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

<GATE5>

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

<EMPRESA>

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:

Betânia Amâncio Pereira , RA 23001181
Pedro Scarpellini dos Santos , RA 23000007
Lavínia Dal Bello e Souza, RA 23000373
Lucas B. S. Rodriguez, RA 23001178
Lucas Eduardo Cruz Alves, RA 23000617
Maria Fernanda Tobias , RA 23001209

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2023

SUMÁRIO

ISSN 1983-6767

1. INTRODUÇÃO

A crescente preocupação com o meio ambiente e a necessidade de promover práticas sustentáveis têm impulsionado o desenvolvimento de soluções inovadoras para o gerenciamento eficiente de resíduos. Nesse contexto, a GATE5, um grupo dedicado à tecnologia e sustentabilidade, apresenta um projeto de uma lixeira inteligente com um aplicativo integrado. Esta lixeira, que combina sensores avançados e conectividade, visa otimizar o processo de coleta de resíduos e promover a conscientização sobre a gestão adequada de resíduos.

A lixeira inteligente desenvolvida pelo grupo é uma solução tecnologicamente avançada que utiliza sensores integrados para monitorar em tempo real o nível de ocupação da lixeira. Estes sensores, estrategicamente posicionados no interior da lixeira, captam informações precisas sobre a quantidade de resíduos presentes, permitindo que os usuários saibam exatamente quando é necessário esvaziá-la. Essa abordagem inteligente não apenas melhora a eficiência da coleta de resíduos, mas também contribui para um ambiente mais limpo e sustentável.

"O estudo realizado por Bhatia e Singh (2020) destaca a importância da utilização de sensores integrados em lixeiras inteligentes para monitorar o nível de ocupação e otimizar o processo de coleta de resíduos. Esses sensores, similares aos aplicados no projeto da lixeira inteligente da GATE5, permitem a coleta de informações precisas sobre a quantidade de resíduos presentes, facilitando a tomada de decisões em relação ao esvaziamento das lixeiras. Além disso, a integração desses sensores com um aplicativo móvel, como proposto pelo grupo GATE5, oferece aos usuários a conveniência de receber notificações em tempo real sobre o status da lixeira, promovendo um sistema de gestão de resíduos mais eficiente e sustentável. Além do aplicativo móvel, também será desenvolvida uma aplicação desktop dedicada para captar e apresentar os dados da lixeira, proporcionando aos administradores e usuários uma experiência completa e acessível em diversas plataformas . Isso permitirá um controle abrangente e conveniente do sistema de gestão de resíduos, tanto em dispositivos móveis quanto em computadores desktop.

ISSN 1983-6767

2. DESCRIÇÃO DA EMPRESA

A Empresa Gate5 é formada por um time multidisciplinar de estudantes ingressados no Centro Universitário da Fundação de Ensino Octávio Barros, mais conhecido como UNIFEOB, com o CNPJ 59.764.555/0001-52. Sua sede está localizada em São João da Boa Vista, no estado de São Paulo, na Avenida Dr. Octávio da Silva Bastos, 2439, Campus II - Mantiqueira, Bairro Nova São João. Nosso objetivo é criar soluções práticas e inteligentes para melhorar o dia a dia das pessoas. Através de uma abordagem colaborativa e criativa, estamos constantemente buscando novas maneiras de resolver problemas e tornar o mundo um lugar melhor.

A Lixeira Inteligente é um exemplo de um produto vindo de nosso compromisso com a inovação. Nós acreditamos que pequenas mudanças podem ter um grande impacto e, ao desenvolver essa solução, estamos trabalhando para tornar a gestão de resíduos mais eficiente e sustentável. Estamos orgulhosos de apresentar nosso produto e compartilhar com você os detalhes do trabalho árduo e da dedicação que temos investido nele.

ISSN 1983-6767

3. PROJETO INTEGRADO

O projeto consiste na automação de lixeiras, onde, através de um sensor, acontece a comunicação entre a lixeira e o aplicativo para que os usuários consigam identificar em qual local determinada lixeira está e a porcentagem de sua lotação.

A crescente preocupação com o meio ambiente e a necessidade de promover práticas sustentáveis têm impulsionado o desenvolvimento de soluções inovadoras para o gerenciamento eficiente de resíduos. Nesse contexto, a GATE5, um grupo dedicado à tecnologia e sustentabilidade, apresenta um projeto de uma lixeira inteligente com aplicativo integrado para auxiliar seus usuários a efetuar a troca das lixeiras observando através do sistema, as lixeiras que precisam ser trocadas,qual seu tipo de resíduo e onde está localizada. Esta lixeira, aliando sensores e conectividade, visa otimizar o processo de coleta e promover a conscientização sobre a gestão de resíduos.

3.1 PROGRAMAÇÃO ORIENTADA A OBJETO

A programação orientada a objetos é um importante conjunto de técnicas que podem ser utilizadas para realizar o desenvolvimento de programas mais eficientes, melhorando a confiabilidade dos programas de computação. Na programação orientada a objetos, os objetos são os elementos principais de construção. Entretanto, a simples compreensão do que é um objeto, ou o uso de objetos em um programa, não significa que estamos programando de uma maneira orientada a objetos. O que conta é o sistema no qual os objetos se interconectam e se comunicam entre si

A programação orientada a objetos é caracterizada por três elementos fundamentais: (1) a utilização de objetos como blocos de construção lógicos em vez de algoritmos, resultando em uma hierarquia de objetos; (2) a criação de objetos como instâncias de classes; e (3) a interconexão entre classes por meio de relações de herança. É importante destacar que, se algum desses elementos estiver ausente, o programa não pode ser considerado orientado a objetos. Quando a herança não está presente, a abordagem se diferencia da programação orientada a objetos e é muitas vezes denominada programação com tipos abstratos de dados ou programação baseada em objetos.

Um objeto, que pode ser visto como um tipo abstrato de dados ou um tipo personalizado, consiste em um conjunto de dados e as funções associadas que operam sobre esses dados. No entanto, a verdadeira potência dos objetos está na capacidade de definir novos objetos a partir de objetos existentes. Esse processo, chamado herança, é o mecanismo fundamental que permite a construção de programas que podem ser facilmente adaptados e modificados para atender a diferentes necessidades e aplicações.

3.1.1 CLASSES E OBJETOS

Segundo Lima (2014, p. 49), “uma classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica”.

Uma Classe é a representação de um conjunto de objetos e inclui métodos e dados que resumem as características comuns desse conjunto. Podemos criar muitos objetos da mesma classe, considerando a classe como a definição de um tipo de objeto. Classes são semelhantes aos tipos de dados e servem como modelos ou moldes que descrevem como construir objetos de um determinado tipo.

Cada vez que instanciamos um objeto a partir de uma classe, estamos criando uma instância daquela classe, tornando o objeto uma variável de tipo objeto. Em termos gerais, os termos "instância" e "objeto" são frequentemente usados de forma intercambiável.

Quando solicitamos a criação de um objeto, a classe responde à mensagem de criação, o que resulta na criação de uma instância desse objeto. Em suma, classes definem estruturas e comportamentos compartilhados por objetos, e a criação de objetos a partir de classes é o processo de instanciar essas estruturas para uso específico.

Um conceito de objeto, segundo Luis Joyanes de Aguiar,autor do livro “Fundamentos de Programação”, é como os tipos abstratos de dados ou tipos definidos pelo usuário, representa uma coleção de dados juntamente com as funções associadas para operar sobre esses dados. No entanto, a verdadeira força dos objetos reside na capacidade de criar novos objetos com base nos existentes. Esse processo, conhecido como herança, é o mecanismo central que facilita a construção de programas que podem ser facilmente adaptados e ajustados para diferentes aplicações.

Os pilares fundamentais da programação orientada a objetos incluem objetos, classes, herança, mensagens e polimorfismo. Em programas orientados a objetos, os objetos

desempenham um papel essencial. Eles se comunicam entre si, trocando mensagens e colaborando para realizar tarefas.

Em resumo, na programação orientada a objetos, os objetos são unidades essenciais que reúnem dados e funcionalidades, e a herança é o mecanismo que permite a construção de hierarquias de objetos, tornando os programas mais flexíveis e adaptáveis.

3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO E HERANÇA.

1. **Atributo:**

Um atributo é uma característica ou propriedade de um objeto que descreve seu estado. Aqui está uma definição amplamente aceita:

Segundo Grady Booch, um autor renomado em POO, um atributo é uma variável que está associada a uma classe ou objeto e contém informações que descrevem o estado desse objeto. Os atributos são usados para representar dados específicos que são relevantes para o objeto e influenciam o comportamento do mesmo. Eles são uma parte fundamental na definição de uma classe, que inclui tanto os atributos quanto os métodos que operam sobre esses atributos para realizar a funcionalidade desejada. Os atributos são muitas vezes chamados de campos ou propriedades e podem ter diferentes tipos de dados, como inteiros, strings, booleanos, entre outros, dependendo das necessidades do objeto.

2. **Métodos:**

Métodos na programação orientada a objetos (POO) são funções associadas a objetos ou classes, que definem comportamentos específicos ou ações que podem ser executadas por esses objetos ou classes.

