

FACULDADE PROJEÇÃO
SISTEMAS de INFORMAÇÃO

WILLIAM VIEIRA BRITO DA SILVA

RECONHECIMENTO FACIAL E SEGURANÇA

Brasília, Setembro de 2021

FACULDADE PROJEÇÃO
SISTEMAS DE INFORMAÇÃO

TEMA

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de informação da Faculdade Projeção, como requisito complementar e obrigatório à obtenção do título de Bacharel em Sistemas de Informação.

Brasília, ____ de _____ de 2021.

Banca Examinadora:

Professor(a) :

Orientador(a)

Professor(a) :

Professor(a) :

Professor Rosa Maria Diekn de Queiroz, Mestre
Coordenadora do curso de Sistemas de Informação

AGRADECIMENTOS

Agradeço a Deus primeiramente, pois através do tempo dele tive a oportunidade de cursar algo que me transformou fortemente, apresentando-me o mundo acadêmico e o mundo do desenvolvimento de software.

Sou muito grato aos amigos que fiz na faculdade, as companhias de intervalo, as ajudas de código que foram de extrema importância para prosseguir em minha caminhada que está só começando. Além de códigos agradecer também pelas partidas de Minecraft e Muck com intensas batalhas e boas risadas a noite que me deixaram muito mais feliz.

Agradeço imensamente aos professores que fizeram e fazem parte da minha formação acadêmica e profissional, sem os ensinamentos e a compreensão dos mesmos não estaria aqui neste momento. Um agradecimento especial aos professores Fernando Feliu Goncales, João Evangelista, João Evangelista, Ivonete Ferreira Sousa, Alexandre Hamilton e ao Ricardo Akl.

Sou absurdamente grato aos que nos antecederam, principalmente aos que de certa forma influenciaram a tecnologia em seu tempo. Agradeço a todos os seres humanos extraordinários que desenvolveram linguagens de programação que atualmente impactam diretamente para o conforto de bilhões de pessoas, muito obrigado.

Link do vídeo de apresentação da primeira entrega do projeto:

<https://youtu.be/a7TtF2dvi3o>

Link do vídeo de apresentação da segunda entrega do projeto:

<https://youtu.be/CCHa69F01Wg>

SUMÁRIO

SUMÁRIO	4
LISTA DE ABREVIATURAS E SIGLAS	8
LISTA DE FIGURAS.....	9
LISTA DE TABELA	10
RESUMO	11
ABSTRACT	12
CAPÍTULO I	13
VISÃO INICIAL.....	13
1. Introdução	13
2. Contextualização	14
3. Problemática	15
4. Solução Proposta.....	16
5. Objetivo Geral	16
Objetivos Específicos.....	16
6. Estrutura do Projeto	16
CAPÍTULO II	18
METODOLOGIA APLICADA	18
1. Metodologia e Modelo de Desenvolvimento	18
2. Técnicas de Levantamento de Requisitos.....	19
2.1 Brainstorming.....	19
2.2 Entrevista.....	19
2.3 Questionários	20
3. Principais tecnologias utilizadas	21
3.1 Arduino IDE	21
3.2 Composer	22
3.3 C/C++	23
3.4 Firebase	24
3.5 Laravel	25
3.6 MySQL	27
3.7 Open CV.....	27
3.8 PHP	28
3.9 Python	30
3.10 Reconhecimento Facial	32
3.11 SQL.....	33
3.12 Visual Studio Code.....	34
4. Análise de Sistemas Existentes.....	35
5. Cronogramas	36
CAPÍTULO III	38

VISÃO DO SISTEMA	38
1. Riscos do Projeto	38
2. Restrições do Projeto.....	38
3. Regras do Negócio	38
4. Requisitos Funcionais	39
5. Requisitos Não Funcionais	39
6. Mensagens do Sistema	40
7. Lista de Casos de Uso.....	41
8. Lista de Atores.....	41
9. Diagrama Geral de Caso de Uso	42
10. Diagrama de Classe	43
11. Diagrama de Processo.....	44
CAPÍTULO IV	45
ESPECIFICAÇÃO DE CASO DE USO.....	45
UC001 – NOTIFICAR ACESSOS	45
UC002 – ABRIR TRANCA POR RFID	47
UC003 – VERIFICAR IDENTIFICAÇÃO POR FACE.....	48
UC004 – ABRIR TRANCA	49
UC005 – EFETUAR LOGIN.....	50
UC006 – MANTER USUÁRIO	52
UC007 – MANTER CÓDIGO RFID	54
UC008 – MANTER ANEXO.....	56
CAPÍTULO V.....	58
MODELAGEM DE DADOS	58
1. Diagrama de Entidade Relacional.....	58
2. Diagrama de Entidade Relacional.....	59
3.1 Dicionário de Dados	60
3.1.1 Tabela TB_USUARIO	60
3.1.2 Tabela TB_PERFIL	61
3.1.3 Tabela TB_TIPO_ACAO	61
3.1.3 Tabela TB_AUDITORIA.....	62
3.1.4 Tabela TB_CONTROLE_ACESSO.....	62
3.1.5 Tabela TB_ANEXO	63
3.1.6 Tabela TB_CODIGO_ACESSO	64
3.1.7 Tabela TB_PARAMETRO	65
CAPÍTULO VI.....	66
PROTOTIPAÇÃO DO SISTEMA	66
1 Login <TL001>	66
1.1 Campos da Tela	67
1.2 Comandos da Tela	67
2 Registrar Administrador <TL002>	68
1.1 Campos da Tela	69

1.2 Comandos da Tela	69
3 Pesquisar Usuários <TL003>.....	70
1.1 Campos da Tela	71
1.2 Comandos da Tela	71
4 Cadastrar Usuário <TL004>	72
1.1 Campos da Tela	73
1.2 Comandos da Tela	73
5 Editar Usuário <TL005>.....	74
1.1 Campos da Tela	75
1.2 Comandos da Tela	75
6 Visualizar Usuário <TL006>	76
1.1 Campos da Tela	77
1.2 Comandos da Tela	77
7 Pesquisar Anexo <TL007>	78
1.1 Campos da Tela	79
1.2 Comandos da Tela	79
8 Cadastrar Anexo <TL008>	80
1.1 Campos da Tela	81
1.2 Comandos da Tela	81
9 Editar Anexo <TL009>.....	82
1.1 Campos da Tela	83
1.2 Comandos da Tela	83
10 Visualizar Anexo <TL010>	84
1.1 Campos da Tela	85
1.2 Comandos da Tela	85
11 Excluir Anexo <TL011>	86
1.1 Campos da Tela	87
1.2 Comandos da Tela	87
12 Pesquisar Código <TL012>.....	88
1.1 Campos da Tela	89
1.2 Comandos da Tela	89
13 Cadastrar Código <TL013>.....	90
1.1 Campos da Tela	91
1.2 Comandos da Tela	91
14 Visualizar Código <TL014>.....	92
1.1 Campos da Tela	93
1.2 Comandos da Tela	93
15 Excluir Código <TL015>	94
1.1 Campos da Tela	95
1.2 Comandos da Tela	95
CONCLUSÃO	96
REFERÊNCIAS.....	97
GLOSSÁRIO.....	100

ANEXO A<descrição do anexo> 101

APÊNDICE <descrição do apêndice> 102

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ARM	Advanced RISC Machine
CFTV	Circuito Fechado de Televisão
CPU	Central Processing Unit
GPU	Graphics Processing Unit
IOT	Internet of Things
PHP	Hypertext Preprocessor
PYPI	The Python Package Index
RAM	Random Access Memory
SoC	System on Chip
SQL	Structured Query Language

LISTA DE FIGURAS

Figura 1: Quadro Kanban.....	19
Figura 2: Arduino IDE 2.X	21
Figura 3: Exemplos de comandos utilizando composer	23
Figura 4: Firabase produtos.....	24
Figura 5: Exemplo de estrutura de projeto Laravel	26
Figura 6: Exemplo de código PHP	29
Figura 7: Exemplo de classe em Python	32
Figura 8: Comandos SQL.....	34
Figura 9: Projeto Aberto no Visual Studio Code	35
Figura 10: Dispositivo de reconhecimento facial	36
Figura 11: Diagrama de Caso de Uso.....	42
Figura 12: Diagrama de Classe.....	43
Figura 13: Diagrama de Processo	44
Figura 14: Diagrama de Entidade Relacional BrModelo	58
Figura 15: Diagrama de Entidade Relacional MySql	59
Figura 16: Tela de Login	66
Figura 17: Tela Registro de Administrador.....	68
Figura 18: Tela de Pesquisa de Usuários	70
Figura 19: Tela Cadastrar Usuário	72
Figura 20: Tela Editar Usuário.....	74
Figura 21: Tela Visualizar Usuário.....	76
Figura 22: Tela Pesquisar Anexo.....	78
Figura 23: Tela Cadastrar Anexo	80
Figura 24: Tela Editar Anexo	82
Figura 25: Tela Visualizar Anexo	84
Figura 26: Tela Excluir Anexo	86
Figura 27: Tela Pesquisar Código	88
Figura 28: Tela Cadastrar Código.....	90
Figura 29: Tela Visualizar Código.....	92
Figura 30: Tela Excluir Código	94

LISTA DE TABELA

Tabela 1: Riscos do Projeto.....	38
Tabela 2: Restrições do Projeto.....	38
Tabela 3: Regras de Negócio.....	38
Tabela 4: Requisitos Funcionais.....	39
Tabela 5: Requisitos Não Funcionais.....	39
Tabela 6: Mensagens do sistema.....	40
Tabela 7: Lista de Caso de Uso.....	41
Tabela 8: Lista de Atores	41
Tabela 9: Campos da tabela TB_USUARIO	60
Tabela 10: Campos da Tabela TB_PERFIL.....	61
Tabela 11: Campos da Tabela TB_TIPO_ACAO.....	61
Tabela 12: Campos da Tabela TB_AUDITORIA.....	62
Tabela 13: Campos da Tabela TB_CONTROLE_ACESSO.....	62
Tabela 14: Campos da Tabela TB_ANEXO.....	63
Tabela 15: Campos da Tabela TB_CODIGO_ACESSO.....	64
Tabela 16: Campos da Tabela TB_PARAMETRO	65

RESUMO

Atualmente o ser humano está na era da conectividade esta que é responsável pelo que podemos chamar “smartificação”, por exemplo, pense em como era a transferência de arquivos na década de 90, ou até mesmo na forma como era garantida a segurança das pessoas em certo ambiente.

Por meio dessa evolução é possível saber diferenciar uma pessoa de outra para, por exemplo, efetuar o desbloqueio de um *smartphone*, utilizar a digital na identificação de entrada em academias, embarque de passageiros identificados por fotos logo antes da viagem dentre outras tantas aplicações que facilitam e impactam positivamente para a segurança em meios digitais.

Fato é que o avanço tecnológico e o aumento do poder computacional permitem utilizar o reconhecimento de rostos mesmo em dispositivos que cabem em nossa mão para através da segurança implícita nesse tipo de técnica certificar-se que o usuário de algo relacionado a isso é realmente quem ele diz ser.

Palavras-Chave: [Reconhecimento facial, segurança, evolução]

ABSTRACT

Currently, human beings are in the era of connectivity, which is responsible for what we can call “smartification”, for example, think about how file transfer was in the 90s, or even how to safety of people in certain environment.

Through this Evolution, it is possible to know how to differentiate one person from another to, for example, unlock a smartphone, use the fingerprint to identify entry into gyms, board passagers identified by photos right before the trip, among many other applications that facilitate and positively impact security in digital media.

The fact is that technological advances and the increase in computational power make it possible to use face recognition even on devices that fit in our hand to, through the security implicit in this type of technique, make sure that the user something related to this is really who he is says to be.

Keywords: [Face recognition, security, evolution]

CAPÍTULO I

VISÃO INICIAL

1. Introdução

Ao mesmo passo, a evolução da IoT, por sua vez, tem permitido a construção das chamadas “cidades inteligentes”, interligadas por sistemas de alta tecnologia que cruzam dados e informações em questões de segundos, a fim de propiciar a vivência em uma cidade que busca reduzir gastos e facilitar a vida de seus moradores. Isso tudo ao mesmo tempo em que sistemas de reconhecimento facial permitem dar a sensação de maior segurança, proporcionando vigilância 24 horas por dia e a identificação de possíveis criminosos mais rapidamente do que a vigilância comum de rondas policiais. (DA SILVA, 2019)

Apesar de cidades cada vez mais inteligentes, dispositivos extremamente facilitadores da vida moderna ainda muitas vezes parte do pressuposto que fechaduras comuns são realmente seguras. É inegável seu valor para a evolução das barreiras físicas encontradas em muitos lugares, mas como em uma era de cidades inteligentes é possível continuar o uso de somente um tipo de fechadura para o controle de acesso residencial? Muitas das vezes sistemas robustos para cumprir tal papel são inviáveis para a maior parte da sociedade brasileira, a fim de permitir o uso da técnica de reconhecimento facial para o auxílio de mecanismos de fechadura pode-se recorrer a soluções *open source*.

Um dos assuntos mais atuais é o reconhecimento de imagens seja na identificação de doenças, tradução simultânea utilizando texto através da câmera de um dispositivo, identificação de sinais de fadiga entre outras inúmeras aplicações relacionadas ao reconhecimento de imagens, por meio da relevância deste tema é proposto para este trabalho falar da aplicação do reconhecimento facial para automação de tranca para o acesso do ambiente residencial.

2. Contextualização

O desenvolvimento da ciência e da tecnologia tem acarretado diversas transformações na sociedade contemporânea, refletindo em mudanças nos níveis econômico, político e social. É comum considerarmos ciência e tecnologia motores do progresso que proporcionam não só desenvolvimento do saber humano, mas, também, uma evolução real para o homem. (PINHEIRO, et al, 2007)

O uso de técnicas de reconhecimento facial tem se tornado cada vez mais difundida pelo mundo afora, seja em sistemas de segurança, embarque de passageiro ou até o acesso domiciliar.

Um dos fatores que diferenciam o reconhecimento facial dos outros sistemas biométricos reside no fato desta tecnologia poder ser utilizada para fins de vigilância. Uma das maiores vantagens desta tecnologia está na capacidade de capturar faces em locais públicos à distância, possibilitando assim minimizar as limitações jurídicas. (PINHEIRO, et al, 2007)

As vantagens são inúmeras no investimento desse tipo de tecnologia, que vão desde maior agilidade e capacidade de identificação de pessoas, acurácia ao identificar pessoas mesmo utilizando máscaras ou outros adereços, tendo ganhado popularidade no auxílio a segurança de grandes locais.

Segundo (Agência Brasil, 2019) “O Brasil tem 37 iniciativas em cidades adotando, de alguma maneira, tecnologias de reconhecimento facial. Mais da metade, 19, foram lançados no período de 2018 a 2019. Essas soluções, em geral, são empregadas nas áreas de segurança pública, transporte e controle de fronteiras”.

Na China, referência mundial no uso de big data e tecnologias de reconhecimento facial para segurança pública, a previsão é de que o número de câmeras de vigilância nas ruas das cidades chegue este ano a 560 milhões. Em 2018, eram 200 milhões. A vigilância, de acordo com o governo chinês, reduz a criminalidade no país. (MONEY TIMES, 2021)

3. Problemática

Normalmente para o acesso ao ambiente residencial usa-se chaves e tranca comuns que de certa forma provê um controle físico do local, mas como fazer se digamos quisesse descobrir se houve acesso ao local em um determinado horário? Será muito provável a sugestão para a adesão de um sistema CFTV, contudo, o grande armazenamento necessário caso queira armazenar todo o conteúdo gerado pelos dispositivos de monitoramento é muito grande.

