

29 DE ABRIL DE 2024



ESTUDO DIAGRAMAS UML

BRUNINI HEITOR
INSTITUTO FEDERAL DA PARAÍBA

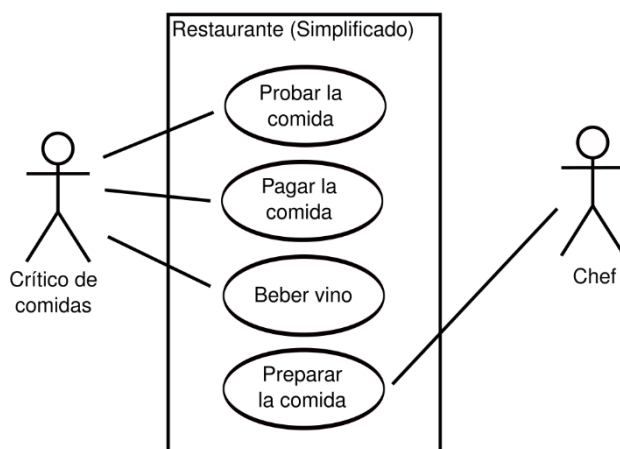
❖ DIAGRAMA CASO DE USO

O Diagrama de Caso de Uso é uma ferramenta da UML (Linguagem de Modelagem Unificada) que descreve as interações entre os usuários (atores) e o sistema, mostrando os principais casos de uso e seus relacionamentos. É uma ferramenta valiosa para capturar os requisitos funcionais do sistema e entender como ele será usado na prática. Aqui estão os principais componentes de um Diagrama de Caso de Uso:

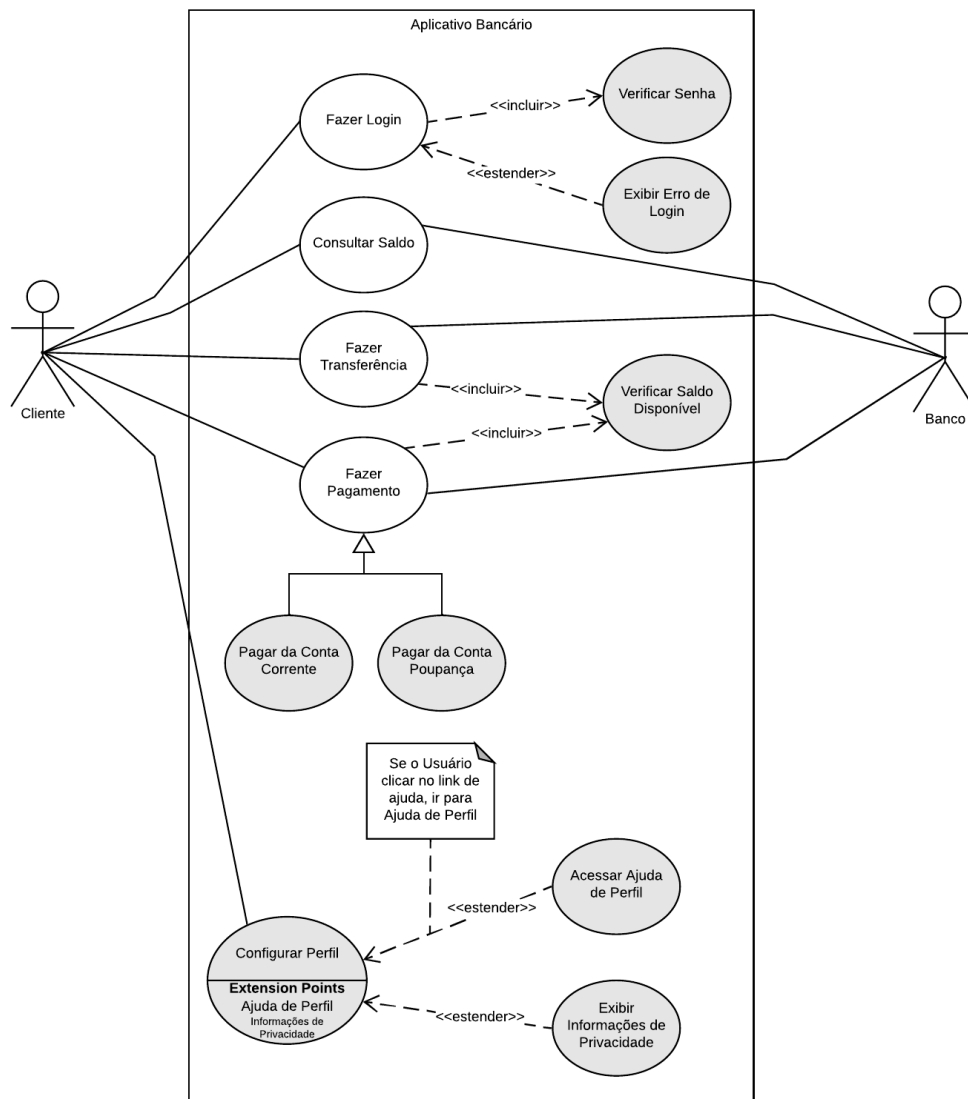
1. **Ator:** Um ator é uma entidade externa que interage com o sistema. Pode ser um usuário humano, outro sistema, dispositivo de hardware, etc. Cada ator é representado por um símbolo de figura humana ou nomeado com um ícone que representa o papel que desempenha em relação ao sistema.
2. **Caso de Uso:** Um caso de uso descreve uma função ou recurso do sistema do ponto de vista do usuário. Ele representa uma sequência de interações entre o sistema e os atores, resultando em um resultado valioso para pelo menos um dos atores. Cada caso de uso é representado por um oval e é nomeado de forma descritiva, descrevendo a ação que o ator deseja realizar.
3. **Relacionamentos:** Os relacionamentos entre os atores e os casos de uso são representados por linhas no diagrama. Os principais relacionamentos incluem:
4. **Associação:** Indica que um ator está associado a um ou mais casos de uso.
5. **Inclusão:** Indica que um caso de uso inclui outro caso de uso, ou seja, o comportamento do caso de uso incluído é sempre executado como parte do comportamento do caso de uso que o inclui.
6. **Extensão:** Indica que um caso de uso estende outro caso de uso, adicionando ou modificando seu comportamento em certas condições.
7. **Sistema:** O sistema em si pode ser representado por um retângulo que envolve todos os casos de uso e atores, indicando os limites do sistema em relação ao ambiente externo.

