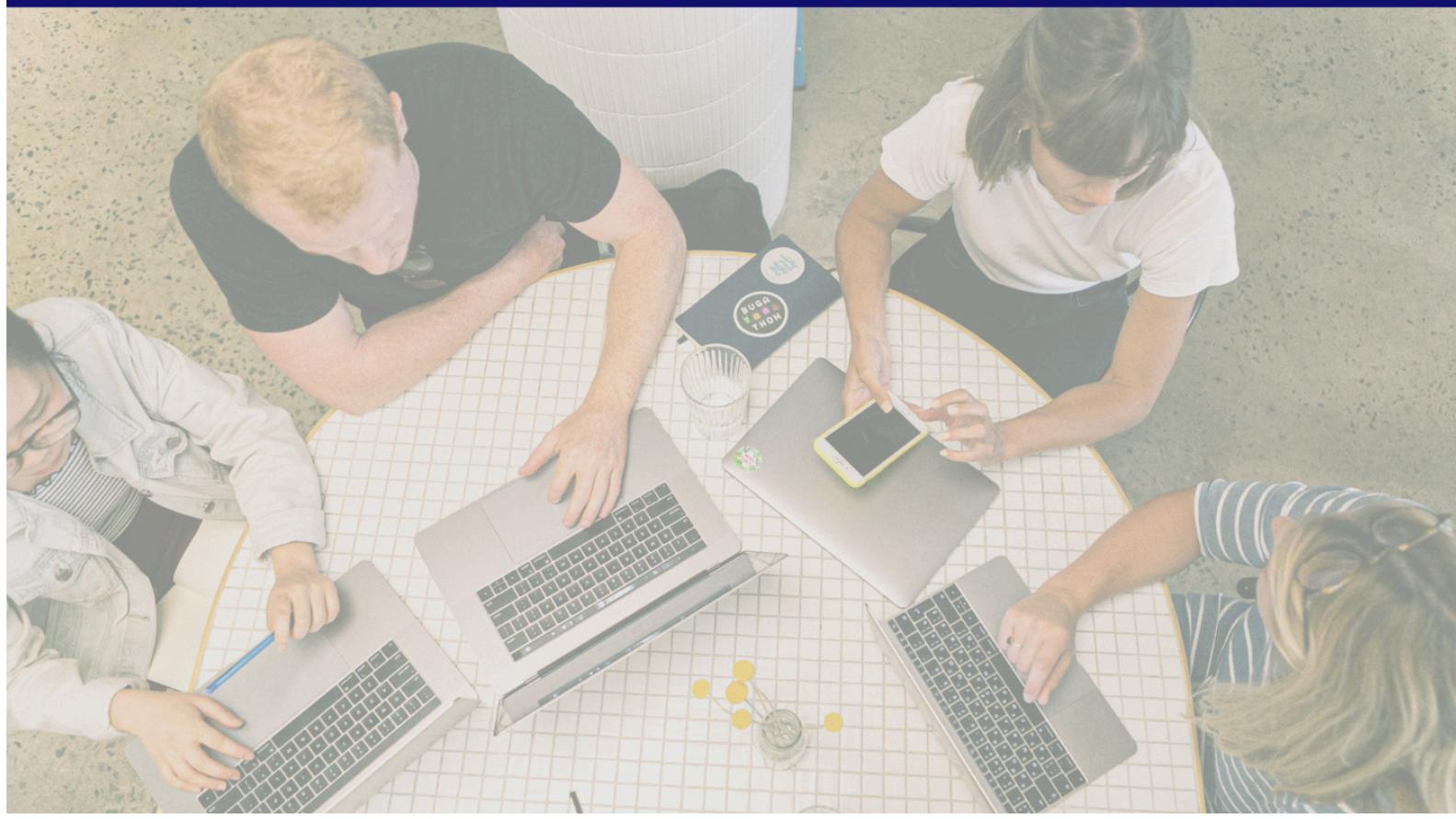


UNifeob
| ESCOLA DE NEGÓCIOS



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**

RESTAURANTE TIJOLEIRO

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**

<RESTAURANTE TIJOLEIRO>

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:

Álex Barbosa, RA 24000685

João Victor Viana Marcolino, RA 24001237

Logan Nakamura, RA 24000961

Maycon Willian Fagundes Dias, RA 24000976

Ray Martins Ajudarte Eleoterio, RA 24001232

SUMÁRIO

1. INTRODUÇÃO	4
2. DESCRIÇÃO DA EMPRESA	7
3. PROJETO INTEGRADO	8
3.1 PROGRAMAÇÃO ORIENTADA A OBJETO	8
3.1.1 CLASSES E OBJETOS	8
3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.	9
3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS	10
3.2 LÓGICA DE PROGRAMAÇÃO	10
3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE	11
3.2.2 DESENVOLVIMENTO DE APLICAÇÕES	11
3.2.3 IMPLEMENTAÇÃO E VALIDAÇÃO	11
3.3 MODELAGEM DE DADOS	11
3.3.1 MODELO CONCEITUAL	12
3.3.2 MODELO LÓGICO E FÍSICO	12
3.3.3 SQL	12
3.4 BUSINESS INTELLIGENCE	12
3.4.1 ORGANIZAÇÃO E IDENTIFICAÇÃO DAS INFORMAÇÕES	12
3.4.2 MANIPULAÇÃO E ANÁLISE DE DADOS	12
3.4.3 CRIAÇÃO DE MODELOS DE ANÁLISE DE DADOS	13
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS	13
3.5.1 GERENCIANDO FINANÇAS	13
3.5.2 ESTUDANTES NA PRÁTICA	14
4. CONCLUSÃO	16
REFERÊNCIAS	17
ANEXOS	18

1. INTRODUÇÃO

O projeto tem como objetivo a criação de um sistema console para coleta e análise de dados sobre os pedidos feitos no restaurante, visando otimizar e identificar os produtos mais vendidos, os dias da semana com mais movimentos e melhorar a experiência oferecida. O sistema será responsável por categorizar os clientes em diferentes perfis, como recorrentes, clientes quentes e aqueles em seu primeiro contato, com base em dados como nome, telefone e formas de contato, como telefone ou redes sociais. Além disso, o sistema registrará o motivo do contato, que pode envolver reservas, pedidos de delivery ou esclarecimento de dúvidas.

