



<u>Desenvolvimento de Sistemas Software</u>

<u>Modelação Estrutural + Notas Finais</u> (<u>Diagrama de Instalação</u>)



Da primeira aula...



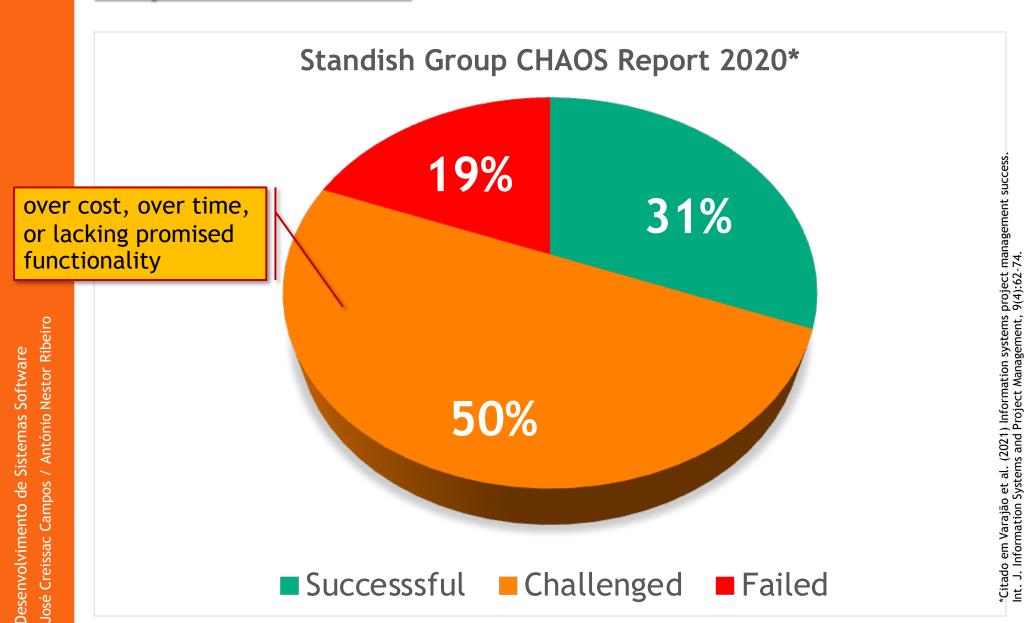
Version 2 announcement:

Among Us 1 (...) started as a tiny local-multiplayer-only game and has grown and grown and grown. (...) Because of this, it's extremely hard to add more things (...) because the game is so fragile. Fixing [it would be] (...) harder than just making a new game.

So the first goal of **Among Us 2 is to be made to withstand growth**. We want to add to it at least as long as Among Us, but **with fewer bugs along the way**.



Da primeira aula...





Standish Group CHAOS Report 2020

- Top 5 Factors in Successful IT Projects
 - User involvement
 - 2. Executive management support
 - 3. Clear Statement of Requirements
 - 4. Proper planning
 - 5. Realistic expectations

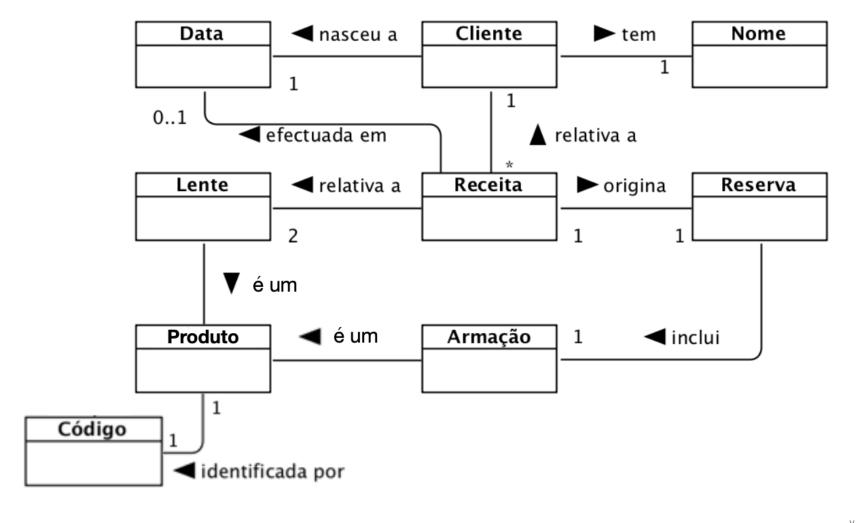
- Top 5 Indicators in Challenged IT Projects
 - 1. Lack of user input
 - Incomplete Requirements & Specifications
 - 3. Changing Requirements & Specifications
 - 4. Lack of executive support
 - 5. Technical incompetence

- Top Factors in Failed IT Projects
 - 1. Incomplete Requirements
 - 2. Lack of user involvement
 - 3. Lack of resources
 - 4. Unrealistic expectations
 - 5. Lack of executive support
 - 6. Changing Requirements & Specifications
 - 7. Lack of planning
 - 8. Didn't need it any longer
 - 9. Lack of IT management
 - 10. Technical illiteracy



