

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



**INTERFACE WEB PARA APOIO AO LETRAMENTO DE PESSOAS COM
SÍNDROME DE DOWN**

LEONARDO DOS SANTOS PRADO

GOIÂNIA

2021

LEONARDO DOS SANTOS PRADO

**INTERFACE WEB PARA APOIO AO LETRAMENTO DE PESSOAS COM
SÍNDROME DE DOWN**

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

Orientador(a): Olegário Correa da Silva Neto

GOIÂNIA

2021

LEONARDO DOS SANTOS PRADO

**INTERFACE WEB PARA APOIO AO LETRAMENTO DE PESSOAS COM
SÍNDROME DE DOWN**

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

Prof. Me. Ludmilla Reis Pinheiro dos Santos

Coordenador(a) de Trabalho de Conclusão de Curso

Banca examinadora:

Orientador: Olegário Correa da Silva Neto

Prof. MSc. Daniel Corrêa da Silva

Prof. Dr. Talles Marcelo G. de A. Barbosa

GOIÂNIA

2021

RESUMO

Uma pessoa com Síndrome de Down tem 47 cromossomos ao invés de 46 como na maioria da população. A síndrome de Down causa algumas alterações que afetam o desenvolvimento cognitivo e corporal de uma pessoa causando comprometimento intelectual. Atualmente saber ler é uma necessidade para qualquer ser humano. Este trabalho propõe o desenvolvimento de uma interface web para auxílio no letramento de pessoas com síndrome de Down através de atividades elaboradas por monitores.

Palavras-chave: Síndrome de Down. Interface Humano Computador. Interface Web. Letramento.

ABSTRACT

Keywords: Down. Interface. Web.

SUMÁRIO

1. INTRODUÇÃO	8
2. REVISÃO BIBLIOGRÁFICA.....	8
3. ESPECIFICAÇÃO DE REQUISITOS	9
3.1. Módulo Monitor	10
3.1.1 Manter Monitor (RF01).....	12
3.1.1.1 Cadastrar Monitor (RF01.1)	12
3.1.1.2 Consultar Monitor (RF01.2).....	13
3.1.1.3 Alterar Monitor (RF01.3)	13
3.1.1.4 Apagar Monitor (RF01.4).....	13
3.1.2 Manter Aprendiz (RF04).....	14
3.1.2.1 Cadastrar Aprendiz (RF04.1)	14
3.1.2.2 Consultar Monitor (RF04.2).....	14
3.1.2.3 Alterar Monitor (RF04.3)	15
3.1.2.4 Apagar Monitor (RF04.4).....	15
3.1.3 Manter Atividades (RF05).....	16
3.1.3.1 Elaborar Atividade (RF05.1)	16
3.1.3.2 Excluir Atividade (RF05.2)	16
3.1.3.3 Alterar Atividade (RF05.3).....	17
3.1.3.4 Relatório de Atividade (RF05.4)	17
3.1.4 Manter Programação de atividades	17
3.1.4.1 Atribuir Atividade (RF06)	18
3.2. Módulo Aprendiz.....	18
4. PROJETO	19
4.1. Modelo Conceitual	19
4.2. Modelo de Classe de Análise.....	20
4.3. Modelo de Interação	21
4.3.1. Módulo Monitor.....	22
4.3.1.1. Cadastro de Monitor	22
4.3.1.2. Alterar Monitor.....	22
4.3.1.3 Apagar Monitor	23
4.3.1.4. Elaborar Atividade.....	23
4.3.1.5. Alterar Atividade	24

4.3.1.6. Excluir Atividade.....	25
4.3.1.7. Relatório de Atividade.....	25
4.3.1.8. Cadastrar Aprendiz.....	26
4.3.1.9. Alterar Aprendiz.....	26
4.3.1.10. Apagar Aprendiz.....	27
4.3.2.11. Realizar Atividade.....	28
4.4. Modelo de Arquitetura.....	28
6. IMPLANTAÇÃO.....	29
7. RESULTADOS OBTIDOS.....	31
7.1. Testes com aprendizes.....	33
8 CONCLUSÃO.....	35
REFERÊNCIAS.....	36
APÊNDICE.....	37

1. INTRODUÇÃO

A síndrome de Down (SD) ou Trissomia 21 é uma alteração genética que ocorre na formação do embrião humano. As pessoas com síndrome de Down têm 47 cromossomos em vez de 46, como na maioria da população. A síndrome de Down causa alterações que afetam o desenvolvimento cognitivo e corporal de uma pessoa causando comprometimento intelectual.

Atualmente saber ler é uma necessidade para qualquer ser humano, pois o mesmo vive em uma sociedade em que a maioria são pessoas letradas e a língua escrita e falada é o maior meio de comunicação (ALMEIDA JÚNIOR, 2017).

Com o objetivo de incluir pessoas com SD, aqui denominados aprendizes, foi iniciado o desenvolvimento de uma interface web para apoio ao letramento. Essa interface conta com dois módulos, um direcionado aos aprendizes, e outro aos monitores, que são pessoas que vão elaborar atividades de letramento. Por meio de um navegador web, o monitor poderá elaborar uma atividade, propô-la ao aprendiz e monitorar a execução da mesma.

2. REVISÃO BIBLIOGRÁFICA

Ao longo dos últimos 10 anos, vários projetos foram iniciados com o objetivo de desenvolver aplicativos para auxiliar no letramento e alfabetização de pessoas com síndrome de Down. Em destaque o Lori's Help (VINICIUS, 2017), o ABCÊ Bulir (Rodrigues, 2014), e o Meu ABC Down (ALMEIDA JÚNIOR, 2017).

O Lori's Help propõe um aplicativo para android que auxilia pessoas com síndrome de Down no processo de alfabetização por meio de atividades, por exemplo, pareamento de vogais. Durante a execução das atividades o aplicativo monitora a atenção demonstrada pelos usuários por meio de um algoritmo de biofeedback chamado Attention Meter, e no final é gerado um relatório de desempenho para cada usuário com marcadores que exhibe o comportamento durante a execução de cada atividade.

O Meu ABC Down é um aplicativo para android voltado para alfabetização e letramento de crianças com Síndrome de Down. O aplicativo permite que a criança adicione mídias de

sons, imagens e textos, onde o mesmo disponibiliza dois jogos, o Jogo da Memória e o Acerte a Imagem, onde os jogos são trabalhados com as mídias adicionadas pela própria criança.

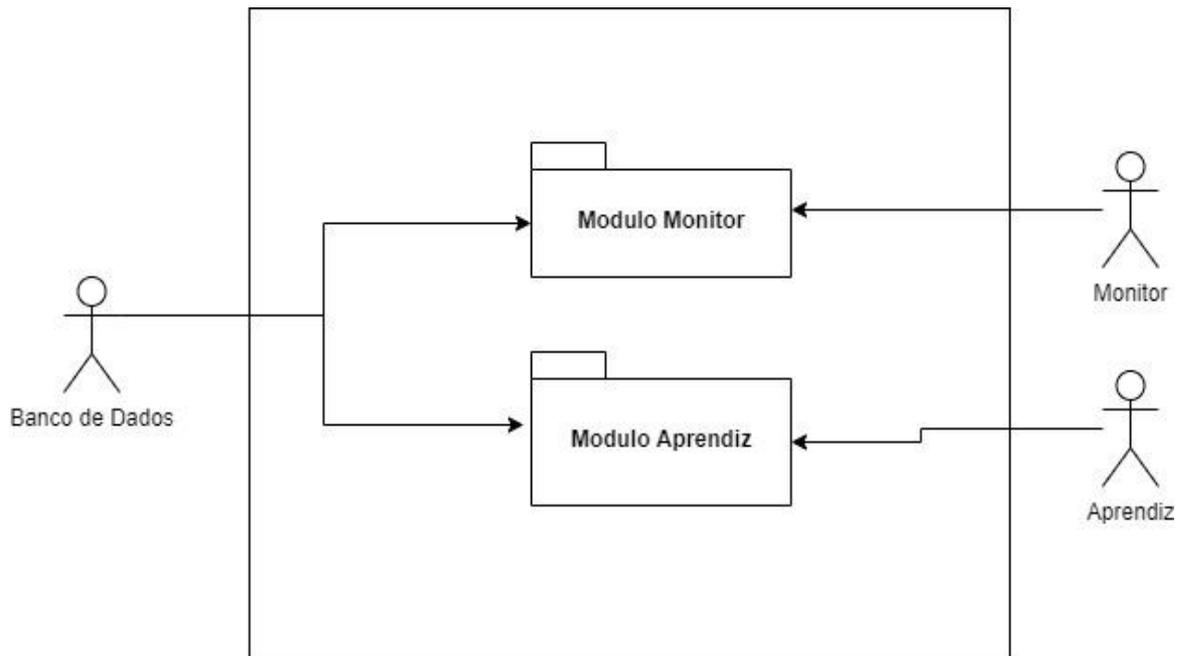
O ABCÊ Bulir (Rodrigues e Felix, 2014) é um software educacional que utiliza o Kinect para promover interação com os usuários para ajudar no processo de alfabetização. Este contém três módulos: apresentações das letras do alfabeto, formação de pequenas palavras e formação de sílabas.

Os projetos citados disponibilizam soluções que buscam incluir pessoas com síndrome de Down, auxiliando tanto no processo de letramento como no de alfabetização. Contudo as soluções propostas são desenvolvidas em plataformas específicas, o que podem causar limitações para alguns usuários que não tem acesso a plataforma conforme foi desenvolvida. Neste contexto, um aplicativo multiplataforma possui suporte a uma quantidade maior de dispositivos disponíveis no mercado e isso permite que usuários de plataformas diferentes utilizem o mesmo aplicativo, o que favorece o objetivo de inclusão. Atualmente com o crescimento do acesso à internet pelo mundo, manter os dados do usuário armazenados em um servidor é importante, para os mesmos serem acessados posteriormente de qualquer lugar ou dispositivo por meio de API.

3. ESPECIFICAÇÃO DE REQUISITOS

A atividade de elicitação de requisitos corresponde à etapa de compreensão do problema aplicada ao desenvolvimento de software (BEZERRA, 2015). Este capítulo mostra como o sistema vai funcionar, os tópicos seguintes mostram respectivamente o módulo monitor (item 3.1) e módulo aprendiz (item 3.2), onde ambos possuem o detalhamento de caso de uso.

Figura 3.1 - Caso de Uso



Fonte: Autoria Própria, 2020

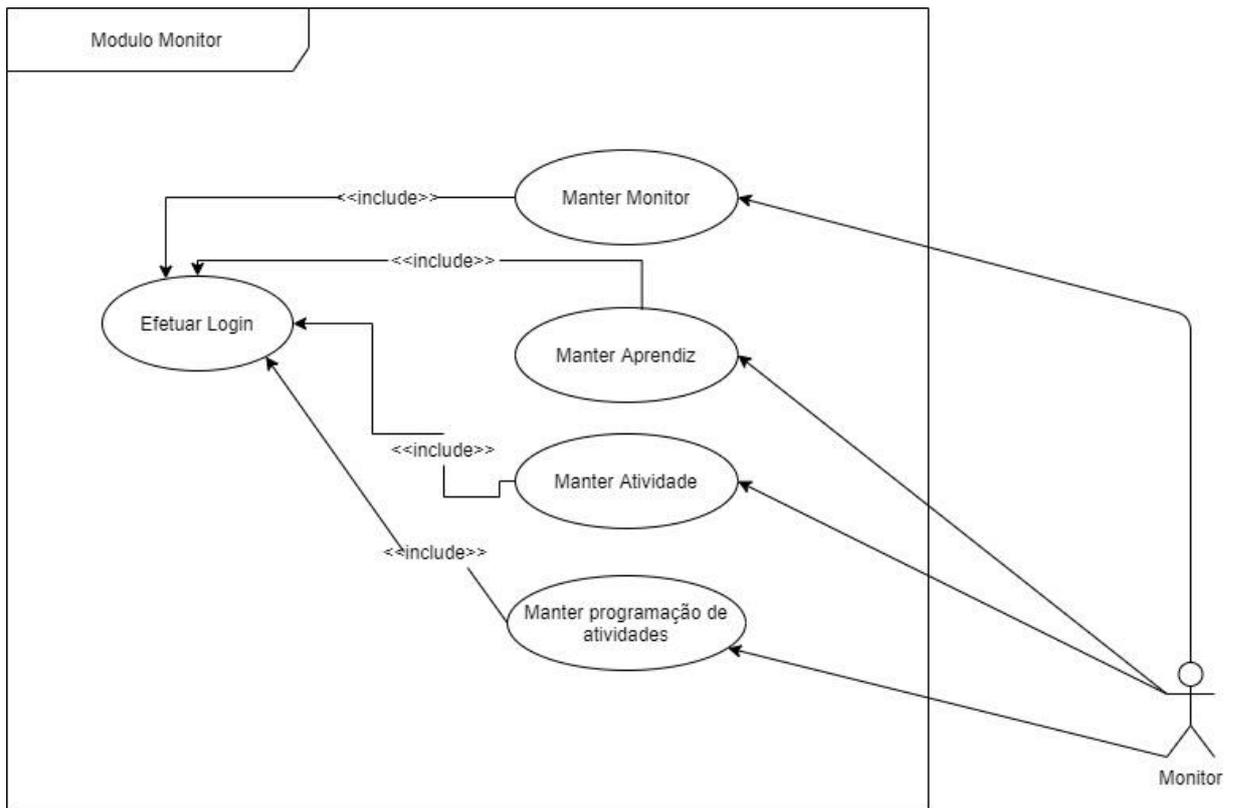
No contexto a interface desenvolvida há dois módulos (**Figura 3.1**):

- **Módulo Aprendiz:** Módulo onde o usuário aprendiz realizará as atividades de letramento propostas pelos monitores.
- **Módulo Monitor:** Módulo responsável pela manutenção de monitores, aprendizes e atividades. Neste módulo também é possível acompanhar o desempenho dos aprendizes através de relatórios.

