

Tech Challenge - FoodieFlow

Grupo 42

RM 350972: Gabriel Falsetti Ferrari Martins

RM 350698: Guilherme Marchesi Endrigo

RM 351351: Lucas Proença Renó

RM 351734: Ederson Ribeiro Paz

Repositório

<https://github.com/gabrielolferrari/FoodieFlow>

Swagger: <http://localhost:8000/docs>

Redoc: <http://localhost:8000/redocs>

Identificação do problema

O negócio fonte de estudo é uma lanchonete, cujo objetivo principal é preparar e servir alimentos e bebidas aos clientes. Atualmente, a lanchonete está passando por um processo de expansão de seus negócios e está enfrentando problemas no atendimento, que estão causando atrasos e erros na entrega, gerando um cenário caótico e confuso.

Atuação de negócios

Com o objetivo de resolver a ineficiência encontrada no atendimento atual, foi solicitado um sistema de controle de pedidos. Através dele, deve ser possível o autoatendimento para realização do pedido.

Para dar clareza ao enfoque do software a ser desenvolvido, o grupo buscou realizar um estudo com o DDD e assim entender quais os domínios de atuação do negocio a solução irá resolver.

Identificação dos domínios do negocio

▼ Subdomínios principais:

- Gestão de alimentos e bebidas
- Serviço ao Cliente

▼ Subdomínios de suporte:

- Marketing e promoção
- Finanças e contabilidade
- Gestão de clientes (CRM)
- Gestão de funcionários (RH)
- Relações com fornecedores (compras)

▼ Subdomínios de Genérico:

- Meios de Pagamento
- Entrega

Uma vez entendido os domínios da lanchonete, pode-se entender com maior clareza que a ferramenta tem como o escopo o subdomínio de Serviço ao Cliente com integrações ao subdomínio de meios de pagamentos (no caso inicialmente ao serviço oferecido do Mercado Pago) e integrar a cozinha (responsável pela Gestão de alimentos e bebidas) o pedido de uma forma mais coerente e que possa ser acompanhado.

Agora com o entendimento do domínio de atuação, foi realizado um a identificação dos atores, eventos, fluxos e Agregados, Entidades e Valores.

Identificação dos atores, eventos, fluxos e Agregados, Entidades e Valores:

Linguagem Ubíqua:

1. **Pedido:** Representa uma solicitação feita por um cliente que pode incluir um lanche, acompanhamento, bebida e/ou sobremesa. Pode estar associado a um cliente identificado, anônimo ou cadastrado.
2. **Cliente:** Uma pessoa que faz o pedido. Pode ser identificado pelo CPF, nome, e-mail ou permanecer anônimo.
3. **Produto:** Pode ser um lanche, acompanhamento, bebida ou sobremesa. Possui nome, categoria, preço, descrição e imagens.
4. **Ingredientes:** São os elementos usados na preparação de um produto. Exemplos: carnes, queijos, vegetais, molhos e temperos
5. **QRCode do Mercado Pago:** Código que representa a transação de pagamento.
6. **Status do Pedido:** Representa o estado atual do pedido e pode ser: Recebido, Em preparação, Pronto e Finalizado.

Fluxos:

a. Realização do pedido e pagamento:

1. O cliente interage com a interface de seleção.
2. O cliente decide se quer se identificar (CPF, nome, e-mail) ou permanecer anônimo.
3. O cliente seleciona os produtos (lanche, acompanhamento e/ou bebida).
4. O sistema calcula o total e apresenta o QRCode do Mercado Pago para o cliente.
5. O cliente realiza o pagamento através do QRCode.
6. O sistema registra o pedido com status "Recebido".

b. Preparação e entrega do pedido:

1. A cozinha recebe o pedido e começa a prepará-lo.
2. O sistema atualiza o status do pedido para "Em preparação".
3. Uma vez que o pedido está pronto, o sistema atualiza o status para "Pronto" e notifica o cliente.
4. O cliente retira o pedido.
5. O sistema atualiza o status do pedido para "Finalizado".

