



Escolaridade

- 2T + 2PL
- Registo de presenças nas PL (regras do RIAPA)

Equipa Docente

- José Creissac Campos (jose.campos@di.uminho.pt) — T / PL
 - horário de atendimento: 5^{as} 14h-17h (marcação prévia!)
- António Nestor Ribeiro (anr@di.uminho.pt) — PL

Canais de comunicação

- Aulas teóricas (global)
- Blackboard (global/turno)
- Aulas PL (turno)



Avaliação

- Exame (≥ 9.0) - uma prova escrita sobre a matéria teórica
- Trabalho Prático (≥ 10.0)
- Classificação Final (≥ 10.0)
 $.5 \text{ Exame} + .5 \text{ Trabalho} - 3(\text{Trabalho} - \text{Exame})/\text{Exame}$
- Congelamentos de nota prática
 - Só notas superiores a 10 de 2010/11
 - Nota em 2011/12 sujeita a um tecto de 13 valores



Trabalho Prático

- Grupos de 3-5 elementos;
 - Aumentar a capacidade de trabalho
 - Fomentar a discussão de soluções alternativas
 - Assumir vários papéis na equipa
- A realizar em duas fases durante o semestre:
 - Fase 1: análise de requisitos
 - até **4 de Novembro** (4+ semanas) - 35% da nota
 - Fase 2: relatório final e o software produzido
 - até **30 de Dezembro** (8 semanas) - 65% da nota
- Apresentação e discussão: semana de 02 de Janeiro 2012



Trabalho Prático

- Enunciado partilhado com Base de Dados
- Em DSS é avaliado o desenvolvimento da camada de negócio
- EM BD é desenvolvida a camada de dados
 - Avaliação independente



Trabalho Prático

- Enunciado partilhado com Base de Dados
- Em DSS é avaliado o desenvolvimento da ca
- EM BD é desenvolvida a camada de dados
 - Avaliação independente
- Tema: Um Sistema de Informação para perscrição médica
 - Permitir ao médico passar receita
 - disponibilizando uma base de dados de medicamentos
 - Permitir ao farmacêutico aviar receitas
 - apresentando possíveis alternativas mais baratas
 - Permitir obter estatísticas sobre medicamentos perscritos / aviados

The image shows a medical prescription form with two main sections: 'Receita Médica Nº' (Medical Prescription Number) and 'Guia de tratamento para o utente' (Treatment Guide for the patient). The 'Receita Médica Nº' section includes fields for the doctor's name, address, and contact information, as well as a table for listing medications. The 'Guia de tratamento para o utente' section includes fields for the patient's name, address, and contact information, as well as a table for listing treatments. The form is designed to be filled out by a doctor and a pharmacist.



O que é um Sistema de Informação?

Sistema. *s. m.* Reunião de partes ligadas entre si, formando uma estrutura complexa. Conjunto de meios, processos destinados a produzir um resultado = Método.

Informação. *s. f.* Conjunto de conhecimentos reunidos sobre um determinado assunto; documentação.

Dicionário de Língua Portuguesa Contemporânea. Academia de Ciências de Lisboa.

