

ENGENHARIA DE SOFTWARE

INTRODUÇÃO

Luís Morgado

2021

ENGENHARIA DE SOFTWARE

- Desenvolvimento de Software como uma actividade de engenharia
 - **Sistemático**
 - **Quantificável**
- **Engenharia de Software:**
 - Aplicação de abordagens sistemáticas, disciplinadas e quantificáveis ao desenvolvimento, operação e manutenção de software [IEEE, 1990]
- Problemas principais
 - **Complexidade**
 - **Mudança**

CRESCIMENTO EXPONENCIAL DE RECURSOS COMPUTACIONAIS



IBM Real-Time Computer Complex - NASA Manned Spacecraft Center
Década de 1960

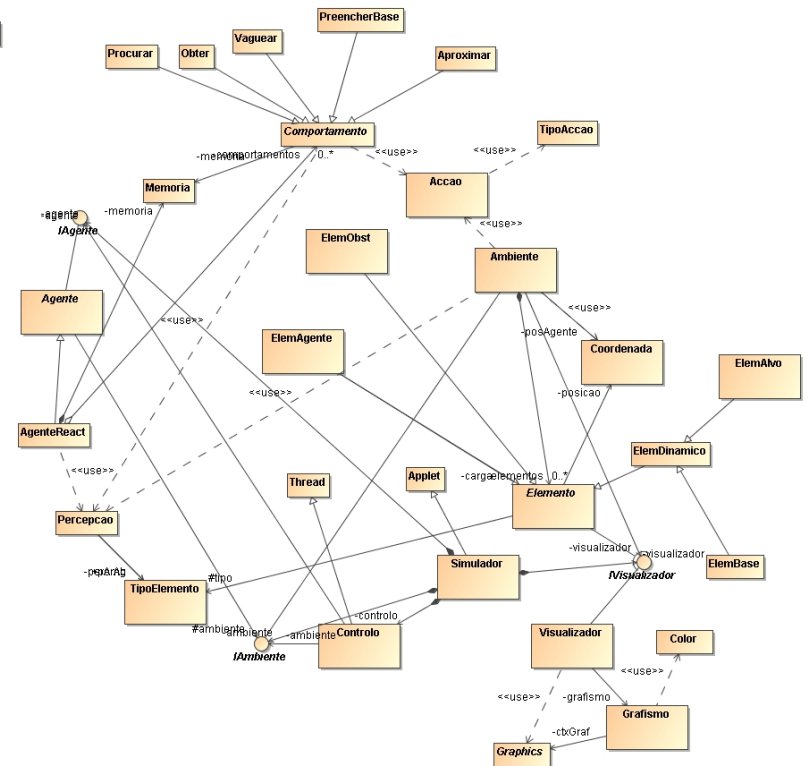
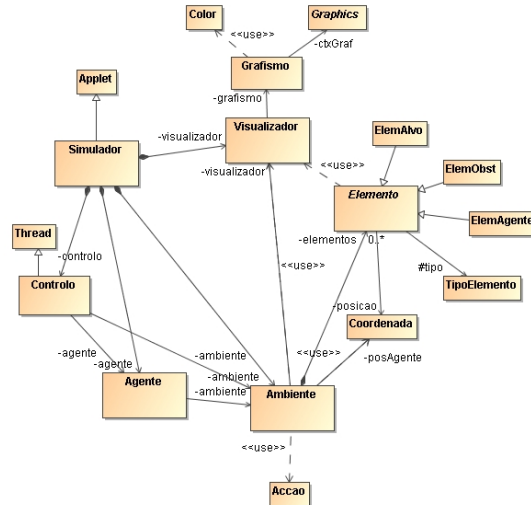
SOFTWARE



