

0303

INFORMAÇÕES GERAIS

Laboratório de Informática

Verificar antes do início da aula a situação do computador; proibido comer e/ou beber nos laboratórios; Proibido jogos de qualquer tipo;

Horários;

Formatação de Trabalhos.

INTRODUÇÃO

O desenvolvimento de software é uma atividade de crescente importância na sociedade contemporânea. A utilização de computadores nas mais diversas áreas do conhecimento humano tem gerado uma crescente demanda por soluções computadorizadas.

CONTEXTUALIZAÇÃO - Marcos Históricos

Máquina a Vapor

Inventada (1712)

Aperfeiçoada (1766 - '69 - '82)

Mecanização da Indústria Têxtil

Tear mecânico (1722)

Máquina de fiar (1764)

Aço (1856 e 1885 – Liga)

Locomotiva a Vapor (Rede de transporte – 1830)

Máquina de Costura (1851)

Motor a combustão interna

Primeiro eficiente (1876)

Produção automobilística em massa (1896)

Energia nuclear (1943)

Uso industrial/Comercial da Eletricidade

Computadores Eletrônicos (1946)

Transistor (1948)



1003

O que é software ?

- Programas de computador
- Entidades abstratas
- Ferramentas que
 - Exploram os recursos de hardware
 - Executam determinadas tarefas
 - Resolvem problemas
 - Torna o computador operacional

Entidade: algo ou alguém que representa determinada função no sistema – nesse caso, o software (todas as aplicações são software, os aplicativos em geral).

O software é algo intangível (aplicações) e o hardware é tangível (celular, televisão).

O software tem que explorar/utilizar os recursos do hardware, mas não pode ultrapassá-los (vai passar o limite do computador). A aplicação é desenvolvida de acordo com o hardware.

O cliente que solicita a execução de tarefas, ele descreve como quer e o que quer.

Tem que deixar o computador pronto para o uso de qualquera um, tornar o computador operacional.

Os softwares ficam no HD do hardware.

Segundo Pressman, software é ...

É o Produto que os profissionais de software constroem e mantêm ao longo de tempo. Abrange programas que executam em qualquer computador, conteúdo que é apresentado ao programa a ser executado e a documentação impressa ou virtual.

1703

Panorama do SW

- O que é? É o produto que os profissionais de software constroem e mantêm ao longo de tempo.
- Quem faz? Engenheiros de Software constroem e mantêm e todas as pessoas que usam de maneira direta e indireta
- Por que é importante? Porque afeta todos os aspectos de nossas vidas e difundiu-se no comércio, cultura e no cotidiano.

Sempre se faz um SW com um objetivo.

Dentro dos engenheiros de SW tem o desenvolvedor Java, PhP...

As funcionalidades que surgiram de acordo com as nossas necessidades são os softwares.

SW comerciais = mercado, restaurante, biblioteca... IOT – Internet das Coisas.

Características do software

- SW não é um elemento físico, é um elemento lógico (não tem propriedades físicas)
- O software não se desgasta com o uso, mas deteriora-se.
- Não há peças de reserva → manutenção, correção e aperfeiçoamento.
- A maioria dos softwares é feita sob medida e não montada a partir de componentes existentes → reusabilidade.

Como desenvolvedor, tem que pensar nessa parte de segurança do SW.

Deteriora-se: Com o passar do tempo ele não atende mais as necessidades e às vezes a empresa têm que trocar o sistema.

Se tem um limite para aperfeiçoamentos e depois de um tempo não é mais possível aperfeiçoar (arrumar).

Atributos essenciais de um bom software

Características	Descrição
Manutenibilidade	O software deve ser escrito de forma que possa evoluir para atender às necessidades dos clientes.
Confiabilidade e proteção	Confiabilidade, proteção e segurança devem fazer parte de um software. Um software confiável não deve causar prejuízos físicos ou econômicos no caso de falha do sistema e usuários maliciosos não devem ser capazes de acessar ou prejudicar o sistema.
Eficiência	O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto eficiência inclui capacidade de resposta, tempo de processamento, uso de memória etc.
Aceitabilidade	Ele deve ser aceitável para o tipo de usuário para o qual foi projetado, isso significa que deve ser compreensível, usável e compatível com outros sistemas usados por ele.

Qualidade de software - Exemplo

- Correto
 - A loja não pode deixar de cobrar por produtos comprados pelo consumidor.
- Robusto
 - A loja não pode parar de vender.
- Eficiente
 - O consumidor não pode esperar.
 - A empresa quer investir pouco em recursos computacionais.
- Amigável e fácil de usar
 - A empresa quer investir pouco em treinamento.

2403

TRABALHO

Tema – “Mulheres e Tecnologia”

Pesquisar sobre o trabalho de mulheres que ajudaram no desenvolvimento da informática (uma para cada integrante do grupo).

O trabalho deverá ter capa, sumário, introdução, desenvolvimento (pequena biografia das mulheres com suas respectivas contribuições) e bibliografia (com data e hora do acesso).

Ao final do trabalho cada integrante fará uma conclusão individual sobre o trabalho todo.

Formatação (ABNT):

Fonte: Arial ou Times New Roman.

Tamanho da fonte: títulos 14 e corpo do texto 12.

Alinhamento: justificado.

Espaçamento entre linhas: 1.5

Bibliografia: data e hora em que os sites foram acessados.

3103 - recesso

0704

Atributos essenciais de um bom software

Características	Descrição
Manutenibilidade	O software deve ser escrito de forma que possa evoluir para atender às necessidades dos clientes.
Confiança e proteção	Confiabilidade, proteção e segurança devem fazer parte de um software. Um software confiável não deve causar prejuízos físicos ou econômicos no caso de falha do sistema e usuários maliciosos não devem ser capazes de acessar ou prejudicar o sistema.
Eficiência	O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto eficiência inclui capacidade de resposta, tempo de processamento, uso de memória etc.
Aceitabilidade	Ele deve ser aceitável para o tipo de usuário para o qual foi projetado, isso significa que deve ser compreensível, usável e compatível com outros sistemas usados por ele.

Atualizam por questões de segurança. Tem mais vírus por causa da quantidade de usuários, quem tem mais vírus é a Microsoft, Linux e depois o Mac IOS.

Qualidade de software - Exemplo

- Correto
 - A loja não pode deixar de cobrar por produtos comprados pelo consumidor.
- Robusto
 - A loja não pode parar de vender.
- Eficiente
 - O consumidor não pode esperar.
 - A empresa quer investir pouco em recursos computacionais.
- Amigável e fácil de usar
 - A empresa quer investir pouco em treinamento.

- **Reutilizável**
 - Várias empresas precisam utilizar partes de um mesmo sistema.
- **Aberto, compatível, de fácil integração com outros sistemas**
 - A empresa já tem controle de estoque, fidelização etc.
- **Portável e independente de plataforma (hardware e software)**
 - A empresa opta por uma determinada plataforma.
- **Baixo custo de instalação e atualização**
 - A empresa tem um grande número de PDVs.

1404

Atividades do processo de desenvolvimento de software

- **Especificação de software** - onde os clientes e engenheiros definem o software que deve ser produzido e as restrições sobre o seu funcionamento.
- **Desenvolvimento de software** - o software é projetado e programado.
- **Validação** - onde o software é verificado para garantir se as necessidades dos clientes foram atendidas.
- **Evolução** - onde ocorrem modificações no software para refletir mudanças de requisitos do cliente e do mercado.

Questões que afetam a maioria dos softwares

- **Heterogeneidade** - cada vez mais, os sistemas são necessários para operar como sistemas distribuídos através de redes que incluem diferentes tipos de computadores e dispositivos móveis.
- **Mudança de negócio e social** - negócio e sociedade estão mudando com uma rapidez incrível, na medida em que as economias emergentes se desenvolvem e as novas tecnologia se tornam disponíveis.
- **Segurança e confiança** - como o software está entrelaçado com todos os aspectos de nossas vidas, é essencial que possamos confiar nele.

2104 – no class

2804

Histórico dos softwares

- **Anos 1940**
 - Iniciou a evolução dos sistemas computadorizados, a maior parte dos esforços, e custos, era direcionada para o desenvolvimento do hardware, devido principalmente das limitações e dificuldades encontradas na época.
- **Anos 1950 - 1965 – Primeira Era**
 - Softwares customizados.
 - Processamento Batch.
- **Anos 1965 – 1975 – Segunda Era**
 - Surgimento dos computadores multiusuários
 - Processamento em tempo real



1940: Tinham muitas limitações de hardware e principalmente de software. Eram máquinas/hardwares monstruosas que faziam uma, duas operações. Eles nem ficavam um dia inteiro, era no máximo 30 minutos e lugar para o dia inteiro era quase impossível.

Primeira Era: o cliente começa a pedir o que ele precisa. Batch – processamento em lote, Digitadores. Tudo era digitado durante um período do dia e a noite esses lotes/arquivos eram processados e atualizava o sistema. Mas poderia fazer as atualizações a qualquer momento, mas para atualizar precisa que ninguém esteja usando e de noite o número de usuários cai. Não pode atualizar um sistema com pessoas usando. Tudo é armazenado num arquivo e de noite é tudo processado.

1965: o mesmo computador era utilizado por várias pessoas com conceito de login e senha. Processamento em tempo real: quando o hardware e o software melhoram isso acontece. O Batch digitava tudo e de noite processava/atualiza e no tempo real manipulando o dado naquele momento há segundos ele aparece atualizado. No saque num banco por exemplo, em segundos aparece que você fez uma transação e no Batch só iria aparecer no dia seguinte.

O computador entendia por meio das linguagens de programação, ele era “ensinado” por meio disso.

A linguagem de baixo nível coloca códigos que somente serão interpretados/entendidos pela máquina e a de alto nível se aproxima da linguagem humana como Java. Com essa melhora, tudo evoluiu digamos assim, mas pessoas ficaram interessadas em programação... Algumas linguagens vão evoluindo e outras caíndo porque não atendem mais as necessidades das pessoas. Na essência todas as linguagens fazem a mesma coisa, mas essas novas linguagens deixam o desenvolvimento de uma aplicação mais fácil. Tem faculdades que dão COBOL, e apesar dela não ser usada mais, tem sistemas programados nessas linguagens antigas e eles precisam de programação. A maior linguagem atual é o CSharp e o Java, mas vão surgindo outras.

Histórico dos softwares

- Anos 1975 – 1985 – Terceira Era
 - Diminuição dos custos dos computadores.
 - Hardware de baixo custo.
 - Demanda crescente por softwares mais complexos
- Processamento distribuído.
- Ano 1985 – Quarta Era
 - Sistemas especialistas.
 - Computação paralela
 - Necessidade da informação estar disponível em curto espaço de tempo para muitas pessoas → INTERNET

Terceira Era: diminuição de custos no mercado externo, principalmente EUA e Europa. Lá todos os equipamentos usados para montar uma área de informática ficou mais barato. A demanda por softwares começou a aumentar, mas não os simples e sim complexos. Complexos do tipo para Usina Nuclear, controle de espaço aéreo... Com isso a computação chegou a mais e mais empresa, porque tinham condições de comprar o equipamento e teriam que ter softwares específicos. Foi o grande boom.

Processamento distribuído: ele não fica centralizado em um único computador, vários computadores podem processar esse dado. Esses vários computadores são responsáveis por processar essa informação.

Quarta Era: computação paralela – tem vários sistemas rodando com comitente, vários sistemas rodando dentro de uma empresa; Sistemas especialistas – evoluem desses softwares complexos e só mudam de nome. São muitos específicos; Necessidade...: rascunho do que é a internet hoje. Se tinha necessidade de obter essa informação. A internet surgiu primeiro com

fins militares – melhorar as comunicações, para atacar a base inimiga de suprimentos alimentares e bélicos, eles atacavam principalmente as comunicações para que quem está sendo atacado não possa pedir ajuda –. Com a internet tinha outros sistemas além das antenas para se pedir ajuda.

A situação no Brasil em termos de tecnologia só começou a mudar em 1990 e em 92 melhorou muito isso porque o Color assinou várias leis que fazia com que as empresas pudessem importar esses equipamentos por um preço mais em conta.

Quinta Era: internet das coisas, inteligência artificial – uma nova realidade que nós temos.

0505

Histórico dos softwares

- Anos 1975 – 1985 – Terceira Era
 - Diminuição dos custos dos computadores.
 - Hardware de baixo custo.
 - Demanda crescente por softwares mais complexos

- Processamento distribuído.

- Ano 1985 – Quarta Era
 - Sistemas especialistas.
 - Computação paralela
 - Necessidade da informação estar disponível em curto espaço de tempo para muitas pessoas → INTERNET

Na época do surgimento do DVD era muito caro mas com a popularização dele foi barateando.

Aplicações de software

■ Software básico

- É uma coleção de programas escritos para dar apoio a outros programas.

- Exemplos

- Compiladores.
- Editores.
- Utilitários para gerenciamento de arquivos



Todas as eras influenciaram no conjunto de softwares que a gente tem hoje. Tem muitos aplicativos porque a demanda aumentou muito.

IEE – normatiza tudo que diz respeito a hardware e de 2019 para cá a parte de software. As publicações são muito utilizadas. Python, Java, C++ e Java Script -> 4 linguagens mais utilizadas/populares de 2020 no ranking feito pelo IEE (estadunidense).

Software Básico: programa na linguagem C precisa ser compilado/traduzido da linguagem de alto nível para baixo nível (linguagem de máquina) e quem faz isso é o compilador (imagem).
Alto nível: próxima da linguagem do dia a dia.

Software de tempo real

- Software que monitora/analisa/controla eventos do mundo real, mas com tempo de resposta em tempo real (milissegundo a 1 minuto).



Monitorar precisa ter uma resposta muito rápida. O caixa tem que estar ligado ao servidor para fazer a transação mais rapidamente possível. Os semáforos inteligentes monitoram o tráfego de veículos e pode ficar mais tempo no verde ou no vermelho – qualquer problema com a central ou energia ele fica no amarelo piscante que já está programado dentro do software para deixar nessa cor para avisar que está ruim –. Se qualquera um desses softwares parar de funcionar vai ter um transtorno gigante. Pode fazer uma manutenção mas vai ficar fora do ar – no caso de banco 24h vai ficar indisponível do mesmo jeito. Com exceção do banco todos eles são sistemas embarcados.