Para uma solução coesa pode-se adotar a certa sugestão do registro de rostos ao se encontrar visíveis para algum dispositivo de captura de imagem pertencente ao sistema, como forma de criação dessas fotos deve-se utilizar o esquema de data/hora no nome do arquivo atestando que foi capturado em certa data. Além dessa precaução os metadados das fotos capturadas devem estar intactos para a manutenção da integridade dos fatos.

A justificativa para o uso desse tipo de tecnologia é citada por (MONEY TIMES, 2021) a criminalidade figura sempre entre as principais preocupações dos brasileiros. Somente em 2019, foram 47.796 mortes violentas intencionais (último dado anual oficial disponível do Fórum de Segurança Pública) com tendência de alta mesmo em ano de pandemia.

Mas seguindo outro caminho há países que investem no uso de tecnologias para a segurança da sociedade como um todo, tendo em vista que é um investimento a longo prazo assim como é o caso do investimento em educação de um país. O investimento em uma sociedade mais segura favorece o crescimento profissional e o aumento de bens de consumo, que por sua vez fortalece economicamente um país, ou seja, gerando o efeito em que todos ganham no processo.

Qual a maneira eficaz e com custo razoável para incrementar a segurança residencial utilizando o reconhecimento facial em um projeto *open source*?

4. Solução Proposta

Com o intuito do aumento de segurança, rastreabilidade do acesso ao ambiente domiciliar e utilizar tecnologias atuais e *open source* o sistema de reconhecimento facial para residências visa disponibilizar uma solução compacta para o aumento de segurança no acesso às áreas residências.

O objetivo é por meio de uma aplicação embarcada seja capaz de reconhecer e diferenciar os moradores cadastrados para o acesso à residência, que por meio do reconhecimento aciona uma tranca de porta e libera a entrada do morador com autorização para tal.

5. Objetivo Geral

Visando implementar um sistema que utiliza reconhecimento facial para permitir o acesso de moradores a uma área residencial aumentando a segurança através de uma solução utilizando tecnologias *opensource* de forma customizável.

Objetivos Específicos

Visando atender ao objetivo geral têm-se os seguintes objetivos específicos:

- Oferecer o controle de acesso através do reconhecimento da face;
- Cadastro de usuário facilitado;
- Informar quem está acessando o ambiente.

6. Estrutura do Projeto

Este projeto foi dividido em oito capítulos, onde se tentou apresentar de forma:

- O Capítulo I trata da introdução e contextualização do tema proposto para o trabalho;
- O Capítulo II especifica tecnologias e metodologias utilizadas no desenvolvimento do trabalho;

- O Capítulo III detalha o Documento de Visão, Regras de Negócio, Requisitos Funcionais e Não Funcionais;

CAPÍTULO II

METODOLOGIA APLICADA

Para o desenvolvimento de produtos de software é necessário primeiro o levantamento de requisitos para então definir quais as melhores tecnologias e atender o escopo do projeto, por meio desse premissa descreve-se as metodologias aplicadas e as técnicas para o levantamento dos requisitos.

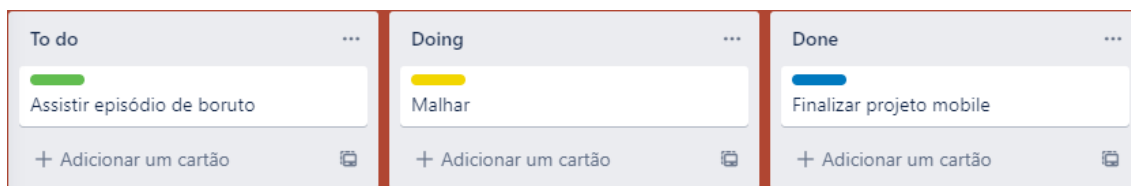
1. Metodologia e Modelo de Desenvolvimento

Kanban é um sistema visual de gestão de trabalho, que busca conduzir cada tarefa por um fluxo predefinido de trabalho. (TOTVS, 2021)

O Kanban, ou mais precisamente o “sistema Kanban para desenvolvimento de software” representa uma implementação mais direta dos princípios de Desenvolvimento Lean de Produtos para o desenvolvimento de software que os métodos ágeis tradicionais. Com foco consistente no fluxo e no contexto, o Kanban oferece uma abordagem menos prescritiva comparada ao Agile, e tem se tornado uma extensão popular dos métodos ágeis tradicionais como Scrum e XP. (BOEG, 2010)

Esse sistema de gerenciamento de tarefas traz uma grande vantagem que é a rastreabilidade da tarefa, pois é possível visualizar sua localização no processo de desenvolvimento da atividade referida ao cartão. Geralmente nota-se o uso das colunas nas seguintes ordens Para fazer, Fazendo, Terminado.

Figura 1: Quadro Kanban



Fonte: Autoria Própria, 2020

2. Técnicas de Levantamento de Requisitos

2.1 Brainstorming

Brainstorming é uma técnica para geração de ideias. Ela consiste em uma ou várias reuniões que permitem que as pessoas sugiram e explorem ideias. (DE MENDONÇA, 2014)

Brainstorming contém duas fases a fase de geração, onde as ideias são coletadas, e a fase de avaliação, onde as ideias coletadas são discutidas. Na fase de geração, as ideias não devem ser criticadas nem avaliadas. Cada ideia pode levar a novas ideias. A técnica de brainstorming leva a uma melhor compreensão do problema para todos e um sentimento de que todos cooperaram para atingir o objetivo. (DE MENDONÇA, 2014)

O Brainstorming foi aplicado na maneira em que será implementado o sistema para o reconhecimento facial na abertura de tranca. Por exemplo, o projeto pode ser implementado utilizando uma API hospedada em nuvem ou mesmo usar o próprio raspberry para o controle e armazenamento das informações, porém deve-se levar em conta o poder computacional que o raspberry disponibiliza.

2.2 Entrevista

A entrevista é uma das técnicas tradicionais mais simples de utilizar e que produz bons resultados na fase inicial de obtenção de dados. Convém que o entrevistador dê margem ao entrevistado para expor as suas ideias. É necessário ter um plano de entrevista para que não haja dispersão do assunto principal e a entrevista fique longa, deixando o entrevistado cansado e não produzindo bons resultados. (BEDANI, 2009)

Segundo (BEDANI, 2009), o planejamento da entrevista deve acontecer depois de outros documentos relacionados ao assunto foram recolhidos e devidamente analisados para a contextualizar os tópicos principais que serão abordados na entrevista.

2.3 Questionários

Questionários consistem em uma maneira de coletar os dados de forma a sofrer o mínimo de mudança possível, preservando a confiabilidade das respostas colhidas para a futura análise.

Segundo (BEDANI, 2009) o uso do questionário é indicado, por exemplo, quando há diversos grupos de usuários que podem estar em diversos locais diferentes do país. Neste caso, elaboram-se pesquisas específicas de acompanhamento com usuários selecionados, que a contribuição em potencial pareça mais importante, pois não seria prático entrevistar todas as pessoas em todos os locais.

Existem vários tipos de questionários que podem ser utilizados. Entre estes podemos listar: múltipla escolha, lista de verificação e questões com espaços em branco. O questionário deve ser desenvolvido de forma a minimizar o tempo gasto em sua resposta. (BEDANI, 2009)

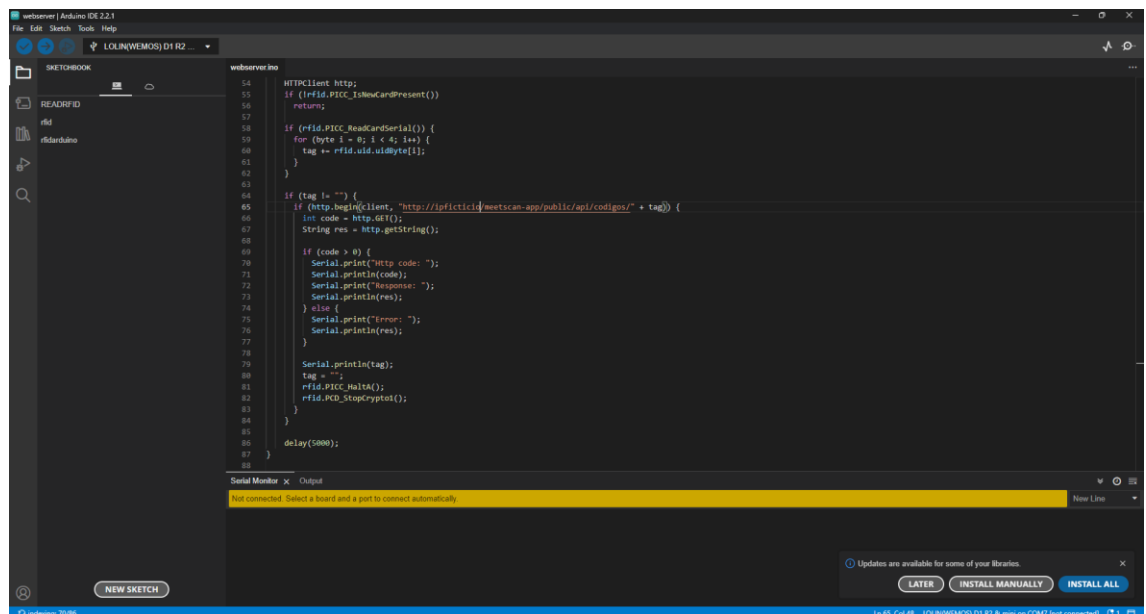
3. Principais tecnologias utilizadas

A grande maioria dos projetos é uma junção de duas ou mais tecnologias heterogêneas, que através de sua união permitem estabelecer a criação de um produto ou serviço no mercado. Visando transcrever a junção que irá compor este projeto são apresentadas as principais ferramentas utilizadas neste projeto.

3.1 Arduino IDE

Arduino IDE é o ambiente de desenvolvimento integrado para iteração de desenvolvedores com placas baseadas na plataforma Arduino, mas não limitando-se a isso podem ser configuradas placas baseadas na família ESP para a confecção de códigos na linguagem C.

Figura 2: Arduino IDE 2.X



Fonte: Autoria Própria, 2024

A IDE foi utilizada para o desenvolvimento de códigos C em conjunto com a placa NodeMCU ESP8266.

3.2 Composer

Composer é uma ferramenta para gerenciamento de dependências em PHP. Ele permite que você declare as bibliotecas das quais seu projeto depende e ele irá gerenciá-las (instalar/atualizar) para você.

Basicamente o Composer fornece ao programador uma forma de lidar com as inúmeras bibliotecas e pacotes que existem para o universo PHP, ao mesmo tempo fornece uma linha de comando amplamente utilizada para criação de projetos utilizando a solução proposta do trabalho que é o Laravel, permitindo criar com facilidades um novo projeto Laravel. (COMPOSER, 2024)

Composer não é um gerenciador de pacotes no mesmo sentido que o Yum ou Apt são. Sim ele lida com “pacotes” ou bibliotecas, mas os gerencia em um projeto base, instalando-os em um diretório, por exemplo, dentro do seu projeto. Por padrão, ele não instala nada globalmente. Assim, é uma dependência gerente. No entanto, ele suporta um projeto “global” por conveniência por meio do comando global. (COMPOSER, 2024)

Figura 3: Exemplos de comandos utilizando composer

A terminal window with a light gray background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains four lines of text, each starting with a comment in brown and a command in black. The commands are: 'composer create-project laravel/laravel example-app', 'composer install', 'composer --version', and 'composer install laravel/sanctum'.

```
# comando para criação de um projeto Laravel  
composer create-project laravel/laravel example-app  
  
# comando de instalação para criação da pasta de  
dependências chamada Vendor  
composer install  
  
# comando para verificar versão do composer  
composer --version  
  
# comando para instalar um pacote dentro de um projeto  
Laravel  
composer install laravel/sanctum
```

Fonte: Autoria Própria, 2024

3.3 C/C++

C é uma linguagem de programação projetada por Dennis Ritchie na Bell Labs. C é muito utilizado, simples e pode ser compilado para uma série de plataformas e sistemas operacionais. C é uma linguagem imperativa, com um pequeno número de palavras chave e um grande número de operadores matemáticos. C também é uma linguagem de programação de nível muito baixo, o que significa que pode ser comunicar diretamente com o *hardware*. (GITHUB, 2024)

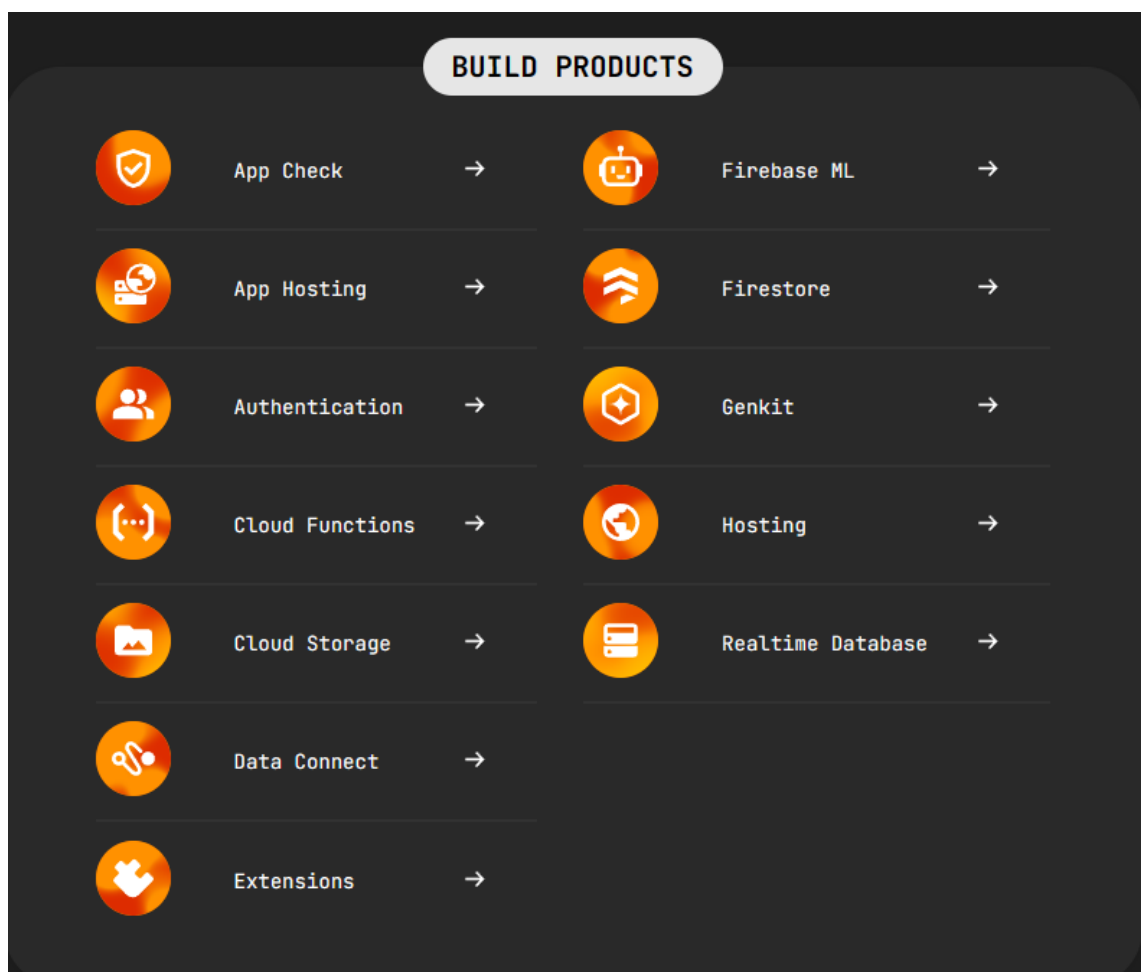
C é uma linguagem utilizada para a construção de outras linguagens de programação, por exemplo, Python, PHP, JavaScript, Julia Language, Lua, entre outras linguagens de programação.