Conforme a perspectiva de Grady Booch, um método é uma função que descreve as ações que os objetos pertencentes a uma classe podem realizar. Os métodos são parte integrante das classes e definem como os objetos dessa classe se comportarão. Eles encapsulam o comportamento específico que um objeto pode executar, ajudando a garantir que o código seja modular e organizado.

Em outras palavras, os métodos são responsáveis por operar sobre os atributos de um objeto e, portanto, permitem que os objetos executem ações ou manipulem dados de acordo com as necessidades do programa. Os métodos podem ser invocados por objetos da classe a que pertencem, permitindo que os objetos colaborem entre si para realizar tarefas complexas.

Assim, os métodos são a maneira pela qual a funcionalidade de um objeto é definida e implementada em POO, desempenhando um papel fundamental na modelagem e execução de

sistemas baseados em objetos. Eles são ações específicas que os objetos podem realizar, tornando a POO uma abordagem poderosa para a criação de software modular e reutilizável.

3. **Encapsulamento:**

O encapsulamento é um dos princípios fundamentais da programação orientada a objetos e está intimamente relacionado à ideia de ocultar os detalhes internos de como os métodos de uma classe funcionam dos objetos que a utilizam. Isso significa que os objetos não precisam se preocupar com a implementação interna de um método, apenas precisam acioná-lo e receber o resultado desejado. O encapsulamento também se estende aos atributos de uma classe, permitindo que um objeto mantenha seus dados protegidos de acesso direto por outros objetos e permitindo que esses dados sejam manipulados e acessados apenas por meio dos métodos da própria classe, garantindo assim maior controle e segurança (TUCKER; NOONAN, 2009).

4. **Herança:**

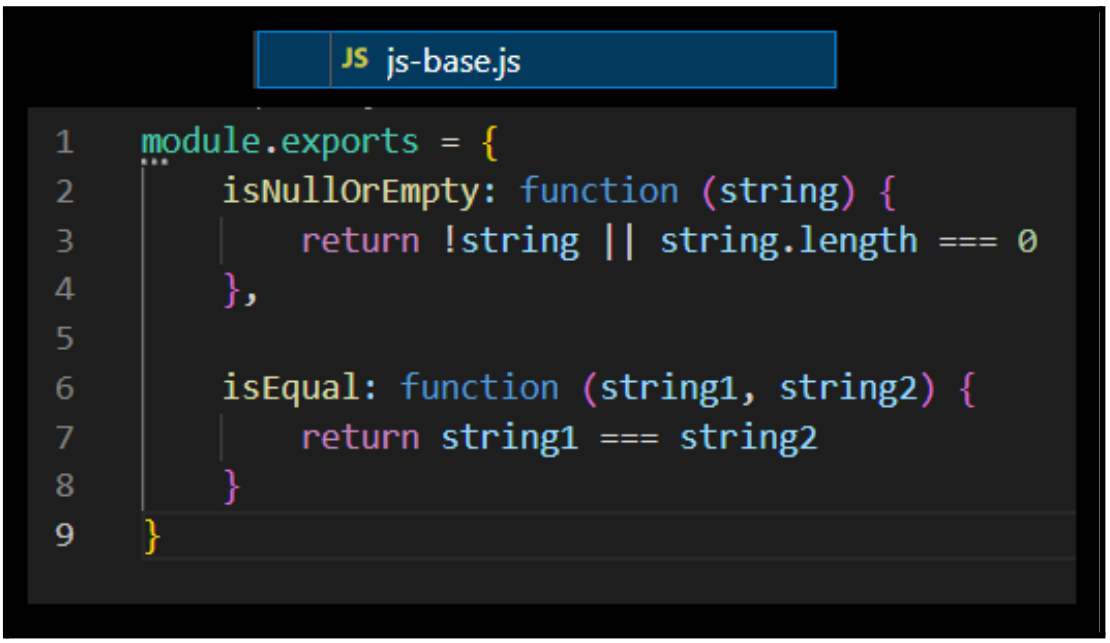
O conceito de herança na programação orientada a objetos é utilizado para se permitir a reutilização de um código. A herança possibilita que as classes compartilhem seus atributos e métodos entre si. Segundo Tucker e Noonan (2009, p. 335),

> [...] o paradigma orientado a objetos suporta a reutilização de código por intermédio da herança. As classes existem em uma linguagem orientada a objetos em uma hierarquia de classes. Uma classe pode ser declarada como uma subclasse de outra classe, que é chamada de classe mãe ou superclasse.

Dentro dessa relação de hierarquia de classes é possível que a subclasse herde atributos e comportamentos da superclasse, simplesmente pelo fato de ser sua subordinada.

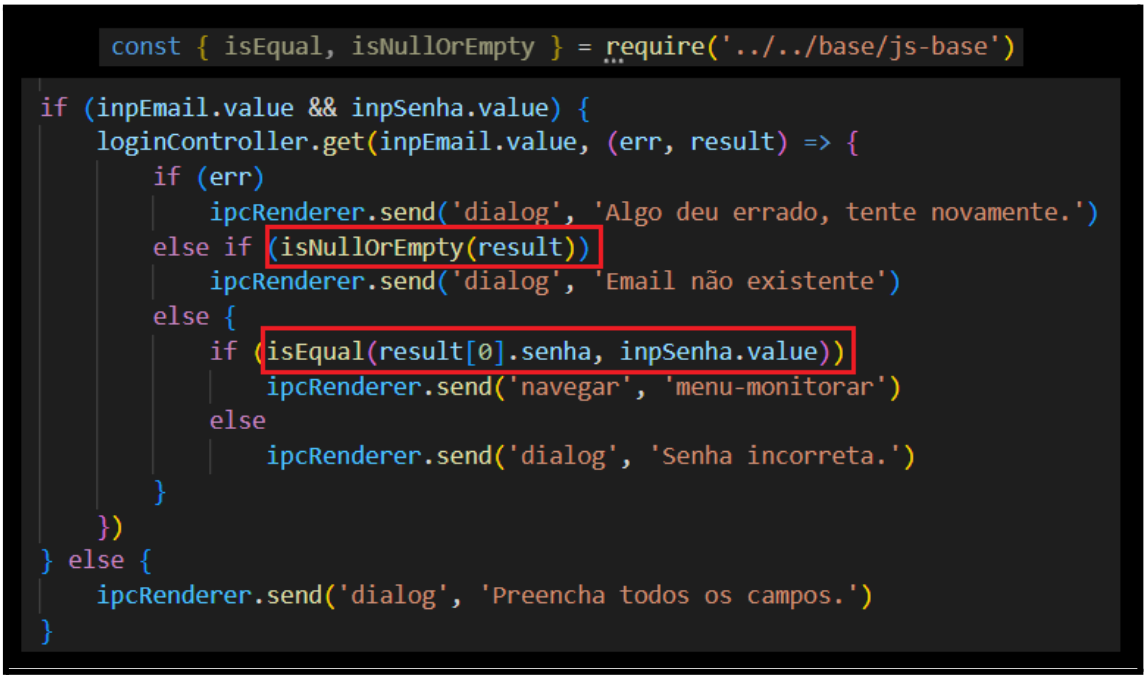
No nosso projeto criamos duas classes bases, uma para ser a base do .CSS e outra para ser a base do .JS, com funções comuns para ser usadas tanto na tela como na lógica.

Figura 1 - Classe base com métodos comuns para lógica do programa



Fonte: autores (2023)

Figura 2 - Instanciação dos métodos base nas variáveis isEqual e IsNullOrEmpty para serem usadas no arquivo de controle da lógica da página login.