Agregados, Entidades e Valores:

Pedido (Agregado):

- Cliente (Entidade)
- Lista de Produtos (Entidade)
- Status do Pedido (Valor)
- Total (Valor)
- QRCode do Mercado Pago (Valor)

Produto (Agregado):

- Nome (Valor)
- Categoria (Valor: Lanche, Acompanhamento, Bebida, Sobremesa)
- Preço (Valor)
- Descrição (Valor)
- Imagens (Valor)

Serviços:

1. **Serviço de Pedido:** Responsável pela criação, atualização e consulta de pedidos.
2. **Serviço de Produto:** Responsável pela criação, atualização e consulta de produtos.
3. **Serviço de Cliente:** Responsável pelo registro e consulta de clientes.

4. **Serviço de Pagamento:** Responsável pela geração do QRCode do Mercado Pago e validação do pagamento.

Repositórios:

1. **Repositório de Pedido:** Armazena e recupera pedidos.
2. **Repositório de Produto:** Armazena e recupera produtos.
3. **Repositório de Cliente:** Armazena e recupera informações de clientes.

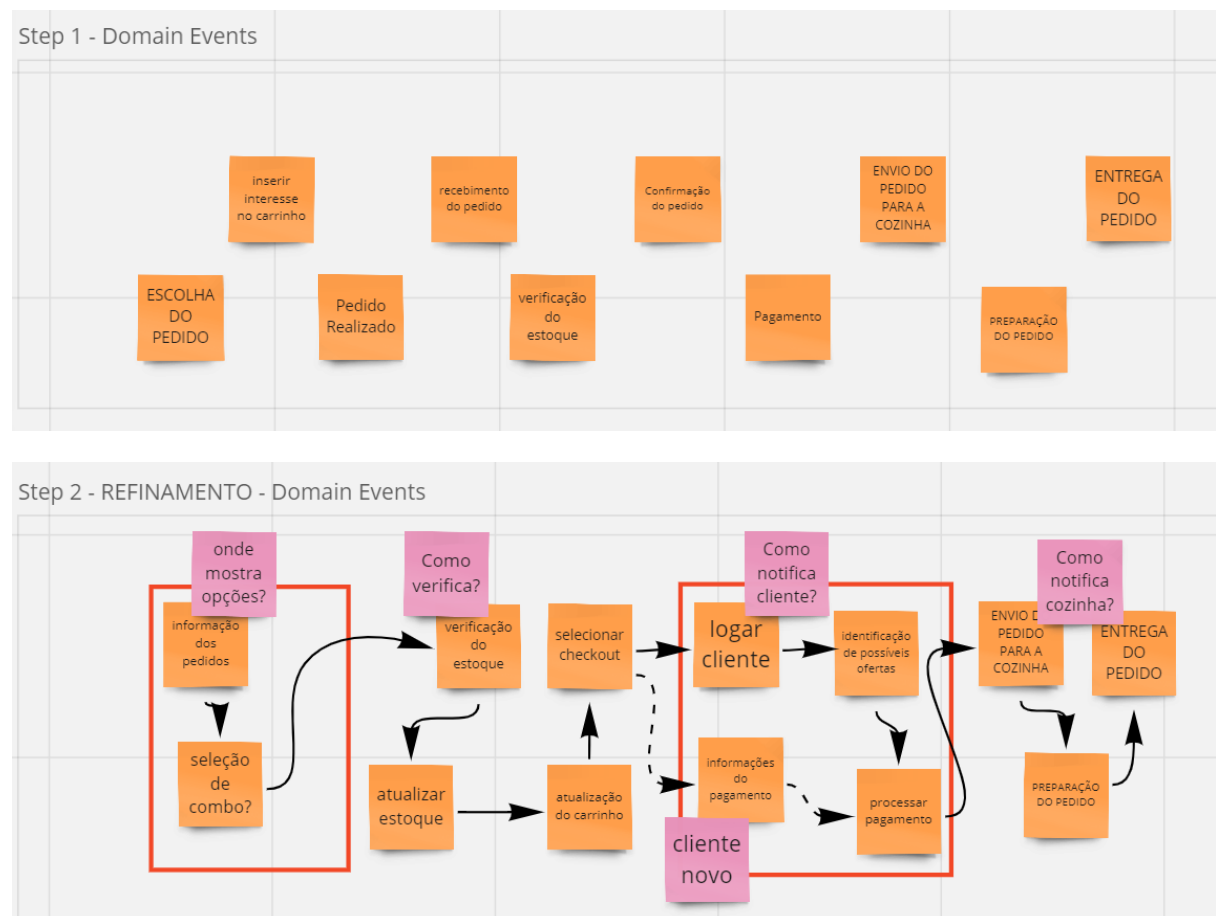
Eventos:

