UNIFEOB
ESCOLA DE NEGÓCIOS

2024

PROJETO INTEGRADO

UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

DESENVOLVIMENTO DE SOLUÇÕES CONSOLE INTEGRADAS PARA EDUCAÇÃO, SUSTENTABILIDADE, INCLUSÃO SOCIAL E EMPREENDEDORISMO

SUPERMERCADO SÃO RAFAEL

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2024

UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

DESENVOLVIMENTO DE SOLUÇÕES CONSOLE INTEGRADAS PARA EDUCAÇÃO, SUSTENTABILIDADE, INCLUSÃO SOCIAL E EMPREENDEDORISMO

SUPERMERCADO SÃO RAFAEL

MÓDULO MODELAGEM E DESENVOLVIMENTO DE SISTEMAS

Business Intelligence – Profª. Mariângela Martimbianco Santos

Programação Orientada a Objeto – Prof. Nivaldo de Andrade

Lógica de Programação – Prof. Marcelo Ciacco Almeida

Modelagem de Dados – Prof. Max Streicher Vallim

Projeto de Modelagem e Desenvolvimento de Sistemas – Profª. Mariângela M. Santos

Estudantes:

Calebe Matheus Moreira Moraes, RA 24000974

Gabriel Henrique Navela Casarini, RA 24000914

Gustavo de Moraes Donadello, RA 24000419

Márcio Augusto Garcia Soares, RA 24000138

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2024

SUMÁRIO

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 E POLIMORFISMO. 11
      3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS 11
   3.2 LÓGICA DE PROGRAMAÇÃO 12
      3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE 12
      3.2.2 DESENVOLVIMENTO DE APLICAÇÕES 13
      3.2.3 IMPLEMENTAÇÃO E VALIDAÇÃO 14
   3.3 MODELAGEM DE DADOS 15
      3.3.1 MODELO CONCEITUAL 15
      3.3.2 MODELO LÓGICO E FÍSICO 16
      3.3.3 SQL 17
   3.4 BUSINESS INTELLIGENCE 18
      3.4.1 ORGANIZAÇÃO E IDENTIFICAÇÃO DAS INFORMAÇÕES 19
      3.4.2 MANIPULAÇÃO E ANÁLISE DE DADOS 19
      3.4.3 CRIAÇÃO DE MODELOS DE ANÁLISE DE DADOS 20
   3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS 21
      3.5.1 GERENCIANDO FINANÇAS 21
      3.5.2 ESTUDANTES NA PRÁTICA 25
4. CONCLUSÃO 26

REFERÊNCIAS 27

ANEXOS 29

ISSN 1983-6767

1. INTRODUÇÃO

Figura 1: Logo Programa Recicla

Fonte: Autores

Neste Projeto Integrado, tivemos como inspiração uma rede de supermercados, focando na criação de um **Sistema de Coleta Seletiva**, com o objetivo de promover práticas sustentáveis e reduzir o impacto ambiental causado pelo descarte inadequado de resíduos. A proposta visa envolver clientes e colaboradores, conscientizando-os sobre a importância da separação correta de materiais recicláveis, reforçando o compromisso da empresa com a sustentabilidade e incentivando os colaboradores com **benefícios** provenientes do próprio mercado como por exemplo: vale-compras, descontos em produtos e prêmios que poderiam ser sorteados entre todos os contribuintes ao **Sistema Recicla**.

Com isso, nosso projeto se alinha ao conceito proposto pelo **Projeto Integrado**, que visa desenvolver um sistema via console nas áreas propostas. A solução desenvolvida contará com uma interface acessível, permitindo que os usuários se cadastrem e acompanhem suas ações de coleta seletiva, gerando dados que auxiliam na otimização de políticas de reciclagem dentro da rede de supermercados. Além de educar a comunidade, o sistema apoia o desenvolvimento de estratégias mais sustentáveis.

Nosso projeto também se destaca pelo incentivo ao **empreendedorismo**, ao transformar o processo de coleta seletiva em uma oportunidade de negócios. O supermercado poderá apoiar e colaborar com empreendedores e cooperativas de reciclagem, promovendo novos modelos de negócios direcionados para a gestão de resíduos e economia circular.

Concluindo, nosso projeto se insere no **âmbito da sustentabilidade e empreendedorismo**, utilizando a tecnologia como ferramenta de inovação e desenvolvimento de soluções ambientalmente responsáveis e economicamente viáveis.

ISSN 1983-6767

2. DESCRIÇÃO DA EMPRESA

O **Supermercado São Rafael**, fundado em 1998, é uma empresa brasileira sediada em São João da Boa Vista-SP. Especializada no ramo do varejo alimentício, onde se trata de uma empresa de ciclo familiar onde temos a Rosana da Silva, dona da empresa, porém a grade de quem cuida gira em torno da família como por exemplo o Rafael (irmão) que é gerente/dono.

A empresa se trata daquele supermercado de bairro onde você sempre encontra rostos conhecidos, aquele lugar pequeno, mas cheio de vida, que é gerenciado por uma família há anos. O pai, a mãe, os filhos, todos trabalham juntos, e isso cria uma sensação de proximidade com os clientes. Sendo um comércio que foi aumentando e evoluindo conforme os anos e a disponibilidade de oportunidades.

O mercado está localizado na **Rua Monsenhor João Ramalho - Jardim Nova Republica, São João da Boa Vista - SP**, com o **CNPJ 02.981.711/0001-10**. Seus sócios-administradores são Rosana da Silva e Rafael. A empresa pode ser contatada pelo telefone **(19) 3631-3528**.

ISSN 1983-6767

3. PROJETO INTEGRADO

O **Projeto Integrado de Sistema de Coleta (Recicla)** desenvolvido no Segundo Semestre conta com as seguintes unidades de estudo: Modelagem de dados, Bussiness Intelligence, Programação Orientada a Objetos e Lógica de Programação. Todas essa unidades em conjunto proporcionaram o desenvolvimento do projeto utilizando as ferramentas necessárias e os conceitos adquiridos durante o semestre dado pelos professores orientadores.

Na **Lógica de Programação**, tivemos abordagens a respeito dos conceitos e do desenvolvimento dos principais pontos da programação (Algoritmos, funções, variáveis, estruturas de controle, dentre outros).

Em **Bussiness Intelligence**, a criação de dashboards foi extremamente abordado durante o semestre, com isso realizamos inúmeras atividades para reforçar o entendimento da criação de interfaces, dessa forma pudemos utilizar o aprendizado para o desenvolvimento do Projeto com o PowerBI.

Na **Modelagem de Dados**, aprendemos conceitos essenciais na criação de um banco de dados, indo desde o modelo conceitual até o modelo físico onde ocorre a população das tabelas com comandos do MySQL (INSERT, SELECT, DELETE, dentre outros).

