

INTRODUÇÃO A UML

UML

A UML (Unified Modeling Language) é uma linguagem de modelagem visual que foi desenvolvida para ser um padrão para a descrição, especificação, projeto e documentação de sistemas de software orientados a objetos. Ela é amplamente usada na engenharia de software para ajudar a projetar, implementar e testar sistemas de software.

A UML foi desenvolvida no final dos anos 90 por um consórcio de empresas de software, liderado por Grady Booch, Jim Rumbaugh e Ivar Jacobson. A linguagem é composta por diversos tipos de diagramas, cada um dos quais é utilizado para representar um aspecto específico do sistema. Esses diagramas podem ser divididos em três categorias principais: diagramas estruturais, diagramas comportamentais e diagramas de interação.

Os diagramas estruturais são usados para representar a estrutura do sistema e incluem o diagrama de classes, o diagrama de objetos, o diagrama de componentes e o diagrama de implantação. O diagrama de classes é um dos mais utilizados e representa as classes, interfaces e seus relacionamentos no sistema. O diagrama de objetos é usado para representar instâncias de classes e seus relacionamentos. O diagrama de componentes representa os componentes do sistema e suas dependências, enquanto o diagrama de implantação representa a distribuição do sistema em diferentes nós de processamento.

Os diagramas comportamentais são usados para representar o comportamento do sistema e incluem o diagrama de atividades, o diagrama de estados e o diagrama de casos de uso. O diagrama de atividades é usado para representar o fluxo de trabalho do sistema e é especialmente útil para modelar processos de negócios. O diagrama de estados é usado para representar o comportamento de uma classe ou objeto em diferentes estados. O diagrama de casos de uso é usado para representar os casos de uso do sistema e como os usuários interagem com ele.

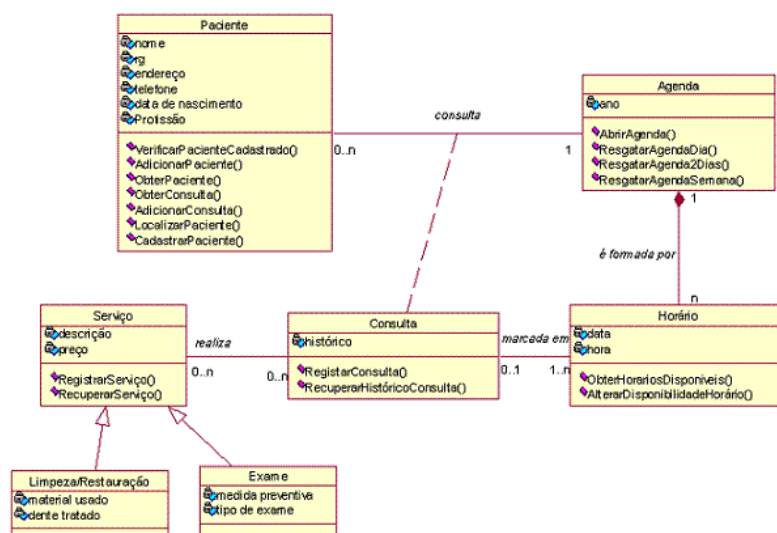
Os diagramas de interação são usados para representar a interação entre objetos e incluem o diagrama de sequência e o diagrama de comunicação. O diagrama de sequência é usado para representar a ordem das mensagens trocadas entre os objetos, enquanto o diagrama de comunicação é usado para representar as mensagens trocadas entre os objetos em uma visão mais geral.

A UML é uma linguagem muito flexível e pode ser usada para modelar uma ampla variedade de sistemas de software. Ela é suportada por uma ampla gama de ferramentas de modelagem de software e é amplamente adotada pela indústria de software como um padrão para modelagem de software.

A linguagem UML é formada por 14 diagramas, a seguir uma breve apresentação de cada um:

1. **Diagrama de Classe** – Descreve a estrutura estática de um sistema de software, mostrando as classes, seus atributos e métodos, e as associações entre elas. As classes representam os objetos do sistema, e as associações representam as relações entre esses objetos. Através desse diagrama é possível visualizar a hierarquia de classes, herança, polimorfismo, interfaces e dependências entre as classes.