Através dessa coleta de informações, o sistema permitirá aos gestores do restaurante oferecer um atendimento mais personalizado, adaptado às necessidades de cada cliente. A análise desses dados possibilitará identificar padrões de comportamento e preferências, o que facilitará a implementação de estratégias de marketing mais direcionadas e a otimização dos serviços prestados. Dessa forma, o sistema não só apoiará a eficiência operacional, mas também contribuirá para a construção de um relacionamento mais próximo e duradouro com os clientes. Além dos benefícios para o restaurante, o projeto visa promover um ambiente de aprendizado para os colaboradores envolvidos, capacitando-os a utilizar a tecnologia de maneira criativa e eficaz na resolução de problemas reais. A solução também integra práticas de sustentabilidade com o objetivo de beneficiar não apenas o restaurante, mas também a comunidade local em que está inserido, criando um impacto positivo no desenvolvimento social e empresarial.

No projeto a tecnologia irá desempenhar um papel crucial na implementação do sistema e na operação do restaurante, o sistema console facilitará na coleta de dados pois permitirá compreender melhor a preferência dos clientes, e com isso levantar dados importantes para a tomada de decisão da diretoria.

2. DESCRIÇÃO DA EMPRESA

O Restaurante Tijoleiro, localizado no município de Jacutinga, MG, opera sob a razão social Restaurante Tijoleiro e possui o CNPJ 16.772.774/0001-70. Fundado em 1987, o restaurante está situado na Rua Artur Sereti Lion, 570, em uma área de intensa circulação, atraindo clientes de várias regiões devido à sua reputação e localização estratégica.

Conhecido por suas incríveis porções, como a famosa porção de torresmo, o Tijoleiro não funciona apenas como restaurante, mas também como um bar, oferecendo um ambiente aconchegante e animado para encontros sociais. O restaurante também se especializa na organização de eventos, como casamentos, aniversários e outros momentos especiais, personalizando tanto o espaço quanto o cardápio conforme as preferências dos clientes. O Tijoleiro é capaz de atender tanto a pequenos quanto a grandes grupos, sempre com um serviço de alta qualidade.

Produtos e Serviços: Além do serviço de marmitas, o Tijoleiro se destaca por seus pratos e porções à la carte, oferecendo uma vasta seleção que vai desde pratos especiais de carnes até diversas porções à moda mineira. Entre as opções mais populares estão os pratos com guarnições típicas da culinária mineira, que podem ser servidos de forma individual ou em porções compartilhadas, tornando o restaurante uma escolha ideal para refeições rápidas ou celebrações em grupo.

Contexto de Mercado: O Tijoleiro atende a uma clientela diversificada, incluindo moradores locais e clientes de outras regiões que buscam seu ambiente descontraído e as delícias da culinária mineira. Além do serviço de bar e restaurante, o Tijoleiro tem se adaptado à demanda crescente por delivery, ampliando sua capacidade de atender a um público ainda maior.

Dada a sua complexidade de operações, incluindo eventos, delivery e atendimento à la carte, há uma crescente necessidade de aprimorar o controle de informações relacionadas a preços, feedbacks e a experiência do cliente. Através do trabalho de banco de dados utilizando POO e ferramentas como o Power BI, será possível consolidar essas informações, o que permitirá à equipe do Tijoleiro tomar decisões estratégicas mais precisas. Isso inclui ajustar preços, melhorar a experiência dos eventos, otimizar o cardápio e responder de maneira mais eficiente tanto aos feedbacks positivos quanto às críticas construtivas, sempre com o objetivo de elevar a qualidade dos produtos e serviços oferecidos.

3. PROJETO INTEGRADO

Nesta etapa do PI, serão apresentados os conteúdos abordados nas disciplinas de BI, POO, Lógica de Programação e Banco de Dados, destacando como esses conhecimentos foram essenciais para o desenvolvimento do projeto. Cada uma dessas áreas foi aplicada de maneira estratégica na empresa escolhida, com o intuito de aprimorar processos e soluções.

Em Business Intelligence (BI), aplicamos conceitos para analisar dados e gerar relatórios que ajudaram a identificar tendências, comportamento de clientes e áreas de melhoria, influenciando decisões estratégicas. Em Programação Orientada a Objetos (POO), a utilização de classes e objetos foi essencial para criar soluções modulares e escaláveis, facilitando a manutenção e a implementação de funcionalidades complexas. Em Lógica de Programação, desenvolvemos algoritmos que otimizam processos e automatizam tarefas, economizando tempo e recursos. No Banco de Dados, projetamos um modelo de dados robusto, utilizando bancos relacionais para garantir a integridade das informações e facilitar a tomada de decisões rápidas e precisas.

3.1 PROGRAMAÇÃO ORIENTADA A OBJETO

A Programação Orientada a Objetos (POO) é uma maneira de organizar e escrever código de forma eficiente. Usando conceitos como classes, objetos, herança e polimorfismo, ela ajuda a criar programas que são mais fáceis de entender e dar manutenção. Por isso, a POO é importante no mercado de trabalho, pois facilita a criação de softwares que podem ser adaptados conforme necessário.

