

ATIVIDADE	UML	1
1. DIAGRAMAS ESTRUTURAIS		3
1.1. DIAGRAMA DE CLASSE		3
1.1.1. Principais Elementos do Diagrama de Classes.....		3
1.1.1.1. Classe		3
1.1.1.2. Relacionamentos entre Classes		3
1.1.1.3. Visibilidade dos Atributos e Métodos		3
1.1.2. Exemplo de Diagrama de Classes		3
1.1.2.1. Cliente		3
1.1.2.2. ContaBancária		3
1.1.2.3. ContaCorrente.....		4
1.1.2.4. ContaPoupança		4
1.1.2.5. Banco.....		4
1.1.2.6. Diagrama		4
1.2. DIAGRAMA DE OBJETOS.....		5
1.2.1. Exemplo de Diagrama de Objetos.....		5
1.2.1.1. Cliente		5
1.2.1.2. ContaCorrente.....		5
1.2.1.3. ContaPoupança		5
1.2.1.4. diagrama		5
1.3. DIAGRAMA DE COMPONENTES		5
1.3.1. Exemplo de Diagrama de Componentes.....		5
1.3.1.1. Banco.....		6
1.3.1.2. Aplicativo Móvel		6
1.3.1.3. Servidor Web		6
1.3.1.4. Banco de Dados.....		6
1.4. DIAGRAMA DE IMPLANTAÇÃO		6
1.4.1. Exemplo de Diagrama de Implantação.....		6
1.4.1.1. Servidor de Aplicação.....		6
1.4.1.2. Servidor de Banco de Dados		6
1.4.1.3. Cliente Web.....		6
1.4.1.4. Cliente Mobile.....		7
1.5. DIAGRAMA DE ESTRUTURA COMPOSTA.....		7
1.5.1. Exemplo de Diagrama de Estrutura Composta		7
1.5.1.1. ContaBancaria		7
1.5.1.2. Cliente		7
1.5.1.3. SaldoConta		7
1.5.1.4. Transacao		7
1.6. DIAGRAMA DE PACOTES		7

1.6.1. Exemplo de Diagrama de Pacotes	7
1.6.1.1. Pacote Cliente	7
1.6.1.2. Pacote Transacoes	8
1.6.1.3. Pacote Servicos	8
1.6.1.4. Pacote Infraestrutura	8
2. DIAGRAMAS COMPORTAMENTAIS.....	8
2.1. DIAGRAMA DE CASOS DE USO	8
2.1.1. Exemplo de Diagrama de Casos de Uso	8
2.1.1.1. Atores.....	8
2.1.1.2. Casos de Uso	8
2.1.1.3. Relacionamentos.....	8
2.1.1.4. Diagrama	9
2.2. DIAGRAMA DE SEQUÊNCIA	9
2.2.1. Exemplo de Diagrama de Sequência	9
2.2.1.1. Objetos.....	9
2.2.1.2. Mensagens	9
2.2.1.3. Diagrama	10
2.3. DIAGRAMA DE COMUNICAÇÃO	10
2.3.1. Exemplo de Diagrama de Comunicação	10
2.3.1.1. Objetos.....	10
2.3.1.2. Mensagens e Conexões.....	10
2.4. DIAGRAMA DE ESTADOS	10
2.4.1. Exemplo de Diagrama de Estados	10
2.4.1.1. Estados.....	11
2.4.1.2. Transições	11
2.5. DIAGRAMA DE ATIVIDADES	11
2.5.1. Exemplo de Diagrama de Atividades.....	11
2.5.1.1. Atividades.....	11
2.5.1.2. Diagrama	11
2.6. DIAGRAMA DE TEMPO.....	12
2.6.1. Exemplo de Diagrama de Tempo	12
2.6.1.1. Objetos e Estados.....	12
2.6.1.2. Linha Temporal	12
2.7. DIAGRAMA DE VISÃO GERAL DE INTERAÇÃO	12
2.7.1. Exemplo de Diagrama de Visão Geral de Interação	12
2.7.1.1. Atividades e Interações.....	12
2.8. DIAGRAMA DE CASOS DE USO DE NEGÓCIO	13
2.8.1. Exemplo de Diagrama de Casos de Uso de Negócio	13
2.8.1.1. Atores.....	13
2.8.1.2. Casos de Uso	13
2.8.1.3. Relacionamentos.....	13

1. DIAGRAMAS ESTRUTURAIS

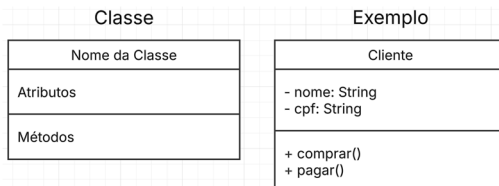
1.1. DIAGRAMA DE CLASSE

O Diagrama de Classes é um diagrama estrutural da UML que representa a estrutura estática de um sistema, detalhando:

- Classes
- Atributos
- Métodos
- Relacionamentos entre as classes (associação, herança, agregação, composição, etc.).