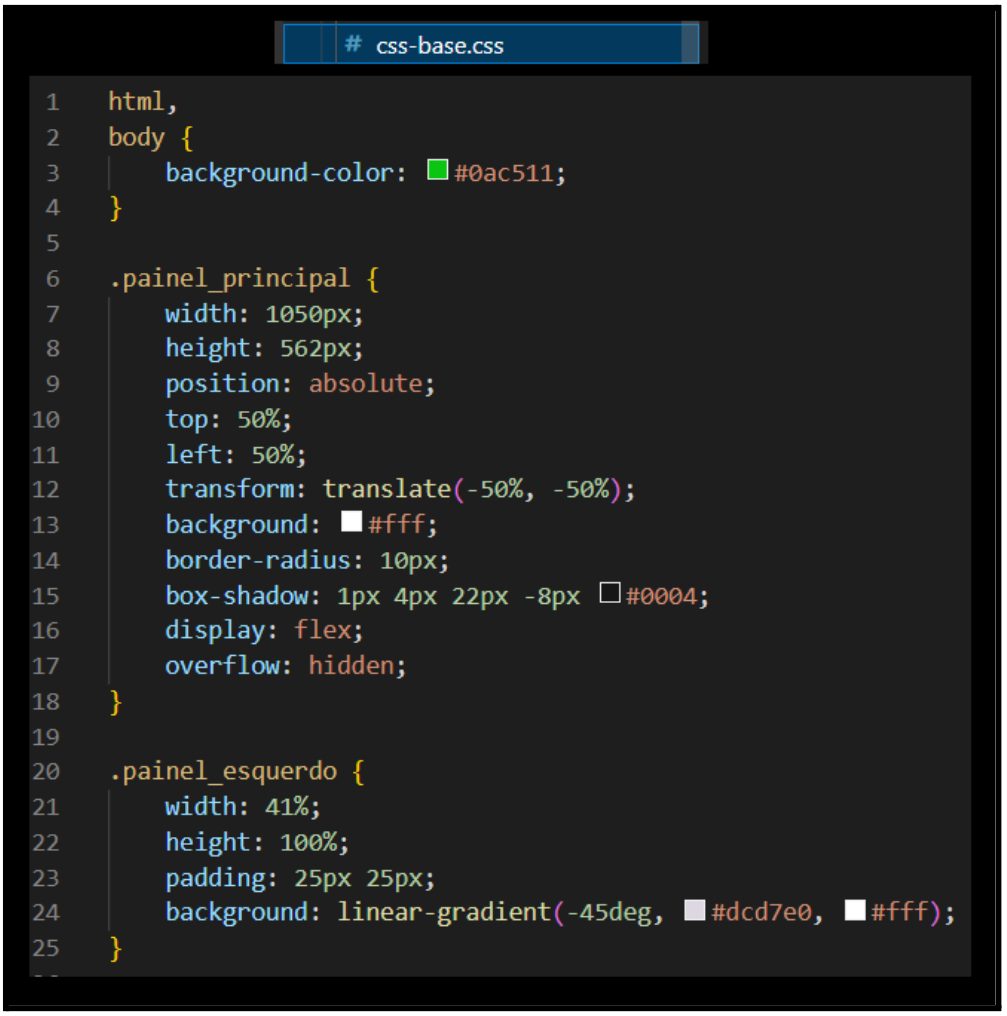


Fonte: autores (2023)

Usando a herança da lógica evitamos o ‘boilerplate’, que é a repetição do mesmo código e lógica múltiplas vezes. O mesmo foi feito para os códigos .CSS com o arquivo

css-base.css, evitando a repetição da criação dos mesmos componentes de tela múltiplas vezes.

Figura 3 - Criação dos componentes base no arquivo css-base.css.



Fonte: autores (2023)

Figura 4 - Exemplo de uso do arquivo base do CSS na tela de login.

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../base/css-base.css" />
    <link rel="stylesheet" href="../../base/bootstrap.css">
</head>

<body>
    <div class="painel_principal">
        <div class="painel_esquerdo">
            <div class="campos">
                <img src="../../../assets/images/unifeob.png" class="feob-logo">
                <h3>Faça seu login</h3>
                <input type="text" id="inp-email" placeholder="EMAIL" class="field-input">
                <input type="text" id="inp-senha" placeholder="SENHA" class="field-input">

                <div class="form-check">
                    <input class="form-check-input" type="checkbox" id="ckb-administrador">
                    <label class="form-check-label" for="ckb-administrador">Administrador</label>
                </div>

                <button class="app-inv-button" id="btn-recuperar-senha">Esqueceu a senha?</button>
                <button class="app-button" id="btn-entrar">ENTRAR</button>
            </div>
```

Fonte: autores (2023)

3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

Os métodos estáticos, também conhecidos como métodos de classe, são recursos fundamentais na programação que pertencem à classe em vez de a instâncias individuais dessa classe. Eles desempenham um papel crucial na realização de operações que não estão intrinsecamente ligadas ao estado específico de um objeto, mas, em vez disso, se relacionam com a classe como uma unidade coesa. A seguir, apresentaremos algumas características e exemplos de métodos estáticos, integrados em nosso projeto:

Os métodos estáticos são empregados para executar tarefas que não dependem das propriedades específicas de um objeto, mas, em vez disso, se relacionam com a classe como um todo. Eles desempenham um papel importante em nosso projeto e são uma maneira eficaz de encapsular funcionalidades que são globalmente relevantes para a classe, contribuindo para a organização e eficiência do código.

Figura 5 - Instanciação da classe LoginModel com os métodos públicos

Fonte: autores (2023)

Na imagem acima podemos ver a instanciação da classe LoginModel e dentro dela os métodos públicos create e get. Esses métodos são públicos pois podem ser acionados externamente por quem for detentor do objeto da classe LoginModel como mostrado na imagem seguinte:

Figura 6 - Uso do método create da classe LoginModel

```
create: function (email, senha, callback) {
    new LoginModel(null, email, senha).create((loginId) => { callback(loginId) })
},
```

Fonte: autores (2023)

Na imagem abaixo podemos ver o uso de um método privado, ou seja, somente a classe detentora do método tem acesso a ele.

Figura 7 - Ao chamarem getAndValidate, a classe LoginModel chama o método privado validateSenha. O '#' indica que é método privado em Javascript.

```
const db = require('../database/database');

class LoginModel {

    constructor(id, email, senha) {
        this.id = id
        this.email = email
        this.senha = senha
    }

    getAndValidate(senha) {
        const sql = 'SELECT * FROM login WHERE email = ?;'
        const values = [this.email]

        db.conexao.query(sql, values, (err, result) => { callback(self.validateSenha(senha)) })
    }

