



UNifeob
| ESCOLA DE NEGÓCIOS



2023

PROJETO INTEGRADO



UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS
ESCOLA DE NEGÓCIOS
A.D.S. E CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO
IOT DATA STREAMER
<FastLane>

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2023

UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS
ESCOLA DE NEGÓCIOS
A.D.S. E CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

IOT DATA STREAMER

<FastLane>

MÓDULO MODELAGEM E DESENVOLVIMENTO DE SISTEMAS

Gestão Financeira – Profa. Renata Elizabeth de Alencar Marcondes

Programação Orientada a Objeto – Prof. Nivaldo Andrade

Lógica de Programação – Prof. Marcelo Ciacco de Almeida

Modelagem de Dados – Prof. Max Streicher Vallim

Projeto de Modelagem e Desenvolvimento de Sistemas – Profª. Mariângela

Martimbianco Santos

Estudantes:

Rodrigo Espinosa Teixeira, RA 23000243

Felipe Fonseca Gimenes, RA 23000242

Enzo Dorigon Leandrini, RA 23000663

Marcos Vinícius Carvalho da Silva, RA 23000327

Rafael Meireles Alves Monteiro, RA 23000481

SUMÁRIO

SUMÁRIO	3
1. INTRODUÇÃO	4
2. DESCRIÇÃO DA EMPRESA	5
3. PROJETO INTEGRADO	6
3.1 PROGRAMAÇÃO ORIENTADA A OBJETO	6
3.1.1 CLASSES E OBJETOS	7
3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA.	8
3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS	9
3.2 LÓGICA DE PROGRAMAÇÃO	11
3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE	11
3.2.2 DESENVOLVIMENTO DE APLICAÇÕES DESKTOP	11
3.3 MODELAGEM DE DADOS	12
3.3.1 MODELO CONCEITUAL	13
3.3.2 MODELO LÓGICO E FÍSICO	15
3.3.3 SQL	16
3.4 GESTÃO FINANCEIRA	17
3.4.1 CLASSIFICAÇÃO DOS CUSTOS	18
3.4.2 CUSTOS DO PROJETO	19
3.4.3 PRECIFICAÇÃO	19
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS	21
3.5.1 GERENCIANDO FINANÇAS	21
3.5.2 ESTUDANTES NA PRÁTICA	24
REFERÊNCIAS	27
ANEXOS	28

1. INTRODUÇÃO

Com o advento da transformação digital, a infraestrutura de Tecnologia da Informação (TI) tornou-se um pilar fundamental para o funcionamento eficiente de organizações em todo o mundo. Nesse contexto, a estabilidade e o desempenho dos servidores tornam-se cruciais para garantir a continuidade operacional e a segurança dos dados. Reconhecendo essa importância, apresentamos o projeto inovador de monitoramento de servidores por meio do uso de sensores de temperatura e umidade.

O monitoramento ambiental de data centers e salas de servidores é uma prática essencial para prevenir falhas, otimizar o desempenho e prolongar a vida útil dos equipamentos. Nosso projeto utiliza sensores de última geração para coletar dados precisos de temperatura e umidade, oferecendo uma visão abrangente das condições ambientais que impactam diretamente no funcionamento dos servidores.

O sensor de temperatura e umidade proposto é projetado para fornecer informações em tempo real, permitindo uma resposta imediata a variações críticas no ambiente. Essa abordagem proativa visa evitar problemas relacionados ao superaquecimento, condensação e outros fatores que podem comprometer a estabilidade dos servidores e, por consequência, a continuidade dos serviços.

Além do monitoramento contínuo, nossa solução oferece recursos avançados, como alertas automáticos e histórico de dados, possibilitando análises retrospectivas para identificação de padrões e tendências. Isso não apenas reduz o risco de falhas inesperadas, mas também permite a implementação de estratégias preventivas, melhorando a eficiência operacional.

Ao adotar essa solução de monitoramento de temperatura e umidade em ambientes críticos de TI, as empresas ganham uma ferramenta poderosa para aprimorar a confiabilidade de seus servidores, reduzir custos relacionados a manutenções corretivas e, acima de tudo, garantir a entrega consistente de serviços digitais. Este projeto representa um passo significativo em direção a uma infraestrutura de TI resiliente e adaptável às demandas do mundo moderno.

2. DESCRIÇÃO DA EMPRESA

A Unifeob (Fundação de Ensino Octávio Bastos) é uma instituição de ensino superior de destaque, situada em São João da Boa Vista, no estado de São Paulo, Brasil. Nossa missão primordial é a oferta de educação de elevado padrão em diversas áreas do conhecimento, visando enriquecer o cenário educacional e profissional da região.

A diversidade de cursos de graduação, pós-graduação e extensão universitária compõem a vasta oferta educacional da instituição, abarcando uma ampla variedade de campos de conhecimento. Além disso, nossa instituição mantém sólidas parcerias com renomadas empresas e instituições, enriquecendo ainda mais a experiência educacional dos alunos.

A Unifeob está intrinsecamente comprometida com a promoção da pesquisa e da inovação. Estimulamos ativamente nossos alunos e docentes a engajarem-se em projetos e atividades científicas que contribuam substancialmente para o progresso do saber em suas respectivas áreas.

Informações adicionais:

CNPJ: 59.764.555/0001-52

CEP: 13874-149

3. PROJETO INTEGRADO

A equipe do projeto trabalha em estreita colaboração com os membros da empresa. Foi realizado um Banner com os fundadores da empresa, bem como com os responsáveis pela gestão e implementação das boas práticas financeiras, a fim de entender melhor a necessidade das pessoas e como as práticas financeiras podem ser aprimoradas para atender a essas necessidades.

Por fim, nossa empresa apresentou um relatório detalhado sobre suas descobertas e recomendações para a vida, a fim de ajudar a empresa a aprimorar ainda mais suas práticas financeiras de alta precisão e atender melhor às necessidades de seus clientes."

