PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

DESENVOLVIMENTO DE UM APLICATIVO PARA AUXILIAR O ESTUDO DE AUTÔMATOS FINITOS DETERMINÍSTICOS

DANIEL FIGUEIREDO PEREIRA

GOIÂNIA
2022

DANIEL FIGUEIREDO PEREIRA

DESENVOLVIMENTO DE UM APLICATIVO PARA AUXILIAR O ESTUDO DE AUTÔMATOS FINITOS DETERMINÍSTICOS

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel Engenharia de Computação.

Orientador(a):
Profa. Ma. Ludmilla Reis Pinheiro dos Santos
Banca examinadora:
Profa. Dra. Carmen Cecília Centeno
Prof. Dr. José Luiz de Freitas Júnior

GOIÂNIA
2022

DANIEL FIGUEIREDO PEREIRA

DESENVOLVIMENTO DE UM APLICATIVO PARA AUXILIAR O ESTUDO DE AUTÔMATOS FINITOS DETERMINÍSTICOS

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Engenharia de Computação, em _____/_____/_____.

_____

Orientadora: Profa. Ma. Ludmilla Reis Pinheiro dos Santos

_____

Profa. Ma. Ludmilla Reis Pinheiro dos Santos

Coordenadora de Trabalho de Conclusão de Curso

GOIÂNIA
2022

Aos meus pais e à minha irmã.

AGRADECIMENTOS

Agradeço primeiramente a Deus, criador de todas as coisas. Aos meus pais que são minha base, minha irmã que me incentiva e me apoia em todos meus projetos. Aos meus tios, primos e meus amigos que estão sempre ao meu lado.

A minha orientadora Ludmilla Reis Pinheiro dos Santos pelas sábias orientações e pela paciência nessa árdua jornada. Aos professores José Luiz de Freitas Júnior e Carmen Cecília Centeno pelos conselhos e por terem aceito o convite para compor a banca de avaliação deste trabalho.

“I don’t believe that man was made to be controlled by machines. Machines can make laws, but they can not preserve justice. Only human beings can do that.”

(NATION, Terry; **Doctor Who - The Keys of Marinus**, 1964)

RESUMO

Linguagens Formais e Autômatos (LFA) é uma das teorias centrais da computação sendo este conteúdo de exigência para cursos superiores nessa área. A disciplina LFA aborda conceitos importantes como o entendimento de sistemas de estados e analisadores de cadeias, temas de relevância para a formação de estudantes.

A disciplina se mostra em alguns momentos complexa, sendo necessário usar ferramentas que auxiliem a aprendizagem do conteúdo. Com o objetivo de facilitar o estudo dos alunos da área da computação e auxiliar os professores, foi desenvolvido o aplicativo moblie que permite o fácil acesso a um ambiente de aprendizagem prática.

Através do aplicativo os professores podem criar exercícios do conteúdo de Autômatos Finitos Determinísticos (AFD) e disponibilizar para os alunos responderem. O aplicativo analisa as respostas e verifica se as mesmas estão erradas ou corretas. As ferramentas que foram utilizadas para o desenvolvimento do aplicativo foram a *engine* Unity, juntamente com a *Integrated Development Environment* (IDE) Visual Studio. O aplicativo foi desenvolvido usando linguagem de programação C#, sendo intuitiva de fácil manuseio pelos alunos e professores.

Palavras-chaves: *Aplicativo, Linguagens Formais e Autômatos.*

ABSTRACT

Formal Languages and Automata are one of the central theories on computing, this is a subject required for superior courses in this area. This subject address important concepts, like state systems, string analyzers, topics of relevance for the students. The students in the computing area must comprehend it.

The subject presents at times has a degree of complexity, there is a need for tools to aid the learning process of this subject. To help the students study and teachers teach was developed an app that easily allows access to a practical environment for the students and is also a teaching tool for teachers.

Through the application, teachers can create exercises on the subject of Deterministic Finite Automata, providing the exercises for the students to answer. The app analyzes the answers and checks if the answer is correct or if it is wrong. The software used to develop the application, was the engine Unity, with the Integrated Development Environment Visual Studio. The app was developed using C# language. The app is easy to use because its user interface is intuitive for teachers and students.

Keywords: *Application, Formal Languages and Automata.*

LISTA DE FIGURAS

Figura 1 - Um autômato finito que modela um interruptor. 21
Figura 2 - Exemplo de diagrama de estados para um AFD. 22
Figura 3 - AFDs equivalentes. 23
Figura 4 - AFD mínimo. 23
Figura 5 - Diagrama em blocos representando o aplicativo. 24
Figura 6 - Diagrama de Casos de Uso. 37
Figura 7 - Interface do Unity. 48
Figura 8 - Interface do Visual Studio. 49
Figura 9 - Interface do GitKraken. 49
Figura 10 - Firestore. 50
Figura 11 - Tela Login. 51
Figura 12 - Tela de Cadastro. 52
Figura 13 - Tela de Recuperação de Senha. 53
Figura 14 - Tela Principal Professor. 54
Figura 15 - Tela Workspace. 55
Figura 16 - Tela de Cadastro do Enunciado. 56
Figura 17 - Tela Workspace com o Menu Opções para usuário professor. 56
Figura 18 - Tela Workspace com o Menu Opções para o usuário aluno. 57
Figura 19 - Workspace com a mensagem de resposta correta 58
Figura 20 - Tela Workspace com o Menu Estado. 59
Figura 21 - Menu Nova Transição - Escolha dos Símbolos. 59
Figura 22 - Menu Nova Transição - Escolha do estado alvo. 60
Figura 23 - Tela Meus Exercícios. 61
Figura 24 - Tela Menu Exercício. 61
Figura 25 - Tela Meus Projetos. 62
Figura 26 - Tela Principal Aluno. 63
Figura 27 - Tela Lista de Exercícios. 63
Figura 28 - Tela Exercícios Resolvidos. 64
Figura 29 - Exemplo de AFDs Equivalente. 65
Figura 30 - Exemplo de AFDs Equivalente - Renomeando Passo I. 65
Figura 31 - Exemplo de AFDs Equivalente - Renomeando Passo II. 66
Figura 32 - Exemplo de AFDs Equivalente - Renomeando Passo III. 66

Figura 33 - Tela de Cadastro do Enunciado. 68
Figura 34 - Tela *Workspace* com AFD de solução..... 69
Figura 35 - Tela *Workspace*. 70
Figura 36 - Workspace com a mensagem de resposta correta. 71

LISTA DE QUADROS

Quadro 1 - Requisitos de Usuário Necessidade. 28
Quadro 2 - RF 001 Manter Cadastro do Usuário. 29
Quadro 3 - RF 002 Fazer login..... 29
Quadro 4 - RF 003 Recuperar senha..... 30
Quadro 5 - RF 004 Manter Cadastro de Exercício..... 30
Quadro 6 - RF 005 Manter Estados. 31
Quadro 7 - RF 006 Manter Transições..... 31
Quadro 8 - RF 007 Verificar Resposta. 32
Quadro 9 - RF 008 Manter Solução Do Exercício..... 32
Quadro 10 - RF 009 Visualizar exercícios cadastrados 33
Quadro 11 - RF 010 Cadastrar a Resposta Do Exercício 33
Quadro 12 - RF 011 Visualizar os Alunos que concluíram o exercício. 34
Quadro 13 - RQ 001 Confiabilidade. 34
Quadro 14 - RQ 002 Usabilidade 35
Quadro 15 - RQ 003 Desempenho. 35
Quadro 16 - DD 001 Dados do usuário..... 35
Quadro 17 - DD 002 Dados do Exercício..... 36
Quadro 18 - CSU 001 Manter Cadastro..... 37
Quadro 19 - CSU 002 Fazer Login..... 38
Quadro 20 - CSU 003 Recuperar Senha 38
Quadro 21 - CSU 004 Manter Estados. 39
Quadro 22 - CSU 005 Manter Transições..... 40
Quadro 23 - CSU 006 Manter Solução. 41
Quadro 24 - CSU 007 Manter Cadastro do Exercício. 42
Quadro 25 - CSU 008 Manter Resposta. 44
Quadro 26 - CSU 009 Enviar Resposta. 46

LISTA DE TABELAS

Tabela 1 - Tabela de Transições do AFD 23
Tabela 2 - Representação da string cabeçario 67
Tabela 3 - Lista de transições. 67
Tabela 4 - Lista de estados finais..... 67

LISTA DE SIGLAS

AFD	Autômatos Finitos Determinísticos
AFND	Autômatos Finitos Não Determinísticos
CSU	Caso de Uso
DD	Dicionário de Dados
GAM	*Ginux Abstract Machine*
IDE	*Integrated Development Environment*
LFA	Linguagens Formais e Autômatos
MEC	Ministério da Educação e Cultura
RF	Requisitos Funcionais
RQ	Requisitos de Qualidade
RUN	Requisitos de Usuário Necessidades

SUMÁRIO

1 INTRODUÇÃO 16
1.1 Objetivo Geral 17
1.2 Objetivos específicos 17
1.3 Procedimentos Metodológicos 17
1.4 Justificativa 18
2 TEORIA DA COMPUTAÇÃO, LINGUAGENS FORMAIS E AUTÔMATOS 19
2.1 Teoria Da Computação 19
2.2 Linguagens Formais e Autômatos 20
2.3 Autômatos Finitos 21
2.4 Autômatos Finitos Determinísticos 21
2.5 Minimização de um Autômato Finito 23
3 DESCRIÇÃO GLOBAL 24
3.1 Aspecto geral do produto 24
3.1.1 Interfaces do Sistema 25
3.1.2 Interfaces Do Usuário 26
3.1.3 Interfaces de Hardware 26
3.2 Funções do aplicativo 26
3.3 Características Do Usuário 26
3.4 Suposições e Dependência 27
4 DOCUMENTAÇÃO 28
4.1 Requisitos de Usuário – Necessidade 28
4.2 Requisitos Funcionais 29
4.3 Requisitos de Qualidade 34
4.4 Dicionário de Dados 35
4.5 Casos de Uso 37
5 DESENVOLVIMENTO DO APLICATIVO 48
5.1 Unity 48
5.2 Visual Studio 48
5.3 GitKraken 49
5.4 Firebase 50
6 IMPLEMENTAÇÃO E RESULTADOS 51
6.1 Telas do Aplicativo 51
6.2 Tela Login 51
6.3 Tela de Cadastro 52
6.4 Tela de Recuperação de Senha 52

6.5 Tela Principal Professor 53
6.6 Tela *Workspace* 54
6.6.1 Tela de Cadastro do Enunciado 55
6.6.2 Menu Opções 56
6.6.3 Mensagem de Resposta Correta 57
6.6.4 Menu estado 58
6.6.5 Menu Nova Transição 59
6.7 Tela Meus Exercícios 60
6.8 Tela Menu Exercício 61
6.9 Tela Meus Projetos 62
6.10 Tela Principal Aluno 62
6.11 Tela Lista de Exercícios 63
6.12 Tela Exercícios Resolvidos 64
6.13 Comparação entre AFDs 64
6.14 Cadastro de Exercício e Envio de Resposta 68
6.14.1 Cadastro de Exercício 68
6.14.2 Envio de Resposta 69
6.15 Resultados 71
7 Considerações Finais 72
7.1 Trabalhos futuros 72
REFERÊNCIAS 73
APÊNDICE A - MANUAL DE USO DO APLICATIVO 75
APÊNDICE B - TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA 84

1 INTRODUÇÃO

Autômatos são modelos computacionais idealizados para estabelecer uma teoria matemática manuseável. Através desse modelo pode-se observar com clareza os problemas complexos tanto no hardware como no software. Segundo Pinheiro (2017), a teoria dos autômatos é uma sub divisão da ciência da computação tendo como objeto de estudo máquinas de estados finitas e suas representações matemáticas. Autômato é um modelo matemático o qual é um objeto de estudo da ciência da computação.

Os autômatos finitos são importantes por serem utilizados tanto em implementações de softwares, quanto em verificações de hardware. Os autômatos finitos podem ser usados em grandes volumes de texto como analisadores léxicos e pesquisas de palavras. Em hardware, podem ser utilizados para verificações de sistemas digitais e sistemas que possuem números finitos de estados (HOPCROFT; MOTWANI; ULLMAN, 2006).

Linguagens Formais e Autômatos (LFA) é uma das teorias centrais da computação, sendo este conteúdo de exigência para todos cursos superiores da área de acordo com as Diretrizes Curriculares Nacionais do Ministério da Educação e Cultura (MEC, 2012). O conteúdo é uma exigência do MEC, os alunos da área da computação precisam compreendê-lo, sendo esse importante para vida acadêmica. A grande maioria dos alunos tem dificuldade pois o conteúdo é altamente teórico, sendo necessário usar de ferramentas com recursos visuais que auxilie o processo de aprendizagem.

Este trabalho apresenta um aplicativo *mobile* para auxiliar o estudo dos conceitos de LFA, especificamente o conteúdo de Autômatos Finitos Determinísticos (AFD). O objetivo deste trabalho é tornar acessível para os estudantes na área da computação uma plataforma onde o aluno pode estudar de forma prática o conteúdo. O aplicativo permite ao professor criar exercícios aos quais são disponibilizados aos alunos para que possam resolvê-los.

Existem alguns trabalhos que têm como objetivo auxiliar os estudantes da computação em relação ao conteúdo, por exemplo o *Ginux Abstract Machine* (GAM) que foi desenvolvido para o sistema operacional Linux (JUKEMURA, 2004). Essa ferramenta tem apresentação intuitiva e fácil de ser usada tendo como objetivo ser um ambiente de estudo do conteúdo de Linguagens Formais e Autômatos.

1.1 Objetivo Geral

Desenvolver e documentar um aplicativo para auxiliar o estudo de Linguagens Formais e Autômatos, mais especificamente do conteúdo de Autômatos Finitos Determinísticos, para os estudantes e para professores de engenharia e ciência da computação.

1.2 Objetivos específicos

- Fazer levantamento dos requisitos funcionais, de usuários necessidade e de qualidade para a implementação do aplicativo;
- Modelar o diagrama de caso de uso;
- Apresentar os casos de uso no modelo descritivo;
- Desenvolver o aplicativo *mobile* para auxiliar o estudo de AFD;
- Proporcionar ao aluno um ambiente de estudo onde o mesmo irá resolver exercícios e modelar diagramas de AFD;
- Proporcionar ao professor uma ferramenta para publicar exercícios do conteúdo de AFD;

1.3 Procedimentos Metodológicos

Este trabalho apresenta um levantamento bibliográfico e experimental. Através de teses, artigos e livros relatando a importância do conteúdo e a complexidade do mesmo, foram utilizadas as técnicas e os métodos presentes no processo de levantamento de requisitos. Dessa forma, foram observados as condições e os resultados do aplicativo, buscando solucionar possíveis falhas com o objetivo de desenvolver um aplicativo com funções que atendam às necessidades dos usuários.

O aplicativo foi desenvolvido usando a Unity, que utiliza a linguagem de programação C#. A Unity foi escolhida para o desenvolvimento, pois é de fácil utilização e contém uma vasta documentação online, que possibilita a resolução de possíveis problemas e *bugs*. Por ser uma ferramenta multiplataforma é possível portar o aplicativo para outras plataformas podendo abranger uma quantidade significativa de usuários.

1.4 Justificativa

O conteúdo de LFA é indispensável na formação dos estudantes na área da computação sendo essa disciplina obrigatória em todas as Diretrizes Curriculares Nacionais para todos os cursos superiores da área de computação, segundo o MEC (MEC, 2012).

A disciplina LFA aborda conceitos importantes como o entendimento de sistemas de estados e analisadores de cadeias, temas de relevância para a formação de estudantes da área. Considerando que o conteúdo é complexo e gera dificuldades de compreensão pelos estudantes, o aplicativo terá como objetivo contribuir de forma positiva para a aprendizagem da disciplina.

Segundo Ramos (2009), a disciplina de LFA é um dos tópicos considerados pelos estudantes excessivamente árduos e abstratos, fato que contribui para reduzir o interesse na aprendizagem. Este trabalho apresenta um aplicativo, com o objetivo despertar o interesse e motivar o estudo dessa matéria fundamental para vida acadêmica dos estudantes e tornar a aprendizagem da mesma eficaz.

2 TEORIA DA COMPUTAÇÃO, LINGUAGENS FORMAIS E AUTÔMATOS

Este capítulo apresenta a definição e a importância da Teoria da Computação. Apresenta também definições de LFA assim como a definição de conceitos centrais da Teoria dos Autômatos, tais como: alfabeto, cadeias e linguagem. Este capítulo também apresenta as formas de representação do AFD, por meio de quíntuplas, diagramas e tabelas. E por último discorre sobre o processo de minimização e comparação entre autômatos.

2.1 Teoria Da Computação

A Ciência da Computação e a Teoria da computação surgiram no início do século XX (JOSÉ NETO, 2009). Matemáticos como George Boole, Turing, Gödel e Chomsky, contribuíram através de pesquisas e soluções de problemas, muitos deles antes mesmo da criação do conceito dos computadores atuais. George Boole, com uma contribuição relevante, desenvolveu a álgebra booleana e formulou questões bases da computação.

> Com as funções por ele idealizadas, tornou-se possível criar modelos precisos, tanto do funcionamento das volumosas, lentas e dispendiosas redes de circuitos lógicos eletromecânicos, utilizadas nos primeiros computadores, como igualmente dos ágeis, compactos e econômicos circuitos digitais microeletrônicos modernos (José Neto, 2009, p.07).

Segundo José Neto (2009), essas teorias estudam os modelos formais de computação, sua aplicabilidade e viabilidade de forma prática. Assim como resolução de problemas, essas teorias foram os alicerces para a criação do computador usado atualmente.

Pinheiro (2017) define a Teoria da Computação como um subcampo da área da ciência da computação, tendo como finalidade encontrar problemas que podem ser computados através da utilização de modelos. A teoria da computação apresenta modelos computacionais que são utilizados para análise de problemas para que se possa entendê-los e resolvê-los.

2.2 Linguagens Formais e Autômatos

Linguagens Formais e Autômatos tem como objeto de estudo as formas de representação das linguagens, suas classes e suas propriedades, como também os modelos computacionais que são capazes de reconhecê-las. As linguagens são caracterizadas por um conjunto de cadeias de comprimento finito de um determinado alfabeto.

Segundo Menezes (2011), a linguagem é um dos conceitos fundamentais da teoria da computação. É uma forma precisa de expressar problemas, o que possibilita o desenvolvimento adequado do estudo da computabilidade.

Os estudos teóricos das linguagens e dos autômatos conduziu a criação dos compiladores, que são ferramentas que traduzem linguagens de alto nível para linguagem de máquina. Os compiladores traduzem linguagens de fácil entendimento pelos seres humanos para a linguagem que é compreendida pelos computadores, possibilitando que o programador execute comandos que são traduzidos pelos compiladores e executados pelos computadores, o que permite desenvolvimento de software com alta complexidade.