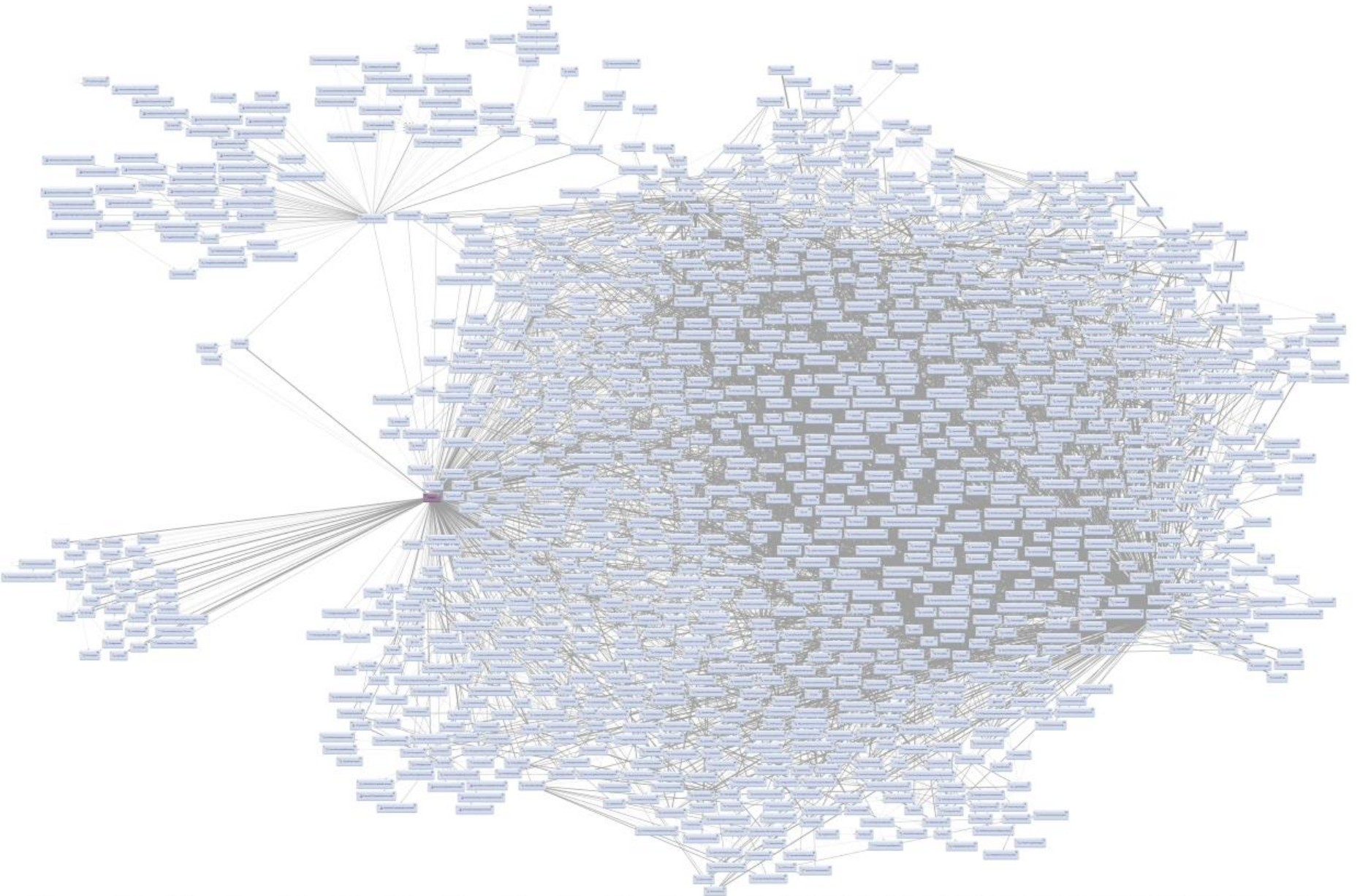
Hoje

Diagrama de relaciones de dependencia entre componentes de un sistema de simulación:

- Thread** depende de **Controlo**.
- Controlo** depende de **Agente** (rol: -agente) y **Simulador** (rol: -controlo).
- Agente** depende de **Controlo** (rol: -agente) y **Simulador** (rol: -agente).
- Simulador** depende de **Controlo** (rol: -controlo) y **Visualizador** (rol: -visualizador).
- Visualizador** depende de **Simulador** (rol: -visualizador) y **Grafismo** (rol: -grafismo).
- Visualizador** depende de **Controlo** (rol: -visualizador).
- Controlo** depende de **Ambiente** (rol: -ambiente).
- Ambiente** depende de **Visualizador** (rol: -visualizador).
- Ambiente** depende de **Accao** (rol: <<use>>).
- Applet** depende de **Simulador**.



SOFTWARE E COMPLEXIDADE



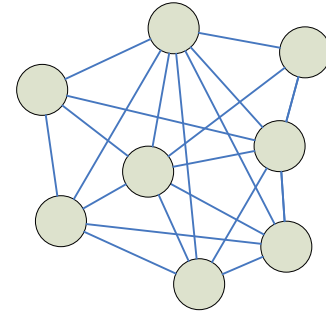
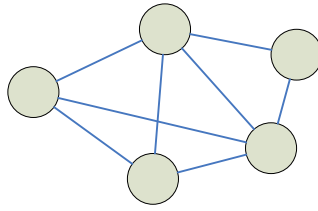
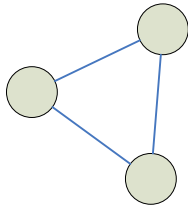
COMPLEXIDADE

Grau de **difículdade de previsão** das propriedades de um sistema dadas as propriedades das partes individuais [Weaver, 1948]

- Relacionada com a **informação** que é necessária para a caracterização de um sistema
- Um sistema é tanto mais **complexo** quanto mais **informação** for necessária para a sua **descrição**
- Reflete-se no **esforço** necessário para geração da **organização (ordem)** do sistema

O PROBLEMA DA COMPLEXIDADE

COMPLEXIDADE ESTRUTURAL



- **UM PROBLEMA DE INTERACÇÃO**

- De partes do sistema
- De elementos de informação
- De elementos das equipas de desenvolvimento

- **EXPLOÇÃO COMBINATÓRIA**

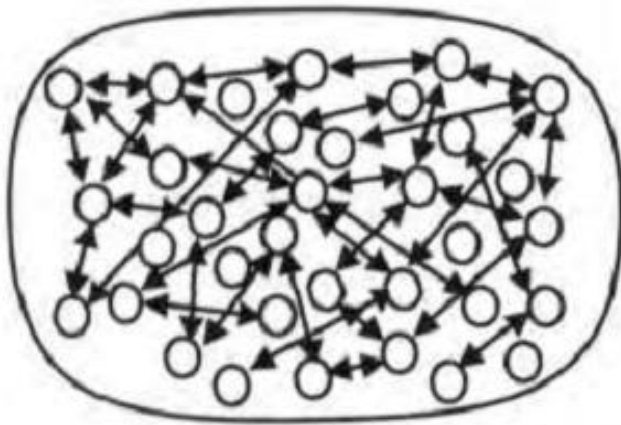
- Um sistema com duas vezes mais partes é muito mais do que duas vezes mais complexo

