



UniCesumar

EDUCAÇÃO PRESENCIAL E A DISTÂNCIA

Diagrama de Sequência



Diagrama de Sequência

- Interação entre os objetos
- Determina a sequência de eventos que ocorrem em um determinado processo
- Quais condições devem ser satisfeitas ...
- Quais métodos devem ser disparados ...
- E em qual ordem ...
- Baseia-se no Diagrama de Casos de Uso
 - 1 Caso de Uso → N Diagramas de Sequência
- Baseia-se, também, no Diagrama de Classes
 - Fornecem as classes e os métodos associados



Diagrama de Sequência

Componentes - ATORES

- Exatamente os mesmos dos Casos de Uso
- Interação → Solicitam serviços → Eventos → Processos
- Não são obrigatórios no Diagrama de Sequência

Diagrama de Sequência

Componentes - OBJETOS

- Representam as instâncias das classes
- Retângulos contendo um texto
- Primeira parte, em minúsculo, o nome do objeto
- Segunda parte, em letras iniciais maiúsculas, o nome da classe
- Informações separadas por dois pontos (:)
- Linha de vida
- Linha vertical tracejada

Diagrama de Sequência

fisica1:
Física

- Componentes - OBJETOS

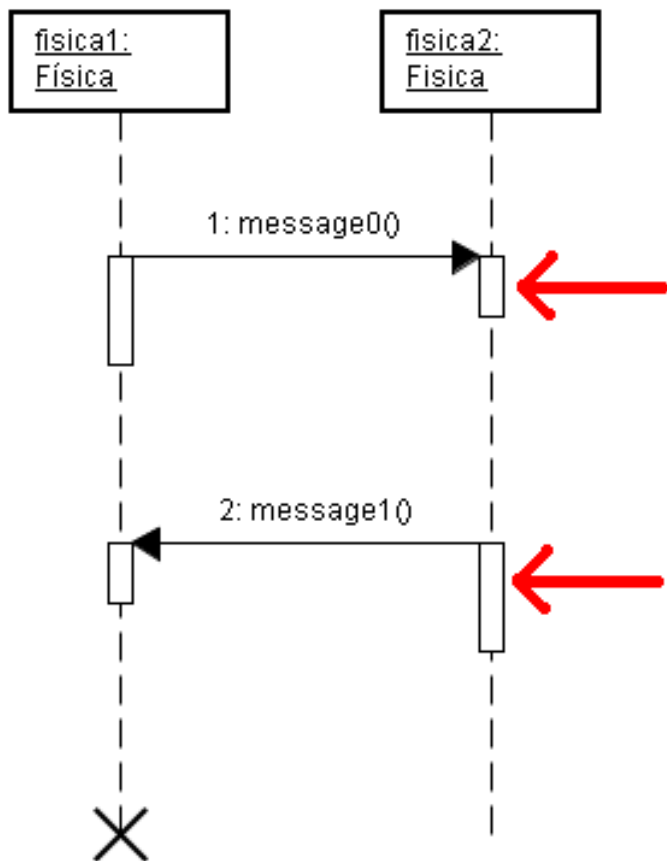
Diagrama de Sequência

- Componentes – LINHAS DE VIDA
- Representa o tempo que um objeto existiu durante um processo;
- Linhas finas verticais tracejadas;
- Iniciam no retângulo que representa o objeto;
- Interrompida por um "X" quando o objeto é destruído.

Diagrama de Sequência

- Componentes – FOCO DE CONTROLE/ATIVAÇÃO
- Indica os períodos em que um determinado objeto está participando ativamente do processo;
 - Executando um ou mais métodos do processo.
- Representados por extensões mais grossas/largas da Linha de Vida.

Diagrama de Sequência



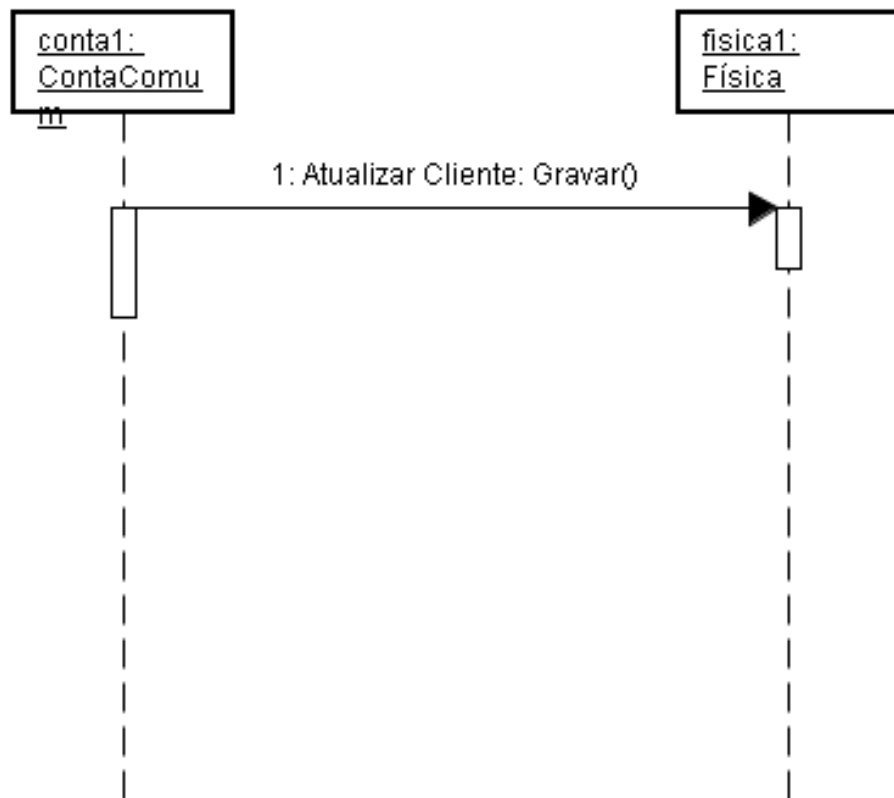
- Componentes – FOCO DE CONTROLE/ATIVAÇÃO

Diagrama de Sequência

Componentes – MENSAGENS/ESTÍMULOS

- Demonstram a ocorrência de eventos que normalmente forçam a chamada de um método em algum dos objetos envolvidos no processo
- Mensagens entre:
 - Ator e Ator;
 - Ator e Objeto;
 - Objeto e Objeto;
 - Objeto e Ator.