Segundo Ramos (2021), o estudo de linguagens formais possui os principais conceitos:

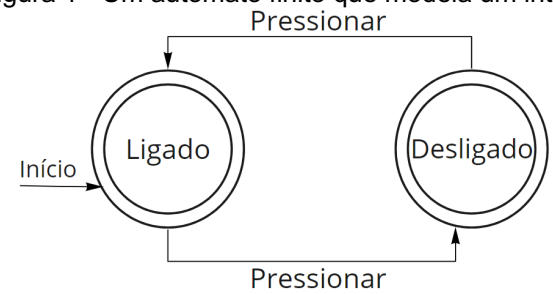
a) Símbolo: é uma representação gráfica utilizada para a construção de cadeias e do alfabeto de linguagens;
b) Alfabeto: é um conjunto de símbolos, sendo representado pela letra grega sigma (Σ), por exemplo: Σ = {0, 1} é o alfabeto binário, que contém 0 e 1 como seus símbolos;
c) Cadeia: corresponde à justaposição de símbolos de um alfabeto. Por exemplo, no alfabeto Σ = {0, 1}, pode ser escolhido 010101 como uma cadeia gerada a partir desse alfabeto;
d) Sentenças: são conjuntos de cadeias pertencentes a uma linguagem formal.

2.3 Autômatos Finitos

“A teoria de autômatos é o estudo dos dispositivos de computação abstratos, ou ‘máquinas’” (HOPCROFT; ULLMAN; MOTWANI, 2006 p.1). A teoria define modelos teóricos ou máquinas abstratas, sendo objeto de estudo da área da computação as quais são empregadas para a representar linguagens regulares e resolver questões computacionais.

Para melhor compreensão do que é uma representação de um autômato finito, pode-se observar o exemplo, de um interruptor liga e desliga, apresentado a Figura 1. Se o interruptor está no estado desligado, ao ser pressionado mesmo conduzirá ao estado ligado e se ao pressionar novamente voltará ao estado desligado, dessa forma alternando entre os estados.

Figura 1 - Um autômato finito que modela um interruptor.



Fonte: Elaborado pelo autor.

Pode-se observar na Figura 1, que os estados são representados por círculos e as transições, as setas entre eles, indicam o estado destino dada a entrada “Pressionar”. No exemplo da Figura 1, ambos os estados possuem dois círculos, o que significa que são estados finais, sendo assim os dois estados de aceitação.

Um autômato finito se define como um modelo matemático de estados finitos que é capaz de fazer a leitura de símbolos de uma determinada cadeia aceitando ou rejeitando essa entrada.

2.4 Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos são máquinas abstratas que possuem transições bem definidas, sendo que para cada símbolo do alfabeto dado, há uma

única opção de transição. O autômato inicia em um estado inicial e avança para próximos estados até o fim da cadeia de símbolos.

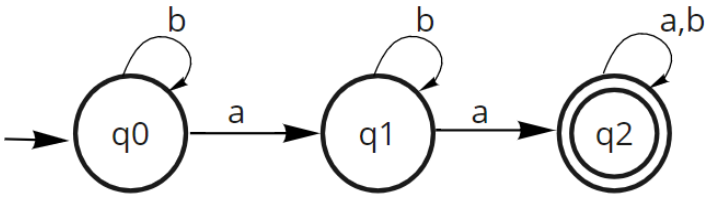
Segundo Sipser (2005 p.55), uma das formas que o autômato finito pode ser representado, é como uma 5-tupla M = (Q, Σ, δ, q0, F), onde:

- Q é um conjunto finito de estados;
- Σ é o alfabeto finito;
- δ : Q × Σ → Q, é a função de transição;
- q0 ∈ Q é o estado inicial;
- F é o conjunto de estados finais de aceitação.

No exemplo da Figura 2, apresenta uma outra forma do AFD ser representado, por um diagrama de estados. Pode-se observar que esse autômato contém 3 estados: q0, q1 e q2, sendo q0 o estado inicial e q2 o estado final. Quando esse recebe a cadeia “baab” ocorrem as seguintes transições:

- Leitura b, transição de q0 para o próprio q0;
- Leitura a, transição de q0 para q1;
- Leitura a, transição de q1 para q2;
- Leitura b, transição de q2 para o próprio q2.

Figura 2 - Exemplo de diagrama de estados para um AFD.



Fonte: Elaborado pelo autor.

Quando o último símbolo da cadeia é lido, se a execução do autômato estiver em um estado final, quer dizer que a cadeia é aceita. Se a última transição do autômato não for para um estado final, a cadeia é rejeitada. A linguagem que é condição para que uma cadeia seja aceita nesse exemplo é que ela deve conter pelo menos dois símbolos “a”.

Outra maneira que o autômato pode ser representado é por tabelas, o mesmo autômato da Figura 2 pode ser representado conforme a Tabela 1.

Tabela 1 - Tabela de Transições do AFD

	a	**b**
q0	q1	q0
q1	q2	q1
q2	q2	q2

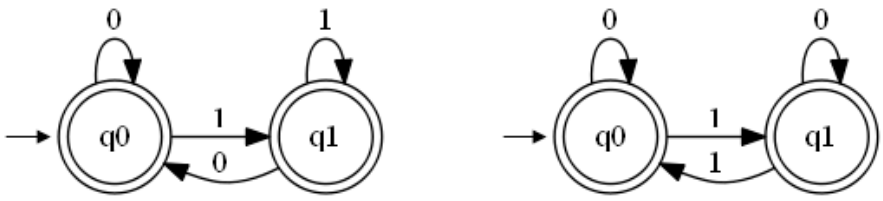
Fonte: Elaborado pelo autor.

2.5 Minimização de um Autômato Finito

A minimização de Autômatos Finitos é o processo que produz um Autômato Finito equivalente com o mínimo de estados possíveis. O Autômato Finito Mínimo tem uma formação única, dessa forma dois autômatos diferentes que aceitam a mesma linguagem, produz o mesmo Autômato Mínimo distinguindo as vezes pela identificação dos estados. O algoritmo de minimização reúne os estados equivalentes diminuindo assim seu número (MENEZES, 2000).

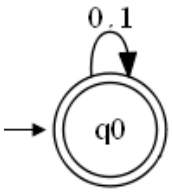
Por exemplo, a Figura 4 apresenta dois AFDs equivalentes que possuem diferentes transições. O AFD mínimo apresentado na Figura 5 é gerado pelo processo de minimização, apesar dos dois AFDs serem diferentes, o resultado gerado é o mesmo para ambos.

Figura 3 - AFDs equivalentes.



Fonte: Elaborado pelo autor.

Figura 4 - AFD mínimo.



Fonte: Elaborado pelo autor.

3 DESCRIÇÃO GLOBAL

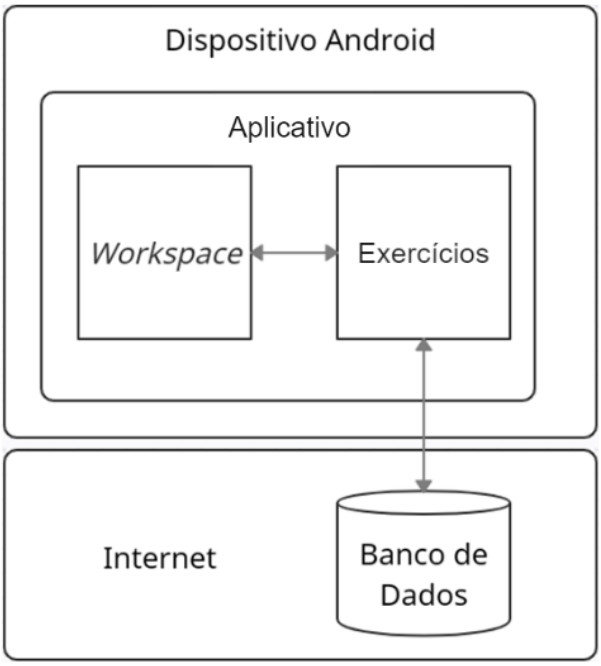
Este capítulo apresenta os perfis dos usuários que são os alunos e professores, a descrição das interfaces, suas funcionalidades e suas limitações.

3.1 Aspecto geral do produto

Os usuários do aplicativo devem realizar um cadastro que permite o acesso aos recursos da mesma, através disso, é possível efetuar o login. O aplicativo apresenta recursos para dois tipos de usuários: professores e alunos. O usuário professor cadastra exercícios, enquanto o usuário aluno pode resolvê-los.

Os usuários tem acesso ao *Workspace,* o qual apresenta a interface onde pode ser construído o diagrama de estados. Depois de elaborado o projeto, os dados são salvos na nuvem, portanto faz comunicação com o banco de dados por meio da internet, como representado na Figura 7.

Figura 5 - Diagrama em blocos representando o aplicativo.



Fonte: Elaborado pelo autor.

Para o cadastramento de questões, o usuário professor informa o enunciado do exercício para o AFD, o alfabeto, o diagrama de estados da solução do exercício na sua forma minimizada. Para a verificação do resultado informado pelo usuário

aluno, o aplicativo faz o processo de minimização do diagrama, e depois compara com o resultado esperado.

3.1.1 Interfaces do Sistema

O aplicativo possui as seguintes interfaces:

- **Tela Login:** Possui campos para receber informações de e-mail, senha, a opção para ir à tela de cadastro e recuperação de senha;
- **Tela de Cadastro:** Possui campos para receber informações para o cadastramento dos usuários;
- **Tela de Recuperação de Senha:** Possui um campo para receber o e-mail do usuário;
- **Tela Principal Professor**: Possui as opções, “Cadastrar exercícios”, “Meus Exercícios”, “Meus Projetos” e “Sair”;
- **Tela Principal Aluno**: Possui as opções, “Lista de Exercícios”, “Meus “Exercícios resolvidos” e “Sair”;
- **Tela *Workspace***: É um espaço de trabalho onde os usuários podem construir o diagrama AFD;
- **Tela Meus Exercícios:** é onde é listado todos os exercícios publicados pelo usuário professor;
- **Tela Meus Projetos:** Lista todos os exercícios criados pelo usuário professor incluindo os exercícios não publicados;
- **Tela Menu Exercício:** Tem a opção de “Editar a Solução” do exercício cadastrado, a opção de “Excluir o Exercício” e apresenta a lista de todos os usuários alunos que concluíram corretamente o exercício selecionado.
- **Tela Lista de Exercícios:** Essa tela lista as questões cadastradas no banco de dados do aplicativo**;**
- **Tela de Exercícios Resolvidos:** Essa tela tem a função de listar os exercícios concluídos pelo usuário.

3.1.2 Interfaces Do Usuário

O usuário precisa possuir um aparelho *smartphone* com o sistema operacional Android 5.1 ou superior e ter acesso à internet.

3.1.3 Interfaces de Hardware

Os requisitos mínimos de hardware para o aplicativo são:

- Processador: 1.0GHz Quad-Core;
- Memória RAM: 1 GBs;
- Armazenamento livre: 90 MBs.

3.2 Funções do aplicativo

O aplicativo possui as seguintes funções:

- Realizar o cadastro do usuário;
- Realizar login;
- Recuperar senha;
- Cadastrar exercício;
- Cadastrar a solução de exercício;
- Verificar se a resposta está correta;
- Enviar resposta.

3.3 Características Do Usuário

O aplicativo é orientado para alunos da área de computação que buscam um auxílio para o entendimento da disciplina de LFA, mais especificamente para os AFD. O aplicativo também é voltado para professores da área da computação com objetivo de auxiliar o ensino de AFD

3.4 Suposições e Dependência

Para a resolução de exercícios o aplicativo faz comunicação com o servidor de banco de dados Firebase, portanto é necessário conexão com a internet. Essa também é necessária para que o aplicativo possa salvar os dados dos usuários e os exercícios cadastrados.

4 DOCUMENTAÇÃO

Este capítulo apresenta os Requisitos de Usuário e de Sistema apresentando, Requisitos de Usuário - Necessidades (RUN), os Requisitos Funcionais (RF) e os Requisitos de Qualidade (RQ).

4.1 Requisitos de Usuário – Necessidade

Esta seção apresenta os Requisitos de Usuário às necessidades do mesmo em relação ao sistema.

Quadro 1 - Requisitos de Usuário Necessidade.

ID	Descrição	Fonte
RUN 001	Os usuários fazem cadastro no aplicativo	Autoria própria
RUN 002	Os usuários fazem login no aplicativo	Autoria própria
RUN 003	Os usuários fazem a recuperação de senha	Autoria própria
RUN 004	O usuário aluno acessa os exercícios cadastrados	Autoria própria
RUN 005	O usuário aluno faz registro da resposta do exercício	Autoria própria
RUN 006	O usuário aluno exclui a resposta do exercício	Autoria própria
RUN 007	O usuário professor faz o registro do alfabeto	Autoria própria
RUN 008	Os usuários fazem o registro de estados	Autoria própria
RUN 009	Os usuários fazem o registro das transições	Autoria própria
RUN 010	O usuário professor faz cadastro de exercício	Autoria própria
RUN 011	O usuário professor exclui o exercício cadastrado	Autoria própria
RUN 012	O usuário professor faz o registro da solução de exercício	Autoria própria
RUN 013	O usuário professor visualiza alunos que fizeram o exercício	Autoria própria

Fonte: Elaborado pelo autor.

4.2 Requisitos Funcionais

Esta seção apresenta os requisitos funcionais do aplicativo.

Quadro 2 - RF 001 Manter Cadastro do Usuário.

Identificador	**Nome**
RF 001	Manter Cadastro do Usuário
Caso de Uso	**Autor**
CSU 001	Daniel Figueiredo Pereira
Descrição	
O usuário preenche os campos com e-mail e senha.	
Critério de Verificação	
• Verificar se o e-mail e senha são válidos; • Verificar se o usuário já tem cadastro.	
Dependência	**Prioridade**
RUN 001, DD 001	Essencial

Fonte: Elaborado pelo autor.

Quadro 3 - RF 002 Fazer login.

Identificador	**Nome**
RF 002	Fazer login
Caso de Uso	**Autor**
CSU 002	Daniel Figueiredo Pereira
Descrição	
O usuário preenche os campos de e-mail e senha.	
Critério de Verificação	
• Verifica se o e-mail possui cadastro; • Verifica se a senha está correta.	
Dependência	**Prioridade**
RUN 002	Essencial

Fonte: Elaborado pelo autor.

Quadro 4 - RF 003 Recuperar senha.

Identificador	**Nome**
RF 003	Recuperar senha
Caso de Uso	**Autor**
CSU 003	Daniel Figueiredo Pereira
Descrição	
Envia um e-mail de recuperação de senha para o usuário.	
Critério de Verificação	
• Verifica se o e-mail possui cadastro no sistema.	
Dependência	**Prioridade**
RUN 003	Essencial

Fonte: Elaborado pelo autor.

Quadro 5 - RF 004 Manter Cadastro de Exercício.

Identificador	**Nome**
RF 004	Manter Cadastro de Exercício
Caso de Uso	**Autor**
CSU 007	Daniel Figueiredo Pereira
Descrição	
Permite o usuário professor ciar exercício podendo editar, excluir e listar.	
Critério de Verificação	
• Verifica se a solução do exercício é um AFD mínimo;	
Dependência	**Prioridade**
RUN 010, RUN 011, RUN 012, RUN 013	Essencial

Fonte: Elaborado pelo autor.

Quadro 6 - RF 005 Manter Estados.

Identificador	**Nome**
RF 005	Manter Estados
Caso de Uso	**Autor**
CSU 004	Daniel Figueiredo Pereira
Descrição	
Permite o usuário fornecer os estados na construção do AFD	
Critério de Verificação	
• Deve conter um estado inicial; • Deve conter pelo menos um estado final.	
Dependência	**Prioridade**
RUN 008	Essencial

Fonte: Elaborado pelo autor.

Quadro 7 - RF 006 Manter Transições.

Identificador	**Nome**
RF 006	Manter Transições
Caso de Uso	**Autor**
CSU 005	Daniel Figueiredo Pereira
Descrição	
Permite o usuário fornece as transições entre os estados para o aplicativo.	
Critério de Verificação	
Dependência	**Prioridade**
RUN 009	Essencial

Fonte: Elaborado pelo autor.

Quadro 8 - RF 007 Verificar Resposta.

Identificador	**Nome**
RF 007	Verificar Resposta
Caso de Uso	**Autor**
CSU 009	Daniel Figueiredo Pereira
Descrição	
O sistema verifica se a resposta que o usuário aluno inseriu está correta	
Critério de Verificação	
• Verifica se a resposta é um AFD	
Dependência	**Prioridade**
RF 005, RF 006, RUN 004, RUN 005	Essencial

Fonte: Elaborado pelo autor.

Quadro 9 - RF 008 Manter Solução Do Exercício.

Identificador	**Nome**
RF 008	Manter Solução do Exercício
Caso de Uso	**Autor**
CSU 006	Daniel Figueiredo Pereira
Descrição	
Permite o usuário professor fornecer a solução do exercício.	
Critério de Verificação	
• Verifica se a solução do usuário professor é um AFD mínimo.	
Dependência	**Prioridade**
RUN 007, RUN 008, RUN 009, RUN 010 RUN 011, RUN 012, RUN 013, RUN 014	Essencial

Fonte: Elaborado pelo autor.

Quadro 10 - RF 009 Visualizar exercícios cadastrados

Identificador	**Nome**
RF 010	Visualizar Exercícios Cadastrados
Caso de Uso	**Autor**
CSU 007	Daniel Figueiredo Pereira
Descrição	
Permite o usuário aluno visualizar os exercícios cadastrados no aplicativo.	
Critério de Verificação	
Dependência	**Prioridade**
RUN 004	Essencial

Fonte: Elaborado pelo autor.

Quadro 11 - RF 010 Cadastrar a Resposta Do Exercício

Identificador	**Nome**
RF 010	Cadastrar a resposta do exercício
Caso de Uso	**Autor**
CSU 007	Daniel Figueiredo Pereira
Descrição	
Permite o usuário fornecer a resposta a um exercício.	
Critério de Verificação	
• Verifica se a resposta está correta.	
Dependência	**Prioridade**
RUN 007, RUN 008, RUN 009, RUN 010 RUN 011, RUN 012, RUN 013, RUN 014	Essencial

Fonte: Elaborado pelo autor.

Quadro 12 - RF 011 Visualizar os Alunos que concluíram o exercício.

Identificador	**Nome**
RF 012	Visualizar os Alunos que concluíram o exercício
Caso de Uso	**Autor**
CSU 006	Daniel Figueiredo Pereira
Descrição	
Permite o usuário professor visualizar os alunos que concluíram os exercícios.	
Critério de Verificação	
Dependência	**Prioridade**
RUN 013	Essencial

Fonte: Elaborado pelo autor.

4.3 Requisitos de Qualidade

Esta seção apresenta as os requisitos de qualidade do aplicativo.

Quadro 13 - RQ 001 Confiabilidade.

Identificador	**Nome**
RQ 001	Confiabilidade
Caso de Uso	**Autor**
	Daniel Figueiredo Pereira
Descrição	
O aplicativo deve ser confiável, armazenando os dados no servidor de forma segura.	
Critério de Verificação	
Dependência	**Prioridade**
	Essencial

Fonte: Elaborado pelo autor.

Quadro 14 - RQ 002 Usabilidade

Identificador	**Nome**
RQ 002	Usabilidade
Caso de Uso	**Autor**
	Daniel Figueiredo Pereira
Descrição O aplicativo deve ser intuitivo para ambos os tipos de usuários, de forma que seja fácil o seu manuseio.	
Critério de Verificação	
Dependência	**Prioridade**
	Essencial

Fonte: Elaborado pelo autor.