Análise - Perceber o problema

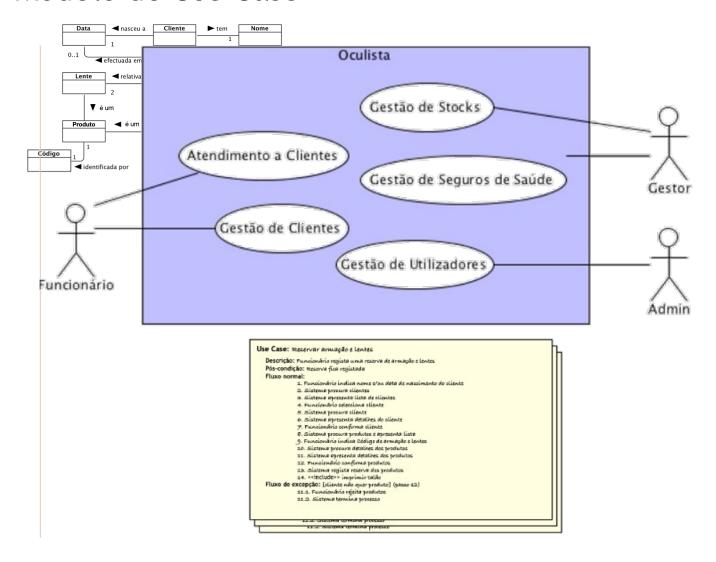
• Modelar o Domínio





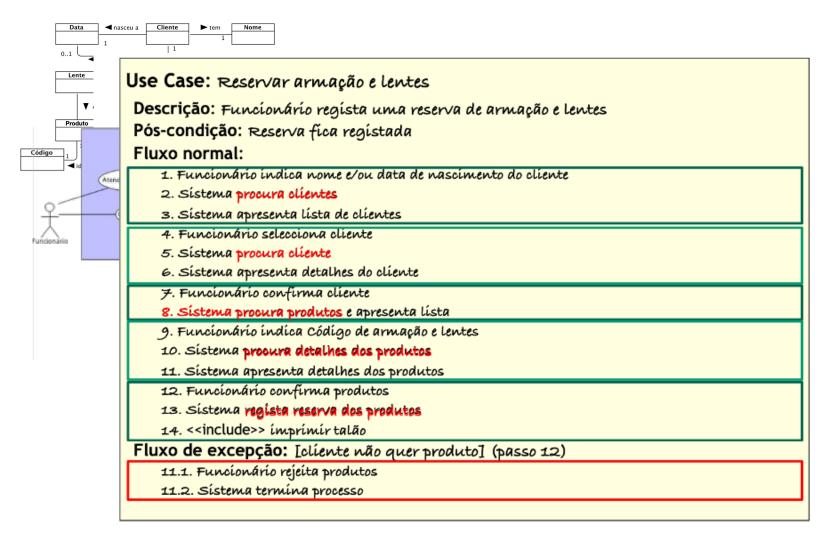
Análise - Perceber os requisitos

Modelo de Use Case



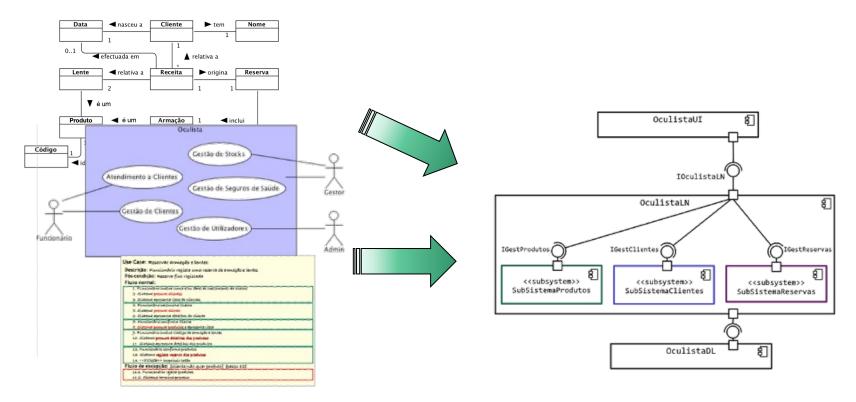


Identificar a API da LN



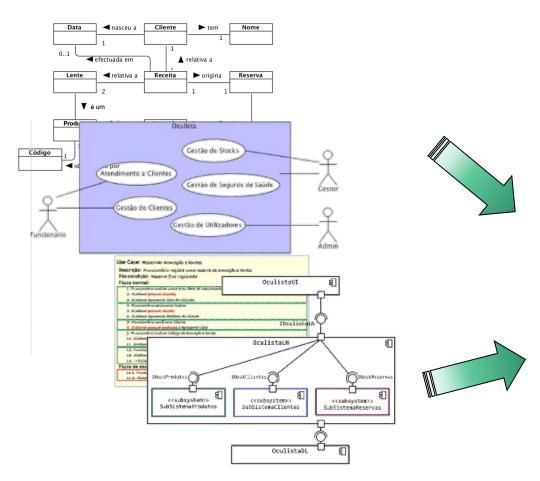


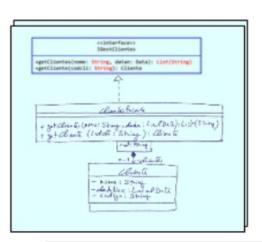
Identificar subsistemas e suas APIs

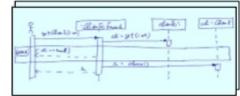




Conceber os subsistemas

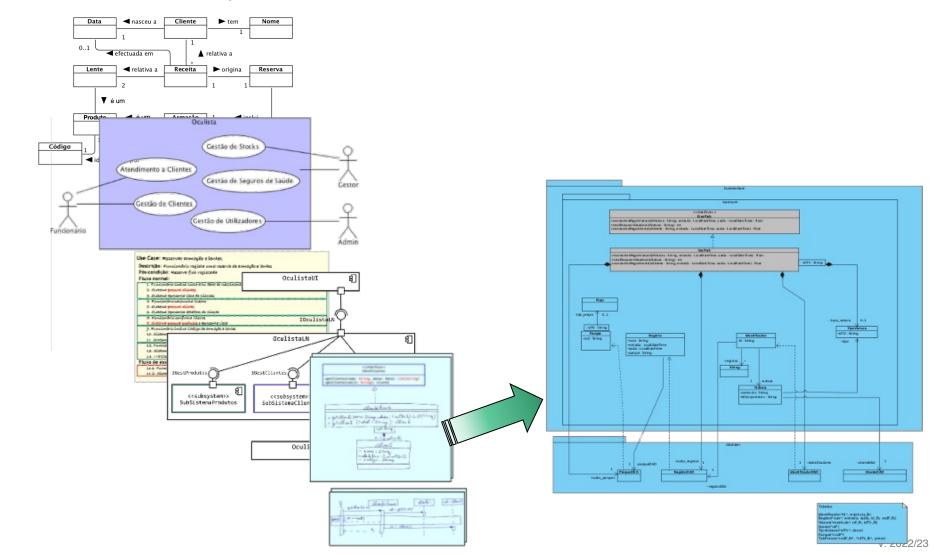






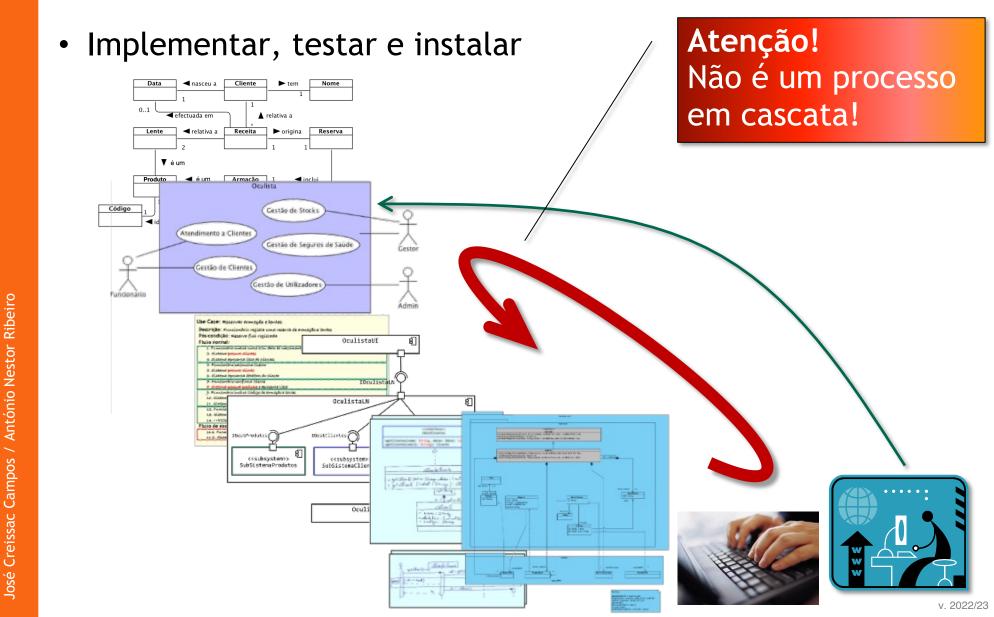


Ajustar à solução tecnológica concreta

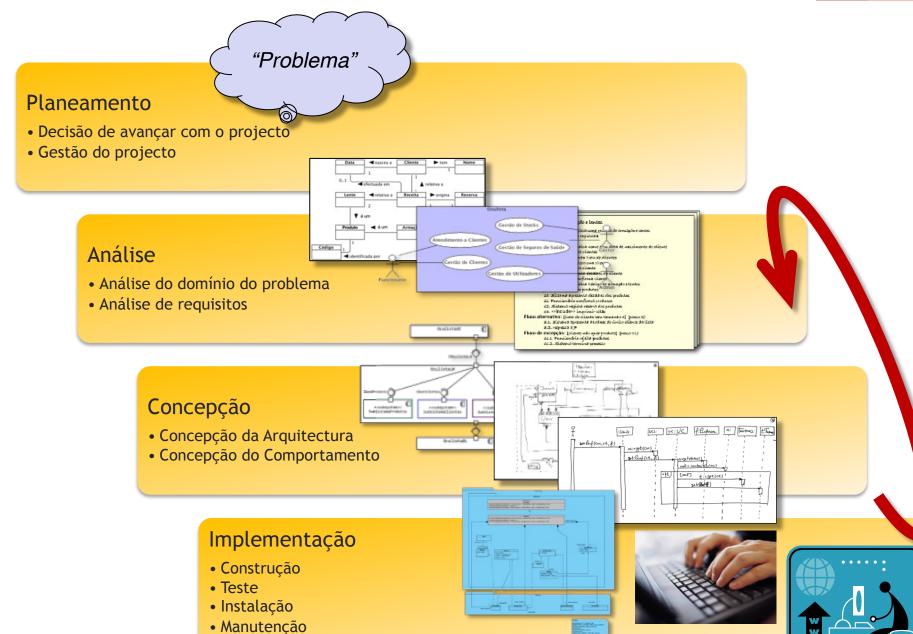