A **Programação Orientada a Objetos** foi essencial para o desenvolvimento visto que nela aprendemos a cerca de classes e objetos na linguagem Python, além disso possibilitou o desenvolvimento de forma fluida do Projeto já que esse paradigma é de extrema importância para sistemas que visam um funcionamento e uma manutenção eficientes.

3.1 PROGRAMAÇÃO ORIENTADA A OBJETO

Segundo o site PythonAcademy, a programação orientada a objetos (**POO**) é um paradigma de programação baseado no conceito de **Classes** e **Objetos**.

Conceitos Essenciais de POO:

- Classes: são tipos de dados definidos pelo desenvolvedor que atuam como um modelo para objetos.

- Objetos: são instâncias de uma classe.
- Métodos: são funções definidas dentro de uma classe que descreve os comportamentos de um objeto.
- Atributos: são definidos na Classe e representam o estado de um objeto.

Pilares de POO:

- Encapsulamento: utilizado para juntar, ou encapsular, dados e comportamentos relacionados em entidades únicas, que são chamados de objetos. O Encapsulamento também afirma que informações importantes devem ser contidas dentro do objeto de maneira privada e apenas informações selecionadas devem ser expostas publicamente. Obs: Para utilizar esse princípio utiliza-se o sinal sublinhado (_).
- Abstração: oculta a complexidade de um sistema, focando nos aspectos essenciais de um objeto enquanto ignora detalhes desnecessários.
- Herança: possibilita a reutilização de código comum em uma relação de hierarquia entre Classes.
- Polimorfismo: permite que objetos de diferentes classes sejam tratados como objetos da mesma classe base, especialmente quando essas classes compartilham uma interface ou herança.

Através de todos esses conceitos abordados durante as aulas de POO, foi possível desenvolver o Projeto Integrado integrando ainda com as outras unidades de estudo.

3.1.1 CLASSES E OBJETOS

De acordo com o autor (Borges, 2014), em sua obra “Python para desenvolvedores” , a classe é uma abstração que define as características e comportamentos que os objetos (instâncias) terão.

No Projeto de Sistema de Coleta (Recicla) temos 11 classes que funcionam com o propósito de fornecer uma solução a locais que procuram se destacar no âmbito da sustentabilidade promovendo como por exemplo processos de reciclagem.

Classes do Sistema:

- UsuarioDB; TipoUsuarioDB;
- SaidaDB; EntradaDB;

- MaterialDB; CategoriaDB
- EstadoDB; CidadeDB;
- ColetorDB; DepósitoDB; ;
- AgendamentoDB.

Detalhamento de uma das classes:

Class UsuarioDB:

Figura 2 (Bibliotecas e Conexão/Desconexão)

```python
import mysql.connector
from mysql.connector import Error
from tabulate import tabulate

class UsuarioDB:
    def __init__(self, host, database, user, password):
        self.host = host
        self.database = database
        self.user = user
        self.password = password
        self.connection = None

    def conectar(self):
        try:
            self.connection = mysql.connector.connect(
                host=self.host,
                database=self.database,
                user=self.user,
                password=self.password
            )
            if self.connection.is_connected():
                print("Conexão com o banco de dados estabelecida.")
        except Error as e:
            print(f"Erro ao conectar no banco de dados: {e}")
            self.connection = None

    def desconectar(self):
        if self.connection.is_connected():
            self.connection.close()
            print("Conexão com o banco de dados encerrada.")
```

- Nessa primeira imagem temos a importação de 2 bibliotecas, a criação do método construtor recebendo as informações para a conexão com o banco de dados, e outros dois métodos chamados conectar() e desconectar(), ambos relacionados com o banco.

ISSN 1983-6767

Figura 3 (Inserir Usuário)

```
def inserir_usuario(self, nome, email, telefone, tipo_usuario_id):
    if self.connection is not None and self.connection.is_connected():
        try:
            cursor = self.connection.cursor()
            sql_query = "INSERT INTO usuario (nome, email, telefone, tipoUsuario_idtipoUsuario) VALUES (%s, %s, %s, %s)"
            cursor.execute(sql_query, (nome, email, telefone, tipo_usuario_id))
            self.connection.commit()
            print(f"Usuário '{nome}' inserido com sucesso!")
        except Error as e:
            print(f"Erro ao inserir dados: {e}")
```

Figura 4 (Listar Usuário)

```
def listar_usuarios(self):
    if self.connection is not None and self.connection.is_connected():
        try:
            cursor = self.connection.cursor()
            cursor.execute("""
                SELECT u.idusuario, u.nome, u.email, u.telefone, t.tipo
                FROM usuario u
                INNER JOIN tipoUsuario t ON u.tipoUsuario_idtipoUsuario = t.idtipoUsuario
            """)
            usuarios = cursor.fetchall()
            if usuarios:
                print("\n=== Usuários Cadastrados ===")
                headers = ["ID", "Nome", "Email", "Telefone", "Tipo de Usuário"]
                print(tabulate(usuarios, headers=headers, tablefmt="grid"))
            else:
                print("Nenhum usuário cadastrado.")
        except Error as e:
            print(f"Erro ao buscar dados: {e}")
```

Figura 5 (Atualizar Usuário)

```
def atualizar_usuario(self, id_usuario, novo_nome, novo_email, novo_telefone, tipo_usuario_id):
    if self.connection is not None and self.connection.is_connected():
        try:
            cursor = self.connection.cursor()
            sql_query = "UPDATE usuario SET nome = %s, email = %s, telefone = %s, tipoUsuario_idtipoUsuario = %s WHERE idusuario = %s"
            cursor.execute(sql_query, (novo_nome, novo_email, novo_telefone, tipo_usuario_id, id_usuario))
            self.connection.commit()
            print(f"Usuário ID {id_usuario} atualizado.")
        except Error as e:
            print(f"Erro ao atualizar dados: {e}")
```

Figura 6 (Excluir Usuário)

```python
def excluir_usuario(self, id_usuario):
    if self.connection is not None and self.connection.is_connected():
        try:
            cursor = self.connection.cursor()
            sql_query = "DELETE FROM usuario WHERE idusuario = %s"
            cursor.execute(sql_query, (id_usuario,))
            self.connection.commit()
            print(f"Usuário ID {id_usuario} excluído com sucesso!")
        except Error as e:
            print(f"Erro ao excluir dados: {e}")
```

- Nestas imagens temos os métodos principais (Create, Read, Update e Delete - CRUD) da classe usuário na qual realizam a inserção de um novo usuário, a listagem de usuários junto da biblioteca tabulate, a atualização e a exclusão de usuários.

