

# Diagramas de UML

O diagrama UML (Linguagem de modelagem unificada) é como um mapa visual que os desenvolvedores usam para representar de forma clara a estrutura e o funcionamento de um sistema. É tipo um conjunto de desenhos que mostra como as partes de um software interagem entre si, como as informações são processadas e como tudo se encaixa. É como um **esboço visual** que ajuda os programadores a entenderem e planejarem melhor seus códigos.

O uml é dividido em duas categorias principais: **diagramas de estrutura e diagramas comportamentais**.

O diagrama de estrutura mostra os objetos em um sistema, e o diagrama comportamental mostra o que vai acontecer, como os objetos vão interagir uns com os outros.

O diagrama de Uml possui 14 tipos de diagramas, são eles:

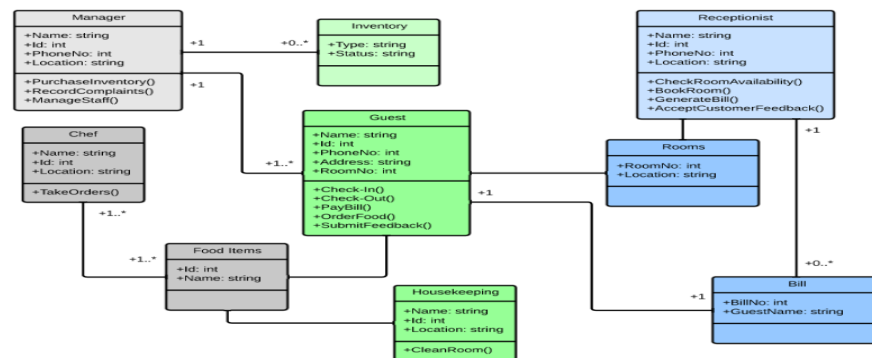
## **Diagramas de Estrutura:**

1. Diagrama de Classes;
2. Diagrama de Componentes;
3. Diagrama de Implantação;
4. Diagrama de Objetos;
5. Diagrama de Pacotes;
6. Diagrama de Perfil;
7. Diagrama de Estrutura Composta;

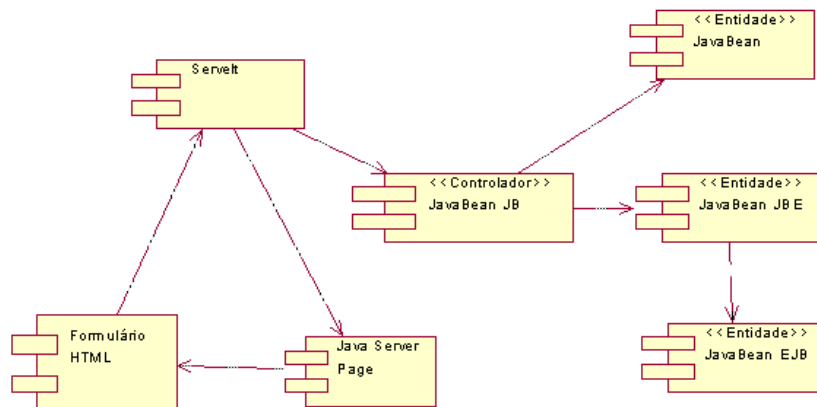
## **Diagramas Comportamentais:**

8. Diagrama de Caso de Uso;
9. Diagrama de Atividades;
10. Diagrama de Máquina de Estado;
11. Diagrama de Sequência;
12. Diagrama de Comunicação;
13. Diagrama de Visão Geral da Interação;
14. Diagrama de Tempo;

1. **Diagrama de Classes:** Este diagrama representa a estrutura e relações das classes que servem de modelo para objetos. Uma classe é representada por um retângulo com três divisões: o nome da classe, seus atributos e seus métodos. É o padrão, todo estudante mesmo que iniciante em banco de dados conhece e já usou ou no mínimo já viu.  
Ponto forte: Como é disparado o mais conhecido de todos os diagramas é fácil de compreender por todos, fora que não é um diagrama tão técnico.  
Ponto fraco: Como o diagrama de classe não é tão complexo, se não bem feito fica confuso

