



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
<ALLTOMATIC>

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
<ALLTOMATIC>

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 – Prof^ª. Mariângela
Martimbianco Santos

Estudantes:

Angelo Marcos R. Ridolfi Carvalho, RA 23001021
Caio Henrique Sibrão da Cruz, RA 23000666
Gabriel Marcos Freire, RA 23000669
Luis Eduardo Leme Silva, RA 23000607
Matheus Guimaraes de Lima C. Gonçalves, RA 23001120
Renan Ruy Rodrigues Reis, RA 23000704

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2023

SUMÁRIO

1. INTRODUÇÃO	4
2. DESCRIÇÃO DA EMPRESA	4
3. PROJETO INTEGRADO	5
3.1 PROGRAMAÇÃO ORIENTADA A OBJETO	5
3.1.1 CLASSES E OBJETOS	5
3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA.	6
3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS	10
3.2 LÓGICA DE PROGRAMAÇÃO	11
3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE	12
3.2.2 DESENVOLVIMENTO DE APLICAÇÕES DESKTOP	12
3.3 MODELAGEM DE DADOS	14
3.3.1 MODELO CONCEITUAL	14
3.3.2 MODELO LÓGICO E FÍSICO	15
3.3.3 SQL	17
3.4 GESTÃO FINANCEIRA	20
3.4.1 CLASSIFICAÇÃO DOS CUSTOS	20
3.4.2 CUSTOS DO PRODUTO	20
3.4.3 PRECIFICAÇÃO	22
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS	22
3.5.1 GERENCIANDO FINANÇAS	22
3.5.2 ESTUDANTES NA PRÁTICA	23
4. CONCLUSÃO	24
REFERÊNCIAS	25

1. INTRODUÇÃO

Nosso objetivo é melhorar o estudo e cultivo sobre as plantas, promovendo-o de uma maneira mais simples e eficaz, para o consumidor.

O foco da nossa empresa é na automatização de estufas pequeno, médio e grande porte, controlado remotamente tanto para estudo e análise do comportamento da planta como para plantação própria, e criar um banco de dados em que contém diversas programações de ciclos de plantio para ajudar pessoas inexperientes a otimizar suas colheitas.

Nossa empresa visa aumentar a sustentabilidade ambiental e a produção agrícola com isso reduzindo a utilização de agrotóxicos. Facilitando o controle de sua estufa, automatizando as funções de iluminação e umidade, com um controle totalmente remoto, sem precisar estar do lado de suas plantas.

Decidimos criar o projeto de estufa, baseado na ideia de facilitar e ajudar na área de plantio, desenvolvimento e pesquisas de plantas, podendo tornar esse meio mais fácil e com maior possibilidade de experimentos, como recriação de ambientes, testes de estresse, ajuda de um futuro banco de dados para agilizar pesquisas e estudos, entre muitas outras vantagens presentes no plantio em estufa.

2. DESCRIÇÃO DA EMPRESA

A empresa para qual o serviço está sendo prestado é o Centro Universitário Fundação De Ensino Octávio Bastos, UNIFEQB, CNPJ 59.765.555/0002-33, com limite territorial circunscrito ao município de São João da Boa Vista, Estado de São Paulo, podendo atuar em todo o território nacional, é instituição de ensino superior mantida pela Fundação de Ensino Octávio Bastos, pessoa jurídica de direito privado, sem fins lucrativos, com sede e foro na Av. Dr. Octávio da Silva Bastos, 2439 - Jardim Nova São João, na cidade de São João da Boa Vista, Estado de São Paulo.

Sua atividade social econômica principal é a Educação De Ensino Superior - Graduação. A Unifeob é um centro universitário que visa entregar ao mercado bons

profissionais. Suas estruturas são modernas e respondem aos principais desafios educacionais do século XXI. Parcerias com grandes nomes da tecnologia garantem o acesso às principais ferramentas encontradas no mercado de trabalho. Aprendizagem e aprimoramento contínuos, inovação, engajamento do time, transparência e sucesso do cliente.

3. PROJETO INTEGRADO

3.1 PROGRAMAÇÃO ORIENTADA A OBJETO

Com a programação orientada a objeto, aprendemos que são estruturas de dados que contêm tanto dados quanto os métodos que operam nesses dados. O objetivo da POO é modelar o mundo real de forma mais fiel, permitindo a representação de entidades, suas interações e hierarquias. Isso resulta em um código mais modular, reutilizável e de fácil manutenção, promovendo a encapsulação, herança e polimorfismo como conceitos-chave. Essa abordagem é amplamente utilizada em linguagens de programação como Java, C++, Python e muitas outras.

Segundo FÉLIX (2016) “A habilidade de desenvolver aplicações para computadores é cada vez mais valiosa para a indústria do século XXI. O desenvolvimento de soluções mais sofisticadas exige uma correta compreensão da arquitetura de software e de diversos conceitos, como a programação orientada a objetos.”

3.1.1 CLASSES E OBJETOS

Classe e objeto são conceitos fundamentais na programação orientada a objetos (POO), um paradigma de programação popular utilizado em muitas linguagens de programação. Vamos explorar o que são classes e objetos:

Classe:

- Uma classe é um modelo ou um molde para criar objetos. Ela define um conjunto de atributos (propriedades) e métodos (funções) que os objetos criados a partir da classe terão.
- Em uma classe, você especifica a estrutura e o comportamento dos objetos dessa classe.
- Ela serve como um modelo para criar múltiplos objetos com características e comportamentos semelhantes.
- Classes são usadas para modelar e representar entidades do mundo real ou conceitos abstratos em seu software.