    #validateSenha(inputSenha) {
        return senha === inputSenha
    }
}

module.exports = LoginModel;
```

Fonte: autores (2023)

ISSN 1983-6767

3.2 LÓGICA DE PROGRAMAÇÃO

Paulo Silveira e Adriano Almeida, autores do livro "Desenvolvimento de Software", explicam a lógica de programação como "a técnica de criar algoritmos para solucionar problemas lógicos". A lógica de programação é a habilidade de ordenar e descrever passos lógicos para a solução de um problema."

A lógica da programação carrega uma abstração que desempenha um papel fundamental no mundo da tecnologia e do desenvolvimento de sistemas. Ela serve como a base sobre a qual toda a construção de software é feita, permitindo que os programadores criem soluções eficazes e eficientes para uma ampla variedade de problemas.

A abstração da lógica em programação é um conceito fundamental que se refere à capacidade de criar modelos simplificados e representações abstratas de sistemas, processos ou problemas complexos. Essa abstração é uma técnica que ajuda os programadores a lidar com a complexidade e a desenvolver soluções mais eficientes. Alguns dos aspectos importantes da abstração lógica na programação são: Abstração de Dados; Abstração de Funções; Abstração de Controle; Abstração de Objetos; Abstração de Algoritmos.

3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

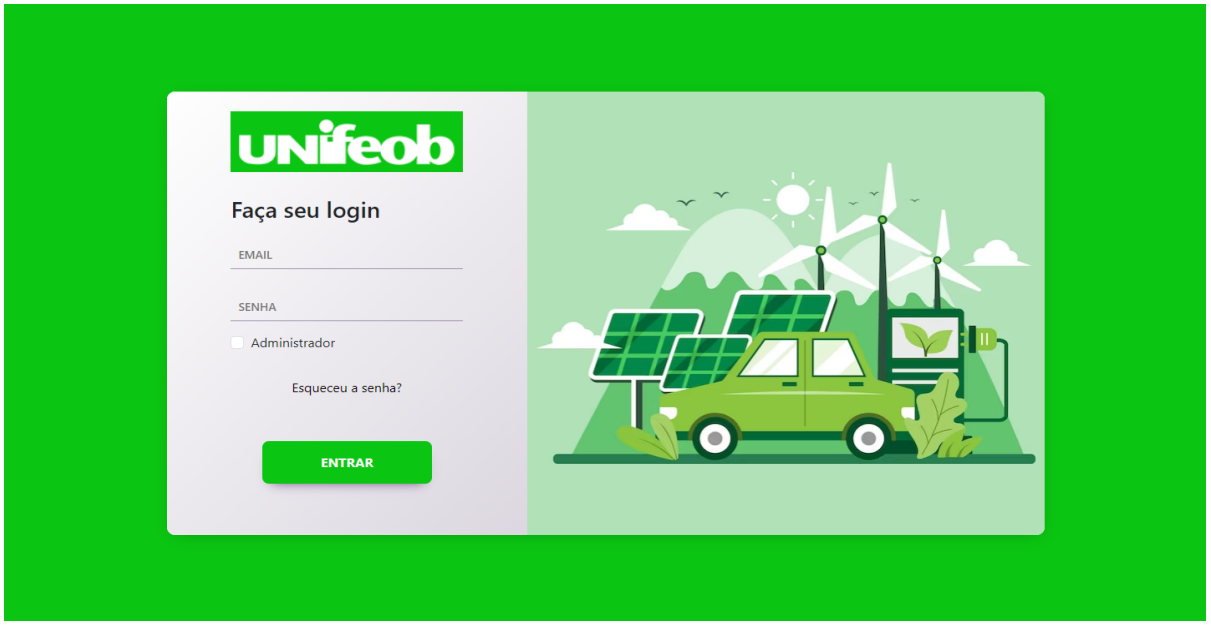
Buscamos os conceitos fundamentais da lógica de programação sendo uma parte do processo de desenvolver algoritmos eficientes para a resolução de problemas incluindo algoritmos , variáveis, tipos de dados , funções, estruturas condicionais e operadores . Podemos explorá-los através das estruturas de controle condicionais como (if , else , switch, for e while) , algoritmos, variáveis, tipos de dados (como inteiros , números de ponto flutuante , strings e valores booleanos).

3.2.2 DESENVOLVIMENTO DE APLICAÇÕES DESKTOP

Neste tópico, exploramos o desenvolvimento de aplicações desktop utilizando o framework Electron.JS, onde fomos capazes de elaborar uma interface para o nosso projeto, composta por 9 telas. Todas as telas foram desenvolvidas utilizando HTML, CSS, JavaScript onde estabelecem uma conexão direta com o banco de dados MySQL Workbench,

possibilitando enviar dados reais da lixeira para o usuário. Essa integração foi alcançada por meio de comandos básicos como configuração de conexão.

Figura 8 - Tela de Login:



Fonte: autores (2023)

Figura 9 - Tela de Cadastro:

Fonte: autores (2023)

O restante das telas foram colocadas em anexo no final do documento.

3.3 MODELAGEM DE DADOS

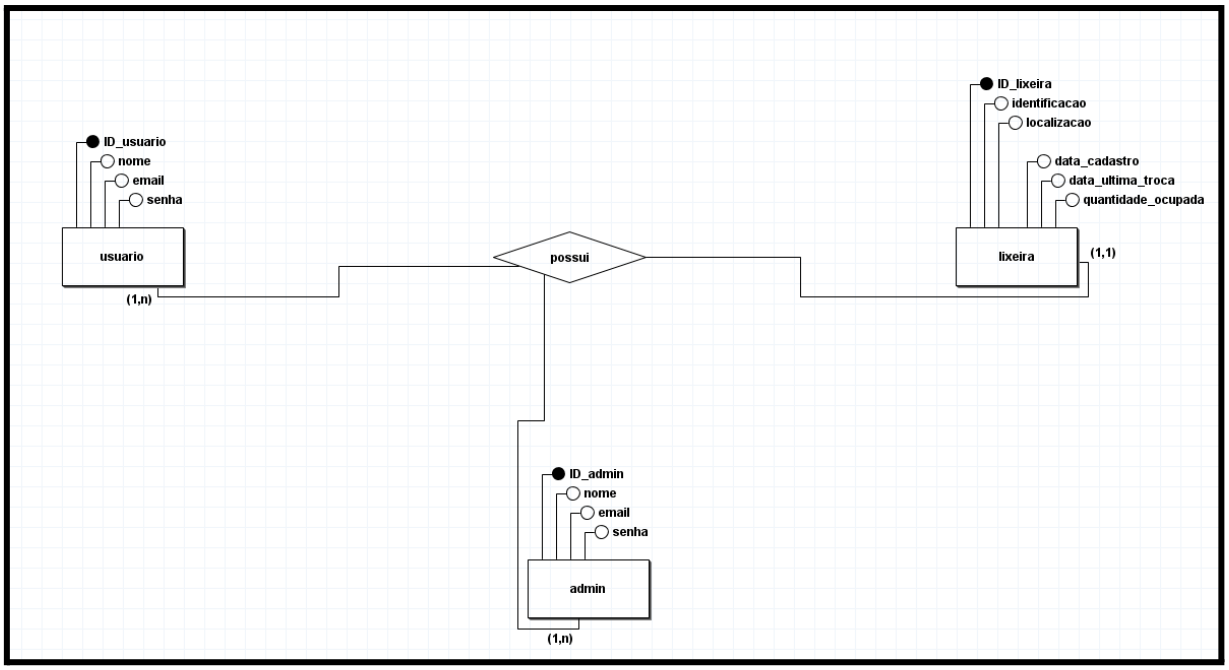
Para Korth, Silberschatz a Sudarshan (2012), um banco de dados “[...] é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”. Eles desempenham um papel fundamental em praticamente todos os aplicativos de software modernos, desde sistemas de gerenciamento de empresas até aplicativos de mídia social e serviços de comércio eletrônico. Para entender melhor o conceito de banco de dados, é útil dividir sua estrutura em alguns aspectos principais: o modelo conceitual, modelo lógico e físico.

3.3.1 MODELO CONCEITUAL

O modelo conceitual de banco de dados é uma representação abstrata e de alto nível de um sistema de banco de dados que descreve as informações que serão armazenadas e a maneira como essas informações estão relacionadas. É uma etapa inicial e fundamental no projeto de banco de dados, pois ajuda a definir a estrutura geral dos dados sem se preocupar com os detalhes técnicos de implementação.

Para Heuser (2009, p. 25), “[...] a técnica de modelagem conceitual mais difundida é a abordagem entidade-relacionamento. Nessa técnica, um modelo conceitual é usualmente representado através de um diagrama”. O MER utiliza elementos gráficos para descrever o modelo de dados de uma aplicação com alto nível de abstração (CALIARI, 2007), identificando entidades, atributos e relacionamentos. Peter Chen, em 1976, idealizou uma notação para realizar a modelagem de dados para ambientes relacionais. Na figura 1 podemos observar o modelo conceitual do projeto, dividido em três tabelas, sendo elas: Usuario (a tabela “usuario” é responsável por armazenar os funcionários cadastrados para acesso no software e nas lixeiras), admin (a tabela "admin" contém informações sobre os administradores responsáveis por gerenciar o software das lixeiras) e lixeira (a tabela “lixeira” armazena as informações das próprias lixeiras e seu status e, administradores e usuários acessam para seu devido controle).

Figura 10 - modelo conceitual

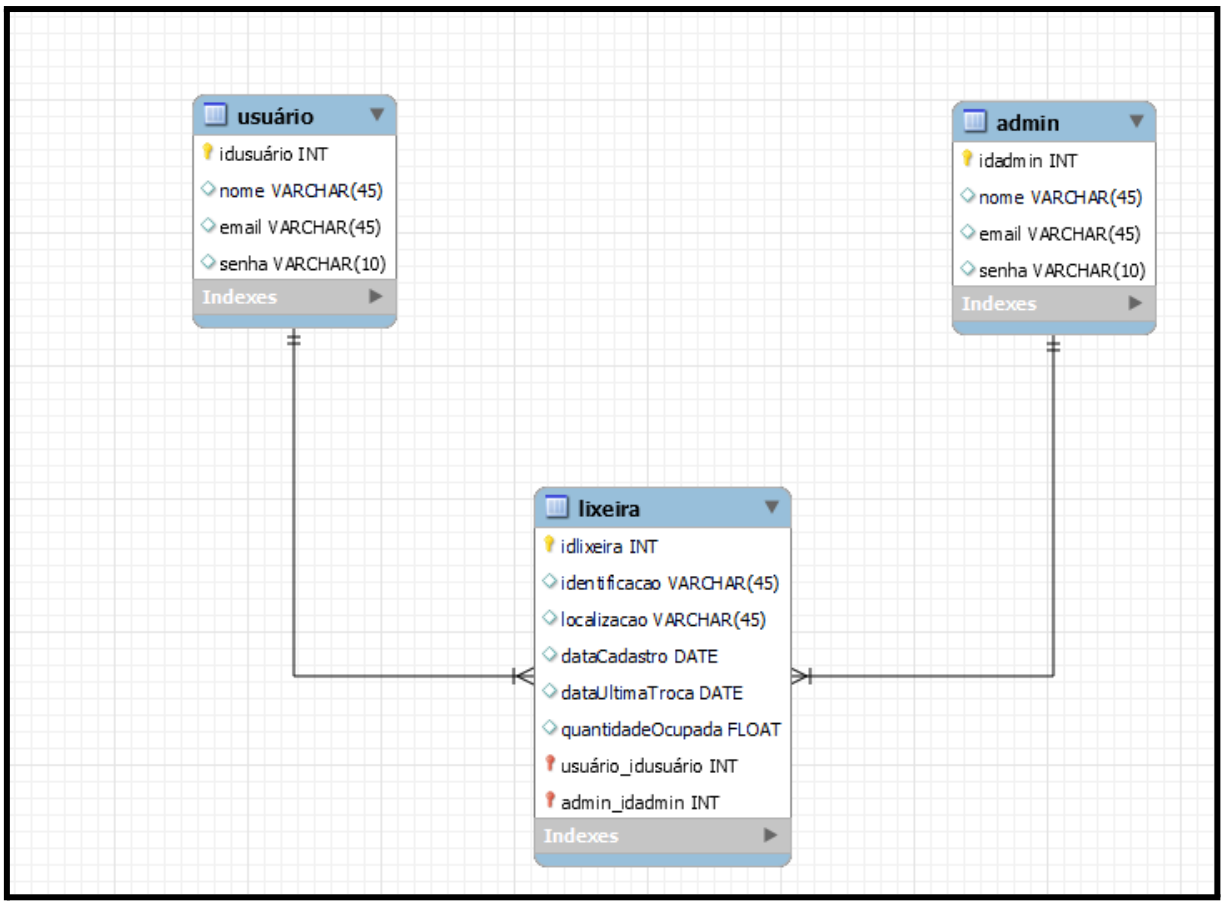


Fonte: autores (2023)

3.3.2 MODELO LÓGICO E FÍSICO

Um modelo lógico de banco de dados é uma representação abstrata da estrutura e organização dos dados em um sistema de gerenciamento de banco de dados (DBMS), independente de detalhes de implementação física. Ele descreve como os dados são organizados em tabelas, relacionamentos entre tabelas, regras de integridade de dados e operações de consulta. O modelo lógico fornece uma visão conceitual que facilita o projeto de dados antes da implementação física, sendo independente de plataformas de hardware ou sistemas de gerenciamento de banco de dados específicos. É uma etapa crucial no desenvolvimento de sistemas de banco de dados. Na imagem abaixo é possível observar o modelo separado em três tabelas com seus respectivos atributos:

Figura 11 - modelo lógico



Fonte: autores (2023)

O modelo físico de banco de dados, por outro lado, trata da implementação real dos dados em um sistema de gerenciamento de banco de dados (DBMS). Ele envolve decisões sobre como os dados serão armazenados em dispositivos de armazenamento, como discos rígidos, memória RAM e índices, bem como otimizações de desempenho para consultas. Jim Gray, um cientista da computação renomado, fez importantes contribuições ao campo de sistemas de gerenciamento de banco de dados, incluindo a otimização de consultas e o projeto de sistemas de armazenamento.

No modelo físico, também é importante considerar o acesso concorrente aos dados, a segurança e a escalabilidade do sistema de banco de dados. Autores como Michael Stonebraker, criador do sistema de gerenciamento de banco de dados relacional Ingres, e outros pesquisadores em sistemas de banco de dados têm contribuído para o desenvolvimento de tecnologias que melhoram o desempenho e a confiabilidade dos sistemas de banco de dados em nível físico.

Em resumo, o modelo lógico descreve a estrutura e a organização dos dados de forma independente da implementação, enquanto o modelo físico lida com a implementação real,

incluindo o armazenamento e o desempenho do banco de dados. Ambos são cruciais para o projeto e a manutenção de sistemas de banco de dados eficientes.

3.3.3 SQL

A linguagem SQL é utilizada para a realização de consultas em dados armazenados em um banco de dados. Ela é muito importante, pois permite a manipulação, de forma simples e poderosa, dos dados presentes em um Sistema de Gerenciamento de Banco de Dados.

Segundo Elmasri e Navathe (2004), a linguagem SEQUEL nasceu unicamente para ser a linguagem de consulta do sistema System R da IBM. Sua evolução originou a linguagem SQL tanto difundida atualmente.

Para realizar a inserção, remoção, edição ou qualquer outra atividade com os dados que irão para o banco de dados ou que já estão nele, é necessária uma interação com o mesmo. Dessa forma, a linguagem SQL (Structure Query Language ou Linguagem de Consulta Estruturada) é a intermediária dessa interação, criando a ponte necessária entre o que precisa ser feito e o banco de dados efetivamente. Assim, para que algo aconteça, é necessário que a linguagem SQL esteja com a sintaxe e os valores desejados corretos. Abaixo estão exemplos de scripts usados no banco de dados do projeto:

Figura 12: Comando de UPDATE feito na tabela ‘lixeira’ para mudar a data da última troca feita na lixeira de id 5.

