✅ 5.1 – Diagrama de Classes

Perguntas Básicas

1. Qual é a função do diagrama de classes?

Representar a **estrutura estática** de um sistema, mostrando **classes**, seus **atributos**, **métodos** e os **relacionamentos** entre elas.

---

2. Para que este diagrama serve?

Serve para:

- Modelar a base estrutural do sistema;
- Apoiar o desenvolvimento orientado a objetos;
- Organizar entidades, suas responsabilidades e interações;
- Auxiliar na documentação e comunicação entre desenvolvedores.

---

3. Explique como modelar uma classe e quais são seus elementos.

Uma classe é modelada geralmente em **2 ou 3 seções**:

1. **Nome da classe**
2. **Atributos**
3. **Operações (métodos)**

Elementos adicionais possíveis:

- visibilidade (+, -, #)

- tipos de dados
- multiplicidade em atributos complexos
- estereótipos
- anotações

---

4. O que é visibilidade, seus tipos e comportamentos?

Define **quem pode acessar** atributos ou métodos da classe.

Tipos:

- **+ Público:** acessível por qualquer classe.
- **- Privado:** acessível apenas dentro da própria classe.
- **# Protegido:** acessível na classe e em subclasses.
- **~ Pacote:** acessível apenas dentro do mesmo pacote.

---

5. O que são associações entre classes e como identificá-las?

Relacionamentos estruturais que indicam **como classes se conectam**.
Identificamos quando:

- uma classe depende de informação/ação da outra;
- uma classe contém objetos da outra;
- objetos interagem de forma estável.

---

6. O que são estereótipos? Explique e exemplifique.

Estereótipos são **rótulos semânticos** usados para especializar classes na modelagem.

- **<<entity>>**
  Representa entidades persistentes ou independentes de interface.
  Ex.: Cliente, Produto.

- **<<persistent>>**
  Objetos gravados em banco de dados.

- **<<boundary>>**
  Elementos de interface com o usuário.
  Ex.: TelaLogin, FormCadastro.

- **<<control>>**
  Gerencia regras de negócio e fluxo.
  Ex.: ControladorPedido.

Perguntas Medianas

1. Associações (comportamentos)

- **a) Unária / Reflexiva:**
  Uma classe se relaciona com ela mesma. Ex.: Pessoa → “é amiga de” → Pessoa.

- **b) Binária:**
  Relação entre duas classes. É o tipo mais comum.

- **c) Ternária / N-ária:**
  Envolve três ou mais classes simultaneamente (mesma relação).

- **d) Agregação:**
  Relação “todo–parte” fraca (partes podem existir sem o todo).
  Ex.: Sala agrega Cadeiras.

- **e) Composição:**
  “Todo–parte” forte (partes não existem sem o todo).
  Ex.: Casa compõe Cômodos.

- **f) Especialização / Generalização:**
  Herança: superclasse → subclasses.

- **g) Dependência:**
  Uma classe **usa/depende** temporariamente de outra.

- **h) Realização:**
  Interface → Classe que implementa.

- **i) Associativa:**
  Quando uma associação tem **propriedades próprias**, criando uma classe associativa.

- **j) Restrição:**
  Condições aplicadas a relacionamentos.
  Ex.: {xor}, {ordered}, {complete}.

---

2. Exemplos de multiplicidade

- **a) 0…1:**
  Zero ou um elemento possível.
  Ex.: Pessoa —tem→ Passaporte.

- **b) 1…1:**
  Exatamente um.

- **c) 0…*:**
  Nenhum ou muitos.

- **d) *...*:**
  Muitos para muitos.

- **e) 1…*:**
  Pelo menos um.

- **f) 3…5:**
  Entre 3 e 5 elementos obrigatoriamente.

Perguntas Específicas

1. Tipos de restrições

- **a) Completa:**
  A generalização cobre **todas as subclasses possíveis**.

- **b) Incompleta:**
  Podem existir subclasses adicionais não representadas.

- **c) Separada / Disjunta:**
  Um objeto só pode pertencer a **uma** das subclasses.

- **d) Sobreposta:**
  Um objeto pode pertencer a **mais de uma** subclasse simultaneamente.

✅ 5.2 – Diagrama de Objetos

1. O que é e como se diferencia do diagrama de classes?

É um diagrama que mostra **instâncias concretas** (objetos) e **seus valores no momento**.
Diferença:

- Diagrama de classes → estrutura **abstrata**.

- Diagrama de objetos → **exemplos reais** das classes em execução.

✅ 5.3 – Diagrama de Sequência

Perguntas Básicas

1. Função do diagrama de sequência

Representar o **fluxo de mensagens ao longo do tempo** entre objetos/atores.

---

2. Identificação dos elementos (1 a 7)

Sem imagem enviada, mas geralmente inclui:

1. Atores
2. Objetos
3. Linha de vida
4. Foco de controle
5. Mensagens síncronas
6. Mensagens assíncronas
7. Retornos

(Se quiser, posso montar exatamente conforme sua imagem — é só enviá-la.)

---

3. Como sinalizar um objeto no diagrama?

Formato:
nomeObjeto : Classe

Ex.:
`c1 : Cliente`

---

4. Função da linha de vida

Representa o **tempo de existência** do objeto durante a execução.
Desenho:
Uma linha vertical tracejada abaixo do objeto.