Objeto:

- Um objeto é uma instância de uma classe. Ele é uma unidade concreta, autocontida criada a partir de uma definição de classe.
- Objetos têm seu próprio estado único (valores de atributos) e podem executar ações (métodos) definidas pela classe.
- Objetos permitem que você trabalhe e manipule dados de forma estruturada e organizada.
- Eles são os principais blocos de construção do seu software e representam as entidades em tempo de execução em seu programa.

3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA.

Atributos (ou Propriedades):

- Atributos são variáveis ou campos que pertencem a um objeto. Eles representam características ou dados associados a um objeto.

Figura 1 - Atributos

```
this.nome = '';  
this.sobrenome = '';  
this.dataNascimento = '';  
this.enderecoCep = '';  
this.enderecoLogradouro = '';  
this.enderecoBairro = '';  
this.enderecoNum = '';  
this.enderecoComplemento = '';  
this.cpf = '';
```

Exemplo de atributos utilizados no desenvolvimento

Fonte: Autores

Métodos:

- Métodos são funções ou ações que um objeto pode executar. Eles representam o comportamento ou as ações que um objeto pode realizar. Exemplo:

Figura 2 - Métodos

```
async showName(data) {  
  await this.setDados();  
  data.innerHTML = `Nome: ${this.nome} ${this.sobrenome}`;  
}  
async showPassword(data) {  
  await this.setDados();  
  data.innerHTML = `Senha: *****`;  
}  
async showEmail(data) {  
  await this.setDados();  
  data.innerHTML = `Email: ${this.usuario}`;  
}  
async showDatas(dataName, dataNameAPI, data) {  
  await this.setDados();  
  if (dataNameAPI === 'dataNascimento') {  
    data.innerHTML = `${dataName}: ${this.convertDate(this.dataNascimento)}`;  
  } else {  
    data.innerHTML = `${dataName}: ${this[dataNameAPI]}`;  
  }  
}  
  
async showNameId(dataName, dataId) {  
  await this.setDados();  
  dataName.innerHTML = `${this.nome} ${this.sobrenome}`;  
  dataId.innerHTML = `#${this.id}`;  
}
```

Exemplo de Métodos usados no desenvolvimento do nosso aplicativo

Fonte: Autores

Figura 3 - Utilização dos Métodos

```
const user = localStorage.getItem('usuario');
const password = localStorage.getItem('senha');

const dataUser = new ShowDatas(user, password);
dataUser.showNameId(nomeUser, idUser);
dataUser.showName(nomeUsuario);
dataUser.showPassword(senha);
dataUser.showEmail(email);
dataUser.showDatas('CPF', 'cpf', cpf);
dataUser.showDatas('Data de Nascimento', 'dataNascimento', dataNascimento);
dataUser.showDatas('Rua', 'enderecoLogradouro', rua);
dataUser.showDatas('Bairro', 'enderecoBairro', bairro);
dataUser.showDatas('CEP', 'enderecoCep', cep);
dataUser.showDatas('Número', 'enderecoNum', numero);
dataUser.showDatas('Complemento', 'enderecoComplemento', complemento);
dataUser.setDados();
```

Exemplo de Utilização dos métodos usados no desenvolvimento do aplicativo

Fonte: Autores

Encapsulamento:

- O encapsulamento é um conceito que visa ocultar os detalhes internos de uma classe ou objeto e fornecer uma interface para interagir com ele. Isso ajuda a proteger os atributos e métodos, permitindo apenas o acesso e a modificação controlados dos dados. Geralmente, é implementado usando modificadores de acesso, como público, privado e protegido.

Figura 4 - Encapsulamento

```
class Login {
  constructor(req, res) {
    this.req = req;
    this.res = res
  }

  async AuthUser(req, res) {
    const user = req.body;
    const exist = await this.#verification(user);
    if (!exist || exist.length === 0) {
      return res.status(400).json({ msg: "User not found!" });
    } else {
      return res.status(200).json(exist);
    }
  }

  async #verification(user) {
    return await userModel.authUser(user)
  }
};
```

```
router.post('/auth/User', loginMiddleware , (req, res) => {  
  login.AuthUser(req, res);  
})
```

Exemplo de Utilização do método privado em uma classe encapsulada

Fonte: Autores

Herança:

- Herança é um conceito que permite que uma classe herde atributos e métodos de outra classe. Uma classe derivada (subclasse) pode estender ou especializar uma classe base (superclasse). Isso promove a reutilização de código e permite a criação de hierarquias de classes. Em resumo, atributos representam dados, métodos representam ações, encapsulamento ajuda a controlar o acesso aos dados e herança permite que as classes compartilhem estrutura e comportamento. Esses conceitos são fundamentais na programação orientada a objetos e são usados para modelar objetos do mundo real de forma eficaz em software.