O Diagrama de Caso de Uso é uma ferramenta poderosa para comunicação entre os stakeholders do projeto, incluindo clientes, gerentes de projeto, analistas de negócios e desenvolvedores. Ele fornece uma visão geral clara e concisa dos requisitos do sistema, ajudando a garantir que todos tenham uma compreensão comum dos objetivos e funcionalidades do sistema a ser desenvolvido.

Figura 1: https://commons.wikimedia.org/wiki/File:UML_diagrama_caso_de_uso.svg



Exemplo completo de caso de Uso:



❖ DIAGRAMA CASO DE CLASSES

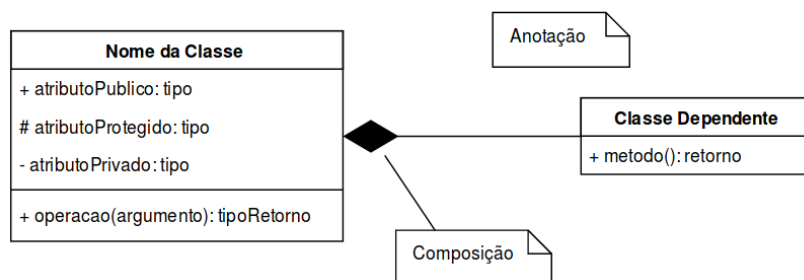
O Diagrama de Classes é uma ferramenta essencial na modelagem de sistemas orientados a objetos. Ele representa a estrutura estática do sistema, mostrando as classes do sistema, seus atributos, métodos e os relacionamentos entre elas. Aqui estão alguns pontos-chave sobre o diagrama de classes:

1. **Classes:** No diagrama de classes, as classes são representadas por retângulos divididos em três compartimentos. O primeiro compartimento contém o nome da classe, o segundo contém os atributos da classe e o terceiro contém os métodos ou operações da classe.
2. **Atributos:** Os atributos representam as características ou propriedades dos objetos da classe. Eles podem incluir informações como nome, tipo de dados e visibilidade (pública, privada, protegida). Os atributos são listados no segundo compartimento do retângulo da classe. A visibilidade dos atributos pode ser indicada por:
 - "+" para atributos públicos