**CRESCIMENTO EXPONENCIAL
DA COMPLEXIDADE**

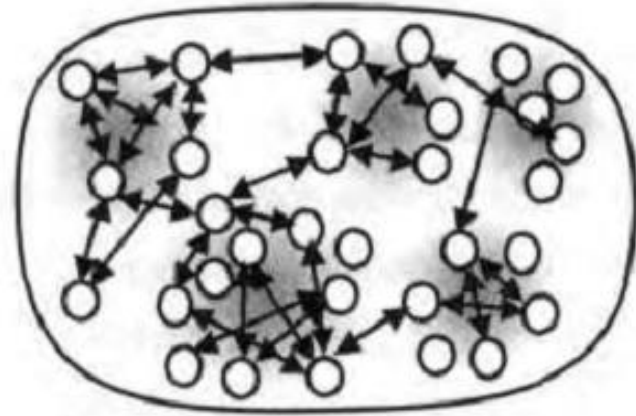
COMPLEXIDADE E ORGANIZAÇÃO

Diferentes tipos de complexidade associados à organização de um sistema

- Ordem, organização, função, propósito
- Desordem, desorganização, perda de função e de propósito



AN UNSTRUCTURED SYSTEM



AN “ORGANIZED” SYSTEM

TIPOS DE COMPLEXIDADE

COMPLEXIDADE DESORGANIZADA

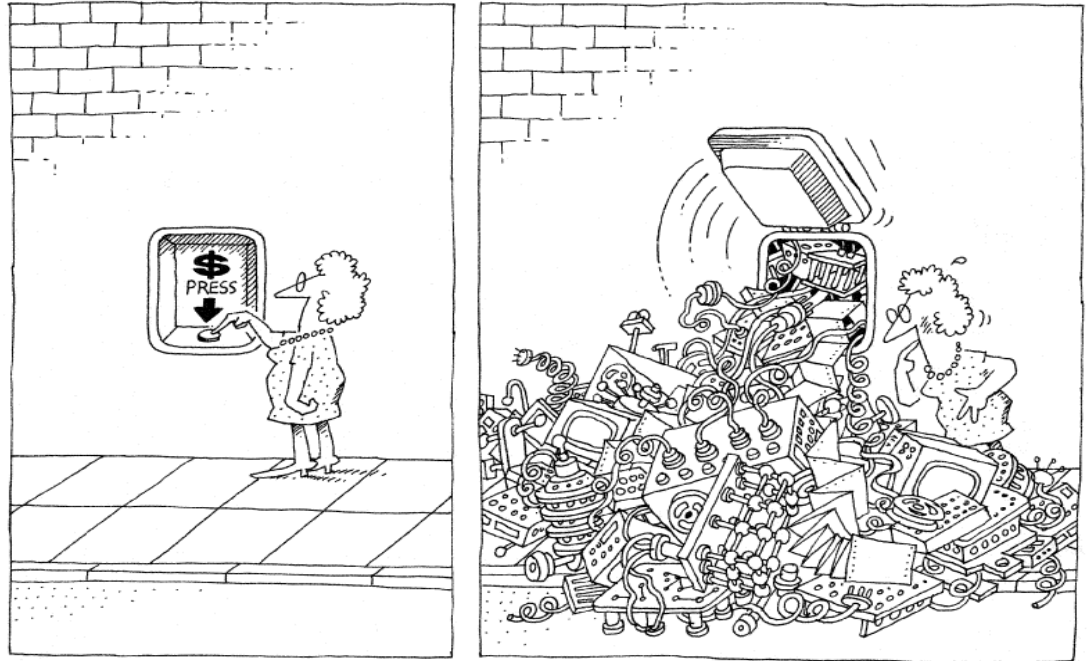
- resulta do **número e heterogeneidade** das partes de um sistema
- as partes podem interactuar entre si, mas a **interacção é irregular**
- as características globais do sistema podem ser inferidas com base em **métodos estatísticos**

COMPLEXIDADE ORGANIZADA

- resulta dos padrões de **inter-relacionamento** entre as partes
- as interacções entre partes obedecem a **padrões correlacionáveis** no espaço e no tempo
- **ORDEM, ORGANIZAÇÃO**
 - Propósito (finalidade)
 - Função

ARQUITECTURA DE SOFTWARE

- MÉTRICAS
- PRINCÍPIOS
- PADRÕES



[Booch, 2004]

COMPLEXIDADE

- Redução
- Controlo

MÉTRICAS DE ARQUITECTURA

- **ACOPLAMENTO**

- Grau de interdependência entre subsistemas

- **COESÃO**

- Nível coerência funcional de um subsistema/módulo (até que ponto esse módulo realiza uma única função)

- **SIMPLICIDADE**

- Nível de facilidade de compreensão/comunicação da arquitectura

- **ADAPTABILIDADE**

- Nível de facilidade de alteração da arquitectura para incorporação de novos requisitos ou de alterações nos requisitos previamente definidos