Diagrama de classe

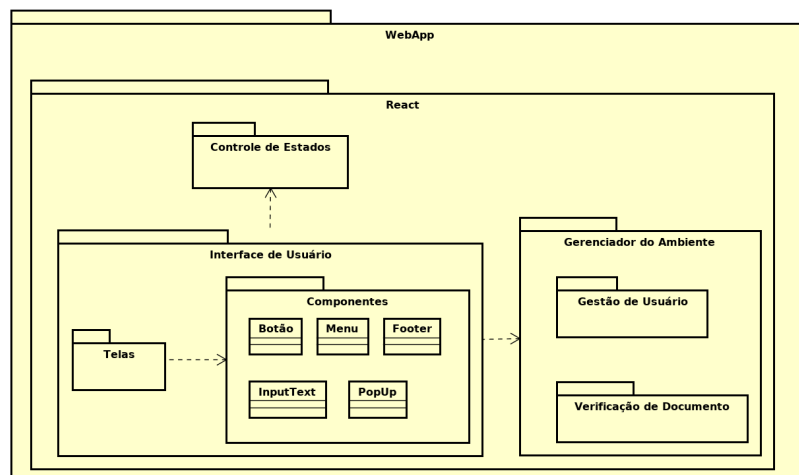


2. **Diagrama de Objeto** – Representa uma instância de um objeto em tempo de execução, mostrando seus atributos e valores. É uma variação do diagrama de classe, que permite visualizar as informações em relação a um objeto específico, ao invés de uma classe inteira. Esse diagrama é útil para entender como os objetos interagem entre si em tempo de execução.

Diagrama de objetos



3. **Diagrama de Pacotes** – Agrupa elementos de modelagem em pacotes



lógicos, mostrando a estrutura de dependência entre os pacotes. Os pacotes representam módulos do sistema e são utilizados para organizar elementos de modelagem relacionados em um único local. Esse diagrama é útil para entender a estrutura geral do sistema e suas dependências.

Diagrama de pacotes

4. **Diagrama de Componentes** – Descreve a estrutura de componentes de um sistema de software, mostrando as interfaces, portas e conexões entre eles. Os componentes representam unidades independentes de software que podem ser

reutilizadas em diferentes contextos. Esse diagrama é útil para entender a estrutura modular do sistema e como os componentes interagem entre si.

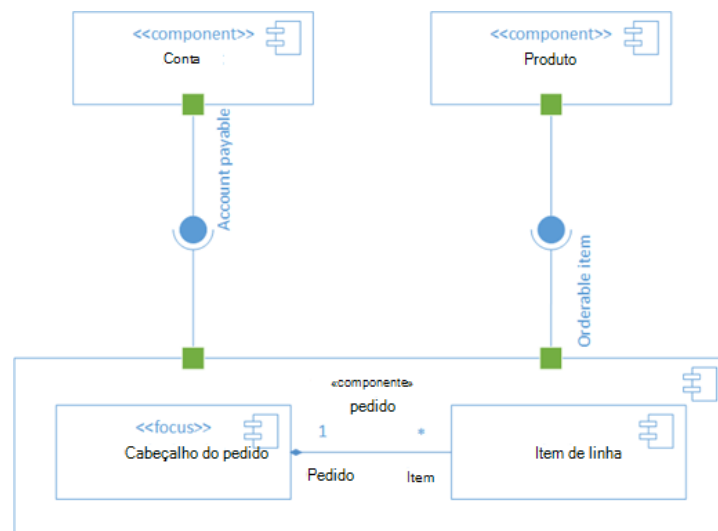
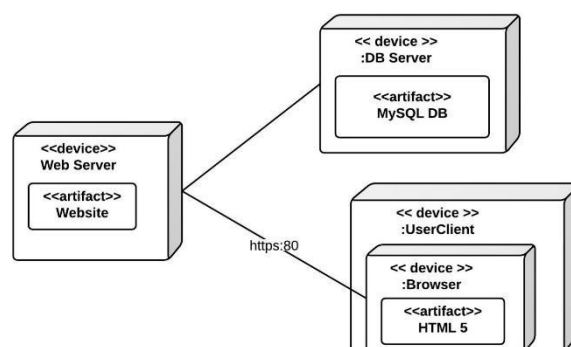


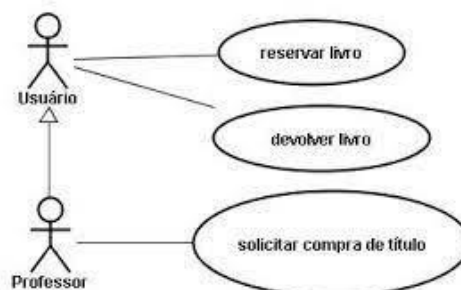
Diagrama de componente

5. **Diagrama de Implantação** - Descreve a implantação física de um sistema de software, mostrando os nós de hardware e os componentes de software que estão instalados neles. Os nós representam os servidores, estações de trabalho e dispositivos de rede, e os componentes representam os módulos de software instalados em cada nó. Esse diagrama é útil para entender a arquitetura de implantação do sistema e sua distribuição física.