O C foi utilizado para a programação da placa NodeMCU ESP8266 em conjunto com Arduino IDE para automação da placa.

3.4 Firebase

O Firebase é uma plataforma de *Backend-as-a-Service (BaaS)*, isto é, ela fornece a infraestrutura de *back-end* pronta para quem desenvolve aplicativos. Logo, ao usar o Firabase, as pessoas desenvolvedoras podem se concentrar na criação da aplicação em si e focar menos na parte de configuração e manutenção de servidores. (ALURA, 2023)

Figura 4: Firabase produtos



Fonte: FIREBASE, 2024

Embora tenha nascido para facilitar o desenvolvimento de aplicativos *mobile* hoje o firabase suporta aplicações *web*, *mobile* android, *mobile* ios e

cross platform flutter. Permitindo a(o) desenvolvedor(a) escolher quais serviços podem utilizar em suas aplicações.

Para este projeto o firabase foi utilizado para armazenar as fotos do usuários através do Cloud Store.

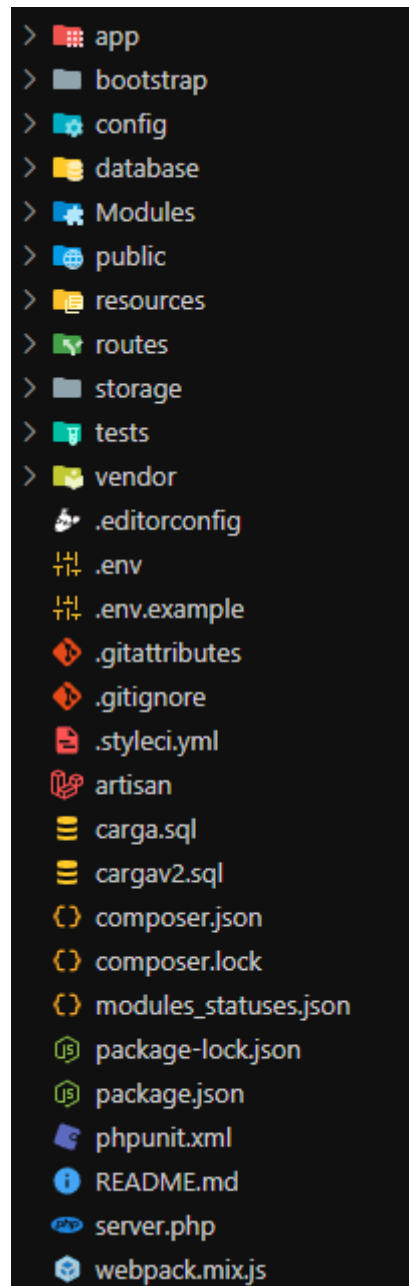
3.5 Laravel

Laravel é um framework de aplicação web com sintaxe expressiva e elegante. O Laravel é acessível, poderoso e fornece ferramentas necessárias para aplicações grandes e robustas. (LARAVEL, 2024)

Laravel é um *framework* para desenvolvimento de aplicações *web* e *apis* utilizando a linguagem de programação PHP. A ferramenta se propõe a oferecer um desenvolvimento de aplicações de forma ágil e segura, conta com uma vasta documentação para as diversas versões do *framework* e uma quantidade grande de exemplos disponíveis em muitas plataformas.

A proposta do *framework* é contar com um ambiente pronto para o desenvolvimento e publicação de aplicações *fullstack*. O Laravel conta com sistema de templates próprio, ORM, Middleware, Gestão de Login, Log para erros, Crons para execução de rotinas diárias, bem como a possibilidade do desenvolvimento de API com vários *clients* similar ao que faz o queridinho Spring na linguagem de programação Java.

Figura 5: Exemplo de estrutura de projeto Laravel



Fonte: Autoria Própria, 2024

O Laravel foi escolhido como plataforma para o desenvolvimento da aplicação web por ser uma estrutura organizada, fácil de trabalhar e rápida para a confecção de aplicações web que atendem o escopo deste projeto.

3.6 MySQL

O MySQL é o banco de dados de código aberto mais conhecido do mundo. De acordo com o DB-Engines, o MySQL é o segundo banco de dados mais popular, ficando atrás do Oracle Database. O MySQL alimenta muitas das aplicações mais acessadas, como Facebook, Twitter, Netflix, Uber, Airbnb, Shopify e Booking.com. (ORACLE, 2024)

Com o MySQL é de código aberto, ele inclui vários recursos desenvolvidos em estreita cooperação com os usuários há mais de 25 anos. Portanto, é muito provável que sua aplicação ou linguagem de programação favorita seja compatível com o MySQL Database. (ORACLE, 2024)

No sistema meetscan o MySQL será utilizado como bando de dados principal para o armazenamento das informações da versão web. Por meio da sua robustez e agilidade é possível modelar de inúmeras formas os objetos relacionais utilizados no sistema.

3.7 Open CV

OpenCV (Open Source Computer Vision Library) é uma biblioteca open source de visão computacional e machine learning. OpenCV foi construída para prover uma infraestrutura comum para aplicações de visão computacional e para acelerar o uso de aprendizado de máquina em produtos comerciais. O produto é licenciado por BSD-license, OpenCV torna fácil a utilização e modificação do código. (OPENCV, 2021)

A biblioteca tem mais de 2500 algoritmos otimizados, que inclui um compreensivo conjunto de visão computacional clássica e mais recente e também algoritmos de aprendizado de máquina. Estes algoritmos podem ser utilizados para detecção e reconhecimentos de faces, identificação de objetos, classificar ações humanas em vídeos, acompanhar movimento em câmera, acompanhar movimento de objetos, extrair modelos 3D de objetos, produzir nuvens 3D a partir de câmeras estéreo, junção de várias imagens para a produzir uma imagem de alta resolução da cena, procurar imagens similares em um conjunto de imagens, remover o efeito de olhos vermelhos causados

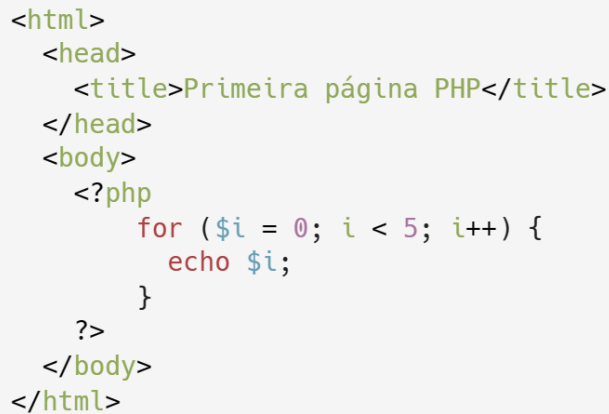
pelo flash da câmera em fotos, seguir o movimento dos olhos, reconhecer o cenário e estabelecer marcadores para sobrepô-lo com realidade aumentada, etc. (OPENCV, 2021)

Tem interface e suporte para a Linguagem C++, Java e MATLAB e suporta os sistemas operacionais Windows, Linux, Android e Mac OS. OpenCV é aplicado principalmente para aplicativos de visão computacional em tempo real e tira proveito das instruções MMX e SSE quando disponíveis. (OPENCV, 2021)

3.8 PHP

O PHP (um acrônimo recursivo para PHP: *Hypertext Preprocessor*) é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML. (PHP, 2024)

Figura 6: Exemplo de código PHP



```
<html>
  <head>
    <title>Primeira página PHP</title>
  </head>
  <body>
    <?php
      for ($i = 0; i < 5; i++) {
        echo $i;
      }
    ?>
  </body>
</html>
```

Fonte: Autoria Própria, 2024

O que distingue o PHP de algo como JavaScript no lado do cliente é que o código é executado no servidor, gerando o HTML, que é então enviado para o navegador. O navegador recebe os resultados da execução desse *script*, mas não sabe qual era o código fonte. Você pode inclusive configurar seu servidor web para processar todos os seus arquivos HTML com o PHP, e então não há como os usuários dizerem o que você tem na sua manga. (PHP, 2024)

A melhor coisa em usar o PHP é que ele é extremamente simples para um iniciante, mas oferece muitos recursos avançados para um programador profissional. Não tenha medo de ler a longa lista de recursos do PHP. Pode entrar com tudo o mais rápido que puder, e começar a escrever *scripts* simples em poucas horas. (PHP, 2024)

Por meio dessa flexibilidade na elaboração de sistemas web o PHP foi escolhido como linguagem principal da versão web do sistema meetscan, com

simplicidade e velocidade no desenvolvimento permite adequar rapidamente o sistemas a novas regras de negócio.

3.9 Python

A linguagem foi criada em 1990 por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI), e tinha originalmente foco em usuários como físicos e engenheiros. O Python foi concebido a partir de outra linguagem existente na época, chamada ABC. (BORGES, 2014)

A linguagem inclui diversas estruturas de alto nível (listas, dicionários, data/hora, complexos e outras) e uma vasta coleção de módulos prontos para uso, além de frameworks de terceiros que podem ser adicionados. Também inclui recursos encontrados em outras linguagens modernas, tais como geradores, introspecção, persistência, metaclasses e unidades de teste. (BORGES, 2014)

Como falado a linguagem Python tem suporte a diversos frameworks e bibliotecas de terceiros que podem ser facilmente encontrados em PYPI¹ que é o site para catalogação e distribuição de pacotes.

Python foi concebida para propósito geral, mas nos últimos anos ganhou muito destaque em certas áreas por conta de ferramentas de terceiros que podem ser instaladas através de seu gerenciador de pacotes chamado PIP. Os frameworks ou bibliotecas que mais se destacam são:

- Pandas²: utilizado para ciência de dados.
- Numpy³: é utilizado principalmente como um apoio para aplicações de machine learning, data science e outras áreas
- Flask⁴: utilizado para o desenvolvimento web

1 Site oficial: [PyPI · The Python Package Index](https://pypi.org/)

2 Pandas site oficial: [pandas - Python Data Analysis Library \(pydata.org\)](https://pandas.pydata.org/)

3 NumPy site oficial: [NumPy](https://numpy.org/)

4 Flask site oficial: [Welcome to Flask — Flask Documentation \(2.0.x\) \(palletsprojects.com\)](https://flask.palletsprojects.com/)

- FastApi⁵: Desenvolvimento de API REST assíncrona
- Kivy⁶: para o desenvolvimento de aplicações multiplataforma e que podem ser executados em Windows, Linux e Android.

Segundo (BORGES, 2014) além de ser utilizado como linguagem principal no desenvolvimento de sistemas, o Python também é muito utilizado como linguagem de script em vários softwares, permitindo automatizar tarefas e adicionar novas funcionalidades, entre eles: LibreOffice, PostgreSQL e Blender.


É possível integrar o Python a outras linguagens, como a Linguagem C e Fortran. Em termos gerais, a linguagem apresenta muitas similaridades com outras linguagens com outras linguagens dinâmicas, como Perl e Ruby. (BORGES, 2014)

Há também o suporte ao paradigma de orientação a objetos na linguagem, permitindo com que o desenvolvimento abarque as vantagens desse tipo de técnica. Abaixo está um exemplo de implementação de código utilizando o paradigma descrito.

5 FastApi site oficial: [FastAPI \(tiangolo.com\)](https://fastapi.tiangolo.com)

6 Kivy site oficial: [Kivy: Cross-platform Python Framework for NUI Development](https://kivy.org/doc/stable/index.html)

Figura 7: Exemplo de classe em Python



```
# Classe
class Pessoa:
    # Método construtor
    def __init__(self) -> None:
        self.__nome = None

    # GET
    def get_nome(self) -> str:
        return self.__nome

    # SET
    def set_nome(self, nome: str) -> None:
        self.__nome = nome
```

Fonte: Autoria Própria, 2021.

3.10 Reconhecimento Facial

O rosto de uma pessoa é o principal fator utilizado para reconhecê-la, no dia a dia. Para o ser humano, reconhecer um rosto familiar é uma tarefa trivial, realizada constantemente; nós somos capazes de guardar uma quantidade imensa de rostos em nossa memória, e mesmo assim reconhecer a maioria delas em uma pessoa em questão de décimos de segundos. (KUROIWA; CARRO, 2015)

Já com os algoritmos de reconhecimento facial são diferentes do reconhecimento feito pelo cérebro humano. Utilizando máquinas para fazer esse trabalho é necessário primeiro treinar o equipamento na distinção de faces e outros objetos que compõem a cena capturada, dessa maneira são

utilizados os pontos da face que além de diferenciar a pessoa a ser reconhecida de outros objetos também distingue o usuário do sistema.

O reconhecimento facial é uma técnica de biometria baseada em traços do rosto humano. O processo de reconhecimento é realizado a partir de pontos de medida do rosto, que fazem uma ligação algorítmica de traços e tamanhos, como exemplo se pode citar a distância exata entre nariz e orelhas, tamanho do crânio, arcada dentária, entre outros detalhes. (OKABE; CARRO, 2015)

3.11 SQL

SQL é a linguagem padronizada mais comum utilizada para acessar bancos de dados relacionais. Dependendo do seu ambiente de programação, você pode inserir diretamente (por exemplo, para gerar relatórios), incorporar instruções SQL ao código gravado em outra linguagem ou usar uma API específica da linguagem que oculte a sintaxe SQL. (ORACLE, 2024)

Com o SQL é possível realizar consultas, exclusões, alterações e cadastros de forma simplificado por meio da utilização direta da mesma ou mesmo utilizando os ORMs de linguagens e frameworks, através da utilização de um ORM é possível abstrair a sintaxe dos diferentes SGBDs presentes no mercado por uma escrita única.

Figura 8: Comandos SQL



```
INSERT INTO `meetscan`.`usuarios` (NO_USUARIO, DS_EMAIL,
DS_SENHA)
VALUES ('Teste', 'email@email.com',
'sadf0asodf0ajsd0fja0sdjfja0dsjf0j0asdf');

SELECT * FROM meetscan.usuarios;

UPDATE SET meetscan.usuarios
set no_usuario = 'Usuário inativo'
where ds_email = 'email@email.com';

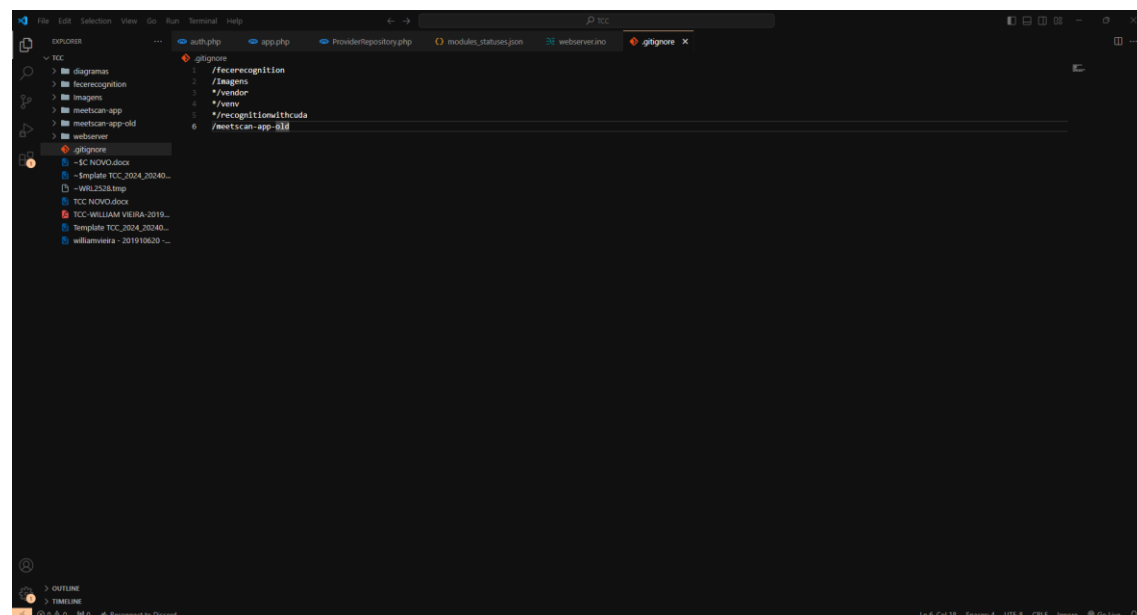
DELETE FROM meetscan.usuarios where ds_email =
'email@email.com';
```

Fonte: Autoria Própria, 2024.

