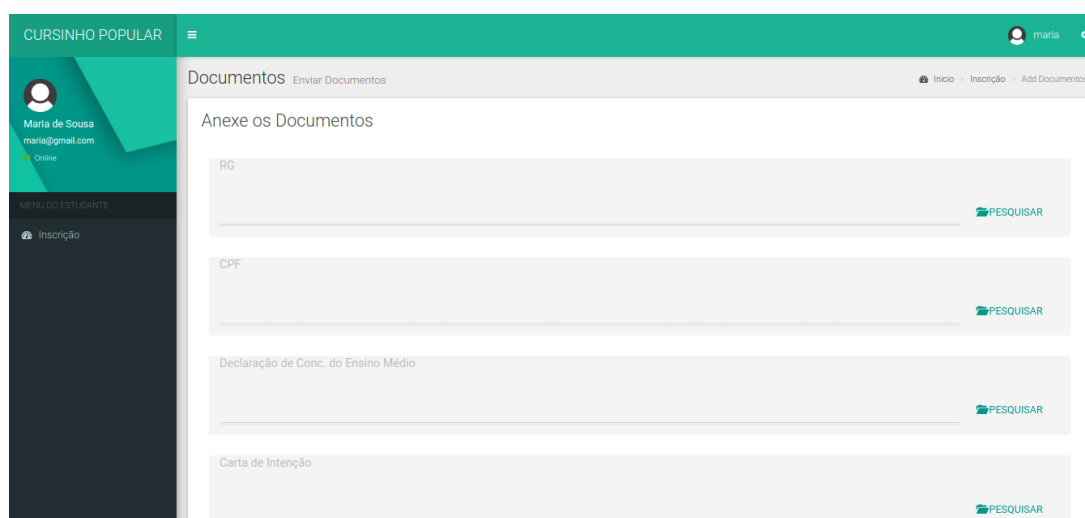


Figura 9 – Tela de Realizar Inscrição

Para concluir a segunda etapa da inscrição, o candidato deverá enviar os documentos necessários para realizar a inscrição. Para isso, basta clicar em “ENVIAR DOCUMENTAÇÃO”, fazer o *upload* de cada um dos documentos em seguida clicar em “Salvar”. A Figura 10 ilustra esse processo.

Figura 10 – Tela *Upload* de Documentação

Após concluir a segunda etapa, o sistema redirecionará o usuário para a tela das etapas da inscrição novamente mostrando que a etapa foi concluída, como mostra a Figura 11. Por fim, para concluir o processo de inscrição em uma turma, o usuário aluno deverá responder a um questionário socioeconômico. Para isso, basta clicar no local descrito na tela e o sistema abrirá uma nova tela com o questionário a ser respondido pelo estudante. Após responder todas as perguntas obrigatórias do formulário, o usuário aluno deverá clicar em “ENVIAR” para salvar e enviar suas respostas ao sistema. A Figura 12 mostra a tela contendo o questionário.

Ao concluir todas as etapas da inscrição o sistema mostrará que a inscrição foi concluída e também apresentará informações da inscrição, como ilustrado na Figura 13.

