

Trabalho de Bancos de Dados II:
Express Food

Bruno Alves Catão Silva - 7553278
Cayo Felipe Barbosa Soares - 13719520
Guilherme Nogueira Pereira - 12543096
Luigi Tramontin - 12727987
Thomas Delfs - 13837175

Repositório: https://github.com/bcatao92/sistema_pedidos

Fundamentação Teórica e Justificativa: Sistema de Banco de Dados para Aplicativo de Delivery

O mercado de delivery no Brasil e no mundo está em constante evolução, apresentando desafios técnicos e oportunidades de inovação que fazem deste setor um caso de estudo ideal para a aplicação de sistemas de banco de dados relacionais. Este documento apresenta uma fundamentação teórica abrangente sobre o tema, destacando dados de mercado, desafios operacionais e a importância dos sistemas de banco de dados para aplicativos de delivery.

1. Panorama do Mercado de Delivery: Crescimento e Tendências

1.1 Cenário Global

O mercado global de delivery de comida online alcançou USD 147,9 bilhões em 2024 e projeta-se que atinja USD 329,4 bilhões até 2033, demonstrando uma taxa de crescimento anual composta (CAGR) de 9,3% durante o período de 2025-2033[1]. Este crescimento exponencial é impulsionado pela ênfase crescente na conveniência do consumidor, pelo aumento do uso de smartphones e acessibilidade à internet, pela rápida urbanização global, pelo crescente número de famílias com dupla fonte de renda e pela integração de recursos avançados e programas de fidelidade nas plataformas[1].

1.2 Cenário Brasileiro

No Brasil, o mercado de delivery também apresenta números expressivos. A receita no setor de delivery online está projetada para alcançar US\$ 20,91 bilhões em 2025, com expectativa de crescimento anual de 7,12% entre 2025-2030[2]. Segundo dados da Associação Brasileira de Bares e Restaurantes (Abrasel), o segmento cresceu entre 7,5% e 8% em 2023, atingindo um valor de mercado de R\$ 38 bilhões[3].

O iFood, principal plataforma do setor no país, encerrou 2024 com 380 mil estabelecimentos parceiros, consolidando o hábito de pedir comida online e alcançando 80,8% do tráfego dos aplicativos de delivery no Brasil em setembro daquele ano[4]. Entre as categorias em crescimento, a marmita liderou com alta de 18% no número de pedidos, seguida por itens de padaria (11%) e açaí (9%)[4].

1.3 Tendências Tecnológicas e Modelos de Negócio

1.3.1 Dark Kitchens

Um modelo de negócio em ascensão é o conceito de "dark kitchen" (cozinha fantasma), que consiste em estabelecimentos de serviço de alimentação que oferecem apenas comida para viagem, sem sala de jantar. A principal vantagem é a redução de custos operacionais, eliminando necessidade de garçons e outras despesas relacionadas ao salão[5]. Esse formato permite que esses restaurantes operem em áreas com aluguéis mais baixos e espaços menores, além da possibilidade de compartilhamento de cozinhas entre diferentes marcas de restaurantes[5].

1.3.2 Inovações em Logística e Entrega

Empresas como o iFood têm investido em tecnologia e inovação para enfrentar desafios logísticos. Em 2024, a plataforma iniciou testes com drones para complementar suas entregas, visando reduzir o tempo de deslocamento e aumentar a eficiência. Em Campinas (SP), os drones reduziram o tempo de entrega de 12 para 3 minutos em determinados trajetos[6].

1.3.3 Personalização da Experiência

A experiência personalizada pós-compra tem se destacado como diferencial competitivo. Fornecer múltiplas opções de entrega, gerenciar expectativas com transparência e oferecer diferentes métodos de entrega são práticas que aumentam a satisfação do cliente e transformam consumidores ocasionais em clientes recorrentes[7].

2. Desafios Operacionais e Logísticos dos Aplicativos de Delivery

2.1 Problemas de Experiência do Usuário

Os aplicativos de delivery enfrentam diversos desafios que impactam diretamente a experiência do cliente e a imagem da marca. Os três problemas-chave mais reportados pelos consumidores são[8]:

1. **Atrasos nas entregas:** Quando o prazo estimado não é cumprido, a expectativa do cliente se transforma em frustração.
2. **Erros nos pedidos:** Recebimento de itens incorretos ou faltantes, o que prejudica a confiança do cliente na capacidade do restaurante em atender corretamente.
3. **Falta de informações:** A ausência de atualizações sobre o andamento da entrega gera ansiedade no consumidor, que deseja acompanhar em tempo real o status do seu pedido.