1.1.1. Principais Elementos do Diagrama de Classes

1.1.1.1. Classe



1.1.1.2. Relacionamentos entre Classes

- Associação (linha simples): Relacionamento comum entre classes.
- Herança (seta vazia): Representa uma especialização (classe filha herda da classe pai).
- Agregação (losango vazio): Uma classe pode conter outra, mas elas podem existir separadamente.
- Composição (losango preenchido): Uma classe depende da outra, ou seja, se uma for deletada, a outra também será.

1.1.1.3. Visibilidade dos Atributos e Métodos

- Público (+): Pode ser acessado por qualquer classe.
- Privado (-): Só pode ser acessado dentro da própria classe.
- Protegido (#): Pode ser acessado pela própria classe e pelas classes filhas.

1.1.2. Exemplo de Diagrama de Classes

Se fosse modelado um sistema bancário, um diagrama de classes básico poderia ter:

1.1.2.1. Cliente

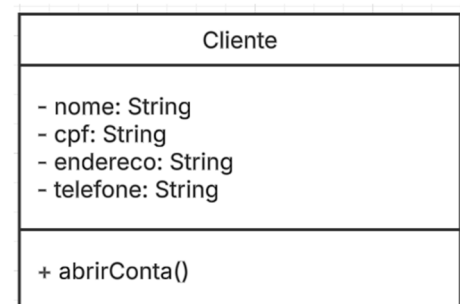
Classe Cliente (Representa o cliente do banco)

Atributos:

1. nome: String → Nome completo do cliente
2. cpf: String → CPF do cliente (documento único)
3. endereco: String → Endereço residencial
4. telefone: String → Número de telefone

Métodos:

1. abrirConta() → Cria uma nova conta para o cliente



1.1.2.2. ContaBancária

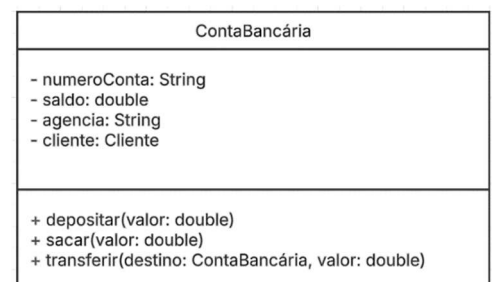
Classe ContaBancária (Classe base para todas as contas)

Atributos:

1. numeroConta: String → Número da conta
2. saldo: double → Saldo disponível
3. agencia: String → Agência bancária vinculada
4. cliente: Cliente → Ligado com a classe Cliente

Métodos:

1. depositar(valor: double) → Adiciona dinheiro à conta
2. sacar(valor: double) → Retira dinheiro da conta
3. transferir(destino: ContaBancária, valor: double) → Realiza transferência entre contas



1.1.2.3. ContaCorrente

Classe ContaCorrente (Herdando de ContaBancária)

Atributos adicionais:

1. limiteChequeEspecial: double → Limite de crédito da conta
2. taxaManutencao: double → Taxa cobrada mensalmente

ContaCorrente
+ limiteChequeEspecial: double + taxaManutencao: double
+ cobrarTarifa()

Métodos adicionais:

1. cobrarTarifa() → Aplica a taxa de manutenção mensal

1.1.2.4. ContaPoupança

Classe ContaPoupança (Herdando de ContaBancária)

Atributos adicionais:

1. taxaJuros: double → Percentual de juros aplicado ao saldo

ContaPoupança
+ taxaJuros: double
+ renderJuros()

Métodos adicionais:

1. renderJuros() → Aplica juros ao saldo da conta]

1.1.2.5. Banco

Atributos:

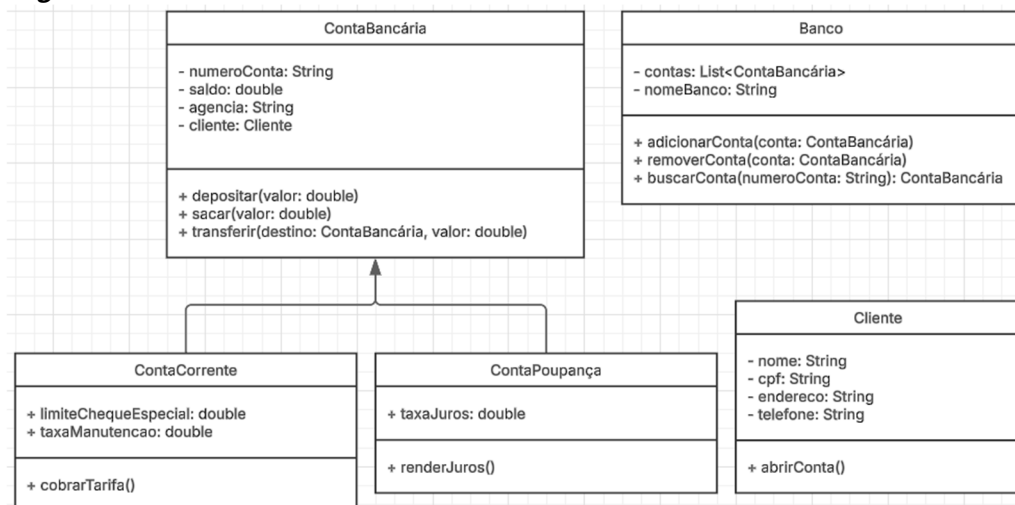
1. contas: List<ContaBancária> → Lista de todas as contas do banco
2. nomeBanco: String → Nome do banco

Métodos:

1. adicionarConta(conta: ContaBancária) → Adiciona uma conta ao banco
2. removerConta(conta: ContaBancária) → Remove uma conta do banco
3. buscarConta(numeroConta: String): ContaBancária → Encontra uma conta pelo número

Banco
- contas: List<ContaBancária> - nomeBanco: String
+ adicionarConta(conta: ContaBancária) + removerConta(conta: ContaBancária) + buscarConta(numeroConta: String): ContaBancária

1.1.2.6. Diagrama



1.2. DIAGRAMA DE OBJETOS

O Diagrama de Objetos é uma representação dinâmica do Diagrama de Classes, mostrando instâncias específicas das classes em um determinado momento da execução do sistema. Ele apresenta objetos e seus respectivos valores atribuídos, permitindo uma visualização concreta das relações entre os elementos do sistema.

1.2.1. Exemplo de Diagrama de Objetos

A seguir, um exemplo baseado em um sistema bancário, mostrando instâncias de classes em um determinado instante:

1.2.1.1. Cliente

Objeto Cliente (Representa um cliente do banco)

Atributos:

1. nome: String → "João Silva"
2. cpf: String → "123.456.789-00"
3. endereco: String → "Rua das Palmeiras, 123"
4. telefone: String → "(41) 99999-9999"

```
Cliente
nome: String → "João Silva"
cpf: String → "123.456.789-00"
endereco: String → "Rua das Palmeiras, 123"
telefone: String → "(41) 99999-9999"
```

1.2.1.2. ContaCorrente

Objeto ContaCorrente (Herdando de ContaBancária)

Atributos:

1. numeroConta: String → "0001-1"
2. saldo: double → 1500.00
3. limiteChequeEspecial: double → 500.00
4. taxaManutencao: double → 12.90
5. agencia: String → "1234"

```
ContaCorrente
numeroConta: String → "0001-1"
saldo: double → 1500.00
limiteChequeEspecial: double → 500.00
taxaManutencao: double → 12.90
agencia: String → "1234"
```

1.2.1.3. ContaPoupança

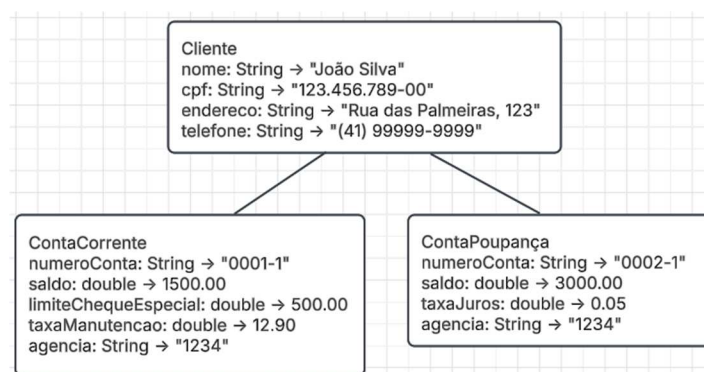
Objeto ContaPoupança (Herdando de ContaBancária)