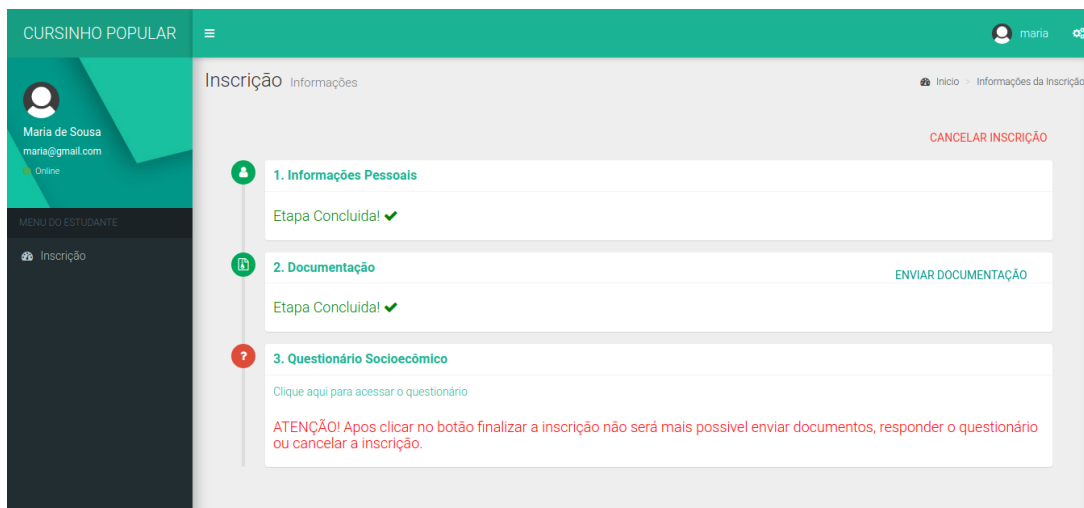


Figura 11 – Tela de Realizar Inscrição

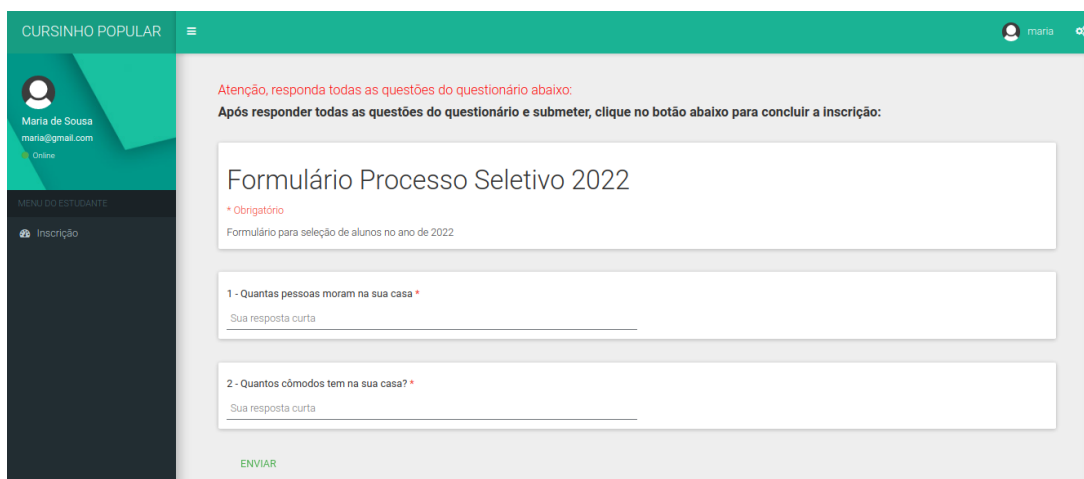


Figura 12 – Tela de Responder Questionário Socioeconômico

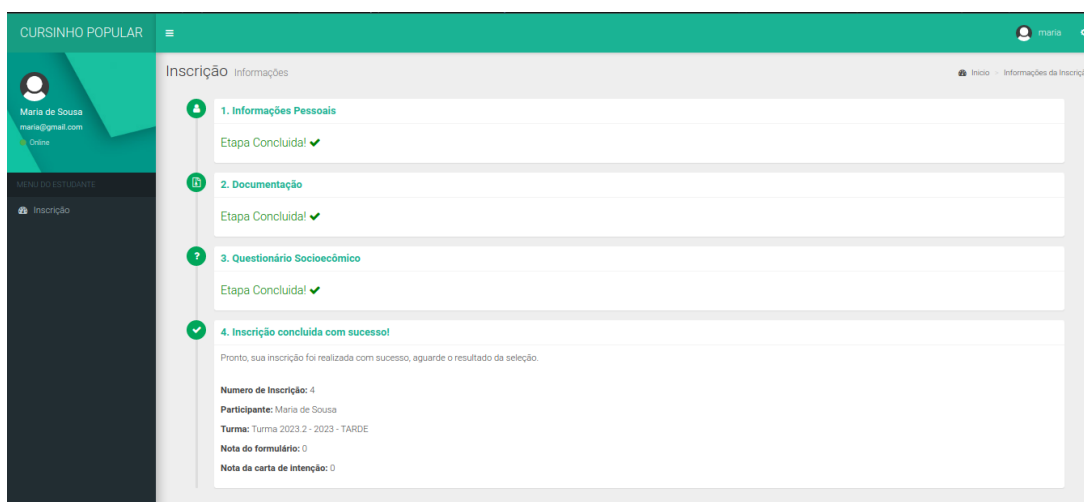


Figura 13 – Tela Inscrição Concluída

5.5.2 Usuário Gerenciador

O usuário gerenciador é aquele que possui acesso a todas as funcionalidades do sistema. Ele tem acesso à visualização administrativa do *Django*, que é uma interface gráfica que permite a edição dos dados armazenados no banco de dados. Essa visualização é fornecida pelo próprio *Django* e pode ser acessada por meio de um navegador da *web*. Nela, é possível adicionar, editar e excluir dados das tabelas do banco de dados. Além disso, o usuário administrador também tem a capacidade de gerenciar usuários do sistema, concedendo ou revogando acesso de outros usuários às diferentes funcionalidades da plataforma.

Na página inicial do sistema, o gerenciador tem a sua disposição as opções de gerenciamento de turma, formulários, gerentes e alunos, dispostos ao lado esquerdo da tela, como mostra a Figura 14.

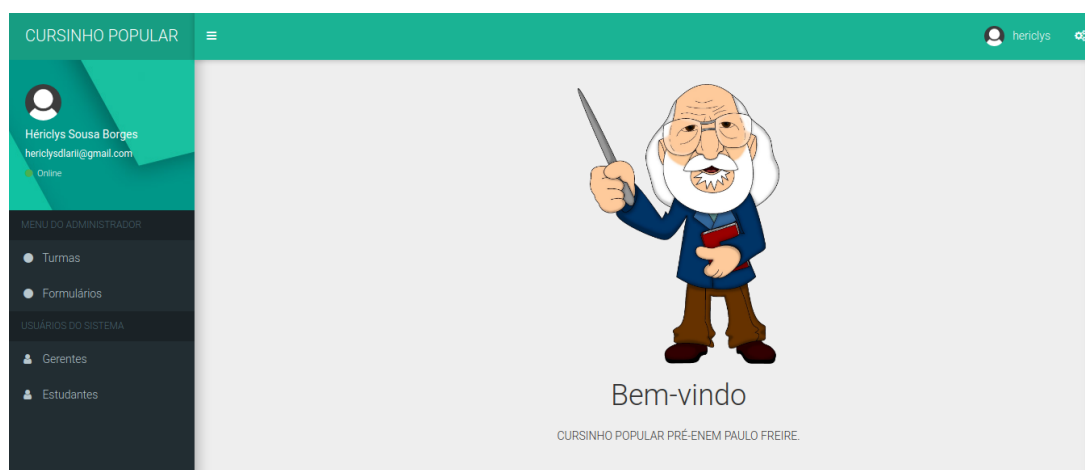


Figura 14 – Tela do Painel do Gerenciador do Sistema

5.5.2.1 Gerenciamento de Turmas

Ao clicar em “Turmas”, no canto esquerdo da tela, o gerenciador poderá visualizar as turmas criadas anteriormente, criar novas turmas, editar as informações das turmas, excluir turmas e avaliar inscritos de uma turma. A Figura 15 mostra como é constituída essa página.

Nessa tela é possível visualizar algumas informações das turmas cadastradas, tais como nome, ano de vigor, turno e período de inscrição. Na barra de “Ações”, ao lado direito das informações da turma, fica disposto ao gerente 4 opções, respectivamente: Visualizar, Editar, Avaliar Inscritos e Excluir. Ao clicar em “Visualizar”, símbolo do olho, o gerenciador poderá visualizar todos os dados da turma referente. Clicando em Editar, o mesmo poderá alterar dados da turma que desejar. O botão de “Avaliar Inscritos” serve para ter acesso aos alunos inscritos na turma, bem como avaliar a inscrição dos mesmos. Por fim, o último botão, símbolo da lixeira, serve para excluir uma turma.

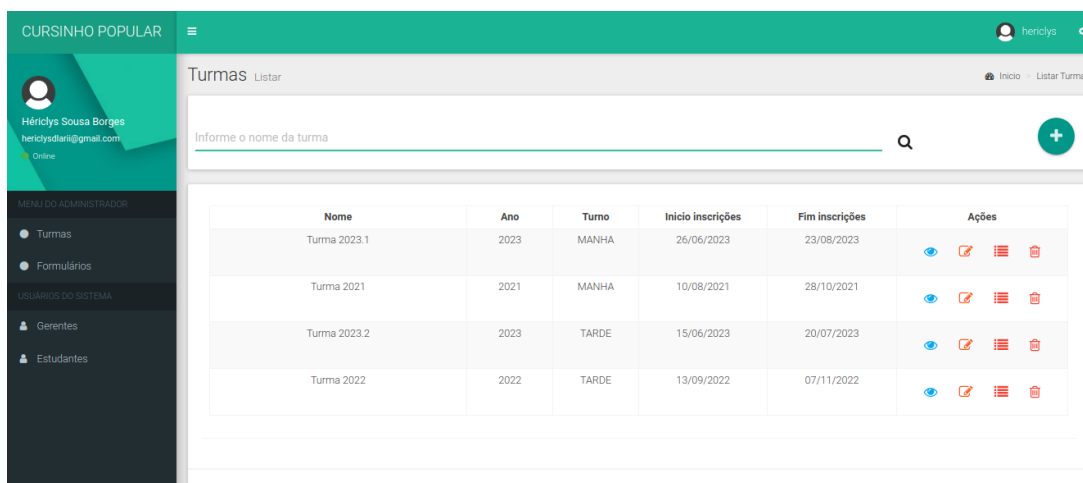


Figura 15 – Tela de Gerenciamento de Turmas

Para que os alunos possam visualizar as turmas abertas o gerenciador terá que cadastrar uma turma, o processo ocorre de forma simples e intuitiva. Para efetuar o cadastro de turmas, o gerente deve clicar no menu “Turmas” e, logo após, no botão “+” e cadastrar as informações necessárias para sua nova turma e clicar em “Salvar”, como referenciado na Figura 16.