3.12 Visual Studio Code

Visual Studio Code é um editor de código-fonte leve, mas poderoso, que é executado em sua área de trabalho e está disponível para Windows, macOS e Linux. Ele vem com suporte embutido para JavaScript, TypeScript e Node.js e tem um rico ecossistema de extensões para outras linguagens (como C++, C#, Java, Python, PHP, Go, .NET). (MICROSOFT, 2024, tradução)

Figura 9: Projeto Aberto no Visual Studio Code



Fonte: Autoria Própria, 2024.

Atualmente uma das ferramentas mais utilizadas do mercado o Visual Studio Code ou simplesmente VScode foi escolhido como o editor de código principal para a confecção da versão web do sistema meetscan.

4. Análise de Sistemas Existentes

A primeira alternativa ao sistema desenvolvido neste trabalho acadêmico tem o título “Controladora de acesso com reconhecimento facial, biometria e RFID Hikvision DS-K1T671MF-L”, trata-se de um aparelho compacto e de aparente fácil instalação.

Trabalha com o sistema operacional baseado em uma distribuição do kernel Linux, além do reconhecimento facial como controle de acesso o equipamento possui também o reconhecimento biométrico, teclado numérico, acesso por QRcode e cartão RFID.

Como cita (EMPORIO DA SEGURANÇA, 2020) o aparelho pode ser integrado ao controle de acesso de portarias de condomínios, já que com seu algoritmo de deep learning permite diferenciar pessoas de fotos tornando assim o controle de acesso confiável e seguro.

Figura 10: Dispositivo de reconhecimento facial



Fonte: HIKVISION, 2021

5. Cronogramas

As fases do projeto devem estar de acordo com o modelo de desenvolvimento selecionado no item 6 do Capítulo I.

	Semanas da Fase 1														
Especificação e documentação	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Elaboração do capítulo I	X	X	X	X	X	X									
Elaboração do capítulo II							X	X	X	X	X				
Elaboração da visão do sistema												X	X	X	X
Especificação de casos de uso															
Prototipação															

	Semanas da Fase 2														
Desenvolvimento,	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

CAPÍTULO III

VISÃO DO SISTEMA

1. Riscos do Projeto

Tabela 1: Riscos do Projeto

Nº	Descrição do Risco	Classificação
1	Falta de conhecimento em alguma das áreas	A
2	Variação no preço dos componentes utilizados no trabalho	A

Classificação: A – Alto; M – Médio e B – Baixo.

2. Restrições do Projeto

Tabela 2: Restrições do Projeto

Nº	Descrição da Restrição	Tipo
1	O projeto não contará com versão mobile	Tecnológica
2	O projeto não contará com Raspberry pi	Econômica
3	Orçamento limitado para execução do projeto	Econômica

Tipo: Econômica; Tecnológica, etc.

3. Regras do Negócio

Tabela 3: Regras de Negócio

Código	Nome	Descrição
RN001	Cadastro de fotos dos moradores	Para o melhor funcionamento do sistema deve ser cadastradas fotos do usuário com rosto em evidência, com fundo neutro e bem iluminadas.
RN002	Acesso a área administrativa o sistema	Apenas o usuário com perfil administrador deve ter acesso ao sistema.
RN003	Limitar a quantidade de anexos por usuário	Cada usuário poderá ter somente um código RFID associado a si.
RN004	Limitar código rfid	Cada usuário poderá ter somente um código associado a si.
RN005	Limitar quantidade de usuários administradores	O sistema deve permitir o cadastro de somente um usuário administrador.

4. Requisitos Funcionais

Tabela 4: Requisitos Funcionais

Código	Nome	Descrição
RF001	Notificar acessos	O caso de uso “Notificar acessos” deverá permitir ao ato o envio de notificação ao morador que interagir com o leitor de rfid.
RF002	Abrir tranca por rfid	O caso de uso “Abrir tranca por rfid” deverá permitir a entrada de morador com um cartão rfid válido e atrelado a um morador registrado no sistema.
RF003	Verificar identificação de face	O caso de uso “Verificar identificação de face” deverá permitir a entrada de moradores com faces cadastradas no sistema.
RF004	Abrir tranca	O caso de uso “Abrir tranca” deverá permitir o acesso do morador identificado através do reconhecimento de face.
RF005	Efetuar login	O caso de uso “Efetuar login” deverá permitir aos usuários administradores acesso a parte administrativa do sistema.
RF006	Manter usuário	O caso de uso “Manter usuário” deverá permitir o gerenciamento dos usuários através do sistema.
RF007	Manter código rfid	O caso de uso “Manter código rfid” deverá permitir o gerenciamento de códigos rfid através do sistema
RF008	Manter anexo	O caso de uso “Manter anexo” deverá permitir o gerenciamento de anexos através do sistema.

5. Requisitos Não Funcionais

Tabela 5: Requisitos Não Funcionais

Código	Nome	Descrição
RN001	Usabilidade	O sistema deve ser intuitivo no manuseio por parte do administrador, também presando pela simplicidade de uso por parte do morador cadastrado, visando a utilização e comprimento da funcionalidade disponibilizada.
RN002	Segurança	Os dados dos usuários devem ser armazenados de forma correta e segura, garantindo o cumprimento da lei geral de proteção de dados.

RN003	Disponibilidade	O sistema deve-se encontrar disponível, com possibilidade da alternância entre a tranca convencional e a tranca comum da porta.
RN004	Armazenar anexos	Os anexos devem ser armazenados em nuvem utilizando o Firebase.

6. Mensagens do Sistema

Tabela 6: Mensagens do sistema

Código	Descrição
MSG001	Não foi possível o cadastro de um novo usuário administrador
MSG002	Usuário administrador registrado com sucesso
MSG003	Credenciais inválidas, por favor, verifique!
MSG004	E-mail não encontrado, por favor, verifique!
MSG005	Usuário administrador já registrado
MSG006	Ocorreu um erro na atribuição de código para o usuário
MSG007	Código atribuído ao usuário com sucesso
MSG008	Não foi possível atualizar o código
MSG009	Código atualizado com sucesso
MSG010	Não foi possível excluir o código
MSG011	Código excluído com sucesso
MSG012	Não foi possível cadastrar o usuário
MSG013	Usuário cadastrado com sucesso
MSG014	Não foi possível atualizar o usuário
MSG015	Usuário editado com sucesso
MSG016	Não foi possível alterar o status do usuário
MSG017	Status do usuário alterado com sucesso
MSG018	Não foi possível achar o usuário atrelado ao código informado
MSG019	Bem-vindo(a) {nome do usuário}!
MSG020	O campo Nome é obrigatório
MSG021	O campo E-mail é obrigatório

MSG022	O campo Senha é obrigatório
MSG023	Deseja alterar o status do usuário?
MSG024	Campos obrigatórios não preenchidos
MSG025	Tem certeza que deseja excluir o anexo selecionado?

7. Lista de Casos de Uso

Tabela 7: Lista de Caso de Uso

Código	Nome	Descrição
UC001	Notificar acessos	Notifica os acessos dos moradores da residência através do telegrama
UC002	Abrir tranca por rfid	Permite o acesso a residência por meio de código rfid
UC003	Verifica identificação de face	Permite a identificação dos usuários registrados no sistema através da face
UC004	Efetuar login	Permite ao administrador acesso a parte administrativa do sistema
UC005	Manter usuário	Permite ao usuário administrador gerenciar os usuários no sistema
UC006	Abrir tranca	Permite ao morador o acesso a residência
UC007	Manter código rfid	Permite ao administrador gerenciar códigos rfid através do sistema
UC008	Manter anexo	Permite ao administrador gerenciar anexos através do sistema

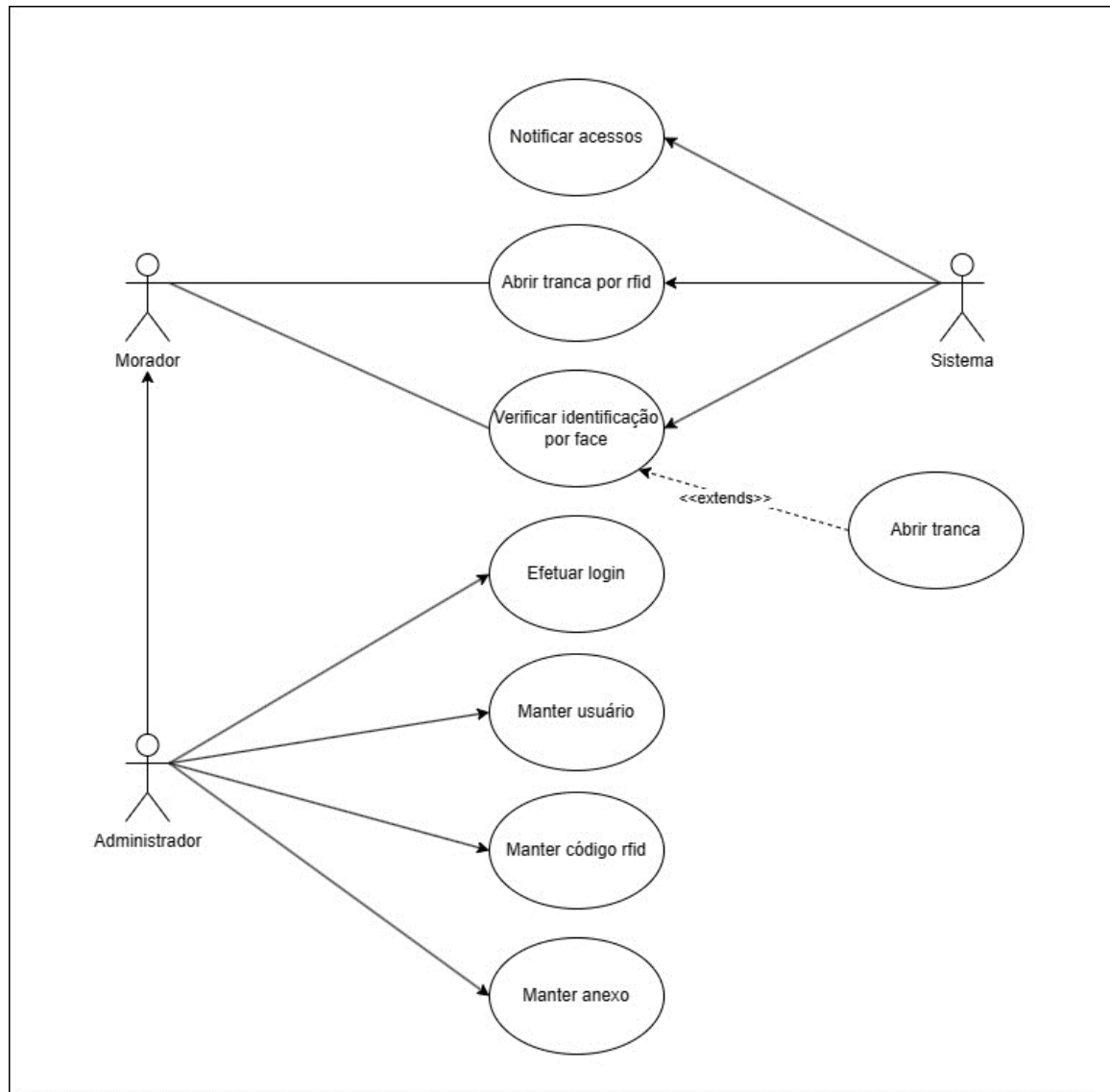
8. Lista de Atores

Tabela 8: Lista de Atores

Código	Nome	Descrição
001	Sistema	Responsável por identificar o usuário através do RFID, registra a entrada e saída do usuário.
002	Morador	Responsável por acessar a residência por meio do RFID.
003	Administrador	Responsável por gerenciar usuários, anexos e códigos RFID.