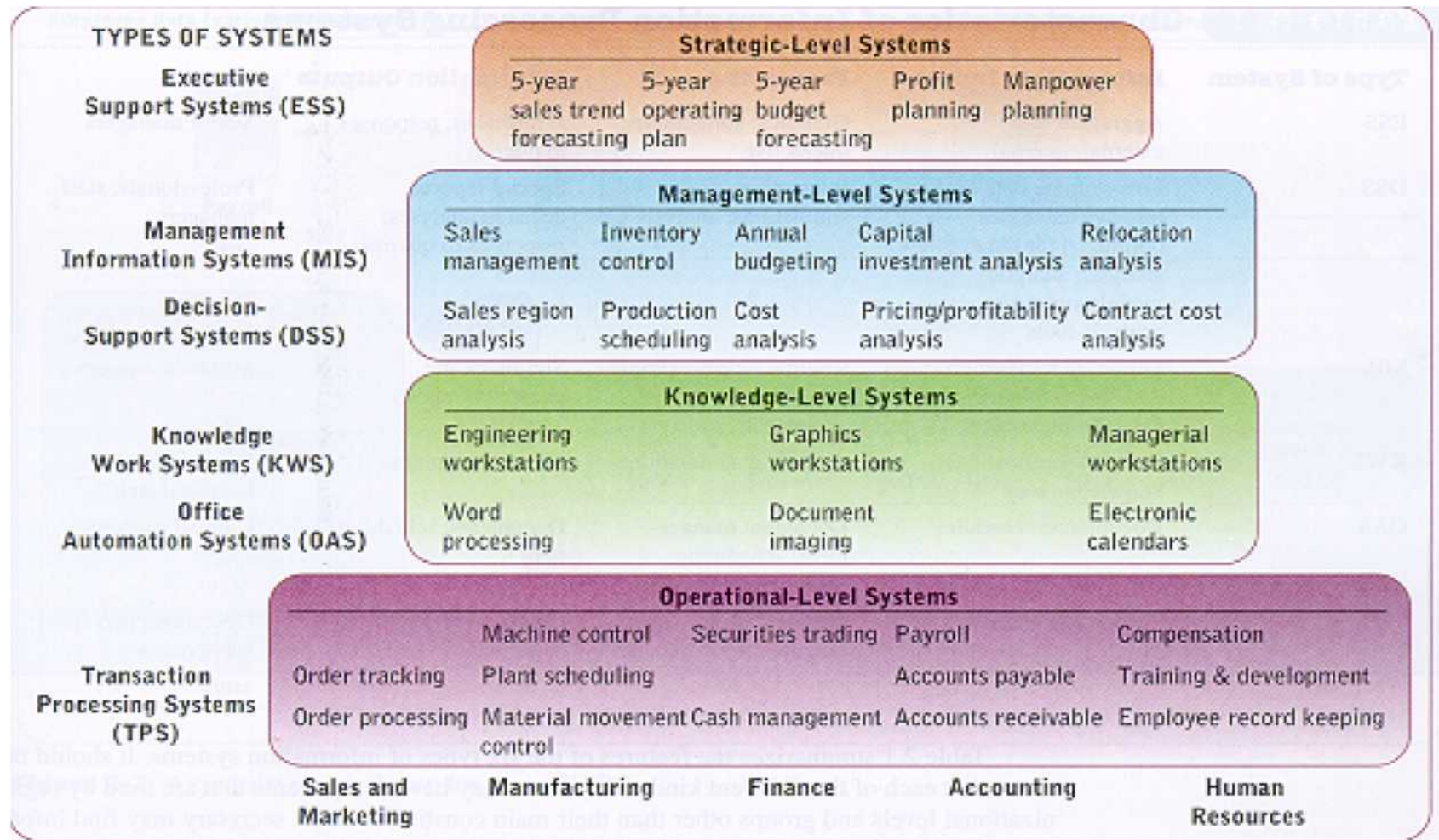
Sistema de Informação:

Fornece um meio para reunir informação de uma dada organização, fornecendo procedimentos para registo, processamento e disponibilização dessa informação, com o objectivo de auxiliar a organização nas suas actividades.

- Não é necessariamente um sistema software;
- Existem vários tipos de sistemas de informação...



Uma taxonomia para Sistemas de Informação





Uma taxonomia para Sistemas de Informação

Sistemas de Processamento de Transacções

- Recolhem e armazenam informação acerca de transacções (eventos de interesse);
- Controlam alguns aspectos relativos às transacções.

Sistemas de Automação de Escritórios

- Fornecem meios para o processamento de dados ao nível do indivíduo;
- Incluem o processamento de texto, folhas de cálculo, etc.

Sistemas de Trabalhos Baseados em Conhecimento

- Auxiliam na criação e integração de novos conhecimentos na organização.



Uma taxonomia para Sistemas de Informação

Sistemas de Suporte à Decisão

- Auxiliam a tomada de decisões fornecendo informação, modelos ou ferramentas de análise.