3.1 PROGRAMAÇÃO ORIENTADA A OBJETO

O projeto FastLane é um sistema baseado em programação orientada a objetos (POO) em Python, desenvolvido para simular a leitura de um sensor. O sensor possui funcionalidades que permitem a leitura de valores, proporcionando uma experiência realista para o usuário. O código foi organizado em três arquivos distintos: `sensor.py`, `simulador.py` e `main.py`, para uma melhor modularidade e organização. A classe `Sensor` encapsula as operações de leitura e obtenção de valores, enquanto a classe `Simulador` utiliza um objeto `Sensor` para simular a leitura e imprimir o valor lido. Este projeto serve como uma base sólida para futuras expansões e integração de funcionalidades adicionais.

Como falamos, a programação orientada a objetos é importante por diversos motivos. Primeiramente, ela promove a reutilização de código por meio do conceito de herança, possibilitando que classes derivadas herdem características e comportamentos de classes base. Isso resulta em um desenvolvimento mais eficiente, pois evita a necessidade de reescrever o mesmo código várias vezes. (Menezes, Elizabeth)

3.1.1 CLASSES E OBJETOS

Main.py: O código cria instâncias de duas classes, `Sensor` e `Simulador`. A classe `Sensor` representa um tipo de sensor, enquanto a classe `Simulador` é utilizada para simular leituras desse sensor. O programa principal (`main()`) cria um objeto `Sensor`, instância um `Simulador` passando o objeto `Sensor` como argumento e, em seguida, simula uma leitura chamando o método `simular_leitura()` do objeto `Simulador`

Tabela 1 - Main.py:

```
from sensor import Sensor
from simulador import Simulador

def main():
    sensor = Sensor()
    simulador = Simulador(sensor)

    simulador.simular_leitura()

if __name__ == "__main__":
    main()
```

Fonte: Grupo 7 ADS/CC

Sensor.py: O código em `Sensor.py` define uma classe chamada `Sensor`. Nesta classe: O método `__init__` inicializa um objeto `Sensor` com um atributo chamado `valor` iniciado com zero. O método `ler_valor` gera um número aleatório entre 0 e 100 e atribui esse valor ao atributo `valor` do objeto `Sensor`. O método `obter_valor` retorna o valor atual do atributo `valor` do objeto `Sensor`:

Tabela 2 - Sensor.py:

```
import random
class Sensor:
    def __init__(self):
        self.valor = 0
    def ler_valor(self):
        self.valor = random.randint(0, 100)
    def obter_valor(self):
        return self.valor
```

Fonte: Grupo 7 ADS/CC

Simulador.py: O código em `Simulador.py` define uma classe chamada `Simulador`. Nesta classe: O método `__init__` é chamado quando um objeto `Simulador` é criado e recebe um objeto `sensor` como argumento. Esse objeto `sensor` é armazenado como um atributo chamado `sensor` no objeto `Simulador`. O método `simular_leitura` chama o método `ler_valor` do objeto `sensor`, que, por sua vez, gera um valor aleatório e o atribui ao atributo `valor` do objeto `sensor`. Em seguida, o método chama o método `obter_valor` para obter o valor atual do objeto `sensor` e o imprime na tela:

Tabela 3 - Simulador.py:

```
class Simulador:
    def __init__(self, sensor):
        self.sensor = sensor
    def simular_leitura(self):
        self.sensor.ler_valor()
        valor = self.sensor.obter_valor()
        print(f'O valor lido pelo sensor é: {valor}')
```

Fonte: Grupo 7 ADS/CC

3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA.

Atributos e Métodos: No contexto do FastLane, podemos adicionar atributos e métodos para tornar o código mais robusto e funcional. Por exemplo, podemos adicionar um atributo para armazenar o nome do sensor e um método para configurar esse nome:

Tabela 4 - Atributos e Métodos:

```
class Sensor:
    def __init__(self, nome):
        self.nome = nome
        self.valor = 0
    def configurar_nome(self, novo_nome):
        self.nome = novo_nome
    # Restante do código..
```

Fonte: Grupo 7 ADS/CC

Encapsulamento: Podemos usar o conceito de encapsulamento para proteger certos atributos e métodos, tornando-os privados e acessíveis apenas dentro da classe:

Tabela 5 - Encapsulamento:

```
class Sensor:
    def __init__(self, nome):
        self.__nome = nome
        self.__valor = 0
    def __ler_valor(self):
        # ...
    def __obter_valor(self):
        return self.__valor
    def configurar_nome(self, novo_nome):
        self.__nome = novo_nome
    # Restante do código...
```

Fonte: Grupo 7 ADS/CC

Herança: Suponha que queremos criar um tipo de sensor mais avançado chamado Sensor Avançado que herda as características do Sensor básico, mas também possui funcionalidades adicionais:

Tabela 6 - Herança:

```
class SensorAvancado(Sensor):
    def __init__(self, nome):
        super().__init__(nome)
        self.calibrado = False
    def calibrar(self):
        self.calibrado = True
    # outras funcionalidades específicas do sensor avançado...
```

Fonte: Grupo 7 ADS/CC

3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

Métodos Estáticos: Os métodos estáticos pertencem à classe em vez de pertencerem a instâncias específicas da classe. Eles podem ser úteis quando uma operação não depende do estado atual do objeto. Exemplo: Converter um valor de temperatura de Celsius para Fahrenheit:

Tabela 7 - Método Estático:

```
class Sensor:
    # ...
    @staticmethod
    def celsius_para_fahrenheit(celsius):
        return (celsius * 9/5) + 32
# Uso:
temp_celsius = 20
temp_fahrenheit = Sensor.celsius_para_fahrenheit(temp_celsius)
print(f"{temp_celsius}°C é igual a {temp_fahrenheit}°F")
```

Fonte: Grupo 7 ADS/CC

Métodos Públicos e Privados: Métodos públicos são acessíveis fora da classe, enquanto métodos privados são destinados a serem usados internamente pela classe e não devem ser acessados externamente. Exemplo: Adicionar um método privado que realiza uma operação de calibração:

Tabela 8 - Métodos Públicos e Privados:

```
class Sensor:
    # ...
    def __calibrar(self):
        self.calibrado = True
    def simular_leitura(self):
        if self.calibrado:
            self.ler_valor()
            valor = self.obter_valor()
            print(f"O valor lido pelo sensor {self.nome} é: {valor}")
        else:
            print(f"O sensor {self.nome} precisa ser calibrado antes da leitura.")

    def calibrar_sensor(self):
        self.__calibrar()
# Uso:
sensor = Sensor("FastLane")
\1')
```

Fonte: Grupo 7 ADS/CC

3.2 LÓGICA DE PROGRAMAÇÃO

A lógica da programação é uma parte fundamental no início do desenvolvimento de sistemas. Nesse tópico, os estudantes aprenderão como funciona a lógica por trás dos computadores e como aplicá-la na prática. Serão abordados conceitos fundamentais, como algoritmos, variáveis, tipos de dados, funções, estruturas condicionais, operadores lógicos e operadores de comparação, voltados para a linguagem de programação JavaScript. Além disso, também trabalharão com prototipação, criação de aplicações desktop com o framework Electron JS e banco de dados relacional (MySQL).