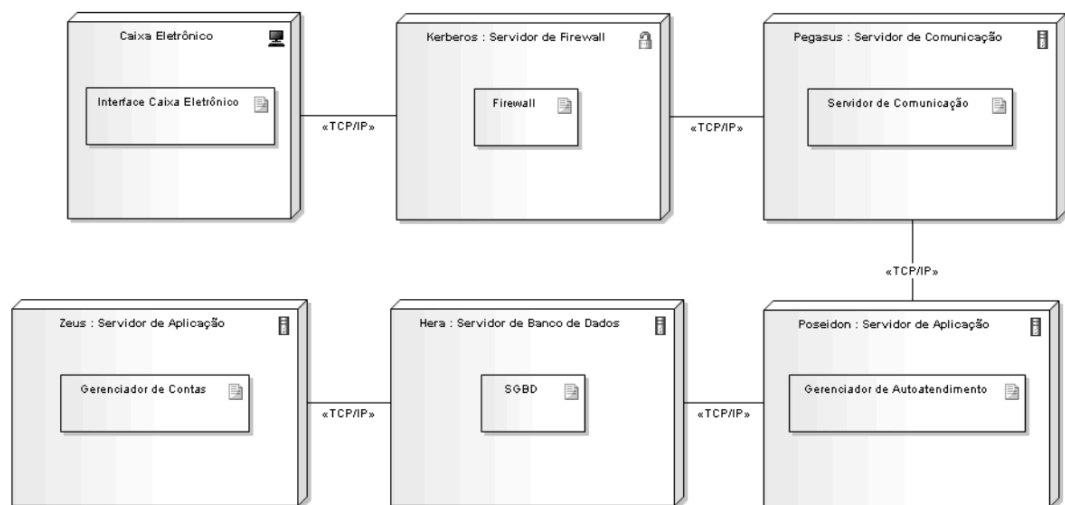


2. **Diagrama de Componentes:** Este diagrama é usado para mostrar a estrutura de sistemas existentes e a criar sistemas novos. Ele representa o relacionamento entre diferentes componentes de um sistema.  
Pontos fortes: Ajuda a entender principalmente a estrutura física do sistema e como os componentes se relacionam.  
Pontos fracos: A complexidade é algo a ser comentado, pois pode aumentar demais a dificuldade de compreensão com o aumento de componentes.

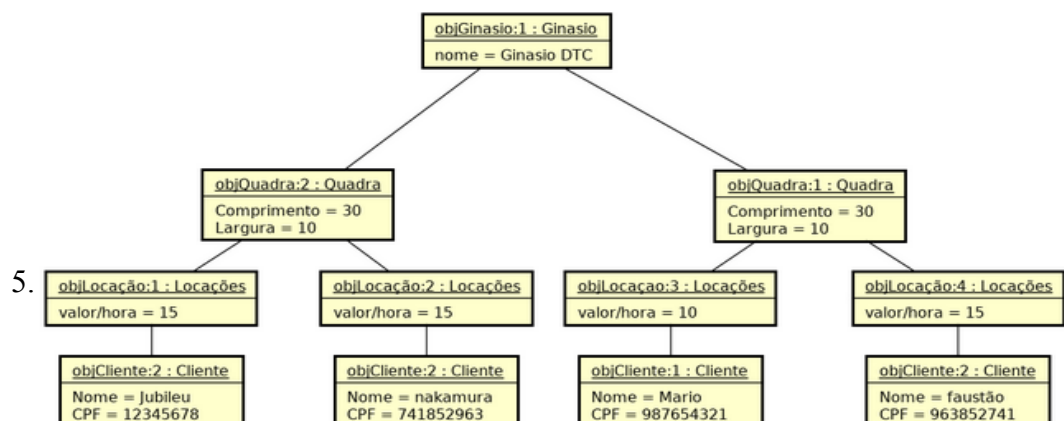


3. **Diagrama de Implantação:** basicamente, o diagrama de implantação UML mostra como o software é distribuído e como ele interage com o hardware. Ele mapeia as partes do software (chamadas de artefatos) para o hardware que vai executá-las. Imagine um prédio, o diagrama de implantação UML seria como o plano que mostra onde cada parte do prédio vai, como a eletricidade, a água, os quartos, etc. No caso de um sistema, o “prédio” é o hardware (como servidores, computadores) e as “partes” são os componentes do software.
- Pontos fortes: Mostra a configuração de tempo de execução dos nós de processamento e os componentes que vivem neles.
- Pontos fracos: Pode não ser útil para sistemas simples ou pequenos.

