

CENTRO PAULA SOUZA
ETEC DE HORTOLÂNDIA
Ensino Médio Integrado ao Técnico em Desenvolvimento de
Sistemas

MIGUEL TRENTINI TORTELLA
PAULO EDUARDO FERREIRA JUNIOR
PEDRO SANTANA FILIPINI
VICENTE MATHEUS COLLIN PEDROSO

SmartAir: SISTEMA DE GERENCIAMENTO
DE ARES-CONDICIONADOS À DISTÂNCIA

Hortolândia
2024

MIGUEL TRENTINI TORTELLA
PAULO EDUARDO FERREIRA JUNIOR
PEDRO SANTANA FILIPINI
VICENTE MATHEUS COLLIN PEDROSO

**SmartAir: SISTEMA DE GERENCIAMENTO
DE ARES-CONDICIONADOS À DISTÂNCIA**

Mini Trabalho de Conclusão de Curso apresentado no Componente de Banco de Dados II do Curso Técnico em Ensino Médio Integrado ao Técnico em Desenvolvimento de Sistemas em 2024 da Etec de Hortolândia, orientado pela Prof. Priscila Batista Martins.

Hortolândia

2024

Sumário

Etapa 1 – Sobre o projeto.....	4
Etapa 1 – Sobre o projeto (Introdução do tema).....	4
Etapa 1 – Sobre o projeto (ODS correspondentes).....	4
Etapa 1 – Sobre o projeto (Resumo e objetivos do projeto).....	5
Etapa 1 – Sobre o projeto (funcionalidades do projeto).....	6
Etapa 2 – Por trás do Banco de Dados.....	7
Etapa 2 – Por trás do Banco de Dados (MER).....	7
Etapa 2 – Por trás do Banco de Dados (Tabelas no SQL).....	8
Etapa 2 – Por trás do Banco de Dados (DER).....	9
Etapa 2 – Por trás do Banco de Dados (Inserção de registros).....	9
Etapa 2 – Por trás do Banco de Dados (Consultas de registros das tabelas).....	10
Etapa 3 – Inner Join e Create View.....	12
Etapa 3 – Inner Join e Create View (Parte Inner Join).....	12
Etapa 3 – Inner Join e Create View (Parte Create View).....	13
Etapa 4 – Consultas, Procedure, Estrutura e Manual da aplicação.....	15
Etapa 4 – Consultas principais.....	15
Etapa 4 – Procedure.....	16
Etapa 4 – Estrutura da aplicação.....	16
Etapa 4 – Manual da aplicação.....	17

SOBRE O PROJETO

O projeto SmartAir, idealizado pelo grupo com o apoio do professor Rafael de Colle, será desenvolvido com base em alguns dos objetivos propostos na Agenda ODS ONU, dos quais se destacam:

- **ODS 7: Energia Acessível e Limpa:**

- **Meta 7.3: Melhorar a eficiência energética global até 2030.**

O projeto SmartAir visa otimizar o uso de ar-condicionado, reduzindo o consumo de energia ao administrar a temperatura ideal virtualmente e evitando, portanto, o uso excessivo de aparelhos.

- **ODS 9: Indústria, Inovação e Infraestrutura:**

- **Meta 9.4: Modernizar a infraestrutura e tornar as indústrias mais sustentáveis, aumentando a eficiência no uso dos recursos e maior adoção de tecnologias e processos industriais limpos e ambientalmente corretos.**

O projeto pode contribuir para a modernização das instalações de laboratórios, utilizando de tecnologias avançadas para gestão de energia e clima.

- **ODS 12: Consumo e Produção Responsáveis:**

- **Meta 12.2: Alcançar a gestão sustentável e o uso eficiente dos recursos naturais.**

O gerenciamento otimizado dos ares-condicionados idealizado pelo projeto pode ser um exemplo de uso eficiente de recursos naturais, diminuindo o consumo de energia e, conseqüentemente, a pegada de carbono (redução de emissões de gases de efeito estufa individuais).

- **ODS 13: Ação Contra a Mudança Global do Clima:**

- **Meta 13.2: Integrar medidas de mudança do clima nas políticas, estratégias e planejamentos nacionais.**

O projeto, ao reduzir o consumo de energia dos ar-condicionados, contribui diretamente para a mitigação das mudanças climáticas, baseado também no argumento da ODS anterior.

- **ODS 17: Parcerias e meios de implementação:**

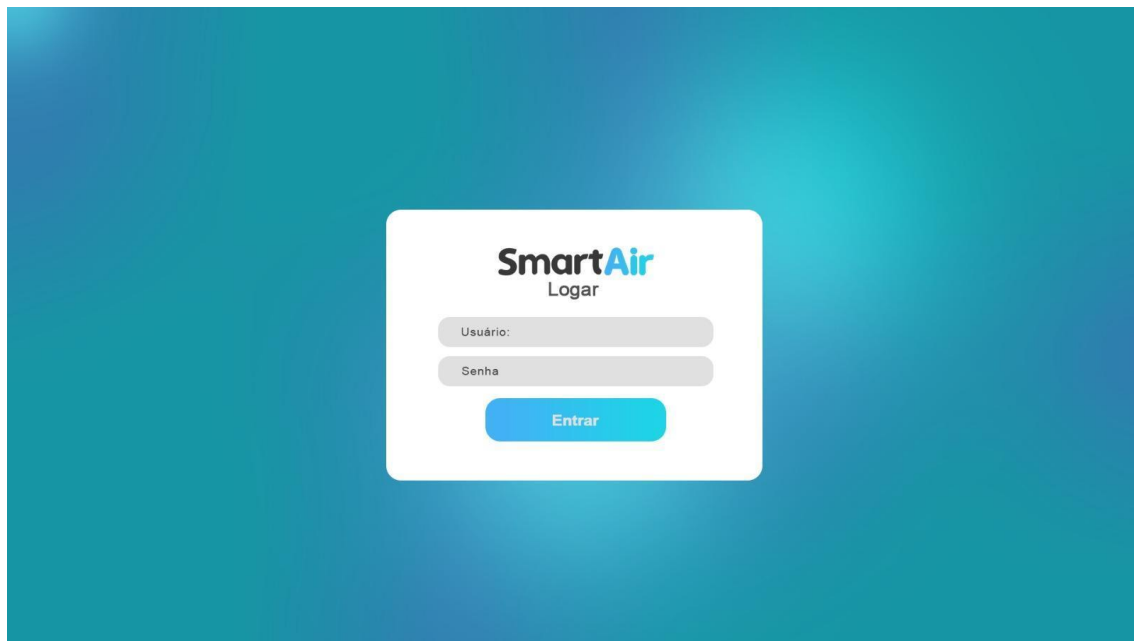
- **Meta 17.7: Promover o desenvolvimento, a transferência, a disseminação e a difusão de tecnologias ambientalmente corretas para os países em desenvolvimento.**

Se tratando de uma ideia promissora, o projeto SmartAir, ao comprovar sua eficiência no âmbito escolar local, pode ter seu sistema tecnológico difundido e expandido a outras áreas, sendo de grande valia para a progressão tecnológica de outros ambientes.