A lógica de programação nada mais é do que uma sequência de passos para resolver um problema. Quem vai resolver o problema, nesse caso, é o computador, baseado nas instruções que passamos a ele. Então, precisamos saber quais tipos de instruções a máquina entende e qual a melhor forma de passarmos os comandos!(Alura, 2023)

3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

A base para criar o sistema do site da Fast Lane foi desenvolvida com as linguagens html, css e javascript, fizemos usando o html o corpo do nosso site os títulos, textos, nomes, cor de fundo e botões presente no site, usamos essa linguagem também para criar links para a parte de gerenciamento e para criar as tabelas como mostra o nosso arquivo index.html e os arquivos das empresas, feobserver.html, martiserver.html e paliunet.html. Utilizamos a linguagem css para dar um melhor enquadramento para os dados contidos no nosso site e para a tabela de dados, também a usamos para colocar cor no fundo da página inicial, já com a linguagem javascript fizemos com e os dados inseridos nas tabelas de cadastro sejam lidos assim criando uma nova tabela com as informações inseridas e sendo armazenadas pelo sistema e após isso limpando as tabelas de cadastro para novas informações.

3.2.2 DESENVOLVIMENTO DE APLICAÇÕES DESKTOP

Utilizando uma programação mais avançada com o electron js fizemos uma conexão do site o ligando com o nosso banco de dados e nosso código sql que é mostrado em nosso arquivo connection.js, também pode se obter a situação dessa conexão com o nosso arquivo database.js, através desse modelo em javascript não apenas fizemos um cadastro mas também com que essa conta criada seja salva e armazenada através do nosso arquivo app.js e o nosso arquivo sensor_controller.js, tem a função de criar uma nova entrada no banco de

dados e salvar esses dados, depois retorna uma promessa que será resolvida com o ID da nova instância se a criação for bem-sucedida, ou rejeitada com um erro se ocorrer algum problema, essa função é responsável por buscar todos os registros existentes do modelo no banco de dados, depois retorna uma promessa que será resolvida com um array contendo todos os registros, ou rejeitada com um erro se a operação de busca falhar. Ambas as funções são exportadas para que possam ser utilizadas em outros arquivos do projeto, como integrar essas funções do nosso site conectado com o nosso banco de dados. (Anexo 2)

3.3 MODELAGEM DE DADOS

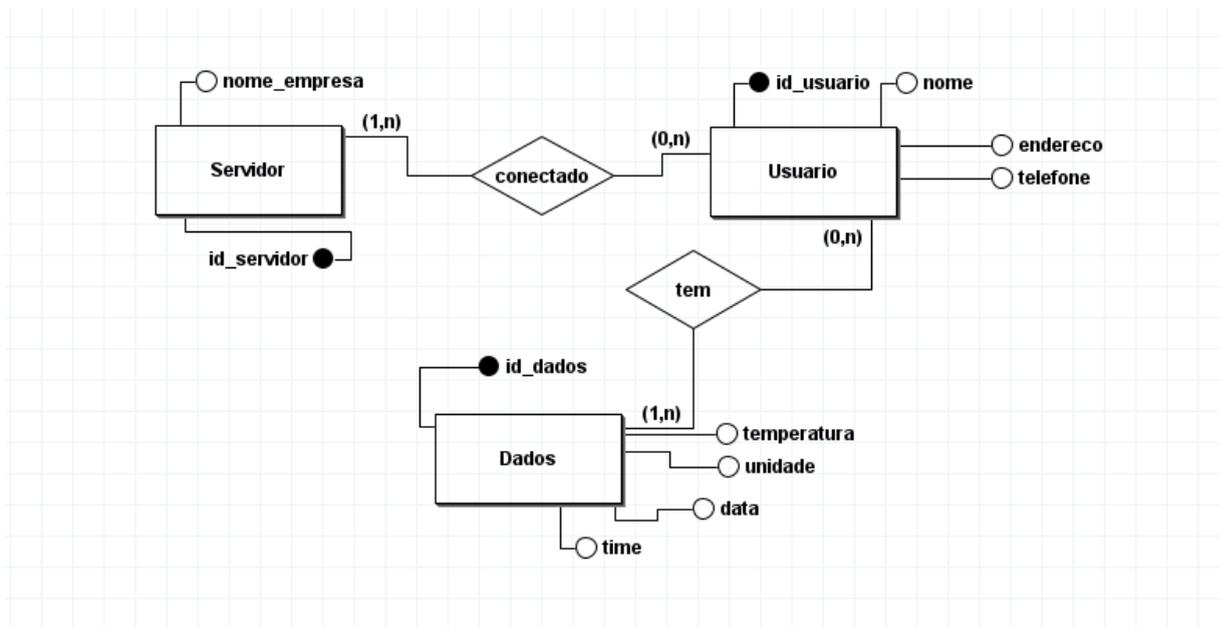
No âmbito da engenharia de software, a fase de desenvolvimento do projeto do banco de dados representa um elemento crucial na construção de sistemas eficientes e funcionais. Esta etapa, que transita desde a concepção do modelo lógico até a concretização do modelo físico, desempenha um papel fundamental na estruturação e organização da informação que será manipulada pelo sistema. Ao adentrar nesse processo, os estudantes são desafiados a traduzir os requisitos e especificações do sistema em uma representação coerente e otimizada no âmbito do banco de dados.