Sistemas de Informação para a Gestão

- Analisam a informação produzida pelos Sistemas de Informação Transaccionais;
- Convertem informação sobre transacções em informação de controlo de desempenho e gestão da organização.

Sistemas de Informação para Executivos

- Evolução dos Sistemas de Informação para a Gestão vocacionada para executivos;
- Permitem análise da informação de forma simples e interactiva e a diferentes níveis de detalhe.



O que é um bom Sistema (de Informação)?

Aquele que satisfaz as necessidades dos seus utilizadores:

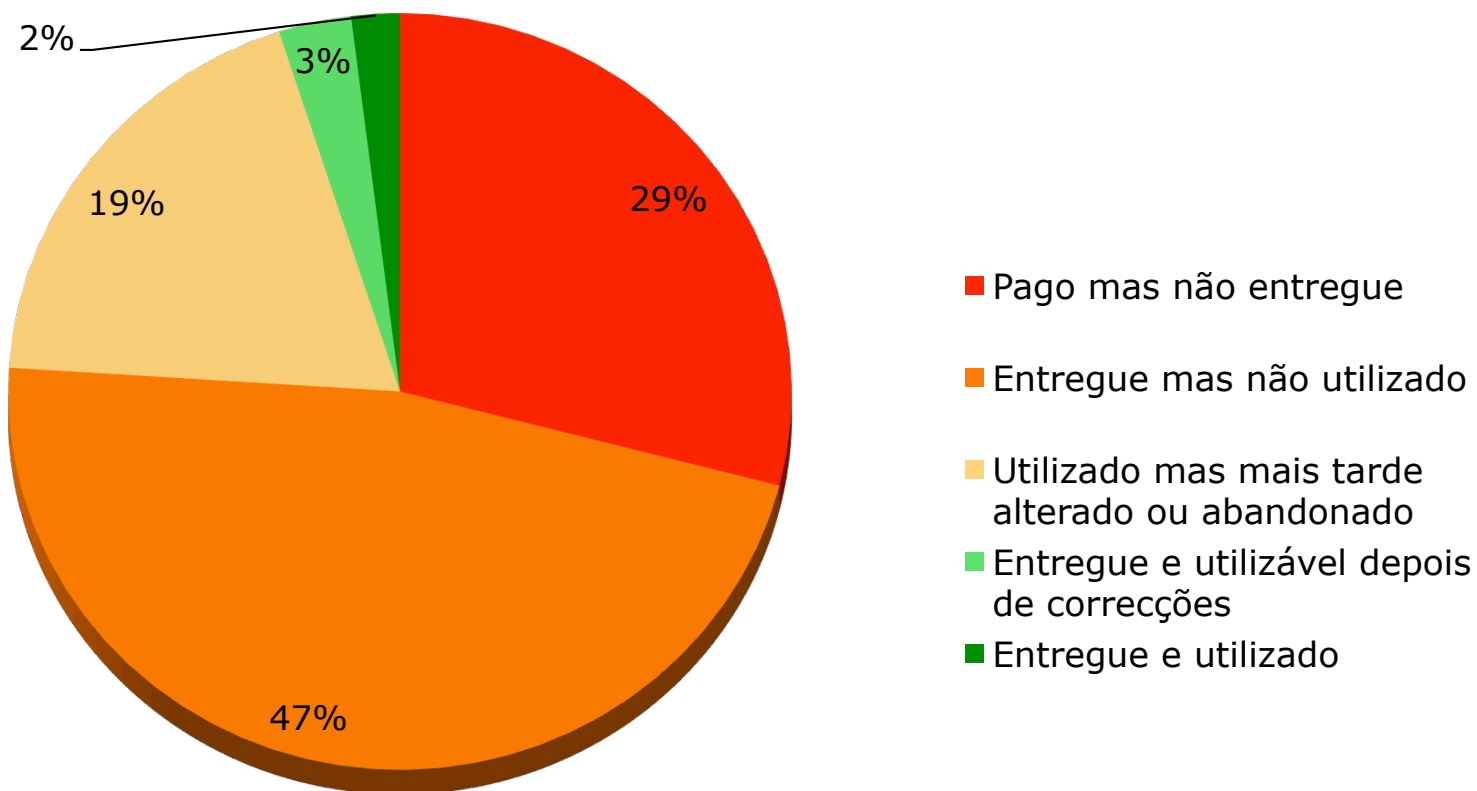
- **útil e utilizável** – bom software facilita a vida dos utilizadores - deve responder às necessidades reais dos utilizadores (usabilidade!);
- **confiável** – sem *bugs*!
- **flexível** – as necessidades dos utilizadores mudam, os *bugs* têm que ser corrigidos
X bug do ano 2k veio mostrar a falta de flexibilidade de muitos sistemas;
- **acessível** (financeiramente) – quer na compra quer na manutenção - fácil e rápido de desenvolver;
- **disponível** – se não está disponível nada mais interessa! - está disponível a tempo e horas? está disponível na plataforma tecnológica pretendida?