Diagrama de Sequência



- Mensagem com disparo de método entre objetos.

Diagrama de Sequência



- Mensagem que instancia um novo objeto

Diagrama de Sequência



- Mensagem que dispara um método destrutor – elimina um objeto não mais necessário.

Diagrama de Sequência

- Mensagem de Retorno ... Linha tracejada.
- Podem retornar valores ou status...

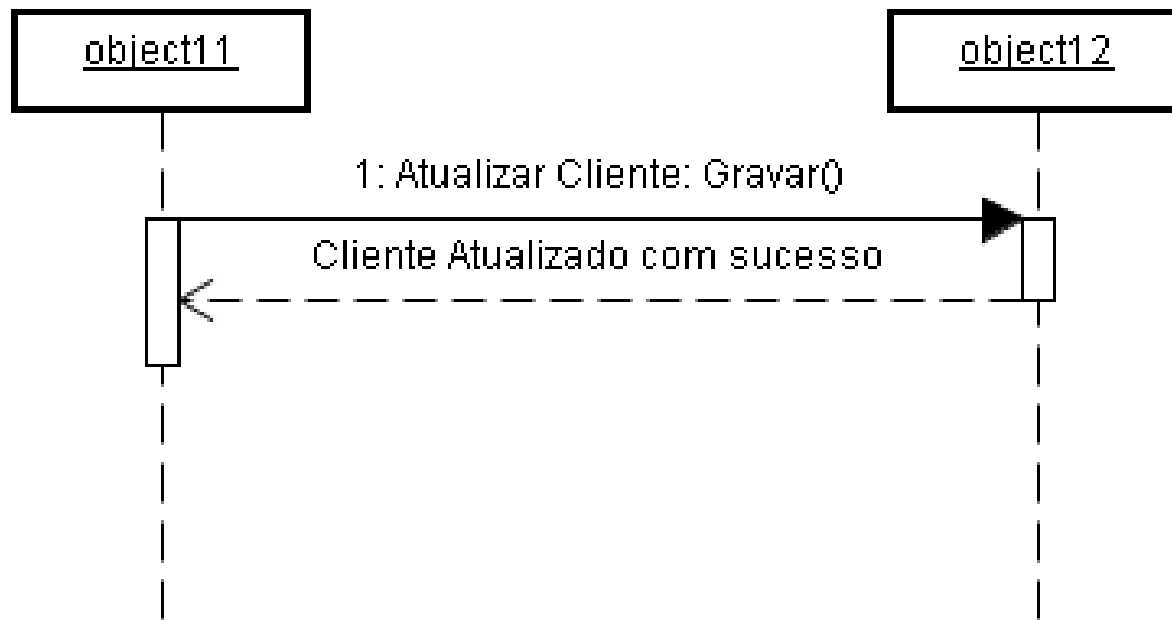


Diagrama de Sequência

- Mensagem com Condição de Guarda
- Entre colchetes ()

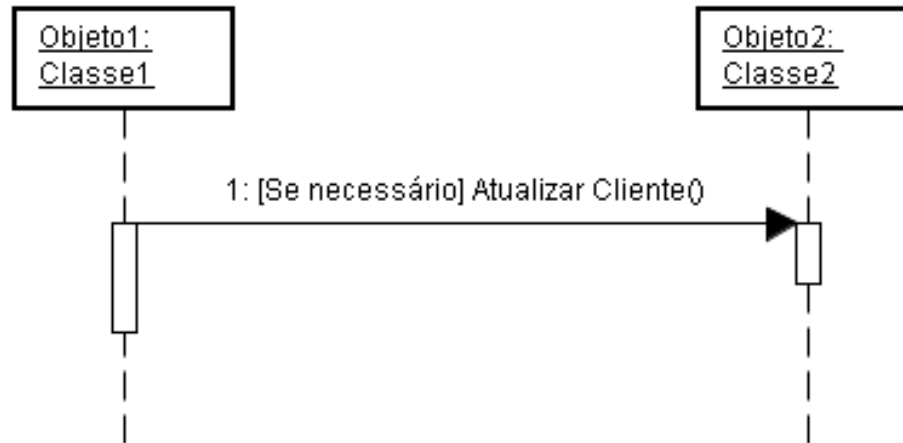
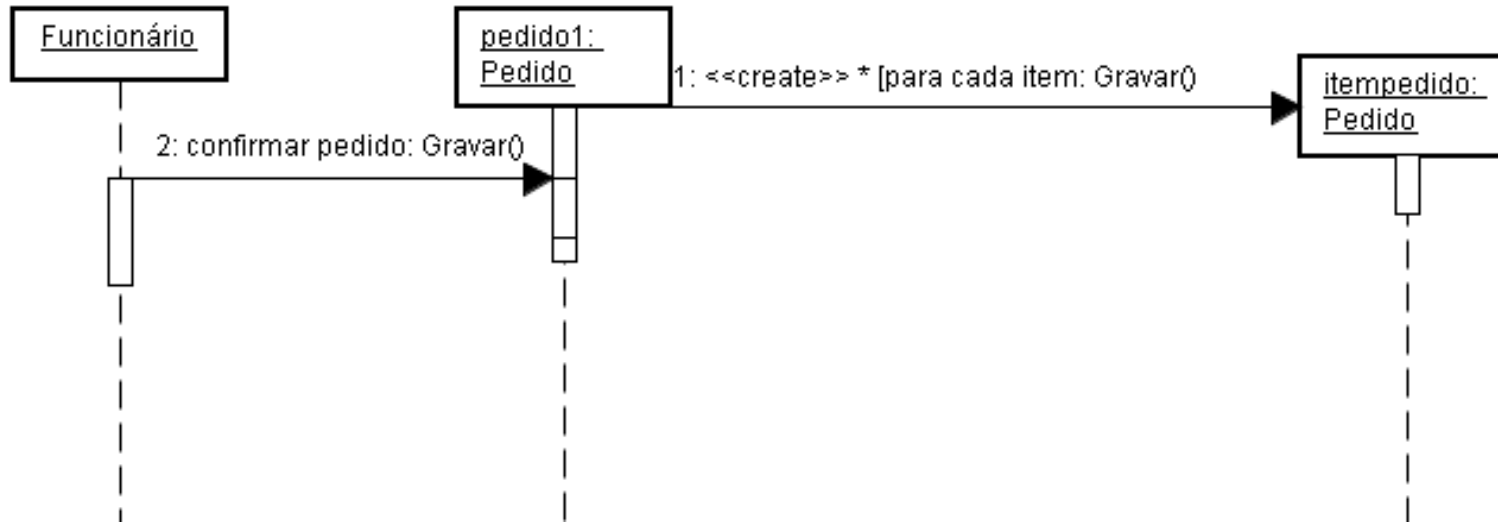


