

**UNIVERSIDADE FEDERAL FLUMINENSE
CAMPUS UNIVERSITÁRIO DE RIO DAS OSTRAS
INSTITUTO DE CIÊNCIA E TECNOLOGIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**EDUARDO MATIAS COMBAT
LUCAS DE AQUINO SILVA**

PAPO UFF:

Plataforma de Aplicações de Provas Online

Rio das Ostras – Rio de Janeiro

2021

EDUARDO MATIAS COMBAT
LUCAS DE AQUINO SILVA

PAPO UFF:

Plataforma de Aplicações de Provas Online

Trabalho de conclusão de curso apresentado à Universidade Federal Fluminense, como parte das exigências para a obtenção do título de Bacharel em Ciência da Computação.

Rio das Ostras, 28 de abril de 2021.

BANCA EXAMINADORA

Prof. Carlos Bazilio Martins - Orientador
Universidade Federal Fluminense

Prof. Dalessandro Soares Vianna - Avaliador
Universidade Federal Fluminense

Prof. Sergio Crespo Coelho da Silva Pinto - Avaliador
Universidade Federal Fluminense

Ficha catalográfica automática - SDC/BRO
Gerada com informações fornecidas pelo autor

C729p Combat, Eduardo Matias
PAPÓ UFF : Plataforma de Aplicações de Provas Online /
Eduardo Matias Combat, Lucas de Aquino Silva ; Carlos Bazilio
Martins, orientador. Niterói, 2021.
94 f. : il.

Trabalho de Conclusão de Curso (Graduação em Ciência da
Computação)-Universidade Federal Fluminense, Instituto de
Ciência e Tecnologia, Rio das Ostras, 2021.

1. Avaliação educacional. 2. Ensino à distância. 3.
Desenvolvimento de sites da web. 4. Produção intelectual. I.
Silva, Lucas de Aquino. II. Martins, Carlos Bazilio,
orientador. III. Universidade Federal Fluminense. Instituto de
Ciência e Tecnologia. IV. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

"Everything that lives is designed to end. They are perpetually trapped in a never-ending spiral of life and death. However...life is all about the struggle within this cycle. That is what "we" believe."

(Nier: Automata)

RESUMO

Neste trabalho apresentamos o PAPO UFF, uma plataforma de aplicação de provas online que pretende contribuir com o cenário acadêmico trazendo uma discussão a respeito da atualização do cenário avaliativo das instituições acadêmicas. A aplicação pretende enfatizar a possibilidade de se ter uma melhor experiência para alunos e professores de diversas matérias, de maneira especial àquelas que estão relacionadas com o panorama computacional, por meio de tecnologias digitais. A plataforma visa disponibilizar, aos professores, ferramentas que lhes possibilitarão criar, em um ambiente totalmente online, questões de diferentes tipos que serão armazenadas em um banco de dados e ficarão disponíveis para que provas possam ser geradas e aplicadas aos alunos em uma interface virtual. Para cada atividade concluída, relatórios gráficos serão montados contendo estatísticas relacionadas ao desempenho de cada um dos alunos participantes. Em suma, pretendemos disponibilizar uma plataforma da qual alunos e professores possam se beneficiar de uma série de facilidades para que o método avaliativo seja mais justo e eficaz ao compararmos com uma prova em papel.

Palavras-chave: Cenário acadêmico. Banco de Dados. Panorama Computacional. Método avaliativo.

ABSTRACT

In this final paper, we present PAPO UFF, an online testing application platform that aims to contribute to the academic scenario, bringing a discussion about updating the evaluative scenario of academic institutions. In order to provide a better experience for students and teachers from different disciplines, especially those that relate to the computational landscape. The platform aims to provide teachers with tools that allow them to create, in a totally online environment, questions of various types that will be stored in a database, so tests can be generated and applied to students in a virtual interface. For each activity carried out, graphic reports containing statistics related to the performance of each of the participating students will be assembled. In short, we intend to provide a platform from which students and teachers can take considerable benefits as tools and functionalities designed for them. So that the evaluation method could be fairer and more effective when compared to a paper exam.

Keywords: Academic scenario. Database. Computational Landscape. Evaluation method.

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Motivação	9
1.2	O Primeiro Protótipo Desenvolvido	11
1.3	O Acesso à Tecnologia no Brasil	12
1.4	Comparativo entre Prova <i>Online</i> e Prova em Papel.....	12
1.5	Objetivos do Projeto	14
1.5.1	Objetivo Geral	14
1.5.2	Objetivos Específicos	14
2	REFERENCIAL TEÓRICO	16
2.1	Conceitos do Cenário <i>Web</i>	16
2.1.1	<i>Front-end</i>	16
2.1.2	<i>Back-end</i>	17
2.1.3	Bibliotecas	17
2.1.4	APIs.....	18
2.1.5	<i>Frameworks</i>	18
3	TRABALHOS RELACIONADOS	20
3.1	Google Sala de Aula.....	20
3.2	URI <i>Online Judge</i>	23
3.3	BOCA <i>Online Contest Administrator</i>	25
3.4	Coderbyte.....	28
3.5	Sistema de Provas <i>Online</i> com Desenvolvimento em Python com Django.....	30
3.6	Cadastramento de Questões e Aplicação de Provas <i>Online</i>	33
4	METODOLOGIA.....	37
5	DESCRIÇÃO DO PROJETO	39
5.1	Organização Interna do PAPO UFF	39

5.1.1 <i>Front-end</i> do Projeto.....	39
5.1.2 <i>Back-end</i> do Projeto	40
5.1.3 Banco de Dados.....	41
5.1.4 Atribuições Referentes ao Professor.....	42
5.1.5 Atribuições Referentes ao Aluno	44
5.2 Descrição das Funcionalidades e Interfaces	46
5.2.1 Interfaces atreladas ao Professor.....	46
5.2.2 Interfaces atreladas ao Aluno	57
6 RESULTADOS E DISCUSSÃO.....	61
6.1 Análise do Histórico do Curso de Ciência da Computação de Rio das Ostras.....	61
6.2 Análise dos resultados do Teste de Usabilidade.....	63
6.3 Considerações para Trabalhos Futuros	68
7 CONSIDERAÇÕES FINAIS	71
REFERÊNCIAS.....	72
APÊNDICE A – Tecnologias Utilizadas no Protótipo.....	75
APÊNDICE B – Interfaces do Protótipo.....	76
APÊNDICE C – Interfaces do PAPO UFF em Escala Real.....	83
APÊNDICE D – Pesquisa de Avaliação do PAPO UFF	90

LISTA DE FIGURAS

Figura 1 – Interface Inicial do Google Sala de Aula	21
Figura 2 – Interface de uma Turma no Google Sala de Aula	21
Figura 3 – Interface de Atividades do Google Sala de Aula	22
Figura 4 – Interface de criação de uma questão do Google Sala de Aula	22
Figura 5 – Página Inicial do usuário no URI	23
Figura 6 – Categorização de uma Questão.....	24
Figura 7 – Enunciado de uma Questão.....	24
Figura 8 – Campo de Resposta de Uma Questão.....	25
Figura 9 – Interface do time para enviar submissões.....	26
Figura 10 – Juiz respondendo uma submissão de um time	26
Figura 11 – Interface do Juiz para responder a perguntas feitas pelos times	27
Figura 12 – Administrador pode fazer o acompanhamento completo da competição, como a verificação das submissões dos times.....	27
Figura 13 – Tarefas que o pessoal de <i>staff</i> tem que executar.....	28
Figura 14 – Descrição de Questão no Coderbyte	29
Figura 15 – Campo de codificação de uma questão no Coderbyte.....	29
Figura 16 – Campo de Execução de Código no Codebyte	30
Figura 17 – Tela Principal de Administração do Site pela Visão de Administradores	31
Figura 18 – Tela de Adição, Edição e Exclusão de Dados.....	32
Figura 19 – Tela de Inscrição de Disciplina.....	32
Figura 20 – Tela de Realização da Prova	33
Figura 21 – Tela de Gabarito de Provas Realizadas.....	33
Figura 22 – Cria e lista as provas já confeccionadas e inclui, exclui ou edita as questões de cada prova.	34
Figura 23 – Criação de novos usuários locais.....	34
Figura 24 – Lista dos testes realizados pelo aluno.....	35
Figura 25 – Comparativo entre ferramentas disponíveis atualmente e o PAPO UFF	36
Figura 26 – Representação dos componentes do PAPO UFF	39
Figura 27 – DER (Diagrama Entidade-Relacionamento) do PAPO UFF	42
Figura 28 – Diagrama de Caso de Uso do Professor.....	43
Figura 29 – Diagrama de Caso de Uso do Aluno	45

Figura 30 – Interface de <i>Login</i> do PAPO UFF	46
Figura 31 – Interface de Perfil do Professor no PAPO UFF	47
Figura 32 – Mensagem pré-carregamento de uma Correção de Prova	47
Figura 33 – Tela do Gráfico com número de Acertos e Erros de uma Prova	48
Figura 34 – Tela com Gráfico das médias de tempo das questões de uma Prova ...	48
Figura 35 – Tela com Gráficos da média de compilações por questão de programação de uma Prova.....	49
Figura 36 – Questões em Destaque de uma Prova	49
Figura 37 – Relação de Notas de uma Prova.....	50
Figura 38 – Lista de Alunos de uma Prova	50
Figura 39 – Lista de Respostas de um Aluno.....	50
Figura 40 – Campo de Alteração de Nota em uma Questão de um Aluno.....	51
Figura 41 – Lista de Resultados da busca no Banco de Questões	51
Figura 42 – Visualização de uma Questão do Banco de Questões	52
Figura 43 – Botão para Adicionar nova Questão no Banco de Questões	52
Figura 44 – Criação de Nova Questão Para adição direta no Banco de Questões...	53
Figura 45 – Campos de Respostas na Criação de Nova Questão Para adição no Banco de Questões	53
Figura 46 – Lista de Matérias das Questões cadastradas no Banco	54
Figura 47 – Lista de Assuntos das Questões cadastradas no Banco de Questões e Adição de novo Assunto.....	54
Figura 48 – Aba de Criação de uma nova Prova.....	55
Figura 49 – Criação de uma Nova Questão no ambiente de Criação de Prova	55
Figura 50 – Definição dos dados de uma Questão para uma Prova	55
Figura 51 – Lista de questões do Bando de Questões visualizadas na Criação de Provas e Botão de Importação	56
Figura 52 – Mensagem de Erro ao Criar uma Prova sem Nome.....	56
Figura 53 – Mensagem de Erro ao Criar uma Prova sem Questões.....	57
Figura 54 – Mensagem de Sucesso ao Criar uma Prova.....	57
Figura 55 – Tela de Inserção de Código de Prova	58
Figura 56 – Tempo Restante da Prova, Lista de Questões e Botão Finalizar Prova.	58
Figura 57 – Interface de uma Questão do tipo Discursiva.....	59
Figura 58 – Interface de uma Questão do tipo Múltipla Escolha	59
Figura 59 – Campo para codificação em uma Questão de Programação	60

Figura 60 – Campos de <i>Input</i> , <i>Output</i> , Compilação e Execução em uma Questão de Programação.....	60
Figura 61 – Mensagem de Confirmação para Entrega de uma Prova	60

LISTA DE GRÁFICOS

Gráfico 1 – Portal da Transparência Análise sobre o Curso de Ciência da Computação de Rio das Ostras.....	62
Gráfico 2 – Questionário PAPO UFF Questão 1	63
Gráfico 3 – Questionário PAPO UFF Questão 2	64
Gráfico 4 – Questionário PAPO UFF Questão 3	64
Gráfico 5 – Questionário PAPO UFF Questão 4	65
Gráfico 6 – Questionário PAPO UFF Questão 5	66
Gráfico 7 – Número de acertos e erros da prova aplicada aos alunos de Linguagens de Programação.....	67
Gráfico 8 – Média de tempo gasto em cada questão pelos alunos de Linguagens de Programação.....	67
Gráfico 9 – Média de compilações na questão de programação da prova de Linguagens de Programação	68

LISTA DE ABREVIATURAS E SIGLAS

IDE *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado

SBC Sociedade Brasileira de Computação

XML *eXpensible Markup Language* ou Linguagem Extensível de Marcação

API *Application Programming Interface* ou Interface de Programação de Aplicação

CPF Cadastro de Pessoas Físicas

DER Diagrama Entidade Relacionamento

MER Modelo Entidade Relacionamento

1 INTRODUÇÃO

Atualmente, vivemos um cenário totalmente atípico por conta do surto pandêmico causado pela Covid-19, o que acarretou significativas mudanças aos mais diversos ramos e negócios em um contexto global. Considerando o contexto no qual as escolas e universidades estão inseridas, nos deparamos com uma urgência por medidas que pudessem manter os estudantes em contato com suas instituições de ensino para que pudessem continuar sua jornada acadêmica. Cada instituição estabeleceu uma maneira particular no que se refere à metodologia de ensino, aplicação de atividades e avaliações para contornar os perigos trazidos pela pandemia, porém, todas foram obrigadas a fazê-las a partir de uma mesma base que seria o ensino *online*, ou seja, via *internet*. Sendo assim, diversas aplicações *web* que antes eram vistas somente como possíveis ferramentas para o ensino e aprendizagem, passaram a desempenhar um papel crucial para que as práticas acadêmicas não fossem interrompidas, mesmo que essas ainda não demonstrem substituir perfeitamente as salas de aulas e os encontros presenciais. É neste cenário que pretendemos, com este trabalho, colaborar com uma ferramenta *online* que possa auxiliar de maneira efetiva professores e alunos (de maneira especial àqueles relacionados com a área de computação) que hoje dependem da *internet* para manter as aulas em dia e assim fazer com que a máquina educacional possa continuar gerando novos profissionais promissores para o mercado.

1.1 Motivação

A ideia do PAPO UFF surgiu muito antes de tudo o que estamos vendo no cenário educacional como citado anteriormente. Para sermos mais exatos, desde o ingressar na faculdade, ao cursarmos a disciplina de Programação 1 em nosso primeiro período, já pensávamos em uma maneira que pudesse contribuir com o entendimento do conteúdo por parte dos alunos e que pudesse passar ao professor uma melhor noção do motivo destes estudantes (calouros do curso de Ciência da Computação, muitas das vezes, tendo seu primeiro contato com programação) estarem tendo tantas dificuldades, visto que o percentual de reprovação acabou sendo demasiado (houveram somente 6 aprovações em uma turma com cerca de 40 alunos).

Podemos considerar como sendo a ideia embrionária por trás do que viria a ser este trabalho, a necessidade que visualizamos de se ter na universidade uma ferramenta que pudesse substituir as provas e trabalhos que demandavam codificação em papel, pois uma parcela considerável dos alunos demonstrava grande dificuldade em expressar suas soluções e visualizar o funcionamento dos seus algoritmos sem contar com um computador, visto que, ao longo de todo o curso, o uso de algum *software* de desenvolvimento, as chamadas *IDE* (do inglês *Integrated Development Environment*, ou seja, Ambiente de Desenvolvimento Integrado) era indispensável, inclusive em sala de aula ao decorrer de todo o semestre.

Com o passar dos períodos, notamos que uma aplicação capaz de aplicar provas e atividades de forma a substituir àquelas em papel poderia ser útil não só para as matérias de computação que envolvem codificação, mas também para diversas outras, desde que essa dispusesse de algumas ferramentas que fizessem sentido em outros cenários. Começamos a vislumbrar algumas funcionalidades pertinentes no meio digital que fizessem do PAPO UFF algo além de um simples programa que emulasse uma IDE para as aulas de programação. Pensamos em funcionalidades que pudessem contribuir com a qualidade de ensino de diversas matérias ao retratar dados importantes sobre o grau de aprendizagem da turma a respeito de um determinado conteúdo, ou como cada aluno, individualmente, estaria lidando com cada questão de uma dada prova, quanto tempo estaria dedicando a cada uma dessas questões, se a visão de alunos e professores seria compatível em relação às questões consideradas fáceis e difíceis, entre outras possíveis análises. Em suma, pudemos perceber que as provas abrangem mais informações além de meras respostas, e essas podem ser muito úteis e interessantes para que o professor possa entender o contexto no qual a sua turma está inserida. Nas atividades feitas em papel, estas informações acabam ficando ocultas, entretanto, acreditamos ser possível extraí-las em um cenário digital.

Por meio do PAPO UFF, pretendemos também, trazer à tona uma discussão que julgamos ser muito relevante e pertinente a respeito do rumo que a educação pode tomar por conta das inovações e adaptações de ferramentas digitais no cotidiano acadêmico. Esta discussão pretende mostrar como o futuro das salas de aula, inclusive após superarmos este período pandêmico, onde as atividades *web* acabam sendo a única opção viável para que as atividades escolares sigam acontecendo, pode se beneficiar com o uso de uma ferramenta que permita uma análise mais

precisa por parte do professor a respeito das características de seus alunos e de suas turmas.

1.2 O Primeiro Protótipo Desenvolvido

Uma importante etapa pela qual passamos até que chegássemos à ideia final que iremos relatar ao longo deste trabalho, foi o protótipo sobre o qual falaremos agora. Este protótipo já se encontrava em uma etapa consideravelmente avançada de desenvolvimento contendo, inclusive, uma série de funcionalidades já bem definidas. Estas funcionalidades permitiam, por exemplo, a criação de questões, criação de provas, gerenciamento de um banco de questões e até a aplicação de uma prova virtual. Porém, visando lapidar melhor a ideia e encaixá-la em um cenário atuante mais amplo, tomamos uma série de decisões que culminaram na descontinuação do programa e passamos a desenvolver uma aplicação totalmente nova, em um paradigma totalmente diferente.

É importante salientar que no momento em que iniciamos o desenvolvimento deste protótipo, adotamos que a aplicação dessas atividades digitais seria realizada em um ambiente muito bem controlado, no caso, o próprio laboratório de computação da universidade, pois acreditávamos que o programa poderia se tornar um recurso em que os professores poderiam simplesmente manter o escopo de uma prova dada em sala, como sempre foi feito, porém, substituindo o papel pelo computador. Dessa forma, o programa funcionaria de forma *Stand-Alone* (termo em inglês para uma aplicação autossuficiente, onde todo o conteúdo necessário para o funcionamento do programa fica instalado na máquina do usuário), isto é, de maneira a depender somente da rede interna do laboratório para que houvesse comunicação entre os computadores nos quais estaria instalado.

A descrição das tecnologias utilizadas no desenvolvimento deste protótipo está disponível no Apêndice A. Todas as interfaces atreladas a cada funcionalidade do protótipo estão disponíveis no Apêndice B.

A decisão por alterar totalmente o escopo do programa acabou sendo, de certa forma, imposta pelas limitações que tínhamos devido a necessidade do distanciamento social. Consideramos que trabalhar com uma aplicação *online*, além de facilitar testes futuros e abranger o potencial do programa, seria uma forma mais adequada tratar o

assunto, visto que este tipo de aplicação tem se tornado cada vez mais presente no dia a dia das pessoas.

1.3 O Acesso à Tecnologia no Brasil

Considerando a ideia de que uma ferramenta digital como o PAPO UFF poderia ser reflexo de um futuro próximo no cenário educacional nacional, buscamos avaliar o nível no qual o país tem crescido no quesito acesso a dispositivos tecnológicos e *internet*.

“Em 2019, o Brasil possuía cerca de 134 milhões de usuários de Internet, ou 74% da população com dez anos ou mais”. (TIC DOMICÍLIOS, 2019 p. 23). Os dados citados acima, referem-se a uma pesquisa que englobou ambientes urbanos e rurais. Um ponto válido a se destacar da conclusão obtida a partir dessa, foi o aumento considerável no acesso a internet em regiões mais remotas. “[...] pela primeira vez na série histórica da pesquisa, mais da metade dos domicílios (51%) contavam com conexão de Internet”. (Ibidem, p. 61).

O interessante nesses dados é o fato de que é nítido que estamos passando por uma etapa de digitalização. Porém, o processo de distribuição de acesso às ferramentas necessárias para chegar à conclusão desta etapa ainda está longe de ser concretizado. De acordo com o crescimento notado e tendências apresentadas, no cenário educacional, caso tenhamos uma modificação efetiva em relação ao método avaliativo que deverá substituir as provas no papel, esta modificação não deve acontecer de forma tão rápida. É esperado que, caso algum tipo de mudança seja de fato implementado, este seria feito inicialmente em países mais desenvolvidos tecnologicamente e, após isso, o processo tenderia a ser replicado pelos demais. De qualquer forma, acreditamos que por meio de estímulos e debates, como os propostos por trabalhos semelhantes, esta digitalização possa ser potencializada e seus resultados possam de fato contribuir com a população de forma geral.

1.4 Comparativo entre Prova *Online* e Prova em Papel

Buscando ingressar nessa discussão entre o quanto devemos considerar pertinente a hipótese de que as provas em papel poderão ser substituídas por uma ferramenta totalmente digital, nos deparamos com um estudo muito interessante sobre

o tema. Nosso intuito seria avaliar se realmente existe algum indício de que isso virá a ocorrer, quando poderia acontecer e quais os reais benefícios que esta digitalização realmente poderia prover no quesito qualidade de ensino.

Na pesquisa realizada, foram considerados os alunos de ensino médio e aplicadas provas de matemática e português. A maneira a qual a aplicação foi feita, garantia que todos os alunos participantes realizassem uma prova online e outra em papel. (Mattos e Da Costa, 2013).

Justamente por se tratar de um cenário de ensino médio, este estudo nos permitiu avaliar de forma mais abrangente o que seria a expectativa pelas consequências em se ter a aplicação de provas *online*. Acreditamos que poderiam ter ocorrido algumas divergências caso fosse realizado algo similar no escopo dedicado aos alunos de Ciência da Computação. Entretanto, a forma com que o tema foi abordado, pode nos fazer refletir sobre algumas vantagens que a substituição do papel poderia proporcionar.

Em termos de desempenho, “Pode-se concluir então que não existe diferença significativa entre as médias das avaliações on-line e escrita para as duas disciplinas”. (Ibidem, p. 9). Ou seja, a variação de notas dos alunos seguiu o mesmo padrão tanto no cenário online quanto na prova impressa. Entretanto, “Os alunos, de forma geral, expressaram maior interesse pelas provas online, ...” (Ibidem).

Neste estudo podemos notar que, simplesmente substituir uma prova em papel por uma prova online, apesar de não contribuir efetivamente para um melhor desempenho dos alunos nas provas, pode trazer alguns benefícios de imediato.

A avaliação online possibilita maior dinâmica na construção de provas e avaliações e os resultados desta pesquisa evidenciam que, do ponto de vista do desempenho do aluno, a modalidade de aplicação não interfere significativamente, nas duas disciplinas as distribuições das notas eram muito semelhantes nas duas modalidades de avaliação, assim como o fato de o aluno apresentar maior desenvolvimento das habilidades de uso do computador e internet também não assegura melhor desempenho nas provas online. Por outro lado, os resultados demonstraram que alunos que apresentam essas habilidades mais desenvolvidas tem melhor desempenho tanto nos dois conteúdos quanto nas duas modalidades de aplicação. (Ibidem).

Além do que foi concluído pelos autores da pesquisa, acreditamos ser válido reforçar que a ideia de uma ferramenta como o PAPO UFF visa não somente replicar a aplicação de uma prova impressa de forma online. Mas sim prover recursos para melhor avaliar o

desempenho desses alunos e traçar algum indicativo que aponte quais seriam os pontos que possam vir a ser os maiores empecilhos para que o aprendizado flua na turma de uma forma geral.

1.5 Objetivos do Projeto

Após passarmos por todo um processo de maturação das nossas ideias em relação ao trabalho a partir do levantamento de pontos importantes tais quais como vantagens e desvantagens observadas com a criação do protótipo, conversas com professores, observações e estudos de materiais auxiliares, pudemos ter uma melhor base do que seria realmente útil para o nosso intuito, que sempre esteve relacionado com criar uma ferramenta que pudesse colaborar com o crescimento da efetividade do ensino acadêmico.

A partir destas análises, organizamos nossos objetivos com este trabalho entre um aspecto geral, mais abrangente e aqueles considerados específicos, que nos permitirão descrever cada estrutura que compõe este estudo de forma mais clara e eficiente.

1.5.1 Objetivo Geral

Temos como objetivo geral desenvolver uma aplicação que permita aos professores uma melhor visualização a respeito do cenário no qual sua turma se encontra em relação à matéria lecionada e que, de forma análoga, permita aos alunos, um ambiente que propicie melhores condições para desenvolver suas respostas e expor seus conhecimentos ao serem avaliados, principalmente, àqueles voltados para o âmbito computacional.

1.5.2 Objetivos Específicos

- Ilustrar, através de estudos a respeito do tema, qual pode ser o impacto no aprendizado a substituição de uma prova em papel por uma atividade digital;
- Apresentar programas e ferramentas semelhantes e apontar seus pontos fortes e fracos pensando no nosso cenário;
- Citar as tecnologias utilizadas para a criação da aplicação do PAPO UFF;

- Explicar o funcionamento de todos os recursos da aplicação PAPO UFF;
- Descrever cada etapa de desenvolvimento, dificuldades e barreiras superadas para a conclusão do programa;
- Analisar os resultados obtidos com a utilização da ferramenta em um teste empírico;
- Propor e recomendar novos recursos que poderiam ser adicionados ao PAPO UFF para que possa ter um poder atuante ainda mais abrangente.

2 REFERENCIAL TEÓRICO

Vejam os nesta seção, algumas definições e conceitos importantes para o entendimento do assunto tratado neste trabalho.

2.1 Conceitos do Cenário *Web*

Vejam os a partir de agora, alguns conceitos muito importantes para a computação voltada para a *web*. Os termos apresentados a seguir irão nos permitir entender melhor o conteúdo no qual este trabalho está inserido e qual seria a base estrutural das tecnologias nele utilizadas.

2.1.1 *Front-end*

Quando nos referimos a parte de *front-end* de uma aplicação *web*, estamos considerando tudo o que engloba a interface de interação com o usuário desta aplicação.

“Podemos classificar como a parte visual de um site, aquilo que conseguimos interagir. Quem trabalha com *front-end* é responsável por desenvolver por meio de código uma interface gráfica, normalmente com as tecnologias base da Web [...]” (SOUTO, 2019, p. 1).