Conforme o site escolalinux.com.br ressalta

A Programação Orientada a Objetos (POO) é um paradigma de programação que organiza o código em torno de objetos, que são instâncias de classes. Ela se baseia em quatro princípios fundamentais: encapsulamento, herança, polimorfismo e abstração.

Na POO, um objeto é uma entidade que contém dados (atributos) e comportamentos (métodos). Os objetos interagem entre si através do envio de mensagens, o que permite a troca de informações e a execução de ações específicas.

Este paradigma também permite proteger e manipular dados de forma segura, o que facilita a integração com áreas como análise de dados e inteligência de negócios. Na prática, pode ser utilizada em áreas como educação, sustentabilidade e empreendedorismo, mostrando como a POO pode auxiliar na solução de problemas complexos.

3.1.1 CLASSES E OBJETOS

Dentro do nosso projeto, com base na programação orientada a objetos, montamos a estrutura da seguinte forma:

Classes principais: Banco, Endereco, Cliente, Finalidade, Meio_de_contato, Produtos, e Atendimentos. Teremos também a classe Menu que terá toda a lógica e organização dos menus que o usuário irá interagir.

Centralizamos dentro da classe bd, o método de conexão com o banco de dados, inserção, alteração e exclusão das informações das classes principais e nelas mantivemos apenas os atributos que irão compor os respectivos objetos.

Como exemplo eu instancio um novo objeto a partir da classe Cliente, para isso eu preciso passar os atributos principais (nome, telefone, id_endereco, ativo) para então enviar essas informações para o banco de dados via comandos em sql (query), realizado este processo temos um novo cliente cadastrado em nosso sistema pronto para realizar um pedido.

O paradigma da programação orientada a objetos contribui para o desenvolvimento de softwares, de maneira que facilita a modularização das funcionalidades, facilita na manutenção e facilita na compreensão de quem vai fazer a manutenção deste código, conforme o site devmedia.com reforça:

A programação orientada a objetos traz outros pontos que acabam sendo mais interessantes no contexto de aplicações modernas. Como o desempenho das aplicações não é uma das grandes preocupações na maioria das aplicações (devido ao poder de processamento dos computadores atuais), a programação orientada a objetos se tornou muito difundida. Essa difusão se dá muito pela questão da reutilização de código e pela capacidade de representação do sistema muito mais perto do que veríamos no mundo real.

3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.

No percurso de desenvolvimento do nosso aplicativo de console estiveram presentes alguns dos pilares da orientação a objetos, dentre eles um dos que utilizamos foi o encapsulamento, conforme ressalta o blog locaweb.com.br

O encapsulamento funciona com a utilização de modificadores de acesso para restringir o acesso aos atributos e aos métodos de um objeto. Garante que somente as classes apropriadas possam acessar as informações e ajudar a assegurar que os atributos e os métodos sejam usados de forma consistente e previsível.

Contudo a estrutura do projeto é principalmente composta por métodos que foram criados dentro da classe banco de dados onde interagimos com os atributos dos objetos instanciados com base nas classes definidas.

Como exemplo, abaixo teremos um trecho do código onde utilizamos o método “inserindo_cliente”, neste método fazemos a inserção de um novo cliente passando os atributos do objeto cliente instanciado, acompanhe:

Trecho de código “Inserção de clientes”

```
#FUNÇÕES PARA CLIENTES

def inserindo_cliente(self, cliente):
    try:
        query = "INSERT INTO tb_cliente (nome, telefone, tb_endereco_idtb_endereco) VALUES (%s, %s, %s);"
        self.cursor.execute(query, (cliente.nome, cliente.telefone, cliente.id_endereco))
        self.conection.commit()
        print(f"O registro foi inserido com sucesso!")
    except Error as e:
        print(f"Erro ao inseir informação: {e}")
```

Fonte: Autores

Seguindo esses conceitos e utilizando-os da maneira correta, construiremos uma solução segura, robusta, onde as informações principais ficarão protegidas, não teremos impactos negativos com a performance, nosso código ficará com a manutenção facilitada e quando houver a necessidade de implementar mais funcionalidades não teremos dificuldades.

3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

Completando a estrutura de nosso projeto temos a classe menu, ela é responsável por fazer a chamada dos métodos de criação, alteração e remoção de dados. O menu é composto por uma série de verificações onde o usuário escolhe qual opção entrar, a partir daí um função irá receber os valores que se tornaram os atributos das classes citadas, essas funções podem ser acessadas em qualquer lugar da aplicação, então tem características de método público.

Abaixo temos um dos exemplos dos métodos públicos da classe “Menu”, onde armazenamos valores em variáveis que em seguida se tornarão atributos da classe “Endereço” e então serão cadastrados no banco de dados.

Trecho de código “Inserção de clientes”

```
#ENDEREÇO

def inserir_endereco(self):
    cidade = input("Cidade: ").upper()
    bairro = input("Bairro: ").upper()
    rua = input("Rua: ").upper()
    numero = input("Numero: ").upper()
    status = 1
    endereco = Endereco(cidade, bairro, rua, numero, status)
    self.db.inserindo_endereco(endereco)
```

Fonte: Autores

3.2 LÓGICA DE PROGRAMAÇÃO

A lógica de programação foi essencial para que possamos avançar no desenvolvimento de sistemas em nosso projeto e aprender a construir soluções eficazes e bem estruturadas, abordando seus princípios e aplicações práticas para desenvolver uma base sólida em nosso PI. Para realizarmos com grande eficiência o código dele, precisamos aplicar diversas funções de lógica de programação que estabelecem regras e ordenam nosso código, um deles é seguir uma sequência lógica de algoritmos, principalmente em um projeto onde precisamos atender clientes, esclarecer dúvidas, realizar pedidos e estabelecer métodos de pagamentos e entregas de produto, para fazer com que um cliente passe por cada etapa necessária, é necessário utilizar da lógica e seguir uma sequência coesa e coerente. Seguindo essa aplicação, utilizamos diversos componentes de programação para construir nosso código, como variáveis para armazenar dados e estruturas condicionais para direcionar o fluxo do programa conforme as necessidades do usuário.