Diagrama de Sequência

- Mensagem com Condição de Guarda
- Disparo de mensagem a vários objetos (*)



Abertura de Conta

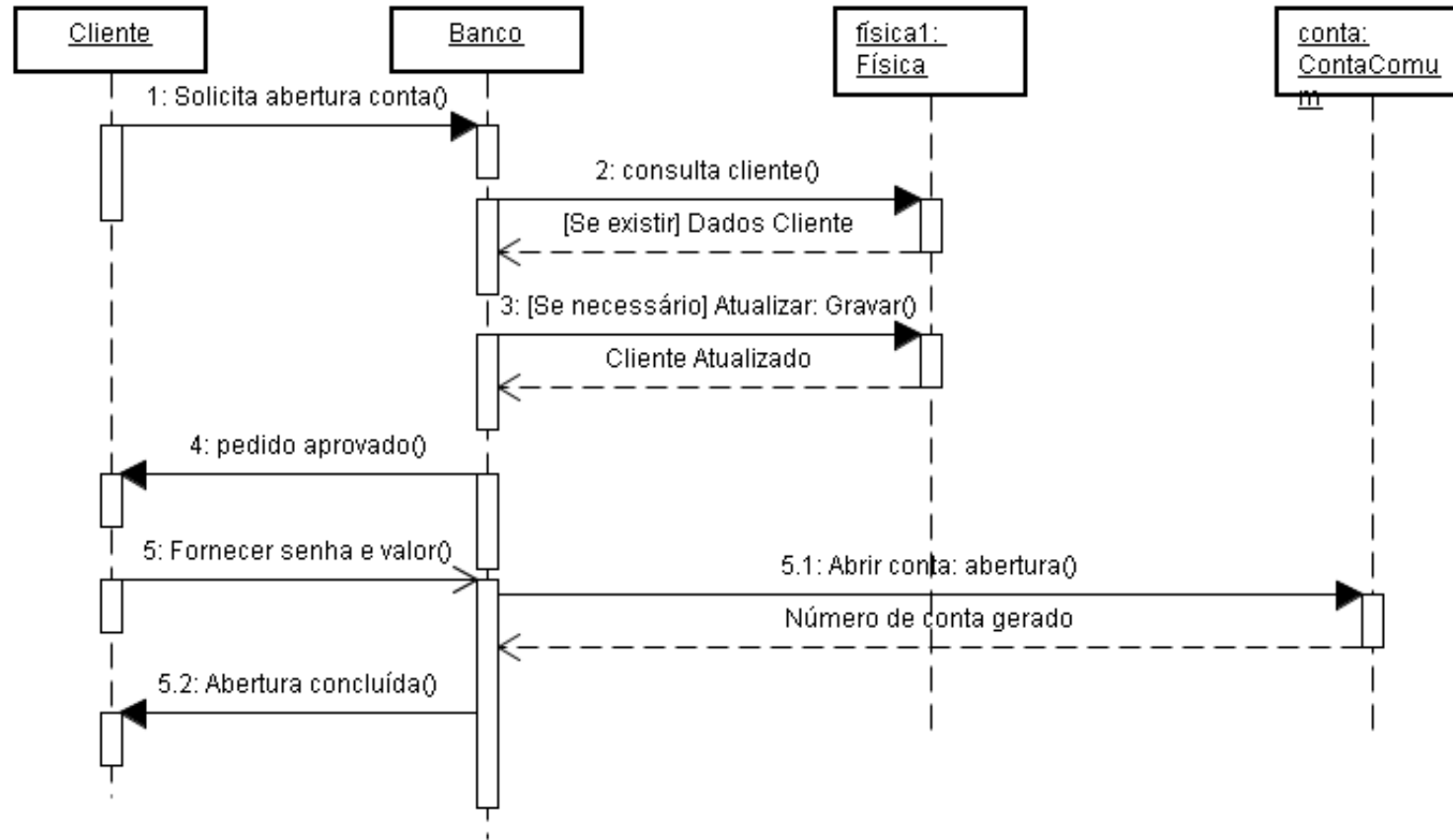


Diagrama de Atividades



Diagrama de Atividades

- É o diagrama com maior ênfase ao nível de algoritmo da UML e provavelmente um dos mais detalhistas.
- Apresenta muitas semelhanças com os antigos fluxogramas.
- Este diagrama preocupa-se em descrever os passos a serem percorridos para a conclusão de um método ou algoritmo específico e não um processo completo como é o diagrama de sequência.