Atributos:

1. numeroConta: String → "0002-1"
2. saldo: double → 3000.00
3. taxaJuros: double → 0.05
4. agencia: String → "1234"

```
ContaPoupança
numeroConta: String → "0002-1"
saldo: double → 3000.00
taxaJuros: double → 0.05
agencia: String → "1234"
```

1.2.1.4. diagrama



1.3. DIAGRAMA DE COMPONENTES

O Diagrama de Componentes é um diagrama estrutural da UML que representa os componentes físicos do sistema, como módulos, bibliotecas e arquivos de implementação, além de suas interações e dependências. Ele auxilia no planejamento da arquitetura do software, garantindo uma melhor organização dos elementos do sistema.

1.3.1. Exemplo de Diagrama de Componentes

A seguir, um exemplo baseado em um sistema bancário, mostrando seus principais componentes e relacionamentos:

1.3.1.1. Banco

Componente Banco (Gerencia todas as contas e clientes)

Interfaces:

1. Gerenciar Contas
2. Gerenciar Clientes
3. Processar Transações

1.3.1.2. Aplicativo Móvel

Componente Aplicativo Móvel (Interface para usuários)

Dependências:

1. Conexão com Banco
2. Autenticação de Usuário

1.3.1.3. Servidor Web

Componente Servidor Web (Intermedia requisições entre cliente e banco)

Interfaces:

1. API RESTful
2. Gerenciamento de Sessões

1.3.1.4. Banco de Dados

Componente Banco de Dados (Armazena informações do sistema)

Tabelas Principais:

1. Clientes
2. Contas Bancárias
3. Transações

1.4. DIAGRAMA DE IMPLANTAÇÃO

O Diagrama de Implantação representa a distribuição física dos componentes do sistema em hardware, incluindo servidores, dispositivos e conexões entre eles. Ele auxilia na visualização da infraestrutura necessária para a execução do sistema.

1.4.1. Exemplo de Diagrama de Implantação

A seguir, um exemplo baseado em um sistema bancário, mostrando sua distribuição em hardware:

1.4.1.1. Servidor de Aplicação

Componente responsável pelo processamento das requisições e lógica do sistema

Atributos:

1. processador: String → "Intel Xeon 3.5GHz"
2. memóriaRAM: int → 32GB
3. sistemaOperacional: String → "Linux Ubuntu Server"

1.4.1.2. Servidor de Banco de Dados

Componente que armazena as informações do sistema bancário

Atributos:

1. tipoBanco: String → "PostgreSQL"
2. armazenamento: int → 2TB
3. replicacao: boolean → true

1.4.1.3. Cliente Web

Dispositivo do usuário acessando o sistema via navegador

Atributos:

1. navegador: String → "Google Chrome"
2. sistemaOperacional: String → "Windows 10"

1.4.1.4. Cliente Mobile

Aplicativo instalado no smartphone do usuário

Atributos:

1. sistemaOperacional: String → "Android 12"
2. resolucaoTela: String → "1080x2400"

1.5. DIAGRAMA DE ESTRUTURA COMPOSTA

O Diagrama de Estrutura Composta detalha a organização interna de uma classe ou componente, mostrando seus elementos internos e como eles se relacionam. Ele é útil para modelar a composição de objetos dentro de um sistema.

1.5.1. Exemplo de Diagrama de Estrutura Composta

A seguir, um exemplo baseado em um sistema bancário:

1.5.1.1. ContaBancaria

Classe que representa uma conta bancária e sua composição interna

Partes:

1. titular: Cliente
2. saldo: SaldoConta
3. transacoes: Lista<Transacao>

1.5.1.2. Cliente

Parte interna que representa o titular da conta

Atributos:

1. nome: String
2. cpf: String
3. endereco: String

1.5.1.3. SaldoConta

Parte que gerencia o saldo disponível na conta

Atributos:

1. valorSaldo: double
2. dataAtualizacao: Date

1.5.1.4. Transacao

Parte que representa uma operação financeira realizada na conta

Atributos:

1. tipo: String
2. valor: double
3. data: Date

1.6. DIAGRAMA DE PACOTES

O Diagrama de Pacotes organiza os elementos do sistema em grupos lógicos, facilitando a modularização e a gestão da complexidade do projeto. Ele mostra dependências entre pacotes e a organização estrutural do sistema.