Figura 5 - Herança:

```
import UserLogado from './UserLogado.js';  
  
export default class ShowDatas extends UserLogado {  
  constructor(usuario, senha) {  
    super(usuario, senha);  
    this.nome = '';  
    this.sobrenome = '';  
    this.dataNascimento = '';  
    this.enderecoCep = '';  
    this.enderecoLogradouro = '';  
    this.enderecoBairro = '';  
    this.enderecoNum = '';  
    this.enderecoComplemento = '';  
    this.cpf = '';  
  }  
}
```

Exemplo de Herança utilizado no desenvolvimento do Aplicativo

Fonte: Autores

Figura 6 - Classe Mãe

```
export default class UserLogado {
  constructor(usuario, senha) {
    this.usuario = usuario;
    this.senha = senha;
    this.url = 'http://186.235.2.225/auth/User';
    this.dados = '';
  }

  async user() {
    try {
      const response = await fetch(this.url, {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({
          username: this.usuario,
          password: this.senha,
        }),
      });
      const json = await response.json();
      this.dados = json[0];
    } catch (err) {
      console.log('Erro na requisição');
    }
  }
}
```

Exemplo de Classe Mãe onde ela passa seus atributos e métodos para uma classe Herdeira

Fonte:Autores

3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

Métodos Estáticos:

- Métodos estáticos são métodos associados a uma classe, não a instâncias de objetos.
- Eles podem ser chamados diretamente na classe, sem a necessidade de criar uma instância da classe.
- Geralmente, são usados para funções utilitárias ou métodos que não dependem do estado de um objeto específico.

Métodos Públicos:

- Métodos públicos são métodos de uma classe que podem ser acessados e chamados a partir de instâncias dessa classe.
- Eles são parte da interface pública da classe e podem ser usados para interagir com os objetos da classe.

Métodos Privados:

- Métodos privados são métodos de uma classe que não são acessíveis a partir de instâncias da classe.
- Eles são usados para encapsular a lógica interna da classe, ocultando-a do mundo externo.
- Geralmente, são precedidos por um modificador de acesso, como "private" (em linguagens que suportam esse recurso).

Em resumo, métodos estáticos pertencem à classe e não às instâncias, métodos públicos são acessíveis a partir de instâncias e fazem parte da interface pública, enquanto métodos privados são ocultos e destinados a serem usados apenas internamente dentro da classe. Esses conceitos ajudam a controlar o acesso e a organização do código em programação orientada a objetos.

3.2 LÓGICA DE PROGRAMAÇÃO

Nessa matéria, aprendemos como funciona a lógica por trás dos computadores e como aplicá-la na prática. Foi abordado 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 trabalhamos com prototipação, criação de aplicações desktop com o framework Electron JS, junto ao banco de dados relacional (MySQL).

Segundo Manzano e Oliveira(2021 p. 32) “Para usar o raciocínio lógico, é necessário ter domínio de pensar, bem como saber pensar e usar a “arte de pensar[...]”. Desta forma, utilizando o conhecimento de lógica de programação foi realizado o desenvolvimento de uma aplicação Desktop, totalmente desenvolvida em linguagem de marcação HTML, CSS e Electron JS que utiliza de uma Rest API desenvolvida em NodeJS e MySQL para o cadastro e realização de requisições de dados da estufa e dos usuários cadastrados.

3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

Visto os fundamentos de lógica de programação, foi adquirido o conhecimento em algoritmos que é o conjunto de instruções e regras que um computador utiliza para realizar tarefas, sobre variáveis que são nomes utilizados para definir valores que são salvos na memória do computador, tipos de dados, classificados em números inteiros, números flutuantes, booleanos, que contém apenas dois possíveis valores, como verdadeiro ou falso, caracteres, strings, que são formadas por uma cadeia de caracteres, formando frases, nomes ou valores e arrays, que são formadas por um conjunto de dados.

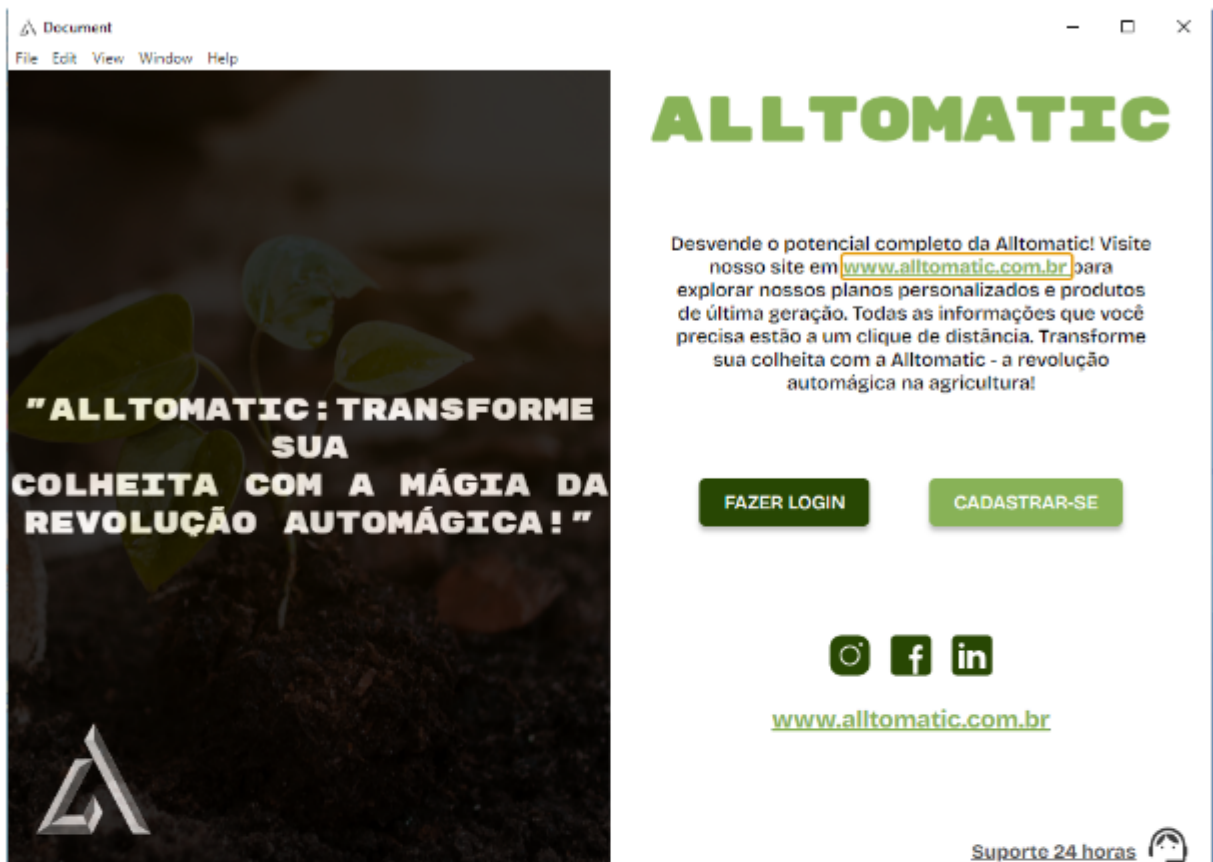
Como ainda, fora visto sobre funções, estruturas condicionais e operadores. Onde, por si, as funções encapsulam um trecho de um código ou algoritmo que realiza uma tarefa e pode ser invocado ou chamado em outro trecho do código, as estruturas condicionais analisam condições para determinam o fluxo que o código irá percorrer para entregar aquilo que foi lhe pedido e a, por fim, os operadores, sendo eles lógicos, que determinam como o código irá se comportar em uma determinada situação, aritméticos que realizam operações matemáticas, como soma, subtração, divisão, multiplicação, exponenciação e etc. E os relacionais que trabalham com verificação dos valores dos operandos e assim, retornam se a condição é verdadeira ou falsa.