2.2 Desafios de Mercado e Competitividade

O mercado brasileiro de food delivery apresenta alta concentração, o que cria um cenário desafiador para empreendedores que desejam atuar nesse setor[3]. A falta de concorrência pode levar à formação de monopólios, o que traz uma série de riscos para lojistas, consumidores e para a economia como um todo.

Recentemente, medidas regulatórias têm sido implementadas para impedir práticas que limitam a concorrência, como contratos de exclusividade entre plataformas de entrega e grandes redes de restaurantes[3].

2.3 Segurança e Violência Urbana

A segurança dos entregadores também representa um desafio significativo. Em 2024, o iFood registrou mais de 13 mil denúncias de violência contra seus entregadores, totalizando 13.576 ocorrências[6]. Somente nos primeiros seis dias de março daquele ano, já haviam sido registradas 810 ocorrências do tipo, evidenciando os riscos enfrentados pelos profissionais do setor.

3. Sistemas de Delivery como Caso de Estudo para Banco de Dados Relacionais

3.1 Complexidade e Inter-relação das Entidades

Os sistemas de delivery envolvem múltiplas entidades interconectadas, como clientes, restaurantes, entregadores, pedidos e produtos. Esta estrutura complexa se beneficia diretamente do modelo relacional de dados, que permite organizar informações em tabelas, filas e colunas, estabelecendo vínculos entre diferentes conjuntos de dados[9].

Por exemplo, quando um aplicativo de processamento de pedidos envia uma solicitação ao banco de dados, o sistema pode acessar a tabela de pedidos do cliente, obter as informações corretas sobre o produto solicitado e usar o ID do cliente para buscar informações de faturamento e entrega na tabela de informações do cliente[9].

3.2 Integridade de Dados e Controle de Redundância

A integridade da relação é fundamental para garantir a exatidão e consistência dos dados em um banco de dados relacional[10]. O modelo exige que cada tabela tenha um identificador único (chave primária), garantindo que cada linha seja identificável pelo seu conteúdo.

O modelo relacional é particularmente eficaz em manter a consistência de dados entre aplicações e cópias de banco de dados. Quando um cliente faz um pedido através de um aplicativo de delivery, ele espera que essa informação seja refletida imediatamente em todas as instâncias do sistema, desde o restaurante até o entregador[9].

3.3 Controle de Concorrência e Acesso Simultâneo

Em um sistema de delivery, múltiplos usuários acessam e modificam os dados simultaneamente: clientes fazem pedidos, restaurantes atualizam status, entregadores marcam entregas concluídas. O controle de concorrência visa gerenciar esse acesso a recursos compartilhados, controlando como múltiplos acessos podem utilizar um recurso sem conflitos que poderiam gerar inconsistência dos dados[11].

Existem diferentes técnicas para gerenciar a concorrência, como monitores, locks e semáforos, que podem ser implementados para garantir que as operações de um sistema de delivery ocorram de forma ordenada e segura, preservando a integridade dos dados[11].

3.4 Monitoramento em Tempo Real e Integração com APIs Externas

3.4.1 Geolocalização

A integração com APIs de geolocalização, como as do Google Maps, é essencial para sistemas de delivery. Estas APIs permitem implementar serviços baseados em localização, fornecendo

rotas precisas para entregadores, localizando rapidamente endereços de clientes e otimizando trajetos para reduzir custos e deslocamentos desnecessários[12].

Com a Directions API, por exemplo, é possível calcular a distância mais curta até o local de entrega e fornecer ao entregador direcionamentos passo a passo, orientações sobre limites de velocidade e notificações sobre condições de trânsito em tempo real[12].

3.4.2 Sistemas de Pagamento

As APIs de pagamento funcionam como infraestrutura crítica que conecta empresas a redes financeiras, permitindo transações monetárias sem fricção[13]. Plataformas como PayPal, Square e Amazon Pay oferecem diferentes vantagens para comerciantes de delivery, como checkout expresso, conformidade global, pagamentos adaptativos e opções de faturamento recorrente[13].

A integração destas APIs com o banco de dados do sistema de delivery exige um design cuidadoso para garantir a segurança e consistência das transações financeiras.