O projeto, em suma, consiste em um sistema de gerenciamento de ar-condicionados, desenvolvido por meio de uma aplicação WPF integrada ao Arduino Esp-32, e que tem por objetivo automatizar e digitalizar a gestão dos aparelhos, de modo que possam ser manuseados à distância por qualquer dispositivo (inicialmente, apenas dispositivos desktop, como o computador local dos professores e um dispositivo central, que seria o computador da sala de manutenção, o qual teria acesso aos aparelhos de todos os ambientes), possibilitando, portanto, uma maior facilidade e mobilidade no controle das temperaturas locais.

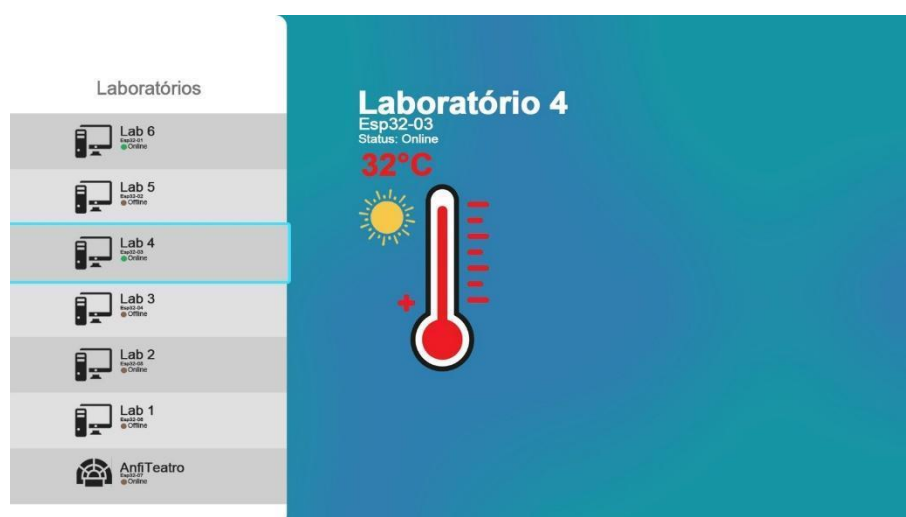
A aplicação WPF do SmartAir seria composta por duas telas principais; a primeira teria a função de login, no qual o usuário (sendo ele professor ou admin), por meio de seu cadastro exclusivo, poderia acessar a tela subsequente. O modelo da tela de login seria semelhante ao mostrado a seguir:

Imagem 1 – Ilustração da tela de login da aplicação SmartAir



A tela seguinte (considerando o acesso pelo dispositivo desktop central) exibiria todos os aparelhos disponíveis, fornecendo informações prévias sobre a situação atual do ar-condicionado (sua localização, Esp correspondente, se está ligado/desligado, etc...). Ao clicar em uma das opções, o usuário teria acesso a informações específicas acerca do aparelho, como a temperatura em que se encontra (caso esteja ligado), a função que esteja exercendo (ventilação, exaustão e afins) e um controle virtual com funções similares ao controle físico, conforme a imagem abaixo:

Imagem 2 – Ilustração hipotética da tela principal/tela ao clicar em um dos dispositivos



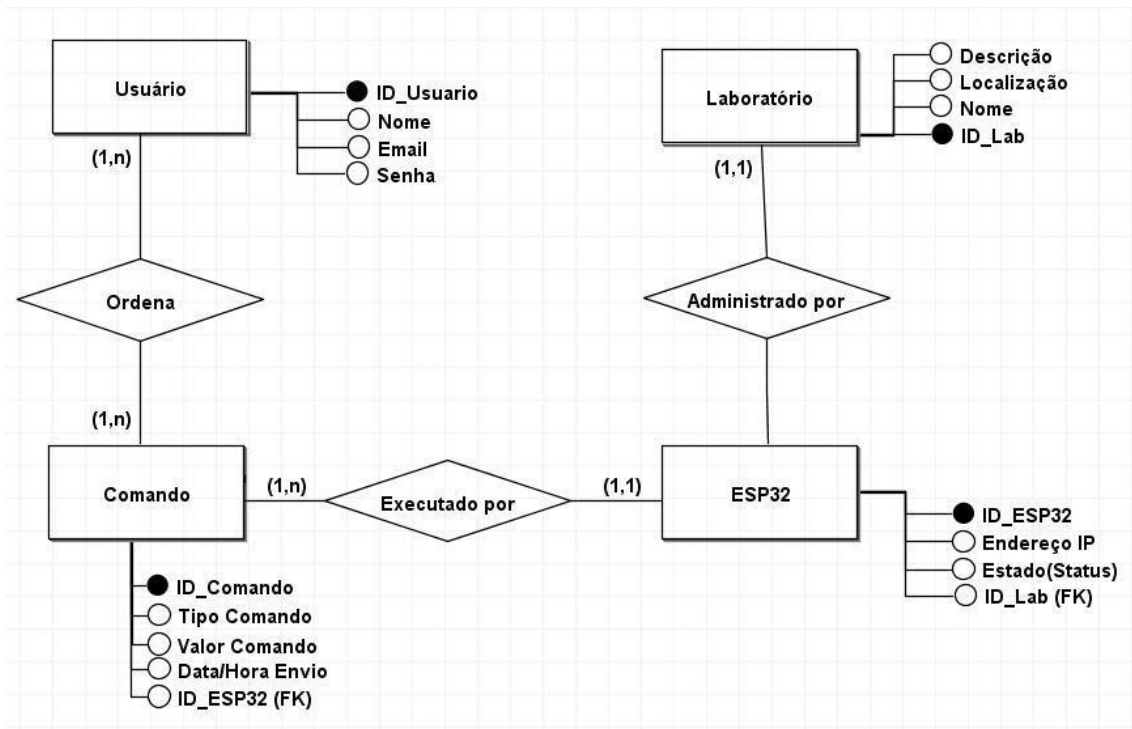
Atrrelado à aplicação WPF, o projeto SmartAir será desenvolvido com a integração do ESP-32, que será responsável por captar (dentro do ar-

condicionado), via radiação infravermelha, comandos enviados pelo dispositivo virtual, que funcionará de modo similar ao controle físico, porém erradicando suas limitações e garantindo um melhor manuseio dos ares-condicionados.