```
3 •  select * from usuarios;
4 •  select * from admin;
5 •  select * from lixeira;
6
7
8 •  UPDATE lixeira SET dataUltimaTroca = '2023-10-27' WHERE idlixeira = 5;
9
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

idlixeira	identificacao	localizacao	dataCadastro	dataUltimaTroca	quantidadeOcupada	usuário_idusuário	admin_idadmin
1	Lixo seco	Prédio A	2023-09-29	2023-09-29	50	1	1
2	Lixo molhado	Prédio B	2023-09-29	2023-09-29	75	2	1
3	Vidro	Prédio C	2023-09-29	2023-09-29	30	3	2
4	Plástico	Prédio D	2023-09-29	2023-09-29	40	5	2
5	Plástico	Prédio F	2023-09-29	2023-10-27	40	3	5

Fonte: autores (2023)

Figura 13: Comando DELETE feito nas tabelas usuarios,lixeira,admin apagando um usuário, uma lixeira e um administrador por seu id.

```
88 •  DELETE FROM usuarios WHERE idusuarios = 1;
89 •  DELETE FROM lixeira WHERE idlixeira = 2;
90 •  DELETE FROM admin WHERE idadmin = 3;
91
```

Fonte: autores (2023)

Figura 14: Comando SELECT com INNER JOIN para retornar os usuários que estão cadastrados na lixeira do prédio C.

```
SELECT u.nome
FROM usuarios u
INNER JOIN lixeira l ON u.idusuarios = l.usuário_idusuário
WHERE l.localizacao = 'Prédio C';
```

Fonte: autores (2023)

3.4 GESTÃO FINANCEIRA

A gestão financeira desempenha um papel crucial na sustentabilidade e no sucesso de uma empresa. Uma parte fundamental dessa gestão é a análise e o acompanhamento dos custos.

A Importância da Gestão Financeira:

1. Tomada de Decisões Estratégicas: A gestão financeira fornece informações que permitem às empresas tomar decisões estratégicas bem fundamentadas. Isso inclui decidir sobre investimentos, expansões, alocação de recursos e estratégias de preços. De acordo com Gitman e Zutter (2018) em "Princípios de Administração Financeira", a análise financeira ajuda a determinar a viabilidade de projetos e a identificar oportunidades de crescimento.

2. Maximização de Lucros: Uma gestão financeira eficaz ajuda a empresa a maximizar seus lucros, otimizando a relação entre receitas e custos. Segundo Horngren et al. (2012) em "Contabilidade de Custos", o controle de custos desempenha um papel fundamental nesse processo.

3. Sustentabilidade Financeira: Manter uma empresa saudável financeiramente requer monitoramento constante dos gastos. Segundo Ross, Westerfield e Jordan (2019) em "Corporate Finance", a gestão financeira ajuda a garantir que os recursos estejam sendo utilizados de forma eficiente e que a empresa possa cumprir suas obrigações financeiras a longo prazo.

A Importância do Controle de Custos:

1. Identificação de Ineficiências: Controlar os custos permite que as empresas identifiquem ineficiências e desperdícios em suas operações. Isso é essencial para a melhoria contínua e a redução de custos operacionais.

2. Precificação Adequada: O controle de custos é fundamental para determinar preços competitivos e manter margens de lucro saudáveis. Como mencionado por Horngren et al. (2012), entender os custos de produção é essencial para estabelecer preços de venda apropriados.

3. Orçamento e Planejamento: O controle de custos desempenha um papel importante no planejamento financeiro e orçamentário. Permite que as empresas aloquem recursos de forma eficaz e evite surpresas desagradáveis no que diz respeito às finanças.

4. Monitoramento do Desempenho: Ao acompanhar os custos, as empresas podem avaliar o desempenho em relação às metas e objetivos estabelecidos. Isso é crucial para a gestão eficaz e a tomada de decisões informadas.

Em resumo, a gestão financeira desempenha um papel essencial na sustentabilidade e no sucesso de uma empresa. O controle de custos é um componente crítico dessa gestão, permitindo a identificação de ineficiências, a precificação adequada, o planejamento financeiro e o monitoramento do desempenho.

3.4.1 CLASSIFICAÇÃO DOS CUSTOS

Abaixo estão elencados os valores das peças e serviços envolvidos no nosso produto. Com esses valores podemos calcular o valor total do projeto e analisar a viabilidade do mesmo. Custo é todo gasto relacionado e utilizado na produção de bens e/ou serviços. São classificados como direto e indireto:

Custo Direto: Esses são os gastos que podem ser diretamente atribuídos ao produto ou projeto em questão. Eles são específicos e claramente relacionados às atividades de produção. Em outras palavras, são os custos que surgem diretamente da aquisição de materiais, mão de obra e outros recursos diretamente envolvidos na fabricação ou execução do projeto. Por exemplo, se você está fabricando um carro, os custos diretos incluiriam o metal usado na carroceria, os salários dos operários da linha de montagem e os pneus. Segundo Schier (2006, p.26):

> Custos diretos são os custos que podem ser identificados e quantificados no produto ou no serviço e valorizados com relativa facilidade. Os materiais diretos, por exemplo, são normalmente requisitados com a identificação prévia de sua utilização, ou seja, ao emitir a requisição para o almoxarifado, o responsável pela produção já que não podem ser identificados de forma fácil, não podem ser apropriados de forma direta para as unidades específicas como, por exemplo, mão-de-obra indireta e matérias indiretas.

Já os custos indiretos, como define Martins (2003, p.32), são custos que não oferecem uma medida para o produto, ele aponta ainda que:

> [...] o rol dos Custos Indiretos inclui Custos Indiretos propriamente ditos e Custos Diretos (por natureza), mas que são tratados como Indiretos em função de sua irrelevância ou da dificuldade de sua medição, ou até do interesse da empresa em ser mais ou menos rigorosa em suas informações. Pode-se inclusive dizer também que, entre os Indiretos, existem os menos indiretos (quase Diretos) .

Para VanDerbeck e Nagy (2001), os custos indiretos são todos os custos que não são identificados diretamente ao produto acabado, eles não podem ser localizados em um ponto específico na produção. Com o avanço da tecnologia e a automatização das fábricas, o percentual de custos indiretos vêm aumentando drasticamente no custo total de fabricação.