3.2.2 DESENVOLVIMENTO DE APLICAÇÕES DESKTOP

Para o desenvolvimento de aplicações desktop foi utilizado o framework Electron Js, que é capaz de utilizar linguagens web como HTML, CSS e JavaScript.

No projeto ele foi essencial para desenvolvimento do front-end, aprimorando as interações das interfaces entre si. Abaixo está a imagem de duas telas desenvolvidas com essa tecnologia:

Figura 7 e 8 - Aplicativo Desktop



Exemplo de duas telas desenvolvidas com o Electron JS. HTML e CSS sendo a primeira tela de login ou cadastro, e a segunda já dentro do Aplicativo mostrando a temperatura dos sensores na data pesquisada

Fonte: Autores

3.3 MODELAGEM DE DADOS

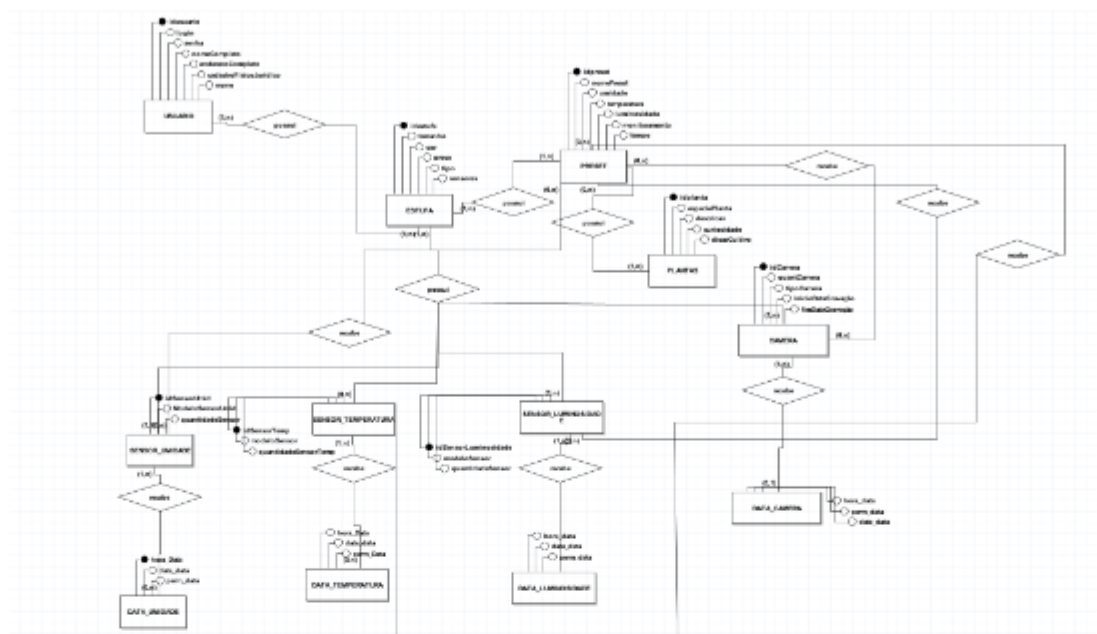
Na disciplina de Modelagem de Dados, são explorados conceitos fundamentais para a criação de estruturas lógicas e a implementação de bancos de dados, utilizando linguagens como MySQL. Esse aprendizado abrange a construção de modelos lógicos, compreendendo diagramas entidade-relacionamento (DER), normalização de dados e projetos conceituais. Além disso, os alunos são instruídos sobre a criação de tabelas, consultas SQL, índices, chaves estrangeiras e procedimentos armazenados no contexto do MySQL, proporcionando habilidades para o desenvolvimento e gestão de sistemas de banco de dados relacionais.