<u>Análise</u> → Implementação









Processo iterativo e incremental - o trabalho

- Fase 1 Perceber os requisitos
 - 5 cenários para perceber o problema
- Fase 2 Primeira iteração de conceber o sistema
 - Focar concepção no Cenário 5
- Fase 3 Primeira iteração de implementar o sistema
 - Focar implementação no Cenário 5 mas fazer iterativamente
 - Ordem sugerida
 - Simular Corridas
 - Configurar Corridas
 - Configurar Campeonato
 - Implementar... mas também testar e instalar!

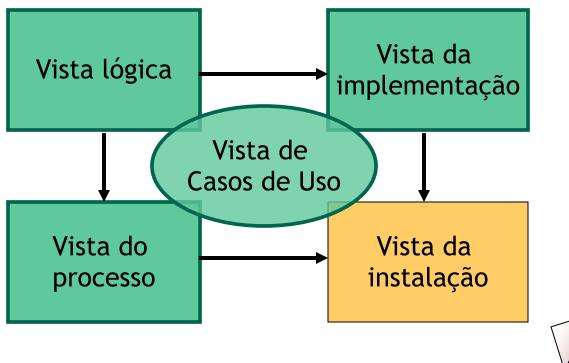


Sobre testar...

- Os Use Case permitem a melhorar a qualidade do processo de testes → sistema mais robusto e confiável