Como vai o desenvolvimento de software?



Estatísticas sobre desenvolvimento de Software

Sobre o produto pago...



- Mais de 75% do software pago não chegou a ser utilizado!
- Apenas 5% do software pago foi utilizado continuamente (deste, 3% necessitou de correcções).



Estatísticas sobre desenvolvimento de Software

Inquérito realizado em 1994 a 352 companhias (Standish Group):

- 56% de todos os *bugs* pode ser atribuídos a erros cometidos durante a fase de análise (i.e., não se esteve a construir o sistema certo!)
- 31% de todos os processos de desenvolvimento de software são cancelados antes de estarem terminados.
- 53% dos projectos custam 189% do estimado.
- 9% dos projectos de grandes companhias respeitam os prazos e o orçamento.
- 16% dos projectos de pequenas companhias respeitam os prazos e o orçamento.

Mais alguns dados sobre grandes projectos (>50,000 linhas de código):

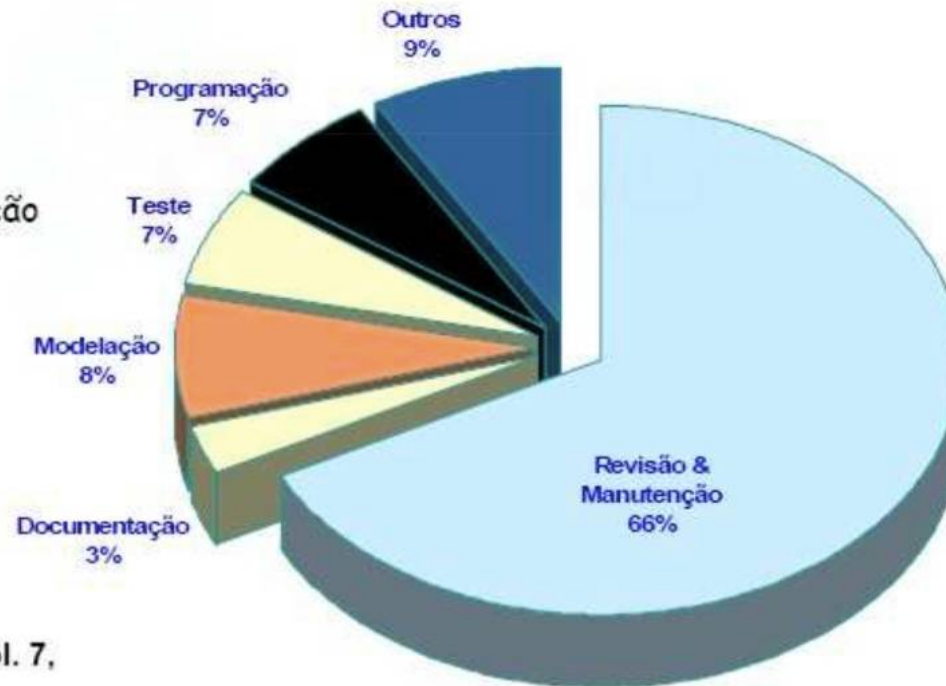
- **produtividade média está abaixo das 10 linhas de código por dia;**
- **em média, encontram-se 60 erros por cada 10,000 linhas de código;**
- **custo de manter o software ultrapassa o dobro do custo de desenvolvimento.**