É importante se atentar ao fato de que a parte de *front-end* está relacionada com códigos que serão interpretados pelos navegadores. Isto é, o profissional dessa área utiliza recursos e ferramentas que têm estreita relação com estes que fazem papel de expor a informação que vem dos servidores para os usuários. Diferentemente do trabalho de um *designer*, por exemplo, que utiliza ferramentas visuais para desenhar interfaces. (Ibidem). Neste trabalho, fazemos uso do React¹, que se trata de uma biblioteca JavaScript focada para criação de interfaces de usuário. Esta escolha deu-se muito por conta da observação do cenário atual do mercado de trabalho que, em geral, tem considerado bastante o uso dessa ferramenta. Podemos notar a popularidade do React por meio da pesquisa anual realizada pelo Stack

¹ Disponível em: <<https://reactjs.org/tutorial/tutorial.html#overview> >. Acesso em 5 abr. 2021.

Overflow². Dentre outras que foram cogitadas para desenvolvimento desse projeto, podemos citar o Angular e o Vue.Js.

2.1.2 *Back-end*

Podemos definir a parte de *back-end* de uma aplicação *web* como sendo o responsável por proporcionar a viabilidade e eficácia de suas funcionalidades. Enquanto que o *front-end* estaria atrelado com tudo o que se relaciona diretamente com a interface a qual o usuário deverá interagir, o *back-end* visa dar suporte para que todos os recursos nela contidos funcionem de forma adequada.

“O *back-end* trabalha em boa parte dos casos fazendo a ponte entre os dados que vem do navegador rumo ao banco de dados e vice-versa, sempre aplicando as devidas regras de negócio [...]”. (SOUTO, 2019, p. 3). Neste trabalho, utilizamos o NodeJs³ como base do nosso *back-end*. Este que se trata de um ambiente de execução JavaScript multiplataforma.

2.1.3 Bibliotecas

O conceito de biblioteca em computação pode ser visto de forma semelhante ao uso da palavra para o ambiente físico no qual pensamos ao lê-la. “A ideia da biblioteca é compartilhar soluções por meio de funções ou métodos” (Zanette, 2017, p. 2).

Isso nos auxilia como é feito seu uso no meio *web*. Uma biblioteca é utilizada de maneira que os desenvolvedores possam aproveitar dos recursos que ela contém de forma fácil e prática em suas aplicações. Ao ser incluída no código, o desenvolvedor poderá fazer acesso de uma determinada função e receber o resultado dessa operação para que seja manipulado da maneira que for necessária.

Um exemplo interessante seriam os programas que utilizam a função de potência na linguagem C. Para isso, deverão incluir a biblioteca `math.h`.

² Disponível em <<https://insights.stackoverflow.com/survey/2020#most-popular-technologies>>. Acesso em 3 mai. 2021.

³ Disponível em: <<https://nodejs.org/en/docs/guides/>>. Acesso em 5 abr. 2021.

2.1.4 APIs

Sendo este, provavelmente, um termo mais restrito àqueles que trabalham diretamente com programação *web*. Podemos dizer que se trata de uma maneira com a qual um desenvolvedor consegue utilizar recursos de outras aplicações de forma integrada com o seu próprio projeto.

“[...]interface que orienta determinado programador quanto ao acesso a comandos, funções, protocolos ou objetos de um software ou plataforma”. (Zanette, 2017, p. 4).

Um exemplo seria o uso do mapa do Google em diversas aplicações. Neste caso, estes softwares utilizam APIs para ter esse recurso. Neste trabalho, por exemplo, fizemos uso do CanvasJS, uma API que nos possibilitou gerar os gráficos para análise do desempenho dos alunos nas provas.

2.1.5 Frameworks

Este pode ser um conceito confuso inclusive para os que trabalham na área. Podendo ser explicado de uma forma mais abstrata, *frameworks* podem ser vistos como uma espécie de forma com a qual o desenvolvedor irá trabalhar.

“Podemos dizer que é um conjunto de códigos abstratos e/ou genéricos que unem códigos com recursos iguais. Ou seja, está ligado à arquitetura do seu *software*”. (Zanette, 2017, p. 5).

Para complementar esta explicação, podemos imaginar um *framework* como sendo um recurso que tratará o seu software como uma fábrica de carros gera os veículos. É notório que vejamos, em uma mesma montadora, carros com estruturas muito semelhantes, como compartilham de um certo padrão de montagem. Isso acontece justamente por tornar muito mais prática a etapa de confecção desses veículos. De forma semelhante, um *framework* visa disponibilizar para o desenvolvedor, recursos que possam ser utilizados para criar suas funcionalidades de forma mais prática e rápida. Entretanto, acabam gerando uma certa dependência em relação a forma que foi desenvolvida.

“O *framework* está ligado diretamente ao código fonte da aplicação. Isso significa que, para cada linguagem de programação, teremos *frameworks* diferentes, mesmo que eles exerçam a mesma função”. (Ibidem, p. 6).

Podemos dizer que na relação entre um *framework* e uma determinada aplicação ocorre um tipo de troca entre praticidade e liberdade de personalização.

Um exemplo, que inclusive foi utilizado neste trabalho, foi o React Bootstrap. Um Framework que nos permitiu trabalhar de forma mais prática a composição da interface do PAPO UFF.

3 TRABALHOS RELACIONADOS

A partir da definição dos nossos objetivos e escopo de atuação do PAPO UFF, pudemos voltar nossos olhares para as plataformas de ensino online já conhecidas no mercado de uma maneira mais crítica. Através de uma análise dessas plataformas, buscamos por características que fossem interessantes em cada uma delas. A partir daí, pudemos definir, mediante alguns ajustes, quais funcionalidades ou particularidades que poderiam ser integradas ao nosso programa. Esta estratégia seria importante pois poderíamos proporcionar uma plataforma mais amigável para aqueles que já estariam habituados com as ferramentas dessas outras aplicações.

Além disso, visando avaliar como outras instituições já teriam ingressado na discussão e proposta de termos um ambiente para aplicação de provas *online*, buscamos alguns trabalhos acadêmicos que tenham implementado algo semelhante ao proposto com este trabalho.

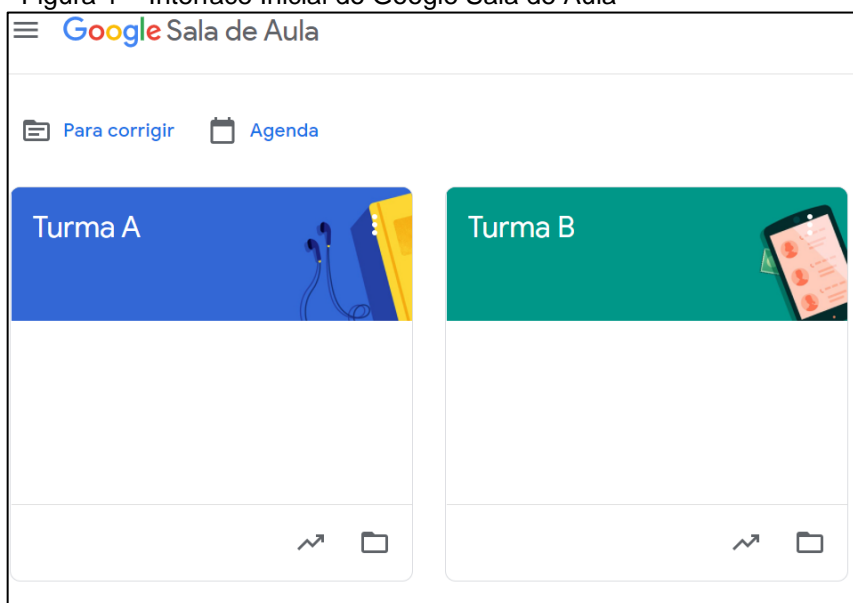
3.1 Google Sala de Aula

Ativo desde 2014, hoje trata-se de uma das plataformas mais utilizadas em meio a pandemia para gerenciamento de aulas *online*. Certamente seu ponto mais forte seria a integração com todo o ambiente Google, entretanto, como principal característica negativa, podemos considerar alguns aspectos de usabilidade. Dentre esses problemas de usabilidade, uma característica marcante seria a tentativa de abstrair suas funcionalidades em botões com uma simbologia confusa. Poderemos notar essa tentativa de impor padrões nas imagens a seguir. Em cada interface, existe algum símbolo que representa uma funcionalidade. Muitas das vezes, são símbolos que não demonstram sua funcionalidade de maneira clara, mas acabam confundindo o usuário justamente por não corresponderem com o que seria o seu intuito.

O Google Sala de Aula⁴ tem como ideia principal a organização do conteúdo por meio de turmas. Já na imagem a seguir, podemos ver o símbolo de uma seta e uma pasta, que significam, respectivamente, um botão para abrir o diário de classe e outro para abrir a pasta da turma no Google Drive (serviço de armazenamento e sincronização de arquivos do Google).

⁴ Disponível em: < <https://classroom.google.com/>>. Acesso em: 7 de dez. de 2020.

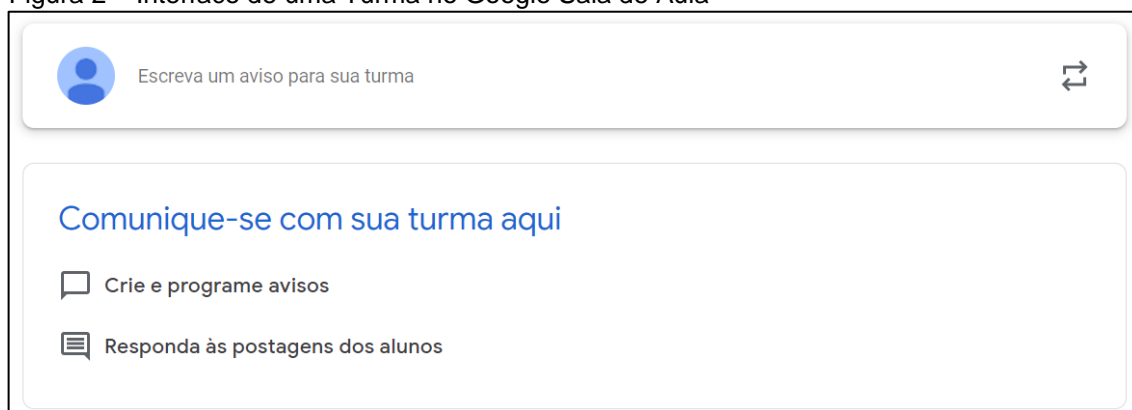
Figura 1 – Interface Inicial do Google Sala de Aula



Fonte: Google Sala de Aula (2020)

Professores podem visualizar as turmas para as quais estão lecionando, agendar e aplicar testes, visualizar as notas dos alunos e se comunicar com eles por meio de um tipo de mural virtual.

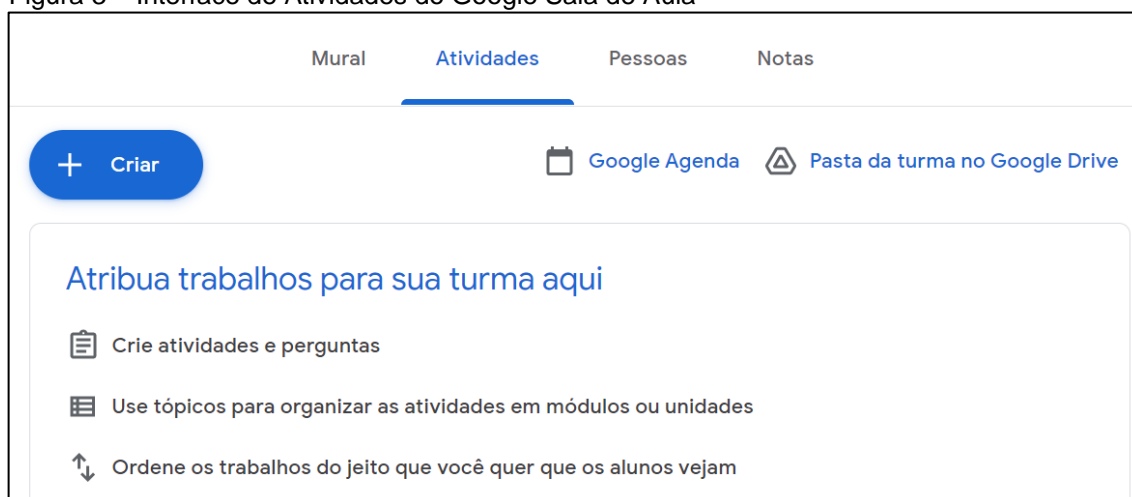
Figura 2 – Interface de uma Turma no Google Sala de Aula



Fonte: Google Sala de Aula (2020)

Dentre as funcionalidades do Google Sala de Aula, aquela que seria mais interessante para nosso contexto, naturalmente, seria a disponibilização das atividades. Podemos notar, na figura 3, que, fugindo do padrão de não explicitar o que o símbolo representaria, a ferramenta mantém escrito “Criar” ao lado do símbolo ‘+’.

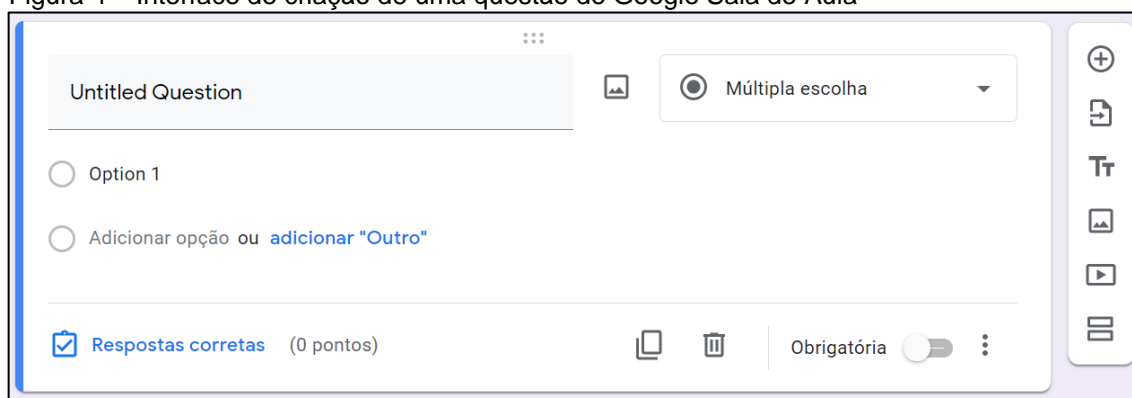
Figura 3 – Interface de Atividades do Google Sala de Aula



Fonte: Google Sala de Aula (2020)

A alternativa disponibilizada pelo Google, apresenta uma interface simplista para a criação de questões, onde é possível criar diferentes tipos de atividades e informar as respostas esperadas. Uma questão pode estar relacionada com algum recurso de mídia, ou seja, alguma imagem ou até mesmo vídeo. Entretanto, a plataforma não tem nenhuma solução ou suporte para o cenário da computação, justamente, o que seria um dos pontos de foco do PAPO UFF.

Figura 4 – Interface de criação de uma questão do Google Sala de Aula



Fonte: Google Sala de Aula (2020)

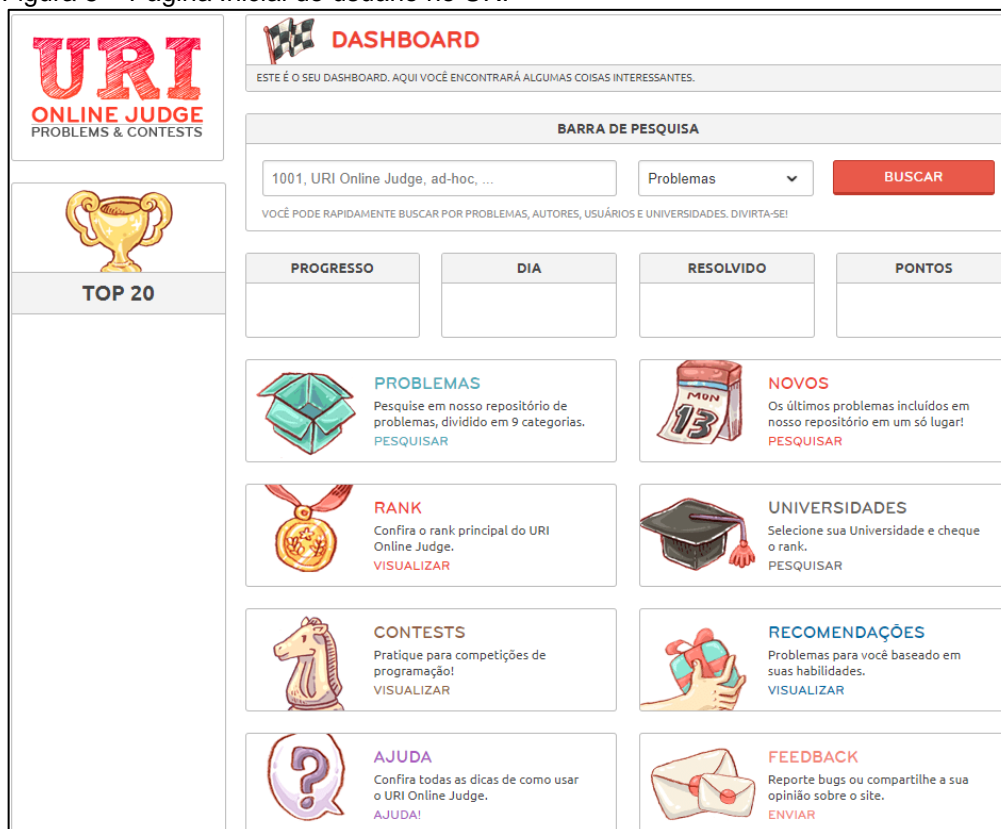
Esta maneira de criar as questões de uma atividade de forma semelhante a criação de blocos, certamente foi a maior contribuição do Google Sala de Aula na criação do PAPO UFF. Inclusive, foi adotada uma estratégia correspondente para a criação das questões de uma prova na nossa plataforma, entretanto, buscamos evitar

a simbologia do Google e adotamos um padrão mais claro, explicitando a funcionalidade de cada botão.

3.2 URI Online Judge

O URI⁵ é uma plataforma digital desenvolvida pela Universidade Regional Integrada, visando colaborar com alunos e professores de algoritmos e linguagens de programação. Utilizada tanto em treinamentos para maratonas de programação como para aplicação de exercícios complementares em aulas, conta com um banco de questões bastante vasto. Além disso, buscando incentivar seus usuários a resolverem os exercícios da plataforma, existe um *ranking* baseado em uma pontuação que considera a quantidade de exercícios respondidos e a qualidade dos códigos das resoluções. A plataforma retorna ao usuário uma porcentagem de acerto baseado em testes de entrada e saída. Vejamos abaixo a imagem da página inicial do URI, após realização de *login* por parte de um aluno.

Figura 5 – Página Inicial do usuário no URI

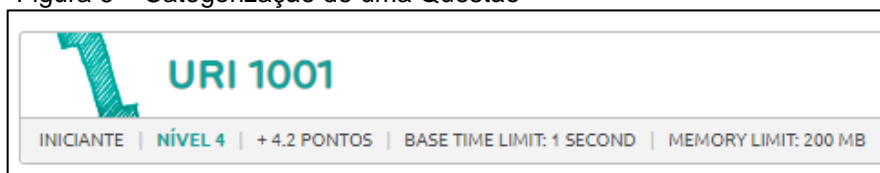


Fonte: URI Online Judge (2020)

⁵ Disponível em: <<https://www.urionlinejudge.com.br/>>. Acesso em: 7 de dez. de 2020.

Cada exercício conta com um nível de dificuldade, exemplos de entrada e saída e um tempo limite de execução. Na figura 6 e 7, podemos visualizar como estas informações são passadas para o aluno no escopo de um exercício.

Figura 6 – Categorização de uma Questão



Fonte: URI *Online Judge* (2020)

Figura 7 – Enunciado de uma Questão

URI Online Judge | 1001 ⌵

Extremamente Básico

Adaptado por Neilor Tonin, URI Brasil

Timelimit: 1

Leia 2 valores inteiros e armazene-os nas variáveis **A** e **B**. Efetue a soma de **A** e **B** atribuindo o seu resultado na variável **X**. Imprima **X** conforme exemplo apresentado abaixo. Não apresente mensagem alguma além daquilo que está sendo especificado e não esqueça de imprimir o fim de linha após o resultado, caso contrário, você receberá "*Presentation Error*".

Entrada

A entrada contém 2 valores inteiros.

Saída

Imprima a mensagem "X = " (letra X maiúscula) seguido pelo valor da variável **X** e pelo final de linha. Cuidado para que tenha um espaço antes e depois do sinal de igualdade, conforme o exemplo abaixo.

Exemplos de Entrada	Exemplos de Saída
10 9	X = 19

Fonte: URI *Online Judge* (2020)

O usuário pode escolher dentre uma lista de linguagens aceitas pela plataforma para enviar sua solução para determinado problema. É possível notar também, na próxima imagem, que existe um campo para identificação da questão por meio de um código. De forma semelhante, no PAPO UFF uma prova recebe um código de acesso no momento de sua criação.

Figura 8 – Campo de Resposta de Uma Questão

The image shows a web form for submitting a solution. It consists of the following elements:

- PROBLEMA:** A text input field containing the number '1477'.
- LINGUAGEM:** A dropdown menu with 'C++17' selected.
- SOURCE CODE:** A large text area for writing code, with a vertical line number indicator on the left showing '1'.
- Buttons:** A grey button on the left with the text 'CONSTRUA A SUA SOLUÇÃO E ENVIE!' and a green button on the right with the text 'ENVIAR'.

Fonte: URI *Online Judge* (2020)

Portanto, o URI se trata de um grande exemplo de ferramenta digital que tem contribuído com o cenário do ensino de computação e realização de maratonas. Entretanto, por conta de ter seu foco na resolução de exercícios com viés voltado para as maratonas, a proposta do URI acaba não sendo compatível com o que se demanda para a aplicação de uma prova. A ferramenta pode contribuir ao longo da matéria de programação de maneira similar ao que é feito com as listas de exercícios, entretanto, o PAPO UFF pretende proporcionar um ambiente mais propício para o que é visto em uma prova. Uma característica interessante que demonstra essa diferença é a possibilidade de se compilar o código de resposta de uma questão no PAPO UFF e executá-lo antes de concluir a prova. Assim, o aluno pode ter segurança em entregar sua resposta, porém, a parte referente à correção, ficará disponível somente para o professor. Por outro lado, o URI já apresenta um feedback direto sobre o código utilizado como resposta, ou seja, se este está de acordo com o que foi pedido ou não, excluindo o que seria o papel do professor na avaliação do aluno.

3.3 BOCA *Online Contest Administrator*

O BOCA⁶ é um sistema de submissão que foi criado para controlar competições visando a forma de atuação da Maratona de Programação da SBC. A ferramenta é bem conhecida no cenário acadêmico justamente pelo suporte prestado aos competidores e organizadores. Tendo sido desenvolvida em 2004, sua interface está

⁶ Disponível em: <<https://www.ime.usp.br/~cassio/boca/campos-ferreira-wei2004.pdf>>. Acesso em: 2 de abr. de 2021.

um pouco datada, mas sua atuação é muito constante nestes eventos até hoje. O sistema funciona com base na ação de quatro principais agentes: o time, os juízes, o administrador e o *staff*. Vejamos, em sequência, quais funcionalidades estão atribuídas a cada um desses agentes e quais seriam as interfaces correspondentes a cada um deles.

O time, que irá anexar os arquivos da sua solução para que possam ser enviados aos juízes.

Figura 9 – Interface do time para enviar submissões

BOCA Username: Universidade A (site=2) 157 minute(s) left

Runs	Score	Clarifications	Tasks	Options	Logout
Run #	Time	Problem	Language	Answer	
2	37	Problema 1	C	No - Compilation error	
3	37	Problema 3	Java	Yes	
4	83	Problema 1	C++	Not answered yet	

To submit a program, just fill in the following fields:

Problem:

Language:

Source code:

Fonte: Modificado de Campos e Ferreira (2004, p. 3)

Os juízes, que poderão responder uma submissão, visualizar e responder as perguntas dos integrantes dos times.

Figura 10 – Juiz respondendo uma submissão de um time

BOCA Username: Juiz 1 (site=2) 118 minute(s) left

Runs (1)	Score	Clarifications (2)	History	Options	Logout
Use the following fields to judge the run:					
Site:	2				
Number:	4				
Time:	83				
Problem Problema 1:	Input: pl.in view	Sol: pl.sol view			
Language:	C++:				
Source code:	AP130.txt view				
Answer:	Not answered yet				

Fonte: Modificado de Campos e Ferreira (2004, p. 6)

Figura 11 – Interface do Juiz para responder a perguntas feitas pelos times

Use the following fields to answer the clarification:

Clarification Site: 2
 Clarification Number: 1
 Clarification Time: 86
 Problem: Problema 1

Clarification:
 O que significa n?

Answer:
 Que n...

Answer all users in the site
 Answer all users in all sites

Answer No response Cancel Clear

Fonte: Modificado de Campos e Ferreira (2004, p. 4)

O administrador, que terá o controle sobre a configuração inicial da competição.

Figura 12 – Administrador pode fazer o acompanhamento completo da competição, como a verificação das submissões dos times

BOCA Username: Administrador 1 (site=2) 114 minute(s) left								
Runs	Score	Clarifications	Users	Problems	Languages	Answers		
Tasks	Site	Contest	Logs	Reports	Options	Logout		
Run #	Site	User	Time	Problem	Language	Status	Judge (Site)	Answer
5	2	team2	98	Problema 1	C	judged	judge1 (2)	Yes
4	2	team1	83	Problema 1	C++	judging	judge1 (2)	Not answered yet
1	2	team1	39	Problema 1	C	deleted	admin1 (2)	Yes
3	2	team1	37	Problema 3	Java	judged	admin1 (2)	Yes
2	2	team1	37	Problema 1	C	judged	judge1 (2)	No - Compilation error

Fonte: Modificado de Campos e Ferreira (2004, p. 7)

O *Staff* (termo em inglês utilizado para se referir ao pessoal do âmbito da organização do evento), que irá prover a devida assistência aos times em relação a problemas de *hardware* ou *software*, acomodações, impressão de material e afins.