Quadro 15 - RQ 003 Desempenho.

Identificador	**Nome**
RQ 003	Desempenho
Caso de Uso	**Autor**
	Daniel Figueiredo Pereira
Descrição O aplicativo deve ter um desempenho satisfatório.	
Critério de Verificação	
• Rápido resultado na realização de buscas.	
Dependência	**Prioridade**
	Essencial

4.4 Dicionário de Dados

Quadro 16 - DD 001 Dados do usuário.

Identificador	**Descrição**			
DD 001	Dados do usuário			
Identificador	**Tamanho**	**Tipo**	**Formato**	**Descrição**
uid	28	String		Id do Usuário
nome	30	String		Nome do usuário
e-mail	35	String	\w+@\w+\.\w+	E-mail do usuário
senha	30	String		Senha do usuário

tipoDeUsuario	1	Bool		Tipo de usuário

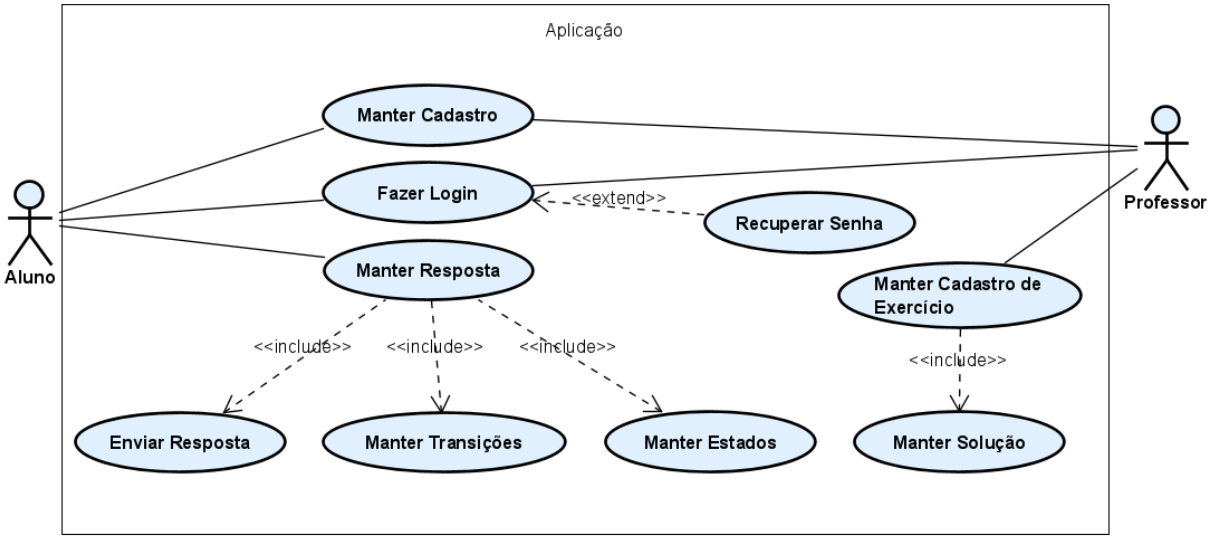
Fonte: Elaborado pelo autor.

Quadro 17 - DD 002 Dados do Exercício.

Identificador	**Descrição**			
DD 002	Dados do Exercício			
Identificador	**Tamanho**	**Tipo**	**Formato**	**Descrição**
IdUser	28	String		Id do Usuário
userName	30	String		Nome do usuário
Enunciado	41	String		Enunciado
quantosEstados	46	Int		E-mail do usuário
alfabeto	40	String		Senha do usuário
Estados	39	String		Tipo de usuário
estadosPosX	43	String		Coordenada X dos Estados
estadosPosY	43	String		Coordenada Y dos Estados
estadoInicial	45	String		Nome do Estado Inicial
estadosFinais	45	String		Nomes do Estados Finais
transistionStates1	50	String		Nomes dos Estados de origem das transições
transistionStates2	50	String		Nomes dos Estados de destino das transições
transistionSymbols	50	String		Símbolos das transições
miniHeader	14	String		Cabeçario da solução do exercício
miniTransitions	47	String		Transições da solução do exercício
miniEstadosFinais	49	String		Estados Finais da solução do exercício
alunos	37	String		Alunos que concluíram os exercícios

4.5 Casos de Uso

Figura 6 - Diagrama de Casos de Uso.



Fonte: Elaborado pelo autor.

Quadro 18 - CSU 001 Manter Cadastro.

Identificador	**CSU 001**
Nome	**Manter Cadastro**
Atores	Aluno, Professor
Requisitos	RF 001
Responsável	Daniel Figueiredo Pereira
Descrição/Resumo	Este caso de uso representa o cadastro dos usuários.
Pré-condições	• Acesso à internet e possuir uma conta e-mail.
Pós-condições	• Cadastro realizado.
Cenário Principal	1. O sistema exibe a tela de login; 2. O usuário seleciona a opção “Cadastre-se”; 3. O sistema exibe a tela de cadastro; 4. O usuário preenche os campos; 5. O usuário seleciona a opção “Confirmar”; 6. O sistema verifica se os dados estão corretos; 7. O sistema armazena os dados no banco de dados; 8. O usuário é redirecionado para a tela principal; 9. Fim do caso de uso.
Cenário alternativo	**(A1) Passos 3 e 4 – O usuário cancela o cadastro** 1.a O usuário seleciona a opção “voltar”; 2.a O sistema retorna ao passo 4 do cenário principal;
Cenário de exceção	(**E1) Passo 6 - Campos em branco.** 1.a O sistema apresenta uma mensagem de erro informando que são necessários os preenchimentos dos campos;

	2.a O sistema retorna ao passo 4 do cenário principal.

Fonte: Elaborado pelo autor.

Quadro 19 - CSU 002 Fazer Login.

Identificador	**CSU 002**
Nome	**Fazer Login**
Atores	Aluno, Professor
Requisitos	RF 002
Responsável	Daniel Figueiredo Pereira
Descrição/Resumo	Este caso de uso representa o login no aplicativo
Pré-condições	• O usuário possuir cadastro no sistema.
Pós-condições	• Usuário efeitua login no aplicativo.
Cenário Principal	1. O sistema exibe a tela de login; 2. O usuário preenche os campos “usuário” e “senha.”; 3. O usuário seleciona a opção “Entrar”; 4. O sistema verifica se os dados estão corretos; 5. O usuário é redirecionado para a tela principal do usuário; 6. Fim do caso de uso.
Cenários alternativo	**(A1) Passo 2 - O usuário ir para tela de cadastro** 1.a O usuário seleciona a opção “Cadastre-se”; 2.a O sistema retorna ao passo 6 do cenário principal.
Cenários de exceção	**(E1) Passo 4 – O usuário informar um e-mail ou senha inválidos** 1.a O sistema exibe uma mensagem de erro informando que os dados informados estão incorretos; 2.a O sistema retorna para o passo 2 do cenário principal.

Fonte: Elaborado pelo autor.

Quadro 20 - CSU 003 Recuperar Senha

Identificador	**CSU 003**
Nome	**Recuperar Senha**
Atores	Aluno, Professor
Requisitos	RF 003
Responsável	Daniel Figueiredo Pereira
Descrição/Resumo	Este caso de uso representa a recuperação de senha dos usuários.
Pré-condições	• Os usuários possuírem cadastro no sistema.
Pós-condições	• Os usuários cadastrar uma nova senha.
Cenário Principal	1. O sistema exibe a tela de login; 2. O sistema exibe a tela de recuperação senha; 3. O usuário preenche o campo com o e-mail;

	4. O sistema verifica se o e-mail é valido; 5. O sistema envia o e-mail de recuperação de senhas; 6. Fim do.
Cenários alternativo	**(A1) Passo 2 - O usuário ir para Tela Login** 1.a O usuário seleciona a opção “Voltar”; 2.a O sistema retorna ao passo 1 do cenário principal.
Cenários de exceção	**(E1) Passo 2 – O usuário informa um e-mail inválido** 1.a O sistema exibe uma mensagem de erro informando que não existe cadastro com o e-mail inserido; 2.a O sistema retorna para o passo 2.b do cenário alternativo. **(E2) Passo 2 – O usuário deixa em branco o campo de e-mail** 1.b O sistema exibe uma mensagem de erro informando que o campo de e-mail está vazio; 2.b O sistema retorna para o passo 2.b do cenário alternativo.

Fonte: Elaborado pelo autor.

Quadro 21 - CSU 004 Manter Estados.

Identificador	**CSU 004**
Nome	**Manter Estados**
Atores	Aluno, Professor
Requisitos	RF 005
Responsável	Daniel Figueiredo Pereira
Descrição/Resumo	Este caso de uso representa as funções de criar, editar e excluir estados.
Pré-condições	
Pós-condições	• Criação de estados.
Cenário Principal	1. O sistema exibe a tela *Workspace*; 2. O usuário pressiona o Botão Adição; 3. O sistema adiciona um novo estado ao *Workspace;* 4. Fim do caso de uso.
Cenários alternativo	**(A1) Passo 2 – Mover o estado** 1.a O usuário clica no estado; 2.a O usuário arrasta o estado para move-lo; 3.a O sistema salva a nova localização do estado; 4.a Fim do caso de uso. **(A2) Passo 2 – Alterar estado para estado inicial**

	1.b O usuário faz um clique longo no estado; 2.b O sistema exibe o Menu Estado; 3.b O usuário seleciona a opção “Estado Inicial”; 4.b O sistema altera o estado para um estado inicial; 5.b Fim do caso de uso. **(A3) Passo 2 – Alterar estado para estado final** 1.c O usuário faz um clique longo no estado; 2.c O sistema exibe o Menu Estado; 3.c O usuário seleciona a opção “Estado Final”; 4.c O sistema altera o estado para um estado final; 5.c Fim do caso de uso. **(A4) Passo 2 – Excluir estado** 1.d O usuário seleciona o estado; 2.d O usuário faz um clique longo no estado; 3.d O sistema exibe na tela o Menu Estado; 4.d O usuário seleciona a opção “Deletar Estado”; 5.d O sistema excluir o estado da tela; 6.d Fim do caso de uso.
Cenário de exceção	

Fonte: Elaborado pelo autor.

Quadro 22 - CSU 005 Manter Transições.

Identificador	**CSU 005**
Nome	**Manter Transições**
Atores	Aluno, Professor
Requisitos	RF 006
Responsável	Daniel Figueiredo Pereira
Descrição/Resumo	Este caso de uso representa o registro das transições entre os estados.
Pré-condições	• Usuário ter adicionado pelo menos um estado.
Pós-condições	• Registro das transições.
Cenário Principal	1. O sistema exibe a tela *Workspace*; 2. O usuário faz um clique longo no estado; 3. O sistema exibe o Menu Estado; 4. O usuário seleciona a opção “Nova Transição”; 5. O sistema exibe o Menu Nova Transição com os símbolos do alfabeto;

	6. O usuário seleciona os símbolos da transição; 7. O sistema exibe os estados adicionados; 8. O usuário seleciona o estado alvo da transição; 9. O sistema cria a transição nos estados selecionados; 10. Fim do caso de uso.
Cenário alternativo	**(A1) Passo 5 ao 8 – O usuário cancela a criação da transição.** 1.a O usuário seleciona a opção “Cancelar”; 2.a O sistema retorna para o passo 3 do cenário principal.
Cenário de exceção	**(E1) Passo 5 – O usuário não seleciona símbolo para a transição** 1.a O sistema exibe uma mensagem de erro informando que é preciso escolher um símbolo para a transição; 2.a O sistema retorna para o passo 5 do cenário principal.

Fonte: Elaborado pelo autor.

Quadro 23 - CSU 006 Manter Solução.

Identificador	**CSU 006**
Nome	Manter Solução
Atores	Professor
Requisitos	RF 008
Responsável	Daniel Figueiredo Pereira
Descrição/Resumo	Este caso de uso representa as ações de listar, criar, editar e excluir uma solução.
Pré-condições	• Usuário professor ter efetuado o login. • Usuário professor ter escolhido o exercício.
Pós-condições	
Cenário Principal	1. O sistema exibe a tela principal; 2. O usuário escolhe a opção “Meus Projetos”; 3. O sistema exibe a tela Meus Projeto; 4. O sistema lista os projetos criados pelo professor; 5. O usuário escolhe a opção “Novo”; 6. O sistema exibe a Tela Workspace com o Menu Enunciado; 7. O usuário preenche os campos de “Alfabeto” e “Enunciado”; 8. O sistema registra o enunciado e o alfabeto no exercício e fecha o Menu Enunciado;

	9. O usuário constrói o AFD solução do exercício proposto; 10. O usuário clica no ícone do Menu Opções; 11. O sistema exibe o Menu Opções; 12. O usuário seleciona a opção salvar; 13. O sistema salva o exercício no banco de dados; 14. Fim do caso de uso.
Cenários alternativo	**(A2) Passo 4 – Abrir e editar solução** 1.a O professor escolhe a opção um projeto da lista; 2.a O sistema exibe a Tela W*orkspace* a solução previamente cadastrada; 3.a O usuário edita o AFD; 4.a O usuário clica no ícone do Menu Opções; 5.a O sistema exibe o Menu Opções; 6.a O usuário seleciona a opção salvar; 7.a O sistema salva o exercício no banco de dados; 8.a Fim do caso de uso.
Cenário de exceção	**(E1) Passo 3 - O usuário não informou o alfabeto ou o enunciado** 1.a O sistema exibe uma mensagem de erro solicitando o preenchimento dos campos; 2.a O sistema retorna para o passo 3 do cenário principal.

Fonte: Elaborado pelo autor.

Quadro 24 - CSU 007 Manter Cadastro do Exercício.

Identificador	**CSU 007**
Nome	Manter Cadastro do Exercício
Atores	Professor
Requisitos	RF 004, RF 008
Responsável	Daniel Figueiredo Pereira
Descrição/Resumo	Este caso de uso representa o cadastro de exercício.
Pré-condições	• Professor ter efetuado o login.
Pós-condições	• Professor ter cadastrado o exercício.
Cenário Principal	1. O sistema exibe a tela principal; 2. O usuário escolhe a opção “Cadastrar Exercício”; 3. O sistema exibe a Tela Workspace com o Menu Enunciado; 4. O usuário preenche os campos de “Alfabeto” e “Enunciado”; 5. O sistema registra o enunciado e o alfabeto no exercício e fecha o Menu Enunciado; 6. O usuário constrói o AFD solução do exercício proposto;

	7. O usuário clica no ícone do Menu Opções; 8. O sistema exibe o Menu Opções; 9. O usuário seleciona a opção publicar exercício; 10. O sistema verifica se o autômato é um AFD; 11.O sistema verifica se o AFD é mínimo; 12. O sistema salva o exercício no banco de dados; 13. Fim do caso de uso.
Cenários alternativo	**(A1) Passo 1 – Listar exercícios cadastrados** 2.a O professor seleciona a opção “Meus Exercícios”; 3.a O sistema acessa o banco de dado e lista os exercícios cadastrados pelo usuário; 5.a Fim do caso de uso. **(A2) Passo 1 – Abrir e editar exercícios cadastrados** 1.a O professor seleciona a opção “Meus Exercícios”; 2.a O sistema acessa o banco de dados e lista os exercícios cadastrados no sistema pelo usuário; 3.a O usuário escolhe um exercício; 4.a O sistema exibe a Menu Exercício; 1.b O professor escolhe a opção “Editar solução” 2.b O sistema exibe a Tela W*orkspace* a solução previamente cadastrada; 3.b O usuário edita o AFD; 4.b O usuário clica no ícone do Menu Opções; 5.b O sistema exibe o Menu Opções; 6.b O usuário seleciona a opção publicar exercício; 7.b O sistema verifica se o autômato é um AFD; 8.b O sistema verifica se o AFD é mínimo; 9.b O sistema salva o exercício no banco de dados; 10.b Fim do caso de uso. **(A3) Passo 4.a – Excluir exercícios** 1.c O usuário seleciona o botão “Excluir”; 2.c O sistema mostra um menu perguntando se o usuário tem certeza; 3.c O usuário clica o botão “sim”; 4.c O sistema exclui o exercício do banco de dados; 5.c Fim do caso de uso;
Cenário de exceção	**(E1) Passo 3 - O usuário não informou o alfabeto ou o enunciado** 1.a O sistema exibe uma mensagem de erro solicitando o preenchimento dos campos;

	2.a O sistema retorna para o passo 3 do cenário principal; **(E2) Passo 10 - O usuário construiu um autômato sem estado inicial** 1.b O sistema exibe na tela uma mensagem de erro dizendo que o autômato não possui estado inicial; 2.b O sistema retorna para o passo 10 do cenário principal (**E3) Passo 10 – O usuário construiu um autômato sem estados finais** 1.c O sistema exibe na tela uma mensagem de erro dizendo que o autômato possui nenhum estado final; 2.c O sistema retorna para o passo 10 do cenário principal. **(E4) Passo 10 - O usuário construiu um autômato faltando transições** 1.d O sistema exibe na tela uma mensagem de erro dizendo qual estado está faltando transições; 2.d O sistema retorna para o passo 10 do cenário principal. **(E5) Passo 11 - O usuário construiu um AFD não mínimo** 1.e O sistema exibe na tela uma mensagem de erro dizendo que o autômato não é um AFD mínimo; 2.e O sistema retorna para o passo 10 do cenário principal.

Fonte: Elaborado pelo autor.

Quadro 25 - CSU 008 Manter Resposta.

Identificador	**CSU 008**
Nome	**Manter Resposta**
Atores	Aluno
Requisitos	RF 011
Responsável	Daniel Figueiredo Pereira
Descrição/Resumo	Este caso de uso representa o cadastro da resposta do usuário aluno.
Pré-condições	• O usuário estar logado no aplicativo; • O usuário ter aberto um exercício.
Pós-condições	• O usuário ter salvo uma resposta.

