

**INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE**

---

# Análise e Projeto Orientados a Objetos

Introdução

---

**Diretoria Acadêmica de Gestão e Tecnologia da Informação  
Curso Técnico Integrado em Informática para Internet**

# Introdução

- Os sistemas computacionais adquiriram extrema importância para as organizações públicas e privadas.
  - Sistemas de informações dão vantagens competitivas.
- O software é um elemento fundamental dos sistemas computacionais.
- Problema: o desenvolvimento de software é uma tarefa complexa.
  - Saber programar é uma, mas não a única, habilidade necessária.

# Introdução

- Quanto maior o escopo do software, mais complexo será o seu desenvolvimento.
- Os sistemas computacionais atuais lidam com vários componentes: interfaces, hardware, procedimentos, fator humano, ...
- Nos primórdios da computação o software era simples e desenvolvido *ad-hoc*.
- A demanda por software mais complexo aumentou.

# Introdução

- Década de 70: identificação da crise do software:
- Fatos da crise do software:
  - Demanda superior à capacidade de desenvolvimento.
  - Qualidade insuficiente dos produtos.
  - Estimativas de custo e prazo raramente cumpridas.

# Introdução

- O Standish Group realiza pesquisas sobre projetos de desenvolvimento de software.



	1994	1996	1998	2000	2002	2004	2006	2009
Successful	16%	27%	26%	28%	34%	29%	35%	32%
Challenged	53%	33%	46%	49%	51%	53%	46%	44%
Failed	31%	40%	28%	23%	15%	18%	19%	24%

# Introdução

- A **Engenharia de Software** surgiu como um dos resultados do esforço para superar a crise do software.
- Engenharia de Software:
  - Área do conhecimento humano que tem como objetivo organizar ferramentas, técnicas e métodos que permitam a construção de software de qualidade a um custo aceitável.

# Introdução

- **Processo de desenvolvimento de software (PDS):**
  - Conjunto de atividades para definir, desenvolver, testar e manter um produto de software.
- **Alguns exemplos de PDS:**
  - RUP, XP, PU, Scrum.
- **Atividades típicas de um PDS:**
  - Análise, projeto, implementação, teste e implantação.

# Introdução

- **Processo de desenvolvimento de software (PDS):**
  - Conjunto de atividades para definir, desenvolver, testar e manter um produto de software.
- **Alguns exemplos de PDS:**
  - RUP, XP, PU, Scrum.
- **Atividades típicas de um PDS:**
  - **Análise, projeto,** implementação, teste e implantação.

Foco da disciplina



# Análise

- Tem como foco a investigação do problema e a identificação dos requisitos.
- Procura descrever **o que** deve ser feito.
- Pode ser dividida em:
  - Análise de requisitos.
  - Análise orientada a objetos (AOO).

# Análise de Requisitos

- Trata de descobrir qual o problema do cliente e quais as suas necessidades.
- Está associada ao processo de descobrir o que o sistema deve realizar e as restrições sobre cada operação.
- Também conhecida como **levantamento de requisitos**.
- Resumo: é uma investigação dos requisitos.

# Análise Orientada a Objetos

- Tem como ênfase encontrar e descrever os objetos – ou conceitos – no domínio do problema.
  - Domínio do problema: área de conhecimento específica na qual um sistema de software está sendo desenvolvido.
- Resumo: é uma investigação dos objetos do domínio.

# Projeto

- Tem com ênfase criar uma solução computacional para as necessidades identificadas na análise.
- Procura descrever **como** deve ser feito.
- O **projeto orientado a objetos** procura definir os objetos de software e como eles colaboram para a satisfação dos requisitos.