9. Diagrama Geral de Caso de Uso

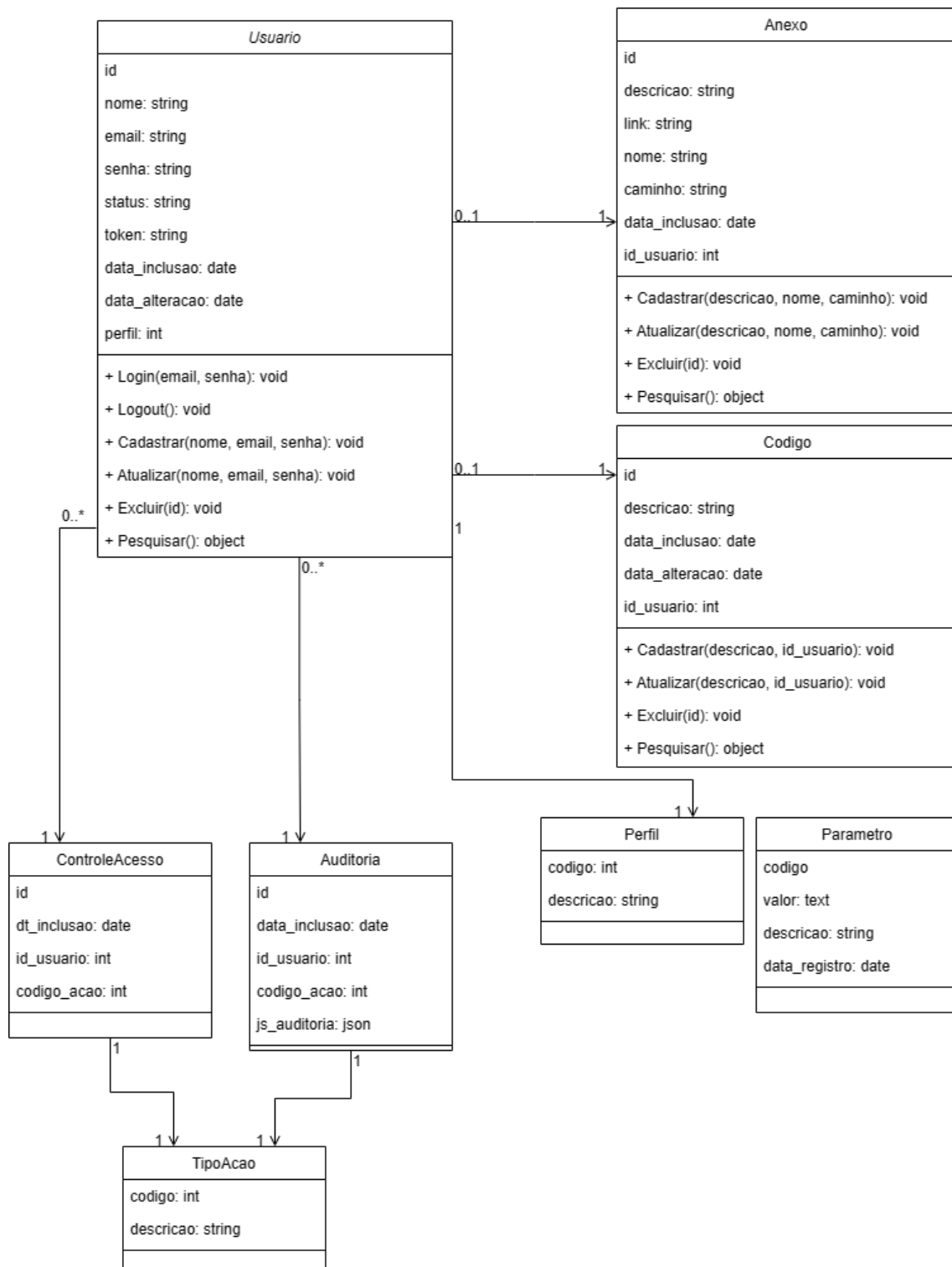
Figura 11: Diagrama de Caso de Uso



Fonte: Autoria Própria, 2024

10. Diagrama de Classe

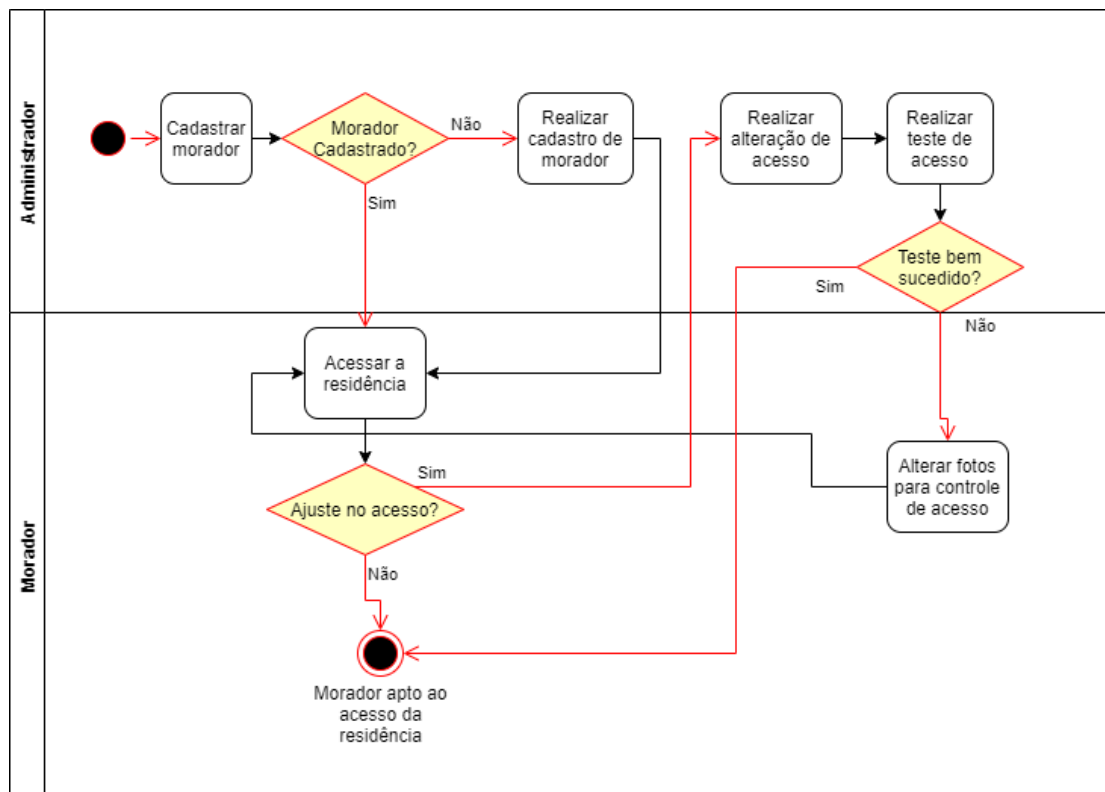
Figura 12: Diagrama de Classe



Fonte: Autoria Própria, 2022

11. Diagrama de Processo

Figura 13: Diagrama de Processo



Fonte: Autoria Própria, 2022

CAPÍTULO IV

ESPECIFICAÇÃO DE CASO DE USO

UC001 – NOTIFICAR ACESSOS

1. Descrição

Este caso de uso tem como finalidade a notificação de acesso dos usuários utilizando o telegrama.

2. Pré-condições

- Ser um usuário cadastrado no sistema;
- Possuir um código rfid associado;
- Possuir um cartão de acesso com código associado a um usuário.

3. Fluxos

3.1 Fluxo Principal – Notificar acesso por rfid

FP01. O caso de uso se inicia quando o ator passa um cartão com código rfid válido e cadastrado no sistema.

FP02. O sistema valida se o código está atrelado a algum usuário cadastrado no sistema.

FP03. Com o código rfid válido então o sistema deverá enviar uma mensagem com o nome do usuário dono do código rfid.

FP04. O caso de uso se encerra

3.2 Fluxos Alternativos

- N/A

3.3 Fluxos de Exceção

FE3.1.Código rfid inválido.

FE3.1.1. O sistema não deve enviar notificações caso o código do cartão rfid não esteja atrelado a um usuário cadastrado no sistema.

FE3.2. Indisponibilidade da api do telegram.

FE3.2.1. O sistema deverá apresentar mensagem caso o sistema não consiga utilizar a api do telegram.

4. Pós-condições

- N/A.

5. Pontos de Extensão

- N/A

6. Pontos de Inclusão

- N/A.

UC002 – ABRIR TRANCA POR RFID

1. Descrição

Este caso de uso tem como finalidade permitir o acesso de moradores a residência utilizando um cartão rfid.

2. Pré-condições

- O ator precisa ser registrado no sistema;
- O ator precisa ter um código rfid atrelado no sistema;
- O ator precisa ter um cartão com o mesmo código registrado no sistema.

3. Fluxos

3.1 Fluxo Principal – Abrir tranca por código rfid

FP01. O caso de uso se inicia quando o ator apresenta o cartão no leitor rfid.

FP02. O sistema realiza a leitura e identificação do cartão apresentado.

FP03. Caso o cartão esteja associado a um usuário o sistema deverá permitir a entrada do usuário.

FP04. O caso de uso se encerra.

3.2 Fluxos Alternativos

- N/A

3.3 Fluxos de Exceção

FE3.1. Código do cartão não encontrado.

FE3.1.1. O sistema deve notificar o ator com a mensagem. [MSG018]

4. Pós-condições

- N/A.

5. Pontos de Extensão

- UC002 – Consultar Dados do Projeto.

6. Pontos de Inclusão

- N/A.

UC003 – VERIFICAR IDENTIFICAÇÃO POR FACE

1. Descrição

Este caso de uso tem como finalidade permitir o acesso de moradores a residência utilizando o reconhecimento de face.

2. Pré-condições

- O ator deve ter uma foto registrada no sistema.
- O ator deve estar em local bem iluminado e sem a utilização de adereços para cobertura do rosto.

3. Fluxos

3.1 Fluxo Principal – Verificar identificação de face

FP01. O caso de uso se inicia quando o ator se aproxima do dispositivo identificador da face.

FP02. O sistema realiza a leitura da imagem e compara com a foto cadastrada no banco de dados.

FP03. O sistema reconhece a face do ator segue para o passo seguinte.

FP04. O caso de uso se encerra.

3.2 Fluxos Alternativos

- N/A

3.3 Fluxos de Exceção

FE3.1. Rosto não identificado.

FE3.1.1. O sistema deverá informar que o usuário é desconhecido.

4. Pós-condições

- N/A.

5. Pontos de Extensão

- UC004 – ABRIR TRANCA.

6. Pontos de Inclusão

- N/A.

UC004 – ABRIR TRANCA

1. Descrição

Este caso de uso é uma extensão da identificação por face, caso o ator seja identificado o caso de uso é acionado.

2. Pré-condições

- O ator precisa ter seu rosto reconhecido para acionar o caso de uso.

3. Fluxos

3.1 Fluxo Principal – Abrir tranca

FP01. O caso de uso se inicia quando o ator tem sua face reconhecida pelo dispositivo de reconhecimento.

FP02. O sistema deverá permitir a abertura da tranca e o acesso a residência.

FP03. O sistema deverá fechar a tranca após dez segundos.

FP04. O ator realiza o acesso ao ambiente residencial.

FP05. O caso de uso se encerra.

3.2 Fluxos Alternativos

- N/A

3.3 Fluxos de Exceção

- N/A

4. Pós-condições

- N/A.

5. Pontos de Extensão

- N/A.

6. Pontos de Inclusão

- N/A.

UC005 – EFETUAR LOGIN

1. Descrição

Este caso de uso tem como finalidade permitir o acesso de um usuário administrador a parte de gestão do sistema.

2. Pré-condições

- O ator deve estar cadastrado no sistema;
- O ator deve ter perfil de administrador;
- O ator deve ter credenciais de acesso válidas.

3. Fluxos

3.1 Fluxo Principal – Efetuar login

FP01. O caso de uso se inicia quando o ator acessa a página de login do sistema.

FP02. O sistema apresenta à tela de login. [TL001][RN002][RF005]

FP03. O ator preenche os campos de e-mail e senha, logo após seleciona a Logar.

FP04. Com todos os campos preenchidos corretamente, o sistema irá permitir o acesso do usuário ao sistema e irá exibir a mensagem **MSG019**, e redirecionará o usuário para a tela “Home” do sistema.

FP05. O caso de uso se encerra.

3.2 Fluxos Alternativos

- N/A

3.3 Fluxos de Exceção

FE3.1. Dados inválidos.

FE3.1.1. O sistema exibe a mensagem **MSG003** e retorna a tela de login.

FE3.2.Dados obrigatórios não informados.

FE3.2.1. O sistema exibe as mensagens **MSG021** e **MSG022** o sistema retorna para a tela de login.

4. Pós-condições

- Acessar a parte administrativa do sistema.

5. Pontos de Extensão

- N/A

6. Pontos de Inclusão

- N/A.

UC006 – MANTER USUÁRIO

1. Descrição

Este caso de uso tem como finalidade a gestão de usuários por parte do usuário administrador.

2. Pré-condições

- O ator precisa ser um usuário administrador;
- O ator deve estar logado no sistema;

3. Fluxos

3.1 Fluxo Principal – Cadastrar usuário

FP01. O caso de uso se inicia quando o ator seleciona o menu Usuários e a opção Novo Usuário.

FP02. O sistema apresenta a tela de cadastro de usuário [TL004] com os campos Nome, E-mail e Senha editáveis.

FP03. O ator preenche os campos e seleciona a opção Salvar.

FP04. O sistema realiza a validação dos campos, o sistema irá registrar o novo usuário no banco de dados e irá exibir a mensagem **MSG013**, confirmando o registro dos dados e redireciona o usuário para a tela de Pesquisa de Usuários [TL003].

FP05. O caso de uso se encerra.

3.2 Fluxos Alternativos

FA3.2.1. Editar Usuário.

FA3.2.1.1. Na tela [TL003] o ator deve acionar o botão com ícone editar.

FA3.2.1.2. O sistema redireciona para a tela de editar usuário. [TL005]

FA3.2.1.3. O ator pode editar informações do usuário e acionar a opção Salvar.

FA3.2.1.4. Com todos os campos preenchidos corretamente, o sistema irá atualizar o registro do usuário no sistema apresentando a mensagem **MSG00**, confirmando a alteração dos dados e redirecionará o ator para a tela de pesquisa de usuários [TL003].

FA3.2.1.5. O caso de uso se encerra.

FA3.2.2. Detalhar dados do usuário.

FA3.2.2.1. Na tela [TL003] o ator deve acionar o botão com ícone editar.

FA3.2.2.2. O sistema redireciona para a tela de visualizar usuário. [TL006]

FA3.2.2.3. O ator pode visualizar as informações do usuário cadastrado no sistema.

FA3.2.2.4. Ao acionar a opção Voltar o sistema deve redirecionar o ator para a tela de pesquisa de usuários. [TL003]

FA3.2.2.5. O caso de uso se encerra.

FA3.2.3. Alterar status do usuário (exclusão lógica).

FA3.2.3.1. Na tela [TL003] o ator deve acionar o botão com ícone editar.

FA3.2.3.2. O sistema apresenta um pop-up com a mensagem **MSG023**, juntamente com as opções Sim e Não.

FA3.2.3.3. Se o ator aciona a opção Sim o sistema altera o status do usuário e apresenta a mensagem **MSG017**.

FA3.2.3.4. Se o ator aciona a opção Não o sistema se mantém na tela [TL003].

FA3.2.3.4. O caso de uso se encerra.

3.3 Fluxos de Exceção**FE3.1.E-mail já utilizado.**

FE3.1.1. O sistema deve apresentar a mensagem **MSG00**, caso o e-mail informado no cadastro do usuário já esteja cadastrado no sistema.

FE3.2.Dados obrigatórios não informados.

FE3.2.1. O sistema exibe a mensagem **MSG024** e retorna ao passo anterior.

4. Pós-condições

- N/A.

5. Pontos de Extensão

- N/A.

6. Pontos de Inclusão

- N/A

UC007 – MANTER CÓDIGO RFID

1. Descrição

Este caso de uso tem como finalidade a gestão de códigos rfid associados aos usuários cadastrados no sistema.

2. Pré-condições

- O ator precisa estar logado no sistema;
- O sistema deve possuir usuários cadastrados;
- Ao menos um usuário não deve estar associado a um código rfid.

3. Fluxos

3.1 Fluxo Principal – Cadastrar Código

FP01. O caso de uso se inicia quando o ator seleciona a opção Códigos.

FP02. O ator deve acionar a opção Novo Código e o sistema redireciona para a tela de cadastro de código [TL013].

FP03. O ator preenche os campos e aciona a opção Salvar.

FP04. Com todos os campos preenchidos corretamente, o sistema irá registrar os dados no banco de dados e irá exibir a mensagem **MSG007**, confirmando o registro dos dados e redirecionará o usuário para tela de pesquisa de códigos [TL012].

FP05. O caso de uso se encerra.

3.2 Fluxos Alternativos

FA3.2.1. Editar.

FA3.2.1.1. Na tela [TL012] o ator deve acionar o botão de editar dados do código.

FA3.2.1.2. O sistema redireciona o ator para a tela editar código.

FA3.2.1.3. O ator deve preencher os campos obrigatórios e acionar o botão Salvar.

FA3.2.1.4. Com todos os campos preenchidos corretamente, o sistema irá alterar o registro de código selecionado no banco de dados e irá exibir a mensagem **MSG007**, confirmando a alteração dos dados e retornando para a tela [TL012].

FA3.2.2. Detalhar dados do código.

FA3.2.2.1. Na tela [TL012] o ator deve acionar o botão de detalhar dados do código.

FA3.2.2.2. O sistema redireciona o ator para a tela de visualizar código. [TL014]

FA3.2.2.3. Ao acionar a opção de voltar o ator é redirecionado para a tela [TL012].

FA3.2.2.4. O caso de uso se encerra.

FA3.2.3. Excluir código.

FA3.2.3.1. Na tela [TL012] o ator deve acionar o botão de exclusão do código.

FA3.2.3.2. O sistema exibe um pop-up com a mensagem **MSG00**, e as opções Sim e Não;

FA3.2.3.3. Se o ator acionar a opção Sim o sistema irá realizar a exclusão física e apresentar a mensagem **MSG011**.

FA3.2.3.4. Se o ator acionar a opção Não o sistema fechará o *pop-up* e permanecerá na tela [TL012].

FA3.2.3.5. O caso de uso se encerra.

FA3.2.4. Voltar

FA3.2.6.1. O ator solicita a opção Voltar.

FA3.2.6.2. O sistema para a tela [TL012].

3.3 Fluxos de Exceção

FE3.1.Dados obrigatórios não informados.

FE3.3.1. O sistema exibe a mensagem **MSG024** e retorna ao passo anterior.

4. Pós-condições

- N/A.

5. Pontos de Extensão

- N/A.

6. Pontos de Inclusão

- N/A

UC008 – MANTER ANEXO

1. Descrição

Este caso de uso tem como finalidade a gestão de anexos dentro do sistema.

2. Pré-condições

- O ator precisa estar logado no sistema;
- O usuário deve ser relacionado a somente um anexo.

3. Fluxos

3.1 Fluxo Principal – Cadastrar Anexo

FP01. O caso de uso se inicia quando o ator aciona a opção Anexos.