4. Justificativa para Escolha do Tema

4.1 Relevância Econômica e Social

O impacto econômico do setor de delivery no Brasil é considerável. Apenas o iFood gerou efeito equivalente a 0,55% do Produto Interno Bruto (PIB) nacional em 2023, criando mais de 900 mil postos de trabalho e movimentando R\$ 110,7 bilhões em atividades econômicas que abrangem diversos setores produtivos[14].

O estudo da Fundação Instituto de Pesquisas Econômicas (Fipe) revelou também que restaurantes pequenos que ingressaram na plataforma geraram, em média, 10,2% mais empregos do que negócios do mesmo porte que não utilizam o aplicativo[14]. Há ainda um efeito multiplicador: a cada R\$ 1.000,00 gastos em restaurantes através do aplicativo, outros R\$ 1.390,00 são injetados na economia brasileira[14].

4.2 Oportunidade para Aplicação de Conceitos de Banco de Dados

O desenvolvimento de um sistema de delivery oferece uma excelente oportunidade para aplicar conceitos avançados de banco de dados, como:

1. **Modelagem conceitual e lógica complexa:** representando entidades interconectadas como clientes, restaurantes, produtos, pedidos e entregadores.
2. **Implementação de mecanismos de controle de concorrência:** garantindo que múltiplos acessos simultâneos não comprometam a integridade dos dados.
3. **Otimização de consultas críticas:** melhorando o desempenho de operações frequentes como busca de restaurantes próximos, cálculo de tempo de entrega e processamento de pagamentos.
4. **Implementação de estratégias de indexação e particionamento:** organizando os dados para acesso eficiente considerando o volume de transações.
5. **Integração com APIs externas:** consolidando dados de diferentes fontes como geolocalização, pagamentos e notificações.

4.3 Potencial de Inovação Tecnológica

O projeto de um banco de dados para aplicativo de delivery oferece amplo espaço para inovação, incluindo:

1. **Sistemas de recomendação personalizados:** utilizando histórico de pedidos e preferências para sugerir novos produtos.
2. **Otimização logística em tempo real:** roteamento dinâmico baseado em condições de trânsito e disponibilidade de entregadores.
3. **Implementação de dark kitchens virtuais:** gerenciando múltiplas marcas operando na mesma cozinha física.
4. **Rastreamento em tempo real:** fornecendo atualizações contínuas sobre o status do pedido.
5. **Integração omnichannel:** unificando dados de diferentes canais de pedido (app, web, telefone) em uma base consistente.

5. Conclusão

O desenvolvimento de um sistema de banco de dados para aplicativo de delivery representa uma oportunidade singular para aplicar conceitos avançados de modelagem de dados, controle de concorrência e otimização de desempenho em um contexto de relevância econômica e social crescente. O setor de delivery, em constante expansão e transformação, apresenta desafios técnicos que podem ser endereçados através de um projeto bem estruturado de banco de dados, contribuindo para melhorar a experiência do usuário e a eficiência operacional dos serviços.

A implementação deste projeto permitirá não apenas demonstrar o domínio de conceitos fundamentais de banco de dados, mas também explorar soluções inovadoras para problemas práticos do mundo real, alinhando-se perfeitamente aos objetivos da disciplina de Laboratório de Banco de Dados.