De acordo com Elmasri e Navathe (2010) “A verdadeira riqueza de uma empresa está em seus dados, e um banco de dados eficaz é a chave para desbloqueá-la”. Visto isso, foi idealizado via modelo conceitual o banco de dados do projeto. Após, foi produzido o modelo lógico, para entender quais tabelas e colunas seriam necessárias, e quais dados iriam ser guardados em suas respectivas tabelas, e por fim, foi criado o modelo físico que está sendo utilizado na aplicação.

3.3.1 MODELO CONCEITUAL

O problema principal enfrentado foi a distribuição de sensores e seus dados. Pois, para armazenar os respectivos dados necessitaria de uma tabela para cada sensor. Foi desenvolvido um modelo conceitual onde havia um tabela que guardava os dados de todos os sensores e eram relacionados com suas chaves estrangeiras. Após pesquisar as melhores maneiras de se desenvolver o banco, a equipe priorizou o modelo onde cada sensor teria uma tabela individual para armazenar os dados. E assim, foi escolhido as seguintes tabelas para o projeto: usuário, estufa, configuração, tipos de sensores, sensores, e uma tabela para cada dado de seu respectivo sensor.

Figura 9 - Modelo Conceitual



Primeiro modelo conceitual da equipe;

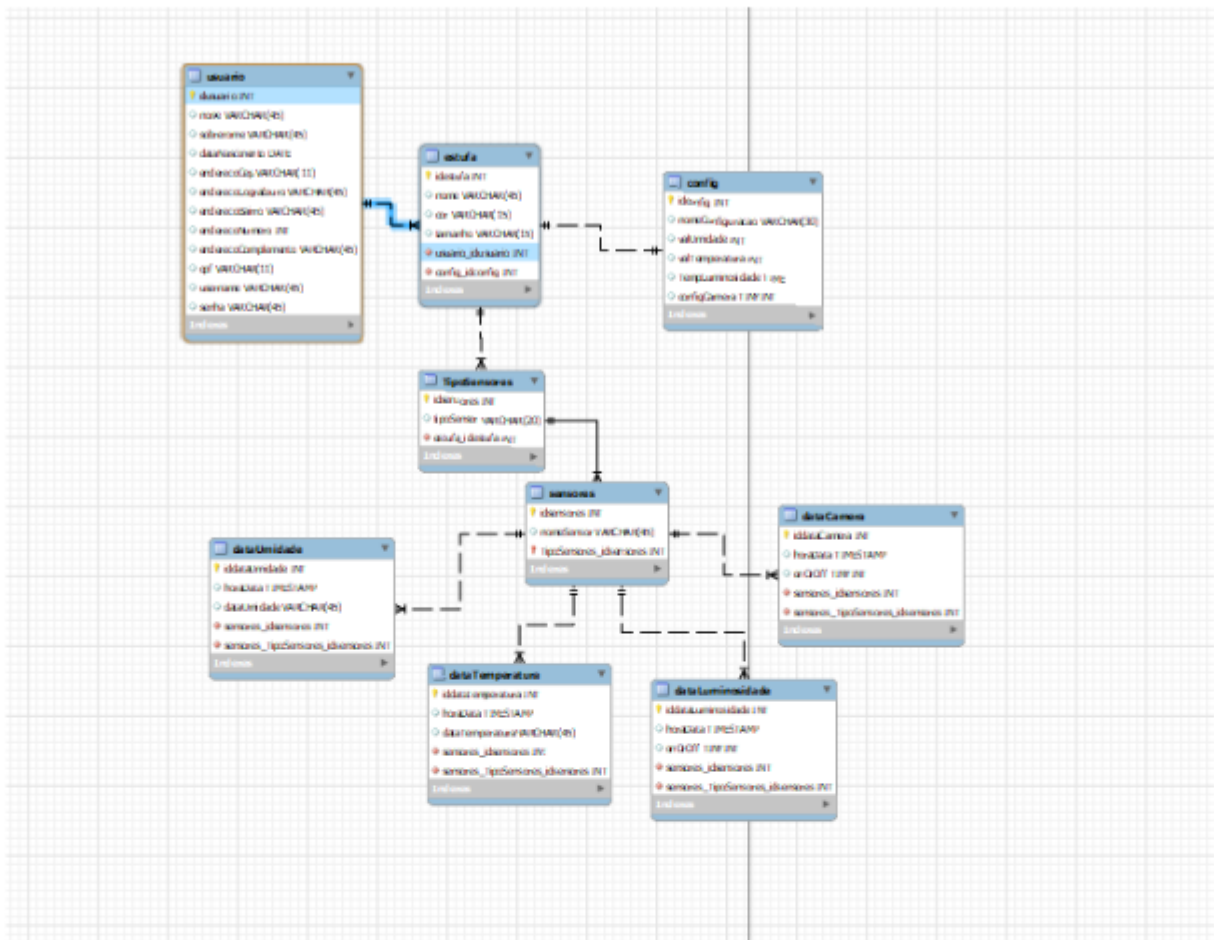
Fonte: Autores

3.3.2 MODELO LÓGICO E FÍSICO

Utilizando o SGBD MySQL, e com a ajuda do modelo conceitual já criado, foi construído primeiramente o modelo lógico, onde fora adotado um modelo mais minimalista e simples quando comparado com o modelo conceitual. Deste modo, em seu novo projeto, o mesmo continha uma quantidade menor de tabelas e ainda assim realizava as mesmas funções de forma mais organizada e melhor relacionada.