Figura 7 (Funções de Interface)

```python
# Funções de interface para o menu
def cadastrar_usuario(db_usuario):
    nome = input("Digite o nome do usuário: ")
    email = input("Digite o email do usuário: ")
    telefone = input("Digite o telefone do usuário: ")
    tipo_usuario_id = input("Digite o ID do tipo de usuário: ")
    db_usuario.inserir_usuario(nome, email, telefone, tipo_usuario_id)

def editar_usuario(db_usuario):
    id_usuario = input("Digite o ID do usuário que deseja editar: ")
    novo_nome = input("Digite o novo nome do usuário: ")
    novo_email = input("Digite o novo email do usuário: ")
    novo_telefone = input("Digite o novo telefone do usuário: ")
    tipo_usuario_id = input("Digite o novo ID do tipo de usuário: ")
    db_usuario.atualizar_usuario(id_usuario, novo_nome, novo_email, novo_telefone, tipo_usuario_id)

def excluir_usuario(db_usuario):
    id_usuario = input("Digite o ID do usuário que deseja excluir: ")
    db_usuario.excluir_usuario(id_usuario)
```

- Nesta terceira imagem temos as funções de interface na qual são chamadas dependendo da opção que o usuário preferir. Através disso um sub menu da opção escolhida irá aparecer e possibilitará realizar uma das ações. Em “Anexos” é possível visualizar as demais classes clicando no repositório do GitHub.

ISSN 1983-6767

3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.

Pilares de POO:

- Encapsulamento: utilizado para juntar, ou encapsular, dados e comportamentos relacionados em entidades únicas, que são chamados de objetos. O Encapsulamento também afirma que informações importantes devem ser contidas dentro do objeto de maneira privada e apenas informações selecionadas devem ser expostas publicamente. Obs: Para utilizar esse princípio utiliza-se o sinal sublinhado (_).
- Abstração: oculta a complexidade de um sistema, focando nos aspectos essenciais de um objeto enquanto ignora detalhes desnecessários.
- Herança: possibilita a reutilização de código comum em uma relação de hierarquia entre Classes.
- Polimorfismo: permite que objetos de diferentes classes sejam tratados como objetos da mesma classe base, especialmente quando essas classes compartilham uma interface ou herança.

Em POO, observamos que todos os conceitos utilizados em conjunto proporcionam um sistema funcional e de fácil manutenção. Sendo assim, através dos aprendizados obtidos com as aulas, aplicamos em nosso projeto os principais fundamentos.

No nosso Projeto, realizamos a criação das classes passando como parâmetro no método construtor as informações necessárias para realizar a conexão com o banco de dados. Todos esses parâmetros são recebidos em uma instância no arquivo “main.py” e ali será onde ocorrerá a execução final do sistema, já visualizado no último tópico na classe usuário.

Em relação aos métodos presentes no código, fizemos 4 métodos principais em cada classe no qual realizam a função do CRUD, dessa forma temos as 4 ações principais no sistema. Além deles, temos outros que realizam a conexão com o banco e mais outros métodos que fazem funções para cada ação principal que seriam as funções de interface.

3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

Tipos de Métodos:

- Métodos Públicos: São métodos acessíveis de fora da classe e representam a interface pública para interagir com os objetos. Permitem que outras partes do código acessem funcionalidades da classe.

- Métodos Privados: São métodos usados internamente na classe e indicados com dois underscores no início do nome (__metodo). Não devem ser acessados diretamente de fora da classe, pois são projetados para encapsular a lógica interna.
- Métodos Estáticos: São declarados com o decorador @staticmethod, não precisam acessar dados específicos da instância (self) e podem ser chamados diretamente pela classe. São usados para operações que não dependem de instâncias.

Em nosso projeto aplicamos os conceitos para que fosse possível tornar nosso sistema melhor e com fácil manutenção. Em nosso código, realizamos a utilização de métodos públicos nas classes, que seriam as 4 funções principais em cada uma (Inserir, Listar, Atualizar e Excluir).

3.2 LÓGICA DE PROGRAMAÇÃO

A **lógica de programação** é a base para o desenvolvimento de sistemas onde aplicamos os conceitos de algoritmos, representando o conjunto de regras que são aplicadas para criar códigos que serão interpretados e executados pelo computador (ALURA, 2023). No contexto do projeto de **Sistema de Coleta Seletiva**, a lógica é aplicada em inúmeras funcionalidades do sistema. Usando conceitos como variáveis, classes, funções, estruturas condicionais e operadores, o sistema organiza e calcula os dados, possibilitando funcionalidades no sistema que tenham por finalidade a promoção de práticas sustentáveis na comunidade.

3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

No desenvolvimento de um Sistema de Coleta Seletiva, é indispensável a compreensão dos conceitos fundamentais da lógica de programação. Sendo assim, entendê-los é essencial para criar algoritmos eficientes e desenvolver soluções de software robustas (CODECRUSH, 2023).

1. **Algoritmos**: Sequências lógicas de instruções para alcançar um objetivo específico, fundamentais para guiar cada ação de um programa.
2. **Variáveis e Tipos de Dados**: Armazenam e definem o tipo de informações, como números e textos.
3. **Funções**: Agrupam instruções que realizam tarefas específicas e podem ser reutilizadas.
4. **Estruturas Condicionais**: Permitem ao programa tomar decisões com base em condições.
5. **Operadores**: Realizam cálculos, comparações e operações lógicas, integrando flexibilidade às funções do sistema.

3.2.2 DESENVOLVIMENTO DE APLICAÇÕES

No desenvolvimento do projeto de Sistema de Coleta Seletiva (Recicla), implementamos conceitos que vão desde a lógica inicial até o paradigma de Programação Orientada a Objetos.

Explicação do funcionamento do Projeto:

O sistema irá receber inicialmente uma entrada de material reciclável, este podendo ser de qualquer categoria (metal, papelão, papel, etc) ou um material de maior dificuldade de descarte (bateria), em um local denominado “depósito” o qual escolhemos o Supermercado São Rafael. Nele as pessoas que descartarem resíduos poderão se cadastrar no sistema e assim seria possível obter uma gestão da quantidade de material descartado por cada usuário.

Por seguinte todo esse material seria destinado a uma saída o qual destinaria-os para um coletor, local este que é responsável por realizar atividades de reciclagem, dessa forma os materiais teriam um destino direto a pontos que possibilitam a reutilização dos resíduos.

Além disso, outra funcionalidade que adicionamos ao sistema foi o agendamento que possibilitaria a escolha do dia pelo coletor para ir buscar os materiais armazenados no depósito quando já houver uma quantidade considerável.

Programação Orientada a Objetos - Classes:

Ao todo foram criadas 11 classes, todas possuindo métodos que possibilitam a inserção, leitura, atualização e a exclusão dos elementos. Todas estão interligadas com o banco de dados MySQL possibilitando assim o CRUD (Create, Read, Update e Delete).