Estatísticas sobre desenvolvimento de Software

os custos envolvidos num projecto de software

- Modelação
- Programação
- Teste
- Documentação
- Revisão e Manutenção
- Outros



Source: DP Budget, Vol. 7,
No. 12, Dec. 1998



Estatísticas sobre desenvolvimento de Software

Exemplos

Alguns exemplos de sistemas com problemas atribuíveis ao software:

- **Sonda Mariner I, Julho de 1962**

Deveria ter voado até Vénus. Apenas quatro minutos após o lançamento despenhou-se no mar. Descobriu-se depois que um operador de negação lógica tinha sido omitido por acidente no código do programa responsável por controlar os foguetes...

- **Therac-25, finais dos anos 80**

Máquina de Raios-X totalmente controlado por software. Diversos problemas provocaram a administração de radiação excessiva a vários doentes.

- **Aeroporto Internacional de Denver, início dos anos 90**

Sistema de tratamento de bagagem envolvendo mais de 300 computadores. O projecto excedeu os prazos de tal forma que obrigou ao adiamento da abertura do aeroporto (16 meses). Foi necessário mais 50% do orçamento inicial para o pôr a funcionar.

- **Colocação de professores, inícios sec. XXI**

Empresa inicialmente contratada não conseguiu desenvolver uma versão funcional do sistema e foi necessário contratar uma nova empresa.



Estatísticas sobre desenvolvimento de Software

Em conclusão:

Problemas com o desenvolvimento de software:

- Atrasos na entrega.
- Incumprimento dos orçamentos.
- Falha na identificação e satisfação das necessidades dos clientes.
- Produtos entregues com falhas.



Sejam de que tipo forem, os SI:

- existem para auxiliar a organização
 - ➡ mais um meio a juntar a (tantos) outros;
- devem ser concebidos em função das necessidades da organização
 - ➡ vão ser utilizados pela organização, não por quem os concebeu.

É necessário perceber a organização para conceber um bom SI.

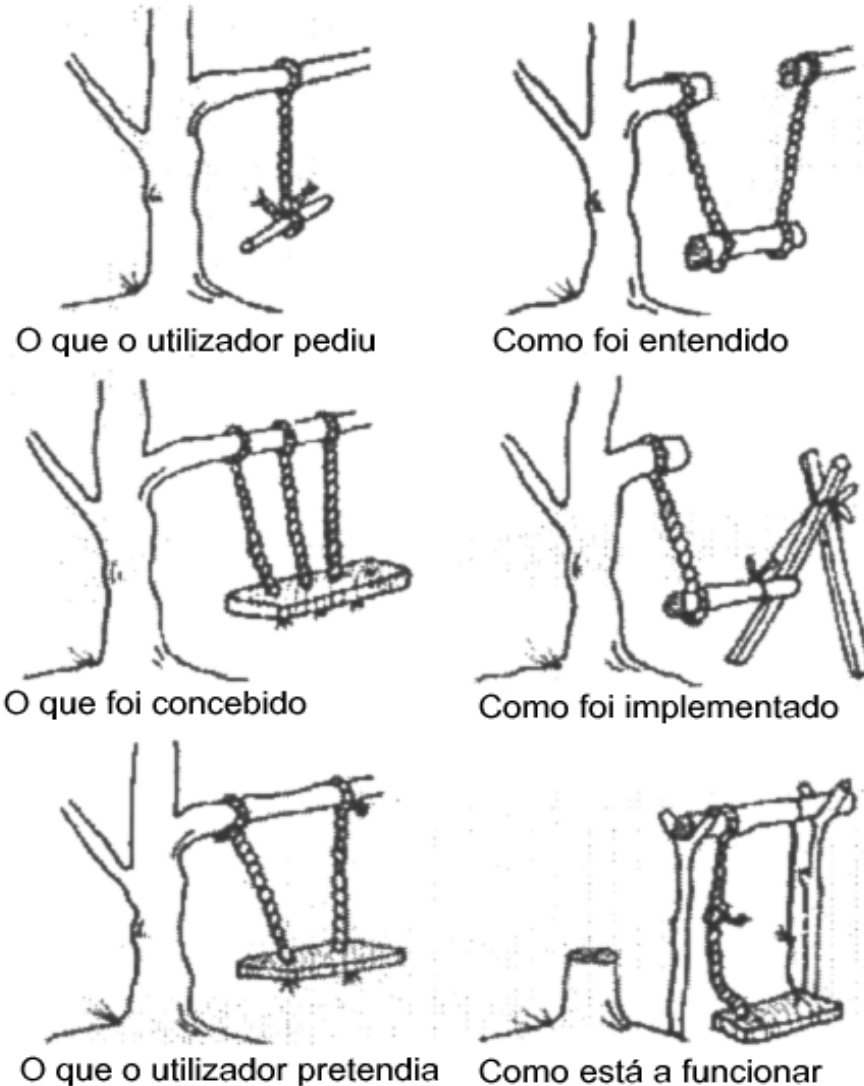
- Quais as actividades da organização a suportar.
- Qual a informação relevante que flui na organização.
- Quais as tarefas das pessoas da organização.