Figura 13 – Tarefas que o pessoal de *staff* tem que executar

BOCA		Username: Pessoa do staff (site=2)			110 minute(s) left	
Tasks (4)		Score	Options	Logout		
Task #	Time	User / Site	Description	File	Status	Actions
1	1	team1(1) / 2	Staff assistance		opentask	get
3	1	team1(1) / 2	File to print	Autoexec.bat	opentask	get
5	83	team1(1) / 2	Delivery to "team1" a balloon for problem Problema 3:		opentask	get
6	98	team2(2) / 2	Delivery to "team2" a balloon for problem Problema 1:		opentask	get

Fonte: Modificado de Campos e Ferreira (2004, p. 8)

Apesar das diferenças de propostas em comparação com o PAPO UFF, o BOCA foi considerado como uma inspiração justamente pela importância que o programa atingiu no cenário das competições. Dessa forma, pretendemos mostrar a relevância que ferramentas digitais já possuem no cenário educacional. Além de fomentarmos uma discussão a respeito da integração dessas ferramentas com as escolas e universidades, incentivando a evolução dessas aplicações que visam contribuir com a melhoria do ensino acadêmico.

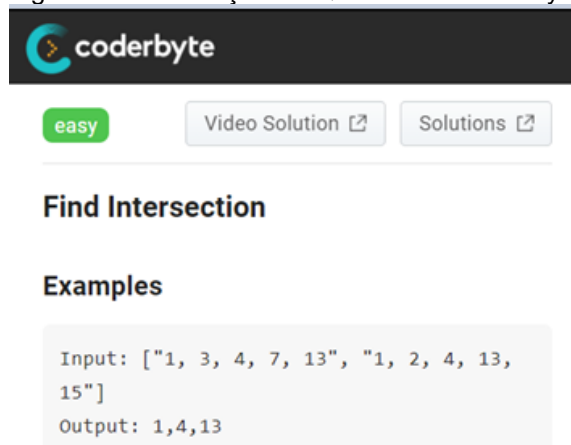
3.4 Coderbyte

Tratando-se de uma aplicação *web* desenvolvida para prática de programação, a Coderbyte⁷ oferece cursos e desafios voltados para o mercado de trabalho. Existe uma integração entre a plataforma e recrutadores para eventuais vagas no mercado, visando um recrutamento prático e preciso para profissionais da área.

Sua interface de aplicação dos desafios é muito funcional, sendo baseada em três partes distintas. O lado esquerdo da página, representado na figura 14, trata das informações do problema a ser respondido, isto é, seu enunciado, uma categoria de dificuldade e exemplos de *input* e *output*.

⁷ Disponível em: <<https://www.coderbyte.com/>>. Acesso em: 7 de dez. de 2020.

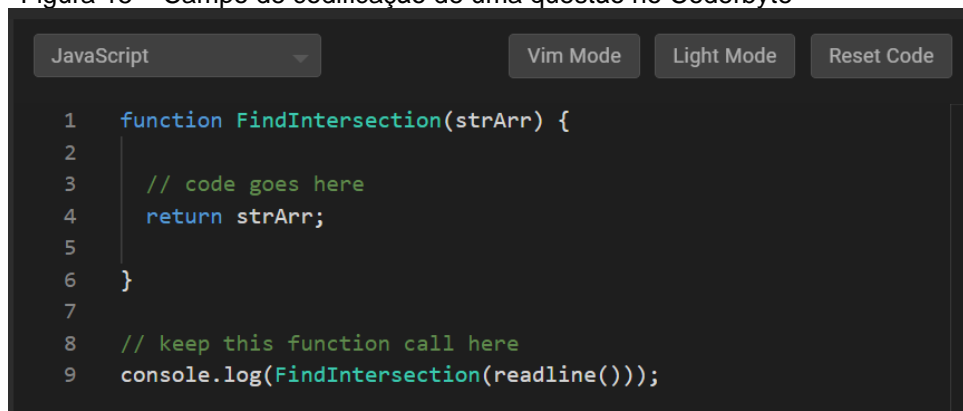
Figura 14 – Descrição de Questão no Coderbyte



Fonte: Coderbyte (2020)

Na próxima imagem, podemos ver o campo correspondente ao meio da página, que se trata do campo reservado para desenvolvimento do código. O Coderbyte permite que a solução de uma determinada questão possa ser feita em diferentes linguagens de programação, além de contar com opções de ajustes como modo claro e escuro.

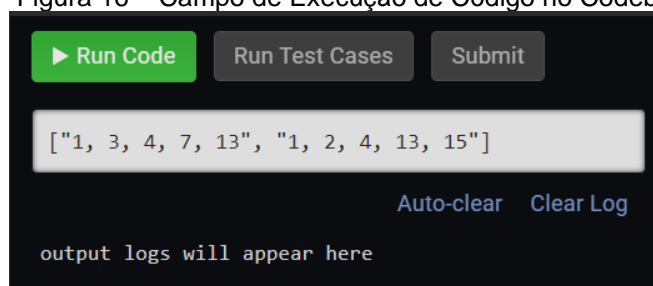
Figura 15 – Campo de codificação de uma questão no Coderbyte



Fonte: Coderbyte (2020)

No lado direito da página, representado na figura 16, nos deparamos com o que acabou sendo uma proposta que serviu de inspiração para o nosso projeto. Nele temos a disponibilização de um campo de *Inputs* que geram *Outputs* para que os usuários que estejam codificando possam testar sua solução e identificar possíveis erros.

Figura 16 – Campo de Execução de Código no Codebyte



Fonte: Coderbyte (2020)

Um ponto forte na integração entre cada uma dessas partes é a possibilidade que o usuário tem de definir qual delas está sendo focada, aumentando e diminuindo cada uma delas como lhe convém.

3.5 Sistema de Provas *Online* com Desenvolvimento em Python com Django

Este projeto, vinculado à UNIRIO, Universidade Federal do Estado do Rio de Janeiro, se trata de um trabalho de conclusão de curso para o título de bacharel em Sistemas de Informação. Tendo como objetivos auxiliar professores e alunos com a disponibilização de provas *online*, integrar o acesso dos alunos ao sistema por meio do Facebook e usar a própria implementação da aplicação como uma forma de estudo da linguagem Python.

Na sua estrutura, existem três atores principais. São eles: o administrador, o professor e o aluno.

Vejamos algumas de suas funcionalidades e como cada um dos atores interage com a ferramenta a fim de compararmos as propostas com aquelas apresentadas pelo PAPO UFF.

O administrador cuida de toda a parte referente a cadastros e permissões da plataforma. Pode definir qual usuário se trata de um professor e cuidar do gerenciamento de turmas, disciplinas, cursos e centros de ensino. Podemos visualizar, na imagem a seguir, o quão abrangente é a função do administrador. Que pode controlar, basicamente, todo o funcionamento da aplicação.

Figura 17 – Tela Principal de Administração do Site pela Visão de Administradores

Administrador Sistema de Provas

Início

Administração do Site

Módulos do Sistema	
Aplicações	
Prova	
Notas	+ Adicionar ≡ Modificar
Provas	+ Adicionar ≡ Modificar
Questões	+ Adicionar ≡ Modificar
Questões e Alunos	+ Adicionar ≡ Modificar
Sistema	
Aluno Disciplinas	+ Adicionar ≡ Modificar
Centros	+ Adicionar ≡ Modificar
Cursos	+ Adicionar ≡ Modificar
Disciplinas	+ Adicionar ≡ Modificar
Turmas	+ Adicionar ≡ Modificar
Usuários	+ Adicionar ≡ Modificar
Administração	
Grupos	+ Adicionar ≡ Modificar
Sites	+ Adicionar ≡ Modificar
Usuários	+ Adicionar ≡ Modificar

Provas
> Correção de Provas
> Relatório de Notas
> Sistema alunos

Ações Recentes
≡ professores Grupo
≡ marcio Usuário
≡ [professor] - Marcio Usuário
≡ professores Grupo
× [aluno] - Usuário

Fonte: Modificado de Vinhaes (2013 p. 41)

O professor tem a função de cadastrar provas, visualizar a lista de alunos que realizaram uma prova, pontuar (corrigir) questões da prova dos alunos e visualizar as notas lançadas no sistema. A figura 18 representa a interface em que o professor poderá editar uma prova, adicionando ou excluindo questões e modificando as informações sobre o prazo para aplicação da prova, turma e semestre associado.

Figura 18 – Tela de Adição, Edição e Exclusão de Dados

Fonte: Modificado de Vinhaes (2013 p. 48)

O aluno, a partir do login via Facebook, consegue se inscrever em uma disciplina, visualizar a lista de provas disponíveis, realizar uma prova e visualizar suas notas. Vejamos as interfaces que serão visualizadas pelo aluno após realização de *login*, no momento da realização de uma prova e visualização de seu desempenho em provas passadas nas figuras 19, 20 e 21, respectivamente:

Figura 19 – Tela de Inscrição de Disciplina

Fonte: Modificado de Vinhaes (2013 p. 52)

Figura 20 – Tela de Realização da Prova



The screenshot shows a web browser window titled "Prova Online Unirio". The address bar shows "Home" and "Notas". The user's name "Aluno: Marina Vinhaes" is visible in the top right corner. The main content area contains two questions, each with a text input field:

- 1 - Pergunta 1 (Valor: 2,0)
- 2 - Pergunta 2 (Valor: 2,0)

At the bottom left, there is a button labeled "Finalizar".

Fonte: Modificado de Vinhaes (2013 p. 53)

Figura 21 – Tela de Gabarito de Provas Realizadas



The screenshot shows a web browser window titled "Prova Online Unirio". The address bar shows "Home" and "Notas". The main content area displays two exam results in separate boxes:

Disciplina	Professor	Prova	Nota	Ação
Banco de Dados	Leonardo Torres	P1	10,0	Visualizar
Gerência de Projetos em Informática	Leonardo Torres	P1	3,0	Visualizar

Fonte: Modificado de Vinhaes (2013 p. 54)

Em geral, apesar de compactuarem do objetivo de servir como auxílio para professores e alunos, as diferenças de cada proposta são bem nítidas. Uma vez que este projeto pretendia criar todo um cenário voltado para o ensino de Python e apresentar integração com Facebook, a parte referente à aplicação das provas conta simplesmente como sendo uma forma de se substituir uma prova impressa.

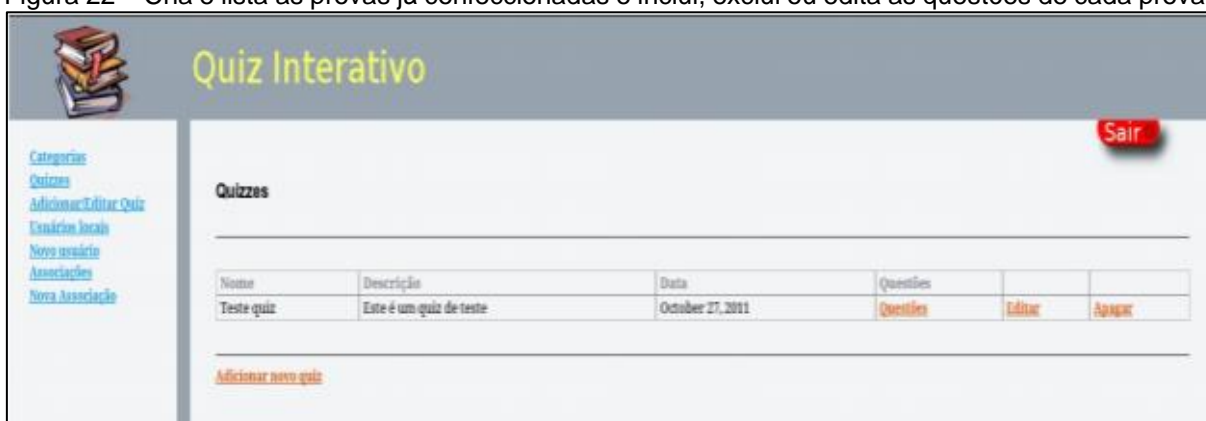
3.6 Cadastramento de Questões e Aplicação de Provas *Online*

Trabalho apresentado ao Colegiado de Computação e Matemática Aplicada do Centro Universitário da Zona Oeste do Rio de Janeiro, como sendo requisito para

obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas. Tendo como objetivo desenvolver um sistema de geração e aplicação de provas *online*.

A aplicação conta com uma divisão entre módulos, havendo o módulo administrador e o módulo aluno. O módulo administrador possibilita a criação de categorias de provas, as provas em si e as suas questões. Além disso, gerencia o cadastro de usuários e as aplicações das provas. Vejamos as interfaces de criação de atividades na figura 22 e criação de novos usuários na figura seguinte:

Figura 22 – Cria e lista as provas já confeccionadas e inclui, exclui ou edita as questões de cada prova.



Fonte: Modificado de DA SILVA (2013 p. 77)

Figura 23 – Criação de novos usuários locais.

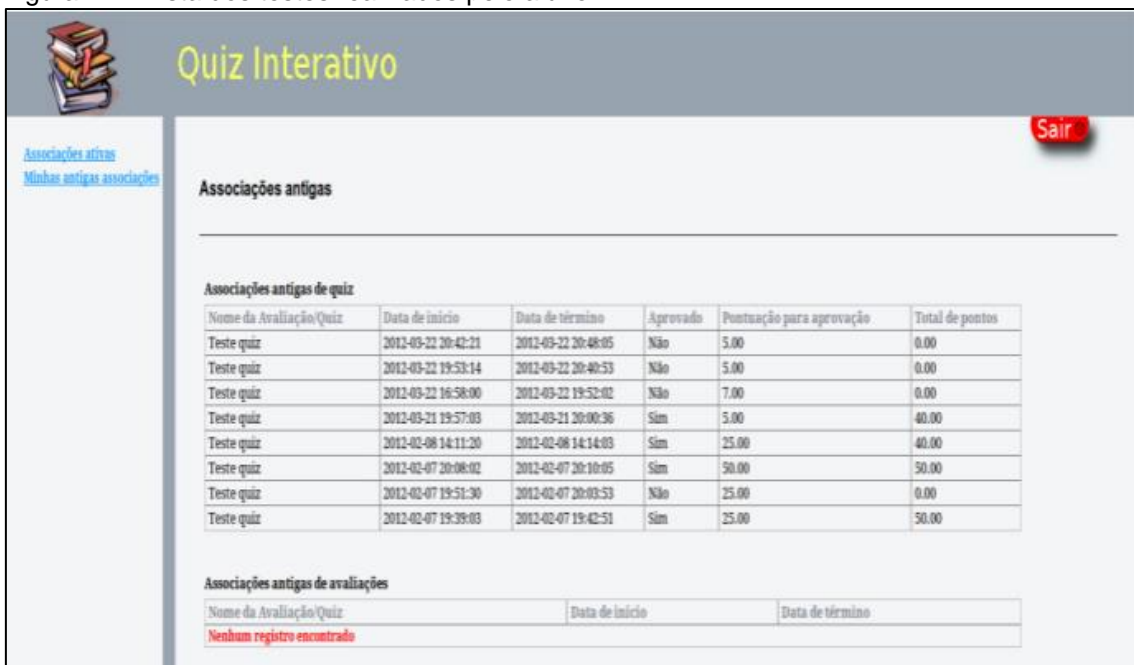


Fonte: Modificado de DA SILVA (2013 p. 85)

Os alunos poderão realizar *login* após cadastro feito pelo administrador. Com o acesso à plataforma, poderão consultar provas ativas, realizar uma dessas provas e visualizar a pontuação de acertos das provas realizadas. Na imagem abaixo, podemos

visualizar a lista de atividades realizadas por um aluno, seu desempenho e data de conclusão:

Figura 24 – Lista dos testes realizados pelo aluno



The screenshot shows a web interface for 'Quiz Interativo'. On the left, there are links for 'Associações ativas' and 'Minhas antigas associações'. The main area is titled 'Associações antigas' and contains a table of 'Associações antigas de quiz'. Below the table, there is a section for 'Associações antigas de avaliações' which currently shows 'Nenhum registro encontrado'.

Nome da Avaliação/Quiz	Data de início	Data de término	Aprovado	Pontuação para aprovação	Total de pontos
Teste quiz	2012-03-22 20:42:21	2012-03-22 20:48:05	Não	5,00	0,00
Teste quiz	2012-03-22 19:53:14	2012-03-22 20:40:53	Não	5,00	0,00
Teste quiz	2012-03-22 16:58:00	2012-03-22 19:52:02	Não	7,00	0,00
Teste quiz	2012-03-21 19:57:03	2012-03-21 20:00:36	Sim	5,00	40,00
Teste quiz	2012-02-08 14:11:20	2012-02-08 14:14:03	Sim	25,00	40,00
Teste quiz	2012-02-07 20:08:02	2012-02-07 20:10:05	Sim	50,00	50,00
Teste quiz	2012-02-07 19:51:30	2012-02-07 20:03:53	Não	25,00	0,00
Teste quiz	2012-02-07 19:39:03	2012-02-07 19:42:51	Sim	25,00	50,00

Associações antigas de avaliações

Nome da Avaliação/Quiz	Data de início	Data de término
Nenhum registro encontrado		

Fonte: Modificado de DA SILVA (2013 p. 97)

Sendo outro trabalho com intuito de contribuir de alguma forma com a aplicação de exercícios e provas em um cenário *online*. Podemos notar que não há um público-alvo para o qual esta aplicação tenha sido direcionada. Dessa forma, a proposta seria, novamente, apenas substituir a prova em papel no cenário educacional geral.

A fim de se ter uma visão geral a respeito da relação entre cada uma das ferramentas apresentadas e o PAPO UFF, foi feito um comparativo de suas respectivas características de forma esquematizada, ilustradas por meio da figura 25.

Figura 25 – Comparativo entre ferramentas disponíveis atualmente e o PAPO UFF

Características	Ferramentas						
	GoogleClassroom	URI	BOCA	Coderbyte	Trabalho UNIRIO	Trabalho UEZO	PAPO UFF
Perfil de Professor	X	X			X	X	X
Perfil de Aluno	X	X			X	X	X
Criação de Turmas	X	X	X				
Questões de programação		X	X	X			X
Questões Objetivas	X				X	X	X
Questões discursivas	X				X	X	X
Banco de questões		X		X	X	X	X
Correção automática	X	X		X			X
Estatísticas pós prova							X

Fonte: Autoria própria

4 METODOLOGIA

A primeira ação tomada após passarmos pelas etapas de pesquisa e debate foi delimitar com precisão qual seria o nosso escopo de atuação. Pretendíamos, a partir daí, desenvolver um programa completo e funcional que pudesse enfatizar os benefícios que uma ferramenta digital pode proporcionar para a área da educação. Essa delimitação foi feita através de duas etapas: a primeira, onde analisamos os programas já em vigor no mercado que visam recriar de alguma maneira um ambiente avaliativo no cenário digital e, a segunda, foi a análise das estratégias que de fato foram adotadas pelas escolas para contornar o empecilho de não poderem ter aulas presenciais.

Do ponto de vista dos programas avaliados, conseguimos extrair alguns conceitos e definir alguns pontos que poderiam, com certa adaptação, ser integrados ao PAPO UFF.

Em relação às escolas, notamos que não houve unanimidade em relação a maneira que cada instituição se adaptou a esta nova forma de lecionar, justamente por conta de cada etapa do ensino demandar necessidades diferentes para alunos e professores.

Sendo assim, pudemos concluir que, caso quiséssemos elaborar uma solução que abrangesse todos esses casos, talvez fosse necessário criar ferramentas dedicadas e não integrar todas as funcionalidades em somente uma plataforma. Portanto, delimitamos nossa área de atuação como sendo uma plataforma *online* que teria seu público-alvo os estudantes universitários do curso de Ciência da Computação.

Após a definição da área de atuação, passamos a buscar pelas ferramentas que nos proporcionariam o devido suporte para desenvolvermos o ambiente online idealizado, ou seja, as bibliotecas e *frameworks* do projeto. Independentemente da nossa escolha, teríamos que lidar com uma tecnologia nova e totalmente diferente do que havia sido feito até então. Visto que não tínhamos muita experiência com programação voltada para a *web*, fatores como a curva de aprendizado, consolidação e abrangência da tecnologia no mercado foram considerados como pontos importantes para realizarmos nossa escolha. Fazendo um estudo comparativo das tecnologias disponíveis, escolhemos o React como a biblioteca responsável por toda a parte de *front-end* do PAPO UFF justamente por mostrar uma certa superioridade

em relação às outras opções nos quesitos de área de atuação no mercado, APIs (do inglês *Application Programming Interface*, ou seja, Interface de Programação de Aplicação) disponíveis e ainda apresentar uma comunidade bastante ativa e ter seu código aberto, além de ter uma base de sustentação muito sólida por estar relacionada com uma grande empresa como o Facebook.

Uma vez tendo definida a parte referente ao *front-end* do projeto, a escolha do *back-end* foi tomada de maneira mais fluída e natural, visto que, o próprio React acabou exercendo um certo direcionamento ao Node.js pelo grau de interação entre as duas tecnologias. Além dessa boa relação entre as ferramentas que comporiam cada parte do projeto, o Node.js se mostrou uma excelente opção por sua leveza, flexibilidade e produtividade.

A tecnologia utilizada para criação do banco de dados foi basicamente a mesma já adotada no próprio protótipo, isto é, por meio do XAMPP e MySQL, porém, toda sua estrutura foi redefinida para atender a este novo escopo.

Todas as funcionalidades foram definidas com base nas ações que seriam designadas ao professor e ao aluno, de forma que pudéssemos manter a ideia de um ciclo de atuação fechado e que consequentemente atendesse ao escopo definido de forma completa, porém, ainda mantendo a noção de que novas funcionalidades poderiam ser integradas.

Ao decorrer do processo de desenvolvimento do programa, planejamos um teste prático a fim de obtermos uma análise empírica sobre a efetividade e viabilidade do uso do PAPO UFF no cenário real da universidade. O teste consistia em colocar o programa em uso no cenário real de uma turma. Além disso, deveria ser feito o preenchimento de um questionário (disponível no Apêndice D) pelos alunos e professores participantes da experiência.

Com o intuito de analisarmos o histórico de desempenho dos alunos de computação, requisitamos à instituição os dados referentes às notas dos períodos anteriores. Dessa maneira, pretendíamos realizar uma análise do impacto da mudança das provas de papel para as atividades acadêmicas online devido a este momento de pandemia.

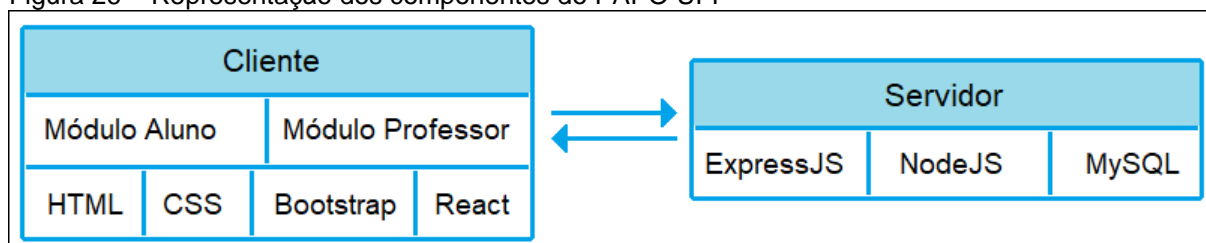
5 DESCRIÇÃO DO PROJETO

Tendo concluído o levantamento de tecnologias, listado as funcionalidades e definido as estruturas que seriam necessárias para o desenvolvimento do projeto, pudemos enfim iniciar sua implementação prática. A base organizacional do trabalho foi feita a partir dos conceitos que estariam divididos em duas vertentes, as que seriam ligadas ao professor e àquelas relacionadas com o aluno.

5.1 Organização Interna do PAPO UFF

Vejamos a partir de agora como ficou designada cada uma das partes de *front-end* e *back-end* do projeto na composição do todo e como se deu a relação entre elas. Através da figura abaixo, visando esquematizar a relação entre cada componente que compõe a estrutura do PAPO UFF, podemos resumir cada um dos tópicos que serão abordados na sequência:

Figura 26 – Representação dos componentes do PAPO UFF



Fonte: Autoria própria

5.1.1 *Front-end* do Projeto

Toda a parte referente ao *front-end* foi baseada no React. Com isso, conseguimos desenvolver as interfaces do projeto mesclando nossas ideias próprias de *layouts* e a utilização de diversas APIs e *frameworks*. Para visualizar o código referente ao front-end do projeto na íntegra, deve-se acessar o repositório correspondente no GitHub⁸. Listamos abaixo alguns desses componentes e suas respectivas funções no funcionamento do PAPO UFF.

⁸ Disponível em: <<https://github.com/Drakz/PAPO-UFF-Front>>. Acesso em 3 mai. 2021.

- React Router: trata-se de uma série de componentes de navegação que, ao ser integrado ao projeto, permitiu que todas as rotas para acessar cada página fosse definida de forma correta, dando fluidez e navegabilidade ao PAPO UFF.
- CanvasJs: uma API que lida com variadas questões referentes a gráficos e exposição de dados. Mostrou-se uma ferramenta perfeita para as nossas intenções em apresentar as análises das provas para o professor.
- React Compound Timer: componente que disponibiliza uma série de funções relacionadas a temporizadores. Foi utilizada para realizarmos o monitoramento do tempo gasto em cada questão por cada aluno em uma prova. Além disso, também foi responsável pela parte de limitar o tempo de prova.
- React Codemirror: componente capaz de manter a formatação de um texto baseando-se no que temos em IDEs, ou seja, trata-se de um editor de código, gerando enumeração de linhas e marcações. Foi responsável pelo campo de código em uma questão de programação.
- React Bootstrap: coleção de componentes de interface. Foram utilizados diversos desses componentes na formatação das páginas do PAPO UFF e na montagem de seu *layout*, a fim de que este fosse claro, agradável e minimalista.

5.1.2 *Back-end* do Projeto

Utilizamos como base para o *back-end* do nosso projeto o NodeJs, mais especificamente, um *framework* dedicado a ele, o ExpressJs⁹. Dessa forma, conseguimos manter toda a interação entre cada parte do programa funcionando da forma que necessitávamos, monitorando as requisições realizadas por um aluno ou professor, buscando informações no banco de dados e provendo todas as funcionalidades que foram definidas para cada uma das páginas do PAPO UFF.