- Fornecem uma descrição clara e concisa dos passos necessários à utilização de uma funcionalidade
 - Ajudam a compreender o que é necessário fazer para testar o sistema
 - Ajudam a garantir que todas as etapas necessárias são incluídas no plano de teste
- Ajudam a identificar potenciais problemas ou áreas do sistema mais propensas a falhas
 - A análise da descrição do Use Case permite identificar potenciais pontos de falha ou áreas onde o sistema pode não se comportar como esperado

※ 〇

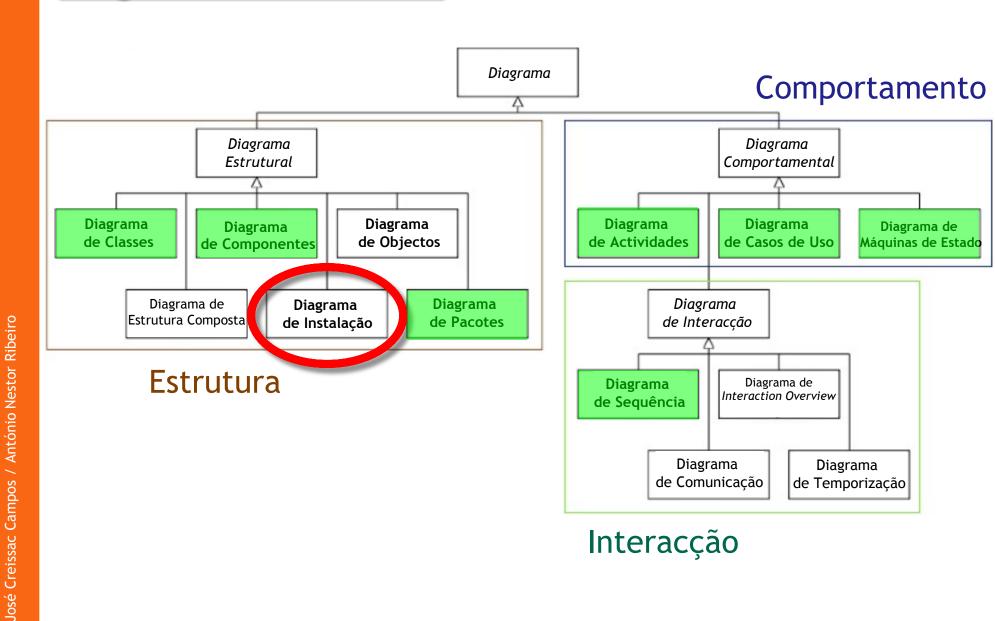
Sobre a instalação do sistema...