Diagrama de Atividades

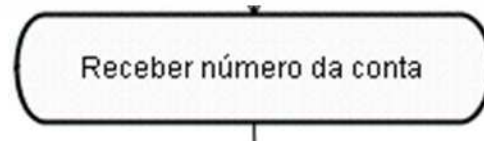
Exemplo de Estado Inicial



Exemplo de Estado Final



Estado de Ação

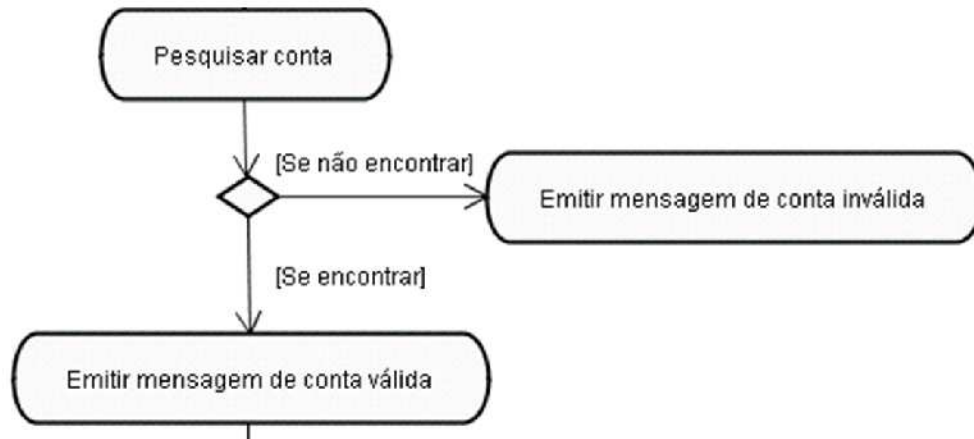


- Estado Inicial
 - Inicia o diagrama de atividades e os processos serão analisados.
- Estado Final
 - Indica o final do Diagrama de Atividades.
- Estado de Ação
 - Representa a realização de uma ação dentro de um fluxo de controle.



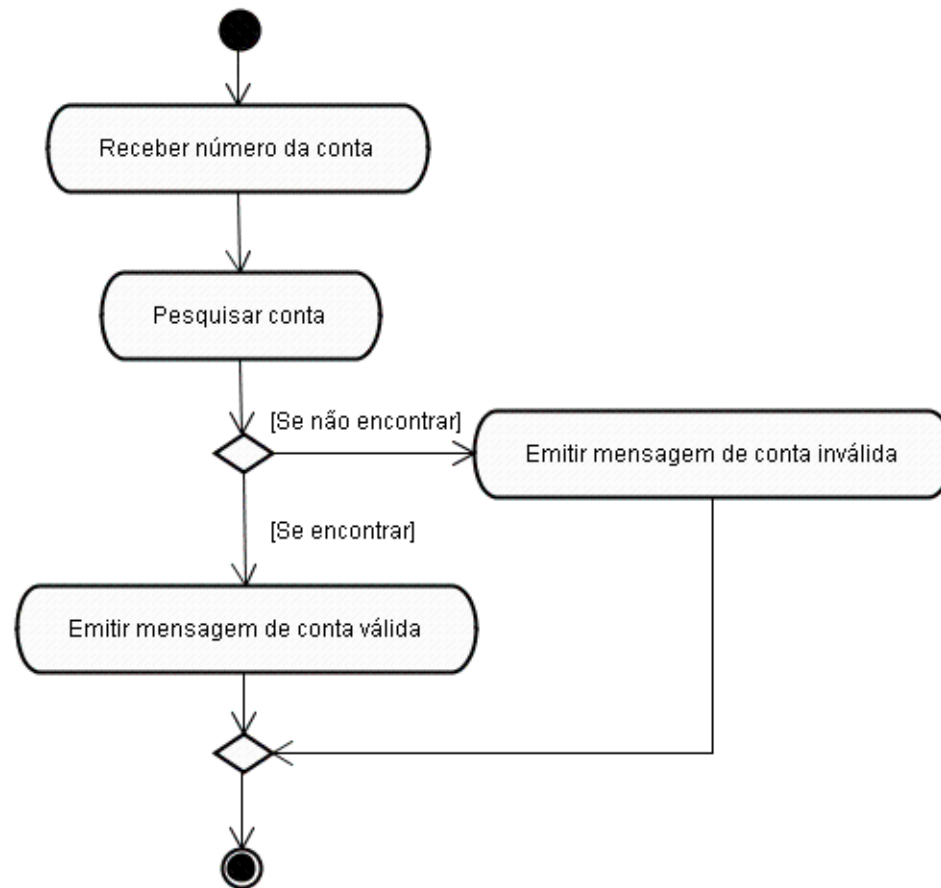
Ponto de Decisão

- Representa um ponto do fluxo que será tomada de decisão.
- O ponto de decisão deve conter a decisão com o texto entre colchetes para determinar qual a condição do teste.