ACOPLAMENTO E COESÃO

- **ACOPLAMENTO**

- Grau de interdependência entre subsistemas
- Característica **inter-modular**

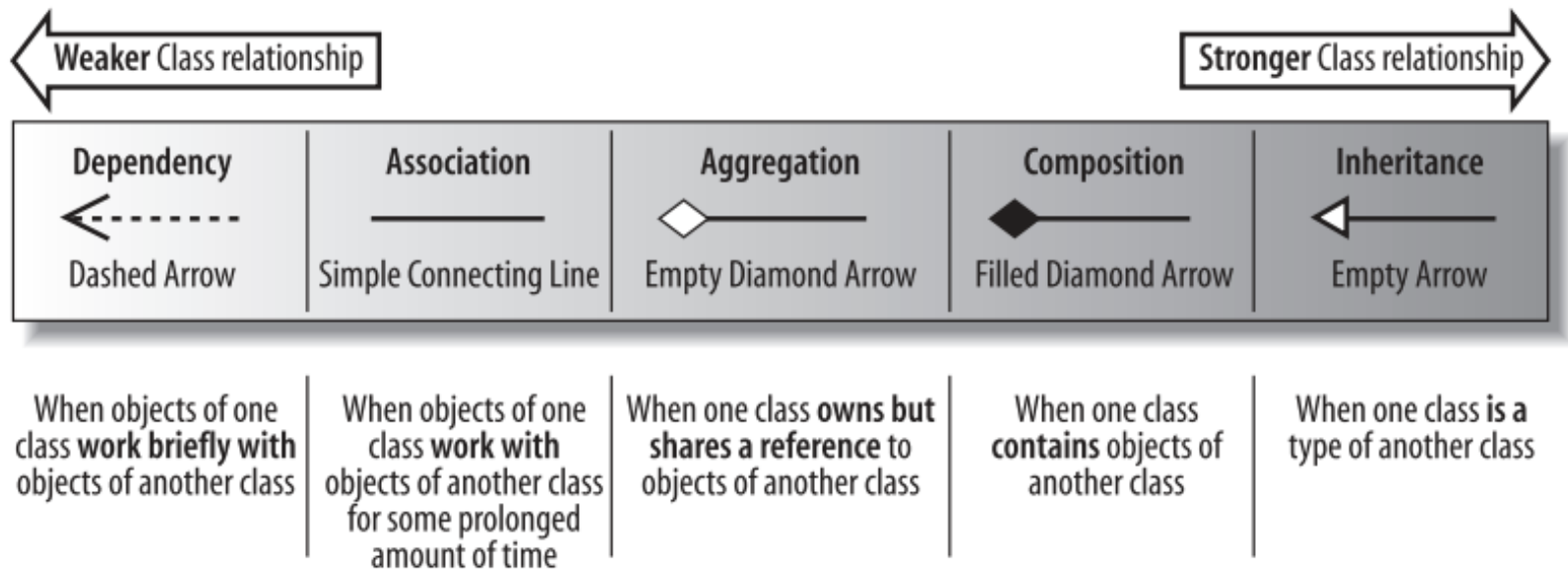
- **COESÃO**

- Nível coerência funcional de um subsistema/módulo (até que ponto esse módulo realiza uma única função)
- Característica **intra-modular**

ACOPLAMENTO EM MODELOS DE ESTRUTURA

Linguagem UML

Relações entre classes e nível de acoplamento



[Miles & Hamilton, 2006]

PRINCÍPIOS DE ARQUITECTURA

- **MODULARIDADE**

- DECOMPOSIÇÃO
- ENCAPSULAMENTO



- ACOPLAMENTO

+ COESÃO



- **FACTORIZAÇÃO**



COMPLEXIDADE

- **ABSTRACÇÃO**

MODULARIDADE

- **DECOMPOSIÇÃO**

- De um sistema em partes coesas
 - Para sistematizar interacções
 - Para lidar com a explosão combinatória
- **FACTORIZAÇÃO**
 - Eliminação de redundância
 - Garantia de consistência

- **ENCAPSULAMENTO**

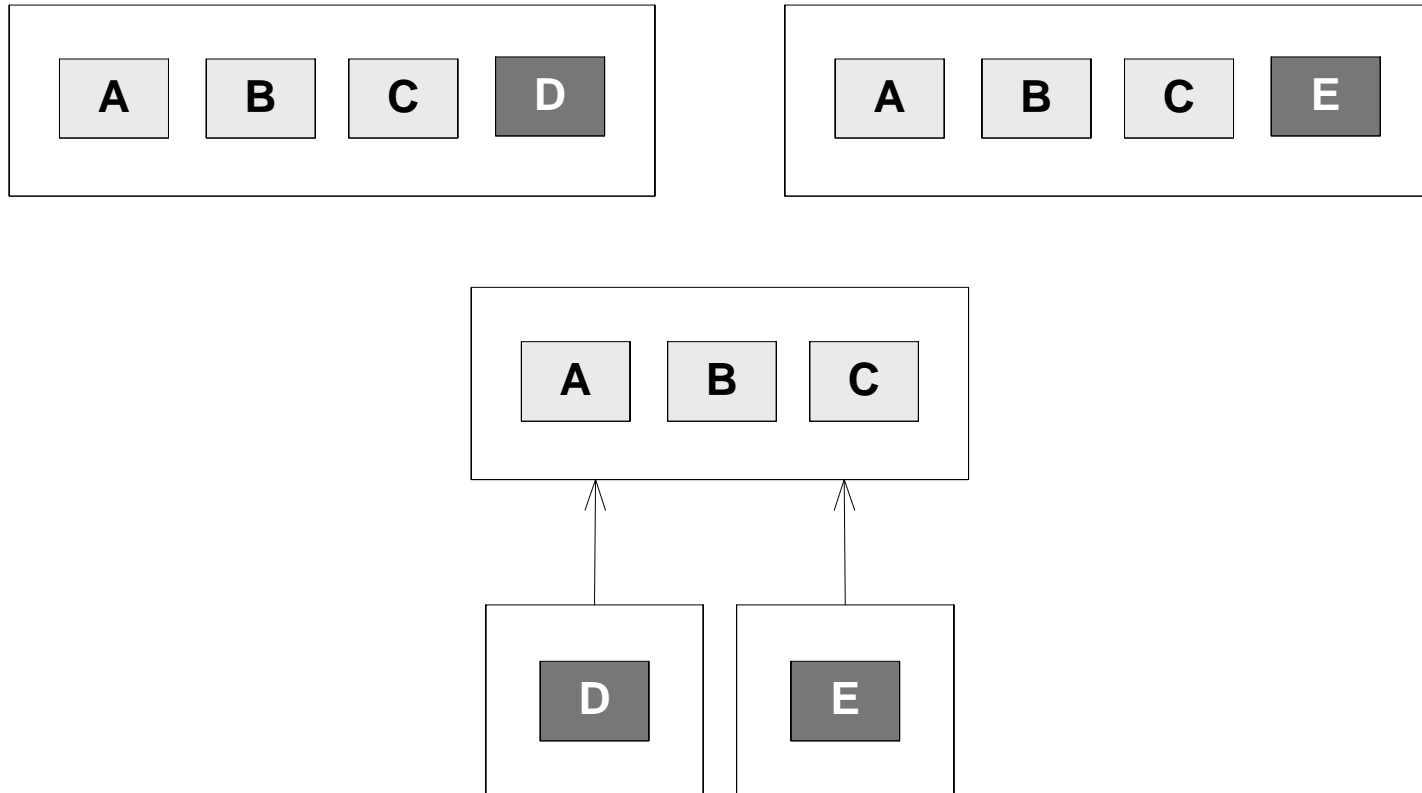
- **Isolamento dos detalhes internos** das partes de um sistema em relação ao exterior
 - Para reduzir dependências (interacções)
 - Relacionar estrutura e função no contexto de uma parte
 - Acesso exclusivo através das interfaces disponibilizadas
- **INTERFACES**
 - **Contractos funcionais** para interação com o exterior