POR TRÁS DO BANCO DE DADOS

No que remete ao Banco de Dados, o back-end desta parte do projeto contará com quatro entidades relacionadas entre si, responsáveis por gerir as informações coletadas e disponíveis na aplicação por meio de seus atributos, conforme o MER a seguir:

Imagem 3 – Ilustração do Modelo de Entidade Relacionamento do projeto



(via BrModelo)

Já as tabelas, desenvolvidas via SQL e responsáveis por conter as informações acerca da aplicação, estarão dispostas conforme os exemplos ilustrados a seguir:

Imagem 4 – Código da tabela da entidade Usuário para inserção dos atributos (à esquerda) e a tabela ao ser executada (à direita)

```

41 CREATE TABLE Usuario (
42     ID_Usuario INT(3) PRIMARY KEY,
43     Nome VARCHAR(100) NOT NULL,
44     Email VARCHAR(100) NOT NULL,
45     Senha VARCHAR(20) NOT NULL
46 );

```

Field	Type	Null	Key	Default	Extra
ID_Usuario	int	NO	PRI	NULL	
Nome	varchar(100)	NO		NULL	
Email	varchar(100)	NO		NULL	
Senha	varchar(20)	NO		NULL	

(via MySQL Workbench)

Imagem 5 – Código da tabela da entidade Comando para inserção dos atributos (à esquerda) e a tabela ao ser executada (à direita)

```

29 CREATE TABLE Comando (
30     ID_Comando INT PRIMARY KEY,
31     ID_ESP32 INT(3),
32     Tipo_Comando VARCHAR(50) NOT NULL,
33     Valor_Comando VARCHAR(100),
34     Data_Hora_Envio DATETIME,
35     FOREIGN KEY (ID_ESP32) REFERENCES ESP32(ID_ESP32)
36 );

```

Field	Type	Null	Key	Default	Extra
ID_Comando	int	NO	PRI	NULL	
ID_ESP32	int	YES	MUL	NULL	
Tipo_Comando	varchar(50)	NO		NULL	
Valor_Comando	varchar(100)	YES		NULL	
Data_Hora_Envio	datetime	YES		NULL	

(via MySQL Workbench)

Imagem 6 – Código da tabela da entidade ESP32 para inserção dos atributos (à esquerda) e a tabela ao ser executada (à direita)

```
17 CREATE TABLE ESP32 (  
18     ID_ESP32 INT(3) PRIMARY KEY,  
19     ID_Lab INT(3),  
20     Endereço_IP VARCHAR(15),  
21     Estado ENUM('online', 'offline'),  
22     FOREIGN KEY (ID_Lab) REFERENCES Lab(ID_Lab)  
23 );
```

Field	Type	Null	Key	Default	Extra
ID_ESP32	int	NO	PRI	NULL	
ID_Lab	int	YES	MUL	NULL	
Endereço_IP	varchar(15)	YES		NULL	
Estado	enum('online','offline')	YES		NULL	

(via MySQL Workbench)

Imagem 7 – Código da tabela da entidade ESP32 para inserção dos atributos (à esquerda) e a tabela ao ser executada (à direita)

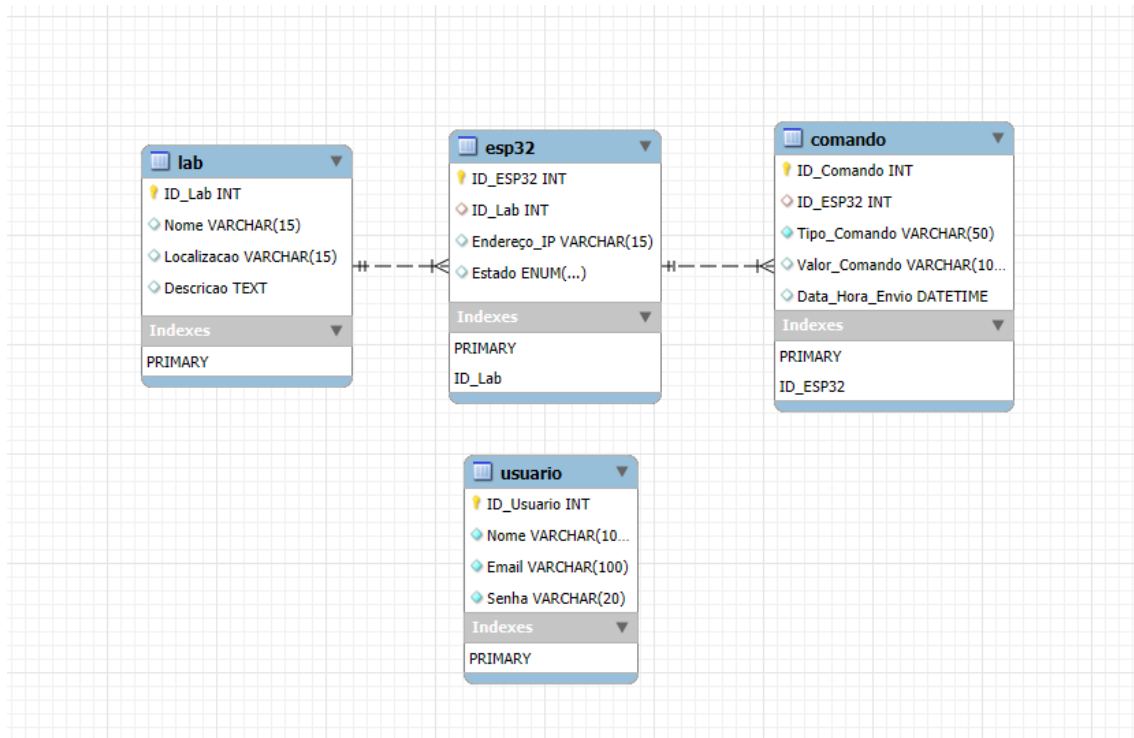
```
4 CREATE TABLE Lab (  
5     ID_Lab INT(3) PRIMARY KEY NOT NULL,  
6     Nome VARCHAR(15),  
7     Localizacao VARCHAR(15),  
8     Descricao TEXT  
9 );
```

Field	Type	Null	Key	Default	Extra
ID_Lab	int	NO	PRI	NULL	
Nome	varchar(15)	YES		NULL	
Localizacao	varchar(15)	YES		NULL	
Descricao	text	YES		NULL	

(via MySQL Workbench)

Nessa mesma linha, o DER do projeto será estruturado conforme a imagem abaixo:

Imagem 8 – Ilustração do Diagrama Entidade-Relacionamento



(via MySQL Workbench)