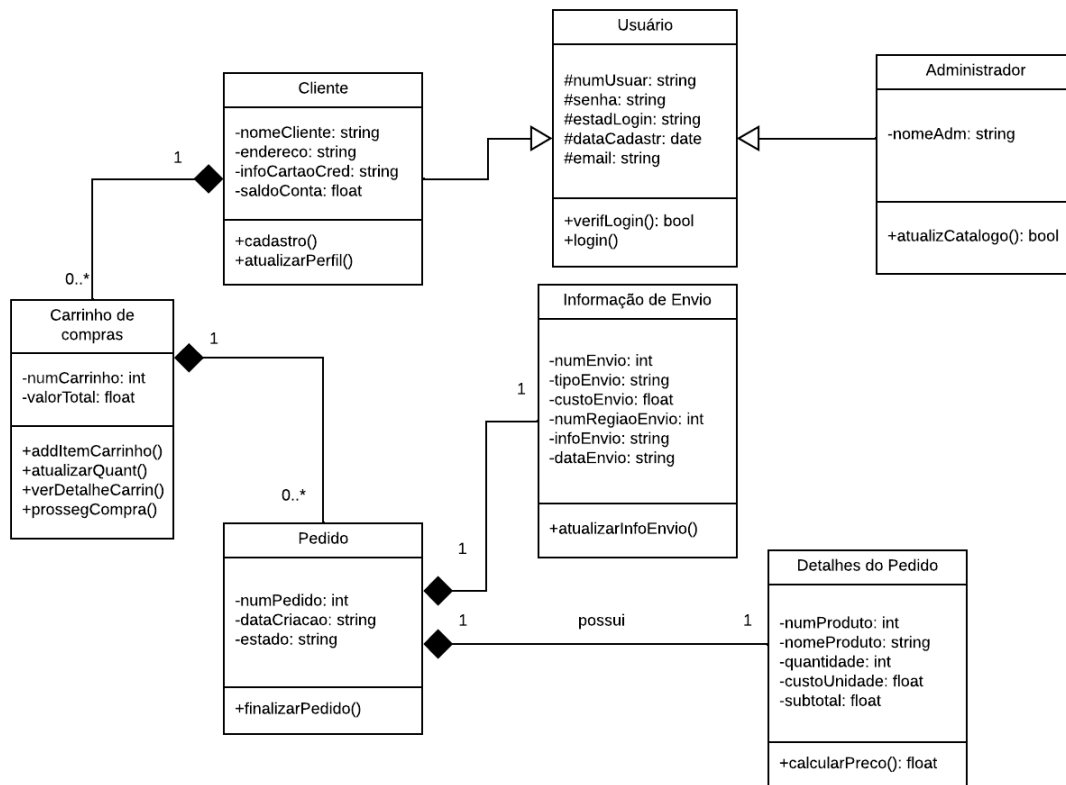
- "-" para atributos privados
 - "#" para atributos protegidos
 - "~" para atributos visíveis para o pacote raiz.
3. **Métodos:** Os métodos representam as operações que os objetos da classe podem realizar. Eles incluem informações como nome, parâmetros e tipo de retorno. Os métodos são listados no terceiro compartimento do retângulo da classe.
 4. **Relacionamentos:** Os relacionamentos entre as classes são representados por linhas no diagrama. Os principais tipos de relacionamentos incluem associação, agregação, composição e herança.
 - Associação: Representa uma relação entre duas classes, indicando que um objeto de uma classe está associado a um objeto de outra classe. É representado por uma linha sólida entre as classes
 - Agregação: Representa uma relação todo-parte entre duas classes, indicando que uma classe é composta por outras classes, mas as partes podem existir independentemente. É representado por uma linha sólida com um losango vazio no lado do todo.
 - Composição: É um tipo mais forte de agregação, indicando que as partes existem apenas como parte de um todo e são destruídas quando o todo é destruído. É representado por uma linha sólida com um losango preenchido no lado do todo.
 - Herança: Representa uma relação de especialização/generalização entre duas classes, indicando que uma classe (subclasse) herda os atributos e métodos de outra classe (superclasse). É representado por uma linha sólida com uma seta apontando para a superclasse.
 - Dependência: Indica que uma classe depende de outra classe de alguma forma, por exemplo, através do uso de parâmetros em um método. É representado por uma linha tracejada com uma seta apontando para a classe de que depende.
 5. **Interfaces:** Interfaces podem ser representadas no diagrama de classes, indicando os serviços que uma classe oferece, mas sem implementar os detalhes de como esses serviços são realizados. As interfaces são representadas por uma linha tracejada com um estereótipo <<interface>> sobre a classe.