FACTORIZAÇÃO

REDUNDÂNCIA

Uma das principais causas de anomalias no desenvolvimento de software

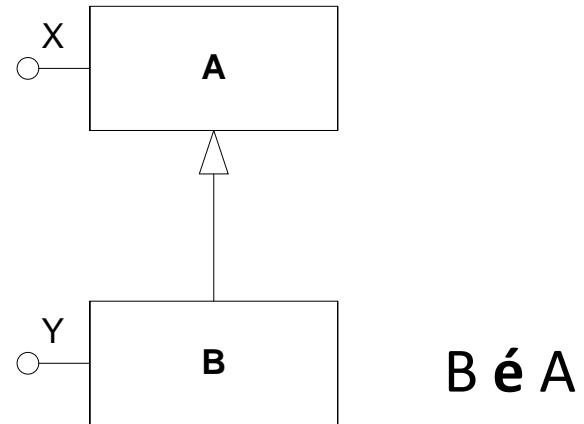
REDUÇÃO DE REDUNDÂNCIA



MECANISMOS DE FACTORIZAÇÃO

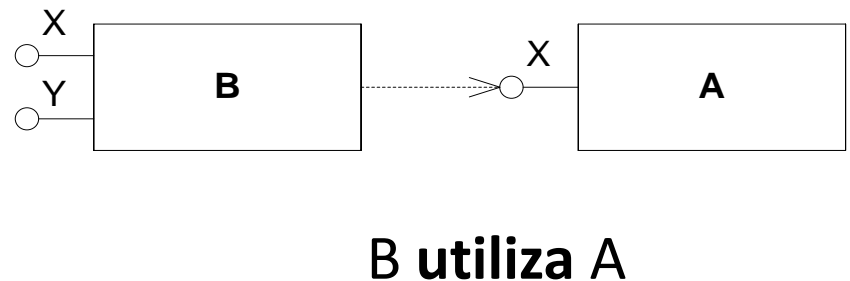
HERANÇA

- Nível de **acoplamento alto**
- B é A



DELEGAÇÃO

- Nível de **acoplamento baixo**
- B utiliza A
- Agregação de partes
- Acoplamento pode variar dinamicamente