Modelagem Entidade Relacionamento

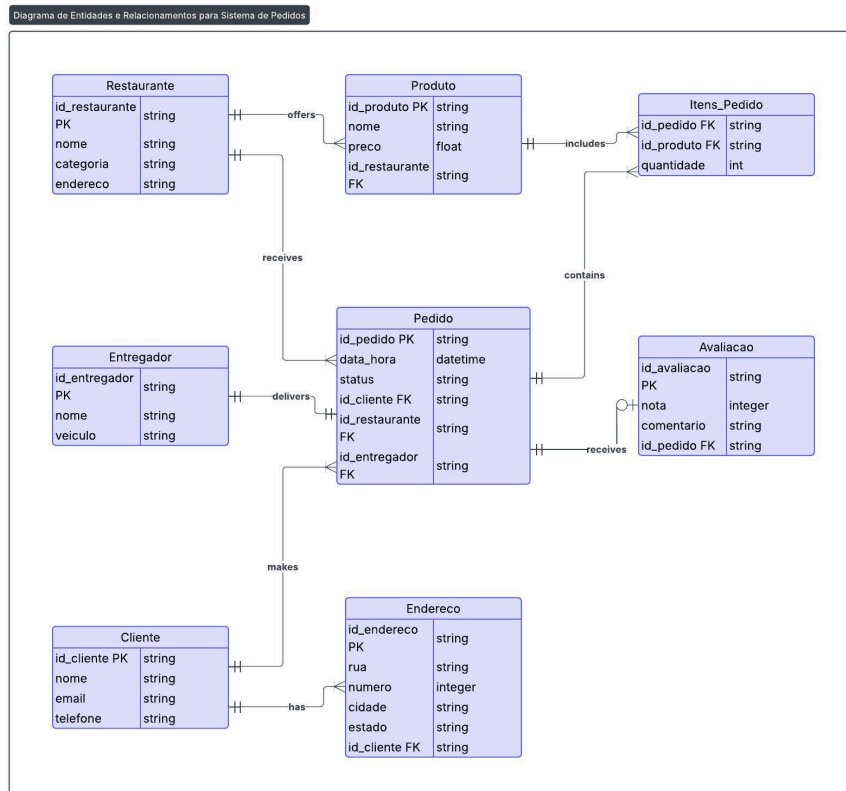
Estrutura de Tabelas:

- Cliente: id_cliente (PK), nome, email, telefone.
- Restaurante: id_restaurante (PK), nome, categoria, endereco.
- Entregador: id_entregador (PK), nome, veiculo.
- Produto: id_produto (PK), nome, preco, id_restaurante (FK).
- Pedido: id_pedido (PK), data_hora (datetime), status, id_cliente (FK), id_restaurante (FK), id_entregador (FK).
- Itens_Pedido: id_item (PK), id_pedido (FK), id_produto (FK), quantidade.
- Avaliação: id_avaliacao (PK), nota, comentario, id_pedido (FK), tipo_avaliacao (cliente/entregador).
- Endereço: id_endereco (PK), rua, numero, cidade, estado, complemento, id_cliente (FK).

Relacionamentos:

- Cliente - Pedido: 1:N (Um cliente pode realizar vários pedidos).

- Restaurante - Pedido: 1:N (Um restaurante pode receber vários pedidos).
- Pedido - Itens_Pedido: 1:N (Um pedido contém múltiplos itens).
- Produto - Itens_Pedido: 1:N (Um produto pode estar presente em múltiplos pedidos).
- Pedido - Avaliação: 1:N (Um pedido pode receber múltiplas avaliações).
- Cliente - Endereço: 1:N (Um cliente pode ter múltiplos endereços).
- Entregador - Pedido: 1:N (Um entregador pode entregar vários pedidos).



Modelo Lógico

1. Cliente

```
CREATE TABLE Cliente (  
    id_cliente VARCHAR(36) PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    telefone VARCHAR(15)  
);
```

2. Endereço

```
CREATE TABLE Endereco (  
    id_endereco VARCHAR(36) PRIMARY KEY,  
    rua VARCHAR(100) NOT NULL,  
    numero INT NOT NULL,  
    cidade VARCHAR(50) NOT NULL,  
    estado VARCHAR(50) NOT NULL,  
    complemento VARCHAR(100),  
    id_cliente VARCHAR(36) NOT NULL,  
    FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)  
);
```

3. Restaurante

```
CREATE TABLE Restaurante (  
    id_restaurante VARCHAR(36) PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    categoria VARCHAR(50),  
    endereco VARCHAR(150)  
);
```

4. Produto

```
CREATE TABLE Produto (  
    id_produto VARCHAR(36) PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    preco DECIMAL(10, 2) NOT NULL,  
    id_restaurante VARCHAR(36) NOT NULL,  
    FOREIGN KEY (id_restaurante) REFERENCES Restaurante(id_restaurante)  
);
```

5. Entregador

```
CREATE TABLE Entregador (  
    id_entregador VARCHAR(36) PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    veiculo VARCHAR(50)
```

);

6. Pedido

```
CREATE TABLE Pedido (  
    id_pedido VARCHAR(36) PRIMARY KEY,  
    data_hora DATETIME NOT NULL,  
    status VARCHAR(20) NOT NULL,  
    id_cliente VARCHAR(36) NOT NULL,  
    id_restaurante VARCHAR(36) NOT NULL,  
    id_entregador VARCHAR(36),  
    FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente),  
    FOREIGN KEY (id_restaurante) REFERENCES Restaurante(id_restaurante),  
    FOREIGN KEY (id_entregador) REFERENCES Entregador(id_entregador)  
);
```