Diagrama de implantação



6. **Diagrama de Caso de Uso** - Descreve a interação entre um sistema de software e seus usuários ou outros sistemas externos, mostrando as principais

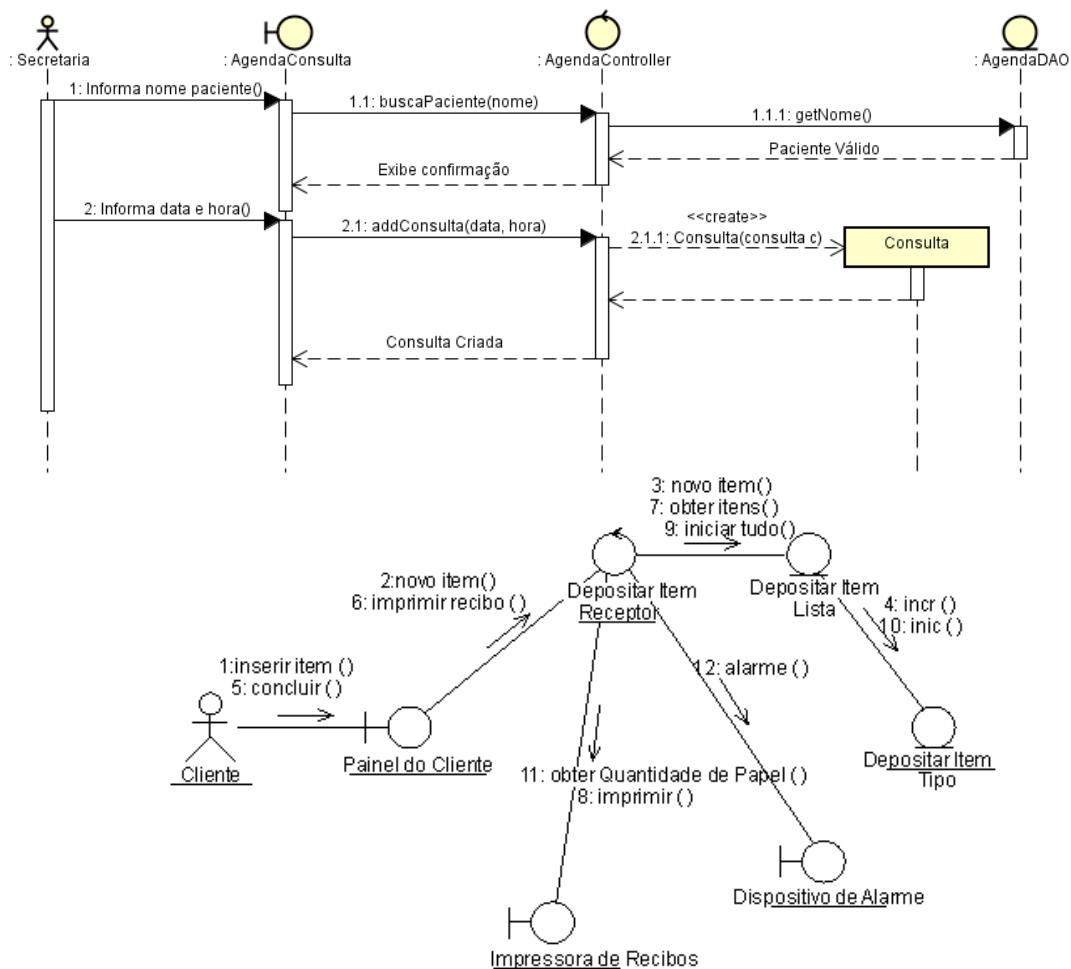


funcionalidades e fluxos de interação. Os casos de uso representam os cenários de uso do sistema, e os atores representam os usuários ou sistemas externos que interagem com o sistema. Esse diagrama é útil para entender os requisitos funcionais do sistema e seus principais fluxos de trabalho.