Sobre os registros, estes são inseridos nas tabelas conforme os exemplos meramente ilustrativos a seguir:

Imagem 9 – Ilustração das inserções de dados correspondentes à tabela Usuários

```
49 • insert into Usuario Values (001, "Pedro", "lerodentario@gmail.com", "hermes");
50 • insert into Usuario Values (002, "Samuel", "zedalinguiça@bol.com", "serieT");
51 • insert into Usuario Values (003, "Paulo", "LigaMetalica@gerson.com", "kripto");
```

(via MySQL Workbench)

Imagem 10 – Ilustração das inserções de dados correspondentes à tabela Comando

```
38 • insert into Comando Values (1, 1, 'Ligar', 'ON', '2024-08-24 10:30:00');
39 • insert into Comando Values (2, 2, 'Temperatura', '16C', '2024-08-24 10:30:00');
40 • insert into Comando Values (3, 3, 'Desligar', 'OFF', '2024-08-24 12:10:00');
```

(via MySQL Workbench)

Imagem 11 – Ilustração das inserções de dados correspondentes à tabela ESP32

```
24 • insert into ESP32 Values (001, 001, "8.206.117.235", "offline");
25 • insert into ESP32 Values (002, 002, "12.138.141.229", "online");
26 • insert into ESP32 Values (003, 003, "15.79.89.102", "online");
```

(via MySQL Workbench)

Imagem 12 – Ilustração das inserções de dados correspondentes à tabela Lab

```
12 • insert into Lab Values (001,"lab 1", "P1", "Apenas exemplo");
13 • insert into Lab Values (002,"lab 2", "P1", "Apenas exemplo");
14 • insert into Lab Values (003,"lab 3", "P2", "Apenas exemplo");
```

(via MySQL Workbench)

Por fim, as consultas acerca das inserções feitas em cada tabela seguirão como os exemplos das imagens seguintes:

Imagem 13 – Ilustração das consultas às inserções
de dados correspondentes à tabela Usuários

	ID_Usuario	Nome	Email	Senha
▶	1	Pedro	lerodentario@gmail.com	hermes
	2	Samuel	zedalinguiça@bol.com	serieT
	3	Paulo	LigaMetalica@gerson.com	kripto
*	NULL	NULL	NULL	NULL

(via MySQL Workbench)

Imagem 14 – Ilustração das consultas às inserções
de dados correspondentes à tabela Comando

	ID_Comando	ID_ESP32	Tipo_Comando	Valor_Comando	Data_Hora_Envio
▶	1	1	Ligar	ON	2024-08-24 10:30:00
	2	2	Temperatura	16C	2024-08-24 10:30:00
	3	3	Desligar	OFF	2024-08-24 12:10:00
*	NULL	NULL	NULL	NULL	NULL

(via MySQL Workbench)

Imagem 15 – Ilustração das consultas às inserções
de dados correspondentes à tabela ESP32

	ID_ESP32	ID_Lab	Endereço_IP	Estado
▶	1	1	8.206.117.235	offline
	2	2	12.138.141.229	online
	3	3	15.79.89.102	online
•	NULL	NULL	NULL	NULL

(via MySQL Workbench)

Imagem 16 – Ilustração das consultas às inserções
de dados correspondentes à tabela Lab

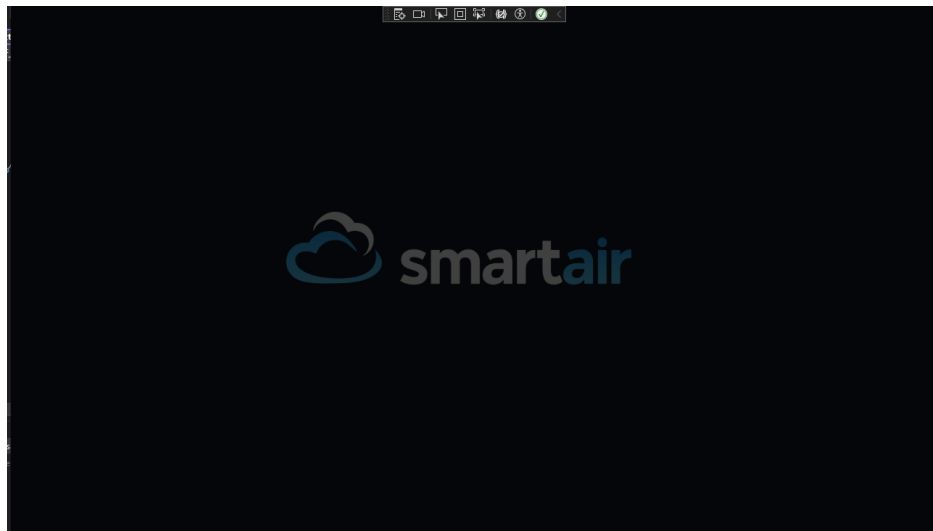
	ID_Lab	Nome	Localizacao	Descricao
▶	1	lab 1	P1	Apenas exemplo
	2	lab 2	P1	Apenas exemplo
	3	lab 3	P2	Apenas exemplo
•	NULL	NULL	NULL	NULL

(via MySQL Workbench)

ESTRUTURA DO PROJETO

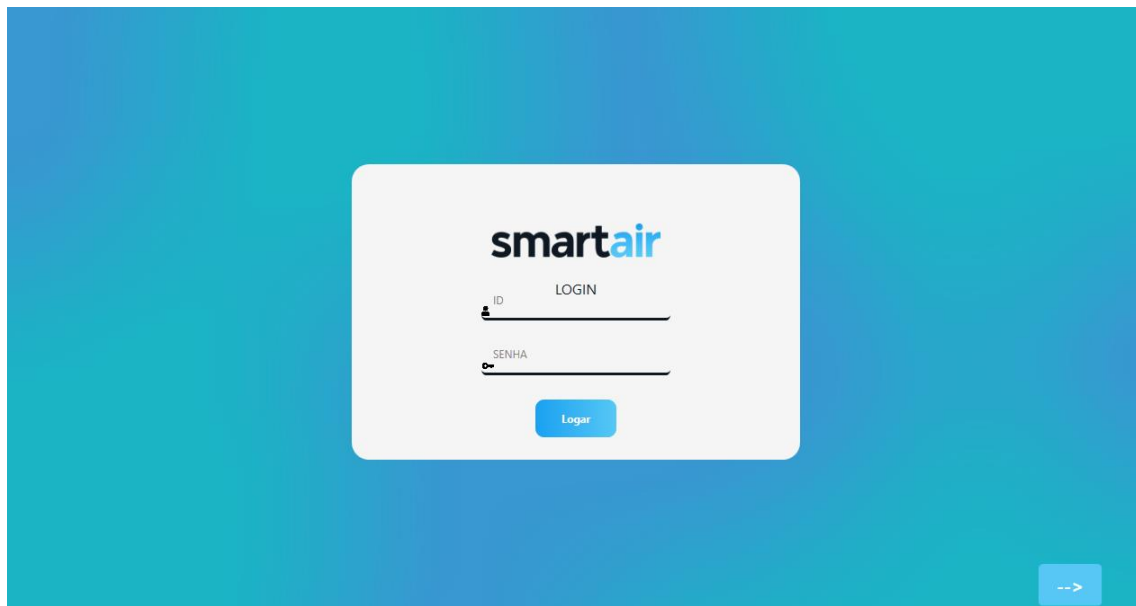
Na parte estrutural do aplicativo, se fazem presentes telas 8 primordiais, as quais podem ser visualizadas a seguir:

Imagem 25 – Tela de loading do projeto SmartAir



(Via Visual Studio 2022)

Imagem 26 – Tela de login do projeto SmartAir



(Via Visual Studio 2022)

Imagem 27 – Tela inicial do projeto SmartAir



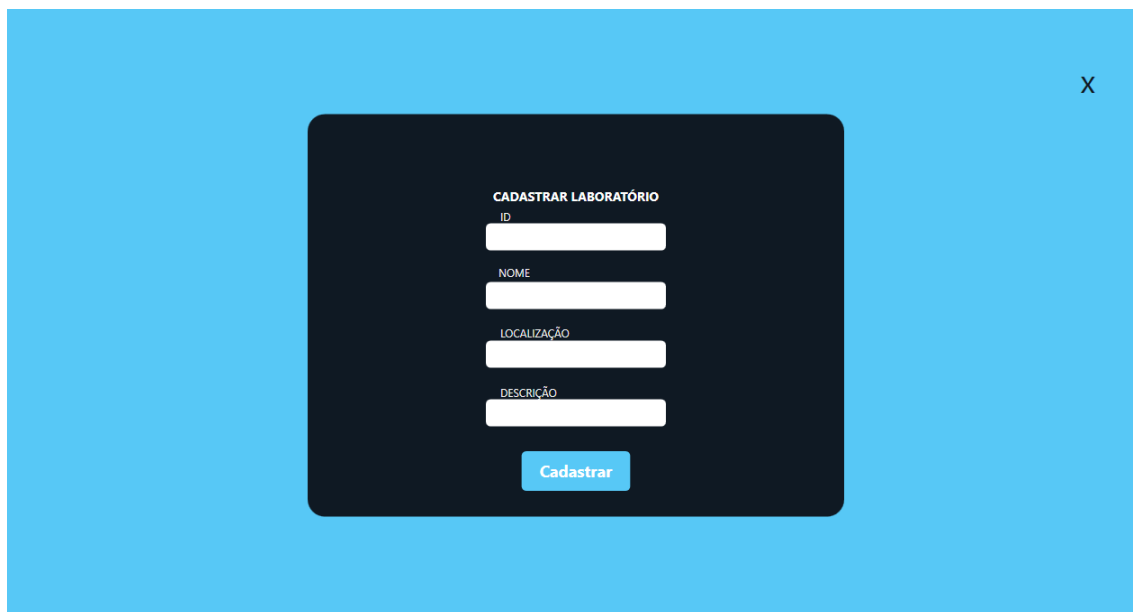
(Via Visual Studio 2022)

Imagem 28 – Tela de cadastro de administradores do projeto SmartAir

A screenshot of a registration form titled 'CADASTRAR ADM'. The form is a dark blue rounded rectangle centered on a light blue background. It contains four white input fields stacked vertically, labeled 'ID', 'SENHA', 'NOME', and 'EMAIL'. Below the fields is a light blue button with the text 'Cadastrar'. In the top right corner of the light blue background, there is a small 'X' icon.

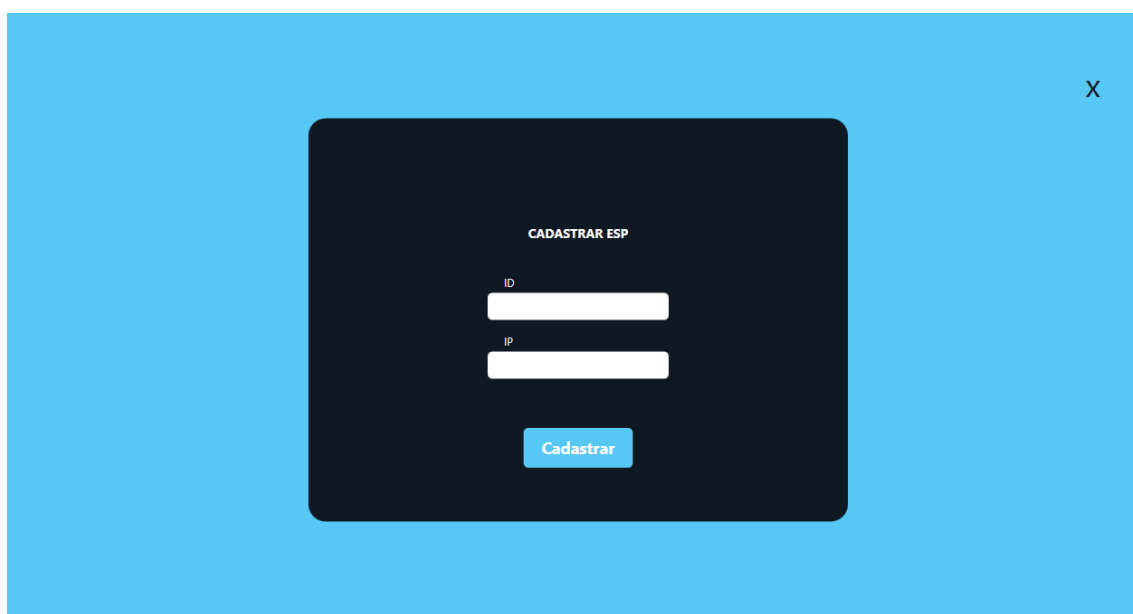
(Via Visual Studio 2022)

Imagem 29 – Tela de cadastro de laboratórios do projeto SmartAir



(Via Visual Studio 2022)













Imagem 30 – Tela de cadastro de ESPs do projeto SmartAir



(Via Visual Studio 2022)