7. Itens_Pedido

```
CREATE TABLE Itens_Pedido (  
    id_item VARCHAR(36) PRIMARY KEY,  
    id_pedido VARCHAR(36) NOT NULL,  
    id_produto VARCHAR(36) NOT NULL,  
    quantidade INT NOT NULL,  
    FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido),  
    FOREIGN KEY (id_produto) REFERENCES Produto(id_produto)  
);
```

8. Avaliação

```
CREATE TABLE Avaliacao (  
    id_avaliacao VARCHAR(36) PRIMARY KEY,  
    nota INT CHECK (nota BETWEEN 1 AND 5),  
    comentario TEXT,  
    id_pedido VARCHAR(36) NOT NULL,  
    FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido)  
);
```

Considerações e Ajustes Realizados:

- As chaves primárias foram padronizadas como **VARCHAR(36)** para permitir o uso de UUIDs.
- As chaves estrangeiras foram definidas em cada tabela para garantir a integridade referencial.

- A tabela **Itens_Pedido** foi criada para representar a relação N:M entre **Pedido** e **Produto**.
- A tabela **Avaliacao** foi ajustada para vincular avaliações aos pedidos, permitindo múltiplas avaliações por pedido.

Script utilizado para popular as tabelas:

-- Habilitar extensão para UUID

```
CREATE EXTENSION IF NOT EXISTS "uuid-oss";
```

-- 1. Inserir 100 Clientes

-- 1. Inserir 100 Clientes com telefone ajustado para até 15 caracteres

```
INSERT INTO Cliente (id_cliente, nome, email, telefone)
```

```
SELECT
```

```
    uuid_generate_v4(),
```

```
    'Cliente ' || g,
```

```
    'cliente' || g || '@exemplo.com',
```

```
    '(' || (10 + (g % 90)) || ')' || '9' || lpad(((900 + (g * 37) % 1000)::text), 3, '0') || '-' || lpad(((1000 + (g * 73) % 9000)::text), 4, '0')
```

```
FROM generate_series(1,100) AS s(g);
```

-- 2. Inserir 100 Endereços (cada um vinculado a um cliente)

```
INSERT INTO Endereco (id_endereco, rua, numero, cidade, estado, complemento,
```

```
id_cliente)
```

```
SELECT
```

```
    uuid_generate_v4(),
```

```
    'Rua ' || chr(65 + (g % 26)),
```

```
    1 + (g * 3) % 500,
```

```
    (ARRAY['São Paulo','Rio de Janeiro','Belo Horizonte','Curitiba','Porto Alegre'])[1 + (g % 5)],
```

```
    (ARRAY['SP','RJ','MG','PR','RS'])[1 + (g % 5)],
```

```
    CASE WHEN g % 3 = 0 THEN 'Apto ' || (100 + g) ELSE 'Casa' END,
```

```
    c.id_cliente
```

```
FROM generate_series(1,100) AS s(g)
```

```
JOIN Cliente c ON c.email = 'cliente' || g || '@exemplo.com';
```

-- 3. Inserir 100 Restaurantes

```
INSERT INTO Restaurante (id_restaurante, nome, categoria, endereco)
```

```
SELECT
```

```
    uuid_generate_v4(),
```

```
    'Restaurante ' || g,
```

```
    (ARRAY['Italiana','Chinesa','Brasileira','Mexicana','Japonesa'])[1 + (g % 5)],
```

```
    'Rua ' || chr(65 + (g % 26)) || ' ' || (1 + (g * 7) % 500)
```

```
FROM generate_series(1,100) AS s(g);
```

-- 4. Inserir 100 Produtos (vinculados a restaurantes)

INSERT INTO Produto (id_produto, nome, preco, id_restaurante)

SELECT

 uuid_generate_v4(),
 'Produto ' || g,
 round((random() * 95 + 5)::numeric, 2),
 r.id_restaurante

FROM generate_series(1,100) AS s(g)

JOIN (

 SELECT id_restaurante, row_number() OVER () AS rn FROM Restaurante
) r ON r.rn = g;

-- 5. Inserir 100 Entregadores

INSERT INTO Entregador (id_entregador, nome, veiculo)

SELECT

 uuid_generate_v4(),
 'Entregador ' || g,
 (ARRAY['Moto','Carro','Bicicleta','Patinete'])[1 + (g % 4)]

