**Diagramas estruturais**

* **Diagrama de classes**
* Diagrama de objetos
* Diagrama de componentes
* Diagrama de instalação ou de implantação
* Diagrama de pacotes
* Diagrama de estrutura composta
* Diagrama de perfil

**Diagramas comportamentais ou dinâmicos**

* Diagrama de caso de uso
* Diagrama de transição de estados ou Máquina de estados
* Diagrama de atividade

**Diagramas de interação**

* **Diagrama de sequência**
* Diagrama Visão Geral de Interação ou de interação
* **Diagrama de colaboração ou comunicação**
* **Diagrama de tempo ou temporal**
* **Estereótipos** são**mecanismos de extensibilidade** que **amplia** o vocabulário da UML, permite a criação de novos tipos de blocos de construção, por exemplo. *Estende o significado de determinado elemento em um diagrama.*
* Etiquetas valoradas - *Permite definir outras propriedades, além das pré-definidas pela linguagem, para determinados elementos de seus diagramas. ​*
* Notas explicativas - *Comenta ou esclarece alguma parte do diagrama.*
* Restrições - *Podem estender ou alterar a semântica natural de um elemento gráfico.*

Os **diagramas de atividades** são uma técnica para descrever lógica de procedimento, processo de negócio e fluxo de trabalho. Assemelha-se aos fluxogramas, mas a principal diferença é o fato dos diagramas de atividades suportarem comportamento paralelo. Um diagrama de atividades é uma série de atividades ligadas por transições, que são setas conectando cada atividade.

Os **casos de uso** são uma técnica para captar os requisitos funcionais de um sistema. Eles servem para descrever as interações típicas entre os usuários de um sistema e o próprio sistema, fornecendo uma narrativa de como o sistema é utilizado.Para requisitos funcionais usa-se Diagrama de Casos de Uso. O grau de abstração de um caso de uso diz respeito à existência ou não de menção à tecnologia a ser utilizada na descrição deste caso de uso.

* Um caso de uso **essencial**não faz menção à tecnologia a ser utilizada.
* Um caso de uso **real**apresenta detalhes da tecnologia a ser utilizada na implementação deste caso de uso.

O diagrama de caso de uso, cuja principal finalidade é auxiliar na comunicação com o usuário do sistema, deve retratar o que o sistema vai fazer, sem especificar como isso será realizado.

**Diagrama de Sequência:** enfatiza a ordem temporal das mensagens, mostra um conjunto de papéis e as mensagens enviadas e recebidas pelas instâncias. A linha tracejada vertical é a **vida** do objeto, e o retângulo estreito nessa linha representa o **tempo** que o objeto desempenha a ação.  Diagrama de Sequência teve a adição de um recurso chamado Fragmentos Combinados utilizados para modelar testes lógicos. "*Um operador de controle, ou****fragmento combinado****, é apresentado como uma****região retangular no diagrama de sequências****. Ele tem uma tag - um rótulo de texto dentro de um pequeno pentágono no canto superior esquerdo - para informar o tipo de operador de controle. O operador aplica-se às linhas da vida que atravessam. Isso é considerado o corpo do operador. Se uma linha da vida não se aplica ao operador, ou fragmento combinado, ela pode ser interrompida no topo do operador de controle retomada na base. Os tipos de controle mais comuns são os seguintes:*

* ***OPT:*** *execução opcional;*
* ***ALT****: execução condicional;*
* ***PAR****: execução paralela;*
* ***LOOP****: execução de loop, iterativa;*

*Obs.: “Há inúmeros outros tipos de operadores, mas esses são os mais úteis e recorrentes em provas de concurso.”.*

Diagrama de Colaboração (hoje conhecido como Comunicação): as informações mostradas no diagrama de comunicação são praticamente as mesmas apresentadas no diagrama de sequência, porém com um enfoque diferente, visto que este diagrama não se preocupa com a temporalidade do processo; exibe mensagens enfatizando relacionamentos.

**Diagrama de Transição de Estados**, ou Diagrama de Máquina de Estados, é uma representação do estado ou situação em que um objeto pode se encontrar no decorrer da execução de processos de um sistema. No diagrama de estado, cada região, em um estado composto, pode ter vários estados funcionando independentemente. Um estado que possui subestados (estados aninhados) é denominado estado composto.