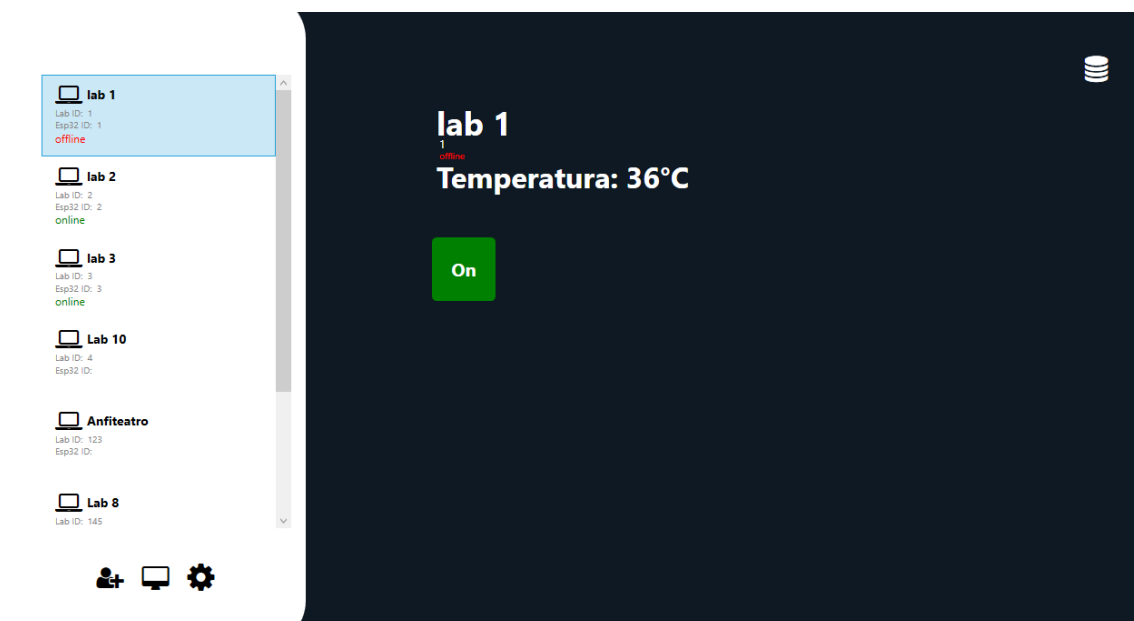
Imagem 31 – Tela de associação dos ESPs aos laboratórios do projeto SmartAir

X

ID	IP	ID do Lab	Ações
1	8.206.117.235	1	 
2	12.138.141.229	2	 
3	15.79.89.102	3	 
111	1123434	145	 
112	122123245	145	 
			 

(Via Visual Studio 2022)

Imagem 32 – Tela individual do controle virtual
(meramente ilustrativo) de cada laboratório



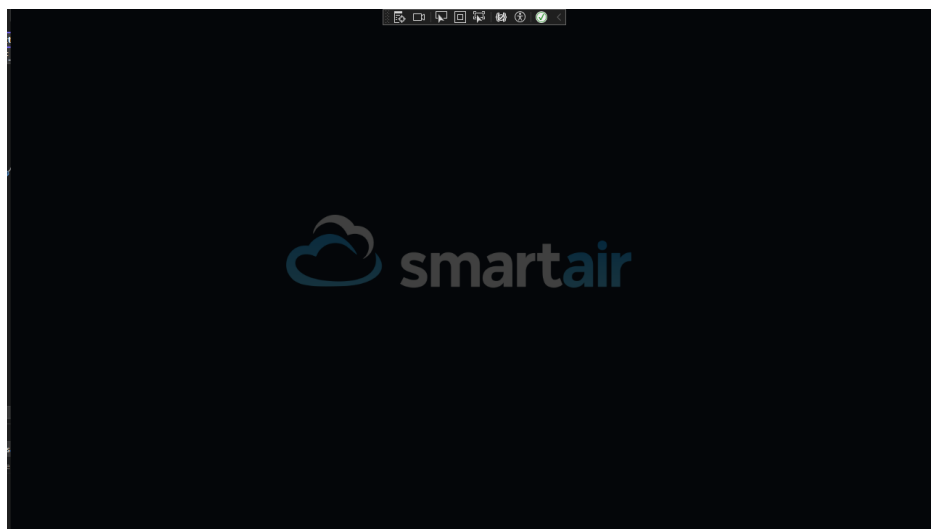
(Via Visual Studio 2022)

Agora, é importante explicar com mais minuciosidade a funcionalidade de cada uma das telas apresentadas e quais foram as ferramentas, plataformas e linguagens em que cada parte do projeto SmartAir foi desenvolvido.

Em geral, o projeto interdisciplinar das matérias técnicas de Banco de Dados II e Desenvolvimento de Sistemas foi desenvolvido na íntegra nas plataformas Visual Studio 2022 (no formato WPF.NET) e MySQL Workbench, com acréscimo integrado da plataforma Arduino, sendo, portanto, desenvolvido nas linguagens C#, SQL e Wiring (variação de C++). Segue individualmente as partes desenvolvidas pelas linguagens e formato de front-end/back-end lógico:

- **C#/WPF.NET:** Nesta parte, pode-se dizer que 7 das 8 telas apresentadas foram desenvolvidas integralmente com o formato WPF.NET via C#, a começar pela tela de loading, que possui uma codificação específica para seu funcionamento.

Figura 01 – Tela de loading do projeto SmartAir



(Via Visual Studio 2022)

Figura 02 – Código da tela de loading do projeto SmartAir

```

16 namespace WPF_MiniTCC_Smartir
17 {
18     public partial class SplashScreen : Window
19     {
20         public SplashScreen()
21         {
22             InitializeComponent();
23
24             // Iniciar o efeito de fade-out quando a janela for carregada
25             Loaded += SplashScreen_Loaded;
26         }
27
28         private void SplashScreen_Loaded(object sender, RoutedEventArgs e)
29         {
30             // Buscar o storyboard de animação e iniciá-lo
31             Storyboard fadeOutStoryboard = (Storyboard)this.FindResource("FadeOutAnimation");
32             fadeOutStoryboard.Begin(blackOverlay); // Passar o retângulo como o alvo da animação
33         }
34     }

```

(Via Visual Studio 2022)

Figura 01: Tela de carregamento utilizada para retardar o acesso do usuário ao sistema enquanto este carrega os dados adicionais.

Figura 02: Código utilizado para a construção do efeito de carregamento da tela.

A seguir, há também a tela de login, onde fornecemos informações já cadastradas para ter acesso às funcionalidades do aplicativo, como ID do administrador e sua senha.