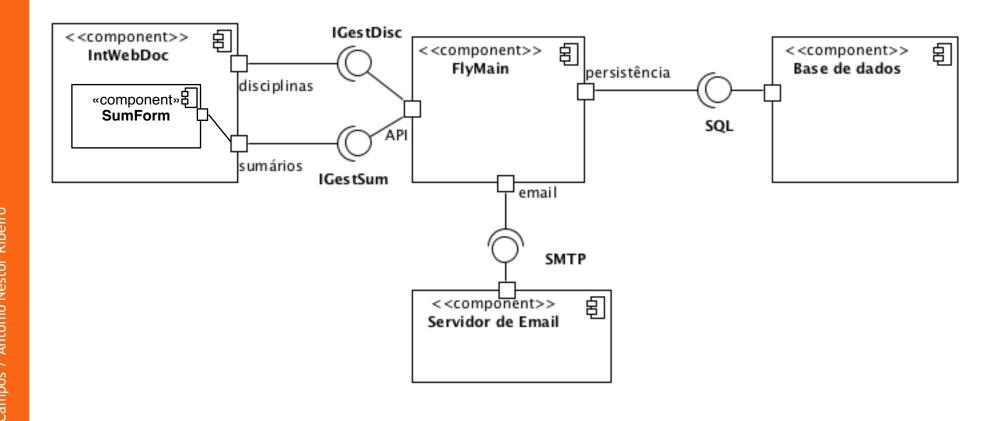
* 〇

Diagramas da UML 2.x





Um sistema divide-se em Componentes





<u>Diagramas de Instalação (Deployment)</u>

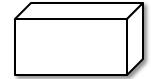
- Qual a disposição física dos componentes que constituem o sistema?
 - Qual é a configuração do sistema em tempo de execução?
- Diagramas de Instalação especificam a arquitectura física do sistema
 - Topologia (ambiente) de hardware sobre a qual são executados os componentes de software
- Permitem:
 - Especificar a distribuição de componentes
 - Identificar estrangulamentos de desempenho



<u>Diagramas de Instalação</u>

- Elementos de um diagrama de deployment
 - Nós
 - Ligações

. Nós (nodes):



- Computadores ou outros dispositivos (hardware)
- Sistema operativo, web servers, application servers, etc. (ambientes de execução)
- Os componentes localizados (*deployed*) em cada nó são representados explicitamente
- É possível agrupar nós em pacotes (packages)

Ligações (connections):

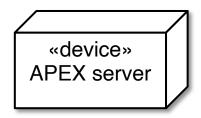
- Representam comunicação entre os nós.
- Podem ser decoradas com multiplicidades.
- · Podem ter estereótipos que indicam o tipo de ligação.
 - Exemplo: «TCP/IP» ou «RMI»



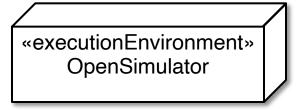


<u>Diagramas de Instalação</u>

Por vezes utiliza-se o estereótipo «device» para identificar os nós de hardware



 Para identificar os ambientes de execução pode utilizar-se o estereótipo «executionEnvironment»