Diante disso, foi realizado o desenvolvimento do modelo físico. Onde contém todos os dados de cadastro dos usuários, que são obtidos por uma inserção em uma Rest API e as informações que são obtidas a partir dos sensores da estufa.

Figura 10 - Modelo Lógico



Modelo lógico desenvolvido

Fonte: autores

Figura 11 - Criação das tabelas

```

-----
-- Table 'mydb'. 'dataUnidade'
-----
CREATE TABLE IF NOT EXISTS 'mydb'. 'dataUnidade' (
  'iddataUnidade' INT NOT NULL AUTO_INCREMENT,
  'horaData' TIMESTAMP NULL,
  'dataUnidade' VARCHAR(45) NULL,
  'sensores_idsensores' INT NOT NULL,
  'sensores_TipoSensores_idsensores' INT NOT NULL,
  PRIMARY KEY ('iddataUnidade'),
  INDEX 'fk_dataUnidade_sensores1_idx' ('sensores_idsensores' ASC, 'sensores_TipoSensores_idsensores' ASC) VISIBLE,
  CONSTRAINT 'fk_dataUnidade_sensores1'
  FOREIGN KEY ('sensores_idsensores', 'sensores_TipoSensores_idsensores')
  REFERENCES 'mydb'. 'sensores' ('idsensores', 'TipoSensores_idsensores')
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Modelo Físico do banco de dados

Fonte: Autores

3.3.3 SQL

Figura 12 - Modelo Físico

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
nome	varchar(45)	NO		NULL	
sobrenome	varchar(45)	NO		NULL	
dataNascimento	date	NO		NULL	
enderecoCep	varchar(11)	NO		NULL	
enderecoLogradouro	varchar(45)	NO		NULL	
enderecoBairro	varchar(45)	NO		NULL	
enderecoNum	int	NO		NULL	
enderecoComplemento	varchar(45)	YES		NULL	
cpf	varchar(11)	NO		NULL	
criadoEm	varchar(45)	NO		NULL	
username	varchar(45)	NO		NULL	
password	varchar(50)	NO		NULL	

Tabela de usuários do Modelo Físico

Fonte: Autores

Figura 13- Tabela de Dados

Field	Type	Null	Key	Default	Extra
dateHourData	datetime	NO	PRI	NULL	
name	varchar(255)	YES		NULL	
date	date	YES		NULL	
dataUmidade	varchar(45)	YES		NULL	

Modelo das tabelas de armazenamento de dados dos sensores

Fonte: Autores

No projeto Alltomatic, foi utilizado Rest API, para a inserir e receber dados do banco, foi utilizado funções assíncronas para uma melhor comunicação e requisição entre usuário e o banco de dados.

Para inserir dados, foi utilizado o comando insert com variáveis que são recebidas pela API, direto do aplicativo desktop.

Figura 14 - Insert no Banco de dados

```
const createUser = async (user) => { // Função que cria os usuários
  const insert = `INSERT INTO users(username, password, nome, sobrenome, dataNascimento, enderecoCep, enderecoLogradouro, enderecoBairro, enderecoCidade, enderecoComplemento, cpf, criadData)`
  const values = `VALUES (${user.username}, ${user.password}, ${user.nome}, ${user.sobrenome}, ${user.dataNascimento}, ${user.enderecoCep}, ${user.enderecoLogradouro}, ${user.enderecoBairro}, ${user.enderecoCidade}, ${user.enderecoComplemento}, ${user.cpf}, ${user.criadaData})`
  const dateUTC = new Date(Date.now()).toLocaleString()
}
```

Utilização da função insert no banco de dados

Fonte: Autores

Para deletar dados, foi utilizado o comando delete na tabela users, a identificação do usuário a ser deletado é feita baseada no seu ID cadastrado no banco, que também é recebido na requisição.

Figura 15 - Delete Banco de dados

```
const deleteUser = async (id) => {
  const removedUser = await connection.execute(`DELETE FROM users WHERE id = ?`, [id])
  return removedUser;
}
```

Utilização da função delete no banco de dados

Fonte: Autores

Para alterar dados, foi utilizado o comando update, onde é recebido pelo aplicativo, via API, a tabela que será alterada, o valor do novo dado e o id do usuário que será alterado.

Figura 14 - Update Banco de dados

```
const updateUser = async (id, column, data) => {
  let query = `UPDATE users SET ${column} = ${data} WHERE id = ${id}`
  if(typeof data == 'string'){
    query = `UPDATE users SET ${column} = '${data}' WHERE id = ${id}`
  }
}
```

Utilização da função update no banco de dados

Fonte: Autores

Para mostrar os dados, foi utilizado o comando select, onde é selecionado todos os dados da tabela users, e enviado via método get para API, para ser apresentado ao usuário no aplicativo desktop.

Figura 16 -Select Banco de Dados