# Análise e Projeto OO

Conceito do domínio

Visualização de conceito do domínio

Projeto OO



Veículo

placa  
nº do chassi

modelo::Veículo

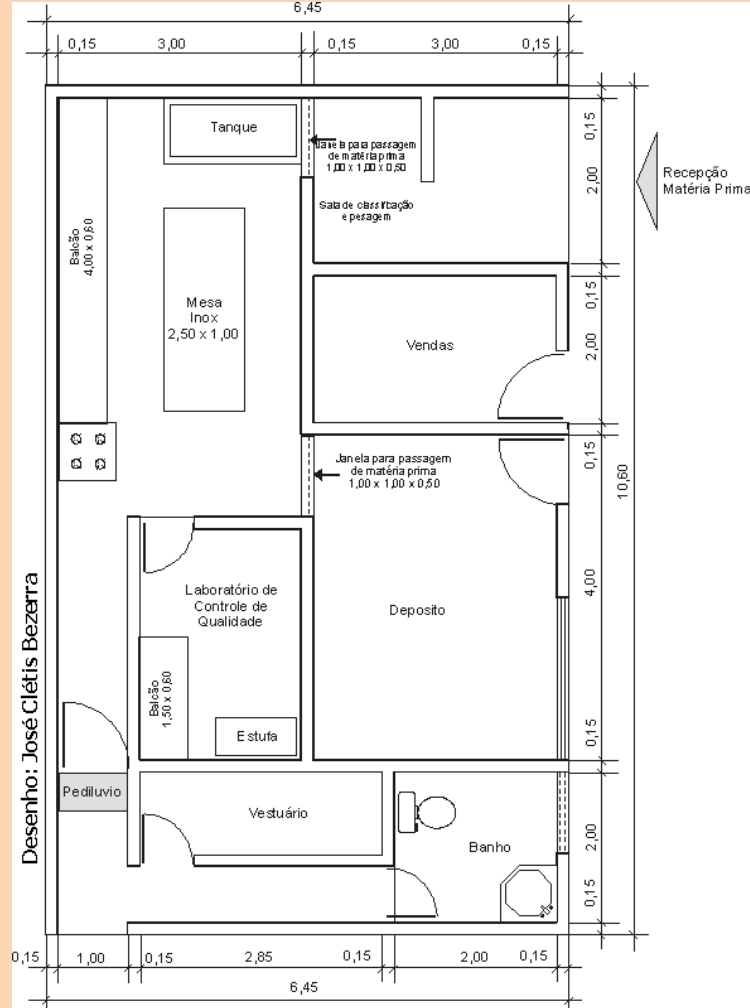
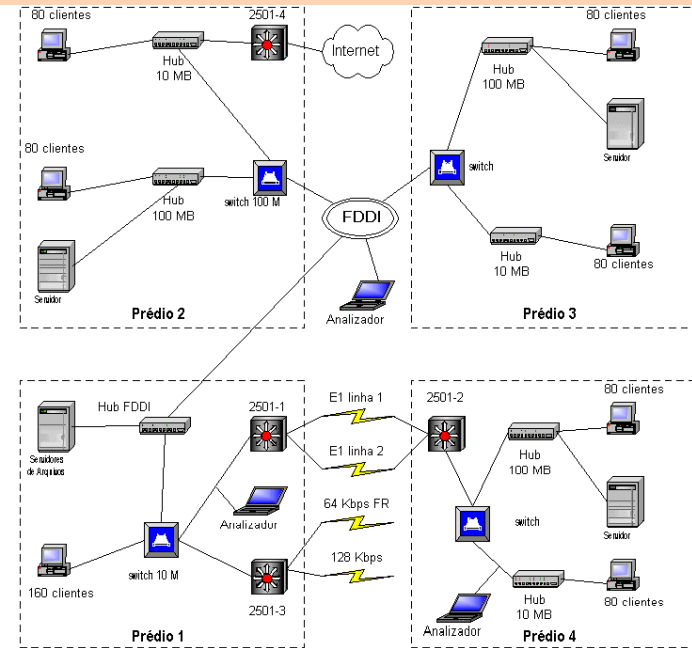
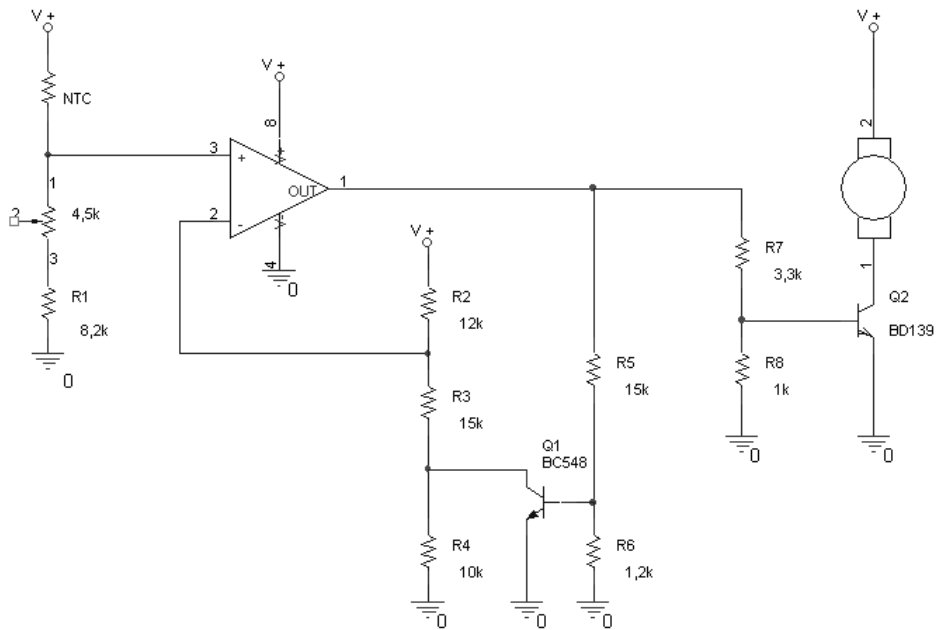
- placa : string  
- chassi : string