3.1. Módulo Monitor

O usuário monitor tem a competência de gerir algumas funções dentro da interface, os casos de uso deste módulo são ilustrados na figura 3.2. Em seguida cada caso de uso é transformado em novos casos de uso para a visualização mais detalhada de cada artefato deste módulo.

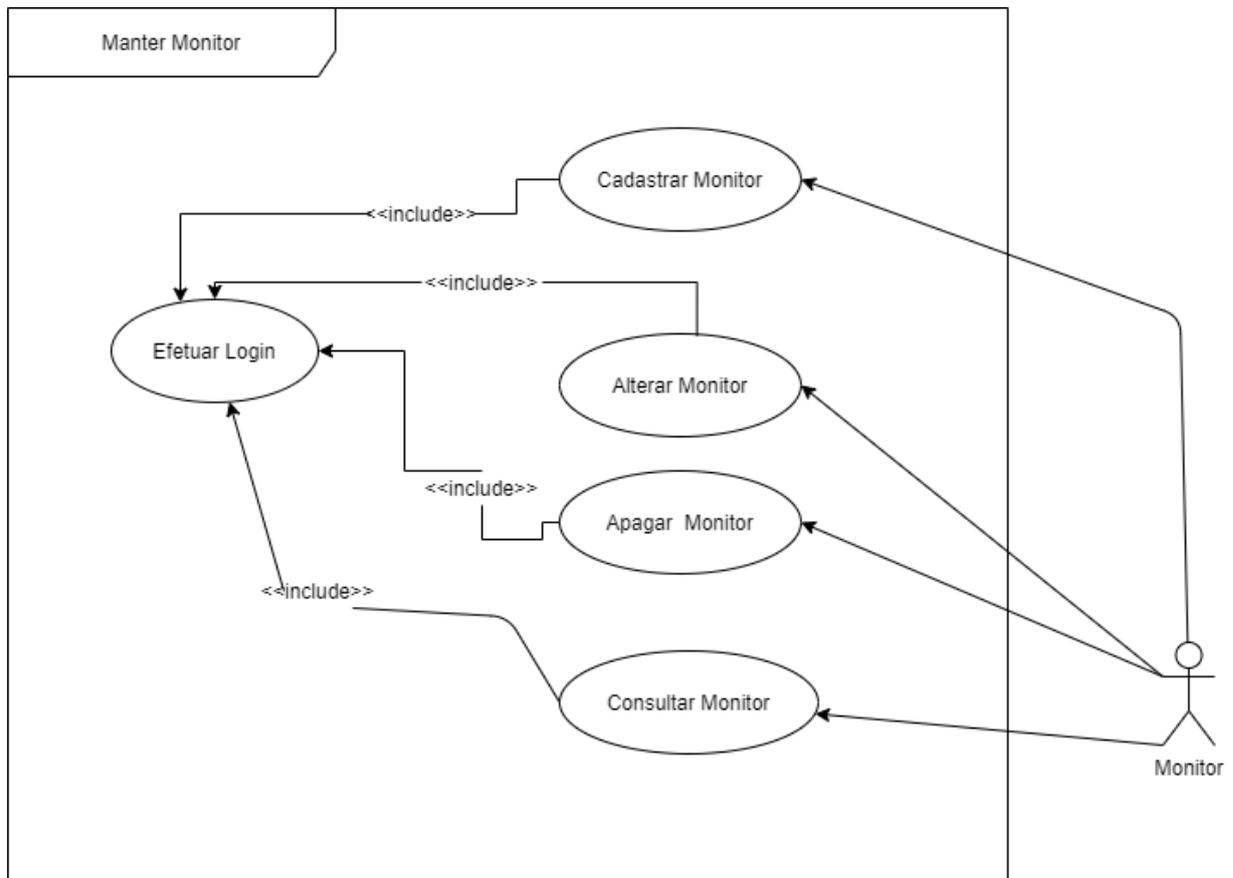
Figura 3.2 - Caso de Uso - Módulo Monitor



Fonte: Autoria Própria, 2020

3.1.1 Manter Monitor (RF01)

Figura 3.2 - Caso de Uso - Manter Monitor



Fonte: Autoria Própria, 2020

Descrição do cenário:

3.1.1.1 Cadastrar Monitor (RF01.1)

Descrição de cenários.

1. O usuário deve estar autenticado no sistema como monitor.
2. O novo monitor deverá estar matriculado na universidade.
3. O usuário informa a matrícula do monitor.
4. O sistema analisa o número de matrícula do monitor e verifica se a matrícula já se encontra cadastrada no sistema. Caso já esteja cadastrado o sistema retorna uma mensagem de erro informando que o respectivo número de matrícula já se encontra cadastrado no sistema.
5. O usuário informa nome, sexo, telefone, data de nascimento, curso matriculado e o e-mail.

6. O usuário efetiva o cadastro através do botão “Cadastrar”

3.1.1.2 Consultar Monitor (RF01.2)

Descrição de cenários.

1. O usuário deve estar autenticado no sistema como monitor.
2. O usuário informa o nome, id ou e-mail do monitor.
3. O sistema busca o nome do monitor e exibe as informações do monitor. Caso não exista o sistema informa que não foi encontrado o monitor.

3.1.1.3 Alterar Monitor (RF01.3)

Descrição de cenários.

1. O usuário deve estar autenticado no sistema como monitor.
2. O usuário informa o nome, id ou e-mail do monitor que deseja alterar.
3. O sistema busca o nome do monitor e exibe as informações do monitor pronto para serem editadas. Caso não exista o sistema informa que não foi encontrado o monitor.
4. O usuário efetiva a alteração através do botão “Salvar”

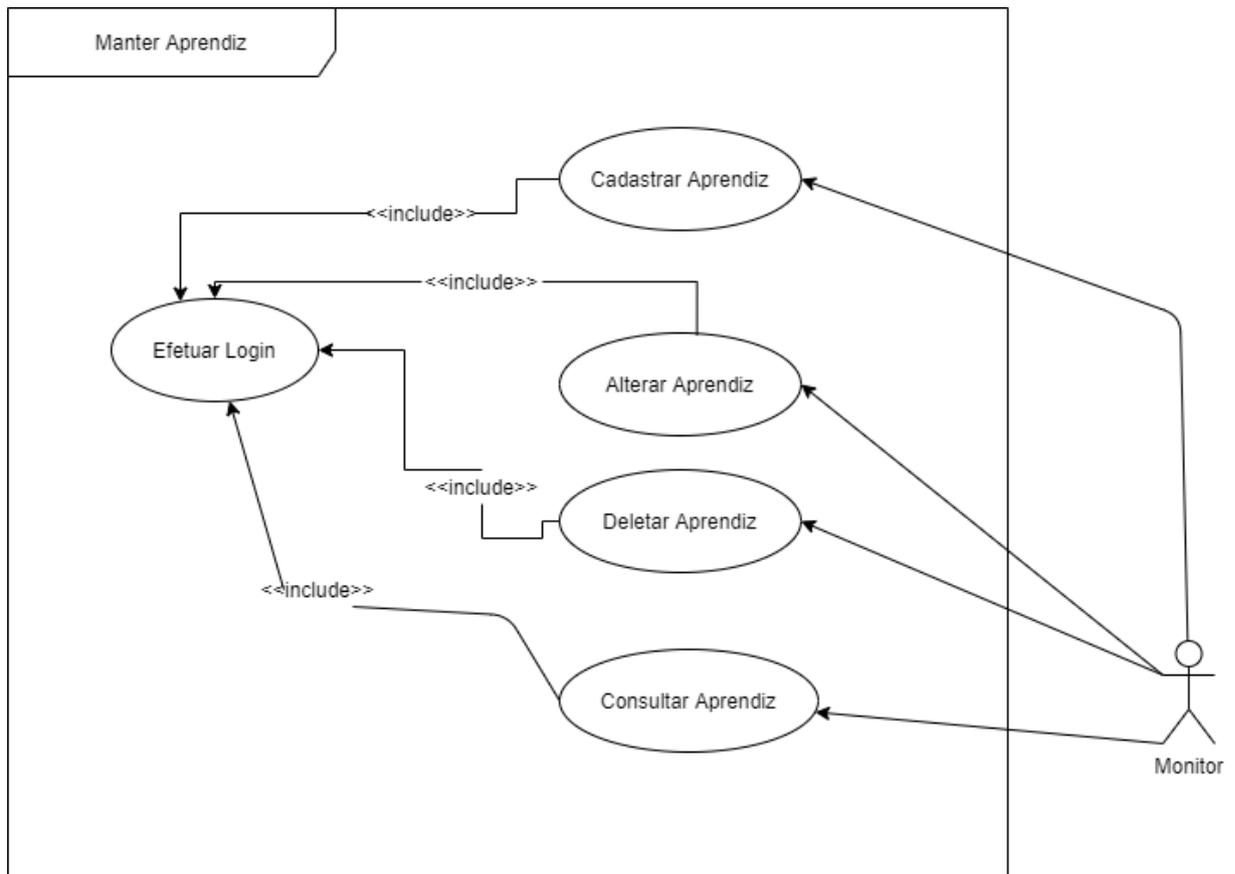
3.1.1.4 Apagar Monitor (RF01.4)

Descrição de cenários.

1. O usuário deve estar autenticado no sistema como monitor.
2. O usuário informa o nome, id ou e-mail do monitor que deseja apagar.
3. O sistema vai listar o usuário e mostrar um botão escrito “apagar”.
4. O usuário efetiva a exclusão através do botão “apagar” e confirmando após selecionar este botão.

3.1.2 Manter Aprendiz (RF04)

Figura 3.3 - Caso de Uso - Manter Aprendiz



Fonte: Autoria Própria, 2021

Descrição do cenário:

3.1.2.1 Cadastrar Aprendiz (RF04.1)

Descrição de cenários.

1. O usuário deve estar autenticado no sistema como monitor.
2. O novo aprendiz deverá conter apelido.
3. O usuário informa nome, sexo, telefone, data de nascimento, e o e-mail.
4. O usuário efetiva o cadastro através do botão “Cadastrar”

3.1.2.2 Consultar Monitor (RF04.2)

Descrição de cenários.

1. O usuário deve estar autenticado no sistema como monitor.
2. O sistema retorna todos os aprendizes cadastrados.
3. Caso não exista o sistema informa que não foi encontrado o aprendiz.

3.1.2.3 Alterar Monitor (RF04.3)

Descrição de cenários.