A tabela apresentada abaixo demonstra os itens usados na produção de uma unidade do nosso sensor e classifica o custo, sendo que são todos diretos. Essa tabela foi utilizada para

o planejamento financeiro, orçamentação e controle de custos para gerenciar os gastos associados ao nosso empreendimento.

Cada linha da tabela inclui três elementos principais:

1. Item: Descreve o nome do componente usado no sensor.

2. Tipo de Custo: Indica se o custo é direto ou indireto, como explicado acima.

3. Valor: Representa o custo monetário associado a cada item. Os valores são listados em reais (R$) e podem ser uma compra única, um valor por unidade ou um valor por hora, dependendo da natureza do item.

Tabela 1 - tabela de custos diretos

ITENS DO SENSOR	**TIPO DE CUSTO**	**VALOR**
Uno R3	DIRETO	R$45,00
Adaptador ESP8266	DIRETO	R$15,00
Cabos Jumper	DIRETO	R$2,25
Sensor Ultra sônico	DIRETO	R$9,40
Modulo ESP8266	DIRETO	R$15,00
Total		R$86,65

Fonte: autores (2023)

A tabela abaixo demonstra o nosso custo fixo mensal, sendo que o tipo de custo é indireto. Esse custo precisa ser pago pela venda total dos sensores.

Tabela 2 - tabela de custos indiretos

CUSTO MENSAL		
Aluguel	INDIRETO	R$800,00
Telefonia	INDIRETO	R$150,00
Salário Funcionários (2)	INDIRETO	R$2658 ,00
Total		R$3608,00

Fonte: autores (2023)

3.4.2 CUSTOS DO PRODUTO / SERVIÇO

O custo da unidade do nosso produto é de R$86,65, que é resultado da soma de todos os itens necessários para a construção de uma única unidade.

Porém dentro desse custo, se faz necessário adicionar o preço fixo dos custos indiretos listados na tabela acima, então o custo fixo deverá ser dividido entre todos os sensores vendidos no mês.

Ao analisarmos os valores comparativos em relação ao nosso produto, observamos que não existem produtos similares no mercado nacional que incorporem a mesma inovação que o nosso. Os produtos analisados no mercado com soluções parecidas com a nossa (não iguais) custam em torno de R$160,00 e R$250,00. Por isso decidimos declarar um valor de venda de R$200,00, fazendo com que a venda mínima mensal para o pagamento dos custos deverá ser de 32 unidades, não havendo lucro e nem prejuízo. Porém estipulamos nossa meta de vendas em 50 unidades, havendo uma markup de 25,94% aplicado.

Portanto, nossa solução se revela altamente viável diante das opções disponíveis no mercado brasileiro.

Os produtos referenciais foram: a lixeira Townew T1 Smart Trash Can. Esta lixeira possui uma abertura e fechamento automáticos e uma função de selagem que fecha automaticamente o saco de lixo cheio e cria um novo saco.

A lixeira iTouchless Deodorizer Automatic Sensor Trash Can. Esta lixeira é equipada com sensores que abrem a tampa automaticamente quando você se aproxima.

3.4.3 PRECIFICAÇÃO

A precificação é o processo de estabelecer o preço de venda de um produto ou serviço. É uma atividade estratégica que envolve considerar diversos fatores, como custos, valor percebido pelo cliente, concorrência, elasticidade da demanda e objetivos de lucro. A precificação eficaz visa equilibrar esses fatores para maximizar a lucratividade e a competitividade no mercado.

Importância da Precificação:

A precificação desempenha grande importância para a empresa, sendo algumas delas listadas abaixo:

1. Lucratividade: A precificação adequada impacta diretamente a lucratividade. Preços muito baixos podem prejudicar a rentabilidade, enquanto preços muito altos podem afastar os clientes. Como afirma Kotler e Armstrong (2018) em "Princípios de Marketing", a precificação é uma das alavancas mais poderosas para aumentar os lucros.

2. Competitividade: Preços competitivos podem ajudar a empresa a conquistar uma fatia maior do mercado e superar a concorrência. De acordo com Monroe (2003) em "Pricing: Making Profitable Decisions", a precificação eficaz é essencial para a conquista de mercado.

3. Maximização de Receitas: Uma estratégia de precificação bem elaborada pode ajudar a empresa a maximizar suas receitas. Isso envolve considerar a elasticidade da demanda, de acordo com Stigler (1987) em "The Economics of Information," e ajustar os preços de acordo.

4. Cobertura de Custos: Os preços devem ser suficientes para cobrir todos os custos de produção e operacionais. Isso é essencial para manter a saúde financeira da empresa (Horngren et al., 2012, "Cost Accounting").

A tabela apresenta informações relevantes para a precificação do nosso produto. Aqui está uma explicação dos elementos da tabela:

- Custo Variável: Refere-se aos custos que variam com a produção de cada unidade. No caso da lixeira, o custo variável é de R$86,65 por unidade, representando o custo da matéria-prima.

- Custo Fixo: São os custos que não variam com a produção, como aluguel, salários fixos, etc. O custo fixo do Sensor é de R$72,16 por unidade.

- **Custo Total:** É a soma do custo variável e do custo fixo. Para a lixeira, o custo total é de R$158,81 por unidade.

- Preço: Representa o preço de venda da lixeira no mercado, que é de R$200,00 por unidade.

- Margem de Contribuição: É a diferença entre o preço de venda e o custo variável. No caso do nosso projeto, a margem de contribuição é de R$41,19 por unidade.

- Margem: É a porcentagem da margem de contribuição em relação ao preço de venda, que neste caso é de 20,60%. Isso significa que 20,60% do preço de venda contribui para cobrir os custos fixos e gerar lucro.

- Mark Up: É um índice que indica quanto o preço de venda é maior do que o custo variável. No caso do nosso produto, o mark-up é de 25,94%.

- Índice de Markup: O índice de markup é uma relação entre o preço de venda e o custo variável. Neste caso, o índice de markup é 1,26.

Portanto, a tabela fornece informações essenciais para a precificação do produto, considerando seus custos e a margem de contribuição. Isso é fundamental para determinar o preço de venda adequado que equilibra a rentabilidade e a competitividade no mercado.

Tabela 3 - tabela de precificação

		Variável						
Produto	**Custo fixo**	**Matéria Prima**	**Custo Total**	**Preço**	**Margem contribuição**	**Margem**	**Mark up**	**Índice markup**
Sensor	R$ 72,16	R$ 86,65	R$158,81	R$ 200,00	R$ 41,19	20,60%	25,94%	1,26

Fonte: autores (2023)

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS

A Formação para a Vida, como um dos eixos do Projeto Pedagógico de Formação por Competências da UNIFEOB, destaca a importância da aplicabilidade e utilidade dos conhecimentos adquiridos na formação acadêmica para a sociedade como um todo. Essa abordagem vai além do ambiente acadêmico, buscando promover a integração dos estudantes com a comunidade e a sua participação ativa em projetos que tragam benefícios sociais.

A extensão universitária desempenha um papel fundamental nesse contexto, pois é por meio dela que os conhecimentos e habilidades desenvolvidos pelos estudantes são colocados em prática, em parceria com a comunidade. Essa interação possibilita a troca de experiências, a identificação de demandas reais e a busca por soluções inovadoras e sustentáveis.

Portanto, a Formação para a Vida, aliada à extensão universitária, busca promover a aplicabilidade dos conhecimentos adquiridos na UNIFEOB, capacitando os estudantes para enfrentar desafios reais e contribuir para o desenvolvimento sustentável da sociedade

3.5.1 GERENCIANDO FINANÇAS

- **Tópico 1:** Introdução aos conceitos econômicos e financeiros básicos.

Neste tópico introdutório sobre conceitos econômicos e financeiros, exploramos a relação entre finanças, economia e contabilidade. Finanças são a gestão do dinheiro, enquanto a economia estuda a produção, circulação e consumo de bens e serviços, com foco em maximizar o lucro (microeconomia) e entender estruturas institucionais (macroeconomia). A

contabilidade analisa variações no patrimônio, o registro das transações e a administração de recursos, diferenciando regimes de caixa e competência.

Para aplicar esses conceitos, é importante entender a classificação dos gastos, como despesas, investimentos e custos. Por exemplo, o pagamento de aluguel é uma despesa, a compra de máquinas e móveis é um investimento, e os itens comprados para produção são custos que se tornam despesas quando contribuem para a geração de receita.

No contexto pessoal, a gestão de custos se aplica da mesma forma, controlando gastos e investimentos para manter a saúde financeira. Ter conhecimento sobre o nível de gastos permite tomar decisões financeiras adequadas. O fluxo de caixa é uma ferramenta essencial para entender as entradas e saídas de recursos, tanto no passado (fluxo de caixa realizado) quanto nas projeções futuras (fluxo de caixa projetado).

Por exemplo, analisar seu extrato bancário é uma forma simples de aplicar o conceito de fluxo de caixa na vida pessoal. Além disso, as decisões financeiras, sejam pessoais ou empresariais, devem estar alinhadas com objetivos de curto e longo prazo. O controle de entradas e saídas de recursos é crucial tanto para empresas quanto para indivíduos, e é essencial para o sucesso financeiro no dia-a-dia.

- **Tópico 2:** Entendendo o ambiente: independência financeira, o valor da minha riqueza e o registro do dia a dia.