1. **Pedido Criado:** Disparado quando um novo pedido é criado.
2. **Status do Pedido Atualizado:** Disparado quando o status de um pedido é alterado.
3. **Produto Adicionado:** Disparado quando um novo produto é adicionado ao sistema.
4. **Cliente Registrado:** Disparado quando um novo cliente se registra no sistema.

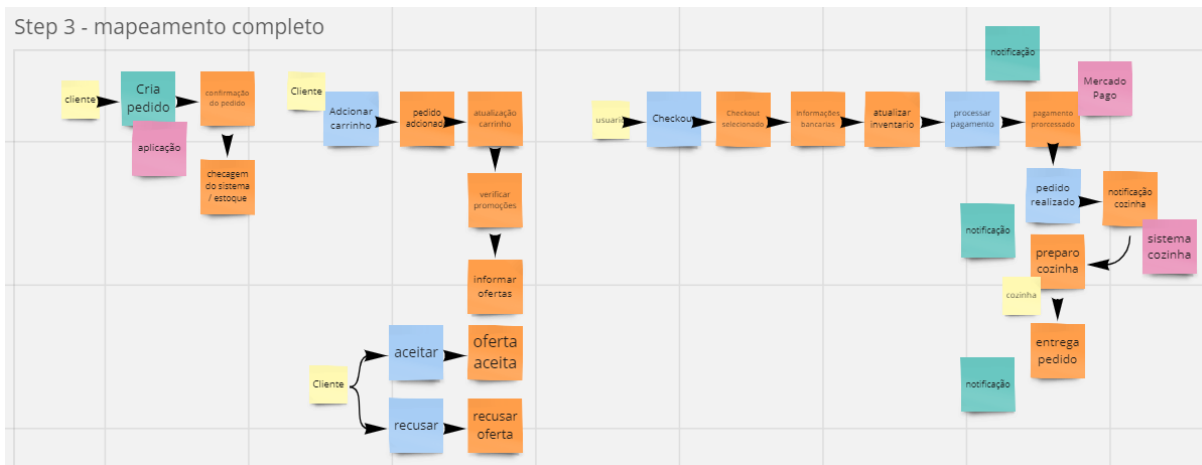
Storming Event

Através do Event Storming buscamos uma maior compreensão de um campo de conhecimento específico e delimitado (um domínio de especialização).