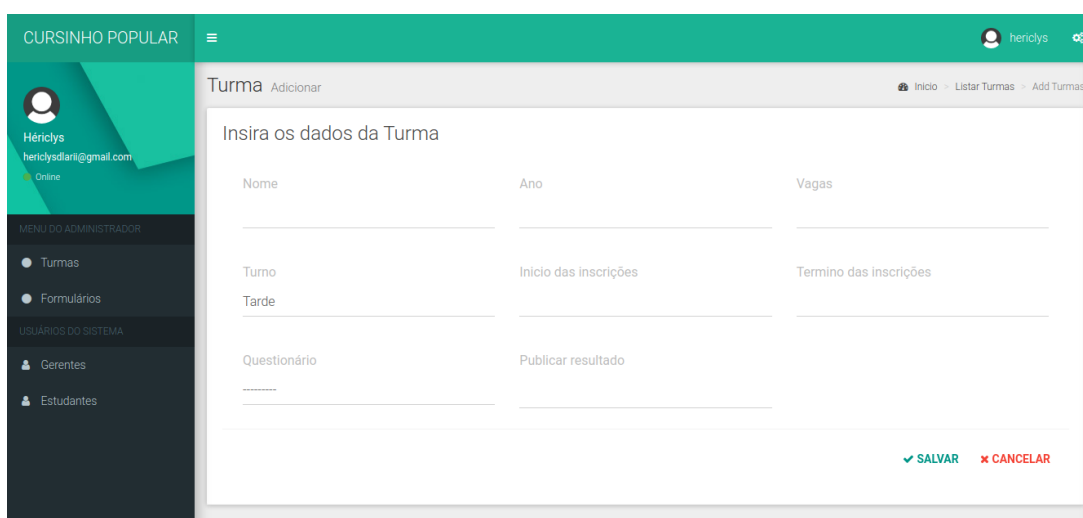
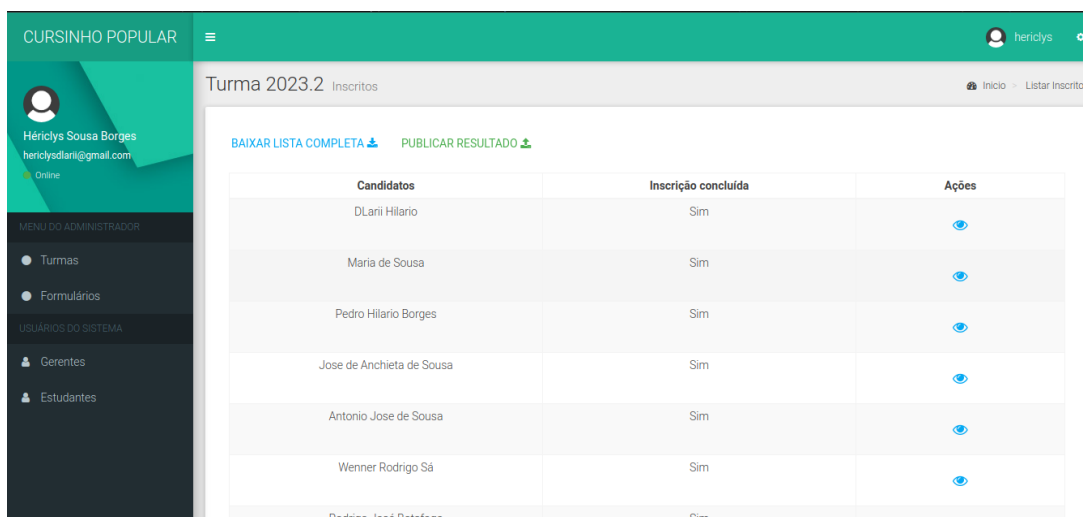


Figura 16 – Tela de Criar Nova Turma

Uma vez que o gerente criou uma turma, automaticamente a turma já fica disponível para alunos se inscreverem e concorrerem às vagas destinadas e ela. Terminado o período de inscrições, o gerente poderá visualizar e avaliar os candidatos clicando em “Avaliar Inscritos” disposto na barra de “Ações”, na tela de Gerenciamento de Turmas. Nesta tela, é possível visualizar os detalhes de todos alunos que estão previamente inscritos na turma selecionada. A Figura 17 exemplifica os elementos contidos nesta seção. Nessa página o gerente poderá visualizar e avaliar a inscrição de cada candidato clicando no botão de ação ao lado direito das informações dos candidatos.

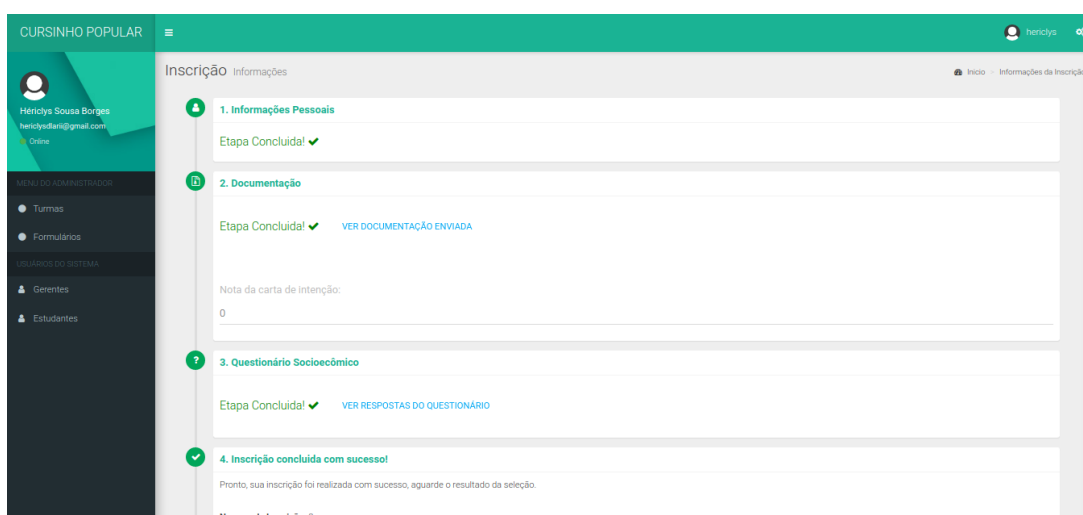


Candidatos	Inscrição concluída	Ações
DLanii Hilario	Sim	
Maria de Sousa	Sim	
Pedro Hilario Borges	Sim	
Jose de Anchieta de Sousa	Sim	
Antonio Jose de Sousa	Sim	
Wenner Rodrigo Sá	Sim	
Rodrigo José Botafogo	Sim	

Figura 17 – Tela de Gerenciamento de Turma Específica

A Figura 18 exemplifica a tela que aparece logo em seguida. Nesta tela, o gerenciador poderá visualizar a documentação do candidato clicando em “VER DOCUMENTAÇÃO ENVIADA”. A Figura 19 mostra a tela de visualização dos documentos, onde o gerente pode optar por visualizar e também fazer o *download* da documentação. Ainda na sessão de “Documentação”, abaixo do botão para visualizar os documentos, é disposto um campo onde servirá para o gerente dar uma nota para a carta de intenção do aluno, que poderá ser visualizada entre os documentos enviados pelo candidato.

Abaixo da área de documentação, na área do questionário, o gerenciador poderá visualizar todas as respostas do questionário do candidato clicando no botão “VER RESPOSTAS DO QUESTIONÁRIO”. Ao clicar nesse botão será disposto ao gerente a tela descrita na Figura 20, onde o avaliador poderá visualizar as respostas do candidato e dar uma nota a cada uma das respostas ao questionário.



Inscrição Informações

- 1. Informações Pessoais**
Etapa Concluída! ✓
- 2. Documentação**
Etapa Concluída! ✓ [VER DOCUMENTAÇÃO ENVIADA](#)
Nota da carta de intenção:
0
- 3. Questionário Socioeconômico**
Etapa Concluída! ✓ [VER RESPOSTAS DO QUESTIONÁRIO](#)
- 4. Inscrição concluída com sucesso!**
Pronto, sua inscrição foi realizada com sucesso, aguarde o resultado da seleção.
Numero de Inscrição: 3

Figura 18 – Tela de Avaliar Inscrição



Figura 19 – Tela de Visualização de Documentos

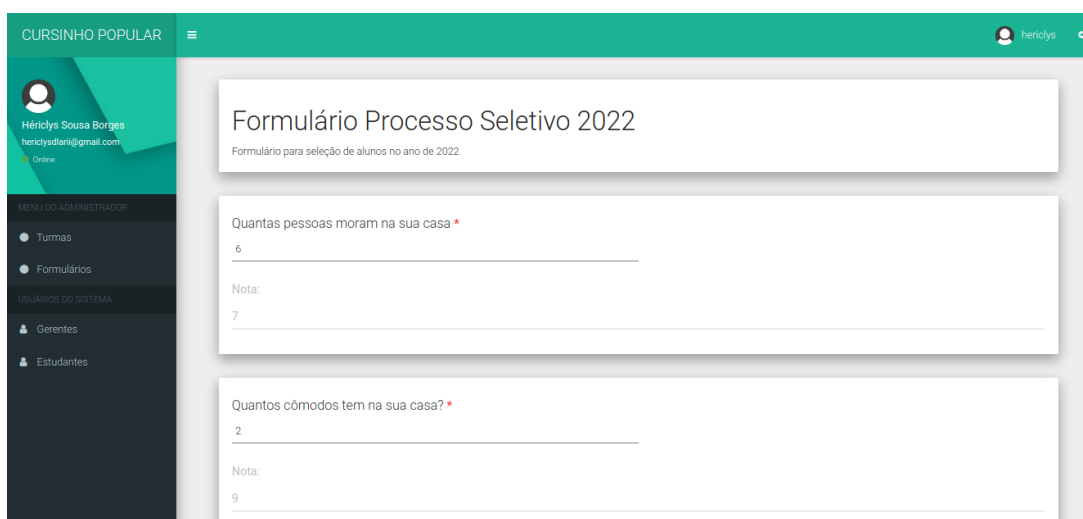


Figura 20 – Tela de Avaliar Respostas do Formulário

Ainda na página de gerenciamento de uma turma, referenciada anteriormente pela Figura 17, o usuário gerenciador poderá visualizar e/ou baixar um documento em formato PDF que o sistema gera com o *ranking* da relação dos alunos aprovados e reprovados após o processo de avaliação dos inscritos, referenciado por a Figura 21. Para visualizar o documento, o gerente deverá clicar em “BAIXAR LISTA COMPLETA”, acima das informações dos inscritos na turma.