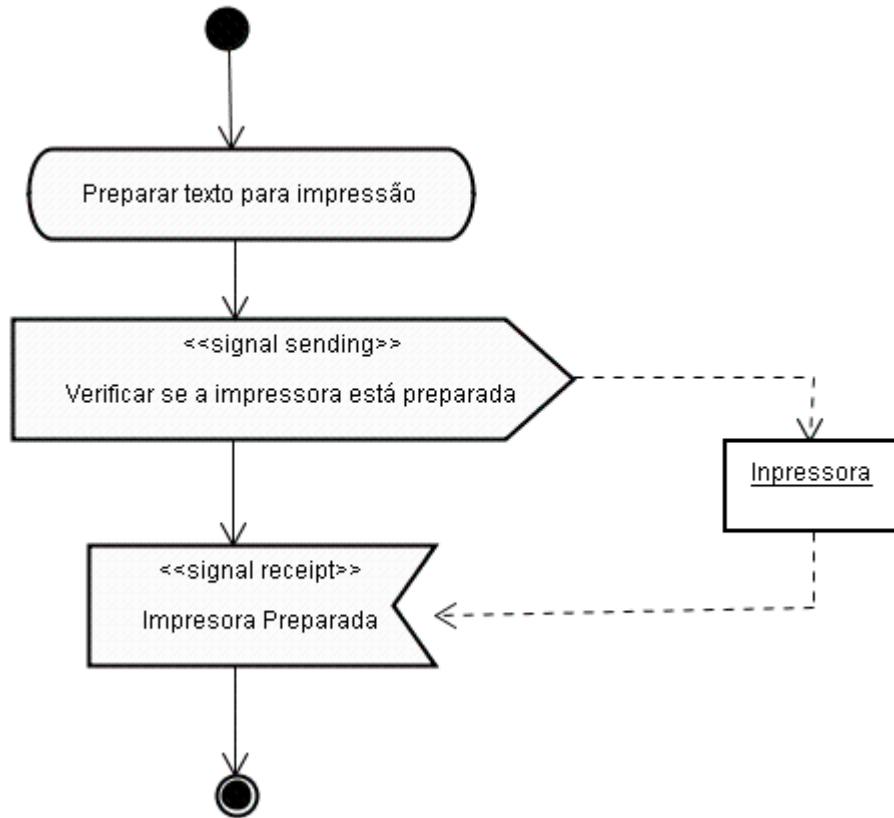


Ponto de Decisão

Exemplo de Diagrama de Atividades



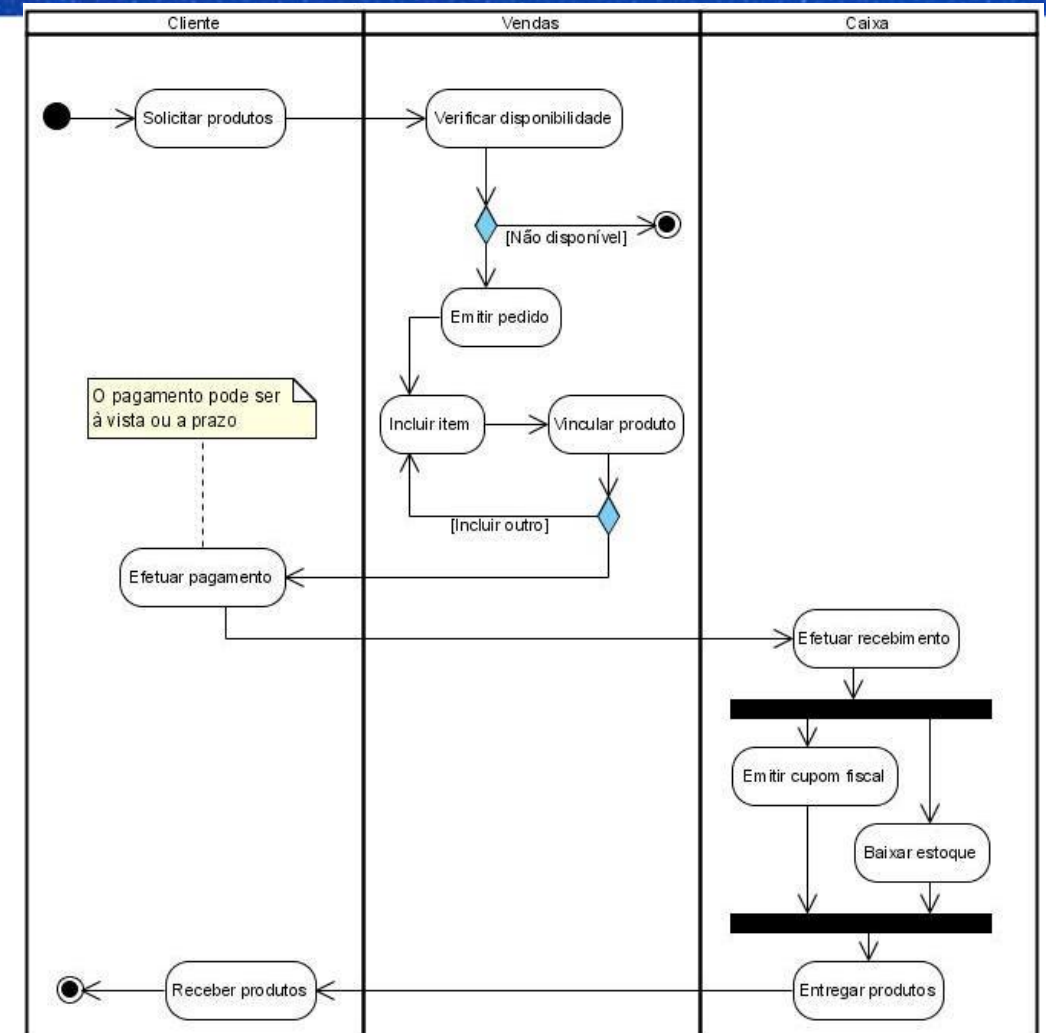
Envio e Recebimento de Sinal



- Representa o recebimento de um sinal de um dispositivo externo, normalmente um item de hardware.

Raias de Natação

- São uma extensão do Diagrama de Atividades, onde procura-se identificar os diversos setores, departamentos ou mesmo os atores que interagem com um processo.





Qualidade de Software



Qualidade de Software

- Criar mecanismos para garantir a qualidade de software evita uma série de transtornos após a implantação.
- Defeitos ligados a falhas de software podem trazer prejuízos financeiros tanto para as empresas que o utilizam, quanto para os usuários.
- Desenvolvedores se esforçam para minimizar erros.
- Realizar testes é uma das premissas das metodologias ágeis de desenvolvimento de software.



O que é qualidade de software?

- É difícil definir o que é Qualidade de Software
- Não há um acordo entre os profissionais da área sobre o que é qualidade de software.
- Software não tem produção em série. Seu custo está no projeto e desenvolvimento.



Qualidade de software é dada por um conjunto de fatores a serem atendidos. Nesse sentido, a aplicação precisa estar em conformidade com as especificações de requisitos tomadas junto ao cliente, além de seguir padrões de qualidade da empresa.