Na prática, a comunicação cliente-servidor é feita diretamente com o *back-end*. Isto é, o cliente entra em contato com o *back-end* de duas maneiras distintas: de forma

⁹ Disponível em: <<https://expressjs.com/pt-br/>>. Acesso em 28 mar. 2021.

ativa por meio de um *proxy* (servidor com papel intermediário no atendimento de chamadas e requisições de clientes por recursos de outros servidores) ou de maneira passiva, por meio de uma API. Assim, o *back-end* será capaz de identificar qual tipo de requisição foi realizada. Caso tenha sido feita de forma direta, esse irá disponibilizar as informações que deverão aparecer na tela do usuário. Caso seja uma chamada da API, irá atender à requisição. Toda a codificação que compõe o back-end do trabalho pode ser consultada através do repositório correspondente no GitHub.

Além disso, para realização de testes e aplicação de uma prova para vários alunos ao mesmo tempo, contamos com o ngrok¹⁰, uma aplicação que nos permitiu disponibilizar o PAPO UFF *online* a partir de uma porta da nossa máquina local de forma simples.

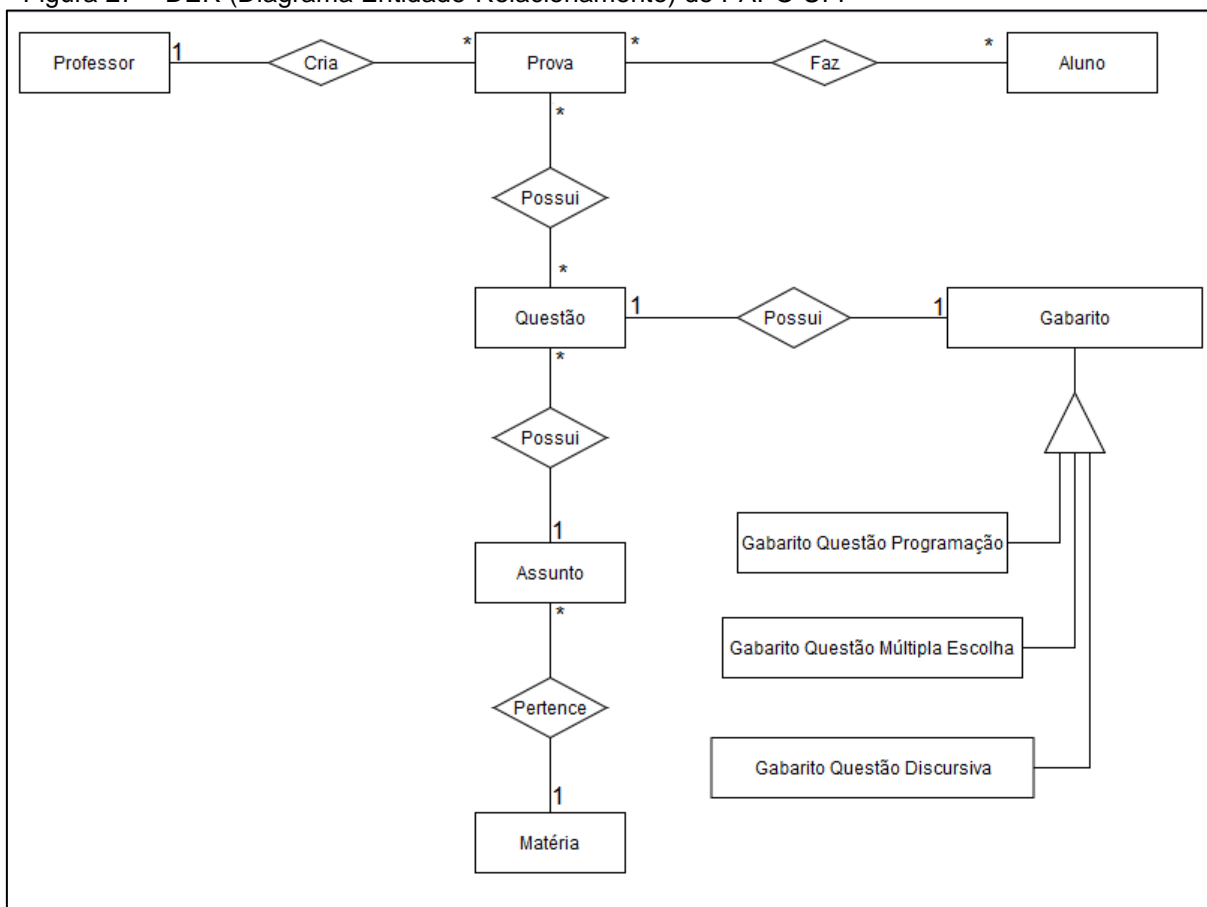
5.1.3 Banco de Dados

A criação do banco de dados e definição de sua organização foi feita através da avaliação dos objetos e requisitos que deveriam ser armazenados, definição de como seria a relação entre eles e como seria realizado o acesso a estas informações. A implementação do banco de dados foi feita em MySQL, por meio do phpMyAdmin. Para criarmos a estrutura do banco de dados, baseado no MER (Modelo Entidade-Relacionamento), utilizamos a aplicação sqlDBM¹¹. Vejamos como ficou o diagrama entidade-relacionamento do banco de dados implementado por meio da figura 27:

¹⁰ Disponível em: <<https://ngrok.com/>>. Acesso em 4 abr. 2021.

¹¹ Disponível em: <<https://sqldb.com/Home/>>. Acesso em 28 mar. 2021.

Figura 27 – DER (Diagrama Entidade-Relacionamento) do PAPO UFF

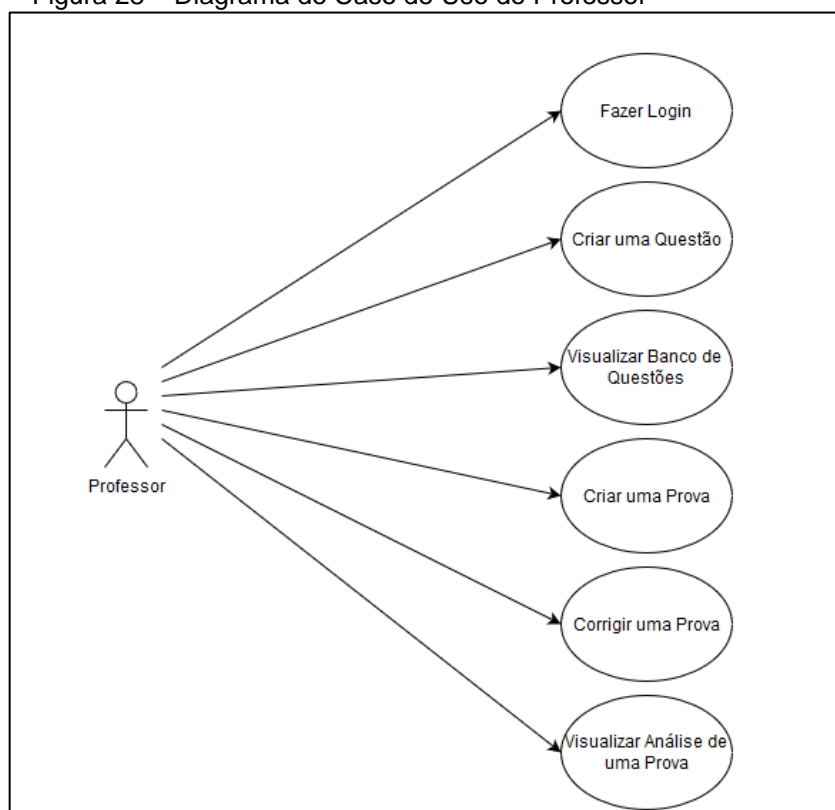


Fonte: Autoria própria

5.1.4 Atribuições Referentes ao Professor

Sendo uma das bases do nosso projeto, a figura do professor no nosso trabalho está relacionada com a maioria de suas funcionalidades. A fim de observarmos a maneira com que interage com o sistema, visualizemos o diagrama de casos de uso abaixo:

Figura 28 – Diagrama de Caso de Uso do Professor



Fonte: Autoria própria

A interação do professor com cada caso citado se dá da seguinte maneira:

- **Fazer *login*:** o professor deverá informar seu nome e senha para seguir com a autenticação no sistema. Todo o conjunto universo dos professores é definido de forma intrínseca ao sistema, ou seja, todos os professores que farão uso do PAPO UFF deverão ser cadastrados previamente pois recebem uma permissão específica para desempenharem o seu papel. Não há possibilidade de requisição ou pedido pela própria plataforma para que um usuário ganhe autenticidade de professor.
- **Criar uma questão:** inserção de uma nova questão no banco de questões. O professor poderá criar uma questão ao visualizar a aba do banco de questões ou ao criar uma prova.
- **Visualizar banco de questões:** todas as questões criadas poderão ser visualizadas através do banco de questões. Com isso, criamos uma espécie de ambiente colaborativo entre professores para que uma questão possa ser

reaproveitada por qualquer outro professor. Entretanto, nenhum desses professores tem permissão para editar ou apagar uma questão criada.

- Criar uma prova: pode ser interpretada como uma seleção de questões que irão ser aplicadas como uma atividade. Pode ser criada a partir de questões totalmente novas, definidas no momento da criação da prova em si, e que depois serão salvas no sistema, ou a partir daquelas já criadas previamente que estão no banco de questões.

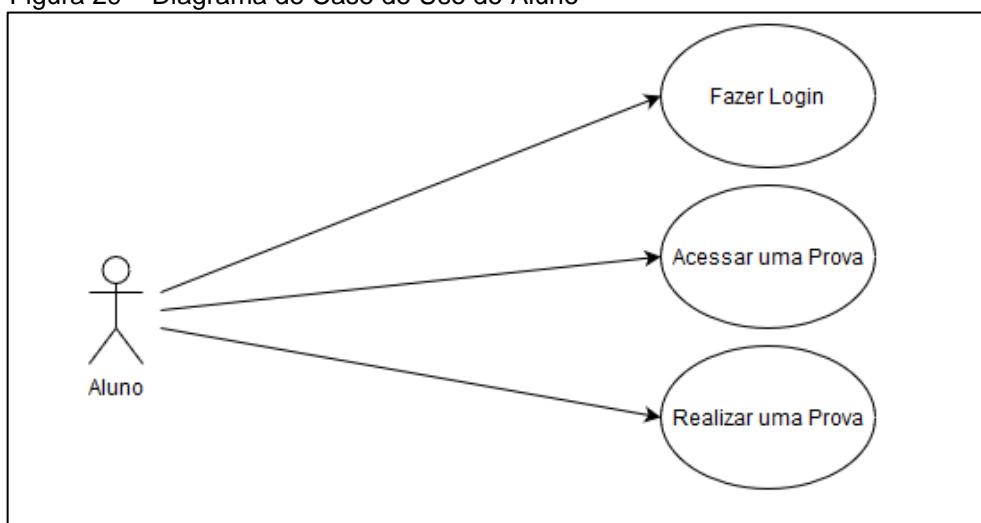
- Corrigir uma prova: o professor dará o aval para encerrar a fase de correção de uma prova. Inicialmente, o sistema de correção fará uma avaliação e indicará, quando possível, o desempenho de cada aluno para uma dada prova, entretanto, aquelas questões que o sistema não conseguir definir uma nota (como as dissertativas interpretativas, de resposta não exata) deverão aguardar um posicionamento do professor.

- Visualizar análise de uma prova: o professor terá acesso a toda análise feita pelo sistema com base no desempenho dos alunos nas provas. Poderá visualizar os gráficos informativos, questões em destaque e o diário de classe, onde as notas dos alunos serão expostas de forma organizada e clara.

5.1.5 Atribuições Referentes ao Aluno

Sendo o outro pilar do PAPO UFF, a figura do aluno complementa o professor na composição das funcionalidades do sistema. Visualizemos abaixo o diagrama de classes correspondente ao seu papel na aplicação:

Figura 29 – Diagrama de Caso de Uso do Aluno



Fonte: Autoria própria

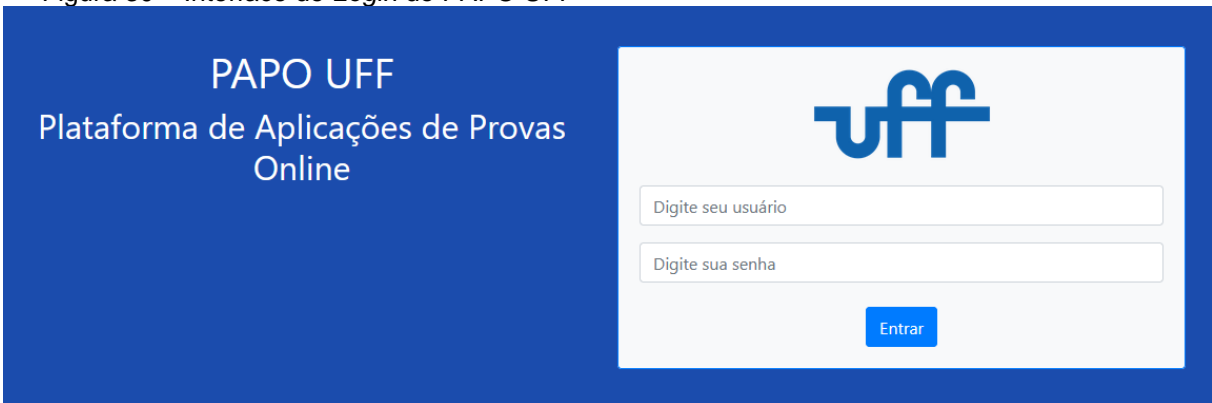
Podemos notar que as funções relacionadas com a figura do aluno buscam única e exclusivamente gerar um volume de dados que poderá ser avaliado posteriormente no momento da correção da prova. A interação com cada caso citado se dá da seguinte maneira:

- Fazer login: o aluno é cadastrado no sistema por meio do seu nome e CPF no momento em que acessa a plataforma pela primeira vez. Sendo assim, todo novo usuário que pretenda acessar o PAPO UFF terá plenas condições de fazê-lo caso seja um aluno.
- Acessar uma prova: o aluno deverá ser informado a respeito do número de identificação de uma prova para que possa acessá-la. Este número é de posse do professor, gerado no momento da criação da prova.
- Realizar uma prova: ao acessar uma prova por meio do seu número de identificação, o aluno terá permissão para, dentro do tempo estipulado, responder cada uma das questões. Neste momento, o aluno terá disponível o recurso de compilação e execução de código para as questões de programação.

5.2 Descrição das Funcionalidades e Interfaces

Vejam os a partir de agora como cada funcionalidade foi implementada e como se relacionam com as interfaces do PAPO UFF. A fim de facilitar o entendimento, separamos a disposição das imagens entre aquelas relacionadas com o professor e aquelas relacionadas com o aluno. Todas as imagens estão aproximadas para que se tenha melhor visualização de seus principais elementos, portanto, para visualizar as telas em seu tamanho real, vide Apêndice C. A única interface comum entre as duas partes é justamente a de efetuar *login*, representada na figura 30.

Figura 30 – Interface de *Login* do PAPO UFF



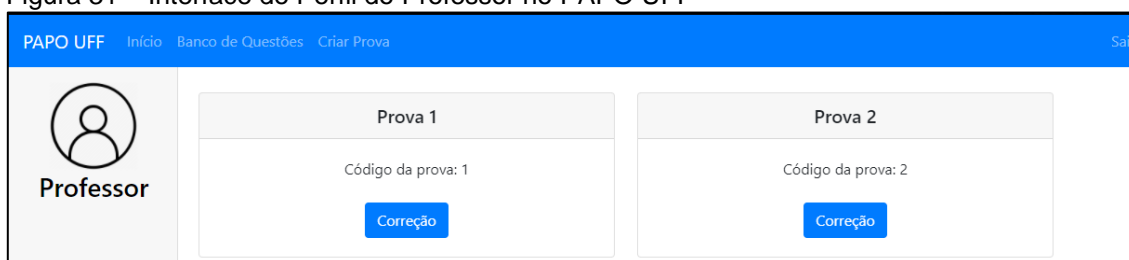
Fonte: Autoria própria

5.2.1 Interfaces atreladas ao Professor

Ao efetuar login, o professor será direcionado para a tela de perfil, como poderemos visualizar na figura 31. Veremos que, a partir daí, o professor poderá navegar por todas as demais funcionalidades que lhe são permitidas por meio de uma barra de menu que se encontra fixa na parte superior da página.

Referente a aba de “Início”, podemos notar elementos de identificação tais como o nome e uma imagem associada ao professor que está fazendo o acesso. Nessa tela, este professor poderá visualizar todas as provas criadas por ele de acordo com os seus respectivos títulos e códigos de acesso.

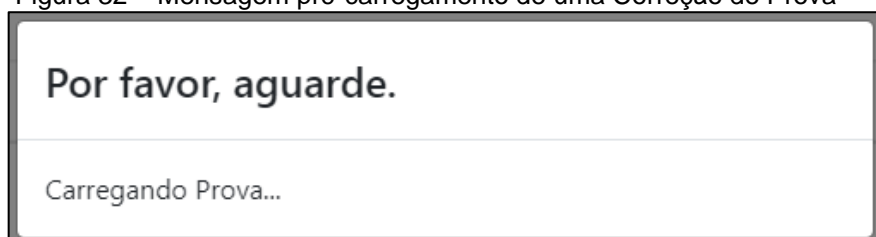
Figura 31 – Interface de Perfil do Professor no PAPO UFF



Fonte: Autoria própria

Ao selecionar o botão “correção” em uma prova, o professor será direcionado para a página que contém as informações específicas referentes à prova selecionada. É nesta página que ocorre a correção automática das questões e o gráfico de análise é gerado. Enquanto esta correção é feita, uma mensagem informa o professor que o processo está sendo executado.

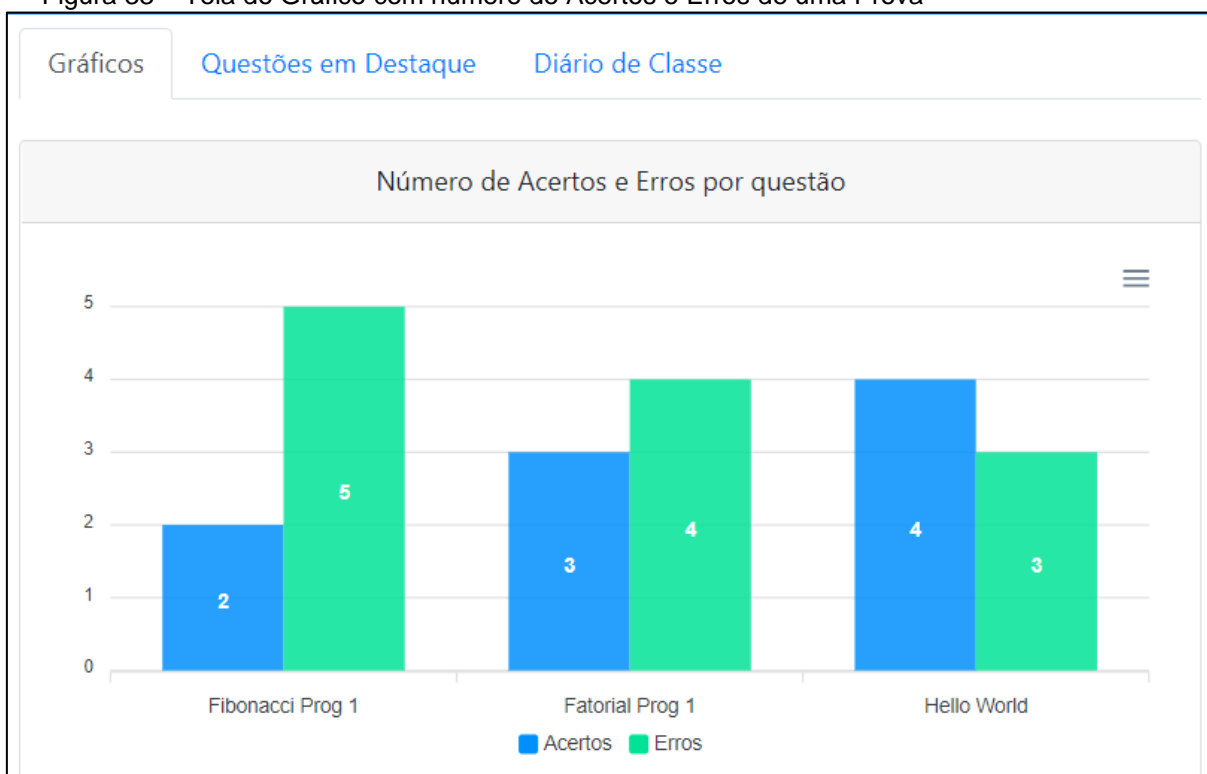
Figura 32 – Mensagem pré-carregamento de uma Correção de Prova



Fonte: Autoria própria

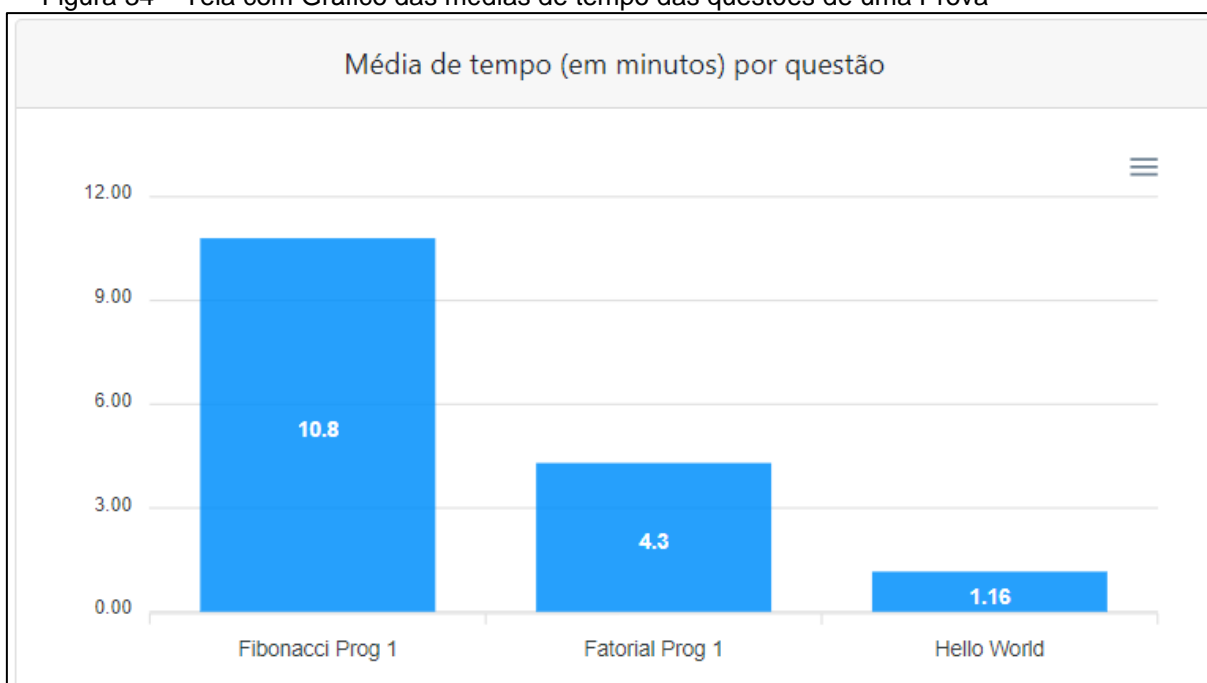
Após concluído o processo de correção das provas, os gráficos de análise serão exibidos, como ilustrado nas figuras 33, 34 e 35. Esses gráficos informam a média de tempo em que cada questão foi feita, a média de compilações das questões de programação e o número de acertos e erros de cada questão.

Figura 33 – Tela do Gráfico com número de Acertos e Erros de uma Prova



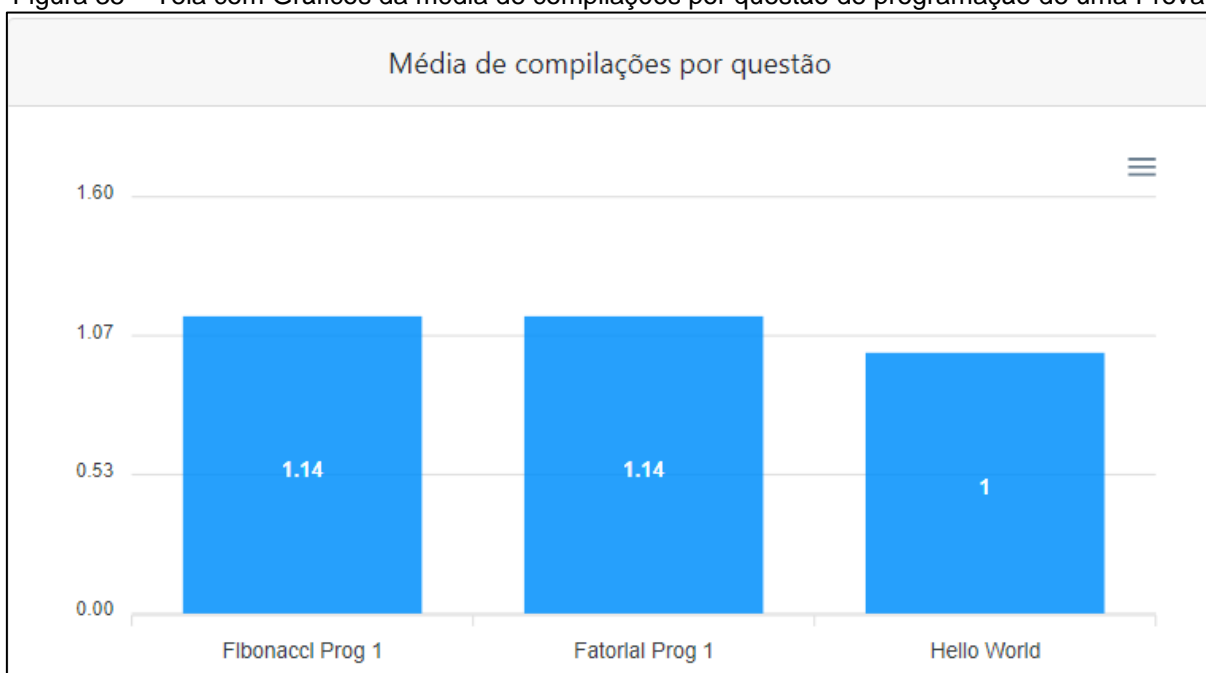
Fonte: Autoria própria

Figura 34 – Tela com Gráfico das médias de tempo das questões de uma Prova



Fonte: Autoria própria

Figura 35 – Tela com Gráficos da média de compilações por questão de programação de uma Prova



Fonte: Autoria própria

Além disso, após o processo de correção, o professor poderá também visualizar as questões em destaque. Podemos notar, por meio da figura 36, que foram consideradas questões de destaque: aquela que apresentar maior tempo médio para ser respondida, menor tempo médio para ser respondida, a questão com mais acertos e a com menos acertos.