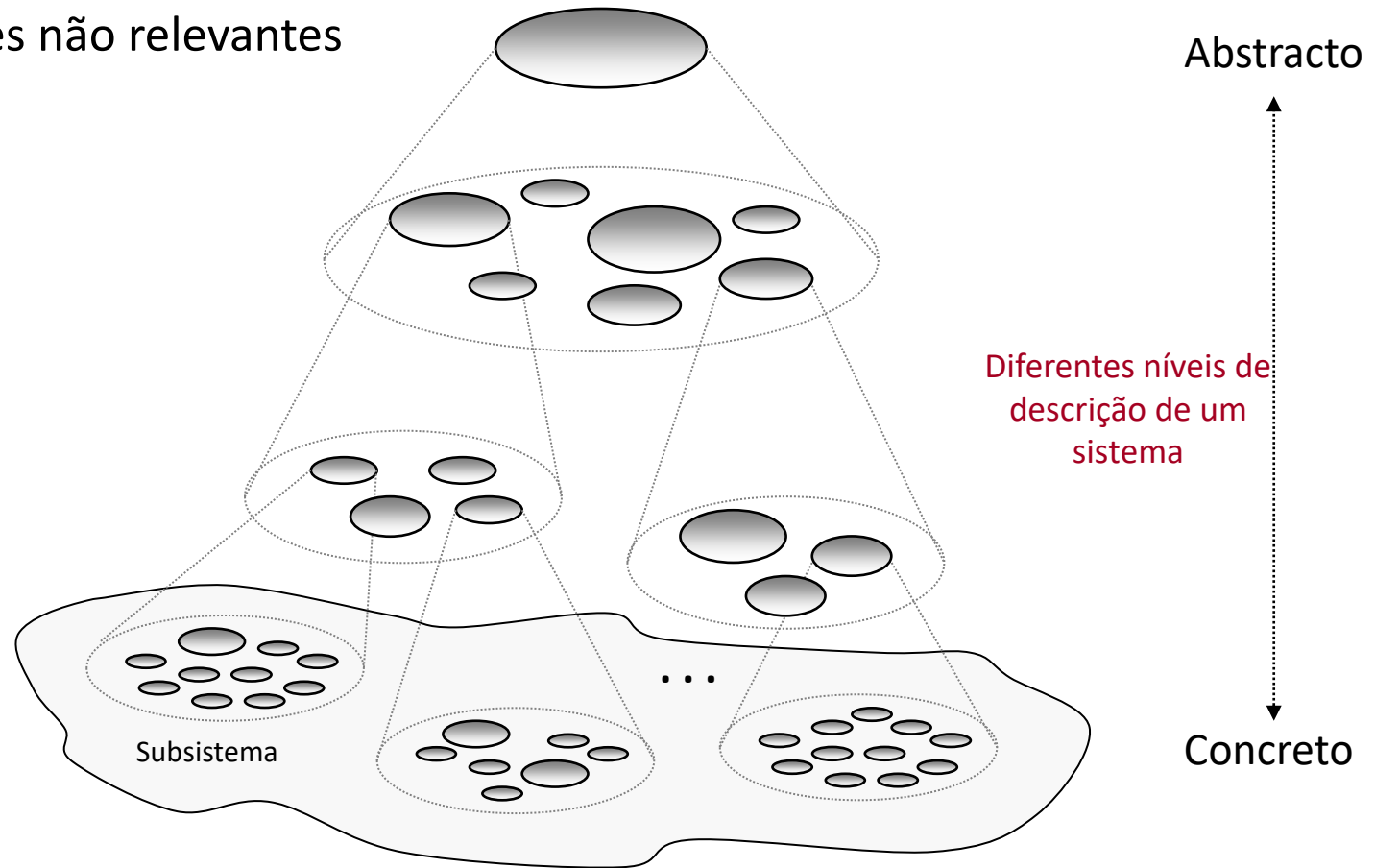
ABSTRACÇÃO

- Processo de descrição de conhecimento a *diferentes níveis de detalhe* (**quantidade de informação**) e *tipos de representação* (**estrutura da informação**) [Korf, 1980]
- **Abstracção é uma ferramenta base para lidar com a complexidade**
 - Identificação de características comuns a diferentes partes
 - Realçar o que é essencial, omitir detalhes não relevantes
 - Modelos
- Desenvolvimento de um sistema complexo
 - Criação de ordem de forma progressiva
 - Processo iterativo guiado por conhecimento

ABSTRACÇÃO

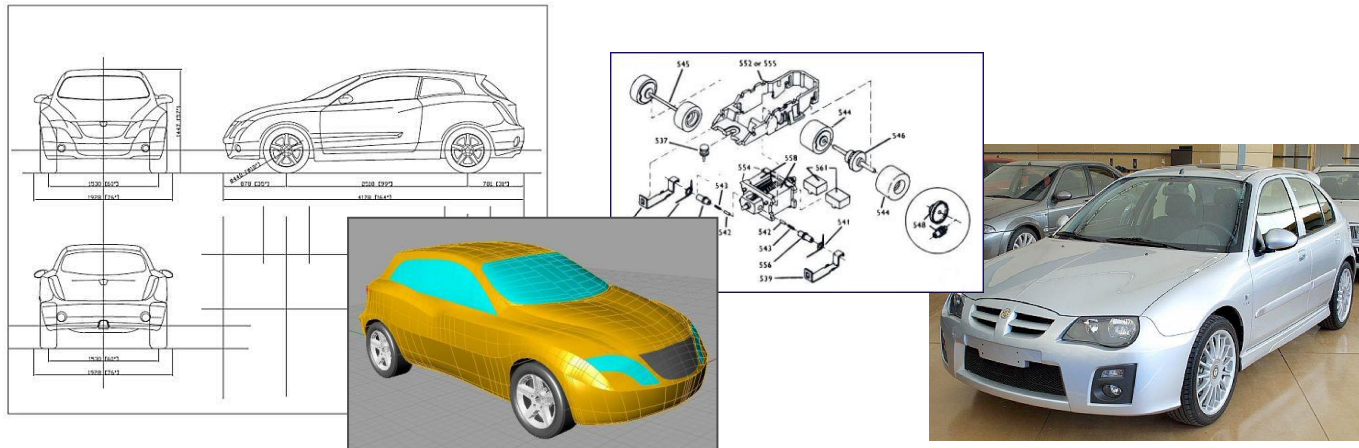
FERRAMENTA BASE PARA LIDAR COM A COMPLEXIDADE

- **Realçar** o que é **essencial**, omitir detalhes não relevantes
- **Simplificação**
- **Focagem**
- **Modelos**



MODELO

- **Representação abstracta de um sistema**
 - Especificação com base em conceitos abstractos das características fundamentais de um sistema
 - Representação de conhecimento acerca de um sistema
- **Meio para lidar com a complexidade**
 - Obtenção e sistematização progressiva de conhecimento
 - Compreensão e comunicação acerca do sistema
 - Especificação de referência para a realização do sistema
 - Documentação de um sistema