O ponto de partida reside na elaboração do modelo lógico, no qual se delineiam as entidades, seus relacionamentos e atributos, proporcionando uma visão abstrata e independente da plataforma. Este estágio inicial exige uma compreensão profunda dos requisitos do sistema, bem como a habilidade de estruturar de forma lógica a inter-relação entre os diferentes elementos do banco de dados.

Lehmkuhl e Eger “O objetivo da modelagem de dados é garantir que todos os objetos de dados existentes em determinado contexto e requeridos pela aplicação estejam representados com precisão dentro do Banco de Dados.”

3.3.1 MODELO CONCEITUAL

Figura 1 - Modelo conceitual exemplo de banco de dados:



Fonte: Grupo 7 ADS/CC

Neste modelo conceitual, buscamos fornecer uma visão abrangente das principais funcionalidades e elementos essenciais para o entendimento do nosso modelo no contexto do aplicativo BrModelo.

Principais Elementos:

1. Entidades:

- Servidor
- Usuário
- Dados

As entidades representam os principais componentes do nosso sistema, cada uma desempenhando um papel fundamental no fluxo de dados e interações.

2. Relacionamentos:

- Conectado
- Tem

Os relacionamentos definem as interconexões entre as entidades, estabelecendo as associações que moldam a dinâmica do sistema.

Detalhamento das Entidades:

1. Servidor:

- Esta entidade representa a infraestrutura física ou virtual que hospeda e gerencia os dados e serviços do sistema.

- Atributos:

- Nome_empresa

- Id_servidor (chave primária)

2. Usuário:

- Esta entidade representa os indivíduos ou entidades que interagem com o sistema, podendo acessar e manipular os dados conforme as permissões atribuídas.

- Atributos:

- Id_usuario (chave primária)

- Nome

- Telefone

- Endereço

3. Dados:

Descrição: Esta entidade encapsula as informações cruciais que são armazenadas e gerenciadas pelo sistema. Esses dados podem variar em natureza, abrangendo desde informações textuais até registros multimídia.

- Atributos:

- Id_dados (chave primária)

- Temperatura

- Time

- Data

- Umidade

Detalhamento dos Relacionamentos:

1. Conectado:

- O relacionamento "Conectado" estabelece a ligação entre um Usuário e um Servidor, indicando a associação entre o acesso de um usuário a um servidor.

- Cardinalidade: Um-para-Muitos

2. Tem:

- O relacionamento "Tem" representa a relação entre um Usuário e os Dados associados a ele. Define a posse ou controle que um Usuário tem sobre determinados conjuntos de dados.

- Cardinalidade: Um-para-Muitos

Esta é uma visão geral do nosso modelo conceitual no ambiente do aplicativo brModelo. Cada elemento e relacionamento desempenha um papel crucial na estrutura e funcionamento do sistema.

3.3.2 MODELO LÓGICO E FÍSICO

1. Tabela servidor:

- Campos:

- `idservidor` (INT): Este campo representa o identificador único de cada servidor.

- `nome_empresa` VARCHAR(45): Este campo armazena o nome da empresa associada a cada servidor.

- Relacionamento:

- A tabela `servidor` está conectada à tabela `dados` em um relacionamento de "um para muitos" (1:n), o que significa que um servidor pode ter vários dados associados a ele, mas cada dado pertence a apenas um servidor.

2. Tabela `dados`:

- Campos:

- `iddados` (INT): Este campo é o identificador único de cada conjunto de dados.

- `temperatura` VARCHAR(45): Aqui é onde a temperatura é registrada.

- `data` (DATE): Armazena a data em que os dados foram registrados.

- `time` (TIME): Armazena a hora do registro.

- Relacionamento:

- A tabela `dados` está conectada à tabela `servidor` através da chave `idservidor`.

Isso significa que cada conjunto de dados é associado a um único servidor.

3. Tabela `usuario`:

- Campos:

- `idusuario` (INT): Este campo é o identificador único de cada usuário.

- `nome` VARCHAR(45): Armazena o nome do usuário.

- `endereco` VARCHAR(45): Armazena o endereço do usuário.

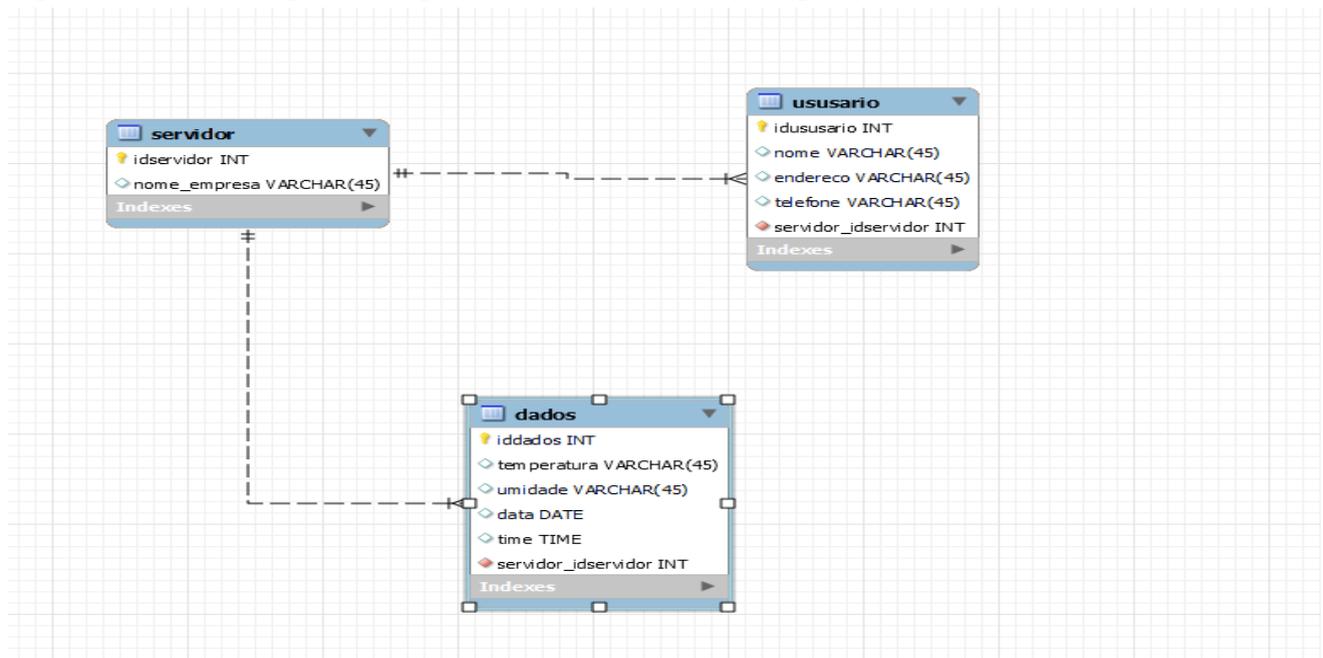
- `telefone` VARCHAR(45): Armazena o número de telefone do usuário.