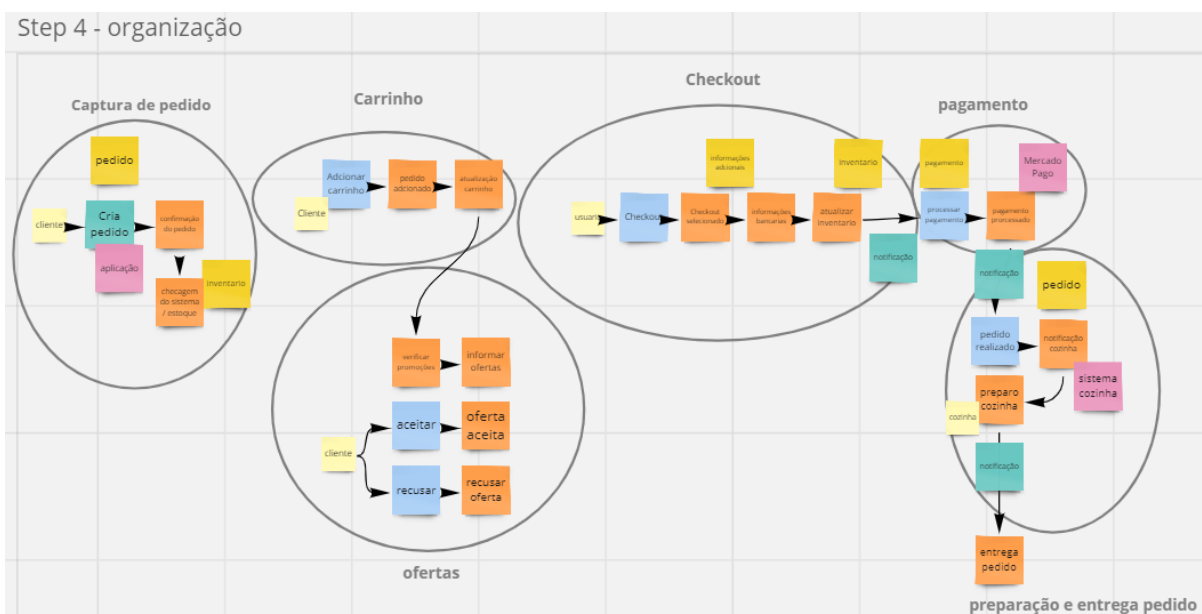
Finalmente, antes de realizar o desenho final, foi criado um Storming Event para entender quais são os eventos que o software deverá possuir para que esse autoatendimento aconteça.



Step 3 - mapeamento completo



Step 4 - organização



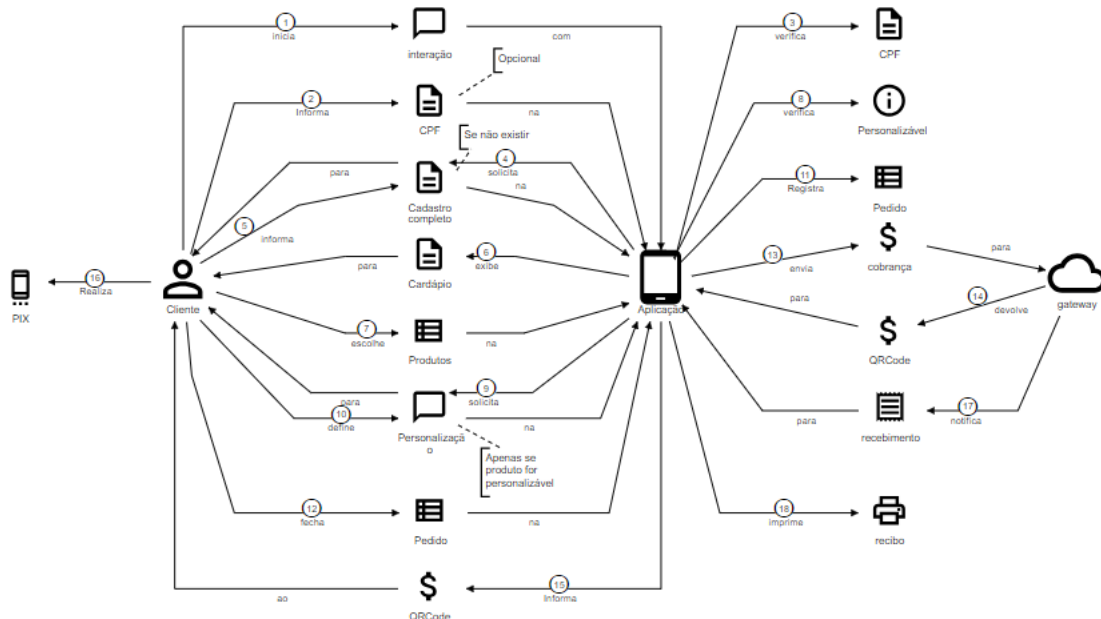
Desenhos com linguagem Ubíqua:

O Domain Storytelling é uma técnica utilizada para entender e comunicar os requisitos de um sistema de forma mais detalhada e envolvente. Ele permite contar histórias sobre o domínio de negócios, envolvendo os principais atores, eventos, fluxos e entidades envolvidas.