1.6.1. Exemplo de Diagrama de Pacotes

A seguir, um exemplo baseado em um sistema bancário, organizando suas classes em pacotes:

1.6.1.1. Pacote Cliente

Contém classes relacionadas aos usuários do sistema

Classes:

1. Cliente
2. ContaBancaria

1.6.1.2. Pacote Transacoes

Agrupar as classes que tratam das operações financeiras

Classes:

1. Transacao
2. Pagamento
3. Transferencia

1.6.1.3. Pacote Servicos

Contém classes responsáveis por fornecer serviços ao sistema

Classes:

1. Autenticacao
2. GerenciadorContas
3. ProcessadorPagamentos

1.6.1.4. Pacote Infraestrutura

Agrupar componentes relacionados à persistência de dados e comunicação

Classes:

1. BancoDeDados
2. ConexaoAPI
3. Logger

2. DIAGRAMAS COMPORTAMENTAIS

2.1. DIAGRAMA DE CASOS DE USO

O Diagrama de Casos de Uso representa as interações entre os usuários (atores) e o sistema, demonstrando as funcionalidades oferecidas e como os atores as utilizam. Ele é útil para entender os requisitos funcionais do sistema.

2.1.1. Exemplo de Diagrama de Casos de Uso

A seguir, um exemplo baseado em um sistema bancário, mostrando os principais casos de uso:

2.1.1.1. Atores

1. Cliente
2. Gerente
3. Sistema Bancário

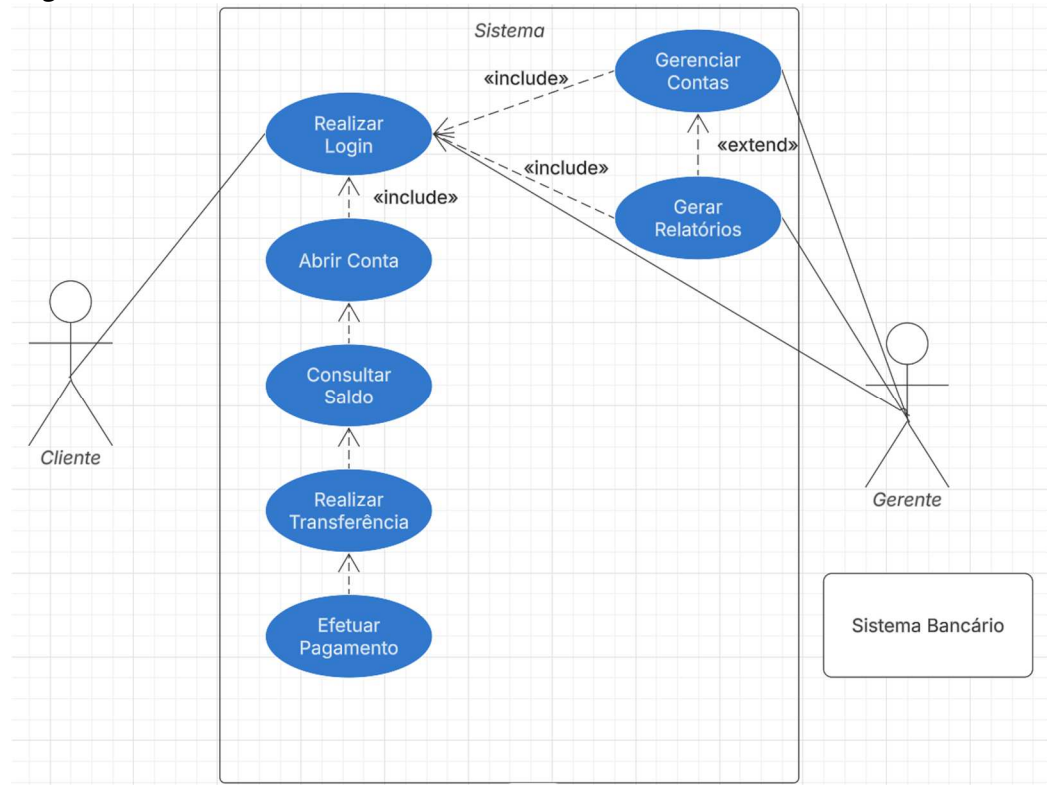
2.1.1.2. Casos de Uso

1. Abrir Conta (Cliente)
2. Realizar Login (Cliente, Gerente)
3. Consultar Saldo (Cliente)
4. Realizar Transferência (Cliente)
5. Efetuar Pagamento (Cliente)
6. Gerenciar Contas (Gerente)
7. Gerar Relatórios (Gerente)

2.1.1.3. Relacionamentos

1. O caso de uso "Realizar Login" é incluído por todos os outros casos de uso do Cliente e do Gerente («include»).
2. O caso de uso "Gerar Relatórios" estende "Gerenciar Contas" («extend»).