As estruturas condicionais desempenham um papel fundamental no desenvolvimento de nosso PI, pois com ela, podemos aplicar diversas alternativas em nosso projeto. Durante o desenvolvimento, serão explorados operadores lógicos e de comparação, que auxiliam no controle preciso do fluxo do código, possibilitando caminhos variados conforme os requisitos do nosso código. Todo o conteúdo foi desenvolvido na linguagem Python, reconhecida por sua simplicidade e acessibilidade, especialmente para iniciantes. Segundo o site: <https://www.alura.com.br/artigos/algoritmos-e-logica-de-programacao#logica-de-programacao>

Na lógica de programação é onde aplicamos todos os conceitos de algoritmos, a definição do passo a passo e transferimos toda a lógica do algoritmo desenvolvido

para uma linguagem de programação. Em linhas gerais, lógica de programação é todo conjunto de regras e conceitos que precisamos aplicar para criar códigos que serão interpretados e executados por um computador.

3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

Durante o desenvolvimento de nosso projeto, utilizamos diversos conceitos de lógica de programação como, variáveis e tipos de dados, funções, estruturas condicionais, operadores lógicos e dentre outros. Demos uso a variáveis e seus tipos de dados no PI para armazenar e classificar as informações que serão fornecidas pelo usuário, nos disponibilizando precisão e eficiência na manipulação dos dados durante a execução do programa, fazendo com que ele seja facilmente interpretado por quem modifica a estrutura de dados ou por quem estiver utilizando.

Funções e estruturas condicionais são de crucial importância dentro do desenvolvimento de um código, utilizamos funções para organizar nosso PI em blocos reutilizáveis, enquanto as estruturas condicionais permitem que o nosso programa tome decisões e adapte seu fluxo conforme condições específicas. Em nosso projeto, as funções aplicarão o funcionamento de cada elemento do código criado, onde podemos organizar cada classe com base em POO (Programação Orientada a Objetos) podendo reutilizar a mesma estrutura de código em outras partes fundamentais do código, tudo isso utilizando a linguagem de programação Python. Enquanto os Operadores Lógicos possibilitam operações lógicas e relacionais, onde podemos relacionar uma classe a outra baseando-se em lógica de programação, pois com essas duas bases, podemos criar códigos mais sólidos e complexos que o normal. Combinados, esses elementos constroem uma base sólida para o desenvolvimento do PI, promovendo a criação de soluções eficazes e escaláveis. De acordo com o site:

<https://awari.com.br/vantagens-processos-de-desenvolvimento-de-software-por-que-adotar-processos-formais/>

Os processos formais no desenvolvimento de software desempenham um papel fundamental na garantia da qualidade, eficiência e sucesso dos projetos. Eles são um conjunto de atividades estruturadas que visam organizar e orientar o fluxo de trabalho durante todo o ciclo de vida do desenvolvimento. Esses processos estabelecem diretrizes claras, definem responsabilidades e fornecem uma base sólida para a equipe de desenvolvimento, resultando em melhores resultados e maior satisfação do cliente.

3.2.2 DESENVOLVIMENTO DE APLICAÇÕES

O projeto utilizou diversas estruturas condicionais, funções, variáveis, métodos e operadores lógicos para aplicar o uso da lógica de programação em nosso código . Utilizando POO, aplicamos diversas classes para gerarmos um código funcional dentro do nosso PI relacionado ao restaurante Tijoleiro, cujo realiza atendimentos ao cliente, utilizando a Lógica de programação em nosso algoritmo, desenvolvemos as seguintes classes:

Atendimento: Essa classe será responsável por realizar o atendimento ao usuário e guardar as informações fornecidas por tal.

Produto: Será responsável por armazenar informações de um produto, como: valor, tamanho, descrição e se o produto está ativo ou não.

Endereco: Irá armazenar as informações de localização do usuário, como: cidade, bairro, rua, número e se está desativado ou ativado.

Finalidade: Essa classe se responsabilizará por armazenar a informação sobre o motivo do contato do cliente com a empresa.

Cliente: Essa classe vai se responsabilizar por armazenar as informações do cliente, como o nome, telefone, endereço (está relacionada a classe “Endereco”.) e se está ativo ou não.

Meio_de_contato: Irá armazenar informações sobre o meio de contato do cliente.

bd: essa classe será responsável por conectar o código ao banco de dados, onde serão armazenadas e criadas todas as informações geradas pelo código.