FP02. O sistema apresenta à tela de pesquisa de anexos [TL007].

FP03. O ator preenche deve acionar a opção Novo Anexo.

FP04. O sistema apresenta a tela de cadastro de anexos [TL008].

FP05. O ator deve preencher os campos e acionar a opção Salvar.

FP06. Com todos os campos preenchidos corretamente, o sistema irá registrar os dados no banco de dados e irá exibir a mensagem **MSG00**, confirmando o registro dos dados e redirecionará o usuário para a tela de pesquisa de anexos [TL007].

FP07. O caso de uso se encerra.

3.2 Fluxos Alternativos

FA3.2.1. Editar Anexo.

FA3.2.1.1. O caso de uso se inicia quando o ator aciona a opção Anexos.

FA3.2.1.2. O sistema apresenta à tela de pesquisa de anexos [TL007].

FA3.2.1.3. O ator aciona a opção de editar anexos.

FA3.2.1.4. O sistema redireciona para a tela de editar anexos [TL009].

FA3.2.1.5. O ator preenche todos os campos e aciona a opção Salvar.

FA3.2.1.4. Com todos os campos preenchidos corretamente, o sistema irá alterar o anexo no banco de dados e armazenar a nova imagem no *firebase* apresentando a mensagem **MSG00**, confirmando a alteração dos dados e redirecionará o ator para a tela de pesquisa de anexos [TL007].

FA3.2.2. Excluir Anexo.

FA3.2.2.1. O caso de uso se inicia quando o ator aciona a opção Anexos.

FA3.2.2.2. O sistema apresenta à tela de pesquisa de anexos [TL007].

FA3.2.2.3. O ator aciona o botão de excluir que irá apresentar um *pop-up* com a mensagem **MSG00**.

FA3.2.2.4. Ao acionar a opção Sim o sistema confirma a exclusão e apresenta a mensagem **MSG025**.

FA3.2.2.5. Ao acionar a opção Não o sistema fecha o *pop-up* e se mantém na tela de pesquisa de anexos [TL007].

FA3.2.3. Visualizar Anexo.

FA3.2.3.1. O caso de uso se inicia quando o ator aciona a opção Anexos.

FA3.2.3.2. O sistema apresenta à tela de pesquisa de anexos [TL007].

FA3.2.3.3. O ator deve acionar a opção para visualizar o anexo.

FA3.2.3.4. O sistema redireciona o ator para a tela de visualização de anexo [TL010].

FA3.2.3.4. Ao acionar a opção Voltar o ator é redirecionado para a tela de pesquisa de anexos [TL007].

FA3.2.4. Voltar

FA3.2.6.1. O ator solicita a opção Voltar.

FA3.2.6.2. O sistema retorna para a tela de pesquisar anexos [TL007].

3.3 Fluxos de Exceção

FE3.1.Dados obrigatórios não informados.

FE3.3.1. O sistema exibe a mensagem **MSG024** e retorna ao passo anterior.

4. Pós-condições

- N/A.

5. Pontos de Extensão

- N/A.

6. Pontos de Inclusão

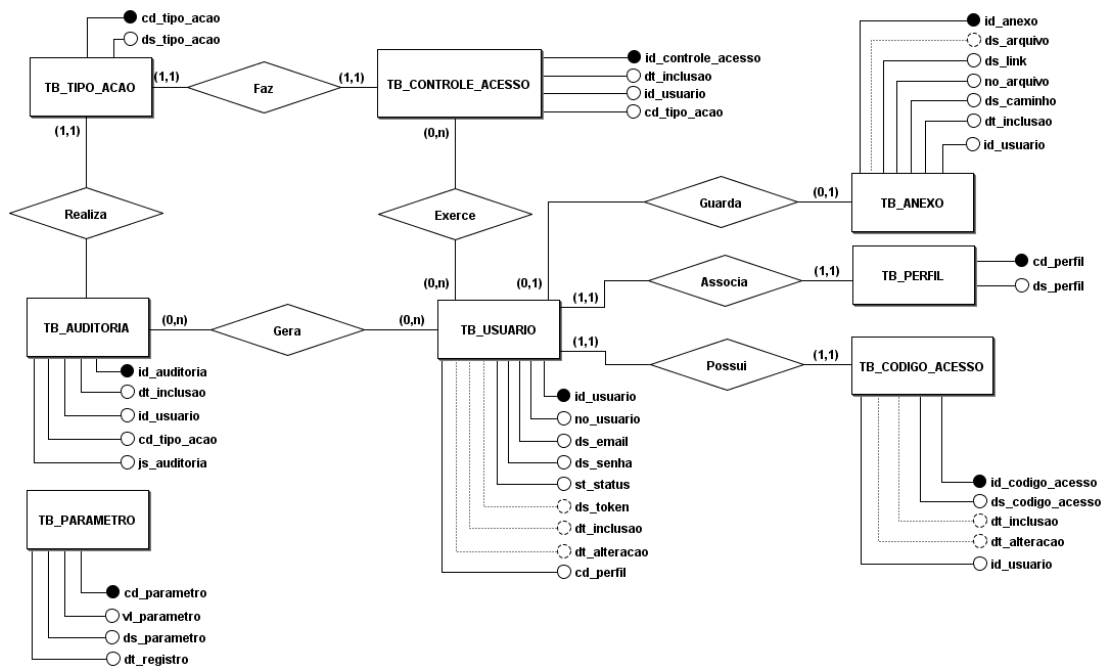
- N/A

CAPÍTULO V

MODELAGEM DE DADOS

1. Diagrama de Entidade Relacional

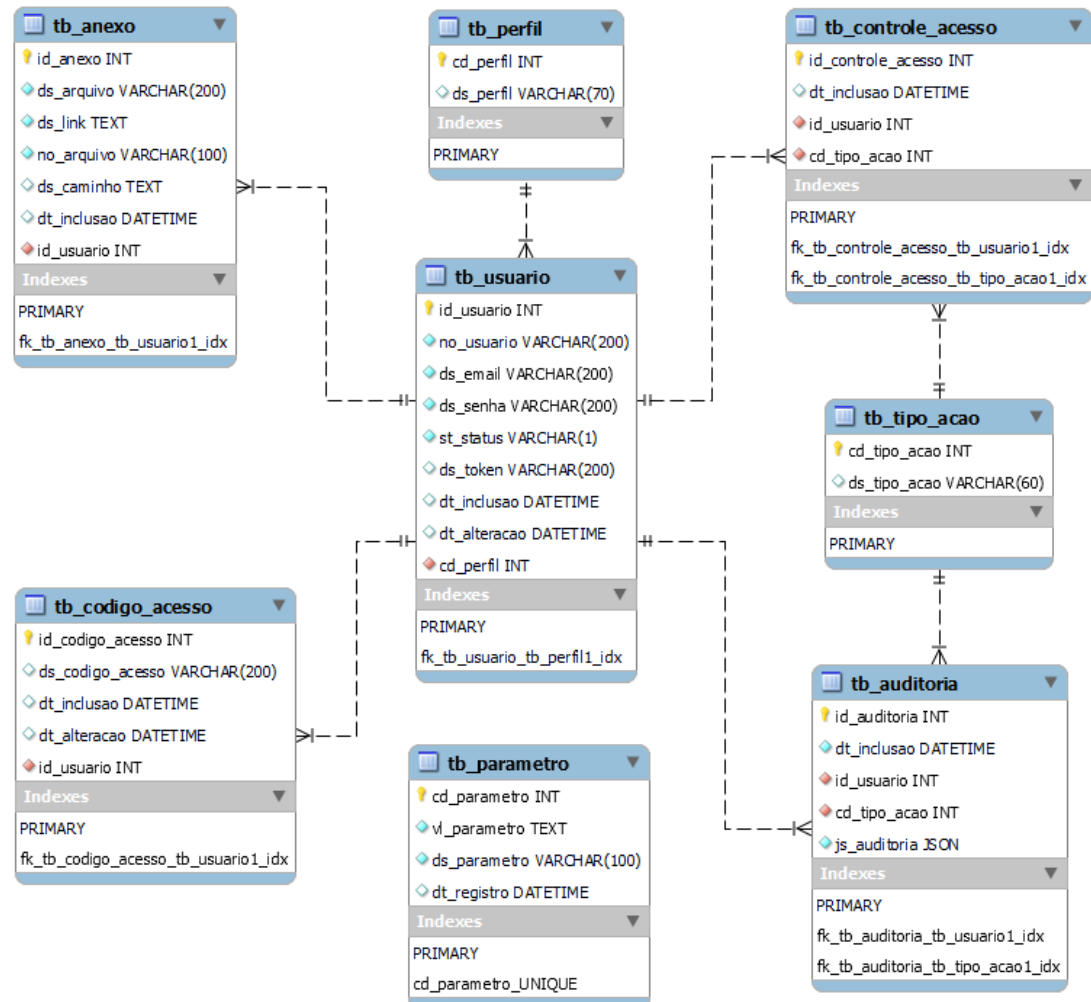
Figura 14: Diagrama de Entidade Relacional BrModelo



Fonte: Autoria Própria, 2024.

2. Diagrama de Entidade Relacional

Figura 15: Diagrama de Entidade Relacional MySQL



Fonte: Autoria Própria, 2024.

3.1 Dicionário de Dados

3.1.1 Tabela TB_USUARIO

Tabela 9: Campos da tabela TB_USUARIO

Campo	Tipo/Tamanho	Obrigatório	Comentário
ID_USUARIO	INTEGER	S	PK da tabela gerada automaticamente após cada inserção
NO_USUARIO	VARCHAR/200	S	Responsável por armazenar o nome do usuário
DS_EMAIL	VARCHAR/200	S	Responsável por armazenar o e-mail do usuário
DS_SENHA	VARCHAR/200	S	Responsável por armazenar a senha do usuário
ST_STATUS	VARCHAR/1	S	Responsável por armazenar o status do usuário. A – Ativo, I - inativo
DS_TOKEN	VARCHAR/200	N	Responsável por armazenar o token de acesso do usuário.
DT_INCLUSAO	DATETIME	N	Responsável por armazenar a data de registro do usuário.
DT_ALTERACAO	DATETIME	N	Responsável por armazenar a data de alteração do usuário.
CD_PERFIL	INT	S	Responsável por armazenar o código o perfil do usuário.
RELACIONAMENTOS			
Tabela	Descrição		
TB_PERFIL	O campo que relaciona as tabelas de perfil e usuário é o CD_PERFIL que é o código do perfil do usuário armazenado no momento do cadastro.		

3.1.2 Tabela TB_PERFIL

Tabela 10: Campos da Tabela TB_PERFIL

Campo	Tipo/Tamanho	Obrigatório	Comentário
CD_PERFIL	INTEGER	S	PK da tabela, deve ser única e inserida por meio de carga
DS_PERFIL	VARCHAR/70	S	Responsável por armazenar a descrição do perfil
RELACIONAMENTOS			
Tabela	Descrição		

3.1.3 Tabela TB_TIPO_ACAO

Tabela 11: Campos da Tabela TB_TIPO_ACAO

Campo	Tipo/Tamanho	Obrigatório	Comentário
CD_TIPO_ACAO	INTEGER	S	PK da tabela, deve ser inserida por meio de carga
DS_TIPO_ACAO	VARCHAR/60	S	Responsável por armazenar a descrição do tipo de ação
RELACIONAMENTOS			
Tabela	Descrição		

3.1.3 Tabela TB_AUDITORIA

Tabela 12: Campos da Tabela TB_AUDITORIA

Campo	Tipo/Tamanho	Obrigatório	Comentário
ID_AUDITORIA	INTEGER	N	PK da tabela gerada automaticamente após cada inserção
DT_INCLUSAO	DATETIME	S	Responsável por armazenar a data de criação do registro
ID_USUARIO	INTEGER	S	Responsável por armazenar o identificador do usuário que realizou a ação
CD_TIPO_ACAO	INTEGERS	S	Responsável por armazenar o código da ação realizada pelo usuário
JS_AUDITORIA	JSON	S	Responsável por armazenar o registro afetado pela ação do usuário
RELACIONAMENTOS			
Tabela	Descrição		
TB_USUARIO	Relacionamento que define a qual usuário pertence o registro de auditoria.		
TB_TIPO_ACAO	Relacionamento que define o tipo de ação realizada pelo usuário no registro de auditoria.		

3.1.4 Tabela TB_CONTROLE_ACESSO

Tabela 13: Campos da Tabela TB_CONTROLE_ACESSO

Campo	Tipo/Tamanho	Obrigatório	Comentário
ID_CONTROLE_ACESSO	INTEGER	S	PK da tabela gerada automaticamente após cada inserção

DT_INCLUSAO	DATETIME	S	Responsável por armazenar a data de criação do registro
ID_USUARIO	INTEGER	S	FK da tabela responsável por identificar o usuário detentor do registro
CD_TIPO_ACAO	INTEGER	S	FK da tabela responsável por identificar a ação realizada pelo usuário
RELACIONAMENTOS			
Tabela	Descrição		
TB_USUARIO	Relacionamento que define qual o usuário realizou a ação.		
TB_TIPO_ACAO	Relacionamento que define o tipo de ação realizada pelo usuário.		

3.1.5 Tabela TB_ANEXO

Tabela 14: Campos da Tabela TB_ANEXO

Campo	Tipo/Tamanho	Obrigatório	Comentário
ID_ANEXO	INTEGER	S	PK da tabela gerada automaticamente após cada inserção
DS_ARQUIVO	VARCHAR/200	S	Responsável por armazenar a descrição do arquivo
DS_LINK	TEXT		Responsável por armazenar o link do arquivo armazenado em nuvem
NO_ARQUIVO	VARCHAR/100		Responsável por armazenar o nome do arquivo enviado pelo usuário
DS_CAMINHO	TEXT		Responsável por armazenar o

			caminho da imagem armazenada localmente
DT_INCLUSAO	DATETIME		Responsável por armazenar a data de criação do registro
ID_USUARIO	INTEGER	S	FK da tabela responsável por identificar a ação realizada pelo usuário
RELACIONAMENTOS			
Tabela	Descrição		
TB_USUARIO	Relacionamento que define a qual usuário o anexo pertence.		

3.1.6 Tabela TB_CODIGO_ACESSO

Tabela 15: Campos da Tabela TB_CODIGO_ACESSO

Campo	Tipo/Tamanho	Obrigatório	Comentário
ID_CODIGO_ACESSO	INTEGER	S	PK da tabela gerada automaticamente após cada inserção
DS_CODIGO_ACESSO	VARCHAR/200	S	Responsável por armazenar o código de acesso RFID
DT_INCLUSAO	DATETIME	S	Responsável por armazenar a data de criação do registro
DT_ALTERACAO	DATETIME	S	Responsável por armazenar a data de alteração do registro
ID_USUARIO	INTEGER	S	FK da tabela responsável por identificar o usuário detentor do registro
RELACIONAMENTOS			

Tabela	Descrição
TB_USUARIO	Relacionamento que define a qual usuário o código de acesso pertence.

3.1.7 Tabela TB_PARAMETRO

Tabela 16: Campos da Tabela TB_PARAMETRO

Campo	Tipo/Tamanho	Obrigatório	Comentário
CD_PARAMETRO	INTEGER	S	PK da tabela responsável por identificar o parâmetro
VL_PARAMETRO	TEXT	S	Responsável por armazenar o valor do parâmetro
DS_PARAMETRO	VARCHAR/100	S	Responsável por armazenar a descrição do parâmetro
DT_REGISTRO	DATETIME	S	Responsável por armazenar a data de criação do parâmetro
RELACIONAMENTOS			
Tabela	Descrição		

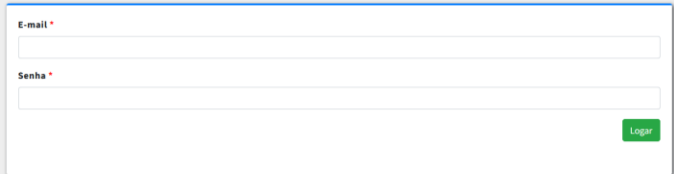
CAPÍTULO VI

PROTOTIPAÇÃO DO SISTEMA

1 Login <TL001>

Tela responsável por permitir aos usuários administradores realizarem o acesso a parte administrativa do sistema

Figura 16: Tela de Login



The image shows a login form centered on a light gray background. The form is a white rectangle with a thin blue border. It contains two input fields: the top one is labeled 'E-mail' with a red asterisk, and the bottom one is labeled 'Senha' with a red asterisk. To the right of the 'Senha' field is a green button with the text 'Logar' in white.