Monitorar precisa ter uma resposta muito rápida. O caixa tem que estar ligado ao servidor para fazer a transação mais rapidamente possível. Os semáforos inteligentes monitoram o tráfego de veículos e pode ficar mais tempo no verde ou no vermelho – qualquer problema com a central ou energia ele fica no amarelo piscante que já está programado dentro do software para deixar nessa cor para avisar que está ruim –. Se qualquera um desses softwares parar de funcionar vai ter um transtorno gigante. Pode fazer uma manutenção mas vai ficar fora do ar – no caso de banco 24h vai ficar indisponível do mesmo jeito. Com exceção do banco todos eles são sistemas embarcados.

Software comercial

- São aplicações que processam e disponibilizam informações para apoiar a tomada de decisão pelos gestores.
 - Sistemas de folha de pagamento.
 - Contas a pagar/receber.
 - Controle de estoque.
 - Transações em ponto de venda.

Rodam nos mercados, escolas, shopping... normalmente eles já estão prontos e compram e utilizam, mas nada impede que você pegue esse software pronto e modifique um pouco/customize mas tem que ter a licença desse software. 70 a 80% dos softwares são comerciais.

Software científico e de engenharia

- São aqueles que auxiliam as aplicações científicas, tem como características algoritmos de intenso processamento de números e cálculos.
- Exemplos
 - Astronomia
 - Vulcanologia.
 - Análise da fadiga mecânica de automóveis.
 - Biofísica molecular

Software comercial

- São aplicações que processam e disponibilizam informações para apoiar a tomada de decisão pelos gestores.
- Sistemas de folha de pagamento.
- Contas a pagar/receber.
- Controle de estoque.
- Transações em ponto de venda.

É aquele software especialista. Análise de fadiga: Qual é o ponto que ele vai parar de funcionar; Vida útil de componentes de segurança, até que ponto se pode usar para não interferir na segurança.

Software embarcado

- Reside na memória de leitura e é usado para controlar produtos e sistemas para os mercados industriais e de consumo. Executam funções limitadas e particulares.

Software científico e de engenharia

- São aqueles que auxiliam as aplicações científicas, tem como características algoritmos de intenso processamento de números e cálculos.
- Exemplos
 - Astronomia
 - Vulcanologia.
 - Análise da fadiga mecânica de automóveis.
 - Biofísica molecular

É um software que não tem um computador para rodar, ele vai rodar de um tipo de memória de leitura desses equipamentos. Micro-ondas, as TV's Smart, Smartwatch... o software reside na memória de leitura e executa funções limitadas, mas alguns têm possibilidade de atualização como a da TV porque está ligado à internet. Todos são passíveis de atualização, mas fáceis para o que tá conectado na internet.

Software de computador pessoal

Tornou-se muito procurado e continua a representar os mais inovadores projetos de interface com seres humanos de toda a indústria de software.

Software embarcado

Reside na memória de leitura e é usado para controlar produtos e sistemas de consumo. Exemplos: televisões, Smartwatches, etc.

Softwares Aplicativos. Dão apoio aos usuários, criação de apresentações, objetos, animação, documentos... Reindenizar: colocou todas as ações e mandou executar.

Software de inteligência artificial

Faz uso de algoritmos não-numéricos para resolver problemas complexos que não sejam favoráveis à computação ou à análise direta.

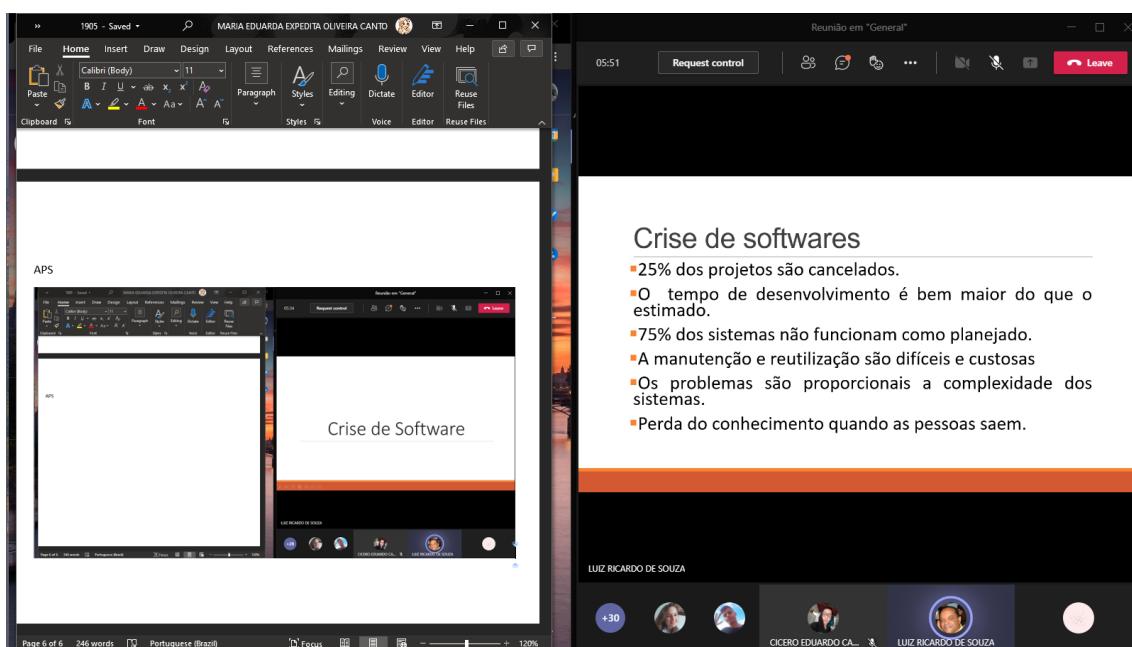
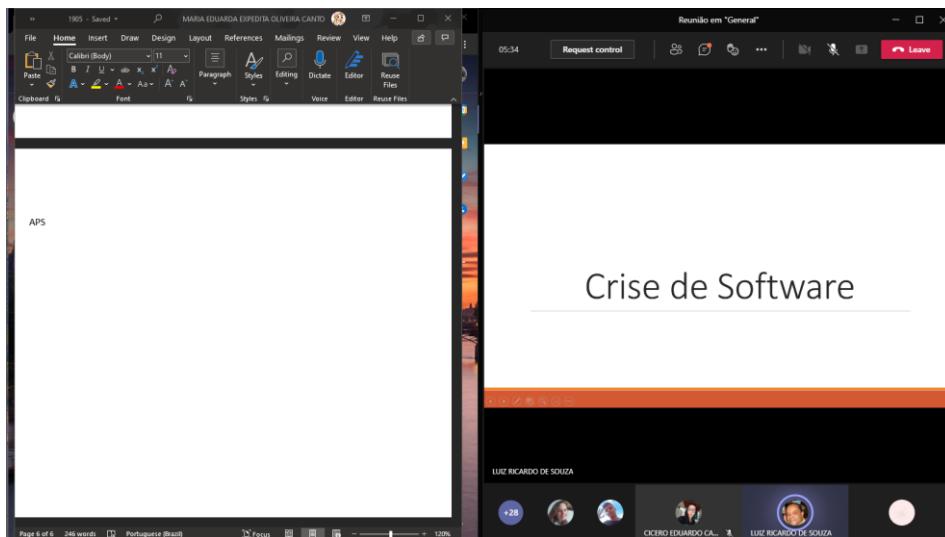
São softwares muito específicos. Aquelas câmeras que ficam no metrô dão para ver bem o seu rosto. Tem hardware que suporta essas aplicações e antigamente não tinha.

1205

Imagens, tabelas... ele decide no dia da apresentação quem vai apresentar e ele vai fazer perguntas. Palavras em inglês coloca em itálico. Onde vai a imagem? Numeração de imagens/figuras, imagens de boa qualidade e do mesmo tamanho. A apresentação é entre 25

e 30 minutos – 20 de apresentação e 10 de perguntas, por exemplo. !1 Como fazer apresentação, dicção... Entregar: 21/07

1905



Os projetos são cancelados por falta de atendimento as reais necessidades da empresa, falta de capacidade técnica da equipe desenvolvedora, custo...

Tempo de desenvolvimento: tendo um prazo para o desenvolvimento, você tem que achar o tempo ideal usando métricas, mas não pode subestimar ou superestimar.

Teste: Não pode programar em um dia, e no dia seguinte entregar pro cliente, um software precisa ser testado e não adianta ser quem programou testar, precisa ser uma pessoa que não escreveu nada daquele código, de preferência um usuário.

Funcionamento: erros na hora do levantamento de requisitos – saber a necessidade do cliente corretamente – o software é do cliente e tem que atender os requisitos dele, você pode achar

algo melhor para se fazer, uma sugestão de aprimoramento, mas tem que falar com o usuário primeiro. O desenvolvedor não pode ficar com dúvida, tem que falar com o cliente quantas fezes for necessário.

Manutenção: se o cliente falou algo errado, na hora da prototipagem pode ocorrer N erros, e se ele quiser que você altera, ele teria que pagar. As vezes a manutenção nem vale a pena, o cliente pode querer não fazer essa manutenção/atualização.

Complexidade: primeiro tem que saber utilizar a ferramenta e depois usufruir de todos os recursos.

Perda: todo o grupo tem que deter as informações, porque se estiver na mão de uma única pessoa, porque se ela sair, o grupo fica órfão e não tem mais conhecimento. Se só tiver uma pessoa que saiba mexer em uma linguagem de programação, e ela sair, vai demandar tempo para achar outra ou alguém aprender e os prazos vão ficar preocupantes; Regra de negócio: o que o sistema vai fazer, terá de executar.

Essas crises ocorrem durante o processo de desenvolvimento do software e tem que prevenir para quer nenhuma delas aconteça.

Todos os contratos devem ser feitos no papel, para não ter problemas. Nunca faça contrato de boca, é sempre no papel, documentação. Evite arrumar computadores de amigos, porque se não funcionar, a culpa é sua.

Delphi orientado a objetos – trabalho

O BANCO DE DADOS PRECISA DE UMA APLICAÇÃO EM OUTRA LINGUAGEM PARA ELES FUNCIONAREM EM CONJUNTO, UM SOZINHO NÃO FAZ NADA. Um complementa um outro, um precisa do outro.

Perda: todo o grupo tem que deter as informações, porque se estiver na mão de uma única pessoa, porque se ela sair, o grupo fica órfão e não tem mais conhecimento. Se só tiver uma pessoa que saiba mexer em uma linguagem de programação, e ela sair, vai demandar tempo para achar outra ou alguém aprender e os prazos vão ficar preocupantes; Regra de negócio: o que o sistema vai fazer, terá de executar.

Essas crises ocorrem durante o processo de desenvolvimento do software e tem que prevenir para quer nenhuma delas aconteça.

Todos os contratos devem ser feitos no papel, para não ter problemas. Nunca faça contrato de boca, é sempre no papel, documentação. Evite arrumar computadores de amigos, porque se não funcionar, a culpa é sua.

Delphi orientado a objetos – trabalho

O BANCO DE DADOS PRECISA DE UMA APLICAÇÃO EM OUTRA LINGUAGEM PARA ELES FUNCIONAREM EM CONJUNTO, UM SOZINHO NÃO FAZ NADA. Um complementa um outro, um precisa do outro.

Causas da crise de software

- **Essenciais**
 - Complexidade dos sistemas
 - Dificuldade de formalização
- **Acidentes**
 - Má qualidade dos métodos, linguagens, ferramentas, processos e modelos de ciclo de vida
 - Falta de qualificação técnica

LUIZ RICARDO DE SOUZA

+35 BEATRIZ DAMAS LUIZ RICARDO DE SOUZA

Complexidade: sistema como um de bolsa de valores, um sistema de folha de pagamento... tem inúmeras variáveis para o desenvolvimento, que podem ser externas e conversam com o software.

Formalização: não entender o que o usuário precisa. Se você não está entendendo, tem que pedir para explicar de novo para ficar claro e desenvolver o sistema corretamente.

Nos acidentes está mais voltado para a equipe de desenvolvimento: falta de experiência, conhecimento da ferramenta que vai ser utilizadas.

Má qualidade de métodos: desenvolvimento e levantamento, metodologia ágil ou tradicional? Qual a mais indicada?

IMPORTANTE!

The screenshot shows a Microsoft Word document on the left and a Microsoft Teams meeting interface on the right. The Word document contains two paragraphs of text. The Teams interface shows a video call with three participants: Beatriz Damas, LUIZ RICARDO DE SOUZA, and another participant whose name is partially visible. The meeting title is "Reunião em 'General'".

Nos acidentes está mais voltado para a equipe de desenvolvimento: falta de experiência, conhecimento da ferramenta que vai ser utilizadas.

Má qualidade de métodos: desenvolvimento e levantamento, metodologia ágil ou tradicional? Qual a mais indicada?

Crise de software

- Estimativas de prazo e de custo
- Produtividade das pessoas
- Qualidade de software
- Software difícil de manter

LUIZ RICARDO DE SOUZA

+34 BEATRIZ DAMAS LUIZ RICARDO DE SOUZA

Estimativas: os desenvolvedores subestimam ou superestimam, normalmente subestimam.

Você desenvolveu um projeto, mas o prazo acabou e ainda falta 20% de desenvolver. Custo: se você colocou um preço e no meio do desenvolvimento viu que não ia dar para terminar com aquele dinheiro, você vai ter prejuízo. Essas coisas têm que ser justas, tem que ver se tem que contratar mais pessoas, se as linguagens de programação são boas o suficiente para desenvolver ou vai ter que aprender outra. Tem que entender a necessidade do cliente para fazer esses levantamentos e atender as necessidades do desenvolvedor.

Produtividade: é um fator de risco. Essas pessoas para fazer projeto, são contratadas e pagas por hora. Tem que ficar em cima daquele empregado para ver se ele está fazendo no tempo correto, isso é tarefa da gestão do projeto, para ver se a pessoa não está sendo coro mole e não vai dar para entregar no prazo, tem que medir essa produtividade de tempo em tempo. É o quanto o programador está sendo eficaz por dia, quanto por dia cada empregado está codificando. O difícil é o convívio com as pessoas, não aprender a desenvolver.

Qualidade: o software tem que ser de qualidade e para isso tem que fazer muito testes do software, mas se estiver dando problema, pode ser do hardware.