1. O usuário deve estar autenticado no sistema como monitor.
2. O usuário informa o nome, id ou e-mail do aprendiz que deseja alterar.
3. O sistema busca o nome do monitor e exibe as informações do aprendiz pronto para serem editadas. Caso não exista o sistema informa que não foi encontrado o monitor.
4. O usuário efetiva a alteração através do botão “Salvar”

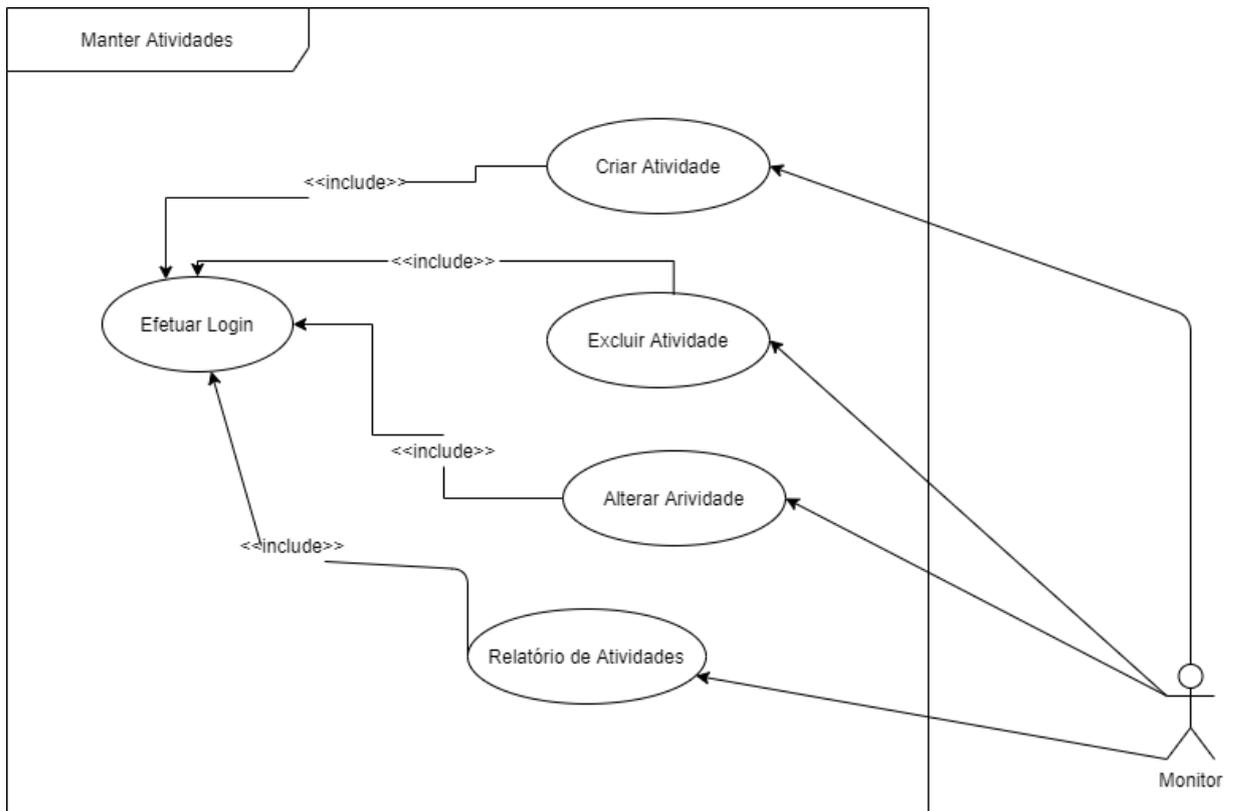
3.1.2.4 Apagar Monitor (RF04.4)

Descrição de cenários.

1. O usuário deve estar autenticado no sistema como monitor.
2. O usuário informa o nome, id ou e-mail do monitor que deseja apagar.
3. O sistema vai listar o usuário e mostrar um botão escrito “apagar”.
4. O usuário efetiva a exclusão através do botão “apagar” e confirmando após selecionar este botão.

3.1.3 Manter Atividades (RF05)

Figura 3.4 - Caso de Uso - Manter Atividades



Fonte: Autoria Própria, 2021

Descrição de cenários.

3.1.3.1 Elaborar Atividade (RF05.1)

Descrição do cenário:

1. O usuário deve estar autenticado no sistema como monitor.
2. O usuário informa o nome da atividade, a palavra a ser acertada, sugestão de resposta, nível de dificuldade e inserir uma imagem relacionado à palavra.
3. O usuário efetiva a elaboração da atividade através da opção “Salvar”.

3.1.3.2 Excluir Atividade (RF05.2)

Descrição do cenário:

1. O usuário deve estar autenticado no sistema como monitor.
2. Na aba de atividades o monitor escolhe a atividade em que deseja excluir.

3. O sistema emite uma mensagem para confirmar a exclusão.
4. O monitor efetiva a exclusão confirmando no sistema.

3.1.3.3 Alterar Atividade (RF05.3)

Descrição do cenário:

1. O usuário deve estar autenticado no sistema como monitor.
2. Na aba de atividade o monitor escolhe a atividade que deseja alterar.
3. O sistema exibe todas as informações da atividade que permite alteração.
4. O monitor efetiva a alteração da atividade através da opção “Salvar”.

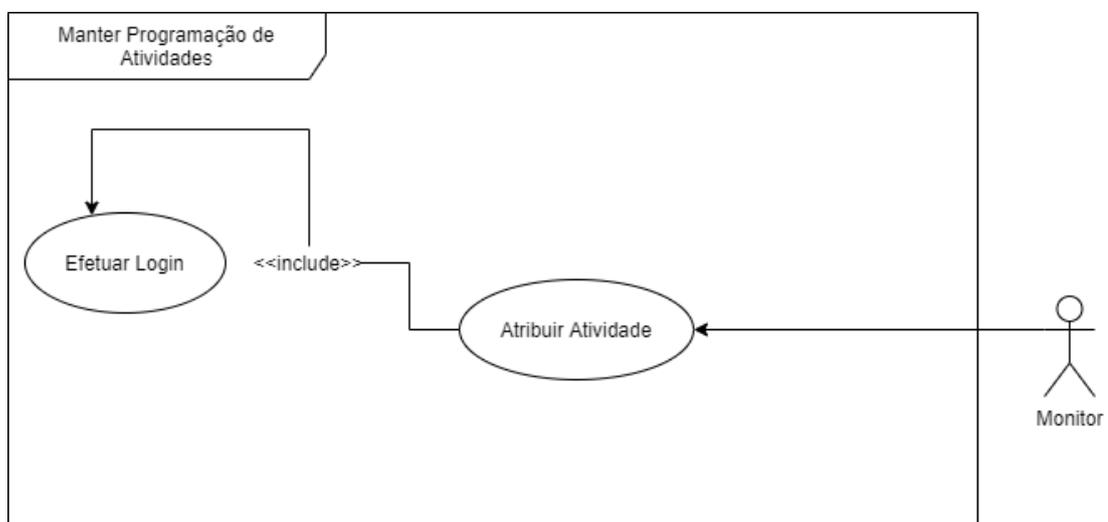
3.1.3.4 Relatório de Atividade (RF05.4)

Descrição do cenário:

1. O usuário deve estar autenticado no sistema como monitor.
2. Na aba de atividades o monitor escolhe a atividade que deseja obter o relatório.
3. O sistema exibe o relatório de execução de atividade de acordo com o aprendiz que a executou.

3.1.4 Manter Programação de atividades.

Figura 3.5 - Caso de Uso - Manter Programação de Atividades



Fonte: Autoria Própria, 2021

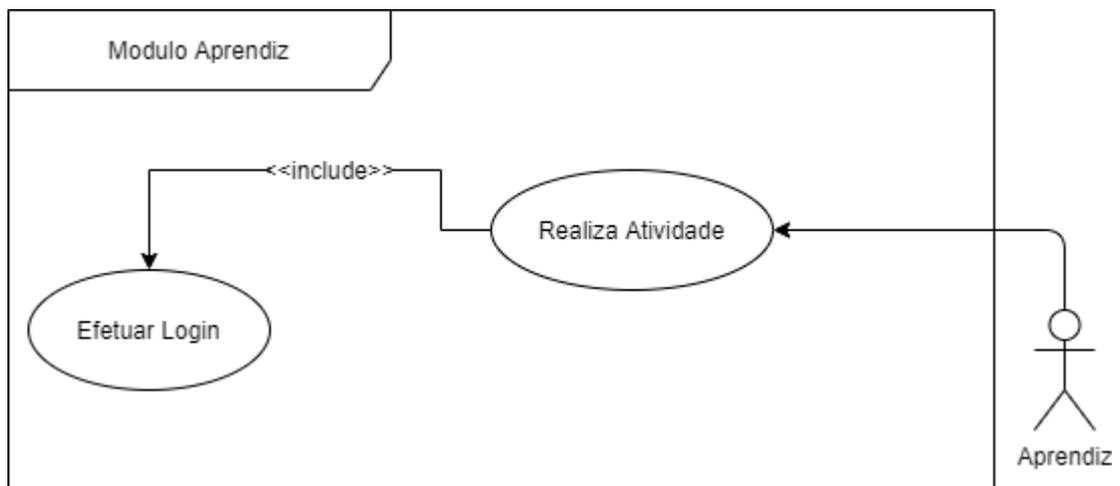
3.1.4.1 Atribuir Atividade (RF06)

Descrição do cenário:

1. O usuário deve estar autenticado no sistema como monitor.
2. O sistema lista as atividades e os aprendizes disponíveis.
3. O usuário deve selecionar a atividade e o aprendiz.
4. O usuário efetiva a atribuição da atividade para o aprendiz através da opção “Salvar”.

3.2. Módulo Aprendiz

Figura 3.4 - Caso de Uso - Manter Programação de Atividades



Fonte: Autoria Própria, 2021

3.2.1 Atribuir Atividade (RF07)

Descrição do cenário:

1. O usuário deve estar autenticado no sistema como aprendiz.
2. O sistema exibe as atividades disponíveis para serem realizadas.
3. O aprendiz deverá realizar a atividade.
4. Após concluir a atividade o sistema vai para a próxima atividade
5. Quando o aprendiz concluir todas atividades o sistema informará que não tem mais atividades disponíveis no momento.

4. PROJETO

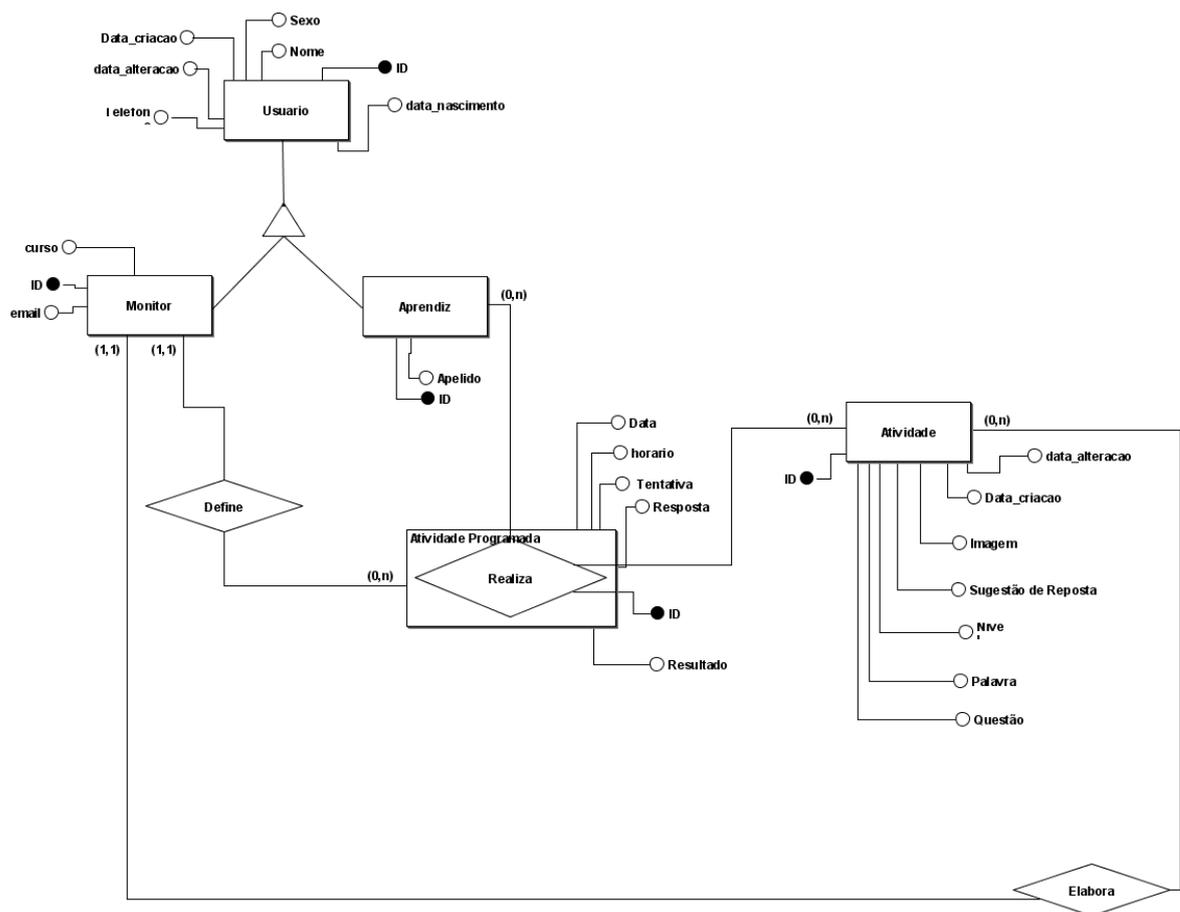
Este capítulo está organizado no seguinte formato: na seção 4.1, descreve o modelo conceitual onde expõe uma descrição do banco de dados da interface. Na seção 4.2, apresenta a modelagem de classe de análise, modelo construído na fase de análise. Na seção 4.3, apresenta o modelo de interação. Na seção 4.4 apresenta o modelo de arquitetura,

4.1. Modelo Conceitual

O modelo conceitual ajuda a entender melhor o sistema de acordo com o mundo real, o modelo conceitual do banco de dados da aplicação.