A nota final do aluno é composta por uma média da nota da carta de intenção com a nota geral do questionário socioeconômico, feita automaticamente pelo sistema. Após o usuário gerenciador concluir as avaliações, o sistema calcula a média e já a disponibiliza para visualização.

Ao lado do botão de visualização do arquivo tem o botão de “PUBLICAR RESULTADOS”, que serve para tornar público o arquivo em PDF contendo as informações dos resultados do processo seletivo. Após clicar nesse botão, automaticamente o documento ficará disponível para qualquer usuário visualizar na página inicial do sistema na aba

de “Ver Resultados”. Após clicar nesse botão qualquer usuário do sistema terá acesso a todos os resultados publicados pelos gerentes do cursinho, como descrito na tela contida na Figura 22.



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CAMPUS SENADOR HELVIDIO NUNES DE BARROS - CSHNB
CURSINHO POPULAR PROFESSOR PAULO FREIRE

RESULTADO
Turma 2023.2

Nome	Nota	Situação
DLarii Hilario	13.5	Aprovado
Ana Maria de Sousa	12.0	Aprovado
Maria de Sousa	12.5	Aprovado
Antonio Jose de Sousa	12.0	Aprovado
Irna Luzia Holanda	11.5	Aprovado
Ronaldo Antonio da Silva	10.0	Aprovado
Pedro Hilario Borges	10.5	Aprovado
Jose de Anchieta de Sousa	10.0	Aprovado
Rodrigo José Botafogo	11.5	Aprovado
Rayssa Soares Cabral	11.0	Aprovado
Ariely Lacerda de Sousa	10.5	Aprovado
Wenner Rodrigo Sá	10.5	Aprovado
Maria Clara Vitoria Sousa B	10.0	Aprovado
Lucas Soares de Sousa	6.0	Reprovado
Italo Sousa Borges	4.5	Reprovado

Figura 21 – PDF Gerado Pelo Sistema

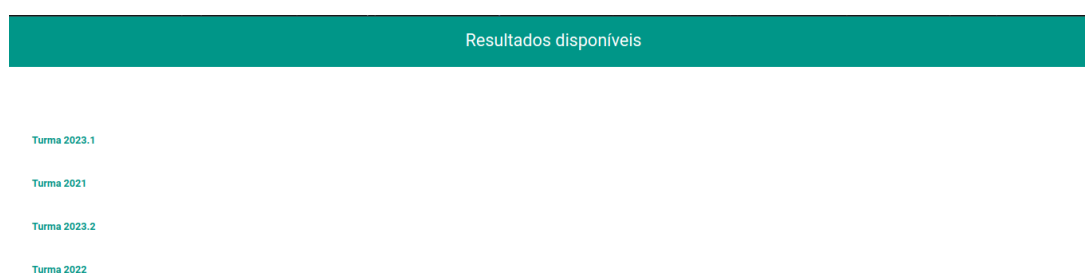


Figura 22 – Tela de Resultados

5.5.2.2 Gerenciamento de Formulários

Antes de criar uma turma, é necessário criar um formulário para ser usado no processo seletivo de uma turma. Para isso, o gerente deve clicar em “Formulários” no Menu do

Administrador. Em seguida será mostrado ao gerente a tela descrita na Figura 23, onde contém todos os formulários previamente cadastrados no sistema. Nesta tela é possível criar, editar ou excluir um formulário. É importante frisar que a gerencia também pode reaproveitar um questionário de outros processos seletivos. Para criar um novo formulário, deve-se clicar em “+NOVO FORMULÁRIO” e informar um título, uma descrição e, em seguida, formular as perguntas e respostas do questionário, como mostra a Figura 24. É importante frisar que a equipe gerenciadora também poderá escolher se as questões serão de respostas obrigatórias. Também contará com algumas opções de respostas, como: múltipla escolha, resposta curta, resposta longa ou *checklist*.

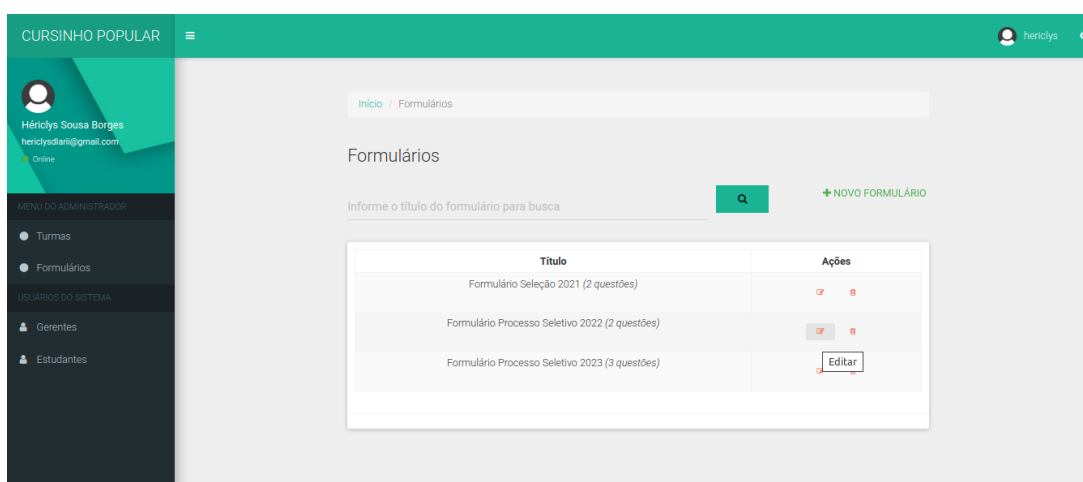


Figura 23 – Tela de Gerenciamento de Formulários

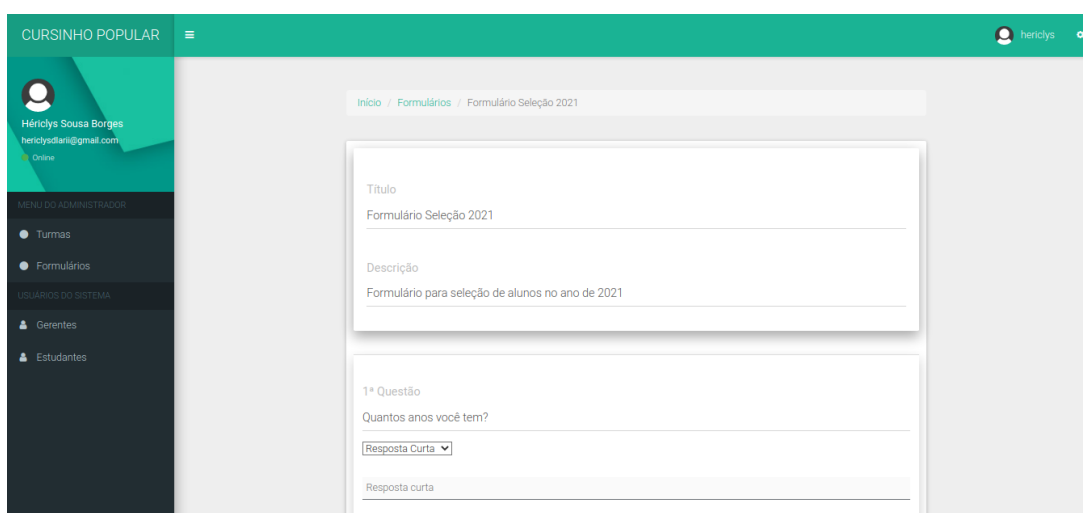


Figura 24 – Tela de Criar Formulário

5.5.2.3 Gerenciamento de Gerentes

Um gerente poderá designar outros usuários para fazer parte do grupo de gerenciadores do sistema. Clicando em “Gerentes”, na aba de Usuários do Sistema, um gerenciador terá