Difícil de manter: fazer um software que vai ter atualização, mas não algo muito caro. Ele tem que ser fácil de administrar, atualizar e fazer manutenções.

Especificação formal: transformar o pedido do cliente em um diagrama, converter a necessidade para um diagrama.

2605

16/06 prova e 23/06 recuperação

Semana do dia 14 e 21 = muitas atividades

Crise de softwares

- 25% dos projetos são cancelados.
- O tempo de desenvolvimento é bem maior do que o estimado.
- 75% dos sistemas não funcionam como planejado.
- A manutenção e reutilização são difíceis e custosas
- Os problemas são proporcionais a complexidade dos sistemas.
- Perda do conhecimento quando as pessoas saem.

Protocolo: conjunto de regras/normas para definir uma situação.

Regra de negócio: o que aquele software vai realmente fazer

Causas da crise de software

- Essenciais
 - Complexidade dos sistemas
 - Dificuldade de formalização
- Acidentes
 - Má qualidade dos métodos, linguagens, ferramentas, processos e modelos de ciclo de vida
 - Falta de qualificação técnica

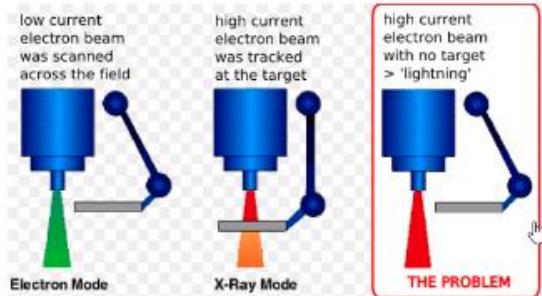
Tem 13 diagramas para fazer a programação direito, linguagem UML – definir os diagramas necessários.

Problemas que essa crise do software agravou:

Therac-25

Equipamento de radioterapia

- Entre 1985 e 1987, ocorreram 6 acidentes causando mortes por overdoses de radiação.
 - Falhas por falta de testes integrados.
 - Falta de documentação.



Aconteceu 6 mortes causadas por overdose de radiação – o equipamento aumentou em muito o nível de radiação. O grande problema foi a falta de testes e documentação. Testes em um equipamento com esse é de suma importância, tem que testar exaustivamente, esse software tem que ser muito testado, já que é para saúde e em softwares mais complexos. Isso foi considerado crime, já que ocorreram mortes.

Aeroporto Internacional de Denver

- Custo do projeto US\$ 4.9 bi.
- 100 mil passageiros/dia.
- 1,200 voos.
- 53 milhas quadradas.
- 94 portões de embarque/desembarque.
- 6 pista de pouso/decolagem.
- Erros no sistema automático de transporte de bagagens.
- Atraso na abertura do aeroporto com custo estimado em US\$ 360 milhões.
- US\$ 86 milhões para corrigir o sistema.



Ariane 5 – O que aconteceu ?

- Explosão após 40 segundos da decolagem.
- O veículo detonou suas cargas explosivas de autodestruição. O foguete perdeu o controle de direção
- Culpa ?
 - Os computadores principais e de backup deram shutdown ao mesmo tempo. Ocorreu um run time error em um programa que convertia um valor em ponto flutuante para inteiro de 16 bits (recebeu um valor fora da faixa permitida).

O run time error foi o problema que fez com que os dois computadores principais desligassem – shut down. Linha de código responsável pelo problema, e nem precisava dela:

Ariane 5 – O que aconteceu ?

strict precondition 1:

```
{      Set."x"=FLPT and Set."y"=INT16  
and -32768 <= x <= +32767  
}
```

program code:

```
y := int(x);
```

postcondition:

```
{Set."x"=FLPT and Set."y"=INT16 and y=int(x)}
```

Isso aconteceu depois de 40s de decolagem:

Ariane 5 – Projeto Espacial Europeu

US\$ 8 bilhões.

10 anos

Garantiria a supremacia europeia no espaço.



Engenharia de Software

- É uma forma de engenharia que aplica os princípios da Ciência da Computação e Matemática para alcançar soluções com melhor custo-benefício para problemas de software.
- Os **fundamentos científicos** para engenharia de software envolvem o uso de modelos abstratos e precisos que permitem ao engenheiro **especificar, projetar, implementar e manter sistemas de software**, avaliando e garantindo suas **qualidades**.

Criar um software depende muito.

0206

The screenshot shows a Microsoft Word document window. The title bar says "0206 - Last Modified: 1... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO". The ribbon menu includes Home, Insert, Draw, Design, Layout, References, Mailings, Review, View, and Help. The main content area displays a slide with the following text:

Engenharia de Software

■ É uma forma de engenharia que aplica os princípios da Ciência da Computação e Matemática para alcançar soluções com melhor custo-benefício para problemas de software.

■ Os **fundamentos científicos** para engenharia de software envolvem o uso de modelos abstratos e precisos que permitem ao engenheiro **especificar, projetar, implementar e manter sistemas de software**, avaliando e garantindo suas **qualidades**.

At the bottom of the slide, there is a footer with names: LUIZ RICARDO DE SOUZA, JOAO VICTOR GONCALVES, and LUIZ RICARDO DE SOUZA. Below the footer, there are icons for +33, AA, JS, and a video camera. The status bar at the bottom right shows "Page 1 of 5" and "Portuguese (Brazil)".

É aplicar métodos científicos para conseguir desenvolver softwares, os métodos envolvem encontro com os usuários, criação de diagramas.... Normatização dos hardwares (que vem na caixa) você sabe que aquele equipamento está de acordo com aquele tipo de rede, e agora vem crescendo essa normatização para software -> W3C ela que normatiza e especificar as condições para o desenvolvimento do html 5, normatiza essa parte para web, o que ajuda a padronizar e ter um desenvolvimento mais uniforme.

The screenshot shows a Microsoft Word document window. The title bar says "0206 - Saved - MARIA EDUARDA EXPEDITA OLIVEIRA CANTO". The ribbon menu includes Home, Insert, Draw, Design, Layout, References, Mailings, Review, View, and Help. The main content area displays a slide with the following text:

Mitos de softwares

At the bottom of the slide, there is a footer with names: LUIZ RICARDO DE SOUZA, JOAO VICTOR GONCALVES, and LUIZ RICARDO DE SOUZA. Below the footer, there are icons for +33, AA, JS, and a video camera. The status bar at the bottom right shows "Page 5 of 5" and "263 words" and "Portuguese (Brazil)".

The screenshot shows a Microsoft Teams meeting interface on the left and a Microsoft Word document on the right. The meeting title is "Reunião em 'General'". The Word document is titled "0206 - Saving..." and is being edited by "MARIA EDUARDA EXPEDITA OLIVEIRA CANTO".

Administrativo

Já temos um manual repleto de padrões e procedimentos para a construção de software. Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?

Realidade:
Será que o manual é usado?
Os profissionais sabem que ele existe?
Ele reflete a prática moderna de desenvolvimento de software?
Ele é completo?

Mitos de softwares

Existe muitas crenças para conseguir desenvolver softwares, os métodos envolvem entendimento de processos, criação de diagramas, e muitos outros que nem sempre são verdade. Mas é comum que essas crenças existam e se acredite que elas são verdadeiras. Vamos desmontar algumas delas.

É comum pensar que para desenvolver um software é necessário ter muitos recursos, mas isso não é verdade. Um bom software pode ser criado com poucos recursos e tempo.

Page 5 of 5 264 words Portuguese (Brazil)

O manual de software é muito importante, mas quase ninguém usa. Tem que criar esse manual, mas as pessoas acabam não utilizando.

The screenshot shows a Microsoft Teams meeting interface on the left and a Microsoft Word document on the right. The meeting title is "Reunião em 'General'". The Word document is titled "0206 - Saved..." and is being edited by "MARIA EDUARDA EXPEDITA OLIVEIRA CANTO".

Administrativo II

Meu pessoal tem ferramentas de desenvolvimento de software de última geração, afinal compramos os mais novos computadores.

Realidade:
É preciso muito mais do que os mais recentes computadores para se fazer um desenvolvimento de software de alta qualidade.

Mitos de softwares

Existe muitas crenças para conseguir desenvolver softwares, os métodos envolvem entendimento de processos, criação de diagramas, e muitos outros que nem sempre são verdade. Mas é comum que essas crenças existam e se acredite que elas são verdadeiras. Vamos desmontar algumas delas.

É comum pensar que para desenvolver um software é necessário ter muitos recursos, mas isso não é verdade. Um bom software pode ser criado com poucos recursos e tempo.

Page 6 of 6 286 words Portuguese (Brazil)

The image shows a Microsoft Word document window. The title bar says 'Reunião em "General"' and the status bar shows '01:19'. The main content is a slide titled 'Administrativo III' with the following text:

Realidade:
O desenvolvimento de software não é um processo mecânico igual à manufatura.
Acrescentar pessoas em um projeto torna-o ainda mais atrasado. Pessoas podem ser acrescentadas, mas somente de uma forma planejada.

Below the slide, there's a footer with names: LUIZ RICARDO DE SOUZA, JOSE FELIPE HIGINO A..., and LUIZ RICARDO DE SOUZA again. There are also icons for +35, a profile picture, JS, JA, and a video camera.

To the right of the Word window, a screenshot of a presentation slide is visible. The slide has a dark background and contains text from the Word document, along with a small image of a person.

Na linha de produção/fábril se precisar aumentar a produção aumenta o número de empregados. Só que no desenvolvimento de software, você tem um projeto de 5 pessoas fazendo um projeto de software e 2 saem, é melhor contratar alguém ou treinar alguém internamente? É melhor treinar porque a pessoa já está em contato com o projeto (as atribuições da vaga da pessoa que saiu vão ser distribuída entre o resto do grupo, mas vai continuar sem uma pessoa, a vaga não vai sumir), já contratar demora muito tempo e dinheiro.

0906

Processo de desenvolvimento de software – engenharia de software. Cada um fazia do jeito que achava que era correta e a engenharia veio para normatizar isso, a engenharia nada mais é que a normatização de um desenvolvimento de software, veio organizar esse desenvolvimento – desenvolvimento dos diagramas, por onde começa, levantamento de requisitos. Em todo mundo se usa um padrão.

O controle de tráfego aéreo é crítico e complexo, sistemas hospitalares são complexos e essa complexidade tem a ver com o desenvolvimento e também tem que ter um entendimento do que o cliente quer e se ele se expressou direito e se o desenvolvedor entendeu o que o usuário quer.

A matéria é para ensinar como, quais os passos para desenvolver um sistema/app/aplicação seja ela qual for.

Algumas empresas chamam de engenheiro de software e outras de desenvolvedor, mas no rústico, eles fazem praticamente a mesma coisa que é criar/programar.

O sistema BET é um sistema de processamento, desenvolvia um sistema e durante um dia inteiro/parte do dia as informações eram alimentadas nesse sistema e o processamento do BET e a noite o sistema seria atualizado e esses dados que forma digitados seriam jogados na base de dados e só apareceria para consulta no dia seguinte, mas poderia levar mais dias. A atualização é pegar esses dados numa área temporária, como um cache e depois seria jogado

na base de dados. O sistema de TEMPO REAL você digita agora e quando clica em salvar esse dado já está disponível.

Na evolução se corrigi bugs das outras versões do software, pode melhorar a aparência, funcionalidades...

As empresas eram muito focadas em aplicações desktop, tinha que ir à empresa para ver o sistema da empresa em outro país. Distribuir os sistemas para o desktop e celular, o conceito de nuvem – grande parte dos sistemas estão na nuvem então de todo parte do mundo você pode acessar e de qualquer dispositivo, não só pelo computador = heterogeneidade – diferentes formas de acessar o sistema da sua empresa, por exemplo.

Nos já estamos na quinta era que começa com a questão do AI e realidade aumentada, mas não se sabe exatamente quando terminou a quarta era.

1606

Proavelmente foi o dia da prova

2306 – palestra?

3006 – palestra?

0707 – no class

1407 – no class

2107 – explicação?

2807

Trabalho

0408

Trabalho

1108

Trabalho

1808

Trabalho

2508

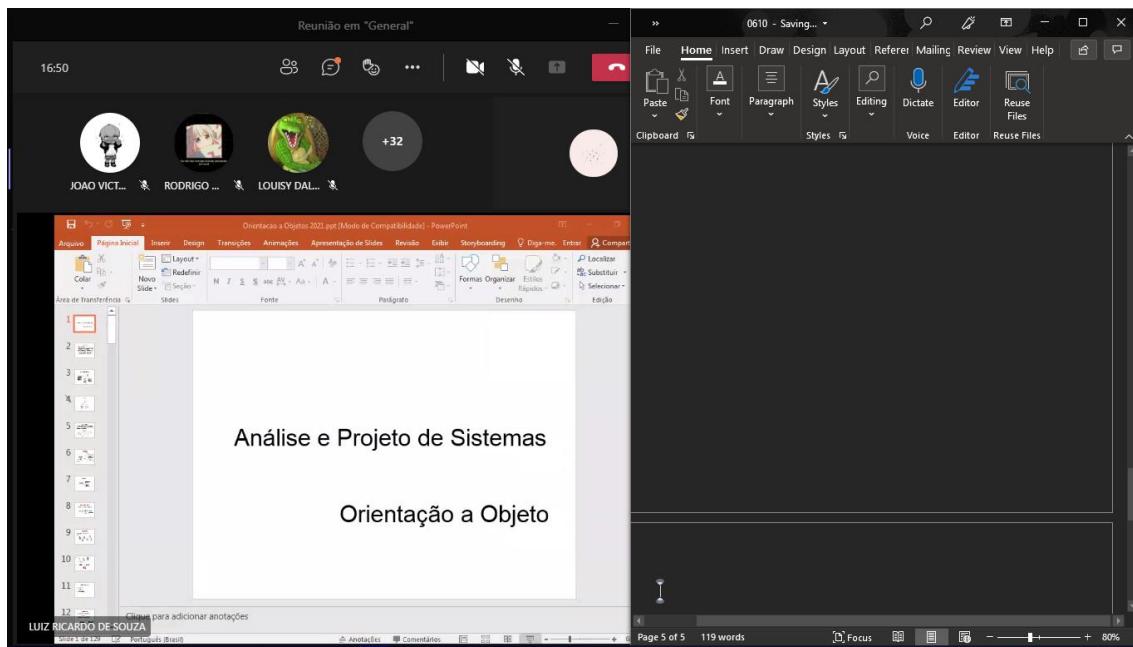
Trabalho

0109

Trabalho

2209

0610



A única coisa que não é Orientação a Objetos é o Banco de Dados porque foi criado com a finalidade de atender linguagens estruturais, já que ele é relacional.