O diagrama de classes é uma ferramenta poderosa para visualizar a estrutura de um sistema de software, identificar as classes e seus relacionamentos, e facilitar a comunicação entre os membros da equipe durante o processo de design e desenvolvimento.

Figura 2: https://pt.wikipedia.org/wiki/Diagrama_de_classes#/media/Ficheiro:Diagrama_de_Classes_com_duas_classes.png



Exemplo completo de diagrama de classes:



❖ DIAGRAMA DE SEQUÊNCIA

Um Diagrama de Sequência na UML (Linguagem de Modelagem Unificada) é uma representação visual da interação entre objetos em um sistema ao longo do tempo. Ele descreve a troca de mensagens entre os objetos em uma sequência temporal, mostrando como os objetos colaboram para realizar uma determinada funcionalidade. Aqui estão os principais pontos sobre os diagramas de sequência:

1. **Objetos:** Os objetos que participam da interação são listados na parte superior do diagrama, ao longo do eixo vertical. Cada objeto é representado por um retângulo com seu nome.
2. **Mensagens:** As mensagens trocadas entre os objetos são representadas por setas horizontais, indicando a direção da comunicação. As mensagens podem ser:
 - Mensagens Síncronas: Representam chamadas de método entre objetos, onde o remetente aguarda uma resposta do destinatário antes de continuar a execução. Elas são representadas por setas sólidas.
 - Mensagens Assíncronas: Representam envio de mensagens entre objetos, onde o remetente não espera uma resposta imediata do destinatário e continua sua execução. Elas são representadas por setas pontilhadas.
 - Mensagens de Retorno:
 - Após o envio de uma mensagem síncrona, o destinatário pode responder com uma mensagem de retorno para indicar o resultado da operação. Essa mensagem de retorno é representada por uma seta pontilhada retornando ao objeto remetente.

