

# Sequence Diagram

**Diagrama UML** que representa **como os objetos interagem entre si** através da troca de mensagens ao longo do tempo.

---

## 1 O Que É um Sequence Diagram?

📌 **Mostra a comunicação entre objetos do sistema** numa **ordem cronológica**.

📌 **Ajuda a visualizar o fluxo de chamadas de métodos** e como as classes interagem.

📌 **Foca-se no comportamento dinâmico do sistema**, diferente do **Diagrama de Classes**, que mostra apenas a estrutura estática.

### ◆ Elementos Principais:

✓ **Objetos (Participantes)** → Representados por **retângulos no topo** do diagrama.

✓ **Lifeline (Linha de Vida)** → Representa a **existência do objeto ao longo do tempo** (linha vertical).

✓ **Mensagens** → Representam **chamadas de métodos e respostas** entre os objetos.

✓ **Blocos de Ativação** → Retângulos na linha de vida que indicam **quando um objeto está ativo/a executar uma ação**.

---

## 2 Para Que Serve um Sequence Diagram?

✓ **Facilita a compreensão do fluxo do sistema.**

✓ **Ajuda a projetar interações entre objetos antes da implementação.**

✓ **Garante que os objetos certos estão responsáveis pelas ações certas.**

✓ **Serve como documentação para a equipa de desenvolvimento.**

---

## 3 Como Criar um Sequence Diagram?

### 📌 Passo 1: Identificar os Objetos Importantes

- Normalmente, vêm do **Diagrama de Classes**.

- Incluem **atores externos (usuários, sistemas)** e **classes internas do sistema**.

### 📌 Passo 2: Definir a Ordem das Interações

- Identificar **quem inicia a interação**.
- Listar as mensagens e a **ordem cronológica das chamadas**.

### 📌 Passo 3: Representar as Mensagens

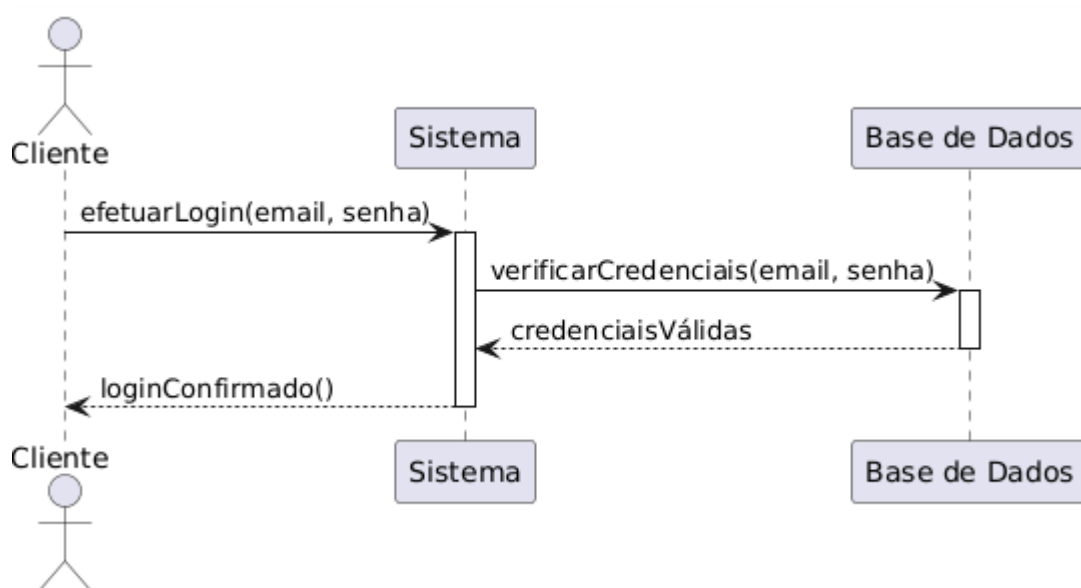
- **Chamadas de métodos** → Setas **sólidas** (→).
- **Mensagens assíncronas** (eventos) → Setas **tracejadas** (→).
- **Retornos** → Setas **tracejadas para trás** (←).

### 📌 Passo 4: Usar Blocos de Ativação

- Indicam **quando um objeto está ativo** e a processar uma mensagem.
- Se uma classe chama outra, o **bloco de ativação da classe original continua** até receber a resposta.

## 4 Exemplo Visual de Sequence Diagram

💡 **Cenário:** Um **Cliente** faz login num sistema. O **Sistema** valida as credenciais na **Base de Dados** e responde ao Cliente.



### Diagrama de Sequência (Desenho)

### ◆ Explicação:

- ✓ O **Cliente** envia a mensagem `efetuarLogin()` para o **Sistema**.
- ✓ O **Sistema** chama `verificarCredenciais()` na **Base de Dados**.
- ✓ A **Base de Dados** responde com `credenciaisVálidas`.
- ✓ O **Sistema** envia `loginConfirmado()` de volta para o **Cliente**.

## 5 Dicas Para Criar um Sequence Diagram Bem Feito

### ✓ 1. Usa Nomes Claros para as Mensagens

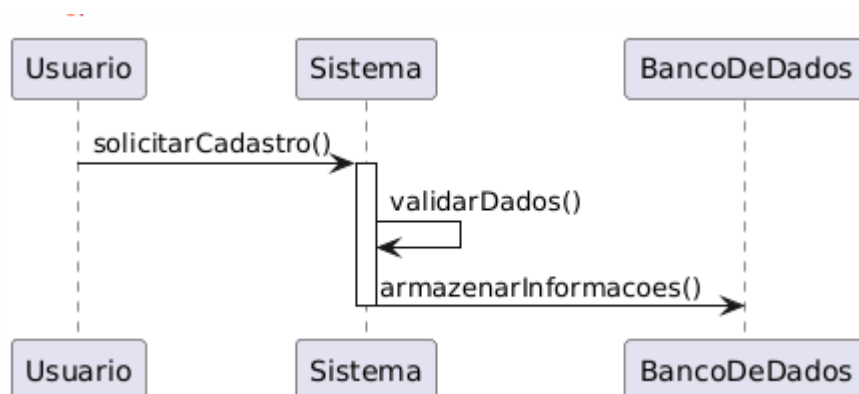
- Preferir **verbos no infinitivo + objeto** ( `validarSenha()` , `processarPagamento()` ).
- Evitar mensagens genéricas como `fazerCoisa()` ou `enviar()` que não explicam nada.

### ✓ 2. Mantém o Diagrama Simples e Lógico

- Se um **diagrama estiver muito complexo**, divide em **múltiplos diagramas menores**.
- Concentra-te no **fluxo principal**, sem excesso de detalhes desnecessários.

### ✓ 3. Usa Blocos de Ativação para Representar Processamento

- Sempre que um objeto **chama um método**, cria um **bloco de ativação** na linha de vida.



📌 O Sistema só termina a ativação depois de armazenar os dados.

💡 **Cenário:**

- Um **Cliente** faz uma compra.
- O sistema percorre uma **lista de produtos** e calcula o preço total.
- Cada produto chama a função **calcularDesconto()** antes de somar ao total.