Figura 03 – Tela de Login do projeto SmartAir

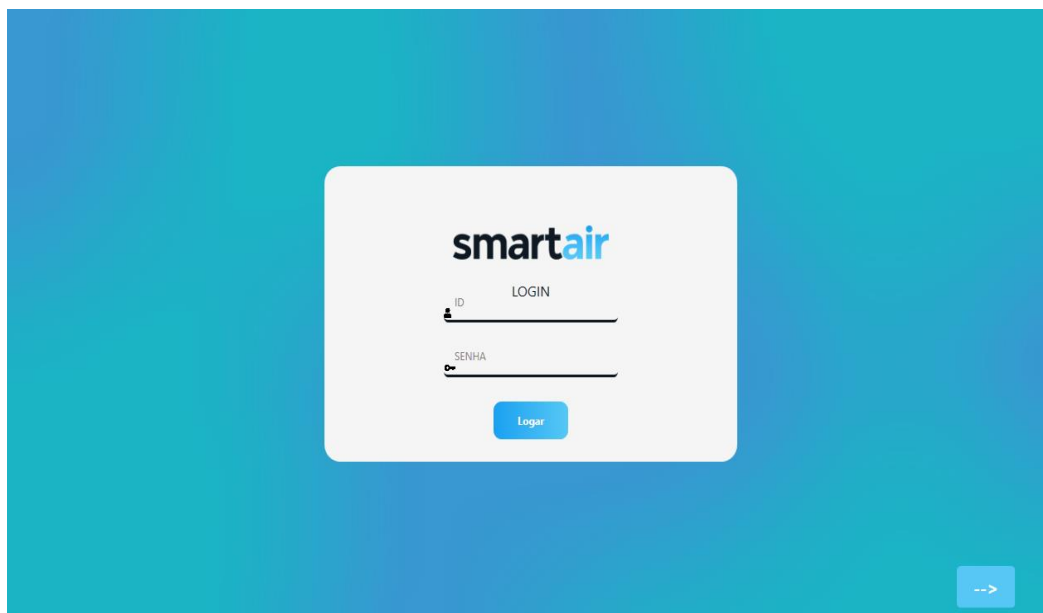


Figura 03: Tela cuja função é realizar o login pelo usuário já cadastrado.

Posteriormente, tem-se a chamada tela inicial, onde todas as partes do projeto estão dispostas e podem ser acessadas pelos ícones da interface. Nela, temos acesso às informações correspondentes aos laboratórios cadastrados (controle, ar-condicionado ligado/desligado, ID do ESP associado, etc.), bem como os ícones direcionadores para os cadastros (seja de administradores, laboratórios ou ESPs) e um ícone direcionador para a tela de edição dos status de um ESP (apagar, associar a um laboratório, etc.).

Figura 04 – Tela inicial do projeto SmartAir



Figura 04: Tela de início do projeto que possui dispostas em seu display todas as funcionalidades do aplicativo.

Em seguida, foi desenvolvido a tela de cadastro dos eventuais novos administradores do aplicativo, na qual são requeridas informações como ID, senha, nome e email do docente, a fim de que este tenha acesso ao controle das funções do aplicativo.

Figura 05 – Tela de cadastro de admins do projeto SmartAir

Figura 05: Tela para cadastrar dados de eventuais novos docentes que venham a ter acesso ao sistema do aplicativo.

Em contexto semelhante, há a tela de cadastro para laboratórios que venham a ter o sistema SmartAir integrado para gerir o ar-condicionado local (pensando em uma eventual expansão do prédio ou uma possível manutenção dos ares-condicionados de determinados ambientes, que não precisam necessariamente serem laboratórios).

Figura 06 – Tela de cadastro de laboratórios do projeto SmartAir

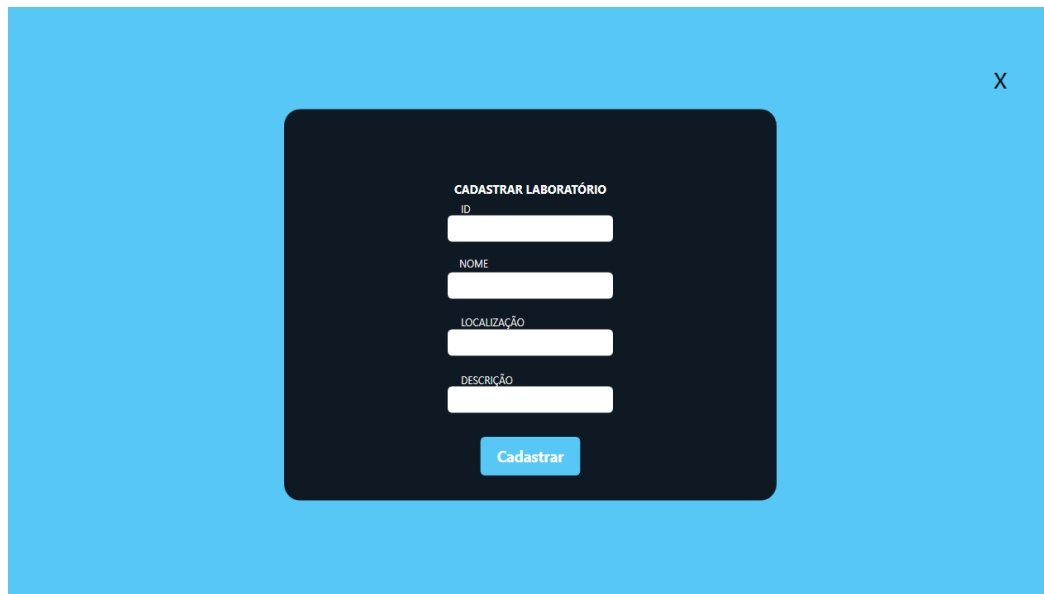
A imagem mostra uma interface de usuário para o cadastro de laboratórios. O fundo é azul claro. No centro, há um formulário escuro com o título "CADASTRAR LABORATÓRIO" em letras brancas. Abaixo do título, há quatro campos de entrada brancos, cada um com um rótulo em letras brancas: "ID", "NOME", "LOCALIZAÇÃO" e "DESCRIÇÃO". Abaixo dos campos, há um botão azul com o texto "Cadastrar" em branco. No canto superior direito do formulário, há um ícone "X" branco para fechar a tela.

Figura 06: Tela para cadastro de eventuais novos ambientes que venham a ter o aplicativo SmartAir integrado em seu ar-condicionado.

Na mesma lógica, tem-se também a tela para cadastro de ESPs, considerando que novos ESPs possam vir a serem utilizados (ora para substituir algum ESP quebrado, ora para implementar o sistema em outro ambiente cuja aplicação se faz ausente).

Figura 07 – Tela de cadastro de ESPs do projeto SmartAir



Figura 07: Tela de cadastro para eventuais novos ESPs que venham a ser integrados em determinados ares-condicionados.