- Relacionamento:

- A tabela `usuario` está conectada à tabela `servidor` em um relacionamento de "um para muitos" (1:n), o que significa que um servidor pode estar associado a vários usuários, mas cada usuário está associado a apenas um servidor.

Este modelo de banco de dados permite o armazenamento de informações sobre servidores, dados (incluindo temperatura, data e hora) e usuários. Os servidores podem estar associados a vários dados e a vários usuários. Cada dado está associado a um servidor específico e cada usuário está associado a um servidor específico também. Isso permite a organização e relacionamento de informações de forma eficiente.

Figura 2 - Modelo lógico exemplo de banco de dados no SqlWorkbench



Fonte: Grupo 7 ADS/CC

Neste modelo lógico e físico estamos entrando no banco de dados, onde criamos as tabelas e colunas que a partir delas criadas, damos sequência fazendo o código SQL no qual utilizando o (www.w3schools.com/MySQL/) conseguimos selecionar, deletar, atualizar e adicionar itens a uma tabela.

3.3.3 SQL

O protótipo do código SQL é como uma "pergunta" que fazemos a um banco de dados para obter informações específicas. É como pedir para o banco de dados nos mostrar uma lista de usuários junto com algumas informações sobre eles.

A parte `SELECT idusuario, nome, endereco, telefone, servidor_idservidor` diz ao banco de dados exatamente quais informações queremos ver. Neste caso, estamos pedindo o ID do usuário, o nome, o endereço, o telefone e o ID do servidor associado a cada usuário.

A parte `FROM usuário` indica de qual "tabela" queremos obter essas informações. Neste caso, a tabela se chama `usuário`. Uma tabela é como uma planilha onde as informações são organizadas.

Quando o código é executado, o banco de dados vai olhar para a tabela `usuario`, pegar as informações que pedimos (ID do usuário, nome, endereço, telefone e ID do servidor) e nos mostrar em um formato de lista. (Anexo 1)

Figura 3 - Modelo físico funcionando dentro do SQL:

	idusuario	nome	endereço	telefone	servidor_idservidor
▶	3	Rosangelo	Rua da saldade	19 992368721	1
	4	Rosangelo	Rua da saldade	19 992368721	2
	5	Rosangelo	Rua da saldade	19 992368721	3
	6	Rosangelo	Rua da saldade	19 992368721	4
	7	Rosangelo	Rua da saldade	19 992368721	5
	8	Rosangelo	Rua da saldade	19 992368721	6
	9	Rosangelo	Rua da saldade	19 992368721	7
	10	Rosangelo	Rua da saldade	19 992368721	8
	11	Rosangelo	Rua da saldade	19 992368721	9
	12	Rosangelo	Rua da saldade	19 992368721	10
▲	NULL	NULL	NULL	NULL	NULL

Fonte: Grupo 7 ADS/CC

3.4 GESTÃO FINANCEIRA

A gestão financeira desempenha um papel fundamental no sucesso de empresas modernas, e a Fast Lane, uma empresa especializada em sensores de temperatura e outros dispositivos eletrônicos, não é exceção. Embora a Fast Lane opere principalmente como uma loja virtual, sua equipe trabalha em um estabelecimento físico, onde a inovação e precisão são as palavras de ordem. Neste cenário, é crucial entender e administrar de forma eficaz os custos diretos e indiretos.

Os custos diretos da Fast Lane incluem a aquisição de peças para os sensores eletrônicos de última geração que produzem. A busca incessante por componentes de qualidade e tecnologicamente avançados é essencial para manter a competitividade. Além disso, os custos indiretos englobam os custos relacionados ao funcionamento do estabelecimento físico, como aluguel, energia e recursos humanos. Gerenciar esses aspectos financeiros com eficiência é um desafio constante.

Como o renomado estudioso financeiro Gonçalves (2022) disse uma vez, "Aprender a controlar seu orçamento é o modo mais prático de cortar gastos e começar a investir." Nesse contexto, a Fast Lane precisa garantir que suas operações financeiras sejam sólidas e bem embasadas. Agora, a pergunta que se impõe é se, apesar desses custos, a Fast Lane tem o potencial de prosperar no mercado altamente competitivo de sensores eletrônicos. Isso será explorado ao longo deste estudo, revelando se essa empresa está destinada a trilhar o caminho do sucesso.

Gestão financeira é o conjunto de processos, métodos e ações que permitem a uma empresa controlar, analisar e planejar suas atividades financeiras. Ela possibilita e fornece os recursos para que os profissionais especializados analisam cenários e traçam metas para a empresa melhorar seus resultados e como utiliza seus recursos. (TOTVS, 2023)

3.4.1 CLASSIFICAÇÃO DOS CUSTOS

Custos Diretos: são todos os custos ligados ao consumo de matéria-prima e mão de obra direta para a produção ou entrega do serviço.

- * KIT(Arduino uno, jumpers, protoboard) – R\$49,97
- * Sensor de Umidade e Temperatura DHT22 – R\$ 7,96
- * RESISTOR 10K – R\$ 4,00
- * Fonte de alimentação – R\$ 11,96
- * Placa PCB De Circuito Impresso: R\$ 4,56
- * Fio 1,5mm Cabo Flexível Rolo 1m: R\$ 1,30

Custos Indiretos: são aquelas que não têm uma ligação tão explícita com os serviços ou produtos oferecidos.