No contexto do projeto "FoodieFlow", o Domain Storytelling foi utilizado para identificar e descrever os diferentes domínios de negócio relacionados à lanchonete. Isso incluiu a realização do pedido, seu pagamento, além da preparação e entrega do pedido.

Através do Domain Storytelling, foi possível ter uma visão mais clara e abrangente do escopo do software a ser desenvolvido, compreendendo os principais fluxos de interação entre os atores e as principais funcionalidades a serem implementadas. Isso facilita a comunicação entre os membros da equipe de desenvolvimento e ajuda a garantir que todos tenham uma compreensão comum do sistema a ser construído.

Realização do pedido e pagamento



Preparação e entrega do pedido

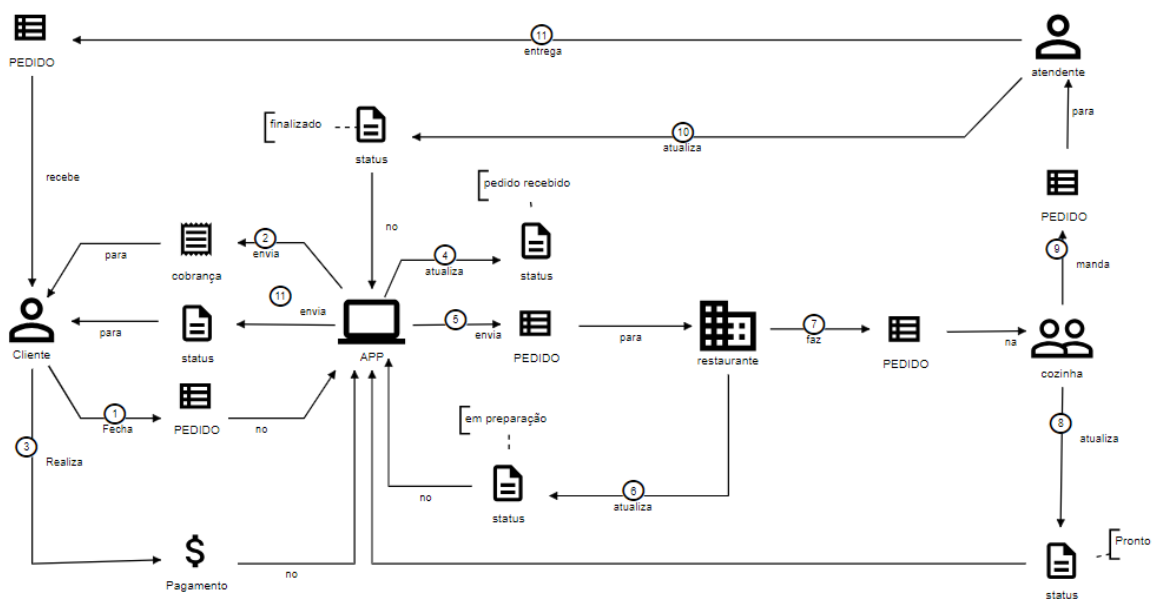


Diagrama de Classes

O diagrama de classes pode ajudar na análise e no projeto de um sistema, fornecendo uma visão geral da estrutura do sistema e das principais entidades envolvidas. Ele permite identificar as classes necessárias para implementar o sistema, bem como as relações entre elas. Além disso, o diagrama de classes pode auxiliar na comunicação entre os membros da equipe de desenvolvimento, servindo como uma representação visual compartilhada do sistema.

No contexto do projeto "FoodieFlow", o diagrama de classes foi utilizado para representar a estrutura do sistema e as principais entidades envolvidas, como Cliente, Produto e Pedido. Ele auxilia na compreensão da organização e das interações entre essas entidades, facilitando o desenvolvimento e a manutenção do sistema.