Descrevendo melhor o modelo, o projeto possui dois tipos de usuário, sendo eles o aprendiz e o monitor, onde o monitor consegue elaborar as atividades, e também definir atividades para um aprendiz. As atividades programadas são aquelas associadas a algum aprendiz, logo n aprendizes podem ter 0 ou N atividades programadas. Uma atividade pode estar programada para nenhum ou vários aprendizes. O Aprendiz tem acesso apenas às atividades que foram programadas para ele.

Figura 4.1 - Modelo Conceitual

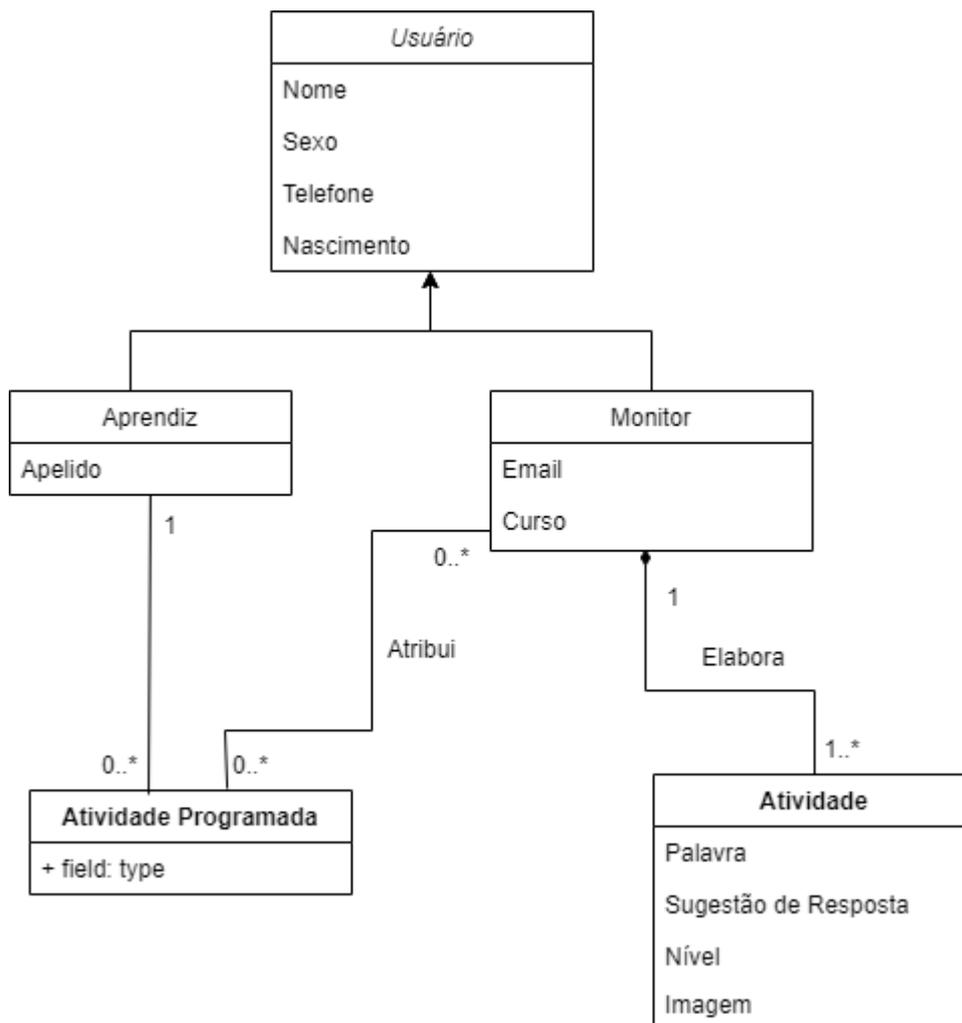


4.2. Modelo de Classe de Análise

A fim de esclarecer o problema do projeto, o modelo de classe de análise representa o conjunto de classes do sistema com base nos casos de uso levantados no capítulo 3.

Compreendendo o modelo de classes elaborado o sistema possui dois tipos de usuários: monitor e aprendiz. O Monitor tem acesso às atividades do sistema para a criação, alteração e visualização da mesma. O aprendiz e o monitor têm acesso à atividade programada. Logo o sistema permite que o monitor programe uma atividade para o aprendiz e que o aprendiz execute aquela atividade programada. O modelo de classe de análise na figura 4.2 ilustra como é o relacionamento entre as classes.

Figura 4.2 - Modelo de Classe de Análise



Fonte: Autoria Própria, 2020

4.3. Modelo de Interação

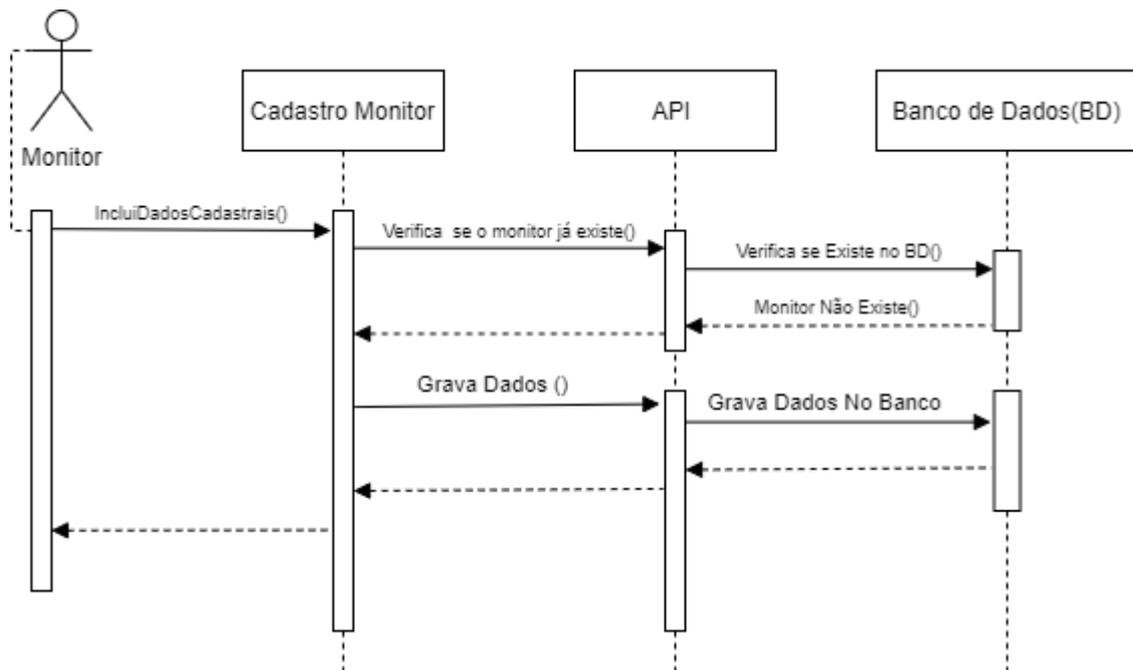
O modelo de interação ilustra as interações formadas por um conjunto de objetos e também seus relacionamentos, incluindo mensagens que podem ser trocadas entre estes objetos. O Modelo de Interação foi dividido em dois tópicos: Módulo Monitor (4.3.1) e Módulo Aprendiz (4.3.2). Para desenvolvimento dos modelos de interação foi utilizado o diagrama de sequência, “usados para modelar interações entre os componentes do sistema” (SOMMERVILLE, 2011, p.86)

4.3.1. Módulo Monitor

Os modelos de interação do módulo monitor foram extraídos com base nos casos de uso do Módulo Monitor (3.1). As interações desses módulos têm como ator o monitor, e como classes a interface de usuário, aplicação e o banco de dados respectivamente em cada diagrama.