A screenshot showing a Microsoft Teams meeting titled "Reunião em 'General'". In the top left, there's a small calendar icon with the date 46. The top right has a red "Leave" button. Below the title bar, there are icons for camera, microphone, and other meeting controls. The main area shows three participant icons: CICERO ED..., LUIZ RICARDO..., and LOUISY DAL... A circular "You're muted. Press Ctrl+Shift+M to unmute your microphone." message is displayed. The bottom status bar shows "RICARDO DE SOUZA" and "Clique para adicionar anotações". To the right of the Teams window is a Microsoft Word document titled "Orientação a Objetos 2022.ppt [Modo de Compatibilidade] - PowerPoint". The slide content includes the title "Conceitos" and a bulleted list: "• Classes e Instâncias", "– Classe é uma categoria geral de objeto que descreve um conjunto de objetos específicos similares que são suas instâncias.", "– É um elemento em OO que descreve o que seus objetos têm e fazem.", "– Servem de base/molde para a criação de novos objetos que serão as instâncias.", "– Instância é um sinônimo para o termo objeto.". The Word ribbon tabs are visible at the top of the document window.

Na linguagem estrutural o programa segue “mais ou menos” que nem no html, faz um bloco de comandos e o compilador ia lendo linha por linha para executar o programa.

Tudo que for característico é da Classe, a classe cliente tem o que caracteriza cliente: nome, endereço, telefone... A Classe Aluno tem nome, RM, e os alunos são as instâncias. A Entidade tem os atributos que são as características e a Classe também vai ter os atributos, mas em Classe tem algo a mais que os atributos e a entidade, que é o método/operação = alguma

função que você vai fazer com aqueles atributos – o Classe Aluno, abaixo dela vai ter as ações como incluir, alterar, deletar...

Tem a Classe de molde para os atributos e as instâncias são os valores que os atributos vão receber/estão recebendo.

The screenshot shows a Microsoft Teams meeting interface. In the top left, there's a video feed of a man named LUIZ RICARDO DE SOUZA. To his right, there are other participants and a '+34' badge. The main window displays a PowerPoint slide with the title 'Conceitos de OO'. The slide content includes a diagram illustrating classes and instances with three cat images. Below the slide, a note says: 'Classe é aquela que tem nome, RM, e os alunos são as instâncias. A Entidade tem os atributos associados às características e a Classe também vai ter os atributos, mas em Classes vemos mais os atributos e a entidade, que é o método/operação – algumas funções que você vai fazer com aqueles atributos – o Classe Aluno, abaixo dela vai ter as ações como incluir, alterar, deletar...'. The bottom right corner of the slide shows 'Page 6 of 6 285 words'.

A Classe seria felino e as instâncias seria os “tipos” de felinos.

This screenshot shows a continuation of the Microsoft Teams meeting. The video feed now shows a woman named RODRIGO. The PowerPoint slide has been updated to say 'A Classe seria felino e as instâncias seria os “tipos” de felinos.' The slide also contains a diagram showing 'instâncias' connected to a central box labeled 'Nome', 'atributos', and 'operações', which is further connected to a box labeled 'dessa'.

Reunião em "General" 01:02:27

Atributos

Objeto

Classe

Nome do Atributo

Valor do Atributo

`Professor`

`titulo nome`

`Professor`

`titulo = "Mestre"
Nome = "Gibson"`

`Professor`

`titulo = "PdD"
nome = "Ivan"`

LUIZ RICARDO DE SOUZA +34 RODRIGO ...

Orientação a Objetos 2021.pdf [Modo de Compatibilidade] - PowerPoint

Page 6 of 6 297 words Focus

Reunião em "General" 01:10:31

Operações

Especificação das operações

`Professor`

`criar()
atualizar()
getTitulo()
getNome()`

LUIZ RICARDO DE SOUZA +35 HENRIQUE ...

Orientação a Objetos 2021.pdf [Modo de Compatibilidade] - PowerPoint

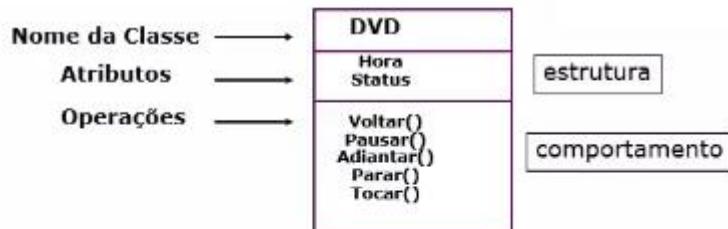
Page 7 of 7 297 words Focus

Os métodos abre e fecha parênteses.

Resumo de classes



- Uma classe é composta por três seções



B

2010

The screenshot shows a Microsoft Word document window. At the top, there is a video conference interface with participants' names: CICERO ED..., RODRIGO..., LOUISY DAL..., and HEITOR AU... A red box highlights the number '+33'. The main content area of the Word document displays a slide with the following text:

ORIENTAÇÃO A OBJETOS
PROFESSOR LUIZ RICARDO

Below the text is a decorative graphic consisting of a grid of white ovals on a black background. The Word ribbon is visible at the top, showing tabs like File, Home, Insert, Draw, Design, Layout, References, Mailings, Review, View, and Help. The status bar at the bottom indicates "Page 6 of 6" and "118 words".

Reunião em "General" 15:16 Request control Leave

CICERO ED... RODRIGO... LUIZ RICARDO... ALBERTO ZL... +33

ABSTRAÇÃO

Page 6 of 6 118 words Portuguese (Brazil) Focus 100%

Tem o mesmo objeto, o gato, com duas visões diferentes. A abstração é compreender o objeto e ver para que ele serve. Logo, tem que programar pensando no objeto, que seria o gato. Tem que imaginar o que esse programa vai realizar para você.

Reunião em "General" 25:20 Request control Leave

LUIZ RICARDO... JOSE FELIPE... CICERO ED... RODRIGO... +35

ABSTRAÇÃO

Mundo REAL

Problema MODELADO

```

graph LR
    A[CARRO DE SOM] -- tem medo --> B[JOÃO]
    B -- toma --> C[MAMADEIRA]
  
```

Page 6 of 6 162 words Portuguese (Brazil) Focus 100%

Reunião em "General"

26:46 Request control Leave

LUIZ RICARDO ... JOSE FELIPE ... HENRIQUE ... RODRIGO ... +35

DOMÍNIO E APLICAÇÃO

Dominio Bancário

CLIENTE JOSE
CLIENTE MARIA
CLIENTE JOÃO
FUNCIONÁRIO JONAS
FUNCIONÁRIO MARCELO
FUNCIONÁRIO RAFAEL

Page 7 of 7 162 words Portuguese (Brazil)

Domínio é a aplicação, o domínio bancário é o sistema bancário. As contas das pessoas são ocorrências da classe conta. Os clientes são ocorrências da classe cliente...

Reunião em "General"

29:20 Request control Leave

LUIZ RICARDO ... JOSE FELIPE ... HENRIQUE ... CICERO ED... +35

DEFINIÇÃO DE OBJETO

- É uma entidade real ou abstrata, que modela um conceito presente na realidade humana, ocupando espaço físico ou lógico.
- Um objeto é uma pessoa, um lugar, é a base para todos os outros conceitos da orientação a objetos.
- Facilita a compreensão do mundo real e oferece uma base real para implementação em computador.
- Um objeto denota uma entidade de natureza física, conceitual ou de software:
 - Entidades físicas: um carro, uma pessoa, um livro.
 - Entidade conceitual: um DER de um sistema.
 - Entidade de software: um radiobutton em uma página web.

Dominio é a aplicação, o domínio bancário é o sistema bancário.

Page 7 of 7 173 words Portuguese (Brazil)

As contas são abstratas e as pessoas são reais, mas todos são objetos.

Reunião em "General"

40:40 Request control Leave

LUIZ RICARDO ... BEATRIZ DA... JOSE FELIPE ... CICERO ED...

+35

CLASSES

- Uma classe é o projeto de um objeto.
- Uma classe representa uma categoria e os objetos são membros dessa categoria.
- Classe é a representação de um conjunto de coisas reais ou abstratas que são reconhecidas como sendo do mesmo tipo.
- Uma classe é considerada uma fábrica de instâncias que inclui atributos e operações dessas instâncias.
- É importante evitar a criação de classes que tentem abranger tudo (ou várias coisas).

Conta
numero
saldo
limite
saca()
deposita()
imprimeExtrato()

DEFINIÇÃO DE OBJETO

As contas são abstratas e as pessoas são reais, mas todos são objetos.

Page 8 of 8 186 words Portuguese (Brazil) Focus

Os atributos número, saldo e limite são da conta. Quando criar o objeto número do João vai ter o número, saldo e limite, já que é tipo uma forma que serve de base para os objetos.
Instancia = objeto.

Reunião em "General"

44:06 Request control Leave

LUIZ RICARDO ... RODRIGO ... BEATRIZ DA... CICERO ED...

+34

CLASSES

Classe Pessoa (grupo de objetos similares que compartilham atributos e comportamentos)

Objetos: ocorrências de uma classe

DEFINIÇÃO DE OBJETO

As contas são abstratas e as pessoas são reais, mas todos são objetos.

CLASSES

- Uma classe é o projeto de um objeto.
- Uma classe representa uma categoria e os objetos são membros dessa categoria.
- Classe é a representação de um conjunto de coisas reais ou abstratas que são reconhecidas como sendo do mesmo tipo.
- Uma classe é considerada uma fábrica de instâncias que inclui atributos e operações dessas instâncias.
- É importante evitar a criação de classes que tentem abranger tudo (ou várias coisas).

Conta
numero
saldo
limite
saca()
deposita()
imprimeExtrato()

Os atributos número, saldo e limite são da conta. Quando criar o objeto número do João vai ter o número, saldo e limite, já que é tipo uma forma que serve de base para os objetos.
Instancia = objeto.

Page 8 of 8 225 words Portuguese (Brazil) Focus

O que vai falar se algum tipo de pessoa vai entrar nessa classe vai depender dos atributos que você colocou.

Reunião em "General" 44:55 Request control Leave

LUIZ RICARDO ... RODRIGO ... BEATRIZ DA... CICERO ED...

ANALOGIA

- Um objeto é como se fosse uma casa ou um prédio.
- Para ser construído, precisa de um espaço físico
- No caso dos objetos, esse espaço físico é algum trecho vago da memória do computador que executa a aplicação.
- No caso das casas e dos prédios, o espaço físico é algum terreno vazio
- Uma classe funciona como uma "receita" para criar objetos. Inclusive, vários objetos podem ser criados a partir de uma única classe
- Assim como várias casas ou prédios poderiam ser construídos a partir de uma única planta; ou vários bolos poderiam ser preparados a partir de uma única receita; ou vários carros poderiam ser construídos a partir de um único projeto

LUIZ RICARDO DE SOUZA

Page 9 of 9 225 words Portuguese (Brazil) Focus 100%

Casas – classe e as pessoas que vão morar dentro seriam os objetos.

Reunião em "General" 45:56 Request control Leave

LUIZ RICARDO ... RODRIGO ... BEATRIZ DA... CICERO ED...

ANALOGIA

The diagram shows a central house icon with several arrows pointing from it to other icons: a car, an open book, and a cake. This visual representation links the concept of a class (the house) to its instances (the car, book, and cake).

LUIZ RICARDO DE SOUZA

Page 9 of 9 237 words Portuguese (Brazil) Focus 100%

Reunião em "General" 01:03:44 Request control Leave

LUIZ RICARDO... JOSE FELIPE... LOUISY DAL... ITALO PEDR...

+34

REPRESENTAÇÃO GRÁFICA DAS CLASSES

```

classDiagram
    class Aluno {
        -nome
        -telefone
        -email
        -dataNascimento
        +fazerMatricula()
        +buscarAlunos()
        +transcarMatricula()
        +alterarAluno()
        +buscarPorId()
    }
    Aluno --> Nome da classe : Nome da classe
    Aluno --> Atributos : Atributos
    Aluno --> Operações ou métodos : Operações ou métodos
  
```

Nome da classe: Minúscula com primeira letra maiúscula.
Ex.: Aluno, Professor e ContaReceber

Atributo: Minúsculo, se necessário primeira letra da segunda palavra maiúscula.
Ex.: nome, telefone e dataNascimento

Métodos: Minúsculo, se necessário primeira letra da segunda palavra maiúscula.
Ex.: inserir(), alterar() e buscarPorId()

Page 10 of 10 273 words English (United Kingdom) Focus 100%

Reunião em "General" 01:05:15 Request control Leave

LUIZ RICARDO... JOSE FELIPE... LOUISY DAL... ITALO PEDR...

+34

PADRONIZAÇÃO DOS NOMES DE UMA CLASSE

```

classDiagram
    class Aluno {
        -nome
        -telefone
        -email
        -dataNascimento
        +fazerMatricula()
        +buscarAlunos()
        +transcarMatricula()
        +alterarAluno()
        +buscarPorId()
    }
    Aluno --> Nome da classe : Nome da classe
    Aluno --> Atributos : Atributos
    Aluno --> Operações ou métodos : Operações ou métodos
  
```

Nome da Classe: Minúscula com primeira letra maiúscula.
Ex.: Aluno, Professor e ContaReceber

Atributo: Minúsculo, se necessário primeira letra da segunda palavra maiúscula.
Ex.: nome, telefone e dataNascimento

Métodos: Minúsculo, se necessário primeira letra da segunda palavra maiúscula.
Ex.: inserir(), alterar() e buscarPorId()

Page 10 of 10 273 words Portuguese (Brazil) Focus 100%

Atributo e Métodos = Quando precisa juntar duas palavras, a primeira letra da segunda palavra é maiúscula.