acesso aos gerentes cadastrados no sistema, estes serão os responsáveis por gerenciar todo o processo de seleção, bem como gerenciar alunos inscritos, turmas, formulários e também disponibilizar os resultados dos processos seletivos. Nesta tela também é possível visualizar, editar e excluir, respectivamente, um usuário gerenciador através dos botões descritos na barra de “Ações”, ao lado direito das informações do usuário. A Figura 25 mostra a página de gerenciamento de gerentes.

Para cadastrar um novo gerente deve-se clicar no botão “+” e inserir os dados do novo gerente e clicar em “SALVAR”, como visto na Figura 37. Feito isso, o sistema redirecionará o gerenciador para a página de gerenciamento de gerentes novamente, onde já será possível visualizar o novo gerente cadastrado.

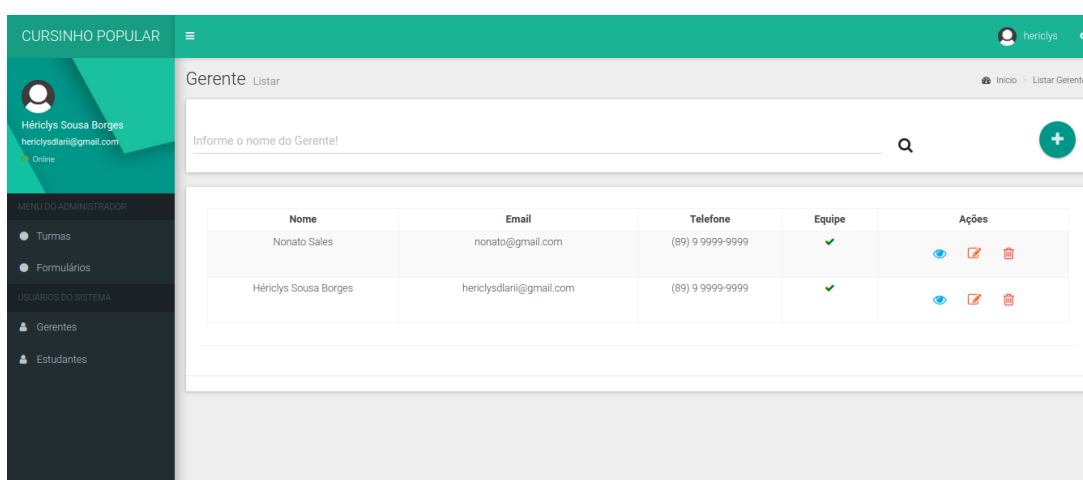


Figura 25 – Tela de Gerenciamento de Gerentes

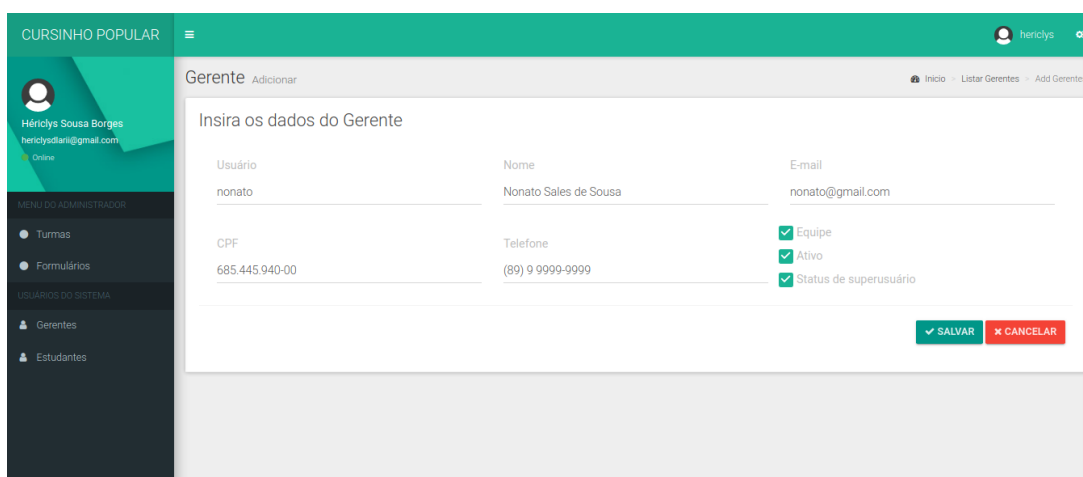


Figura 26 – Tela de Criar Novo Gerente

5.5.2.4 Gerenciamento de Alunos

Na última opção do menu dos usuários do sistema, clicando em “Estudantes”, os gerenciadores poderão visualizar todos os alunos cadastrados no sistema independente da

turma cadastrada. Nesta tela ele poderá visualizar, editar e excluir dados dos estudantes. Caso necessário, nessa página também é possível adicionar um novo aluno.

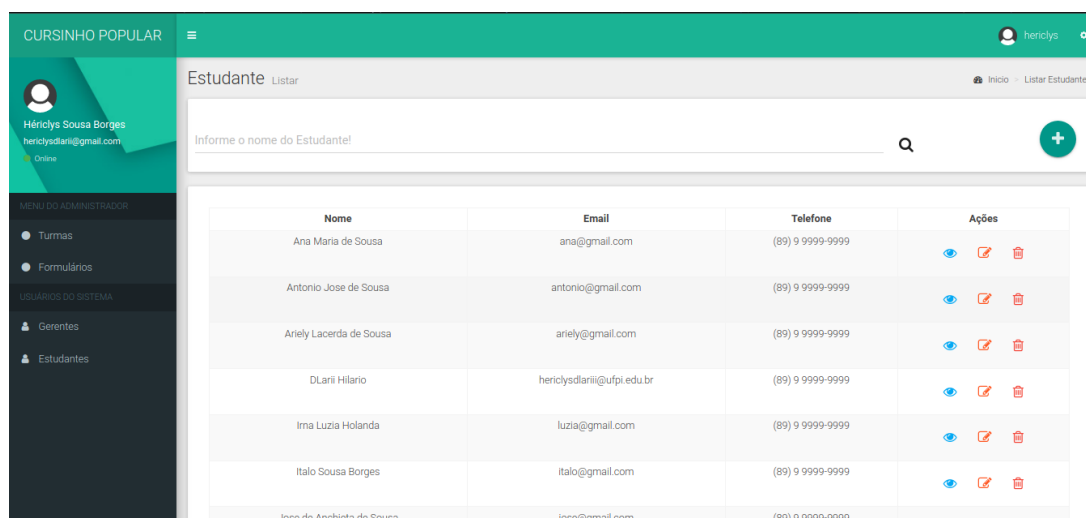


Figura 27 – Tela de Gerenciar Alunos

6 Resultados e Discussões

Este capítulo nós apresentamos os resultados obtidos na avaliação de usabilidade de *software* e Interação Humano Computador (IHC) alcançada através de uma pesquisa realizada com um grupo de pessoas compostas por estudantes universitários de diferentes instituições do estado e pessoas julgadas como possíveis candidatos a alunos do cursinho, onde foram coletadas respostas de 18 participantes.