4.3.1.1. Cadastro de Monitor

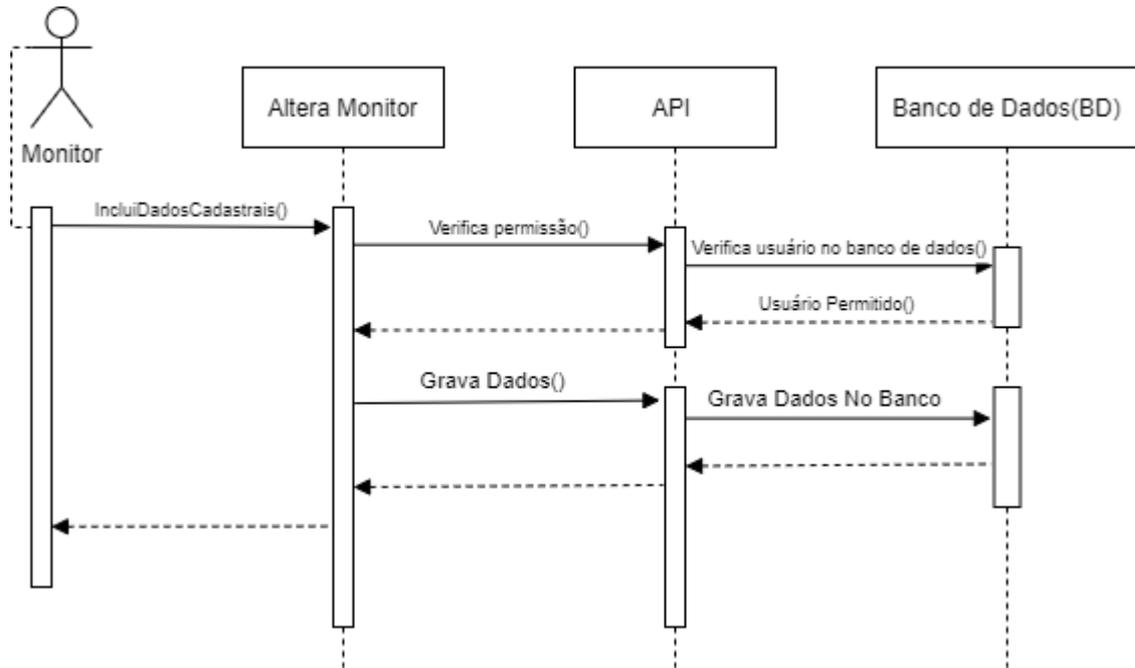
Figura 4.3 - Cadastrar Monitor



Fonte: Autoria Própria, 2020

4.3.1.2. Alterar Monitor

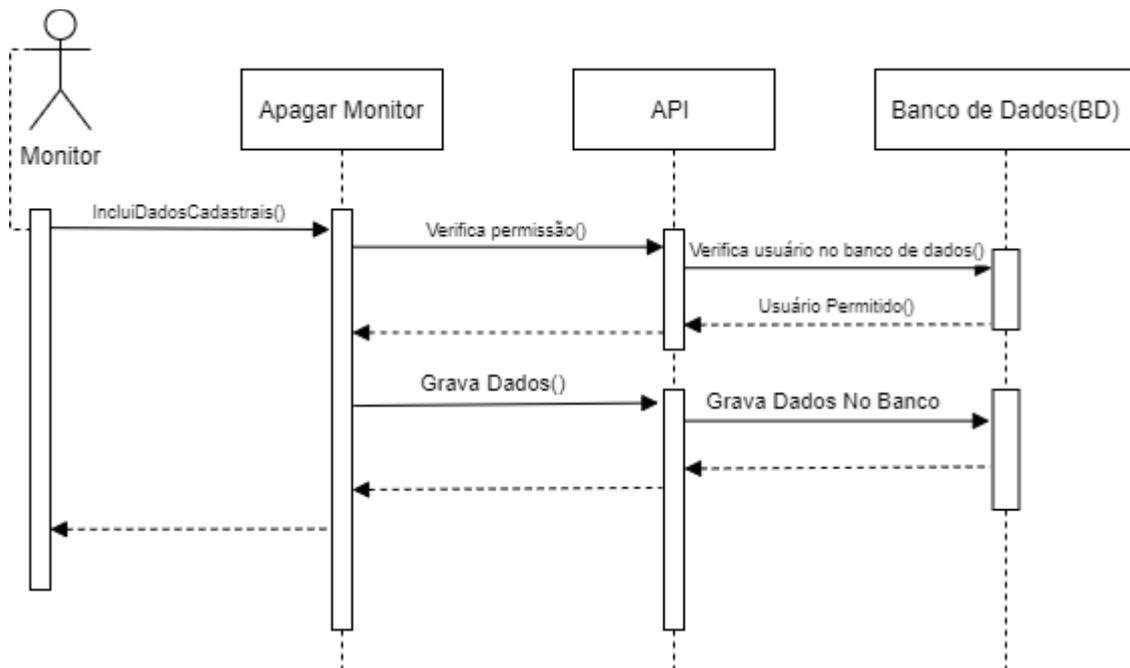
Figura 4.3 - Alterar Monitor



Fonte: Autoria Própria, 2020

4.3.1.3 Apagar Monitor

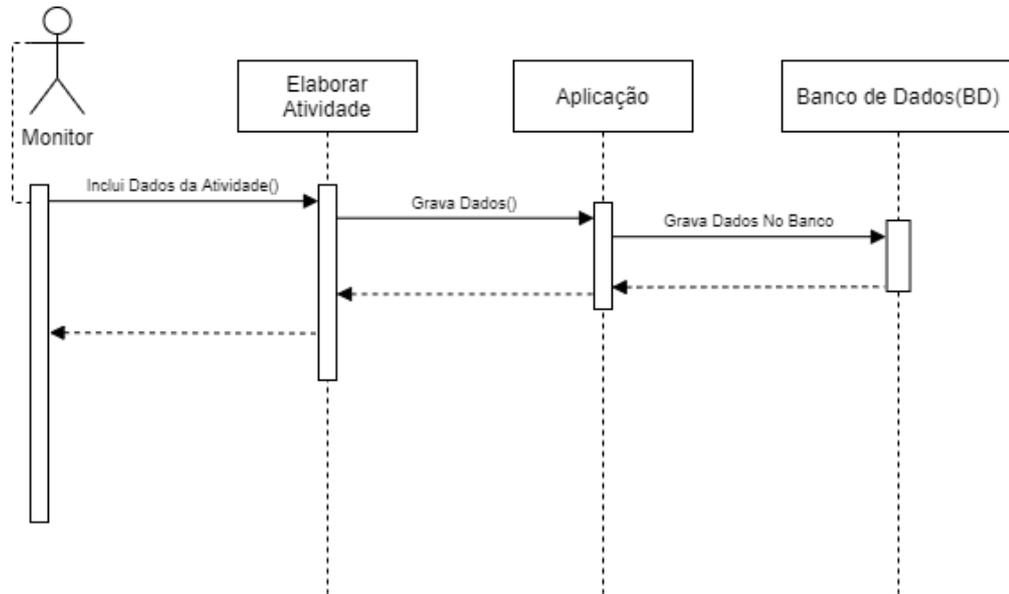
Figura 4.4 -Apagar Monitor



Fonte: Autoria Própria, 2020

4.3.1.4. Elaborar Atividade

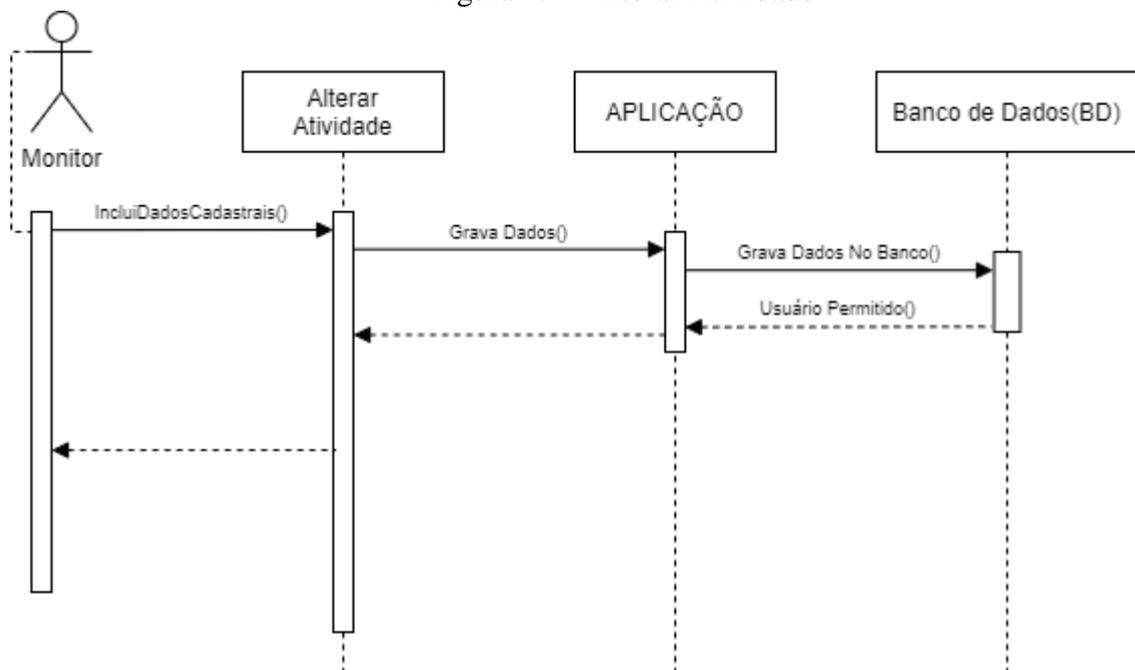
Figura 4.4 - Elaborar Atividade



Fonte: Autoria Própria, 2020

4.3.1.5. Alterar Atividade

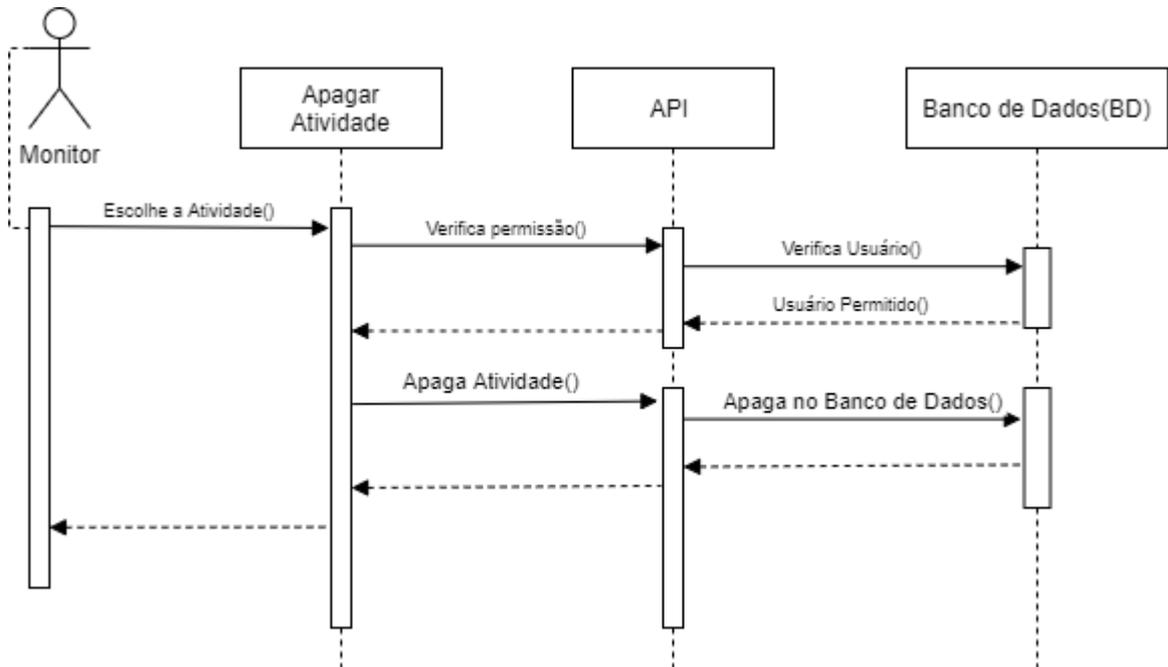
Figura 4.4 - Alterar Atividade



Fonte: Autoria Própria, 2020

4.3.1.6. Excluir Atividade

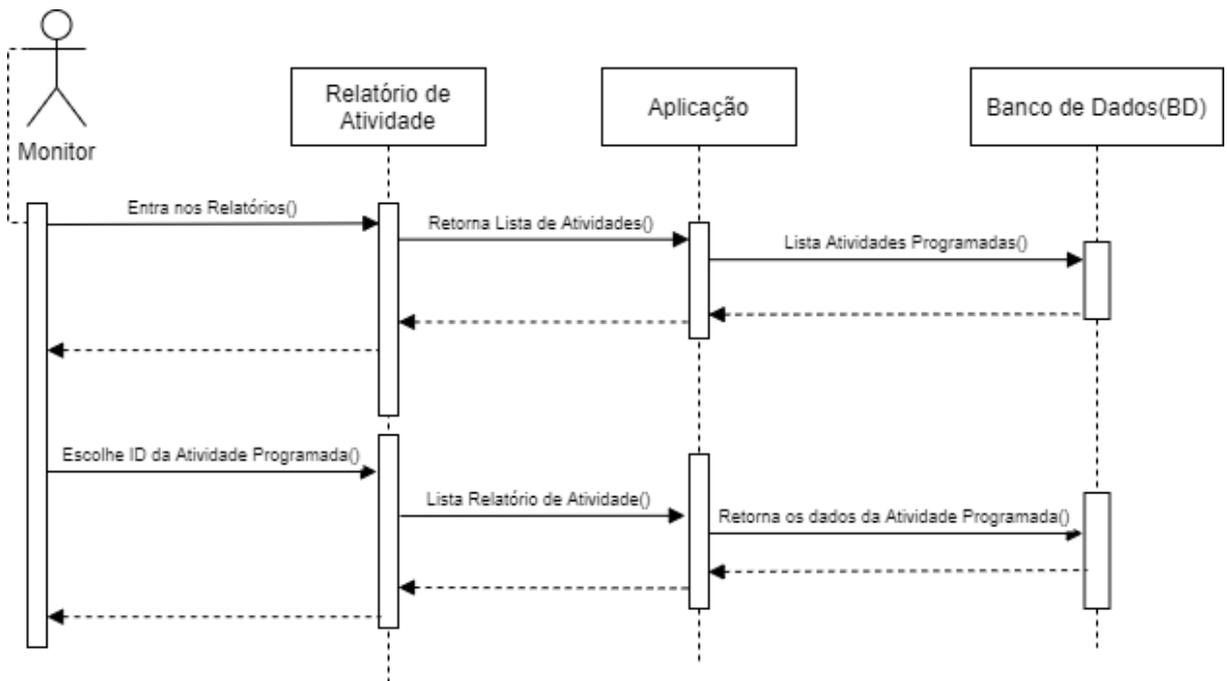
Figura 4.5 - Apagar atividade



Fonte: Autoria Própria, 2020

4.3.1.7. Relatório de Atividade

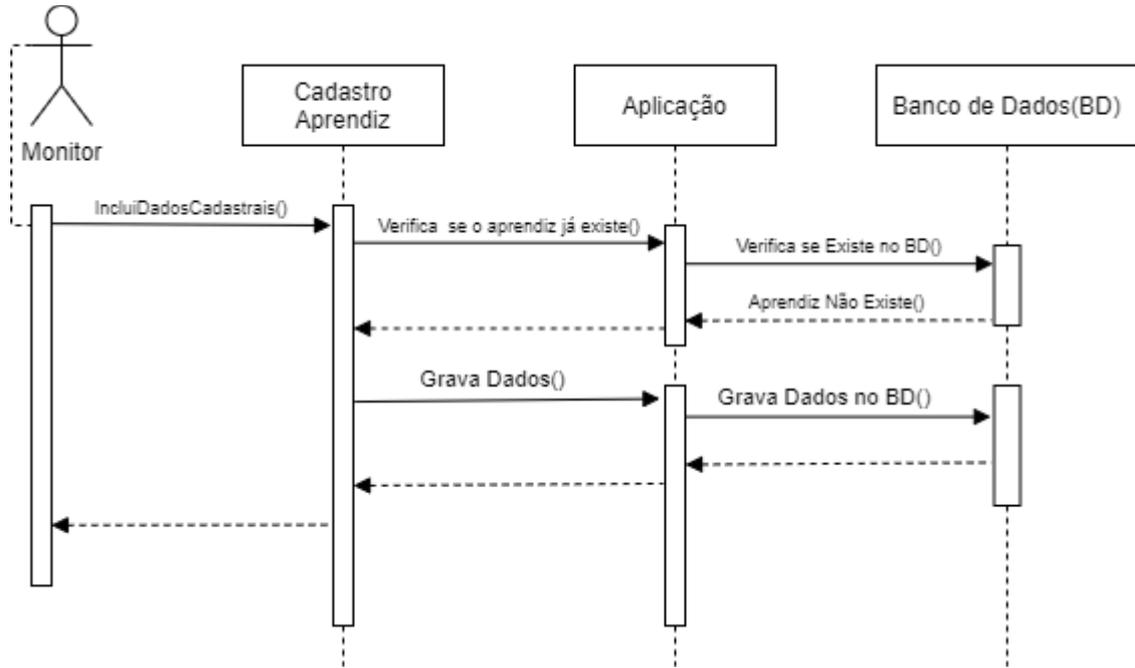
Figura 4.6 - Apagar atividade



Fonte: Autoria Própria, 2020

4.3.1.8. Cadastrar Aprendiz

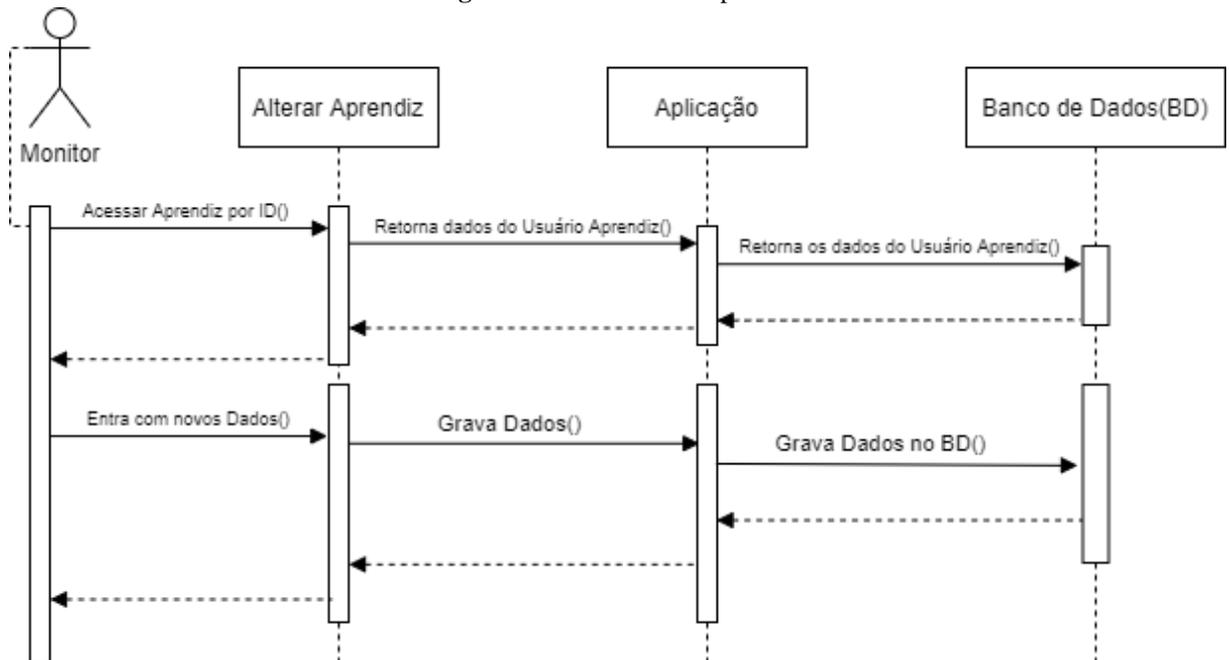
Figura 4.7 - Cadastro de aprendiz



Fonte: Autoria Própria, 2020

4.3.1.9. Alterar Aprendiz

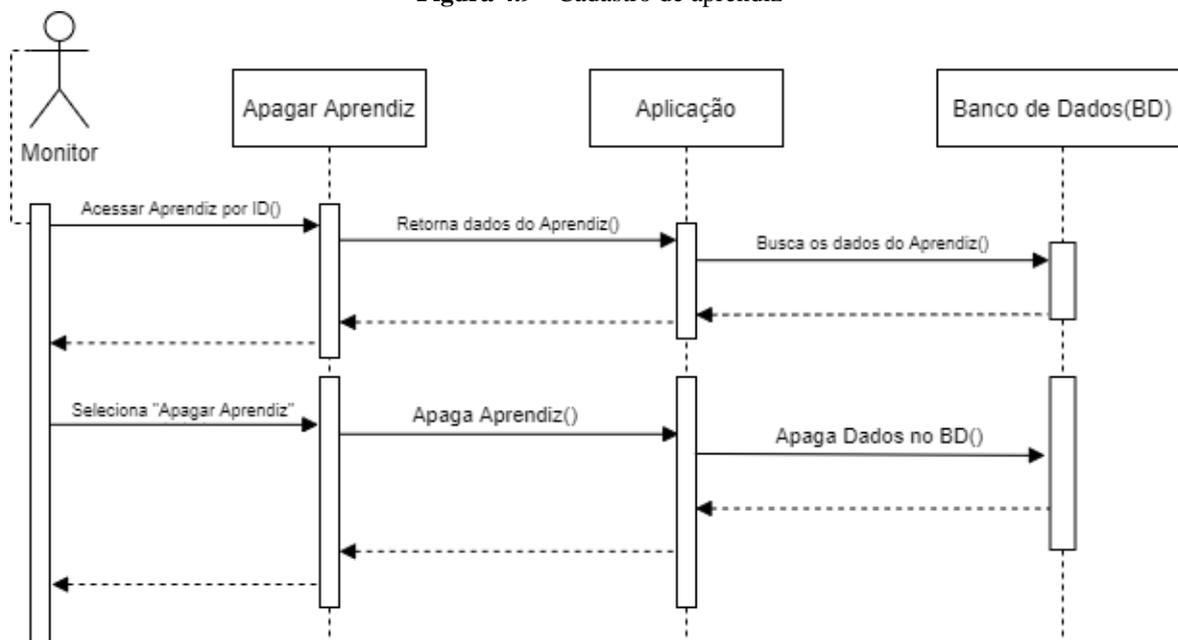
Figura 4.8 - Cadastro de aprendiz



Fonte: Autoria Própria, 2020

4.3.1.10. Apagar Aprendiz

Figura 4.9 - Cadastro de aprendiz



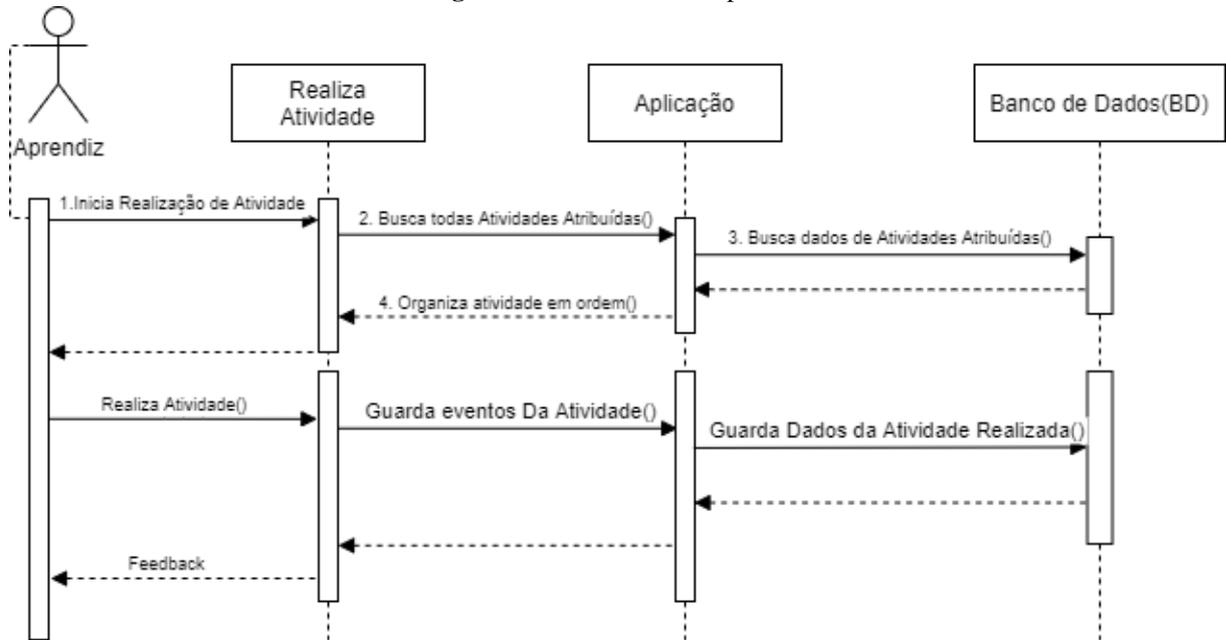
Fonte: Autoria Própria, 2020

4.3.2. Módulo Aprendiz