Cenário Principal	1. O sistema exibe a Tela Principal Aluno; 2. O usuário seleciona no menu a opção “Lista de Exercícios”; 3. O sistema exibe a Tela Lista de Exercícios e lista todos os exercícios cadastrados no aplicativo; 4. O usuário seleciona um exercício da lista; 5. O sistema exibe a Tela Workspace do exercício selecionado; 6. O usuário constrói o AFD para responder o exercício; 7. O usuário clica no ícone do Menu Opções; 8. O sistema exibe o Menu Opções; 9. O usuário seleciona a opção “Enviar a Resposta”; 10. O sistema verifica se o autômato construído é um AFD; 11. O sistema minimiza o AFD construído pelo usuário e compara com o AFD solução; 12. O sistema exibe a mensagem de parabéns; 13. O sistema salva a resposta do usuário aluno; 14. O usuário seleciona a opção OK; 15. Fim do caso de uso.
Cenário alternativo	**A1) Passo 12 – O usuário aluno insere uma resposta errada;** 1.a O sistema exibe uma mensagem informando que a resposta está incorreta; 2.a O sistema retorna para o passo 11 do cenário principal. **A2) Passo 2 – O usuário lista os exercícios resolvidos;** 1.b O usuário seleciona no menu a opção “Exercícios Resolvidos”; 2.b O sistema exibe a tela de Exercícios Resolvidos e lista todos os exercícios resolvidos pelo usuário; 3.b Fim do caso de uso. **A3) Passo 2.b – O usuário visualiza e edita a resposta cadastrada;** 1.c O usuário seleciona um exercício da lista; 2.c O sistema exibe a Tela *Workspace* com a resposta cadastrada pelo usuário aluno; 3.c O usuário clica no ícone do Menu Opções; 4.c O sistema exibe o Menu Opções; 5.c O usuário seleciona a opção “Enviar a Resposta”; 6.c O sistema verifica se o autômato construído é um AFD; 7.c O sistema minimiza o AFD construído pelo usuário e compara com o AFD solução; 8.c O sistema exibe a mensagem de parabéns;

	Fim do caso de uso.
Cenário de exceção	**(E1) Passo 11 - O usuário construiu um autômato sem estado inicial** 1.a O sistema exibe na tela uma mensagem de erro dizendo que o autômato não possui estado inicial; 2.a O sistema retorna para o passo 11 do cenário principal. (**E2) Passo 11 – O usuário construiu um autômato sem estados finais** 1.b O sistema exibe na tela uma mensagem de erro dizendo que o autômato possui nenhum estado final; 2.b O sistema retorna para o passo 11 do cenário principal. **(E3) Passo 11 - O usuário construiu um autômato faltando transições** 1.c O sistema exibe na tela uma mensagem de erro dizendo qual estado está faltando transições; 2.c O sistema retorna para o passo 11 do cenário principal.

Fonte: Elaborado pelo autor.

Quadro 26 - CSU 009 Enviar Resposta.

Identificador	**CSU 009**
Nome	**Enviar Resposta**
Atores	Aluno
Requisitos	RF 007
Responsável	Daniel Figueiredo Pereira
Descrição/Resumo	Este caso de uso representa o ato de verificar se a resposta está correta.
Pré-condições	• O usuário ter construído o AFD de resposta.
Pós-condições	• O sistema verificar se a resposta está certa ou errada.

Cenário Principal	1. O usuário seleciona a opção “Enviar a Resposta”; 2. O sistema verifica se o autômato construído é um AFD; 3. O sistema minimiza o AFD construído pelo usuário e compara com o AFD solução; 4. O sistema verifica que a resposta está correta; 5. Fim do caso de uso.
Cenário alternativo	**A1) Passo 4 – O usuário aluno insere uma resposta errada;** 1.a O sistema exibe uma mensagem informando que a resposta está incorreta; 2.a O sistema retorna para o passo 1 do cenário principal.
Cenário de exceção	**(E1) Passo 3 – O usuário aluno insere uma resposta errada;** 1.a O sistema exibe uma mensagem informando que a resposta está incorreta; 2.a O sistema retorna para o passo 1 do cenário principal. **(E2) Passo 2 - O usuário construiu um autômato sem estado inicial** 1.b O sistema exibe na tela uma mensagem de erro dizendo que o autômato não possui estado inicial; 2.b O sistema retorna para o passo 1 do cenário principal. (**E3) Passo 2 – O usuário construiu um autômato sem estados finais** 1.c O sistema exibe na tela uma mensagem de erro dizendo que o autômato possui nenhum estado final; 2.c O sistema retorna para o passo 1 do cenário principal. **(E4) Passo 2 - O usuário construiu um autômato faltando transições** 1.d O sistema exibe na tela uma mensagem de erro dizendo qual estado está faltando transições; 2.d O sistema retorna para o passo 1 do cenário principal.

Fonte: Elaborado pelo autor.

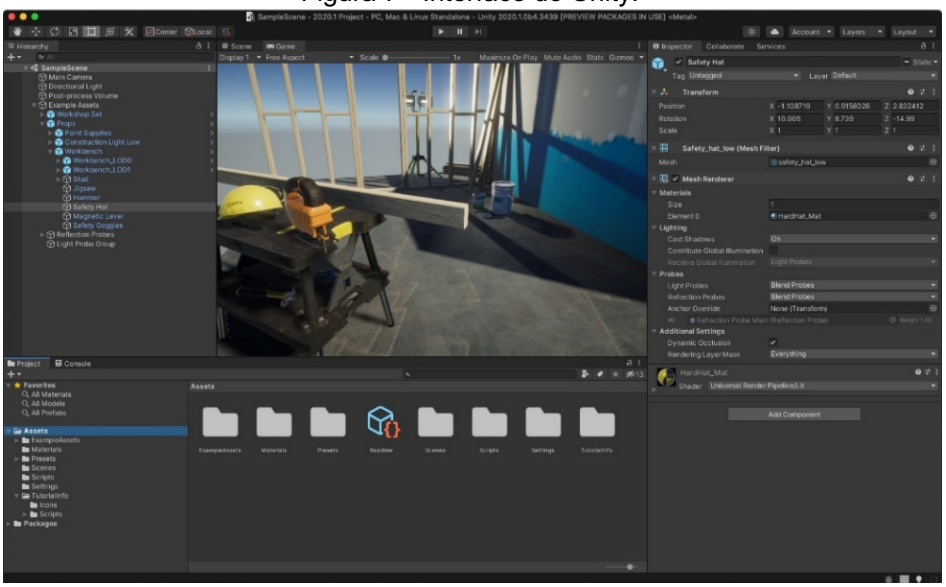
5 DESENVOLVIMENTO DO APLICATIVO

Este capítulo apresenta as ferramentas utilizadas para o desenvolvimento do aplicativo. O ambiente escolhido para a implementação foi a engine Unity, juntamente com a IDE Visual Studio. A ferramenta utilizada para controle de versão foi o GitKraken e o Firebase para autenticação e banco de dados.

5.1 Unity

Unity é um motor gráfico multiplataforma que possibilita desenvolvedores criarem aplicativos e jogos usando a linguagem de programação C#, o que possibilita portar os programas para outras plataformas que ele oferece suporte, como: Android, iOS, Windows, Linux e consoles de jogos. A Figura 9 exibe um exemplo de uso da interface da Unity. (UNITY, 2022).

Figura 7 - Interface do Unity.

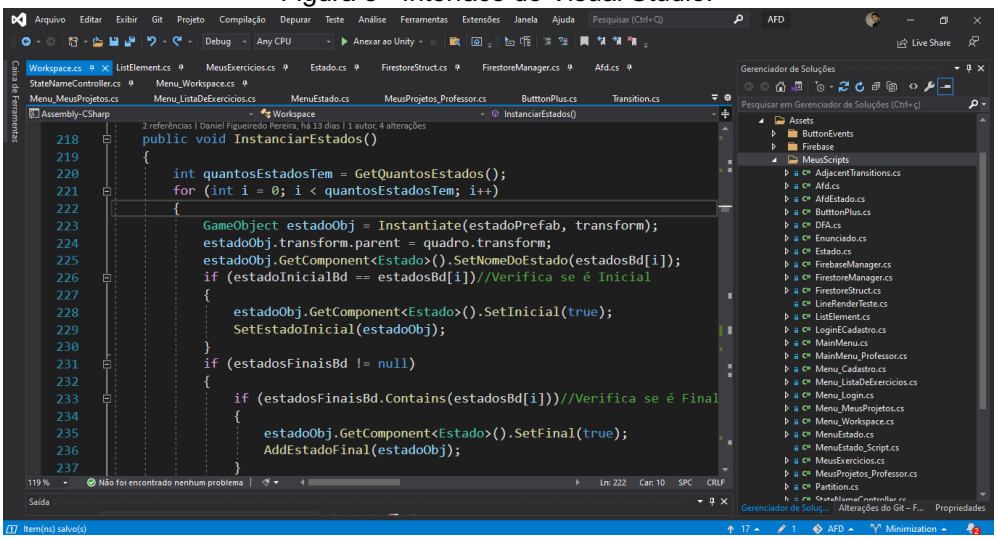


Fonte: Unity, 2022.

5.2 Visual Studio

Para a implementação do aplicativo foi utilizado a IDE Visual Studio para a escrita do código fonte do aplicativo. A ferramenta proporciona integrações com Unity facilitando a resolução de *bugs* e falhas (MICROSOFT, 2022). A Figura 10 exibe a interface da ferramenta Visual Studio.

Figura 8 - Interface do Visual Studio.

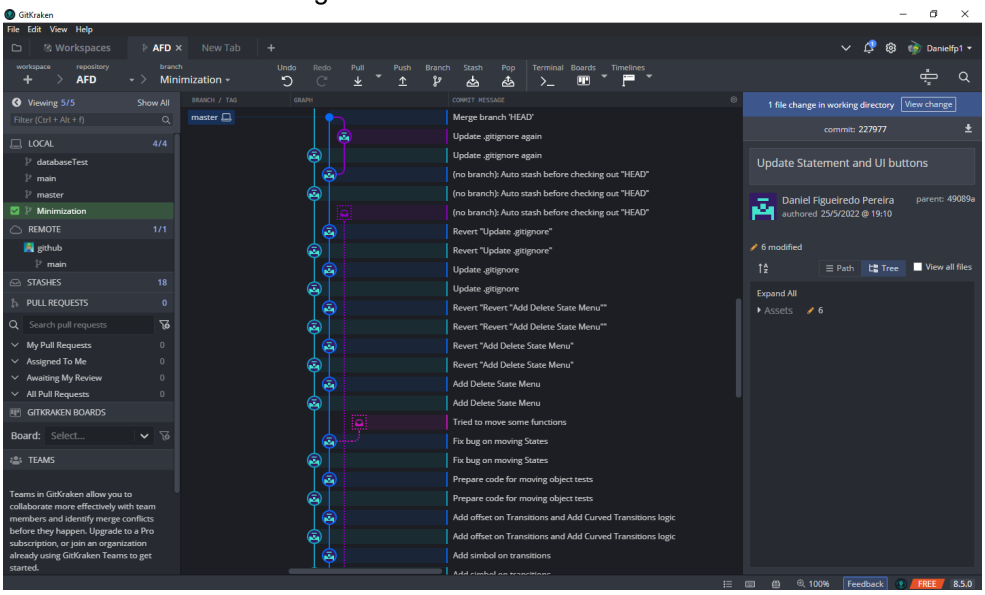


Fonte: Elaborado pelo autor.

5.3 GitKraken

GitKraken é uma ferramenta que usa o sistema de controle de versões distribuído Git, para gerenciar as alterações dos *scripts* do aplicativo, ela registra o histórico de edições dos arquivos de desenvolvimento do projeto (GITKRAKEN, 2022). Por meios gráficos é permitido facilmente fazer comparações de código e rastreabilidade das mudanças ao longo do tempo como representado na Figura 11.

Figura 9 - Interface do GitKraken.



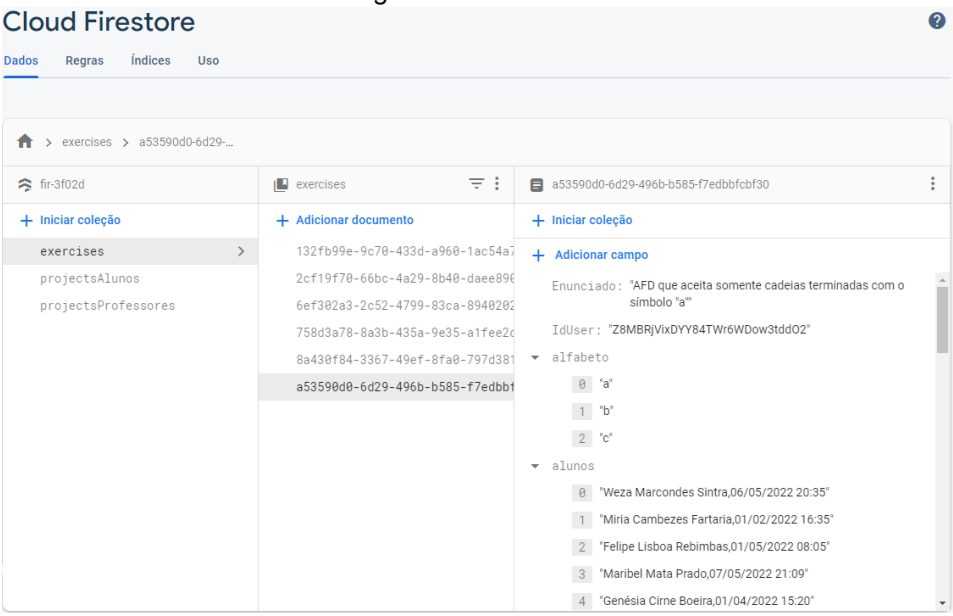
Fonte: Elaborado pelo autor.

5.4 Firebase

Firebase é uma plataforma projetada pela Google para a criação de aplicativos *mobile,* permitindo facilmente a incorporação de outros produtos da Google, como o Firebase *Authentication* para realizar cadastro e login de usuários, e o Firestore para banco de dados, o qual armazena as informações dos usuários (FIREBASE, 2022).

O sistema de armazenamento na nuvem Firestore facilita ao desenvolvedor a implementação das consultas de dados. Os dados são armazenados em uma estrutura de coleções e em documentos como apresentado na Figura 12.

Figura 10 - Firestore.



Fonte: Elaborado pelo autor.

6 IMPLEMENTAÇÃO E RESULTADOS

Este capítulo apresenta as telas do aplicativo, descrevendo as suas funções, as ações que os usuários podem realizar e demonstra como é feita a comparação dos AFDs. Mostrando o uso dos dois usuários, as telas da perspectiva do usuário professor e da perspectiva do usuário aluno. No Apêndice A apresenta um manual com instruções e modos de utilização de forma mais detalhada.

6.1 Telas do Aplicativo

Esta seção apresenta e descreve as telas presentes no aplicativo.

6.2 Tela Login

A Figura 13 apresenta a Tela de Login, onde os usuários, professor e aluno, fazem o acesso ao aplicativo. Essa tela apresenta os seguintes itens:

- E-mail: Campo onde o usuário informa o e-mail cadastrado no aplicativo;
- Senha: Campo onde o usuário informa a senha do cadastro;
- Botão Login: Botão que envia as informações dos campos para a realização da autenticação do usuário;
- Botão Cadastre-se: Botão que direciona o usuário para a Tela de Cadastro;
- Botão Esqueci minha senha: Botão que direciona o usuário para a Tela de Recuperação de Senha.

Figura 11 - Tela Login.

Fonte: Elaborado pelo autor.

6.3 Tela de Cadastro

A Figura 14 apresenta a Tela de Cadastro, onde o usuário, professor e aluno, faz o cadastro no aplicativo. Possui os seguintes itens:

- Nome: Campo onde o usuário informa o nome para o cadastro;
- E-mail: Campo onde o usuário informa o e-mail para o cadastro;
- Senha: Campo onde o usuário cria a senha para acesso ao aplicativo;
- Confirmação da Senha: Campo onde o usuário confirma a senha;
- Identificação do tipo do usuário: É uma caixa de seleção onde o usuário seleciona entre as opções “Aluno” ou “Professor”;
- Botão Voltar: Botão que direciona o usuário para a Tela Login;
- Botão Cadastrar: Botão que envia os dados dos campos para a realização do cadastro.

Figura 12 - Tela de Cadastro.

Fonte: Elaborado pelo autor.

6.4 Tela de Recuperação de Senha

A Figura 15 apresenta a Tela de Recuperação de Senha, onde os usuários, professor e aluno, fazem a recuperação de senha do cadastro. Esta tela possui os seguintes itens:

- E-mail: Campo onde o usuário informa o e-mail para a recuperação do acesso;
- Botão Voltar: Botão que direciona o usuário para a Tela Login;

- Botão Recuperar: Botão que envia a solicitação de recuperar a senha para o e-mail.

Figura 13 - Tela de Recuperação de Senha.

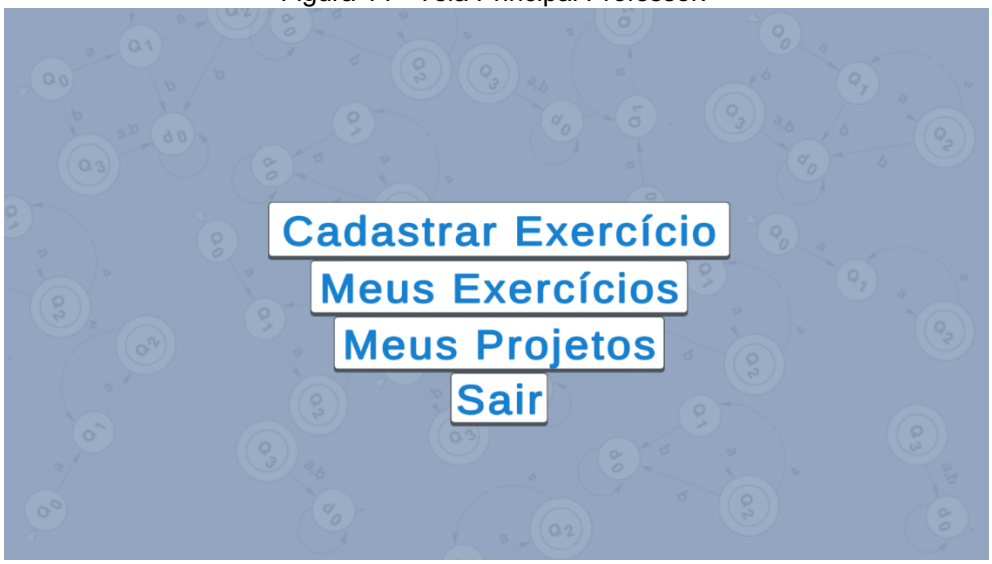
Fonte: Elaborado pelo autor.

6.5 Tela Principal Professor

A Figura 16 apresenta a Tela Principal Professor, que é exibida para o usuário professor após ter feito o login. Esta tela possui os seguintes itens:

- Botão Cadastrar Exercício: Direciona o usuário professor para a Tela *Workspace*;
- Botão Meus Exercícios: Direciona o usuário professor para a tela Meus Exercícios;
- Botão Meus Projetos: Direciona o usuário professor para a Tela Meus Projetos
- Botão Sair: Efetua o *logoff* e direciona o usuário professor para a Tela Login.

Figura 14 - Tela Principal Professor.



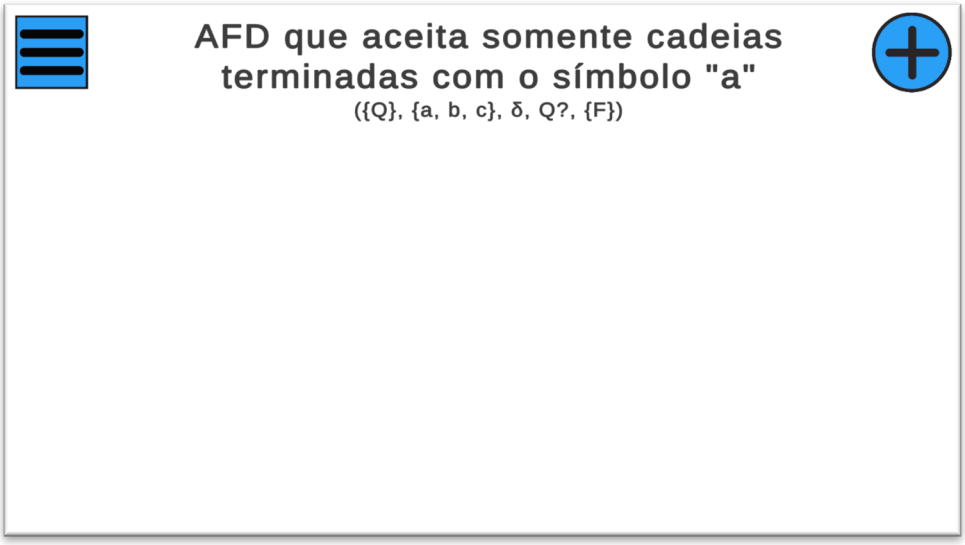
Fonte: Elaborado pelo autor.