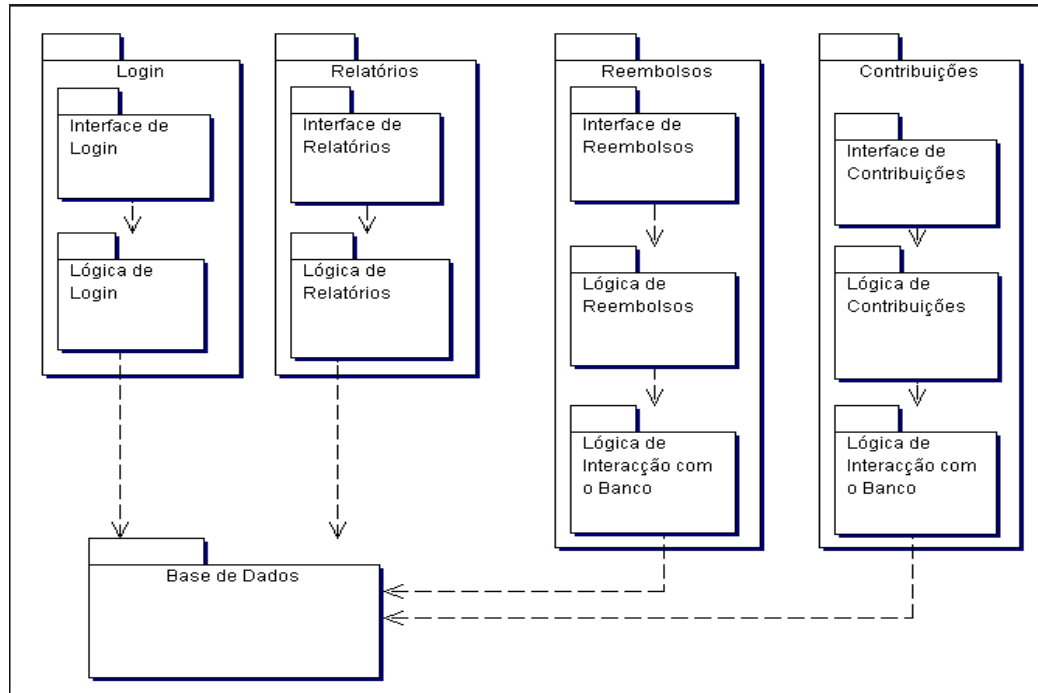
deployment Sistema de Controle Bancário



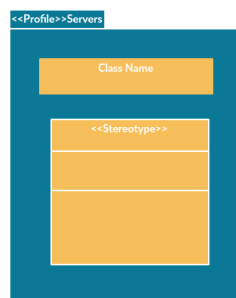
4. **Diagrama de Objetos:** Imagine que você está organizando uma festa de aniversário. Você tem vários “objetos” que fazem parte da festa, como convidados, bolo, balões, presentes. Cada um desses objetos tem características próprias, tipo, cada convidado tem um nome, uma idade, e uma relação com o aniversariante. O bolo tem um sabor e um tamanho. Os balões têm cores, formas, e tamanhos. O diagrama de Objetos e o de classes são bem parecidos num geral, porém a diferença entre eles é que o Diagrama de Classes é mais abstrato. Por outro lado, o Diagrama de Objetos é mais concreto.



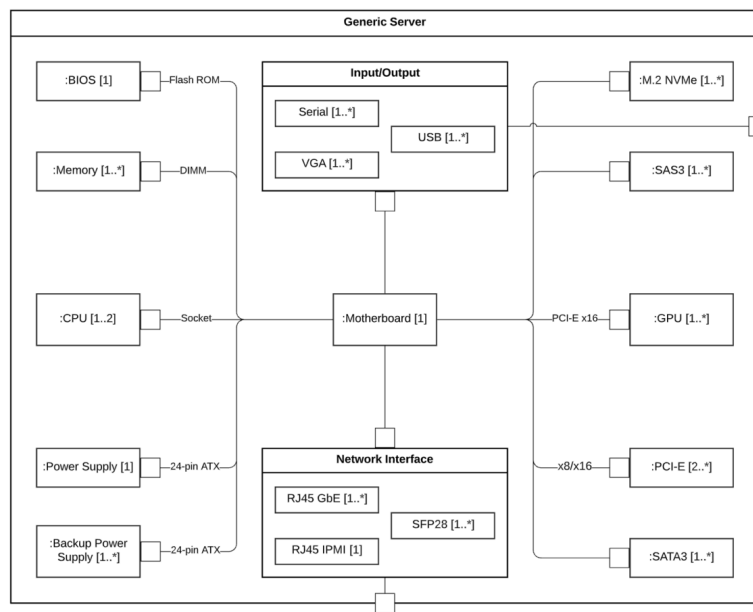
Essas coisas podem ser diagramas, documentos, classes e até outros pacotes. Os Diagramas de Pacotes são super úteis para simplificar diagramas de classes complexos e dar uma **visão geral** de projetos e sistemas de grande escala. Pontos fortes: Ajuda a organizar e agrupar classes relacionadas. Pontos fracos: Não mostra detalhes de como as classes dentro do pacote interagem.