Diagrama de caso de uso

7. **Diagrama de Sequência** - Descreve a interação entre objetos em um determinado cenário, mostrando a ordem de mensagens trocadas entre eles. As mensagens representam as chamadas de métodos entre os objetos, e a sequência delas representa o fluxo de controle do sistema. Esse diagrama é útil para entender o comportamento dinâmico do sistema em um determinado cenário.

Diagrama de sequência



8. **Diagrama de Comunicação** - É uma variação do diagrama de sequência, que mostra as mesmas informações em um formato mais simplificado. Nesse diagrama,

as mensagens são representadas como setas direcionais, sem a indicação explícita da ordem em que elas são trocadas. Esse diagrama é útil para comunicar de forma mais clara e concisa o comportamento dinâmico

Diagrama de comunicação

9. **Diagrama de Estado** (State Diagram): usado para descrever o comportamento do sistema em diferentes estados, mostrando as transições entre eles.

Diagrama de estado



10. **Diagrama de Atividade** (Activity Diagram): usado para descrever a lógica de negócios do sistema, mostrando as atividades e as decisões que o sistema deve realizar para atingir seus objetivos.

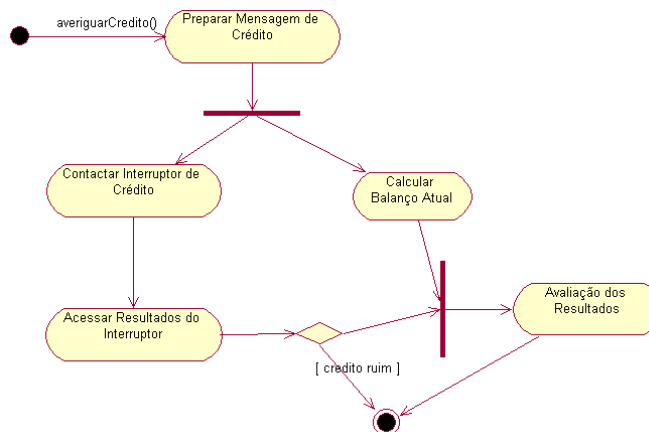
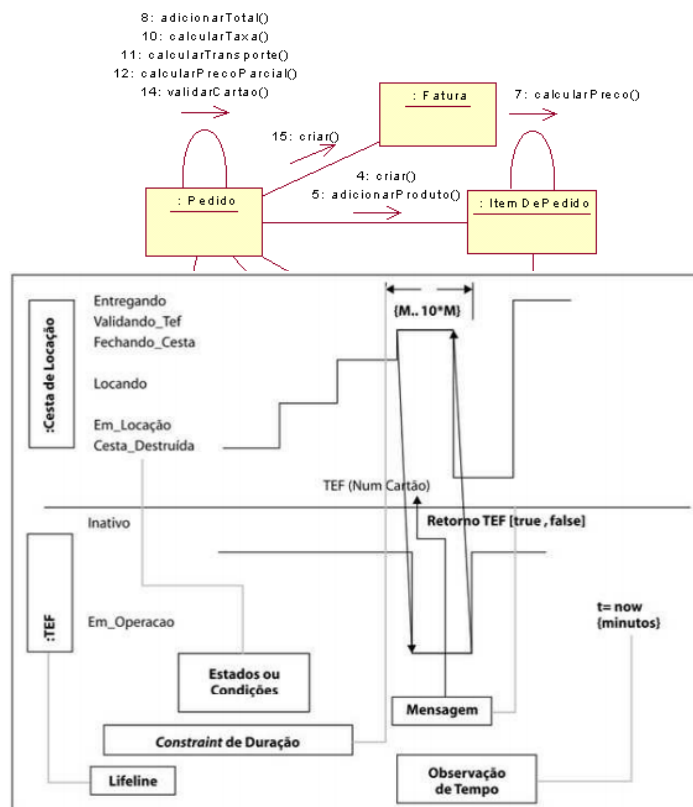


Diagrama de Atividade

11. **Diagrama de Colaboração** (Collaboration Diagram): semelhante ao Diagrama de Comunicação, mas enfatiza a colaboração entre os objetos em vez da ordem das mensagens.