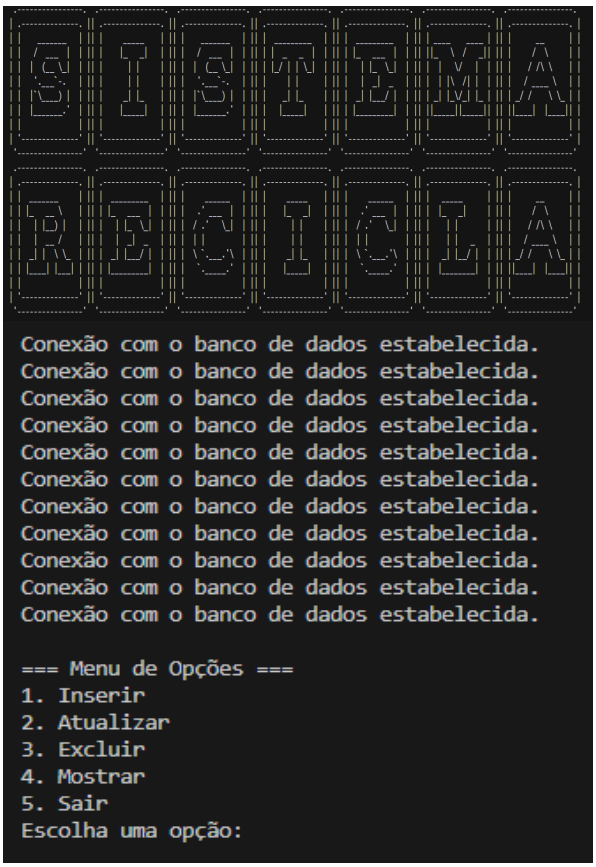
Em todas as classes realizamos as conexões com o banco de dados através da biblioteca mysql.connector. No próprio método construtor das classes colocamos as informações necessárias para poder conectar com o banco e assim poder instanciar as informações de forma que a conexão seja estabelecida no arquivo “main.py”, arquivo este que possibilita a execução total do sistema. Em “Anexos” é possível observar todas as classes e funções presentes no projeto ao clicar no repositório do GitHub.

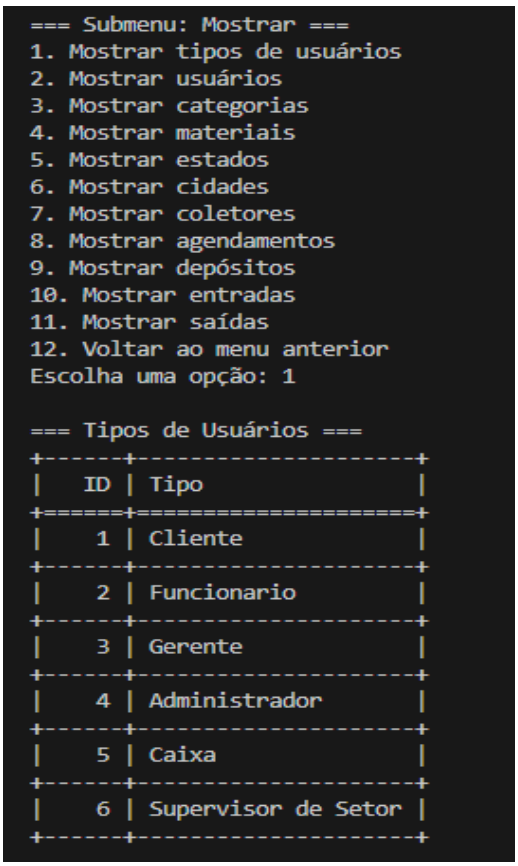
3.2.3 IMPLEMENTAÇÃO E VALIDAÇÃO

A validação do Sistema de Coleta Seletiva (Recicla) foi realizada e tivemos êxito na organização e na execução do sistema, sendo assim todos os procedimentos que eram visados no início do projeto foram alcançados e possibilitaram um sistema console que demonstrasse o funcionamento com opções de inserção, leitura, atualização e exclusão.

- Imagens do funcionamento do Sistema no **TERMINAL** - Tabelas de informações (Conectado com o banco de dados)

Figura 8 e 9 (Tela Inicial/ SubMenu)





Além dessas interfaces, temos outras que possibilitam a visualização de outras informações, sendo possível vê-las ao clicar no link do GitHub presente em “Anexos”.

ISSN 1983-6767

3.3 MODELAGEM DE DADOS

“Em sua essência, a modelagem de dados envolve a criação de modelos conceituais, lógicos e físicos que descrevem como os dados estão relacionados entre si e como são armazenados e acessados dentro de um sistema.” (ALURA 2024)

Tal conceito pode ser aplicado em inúmeros locais e em todos eles a modelagem faz um papel fundamental para o funcionamento adequado. Um exemplo seria na área da Saúde onde a modelagem de dados se faz essencial em algumas atividades de uma clínica tendo como exemplo: o gerenciamento de registros médicos eletrônicos, análise de resultados de testes clínicos, previsão de surtos de doenças e desenvolvimento de tratamentos personalizados.

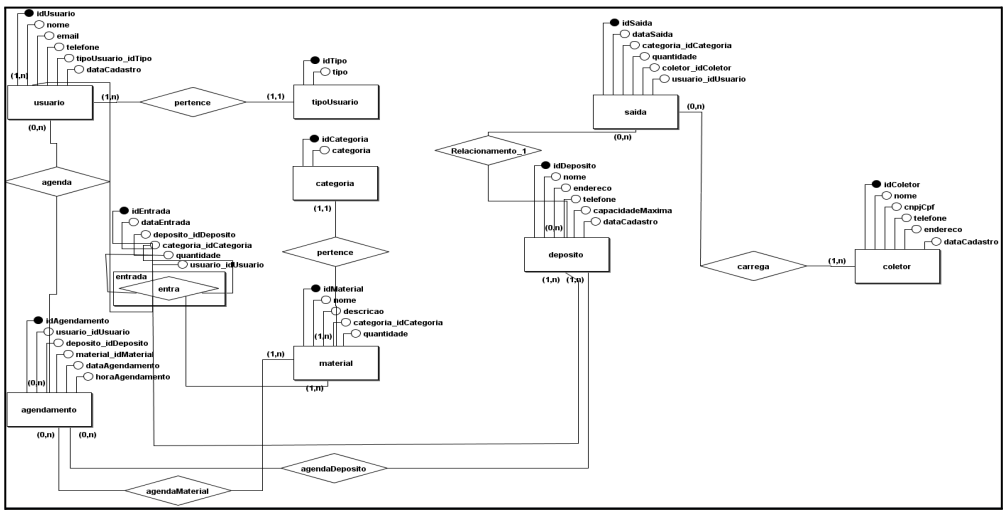
A modelagem de dados sendo assim procura desenvolver modelos que possam facilitar o entendimento dos dados e assim poder promover um banco de dados relacional e funcional para um sistema. Com o nosso Projeto não foi diferente, nele procuramos aperfeiçoar os modelos Conceitual, responsável por definir as principais entidades, Lógico, que descreve relacionamentos e estruturas de dados mais detalhadamente e o Físico que acaba colocando tudo isso em prática para que torne possível realizar a conexão com o código do Sistema de Coleta (Recicla).