6.6 Tela *Workspace*

A Figura 17 apresenta a Tela *Workspace*, onde AFD é construído. Esta tela é apresentada para os dois usuários, o professor e o aluno. Para o professor esta tela é usada para o registro da solução do exercício. Para o aluno, é onde ele adiciona suas respostas. Esta tela possui os seguintes itens:

- Enunciado: É exibido o enunciado descrito pelo usuário professor;
- Quíntupla: É exibida na tela de acordo com a construção do AFD;
- Botão Adição: Faz a adição de novos estados no AFD;
- Botão Opções: Abre o menu opções, detalhado da seção 6.6.2.

Figura 15 - Tela *Workspace*.



Fonte: Elaborado pelo autor.

6.6.1 Tela de Cadastro do Enunciado

A Figura 18 apresenta a Tela de Cadastro do Enunciado, tem como função realizar o cadastro do alfabeto a ser utilizado e do enunciado do exercício. Esta tela possui os seguintes itens:

- Campo do alfabeto: Permite ao usuário professor adicionar o alfabeto do exercício a ser cadastrado;
- Campo do enunciado: Permite ao usuário professor adicionar o enunciado do exercício;
- Botão Voltar: Direciona o usuário à tela do usuário principal professor;
- Botão OK: Faz o cadastro do alfabeto e do enunciado no exercício, e direciona à Tela *Workspace*;

Figura 16 - Tela de Cadastro do Enunciado.

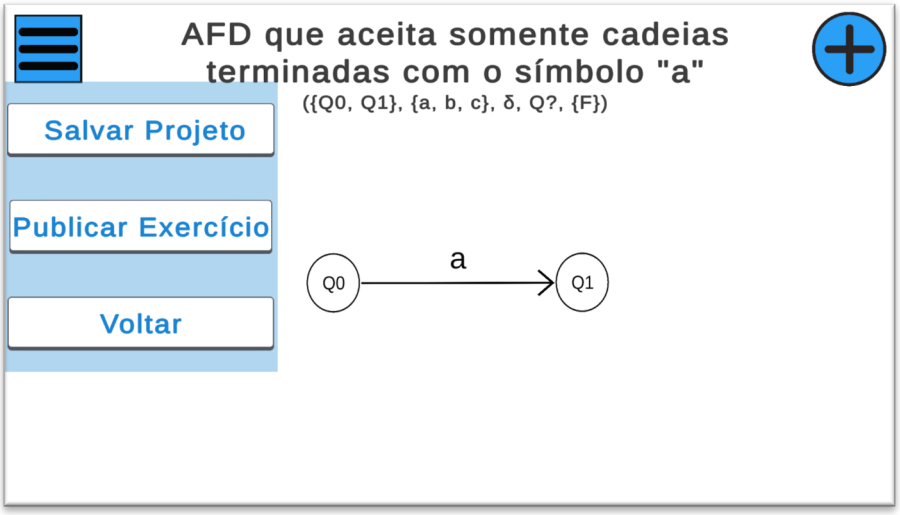
Fonte: Elaborado pelo autor.

6.6.2 Menu Opções

O Menu Opções assim como a Tela *Workspace* muda de funções de acordo com o usuário. O Menu Opções para o usuário professor apresentado na Figura 19, possui os seguintes itens:

- Botão Salvar Projeto: Salva o AFD no banco de dados;
- Botão Publicar Exercício: Verifica se a solução inserida é um AFD mínimo, se sim, o exercício é salvo no banco de dados e é publicado para que usuários alunos possam resolve-lo;
- Botão Voltar: Direciona o usuário professor para Tela Principal Professor.

Figura 17 - Tela Workspace com o Menu Opções para usuário professor.

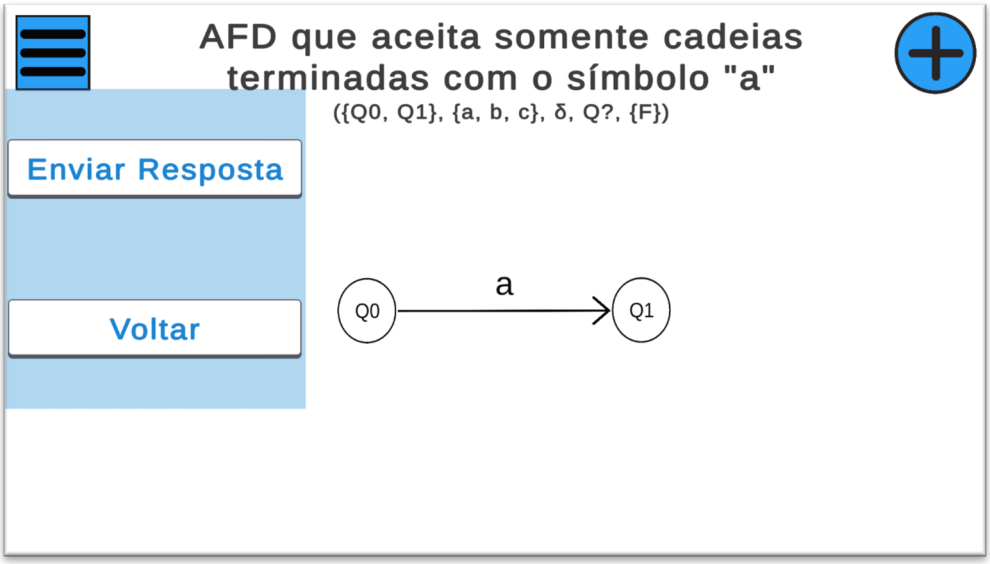


Fonte: Elaborado pelo autor.

O Menu Opções para o usuário aluno apresentado na Figura 20, possui os seguintes itens:

- Botão Enviar Resposta: Faz o processo de minimização no AFD modelado e compara com a solução cadastrada pelo usuário professor. Se a resposta for correta, é exibida a mensagem de resposta correta. Caso contrário, é exibida uma mensagem de resposta errada;
- Botão Voltar: Direciona o usuário aluno para a Tela Principal Aluno.

Figura 18 - Tela Workspace com o Menu Opções para o usuário aluno.



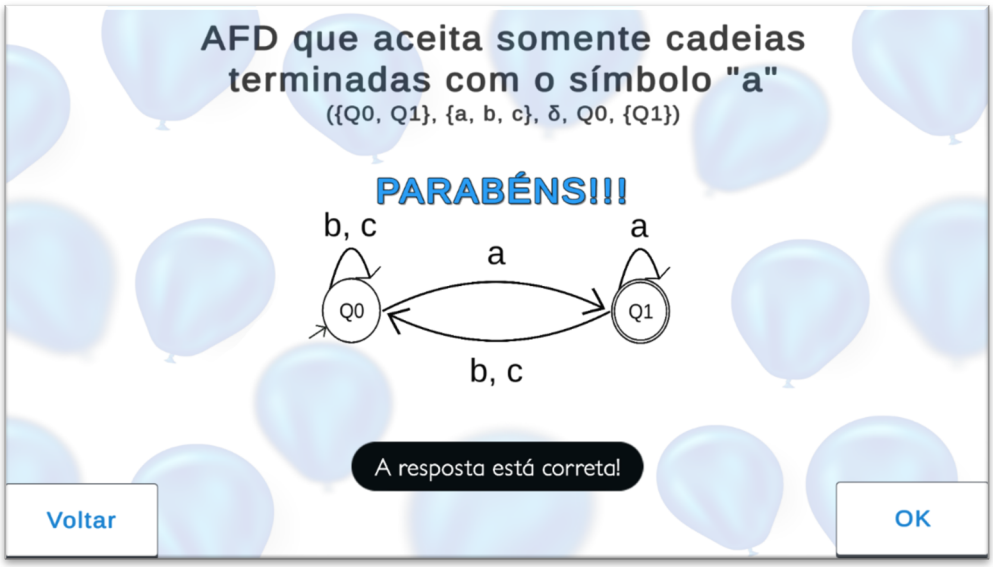
Fonte: Elaborado pelo autor.

6.6.3 Mensagem de Resposta Correta

A Figura 21 apresenta a Mensagem de Resposta Correta, é exibida quando o usuário aluno envia a resposta e o aplicativo verifica que ela é equivalente à solução correta do exercício. A tela possui os seguintes botões:

- Botão Voltar: Fecha a mensagem;
- Botão OK: Direciona o usuário aluno para a Tela de Exercícios.

Figura 19 - Workspace com a mensagem de resposta correta



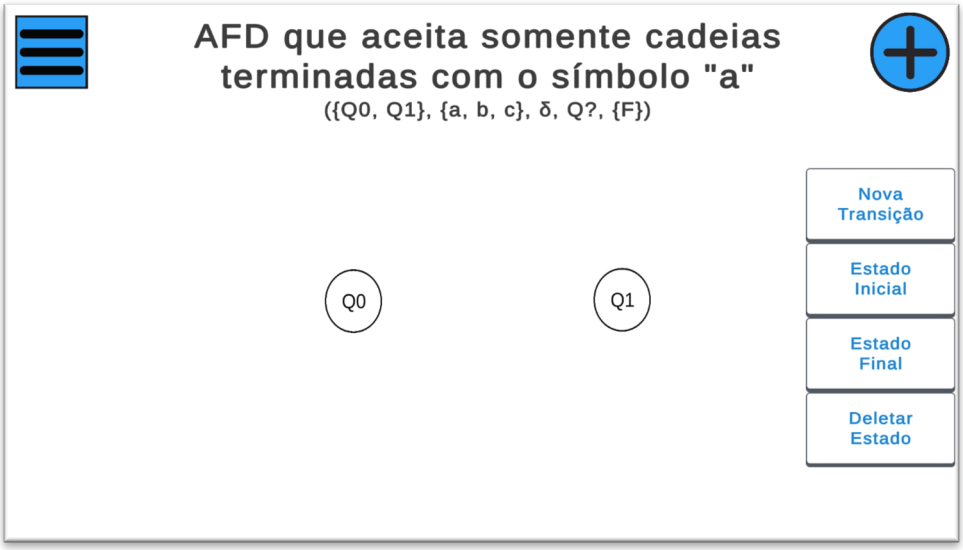
Fonte: Elaborado pelo autor.

6.6.4 Menu estado

A Figura 22 apresenta o menu estado, que pode ser aberto com um clique longo em qualquer estado adicionado no *Workspace*. Nele há todas as operações possíveis com o estado escolhido, são elas:

- Nova Transição: Abre a Tela Nova Transição, para a criação de uma transição, detalhado da seção 6.6.3;
- Estado Inicial: Se o projeto não possuir um estado inicial, torna o estado selecionado em estado inicial. Se o projeto possuir um estado inicial, a opção não é ativada;
- Estado Final: Torna o estado selecionado em estado final;
- Deletar Estado: Deleta o estado selecionado e as transições associadas.

Figura 20 - Tela *Workspace* com o Menu Estado.



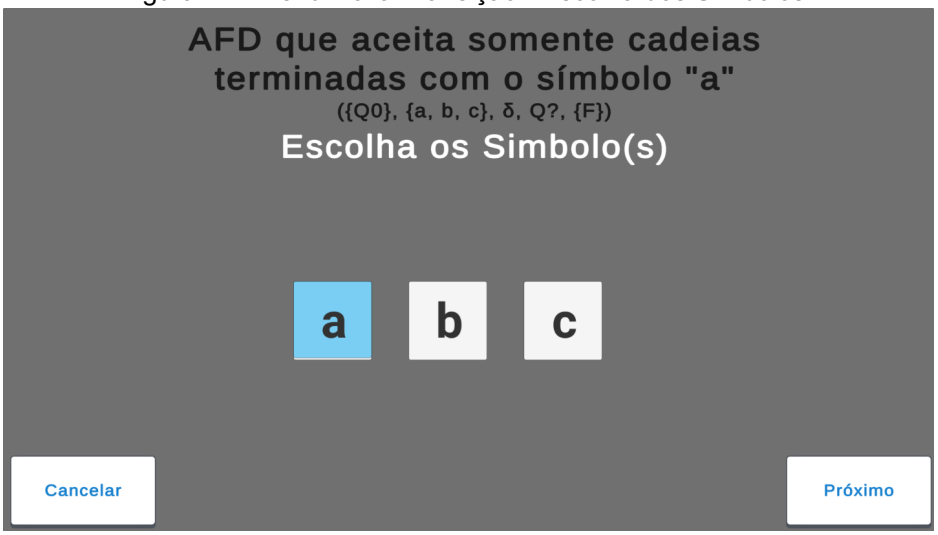
Fonte: Elaborado pelo autor.

6.6.5 Menu Nova Transição

O Menu Nova Transição possui duas interfaces, a primeira é a escolha do símbolo apresentado na Figura 23, e a segunda é a escolha do estado alvo da transição, Figura 24. Na escolha do símbolo possui os seguintes itens:

- Símbolos do alfabeto: para o usuário selecionar qual ou quais símbolos fazem parte da transição;
- Botão Cancelar: Cancela a ação e fecha o menu Nova Transição;
- Botão Próximo: Avança para a segunda interface do Menu Nova Transição.

Figura 21 - Menu Nova Transição - Escolha dos Símbolos.

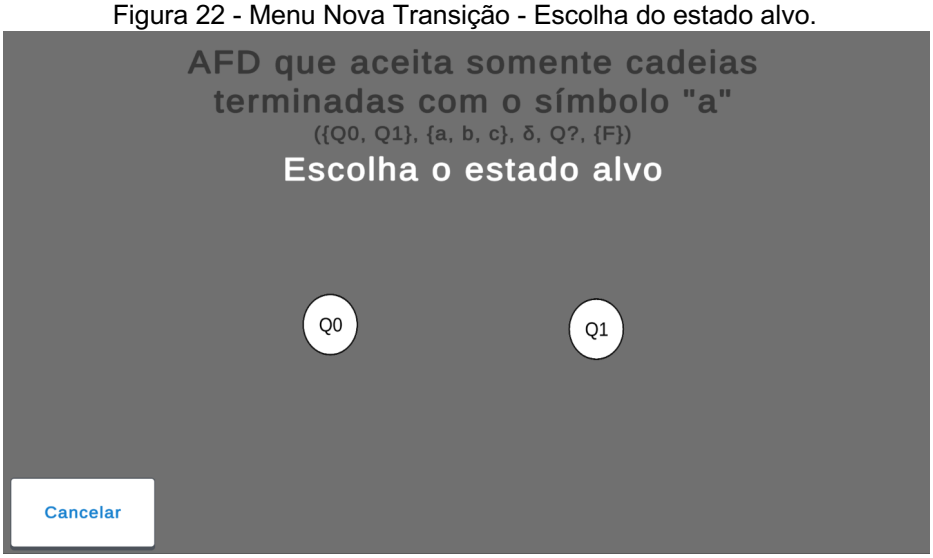


Fonte: Elaborado pelo autor.

A segunda interface do Menu Nova Transição apresentada na Figura 24 possui:

- Botão Cancelar: Cancela a ação e fecha o menu Nota Transição;
- Os estados adicionados pelo usuário: Para a escolha do alvo da transição.

Figura 22 - Menu Nova Transição - Escolha do estado alvo.



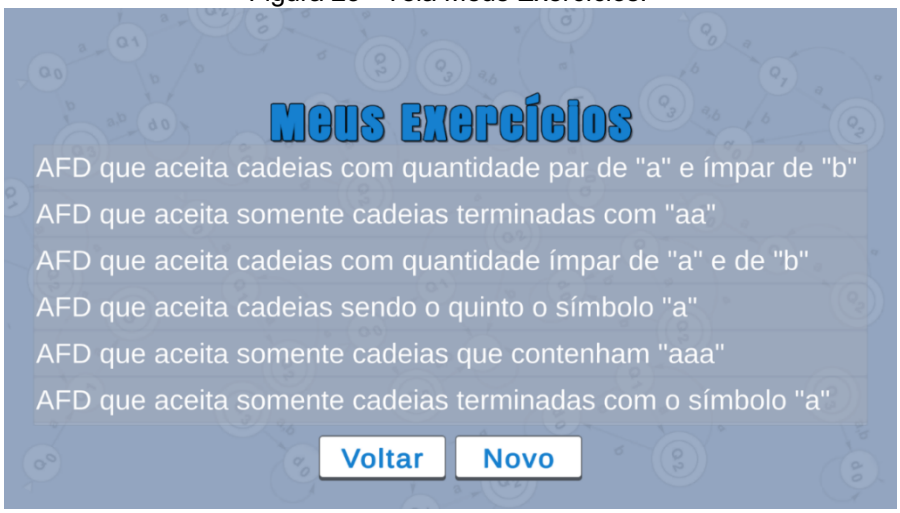
Fonte: Elaborado pelo autor.

6.7 Tela Meus Exercícios

A Figura 25 apresenta a Tela Meus Exercícios, onde é listado todos os exercícios publicados pelo usuário professor. Ao ser selecionado um exercício o usuário professor é direcionado a Tela Meus Exercícios, que possui os seguintes itens:

- Lista de Exercícios: Campo onde é exibida a lista de exercícios publicados pelo usuário professor;
- Botão Novo: Direciona o usuário professor para a tela de Workspace para o cadastro de um novo exercício;
- Botão Voltar: Direciona o usuário para a Tela Principal Professor.

Figura 23 - Tela Meus Exercícios.



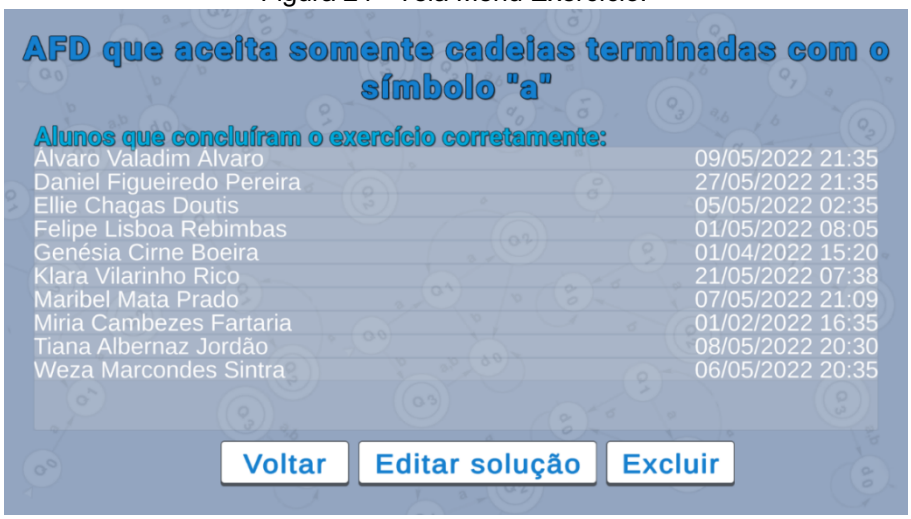
Fonte: Elaborado pelo autor.