Reunião em "General"

01:11:19 Request control Leave

LUIZ RICARDO ... JOSE FELIPE ... BEATRIZ DA ... LOUSY DAL ... +34

TIPOS DE DADOS

Aluno

- nome : string
- telefone : string
- email : string
- dataNascimento : date

- +fazerMatricula()
- +buscarAlunos()
- +transcarMatricula()
- +alterarAluno()
- +buscarPorId()

Tipos de Dados:

- ✓ String
- ✓ Integer
- ✓ Boolean
- ✓ Date
- ✓ Double

PADRORIZAÇÃO DOS NOMEIS DE UMA CLASSE

Nome da Classe: Inicia com uma letra maiúscula. Ex: Aluno, Professor e Conferencista.

Nome das Variáveis: Inicia com uma letra maiúscula e pode ter outras letras maiúsculas ou minúsculas. Ex: nome, telefone e dataNascimento.

Nome das Constantes: Inicia com uma letra maiúscula e pode ter outras letras maiúsculas ou minúsculas. Ex: pi, numero_pi e piConstante.

Atributo e Métodos = Quando precisa juntar duas palavras, a primeira letra da segunda palavra é maiúscula.

Page 11 of 11 290 words Portuguese (Brazil) Focus 100%

2710

Reunião em "General"

30:56 Leave

HENRIQUE ... LUIZ RICARDO ... GABRIEL YU ... CICERO ED... ITALO PEDR...

FELIPE FERREIRA DE SOUZA parou em tipos de dados

CRIAÇÃO DO OBJETO (PROGRAMAÇÃO)

Aluno

- nome
- telefone
- email
- dataNascimento

- +fazerMatricula()
- +buscarAlunos()
- +transcarMatricula()
- +alterarAluno()
- +buscarPorId()

Instância da Classe – Momento no código fonte que é gerado uma “Variável” do tipo da classe, que herda a estrutura de dados (atributos) e suas operações (métodos).

SobjAluno = new Aluno;

Aluno

- nome = João da Silva
- telefone = 3444-3333
- email = joao@silva.com.br
- dataNascimento = 2008-01-01

- +fazerMatricula()
- +buscarAlunos()
- +transcarMatricula()
- +alterarAluno()
- +buscarPorId()

Page 1 of 1 0 words Focus 100%

Reunião em "General" 36:48

HENRIQUE ... LUIZ RICARDO ... GABRIEL YU... CICERO ED... ITALO PEDR... +34

CHAMADA DOS MÉTODOS

```

Aluno
-nome
-telefone
-email
-dataNascimento
+fazerMatricula()
+buscarAlunos()
+transcarMatricula()
+alterarAluno()
+buscarPorId()

```

→ \$objAluno = new Aluno;
 → \$objAluno->buscarAlunos();

LUIZ RICARDO DE SOUZA

AutoSave File Home Insert Draw Design Layout References Mailings Review View Help

CRIAÇÃO DO OBJETO (PROGRAMAÇÃO)

Reunião em "General" 40:16

HENRIQUE ... LUIZ RICARDO ... GABRIEL YU... CICERO ED... ITALO PEDR... +34

RETORNO DE MÉTODOS

```

Aluno
-nome
-telefone
-email
-dataNascimento
+fazerMatricula()
+buscarAlunos()
+transcarMatricula()
+alterarAluno()
+buscarPorId()

```

→ \$objAluno = new Aluno;
 → \$objAlunos = \$objAluno->buscarAlunos();

Tipos de Retorno:

- ✓ Boolean
- ✓ Objeto
- ✓ Coleção de Objetos

LUIZ RICARDO DE SOUZA

AutoSave File Home Insert Draw Design Layout References Mailings Review View Help

CRIAÇÃO DO OBJETO (PROGRAMAÇÃO)

Reunião em "General" 43:14

MÉTODOS GET E SET

Aluno

-nome
-telefone
-email
-dataNascimento
+fazerMatricula()
+buscarAlunos()
+transferMatricula()
+alterarAluno()
+buscarPorId()

GET: Buscar dados dentro do objeto
SET: Armazenar dados dentro do objeto

```

function get($atributo)
{
    return $this->$atributo;
}

function set($atributo, $valor)
{
    $this->$atributo = $valor;
}

```

RETOURNO DE MÉTODOS

TIPOS DE RETORNO:

- ✓ Objetos
- ✓ Classes
- ✓ Coleções de Objetos

Page 2 of 2 0 words Focus 100%

Reunião em "General" 44:19

REFERÊNCIA AOS ATRIBUTOS

Aluno

-nome
-telefone
-email
-dataNascimento
+fazerMatricula()
+buscarAlunos()
+transferMatricula()
+alterarAluno()
+buscarPorId()

Referência Fora da Classe

```

$objAluno = new Aluno;
print $objAluno->get('telefone');

```

Referência Dentro da Classe

```

print $this->telefone;

```

RETOURNO DE MÉTODOS

MÉTODOS GET E SET

Aluno	GET: Buscar dado dentro do objeto SET: Armazenar dado dentro do objeto
-nome	function get(\$atributo)
-telefone	{ return \$this->\$atributo;
-email	}
-dataNascimento	function set(\$atributo, \$valor)
+fazerMatricula()	{ \$this->\$atributo = \$valor;
+buscarAlunos()	}
+transferMatricula()	
+alterarAluno()	
+buscarPorId()	

Page 2 of 2 0 words Focus 100%

Reunião em "General" 44:11

CICERO EDUARDO CAMPOS LEITE BISPO
Esse python ta parecendo java script

HENRIQUE ... LUIZ RICARDO ... GABRIEL YU ... CICERO ED... ITALO PEDR...

VISIBILIDADE DA CLASSE, ATRIBUTOS E MÉTODOS

- + ou public → Acesso dentro ou fora da classe instanciada.
- # ou protected → Acesso dentro da classe ou pelas classes dependentes (herança).
- ou private → Acesso somente dentro da classe.
- ~ ou package → Acesso somente dentro do mesmo pacote.

MELO, Ana Cristina 2004

LUIZ RICARDO DE SOUZA

AutoSave Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Styles Voice Editor Reuse Files

Referência Pela Classe
Aluno objAluno = new Aluno();
print objAluno.get("telefone");

Referência Dentro da Classe
print objAluno.get("telefone");

- sharp. O package é muito específico.

Reunião em "General" 54:16

CICERO EDUARDO CAMPOS LEITE BISPO
Esse python ta parecendo java script

HENRIQUE ... LUIZ RICARDO ... GABRIEL YU ... CICERO ED... ITALO PEDR...

VISIBILIDADE DA CLASSE

```
public class Aluno {
    // atributos
    // métodos
}
```

MELO, Ana Cristina 2004

- sharp. O package é muito específico.

AutoSave Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Styles Voice Editor Reuse Files

VISIBILIDADE DA CLASSE, ATRIBUTOS E MÉTODOS

- + ou public → Acesso dentro ou fora da classe instanciada.
- # ou protected → Acesso dentro da classe ou pelo package.
- ou private → Acesso somente dentro da classe.
- ~ ou package → Acesso somente dentro do mesmo pacote.

MELO, Ana Cristina 2004

LUIZ RICARDO DE SOUZA

Reunião em "General"

54:33

HENRIQUE ... LUIZ RICARDO ... GABRIEL YU... CICERO ED... ITALO PEDR...

+34

VISIBILIDADE DA CLASSE E MÉTODO

```
public class Aluno {
    // atributos
    public function buscarAlunos() {
        }
}
```

Aluno

- nome
- telefone
- email
- dataNascimento
- +fazerMatricula()
- +buscarAlunos()
- +transcarMatricula()
- +alterarAluno()
- +buscarPorId()

Page 4 of 4 8 words Focus 100%

Reunião em "General"

55:05

HENRIQUE ... LUIZ RICARDO ... GABRIEL YU... CICERO ED... ITALO PEDR...

+34

VISIBILIDADE DA CLASSE, MÉTODO E ATRIBUTOS

```
public class Aluno {
    private var $nome;
    private var $telefone;
    private var $email;
    private var $dataNascimento;
    public function buscarAlunos() {
    }
}
```

Aluno

- nome
- telefone
- email
- dataNascimento
- +fazerMatricula()
- +buscarAlunos()
- +transcarMatricula()
- +alterarAluno()
- +buscarPorId()

Page 4 of 4 8 words Focus 100%

Esses são exemplos feitos em Phyton

Reunião em "General"

55:33

HENRIQUE ... LUIZ RICARDO ... GABRIEL YU ... CICERO ED... ITALO PEDR... +34

OBJETOS, ATRIBUTOS E MÉTODOS

DOMÍNIO BANCÁRIO APPLICAÇÃO SISTEMA

LUIZ RICARDO DE SOUZA

Reunião em "General"

01:02:23

HENRIQUE ... LUIZ RICARDO ... GABRIEL YU ... CICERO ED... ITALO PEDR... +32

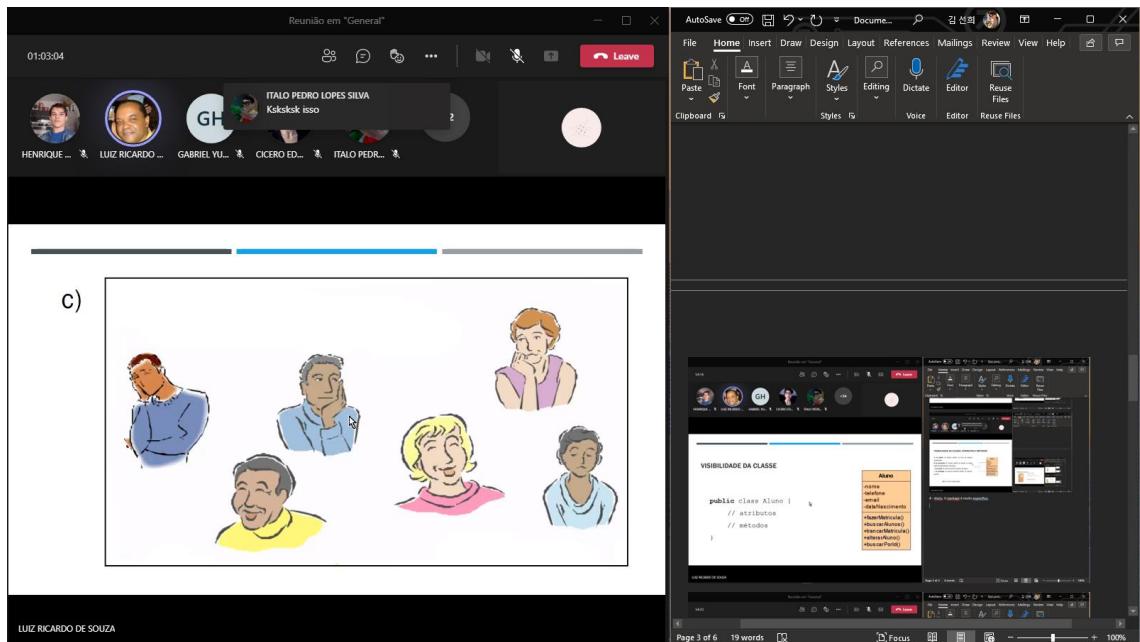
EXERCÍCIOS: IDENTIFIQUE AS CLASSES, ATRIBUTOS E OPERAÇÕES

a)

b)

Esse é um exemplo feito em Python

LUIZ RICARDO DE SOUZA



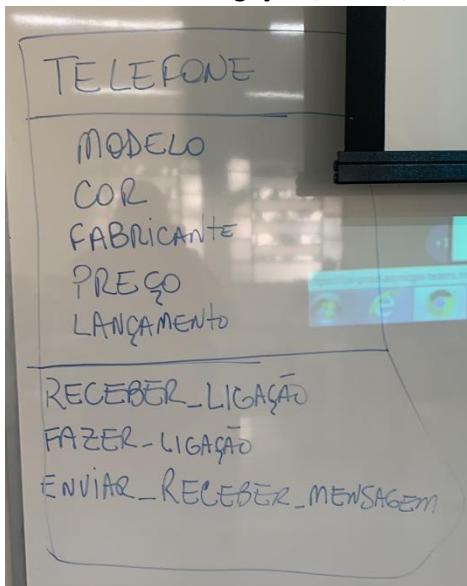
c)



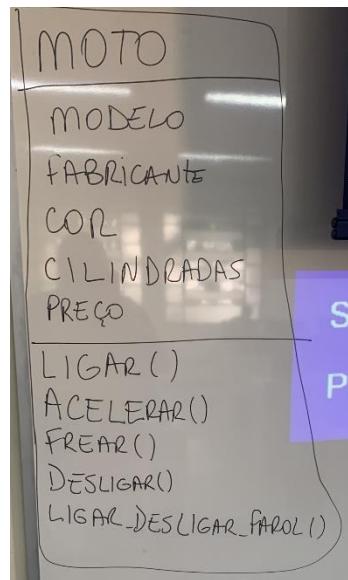
- A) Classe: Tele móvel; Atributos: Cor preta, cor branca, cor prateada, cor azul, botões, visor; Métodos: fazer ligações, enviar mensagens, tocar música.
- B) Classe: Moto; Atributos: Cor vermelha, desenho 1D, 2D, 3D, rodas, guidão, freio, embreagem...; Métodos: dirigir, buzinar, dar seta...
- C) Classe: pessoas; Atributos: branca, negra, amarela, rosada, mulher, homem; Métodos: comer, dormir, estudar, trabalhar, emoções.

0311

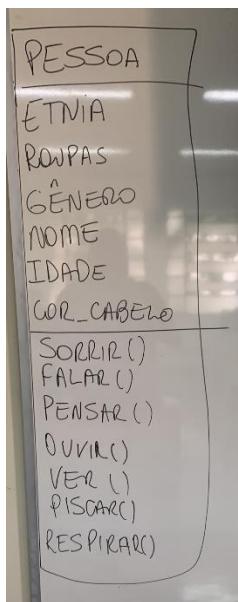
- A) Classe: Telefone; Atributos: Modelo, cor, fabricante, preço, lançamento; Métodos: receber e efetuar ligações, enviar/receber mensagens.



- B) Classe: Motos; Atributos: Modelo, fabricante, cor, cilindradas, preço; Métodos: Ligar, acelerar, frear, desligar, ligar e desligar farol.



c)



1011

UML – serve p gente documentar sistemas. É um conjunto de digramas.