3.3.1 MODELO CONCEITUAL

O modelo conceitual foi a etapa inicial no desenvolvimento do banco de dados. Ela representou os dados de forma abstrata, descrevendo apenas as entidades, atributos e relacionamentos do sistema sem considerar aspectos técnicos de implementação. Esse modelo é independente de qualquer SGBD, ele nos permitiu ser um guia de entendimento tanto para desenvolver quanto para testar funcionalidades.

A utilização do BRModelo foi a principal ferramenta que utilizamos para fazer a parte conceitual da base de dados, ela oferece várias vantagens, incluindo a criação de diagramas de entidade-relacionamento (ER) com clareza e facilidade. Estes diagramas ajudam a organizar e visualizar a estrutura do banco de dados de maneira a facilitar a análise e coleta de requisitos do sistema.

Figura 10 (Diagrama Entidade Relacionamento - DER)



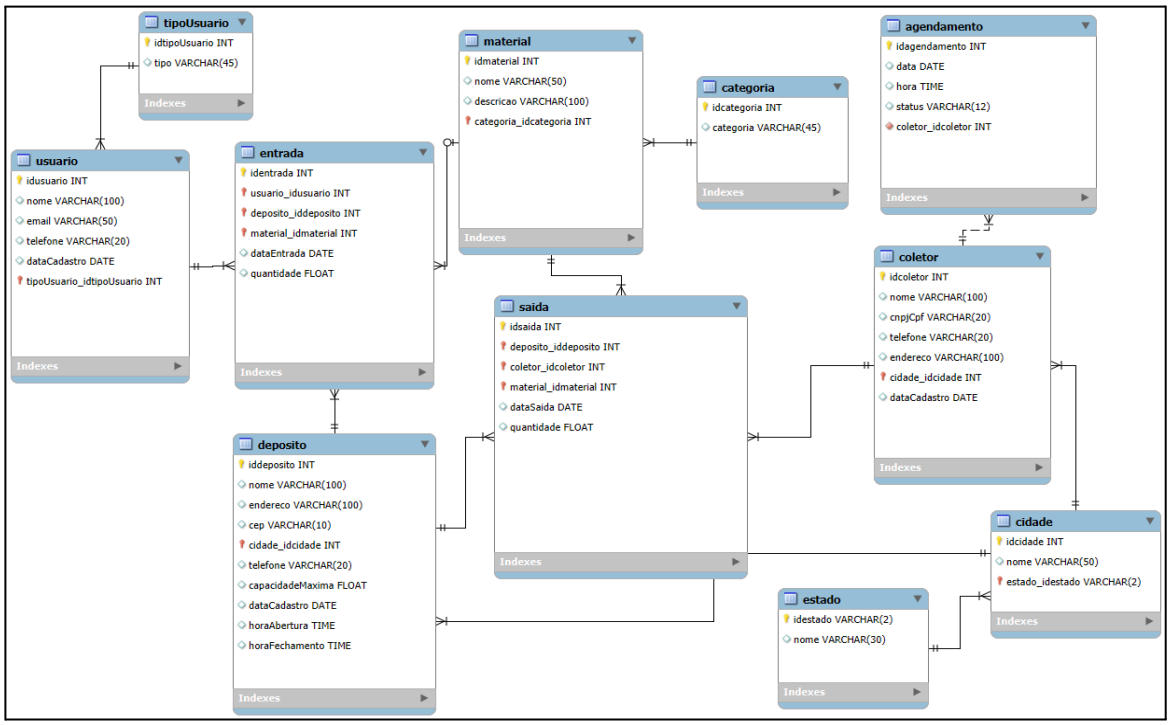
3.3.2 MODELO LÓGICO E FÍSICO

Foi utilizado o MySQL para transformar nosso modelo conceitual, criado no BRModelo, em um banco de dados implementável, facilitando a criação do modelo físico por meio da geração de scripts SQL automáticos. Essa escolha foi por conta do software ser de código aberto, amplamente utilizado por sua performance e escalabilidade.

O modelo lógico é a etapa intermediária que ocorre após a modelagem conceitual e que antecede o modelo físico. Ele representa a estrutura do banco de dados em um formato mais detalhado e específico, definindo as tabelas, atributos e relacionamentos de forma que sejam compatíveis com o sistema de gerenciamento de banco de dados escolhido.

Especificamos cada atributo das tabelas com tipos de dados e restrições, como NOT NULL, UNIQUE E DEFAULT, que são cruciais para garantir a consistência e integridade dos dados. A normalização foi aplicada para evitar redundância e melhorar a eficiência da base de dados, dividindo as informações em tabelas apropriadas e estabelecendo relacionamentos de forma clara.

Figura 11 (Modelo Lógico)



O modelo físico é a última fase da criação e representa a implantação real no SGBD. No MySQL, o modelo lógico já desenvolvido foi transformado no modelo físico por meio da geração de scripts SQL automáticos, que criam as tabelas e definem chaves e índices. O script foi gerado para criar cada tabela com seus atributos e restrições, Além disso, foram incluídos índices (INDEX) para otimizar a performance do banco de dados e chaves estrangeiras para garantir a integridade referencial. Uma forma de boas práticas para ser integrado em diferentes ambientes e suportar grandes volumes de informações.

3.3.3 SQL

“SQL é uma linguagem padrão para gerenciar e manipular bancos de dados relacionais. É usada para definir, consultar e modificar dados armazenados, além de gerenciar a estrutura das informações. Sua versatilidade e facilidade de uso fazem dela uma ferramenta essencial para o desenvolvimento de sistemas de armazenamento e análise de dados.” (ALURA 2023)

No projeto, foi necessário criar diversas consultas SQL para obter dados relevantes de múltiplas tabelas. A habilidade de construir consultas utilizando SELECT, WHERE, e agregações com GROUP BY permitiu extrair insights detalhados e transformar dados brutos em informações significativas.

O uso de INNER JOIN foi indispensável para unir informações de tabelas diferentes e criar uma visão mais completa dos dados. Por exemplo, para compilar informações de cadastros de usuários com a tabela tipos de usuários. Nas aulas de SQL mostraram na prática como fazer junções precisas e eficientes.

Foi criada uma VIEW unindo todas as tabelas, ela foi pensada para otimizar a integração com o Power Bi e garantir que as consultas fossem mais rápidas e organizadas sem a necessidade de edição dos campos. Com isso, foi possível pré-processar os dados no banco e diminuir a carga de trabalho do Power Bi, tornando a criação de relatórios mais eficiente.