6.8 Tela Menu Exercício

A Figura 26 apresenta a Tela Menu Exercício, onde é listado todos os alunos que concluíram o exercício selecionado. Esta tela possui os seguintes itens:

- Lista de Alunos: Campo onde é exibida a lista dos usuários alunos que concluíram corretamente o exercício selecionado, juntamente com a data e hora da conclusão;
- Botão Editar Solução: Direciona o usuário a Tela *Workspace* para edição da solução previamente cadastrada pelo usuário professor;
- Botão Excluir: Exclui o exercício publicado do banco de dados do aplicativo;
- Botão Voltar: Direciona o usuário para a Tela Meus Exercícios.

Figura 24 - Tela Menu Exercício.



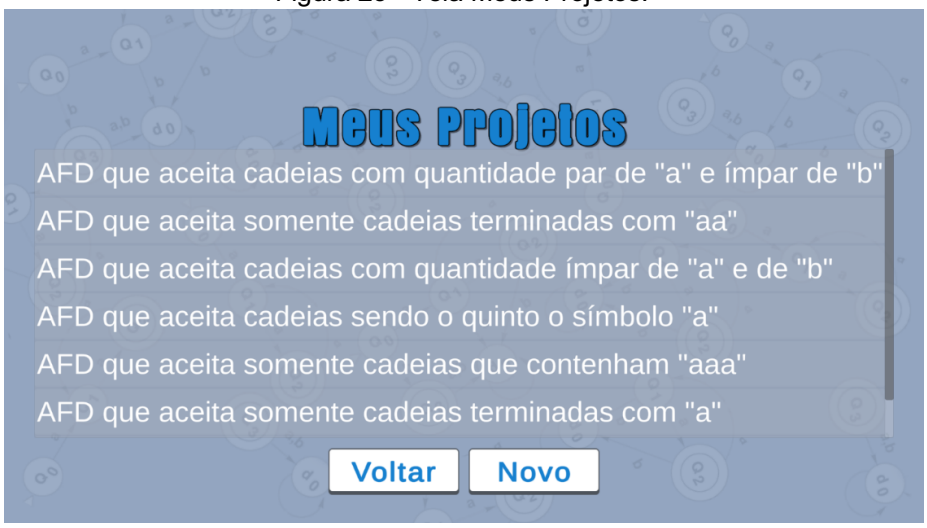
Fonte: Elaborado pelo autor.

6.9 Tela Meus Projetos

A Figura 27, apresenta a Tela Meus Projetos onde é listado todos os exercícios criados pelo usuário professor. Esta tela possui os seguintes itens:

- Lista de Projetos: Campo onde é exibida a lista de projetos criados pelo usuário professor;
- Botão Novo: Redireciona o usuário professor para a tela de Workspace para a criação de um novo projeto;
- Botão Voltar: Direciona o usuário para a Tela Principal Professor.

Figura 25 - Tela Meus Projetos.



Fonte: Elaborado pelo autor.

6.10 Tela Principal Aluno

A Figura 28 apresenta a Tela Principal Aluno, que é a tela exibida ao usuário aluno após o login. Esta tela possui os seguintes itens:

- Botão Lista de Exercícios: Direciona o usuário aluno para a Tela Lista de Exercícios;
- Botão Exercícios Resolvidos: Direciona o usuário aluno para a Tela Lista de Exercícios Resolvidos;
- Botão Sair: Direciona o usuário aluno para a Tela Login.

Figura 26 - Tela Principal Aluno.

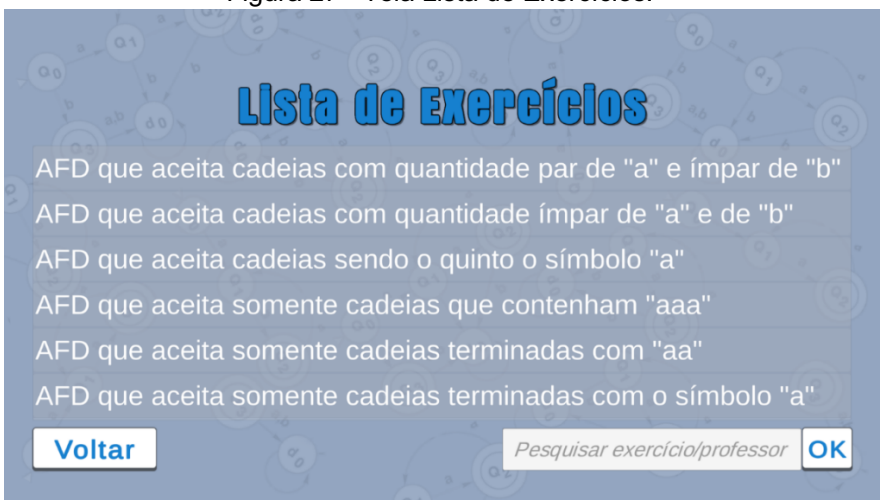
Fonte: Elaborado pelo autor.

6.11 Tela Lista de Exercícios

A Figura 29 apresenta a Tela Lista de Exercícios, onde é listada todos os exercícios publicados no aplicativo. Nela o usuário aluno pode pesquisar pelo enunciado do exercício e pelo professor que publicou o exercício. Esta tela possui os seguintes itens:

- Lista de Exercícios: Campo onde é exibida a lista de exercício publicados;
- Campo de Pesquisa: Permite ao usuário aluno inserir o enunciado ou o nome do professor para realizar a pesquisa;
- Botão OK: Botão que confirma a realização da pesquisa;
- Botão Voltar: Botão que direciona o usuário à Tela Principal Aluno.

Figura 27 - Tela Lista de Exercícios.



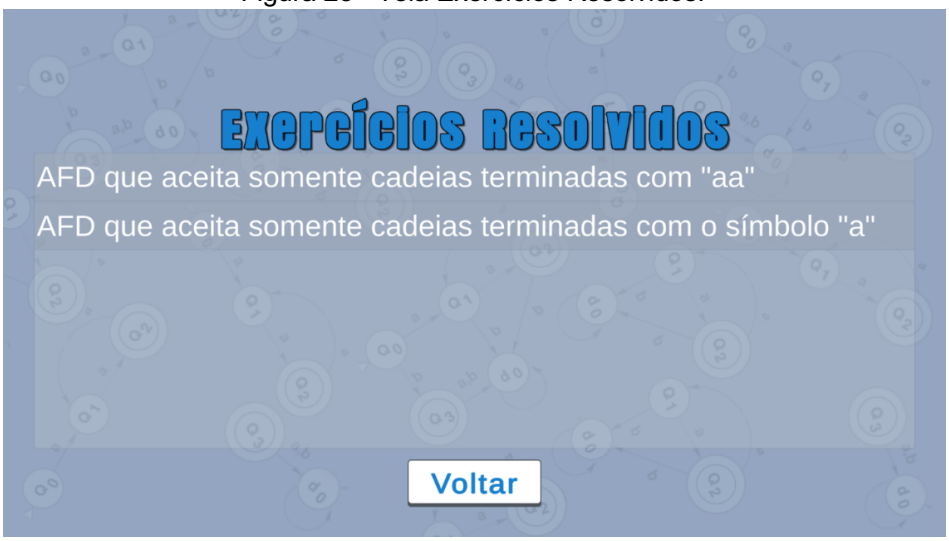
Fonte: Elaborado pelo autor.

6.12 Tela Exercícios Resolvidos

A Figura 30 apresenta a Tela Exercícios Resolvidos, nesta tela é listada todos os exercícios que o aluno resolveu. Nela o usuário aluno pode acessar as respostas enviadas do seu usuário. Esta tela possui os seguintes itens:

- Lista de Exercícios Resolvidos: Campo onde é exibida a lista de exercício resolvidos;
- Botão Voltar: Direciona o usuário à Tela Principal Aluno.

Figura 28 - Tela Exercícios Resolvidos.



Fonte: Elaborado pelo autor.

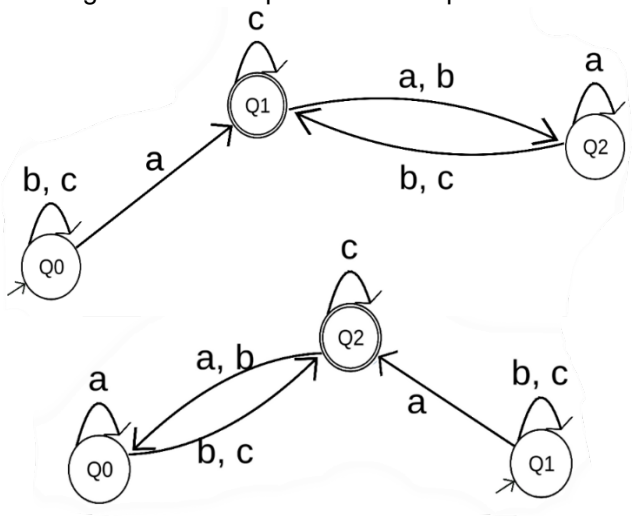
6.13 Comparação entre AFDs

Para ser possível a comparação dos AFDs equivalentes é necessário fazer o processo de minimização. No processo de minimização, foi utilizado o algoritmo de Brzozowski que faz a remoção os estados inacessíveis e redundantes.

Como a forma do AFD minimizado é único, diferenciando apenas das identificações dos estados, é preciso renomear cada estado para um padrão, sendo assim possível comparar os 2 AFDs mínimos. Foi adotada uma forma que se baseia na distância do estado em questão, ao estado inicial e o símbolo da transição recebida por ele.

Na Figura 29 apresentam dois AFDs mínimos e equivalentes, porém, todos os estados possuem nomes diferentes.

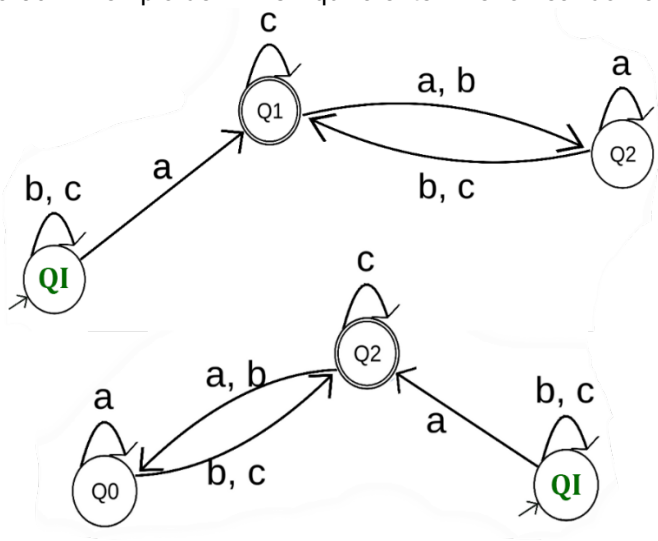
Figura 29 - Exemplo de AFDs Equivalente.



Fonte: Elaborado pelo autor.

Para resolução desse problema foi implementado o algoritmo para renomear em uma forma padronizada. Primeiramente é renomeado o estado inicial para “QI”, como apresenta a Figura 30.

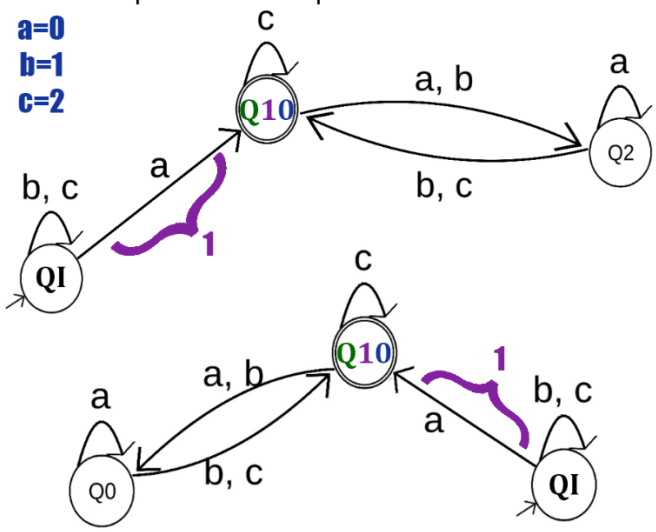
Figura 30 - Exemplo de AFDs Equivalente - Renomeando Passo I.



Fonte: Elaborado pelo autor.

Em seguida é buscado o estado destino da próxima transição seguindo o índice do alfabeto. Por exemplo, na Figura 31 os AFDs da comparação possuem o alfabeto com os símbolos: “a”, “b” e “c”, portanto seus índices são respectivamente “0”, “1” e “2”. Assim encontrado o estado, seu novo nome é atribuído baseando-se na sua distância ao estado inicial e no índice da transição. “Q” seguido da distância: “1” e do índice “0”, resultando em “Q10”

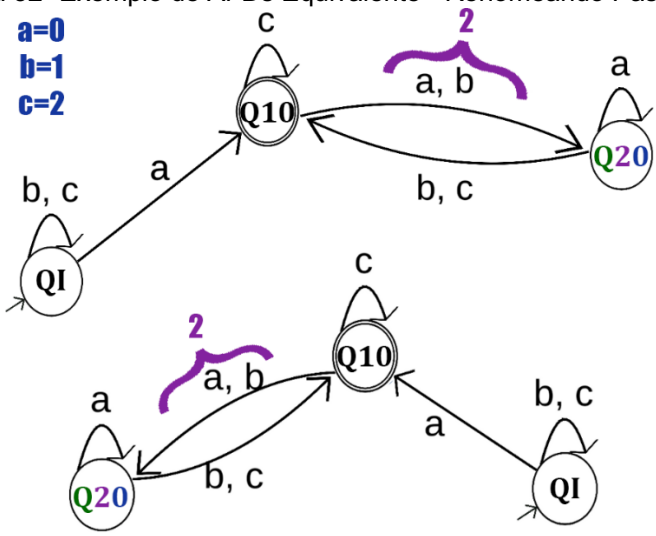
Figura 31 - Exemplo de AFDs Equivalente - Renomeando Passo II.



Fonte: Elaborado pelo autor.

Esse processo é repetido até que todos os estados sejam renomeados, resultando na Figura 32 com a renomeação do último estado para “Q20”.

Figura 32- Exemplo de AFDs Equivalente - Renomeando Passo III.



Fonte: Elaborado pelo autor.

Após renomeados os AFDs são representados no código por uma série de strings:

- Cabeçario;
- Lista de transições;
- Lista dos estados finais.

A string cabeçario é composta por: número de estados, número de transições, nome do estado inicial e o número de estados finais. A Tabela 2 representa a string cabeçario do exemplo dado da Figura 32.

Tabela 2 - Representação da string cabeçario

Nº de estados	Nº de transições	Estado inicial	Nº de estados finais
3	9	QI	1

Fonte: Elaborado pelo autor.

A lista de transições é composta por strings contendo: nome do estado de origem, símbolo da transição e o nome do estado destino. A Tabela 3 representa a lista de transições do exemplo dado da Figura 32.

Tabela 3 - Lista de transições.

Estado de origem	Símbolo da transição	Estado de destino
QI	a	Q10
QI	b	QI
QI	c	QI
Q10	a	Q20
Q10	b	Q20
Q10	c	Q10
Q20	a	Q20
Q20	b	Q10
Q20	c	Q10

Fonte: Elaborado pelo autor.

Os estados finais do AFD são representados por uma lista de strings com os nomes dos estados finais. A Tabela 4 representa a lista de estados finais dos AFDs do exemplo dado da Figura 32.

Tabela 4 - Lista de estados finais

Estados Finais
Q10

Fonte: Elaborado pelo autor.

Fazendo a comparação dessas listas e verificando se todos os itens estão presentes indiferente da posição, é possível comprovar que os dois AFDs são iguais.

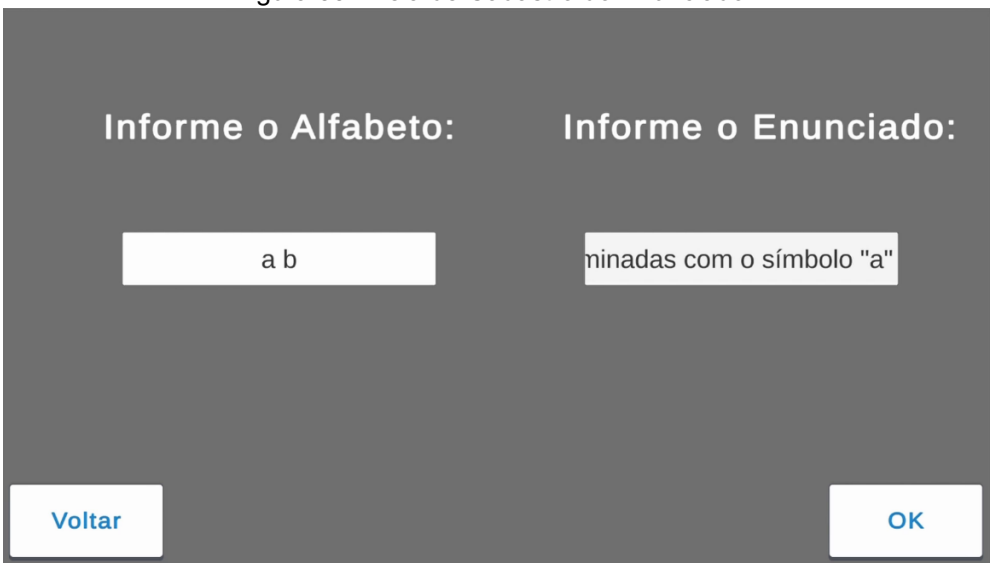
6.14 Cadastro de Exercício e Envio de Resposta

Essa seção mostra o fluxo de cadastro de exercício feito pelo usuário professor e o fluxo de envio de resposta feito pelo usuário aluno, utilizando o mesmo exercício como exemplo.

6.14.1 Cadastro de Exercício

Para o cadastro do exercício o usuário professor deve informar o alfabeto do AFD e o enunciado como demostra a Figura 33.