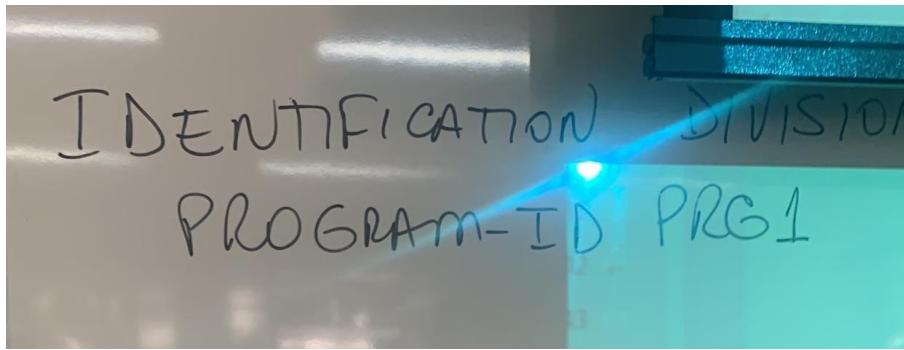
DEFINIÇÃO

É um conjunto de ferramentas (diagramas) que permitirão representar o modelo de um sistema qualquer, porém, por meio do paradigma de Orientação a objetos.

UML – serve para documentar sistemas. É um conjunto de diagramas.

Tinha a análise estrutura – representava o sistema como todos, mas era um outro tipo de linguagem estrutura, como o html se fosse uma linguagem = lê linha por linha e monta aquela página.

Início de programa em Cobol (hoje não se cria mais nada, só dá manutenção para coisas que utilizam essa linguagem – tem que escrever muito e tem uma série de regras):



A UML veio para corrigir isso, já que vai usar classes e objetos.

OBJETIVOS

- A modelagem de sistemas (não apenas de software) usando os conceitos da orientação a objetos.
- Estabelecer uma união fazendo com que métodos conceituais sejam também executáveis.
- Criar uma linguagem de modelagem usável tanto pelo homem quanto pela máquina.

Tinha a análise estrutura – representava o sistema como todos, mas era um outro tipo de linguagem estrutura, como o [html](#) se fosse uma linguagem = lê linha por linha e monta aquela página.

Início do programa em Cobol:

**IDENTIFICATION DIVISION
PROGRAM-ID PRES1**

A UML veio para corrigir isso, já que vai usar classes e objetos.

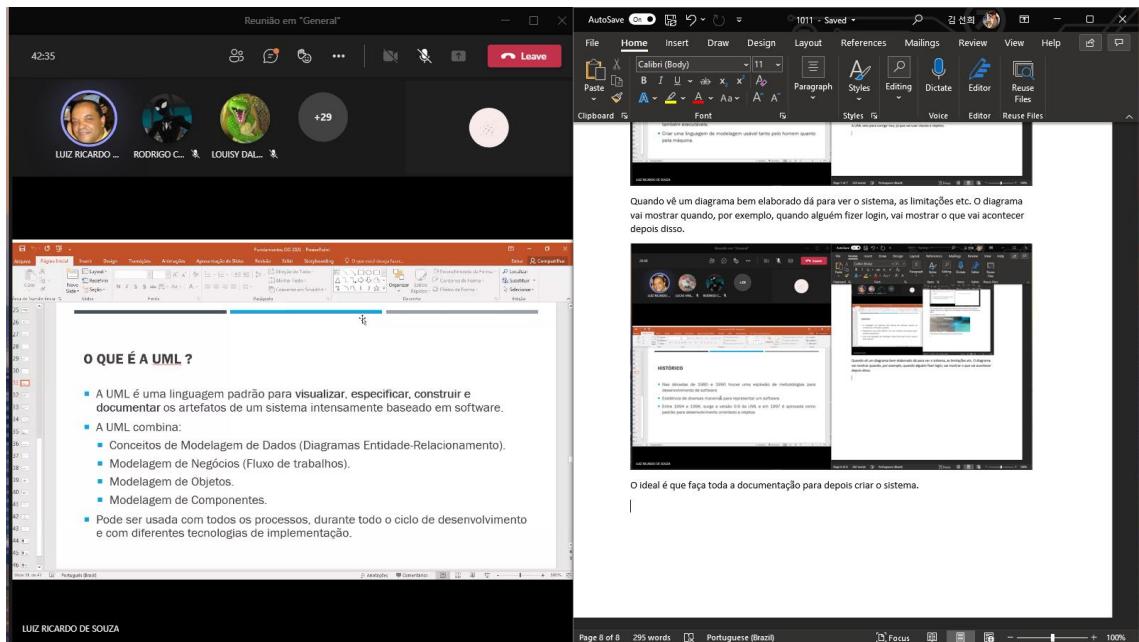
Quando vê um diagrama bem elaborado dá para ver o sistema, as limitações etc. O diagrama vai mostrar quando, por exemplo, quando alguém fizer login, vai mostrar o que vai acontecer depois disso.

HISTÓRICO

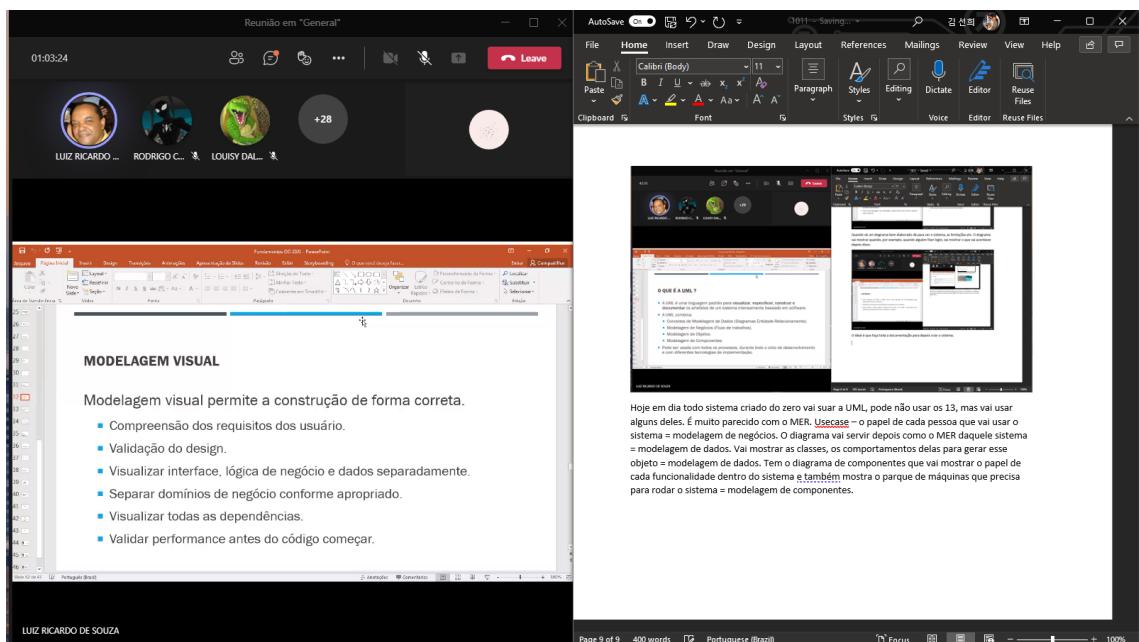
- Nas décadas de 1980 e 1990 houve uma explosão de metodologias para desenvolvimento de software.
- Existência de diversas maneiras para representar um software.
- Entre 1994 e 1996, surge a versão 0,9 da UML e em 1997 é aprovada como padrão para desenvolvimento orientado a objetos

Quando vê um diagrama bem elaborado dá para ver o sistema, as limitações etc. O diagrama vai mostrar quando, por exemplo, quando alguém fizer login, vai mostrar o que vai acontecer depois disso.

O ideal é que faça toda a documentação para depois criar o sistema.



Hoje em dia todo sistema criado do zero vai suar a UML, pode não usar os 13, mas vai usar alguns deles. É muito parecido com o MER. Usecase – o papel de cada pessoa que vai usar o sistema = modelagem de negócios. O diagrama vai servir depois como o MER daquele sistema = modelagem de dados. Vai mostrar as classes, os comportamentos delas para gerar esse objeto = modelagem de dados. Tem o diagrama de componentes que vai mostrar o papel de cada funcionalidade dentro do sistema e também mostra o parque de máquinas que precisa para rodar o sistema = modelagem de componentes.



Vai traduzir todos os requisitos em um diagrama. Vai conseguir olhar separadamente – código é o programa em si, a finalidade das conexões, mas o cliente só quer saber como vai funcionar = visualizar interface. Dependência = o NSA tem uma dependência do site do vestibulinho para saber as informações como nome, endereço...

Reunião em "General"

01:18

LUIZ RICARDO ... RODRIGO C... LOUISY DAL...

+28

AutoSave 1011 - Saving... File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

alguns deles. É muito parecido com o MIER. **Usecase** = o papel de cada pessoa que vai usar o sistema = modelagem de negócios. O diagrama vai servir depois como o MER daquele sistema = modelagem de dados. Vai mostrar as classes, os comportamentos delas para gerar esse objeto = modelagem de dados. Tem o diagrama de componentes que vai mostrar o papel de cada funcionalidade dentro do sistema e também mostra o parque de máquinas que precisa rodar o sistema = modelagem de componentes.

USOS DA UML

- Mostrar a interação da aplicação com o mundo interno e externo.
- Mostrar as interações dos usuários com o sistema.
- O comportamento da aplicação.
- A estrutura do sistema.
- Os componentes do sistema.
- A arquitetura da empresa.

Vai traduzir todos os requisitos em um diagrama. Vai conseguir olhar separadamente – código é o programa em si, a finalidade das conexões, mas o cliente só quer saber como vai funcionar = visualizar interface. **Dependência** = NSA tem uma dependência do site do vestibulinho para saber as informações como nome, endereço...

Page 9 of 9 452 words Portuguese (Brazil) Focus

O comportamento = como ela deve funcionar corretamente (tem diagrama para componentes, empresa...)

Reunião em "General"

01:12:02

LUIZ RICARDO ... RODRIGO C... LOUISY DAL...

+28

AutoSave 1011 - Saving... File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

o problema é que a aplicação desse requisito, mas o cliente só quer saber como vai funcionar = visualizar interface. **Dependência** = NSA tem uma dependência do site do vestibulinho para saber as informações como nome, endereço...

DIAGRAMAS UML 2.5

```

graph TD
    Diagrama[Diagrama] --> DiagramaEstruturas[Diagrama de Estruturas]
    Diagrama --> DiagramaComportamentos[Diagrama de Comportamentos]
    DiagramaEstruturas --> DiagramaClasses[Diagrama de Classes]
    DiagramaEstruturas --> DiagramaComponentes[Diagrama de Componentes]
    DiagramaEstruturas --> DiagramaObjetos[Diagrama de Objetos]
    DiagramaComportamentos --> DiagramaAtividades[Diagrama de Atividades]
    DiagramaComportamentos --> DiagramaCasosUso[Diagrama de Casos de Uso]
    DiagramaAtividades --> DiagramaInteracao[Diagrama de Interação]
    DiagramaAtividades --> DiagramaMSE[Diagrama de Máquina de Estados]
    DiagramaInteracao --> DiagramaPerfis[Diagrama de Perfil]
    DiagramaInteracao --> DiagramaEstruturasCompuestas[Diagrama de Estruturas Compostas]
    DiagramaInteracao --> DiagramaImplantacao[Diagrama de Implantação]
    DiagramaInteracao --> DiagramaPaquetes[Diagrama de Paquetes]
    DiagramaCasosUso --> DiagramaSequencia[Diagrama de Sequência]
    DiagramaCasosUso --> DiagramaComunicacao[Diagrama de Comunicação]
    DiagramaCasosUso --> DiagramaVGI[Diagrama de Visão Geral de Intereração]
    DiagramaCasosUso --> DiagramaTempo[Diagrama de Tempo]
  
```

Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

LUIZ RICARDO DE SOUZA

Page 10 of 10 465 words Portuguese (Brazil) Focus

Diagrama de Classes – Estrutura e Diagrama de Casos de Uso – Comportamentos. Quanto mais para baixo, mais específico o diagrama é.

1711

The screenshot shows a Microsoft Teams meeting interface at the top and a PowerPoint slide below. The Teams interface includes participant icons for LUIZ RICARDO, JOAO VICT..., and RODRIGO, with a '+32' button. The PowerPoint slide has a dark background and contains the following text:

DIAGRAMA USE CASE (CASOS DE USO)

- É o diagrama mais abstrato, flexível e informal da UML.
- Normalmente, é utilizado no início da modelagem para identificar os requisitos do sistema.
- São utilizados para captar o comportamento pretendido do novo sistema (macro atividades).
- Pode ser utilizado como base para criação de outros diagramas.
- Estão associados aos requisitos funcionais do sistema.
- Representam o comportamento do sistema do ponto de vista do usuário.
- Usando uma linguagem simples, permite que qualquer pessoa compreenda o comportamento externo do sistema.

LUIZ RICARDO DE SOUZA

Quanto mais detalhado melhor. Esse diagrama só mostra os requisitos funcionais, que são ligados diretamente a programação, que você programa. E os não funcionais são aqueles que não programa, que estão além da programação – como o login tem que ser preenchido em 10s. Já o funcional: o sistema deverá cadastrar os alunos e as disciplinas. Vai ter a visão que o usuário vai ter, é muito simples e vai ter o que cada “ator” faz (o que o aluno faz...).

The screenshot shows a Microsoft Teams meeting interface at the top and a PowerPoint slide below. The Teams interface includes participant icons for LUIZ RICARDO, JOAO VICT..., and RODRIGO, with a '+32' button. The PowerPoint slide has a dark background and contains the following text:

COMPONENTES PRINCIPAIS

O modelo de casos de uso tem como principais componentes:

- atores: entidade externa que interage com os casos de uso.
- casos de uso: especifica uma funcionalidade completa do sistema.
- relacionamentos: mostra a comunicação entre o ator e o caso de uso.

LUIZ RICARDO DE SOUZA

São as pessoas ou as coisas que vão interagir com o sistema = atores. Casos de usos = funcionalidades do sistema (efetuar login, deletar registro, gerar senha...). Relacionamentos = diferente de BD, é só para mostrar a relação entre os atores, e até mesmo entre os caso de uso.

ATOR

- Representam os papéis desempenhados pelos diversos usuários que poderão utilizar ou interagir com os serviços e funções do sistema.
- Pode ser qualquer elemento externo que interaja com o sistema, inclusive um software ou hardware.
- Exemplos típicos: cliente, aluno, supervisor, professor, impressora fiscal, dispositivo de conexão de rede, sistema de faturamento etc.