**Diagrama de pacotes mostra, por exemplo, a decomposição do próprio modelo em unidades organizacionais e suas dependências. Agrupam elementos semanticamente próximos. Obs.: Pacotes bem estruturados são fracamente acoplados e muito coesos.**

**Diagrama de classes** - conforme cita no livro UML - Guia do usuário: "Ao especificar a visibilidade características de um classificador, geralmente você deseja ocultar todos os seus detalhes de implementação..." Ou seja, vamos do nível mais restritivo para o menos restritivo. São relacionamentos do diagrama de classes:

* **Associação simples**: seta com linha sólida. Indica que objetos de um elemento estão ligados a objetos de outro elemento. Ex.: PESSOA possui CONTA.
* **Associação por agregação:** seta de linha sólida com losango aberto. A parte existe sem o todo. Ex.: PESSOA participa de um CLUBE. Ou seja, PESSOA (parte) pode existir sem CLUBE (todo).
* **Associação por composição:** seta de linha sólida com losango fechado. O todo controla o cliclo de vida da parte. Ex.: EMPRESA possui DEPARTAMENTO. O DEPARTAMENTO (parte) não pode existir sem a EMPRESA (todo).
* **Dependência:** seta de linha tracejada. Mudanças em um elemento pode causar mudanças no outro. Ex.: GUI depende de FORMULÁRIO. A interface GUI depende da classe FORMULÁRIO, logo, mudanças na classe formulário podem afetar a interface.
* **Generalização:** seta de linha sólida com triângulo aberto. Relacionamento "é um tipo de". Ex.: POUPANÇA é um tipo de CONTA.
* **Realização:**seta de linha tracejada com triângulo aberto. Também possui a notação de bola soquete que é utilizada para modelar uma dependência e uma realização entre duas classes e uma interface.

Portanto:

* 1°. Private está relacionado a classes;
* 2°. Package está relacionado ao pacote;
* 3°. Protected está relacionado aos descendentes (herança);
* 4°. Pública está relacionada ao sistema como um todo;

**DIAGRAMA DE IMPLANTAÇÃO/IMPLEMENTAÇÃO** ->  Modela a configuração física do sistema, revelando que pedaços de software rodam em que equipamentos de hardware.

Inclui: ◦ Nós – Dispositivos (Hardware) e  Ambientes de Execução.

◦ Artefatos – Código fonte, Código binário e Executáveis, etc.

**Diagrama de estrutura composta**

* Descreve a estrutura interna (classes, objetos e interfaces) de um classificador modelando colaborações;
* Descreve estruturas de partes (instâncias contidas em outro elemento) ou instâncias interconectadas por portas (ponto de interação entre os elementos);
* Usado para associar o diagrama de objetos com o diagrama de classes/interfaces em tempo de execução;
* Notação de um **círculo tracejado** (colaboração para execução de determinada atividade) que contém partes de **diagramas de objetos** e partes de **classes estruturadas**;

**Reforço:**

Nos modelos UML, **um diagrama de estrutura composta mostra a estrutura interna dos classificadores estruturados utilizando peças, portas e conectores**. Um classificador estruturado define a implementação de um classificador e pode incluir uma classe, um componente ou um nó de implementação. Você pode utilizar o diagrama de estrutura composta para mostrar os detalhes internos de um classificador e descrever os objetos e funções que trabalham juntos para executar o comportamento do classificador contido.

Quando a banca fala em **diagrama de interação** ela quer se referir ao **Diagrama de Interação Geral**. De fato pode haver uma combinação do Diagrama de Interação Geral com os diagramas de sequência e de comunicação para facilitar o entendimento dos requisitos do sistema.

Visões UML(5)

* Visão de Caso de USO
* Visão de Projeto/ Lógica
* Visão de Implementação ou Componente
* Visão de Processo/ Concorrência
* Visão de Implantação

**"Composição é uma variação da agregação, é um vínculo mais forte entre os objetos-todo e os objetos-parte. Os objetos-parte têm de estar associados a um único objeto-todo."**

Restrição sobre generalização/herança:

* **Sobreposição (overlapping)**: Um objeto da superclasse pode pertencer **simultaneamente a mais** de uma subclasse.
* **Disjuntiva (disjoint)**: superclasses podem se especializar em **apenas** uma subclasse.
* **Completa**: é uma generalização que já foram especificadas todas as subclasses.
* **Incompleta**: é uma generalização que existe a possibilidade de uma outra especialização, ou seja, caso um objeto da superclasse pode não pertencer a nenhuma das subclasses.