Antes de responder ao questionário de avaliação, os entrevistados tiveram acesso a plataforma acessando um *link* disponibilizado através de uma lista de transmissão por meio do aplicativo *WhatsApp*, onde puderam testar as funcionalidades da plataforma de forma individual.

Após o contato com a aplicação, de maneira análoga, foi disponibilizado o *link* de um formulário criado através da ferramenta *Google Forms*¹ coletando dados sobre a usabilidade, facilidade de navegação, funcionalidades e outros aspectos relevantes. A partir das respostas obtidas, foram realizadas análises e discussões visando identificar pontos positivos e negativos do sistema, bem como propor possíveis melhorias para aprimorá-lo.

O questionário de avaliação contém dez (10) perguntas acerca da interface e funcionalidades presentes no sistema. As perguntas estão descritas na Tabela 4 abaixo:

Tabela 4 – Perguntas do Questionário de Avaliação

Questão	Pergunta
1	A plataforma possui uma interface simples e amigável?
2	Você sentiu facilidade ao navegar dentro da plataforma?
3	Eu acho o sistema desnecessariamente complexo.
4	Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5	Eu acho que as várias funções do sistema estão muito bem integradas.
6	Eu acho que o sistema apresenta muita inconsistência.
7	Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8	Eu achei o sistema difícil de usar.
9	Eu me senti confiante ao usar o sistema.
10	Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Agregamos ao questionário de avaliação a metodologia da Escala *Likert* para mensurar o grau de satisfação (ou insatisfação) dos usuários em relação à usabilidade do sistema proposto. As respostas foram coletadas em uma escala de 1 a 5, onde 1 significa “Discordo Totalmente” e 5 significa “Concordo Totalmente”. Os critérios de avaliação foram efetividade, eficiência e satisfação do usuário. A pesquisa mostrou que a maioria dos participantes apresentou uma aceitação positiva em relação à usabilidade do *software*.

¹ <https://www.google.com/intl/pt-BR/forms/about/>

Com a execução da pesquisa, foi possível concluir que a aceitação dos universitários que participaram do processo de avaliação em relação a usabilidade do *software* foi em sua maioria positiva, na qual, os dados colhidos são apresentados na Tabela 5. Como pode ser observado, a maioria dos itens do questionário alcançaram um nível de satisfação igual ou superior a 70%, sendo que esta foi a métrica definida inicialmente para que a pesquisa possua um resultado satisfatório.

Tabela 5 – Resultados da Pesquisa de Usabilidade

Pergunta	Discordo Totalmente	Discordo	Neutro	Concordo	Concordo Totalmente
A plataforma possui uma interface simples e amigável?	0%	0%	11,1%	44,4%	44,4%
Você sentiu facilidade ao navegar dentro da plataforma?	0%	0%	11,1%	56,6%	33,3%
Eu acho o sistema desnecessariamente complexo.	44,4%	16,7%	5,6%	27,8%	5,6%
Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	38,9%	33,3%	16,7%	11,1%	0%
Eu acho que as várias funções do sistema estão muito bem integradas.	0%	5,6%	0%	70,6%	23,5%
Eu acho que o sistema apresenta muita inconsistência.	22,2%	50%	11,1%	16,7%	0%
Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	0%	0%	27,8%	27,8%	44,4%
Eu achei o sistema difícil de usar.	50%	16,7%	16,7%	16,7%	0%
Eu me senti confiante ao usar o sistema.	0%	0%	5,6%	44,4%	50%
Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	27,8%	44,4%	22,2%	0%	5,6%

Após a análise das respostas do questionário de usabilidade, algumas sugestões podem ser consideradas a fim de melhorar a usabilidade do sistema, como:

1. Simplificação da interface do usuário, tornando-a mais intuitiva e fácil de usar;
2. Melhoria da consistência do sistema, de forma a minimizar as dificuldades encontradas pelos usuários;
3. Criação de tutoriais ou outras ferramentas de ajuda, para facilitar o aprendizado das funcionalidades do software.

7 Conclusão

Os sistemas *web* apresentam uma forma eficaz de simplificar tarefas trabalhosas e de grande consumo de tempo, esforço físico e mental. Tendo isso em vista, o presente trabalho abordou o desenvolvimento do CPPPF, um sistema *web* para o gerenciamento de processo seletivo de vagas destinadas a alunos de baixa renda, com o intuito de facilitar o cotidiano dos usuários e garantir o controle automatizado das atividades realizadas manualmente.

Para avaliar o sistema, foi aplicado um formulário a alunos universitários de diferentes universidades para coletar dados sobre sua usabilidade, facilidade de navegação, funcionalidades e outros aspectos relevantes. As respostas obtidas foram analisadas e discutidas visando identificar pontos positivos e negativos do sistema e propor possíveis melhorias para aprimorá-lo.

Concluiu-se que o sistema obteve uma boa aceitação, a maioria concordou que o sistema é fácil de usar, que as funções estão bem integradas e que se sentiram confiantes ao usar o sistema. No entanto, alguns participantes acharam o sistema desnecessariamente complexo e difícil de usar. Por fim, implementações de novas funcionalidades podem tornar o processo de gerenciamento do *site* mais amplo e trazer mais eficiência e interação aos usuários finais.

7.1 Trabalhos Futuros

Existem vários recursos que podem ser aprimorados, como interface, funcionalidades, desempenho, entre outros. Como sugestões de trabalhos futuros, propõem-se algumas melhorias para o sistema, como:

- Simplificar a interface do usuário, tornando-a mais intuitiva e fácil de usar;
- Criação de tutoriais ou outras ferramentas de ajuda, para facilitar o aprendizado das funcionalidades do software;
- Implementar novo login para acesso dos professores;
- Implementar gerenciamento de simulados.

Referências

- ARAFAT, M. et al. Analisis dan perancangan website sebagai sarana informasi pada lembaga bahasa kewirausahaan dan komputer akmi baturaja menggunakan php e mysql. *Jurnal Ilmiah MATRIK*, v. 19, n. 1, p. 1–10, 2017. Citado 2 vezes nas páginas 27 e 29.
- BARROS, M. de O. Sistema de provas online com desenvolvimento em python com django. 2013. Citado na página 23.
- BERNERS-LEE, T. J. The world-wide web. *Computer networks and ISDN systems*, Elsevier, v. 25, n. 4-5, p. 454–459, 1992. Citado na página 19.
- BORTOLOSSI, H. J. Criando conteúdos educacionais digitais interativos em matemática e estatística com o uso integrado de tecnologias: Geogebra, javaview, html, css, mathml e javascript. *Revista do Instituto GeoGebra Internacional de São Paulo. ISSN 2237-9657*, v. 1, n. 1, 2012. Citado 2 vezes nas páginas 24 e 25.
- CALDEIRA, L. M. *Criação de um Website dinâmico e versátil dada uma base de dados remota e recorrendo à linguagem de script PHP*. Tese (Doutorado), 2015. Citado 2 vezes nas páginas 27 e 29.
- CARVALHO, V. *PostgreSQL: Banco de dados para aplicações web modernas*. [S.l.]: Editora Casa do Código, 2017. Citado na página 26.
- COULOURIS, G. et al. *Sistemas Distribuídos-: Conceitos e Projeto*. [S.l.]: Bookman Editora, 2013. Citado na página 20.
- DATE, C. J. *Introdução a sistemas de bancos de dados*. [S.l.]: Elsevier Brasil, 2004. Citado na página 25.
- ELMASRI, R. et al. *Sistemas de banco de dados*. [S.l.]: Pearson Addison Wesley São Paulo, 2005. Citado na página 25.
- GARRIDO, J. C. Arquitectura y diseño de sistemas web modernos. *InforMAS, Revista de Ingeniería Informática del CIIRM*, n. 1, 2004. Citado na página 19.
- GRANNELL, C. *The essential guide to CSS and HTML web design*. [S.l.]: Apress, 2008. Citado na página 24.
- GUTIÉRREZ, J. J. ¿ qué es un framework web. Available in: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf Accessed May, v. 12, 2014. Citado na página 19.
- JUNIOR, L. A. Z. *Sistemas de informação baseados na tecnologia web: um estudo sobre seu desenvolvimento*. Tese (Doutorado) — Universidade de São Paulo, 2003. Citado na página 20.
- KATO, D. S. O papel dos cursinhos populares nos acessos e mudanças de perspectivas de seus participantes. *Cadernos CIMEAC*, v. 1, n. 1, p. 5–24, 2015. Citado na página 15.
- LAMSAL, K. *Projetando e desenvolvendo um site dinâmico usando php*. 2020. Citado 2 vezes nas páginas 28 e 29.