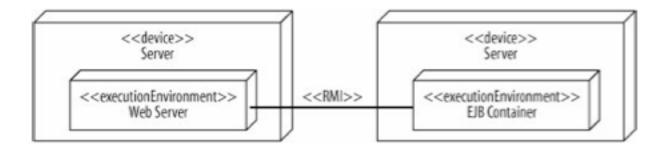
· Comunicação entre dois nós



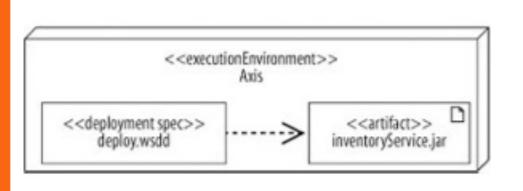


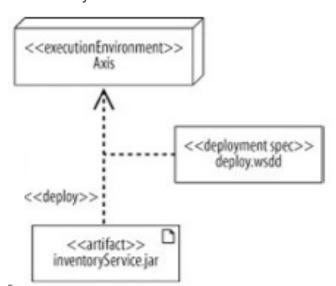
<u>Diagramas de Instalação</u>

 A descrição pode ser refinada para detalhar os ambientes de execução em cada nó



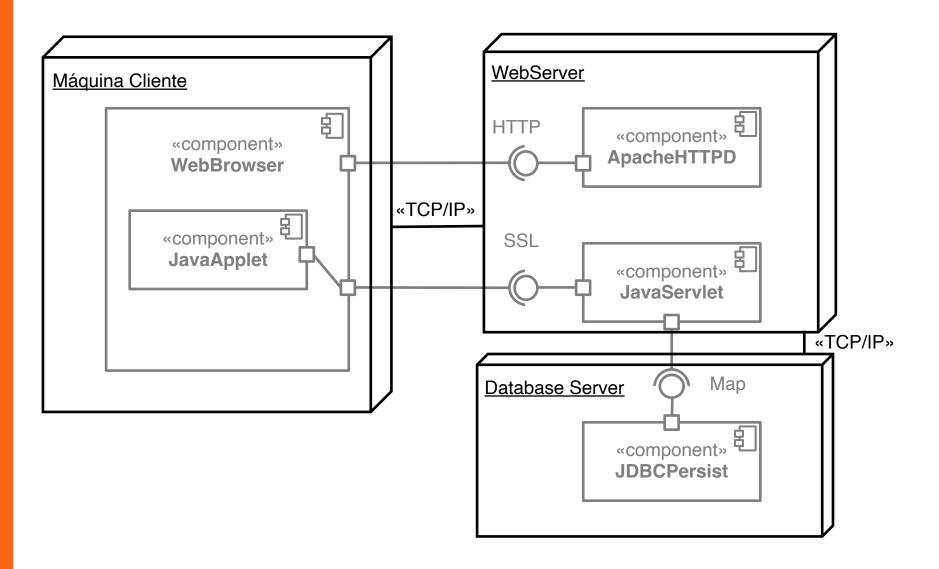
• Especificação de dependências em tempo de execução





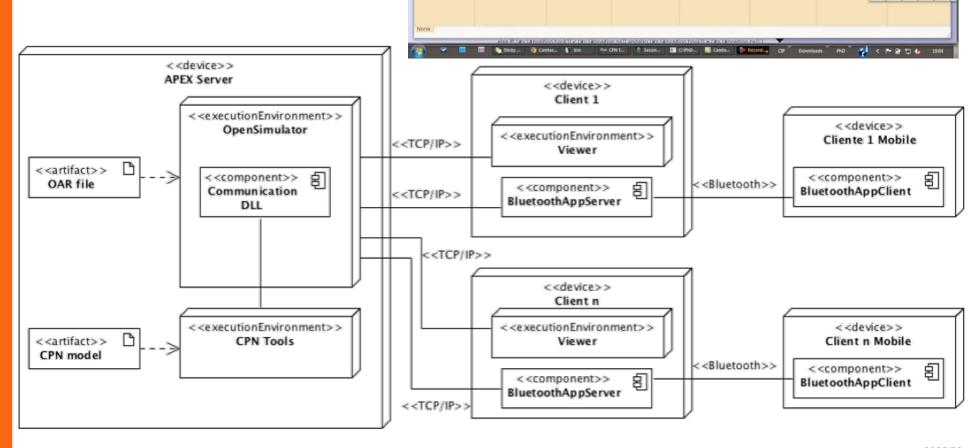


Instalação e componentes





Um exemplo



1' {id="Jose Silva",userList=["book12

initial_GPS

initial_light_SENSOR

initial_OBJ

action acceptCo

Create Sim SS Net

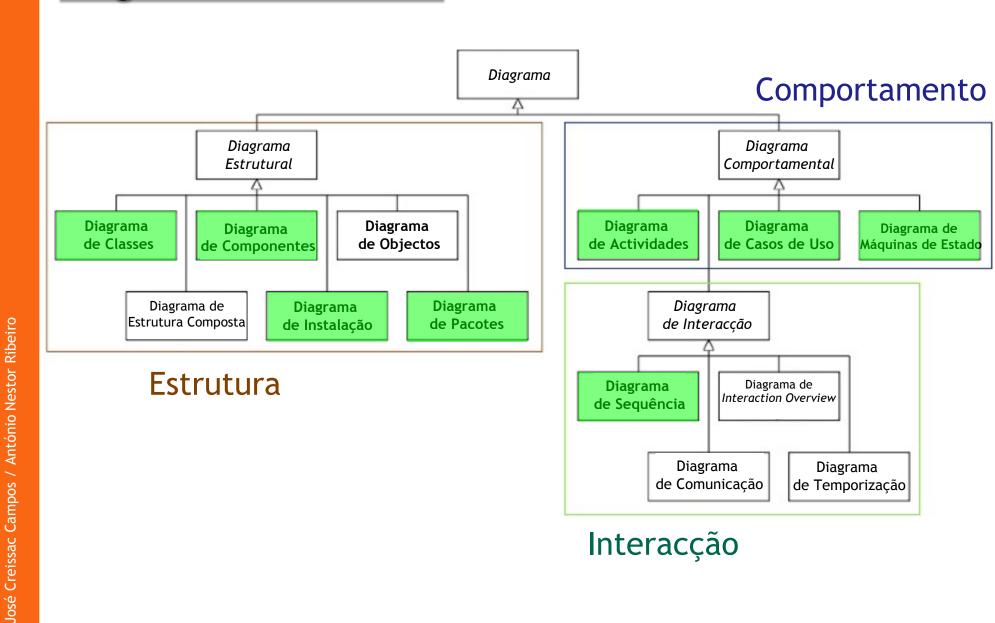
1 {userID="Jose Silva",year=2012,m onth=4,day=20,hour=17,min=57,sec