FROM generate_series(1,100) AS s(g);

-- 6. Inserir 100 Pedidos (vinculados a cliente, restaurante e entregador)

INSERT INTO Pedido (id_pedido, data_hora, status, id_cliente, id_restaurante,
id_entregador)

SELECT

 uuid_generate_v4(),
 NOW() - (interval '1 day' * (random() * 30)),
 (ARRAY['Pendente','Em andamento','Entregue','Cancelado'])[1 + (g % 4)],
 c.id_cliente,
 r.id_restaurante,
 e.id_entregador

FROM generate_series(1,100) AS s(g)

JOIN (

 SELECT id_cliente, row_number() OVER () AS rn FROM Cliente
) c ON c.rn = g

JOIN (

 SELECT id_restaurante, row_number() OVER () AS rn FROM Restaurante
) r ON r.rn = g

JOIN (

 SELECT id_entregador, row_number() OVER () AS rn FROM Entregador
) e ON e.rn = g;

-- 7. Inserir 100 Itens_Pedido (vinculados a pedidos e produtos)

INSERT INTO Itens_Pedido (id_item, id_pedido, id_produto, quantidade)

SELECT

 uuid_generate_v4(),
 p.id_pedido,
 pr.id_produto,

```

1 + (g % 5)
FROM generate_series(1,100) AS s(g)
JOIN (
    SELECT id_pedido, row_number() OVER () AS rn FROM Pedido
) p ON p.rn = g
JOIN (
    SELECT id_produto, row_number() OVER () AS rn FROM Produto
) pr ON pr.rn = g;

-- 8. Inserir 100 Avaliações (vinculadas a pedidos)
INSERT INTO Avaliacao (id_avaliacao, nota, comentario, tipo_avaliacao, id_pedido)
SELECT
    uuid_generate_v4(),
    1 + floor(random() * 5)::int,
    'Comentário ' || g,
    (ARRAY['cliente','entregador'])[1 + (g % 2)],
    p.id_pedido
FROM generate_series(1,100) AS s(g)
JOIN (
    SELECT id_pedido, row_number() OVER () AS rn FROM Pedido
) p ON p.rn = g;

```

Uso de IA Generativa

Introdução e justificativa

Durante a fase inicial do projeto, foi solicitado o uso da Inteligência Artificial Generativa (IAG) para definir e organizar a divisão dos temas a serem abordados por cada integrante do grupo, considerando o tema escolhido (sistema de delivery semelhante ao iFood) e as especificações do trabalho.

Nesta primeira fase, a IA utilizada foi o ChatGPT, ao qual foi solicitada uma estrutura organizacional para a equipe com base em análise dos tópicos principais do trabalho. A estrutura resultante foi:

- Tema e Justificativa: Pesquisa sobre o contexto e motivação do projeto.
- Modelagem ER e Modelo Lógico: Estruturação da base de dados, entidades, relacionamentos e atributos.
- Implementação do Banco de Dados: Criação dos scripts SQL para a base de dados relacional.
- Backup da Base de Dados: Estruturação dos scripts de backup e recuperação dos dados.
- Visão do Uso da IAG: Relatório específico sobre o uso da IAG no projeto, considerando as etapas desenvolvidas.
- Criação dos Slides: Estruturação da apresentação final em formato .pptx

Após a divisão do trabalho entre os membros, a IA generativa foi usada para escrever sua introdução. A IA utilizada foi a Perplexity, que recebeu um extenso script contendo o tema e os objetivos definidos para o projeto, bem como instruções detalhadas do que era necessário para se obter uma boa resposta. Após alguns ajustes no texto e conferência das fontes citadas, o resultado foi utilizado.

A Perplexity também foi utilizada para criar um slide para a apresentação do tema do trabalho, com base em código LaTeX.