Só as pessoas ou as coisas que vão interagir com o sistema = atores. Casos de usos = funcionalidades do sistema (efetuar login, deletar registro, gerar senha...). Relacionamentos = diferente de BD, é só para mostrar a relação entre os atores, e até mesmo entre os casos de uso.

O hardware pode interagir, por exemplo, um balcão para solicitar a senha automática, que aperta algum botão e gera ela.

ATOR

- Representam os papéis desempenhados pelos diversos usuários que poderão utilizar ou interagir com os serviços e funções do sistema.
- Pode ser qualquer elemento externo que interaja com o sistema, inclusive um software ou hardware.
- Exemplos típicos: cliente, aluno, supervisor, professor, impressora fiscal, dispositivo de conexão de rede, sistema de faturamento etc.

Só as pessoas ou as coisas que vão interagir com o sistema = atores. Casos de usos = funcionalidades do sistema (efetuar login, deletar registro, gerar senha...). Relacionamentos = diferente de BD, é só para mostrar a relação entre os atores, e até mesmo entre os casos de uso.

Embaixo do boneco coloca o nome = cliente, supervisor...

Reunião em "General"

20:52

LUIZ RICARDO, ITALO PEDR., RODRIGO +34

Fundamentos OO 2022.pptx - PowerPoint

IDENTIFICANDO ATORES

- Algunas perguntas podem auxiliar a identificar atores:
- Quem utiliza a principal funcionalidade do sistema?
- Quem vai precisar de suporte do sistema para realizar suas tarefas diárias?
- Quem precisa manter, administrar e deixar o sistema "rolando"?
- Quais dispositivos de hardware o sistema precisa manipular?
- Com quais outros sistemas o sistema precisa interagir?
- Quem ou o que tem interesse nos resultados produzidos pelo sistema?

Page 7 of 7 289 words Portuguese (Brazil)

LUIZ RICARDO DE SOUZA

Quais são as pessoas que utilizam esse sistema? – essas pessoas possivelmente serão atores dentro desse sistema, não importa se tem uma ou um milhão de funções, mas se não tiver nenhuma função não coloca.

Quem administra o sistema?

Os dispositivos como impressora, que vai rodar um relatório no sistema e vai imprimir na impressora.

Sistema da escola interagindo com o sistema do banco.

Reunião em "General"

23:36

LUIZ RICARDO, ITALO PEDR., RODRIGO +34

Fundamentos OO 2022.pptx - PowerPoint

CASOS DE USO

- Casos de uso são utilizados para modelar os requisitos funcionais do sistema.
- Descrevem as funcionalidades do sistema, conforme esperadas pelos usuários, retratando um "diálogo" que uma entidade externa, realiza com o sistema.
- Um caso de uso é baseado em um cenário descritivo, mostrando como o ator interage com o sistema.
- Deve ser completo, isto é, não possível "quebrar" um caso de uso grande em diversos casos de usos menores.
- Realizado em nome de um ator que, por sua vez, deve pedir direta ou indiretamente ao sistema tal realização.
- Pode se relacionar com um ou mais atores através de associações.

Page 8 of 8 353 words Portuguese (Brazil)

LUIZ RICARDO DE SOUZA

Representação gráfica de um requisito de sistema – funcionais. Todos eles são representados por um case.

Mostra do ponto de vista do usuário, o que o usuário professor pode fazer, o que o aluno pode fazer?...

Com base na descrição gira um caso de uso. "Quebrar" = não pode dividir um caso de uso.

Realizar em nome do ator = representar as funcionalidades que o ator tem para realizar seu trabalho (usuário professor = fazer chamada, registrar menções...).

O mesmo case pode ter um ou mais autores usando o mesmo case.

Reunião em "General"

Fundamentos OO 2021.pptx - PowerPoint

CASOS DE USO

- Casos de uso são utilizados para modelar os requisitos funcionais do sistema.
- Descrevem as funcionalidades do sistema, conforme esperadas pelos usuários, retratando o sistema.
- Um caso de uso é interage com o sistema.
- Deve ser completo.
- Realizado em nome de um ator que, por sua vez, deve pedir direta ou indiretamente ao sistema tal realização.
- Pode se relacionar com um ou mais atores através de associações.

Com base na descrição gira um caso de uso. "Quebrar" = não pode dividir um caso de uso. Realizar em nome do ator = representar as funcionalidades que o ator tem para realizar seu trabalho (usuário professor = fazer chamada, registrar menções...). O mesmo case pode ter um ou mais autores usando o mesmo case.

Reunião em "General"

Fundamentos OO 2021.pptx - PowerPoint

IDENTIFICANDO CASOS DE USO

As respostas às seguintes perguntas podem auxiliar a encontrar casos de uso:

- Quais funções o ator requer do sistema? O que o ator precisa fazer?
- O ator precisa criar, ler, destruir, modificar ou armazenar algum tipo de informação dentro do sistema?
- O ator precisa ser notificado de eventos do sistema? O ator precisa notificar o sistema sobre algum evento?
- O trabalho diário do ator poderia ser simplificado ou tornado mais eficiente através de novas funcionalidades do sistema?
- Quais entradas e saídas o sistema precisa?
- Quais os principais problemas com o atual sistema?

Com base na descrição gira um caso de uso. "Quebrar" = não pode dividir um caso de uso. Realizar em nome do ator = representar as funcionalidades que o ator tem para realizar seu trabalho (usuário professor = fazer chamada, registrar menções...). O mesmo case pode ter um ou mais autores usando o mesmo case.

Aqui vai representar todas as funcionalidades do sistema.

O ator professor pode cadastrar menções, efetuar registros, fazer chamada – 1

Se precisar armazenar, tem que ter os respectivos casos de usos - 2

Gerar senha, se tiver um balcão de informações, esse ator, quando pede a senha vai gerar a resposta que é a senha, e depois o ator externo vai chamar a senha – 3

Trabalho diário, se sim, vai gerar o case – 4

Pode ser um relatório que pode ser impresso ou não – sair – 5

Se tiver que fazer login, vai virar caso de uso, se não tiver – 6

Gerar senha, se tiver um balcão de informações, esse ator, quando pede a senha vai gerar a resposta que é a senha, e depois o ator externo vai chamar a senha – 3

Trabalho diário, se sim, vai gerar o case – 4

Pode ser um relatório que pode ser impresso ou não – sair – 5

Se tiver que fazer login, vai virar caso de uso, se não tiver – 6

CDU01 = identificação do requisito. Entre o técnico e o gerente tem uma relação de herança, porque todas as funcionalidades que o técnico fizer, o gerente pode fazer também.

Se tiver que fazer login, vai virar caso de uso, se não tiver – 6

CDU01 – identificação do requisito. Entre o técnico e o gerente tem uma relação de herança, porque todas as funcionalidades que o técnico fizer, o gerente pode fazer também.

O aluno tem as coisas específicas dele, mas pode fazer as coisas da pessoa também = herança.

Reunião em "General"

35:13

LUIZ RICARDO, ITALO PEDR., RODRIGO +35

Fundamentos OO 2022.pptx - PowerPoint

CDU01 - Identificação do requisito: Entre o técnico e o gerente tem uma relação de herança, porque todas as funcionalidades que o técnico fizer, o gerente pode fazer também.

O aluno tem as coisas específicas dele, mas pode fazer as coisas da pessoa também = herança.

Reunião em "General"

36:19

LUIZ RICARDO, LOUSY DAI..., ITALO PEDR. +35

RELACIONAMENTOS - ASSOCIAÇÃO

São representadas por uma linha que liga o ator ao caso de uso.

Podem existir setas para indicar a navegabilidade.

Professor —> Lançar Notas

CDU01 - Identificação do requisito: Entre o técnico e o gerente tem uma relação de herança, porque todas as funcionalidades que o técnico fizer, o gerente pode fazer também.

Page 10 of 10 588 words Portuguese (Brazil) Focus 120%

GENERALIZAÇÃO/ESPECIALIZAÇÃO

Forma de associação na qual existem dois ou mais casos de uso com características semelhantes.

Existem algumas diferenças entre os casos de uso associados:

- Também é possível com atores.

```

graph TD
    Pessoa --> Aluno
    Pessoa --> Professor
    Pessoa --> Assistente
    Professor --> General
    Professor --> Linguagem
    Professor --> Matematica
    Professor --> Materiais
    Assistente --> Materiais
    
```

Essa seta “não pintada” significa que é um relacionamento de herança:

Pessoa

```

graph TD
    Pessoa --> Aluno
    Pessoa --> Professor
    Pessoa --> Assistente
    
```

Herança/generalização = o que for comum a todos os outros atores, coloca no ator pessoa, ai o que for específico de cada um coloca os cases de cada um.

Especialização = coisas/cases específicos.

Fundamentos OO 2022.pptx - PowerPoint

Arquivo Página Inicial Insert Design Transições Animações Apresentação de Slides Revisão Exibe Storyboarding Diga-me Entrar Compartilhar

Área de Transferência

Slide 46 de 47 Português (Brasil)

INCLUDE

- A execução de um caso de uso obriga a execução de um outro.
- Pode ser comparado à chamada de uma sub-rotina.

EXTEND

- Descreve cenários opcionais de um caso de uso.
- Só ocorrerá se uma determinada condição for satisfeita.

Diagrama UML:

```

sequenceDiagram
    actor Almacarife
    participant UC001 as UC001 - Selecionar material
    participant UC002 as UC002 - Checar estoque
    participant UC003 as UC003 - Comprar material
    participant UC004 as UC004 - Emitir pedido de compra

    Almacarife->>UC001: 
    UC001->>UC002: <<include>>
    UC001->>UC003: <<extend>>
    UC002-->>Pessoa: Herança
    UC003-->>Pessoa: Generalização
    UC004-->>Pessoa: Especialização
  
```

Word Document Content:

- Herança/generalização = o que for comum a todos os outros atores, coloca no ator pessoa, aí o que for específico de cada um coloca os casos de cada um.
- Especialização = coisas/cases específicos.

Include pode ser chamado de sub-rotina: quando solicita um material, tem que checar para ver se existe esse material no estoque. Tem que obrigatoriamente executar esses dois casos.

Fundamentos OO 2022.pptx - PowerPoint

Arquivo Página Inicial Insert Draw Design Layout References Mailings Review View Help

Drawing Tools

Slide 47 de 47 Português (Brasil)

INCLUDE

- A execução de um caso de uso obriga a execução de um outro.
- Pode ser comparado à chamada de uma sub-rotina.

EXTEND

- Descreve cenários opcionais de um caso de uso.
- Só ocorrerá se uma determinada condição for satisfeita.

Diagrama UML:

```

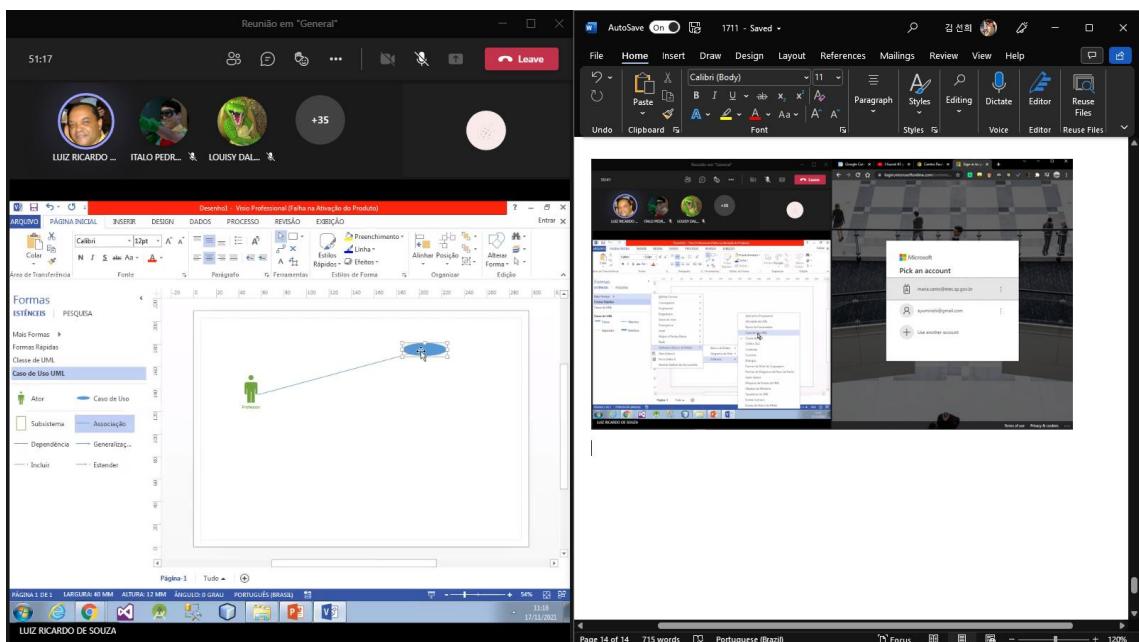
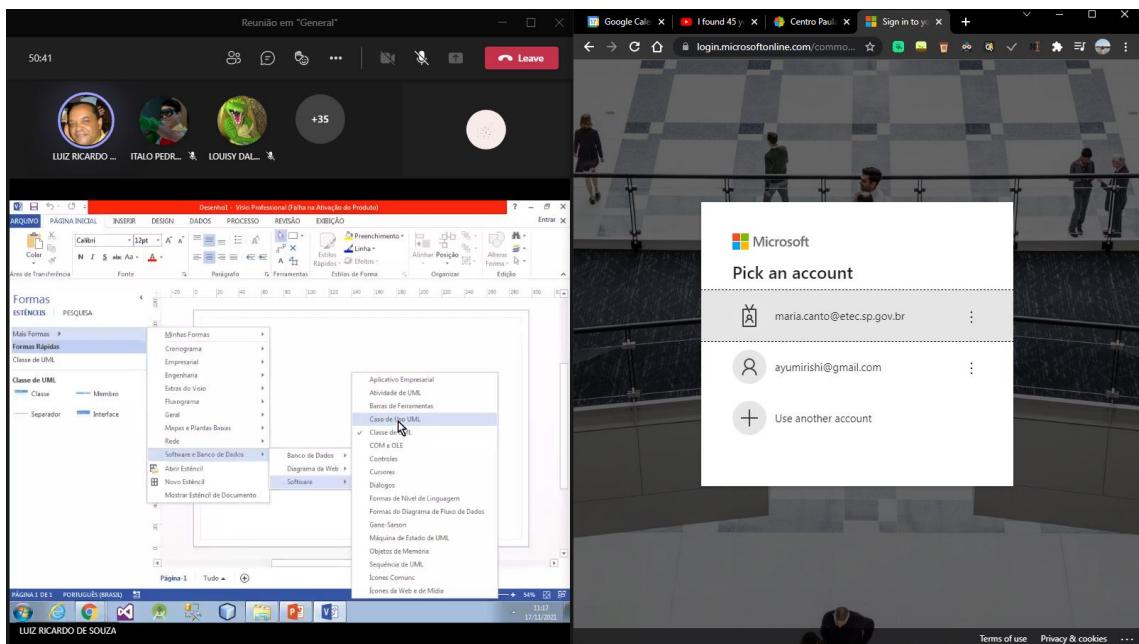
sequenceDiagram
    actor Almacarife
    participant UC001 as UC001 - Selecionar material
    participant UC002 as UC002 - Checar estoque
    participant UC003 as UC003 - Comprar material
    participant UC004 as UC004 - Emitir pedido de compra

    Almacarife->>UC001: 
    UC001->>UC002: <<include>>
    UC001->>UC003: <<extend>>
    UC002-->>Pessoa: Herança
    UC003-->>Pessoa: Generalização
    UC004-->>Pessoa: Especialização
  
```

Word Document Content:

- Include pode ser chamado de sub-rotina: quando solicita um material, tem que checar para ver se existe esse material no estoque. Tem que obrigatoriamente executar esses dois casos.
- Especialização = coisas/cases específicos.