MODELAÇÃO DE UM SISTEMA

DEFINIÇÃO DOS PADRÕES DE ORGANIZAÇÃO DO SISTEMA

CARACTERÍSTICAS IMPORTANTES DE UM MODELO

– ABSTRACÇÃO

- Foco nos aspectos importantes, remoção de aspectos não relevantes

– COMPREENSÃO

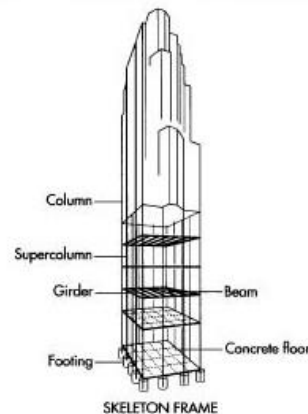
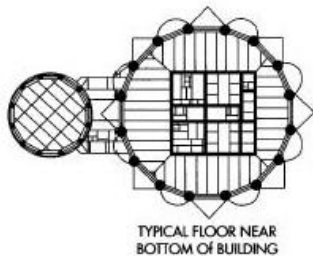
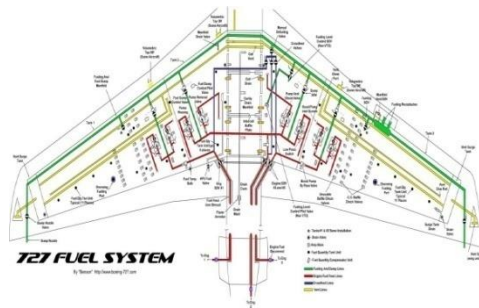
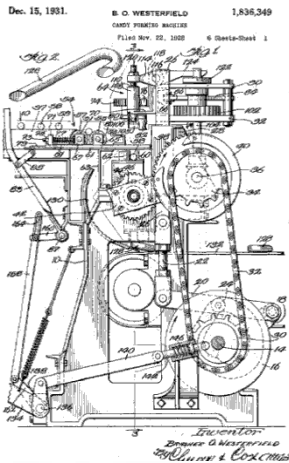
- Facilidade de transmissão e compreensão das ideias envolvidas

– PRECISÃO

- Representação correcta e rigorosa do sistema

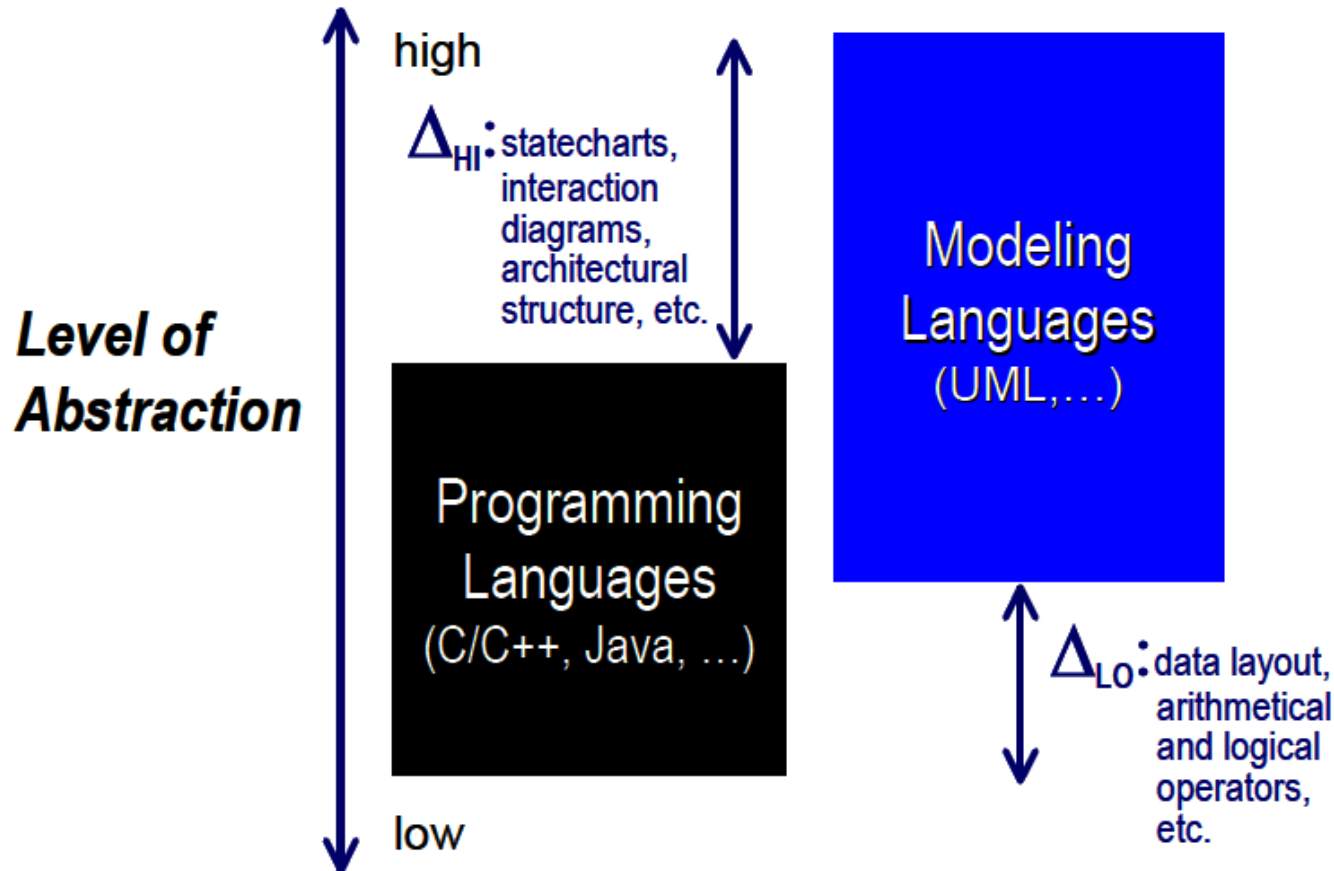
– PREVISÃO

- Possibilidade de inferência de conhecimento correcto acerca do sistema descrito