- * Conta de Energia R\$ 10,00
- * Conta de água R\$ 7,00
- * Marketing: R\$ 7,00
- * Aluguel e wifi: R\$ 57,00

Custos Variáveis vs. Custos Fixos:

* **Custos Variáveis:** São aqueles que variam de acordo com a produção ou vendas, como as peças para sensores. Eles somam os valores de todas as peças utilizadas na produção dos sensores de temperatura e umidade.

* **Custos Fixos:** São aqueles que permanecem relativamente constantes, independentemente do nível de produção ou vendas. Incluem o aluguel do estabelecimento, contas de energia e água, manutenção, limpeza, marketing e outros custos indiretos.

3.4.2 CUSTOS DO PROJETO

Pesquisas no mercado foram feitas e concluímos que de acordo com o site BassAutomação e entre outros, os preços começam em cerca de R\$6.000,00 para as configurações básicas e chegam a R\$30.000,00 para casas totalmente automatizadas com monitoramento. Nosso produto é altamente viável no mercado por ser um produto de custo mediano de R\$287,90 reais. Nosso produto pode oferecer um conjunto mais restrito de funcionalidades em comparação com soluções mais caras, o que justifica a diferença de preço, também buscamos tornar a automação acessível a um público mais amplo, expandindo o acesso a essas tecnologias. Estamos confiantes de que ele pode alcançar o sucesso e gerar resultados positivos para todas as partes envolvidas. Estamos ansiosos para embarcar nessa jornada e demonstrar o potencial do nosso produto no mercado.

3.4.3 PRECIFICAÇÃO

Para determinar o preço de venda, é necessário considerar tanto os custos diretos quanto os custos indiretos, além de incluir uma margem de lucro para garantir a sustentabilidade do negócio. Aqui está uma sugestão de cálculo:

1. Total de Custos Diretos Por Unidade: R\$ 79,75
2. Total de Custos Indiretos Por Unidade: R\$ 79,00
3. Margem de Lucro: (custos totais (diretos + indiretos))
Margem de Lucro (32,70%):** 1,327
4. Impostos: R\$ 15,00
5. Comissão: R\$ 20,00
6. Cálculo do Preço de Venda:

Preço de Venda = (Custos Diretos + Custos Indiretos) x Margem de Lucro

Preço de Venda = (R\$ 79,75 + R\$ 79,00 + R\$15,00 + R\$20,00) x 1,327

Visando os nossos custos para ter o nosso produto e manter o nosso negócio, o preço foi calculado da seguinte forma, na parte de custo fixo o rendimento foi distribuído em 3 partes, sendo: R\$ 500,00 de aluguel, R\$50,00 de Wi-fi e R\$240,00 em energia elétrica/água/marketing, resultando no total de R\$ 790,00 reais em custo fixo. O preço ficou em torno de R\$221,80 reais.

Figura 4 - Tabela da formação de preço do projeto

Formação de Preço									
Custo Fixo	Custo Variável			Custo Total	Preço	Margem de Contribuição	Margem	Mark Up	Índice de Markup
	Matéria Prima	Comissão	Impostos						
R\$ 79,00	R\$ 79,75	R\$ 20,00	R\$ 15,00	R\$ 193,75	R\$ 287,90	R\$94,15	32,70%	48,59%	1,49

Fonte: Grupo 7 ADS/CC

| Custo Fixo: R\$79,00 Despesas que permanecem constantes, independentemente da quantidade de bens ou serviços produzidos ou vendidos. Exemplos incluem aluguel, salários administrativos e seguro. |

| Matéria Prima: R\$ 79,75 | Os materiais essenciais utilizados na produção de um produto ou serviço. Eles são transformados durante o processo de produção para criar o produto final. |

| Comissão: R\$20,00 | Uma porcentagem do valor de uma transação de venda que é paga a um vendedor ou intermediário como forma de compensação pelo seu papel na venda. |

| Impostos: R\$ 15,00 | Pagamentos obrigatórios ao governo sobre a renda, propriedade ou bens e serviços. Eles são usados para financiar os serviços públicos e programas governamentais. |

| Custo Total: R\$ 193,75 | A soma de todos os custos associados à produção de um bem ou serviço, incluindo custos fixos e variáveis, matéria-prima, mão de obra, etc. |

| Preço R\$:287,90 | O valor monetário pelo qual um produto ou serviço é vendido no mercado. É determinado por vários fatores, incluindo custos, demanda e concorrência. |

| Margem de Contribuição R\$: 94,15 | A diferença entre o preço de venda de um produto e seus custos variáveis (como matéria-prima e mão de obra direta). É usado para cobrir os custos fixos e gerar lucro. |

| Margem: 32,7% | A porcentagem de lucro em relação ao preço de venda. Pode ser expressa como uma porcentagem do preço de venda ou como uma porcentagem do custo. |

| Markup: 48,59% | A diferença entre o custo de um produto e seu preço de venda. É expresso como uma porcentagem do custo e é usado para determinar o preço de venda. |

| Índice de Markup: 1,49 | Uma medida que indica quantas vezes o custo de um produto é marcado para determinar seu preço de venda. É calculado dividindo o preço de venda pelo custo. Quanto maior o índice, maior o markup. |

| Markup: 26,20% | A diferença entre o custo de um produto e seu preço de venda. É expresso como uma porcentagem do custo e é usado para determinar o preço de venda. |

| Índice de Markup: 1,26 | Uma medida que indica quantas vezes o custo de um produto é marcado para determinar seu preço de venda. É calculado dividindo o preço de venda pelo custo. Quanto maior o índice, maior o markup. |

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS

A Formação para a Vida é um dos eixos do Projeto Pedagógico de Formação por Competências da UNIFEOB.

Esta parte do projeto está diretamente relacionada com a extensão universitária, ou seja, o objetivo é que seja aplicável e que tenha real utilidade para a sociedade, de um modo geral.

3.5.1 GERENCIANDO FINANÇAS

- **Tópico 1:**

A compreensão dos conceitos econômicos e financeiros é essencial para indivíduos e sociedades navegarem eficientemente em um mundo cada vez mais interconectado. Esses conceitos formam a base para as decisões que impactam diretamente o nosso cotidiano, desde escolhas pessoais até as políticas macroeconômicas adotadas por governos.