initialise animation

initial_time_SENSO

※ ○

Diagramas da UML 2.x

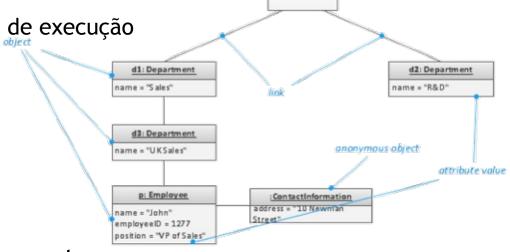




Diagramas de que não falamos

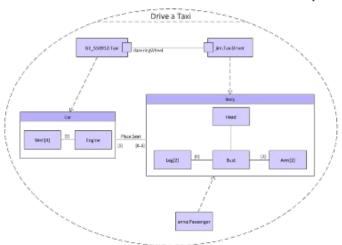
Diagramas de objectos

Mostra instâncias em tempo de execução



c: Company

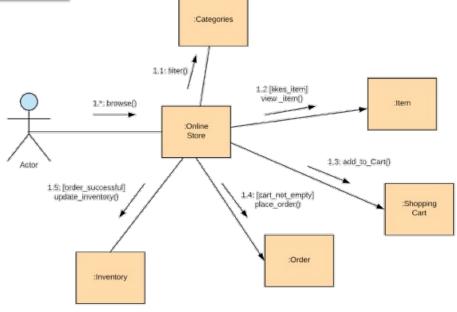
- Diagramas de estrutura composta
 - Mostra a estrutura interna de uma classe em termos de suas partes e das relações entre elas



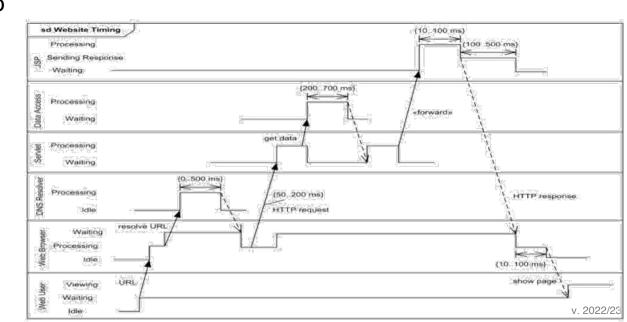
* 〇

Diagramas de que não falamos

- Diagramas de Comunicação
 - Uma alternativa aos diagramas de sequência, focada na relação estrutural entre os objectos