Entender o problema antes de desenvolver a solução!



Desenvolvimento de Software (Riscos)

Desenvolver um bom sistema não é tarefa trivial

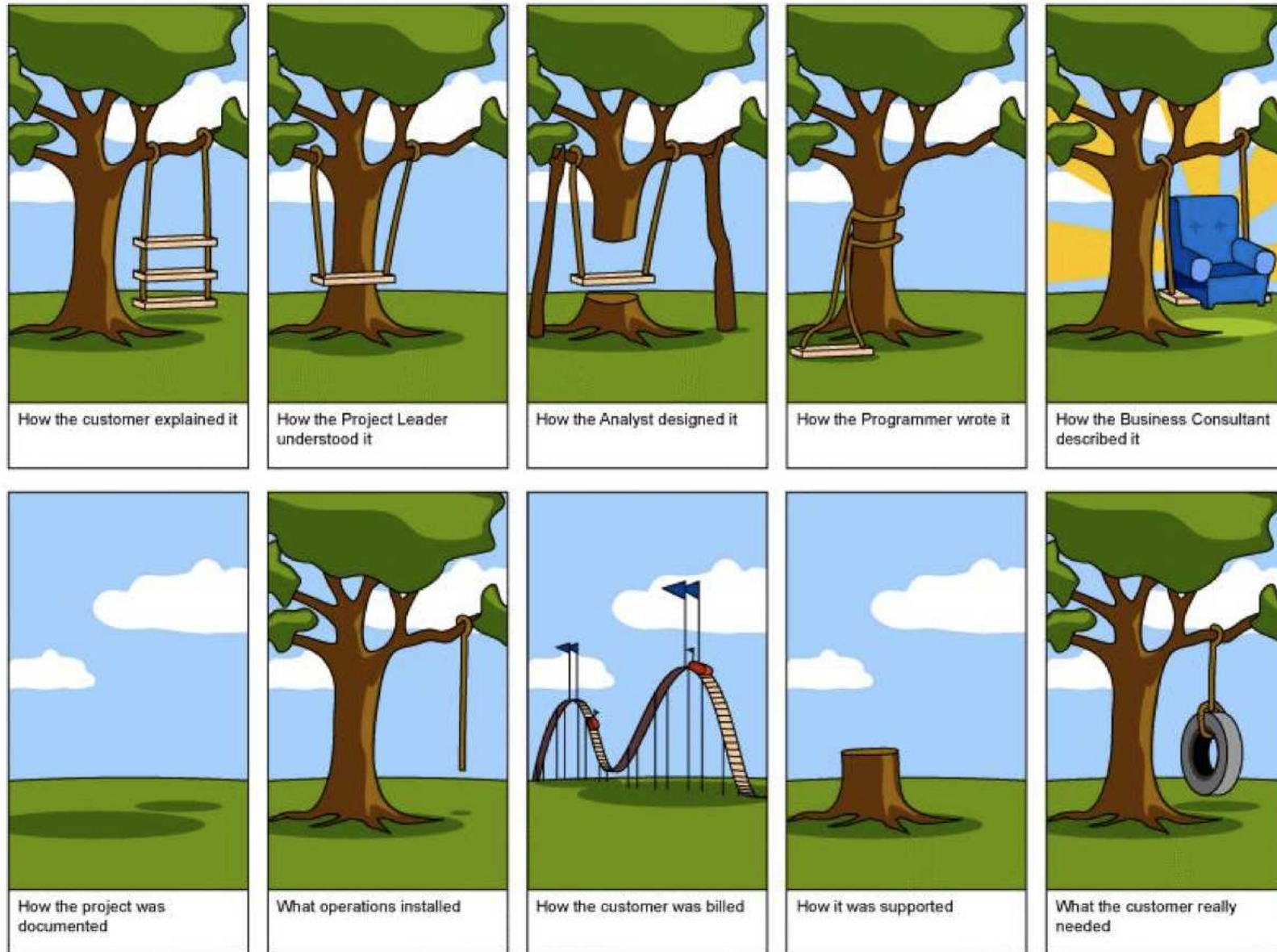


- Riscos associados aos requisitos.
- Riscos tecnológicos.
- Riscos de competência.
- Riscos políticos.



Desenvolvimento de Software (Riscos)

Desenvolver um bom sistema não é tarefa trivial (versão moderna!)





Desenvolvimento de Software (Riscos) (II)

Riscos associados aos requisitos

É necessário comunicar com os peritos da organização para:

- compreender que tarefas o sistema deve suportar;
- compreender como o sistema *encaixa* nas actividades da organização.

Um dos maiores desafios é construir o sistema *certo*.

Riscos Tecnológicos

- Qual a tecnologia mais apropriada?
- Como controlar a complexidade?

É necessário validar as soluções tecnológicas o mais cedo possível.

Riscos de Competência

- É necessário saber-se o que se está a fazer (obviamente?).
- Exemplo de OO: fácil de aprender/difícil de dominar.

Riscos Políticos

- Por muito bom que seja o SI só terá sucesso se tiver o apoio das *pessoas certas*.



Respostas I - Tecnologia

- Primeiras abordagens à *Crise do Software* preocupavam-se mais com a produtividade do que com a qualidade.
- As tecnologias de programação têm vindo a tornar-se cada vez mais sofisticadas (tanto aos níveis dos paradigmas como das ferramentas).