Neste, aprofundamos nosso entendimento sobre independência financeira, o valor da riqueza pessoal e o registro do dia a dia das finanças. Abaixo estão os pontos importantes:

1. Fonte de Rendimentos: Para alcançar um resultado financeiro positivo, é fundamental identificar como ganhar dinheiro e decidir como alocar os recursos restantes após os custos. Fontes de renda incluem faturamento para empresas, salários para indivíduos e até empreendedorismo.

2. Redução de Custos: Além de gerar receita, é essencial controlar os gastos. Para reduzir custos, evite dívidas bancárias com altas taxas de juros, negocie preços com fornecedores, tome decisões para controlar gastos e avalie despesas menos relevantes, como refeições fora de casa.

3. Conceitos de Investimentos: Discutimos gastos para investimento, que visam ao aumento do patrimônio, como a compra de um carro ou uma casa. Também exploramos

investimentos financeiros, onde é importante escolher opções alinhadas com seu perfil de risco. Exemplos incluem poupança, tesouro direto, títulos de renda fixa, ações e outros.

4. Demonstrações Financeiras: Recomendamos a criação de relatórios financeiros periódicos para análise e tomada de decisões. As empresas têm demonstrações contábeis, mas as pessoas físicas podem trabalhar com o fluxo de caixa, que registra movimentações em três categorias: atividades operacionais, de investimento e de financiamento.

5. Exemplo de Fluxo de Caixa: Fornecemos um exemplo simples de um fluxo de caixa pessoal por seis meses, mostrando entradas (como salário) e saídas (como despesas pessoais). O saldo final é calculado após cada mês.

No dia a dia, as pessoas podem aplicar esses conceitos para melhorar a gestão de suas finanças, alcançar independência financeira e tomar decisões inteligentes sobre investimentos. Essas práticas valem tanto para empresas quanto para indivíduos, independentemente do tamanho de seus recursos.

- **Tópico 3:** Dívidas e juros compostos, opções de empréstimo e alternativas ao endividado.

Neste tópico, abordamos Matemática Financeira com foco em dívidas, juros simples e juros compostos, opções de empréstimo e alternativas para quem está endividado. Aqui está uma síntese dos principais pontos:

Matemática Financeira:

- A Matemática Financeira estuda o valor do dinheiro no tempo, usando análises de fluxos de caixa que representam entradas (seta para cima) e saídas (seta para baixo) de valores monetários.

Juros Simples:

- No sistema de juros simples, apenas o capital inicial é usado no cálculo dos juros durante o período da aplicação.

- Exemplo: Se você investe R$1.000 a uma taxa de 10% ao mês, o cálculo de juros simples é: VF = VP + (VP x i x n).

- Cálculo: VF = 1.000 + (1.000 x 0,10 x 3) = 1.300,00.

Juros Compostos:

- O sistema de juros compostos calcula juros sobre o montante acumulado, levando em consideração os juros acumulados anteriormente.

- Exemplo: Com os mesmos R$1.000 a 10% ao mês, o cálculo de juros compostos é: VF = VP + (1 + i)^n.

- Cálculo: VF = 1.000 + (1 + 0,10)^3 = 1.331,00.

Concessão de Crédito:

- Para determinar o valor do crédito, é essencial coletar informações sobre o histórico da empresa (para pessoas jurídicas) ou documentos como comprovantes de renda (para pessoas físicas).

- A análise de crédito é fundamental para evitar riscos em operações de recebimentos duvidosos.

Organização das Finanças:

- Controle de entradas e saídas financeiras, definição de prioridades e elaboração de um orçamento financeiro são ações importantes para a organização financeira pessoal ou empresarial.

Planejamento de Ações Orçamentárias:

- A estruturação do planejamento financeiro envolve dois principais componentes: planejamento financeiro e planejamento estratégico. Eles estão interligados e direcionam as ações futuras.

- O controle é uma parte essencial desse processo para garantir que as metas sejam alcançadas.

Neste contexto, a Matemática Financeira desempenha um papel crucial em tomadas de decisão financeira, seja para investir, emprestar ou gerenciar dívidas. É importante entender como diferentes sistemas de juros afetam seus ganhos e dívidas, bem como como a análise de crédito é vital para evitar riscos. Além disso, o planejamento financeiro e estratégico são fundamentais tanto para indivíduos quanto para empresas.

- **Tópico 4:** Estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos.

Neste tópico final sobre "Gerenciando Finanças", exploramos como estabelecer metas para a realização de sonhos e envolver grupos para atingir objetivos. Aqui está uma síntese dos principais pontos, acompanhados de exemplos práticos:

Sonhos e Projetos:

- Todos nós temos sonhos e desejos para o futuro, que variam em magnitude e custo.

- Alguns sonhos requerem um esforço financeiro maior, como uma viagem internacional, enquanto outros são mais simples, como comprar um par de sapatos novos.

- A educação financeira é fundamental para equilibrar as finanças pessoais e realizar esses sonhos.

Planejamento de Sonhos:

- Transformar sonhos em projetos envolve planejamento e organização.

- Exemplo: Se deseja uma viagem à Espanha para um curso de espanhol, pergunte a si mesmo quanto tem guardado, qual será o custo da hospedagem, transporte, o curso, alimentação, etc.

- Planejar ajuda a determinar se é o momento certo para realizar o projeto ou se é melhor adiá-lo.

Mitigos nas Finanças Pessoais:

- Abordamos mitos que podem prejudicar o gerenciamento financeiro.

- Exemplo: O mito de que apenas pessoas com grandes somas de dinheiro podem investir é derrubado, mostrando opções de investimento acessíveis a partir de pequenos valores.

Atitudes para o Sucesso Financeiro:

- Além do planejamento, são necessárias ações para alcançar objetivos financeiros.

- Exemplo: Traçar objetivos claros, estabelecer prazos razoáveis e planejar recursos são passos essenciais para realizar os sonhos.

Opções Financeiras para a Aposentadoria:

- Discutimos a importância do planejamento da aposentadoria e oferecemos opções, incluindo planos de aposentadoria complementar, como a previdência privada e investimentos de longo prazo.

- Exemplo: Se começar a poupar R$200 mensalmente a partir dos 25 anos, pode economizar um valor substancial para a aposentadoria.

O encerramento enfatiza que a atitude, o foco e a organização nas finanças pessoais são fundamentais para garantir a realização de sonhos e uma aposentadoria tranquila. O tópico destaca a importância de evitar mitos financeiros e tomar decisões informadas para construir um futuro financeiramente seguro.

3.5.2 ESTUDANTES NA PRÁTICA

No podcast abaixo foi abordado as principais tomadas de decisão para que ocorra uma boa gestão financeira.

https://youtu.be/fOZGLjkAlmQ?si=MkGcTr7FNV_1PU9a

ISSN 1983-6767

4. CONCLUSÃO

Após a conclusão deste Projeto de Modelagem e Desenvolvimento de Sistemas, podemos destacar os principais pontos abordados e os resultados alcançados. Ao longo de todo o processo, a equipe do projeto formada pelo Grupo GATE5 em parceria com a UNIFEOB trabalhou para desenvolver soluções inovadoras e sustentáveis na área de gestão de resíduos de diversos ambientes.

Ao longo do projeto, conhecimentos de disciplinas como Modelagem de Dados, Gestão Financeira, Lógica de Programação, Programação Orientada a Objeto e Autoconhecimento foram aplicados para criar um aplicativo de interface amigável para a integração de soluções empresariais em relação ao lixo produzido nos âmbitos. Ao integrar banco de dados, conectividade e aplicações intuitivas, as nossas soluções proporcionam uma abordagem eficiente e prática coordenação da sustentabilidade local.

Durante o desenvolvimento do projeto enfrentamos desafios logísticos e de agrupamento como atividade de integração de banco de dados, programação de software e criação de interfaces. Mas, com o esforço e trabalho em equipe de todos, superamos essas dificuldades e alcançamos resultados satisfatórios.

Além disso, a aplicação de conceitos de marketing digital permite-nos criar uma imagem de marca sólida através do desenvolvimento de um site completo e da utilização das redes sociais para promover as nossas soluções. Isso nos permite atingir um público mais amplo e aumentar a visibilidade do projeto.

O trabalho em equipe é crucial para o sucesso do projeto, pois cada membro contribui com suas habilidades e conhecimentos específicos. A colaboração e a troca de ideias são fundamentais para a criação de um produto final de qualidade.

O projeto oferece uma valiosa oportunidade de aplicar os conhecimentos adquiridos ao longo do curso e desenvolver habilidades práticas exigidas no mercado de trabalho. Através da combinação entre tecnologia, sustentabilidade e empreendedorismo, a equipe GATE5 e a UNIFEOB conseguiram criar uma solução eficiente e relevante para os usuários.

Apesar dos desafios encontrados, os resultados finais demonstraram o comprometimento e a capacidade de superar obstáculos para atingir os objetivos propostos.

Estamos orgulhosos do trabalho que realizamos e acreditamos que os nossos contentores inteligentes podem dar um contributo significativo para melhorar a gestão de resíduos e proteger o ambiente.