Figura 36 – Questões em Destaque de uma Prova



Fonte: Autoria própria

No diário de classe, o professor poderá visualizar todas as notas dos alunos nessa prova. Vejamos:

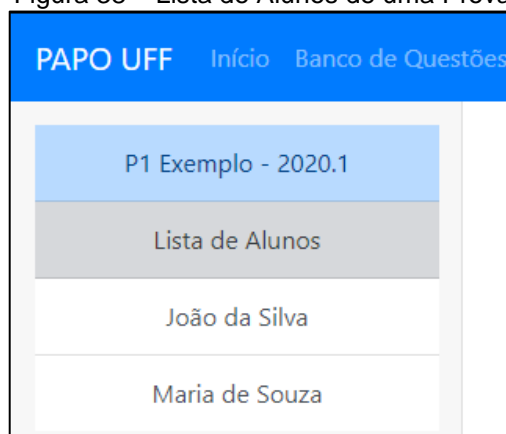
Figura 37 – Relação de Notas de uma Prova

Gráficos Questões em Destaque Diário de Classe	
Nome do Aluno	Nota
João da Silva	78
Maria de Souza	85

Fonte: Autoria própria

Ainda na página da prova selecionada, ao lado esquerdo, o professor poderá visualizar uma lista com o nome de cada aluno que realizou a prova.

Figura 38 – Lista de Alunos de uma Prova



Fonte: Autoria própria

Ao selecionar um desses alunos, poderá visualizar sua prova completa, com a lista de questões com as respostas deste aluno, além da nota definida pela correção automática.


Figura 39 – Lista de Respostas de um Aluno

Testando discursiva 2
Teste Programação 1
Testando discursiva 1
Teste Discursiva 3

Fonte: Autoria própria

Caso seja necessário alterar alguma nota de uma dada questão, o professor poderá fazê-lo ao acessá-la, alterar o campo de “Nota desta Questão” e clicar em “Salvar Nota”.

Figura 40 – Campo de Alteração de Nota em uma Questão de um Aluno

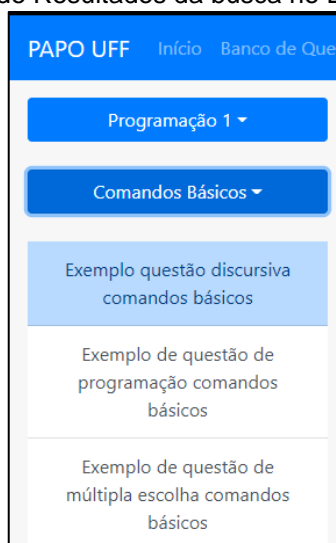


Nota desta Questão: [Salvar Nota](#)

Fonte: Autoria própria

Na aba “Banco de Questões”, o professor poderá visualizar, de acordo com a matéria e assunto, todas as questões que estão cadastradas no sistema. Todas essas questões apresentam um título único, enunciado e resposta, além de uma categorização referente à matéria, ao assunto, à dificuldade e ao seu tipo (que pode ser discursiva, múltipla escolha e programação). A seguir, podemos visualizar, através da figura 41, a lista de questões cadastradas no banco e, por meio da figura 42, como se dá a apresentação de uma dessas questões para visualização do professor nesta mesma tela.

Figura 41 – Lista de Resultados da busca no Banco de Questões



PAPO UFF Início Banco de Questões

Programação 1 ▾

Comandos Básicos ▾

Exemplo questão discursiva comandos básicos

Exemplo de questão de programação comandos básicos

Exemplo de questão de programação comandos básicos

Exemplo de questão de múltipla escolha comandos básicos

Fonte: Autoria própria

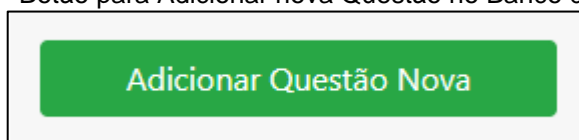
Figura 42 – Visualização de uma Questão do Banco de Questões

Matéria	Assunto	Tipo	Dificuldade
Programação 1 ▾	Vetores ▾	Discursiva ▾	Fácil ▾
Título			
Exemplo questão discursiva vetor			
Enunciado			
Exemplo questão discursiva			
Resposta correta: Exemplo questão discursiva			

Fonte: Autoria própria

Além de visualizar as questões cadastradas, o professor também pode criar uma nova questão nessa mesma aba, necessitando apenas inserir os mesmos campos que foram citados anteriormente. Assim, essa nova questão poderá ser usada futuramente para a criação de uma prova deste ou até mesmo de outros professores. Vejamos a sequência de imagens que demonstra o processo para adicionar uma questão no banco de questões:

Figura 43 – Botão para Adicionar nova Questão no Banco de Questões



Fonte: Autoria própria

Figura 44 – Criação de Nova Questão Para adição direta no Banco de Questões

O formulário "Adicionar Questão no Banco" apresenta os seguintes elementos:

- Matéria:
- Assunto:
- Tipo:
- Dificuldade:
- Título:
- Enunciado:
- Botões: "Editar Resposta" (azul) e "Adicionar no Banco" (verde).

Fonte: Autoria própria

Figura 45 – Campos de Respostas na Criação de Nova Questão Para adição no Banco de Questões

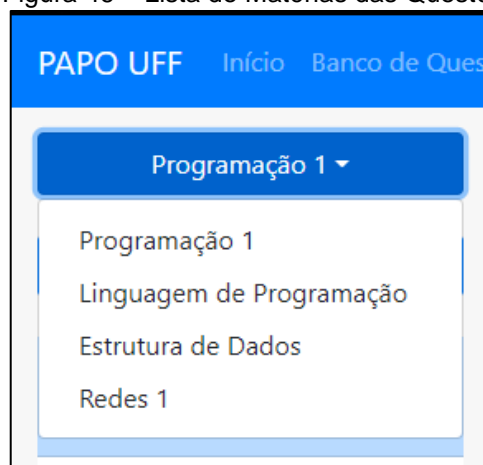
O formulário "Adicionar Questão no Banco" focado nos campos de resposta apresenta os seguintes elementos:

- Enunciado:
- Resposta:
- Botão: "Salvar" (verde).

Fonte: Autoria própria

As matérias que podem ser selecionadas são definidas de forma intrínseca ao sistema, ou seja, compõem um conjunto universo fechado. Estas matérias foram categorizadas com base na grade do curso de Ciência da Computação da Universidade Federal Fluminense de Rio das Ostras no período de 2020.1. Na figura 46, temos a representação de como o professor poderá visualizar essa lista de matérias.

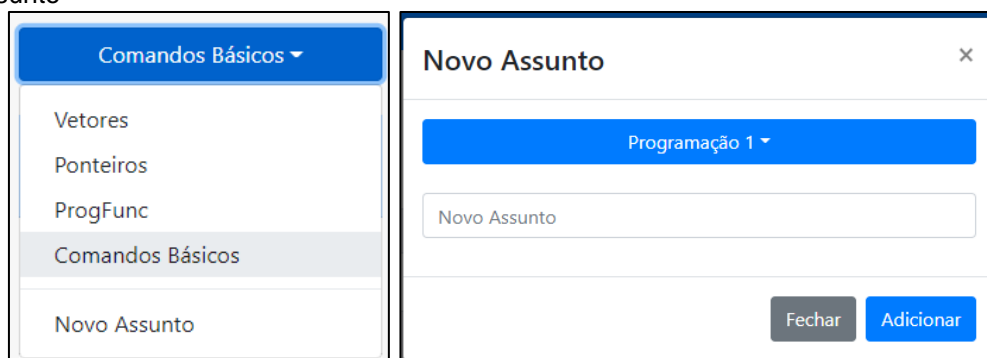
Figura 46 – Lista de Matérias das Questões cadastradas no Banco



Fonte: Autoria própria

Por outro lado, os assuntos podem ser adicionados à medida que o professor julgar necessário, visando sempre uma melhor organização e identificação das questões cadastradas. Vejamos:

Figura 47 – Lista de Assuntos das Questões cadastradas no Banco de Questões e Adição de novo Assunto



Fonte: Autoria própria

Na aba “Criar Prova”, como representado na figura 48, o professor irá se deparar com uma interface composta de um campo para definir o nome da prova e três botões que tratam, respectivamente, da criação de uma nova questão, da busca de uma questão no banco e da finalização do processo de criação da prova.

Figura 48 – Aba de Criação de uma nova Prova

Fonte: Autoria própria

O botão “Nova questão” trata do caso onde o professor queira criar uma questão totalmente nova ao criar uma prova, ou seja, ele não irá aproveitar uma questão já criada que esteja salva no sistema. Ao ser selecionado, será gerada uma questão em branco onde o professor deverá inserir todos os campos assim como feito na aba do “Banco de Questões”. Nesta aba, entretanto, por estar associada a uma prova, algumas outras informações são requisitadas.

Figura 49 – Criação de uma Nova Questão no ambiente de Criação de Prova

Fonte: Autoria própria

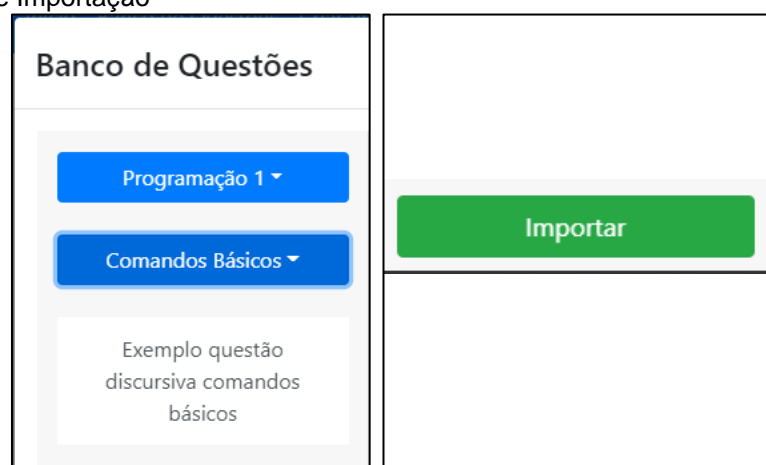
Relacionadas diretamente com a prova, estas outras informações tratam do valor dessa questão no escopo da prova e o número de compilações que o aluno poderá realizar no momento da aplicação (para aquelas do tipo programação) e devem ser inseridas ao selecionar “Editar Resposta”.

Figura 50 – Definição dos dados de uma Questão para uma Prova

Fonte: Autoria própria

O botão “Banco de Questões” possibilita ao professor importar uma questão que esteja salva no sistema para a prova que está sendo criada. O professor poderá visualizar todo o conteúdo da questão antes de adicioná-la à prova selecionando o botão “importar”. E para que seja salva, basta realizar a inclusão dos campos referentes a esta questão no escopo da prova criada, como comentado anteriormente.

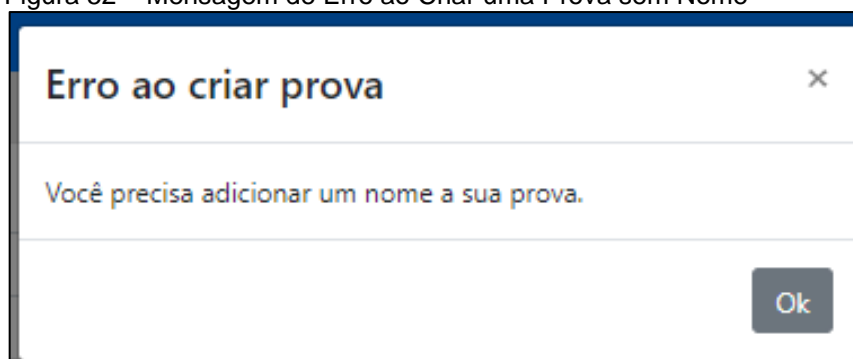
Figura 51 – Lista de questões do Banco de Questões visualizadas na Criação de Provas e Botão de Importação



Fonte: Autoria própria

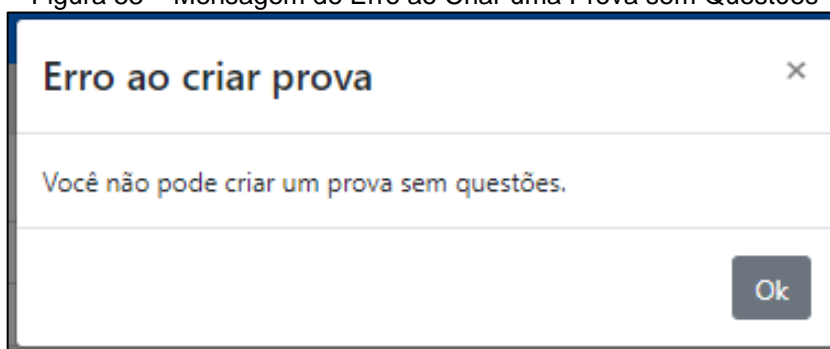
O botão “Finalizar prova” deve ser selecionado quando toda a estrutura da prova for terminada. Nesse momento, será checado se houve algum erro no escopo da prova ou se existe alguma informação pendente. Caso algum problema seja encontrado, o professor será informado por mensagens de alerta.

Figura 52 – Mensagem de Erro ao Criar uma Prova sem Nome



Fonte: Autoria própria

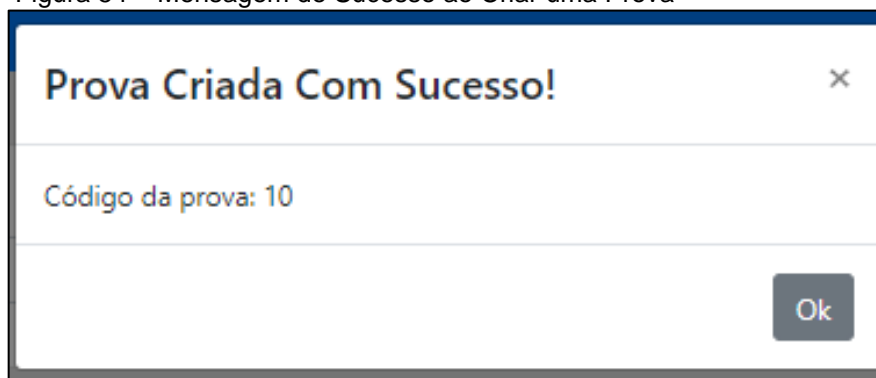
Figura 53 – Mensagem de Erro ao Criar uma Prova sem Questões



Fonte: Autoria própria

Caso a verificação dê o aval para criação de uma nova prova, uma mensagem de confirmação será apresentada e o código de acesso dessa prova será informado.

Figura 54 – Mensagem de Sucesso ao Criar uma Prova



Fonte: Autoria própria

5.2.2 Interfaces atreladas ao Aluno

Ao realizar *login*, o aluno será direcionado para a tela de acesso de provas, ilustrada na figura 55. Por meio dela, conseguirá acessar a sua prova desde que detenha o código de acesso correto disponibilizado pelo professor.

Figura 55 – Tela de Inserção de Código de Prova

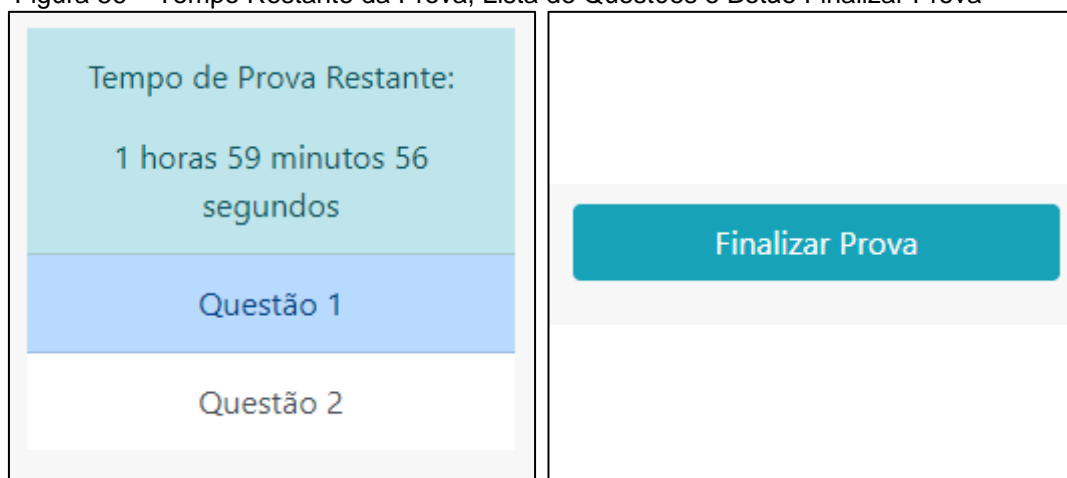


A interface de usuário para a inserção do código de prova. No topo, há o logotipo "PAPOuff" em azul. Abaixo dele, uma mensagem de boas-vindas: "Bem vindo, 'Nome do Aluno!'". O campo principal contém o texto "Digite o código da prova" e um campo de entrada com o exemplo "Ex.: 12345". Um botão azul com o texto "Entrar" está posicionado abaixo do campo de entrada. Na base da tela, há uma linha de texto: "Em caso de dúvidas ou problemas, por favor, comunique ao professor."

Fonte: Autoria própria

Ao inserir o código da prova e clicar em “Entrar”, o aluno será direcionado para a prova correspondente. Nessa tela é possível notar o tempo de prova disponível, uma lista com todas as questões e um botão para finalizar a atividade e entregar a prova.

Figura 56 – Tempo Restante da Prova, Lista de Questões e Botão Finalizar Prova

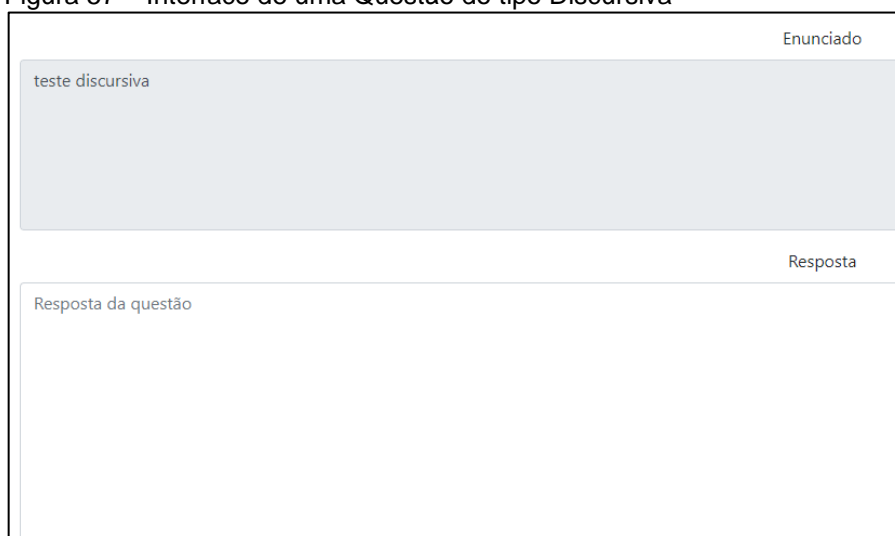


A interface de usuário durante a prova. À esquerda, há um painel com o texto "Tempo de Prova Restante:" seguido de "1 horas 59 minutos 56 segundos". Abaixo disso, há uma lista de questões com "Questão 1" e "Questão 2". À direita, há um botão azul com o texto "Finalizar Prova".

Fonte: Autoria própria

Além desses elementos, ao selecionar uma questão, o aluno poderá visualizá-la de acordo com o seu tipo. As figuras 57, 58 e 59 representam, respectivamente, uma questão discursiva, múltipla escolha e programação. Questões do tipo discursiva contam apenas com um campo de enunciado e uma área reservada para o aluno inserir sua resposta.

Figura 57 – Interface de uma Questão do tipo Discursiva



A interface de uma questão discursiva é apresentada em um formulário dividido em duas seções principais. A seção superior, rotulada "Enunciado", contém o texto "teste discursiva". A seção inferior, rotulada "Resposta", contém o texto "Resposta da questão".

Fonte: Autoria própria

Questões de múltipla escolha contém um enunciado e as alternativas. O aluno deverá marcar a caixa correspondente a alternativa de acordo com o requisitado.

Figura 58 – Interface de uma Questão do tipo Múltipla Escolha



A interface de uma questão de múltipla escolha apresenta três alternativas, cada uma com uma caixa de seleção vazia à esquerda. As alternativas são:

- teste mult a
- teste mult b
- teste mult c

Fonte: Autoria própria

Por fim, as questões do tipo programação são aquelas que apresentam maior diferencial em sua estrutura. Além de contar com o enunciado, assim como todas as outras, este tipo de questão conta com um campo "código", local onde o aluno deverá escrever sua resposta, neste caso, seu programa na linguagem C. Este campo, diferentemente do campo de resposta de uma questão discursiva, visa disponibilizar para o aluno a mesma estrutura que se é vista em IDEs. Isto é, o seu código irá contar com um campo de alto contraste, numeração de linhas e *highlights* (expressão em inglês que significa destaque, uma maneira de diferenciar elementos que compõem o código).

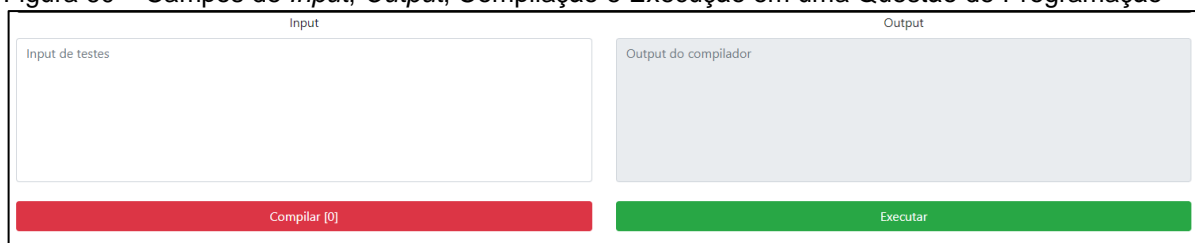
Figura 59 – Campo para codificação em uma Questão de Programação



```
1 #include <stdio.h>
2
3 int main(){
4     printf("PAPO UFF");
5     return 0;
6 }
```

Fonte: Autoria própria

Além disso, o aluno poderá compilar e executar seu código. O número de compilações é definido pelo professor no momento da criação da prova. O resultado da compilação e da execução do código pode ser visualizado pelo campo de *Output* e, caso seja necessário algum *Input*, este pode ser inserido no campo dedicado.

Figura 60 – Campos de *Input*, *Output*, Compilação e Execução em uma Questão de Programação

The interface consists of two main input areas at the top: 'Input' (containing 'Input de testes') and 'Output' (containing 'Output do compilador'). Below these are two buttons: a red 'Compilar [0]' button and a green 'Executar' button.

Fonte: Autoria própria

Uma vez que a prova seja terminada, o aluno deverá selecionar o botão “Finalizar Prova” para entregá-la. Após acionado, será mostrada uma mensagem de confirmação para que a prova seja de fato concluída e as informações salvas para eventual correção futura.

Figura 61 – Mensagem de Confirmação para Entrega de uma Prova



The dialog box is titled 'Finalizar Prova' and contains the text: 'Você está prestes a entregar a prova. Tem certeza disso?'. At the bottom, there are two buttons: 'Voltar' (grey) and 'Confirmar' (blue).

Fonte: Autoria própria

6 RESULTADOS E DISCUSSÃO

Neste capítulo iremos fazer uma reflexão a respeito do histórico do desempenho dos alunos na disciplina de programação. Após isso, avaliaremos o resultado do questionário apresentado para aqueles que participaram do teste de usabilidade. Por fim, apontaremos, com base no *feedback* de professores, dos próprios alunos por meio do questionário e nossa própria experiência com o PAPO UFF quais poderiam ser as recomendações para trabalhos futuros.

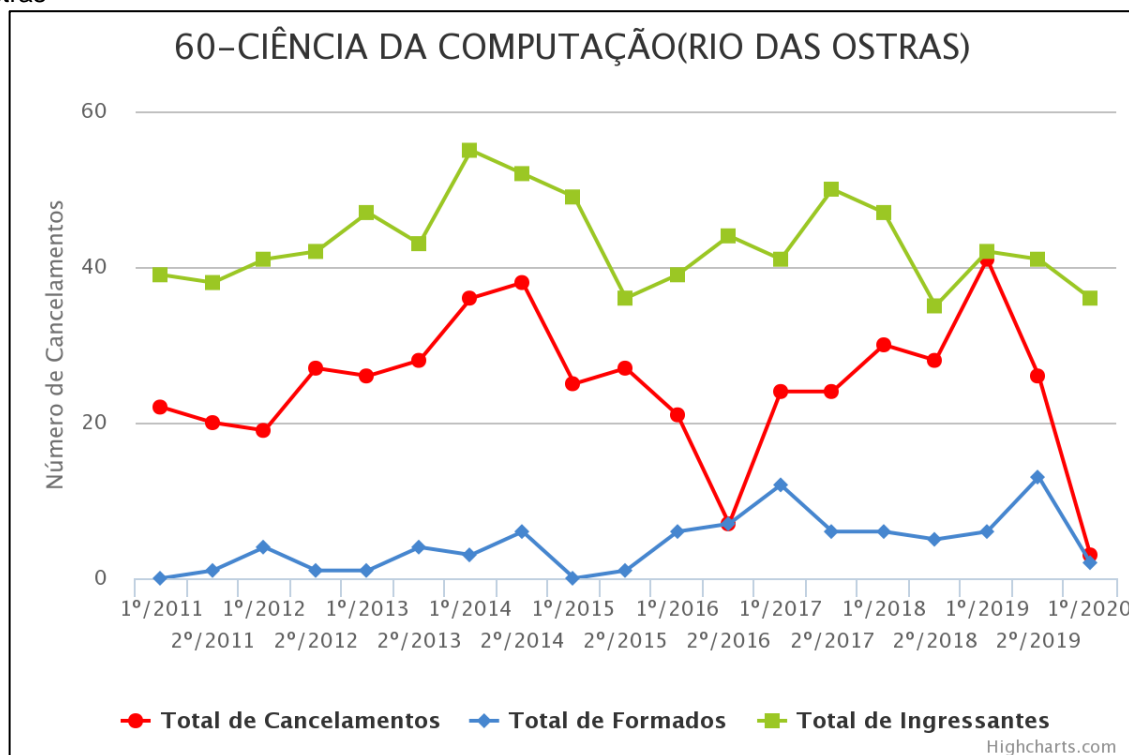
6.1 Análise do Histórico do Curso de Ciência da Computação de Rio das Ostras

Como comentado na introdução deste trabalho, uma de nossas motivações para criação do PAPO UFF seria contribuir de alguma maneira com a qualidade de ensino e prover melhores condições para que as avaliações dos alunos sejam feitas da maneira mais justa e eficiente possível. A visível dificuldade que os alunos das disciplinas de programação encontram ao longo das disciplinas, muitas das vezes, acabam levando a um grave grau de desmotivação ou, nos piores casos, na desistência do curso.