3.4 BUSINESS INTELLIGENCE

O Power BI, desenvolvido pela Microsoft, é uma ferramenta poderosa de Business Intelligence que transforma dados brutos em insights visuais de forma interativa e acessível. Com ele, é possível conectar, modelar e visualizar informações de várias fontes em uma plataforma unificada, o que ajuda as empresas a monitorar e entender suas operações em tempo real por meio de relatórios e paineis visuais.

Entre suas funcionalidades, o Power BI permite criar relatórios dinâmicos e personalizáveis, usar gráficos interativos, analisar KPIs (Indicadores de Desempenho) e integrar dados de diferentes plataformas, como bancos de dados, planilhas e aplicativos. Assim, ele se torna essencial para decisões baseadas em dados, fornecendo uma compreensão detalhada

> Ter um negócio significa gerenciar uma série de informações sobre ele para compreender quais ações tomarão o futuro da empresa. Só que isso envolve dados de mais de um setor, e pode ficar um pouco complicado juntar e analisar tudo. É nesse momento que você entende a necessidade de contar com o Microsoft Power BI. (FREITAS, 2022)

Essa ferramenta facilita a criação de relatórios e o controle das operações, tornando o trabalho mais produtivo.

No contexto de um projeto de coleta seletiva em uma rede de supermercados, o Power BI atua como aliado estratégico. Ele permite centralizar e monitorar dados importantes, como volumes de materiais reciclados e o perfil dos consumidores, ajudando o supermercado a ajustar a estratégia de coleta seletiva e maximizar o impacto ambiental e o engajamento dos clientes.

ISSN 1983-6767

3.4.1 ORGANIZAÇÃO E IDENTIFICAÇÃO DAS INFORMAÇÕES

Com base nos dados que coletamos e organizamos, as principais informações identificadas na utilização do nosso sistema são:

- Dados de participação, em que consiste em número de usuários cadastrados e quantidade de cadastros;
- Tipagem de materiais coletados e categorias de materiais recicláveis (plástico, papel, vidro, metal), quantidade de cada tipo de material coletado, percentual de materiais recicláveis em relação ao total coletado;
- Localização de pontos de coleta, áreas ou bairros atendidos (lojas/supermercados), mapas de pontos de coleta;
- Impacto ambiental, como volume total de resíduos coletados
- Resultados e relatórios, que consiste em relatórios periódicos sobre o desempenho do sistema e análise de dados para ajustes e melhorias;
- Controle de retirada, consiste no registro de uma empresa terceirizada retirar os resíduos recicláveis, dados como retirada, registro de resíduos, quantidade reciclada, etc.

Essas informações oferecem uma visão abrangente da operação do sistema de coleta seletiva e são fundamentais para a formulação de estratégias de melhoria contínua. Ao analisar dados sobre a participação dos usuários, a tipagem e a quantidade de materiais recicláveis, a localização dos pontos de coleta e o impacto ambiental, podemos identificar oportunidades para aumentar a eficácia do programa.

Além disso, os relatórios periódicos e o controle da retirada de resíduos garantem uma gestão eficiente e transparente, promovendo o engajamento da comunidade e reforçando o compromisso do supermercado com a sustentabilidade.

3.4.2 MANIPULAÇÃO E ANÁLISE DE DADOS

Para garantir uma visão abrangente da eficácia do sistema de coleta seletiva no supermercado, se faz fundamental trabalhar com uma base de dados bem organizada, integrando dados internos e externos. Os dados internos são gerados diariamente pelo sistema de coleta, incluindo a quantidade de material reciclável coletado, categorizado por tipo (plástico, vidro, papel, etc.), e informações específicas sobre local e horário de cada coleta.

Esses dados são essenciais para medir a eficiência operacional do sistema e entender padrões de comportamento no uso.

Os dados externos complementam os internos, proporcionando uma visão mais contextual e estratégica. Eles contêm informações demográficas da região, taxas de reciclagem locais e dados de parceiros que indicam a quantidade de material reciclado. Essas informações são importantes para comparar o desempenho do sistema com metas regionais e identificar áreas com potencial de melhoria. Como afirma a Redação Impacta, "quando se possui tanto um banco de dados bem administrado como informações de qualidade, o resultado geralmente é o sucesso."

Com uma base de dados organizada, é possível criar indicadores-chave de desempenho (KPIs) que monitoram e avaliam o sistema. Por exemplo, uma quantidade total de material reciclado permite visualizar a contribuição de cada supermercado para a sustentabilidade. Os indicadores são exibidos em um dashboard, permitindo que a equipe visualize o progresso, compreenda padrões de uso e tome decisões informadas para otimizar a coleta seletiva. Assim, o projeto se torna uma ferramenta poderosa para promover a sustentabilidade e o engajamento da comunidade com a reciclagem.

3.4.3 CRIAÇÃO DE MODELOS DE ANÁLISE DE DADOS

O modelo a seguir aborda a criação dos principais dashboards que facilitarão a mensuração e a análise de dados para o nosso projeto de coleta seletiva. Esses dashboards serão essenciais para oferecer insights detalhados e promover uma compreensão mais clara das informações, permitindo uma gestão eficiente e orientada por dados na implementação da coleta seletiva:

Figura 12 (Menu Power BI)

ISSN 1983-6767

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS

O projeto de um Sistema de Coleta Seletiva, sob a perspectiva de Meio Ambiente e Sustentabilidade, está diretamente relacionado com a extensão universitária em várias partes:

Práticas Sustentáveis -

A extensão universitária busca fomentar práticas que beneficiem a sociedade. O projeto pode incluir a criação de pontos de coleta seletiva em locais estratégicos, além de parcerias com cooperativas de reciclagem. Isso não só melhora a gestão de resíduos, mas também gera empregos e renda para os catadores.

Ações Comunitárias -

A implementação do projeto deve envolver a comunidade local, promovendo ações como mutirões de limpeza e oficinas de reutilização de materiais. Essas atividades não só aumentam a participação da população, mas também fortalecem a conexão entre a universidade e a comunidade, essencial para a extensão universitária.

Avaliação de Impacto -

O projeto deve incluir um plano para avaliar seu impacto na comunidade, como a quantidade de resíduos reciclados e a mudança de comportamento da população. Essa avaliação pode ser um campo de estudo para alunos, contribuindo para o conhecimento acadêmico e a melhoria contínua do sistema.

Em resumo, o Sistema de Coleta Seletiva não apenas promove a sustentabilidade, mas também fortalece a relação entre o supermercado e a sociedade, oferecendo um modelo de aprendizado e aplicação que beneficia ambos.

3.5.1 GERENCIANDO FINANÇAS

Após a leitura da apostila "Gerenciando Finanças", compreendemos os tópicos abordados e elaboramos uma síntese sobre a influência da globalização e dos avanços tecnológicos nas finanças pessoais. Observamos que essas mudanças rápidas afetam a forma como gerenciamos nossos recursos, desde a facilidade de acesso a informações financeiras até

as novas ferramentas digitais que facilitam o controle orçamentário. Destacamos também a importância de encarar essas inovações como oportunidades para melhorar nossa educação financeira e tomar decisões mais informadas. Ao adaptar-nos a essas mudanças, podemos desenvolver habilidades que nos ajudem a alcançar uma maior estabilidade financeira, promovendo um crescimento pessoal e profissional significativo.