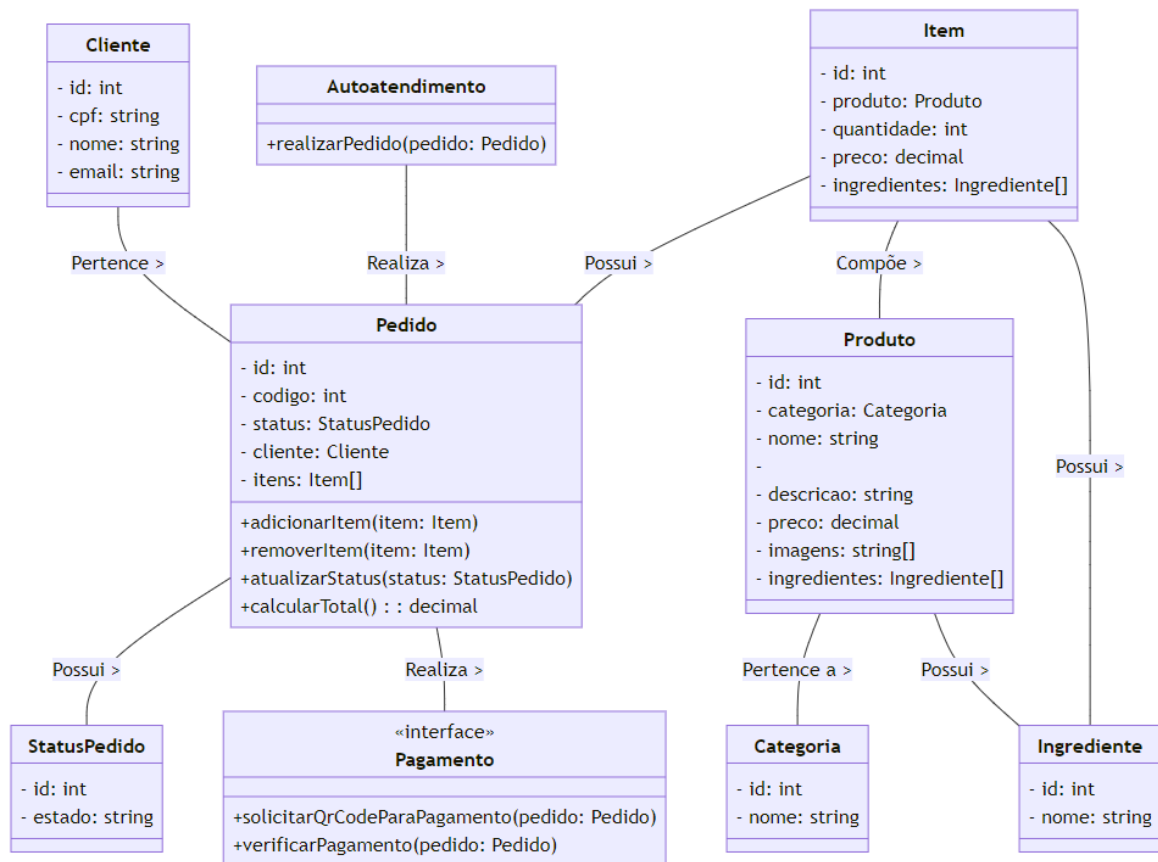


Diagrama de Status

Um diagrama de status é uma representação visual que mostra os diferentes estados ou etapas pelos quais um objeto ou processo pode passar. Ele ajuda a identificar e compreender as transições de estado, permitindo uma melhor visualização do fluxo de informações ou eventos dentro de um sistema.

No contexto do projeto "FoodieFlow", o diagrama de status é utilizado para representar os diferentes estados pelos quais um pedido pode passar, como "Recebido", "Em preparação", "Pronto" e "Finalizado". Ele ajuda a entender o fluxo do pedido, desde o momento em que é feito até a sua conclusão, permitindo uma melhor organização e controle do processo.

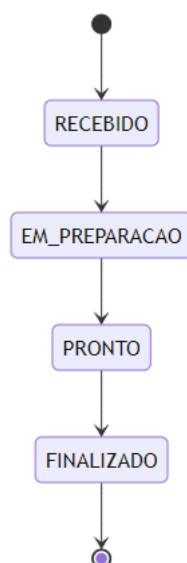


Diagrama Entidade-Relacionamento (ER)

O diagrama apresentado é um Diagrama Entidade-Relacionamento (ER), que é uma representação visual dos relacionamentos entre as entidades de um sistema e sua estrutura de dados. Ele mostra como as entidades estão conectadas e interagem entre si por meio de relacionamentos.