Equipe Monitora

Normas nacionais e internacionais de qualidade

NORMAS	COMENTÁRIOS
ISO 9126	Características da qualidade de produtos de software.
NBR 13596	Versão brasileira da ISO 9126.
ISO 14598	Guias para a avaliação de produtos de software, baseados na utilização prática da norma ISO 9126.
ISO 12119	Características de qualidade de pacotes de software (software de prateleira, vendido como um produto embalado).
IEEE P1061	Standard for Software Quality Metrics Methodology. Norma que trata das metodologias para padronização da qualidade de software, incluindo algumas abordagens de medição.
ISO 12207	Software Life Cycle Process. Norma para a qualidade do processo de desenvolvimento de software.
NBR ISO 9001	Sistemas de qualidade – Modelo para garantia de qualidade em projeto, desenvolvimento, instalação e assistência técnica (processo).
NBR ISO 9000-3	Gestão de qualidade e garantia de qualidade. Aplicação da norma ISO 9000 para o processo de desenvolvimento de software.
NBR ISO 10011	Auditoria de Sistemas de Qualidade (processo).
CMMI	Capability Maturity Model Integration. Modelo da SEI (Instituto de Engenharia de Software do Departamento de Defesa dos USA) para avaliação da qualidade do processo de desenvolvimento de software. Não é uma norma ISO, mas é muito bem aceita no mercado.
SPICE ISO 15504	Projeto da ISO/IEC para avaliação do processo de desenvolvimento de software. Ainda não é uma norma oficial ISO, mas o processo está em andamento.

Qualidade de Software na visão do usuário

- A experiência do usuário, além das qualidades técnicas do software, é um fator determinante para a construção de sistemas de maior qualidade.
- Faz sentido verificar se o software está ok, pela percepção de quem vai utilizar.
- O usuário procura respostas para questões como:
 - As funcionalidades estão disponíveis e são executadas eficientemente?
 - O software funciona corretamente em imprevistos?
 - O software é seguro, ou seja, evita que pessoas ou sistemas não autorizados tenham acesso às informações?
 - É fácil de usar ou requer muito treinamento?
 - É fácil de integrar com outros sistemas existentes?



Fatores de qualidade de software



Fatores de qualidade de software

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	SIGNIFICADO
Funcionalidade O conjunto de funções satisfazem as necessidades explícitas e implícitas para a finalidade a que se destina o produto?	Adequação	Propõe-se a fazer o que é apropriado?
	Acurácia	Gera resultados corretos ou conforme acordados?
	Interoperabilidade	É capaz de interagir com os sistemas especificados?
	Segurança de acesso	Evita o acesso não autorizado, acidental ou deliberado a programas e dados?
	Conformidade	Está de acordo com normas e convenções previstas em leis e descrições similares?
Confiabilidade O desempenho se mantém ao longo do tempo e em condições estabelecidas?	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas como ele reage?
	Recuperabilidade	É capaz de recuperar dados após uma falha?
Usabilidade É fácil utilizar o software?	Inteligibilidade	É fácil entender os conceitos utilizados?
	Apreensibilidade	É fácil aprender a usar?
	Operacionalidade	É fácil de operar e controlar a operação?
Eficiência Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?	Comportamento em relação ao tempo	Qual é o tempo de resposta e de processamento?
	Comportamento em relação aos recursos	Quanto recurso utiliza?
Manutenibilidade Há facilidade para correções, atualizações e alterações?	Analísabilidade	É fácil encontrar uma falha quando ocorre?
	Modificabilidade	É fácil modificar e remover defeitos?
	Estabilidade	Há grandes riscos de bugs quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
Portabilidade É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?	Adaptabilidade	É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado?
	Capacidade para ser instalado	É fácil instalar em outros ambientes?
	Capacidade para substituir	É fácil substituir por outro software?
	Conformidade	Está de acordo com padrões ou convenções de portabilidade?

Elementos de garantia da qualidade de software

Padrões	Organizações que padronizam requisitos de qualidade
	São adotados voluntariamente pela organização
	Algumas vezes exigidos pelos clientes
Revisões e Auditorias	Possuem o objetivo de garantir a qualidade da entrega
	Objetivo é revelar erros
Testes	Atividade de controle de qualidade
	Objetivo de descobrir erros
Coleta e Análise de Erros	Medição de desempenho
	Analisar dados com o objetivo de resolver a causa raiz
Administração da Segurança	Definir políticas para proteger os dados e operações
	Utiliza-se softwares de proteção para auxiliar na atividade
Proteção	Avaliar riscos
	Avaliar consequências
	Objetivo de reduzir os problemas