2.1.1.4. Diagrama



2.2. DIAGRAMA DE SEQUÊNCIA

O Diagrama de Sequência representa a interação entre objetos ao longo do tempo, destacando a ordem das mensagens trocadas entre eles. Ele é útil para modelar fluxos de execução e entender a dinâmica do sistema.

2.2.1. Exemplo de Diagrama de Sequência

A seguir, um exemplo baseado em um sistema bancário, mostrando o fluxo de uma transferência entre contas:

2.2.1.1. Objetos

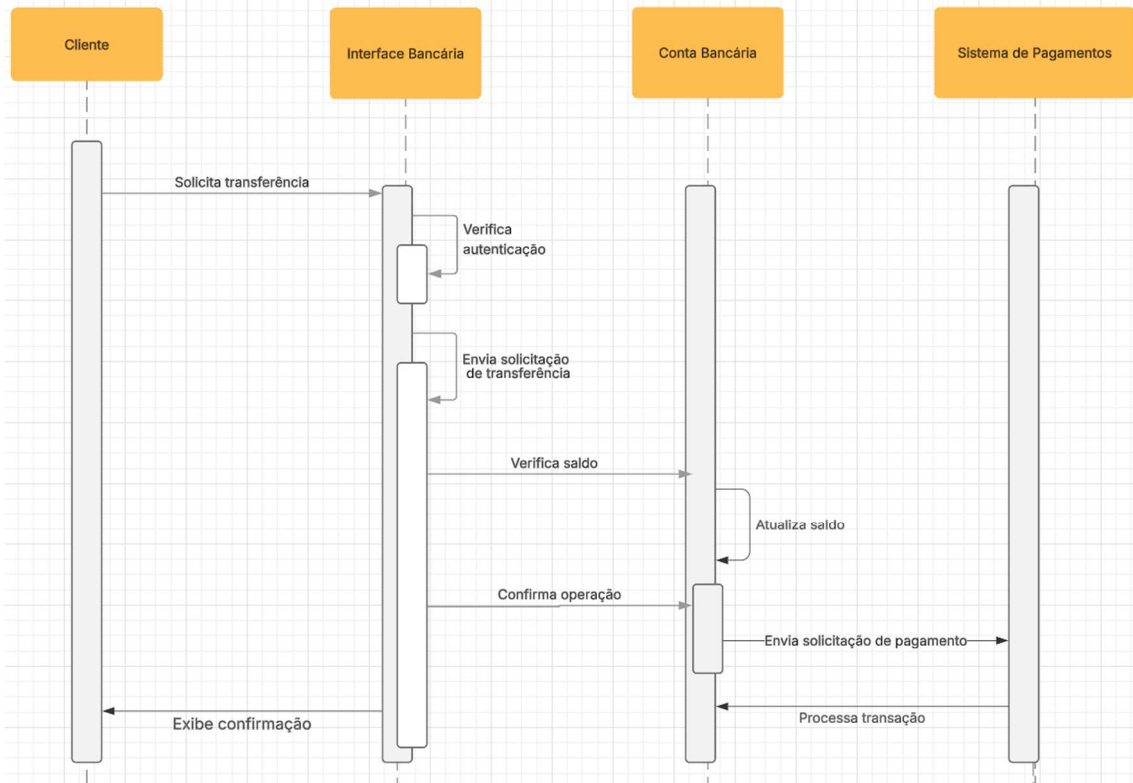
1. Cliente
2. Interface Bancária
3. Conta Bancária
4. Sistema de Pagamentos

2.2.1.2. Mensagens

1. O Cliente solicita a transferência pela Interface Bancária.
2. A Interface Bancária envia a solicitação à Conta Bancária.
3. A Conta Bancária verifica saldo e confirma a operação.
5. A Conta Bancária envia a solicitação ao Sistema de Pagamentos.
6. O Sistema de Pagamentos processa a transação.
7. A Conta Bancária atualiza o saldo do Cliente.
8. A Interface Bancária exibe a confirmação ao Cliente.

O diagrama deve apresentar os objetos no topo e as mensagens como setas verticais, indicando a sequência das interações ao longo do tempo.