Representação em uma linguagem de programação OO

```
class Veiculo{  
    private String placa;  
    private String chassi;  
    public void vender(){...}  
}
```

# Expressando projetos

- Para expressar projetos (arquitetônicos, navais, mecânicos, etc), os engenheiros utilizam padrões estabelecidos e os aplicam nas plantas.



# Unified Modeling Language

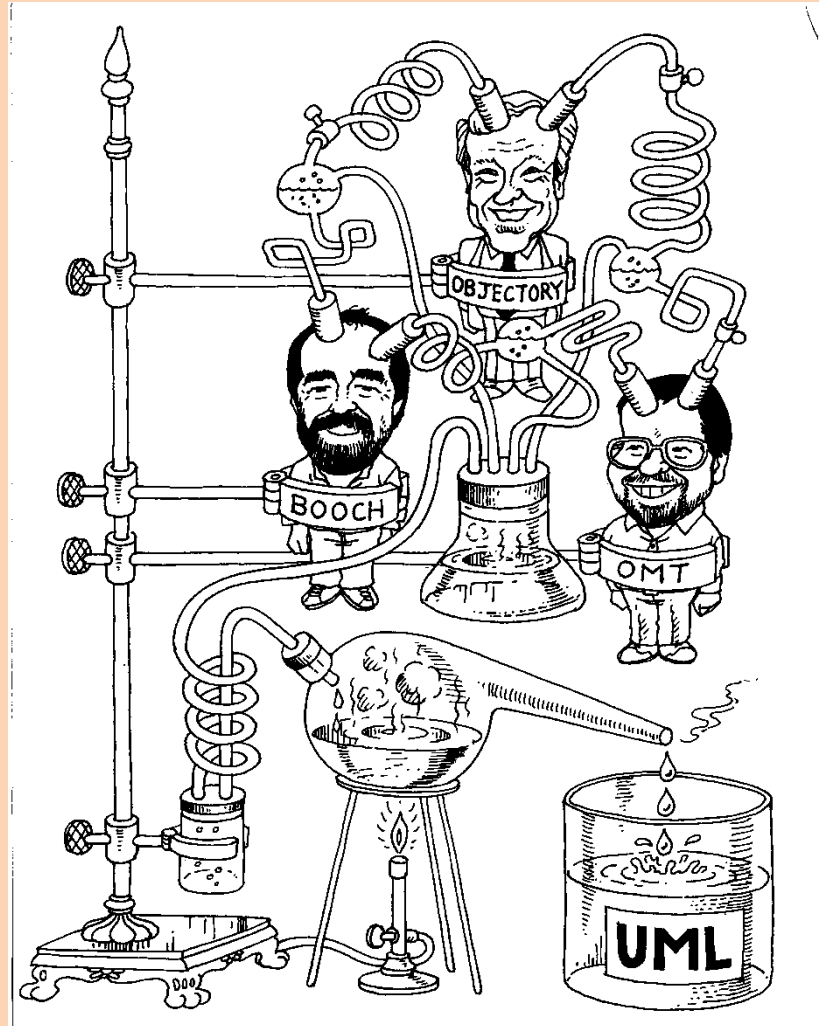


- É uma linguagem visual para especificar, construir e documentar os artefatos de um sistema de software.
- A versão 2 é a mais recente.
- Define um conjunto de diagramas que podem ser classificados como estruturais e comportamentais.
- É independente de linguagem de programação.