De modo a quantificar este percentual nos últimos períodos, entramos em contato com os professores e a coordenação do curso de Ciência da Computação da Universidade Federal Fluminense de Rio das Ostras para obter os dados referentes ao desempenho dos alunos nos períodos passados. Além disso, visualizamos por meio do portal da transparência que as disciplinas de programação sempre se mantiveram como uma das disciplinas com maior número de reprovações ao longo dos anos. É válido salientar que o fato de uma determinada disciplina apresentar um grande percentual de aprovação por parte dos alunos não significa diretamente que todo o conteúdo foi absorvido com sucesso e que não houve dificuldades no aprendizado. Entretanto, devido aos números alarmantes em relação ao curso de Ciência da Computação, especificamente, em comparação com outros cursos do próprio polo universitário, acreditamos ser válido trazer estas informações para que sirvam como reflexão.

Vejamos o índice comparativo entre os ingressantes, os desistentes e os formados desde 2011:

Gráfico 1 – Portal da Transparência Análise sobre o Curso de Ciência da Computação de Rio das Ostras



Fonte: Portal da Transparência (2021)

Embora o número de ingressantes se mantenha em torno dos 40 alunos, o número de formandos já chegou a atingir a nulidade. Outro aspecto preocupante, seria que a quantidade de cancelamentos esteve por muito tempo em uma média acima dos 20 alunos. Devido ao grande impacto causado pela pandemia e todas as alterações as quais os alunos e a própria instituição foram submetidos, não consideremos nesta avaliação o período a partir de 2020.1.

Concluindo a análise deste retrospecto que tem se mostrado tão negativo, salientamos que a sua causa certamente está relacionada com fatores tão diversos que talvez seja impossível delimitar apenas um motivo como o foco do problema, visto que, muitas das vezes, estes problemas são externos ao ambiente universitário. Entretanto, acreditamos que, por meio de uma avaliação mais aprofundada da situação, muitos destes motivos possam ser identificados e as devidas medidas de controle possam ser tomadas. Com este intuito, uma ferramenta como o PAPO UFF poderia atuar como um potencial identificador destes focos, uma vez que as análises realizadas para cada prova aplicada visa passar ao professor uma melhor descrição do cenário no qual os seus alunos se encontram no contexto do ensino em sala de aula.

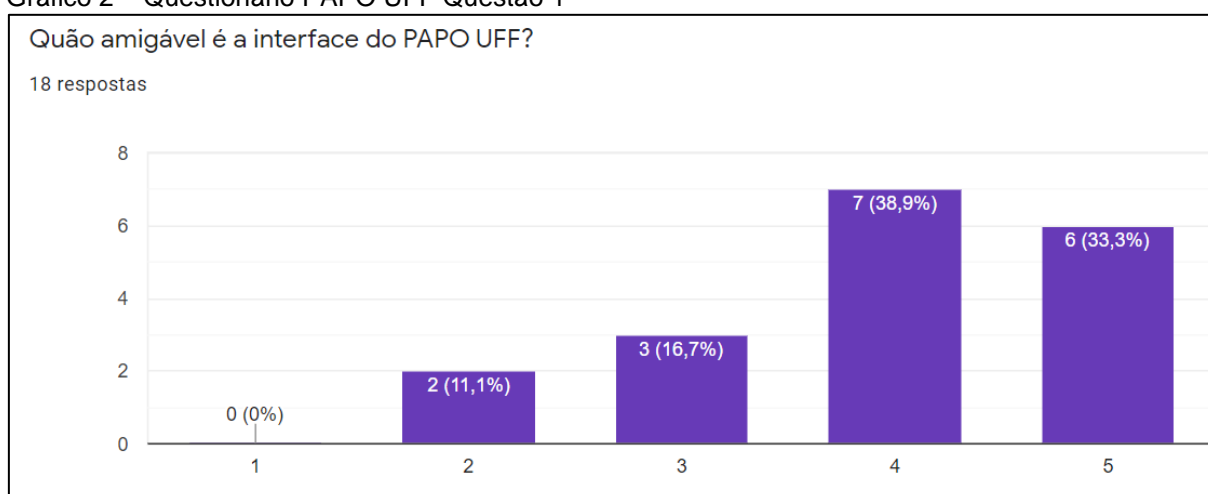
6.2 Análise dos resultados do Teste de Usabilidade

O processo de aplicação dos testes de usabilidade foi muito interessante pois pudemos observar o comportamento dos alunos mediante à aplicação. Infelizmente, o número de participantes foi muito abaixo do esperado. Foram realizadas 27 provas pelos alunos e apenas 18 responderam ao questionário. O acesso à plataforma foi disponibilizado para os alunos de Linguagem de Programação e Programação 1 do curso de Ciência da Computação da Universidade Federal Fluminense de Rio das Ostras.

O questionário foi elaborado de maneira que o aluno pudesse expressar o que de fato tenha sido a sua experiência com o PAPO UFF de maneira rápida e completa. Foram dadas notas para alguns aspectos considerados chave. Vejamos o resultado da avaliação de cada um desses aspectos.

Inicialmente, questionamos o quão amigável seria a interface do PAPO UFF.

Gráfico 2 – Questionário PAPO UFF Questão 1

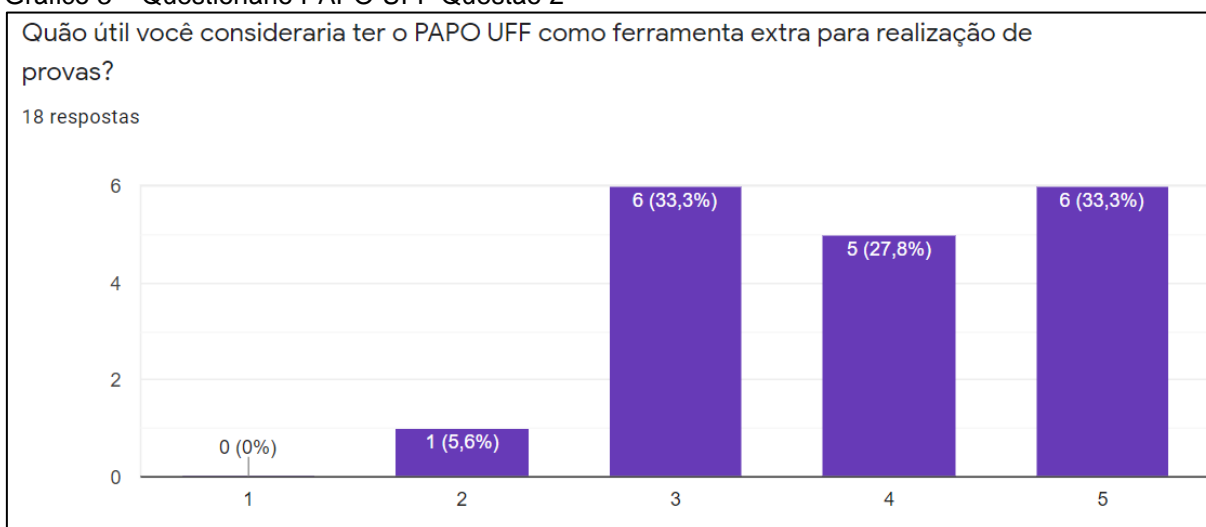


Fonte: Autoria própria

A grande maioria categorizou a interface da aplicação como sendo nota 4 ou 5 em uma escala que considera o mais próximo do 1 como “Nada amigável” e o mais próximo do 5 como sendo “Extremamente amigável”.

A segunda questão seria a respeito do grau de utilidade do PAPO UFF.

Gráfico 3 – Questionário PAPO UFF Questão 2

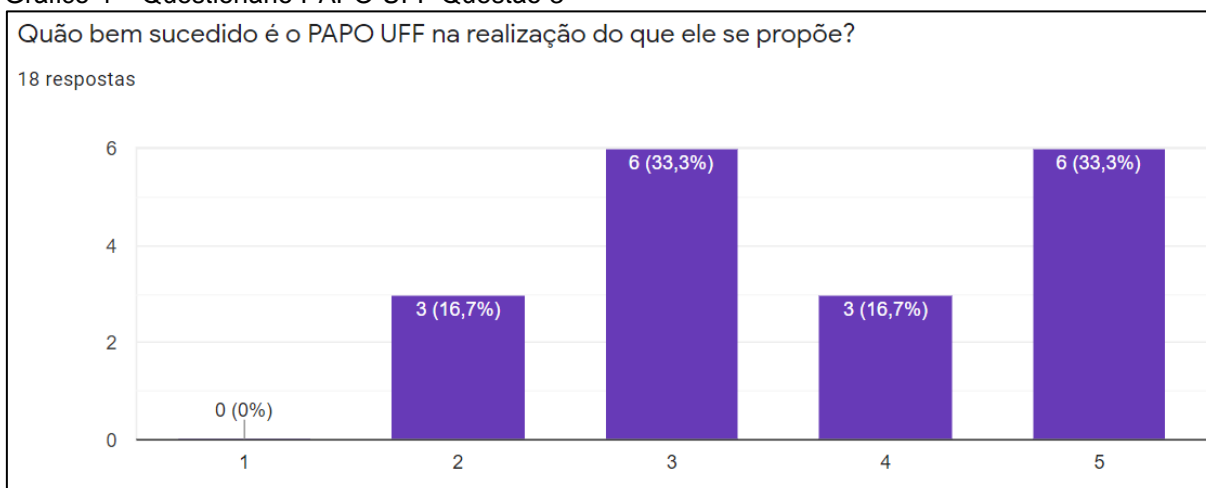


Fonte: Autoria própria

Sendo uma avaliação em relação a visão dos alunos mediante ao que estaria disponível da ferramenta para eles, a maioria ainda ficou dentro de um percentual positivo. Uma grande quantidade de alunos julgou não considerar a ferramenta “Nada útil” e nem “Extremamente útil”, ficando no meio termo.

A terceira questão seria a respeito do desempenho do PAPO UFF em relação ao que é proposto pela plataforma.

Gráfico 4 – Questionário PAPO UFF Questão 3

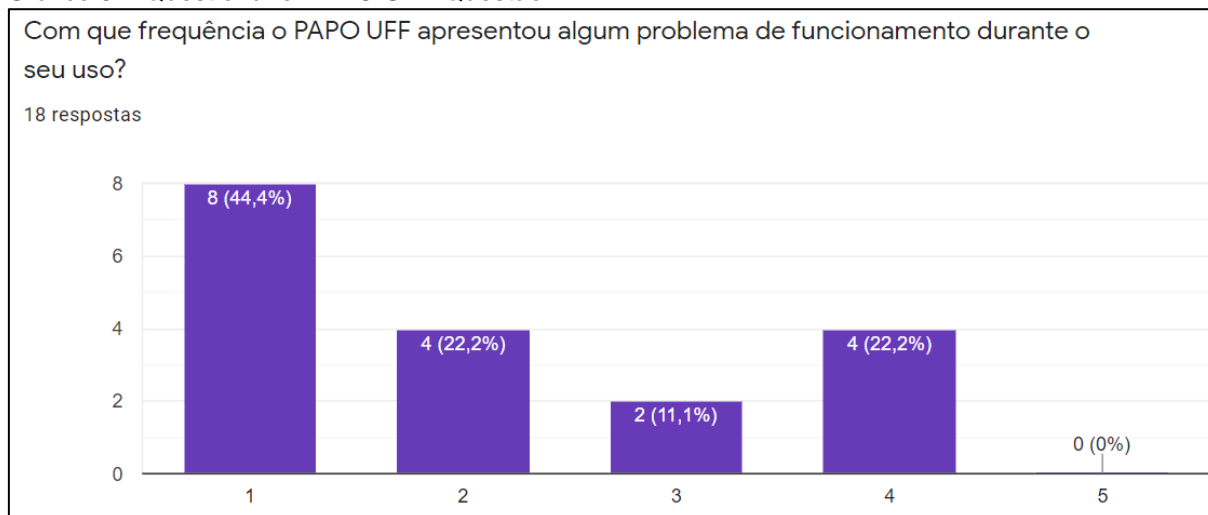


Fonte: Autoria própria

Assim como o aspecto anterior, houve uma certa divergência em relação aos alunos quanto a esta questão. Ficando a maioria dentro do escopo positivo, ainda tivemos um grande número de alunos que acabaram escolhendo a alternativa mediana.

O quarto elemento de análise foi a respeito de eventuais problemas que possam ter ocorrido com os alunos ao responderem uma prova.

Gráfico 5 – Questionário PAPO UFF Questão 4

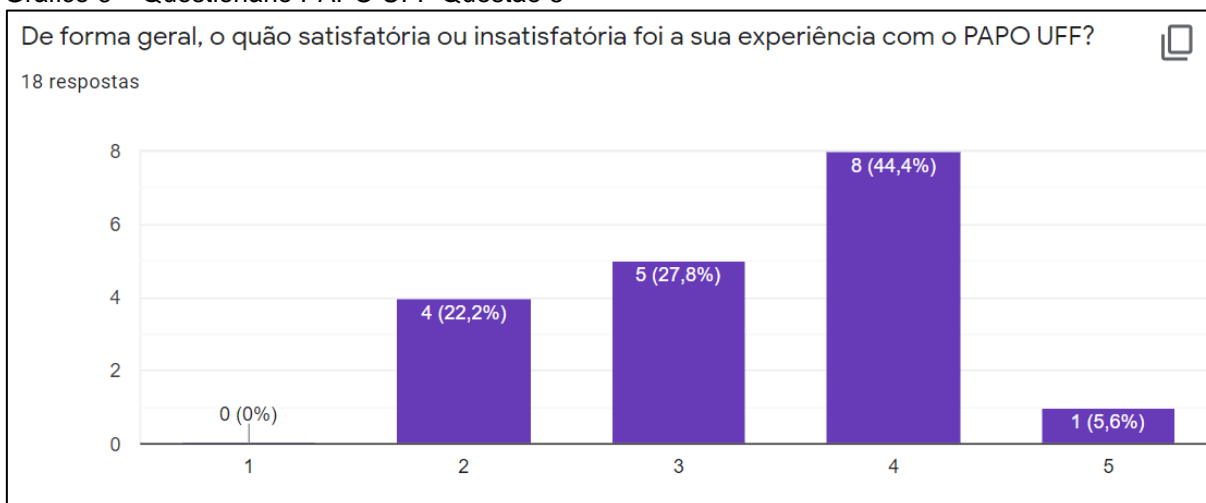


Fonte: Autoria própria

Esta escala considera 1 como sendo “Nada frequente” e 5 como sendo “Extremamente frequente”. Infelizmente, não obtivemos o *feedback* do que teria acontecido com todos aqueles que marcaram as opções referentes a ter passado por algum problema. Entretanto, dentre dos questionários que contavam com esta descrição, foram informados os seguintes problemas: uso da plataforma no celular, problemas com a leitura de uma questão com enunciado muito grande, questionamentos a respeito do processo de *login*, problema com indentação na parte de código das questões de programação ao usar a tecla TAB. Dentre estes problemas relatados, é válido considerar que a questão do *login* e a disponibilidade da plataforma para uso por meio de um celular foram aspectos fora do nosso escopo. Ou seja, o *login* dos alunos foi feito de forma totalmente aberta, isto é, não houve nenhum tipo de barreira justamente para que o acesso fosse feito sem necessidade de cadastros. Em relação aos problemas com celular, realmente, este tipo de acesso não foi considerado no nosso escopo. A recomendação era fazer uso por meio de um computador. Todas estas reclamações e recomendações servem como inspiração para melhorias a serem adotadas em futuras versões do programa, onde os critérios de usabilidade poderiam ser revistos.

Por fim, foi questionado quão satisfatória teria sido a experiência dos alunos com o PAPO UFF.

Gráfico 6 – Questionário PAPO UFF Questão 5



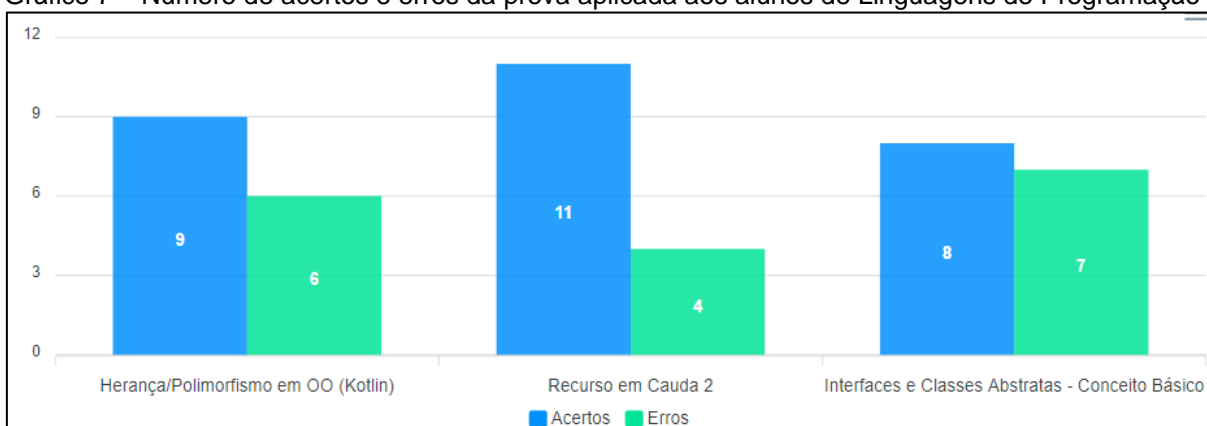
Fonte: Autoria própria

De forma geral, o resultado foi positivo, exceto alguns alunos que apresentaram os problemas relatados anteriormente.

Embora o número de participantes tenha sido bem reduzido, alguns *feedbacks*, foram bastante interessantes e serviram para alguns pontos tratados no tópico que veremos a seguir, como considerações para trabalhos futuros. Portanto, apesar das dificuldades em relação ao número de alunos participantes, esta acabou sendo uma experiência válida e muito positiva para avaliação do PAPO UFF.

Sendo considerada uma atividade complementar para a matéria de Linguagens de Programação, a prova aplicada a estes alunos possibilitou uma observação interessante. Tratando-se de uma atividade com 3 questões, o professor pode constatar que, aquela que se esperava um melhor aproveitamento por parte da turma, foi justamente a que gerou a maior incidência de erros. Com essa informação, foi possível reaver os conceitos envolvidos na questão em aula e reforçar o conteúdo com os alunos. Dentre os alunos participantes do teste em geral, 15 eram da turma de Linguagens de Programação. Vejamos no gráfico 7 como ficou distribuída a relação entre o número de acertos e erros por parte desses alunos.

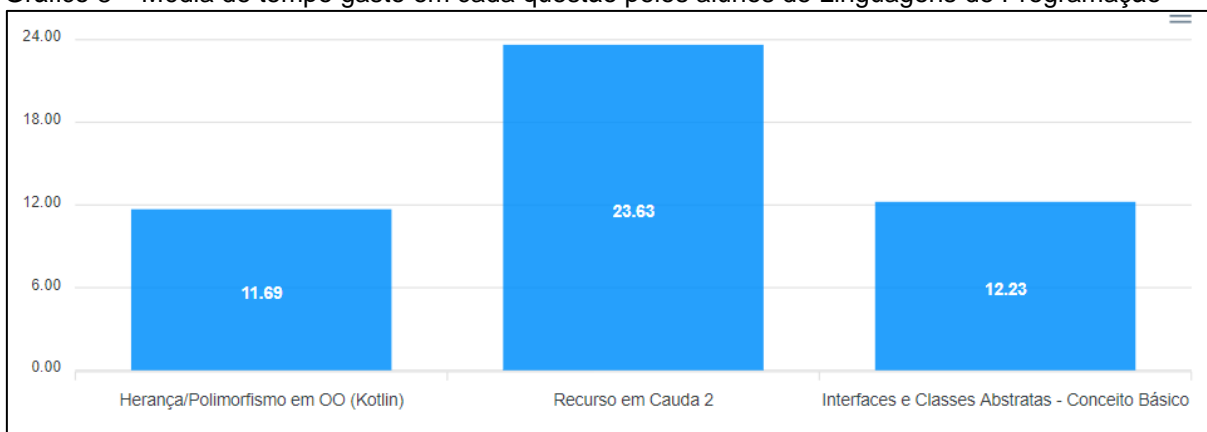
Gráfico 7 – Número de acertos e erros da prova aplicada aos alunos de Linguagens de Programação



Fonte: Autoria própria

As demais informações a respeito do tempo gasto em cada questão e do número de compilações por cada aluno ficaram dentro do esperado por parte do professor. O gráfico 8 demonstra o tempo gasto em cada questão.

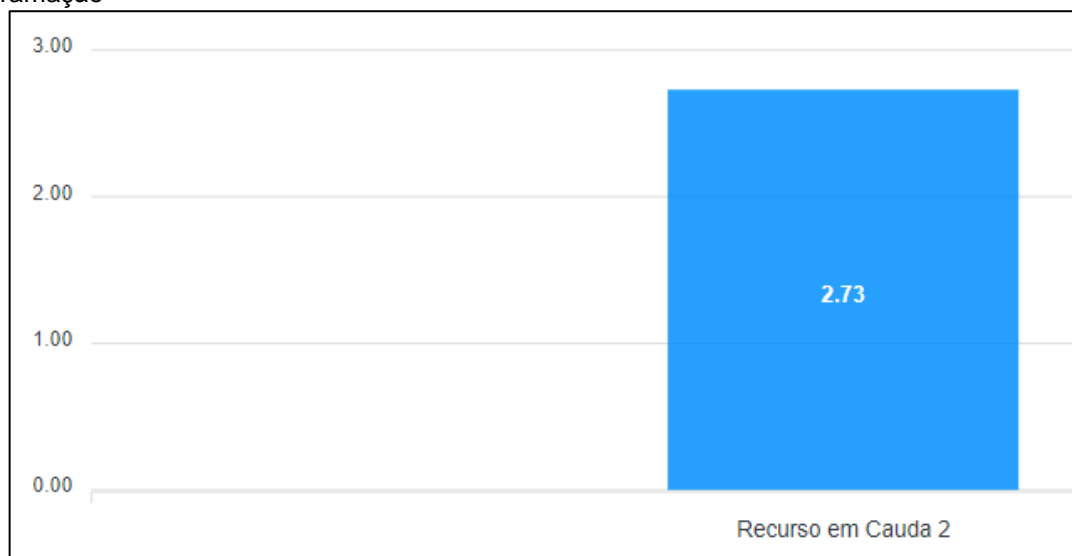
Gráfico 8 – Média de tempo gasto em cada questão pelos alunos de Linguagens de Programação



Fonte: Autoria própria

Como apenas uma das questões era do tipo programação, o gráfico a seguir retrata justamente a sua média de compilações perante a turma.

Gráfico 9 – Média de compilações na questão de programação da prova de Linguagens de Programação



Fonte: Autoria própria

6.3 Considerações para Trabalhos Futuros

Mediante as considerações feitas a partir dos resultados dos testes, preenchimento do questionário pelos alunos participantes, conversas com professores e uma série de autoavaliações, fizemos o levantamento de alguns recursos que seriam interessantes para a plataforma.

Primeiramente, consideramos que deveria ser feita uma remodelagem em alguns aspectos da interface do programa para que pudesse prover uma melhor experiência visando usabilidade e acessibilidade. Como informado a respeito do escopo tratado neste trabalho, não consideramos a adição de recursos de acessibilidade para tornar o uso da plataforma eficiente por usuários com necessidades especiais. Para inserção de uma aplicação deste tipo no cenário real, certamente estas medidas deveriam ser consideradas e diversos outros testes de usabilidade deveriam ser realizados para se chegar ao resultado desejado, no caso, sendo a maior inclusão possível.

Outra questão que gostaríamos de salientar, seria a respeito da possibilidade de integração de uma ferramenta deste tipo com o sistema da faculdade. Dessa forma, poderíamos controlar quem teria acesso à plataforma de maneira mais simples. Além disso, também seria possível integrar as notas obtidas nas avaliações diretamente no

sistema da universidade. Esta integração daria muito mais consistência para a aplicação, tornando-a muito mais viável de entrar na rotina das turmas.

Considerando os dados a respeito do uso de celulares por parte da população, acreditamos que uma adaptação da ferramenta para o cenário *mobile* também seria inevitável para que pudesse alcançar a maior parte do público-alvo possível. Além dos casos onde alunos que não possuem um computador para exercer suas tarefas acadêmicas, os demais também seriam beneficiados com uma maneira mais prática de acessar suas atividades.

A partir da implementação da ferramenta no cenário real das atividades corriqueiras das turmas, alguns ajustes poderiam ser considerados para criação de funcionalidades complementares e no controle de acesso aos dados por parte dos alunos e professores. Podemos citar, por exemplo, a criação de uma página de perfil para os alunos, onde poderiam visualizar seu retrospecto nas atividades realizadas e outras possíveis informações de interesse. Além disso, considerando as atuais atribuições da figura do professor, poderia ser realizada alguma alteração na maneira com que acontece a interação com o banco de questões. Atualmente, somente é permitida a inserção de novas questões por parte dos professores, entretanto, a adição de um recurso de atualização ou exclusão de questões pode ser válida.

Tratando dos aspectos mais minuciosos, diretamente ligados ao PAPO UFF, podemos citar ajustes personalizados em cada campo da interface, como, por exemplo, a possibilidade de alterar o tamanho de cada elemento da página de acordo com a necessidade do usuário. Outro ponto seria a inclusão de mais linguagens consideradas numa questão de programação, que ficou limitada pela linguagem C.