2.2.1.3. Diagrama



2.3. DIAGRAMA DE COMUNICAÇÃO

O Diagrama de Comunicação representa a interação entre objetos, enfatizando a organização estrutural e os relacionamentos entre eles. Diferente do Diagrama de Sequência, que prioriza a ordem das mensagens ao longo do tempo, este diagrama destaca como os objetos estão conectados e quais mensagens trocam entre si.

2.3.1. Exemplo de Diagrama de Comunicação

A seguir, um exemplo baseado em um sistema bancário, mostrando o processo de realização de uma transferência:

2.3.1.1. Objetos

1. Cliente
2. Interface Bancária
3. Conta Bancária
4. Sistema de Pagamentos

2.3.1.2. Mensagens e Conexões

1. O Cliente interage com a Interface Bancária para iniciar a transferência.
2. A Interface Bancária se comunica com a Conta Bancária para validar o saldo.
3. A Conta Bancária envia uma solicitação ao Sistema de Pagamentos.
5. O Sistema de Pagamentos confirma a transação e informa a Conta Bancária.
6. A Conta Bancária atualiza o saldo e retorna a confirmação à Interface Bancária.
7. A Interface Bancária exibe o resultado ao Cliente.

O diagrama deve organizar os objetos e representar as mensagens numeradas entre eles, indicando a troca de informações dentro do sistema.

2.4. DIAGRAMA DE ESTADOS

O Diagrama de Estados representa os estados possíveis de um objeto e as transições entre eles, com base em eventos que ocorrem durante o ciclo de vida do objeto. Ele é útil para modelar o comportamento dinâmico de entidades dentro do sistema.

2.4.1. Exemplo de Diagrama de Estados

A seguir, um exemplo baseado em um sistema bancário, mostrando os estados de uma conta bancária:

2.4.1.1. Estados

1. Conta Criada
2. Conta Ativa
3. Conta Bloqueada
4. Conta Encerrada

2.4.1.2. Transições

1. Uma conta recém-criada entra no estado "Conta Criada".
2. Após ativação e depósito inicial, ela transita para "Conta Ativa".
3. Se houver tentativas de acesso indevidas ou saldo negativo por tempo prolongado, a conta passa para "Conta Bloqueada".
4. Se o cliente solicitar o fechamento da conta, ela transita para "Conta Encerrada".

O diagrama deve representar os estados como nós e as transições como setas, indicando os eventos que levam de um estado a outro.

2.5. DIAGRAMA DE ATIVIDADES

O Diagrama de Atividades modela fluxos de trabalho e processos, representando a sequência de ações em um sistema. Ele é útil para visualizar lógica de negócio, fluxos de processos e decisões dentro do sistema.

2.5.1. Exemplo de Diagrama de Atividades

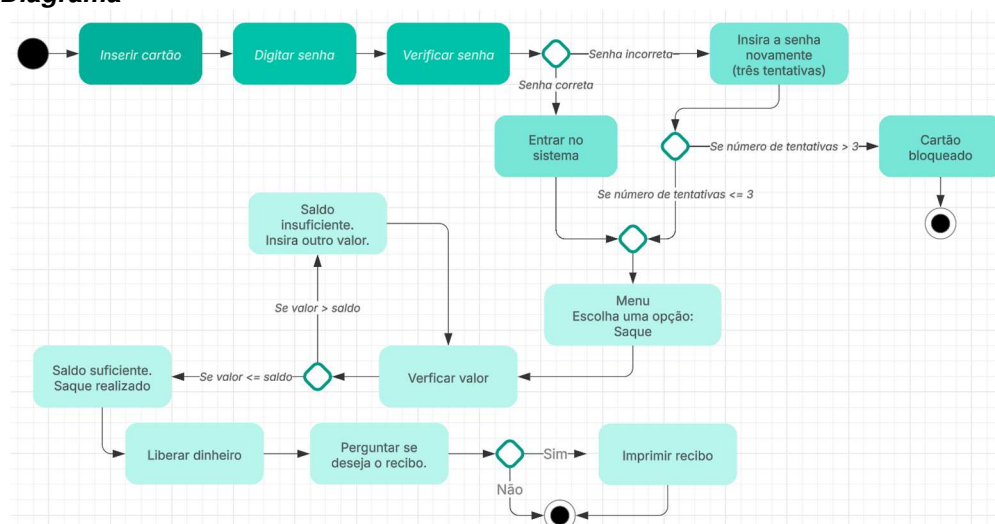
A seguir, um exemplo baseado em um sistema bancário, mostrando o fluxo para realizar um saque:

2.5.1.1. Atividades

1. Início
2. Cliente insere o cartão
3. Cliente digita a senha
4. Verificação da senha
 - Se correta, prossegue
 - Se incorreta, solicita nova tentativa (até três vezes, depois bloqueia)
5. Cliente seleciona "Saque"
6. Cliente informa o valor
7. Verificação de saldo
 - Se suficiente, prossegue
 - Se insuficiente, exibe mensagem de erro
8. Liberação do dinheiro
9. Fim

O diagrama deve conter nós de início e fim, atividades representadas por retângulos arredondados e transições entre elas indicadas por setas. Decisões são representadas por losangos, mostrando caminhos alternativos no fluxo.

2.5.1.2. Diagrama



2.6. DIAGRAMA DE TEMPO

O Diagrama de Tempo representa o comportamento do sistema ao longo do tempo, mostrando a variação do estado dos objetos em uma linha temporal. Ele é útil para modelar processos que dependem de restrições temporais e mudanças de estado em intervalos específicos.

2.6.1. Exemplo de Diagrama de Tempo

A seguir, um exemplo baseado em um sistema bancário, mostrando o comportamento de uma transação bancária ao longo do tempo:

2.6.1.1. Objetos e Estados

1. Cliente
 - Inicia transação
 - Aguarda processamento
 - Recebe confirmação
2. Conta Bancária
 - Aguarda solicitação
 - Processa transação
 - Atualiza saldo
3. Sistema de Pagamentos
 - Valida transação
 - Deduz valor da conta origem
 - Credita valor na conta destino

2.6.1.2. Linha Temporal

1. Cliente solicita transferência (t0)
2. Conta Bancária recebe a solicitação (t1)
3. Sistema de Pagamentos valida e processa (t2)
4. Conta Bancária atualiza saldo e confirma operação (t3)
5. Cliente recebe confirmação (t4)

O diagrama deve conter linhas horizontais representando os objetos e suas mudanças de estado ao longo da linha do tempo, com eventos marcando as transições entre estados.

2.7. DIAGRAMA DE VISÃO GERAL DE INTERAÇÃO

O Diagrama de Visão Geral de Interação combina elementos de diagramas de sequência e atividades para representar fluxos complexos dentro do sistema. Ele permite visualizar a interação entre diferentes partes do sistema, destacando decisões e interações em processos dinâmicos.

2.7.1. Exemplo de Diagrama de Visão Geral de Interação

A seguir, um exemplo baseado em um sistema bancário, mostrando o fluxo para realizar uma transferência entre contas:

2.7.1.1. Atividades e Interações

1. Início
2. Cliente acessa o aplicativo
3. Cliente seleciona "Transferência"
 - Se não estiver autenticado, redireciona para login
4. Cliente informa os dados da conta de destino e o valor
5. Sistema verifica saldo disponível
 - Se saldo insuficiente, exibe mensagem de erro e encerra o fluxo
6. Se saldo suficiente, o sistema solicita autenticação via token
7. Cliente insere o token e confirma a operação
8. Sistema processa a transação e comunica o banco
9. Banco confirma a transação e retorna a resposta
10. Sistema atualiza saldo e exibe confirmação ao cliente
10. Fim

2.8. DIAGRAMA DE CASOS DE USO DE NEGÓCIO

O Diagrama de Casos de Uso de Negócio representa os processos de negócio sem focar na tecnologia ou no sistema de software. Ele modela interações entre participantes (atores) e atividades-chave do negócio.

2.8.1. Exemplo de Diagrama de Casos de Uso de Negócio

A seguir, um exemplo baseado em um banco, destacando os principais processos de negócio:

2.8.1.1. Atores

1. Cliente
2. Gerente
3. Caixa Bancário

2.8.1.2. Casos de Uso

1. Abrir Conta (Cliente, Gerente)
2. Depositar Dinheiro (Cliente, Caixa Bancário)
4. Sacar Dinheiro (Cliente, Caixa Bancário)
5. Solicitar Empréstimo (Cliente, Gerente)
6. Aprovar Empréstimo (Gerente)
7. Gerenciar Conta (Gerente)

2.8.1.3. Relacionamentos

1. O caso de uso "Abrir Conta" pode incluir "Verificar Documentação" («include»).
2. "Solicitar Empréstimo" estende "Análise de Crédito" («extend»).
3. "Gerenciar Conta" inclui "Atualizar Dados do Cliente" e "Encerrar Conta" («include»).