O modelo de Interação do módulo aprendiz foi extraído do caso de uso do Módulo Aprendiz (3.2). O diagrama de interação deste módulo tem como ator o aprendiz, e como classe a interface de usuário, aplicação e o banco de dados respectivamente.

4.3.2.11. Realizar Atividade

Figura 4.10 - Cadastro de aprendiz



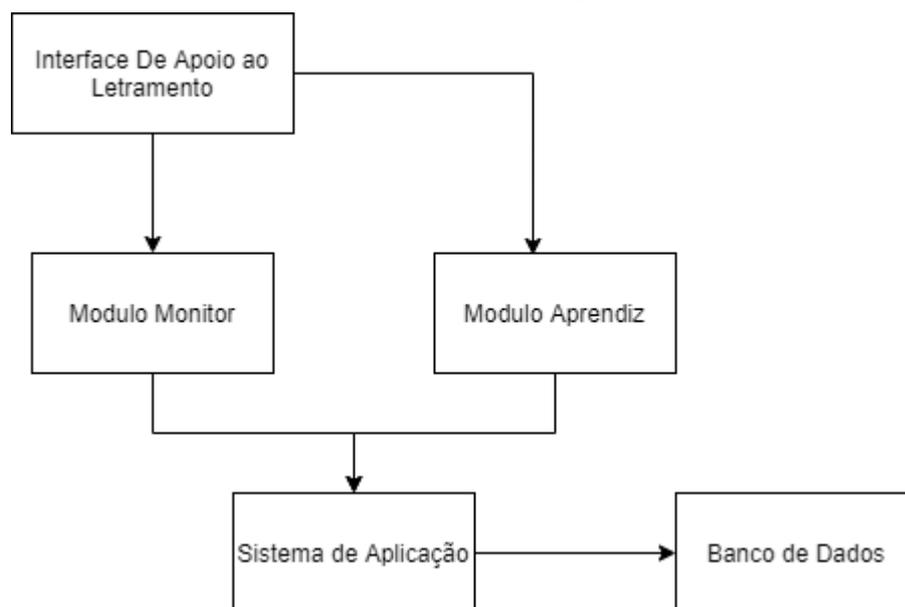
Fonte: Autoria Própria, 2020

4.4. Modelo de Arquitetura

“O projeto de arquitetura está preocupado com a compreensão de como um sistema deve ser organizado e com a estrutura geral desse sistema,” (SOMMERVILLE, 2011, p.103).

No modelo de arquitetura do sistema (figura 4.11) é possível observar o comportamento abstraido do sistema, o projeto possui dois módulos, onde esses módulos vão se relacionar com um sistema de aplicação, e o sistema de aplicação irá se relacionar com o Banco de Dados.

Figura 4.11 - Modelo de Arquitetura

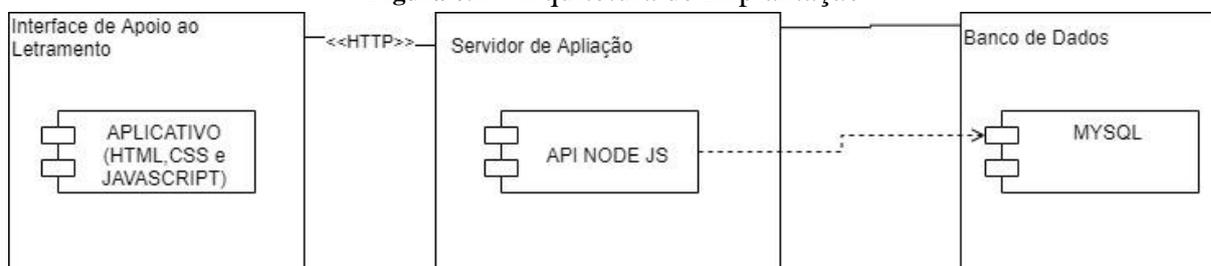


Fonte: Autoria Própria, 2020

6. IMPLANTAÇÃO

A Interface de Apoio ao Letramento foi estruturada em três camadas, o aplicativo web, servidor de aplicação e o banco de dados. Para desenvolvimento da Interface foram utilizadas as tecnologias que são suportadas pelos principais navegadores web, no servidor de aplicação foram empregados serviços web através de tecnologias que englobam o javascript, já no banco de dados foi utilizado um sistema de gerenciamento de banco de dados mysql.