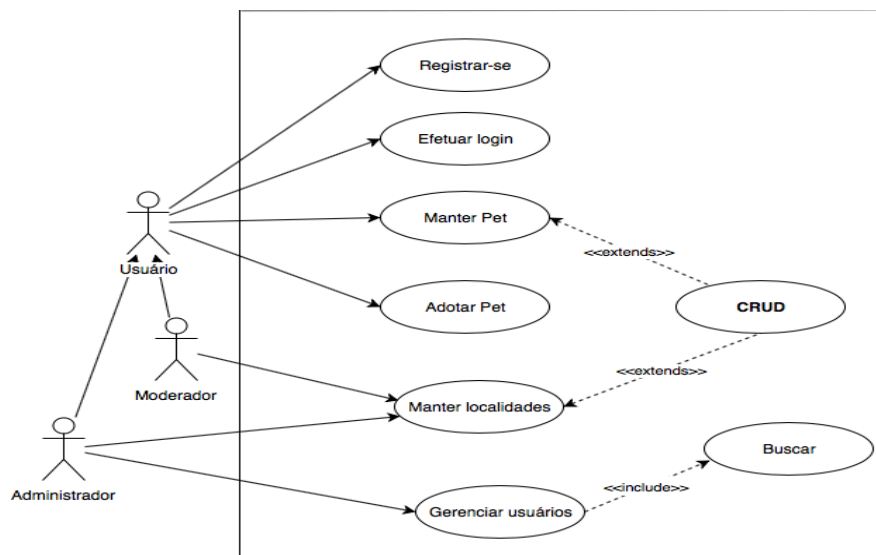
6. **Diagrama de Perfil:** Encontrei muito pouco deste diagrama, até porque ele é raramente utilizado, talvez por ser um diagrama introduzido recentemente. O diagrama de perfil permite aumentar as classes padrão com informações específicas para uma área. Por exemplo, se você está projetando algo relacionado à saúde, poderia criar um perfil de saúde que adiciona propriedades ou comportamentos específicos para lidar com dados médicos. Pontos fracos: Pouco conhecida, dificulta muito a compreensão de quem vê de fora.



7. **Diagrama de Estrutura Composta:** permite que você agrupe partes menores (componentes) para criar algo maior e mais complexo. É como montar peças de LEGO para formar uma construção maior. Então, se você tem várias classes ou componentes que funcionam bem juntas e quer mostrar como elas se encaixam para formar uma parte específica do seu sistema, utilize o diagrama de estrutura composta. É uma forma de organizar e visualizar como as partes se unem para criar uma funcionalidade maior e mais completa.
- Pontos fortes: Mostra a estrutura interna de um classificador.
- Pontos fracos: Pode se tornar complexo se tiver com muitos componentes.