---

5. Foco de controle

Representado por um **retângulo estreito** sobre a linha de vida.
Indica quando o objeto está **ativo processando** uma ação.

Perguntas Medianas

1. Mensagens / estímulos

- **a) ator → ator:** comunicação entre papéis humanos.
- **b) ator → objeto:** inicia um processo no sistema.
- **c) objeto → objeto:** colaboração interna do sistema.

---

2. Desenho das setas

- **Seta cheia + ponta cheia:** chamada de método síncrona.
- **Linha tracejada + ponta aberta:** retorno.
- **Seta aberta:** chamada assíncrona.

---

3. Diferença entre mensagem de retorno e autochamada

- **Retorno:** objeto devolve resultado para quem chamou.
- **Autochamada:** o **mesmo objeto** chama um de seus métodos internos.

Perguntas Específicas

1. Como identificar condição em mensagem?

Coloca-se entre colchetes:

Ex.:

```
[saldo >= valor] debitar()
```

---

2. Passos para ensinar a construir um diagrama de sequência

1. Identificar atores.
2. Identificar objetos envolvidos.
3. Definir o cenário/metodologia.
4. Colocar objetos na ordem horizontal.
5. Desenhar linhas de vida.
6. Inserir mensagens na ordem temporal.
7. Incluir retornos e focos de controle.
8. Revisar e validar.

---

3. Diagramas hipotéticos (descrição)

Se quiser, posso gerar os diagramas visuais também!
Aqui vai o texto:

Alterar Produto

Ator Admin → TelaProduto → ControladorProduto → Produto → BancoDados → Produto → Controlador → Tela.

Cancelar Produto

Fluxo semelhante, substituindo operação por cancelar().

✅ 5.4 – Diagrama de Componentes

Perguntas Básicas

1. Função

Modelar a **arquitetura física** do sistema: módulos, bibliotecas e dependências.

---

2. O que é um componente?

Unidade modular substituível do sistema.
Representação: retângulo com ícone de “caixa”.

---

3. Além do nome, o que pode compor a representação?

- Interfaces fornecidas
- Interfaces requeridas
- Portas
- Estereótipos
- Dependências

---

4. O que é uma dependência?

Relação indicando que um componente **usa/requer** outro.

---

5. O que é e como representar uma interface?

Conjunto de serviços expostos.
Representação:

- Bola (lollipop) → interface fornecida.
- Semicírculo → interface requerida.

Perguntas Medianas

Estereótipos:

- **<<executable>>** executável.
- **<<library>>** biblioteca.
- **<<table>>** tabela de BD.
- **<<document>>** arquivo de documentação.
- **<<file>>** arquivo físico.

---

2. Partes da seta de dependência

- **Base:** origem (quem depende).
- **Ponta tracejada:** destino (de quem depende).

Perguntas Específicas

1. Linha cheia vs. tracejada

- **Cheia:** relação estrutural.
- **Tracejada:** dependência.

✅ 5.5 – Diagrama de Implantação

Perguntas Básicas

1. Para que serve?

Modelar a **distribuição física** do sistema: servidores, máquinas, dispositivos.

---

2. Quando é útil?

Em projetos que envolvem:

- infraestrutura
- comunicação entre máquinas
- arquitetura distribuída

---

3. O que são nós?

Elementos físicos que executam componentes.
Ex.: Servidor, smartphone, roteador.

---

4. Nós com componentes

Um nó que **contém** componentes de software implantados nele.

---

5. Associação física entre nós

Representada com conexões (linhas) indicando comunicação:

- cabos
- redes
- protocolos

✅ 5.6 – Diagrama de Comunicação / Colaboração

1. O que é e para que serve?

Representa a comunicação entre objetos enfatizando a **estrutura** da colaboração.

---

2. Diferença e semelhanças com diagrama de sequência

Semelhança: ambos mostram troca de mensagens.
Diferença:

- sequência → enfoque no **tempo**.
- comunicação → enfoque na **estrutura/ligação**.

---

3. Componentes

- Objetos
- Links
- Mensagens numeradas
- Sequência de chamadas

---

4. Símbolos boundary, control, model

Posso desenhar se quiser em imagem — apenas pedir!
Representação textual:

- `obj : Classe <<boundary>>`

- `obj : Classe <<control>>`
- `obj : Classe <<entity>>`

✅ 5.7 – Diagrama de Atividades

Perguntas Básicas

1. O que é e funcionalidade?

Diagrama de fluxo que mostra **processos**, decisões e paralelismos.

---

2. 4 componentes básicos

- Estado de ação (retângulo arredondado)
- Decisão (losango)
- Inicial (círculo cheio)
- Final (círculo com borda dupla)

---

3. Sub-atividade

Estado que representa uma atividade detalhada em outro diagrama (zoom).
Usado para simplificar.

---

4. Fluxo de objetos

Mostra **objetos passando** entre atividades.
Desenho: seta com objeto rotulado.

---

5. Símbolos de envio/recebimento de eventos

Posso desenhar; textual:

- Envio: envelope aberto
- Recebimento: envelope fechado

---

6. Barras de sincronização

Representam **paralelismos**: fork/join.
Linha grossa horizontal.