No contexto do projeto "FoodieFlow", o Diagrama ER é utilizado para representar a estrutura de dados do sistema, mostrando as entidades principais, como Cliente, Produto e Pedido, e seus relacionamentos. Ele ajuda a entender como essas entidades estão relacionadas e como os dados são armazenados e recuperados no sistema.

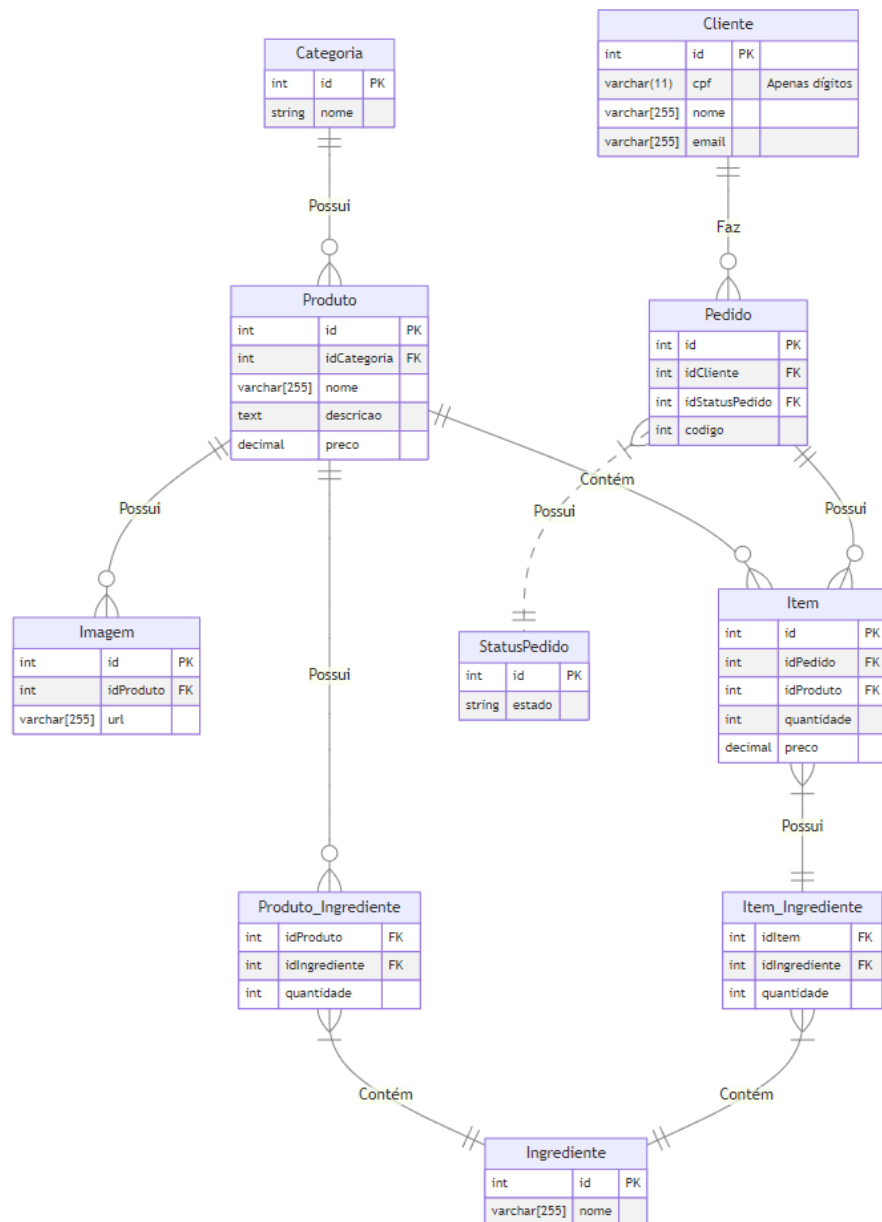
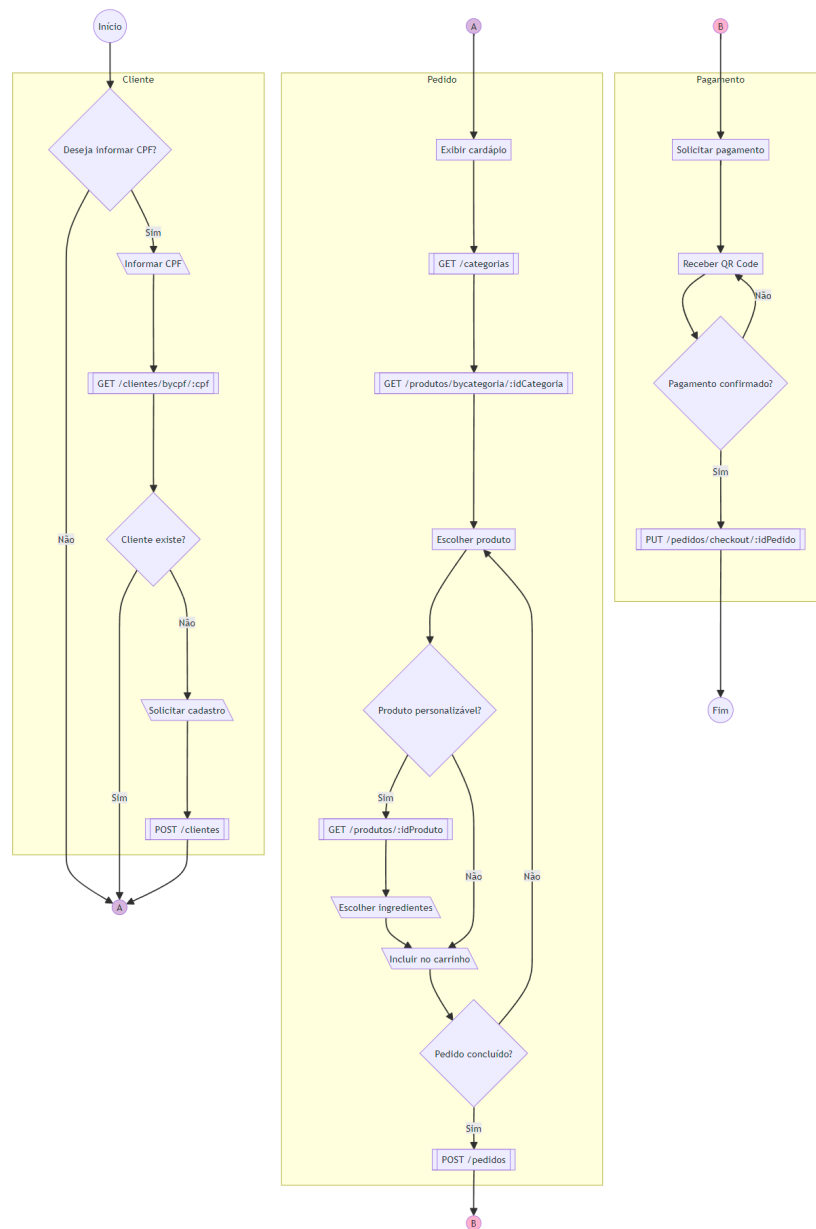


Diagrama de Fluxo com Endpoints

O diagrama de fluxo é uma representação visual que mostra a sequência de passos ou etapas em um processo. Ele é usado para ilustrar a ordem das ações e a interação entre os elementos em um sistema ou fluxo de trabalho.

No contexto do projeto "FoodieFlow", o diagrama de fluxo é utilizado para representar as etapas envolvidas na realização do pedido. Ele mostra a ordem das ações, as decisões tomadas e as interações com a API.



Arquitetura hexagonal

O diagrama abaixo representa os módulos e pacotes escolhidos para representar a arquitetura que será aplicada. Dentre os diversos elementos presentes na arquitetura, foram selecionadas as classes Cliente e Produto para exemplificar a estrutura. Devido à complexidade e extensão do diagrama completo, optou-se por apresentar apenas essas duas classes como representação simplificada.