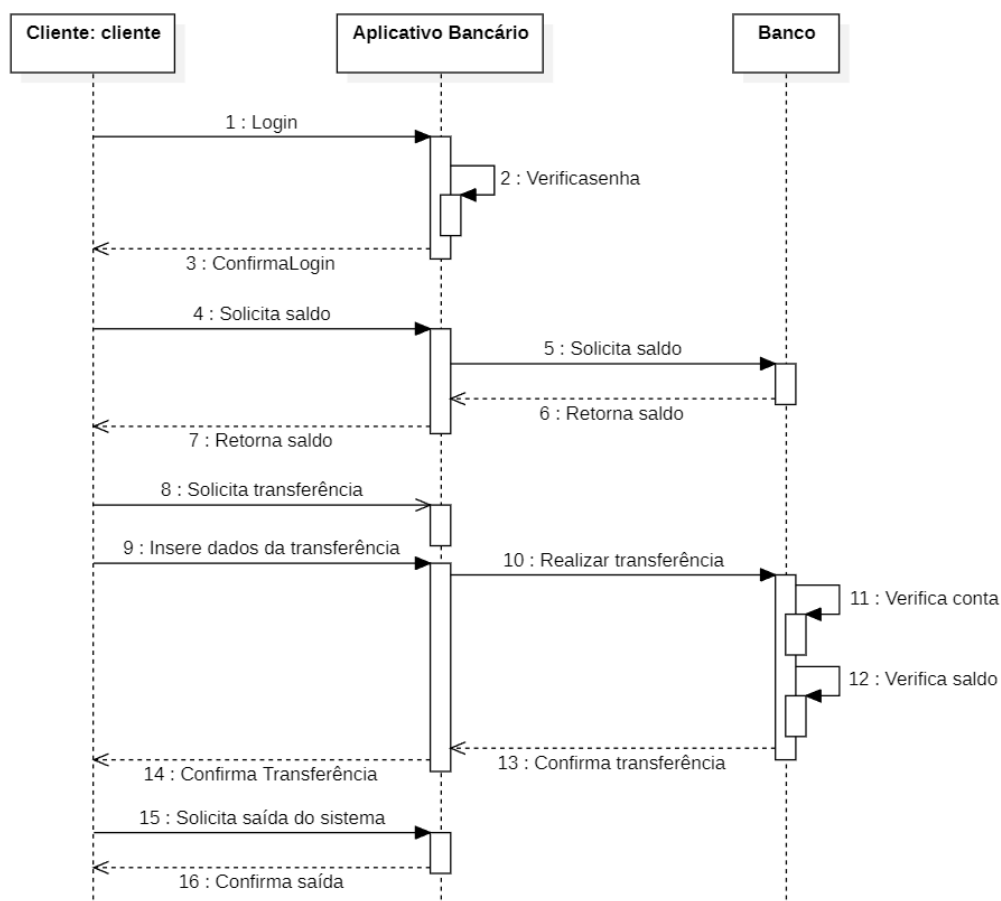
3. **Ativação:** A ativação de um objeto é representada por uma linha vertical tracejada que indica quando o objeto está ativo e processando uma mensagem.
4. **Interação:** A interação entre os objetos é mostrada através da ordem e do tempo em que as mensagens são trocadas. A ordem das mensagens ao longo do tempo é representada da esquerda para a direita.
5. **Fragments de Interação:** São usados para representar estruturas de controle de fluxo, como loops, ramificações e iterações dentro do diagrama de sequência. Os fragmentos de interação incluem fragmentos combinados, alternativos, opcionais e de loop.
6. **Foco da Sequência:** O foco da sequência é uma maneira de mostrar detalhes de uma parte específica de um diagrama de sequência em uma caixa retangular dentro do diagrama. Ele destaca a interação entre um conjunto específico de objetos em uma parte do diagrama de sequência.

Os diagramas de sequência são amplamente utilizados durante a fase de projeto de software para visualizar e comunicar a interação dinâmica entre os objetos em um sistema. Eles ajudam a identificar os requisitos de comunicação entre objetos, a validar o comportamento esperado do sistema e a entender como os diferentes componentes interagem entre si durante a execução do sistema. Esses diagramas são especialmente úteis para entender casos de uso complexos e cenários de interação entre diferentes partes do sistema.

Como criar um diagrama de sequência:

1. Selecionar um Caso de Uso no diagrama de casos de uso
2. Escrever uma descrição detalhada do que o caso de uso faz.
3. Identificar os atores e objetos que interagem
4. Identificar as mensagens que os objetos trocam entre si
5. Determinar as condições especiais (loops, fluxos alternativos, etc.)
6. Desenhar o diagrama.

Exemplo de diagrama de sequência:



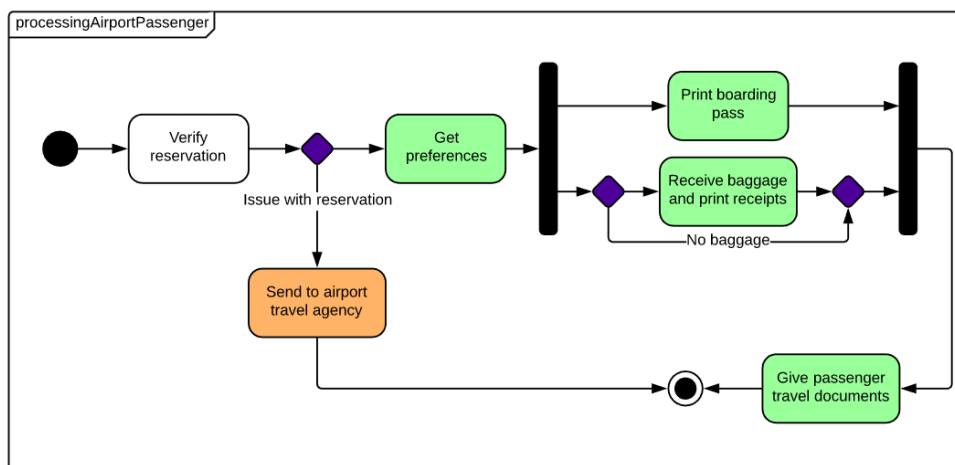
❖ DIAGRAMA DE ESTADOS

O Diagrama de Estados na UML é usado para visualizar o comportamento de um objeto ou sistema ao longo do tempo, mostrando os diferentes estados que o objeto pode assumir e as transições entre esses estados em resposta a eventos externos. Aqui estão os principais pontos sobre o diagrama de estados:

1. **Estado:** Um estado representa uma condição específica em que um objeto ou sistema pode existir em um determinado momento. Por exemplo, um objeto pode estar em um estado "ligado" ou "desligado".
2. **Transição:** Uma transição representa a mudança de um estado para outro em resposta a um evento. Os eventos podem ser acionados por estímulos externos, como a entrada do usuário, ou por condições internas, como um temporizador expirado.
3. **Evento:** Um evento é uma ocorrência que desencadeia uma transição de estado. Os eventos podem ser representados por gatilhos como "entrada de dados", "clique de botão" ou "temporizador expirado".
4. **Ação:** Uma ação é uma atividade que é executada quando ocorre uma transição de estado. As ações podem incluir operações específicas, como atualizar variáveis, exibir mensagens, notificar outros objetos, etc.
5. **Estado Inicial e Final:** Um diagrama de estados geralmente começa com um estado inicial, que representa o estado inicial do objeto ou sistema. Ele também pode incluir um ou mais estados finais, que representam o término do ciclo de vida do objeto ou sistema.
6. **Diagrama de Estados Hierárquico:** Em sistemas complexos, pode ser útil criar um diagrama de estados hierárquico, onde os estados são agrupados em estados compostos e sub-estados. Isso ajuda a organizar e visualizar de forma mais clara os diferentes estados e transições em um sistema grande e complexo.
7. **Guardas:** é comum usar condições (também conhecidas como guardas) para especificar quando uma transição de estado deve ocorrer. Essas condições determinam se uma transição de estado é ativada ou não com base em certas circunstâncias, são representadas por um losango preto.

Os diagramas de estados são amplamente utilizados na modelagem de sistemas reativos, como sistemas embarcados, sistemas de controle e software baseado em eventos. Eles ajudam a entender o comportamento dinâmico de um sistema ao longo do tempo e fornecem uma visão clara das diferentes condições e ações que podem ocorrer durante a execução do sistema. Esses diagramas são valiosos para identificar requisitos de comportamento, validar o design do sistema e implementar lógica de controle adequada em sistemas de software.

Figura 3: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-maquina-de-estados-uml>