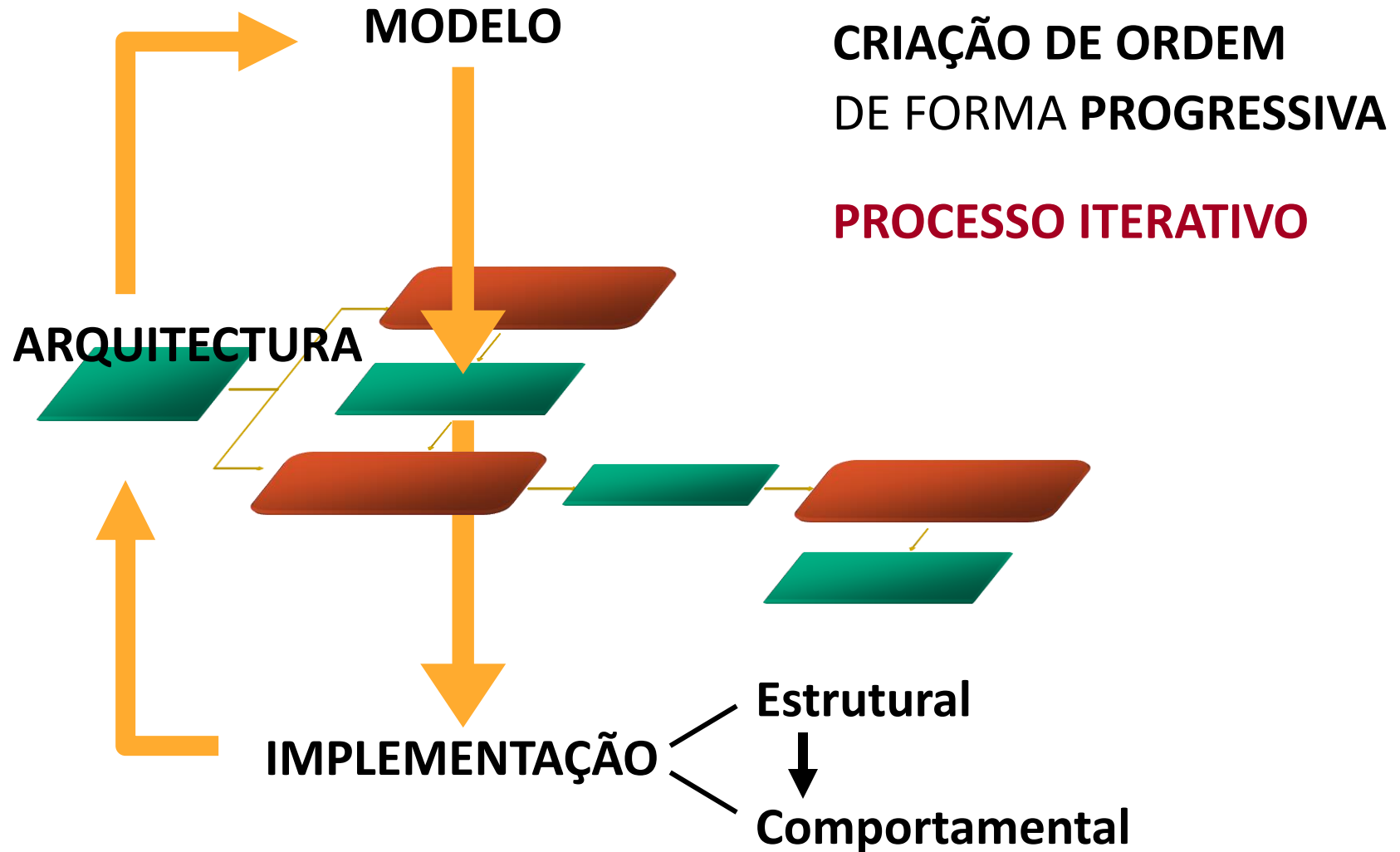


LINGUAGENS DE MODELAÇÃO

DESCRIÇÃO DO SISTEMA A DIFERENTES NÍVEIS DE ABSTRACÇÃO



PROCESSO DE DESENVOLVIMENTO



BIBLIOGRAFIA

[Pressman, 2003]

R. Pressman, *Software Engineering: a Practitioner's Approach*, McGraw-Hill, 2003.

[Booch et al., 1998]

G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1998.

[Miles & Hamilton, 2006]

R. Miles, K. Hamilton, *Learning UML 2.0*, O'Reilly, 2006.

[Eriksson et al., 2004]

H. Eriksson, M. Penker, B. Lyons, D. Fado, *UML 2 Toolkit*, Wiley, 2004.

[Douglass, 2009]

B. Douglass, *Real-Time Agility: The Harmony/ESW Method for Real-Time and Embedded Systems Development*, Addison-Wesley, 2009.

[SRC, 2015]

Semiconductor Research Corporation, *Rebooting the IT Revolution*, 2015.

[Korf, 1980]

R. Korf, Toward a model of representation changes, *Artificial Intelligence*, Volume 14, Issue 1, 1980.