1. Oferta e Demanda:

Um dos pilares fundamentais da economia é a lei da oferta e demanda. Em termos simples, a oferta refere-se a quantidade de um bem ou serviço disponível, enquanto a demanda é a quantidade que os consumidores estão dispostos a adquirir a determinado preço. Por exemplo, durante períodos de alta demanda por um produto, seu preço tende a aumentar, incentivando mais produção e equilibrando o mercado

2. Inflação:

A inflação representa aumento geral dos preços de bens e serviços ao longo do tempo, se a inflação é alta, o dinheiro perde valor, impactando diretamente o custo de vida. Na prática, seria possível observar como podem subir os preços de itens básicos.

3. Juros:

Os juros são uma compensação pelo uso do dinheiro ao longo do tempo. Taxas de juros impactam desde o custo de empréstimos até os ganhos de investimentos. Assim como, ao fazer um empréstimo, os juros aumentam o valor final a ser pago.

4. Orçamento Pessoal:

O orçamento pessoal é essencial para o equilíbrio financeiro, envolve o acompanhamento dos gastos e receitas, garantindo a eficiência como o dinheiro é alocado. No dia a dia, isso entraria no controle efetivo das finanças pessoais.

5. Mercado de ações:

O mercado de ações é um componente vital da atividade econômica, compreender o funcionamento de ações e seu impacto na economia é crucial, pois afeta empresas e investidores. No entanto, o desempenho de uma empresa na bolsa pode atrair investidores, proporcionando capital para expansão.

- **Tópico 2:**

A administração financeira pessoal é comparada à gestão financeira de uma empresa, onde o objetivo principal é maximizar a riqueza. Para alcançar independência financeira, é crucial compreender e gerenciar as fontes de renda, como salários, investimentos e outras formas de ganhos. A liquidez, representando o dinheiro disponível após o pagamento de despesas, é essencial na gestão financeira pessoal.

Para gerar resultados financeiros positivos, é necessário equilibrar receitas e despesas. Reduzir custos é fundamental, e estratégias como evitar dívidas bancárias, renegociar fornecedores, controlar gastos e otimizar processos podem contribuir para esse objetivo. Além disso, investir sabiamente, seja em propriedades, ações, renda fixa ou outros ativos, é uma maneira de aumentar a riqueza ao longo do tempo.

O perfil do investidor desempenha um papel importante na escolha de estratégias de investimento, variando de conservador a agressivo. A compreensão dos conceitos de investimento, como renda fixa e variável, é crucial para tomar decisões informadas.

Manter registros financeiros detalhados, assim como as empresas elaboram demonstrações contábeis, é essencial para avaliar o desempenho, identificar áreas de melhoria e tomar decisões informadas. A cultura de criar relatórios financeiros regulares facilita a análise e a tomada de decisões eficazes.

Alcançar a independência financeira envolve a gestão cuidadosa das finanças pessoais, a geração de receitas, a redução de custos, o investimento inteligente e a manutenção de registros financeiros precisos para uma tomada de decisão informada.

- **Tópico 3:**

Nesse tópico vimos como a Matemática Financeira estuda o valor do dinheiro no tempo. Os objetivos dela podem ser considerados sob dois aspectos: de quem aplica (procura taxas de juros mais altas) e de quem capta (procura taxas de juros mais baixas).

Existem dois tipos de juros em uma operação financeira: o simples e o composto. O primeiro, assim como seu nome diz, é mais fácil de ser compreendido.

No Juros Simples, apenas o capital inicial será usado no cálculo dos juros durante a aplicação. Temos assim que caso uma pessoa faça uma aplicação de R\$ 100,00 por 6 meses, a uma taxa de 10% teremos como resultado um juros de 10 reais ao mês, ou seja no primeiro mês será R\$100,00 + R\$10,00 (de juros) totalizando R\$110,00, no mês subsequente teremos os mesmos R\$ 100,00 da aplicação + a taxa de 10% + o mês anterior totalizando R\$120,00, no fim dos 6 meses teremos então um saldo total de R\$160,00.

Para facilitar ainda mais temos uma formula para realizar o cálculo: $VF = VP + VP * i * n$.

Onde:

VF = valor futuro

VP = valor presente

i = taxa de juros

n = quantidade de períodos

Já o Juros Compostos determina que o juros de períodos anteriores são adicionados ao capital inicial e em cima desse novo valor é refeito o cálculo. Temos assim o exemplo de que uma pessoa faça uma aplicação de R\$ 100,00 por 6 meses, a uma taxa de 10% ao mês, ou seja no primeiro mês será R\$100,00 + R\$10,00 (de juros) totalizando R\$110,00, no mês subsequente teremos R\$ 110,00 da aplicação + a taxa de 10% totalizando R\$121,00, no fim dos 6 meses teremos então um saldo total de R\$177,16.

Para facilitar temos uma formula para realizar o cálculo: $VF = VP * (1+i)^n$.

Onde:

VF = valor futuro

VP = valor presente

i = taxa de juros

n = quantidade de períodos

Temos também a Possibilidade de crédito, onde a pessoa vai atrás de um empréstimo para pagar algo, nesse caso o Doador deve realizar uma avaliação para descobrir se deve realizar a concessão de crédito para novos clientes, nessa avaliação o Doador vai atrás de

descobrir referências bancárias; obtenção de garantias reais. Já para o Tomador do crédito, os fatores analisados devem ser: qual o capital que se pretende pegar emprestado.

- **Tópico 4:**

Construir um caminho sólido no presente é crucial para garantir estabilidade e conforto no futuro. Tanto na gestão empresarial quanto na financeira pessoal, é fundamental adotar hábitos e controles para evitar surpresas desagradáveis.

A gestão financeira requer equilíbrio, com um planejamento sólido e controles eficientes. Sonhos, sejam simples ou ambiciosos, ressaltam a importância da educação financeira, que desempenha papel crucial na conquista de objetivos.

A realização de sonhos exige um projeto bem elaborado, com prioridades definidas e metas estabelecidas. O comprometimento com a organização financeira é essencial para construir um futuro sólido e recompensador.