Figura 6.1 - Arquitetura de Implantação



Fonte: Autoria Própria, 2021

6.1. Aplicativo Web

Para o desenvolvimento do aplicativo WEB, foi utilizado um framework React JS, uma tecnologia que tem como base as linguagens HTML, CSS e JavaScript. Com base na tecnologia adotada, os dois módulos foram desenvolvidos dentro do mesmo projeto, onde foram instanciadas rotas (caminhos inseridos no navegador) diferentes para cada módulo. Pensando

na divisão de rotas, existem dois caminhos distintos, o primeiro leva a interface para o ambiente do aprendiz e o segundo para o ambiente do monitor, como discutido em capítulos anteriores, ambos precisam estar autenticados para conseguirem ter acesso ao seu ambiente.

O projeto do aplicativo web encontra-se no seguinte repositório dentro do Github github.com/leonardosprado/tcc-frontend.

6.2. Servidor de Aplicação

Para o desenvolvimento do servidor de aplicação foi utilizado uma interface de programação de aplicação (API). Essa interface possibilita a comunicação do aplicativo web (seção 6.1) com o banco de dados através de padrões e protocolos. A API consiste em uma solução Web Service. A API Web Service possibilita universalizar o meio de transferência de dados entre as camadas do projeto, através do JSON. O JSON (em inglês, JavaScript Object Notation) é um formato para troca de dados entre sistemas. A vantagem deste formato é que ele é independente de linguagem de programação e é de fácil entendimento para humanos. Através de requisição HTTP o servidor de aplicação executa operações para a manipulação de dados, sendo as principais POST, GET, DELETE e PUT. A requisição POST tem o objetivo de criar dados dentro do servidor, GET fazer a leitura dos dados, DELETE apagar os dados e PUT fazer atualização nos dados.

Através da tecnologia node JS foi desenvolvido a solução de API Web Service. O Node JS é um ambiente que possibilita a utilização da linguagem de programação Javascript para a criação de aplicações sem a necessidade de um navegador web.

O servidor de Aplicação encontra-se no seguinte repositório dentro do GitHub: github.com/leonardosprado/tcc-api

6.3. Banco de Dados

Para o desenvolvimento do Banco de Dados foi utilizado o gerenciador de banco de dados MYSQL, o modelo conceitual do banco de dados é ilustrado na seção 4.1.

7. RESULTADOS OBTIDOS

Dentro do módulo do aprendiz foram desenvolvidas duas interfaces para facilitar o acesso ao ambiente de atividade. Já no módulo do monitor foram desenvolvidos uma sequência de interfaces com um menu lateral para navegação de acordo com os casos de uso levantados na seção 3.1 (Módulo Monitor) deste projeto. Já nos testes das atividades com os aprendizes aconteceram de forma remota, ou seja, o aprendiz fez o teste fora de um ambiente controlado.

No módulo aprendiz conta com uma interface de autenticação do usuário, onde é exigido o primeiro nome do aprendiz e uma senha. Após a autenticação do aprendiz, o sistema o direciona para uma nova interface com a primeira atividade do aprendiz, onde com o auxílio devido, poderá alterar o texto para cursivo ou manter no formato bastão. Esta interface também possui uma figura animada de um rosto representando a felicidade quando a palavra está certa e triste quando começa a errar a palavra. Na mesma tela também possui uma frase e uma imagem. A frase pode ser uma dica ou pergunta inserida pelo monitor que elaborou a atividade, tanto esta frase e a imagem ajudam o aprendiz na execução da atividade de letramento. Na realização da atividade o aprendiz irá visualizar as letras e poderá ir selecionando a letra de acordo, após concluir a atividade o sistema mudará para a próxima atividade. Após a conclusão da atividade, caso esteja correta o sistema toca um som animado e exibe balões comemorativos ilustrando que o aprendiz acertou a resposta. No caso de não estar correta o sistema não exibe mensagem, apenas aguarda a finalização e segue para a próxima atividade. Ao concluir a atividade o sistema avança para a próxima atividade, caso não tenha o sistema emite uma mensagem informando que não possui atividade. Na figura 7 é possível ver como está organizada a tela de atividade do aprendiz.

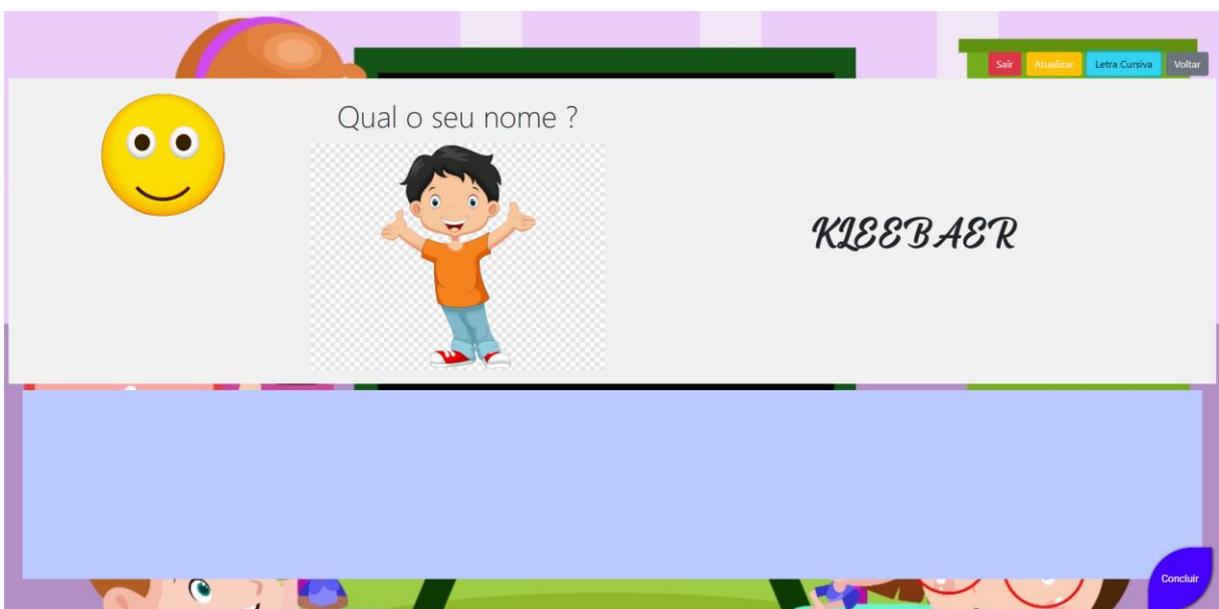
figura 7 - Letras no formato bastão



fonte: Autoria Própria, 2021

Como algumas pessoas possuem dificuldade para o entendimento de letras cursivas, foi inserido na interface uma funcionalidade para permitir a troca para a letra no formato bastão, que são letras que são utilizadas no processo de alfabetização.

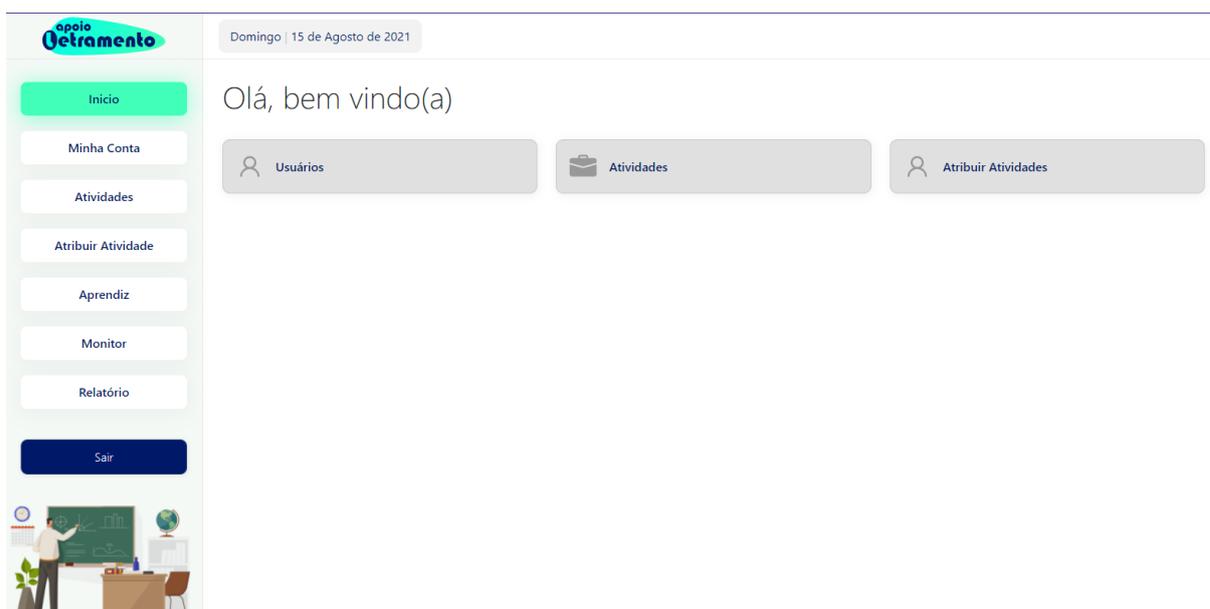
Figura 7.1 - Letras Cursivas



fonte: Autoria Própria, 2021

O módulo do monitor, foi construído com uma interface que busca facilitar a administração de toda plataforma, este módulo segue o diagrama de caso de uso do módulo do monitor (seção 3.1). Com um menu lateral contendo as opções que mantêm cada artefato. O monitor consegue uma navegabilidade de fácil entendimento. Para acesso a este módulo é necessário a autenticação com usuário e senha. Na figura 7.2 é possível visualizar o resultado desta tela.

Figura 7.2 - Módulo Monitor



fonte: Autoria Própria, 2021

7.1. Testes com aprendizes

O teste da interface com os aprendizes foi realizado de forma remota. O responsável do aprendiz o auxiliou na realização das atividades em um computador ou celular. Com a base de dados de educandos do projeto Alfadown da PUC-Goiás, foi possível realizar os testes com pessoas que têm síndrome de down. No teste foram realizadas 5 atividades de letramento com cada educando. Para a realização do teste foram escolhidos educandos que tiveram auxílio do pai ou responsável.

Foram escolhidos 5 educandos com autorização dos responsáveis para realização dos testes. O teste foi composto por um conjunto de atividades de letramento focadas em formar uma palavra escolhendo as letras. Foi proposto um conjunto com 5 atividades para serem

realizadas em até 5 minutos. As cinco atividades foram submetidas em um único teste. A tabela 7 ilustra as atividades proposta, o tempo médio de resposta de cada uma, os acertos e os erros.

Tabela 7 – Relatório de Atividades Proposta

Atividade	Pergunta	Resposta	Tempo Médio de Resposta (Segundos)	Acertos	Erros
1	Qual o seu nome?	O nome do Aprendiz	58,4	5	0
2	Qual o animal?	Cachorro	32,9	4	1
3	Qual o animal?	Coelho	63,4	4	1
4	Qual o objeto?	Óculos	28	3	2
5	Qual a Fruta?	Maçã	58,2	4	1

O tempo médio das respostas do conjunto de atividades de cada aprendiz atingiu aproximadamente de 4 minutos. Logo o tempo médio ficou abaixo do tempo máximo esperado de 5 minutos.

8 CONCLUSÃO

Os projetos desenvolvidos para auxílio de pessoas com síndrome de down no letramento, dão o suporte para execução de atividades que apoiam o aprendizado. Porém, as aplicações são restritas a plataformas que restringem o acesso dos usuários, permitido apenas para usuários do Android

A interface, desenvolvida neste projeto, eliminou tal limitação possibilitando o acesso de pessoas que possuem computadores, celulares ou dispositivos com suporte aos navegadores web. A interface possui telas com responsividade para comportar-se em qualquer tamanho de tela, ficando o aprendiz livre para escolher o computador com a utilização do mouse, ou o celular com touch screen.

A solução web service, permitiu o incremento de novos módulos dentro do aplicativo, assim podendo deixar o projeto mais apto a novas atualizações. Sendo assim, por ser um projeto web, pode ser disponibilizado em servidores online, podendo o usuário (aprendiz ou monitor) com acesso à internet se conectar à plataforma. O acompanhamento do aprendiz também ficou flexível, podendo em ambientes de ensino ou até mesmo remotamente, apenas com o auxílio dos pais.

