A **Modelagem UML** (Unified Modeling Language, ou Linguagem de Modelagem Unificada) é uma linguagem padrão usada para descrever, especificar, projetar e documentar sistemas de software. Ela fornece uma representação visual dos componentes de um sistema e suas interações. A UML é amplamente usada no desenvolvimento de software orientado a objetos, mas também pode ser aplicada em outras abordagens de desenvolvimento.

A UML é composta por diversos tipos de diagramas, cada um com uma finalidade específica, como:

1. **Diagramas Estruturais** (ex.: diagrama de classes, diagrama de objetos, diagrama de componentes, etc.)
2. **Diagramas Comportamentais** (ex.: diagrama de casos de uso, diagrama de sequência, diagrama de atividades, etc.)

**Exemplo de Modelagem UML**

Vamos considerar um exemplo simples de um sistema de **gestão de biblioteca**.

**1. Diagrama de Casos de Uso**

Este diagrama mostra os diferentes casos de uso (funcionalidades) que um usuário pode executar no sistema, além de representar a interação entre o ator (usuário) e o sistema.

**Casos de uso para a Biblioteca**:

* **Usuário** pode **Consultar livros**
* **Usuário** pode **Reservar livros**
* **Bibliotecário** pode **Adicionar livros**
* **Bibliotecário** pode **Remover livros**
* **Bibliotecário** pode **Gerenciar empréstimos**

**2. Diagrama de Classes**

O diagrama de classes descreve as classes do sistema e seus relacionamentos. Cada classe pode ter atributos (dados) e métodos (ações).

**Exemplo de Classes para a Biblioteca**:

* **Livro**: Tem atributos como título, autor, e disponibilidade. Métodos incluem emprestar() e reservar().
* **Usuário**: Atributos como nome e ID, com métodos como consultarLivro() e reservarLivro().
* **Bibliotecário**: Atributos como nome e ID, com métodos como adicionarLivro() e removerLivro().

O diagrama de classes pode parecer com isso:

+----------------+ +----------------+ +----------------+

| Livro | | Usuário | | Bibliotecário |

+----------------+ +----------------+ +----------------+

| -titulo: String| | -nome: String | | -nome: String |

| -autor: String | | -id: Integer | | -id: Integer |

| -disponibilidade: Boolean| | | | |

+----------------+ +----------------+ +----------------+

| +consultarLivro()| | +consultarLivro()| | +adicionarLivro()|

| +reservar() | | +reservarLivro() | | +removerLivro() |

+----------------+ +----------------+ +----------------+

Neste exemplo:

* O **Livro** possui um título, autor e um status de disponibilidade.
* O **Usuário** pode consultar e reservar livros.
* O **Bibliotecário** pode adicionar ou remover livros do sistema.

**3. Diagrama de Sequência**

O diagrama de sequência descreve como os objetos interagem ao longo do tempo em uma operação ou cenário. Ele mostra a sequência de mensagens trocadas entre os objetos.

**Exemplo de Sequência: Usuário reservando um livro**

1. O **Usuário** envia uma mensagem ao sistema para **consultar o livro**.
2. O **Sistema** verifica se o livro está disponível e responde.
3. Se o livro estiver disponível, o **Usuário** envia uma mensagem para **reservar o livro**.
4. O **Sistema** confirma a reserva.

O diagrama de sequência seria algo como:

Usuário Sistema Livro

| | |

|----consultar()-->| |

| |---disponível()?-->|

| |<---sim----------|

|----reservar()-->| |

| |---reservar()-->|

| |<---confirmado--|

**Conclusão**

A **modelagem UML** permite uma visualização clara e precisa do sistema em desenvolvimento, o que ajuda a entender e comunicar a estrutura e o comportamento do software. Ela facilita a documentação, a análise de requisitos, e a comunicação entre desenvolvedores, designers e clientes.