Diagrama de Colaboração

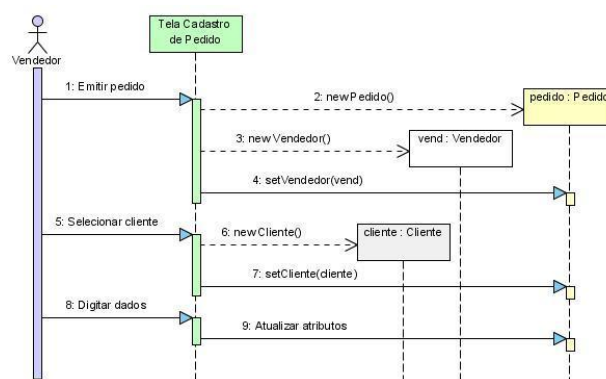


12. **Diagrama de Tempo** (Timing Diagram): usado para descrever a evolução temporal dos eventos do sistema, mostrando como eles se relacionam no tempo.

Diagrama de Tempo

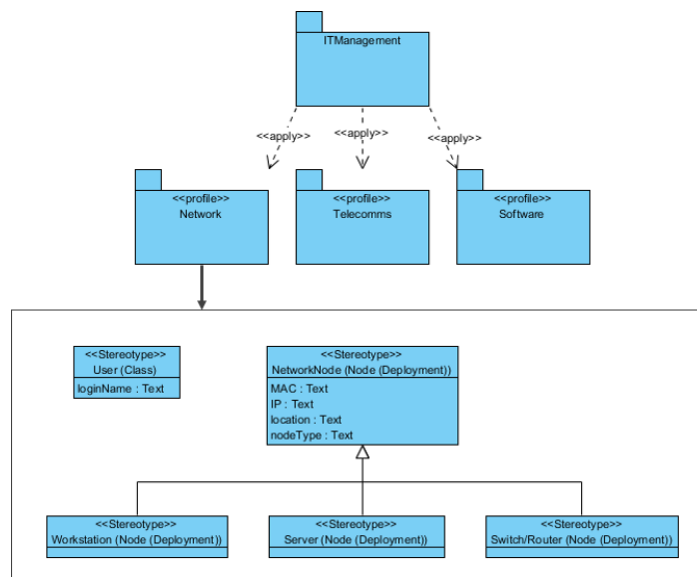
13. **Diagrama de Interação Geral** (Interaction Overview Diagram): usado para descrever a visão geral de uma interação complexa, mostrando como os diferentes elementos se relacionam de forma geral.

Diagrama de interação



14. **Diagrama de Perfis** (Profile Diagram): usado para descrever extensões personalizadas da UML, mostrando como o modelo UML é estendido para atender a requisitos específicos de modelagem.

Diagrama de perfil



Acima vimos os 14 diagramas que formam a UML, desses 14 estudaremos dois de forma mais aprofundada, o diagrama de caso de uso e diagrama de classe

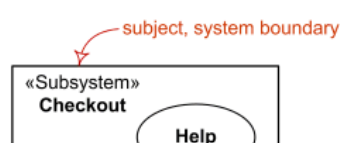
UML — Diagrama de Casos de Uso

O diagrama de Casos de Uso auxilia no levantamento dos requisitos funcionais do sistema, descrevendo um conjunto de funcionalidades do sistema e suas interações com elementos externos e entre si.

Cenários: quando falamos de casos de uso, temos que ter em mente o conceito de cenários, que seriam instâncias de casos de uso.

Um cenário pode ser compreendido como uma sequência de passos que descreve uma interação entre um usuário e o sistema.

Típico cenário de um diagrama de Casos de Uso



Em resumo, os diagramas de Casos de Uso:

- auxiliam na comunicação entre o cliente e os analistas.
- apresentam as principais funcionalidades do sistema com foco no cliente.
- descrevem cenários de interação entre as partes internas/externas de um sistema, com foco no usuário.
- é muito utilizado na fase de levantamento de requisitos.

Notação

Os principais elementos de um diagrama de casos de uso são:

- Atores: boneco com rótulo que representa um humano ou um sistema computacional.

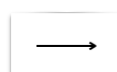


- Caso de Uso: elipse com rótulo que representa uma funcionalidade do sistema, sendo que esta pode estar estruturada em outra(s). Um caso de uso pode ser concreto, quando é iniciado diretamente por um ator, ou abstrato, quando é uma extensão de um outro caso de uso. Além disso há casos de uso primários e secundários. O primeiro representa os objetivos dos atores, já o segundo são funcionalidades do sistema que precisam existir para que este funcione corretamente.

caso de uso



- Relacionamentos: auxiliam na descrição dos casos de uso, podendo ser: entre um ator e um caso de uso, entre atores e entre casos de uso.



Relacionamento