Menu: Essa classe irá se responsabilizar por conter toda a lógica de programação do nosso código e estruturas condicionais, onde poderá relacionar uma classe a outra, se irá registrar, alterar ou excluir uma informação de alguma classe. Segue uma imagem:

Trecho de código “Classe Menu”

```

10 class Menu:
11
12     def __init__(self, db):
13         self.db = db
14
15     def menu(self):
16         while True:
17             print("--- Menu Principal ---")
18             print("1. Endereço")
19             print("2. Cliente")
20             print("3. Forma Contato")
21             print("4. Finalidade")
22             print("5. Produtos ")
23             print("6. Atendimentos")
24             print("7. Atendimentos x Produtos.")
25             print("8. Atualizar Views (B.I)")
26             print("9. Sair")
27             opcao = input("Escolha uma opção: ")
28
29             if opcao == "1":
30                 while True:
31                     print("--- Menu Endereços ---")
32                     print("1. Cadastrar Endereço")
33                     print("2. Listar Endereço")
34                     print("3. Alterar Endereço")
35                     print("4. Desativar/Ativar Endereço")
36                     print("9. Voltar ao Menu Principal")
37                     opcao1 = input("Escolha uma opção:")
38

```

Fonte: Autores(classe menu do código do projeto tijoleiro)

3.2.3 IMPLEMENTAÇÃO E VALIDAÇÃO

Os módulos criados durante o desenvolvimento de nosso projeto, foram implementados com funções, como variáveis, manipulação de dados e interação com o banco de dados. Cada função encapsula uma operação necessária para o funcionamento do nosso projeto, tornando o código mais organizado e facilitando sua reutilização, ao mesmo tempo que faz cadastros de cada dado inserido dentro do código. Por exemplo, uma função pode ser responsável por validar os dados de entrada, enquanto outra realiza a conexão com o banco de dados para armazenar informações.

As estruturas condicionais irão trazer todas as funcionalidades do nosso projeto. Elas permitirão que o código do projeto tome decisões com base em condições específicas, garantindo que as operações sejam executadas apenas quando os dados inseridos pelo usuário são atendidos, fazendo com que nosso código siga com perfeição a lógica de programação. Todas as classes criadas desempenham suas funções, sendo todas aplicadas em nossa classe “menu.py”, fazendo com que sejamos capazes de atender o usuário como planejamos em nosso projeto.

3.3 MODELAGEM DE DADOS

A modelagem de dados é uma etapa fundamental no desenvolvimento de sistemas de informação do nosso projeto, pois fornece a estrutura e a organização dos dados que serão

utilizados. Com a forte demanda por informações, a necessidade de uma representação clara e eficaz dos dados se tornou essencial para garantir a integridade, a eficiência e a acessibilidade das informações.

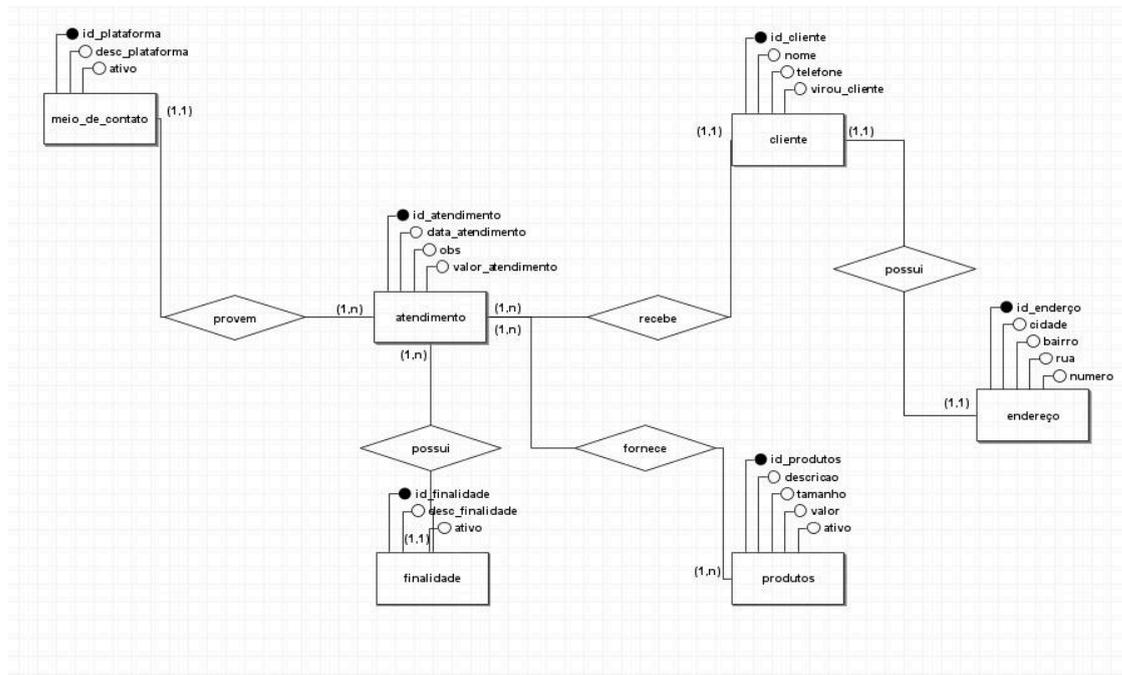
Neste contexto, a modelagem de dados se torna de extrema importância para o desenvolvimento do projeto, pois a partir dela serão criados os modelos conceitual, lógico e físico do sistema: O **modelo conceitual** oferece uma visão ampla dos dados, definindo os principais elementos e suas relações, sem incluir detalhes técnicos, permitindo uma compreensão inicial do sistema. Em seguida, o **modelo lógico** traduz essa visão geral em um formato mais técnico, detalhando atributos e relacionamentos para estabelecer como os dados se interconectam. Por fim, o **modelo físico** define como esses dados serão armazenados fisicamente no banco de dados. Segundo o site: aws.amazon.com/pt/what-is/data-modeling/

Modelagem de dados é o processo de criar uma representação visual, ou esquema, que define os sistemas de coleta e gerenciamento de informações de qualquer organização. Esse blueprint ou modelo de dados ajuda diferentes partes interessadas, como analistas de dados, cientistas e engenheiros, a criar uma visão unificada dos dados da organização. O modelo descreve quais dados a empresa coleta, a relação entre diferentes conjuntos de dados e os métodos que serão usados para armazenar e analisar esses dados.