O extend é opcional, se ocorre quando algo acontecer. Checou o estoque, não tem o material, ele vai comprar o material, que pode ou não ser comprado, aí se for efetuado a compra, tem que emitir o pedido de compra. – como a essência de baunilha, pode ou não comprar, já que não é algo essencial.



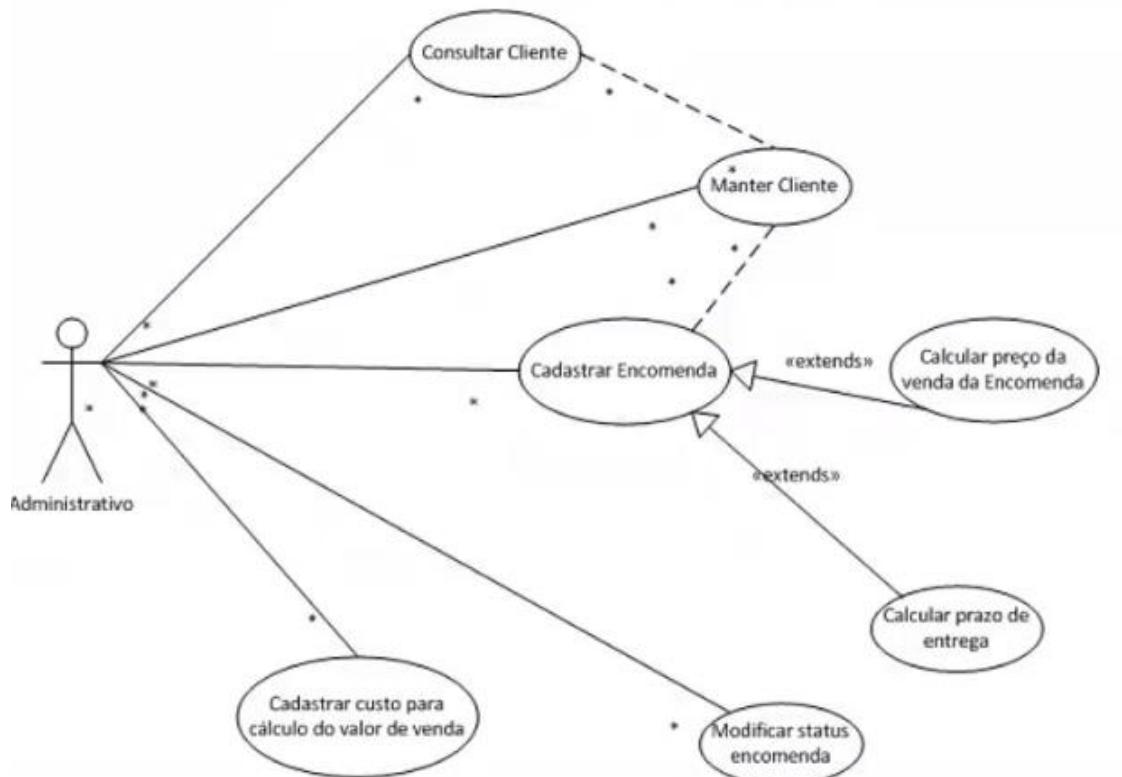
2411

O cliente vai apenas fazer o pedido, ele não vai manipular o sistema, mas se fosse um site em que o cliente fosse pedir uma encomenda, seria outra coisa.

Não vai usar o nome João, e sim administrador/operador de sistema nesse caso. Ele é o único ator nesse caso.

Cases – encomenda, recibo,

Exercício 1 – Encomenda de Placas



Consultar – para ver se o cliente existe dentro da base de dados.

Manter – dentro do case tem de incluir, alterar, deletar. O consultar é sempre fora do case manter.

Cadastrar – todas as informações como tamanho da placa, cliente que está solicitando, o recibo está aqui porque é um subproduto da encomenda (se fosse um relatório, como um boletim, teria que ter um case específico) ...

Calcular prazo – de acordo com o que vai ser solicitado no cadastro encomenda.

Calcular preço – aquela tabela.

Modificar o status – para quando estiver pronto.

Cadastrar custo – os parâmetros que vai calcular aqui, valor da placa... custos para criar essa placa.

Os atributos são usados apenas no diagrama de classes.

As linhas tracejadas vão depender de outros cases, a relação de dependência. Todo diagrama é de cima para baixo que lê.

Exercício 3:

80 • Exercitando Modelagem em UML

São controles oferecidos pela aplicação:
Quando há uma nova solicitação, é possível obter de cada item a lista dos três menores preços que já foram pagos para o referido produto, incluindo na listagem o local onde foi comprado.
A lista de compras é impressa para todos os itens que não foram fechados, de todas as solicitações de compra que estejam com status em aberto.
Uma solicitação pode ser cancelada (status = "cancelado").
Quando todos os itens de uma solicitação tiverem sido comprados, o sistema atualiza automaticamente o status dessa solicitação para "fechado".
Deve ser emitida uma listagem de todos os produtos já comprados, com seu somatório de quantidade e de valor.

Exercício:
A partir do cenário descrito, desenhe o diagrama de casos de uso desse sistema. Escreva, também, os cenários. Considere que todas as operações são feitas pelo Alvaro, que pode ser identificado como Responsável pela Obra.

RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):

84-85 / 129

Exercício 4:

88 • Exercitando Modelagem em UML

Para cada caixa, deve-se ter um histórico de atendimentos, para se obter estatística. A estatística deve ser detalhada quanto ao tempo mínimo, médio e máximo de atendimento por caixa e por dia, além do número de atendimentos por assunto.
A qualquer momento é preciso saber que caixa está com um determinado número de atendimento.

Exercício:
A partir do cenário descrito, desenhe o diagrama de casos de uso desse sistema. Escreva, também, os cenários. Considere que as tarefas de controlar os assuntos e os caixas, obter estatísticas e relatórios são do Setor Administrativo. O Caxa se responsabiliza por controlar a próxima senha e o Balcão de Informações é que gera novas senhas.

RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):

82-83 / 129

MANTER CAIXAS

Este caso de uso tem por objetivo permitir a manutenção (inclusão, alteração e exclusão) dos caixas controlados no atendimento.

Ator: Administrativo

Cenário Principal:

- O sistema prepara uma lista de caixas cadastrados.
- O usuário informa, no caso de "Alteração" ou "Inclusão".
- O sistema exibe os dados cadastrados e os habilita para edição.
- No caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
- No caso de exclusão, o sistema solicita a confirmação.
- O sistema prepara uma lista de caixas cadastrados.
- O usuário informa, no caso de "Alteração" ou "Inclusão".
- O sistema habilita os dados cadastrais do assunto.
- No caso de inclusão, o sistema gera automaticamente um código de identificação.

Cenário Alternativo:
- Exclusão não permitida
Não é possível excluir um assunto que esteja associado a um atendimento.

92-93 / 129

Exercício 5:

Exercitando a identificação de Casos de Uso • 97

- Limite de números apostados
O intervalo permitido para a quantidade de números apostados é de valor de quantidade mínima de apostas do jogo lotérico como limite inferior, e valor de quantidade máxima de apostas do jogo lotérico como limite superior.

CONTROLE PARTICIPANTES E PAGAMENTO DE COTA

Descrição: Este caso de uso tem por objetivo permitir a manutenção (inclusão, alteração e exclusão) dos participantes efetivos de um bolão e o pagamento de suas cotas.

Ator: Gestor do Bolão

Pré-condição: Receber a identificação do bolão.

Centro Principal:

- O sistema busca e exibe a lista de participantes já cadastrados para o bolão, exibindo para cada:
 1.1. o nome do participante
 1.2. o ramal do participante
 1.3. o número de cota
 1.4. o valor da cota pagas
- O usuário pode alterar a quantidade de cotas ou o status do pagamento de um dos participantes ou cadastrar um novo participante.
 2.1.0 sistema prepara uma lista de todos os participantes credenciados para o bolão que está
- 2.2.0 usuário informa um novo participante, selecionado de uma lista preexistente.
 2.3.0 usuário informa a quantidade de cotas e se as cotas já foram pagas.

41. CENÁRIO: ESTACIONAMENTO

Bruno e seu pai compraram um terreno e inaugurarão um estacionamento. Para ajudar, a irmã de Bruno está desenvolvendo uma aplicação de controle de estacionamento.

Quando o veículo entra no estacionamento, o atendente observa sua placa e a mesma é cadastrada, juntamente com o modelo do veículo e sua cor. A hora de entrada é registrada automaticamente, correspondendo ao momento do cadastramento da placa. Após estacionar o veículo, o cliente pega o ticket onde está impresso: o número da placa, o modelo do veículo, a cor, a data e a hora da entrada.

98 • Exercitando Modelagem em UML

Ao retornar ao estacionamento, o cliente entrega o ticket. O tempo de permanência é calculado. Considerando esse tempo de permanência, é aplicada a tabela de preços, sabendo-se que a tabela de sábado não é a mesma dos dias úteis e, às vezes, dependendo da época do ano, os donos lançam promoções durante os dias úteis. Veja exemplo das tabelas de preço:

Segunda à sexta	Sábado
1ª hora = R\$ 2,00	Preço único = R\$ 3,00
a partir da 2ª hora (inteiro ou fração) = + R\$ 1,00	

Os donos precisam de relatórios de faturamento diário e semanal.

RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):

102-103 / 129

98 • Exercitando Modelagem em UML

Ao retornar ao estacionamento, o cliente entrega o ticket. O tempo de permanência é calculado. Considerando esse tempo de permanência, é aplicada a tabela de preços, sabendo-se que a tabela de sábado não é a mesma dos dias úteis e, às vezes, dependendo da época do ano, os donos lançam promoções durante os dias úteis. Veja exemplo das tabelas de preço:

Segunda à sexta	Sábado
1ª hora = R\$ 2,00	Preço único = R\$ 3,00
a partir da 2ª hora (inteiro ou fração) = + R\$ 1,00	

Os donos precisam de relatórios de faturamento diário e semanal.

RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):

Meeting in "General"

22:43 Request control JOSE FELIPE... +35 LUIZ RICARDO DE SOUZA

Exercício 1 – Encomenda de Placas

Document1 - Word

File Home Insert Draw Design Layout References Mailings Review View Help

Exercício 1 - Encomenda de Placas

Exercício 2 – Jogo de volei

Apóio Técnico

Técnico

Auxiliar Técnico

Jogador

Manter jogador

Manter jogo

Controlar Placar

Inicia Set

Registrar Ponto

Listar Maiores Pontuadores

Dentro do manter = cadastrar, atualizar e deletar e só o consultar seria fora. Foi definido um papel para cada ator, logo eles não precisam se interligar.

Document1 - Word

File Home Insert Draw Design Layout References Mailings Review View Help

Exercício 2 – Jogo de volei

Apóio Técnico

Técnico

Auxiliar Técnico

Jogador

Exercício 3 – Engenharia

Responsável pela Obra

Consultor Produto

Manter Produto

Manter Item da solicitação de compra

Manter solicitação de compra

Manter local de compra

Imprimir lista de compras

Emitir relatório geral de produtos comprados

Registrar compra efetuada

The Microsoft Word slide contains a screenshot of a Windows desktop. On the left is a Microsoft Word ribbon. In the center is a Microsoft Calculator application showing the calculation $100 \times 0.3 = 30$. To the right is a process diagram titled "Exercício 4 – Empresa de atendimento". The diagram shows a flow from a "Caixa" (Counter) user to various tasks: "Obter próximo número", "Registrar término atendimento", "Verificar distribuição atual de senhas", "Manter caixa", "Gerar senha", and "Obter estatísticas do histórico de atendimento". From another "Caixa" user, there is a connection to "Manter assuntos". A "Admin sistema" user is connected to "Manter assuntos" and "Manter caixa".

The Microsoft Word slide contains a screenshot of a Windows desktop. On the left is a Microsoft Word ribbon. In the center is a PDF document titled "EXERCÍCIOS DO LIVRO FÍSICA - PAULO UENO VOLUME ÚNICO". It shows two physics problems (23 and 24) and a list of power values (1 to 24). To the right is a process diagram titled "Exercício 5 – Estacionamento". The diagram shows a flow from an "Atendente" (Attendant) user to tasks: "Registrar entrada do veículo", "Registrar saída do veículo", "Manter tabela de preços", "Emitir ticket de estacionamento", and "Gerar relatório de faturamento diário". From another "Atendente" user, there is a connection to "Manter tabela de preços". A "Gerente" (Manager) user is connected to "Gerar relatório de faturamento diário".

Como a pessoa só vai entrar com o carro, não precisa colocar o “entrar e sair” dentro do manter, porque no máximo vai pegar a placa do carro