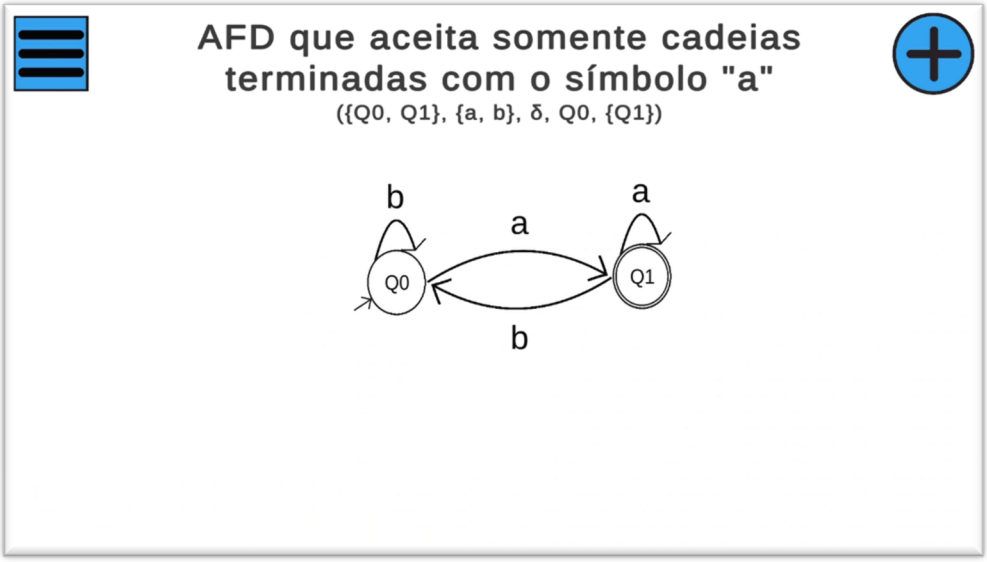
Figura 33 - Tela de Cadastro do Enunciado.



Fonte: Elaborado pelo autor.

Em seguida, o usuário professor tem acesso a área *Workspace* onde ele constrói o AFD de solução do exercício a ser cadastrado, como demonstrado na Figura 34. No exemplo dado o enunciado do exercício é: AFD que aceita somente cadeias terminadas com o símbolo "a". E o alfabeto do AFD é “a” e “b”.

Figura 34 - Tela *Workspace* com AFD de solução.



Fonte: Elaborado pelo autor.

Para o registro da solução do exercício, o AFD construído pelo usuário professor passa pelas seguintes verificações:

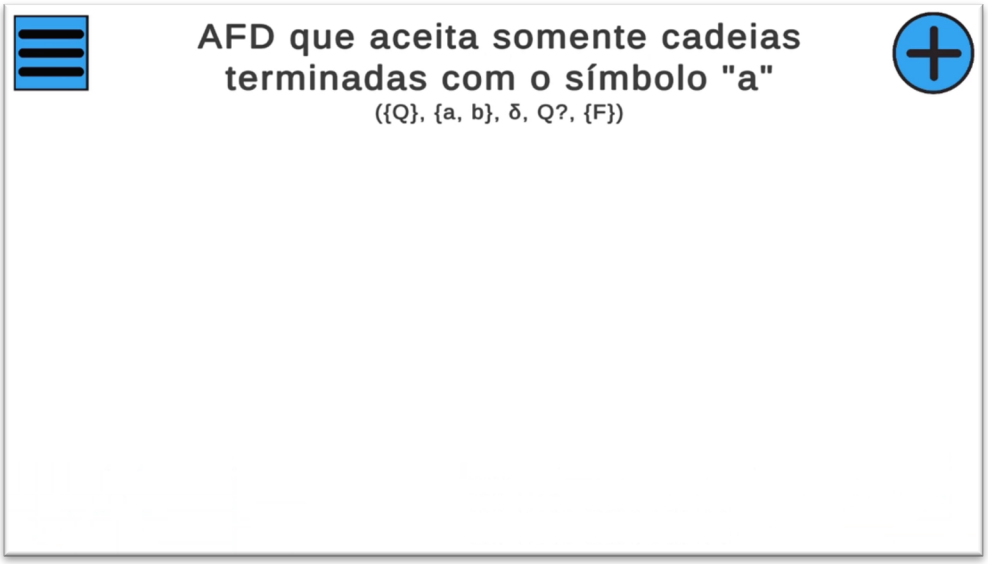
- Verifica se é um AFD;
  - Se possui estado Inicial;
  - Se possui pelo menos um estado final;
  - Se todos os estados possuem transições com todos os símbolos.
- Verifica se é mínimo.
  - Faz o processo de comparação discutido na seção 6.13;
  - Se o AFD mínimo for igual ao AFD construído.

Depois de executadas essas verificações, o exercício é cadastrado no aplicativo e é salvo no banco de dados.

6.14.2 Envio de Resposta

Para o envio de resposta o usuário aluno deve escolher um exercício previamente cadastrado por um usuário professor. Em seguida, é exibida a Tela de *Workspace* para ser construído o AFD de resposta pelo usuário aluno, como é apresentado a Figura 35.

Figura 35 - Tela *Workspace*.



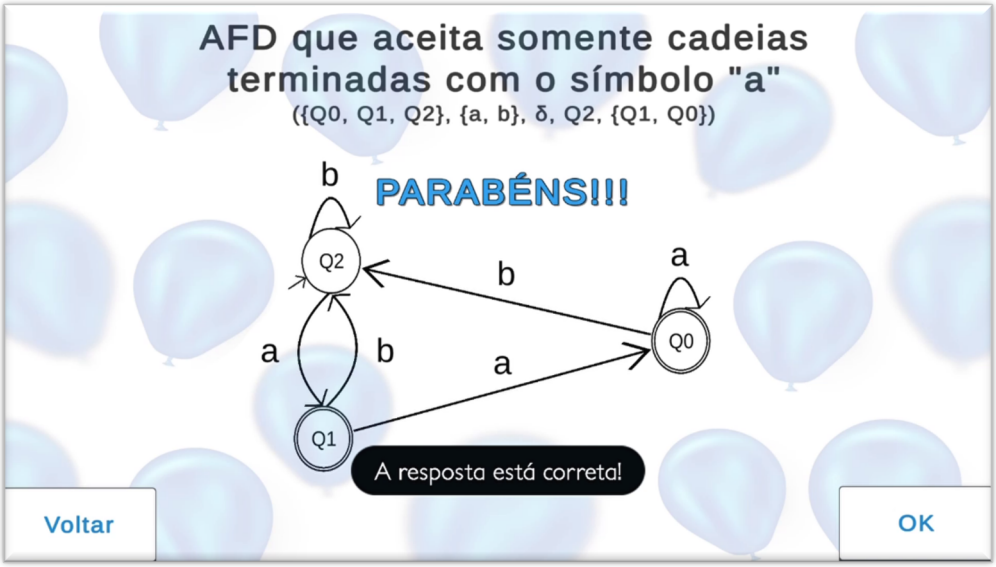
Fonte: Elaborado pelo autor.

Para o envio da resposta, o autômato construído pelo usuário aluno passa pelas seguintes verificações:

- Verifica se é um AFD;
  - Se possui estado Inicial;
  - Se possui pelo menos um estado final;
  - Se todos os estados possuem transições com todos os símbolos.

Depois de executadas essas verificações, o AFD passa pelo processo de comparação, apresentado na seção 6.13, com o AFD de solução do exercício. Se o AFD construído pelo usuário aluno for equivalente ao AFD de solução cadastrado pelo usuário professor, o exercício é concluído corretamente, como é demonstrado na Figura 36.

Figura 36 - Workspace com a mensagem de resposta correta.



Fonte: Elaborado pelo autor.

6.15 Resultados

Os resultados obtidos foram satisfatórios, o aplicativo permite que os professores criem exercícios que são armazenados no banco de dados. Os alunos podem acessar o ambiente de estudo e resolver os exercícios cadastrados no aplicativo, gerando como resultado, resposta errada ou resposta correta.

7 CONSIDERAÇÕES FINAIS

O aplicativo foi desenvolvido para dispositivos *mobiles* com o sistema operacional Android. O objetivo desse aplicativo foi contribuir para o estudo dos alunos da área da computação em relação ao conteúdo Linguagens Formais e Autômatos. Os Autômatos se mostram um conteúdo de relevância aos profissionais da computação, pois através desses se pode modelar problemas computacionais.

O trabalho apresentou a documentação utilizada no desenvolvimento do aplicativo, levantamento dos requisitos funcionais, usuários necessidade e de qualidade do aplicativo. Apresentou o modelo do diagrama de caso de uso e sua descrição, assim como ferramentas utilizadas para a criação do aplicativo.

O ambiente virtual proporciona a experiência prática do conteúdo na resolução dos exercícios. O aplicativo possibilita resolver questões sobre o conteúdo de Autômatos Finitos Determinísticos, e que professores podem cadastrar problemas para que os alunos da plataforma possam resolvê-los.

Algumas dificuldades foram encontradas no desenvolvimento do aplicativo, tais como a implementação do algoritmo de minimização na linguagem utilizada e implementações das representações gráficas do diagrama AFD na ferramenta Unity.

7.1 Trabalhos futuros

- Adicionar a função para verificar se uma cadeia de símbolos é aceita ou rejeitada pelo AFD construído;
- Adicionar a possibilidade de visualizar as transições dada uma cadeia de símbolos, por meio de uma animação.

REFERÊNCIAS

DIVEIRO, Tiarajú Asmuz; MENEZES, Paulo Baluth. **Teoria da Computação.** 5 ed. Sagra Luzzato, Porto Alegre 1999

FARIAS PINHEIRO, Álvaro. Autômatos. Revista **Fundamentos da Engenharia de Software**. Ano II – Número 5 – Recife, p. 04-28, Maio, 2017 Selo Editorial: Publicação Independente.

FIREBASE. **Cloud Firestore**. 2022a. Disponível em: <https://firebase.google.com/docs/firestore>. Acesso em: 25 de maio de 2022.

GARCÍA, Pedro; LÓPEZ, Damián; PARGA, Manuel Vázquez; **DFA minimization: from Brzozowski to Hopcroft.** Disponível em: <https://riunet.upv.es/handle/10251/27623> Acesso em: 15 de junho de 2022.

GITKRAKEN, **Branching in Git.** Disponível em: < https://www.gitkraken.com/learn/git/branch> Acesso em: 25 de maio de 2022.

HOPCROFT, Jonh E.; MOTWANI, Rajeev; ULLMAN, Jeffrey D. **Introduction to Automata Theory, Languages, and Computations**. 3. ed. Pearson, Boston, 2006.

JOSÉ NETO, João. **A Teoria da Computação e o profissional de informática.** Revista de Computação e Tecnologia da PUC-SP São Paulo, v. 1, n.1, out. 2009.

JUKEMURA, Anibal Santos. **Ginux Abstract Machine Proposta De Um Simulador De Operações Sobre Autômatos Em Ambiente Linux Para Auxílio Ao Estudo Acadêmico De Linguagens Formais E Máquinas Abstratas**. Minas Gerais 2004.

MENEZES, Paulo Baluth. **Linguagem Formais e Autômatos**. 3. Ed. Sagra Luzzatto, Porto Alegre. 2000.

MICROSOFT, **É assim que você faz o software.** 2022b. Disponível em: <https:// https://visualstudio.microsoft.com/> Acesso em: 25 de maio de 2022.

MIDENA RAMOS, Vinícius. **Ensino de linguagens formais e autômatos em cursos superiores de computação.** Revista de Computação e Tecnologia da PUC-SP São Paulo, v. 1, n.1, out. 2009.

MINISTÉRIO DA EDUCAÇÃO, **Diretrizes Curriculares Nacionais para os cursos de graduação em Computação** Brasília - DF, 2012. Disponível em: <http://portal.mec.gov.br/component/content/article?id=12991> Acesso em 16 de Agosto de 2021.

SIPSER, Michael. **Uma Introdução à Teoria da Computação**. 3. ed. PWS Publishing Co., 2005.

RAMOS, Marcus V. M**. Linguagens Formais: Exercícios e Soluções.** 1. ed. São Paulo 2021.

UNITY, **Unity 2021 LTS.** 2022b. Disponível em: <https://unity.com/pt/releases/2021-lts> Acesso em: 25 de maio de 2022.

APÊNDICE A - MANUAL DE USO DO APLICATIVO

1 CADASTRO

Para o cadastro no aplicativo deve ser preenchido na Tela de Cadastro, apresentada na Imagem 1, os campos com o nome, e-mail, senha, confirmação da senha e informar o tipo de usuário do cadastro. Após preenchidos os campos, o usuário deve selecionar a opção “Cadastrar”.

Imagem 1 - Tela de Cadastro.

Fonte: Elaborado pelo autor.

2 FAZER LOGIN

Para fazer login no aplicativo, o usuário deve estar cadastrado e preencher na Tela de Login, representada na Imagem 2, os campos com o e-mail e a senha informados no cadastro. Após isso o usuário deve selecionar a opção “Login”.

Imagem 2 - Tela Login.

Fonte: Elaborado pelo autor.

3 RECUPERAR A SENHA

Para recuperar a senha o usuário deve acessar a Tela de Recuperação de Senha, apresentada na Imagem 3, informar o e-mail do cadastro e selecionar a opção “Recuperar”. Após isso o usuário receberá um e-mail no endereço informado para a criação de uma nova senha.

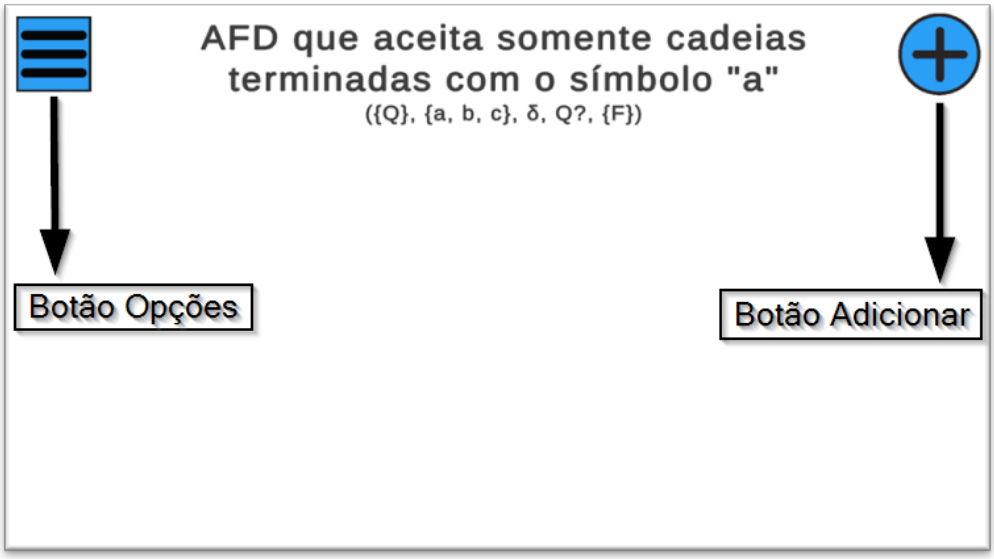
Imagem 3 - Tela de Recuperação de Senha.

Fonte: Elaborado pelo autor.

4 USO DO *WORKSPACE*

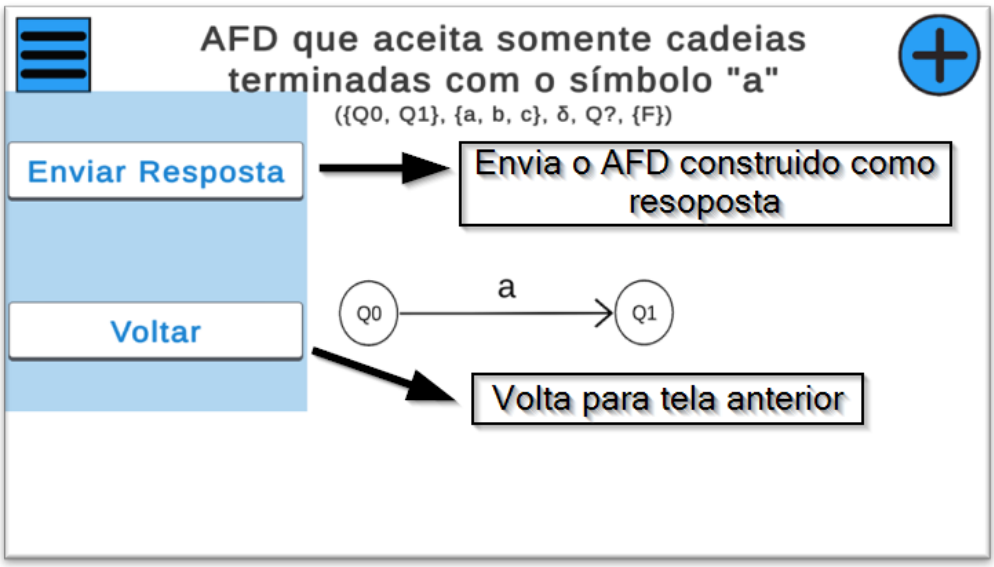
A Tela *Workspace,* apresentada na Imagem 4, é onde os usuários constroem o diagrama do AFD. Por meio do Botão Adicionar, são adicionados novos estados. Por meio do Botão Opções, abre o menu opções, apresentado nas Imagens 5 e 6.

Imagem 4 - Tela *Workspace*.



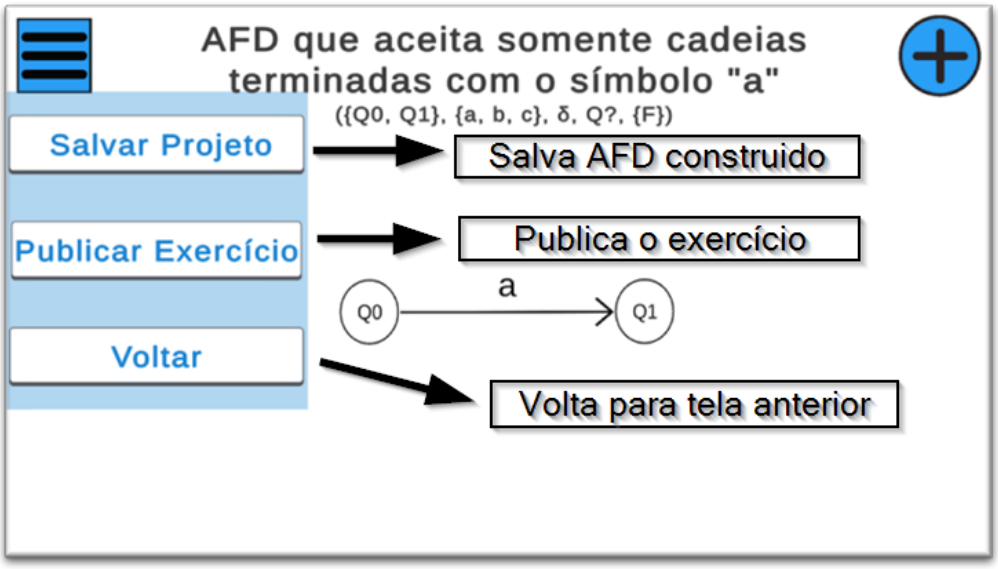
Fonte: Elaborado pelo autor.

Imagem 5 - Tela *Workspace* com o Menu Opções para usuário aluno.



Fonte: Elaborado pelo autor.