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	E-mail	Alfanumérico	200	N/A	S	N/A	S	ds_email	S
002	Senha	Alfanumérico	200	N/A	S	N/A	S	ds_senha	N

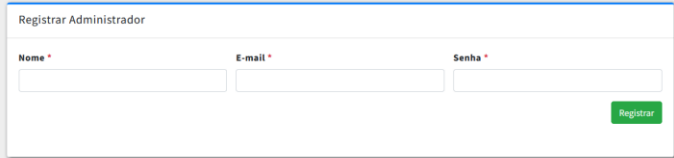
1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Logar	Realiza a checagem das credenciais para permitir o acesso do administrador no sistema	Todos os campos obrigatórios devem ser preenchidos

2 Registrar Administrador <TL002>

Tela responsável por permitir o cadastro de um novo usuário administrador no sistema.

Figura 17: Tela Registro de Administrador



O formulário, intitulado "Registrar Administrador", está centralizado na tela. Ele contém três campos de entrada para "Nome", "E-mail" e "Senha", cada um com um asterisco vermelho indicando obrigatoriedade. Os campos são alinhados horizontalmente. À direita dos campos, há um botão verde com o texto "Registrar" em branco.

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Nome	Alfanumérico	200	N/A	S	N/A	S	no_usuario	S
002	E-mail	Alfanumérico	200	N/A	S	N/A	S	ds_email	S
003	Senha	Alfanumérico	200	N/A	S	N/A	S	ds_senha	S

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Registrar	Grava um novo usuário administrador na base	Todos os campos obrigatórios devem ser preenchidos

3 Pesquisar Usuários <TL003>

Tela responsável por permitir ao administrador a pesquisa e gestão de usuários.

Figura 18: Tela de Pesquisa de Usuários

MeetScan Usuários Anexos Codigos Sair

Pesquisa de usuários




Nome E-mail Perfil Status

Seleção Seleção

Novo Usuário Pesquisar

Resultado da pesquisa

Mostrando 10 registros por página Pesquisar

#	Nome	E-mail	Perfil	Data Cadastro	Status	Ações
4	William Vieira	vieiraork@gmail.com	ADMINISTRADOR	01/04/2024	Ativo	  

Mostrando página 1 de 1

Anterior 1 Próximo

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Nome	Alfanumérico	100	N/A	N	N/A	S	Nome do usuário	S
002	E-mail	Alfanumérico	100	N/A	N	N/A	S	E-mail do usuário	S
003	Perfil	Combo box	N/A	N/A	N	N/A	S	Código do perfil associado ao usuário	S
004	Status	Combo box	N/A	N/A	N	N/A	S	Tipo de status do usuário	S

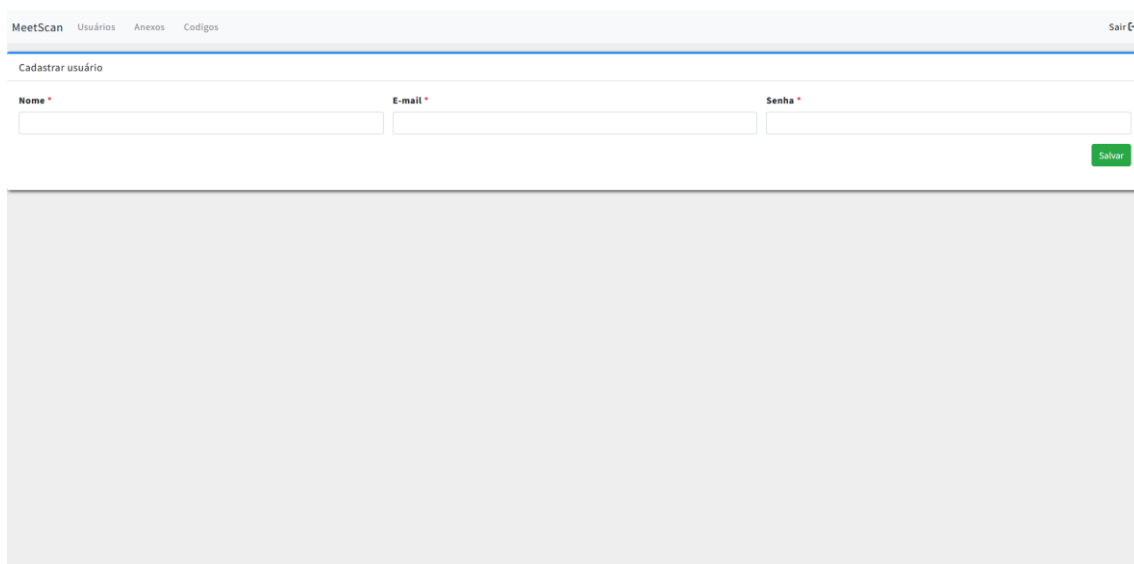
1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Novo	Redireciona o usuário para funcionalidade de cadastro de usuário	
02	Pesquisar	Pesquisa usuários cadastrados com base nas opções de filtro	Para apresentação de resultados na tabela há necessidade de usuários cadastrados
03	Visualizar	Visualiza as informações dos usuários	Há necessidade de usuários cadastrados no sistema
04	Editar	Redireciona o para funcionalidade de editar	Há necessidade de usuários cadastrados no sistema
05	Alterar status	Alterar o status do usuário (exclusão lógica)	O usuário não deve inativar o próprio cadastro.

4 Cadastrar Usuário <TL004>

Tela responsável por permitir ao administrador a pesquisa e gestão de usuários.

Figura 19: Tela Cadastrar Usuário



A imagem mostra a interface de usuário para o cadastro de um novo usuário no sistema MeetScan. No topo, há uma barra de navegação com o nome do sistema 'MeetScan' e links para 'Usuários', 'Anexos' e 'Códigos'. No canto superior direito, há um link 'Sair' com uma seta para fora. Abaixo da barra, o título da página é 'Cadastrar usuário'. O formulário principal contém três campos de entrada: 'Nome *', 'E-mail *' e 'Senha *'. Cada campo é precedido por seu respectivo rótulo e um asterisco indicando obrigatoriedade. Os campos são retangulares e brancos. À direita dos campos, há um botão verde com o texto 'Salvar' em branco. Abaixo do formulário, há uma grande área cinza vazia, possivelmente para uma mensagem de confirmação ou para uma imagem de perfil.

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Nome	Alfanumérico	200	N/A	S	N/A	S	no_usuario	S
002	E-mail	Alfanumérico	200	N/A	S	N/A	S	ds_email	S
003	Senha	Alfanumérico	200	N/A	S	N/A	S	ds_senha	S

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Salvar	Grava um novo usuário no sistema	Todos os campos obrigatórios devem estar preenchidos. O usuário deve conter um e-mail único no sistema.

5 Editar Usuário <TL005>

Tela responsável por permitir ao administrador a alteração das informações relacionados ao usuário cadastrado no sistema.

Figura 20: Tela Editar Usuário

The screenshot displays the 'Editar usuário' (Edit User) interface within the MeetScan application. The top navigation bar includes 'MeetScan', 'Usuários', 'Anexos', 'Codigos', and a 'Sair' (Logout) button. The main heading is 'Editar usuário'. The form contains three input fields: 'Nome *' (Name) with the value 'William Vieira', 'E-mail *' (Email) with the value 'vieiraork@gmail.com', and 'Senha' (Password). A green 'Salvar' (Save) button is positioned to the right of the password field. Below the form is a large, empty gray rectangular area.

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Nome	Alfanumérico	200	N/A	S	N/A	S	no_usuario	S
002	E-mail	Alfanumérico	200	N/A	S	N/A	S	ds_email	S
003	Senha	Alfanumérico	200	N/A	S	N/A	S	ds_senha	S

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Editar	Altera o registro do usuário no sistema	Todos os campos obrigatórios devem ser preenchidos

6 Visualizar Usuário <TL006>

Tela responsável por permitir ao administrador a visualização das informações dos usuários.

Figura 21: Tela Visualizar Usuário

MeetScan Usuários Anexos Codigos Sair ↗

Usuário
Nome: William Vieira
Perfil: ADMINISTRADOR
Status: Ativo
Data de cadastro: 19/03/2024 20:56:05
Data alteração:

Voltar

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Nome	Alfanumérico	200	N/A	S	N/A	N	no_usuario	S
002	Perfil	Texto	70	N/A	S	N/A	N	ds_perfil	S
003	Status	Texto	1	N/A	S	N/A	N	st_status	S
004	Data de Cadastro	Data/hora	N/A	N/A	N	N/A	N	dt_inclusao	S
005	Data Alteração	Data/hora	N/A	N/A	N	N/A	N	dt_alteracao	S

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Voltar	Retorna o usuário para a tela de pesquisa	N/A

7 Pesquisar Anexo <TL007>

Tela responsável por permitir ao administrador a pesquisa e gestão dos anexos associados aos usuários.

Figura 22: Tela Pesquisar Anexo

MeetScan Usuários Anexos Codigos Sair

Nome do arquivo Data início Data fim

dd / mm / aaaa dd / mm / aaaa

Novo Anexo Pesquisar

Resultado

Mostrando 10 registros por página Pesquisar

#	Descrição	Data registro	Usuario	Ações
3	Anexo para o usuário teste	01/04/2024	William Vieira	

Mostrando página 1 de 1

Anterior 1 Proximo

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Nome do Arquivo	Alfanumérico	200	N/A	N	N/A	S	no_arquivo	S
002	Data inicio	Data	N/A	N/A	N	N/A	S	dt_inclusao	S
003	Data fim	Data	N/A	N/A	N	N/A	S	dt_inclusao	S

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Novo Anexo	Redireciona o usuário para a tela de cadastro de anexo	N/A
02	Pesquisar	Realiza a pesquisa de anexos com base nos filtros preenchidos	N/A
03	Visualizar	Visualiza detalhadamente as informações do anexo	N/A
04	Editar	Redireciona o usuário para a tela de alteração de anexo	N/A
05	Excluir	Realiza a exclusão do registro no banco	N/A

8 Cadastrar Anexo <TL008>

Tela responsável por permitir ao administrador cadastrar novos anexos associados aos usuários.

Figura 23: Tela Cadastrar Anexo

A interface de usuário para o cadastro de anexos no sistema MeetScan. No topo, há uma barra de navegação com o nome do sistema 'MeetScan' e links para 'Usuários', 'Anexos' e 'Códigos'. Um botão 'Sair' com uma seta para fora está no canto superior direito. Abaixo, o título da seção é 'Cadastro de arquivo'. O formulário principal contém três campos: 'Descrição do arquivo' (um campo de texto), 'Usuário' (um menu suspenso com a opção 'Selecione') e 'Arquivo' (um campo com um botão 'Procurar...' e o texto 'Nenhum arquivo selecionado.'). Um botão verde 'Salvar' está localizado à direita do formulário. A área principal da tela abaixo do formulário é cinza e vazia.

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Descrição do arquivo	Alfanumérico	200	N/A	N	N/A	S	ds_arquivo	S
002	Usuário	Combo box	N/A	N/A	S	N/A	S	id_usuario	S
003	Arquivo	Arquivo	N/A	N/A	S	N/A	S	no_arquivo	S

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Salvar	Grava uma referência do anexo no banco e realiza o upload da imagem no firebase.	Todos os campos obrigatórios devem ser preenchidos. Usuário selecionado não deve ter anexos já associados a ele.

9 Editar Anexo <TL009>

Tela responsável por permitir ao administrador cadastrar novos anexos associados aos usuários.

Figura 24: Tela Editar Anexo

MeetScan Usuários Anexos Códigos Sair ↗

Editar de arquivo

Detalhes de Usuário

Nome do Usuário: William Vieira

Descrição do arquivo

Teste inserção

Arquivo *

Procurar... Nenhum arquivo selecionado.

Salvar

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Descrição do arquivo	Alfanumérico	200	N/A	N	N/A	S	ds_arquivo	S
002	Usuário	Combo box	N/A	N/A	S	N/A	S	id_usuario	S
003	Arquivo	Arquivo	N/A	N/A	S	N/A	S	no_arquivo	S

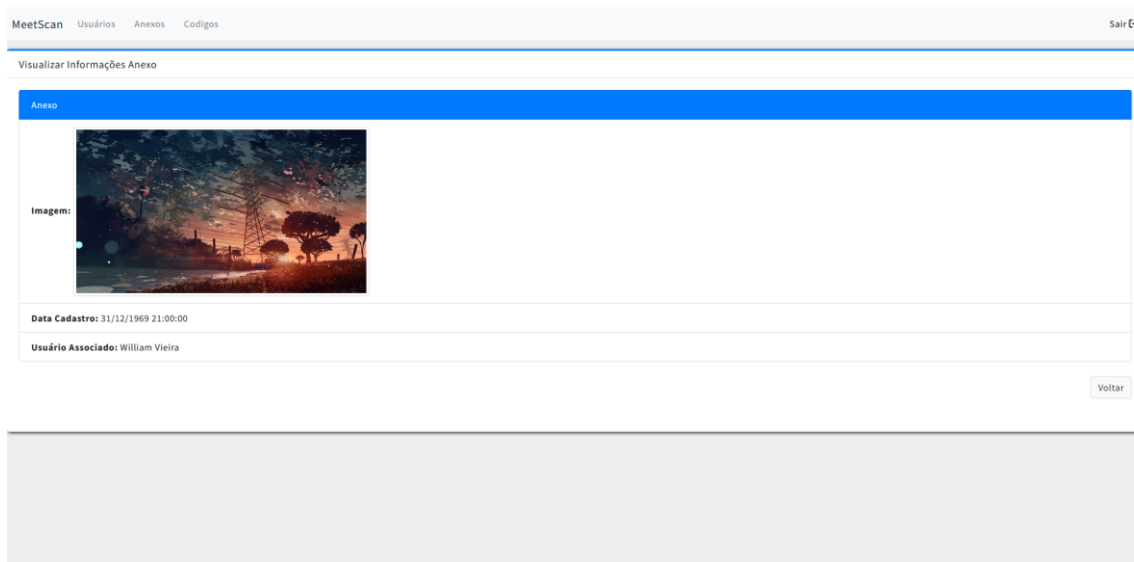
1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Salvar	Excluir o anexo já cadastrado e realiza o upload do novo arquivo para o firebase.	Todos os campos obrigatórios devem ser preenchidos São aceitos arquivos com as extensões png e jpg

10 Visualizar Anexo <TL010>

Tela responsável por permitir ao administrador visualizar os anexos cadastrados no sistema.