- Diagramas de Temporização
 - Focados na representação de tempo





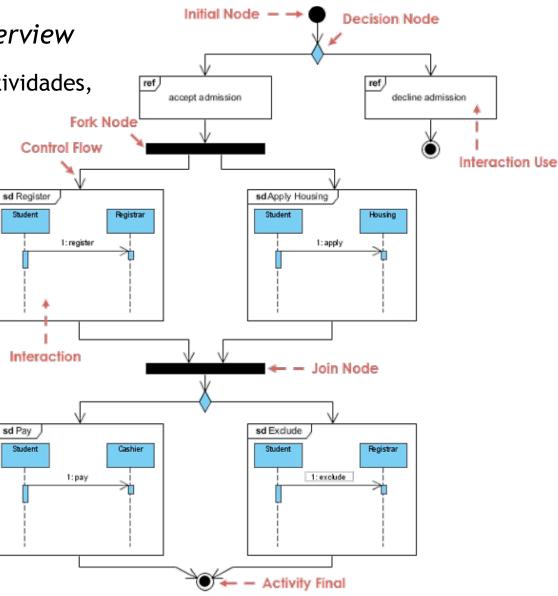
<u>Diagramas de que não falamos</u>

• Diagramas de Interaction Overview

Semelhante ao diagrama de actividades,

com diagramas de sequência

como actividades



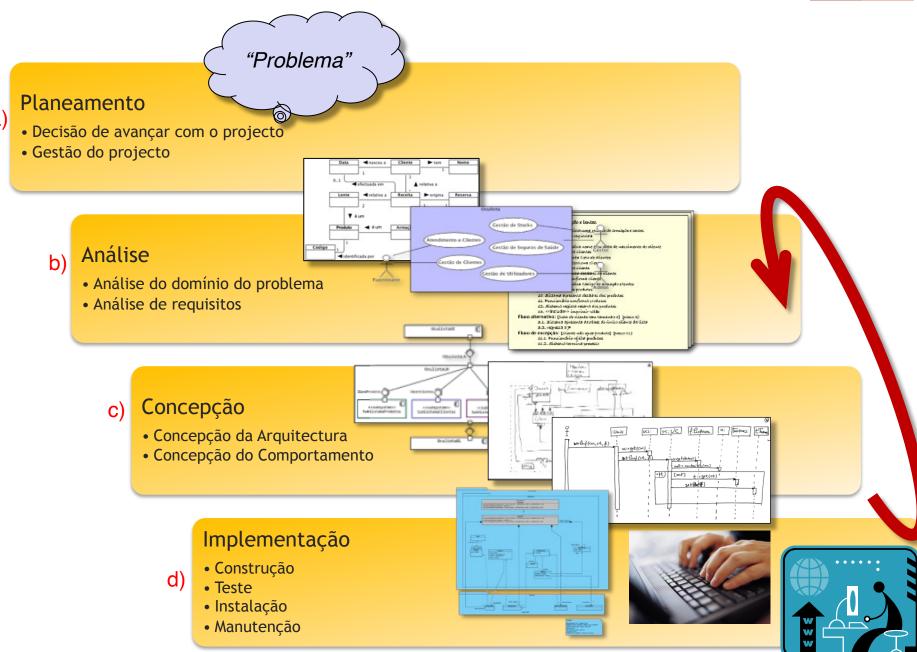




Desenvolvimento de Sistemas Software

Notas finais





Futuro da (na!) programaçã

Forbes



Robots Replacing Developers? This Startup Uses



I am surprised that most answers seem to be in the negative, or think that programming is one of the last fields where human labor becomes obsolete. General artificial intelligence will of course replace programmers at some point: the question is not if, but when, and at what level of sophistication of AI.

Home Answer Anstifications

programmers in the future?

How well do you know your tools and platforms?

Answer

73 Answers

The Little Black Book of

Promoted by VisionMobile

developers and find out!

Start now at vmob.me ₽

Joscha Bach

Answered Jun 9, 2016

Could artificially intelligent computers replace

Take the new developer economics survey, along with another 40,000

Follow 124 Comment 1 Downvote

Q Search Quora

¥ 1 ····

ACM TECHNEWS Al Will Replace Coders by 2040, Warn Academics



SHARE:

By V3.co.uk December 8, 2017

Comments

Coders and programmers could find themselves becoming marginalized by emerging technologies such as artificial intelligence, with humans being replaced in these jobs by 2040, according to a study from academic researchers published by Oak Ridge National Laboratory.

The report predicted by 2040, machine learning and natural language processing technologies will have become so advanced they will be able to write better software code faster than the best human coders.

In addition, "the major technologies that will drive the creation and adoption of machine-generated code already exist, either at

SIGN IN MORE NEWS & OPINIONS As Scrutiny Of Social Networks **Grows, Influence Attacks** Continue In Real Time **Connectivity Hacking Back**

Makes a Comeback - Rut It's Still

Create an ACM Web Account

SIGN IN for Full Access

User Name

Password

Forgot Password?

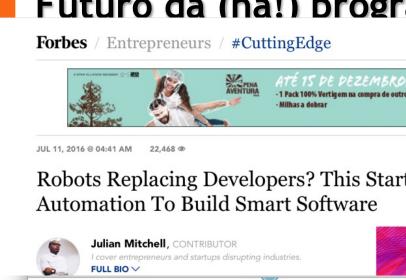
Best No-Code Development Platforms Software

HIGHEST RATED

that allow businesses to develop software quickly without coding. The platforms provide WYSIWYG editors and drag-and-drop

No-Code Development Platform provide drag-and-drop tools that

SHOW MORE .



Home / News / Al Will Replace Coders by 2040, Warn Academics / Full Text





• Decisão de avançar com o projecto

• Gestão do projecto

Análise

- Análise de requisitos

Concepção

- Concepção da Arquitectura
- Concepção do Comportamento

Implementação

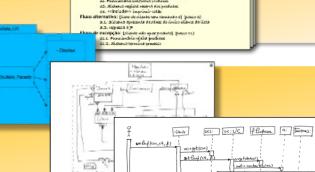
- Construção
- Teste
- Instalação
- Manutenção











estão de Stocks





Relembrar...

- ECTS
 - 5 ECTS = 140h
 - 15 semanas -> 140/15 = 9h20m/semana (para DSS!)
 - Aulas = 4h/semana (na verdade são só 13 semanas!)
 - Trabalho autónomo = 5h20m/semana = ~1h/dia útil
 - 1h por dia = 75h de trabalho autónomo em 15 semanas (por aluno!)
- Teste / Exame
 - Podem levar duas folhas (4 páginas) com a informação que considerarem relevante

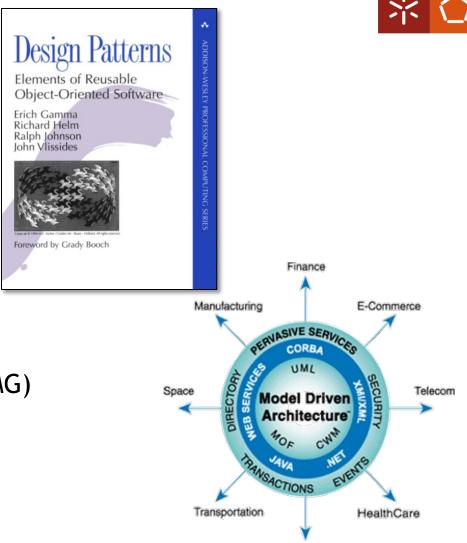
Mais..

Software design patterns



- - Model Driven Arquitecture (OMG)

- <pub>Engenharia de Aplicações</pub>
 - Desenvolvimento de Aplicações Multi-camada (Web)



More...