3.3.1 MODELO CONCEITUAL

Para atender as demandas do projeto começamos a pensar em como desenvolver o modelo conceitual, suas entidades e relacionamentos, que seriam o primeiro passo para a construção do modelo lógico, dentro dele começamos a partir das entidades que são as partes principais do modelo conceitual que fizemos com base no que o projeto necessitava, logo após partimos para a criação dos relacionamentos que interligam as entidades, em seguida criamos os atributos das entidades e ajustamos as cardinalidades entre os relacionamentos para que tudo ficasse de acordo com o projeto. Segue uma imagem:

Modelo Conceitual



Fonte: Autores(modelo conceitual do projeto tijoleiro)

O modelo conceitual conta com 6 entidades que o compõem, cada entidade tem grande importância sendo atendimento a entidade central, ela que irá receber a maior parte das informações das outras entidades, como por exemplo id plataforma que recebe os meios de contato, id clientes que liga os clientes e seus endereços ao atendimento, finalidade que seria o porque um cliente entrou em contato para delivery ou reserva do restaurante e uma das mais importantes a entidade produto, que tem ligação ao atendimento ela que mostrará que produtos foram enviados a determinado cliente de um ou mais atendimentos, todas são muito importantes e de certa forma tem conexão com a entidade atendimentos. Segundo o site: aws.amazon.com/pt/what-is/data-modeling/

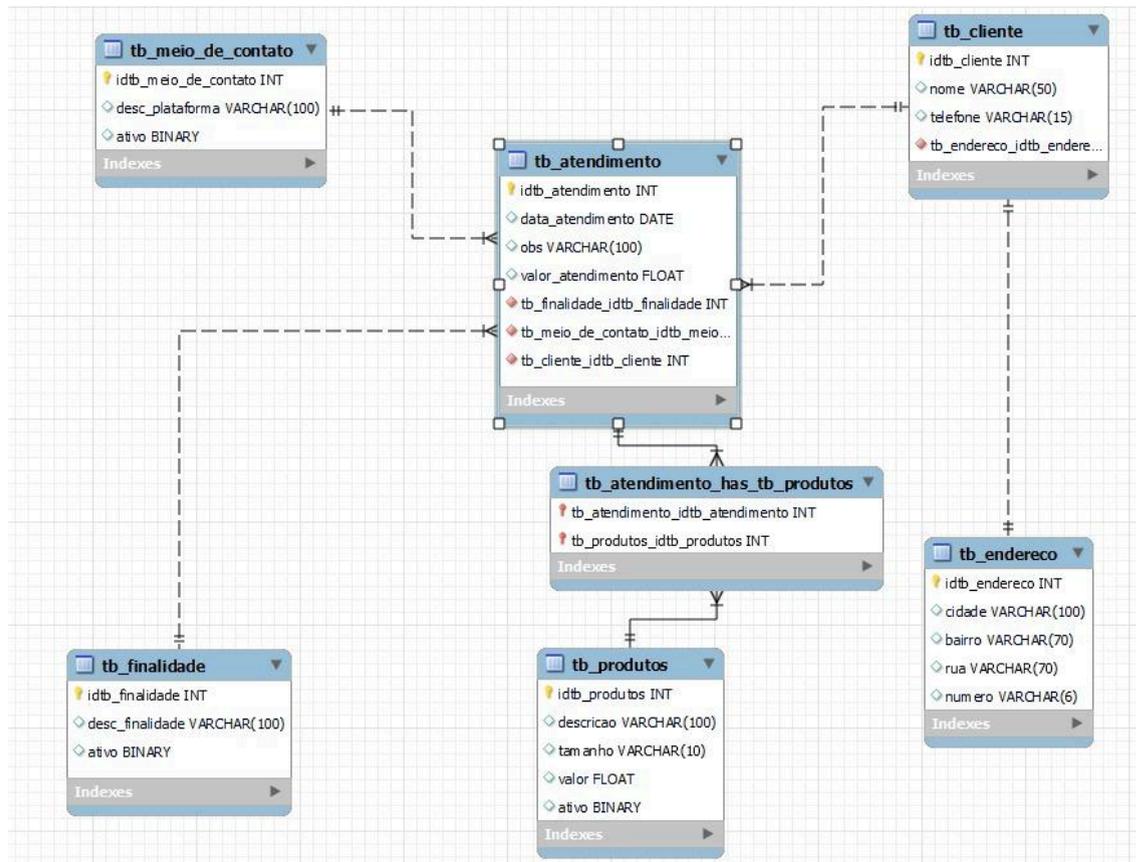
As partes interessadas e os analistas de negócios normalmente criam o modelo conceitual. Trata-se de uma representação diagramática simples que não segue as regras formais de modelagem de dados. O que importa é que ele ajuda as partes interessadas técnicas e não técnicas a compartilhar uma visão comum e concordar com a finalidade, o escopo e o design de seus projetos de dados.