Figura 25: Tela Visualizar Anexo



Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Imagem	Imagem	N/A	N/A	S	N/A	N	ds_link	S
002	Data Cadastro	Data/Hora	N/A	N/A	S	N/A	N	dt_inclusao	S
003	Usuário Associado	Alfanumérico	200	N/A	S	N/A	N	no_usuario	S

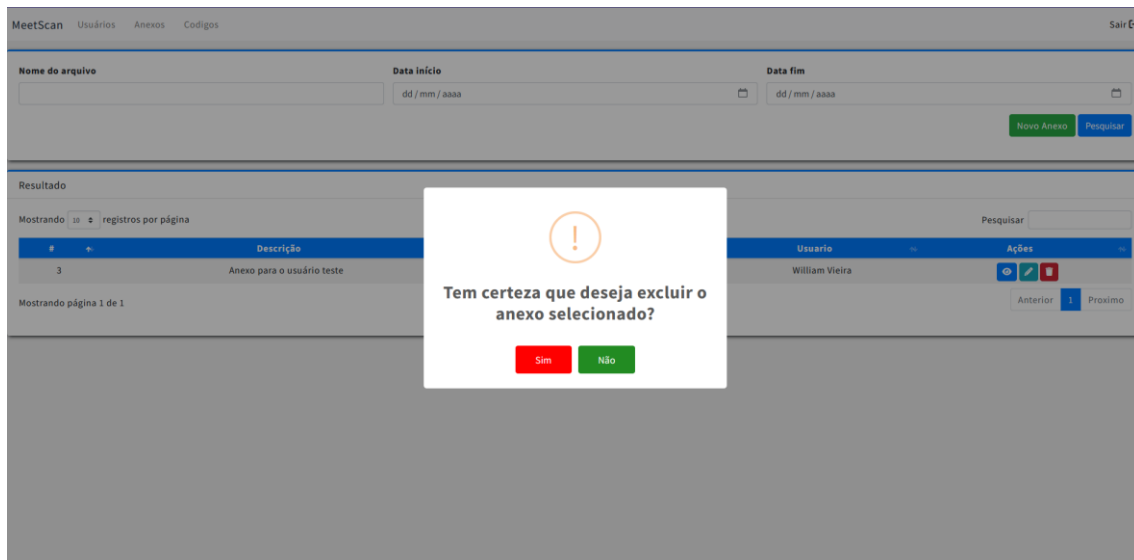
1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Voltar	Redireciona o usuário para tela de pesquisa de anexos	N/A

11 Excluir Anexo <TL011>

Tela responsável por permitir ao administrador excluir anexos cadastrados no sistema.

Figura 26: Tela Excluir Anexo



Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Sim	Confirma a ação de exclusão do anexo	N/A
02	Não	Cancela a ação de exclusão e se mantém na página de pesquisa de anexo	N/A

12 Pesquisar Código <TL012>

Tela responsável por permitir ao administrador realizar a pesquisa e gestão dos códigos rfid.

Figura 27: Tela Pesquisar Código

MeetScan Usuários Anexos Códigos Sair

Pesquisar código

Código Usuário Data início Data fim

dd/mm/aaaa dd/mm/aaaa

Novo Código Pesquisar

Resultado da pesquisa

Mostrando 10 registros por página Pesquisar

#	Código	Data Registro	Data Alteração	Usuário	Ação
1	950394859	01/04/2024		William Vieira	

Mostrando página 1 de 1 Anterior Proximo

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Código	Alfanumérico	200	N/A	N	N/A	S	ds_codigo_acesso	S
002	Usuário	Alfanumérico	200	N/A	N	N/A	S	no_usuario	S
003	Data início	Data	N/A	N/A	N	N/A	S	dt_inclusao	S
004	Data fim	Data	N/A	N/A	N	N/A	S	dt_inclusao	S

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Novo Código	Redireciona o usuário para tela de cadastro de código	N/A
02	Pesquisar	Realiza a pesquisa com base nos campos de filtro preenchidos pelo usuário	N/A
03	Visualizar	Redireciona o usuário para a tela de visualizar detalhes do código	N/A
04	Excluir	Redireciona o usuário para a funcionalidade de excluir código	N/A

13 Cadastrar Código <TL013>

Tela responsável por permitir ao administrador realizar o cadastro de um novo código rfid.

Figura 28: Tela Cadastrar Código

A imagem mostra a interface de usuário para o cadastro de um novo código rfid no sistema MeetScan. No topo, há uma barra de navegação com o nome do sistema 'MeetScan' e links para 'Usuários', 'Anexos' e 'Códigos'. No canto superior direito, há um link 'Sair' com uma seta para fora. Abaixo da barra de navegação, o título da página é 'Cadastrar código'. O formulário principal contém dois campos obrigatórios, indicados por um asterisco vermelha: 'Código' e 'Usuário'. O campo 'Código' é uma caixa de texto vazia. O campo 'Usuário' é um menu suspenso com o texto 'Selecione' e uma seta para baixo. À direita dos campos, há dois botões: 'Voltar' (cinza) e 'Salvar' (verde). Abaixo do formulário, há uma grande área cinza vazia, provavelmente para uma imagem ou mensagem de confirmação.

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Código	Alfanumérico	200	N/A	S	N/A	S	ds_codigo_acesso	S
002	Usuário	Numérico Inteiro	N/A	N/A	S	N/A	S	id_usuario	S

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Salvar	Grava um novo código no sistema	Todos os campos obrigatórios devem ser preenchidos Cada usuário pode ser associado a um único código
02	Voltar	Retorna para a tela de pesquisa de código	N/A

14 Visualizar Código <TL014>

Tela responsável por permitir ao administrador visualizar o código rfid atrelado a um usuário.

Figura 29: Tela Visualizar Código

MeetScan Usuários Anexos Códigos Sair ↗

Código
Código: 950394859
Usuário: William Vieira
Data de cadastro: 01/04/2024 20:41:16
Data alteração:

Voltar

Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Código	Alfanumérico	200	N/A	S	N/A	N	ds_codigo_acesso	S
002	Usuário	Alfanumérico	200	N/A	S	N/A	N	no_usuario	S
003	Data de Cadastro	Data/Hora	N/A	N/A	S	N/A	N	dt_inclusao	S
004	Data Alteração	Data/Hora	N/A	N/A	N	N/A	N	dt_alteracao	S

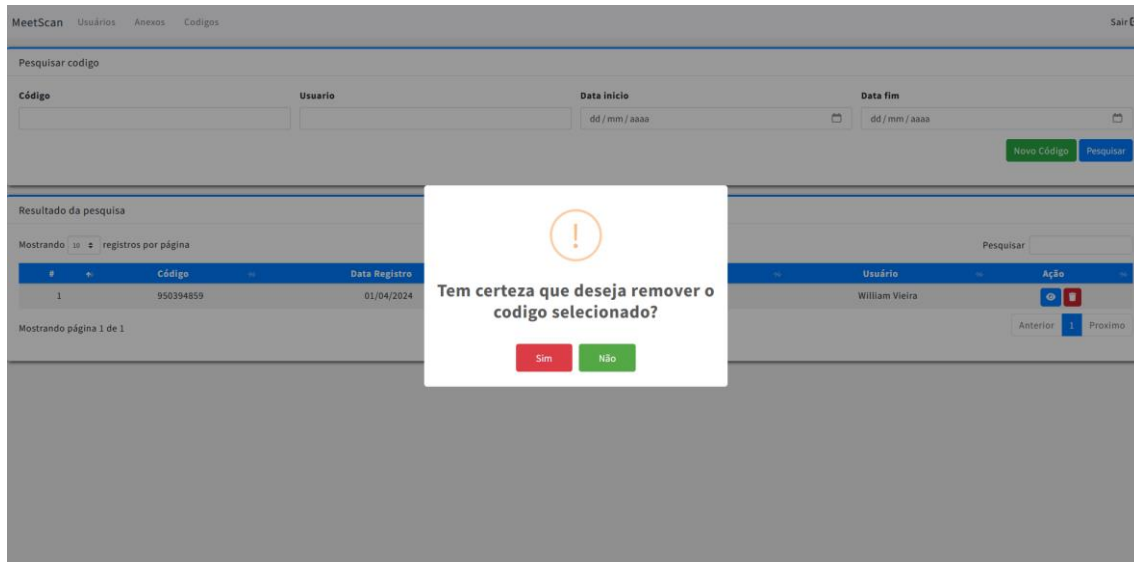
1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Voltar	Retorna o usuário para a tela de pesquisa de códigos	N/A

15 Excluir Código <TL015>

Tela responsável por permitir ao administrador efetuar a exclusão de códigos rfid atrelados aos usuários.

Figura 30: Tela Excluir Código



Fonte: Autoria Própria, 2024

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Sim	Confirma a ação de exclusão do código	N/A
02	Não	Cancela a ação de exclusão e se mantém na página de pesquisa de código	N/A

CONCLUSÃO

[Informar as considerações finais relacionado ao trabalho]

REFERÊNCIAS

Dissertações, teses, trabalhos individuais, etc.

BORGES, Luiz Eduardo. **Python para desenvolvedores: aborda Python 3.3**. Novatec Editora, 2014.

FACELI, Katti. **Combinação de métodos de inteligência artificial para fusão de sensores**. 2001. Tese de Doutorado. Universidade de São Paulo.

Artigo de periódico

BOEG, Jesper. Kanban em 10 passos. Tradução de Leonardo Campos, Marcelo Costa, Lúcio Camilo, Rafael Buzon, Paulo Rebelo, Eric Fer, Ivo La Puma, Leonardo Galvão, Thiago Vespa, Manoel Pimentel e Daniel Wildt. C4Media, 2010.

DA SILVA, Rosane Leal; DA SILVA, Fernanda dos Santos Rodrigues. Reconhecimento Facial e segurança pública: os perigos do uso da tecnologia no sistema penal seletivo brasileiro. In: **5 Congresso Internacional de Direito e Contemporaneidade: mídias e direitos da sociedade em rede**. 2019.

DE MENDONÇA, Ricardo Augusto Ribeiro. Levantamento de requisitos no desenvolvimento ágil de software. **Semana da Ciência e Tecnologia da PUC Goiás**, p. 12, 2014.

KUROIWA, Bruno Tsutsumi; CARRO, Silvio Antonio. Detecção de intrusão com reconhecimento facial em imagens geradas por câmeras de segurança. In: **Colloquium Exactarum**. ISSN: 2178-8332. 2015. p. 58-72.

OKABE, Rogerio Kazuhiro; CARRO, Silvio Antonio. Reconhecimento facial em imagens capturadas por câmeras digitais de rede. In: **Colloquium Exactarum**. ISSN: 2178-8332. 2015. p. 106-119.

PINHEIRO, Nilcéia Aparecida Maciel; SILVEIRA, Rosemari Monteiro Castilho Foggiatto; BAZZO, Walter Antonio. Ciência, tecnologia e sociedade: a relevância do enfoque CTS para o contexto do ensino médio. **Ciência & Educação (Bauru)**, v. 13, p. 71-84, 2007.

Em meio eletrônico

Agencia Brasil. Tecnologias de reconhecimento facial são usadas em 37 cidades do país. Disponível em: <[Tecnologias de reconhecimento facial são usadas em 37 cidades no país | Agência Brasil \(ebc.com.br\)](https://www.ebc.com.br/tecnologia/37-cidades-reconhecimento-facial)>. Acesso em: out. 2021.

ALURA. O que é Firebase? Para que serve, principais característica e um Guia dessa ferramenta Google. Disponível em: <[O que é Firebase? Um Guia dessa ferramenta Google | Alura](https://www.alura.com.br/artigo/o-que-e-firebase)>. Acesso em: mai. 2024.

BEDANI, Janaína. Engenharia de Software 2 - Técnicas para levantamento de Requisitos. 2009. Disponível em: <[Técnicas para levantamento de Requisitos \(devmedia.com.br\)](https://devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos)> . Acesso em: nov. 2021

COMPOSER. Introduction. Disponível em: <[Introduction - Composer \(getcomposer.org\)](https://getcomposer.org/doc/01-introduction.md)>. Acesso em: mai. 2024.

CRISTIAN AQUINO. Como a tecnologia pode ajudar a segurança pública nas cidades. Disponível em: <[Cristian Aquino: Como a tecnologia pode ajudar a segurança pública nas cidades \(moneytimes.com.br\)](https://moneytimes.com.br/seguranca/como-a-tecnologia-pode-ajudar-a-seguranca-publica-nas-cidades)>. Acesso em: out. 2021.

EMPORIO DA SEGURANÇA. Leitor Reconhecimento Facial DS-K1T671 | HIKVISION, 2020. Disponível em: <[Leitor Reconhecimento Facial DS-K1T671 | HIKVISION - YouTube](https://www.youtube.com/watch?v=...)>. Acesso em: out. 2021.

FIREBASE. Products. Disponível em: <[Firebase Products \(google.com\)](https://firebase.google.com/products)>. Acesso em: mai. 2024.

G1. Governo revê para cima projeção de inflação, e salário mínimo em 2022 pode subir para R\$ 1.192,00. Disponível em: <[Governo revê para cima projeção de inflação, e salário mínimo em 2022 pode subir para R\\$ 1.192 | Economia | G1 \(globo.com\)](https://g1.globo.com/economia/noticia/governo-reve-para-cima-projecao-de-inflacao-e-salario-minimo-em-2022-pode-subir-para-r-1.192-00.ghtml)>. Acesso em: out. 2021.

GITHUB. Topics C. Disponível em: <[c · GitHub Topics · GitHub](https://github.com/topics/c)>. Acesso em: mai. 2024.

Laravel. Laravel. Disponível em: <[GitHub - laravel/laravel: Laravel is a web application framework with expressive, elegant syntax. We've already laid the foundation for your next big idea — freeing you to create without sweating the small things.](https://laravel.com/docs/7.x/introduction)>. Acesso em: mai. 2024.

MICROSOFT. Getting Started. Disponível em: <[Documentation for Visual Studio Code](https://docs.microsoft.com/en-us/visualstudio/getting-started)>. Acesso em: mai. 2024.

OPENCV. Descrição da biblioteca. Disponível em:<<https://opencv.org/>>. Acesso em: out. 2021.

ORACLE. O que é o MySQL? Disponível em: <[O que é o MySQL | Oracle Brasil](#)>. Acesso em: mai. 2024.

PHP. O que é o PHP? Disponível em: <[PHP: O que é o PHP? - Manual](#)>. Acesso em: mai. 2024.

Tecmundo. Facebook desativa outra IA após novo caso de racismo. Setembro 2021. Disponível em: <[Facebook desativa outra IA após novo caso de racismo - TecMundo](#)>. Acesso em: set. 2021.

TOTVS. Kanban: Conceito, como funciona, vantagens e implementação. Disponível em: <[Kanban: Conceito, como funciona, vantagens e implementação - TOTVS](#)>. Acesso em: out. 2021.

GLOSSÁRIO

[É optativo]

ANEXO A<DESCRIÇÃO DO ANEXO>

Destinam-se à inclusão de informações complementares ao trabalho, mas que não são essenciais à sua compreensão. Os Apêndices devem apresentar material desenvolvido pelo próprio autor, formatado de acordo com as normas. Já os Anexos destinam-se à inclusão de material como cópias de artigos, manuais, etc., que não necessariamente precisam estar em conformidade com o modelo, e que não foram desenvolvidos pelo autor do trabalho. A contagem das páginas nos Apêndices e Anexos segue normalmente. Nos Anexos os números não precisam ser indicados, a não ser na página inicial de cada um.

No caso de haver apenas um anexo, não utiliza-se as letras para enumerá-los. Usa-se a palavra ANEXO no singular.

APÊNDICE <DESCRIÇÃO DO APÊNDICE>

Destinam-se à inclusão de informações complementares ao trabalho, mas que não são essenciais à sua compreensão. Os Apêndices devem apresentar material desenvolvido pelo próprio autor, formatado de acordo com as normas. Já os Anexos destinam-se à inclusão de material como cópias de artigos, manuais, etc., que não necessariamente precisam estar em conformidade com o modelo, e que não foram desenvolvidos pelo autor do trabalho. A contagem das páginas nos Apêndices e Anexos segue normalmente. Nos Anexos os números não precisam ser indicados, a não ser na página inicial de cada um.

No caso de haver apenas um apêndice, não utiliza-se as letras para enumerá-los, a utilização de letras é dispensada. Usa-se a palavra APÊNDICE no sin