Segue o script utilizado:

Quero que você me ajude a desenvolver a fundamentação teórica e a justificativa para um projeto de uma matéria de banco de dados aplicado a um aplicação de delivery. Para isso, siga os passos:

- 1. Pesquise e resuma dados atuais sobre o crescimento do mercado de delivery no Brasil e no mundo, incluindo projeções de faturamento, número de usuários e tendências tecnológicas (como dark kitchens, personalização e integração de canais).*
- 2. Identifique e explique os principais desafios enfrentados por apps de delivery, tanto do ponto de vista operacional/logístico (atrasos, erros de pedidos, otimização de rotas, custos, integração de sistemas) quanto de experiência do usuário.*
- 3. Explique por que sistemas de delivery são um excelente caso de estudo para banco de dados relacionais, destacando:*
- 4. Complexidade e inter-relação das entidades (clientes, restaurantes, entregadores, pedidos, produtos)*
- 5. Importância da integridade de dados, controle de redundância e acesso simultâneo*
- 6. Necessidade de monitoramento em tempo real, rastreabilidade e integração com APIs externas (pagamento, geolocalização, notificações)*
- 7. Redija uma justificativa para a escolha do tema, relacionando a relevância econômica e social do delivery, a oportunidade de aplicar conceitos de banco de dados e o potencial de inovação tecnológica.*
- 8. Estruture a resposta em tópicos claros, com dados e exemplos, e sempre cite as fontes dos dados e conceitos utilizados.*
- 9. Use como referência sites como Statista, Kitchen Central, Univesp/Novotec, artigos científicos sobre prototipagem de apps de delivery (ex: UDESC), e estudos de impacto do iFood no Brasil.*

Leve em conta o que se espera do seguinte planejamento de entrega:

"Tema e Justificativa:

Realizar pesquisa sobre o setor de delivery, identificando problemas e oportunidades de otimização por meio do uso de banco de dados;

Definir o tema central do projeto, destacando a relevância da aplicação de um sistema de delivery e a importância da organização eficiente dos dados;

Redigir a justificativa do projeto, abordando os impactos esperados na operação do sistema de delivery;

Documentar a pesquisa em um documento Word/Google Docs para referência futura."

Considere as especificações do trabalho que estão em anexo.

Me indique também como poderia fazer apenas um slide de apresentação sobre isso.

Por fim, peço que tudo isso esteja escrito de forma coesa e formatado corretamente para que eu insira no trabalho como a parte de "Tema e Justificativa"

O resultado original da prompt pode ser visto no link:

- <https://www.perplexity.ai/search/quero-que-voce-me-ajude-a-dese-5JqwmU9UTuOogeUMXuY3Bg>

Modelagem ER e Modelo Lógico

Em relação à Modelagem ER e Modelo Lógico, por sugestão do ChatGPT, foi utilizada a plataforma LucidChart, que possui um assistente de IA para geração de diagramas. Através de descrições textuais dos elementos (entidades, atributos e relacionamentos), um modelo inicial foi criado, com algumas cardinalidades necessitando ajustes manuais por causa de erros cometidos pela IA.

Dentre os ajustes ao modelo sugerido pelo assistente da plataforma LucidChart, podemos citar:

- Inicialmente, a relação entre Cliente e Pedido foi gerada como 1:1, necessitando ser mudada para 1:N, posto que um cliente pode fazer múltiplos pedidos.
- A relação entre Restaurante e Pedido também foi ajustada para 1:N após ser encontrado um erro na sugestão da IAG.
- Foi necessária uma inclusão totalmente manual da tabela intermediária Itens_Pedido, posto que a IA não gerou a estrutura N:M como necessária.

Benefícios e Limitações do Uso da IAG:

Os benefícios encontrados no uso da IA nesta etapa do trabalho incluem a redução do tempo de desenvolvimento inicial do modelo ER, geração automatizada

das estruturas básicas e sugestões de atributos e tipos de dados baseados em padrões comuns de sistemas de delivery.

Já sobre as limitações encontradas, podemos citar os erros de cardinalidades e necessidade revisão manual, ausência de uma estrutura intermediária para relações N:M e a generalidade de sugestões, exigindo refinamento posterior.

Criação do modelo lógico

Durante a criação do modelo lógico, as IAGs utilizadas foram o ChatGPT e o Perplexity, com o GPT fundamentando a estruturação e organização dos comandos SQL utilizados para criar as tabelas e o Perplexity fornecendo o script para popular os campos da tabela.

Com base no modelo relacional original, foi pedido que a IAG gerasse um modelo lógico SQL, incluindo a definição de tabelas, atributos e relacionamentos.

Por sugestão da IA, foi utilizado o tipo UUID para a chave primária, além de NUMERIC para campos monetários e DATETIME para registros temporais, assegurando a consistência dos dados.

O script fornecido pelo ChatGPT foi rodado no pgAdmin 4, criando as tabelas iniciais para o banco de dados, chamado aqui de SISTEMA_PEDIDOS. Uma vez observado que as tabelas haviam sido criadas corretamente, partimos para o seu preenchimento.