3.3.2 MODELO LÓGICO E FÍSICO

A partir do modelo conceitual começamos a criação do lógico, que usa as entidades que criamos, seus atributos e relacionamentos para criar tabelas, dentro disso, precisamos construir corretamente as funções das tabelas, como por exemplo colocar identificadores

como chaves primárias, e deixar cada atributo com a tipagem correta, se um atributo necessitasse de texto usávamos varchar, e se fossem somente numéricas utilizamos int ou float e para ativo e não ativo usamos binário/binary, logo após a criação do modelo lógico usamos o próprio programa para transformá-lo no modelo físico utilizando o forward engineer para a criação do modelo lógico. Segue uma imagem sobre o modelo lógico:

Modelo Lógico



Fonte: Autores(modelo lógico do projeto tijoleiro)

O modelo lógico como descrito anteriormente usa as entidades, atributos e relacionamentos do modelo conceitual para gerar tabelas e seus relacionamentos, com exceção da tabela tb_atendimento_has_tb_produtos que usa a id dos produtos e a dos atendimentos para registrar o que o cliente pediu, essa é a única tabela que não foi gerada a partir de uma entidade e sim a partir de um relacionamento de muitos para muitos ou seja n para n.

3.3.3 SQL

Com o banco de dados pronto começamos a popular as tabelas a partir da tabela produtos usando o comando insert: insert into tb_produtos (descricao,tamanho,valor,ativo) values ('MARMITA','G','25',1), fizemos isso para todos os produtos necessários na tabela, passando para a tabela meio de contatos, que também utilizamos o comando insert mas com os dados necessários para aquela tabela, fizemos isso também para as tabelas: finalidade, endereco, clientes, atendimento e atendimento has produtos que conecta produtos a atendimentos, tivemos alguns problemas que foram resolvidos utilizando os comandos update e alter. Segue alguns comandos que usamos e uma breve explicação:

"INSERT INTO tb_endereco (CIDADE, BAIRRO, RUA, NUMERO) VALUES (%s, %s, %s, %s);" o insert serve para inserir dados nas tabelas, porém esse insere mais especificamente na tb_endereco, que salva os endereços dos clientes.

"SELECT idtb_endereco, cidade, bairro, rua, numero FROM tb_endereco where ativo = 1;"

O comando select serve para selecionar e mostrar os valores descritos nele com a condição ativo, ou seja mostrará apenas os campos ativos da tb_endereco.

"update tb_endereco set cidade = %s, bairro = %s, rua = %s, numero = %s where idtb_endereco = %s;" o comando update serve para atualizar os valores e campos de uma tabela, como na de exemplo que é a endereço, ele pode ser usado em caso de erro em algum dado ou dados antigos que precisam ser atualizados.

3.4 BUSINESS INTELLIGENCE

Nesta parte do Projeto Integrado, o objetivo é desenvolver dashboards para o Restaurante Tijoleiro, utilizando práticas de Business Intelligence (BI) para fornecer uma visão sobre os atendimentos e o uso do sistema. O dashboard permite que dados sejam analisados de maneira visual, ajudando na compreensão de operações do restaurante. Segundo o site da Microsoft PowerBI:

Uma estratégia de BI é um plano para implementar, usar e gerenciar dados e análises. Você define sua estratégia de BI começando com um planejamento estratégico de BI. O planejamento estratégico ajuda você a identificar as áreas de foco e objetivos de BI. Para determinar o caminho para progredir em direção aos seus objetivos de BI, você descreve os principais resultados específicos usando o planejamento tático.

3.4.1 ORGANIZAÇÃO E IDENTIFICAÇÃO DAS INFORMAÇÕES

Para estruturar o dashboard, seguimos o modelo de análise SWOT, uma ferramenta estratégica que auxilia na organização. Segundo Sherman (2014), “identificar dados relevantes é essencial para que o BI agregue valor, pois facilita a leitura e análise das operações empresariais” (Business Intelligence Guidebook). As informações mapeadas incluem: Volume e frequência de atendimentos. Produtos mais vendidos e recorrência de pedidos. Preferências de contato e fidelização dos clientes.

3.4.2 MANIPULAÇÃO E ANÁLISE DE DADOS

Na organização dos dados coletados do sistema SQL , aplicamos "Data Cleaning" limpeza de dados e "Data Transformation" para assegurar a qualidade dos dados e prepará-los para análise. Segundo o guia “Análise de Dados para Leigos” de Michael Alexander e John Walkenbach (2020): “limpar e transformar dados são etapas essenciais para garantir que eles estejam corretos e úteis para análise” (Análise de Dados para Leigos).

3.4.3 CRIAÇÃO DE MODELOS DE ANÁLISE DE DADOS

Para a construção do dashboard, utilizamos o Power BI e aplicamos princípios de visualização de dados. De acordo com o site Awari: “a visualização de dados é essencial para transformar informações complexas em insights claros, ajudando a alcançar os objetivos de forma mais eficaz” (Awari). Com isso, gráficos, tabelas e segmentações interativas no dashboard permitem ao restaurante Tijoleiro identificar padrões de consumo e desempenho, facilitando a tomada de decisões estratégicas.

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS

A gestão financeira é uma habilidade essencial para a vida cotidiana, permitindo que indivíduos tomem decisões mais conscientes e responsáveis sobre seu dinheiro. Esta parte do projeto tem como objetivo abordar a importância de desenvolver competências financeiras, fornecendo ferramentas práticas que ajudem as pessoas a lidar com suas finanças pessoais de

maneira eficiente e sustentável. Ao aprender a administrar suas finanças, as pessoas se tornam mais capacitadas para enfrentar desafios econômicos, alcançar seus objetivos e promover um futuro financeiro mais estável.

3.5.1 GERENCIANDO FINANÇAS

- **Tópico 1:** Inicialmente quando falamos deste tema, dinheiro é o que vem a nossa mente e ele é o principal agente dessa equação, este tema também tem forte relação com as disciplinas Administração, Contabilidade e Economia. Então se você busca ou pretende se tornar um expert na área são esses temas que você deve buscar.

Resumidamente conseguimos separar a Economia em dois principais eixos: Macroeconomia e Microeconomia.