- MILANI, A. *PostgreSQL-Guia do Programador*. [S.l.]: Novatec Editora, 2008. Citado na página 26.
- MITRULIS, E.; PENIN, S. T. d. S. Pré-vestibulares alternativos: da igualdade à equidade. *Cadernos de Pesquisa*, SciELO Brasil, v. 36, n. 128, p. 269–298, 2006. Citado na página 15.
- MUSLIHUDIN, M.; LARASATI, A. O design do novo sistema de aplicação de admissão de alunos na estação pringsewu usando php e mysql. *TAM Journal (Technology Acceptance Model)*, v. 3, p. 32–39, 2017. Citado 2 vezes nas páginas 27 e 29.
- NASCIMENTO, A. do. Universidade e cidadania: o movimento dos cursos pré-vestibulares populares. *Lugar Comum*, v. 17, p. 45–citation_lastpage, 2002. Citado na página 17.
- NATIONS, D. What exactly is a web application? *Improve your understanding of web-based application programs*. *Lifewire*, v. 22, 2018. Disponível em: <<https://www.lifewire.com/what-is-a-web-application-3486637>>. Citado na página 19.
- NURHASANAH SITI E RUSDAN, M. Desenvolvimento de front-end em aplicativos de turismo e viagens usando python e django framework em pt. industri telekomunikasi indonesia. *Journal of ICT (Informática e Tecnologia de Comunicação)*, v. 2, n. 1, p. 31–38, 2020. Citado 2 vezes nas páginas 28 e 29.
- OLIVEIRA, D. d. P. R. d. Sistema de informações gerenciais: estratégicas, táticas, operacionais. 11ª edição. *São Paulo: Atlas*, 2007. Citado na página 20.
- OLUWATOSIN, H. S. Client-server model. *IOSR J Comput Eng (IOSR-JCE)*, v. 16, n. 1, p. 67, 2014. Citado na página 18.
- PIETRO, F. E. de; PETRUCELLI, E. E. Website fotográfico desenvolvido em php. *Revista Interface Tecnológica*, v. 16, n. 1, p. 254–265, 2019. Citado 2 vezes nas páginas 28 e 29.
- ROSSUM, G. *Python Reference Manual*. NLD, 1995. Citado na página 21.
- ROSSUM, G. V. et al. Python programming language. In: *USENIX annual technical conference*. [S.l.: s.n.], 2007. v. 41, p. 36. Citado na página 22.
- SILVA, F. M. G. M. d. S. *Management Information System (Django)*. Dissertação (Mestrado), 2014. Citado na página 23.
- TANENBAUM, A. S.; FILHO, N. M. *Sistemas operacionais modernos*. [S.l.]: Prentice-Hall, 1995. Citado na página 18.
- Tiwari, U. et al. Design of python based lost and found website for college campus. In: *2019 International Conference on Power Electronics, Control and Automation (ICPECA)*. [S.l.: s.n.], 2019. p. 1–5. Citado 2 vezes nas páginas 28 e 29.
- TORRES, V. M. Html e seus componentes. *Revista Ada Lovelace*, v. 2, p. 99–101, 2018. Citado na página 24.
- VAINIKKA, J. Full-stack web development using django rest framework and react. *Metropolia Ammattikorkeakoulu*, 2018. Citado 2 vezes nas páginas 28 e 29.

WINCKLER, M.; PIMENTA, M. S. Avaliação de usabilidade de sites web. *ESCOLA REGIONAL de Informática. Porto Alegre: SBC*, p. 1–54, 2002. Citado 2 vezes nas páginas 19 e 20.

Wu, J. H. e H. Qu e L. Projeto e implementação de plataforma de armazenamento em nuvem privada baseada em openstack. In: *2015 IEEE International Conference on Smart City / SocialCom / SustainCom (SmartCity)*. [S.l.: s.n.], 2015. p. 1098–1101. Citado 2 vezes nas páginas 27 e 29.

YOSHIURA, V. T. *Desenvolvimento e implantação de um sistema web para monitoramento da rede de atenção em saúde mental*. Tese (Doutorado) — Universidade de São Paulo, 2015. Citado na página 25.

ZAGO, N. Cursos pré-vestibulares populares: limites e perspectivas. *Perspectiva*, v. 26, n. 1, p. 149–174, 2008. Citado na página 17.

ZHOU, Y. *PROJECT REPORT OF BUILDING COURSE MANAGEMENT SYSTEM BY DJANGO FRAMEWORK*. Tese (Doutorado) — ZHEJIANG UNIVERSITY, 2010. Citado 2 vezes nas páginas 27 e 29.

Apêndices

APÊNDICE A – Resultados da Avaliação.