3.5.2 ESTUDANTES NA PRÁTICA

Após realizar a síntese dos conteúdos, é hora dos estudantes colocarem a “mão na massa”, de produzirem algo da formação para a vida que possa ser utilizado na prática pela sociedade.

Figura 5 - Banner



BOAS PRÁTICAS FINANCEIRAS

CONQUISTE SUA LIBERDADE FINANCEIRA: DICAS PARA UM FUTURO FINANCEIRO SÓLIDO!

- ### 1. DIVERSIFIQUE SUAS FONTES DE RENDA.

"Diversificar Fontes de Renda" significa criar diferentes fontes de ganho além do salário, como investimentos ou empreendimentos secundários. Essa prática aumenta a estabilidade financeira, reduzindo a dependência de uma única fonte de renda.
- ### 2. REDUZA DESPESAS DESNECESSÁRIAS.

"Reduza Despesas Desnecessárias" sugere a importância de identificar e cortar gastos não essenciais, otimizando o orçamento. Ao eliminar despesas supérfluas, é possível economizar recursos financeiros para direcioná-los a objetivos mais significativos.
- ### 3. INVISTA DE FORMA INTELIGENTE.

"Invista de Forma Inteligente" destaca a importância de tomar decisões financeiras estratégicas ao aplicar dinheiro. Isso envolve escolher investimentos alinhados com metas e tolerância ao risco, buscando maximizar ganhos e minimizar riscos.
- ### 4. CONSTRUA UM FUNDO DE EMERGÊNCIA.

"Construa um Fundo de Emergência" significa reservar uma quantia financeira para imprevistos, oferecendo segurança em situações inesperadas. Esse fundo atua como uma rede de proteção financeira, proporcionando estabilidade durante crises e evitando endividamento desnecessário.

 Fast Lane

4. CONCLUSÃO

Após concluirmos as propostas para o projeto proposto por cada unidade de estudo, conseguimos desenvolver várias habilidades técnicas. Com nossos estudos em programação orientada a objeto, aprendemos a desenvolver um código que está relacionado ao dispositivo do sensor de temperatura e umidade. Também, por conta disso, fizemos com que nele sejam executados os comandos. Então, tivemos que criar tanto uma parte para apresentar informações, quanto outra para fazer a leitura da temperatura e da umidade.

Através das habilidades que foram desenvolvidas em lógica da programação, conseguimos construir um sistema que apresenta toda a parte front-end do site do nosso sensor, bem como um sistema de back-end para registrar os dados e executar funções. Tivemos que construir um site com todas as informações do cliente e opções para ele escolher como o sensor vai funcionar. Aprendemos, através dos conhecimentos obtidos na matéria de modelagem de banco de dados, a criar uma estrutura para registrar todos os dados que nosso sistema vai precisar e fazer a ligação entre eles.

Em seguida, tivemos que criar um banco que fizesse a ligação entre os dados preenchidos pelo cliente e a temperatura e a umidade que representam o funcionamento do sensor. No estudo de gestão financeira, aprendemos a gerenciar e calcular todos os gastos e lucros que teremos na nossa empresa, como o que teremos que arcar para construir os aparelhos e suas manutenções, além das vendas que teremos. Através disso, estamos trilhando um caminho melhor em direção ao sucesso financeiro.

REFERÊNCIAS

ALURA. **Lógica de programação**. Disponível em:

<https://www.alura.com.br/formacao-programacao> Acesso em: 23 nov. 2023

ESTRATÉGIA. **Programação orientada a objeto**. Disponível em:

<https://www.estrategiaconcursos.com.br/blog/programacao-orientada/> Acesso em: 23 nov 2023

FRANÇA, R. SQL: **O que é e como usar os principais comandos básicos SQL**.

Disponível em: <https://blog.betrybe.com/sql/#1> Acesso em 23 nov.2023

GONÇALVES, Douglas. **Aprender a controlar seu orçamento é o modo mais prático de cortar gastos e começar a investir**. 2022.

GRUPO TOTVS. **Gestão Financeira**: O que é, para que serve e dicas. 2023. Disponível

em: <https://www.totvs.com/blog/servicos-financeiros/gestao-financeira/> Acesso em: 23 nov. 2023.

ANEXOS

Anexo 1

3.3.3

-- INSERIR servidores

```
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('01', 'MALTITENIS');
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('02', 'FLUAMBURGO');
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('03', 'MARCOS');
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('04', 'CCNETGRAPH');
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('05', 'TEMASNETON');
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('06', 'CSGOINTERNET');
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('07', 'LOLPOKANET');
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('08', 'TANTENET');
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('09', 'ENPATANET');
INSERT INTO servidor (idservidor, nome_empresa) VALUES ('10', 'INFRATENIS');
```

```
SELECT * FROM servidor;
DELETE FROM servidor;
```

-- INSERIR dados

```
INSERT INTO dados (temperatura, umidade) VALUES ('Alta', 'Baixa');
INSERT INTO dados (temperatura, umidade) VALUES ('media', 'baixa');
INSERT INTO dados (temperatura, umidade) VALUES ('baixa', 'baixa');
INSERT INTO dados (temperatura, umidade) VALUES ('baixa', 'alta');
INSERT INTO dados (temperatura, umidade) VALUES ('alta', 'media');
INSERT INTO dados (temperatura, umidade) VALUES ('media', 'media');
INSERT INTO dados (temperatura, umidade) VALUES ('baixa', 'alta');
INSERT INTO dados (temperatura, umidade) VALUES ('alta', 'alta');
INSERT INTO dados (temperatura, umidade) VALUES ('baixa', 'baixa');
INSERT INTO dados (temperatura, umidade) VALUES ('baixa', 'media');
```

```
SELECT * FROM dados;
DELETE FROM dados;
```

-- INSERIR usuario

```
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '01');
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '02');
```

```
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '03');
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '04');
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '05');
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '06');
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '07');
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '08');
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '09');
INSERT INTO usuario (nome, endereco, telefone, servidor_idservidor) VALUES
('Rosangelo', 'Rua da saldade', '19 992368721', '10');
```

```
SELECT * FROM usuario;
DELETE FROM usuario;
```

Anexo 2:

[Código Sensor](#)