Microeconomia: seria o estudo relacionado a determinado produto ou grupo de produtos e dos compradores e vendedores de tais produtos.

Macroeconomia: Estuda o funcionamento da economia como um todo, busca estudar e medir as variáveis de volume de produção, emprego e preços. Contabilidade: traz para a empresa uma visão clara da situação financeira, organizando seus recursos e ajudando na tomada de decisões. Com essas informações em mente conseguimos classificar nossos gastos em: despesas, custos, desembolsos e investimentos e com isso saber tomar as melhores decisões sem colocar em risco a nossa saúde financeira.

- **Tópico 2:** Independência financeira é quando você tem fontes de renda suficientes para cobrir seus gastos sem depender do trabalho ativo. Para isso, você precisa de uma gestão financeira inteligente. Isso inclui gerar renda, controlar despesas, investir com sabedoria e buscar otimização constante dos seus recursos.

Para registrar e controlar as finanças no dia a dia:

1. Registre todas as entradas e saídas de dinheiro.
2. Monitore sua liquidez – o quanto sobra após pagar suas despesas.
3. Controle os custos, renegociando fornecedores e evitando dívidas.
4. Invista o excedente em aplicações que se alinhem ao seu perfil (conservador, moderado ou agressivo).
5. Gere relatórios periódicos para revisar suas finanças e ajustar seu plano.

- **Tópico 3:** Você já se perguntou como os juros podem transformar pequenas dívidas em grandes problemas? Os juros simples são calculados apenas sobre o valor inicial da dívida.

Mas os juros compostos são calculados sobre o saldo total, incluindo os juros anteriores. Isso faz com que a dívida cresça muito mais rápido.

Se você está endividado, existem algumas opções. A primeira é negociar com o credor, tentando reduzir o valor ou os juros. Outra opção é consolidar suas dívidas, ou seja, trocar todas as dívidas que você possui por uma só.

- **Tópico 4:** Para estabelecer metas financeiras, é importante definir objetivos específicos, com prazos claros e dividi-los em etapas menores. Priorize metas de curto, médio e longo prazo, e inclua sua família e círculo de amigos nos planos para garantir o alinhamento das prioridades em seu percurso.

Em relação aos sonhos de longo prazo, como comprar uma casa ou garantir uma aposentadoria tranquila, o planejamento e o investimento regular são essenciais. Criar uma reserva de emergência é crucial para lidar com imprevistos sem comprometer os projetos de longo prazo. A chave para o sucesso está na disciplina e no ajuste contínuo dos planos.

3.5.2 ESTUDANTES NA PRÁTICA

Para criar o vídeo, o roteiro foi dividido em 6 partes a fim de organizar melhor o vídeo e as ideias inseridas nele. Essas partes começam com a introdução onde o tema do vídeo é apresentado, seguindo para o tópico 1 falando sobre conceitos econômicos e financeiros básicos, após esse seguimos com tópico 2 que fala sobre independência financeira e gestão de riqueza, continuando ao tópico 3 onde apresentamos sobre dívidas, juros compostos e alternativas para o endividado, logo em seguida falamos sobre o tópico 4 que conta com o estabelecimento de metas e realização de sonhos, então acabamos com a finalização do vídeo agradecendo aos envolvidos e a todos os professores.

4. CONCLUSÃO

Em conclusão, nosso projeto alcançou seus objetivos principais, criando uma estrutura organizada e funcional para o projeto da criação do sistema de console de coleta e análise de dados sobre os clientes de um restaurante. Os resultados obtidos refletem o compromisso com as metas estabelecidas, proporcionando uma base sólida e bem estruturada para otimizar e melhorar a coleta de dados.

No decorrer do desenvolvimento, foram encontradas algumas dificuldades, como os donos do restaurante não saberem direito o que queriam e sempre mudarem alguma coisa no projeto, tivemos algumas dificuldades em relação a criação de algumas partes do projeto de forma integral. Esses desafios exigiram soluções criativas e uma adaptação contínua ao longo do processo.

Apesar das dificuldades, o projeto ofereceu uma experiência valiosa, permitindo à equipe aprofundar conhecimentos em BI, POO, modelagem de dados e lógica de programação, isso nos ajudou a desenvolver habilidades práticas que serão úteis em futuros projetos acadêmicos e profissionais.

REFERÊNCIAS

ALURA.com.br **Algoritmos e lógica de programação: O que são e qual a importância:** [algoritmos e lógica de programação](#)

AWARI.com.br/ **A Importância da Visualização de Dados para Alcançar Seus Objetivos:** [A importância da visualização de dados objetivo](#)

AWARI.com.br/ **Vantagens processos de desenvolvimento de software:** [Vantagens processos de desenvolvimento de software](#)

AWS.amazon.com/pt **O que é modelagem de dados: o que é modelagem de dados?**

devmedia.com.br **POO: Os 4 pilares da Programação Orientada a Objetos:** [os 4 pilares da programação orientada a objetos](#)

LEARN.microsoft.com/pt **Planejamento de implementação do Power BI:** [Planejamento de solução de BI](#)

LOCAWEB.com.br **O que é encapsulamento na programação orientada a objetos:** [o que é encapsulamento na programação orientada a objetos](#)

NOVA.escolalinux.com.br **Programação orientada a objetos - quando utilizar esse recurso:**[programação orientada a objetos](#)

SHERMAN, Rick. Business Intelligence Guidebook - From Data Integration Analytics (2014) [Livro](#)

WALKENBACH, John. Análise de dados para leigos - 2ª Edição (2020) [Livro](#)

ANEXOS

Link vídeo formação para vida: <https://youtu.be/CXrQL6E3VUk>