Primeiro tentamos fazer com que o ChatGPT inserisse na tabelas dados condizentes com um .csv retirado do [Kaggle.com](https://www.kaggle.com), porém, como já havíamos estourado o limite de dados grátis desta IAG, partimos para o Perplexity, que não conseguiu abrir o arquivo no modo gratuito. Então pedimos um script que populasse as tabelas de modo coerente.

Por padrão, o Perplexity gerou um script Python que, após combinado com um script SQL, serviria para preencher as tabelas. Por limitações de tempo, pedimos que gerasse o script SQL com as nossas necessidades, gerando um total de 100 linhas para cada tabela, sabidamente um valor pequeno para um SGBD, mas suficiente para comprovar que o sistema estava funcionando.

Vantagens:

Sendo o tempo a principal limitação encontrada neste trabalho, com os membros do grupo trabalhando em diversos projetos simultâneos, pode-se dizer que a principal vantagem do uso de IAGs nesta etapa foi o tempo economizado. A estruturação das tabelas e escolha de tipo de variável é simples, mas preenchê-las seria um trabalho longo e tedioso de se fazer manualmente.

Desvantagens:

O uso de IAG tomou o lugar do que poderia ser uma experiência valiosa de estruturação do modelo lógico para os membros do grupo. Também vale apontar que a versão original do código da Perplexity veio com erros de inserção de dados no campo telefone, que precisamos corrigir através da própria IA.

Referências

[3] Delivery no Brasil: Competitividade saudável contribui para crescimento do setor. G1, 2024.

[1] Online Food Delivery Market Size, Share Global Industry Report 2033. IMARC Group, 2024.

[5] Dark Kitchen é a nova tendência no setor de food service. Scuada, 2023.

[7] Three tips to create a personalized delivery experience. Seven Senders, 2024.

[2] Online Food Delivery - Brazil. Statista Market Forecast, 2025.

[6] Setor de food delivery enfrenta desafios na logística. UAI, 2025.

[14] Impacto do iFood elevou PIB brasileiro em 0,55% em 2023. Brasil 247, 2024.

[11] O que é controle de concorrência? Stack Overflow, 2017.

[10] Integridade da relação. Wikipedia, 2018.

[12] Por que integrar as APIs Google Maps em sistemas de delivery? Geoambiente, 2025.

[13] Top 10 Melhores APIs de Pagamento em 2025. Apidog, 2025.

[8] Problemas com delivery: quais são os principais e como agir. Goomer, 2024.

[9] O que é um banco de dados relacional (RDBMS)? Oracle, 2025.

[4] Delivery cresce no Brasil com 380 mil estabelecimentos cadastrados no iFood em 2024. E-commerce Brasil, 2024.

[15] Mercado de Delivery no Brasil: O Que Esperar do Setor? Kitchen Central, 2025.

1. <https://g1.globo.com/sc/santa-catarina/especial-publicitario/delivery-much/noticia/2024/09/17/delivery-no-brasil-competitividade-saudavel-contribui-para-crescimento-do-setor.ghml>
2. <https://blog.sevensenders.com/en/3-tips-customer-experience-shipping>
3. Especificacao-do-Trabalho-de-Laboratorio.pdf
4. <https://www.oracle.com/br/database/what-is-a-relational-database/>
5. <https://www.imarcgroup.com/online-food-delivery-market>
6. <https://www.statista.com/outlook/emo/online-food-delivery/brazil>
7. <https://www.scuada.com.br/blog/dark-kitchen-e-a-nova-tendencia-no-setor-de-food-service/>

8. <https://www.statista.com/outlook/emo/online-food-delivery/meal-delivery/worldwide>
9. <https://goomer.com.br/blog/problemas-com-delivery>
10. <https://pt.stackoverflow.com/questions/211394/o-que-é-controle-de-concorrência>
11. <https://cloud.google.com/learn/what-is-a-relational-database>
12. https://pt.wikipedia.org/wiki/Integridade_da_relação
13. <https://www.geoambiente.com.br/blog/3-vantagens-de-integrar-as-apis-de-google-maps-a-sistemas-de-delivery/>
14. <https://scispace.com/pdf/desenvolvimento-de-um-sistema-produto-servico-com-base-e-m-2nwd62hw1e.pdf>
15. <https://goomer.com.br/blog/crescimento-delivery-brasil>