Ainda sobre ESPs, foi adicionada uma tela com DataGrid cuja função é associar aos laboratórios eventuais novos ESPs cadastrados, bem como excluir os que não são mais utilizados pelo sistema. É possível associar somente um ESP por laboratório.

Figura 08 – Tela de associação dos ESPs aos laboratórios do projeto SmartAir

ID	IP	ID do Lab	Ações
1	8.206.117.235	1	
2	12.138.141.229	2	
3	15.79.89.102	3	
111	1123434	145	
112	122123245	145	

Figura 08: Tela para emparelhamento de novos ESPs aos ambientes cadastrados via ID próprio e endereço de IP.

Por fim, a tela do controle virtual exemplifica as funções principais do controle físico, o qual cada laboratório terá individualmente em sua representação. A janela alterna de visual de acordo com o status do ar-condicionado local (se está ligado ou desligado).

Figura 09 – Tela do controle virtual ilustrado em uma tela de laboratório cadastrado (ligado)

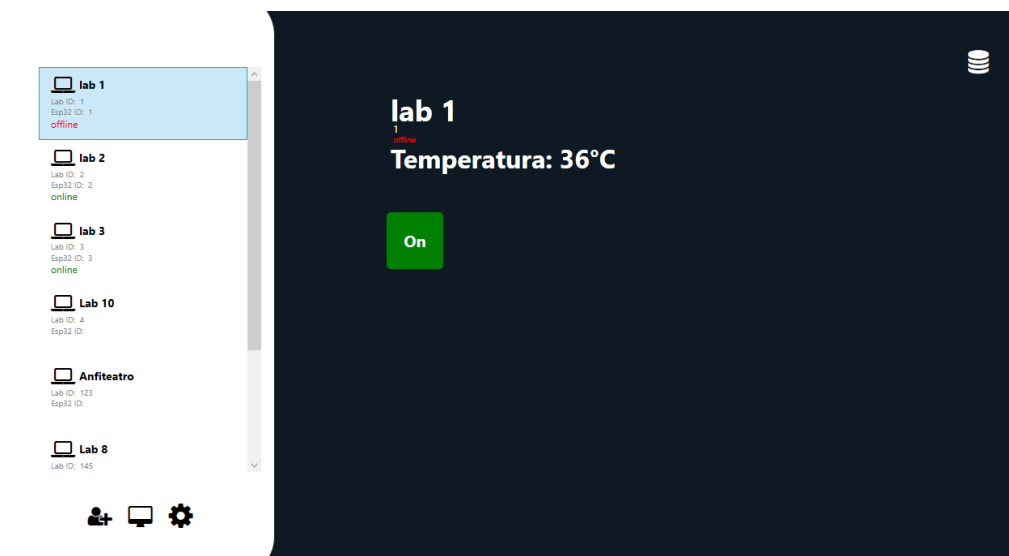
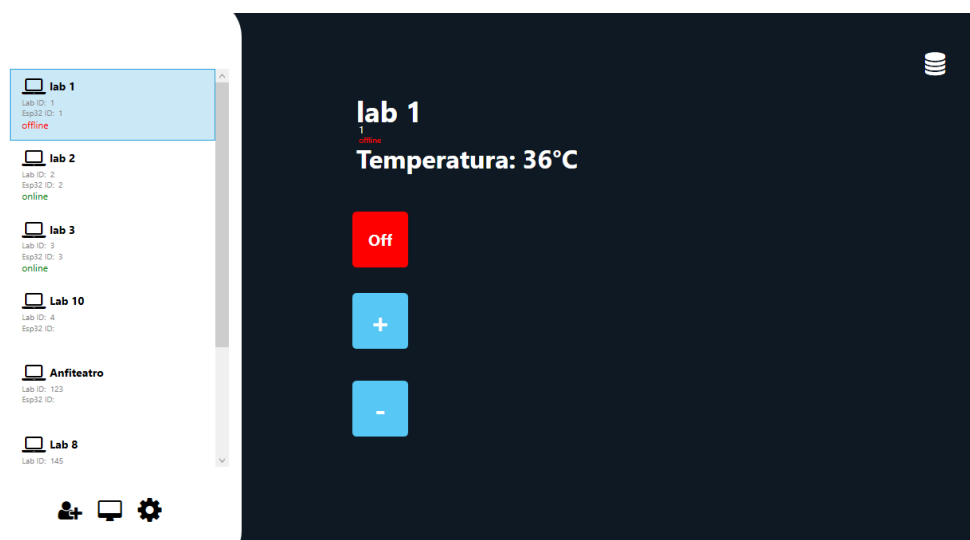


Figura 10 – Tela do controle virtual ilustrado em uma tela de laboratório cadastrado (ligado)



Figuras 09 e 10: Telas com o controle que, ao serem clicados os botões, enviará os comandos via IoT para o ESP que, por sua vez, emitirá os códigos via infravermelho para o ar-condicionado.

- **Arduino (Wiring)/WPF.NET:** Esta parte é brevemente pois representa apenas a parte dos controles virtuais, na qual foi implementado uma biblioteca para detecção e emissão de infravermelho NEC (IRremote), onde, via IoT, os códigos serão implementados no banco de dados da janela que, ao recebê-los e alocá-los corretamente nos botões virtuais, será capaz de transmitir os códigos dispostos em hexadecimal de cada sinal infravermelho cordial aos botões do controle físico, gerando maior praticidade e mobilidade no gerenciamento dos ares-condicionados.

Figura 11: Parte do código que exemplifica a lógica utilizada para a função de ligar o ar-condicionado.

```
1
2
3 #include <IRremote.h>
4
5 const int recvPin = 11;
6 const int irSendPin = 3;
7 const int buttonPin = 5;
8
9 IRrecv irrecv(recvPin);
10 IRsend irsend;
11 decode_results results;
12
13 void setup() {
14     Serial.begin(9600);
15     irrecv.enableIRIn(); // Inicia o receptor IR
16     pinMode(buttonPin, INPUT);
17 }
18
19 void loop() {
20     if (irrecv.decode(&results)) {
21         Serial.println(results.value, HEX); // Imprime o código IR capturado
22         irrecv.resume(); // Prepara para o próximo valor
23     }
24
25     //Para ligar lab 4
26
27     if (digitalRead(buttonPin) == HIGH) {
28
29         irsend.sendNEC(0xB24D1FE0, 32);
30         Serial.println('Codigo enviado');
31         delay(1000); // Evita múltiplos envios
32     }
33
34 }
```

Figura 11: Breve exemplo de código que será utilizado para implementar as funcionalidades que prometem replicar os comandos dos controles físicos dos ares-condicionados virtualmente.