Por último, sugerimos alguns módulos extras que poderiam ser considerados. Uma maneira de incentivar os alunos a realizarem mais atividades a partir de uma lista de questões que estariam disponíveis a todos e que gerariam um ranqueamento na plataforma. Este incentivo poderia ser feito de forma similar ao que se tem disponível no URI, a partir de uma prática chamada gamificação, uma estratégia relativamente recente no cenário educacional. Esta estratégia se baseia na adoção de recursos e técnicas atreladas a jogos em outros contextos, com a finalidade de que o público-alvo tenha maior interesse no assunto.

Um módulo ou recurso antitrapaça, pois, infelizmente, a tarefa de limitar os recursos disponíveis ou evitar que os alunos tenham acesso a conteúdo externo estando online é quase impossível. Acreditamos que a maneira mais efetiva de tratar

o problema seria limitar o acesso à ferramenta na própria faculdade, de forma local (caso que não se aplicaria neste momento de pandemia) ou criar maneiras de rastrear e reportar atividades suspeitas feitas pelos usuários.

Enfim, acreditamos que uma ferramenta como o PAPO UFF pode se expandir de maneira que consiga integrar diversos cenários diferentes. Baseando-se em cada necessidade de um público-alvo, pode existir um tipo de questão diferente a ser considerado em uma prova, além de várias alternativas e recursos que possam ser considerados como interessantes no momento da análise de uma determinada avaliação.

7 CONSIDERAÇÕES FINAIS

Considerando todas as etapas pelas quais este projeto passou, todo o esforço nele aplicado e toda a expectativa por um futuro que possa se beneficiar do assunto nele tratado, consideramos importante fazer alguns comentários e ponderações finais.

Tendo como objetivo enfatizar um possível processo de atualização na maneira com a qual as instituições acadêmicas têm realizado o processo avaliativo de seus alunos, o PAPO UFF visa ser um exemplo de como este processo pode ser construído. A partir do empenho de professores e alunos, acreditamos que a utilização de ferramentas tecnológicas pode contribuir cada vez mais com o nível de ensino e aprendizagem nos mais diversos cenários.

Acreditamos que o PAPO UFF cumpriu seu papel ao mostrar, na prática, exemplos de como poderiam ser extraídas informações relevantes a respeito dos alunos nas provas digitais. Em comparação com o que vemos nas provas em papel, este seria um dos grandes benefícios provido pela tecnologia, além da facilidade para criação e disponibilização das provas. Além desses benefícios para um público geral, a tecnologia pode ter papel determinante na inclusão e melhoria de vida de pessoas que têm necessidades especiais. A acessibilidade seria mais um ponto positivo na comparação com a prova em papel tradicional.

Por outro lado, pudemos ver também que este processo de substituição da prova em papel tende a demorar para acontecer. A infraestrutura tecnológica do país ainda não está preparada para prover tais alterações de forma imediata ou em curto prazo. Porém, acreditamos que esta seja uma evolução natural e que deve ser implementado algo semelhante, inicialmente, em países mais bem desenvolvidos.

Através dos testes práticos, pudemos notar que os próprios alunos tendem a aceitar a ideia de que uma ferramenta digital possa contribuir de maneira significativa para o desempenho acadêmico e que a experiência geral no uso desse tipo de ferramenta é satisfatória. Afinal, professores e alunos teriam benefícios múltiplos.

Acreditamos que contribuir com uma melhor qualidade da educação no país é contribuir com a qualidade de vida de toda uma nação.

REFERÊNCIAS

CODERBYTE. *Coderbyte for Developers*, 2020. Página de Resolução de Exercício. Disponível em: <<https://www.coderbyte.com/>>. Acesso em: 7 de dez. de 2020.

DA SILVA, Leonardo Ivo Neves. 2013. CADASTRAMENTO DE QUESTÕES E APLICAÇÃO DE PROVAS ONLINE. Disponível em: <<http://www.uezo.rj.gov.br/tcc/tads/Leonardo-Ivo-Neves-da-Silva.pdf>>. Acesso em 4 abr. 2021.

DE CAMPOS, Cassio P.; FERREIRA, Carlos E. BOCA: *A Support System for Programming Contests*. In: *Workshop de Educação em Computação (Brazilian Workshop on Education in Computing)*, 2004, Salvador. Anais do Congresso da SBC. 2004. Disponível em: <[BOCA: um sistema de apoio a competições de programação](#)>. Acesso em: 2 de abr. de 2021.

DIAGRAMS.NET. *Security-first diagramming for teams*. Disponível em: <<https://app.diagrams.net/>>. Acesso em 6 abr. 2021.

FENOPIX. CanvasJS, 2021. Disponível em: <<https://canvasjs.com/react-charts/>>. Acesso em 25 mar. 2021.

GNU FREE DOCUMENTATION LICENSE. BOCA, 2020. manualBOCA. Disponível em: <<http://maratona.ime.usp.br/manualBOCA.html/>>. Acesso em: 7 de dez. de 2020.

GOOGLE. *Google Classroom*, 2020. Sala de Aula. Disponível em: <<https://classroom.google.com/>>. Acesso em: 7 de dez. de 2020.

URI ONLINE JUDGE FORUM. URI, 2020. Página inicial. Disponível em: <<https://www.urionlinejudge.com.br/>>. Acesso em: 7 de dez. de 2020.

MATTOS, Thácyra A. F.; DA COSTA, José Wilson. 2013. Avaliação da aprendizagem: do papel para o computador. Disponível em: <<https://br-ie.org/pub/index.php/wcbie/article/view/2687/2341>>. Acesso em 4 abr. 2021.

MDN Web Docs. 2021. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy>. Acesso em 28 mar. 2021.

MILLER, Mikki, 2019. EmergingEdtech. Disponível em: <<https://www.emergingedtech.com/2019/03/7-advantages-digital-assessments-have-over-paper-tests-exams/>>. Acesso em 4 abr. 2021.

NGROK. Public URLs, 2021 Disponível em: <<https://ngrok.com/>>. Acesso em 4 abr. 2021.

NODEJS. *Introduction to Node.js*, 2021. Disponível em: <<https://nodejs.org/en/docs/guides/>>. Acesso em 5 abr. 2021.

ORACLE. MySQL, 2021. Disponível em: <<https://www.mysql.com/>>. Acesso em 5 abr. 2021.

PHPMYADMIN. *Bringing MySQL to the web*. Disponível em: <<https://www.phpmyadmin.net/>>. Acesso em 5 abr. 2021.

REACT BOOTSTRAP. Disponível em: <<https://react-bootstrap.github.io/>>. Acesso em 4 abr. 2021.

REACT CODE MIRROR. Disponível em: <<https://uiwjs.github.io/react-codemirror/>>. Acesso em 4 abr. 2021.

REACT COMPOUND TIMER. Disponível em: <<https://www.npmjs.com/package/react-compound-timer>>. Acesso em 5 abr. 2021.

REACT TRAINING. React Router, 2021. Disponível em: <<https://reactrouter.com/>>. Acesso em 04 abr. 2021.

REACT. Tutorial: *Intro to React*, 2021. Disponível em: <<https://reactjs.org/tutorial/tutorial.html#overview>>. Acesso em 5 abr. 2021.

SOUTO, Mario. 2019. Alura. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>>. Acesso em 4 abr. 2021.

SQLDBM, 2021. Disponível em: <<https://sqldbm.com/Home/>>. Acesso em 28 mar. 2021.

STACK OVERFLOW, 2021. 2020 *Developer Survey*. Disponível em: <<https://insights.stackoverflow.com/survey/2020#most-popular-technologies>>. Acesso em 3 mai. 2021.

STRONGLOOP. Express, 2021. Disponível em: <<https://expressjs.com/pt-br/>>. Acesso em 28 mar. 2021.

TIC DOMICÍLIOS, 2019. Pesquisa Sobre o Uso das Tecnologias de Informação e Comunicação nos Domicílios Brasileiros. Disponível em: <https://www.cetic.br/media/docs/publicacoes/2/20201123121817/tic_dom_2019_livro_eletronico.pdf>. Acesso em: 4 abr. 2021.

UFF, 2021. Sistema de Transparência da UFF. Disponível em: <https://app.uff.br/transparencia/ingressantes_formados_cancelados>. Acesso em 6 abr. 2021.

VILLADORA. Express Http proxy. Disponível em: <<https://github.com/villadora/express-http-proxy#readme>>. Acesso em 03 mar. 2021.

VINHAES, Marina de Lima. 2013. Sistema de Provas Online com desenvolvimento em Python com Django. Disponível em: <<https://www.uniriotec.br/~bsi/tcc/textos/2013Vinhaes.pdf>>. Acesso em 5 abr. 2021.

ZANETTE, Alysson. 2017. BeCode. Disponível em: <<https://becode.com.br/framework-biblioteca-api-entenda-as-diferencas/>>. Acesso em 5 abr. 2021.

APÊNDICE A – Tecnologias Utilizadas no Protótipo

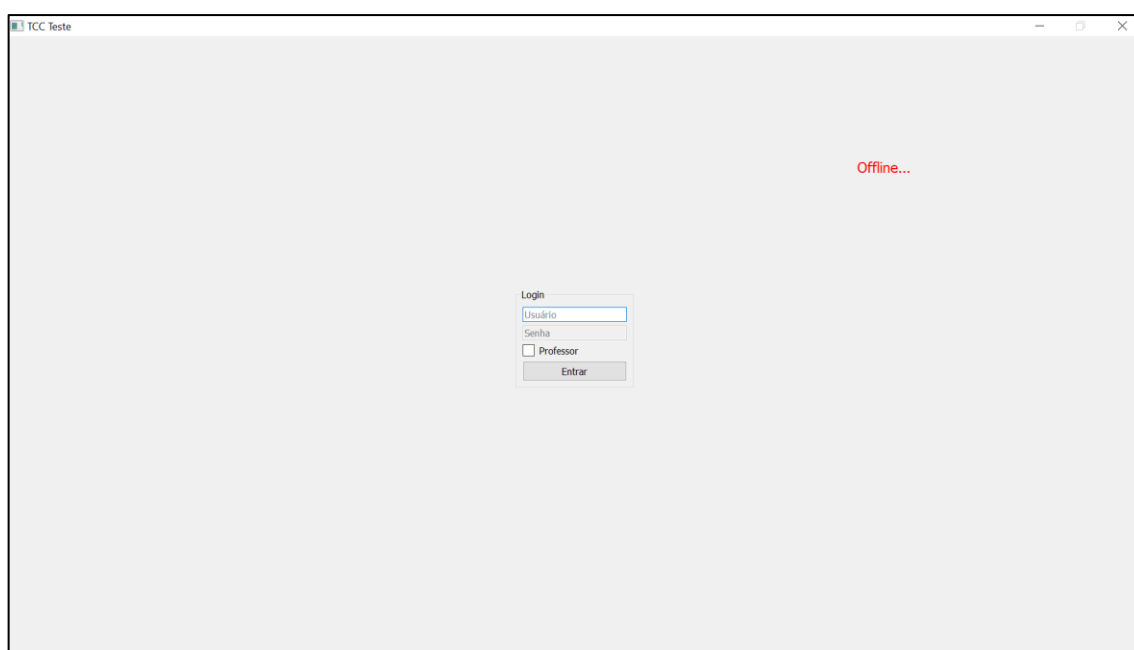
Citaremos de maneira breve e sucinta as tecnologias e ferramentas utilizadas neste protótipo com o intuito de passar uma melhor noção comparativa com o resultado obtido no trabalho final, onde poderemos notar as diversas formas que consideramos plausíveis para desempenhar as atribuições que imaginávamos cabíveis para o programa e quais as vantagens de cada uma dessas propostas.

Todo o programa foi desenvolvido através do Qt Creator, que se trata de uma IDE multiplataforma com diversos recursos. Por meio dela, conseguimos uma boa integração entre a interface com a qual estávamos trabalhando e o código fonte que geria cada funcionalidade que, por sua vez, foi todo desenvolvido utilizando a linguagem de programação C++. O banco de dados utilizado foi o MySQL, configurado por meio do aplicativo phpMyAdmin e suporte do aplicativo XAMPP. A comunicação entre os alunos e o professor foi feita por meio da implementação de um sistema cliente servidor onde o *login* de um aluno geraria um *Socket* que se conectaria ao *Server* iniciado pelo professor no momento da aplicação da prova. O professor poderia gerar provas que seriam salvas em formato XML (do inglês eXpensible Markup Language, ou seja, Linguagem Extensível de Marcação) para que fossem aplicadas futuramente.

APÊNDICE B – Interfaces do Protótipo

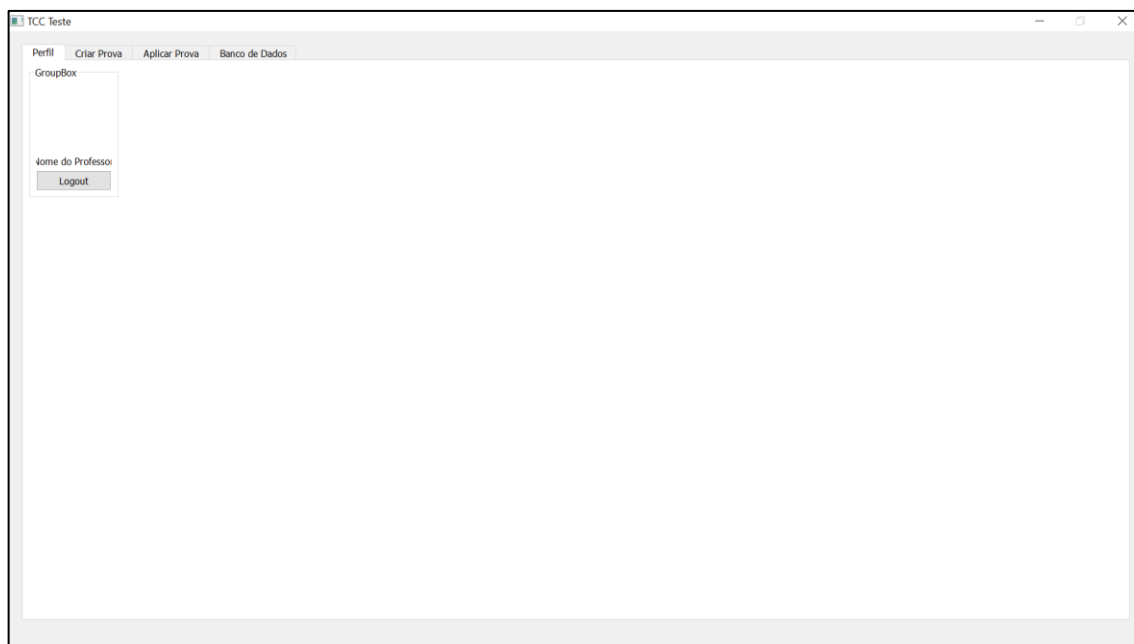
Vejamos a seguir as interfaces funcionais do protótipo apresentado. Podemos notar que, apesar de não estarem prontas do ponto de vista estético, no aspecto funcional, as operações idealizadas já estavam bem definidas.

Tela inicial do programa, onde seria feita a distinção do perfil de usuário entre aluno e professor. Esta tela também informa o *status* da conexão com o Banco de Dados, que neste exemplo, mostra-se *Offline*.



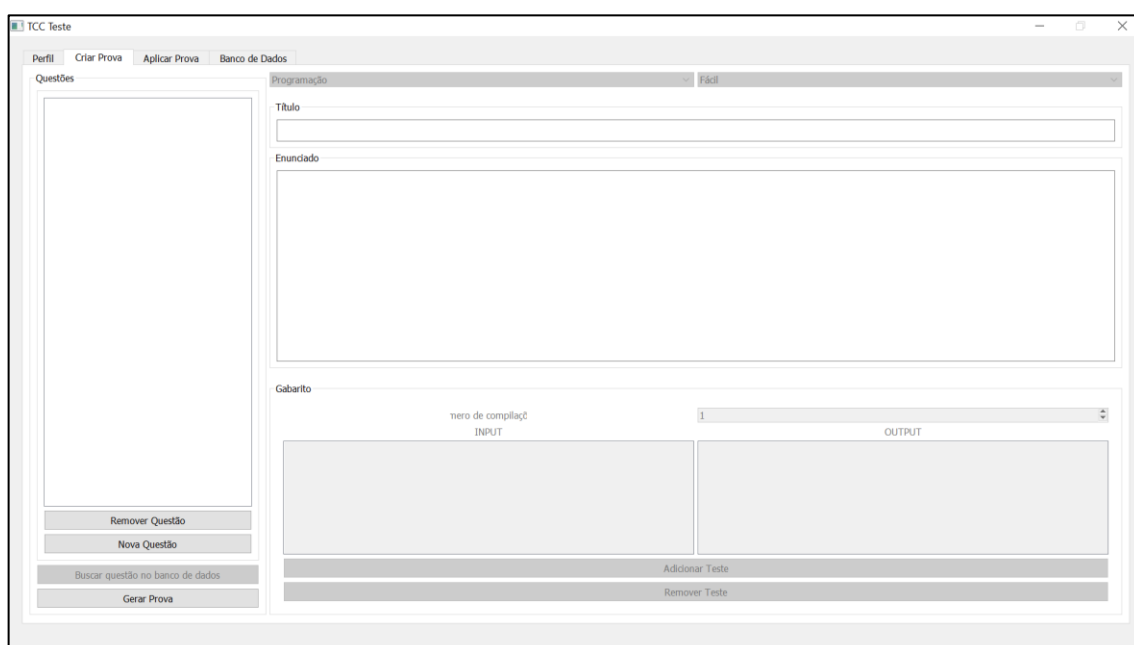
Fonte: Autoria própria

Aba perfil do usuário professor. Como ainda não havíamos definido exatamente o que seria relevante para esta parte, definimos apenas um campo para identificação com base no nome e uma imagem.



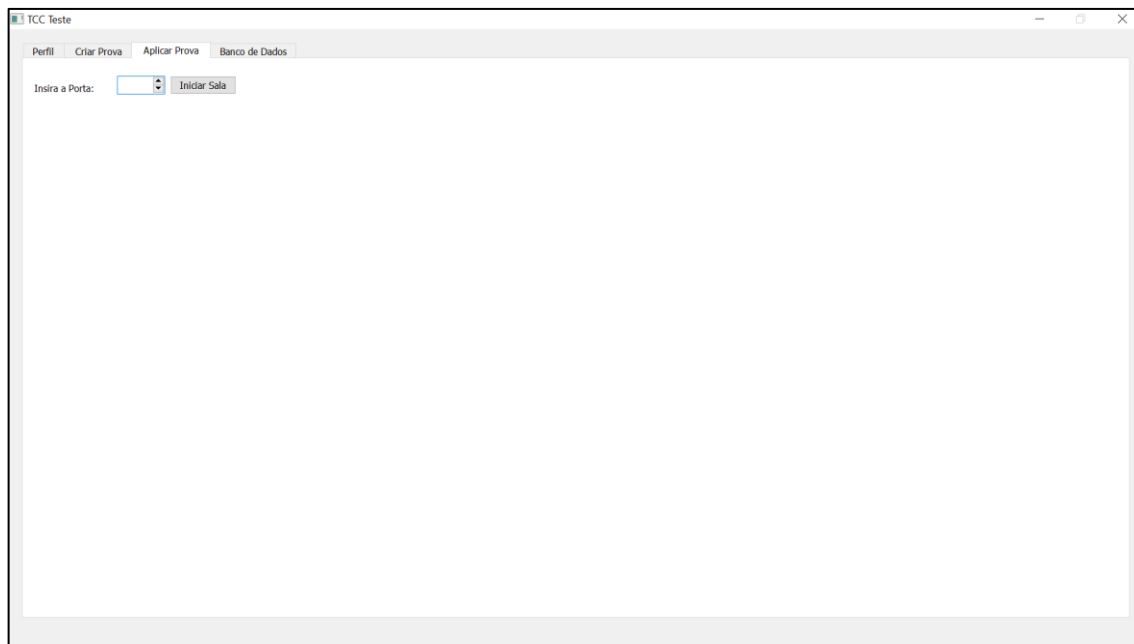
Fonte: Autoria própria

Aba para criação de uma prova. Aqui o professor poderia definir as questões de uma dada prova que seria aplicada futuramente, gerando novas questões ou buscando-as do banco de dados.



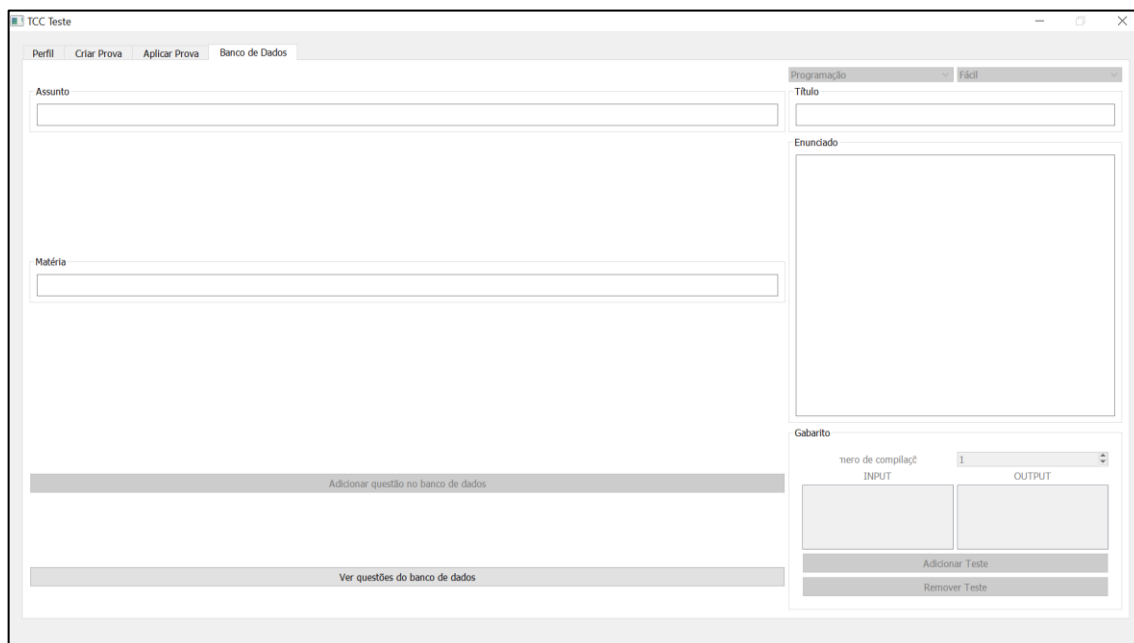
Fonte: Autoria própria

Aba para aplicação de prova, onde o professor iria criar o código da sala com a qual os alunos iriam se conectar.



Fonte: Autoria própria

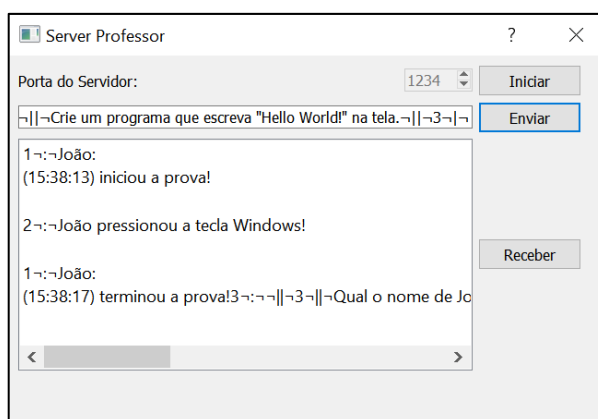
Aba do banco de dados, onde o professor poderia visualizar as questões já criadas (inclusive por outros professores) ou adicionar uma questão nova para que fosse salva nesse banco de questões compartilhado.



Fonte: Autoria própria

Janela que indicaria o *status* do *Server* gerado pelo professor ao iniciar uma prova. Na versão desenvolvida até então, esta janela funcionava somente para auxiliar

na visualização do fluxo de troca de informações entre alunos e professores, tais quais como o sucesso no envio da prova, quando cada aluno teria iniciado e terminado sua atividade, além de algumas teclas e *hotkeys* (combinação de teclas com o intuito de realizarem uma determinada ação) que seriam monitoradas durante todo o período de aplicação da prova. Podemos notar alguns caracteres especiais nas mensagens que chegam ao servidor, esta foi uma forma com a qual pretendíamos elaborar uma codificação que seria própria da aplicação, porém, até então, tratava-se apenas de uma fase de testes.



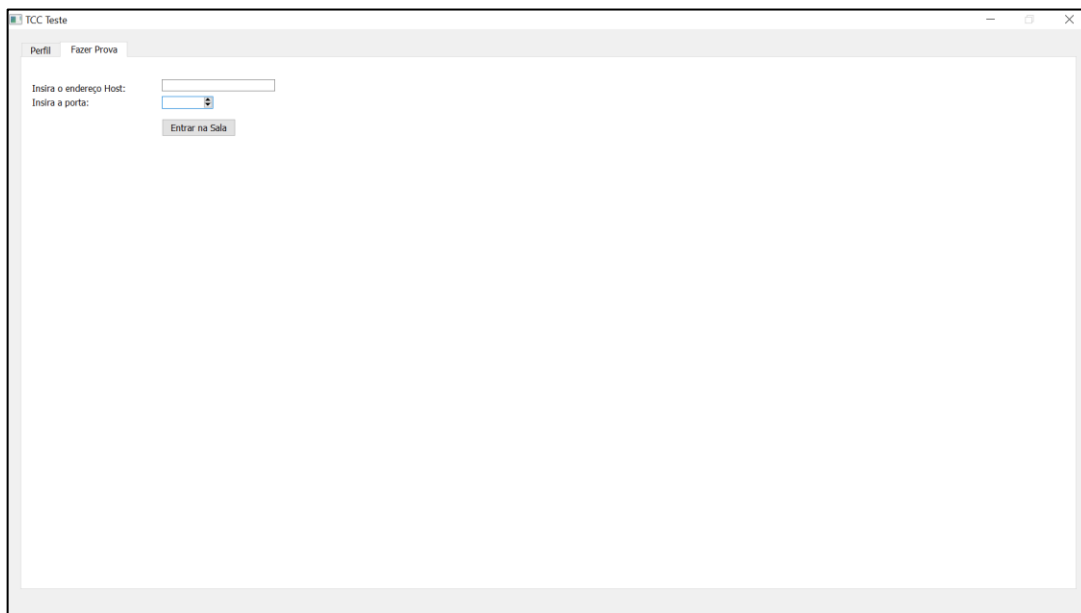
Fonte: Autoria própria

Aba perfil do usuário Aluno. Até então, incluía somente a opção de voltar à página inicial do programa.