- Introdução aos conceitos econômicos e financeiros básicos

A compreensão dos conceitos econômicos e financeiros básicos é essencial para a gestão eficiente das finanças pessoais e a tomada de decisões informadas. Por exemplo, entender a diferença entre despesas fixas, como aluguel e contas de serviços públicos, e despesas variáveis, como alimentação e lazer, permite um planejamento orçamentário mais preciso. A gestão de custos pessoais, como criar uma planilha de controle de gastos, ajuda a identificar onde é possível economizar e a garantir estabilidade financeira. Além disso, desenvolver habilidades de decisão financeira, como a análise de investimentos ou a comparação de preços antes de compras importantes, capacita os indivíduos a otimizar o uso de seus recursos.

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

No caminho para a independência financeira, entender o valor da riqueza e registrar o fluxo financeiro do dia a dia são práticas fundamentais. Por exemplo, manter um diário de despesas ajuda a visualizar os gastos mensais e a identificar padrões de consumo. As fontes de rendimentos, como um emprego fixo ou renda extra por meio de freelancing, são a base para construir patrimônio. Reduzir custos, como optar por transporte público em vez de carro, pode liberar recursos para investimentos. O conhecimento sobre investimentos, como a aplicação em uma conta poupança ou em ações, permite decisões estratégicas em diferentes contextos econômicos. As demonstrações financeiras, como extratos bancários e relatórios de gastos, são essenciais para acompanhar o progresso financeiro e realizar ajustes necessários no planejamento.

- Dívidas e juros compostos, opções de empréstimo e alternativas ao endividado

A gestão de dívidas e a compreensão dos juros, tanto simples quanto compostos, são fundamentais para evitar o endividamento. Por exemplo, calcular o custo total de um

empréstimo usando a fórmula de juros compostos ajuda a tomar decisões mais informadas sobre o crédito. Avaliar as opções de empréstimo, como comparar taxas de juros entre instituições financeiras, é crucial para quem busca maior controle sobre suas finanças. Manter as finanças organizadas com um plano de ação orçamentário, como elaborar um orçamento mensal, é um passo decisivo para equilibrar receitas e despesas, prevenindo o acúmulo de dívidas e construindo uma trajetória de prosperidade a longo prazo.

- Estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos

Estabelecer metas claras para realizar sonhos é um processo que pode ser fortalecido ao envolver o grupo ou a comunidade a que pertencemos. Por exemplo, criar um grupo de apoio financeiro com amigos ou familiares pode aumentar as chances de sucesso em objetivos financeiros compartilhados, como poupar para uma viagem ou um projeto comum. Adotar hábitos saudáveis na gestão das finanças, como revisar regularmente o orçamento e fazer uma poupança mensal, é essencial para manter as finanças organizadas. Desmistificar mitos financeiros, como a crença de que é necessário ter um alto salário para poupar, ajuda a evitar armadilhas comuns. Atitudes positivas, como a disciplina em seguir um plano financeiro, são fundamentais para alcançar o sucesso. Por fim, planejar para a aposentadoria desde cedo, considerando opções financeiras adequadas, como planos de previdência privada ou contas de investimento, garante uma maior tranquilidade e segurança no futuro.

- Introdução aos conceitos econômicos e financeiros básicos

A compreensão dos conceitos econômicos e financeiros básicos é essencial para a gestão eficiente das finanças pessoais e a tomada de decisões informadas. Por exemplo, entender a diferença entre despesas fixas, como aluguel e contas de serviços públicos, e despesas variáveis, como alimentação e lazer, permite um planejamento orçamentário mais preciso. A gestão de custos pessoais, como criar uma planilha de controle de gastos, ajuda a identificar onde é possível economizar e a garantir estabilidade financeira. Além disso, desenvolver habilidades de decisão financeira, como a análise de investimentos ou a comparação de preços antes de compras importantes, capacita os indivíduos a otimizar o uso de seus recursos.

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

No caminho para a independência financeira, entender o valor da riqueza e registrar o fluxo financeiro do dia a dia são práticas fundamentais. Por exemplo, manter um diário de despesas ajuda a visualizar os gastos mensais e a identificar padrões de consumo. As fontes de rendimentos, como um emprego fixo ou renda extra por meio de freelancing, são a base para construir patrimônio. Reduzir custos, como optar por transporte público em vez de carro, pode liberar recursos para investimentos. O conhecimento sobre investimentos, como a aplicação em uma conta poupança ou em ações, permite decisões estratégicas em diferentes contextos econômicos. As demonstrações financeiras, como extratos bancários e relatórios de gastos, são essenciais para acompanhar o progresso financeiro e realizar ajustes necessários no planejamento.

- Dívidas e juros compostos, opções de empréstimo e alternativas ao endividado

A gestão de dívidas e a compreensão dos juros, tanto simples quanto compostos, são fundamentais para evitar o endividamento. Por exemplo, calcular o custo total de um empréstimo usando a fórmula de juros compostos ajuda a tomar decisões mais informadas sobre o crédito. Avaliar as opções de empréstimo, como comparar taxas de juros entre instituições financeiras, é crucial para quem busca maior controle sobre suas finanças. Manter as finanças organizadas com um plano de ação orçamentário, como elaborar um orçamento mensal, é um passo decisivo para equilibrar receitas e despesas, prevenindo o acúmulo de dívidas e construindo uma trajetória de prosperidade a longo prazo.

- Estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos

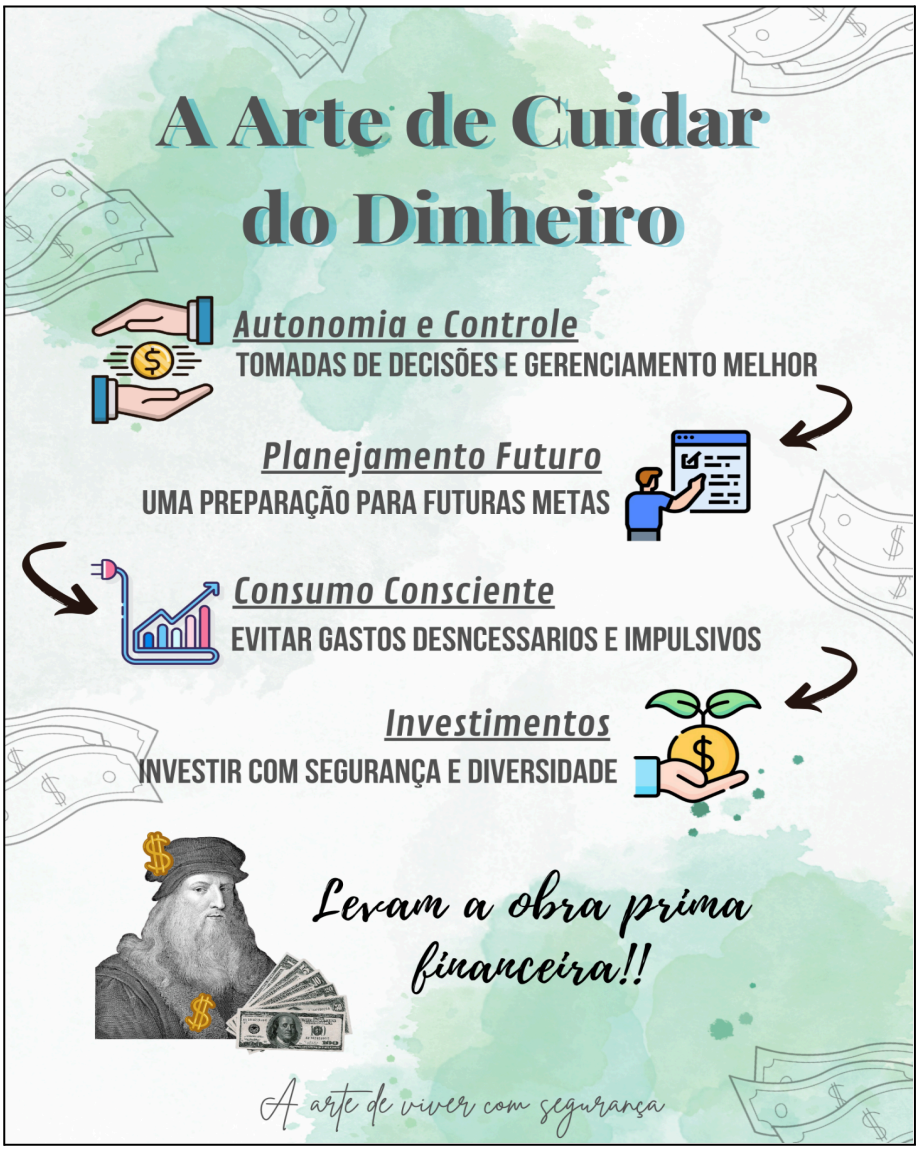
Estabelecer metas claras para realizar sonhos é um processo que pode ser fortalecido ao envolver o grupo ou a comunidade a que pertencemos. Por exemplo, criar um grupo de apoio financeiro com amigos ou familiares pode aumentar as chances de sucesso em objetivos financeiros compartilhados, como poupar para uma viagem ou um projeto comum. Adotar hábitos saudáveis na gestão das finanças, como revisar regularmente o orçamento e fazer uma poupança mensal, é essencial para manter as finanças organizadas. Desmistificar mitos financeiros, como a crença de que é necessário ter um alto salário para poupar, ajuda a evitar armadilhas comuns. Atitudes positivas, como a disciplina em seguir um plano financeiro, são fundamentais para alcançar o sucesso. Por fim, planejar para a aposentadoria desde cedo,

considerando opções financeiras adequadas, como planos de previdência privada ou contas de investimento, garante uma maior tranquilidade e segurança no futuro.

3.5.2 ESTUDANTES NA PRÁTICA

A educação financeira é fundamental para o desenvolvimento de habilidades que ajudam na gestão do dinheiro e na tomada de decisões financeiras conscientes. Ao compartilhar dicas e tópicos sobre este assunto, buscamos capacitar indivíduos a planejar seu futuro, evitar dívidas desnecessárias e investir de maneira inteligente. Através do conhecimento financeiro, podemos construir uma sociedade mais sustentável e próspera, onde todos tenham acesso a uma vida financeira saudável e equilibrada.

BANNER:



ISSN 1983-6767

4. CONCLUSÃO

Em resumo, nosso projeto busca promover a sustentabilidade e a consciência ambiental dentro de mercados ao estabelecer um sistema de coleta para materiais recicláveis e itens que exigem descarte especial. Com um ponto de coleta centralizado, a comunidade é incentivada a descartar corretamente os resíduos, garantindo um destino adequado para cada tipo de material.

O sistema facilita a gestão dessas contribuições e transforma o mercado em um agente de mudança, reforçando práticas ambientais responsáveis. Ao longo do tempo, ele amplia o impacto positivo do mercado, promovendo uma cultura de sustentabilidade e incentivando a comunidade a adotar uma postura mais ativa na preservação do meio ambiente.

Além de contribuir para a redução do impacto ambiental, o projeto fortalece o papel do mercado como ponto de conscientização e referência em práticas sustentáveis, aproximando a comunidade da causa ambiental e ajudando a construir um futuro mais verde e consciente.

ISSN 1983-6767

REFERÊNCIAS

RAMOS, Vinícius. **Programação Orientada a Objetos no Python: Introdução.** PythonAcademy, 2021. Disponível em: <Introdução à Programação Orientada a Objetos no Python> Acesso em: 26 out. 2024

HENRIQUE, João. **POO: o que é programação orientada a objetos?** Alura, 2023. Disponível em: <POO: o que é programação orientada a objetos? | Alura> Acesso em: 26 out. 2024

BORGES, Luiz Eduardo. **Python para Desenvolvedores**: Aborda Python. 3ª ed. São Paulo: Novatec, 2014.

BESSA, André. **Algoritmos e Lógica de programação**: O que são e qual a importância? Alura, 2023. Disponível em: <https://www.alura.com.br/artigos/algoritmos-e-logica-de-programacao>. Acesso em: 25 out. 2024.

PIUBELLO, Henrique. **Guia Completo de Lógica de Programação**: Conceitos Essenciais e Exemplos Práticos. CodeCrush, 2023. Disponível em: <Guia Completo de Lógica de Programação: Conceitos Essenciais e Exemplos Práticos>. Acesso em: 26 out. 2024

IGOR, Nascimento Alves. **O que é e para que serve a modelagem de dados?** Alura, 2024. Disponível em: <O que é e para que serve a modelagem de dados? | Alura> Acesso em: 01 nov. 2024

OLIVEIRA, Danielle. **SQL: O que é SQL e para que serve ?** Alura, 2023. Disponível em: <SQL: O que é, quais os comandos e como utilizar? | Alura> Acesso em: 08 nov. 2024

Freitas, Gabriel. **Power BI: por que meu negócio precisa dessa ferramenta?** v4Company, 2022. Disponível em:
<https://v4company.com/blog/trafego/power-bi>
Acesso em: 01 nov. 2024

Entenda a importância de um banco de dados em uma organização. BlogImpacta, 30 de Janeiro de 2017. Disponível em:
<https://www.impacta.com.br/blog/entenda-a-importancia-de-um-banco-de-dados-em-uma-organizacao/>
Acesso em: 01 nov. 2024