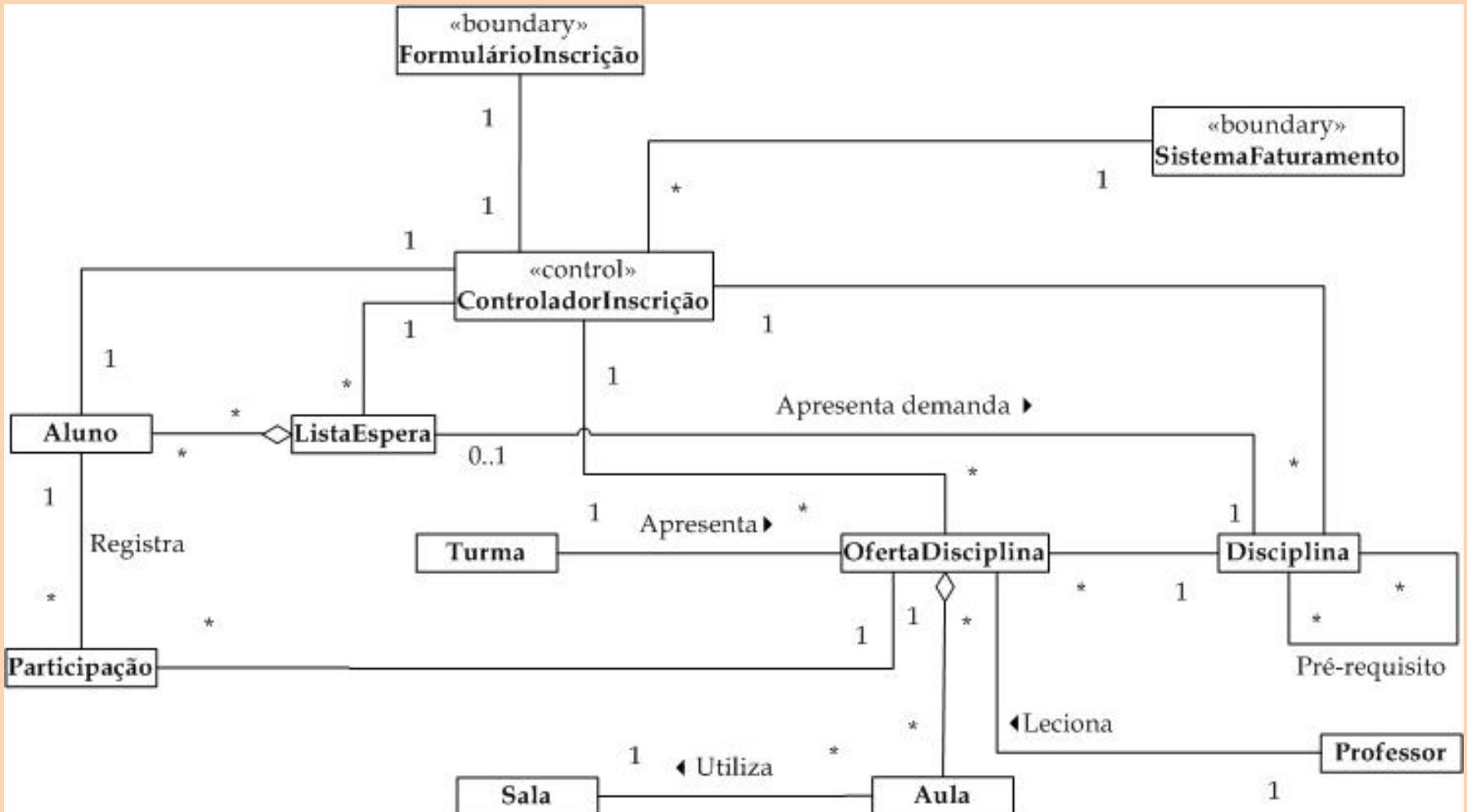


# UML

- É independente do processo de desenvolvimento.
- **Não** é uma técnica de modelagem.
- Foi criada inicialmente pelos 3 amigos: Booch, Rumbaugh e Jacobson.
- O desenvolvimento da UML é controlado pelo OMG.



# Exemplo de Diagrama



# Modos de aplicar a UML

- **Como rascunho:** diagramas incompletos e informais criados para explorar partes difíceis do problema.
- **Como planta de software:** diagramas detalhados usados para documentação, geração de código ou como resultado de engenharia reversa.

# Referências

- BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro: Elsevier, 2002
- LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objeto e ao desenvolvimento iterativo**. Porto Alegre: Bookman, 2007, 3. ed.
- Transparências do professor Raul Sidnei Wazlawic (UFSC).
- Notas de aula do professor Ricardo de Almeida Falbo (UFES).