Controle ou Garantia da Qualidade de Software

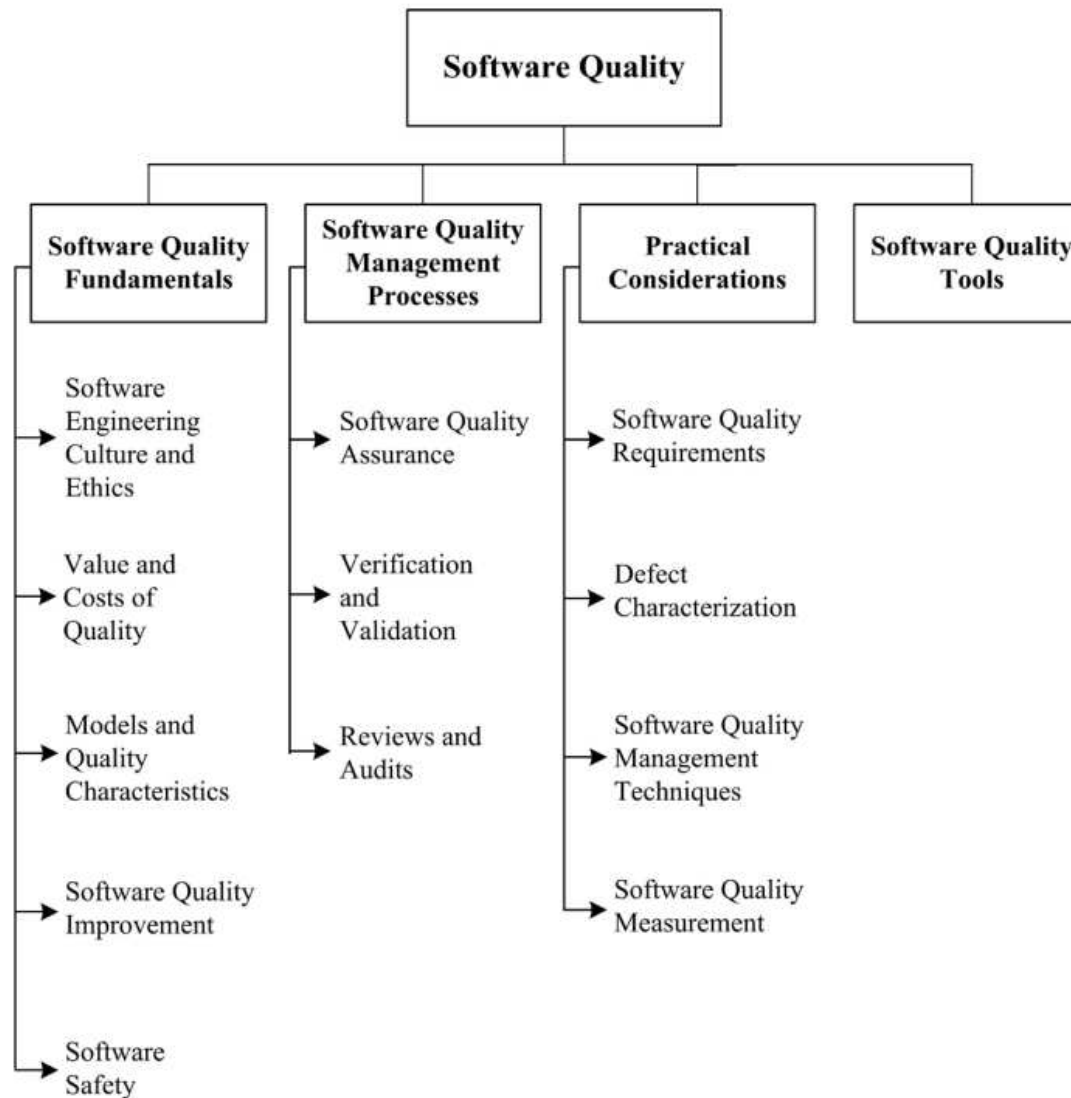
- Importante não confundir os termos:
 - Controle da Qualidade (Quality Control)
 - Garantia da Qualidade (Quality Assurance).
- Embora usados de maneira errônea em muitos lugares, ambos os termos têm propósitos totalmente diferentes.

Controle ou Garantia da Qualidade de Software

Quality Assurance	Quality Control
1. Garantia da qualidade garante que o processo é definido e apropriado.	1. As atividades de controle da qualidade focam na descoberta de defeitos em i específicos.
2. Metodologia e padrões de desenvolvimento são exemplos de garantia da qualidade.	2. Um exemplo de controle da qualidade poderia ser: "Os requisitos definidos são os requisitos certos?".
3. Garantia da qualidade é orientada a processo.	3. Controle da qualidade é orientado a produto.
4. Garantia da qualidade é orientada a prevenção.	4. Controle da qualidade é orientado a detecção.
5. Foco em monitoração e melhoria de processo.	5. Inspeções e garantia de que o produto de trabalho atenda aos requisitos especificados.
6. As atividades são focadas no início das fases no ciclo de vida de desenvolvimento de software.	6. As atividades são focadas no final das fases no ciclo de vida de desenvolvimento de software.
7. Garantia da qualidade garante que você está fazendo certo as coisas e da maneira correta.	7. Controle da qualidade garante que os resultados do seu trabalho são os esperados conforme requisitos.



Qualidade de Software segundo SWEBOK



CMMI - Capability Maturity Model Integration

- Um processo representa, dentro da área de software, um conjunto de atividades cujo objetivo é atingir uma meta previamente estipulada. Já por capacidade e maturidade de um processo, deve-se ter a noção do grau de qualidade com o qual um processo atinge um resultado esperado.
- Dentre os principais benefícios da implantação do CMMI, vale a pena destacar:
 - Uma maior confiabilidade no que refere ao cumprimento de prazos e custos que foram acordados, inicialmente, perante o cliente que solicitou o desenvolvimento de um sistema. Essa previsibilidade é decorrente do rigor que o CMMI exige quanto à medição dos processos, fato este que conduz à obtenção de uma base histórica realista e confiável para estes fins;
 - O gerenciamento das atividades relativas à produção de software aumenta consideravelmente;
 - Uma maior qualidade nos softwares criados, já que processos bem definidos e controlados conduzem à produção de produtos mais confiáveis;
 - A menor dependência da empresa de desenvolvimento para com seus especialistas. Com um foco voltado para processos e melhoria contínua, além do uso intensivo de informações históricas, a organização deixa de depender única e exclusivamente de profissionais com um elevado grau de conhecimento técnico;
 - A busca por melhorias contínuas nos processos cotidianos.