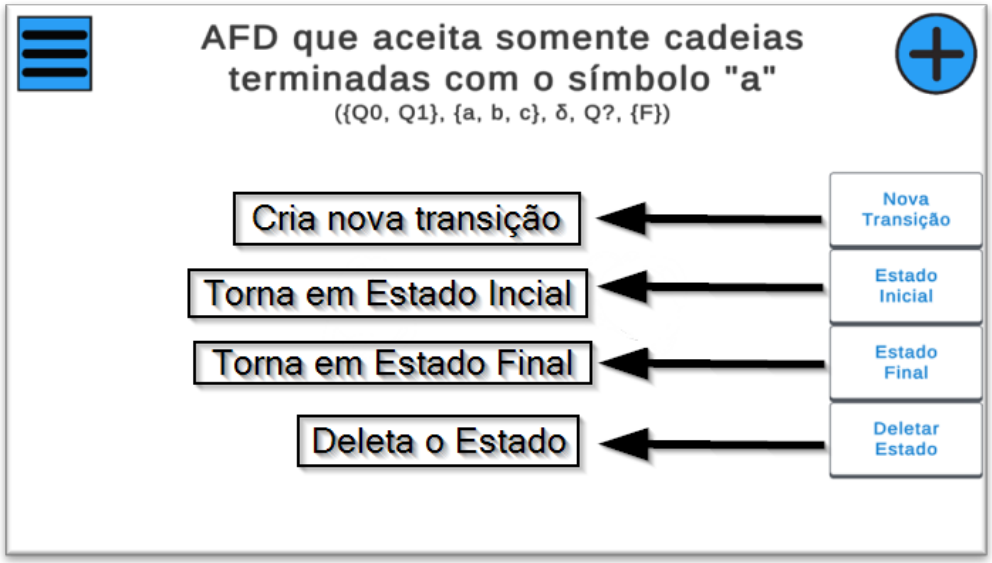
Imagem 6 - Tela *Workspace* com o Menu Opções para o usuário professor.



Fonte: Elaborado pelo autor.

Ao fazer um clique longo no estado adicionado, abre o Menu Estado, como apresentado na Imagem 7.

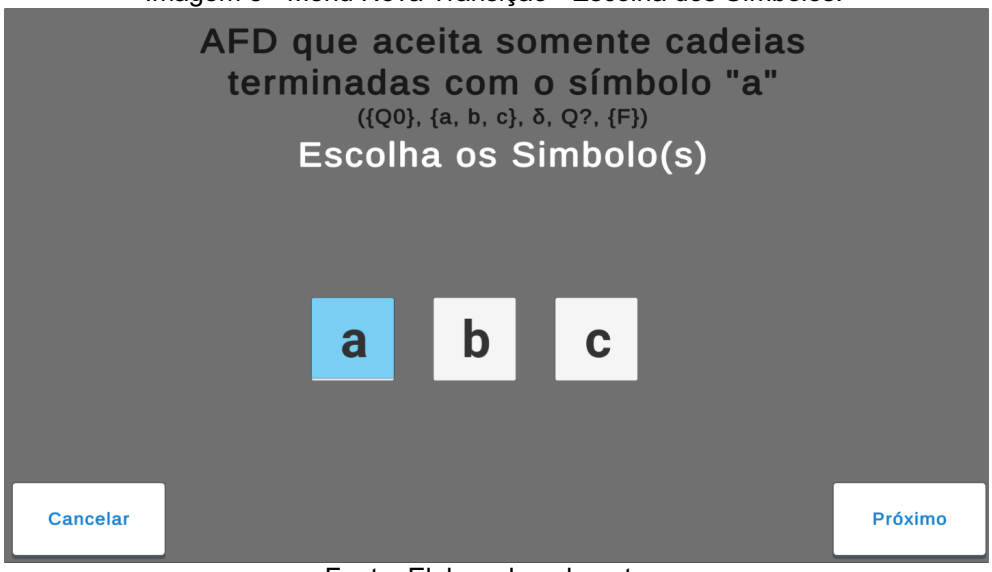
Imagem 7 - Tela *Workspace* com o Menu Estado.



Fonte: Elaborado pelo autor.

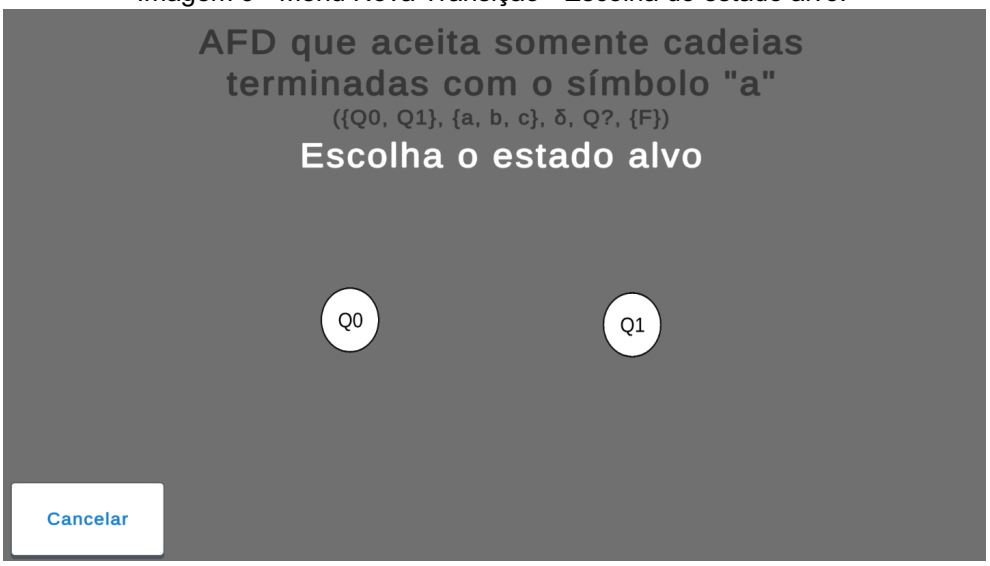
Para Adicionar uma nova transição o usuário deve escolher a opção “Nova Transição” no Menu Estado, após isso é preciso escolher o simbolo da transição e por último o estado alvo da transição, como representados nas Imagens 8 e 9.

Imagem 8 - Menu Nova Transição - Escolha dos Símbolos.



Fonte: Elaborado pelo autor.

Imagem 9 - Menu Nova Transição - Escolha do estado alvo.

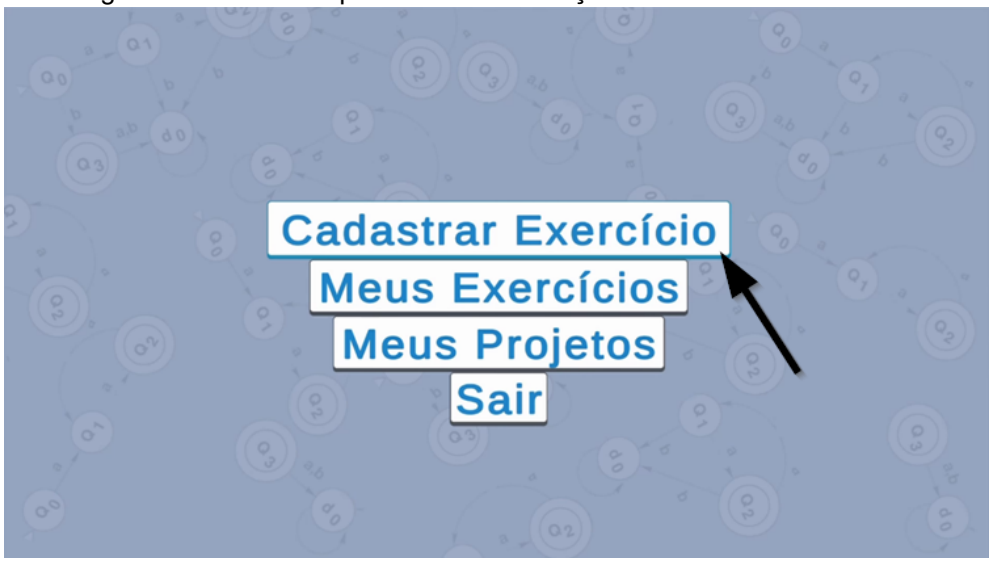


Fonte: Elaborado pelo autor.

5 CADASTRO DE EXERCÍCIO

Para cadastro de exercício o usuario professor deve acessar a opção “Cadastrar Exercício” na Tela Principal do Professor, como representado na imagem 10.

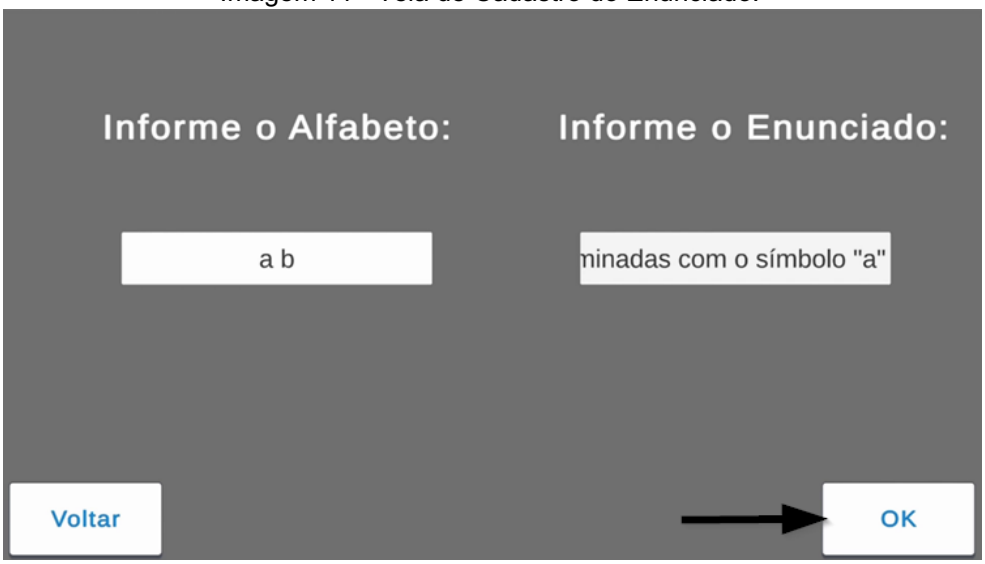
Imagem 10 - Tela Principal Professor - Seleção de Cadastro de Exercício.



Fonte: Elaborado pelo autor.

Em seguida deve ser preenchido os campos do alfabeto e do enunciado e selecionar a opção “OK”, como mostra a Imagem 11.

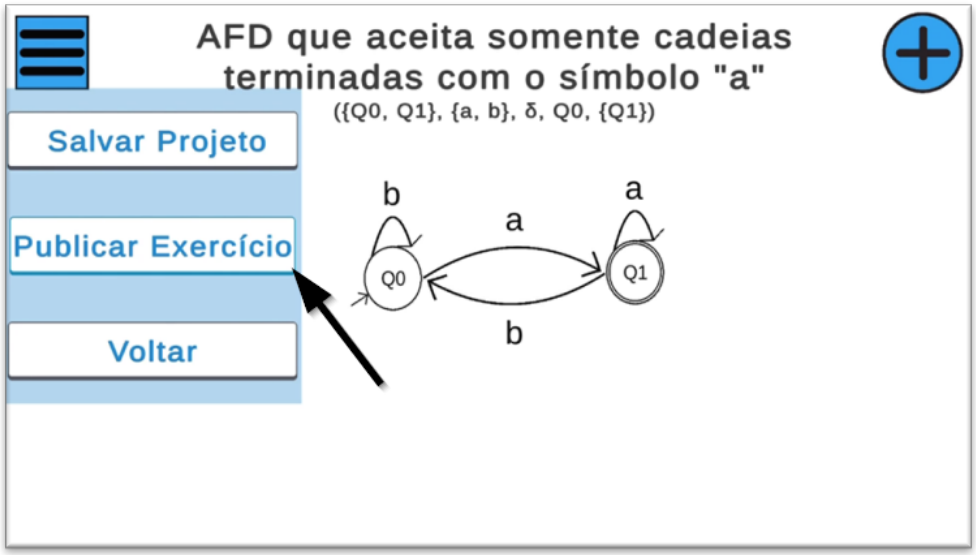
Imagem 11 - Tela de Cadastro do Enunciado.



Fonte: Elaborado pelo autor.

O AFD solução do exercício deve ser modelado em sua forma minimizada e depois disso, deve ser selecionado a opção “Publicar o Exercício” no menu opções, como demonstra a Imagem 12.

Imagem 12 - Tela Workspace com o Menu Opções para usuário professor.



Fonte: Elaborado pelo autor.

6 RESOLUÇÃO DE EXERCÍCIO

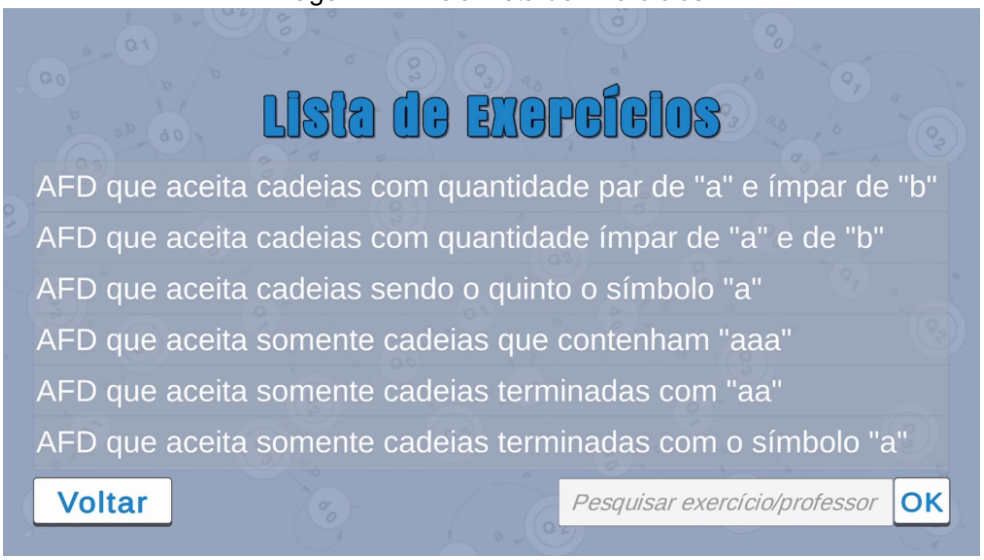
Para a resolução do exercício o usuário aluno deve primeiramente listar os exercícios cadastrados no aplicativo, para isso ele deve selecionar a opção “Lista de Exercícios” na Tela Principal Aluno. Assim como demonstrado na Imagem 13.

Imagem 13 - Tela Principal Aluno - Seleção da Lista de Exercícios.

Fonte: Elaborado pelo autor.

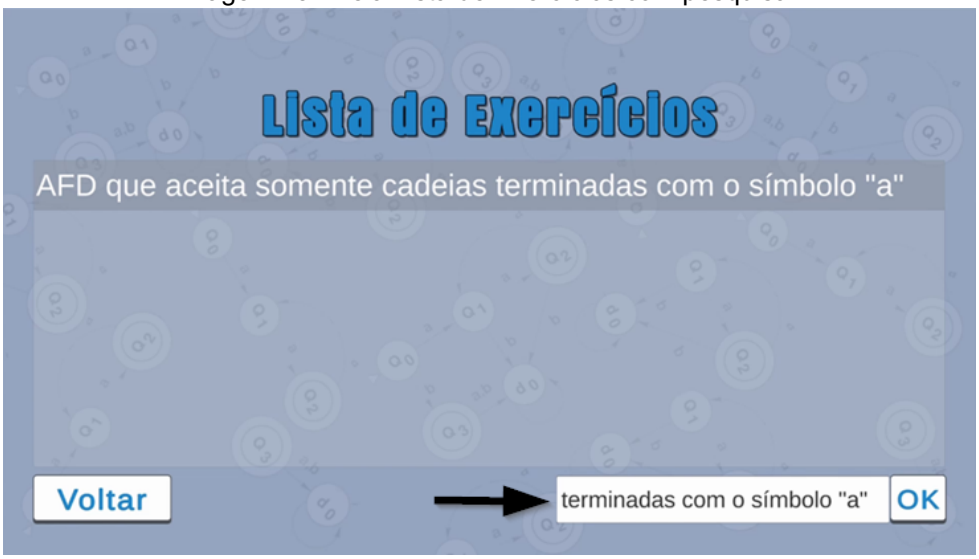
Na tela Lista de Exercícios, como apresenta a imagem 14, o usuário aluno deve selecionar um exercício para ser resolvido, podendo fazer uma busca pelo exercício desejado, como representado na Imagem 15.

Imagem 14 - Tela Lista de Exercícios.



Fonte: Elaborado pelo autor.

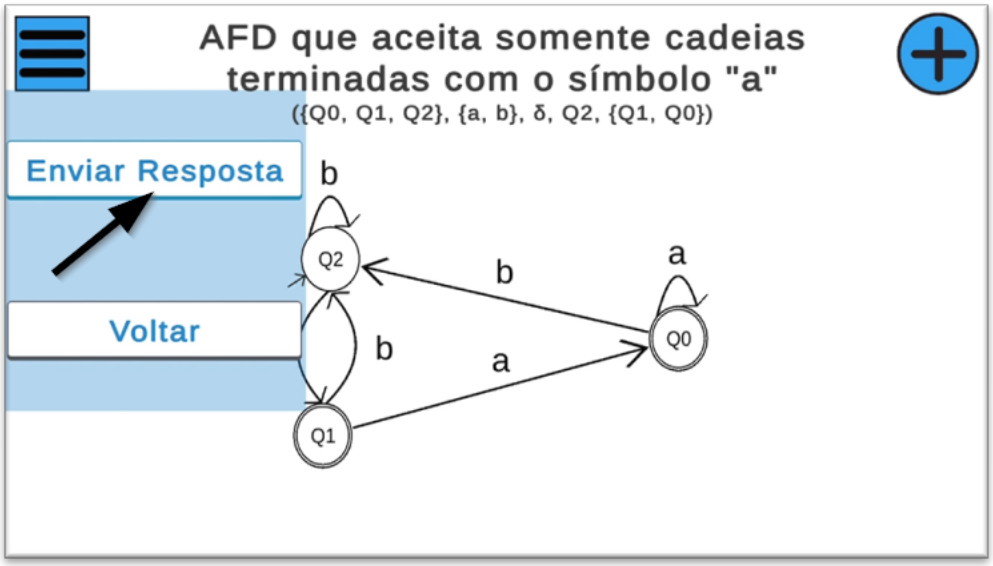
Imagem 15 - Tela Lista de Exercícios com pesquisa.



Fonte: Elaborado pelo autor.

Depois de construído o diagrama do AFD o usuário aluno pode enviar a resposta a partir da opção “Enviar Resposta” no Menu Opções, como apresentado na imagem 16.

Imagem 16 - Tela Workspace com o Menu Opções para o usuário aluno.



Fonte: Elaborado pelo autor.

APÊNDICE B - TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA

PUC GOIÁS

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 • Setor Universitário
Caixa Postal 86 • CEP 74605-010
Goiânia • Goiás • Brasil
Fone: (62) 3946 1000
www.pucgoias.edu.br • reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Daniel Figueredo Pereira
do Curso de Engenharia de Computação, matrícula 2014100330465,
telefone: 62 98602-6410 e-mail danilfpc@hotmail.com na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Um aplicativo para o ensino de Autômatos Finitos Determinísticos, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 17 de Março de 2022.

Assinatura do(s) autor(es): Daniel Figueredo Pereira

Nome completo do autor: Daniel Figueredo Pereira

Assinatura do professor-orientador: Ludmilla R. P. dos Santos

Nome completo do professor-orientador: Ludmilla Reis Pinheiro dos Santos