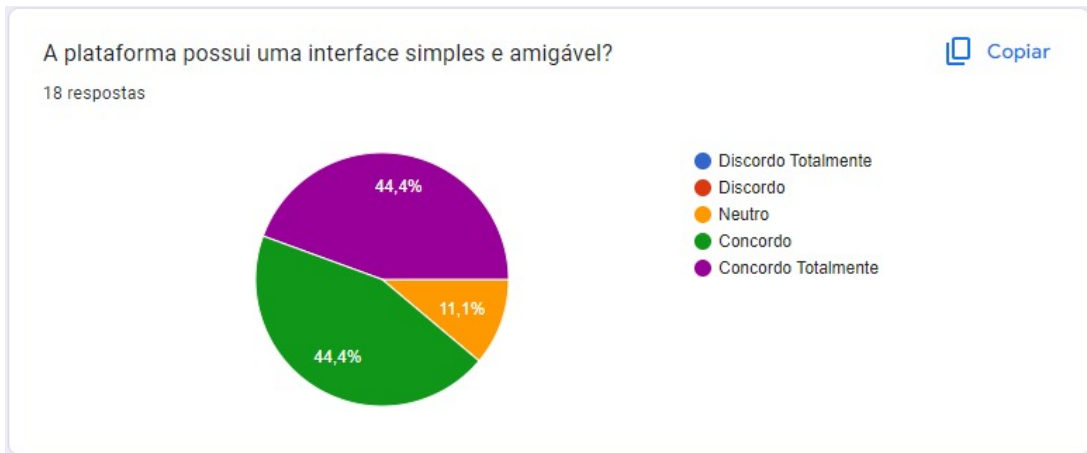


Figura 28 – Questão 1

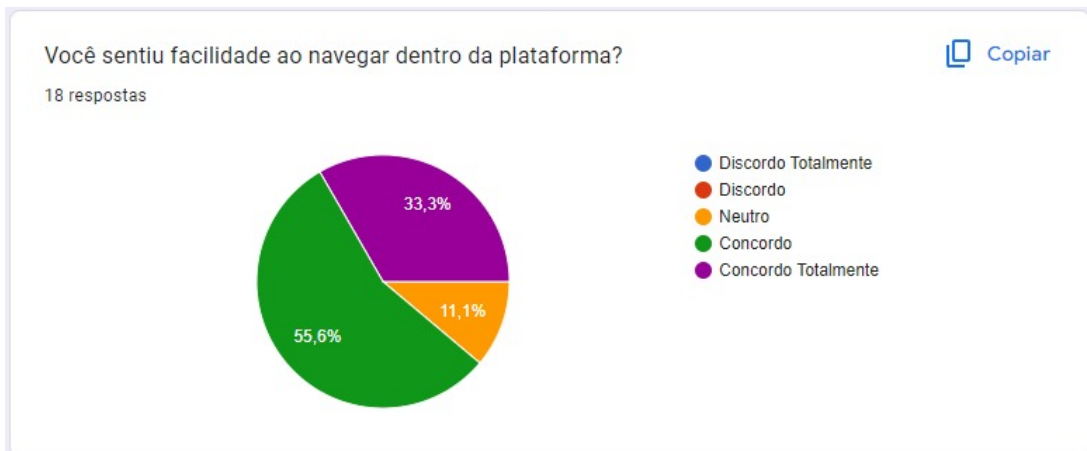


Figura 29 – Questão 2

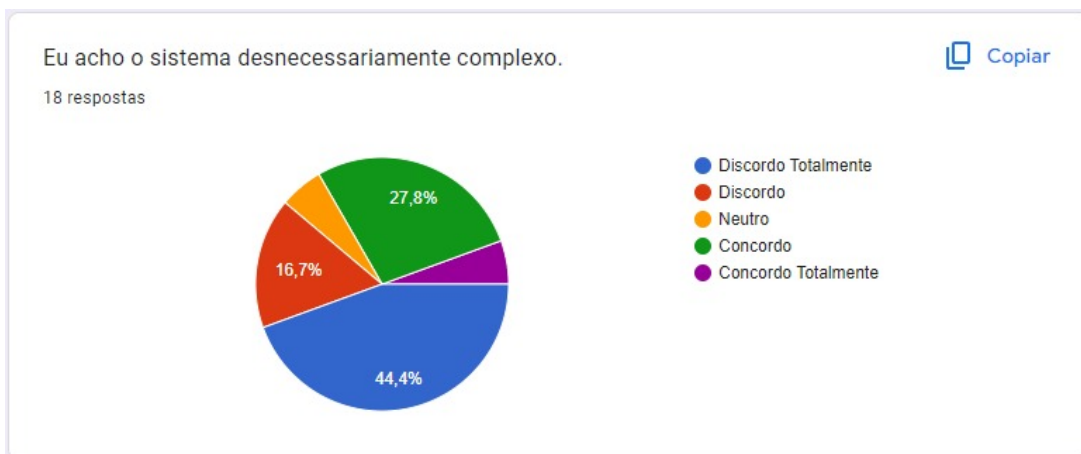


Figura 30 – Questão 3



Figura 31 – Questão 4



Figura 32 – Questão 5



Figura 33 – Questão 6

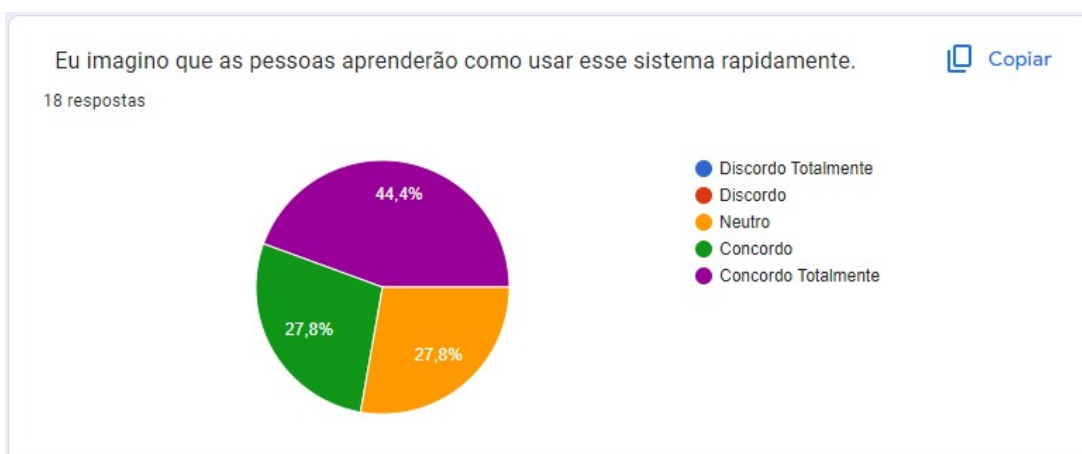


Figura 34 – Questão 7

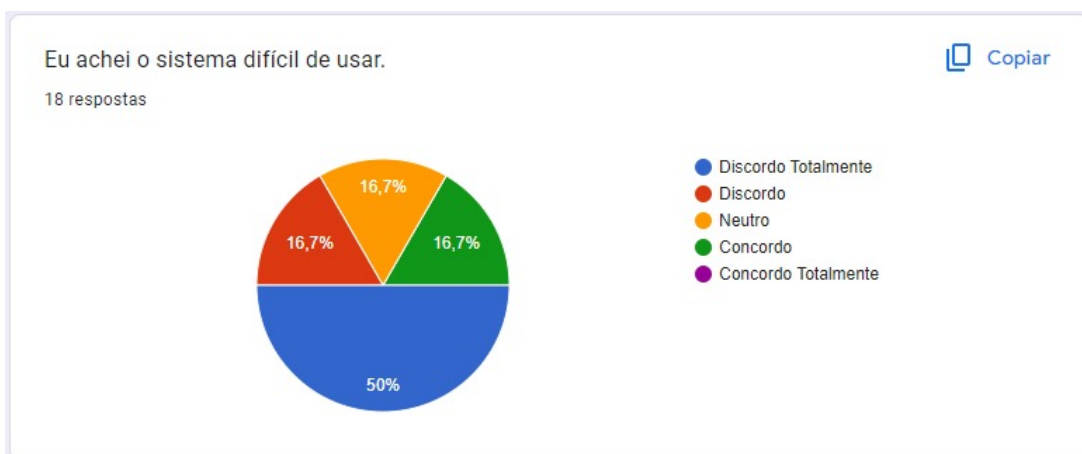


Figura 35 – Questão 8



Figura 36 – Questão 9

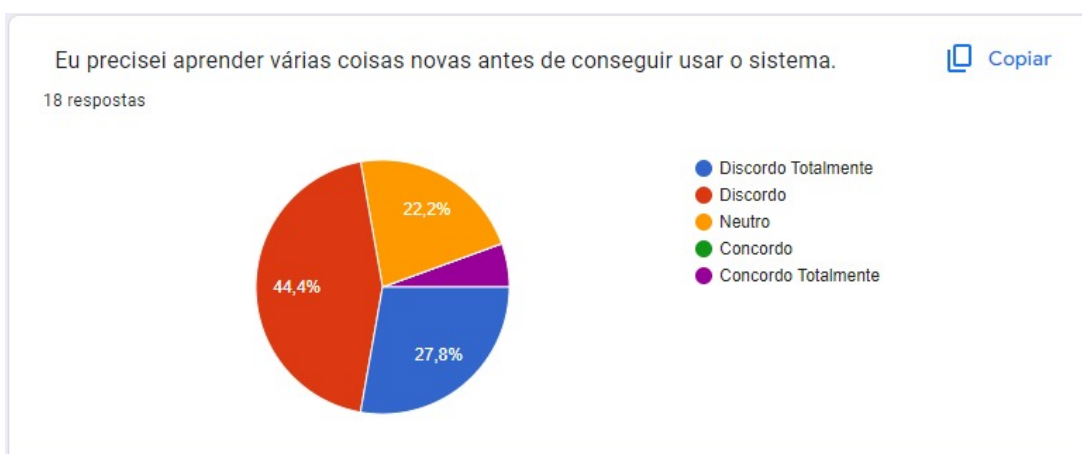


Figura 37 – Questão 10



**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
“JOSÉ ALBANO DE MACEDO”**

Identificação do Tipo de Documento

- Tese
- Dissertação
- Monografia
- Artigo

Eu, **Hériclys Sousa Borges**, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação “**Sistema de Gerenciamento de Processo Seletivo Para Cursinho Popular Pré-ENEM Paulo Freire**” de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI, 23 de agosto de 2023.


Assinatura