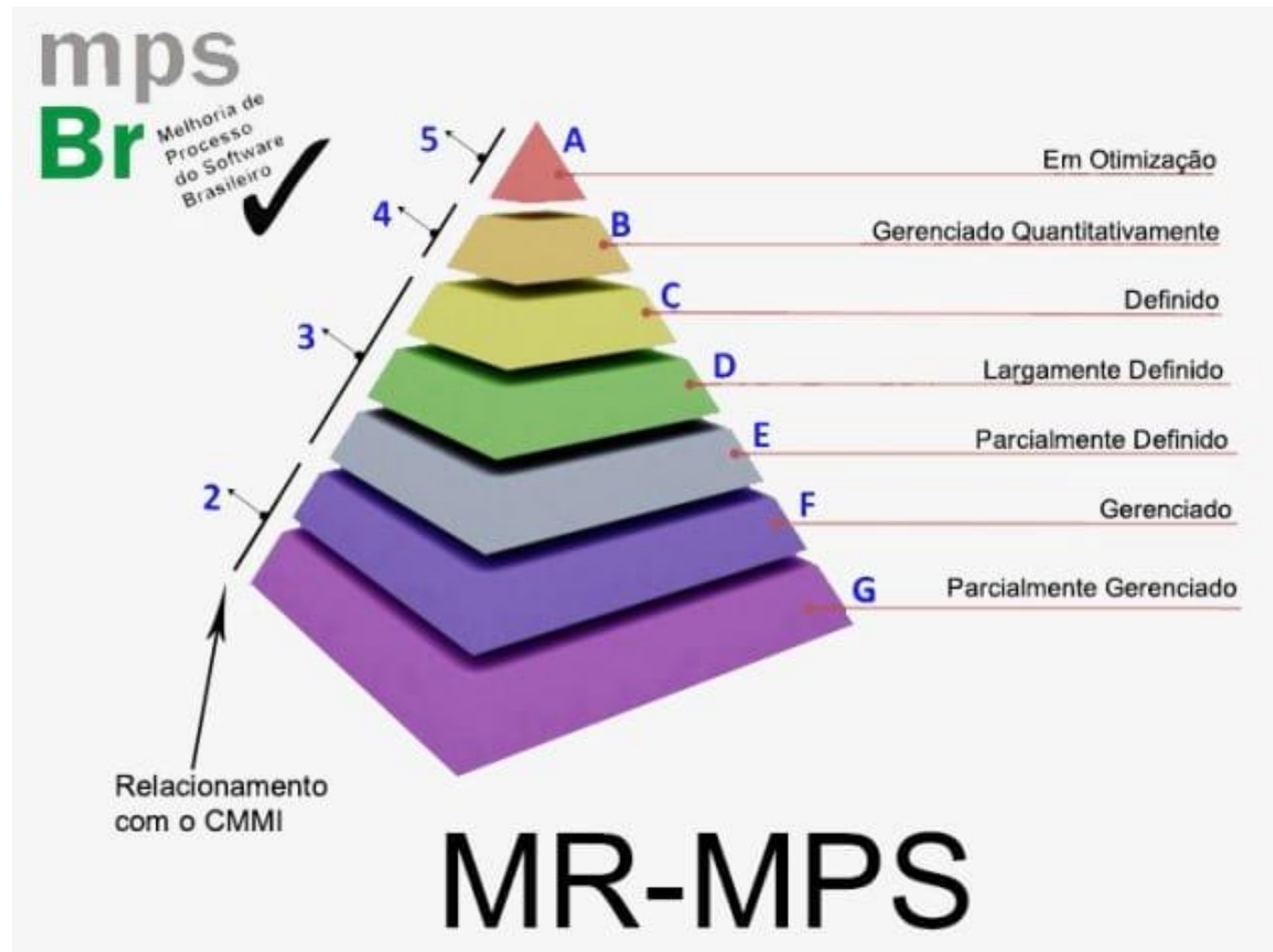


CMMI - Capability Maturity Model Integration

- Conseguir o que este modelo propõe, a organização interessada na implantação do CMMI deverá evoluir progressivamente, considerando para isto uma sucessão de diferentes de níveis.
- Cada nível indica, por sua vez, o grau de maturidade dos processos num determinado instante:
 - Nível 1 - Inicial: os processos normalmente estão envoltos num caos decorrente da não-obediência ou ainda, inexistência de padrões;
 - Nível 2 - Gerenciado: os projetos têm seus requisitos gerenciados neste ponto. Além disso, há o planejamento, a medição e o controle dos diferentes processos;
 - Nível 3 - Definido: os processos já estão claramente definidos e são compreendidos dentro da organização. Os procedimentos se encontram padronizados, além de ser preciso prever sua aplicação em diferentes projetos;
 - Nível 4 - Gerenciado Quantitativamente: ocorre o aumento da previsibilidade do desempenho de diferentes processos, uma vez que os mesmos já são controlados quantitativamente;
 - Nível 5 - Otimizado: existe uma melhoria contínua dos processos.



MSP-BR



Tipos de Testes

Tipo de Teste	Descrição
Teste de Unidade	Teste em um nível de componente ou classe. É o teste cujo objetivo é um “pedaço do código”.
Teste de Integração	Garante que um ou mais componentes combinados (ou unidades) funcionam. Podemos dizer que um teste de integração é composto por diversos testes de unidade.
Teste Operacional	Garante que a aplicação pode rodar muito tempo sem falhar.
Teste Positivo-negativo	Garante que a aplicação vai funcionar no “caminho feliz” de sua execução e vai funcionar no seu fluxo de exceção.
Teste de Regressão	Toda vez que algo for mudado, deve ser testada toda a aplicação novamente.
Teste de Caixa-preta	Testar todas as entradas e saídas desejadas. Não se está preocupado com o código, cada saída indesejada é visto como um erro.
Teste Caixa-branca	O objetivo é testar o código. Às vezes, existem partes do código que nunca foram testadas.
Teste Funcional	Testar as funcionalidades, requerimentos, regras de negócio presentes na documentação. Validar as funcionalidades descritas na documentação (pode acontecer de a documentação estar inválida)



Tipos de Testes

Tipo de Teste	Descrição
Teste de Interface	Verifica se a navegabilidade e os objetivos da tela funcionam como especificados e se atendem da melhor forma ao usuário.
Teste de Performance	Verifica se o tempo de resposta é o desejado para o momento de utilização da aplicação.
Teste de Carga	Verifica o funcionamento da aplicação com a utilização de uma quantidade grande de usuários simultâneos.
Teste de Aceitação do usuário	Testa se a solução será bem vista pelo usuário. Ex: caso exista um botão pequeno demais para executar uma função, isso deve ser criticado em fase de testes. (aqui, cabem quesitos fora da interface, também).
Teste de Volume	Testar a quantidade de dados envolvidos (pode ser pouca, normal, grande, ou além de grande).
Testes de Stress	Testar a aplicação sem situações inesperadas. Testar caminhos, às vezes, antes não previstos no desenvolvimento/documentação.
Testes de Configuração	Testar se a aplicação funciona corretamente em diferentes ambientes de hardware ou de software.
Testes de Instalação	Testar se a instalação da aplicação foi OK.
Testes de Segurança	Testar a segurança da aplicação das mais diversas formas. Utilizar os diversos papéis, perfis, permissões, para navegar no sistema.



Aonde você quer chegar?
Vai com a



UniCesumar

EDUCAÇÃO PRESENCIAL E A DISTÂNCIA