```
const getUser = async () => { // Função que obtém os usuários do banco de dados
  const [users] = await connection.execute('SELECT * FROM users')
  return users
}
```

Utilização da função select no banco de dados

Fonte: Autores

Figura 17 -Dados inseridos no banco

7	Matheus	Gonçalves	2000-08-04	13870440	Loren	Loren	333	Loren	cancelado	Mon, 23 Oct 2023 21:38:30 GMT	falsado
8	Matheus	Guimarães	2000-08-05	13870138	Rua Fioravante Galvani	Rosário	333	Loren	cancelado	Mon, 23 Oct 2023 21:39:45 GMT	falsado123456
9	Matheus	Gonçalves	2000-08-04	13870440	Rua Manoel João Batista	Jardim Sathélite	656	Loren	cancelado	Mon, 23 Oct 2023 21:38:54 GMT	sfsd/df
10	Matheus	Gonçalves	2000-08-04	13870440	Rua Manoel João Batista	Jardim Sathélite	45454545	adadadadadadad	cancelado	Mon, 23 Oct 2023 21:39:55 GMT	fghfgg
11	Matheus	Gonçalves	2000-08-04	13870238	Rua Fioravante Galvani	Rosária	2245465	adadadadadadad	cancelado	Mon, 23 Oct 2023 21:42:25 GMT	writerente
12	Matheus	Gonçalves	2000-08-04	13870440	Rua Manoel João Batista	Jardim Sathélite	66454654	edfEdadadadad	cancelado	Mon, 23 Oct 2023 21:39:47 GMT	matheusguimaraes123@gmail.com

Exemplo de dados inseridos no Banco

Fonte: Autores

Figura 18 - API