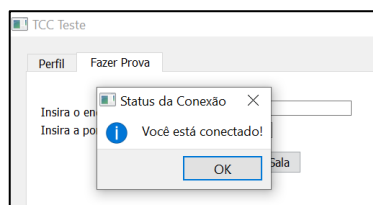
Fonte: Autoria própria

Aba na qual um aluno deveria informar as informações do servidor criado pelo professor para que pudesse realizar uma prova.



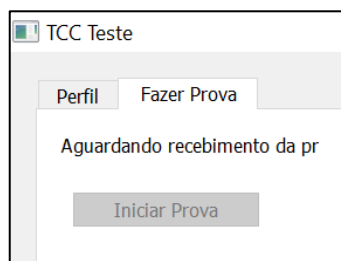
Fonte: Autoria própria

Ao informar as informações sobre a prova, uma mensagem alertaria a respeito do *status* da conexão com o professor, neste caso, vemos o cenário onde a conexão foi feita com sucesso.



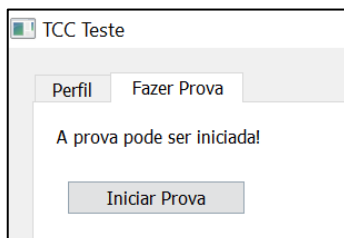
Fonte: Autoria própria

Após conectar-se ao servidor do professor, o aluno aguardaria até que fosse liberado o início da prova.



Fonte: Autoria própria

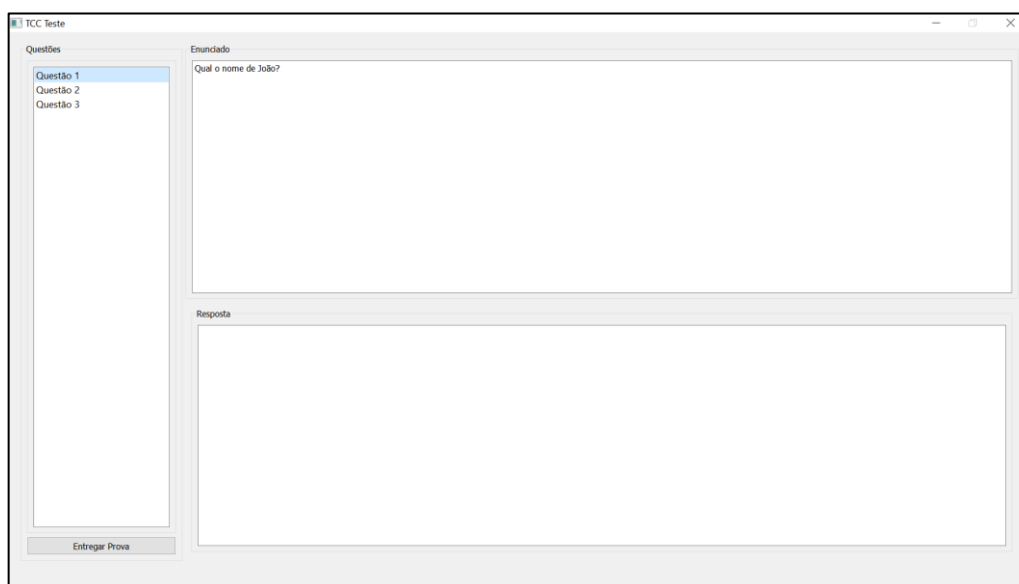
Dado o aval por parte do professor, os alunos poderiam selecionar o botão que daria início à prova.



Fonte: Autoria própria

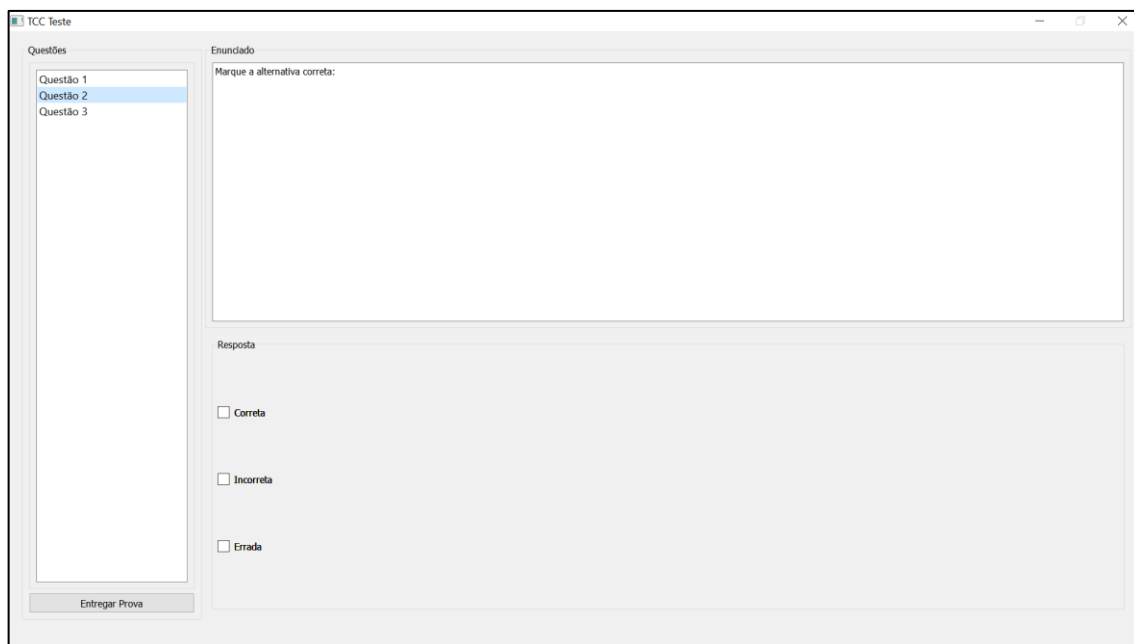
A interface para realização de prova consta com uma lista de questões no lado esquerdo da tela e o escopo da questão selecionada no lado direito. Veremos os diferentes tipos de questões com as quais os alunos poderiam se deparar em uma prova.

Interface de uma questão discursiva.



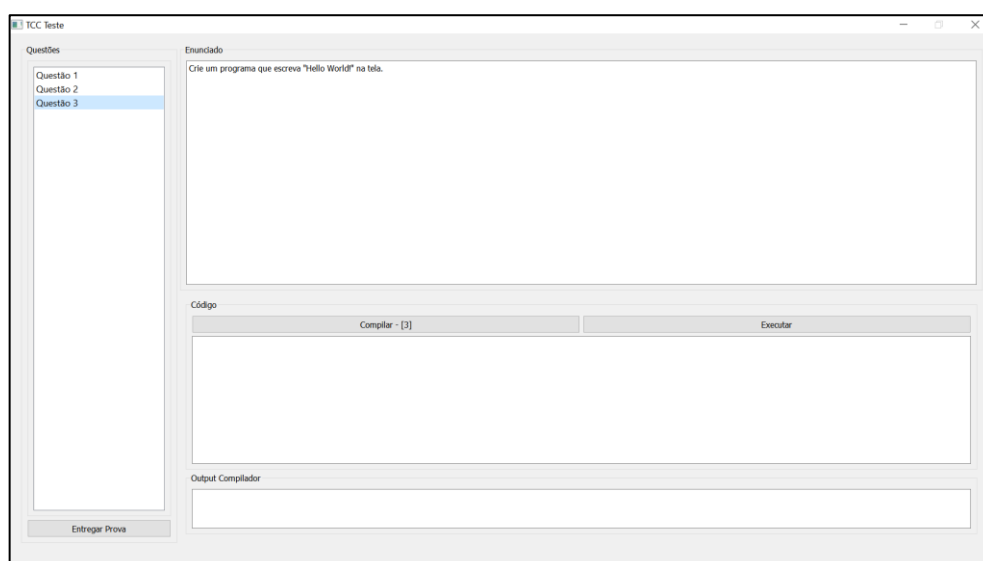
Fonte: Autoria própria

Interface de uma questão de múltipla escolha.



Fonte: Autoria própria

Interface de uma questão de programação. Este tipo de questão, contaria com o recurso de um compilador onde os alunos teriam a possibilidade de testar seus códigos antes de confirmar sua resposta e obter um retorno do compilador em relação ao processo de compilação. O número de compilações seria definido pelo professor, porém, o aluno poderia executar o programa compilado quantas vezes julgasse necessário.



Fonte: Autoria própria

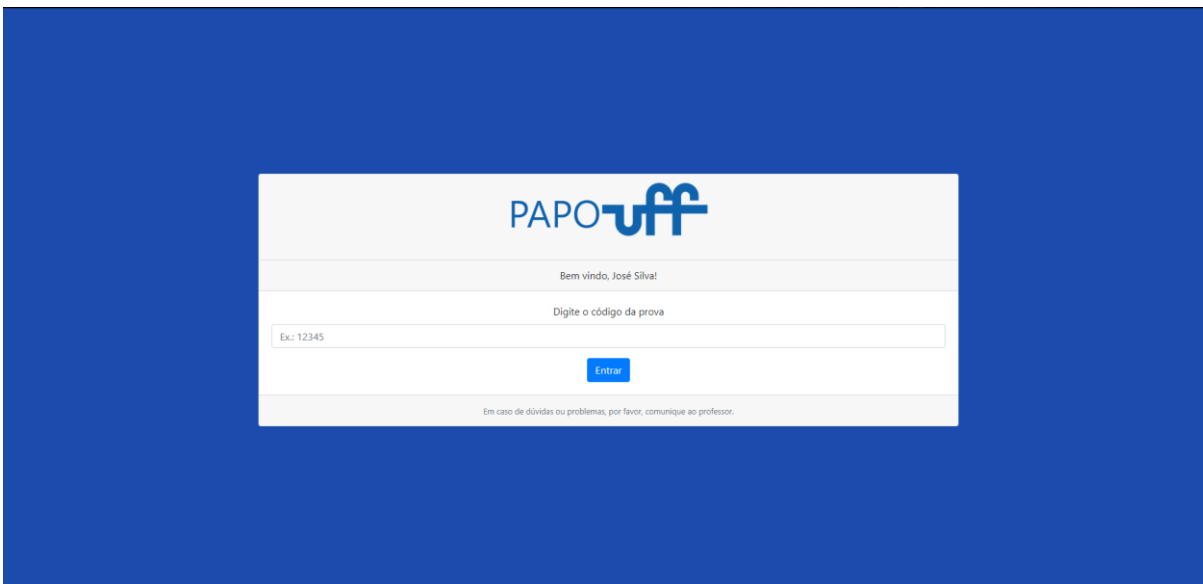
APÊNDICE C – Interfaces do PAPO UFF em Escala Real

Página inicial para realização de *login*.

The image shows the login page of the PAPO UFF platform. The background is a solid blue color. On the left side, the text "PAPO UFF" is displayed in white, with "Plataforma de Aplicação de Provas Online" underneath it. On the right side, there is a white rectangular box containing the "uff" logo at the top. Below the logo are two input fields: "Digite seu usuário" and "Digite sua senha". A blue button labeled "Entrar" is positioned below the second input field.

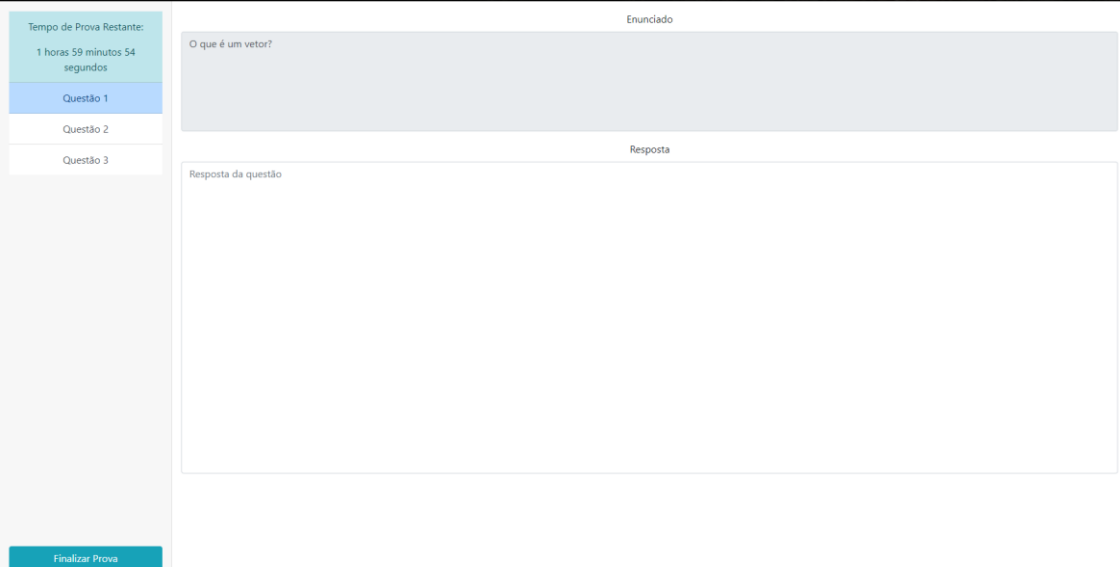
Fonte: Autoria própria

Página seguinte ao *login* realizado pelo aluno. Onde deverá ser inserido o código de uma prova.

The image shows the page after a successful login. The background is blue. A white rectangular box is centered on the page. At the top of the box is the "PAPO-uff" logo. Below the logo, the text "Bem vindo, José Silva!" is displayed. Underneath that is the instruction "Digite o código da prova". A text input field contains the example code "Ex.: 12345". A blue "Entrar" button is located below the input field. At the bottom of the white box, there is a small line of text: "Em caso de dúvidas ou problemas, por favor, comunique ao professor."

Fonte: Autoria própria

Interface de uma questão do tipo discursiva.



Tempo de Prova Restante:
1 horas 59 minutos 54 segundos

Questão 1
Questão 2
Questão 3

Finalizar Prova

Enunciado

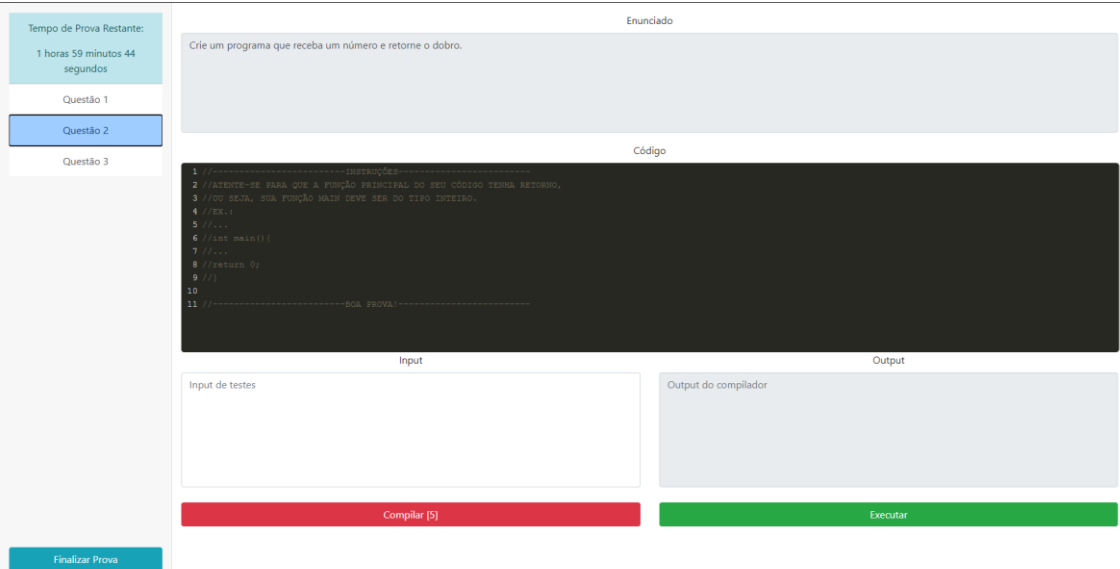
O que é um vetor?

Resposta

Resposta da questão

Fonte: Autoria própria

Interface de uma questão do tipo programação.



Tempo de Prova Restante:
1 horas 59 minutos 44 segundos

Questão 1
Questão 2
Questão 3

Finalizar Prova

Enunciado

Crie um programa que receba um número e retorne o dobro.

Código

```
1 //-----SUA RESPOSTA-----
2 //ATENÇÃO: PARA QUE A FUNÇÃO PRINCIPAL DO SEU CÓDIGO TENHA RETORNO,
3 //OU SEJA, SUA FUNÇÃO MAIN DEVE SER DO TIPO INTEIRO.
4 //EX:
5 //...
6 //int main()
7 //{
8 //return 0;
9 //}
10
11 //-----SUA PROVA-----
```

Input

Input de testes

Output

Output do compilador

Compilar [5]

Executar

Fonte: Autoria própria

Interface de uma questão do tipo múltipla escolha.

The screenshot shows a test interface. On the left, there is a sidebar with a timer indicating 'Tempo de Prova Restante: 1 horas 59 minutos 26 segundos'. Below the timer are three question tabs: 'Questão 1', 'Questão 2', and 'Questão 3', with 'Questão 3' selected. At the bottom of the sidebar is a 'Finalizar Prova' button. The main area is titled 'Enunciado' and contains the question 'Qual vetor está correto?'. Below the question are three radio button options: int a = 1,2,3; int a[] = 1,2,3; and int a[3] = 1,2,3;

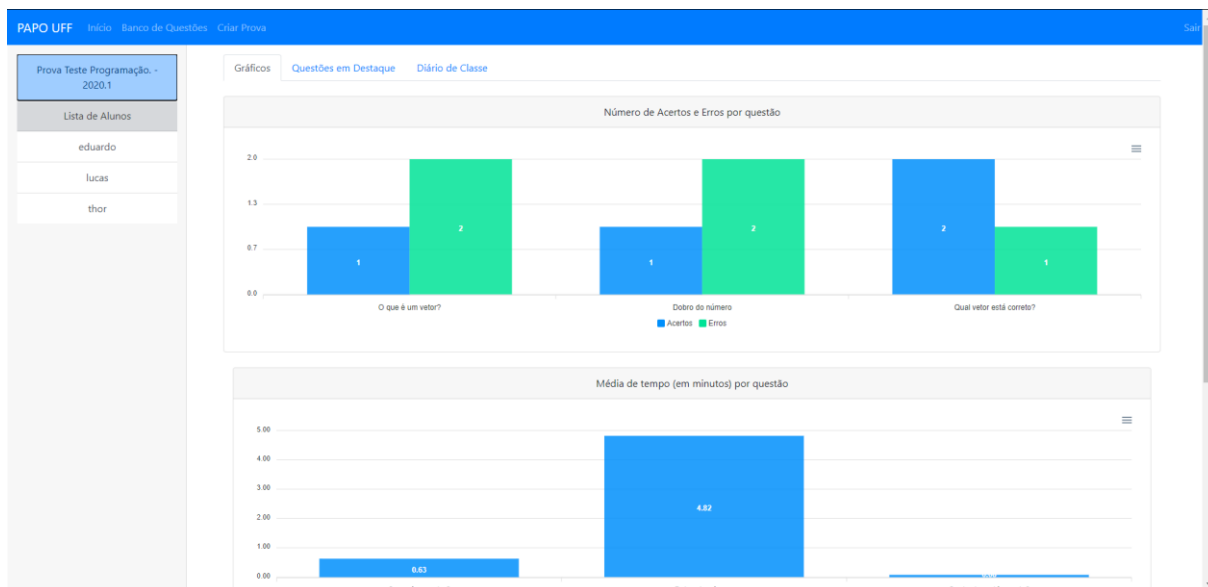
Fonte: Autoria própria

Tela seguinte ao *login* do professor. Página inicial, com perfil e informações sobre provas aplicadas.

The screenshot shows a professor's dashboard. At the top, there is a blue navigation bar with 'PAPO UFF' and links for 'Início', 'Banco de Questões', and 'Criar Prova'. On the left, there is a profile section with a user icon and the label 'Professor'. The main content area displays two test result cards. The first card is titled 'P1 Exemplo 2020.2' and shows 'Código da prova: 22' with a 'Correção' button. The second card is titled 'P2 Exemplo - 2020.2' and shows 'Código da prova: 26' with a 'Correção' button.

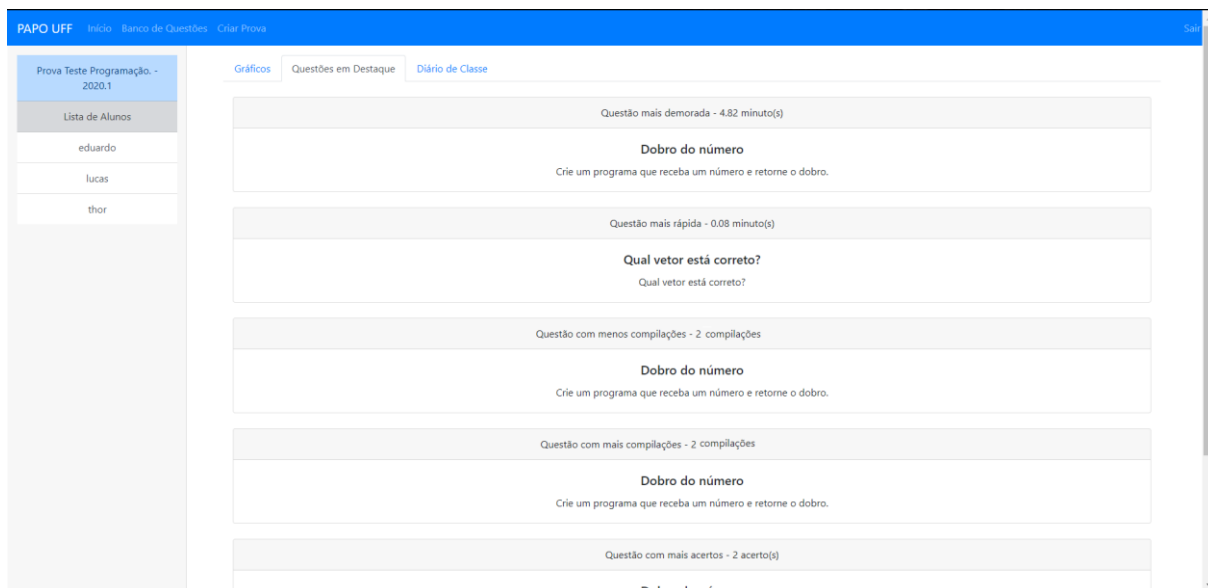
Fonte: Autoria própria

Página de correção de uma prova. Onde é possível analisar os gráficos e resultados a partir do desempenho de cada aluno.



Fonte: Autoria própria

Aba de questões em destaque.



Fonte: Autoria própria

Aba diário de classe. Onde constam os nomes e notas dos alunos participantes de uma determinada prova.

PAPO UFF Início Banco de Questões Criar Prova Sair

Prova Teste Programação. - 2020.1

Gráficos Questões em Destaque Diário de Classe

Lista de Alunos

Nome do Aluno	Nota
eduardo	20
lucas	20
thor	6

Fonte: Autoria própria

Revisão de uma questão respondida por um aluno em uma determinada prova.

PAPO UFF Início Banco de Questões Criar Prova Sair

Prova Teste Programação. - 2020.1

Lista de Alunos

eduardo

lucas

thor

O que é um vetor?

Dobro do número

Enunciado

Crie um programa que receba um número e retorne o dobro.

Feedback

Questão Correta.

Resposta Aluno

```
#include <stdio.h>

int main()
{
    int a;
    scanf("%d", &a);
    printf("%d", 2*a);
    return 0;
}
```

Gabarito

< Input: 1 Output: 2 >,< Input: 2 Output: 4 >,< Input: 3 Output: 6 >

Nota desta Questão: Salvar Nota

Qual vetor está correto?

Fonte: Autoria própria

Tela de banco de questões.

The screenshot shows the PAPO UFF interface. At the top, there are navigation links: "Início", "Banco de Questões", and "Criar Prova". The main area is divided into a left sidebar and a main content area. The sidebar has a "Programação 1" dropdown and a "Vetores" dropdown. Below these are several example question titles: "Exemplo questão discursiva vetor", "Exemplo questão discursiva pilha", "testando inserção de questão", "O que é um vetor?", "Dobro do número", "Qual vetor está correto?", "Hello World", "Fatorial Prog 1", and "Fibonacci Prog 1". At the bottom of the sidebar is a green button labeled "Adicionar Questão Nova". The main content area has four dropdown filters: "Matéria" (Programação 1), "Assunto" (Vetores), "Tipo" (Discursiva), and "Dificuldade" (Fácil). Below these are two text input fields: "Título da questão" and "Enunciado da questão".

Fonte: Autoria própria

Opção de adicionar nova questão na tela do banco de questões.

This screenshot shows the same PAPO UFF interface as above, but with a modal window titled "Adicionar Questão no Banco" open. The modal contains the same filter dropdowns (Matéria: Programação 1, Assunto: Vetores, Tipo: Discursiva, Dificuldade: Fácil) and text input fields for "Título da questão" and "Enunciado da questão". At the bottom of the modal, there is a blue button labeled "Editar Resposta" and a green button labeled "Adicionar no Banco". The background interface is dimmed.

Fonte: Autoria própria

Tela de criação de uma prova.

PAPO UFF [Início](#) [Banco de Questões](#) [Criar Prova](#) [Sair](#)

Nome da Prova
Ex.: P1 Programação 1 - 2021,1

Nova Questão - 10 pontos

Nova Questão - 10 pontos

Matéria Assunto Tipo Dificuldade

Título

Enunciado

[Editar Resposta](#) [Apagar](#)

[Nova Questão](#)
[Banco de Questões](#)
[Finalizar Prova](#)

Fonte: Autoria própria

Com que frequência o PAPO UFF apresentou algum problema de funcionamento durante o seu uso? *

	1	2	3	4	5	
Nada Frequente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremamente Frequente

⋮

De forma geral, o quão satisfatória ou insatisfatória foi a sua experiência com o PAPO UFF? *

	1	2	3	4	5	
Extremamente insatisfatória	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremamente satisfatória

Quais as suas recomendações ou críticas a respeito do PAPO UFF?

Texto de resposta longa