Este projeto é um marco na nossa formação acadêmica e um passo em direção a um futuro mais sustentável e tecnologicamente avançado. Estamos gratos pela oportunidade de participar neste desafio e esperamos ver o impacto positivo que as nossas soluções têm na sociedade.

ISSN 1983-6767

REFERÊNCIAS

Ana Clara: MARQUES SIQUEIRA, Gestão de custos: um enfoque no custeio de projetos da Associação Junior Achievement de Minas Gerais. Belo Horizonte, 2020. Disponível em: https://www.pucminas.br/iceg/Documents/cont/Monografia%20Ana%20Clara%20Marques%20Siqueira%202.2020.pdf. Acesso em: 27 de setembro de 2023.

Duckett, J. (2011). HTML and CSS: Design and Build Websites.

EUSER, Carlos Alberto. Projeto de Banco de Dados. 4. ed. Rio Grande do Sul: Sagra, 1994.

GUILAR, Luis J. Fundamentos de programação. Grupo A, 2008. E-book. ISBN 9788580550146. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788580550146/. Acesso em: 21 ago.2023.

GITMAN, L. J., & ZUTTER, C. J. Princípios de Administração Financeira. Pearson, 2018.

HORNGREN, C. T., DATAR, S. M., & RAJAN, M. V. Contabilidade de Custos. Pearson, 2012.

KORTH, H. F.; SILBERSCHATZ, A.; SUDARSHAN, S. Sistema de banco de dados. 6. ed. Rio de Janeiro: Campus, 2012.

Kotler, P., & Armstrong, G. Princípios de Marketing. Pearson, 2018.

Lea Verou. (2015). CSS Secrets: Better Solutions to Everyday Web Design Problems.

MARTINS, Eliseu. Contabilidade de custos. 9. ed. São Paulo: Atlas, 2003.

Monroe, K. B. Pricing: Making Profitable Decisions. McGraw-Hill/Irwin, 2013.

NAGY, Charles F.; VANDERBECK, Edward J. Contabilidade de custos. Trad. Robert Brian Taylor. 11. ed. São Paulo: Cengage Learning, 2001.

ROSS, S. A., WESTERFIELD, R. W., & JORDAN, B. D. Corporate Finance. McGraw-Hill, 2019.

SCHIER, Carlos Ubiratan da Costa. Gestão de custos. Curitiba: Ibpex, 2006.

SEBESTA, Robert. Conceitos de linguagens de programação. Grupo A, 2018. E-book. ISBN 9788582604694. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/. Acesso em: 21 ago. 2023.

Silveira, P., & Almeida, A. 2016. Desenvolvimento de Software. Casa do Código.

SILVA, Fabricio M.; LEITE, Márcia C D.; OLIVEIRA, Diego B. Paradigmas de programação. Grupo A, 2019. E-book. ISBN 9788533500426. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788533500426/. Acesso em: 21 ago. 2023.

STIGLER, G. J. The Economics of Information. Journal of Political Economy, 95(1), 1-21, 1987.

Townew T1 Smart Trash Can (15,5 liter). townew, 2021. Disponível em: https://townew.eu/product/townew-t1-smart-trash-can/. Acesso em: 27 de setembro de 2023.

"Algoritmos e Estruturas de Dados" por Niklaus Wirth. JavaScript: O guia definitivo .Lógica de Programação e Algoritmos com JavaScript. Flanagan, D. (2016).

ISSN 1983-6767

ANEXOS

Imagens das outras telas desenvolvidas para a matéria de Lógica de Programação:

Figura 15 - Tela de Recuperação de senha:

Fonte: autores (2023)

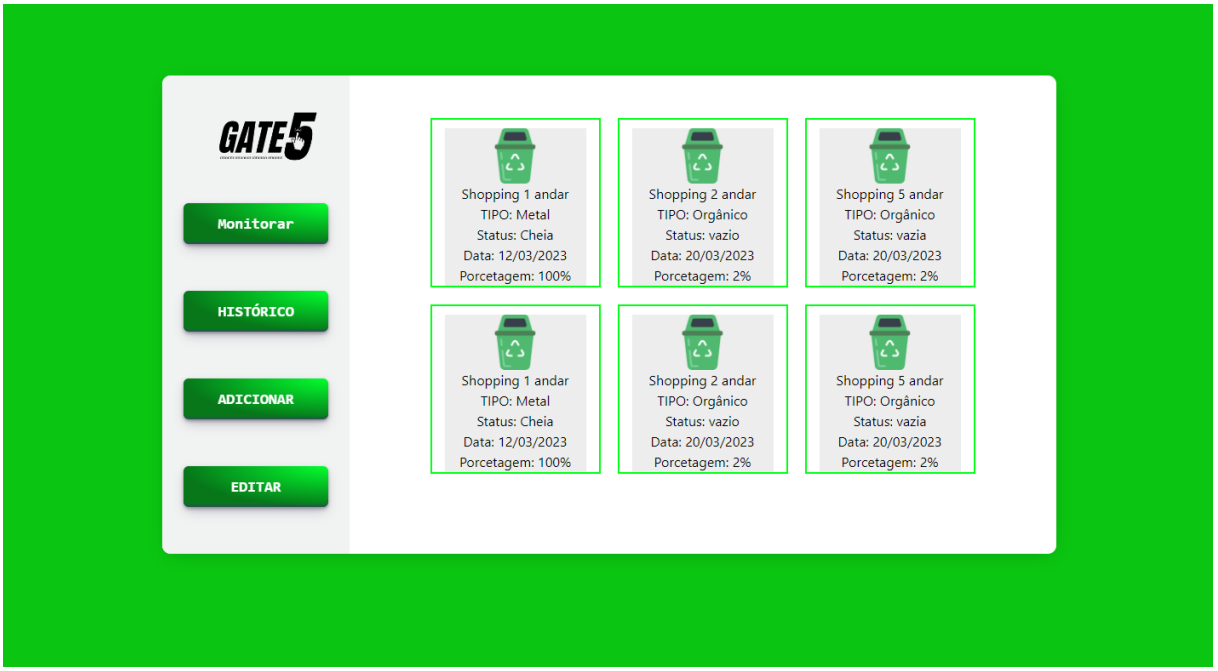
Figura 16 - Tela de Verificação:

Fonte: autores (2023)

Figura 17 - Tela de Alteração de senha:

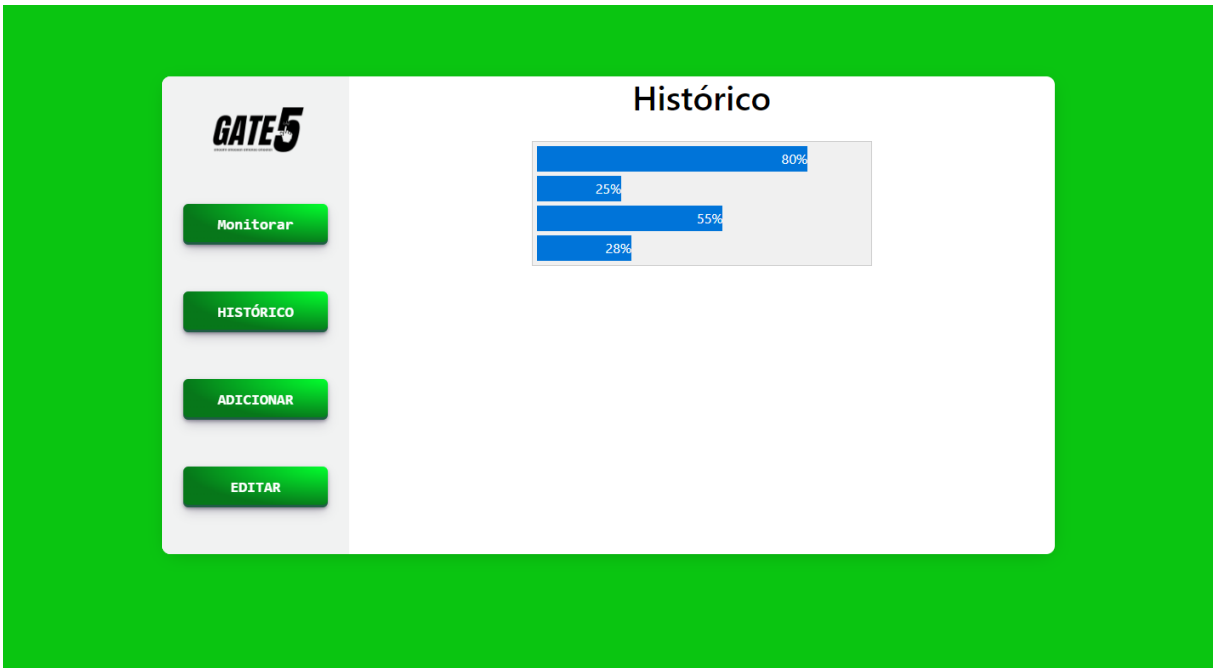
Fonte: autores (2023)

Figura 18 - Tela de Monitoramento:



Fonte: autores (2023)

Figura 19 - Tela de Histórico:



Fonte: autores (2023)

Figura 20 - Tela de Adicionar a Lixeira:

Fonte: autores (2023)

Figura 21 - Tela de Editar a Lixeira:

Fonte: autores (2023)