Como trabalho futuro podem ser implementadas novas atividades de letramento. Por exemplo, uma atividade que possibilita a adição do uso do comando de voz, permitindo ao aluno falar a letra que deseja escolher. Além da possibilidade de analisar a pronúncia das letras, sílabas e palavras.

Também como trabalho futuro pode ser implementado o monitoramento da atenção do aprendiz utilizando o algoritmo de biofeedback, Attention Meter, utilizado na aplicação Lori's Help (Seção 2).

REFERÊNCIAS

ALVES, JOSIMAR. Meu ABC Down: Aplicativo para Alfabetização e Letramento de Crianças com Síndrome de Dow. Congresso Nacional de Educação, 4, 2017, João Pessoa. In: **Anais do IV Congresso Nacional de Educação – CONEDU**. João Pessoa-PB. Disponível em: Acesso em:

https://www.editorarealize.com.br/revistas/conedu/trabalhos/TRABALHO_EV127_MD1_S5_ID5508_26092019183239.pdf 10 mai. 2020

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 3. ed. rev. e atual. Rio de Janeiro: Elsevier, 2015.

O que é?, .Síndrome de Down. Disponível em:

<<http://www.movimentodown.org.br/sindrome-de-down/o-que-e/>>. Acesso em 30 de abril de 2020.

RODRIGUES, Michel; FÉLIX, Zildomar. ABCÊ Bulir: Uma Proposta de Software Educacional para o auxílio no Processo de Alfabetização e Desenvolvimento Cognitivo de Crianças com Síndrome de Down. Congresso brasileiro de informática na educação, 3,2014.. **Anais do 20º Workshop de Informática na Escola WIE 2014**. Dourados: Sociedade Brasileira de Computação,2014. p.11-15. Disponível em:

<https://www.br/ie.org/pub/index.php/wie/issue/view/88> Acesso em: 10 mai. 2020.

SOMMERVILLE, Ian. Engenharia de software, 9ª edição. São Paulo: Pearson Addison Wesley, 2011.

VINICIUS, Marcus et al. Lori's Help: um aplicativo para auxílio na alfabetização de pessoas com Síndrome de Down. **Revista Brasileira de Computação Aplicada (ISSN 2176-6649)**, Passo Fundo, p. 2-13. jul. 2017.

APÊNDICE

APÊNDICE A – BANCO DE DADOS EXPORTADO EM MYSQL

```
-- phpMyAdmin SQL Dump
-- version 5.0.2
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1:3306
-- Tempo de geração: 23-Ago-2021 às 03:44
-- Versão do servidor: 5.7.31
-- versão do PHP: 7.4.9

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Banco de dados: `apoiopletramento`
--

-----

--
-- Estrutura da tabela `aprendiz`
--

DROP TABLE IF EXISTS `aprendiz`;
CREATE TABLE IF NOT EXISTS `aprendiz` (
  `id_aprendiz` int(11) NOT NULL AUTO_INCREMENT,
  `apelido` varchar(255) DEFAULT NULL,
  `id_usuario` int(11) NOT NULL,
  PRIMARY KEY (`id_aprendiz`),
  UNIQUE KEY `apelido` (`apelido`),
  KEY `fk_UsuApre` (`id_usuario`)
) ENGINE=MyISAM AUTO_INCREMENT=16 DEFAULT CHARSET=latin1;

-----

--
-- Estrutura da tabela `atividade`
--
```

```

DROP TABLE IF EXISTS `atividade`;
CREATE TABLE IF NOT EXISTS `atividade` (
  `id_atividade` int(11) NOT NULL AUTO_INCREMENT,
  `data_criacao` datetime DEFAULT NULL,
  `data_alteracao` datetime DEFAULT NULL,
  `questao` text NOT NULL,
  `sugestao_resposta` varchar(255) DEFAULT NULL,
  `nivel` varchar(255) DEFAULT NULL,
  `palavra` varchar(255) DEFAULT NULL,
  `img_name` varchar(255) NOT NULL,
  `url_img` longtext NOT NULL,
  `id_monitor` int(11) NOT NULL,
  PRIMARY KEY (`id_atividade`),
  KEY `fk_MonAtiv` (`id_monitor`)
) ENGINE=MyISAM AUTO_INCREMENT=56 DEFAULT CHARSET=latin1;

-----

--
-- Estrutura da tabela `atividade_programada`
--

DROP TABLE IF EXISTS `atividade_programada`;
CREATE TABLE IF NOT EXISTS `atividade_programada` (
  `id_atividade_programada` int(11) NOT NULL AUTO_INCREMENT,
  `data_realizacao` datetime DEFAULT NULL,
  `data_criacao` datetime DEFAULT NULL,
  `data_alteracao` datetime DEFAULT NULL,
  `tentativa` text,
  `status` varchar(255) DEFAULT NULL,
  `resposta` varchar(255) DEFAULT NULL,
  `resultado` varchar(255) DEFAULT NULL,
  `id_monitor` int(11) NOT NULL,
  `id_atividade` int(11) NOT NULL,
  `id_aprendiz` int(11) NOT NULL,
  PRIMARY KEY (`id_atividade_programada`) USING BTREE,
  KEY `fk_AtivReali` (`id_atividade`),
  KEY `fk_ApreReali` (`id_aprendiz`),
  KEY `fk_MonReali` (`id_monitor`)
) ENGINE=MyISAM AUTO_INCREMENT=139 DEFAULT CHARSET=latin1;

-----

--
-- Estrutura da tabela `monitor`
--

DROP TABLE IF EXISTS `monitor`;
CREATE TABLE IF NOT EXISTS `monitor` (

```

```

`id_monitor` int(11) NOT NULL AUTO_INCREMENT,
`email` varchar(255) DEFAULT NULL,
`curso` varchar(255) DEFAULT NULL,
`id_usuario` int(11) NOT NULL,
PRIMARY KEY (`id_monitor`),
UNIQUE KEY `email` (`email`),
KEY `fk_UsuMoni` (`id_usuario`)
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;

-----

--
-- Estrutura da tabela `usuario`
--

DROP TABLE IF EXISTS `usuario`;
CREATE TABLE IF NOT EXISTS `usuario` (
  `id_usuario` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `nome` varchar(255) DEFAULT NULL,
  `sexo` varchar(255) DEFAULT NULL,
  `telefone` varchar(155) DEFAULT NULL,
  `data_criacao` datetime DEFAULT NULL,
  `data_alteracao` datetime DEFAULT NULL,
  `data_nascimento` date DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id_usuario`,`username`) USING BTREE,
  UNIQUE KEY `username` (`username`)
) ENGINE=MyISAM AUTO_INCREMENT=40 DEFAULT CHARSET=latin1;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

APÊNDICE B – COMO CONFIGURAR O SERVIDOR WEB SERVICE

1. Para iniciar o servidor da aplicação é necessário a instalação do **NODE JS** versão 14.15.4, **NPM** e o **YARN**:
 - a. Para instalar o Node JS: <https://nodejs.org/en/>
 - b. Para instalar o YARN: <https://classic.yarnpkg.com/en/docs/install/#windows-stable>
2. Fazer download do diretório no seguinte repositório e logo em seguida extrair:
github.com/leonardoprado/tcc-api
3. Dentro do diretório é necessário criar o arquivo com o nome: `.env`
O arquivo env deve conter as seguintes variáveis de ambiente:

SECRET = Texto secreto para utilizar na autenticação de usuários.

MYSQL_HOST = Host do banco de dados

MYSQL_PORT= Porta do Banco de Dados

MYSQL_USER= Usuário do Banco de dados

MYSQL_PASSWORD= Senha do Banco de Dados

MYSQL_DATABASE= Nome do Banco de Dados

Exemplo de arquivo `.env`:

```
#.env, don't commit to repo
SECRET=prado
MYSQL_HOST=localhost
MYSQL_PORT=3306
MYSQL_USER=root
MYSQL_PASSWORD=
MYSQL_DATABASE=apoioletramento
```

4. Abrir o TERMINAL/PowerShell/CMD e entrar dentro do diretório do projeto:

Exemplo PowerShell do Windows:

```
> cd C:\api-main\tcc-api-main
```

Dentro do PowerShell dentro do diretório é necessário digitar o seguinte comando para a instalação das dependências do projeto :

```
yarn install ou npm install
```

Exemplo:

```
C:\api-main\tcc-api-main > yarn install
```

Posteriormente todas dependências do projeto serão instaladas. Em seguida para executar o projeto digite o seguinte comando no PowerShell:

```
yarn start
```

Exemplo:

```
C:\api-main\tcc-api-main > yarn start
```

Se tudo ocorrer corretamente vai ser informado no PowerShell a seguinte mensagem:

```
PS C:\Users\renat\Documents\tcc-api-main\tcc-api-main> yarn start
yarn run v1.22.10
$ nodemon
[nodemon] 2.0.12
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting node server.js
consign v0.1.6 Initialized in api
! Entity not found C:\Users\renat\Documents\tcc-api-main\tcc-api-main\api\data
+ .\controllers\atividade.js
+ .\controllers\authentication.js
+ .\controllers\authorization.js
+ .\controllers\user.js
+ .\routes\atividade.js
+ .\routes\authentication.js
+ .\routes\authorization.js
+ .\routes\user.js
Servidor rodando na porta 3000
```

Fonte: Autoria própria, 2021

5. Normalmente o servidor vai ser iniciado na porta 3000, lembrando que todas as rotas estão na versão 1 da API, logo a aplicação terá a seguinte rota: *localhost:3000/api/v1/*

APÊNDICE C – COMO CONFIGURAR O APLICATIVO

1. Para iniciar o aplicativo é necessário a instalação do **NODE JS** versão 14.15.4, **NPM** e o **YARN**:
 - a. Para instalar o Node JS: <https://nodejs.org/en/>
 - b. Para instalar o YARN: <https://classic.yarnpkg.com/en/docs/install/#windows-stable>
2. Fazer download do diretório no seguinte repositório e logo em seguida extrair:
github.com/leonardoprado/tcc-frontend
3. Dentro do diretório é necessário configurar o arquivo *.env.production* e *.env.development*
.env.production: para configurar o aplicativo em produção;
.env.development: para configurar o aplicativo em desenvolvimento.
Neste caso como está sendo configurando o aplicativo no modo desenvolvimento é necessário configurar o *.env.development*

Explicando cada variável:

REACT_APP_API_URL_HOST= Caminho que está rodando a API

Exemplo: `http://45.90.109.43:3000/api/v1/`

REACT_APP_URL_API= Caminho que está rodando o servidor

Exemplo: `http://45.90.109.43:3000/`

REACT_APP_VERSION_API= Versão da API
`api/v1/`

4. Abrir o TERMINAL/PowerShell/CMD e entrar dentro do diretório do projeto:

Exemplo Prompt de Comando do Windows:

```
> cd C:\api-main\tcc-frontend-main
```

No PowerShell dentro do diretório é basta digitar o seguinte comando:

```
yarn install ou npm install
```

Exemplo:

```
C:\api-main\tcc-frontend-main > yarn install
```

Posteriormente todas dependências do projeto serão instaladas e em seguida para executar o programa basta digitar o seguinte comando no PowerShell:

```
yarn react-scripts start
```

Exemplo:

```
C:\api-main\tcc-api-main > yarn react-scripts start
```

5. O servidor vai ser iniciado na porta 3000, caso a porta 3000 esteja ocupada pelo servidor API, é necessário confirmar a abertura do servidor na porta 3001. Logo em seguida o servidor do aplicativo iniciará em modo desenvolvimento na porta 3001.
6. Gerar a Build do Projeto (Projeto em Produção):

Para gerar os arquivos de *build* do projeto é necessário dentro do diretório do projeto inserir o comando:

```
yarn build
```

Exemplo: C:\api-main\tcc-api-main > *yarn build*

Se o processo ocorrer corretamente, dentro do projeto vai ser criado um diretório com o nome *build* com todos os arquivos do projeto necessário para colocá-lo em produção. Para funcionamento correto do aplicativo é necessário configurar o servidor em que o projeto ficará hospedado.

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Leonardo dos Santos Prado
do Curso de engenharia de computação, matrícula 20142003300405,
telefone: 62982732049 e-mail: LEONARDOPRADO19@GMAIL.COM, na qualidade de titular dos
direitos autorais, em consonância com a Lei n° 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
INTERFACE WEB PARA APOIO AO LETRAMENTO DE PESSOAS COM
SÍNDROME DE DOWN, 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, 18 de Agosto de 2021.

Assinatura do(s) autor(es): Leonardo dos Santos Prado

Nome completo do autor: Leonardo dos Santos Prado

Assinatura do professor-orientador: _____

Nome completo do professor-orientador: _____