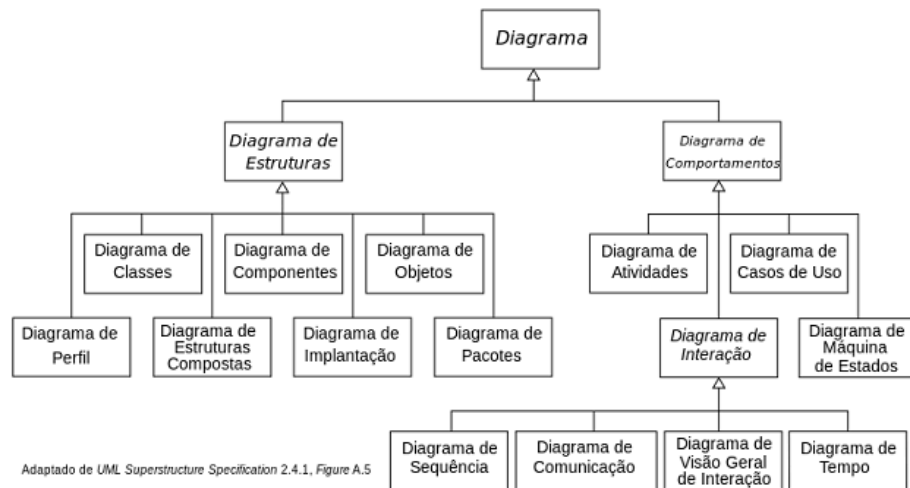
8. **Diagrama de Caso de Uso:** Este diagrama é como um storyboard de um filme. Ele mostra todos os personagens (ou “atores”) e o que eles podem fazer com o sistema. É como se fosse um resumo de todas as coisas interessantes que podem acontecer.
- Pontos fortes: Representa as funcionalidades do sistema a partir da perspectiva do usuário.
- Ponto fraco: Não mostra a estrutura interna do sistema.



9. **Diagrama de Atividades:** Imagine que você está cozinhando seguindo uma receita. O diagrama de atividades é como essa receita, mostrando todas as etapas que você precisa seguir para chegar ao resultado final. Ele mostra o fluxo de controle de uma atividade para outra.

Pontos fortes: Representa o fluxo de controle ou fluxo de objetos entre atividades.

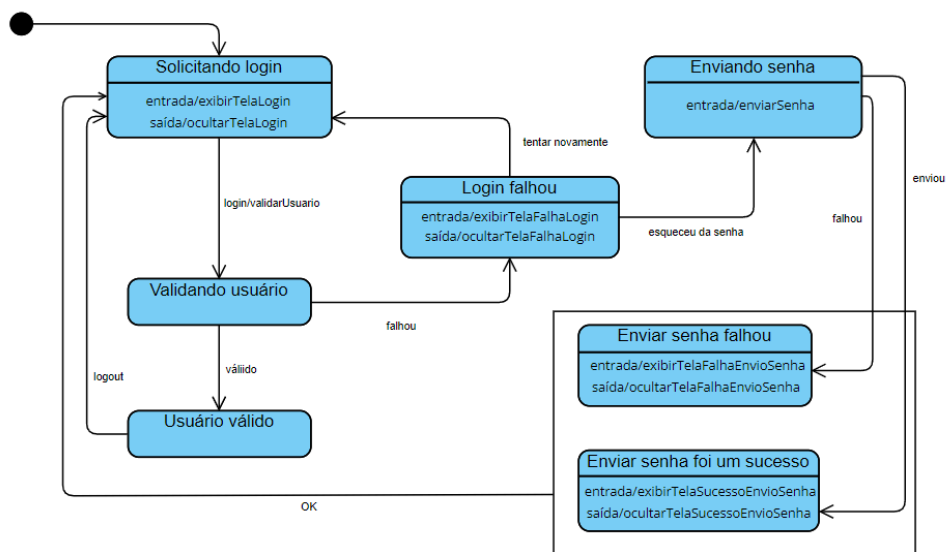
Pontos fracos: Pode se tornar complexo com muitas atividades.



10. **Diagrama de Máquina de Estado:** Este diagrama é como um mapa do humor de uma pessoa. Ele mostra todos os diferentes “estados” que um objeto pode ter, e como ele muda de um estado para outro com base em diferentes eventos.

Pontos fortes: Mostra o comportamento de uma entidade individual ao longo do tempo.

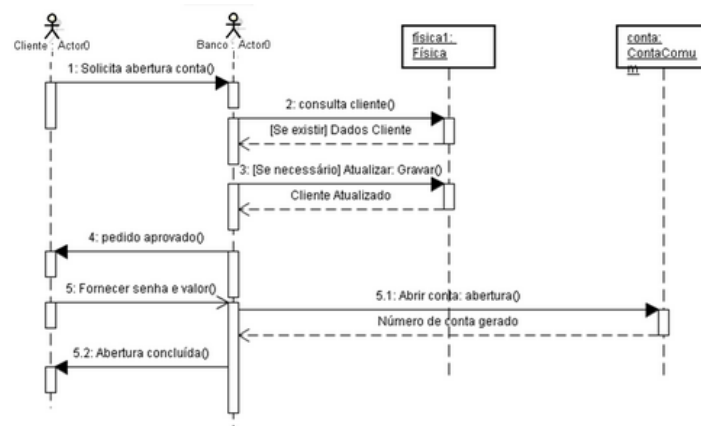
Ponto fraco: Não é útil para representar o comportamento de todo o sistema.