```
▼ 40:
  id: 41
  nome: "Matheus"
  sobrenome: "Guimarães"
  dataNascimento: "2000-08-04T03:00:00.000Z"
  enderecoCep: "13870330"
  enderecoLogradouro: "Rua Fioravante Galvani"
  enderecoBairro: "Rosário"
  enderecoNum: 45
  enderecoComplemento: "Apartamento 14"
  cpf: "45698712365"
  criadoEm: "Sat, 11 Nov 2023 01:25:18 GMT"
  username: "matheusguimaraes123@gmail.com"
  password: "123"
```

Exemplo de dados enviados pela API

Fonte: Autores

3.4 GESTÃO FINANCEIRA

Dentro da matéria de Gestão Financeira nos forneceu a importância de realizarmos e definirmos o preço, junto ao custo do produto referente ao nosso negócio, analisarmos o mercado em nossa volta, assim podendo melhorar nossa produtividade e rendimento de nossa empresa. Pudemos realizar o custo de um produto realizada a classificação dos custos do nosso projeto, sendo estes diretamente ligados ao produto ou indiretamente, e com isso precificar. Sendo assim é possível verificar se o preço está compatível com o mercado e se o produto seria viável.

3.4.1 CLASSIFICAÇÃO DOS CUSTOS

Segundo Martins (2001) descreve que custo é o esforço financeiro despendido para a aquisição de um bem ou serviço utilizado na produção de outro bem ou serviço.

Deste modo, foi realizado o cálculo de custos sendo eles diretos e indiretos. Custo direto é todo custo que tem como alvo principal a produção do produto, já o custo indireto é todo aquele que é utilizado para o produto mas que não estão diretamente ligados na atividade principal da empresa.

Como custos diretos na Alltomatic tem-se sensores e peças para a montagem da estufa e como custos indiretos sendo contas de água e luz, marketing, design, hospedagem, domínio e serviços de montagem.

3.4.2 CUSTOS DO PRODUTO

Na separação dos custos do produto, foi selecionado um produto referência para o comparativo e realização da análise de mercado.

Como resultado, obteve-se as tabelas de custos:

CUSTOS INDIRETOS	
LUZ	R\$ 100,00
ÁGUA	R\$ 40,00
INTERNET	R\$ 80,00
MARKETING	R\$ 100,00
DESIGN	R\$ 100,00
HOSPEDAGEM	R\$ 36,00
DOMÍNIO	R\$ 40,00
TOTAL=	R\$ 496,00

CUSTOS DIRETOS	
SENSOR DE UMIDADE	R\$ 9,40
ARDUINO UNO	R\$ 63,45
SENSOR DE UMIDADE E TEMPERATURA DHT11	R\$ 14,40
LCD 16x2	R\$ 14,18
MÓDULO BLUETOOTH	R\$ 22,41
CABOS JUMPER	R\$ 9,58
RELÉ	R\$ 26,90
VÁLVULA SOLENÓIDE	R\$ 28,90
LÂMPADA LED 12W	R\$ 11,07
PROTOBOARD	R\$ 25,76
3M DE TABUA(PINOS)	R\$ 18,46
FIO 2.5MM(1M)	R\$ 15,99
INTERRUPTOR	R\$ 5,35
CAIXA DE INTERRUPTOR	R\$ 7,87
MANTA REFLETORA	R\$ 15,57
TOTAL=	R\$ 289,29

GASTO TOTAL(INDIRETOS E DIRETOS)	R\$ 785,29
---	-------------------

Na primeira tabela pode-se observar uma estimativa de gastos indiretos com o produto. Na segunda mostrando os custos diretos, fazendo a soma dos gastos total chegando a um valor de R\$785,29.

Fazendo uma análise de mercado e comparando gastos da empresa Alltomatic com alguns produtos semelhantes, foi atingido um preço viável para o comércio. Pela pesquisa o

produto que mais se assemelha, custa na média de R\$4167,00 (Grow da Maria) e não contém tudo que a estufa indoor oferece.

3.4.3 PRECIFICAÇÃO

O custo de venda do produto é de R\$2.000,00. Podemos observar que com os dados de custos, conseguimos achar a Margem de contribuição que é o valor que sobra das vendas após subtrair os custos variáveis (custo total do produto - preço de venda) que ficou com o valor de R\$429,80, a Margem que é a porcentagem de lucro obtido a partir do produto (margem de contribuição - custo preço de venda) que ficou com a porcentagem de 21,49%, Markup e o percentual aplicado sobre o custo de um produto para determinar o preço de venda (margem de contribuição - custo total) 27,37 e índice de markup que é a relação entre preços de venda e os custos do produto ou serviços (markup + 1) 1,27.

		Custo Variável						
Custo Fixo	Design	Hospedagem	Impostos	Sensores	Custo Total	Preço	Margem de Contribuição	
R\$ 785,00	R\$ 300,00	R\$ 196,00	R\$ 144,60	R\$ 144,60	R\$ 1.570,20	R\$ 2.000,00	R\$429,80	
Margem	Mark Up	Índice de Markup						
21,49%	27,37%	1,27						

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 UNIFEOP.

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

Em gerenciamento e finanças foi desenvolvido a capacidade de gerenciar melhor seu dinheiro e assim transformando as mudanças em oportunidades para obter sucesso em todas as áreas da vida abordando alguns temas como:

- **Introdução aos conceitos econômicos e financeiros básicos:** O conceito de economia engloba a noção de como as sociedades utilizam os recursos para produção de bens com valor e a forma como é feita a distribuição desses bens entre os indivíduos. Já os financeiros é uma forma mais ampla, dizemos que ela trata do processo, instituições, mercados e instrumentos envolvidos na transferência de fundos entre pessoas, empresas e governos.
- **Independência Financeira, o valor da minha riqueza e o registro do dia a dia:** é o estágio em que uma pessoa conseguiu acumular um patrimônio suficiente para ser possível viver da renda passiva que esse patrimônio gera. Em outras palavras, chega um momento em que você não está mais dependente de ter que gerar renda com o seu esforço.
- **Dívidas e juros compostos, opções de empréstimo e alternativas ao endividado:** O conhecimento sobre os juros, empréstimos e alternativas dos indivíduos é essencial para lidar com as adversidades financeiras, ele nos ajuda a obter informações de como podemos lidar com esses problemas para não ficarmos com o nome sujo ou até mesmo saldo negativo na conta.
- **Estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos:** As metas financeiras são os objetivos e estratégias que você definirá para o seu dinheiro a fim chegar a uma conquista após um período planejado.

3.5.2 ESTUDANTES NA PRÁTICA

Em nosso Podcast, abordamos o tema de Gerenciando Finanças, com os tópicos de: Introdução aos conceitos econômicos e financeiros básicos, Independência Financeira, o valor da minha riqueza e o registro do dia a dia, Dívidas e juros compostos, opções de empréstimo e

alternativas ao endividado e Estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos.

4.CONCLUSÃO

No geral, o nosso projeto demonstrou a aplicação prática e integrada de diversos conhecimentos adquiridos em diferentes disciplinas. O aprendizado de programação orientada objeto foi amplamente utilizado na parte de desenvolvimento do aplicativo desktop. Com a matéria de Modelagem de Dados, foi desenvolvida a capacidade de criar, estruturar e desenvolver um Banco de Dados para nosso sistema, e com a Lógica de Programação, conseguimos integrar o aplicativo com o banco de dados, através de uma API e na matéria de Gestão Financeira, foi desenvolvida a capacidade de fazer o rateio, e a precificação, possibilitando assim um cálculo mais preciso dos custos diretos e indiretos relacionados ao nosso produto, e possibilitando achar um preço justo de comércio.

Isso destaca a importância de uma abordagem multidisciplinar e evidencia como a combinação de habilidades e conhecimentos de áreas diversas pode resultar em soluções inovadoras e completas.

Realizamos esse projeto em prol da UNIFEOB, na cidade de São João da Boa Vista visando melhorar o meio agrônomo e botânico, voltado para a automação do cultivo e estudos. Sendo a base do nosso projeto de extensão, cujo objetivo é a automatização de algo dentro do Campus UNIFEOB, para melhorarmos. Planejamos seguir com o projeto ao longo dos semestres decorrentes, buscando sempre atualizar e implementar cada vez mais melhorias para facilitar o uso e eficiência para a área na qual foi desenvolvido.

REFERÊNCIAS

FÉLIX, Rafael (org.). **Programação orientada a objetos**. São Paulo, SP: Pearson, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 09 nov. 2023.

Grow da maria. Link do produto referência para análise de mercado. Disponível em: https://growdamaria.com/produto/kit-de-cultivo-automatizado-90x90x180/?utm_source=Google+Shopping&utm_medium=cpc&utm_campaign=Produtos+de+google+shopping&gclid=CjwKCAjwsKqoBhBPEiwALrrqiJ34WShWBk_YWd-ThVmVpjOtE03yyHRN4B1Sc61--Bfg1lQ4r-DwRoCowgQAvD_BwE Acesso em: 27 set.2023

MANZANO, José e OLIVEIRA, Jaya. **Algoritmos - Lógica Para Desenvolvimento de Programação de Computadores**, 29ª edição, Bela Vista - SP - 2021

MARTINS, Eliseu. **Contabilidade de Custos**. 8. ed. São Paulo: Atlas, 2001.

RAMEZ, Elmasri, NAVATHE, Shamkant. **Banco de Dados**. 6. ed. Reino Unido: Londres, 2010.

ANEXOS

Link do nosso PodCast no Youtube: <https://www.youtube.com/watch?v=W7WqM-M1iyo>