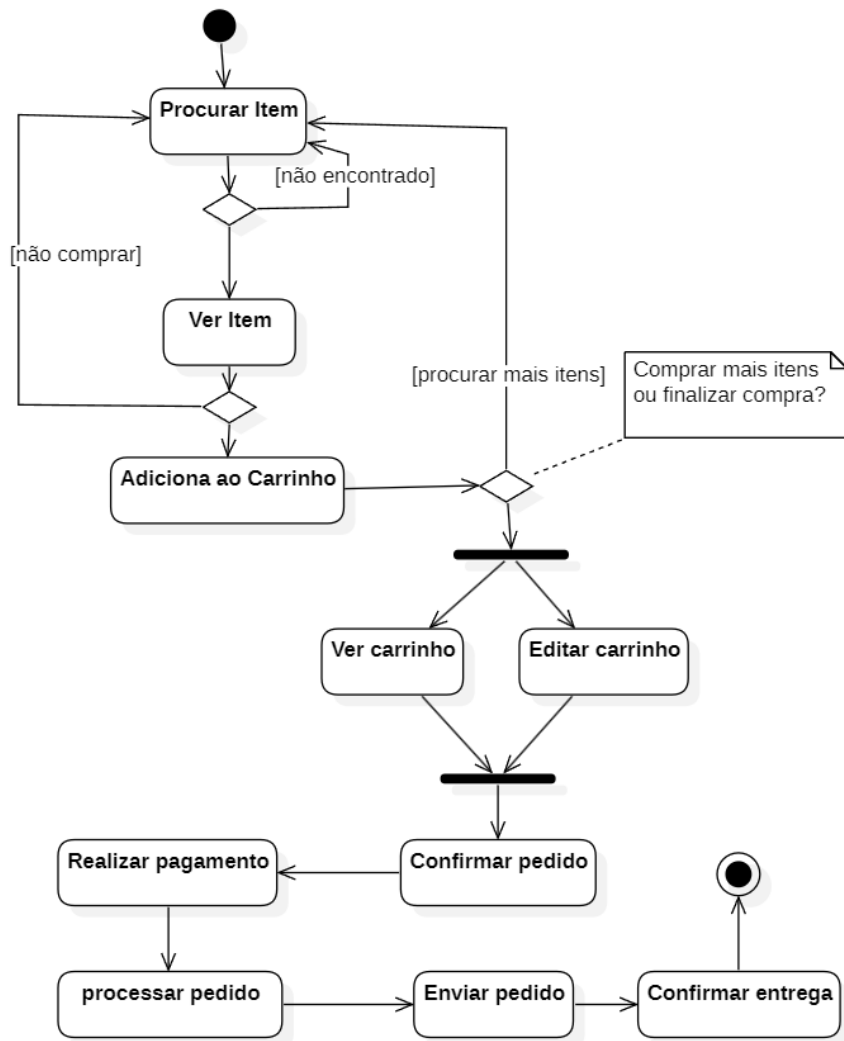
❖ DIAGRAMA DE ATIVIDADES

O Diagrama de Atividades na UML (Linguagem de Modelagem Unificada) é usado para modelar o fluxo de controle de atividades em um sistema, mostrando como as atividades são executadas em sequência ou em paralelo. Aqui estão os principais pontos sobre o diagrama de atividades:

- **Atividade**: Uma atividade é uma ação ou conjunto de ações que são realizadas em um sistema. Elas são representadas por retângulos no diagrama de atividades e podem incluir atividades simples, como "enviar e-mail" ou "processar pedido", bem como atividades mais complexas, como "verificar disponibilidade de estoque" ou "processar pagamento".
- **Fluxo de Controle**: O fluxo de controle das atividades é representado por setas que conectam as atividades. As setas mostram a ordem em que as atividades são executadas, indicando a direção do fluxo de controle.
- **Decisões**: Decisões são representadas por losangos no diagrama de atividades e são usadas para modelar pontos de decisão no fluxo de controle. Dependendo de uma condição específica, o fluxo pode seguir por caminhos diferentes.
- **Fork e Join**: Forks e joins são usados para modelar paralelismo no diagrama de atividades. Um fork divide o fluxo de controle em múltiplos caminhos paralelos, enquanto um join combina os caminhos paralelos de volta em um único fluxo de controle.
- **Bifurcações e Junções**: Bifurcações e junções são usadas para modelar concorrência no diagrama de atividades. Uma bifurcação cria múltiplos caminhos de execução concorrentes, enquanto uma junção combina esses caminhos de volta em um único caminho.
- **Atividades de Início e Fim**: O diagrama de atividades geralmente começa com uma atividade de início, que representa o início do fluxo de controle, e termina com uma atividade de fim, que representa o término do fluxo de controle.
- **Partições de Natação (Swimlanes)**: Partições de natação são usadas para agrupar atividades relacionadas ou atribuir atividades a diferentes atores ou unidades organizacionais. Elas são representadas por linhas horizontais no diagrama de atividades e ajudam a organizar e visualizar as responsabilidades de cada participante ou grupo de atividades.

Os diagramas de atividades são frequentemente usados durante a análise e o projeto de sistemas de software para modelar processos de negócios, fluxos de trabalho e algoritmos complexos. Eles fornecem uma representação visual clara e concisa do comportamento dinâmico do sistema e ajudam a identificar requisitos de funcionamento, validar a lógica do sistema e comunicar efetivamente o fluxo de controle entre os stakeholders do projeto.

Exemplo de diagrama de Atividades:



Fontes de estudo utilizadas:

[Bóson Treinamentos - YouTube](#)

[Lucid Software Português - YouTube](#)

[ChatGPT | OpenAI](#)