**11. Diagrama de Sequência:** Este diagrama é como uma troca de mensagens entre amigos. Ele mostra a ordem em que as mensagens são enviadas e recebidas entre diferentes objetos ao longo do tempo.

Pontos fortes: Mostra a interação entre objetos em uma sequência temporal.

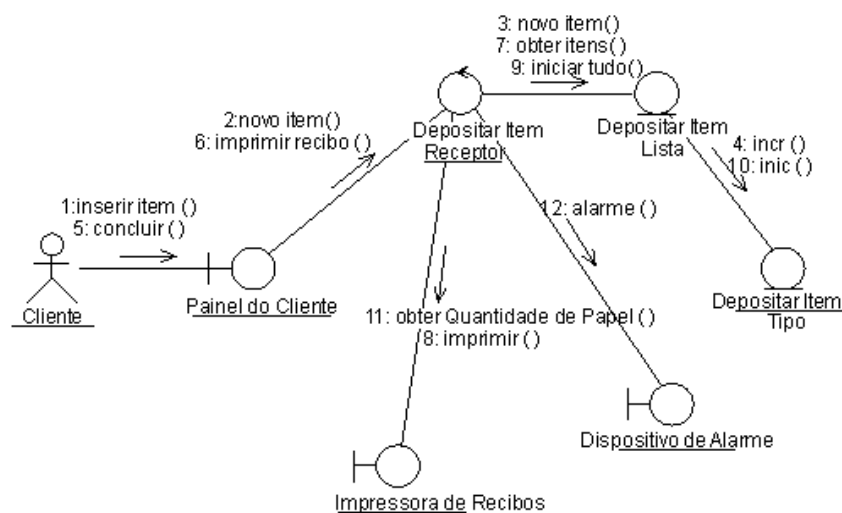
Ponto fracos: Não é útil para representar atividades simultâneas.



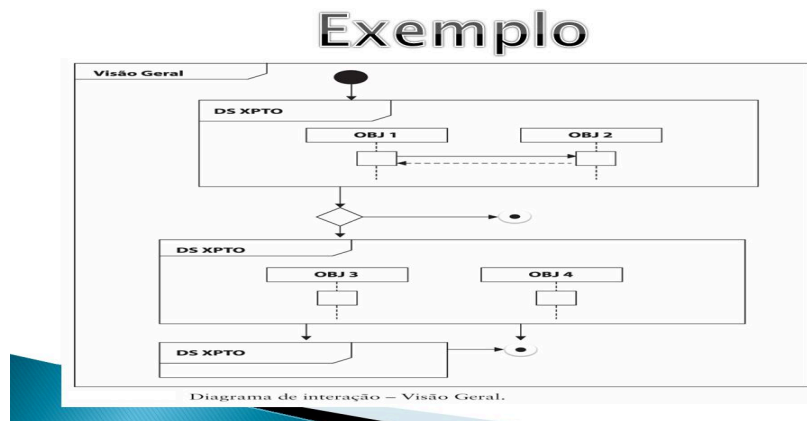
**12. Diagrama de Comunicação:** Este diagrama é como um jogo de telefone sem fio. Ele mostra como as mensagens são passadas de um objeto para outro. É semelhante ao diagrama de sequência, mas se concentra mais na estrutura das mensagens do que na ordem em que são enviadas.

Ponto forte: Mostra a interação entre objetos enfocando a comunicação.

Ponto fraco: Não mostra a sequência temporal das mensagens.



- 13. Diagrama de Visão Geral da Interação:** Este diagrama é como um resumo de um livro. Ele fornece uma visão geral de alto nível de uma interação, mostrando o fluxo de controle entre outros diagramas de interação. Ponto forte: Combina o diagrama de sequência e o diagrama de atividades para representar o fluxo de controle. Ponto fraco: Pode se tornar complexo com muitas interações.



- 14. Diagrama de Tempo:** Este diagrama é como uma linha do tempo de um evento histórico. Ele mostra a sequência de eventos em relação ao tempo. É útil para modelar o comportamento de um sistema em tempo real. Ponto forte: Mostra o comportamento de objetos ao longo do tempo. Ponto fraco: Não é útil para representar a interação entre vários objetos.