Paradigmas de Programação

O modo como estruturamos o código tem vindo a evoluir como resposta ao aumento da complexidade do software:

- Programação estruturada (70's) - Estruturar o código para controlar complexidade.
- Programação modular - Estruturar o código, mas também os dados.
- Programação orientada aos objectos (80's) - Aumenta o poder expressivo na estruturação de dados/código.

Ferramentas

Ambientes de Desenvolvimento Integrado (IDEs) cada vez mais sofisticados procuram facilitar a tarefa de programação (e também de análise).



Respostas II - Métodos de Desenvolvimento

- A tecnologia só por si não resolve os problemas (e estes têm vindo a aumentar!)
- A tecnologia é apenas uma ferramenta, é necessário saber como utilizá-la
- Hoje em dia a *Crise do Software* tem muito a ver com a qualidade do produto final.
- São necessários métodos de desenvolvimento que garantam produtividade e qualidade.

método

do Lat. methodu < Gr. métodos, caminho para chegar a um fim

s. m., processo racional que se segue para chegar a um fim; modo ordenado de proceder; processo; ordem; conjunto de procedimentos técnicos e científicos;

(<http://www.priberam.pt/dlpo/>)

O desenvolvimento de software não pode ser encarado como *arte*,
mas como Engenharia.

Necessitamos de métodos e ferramentas apropriados.

Em DSS apresenta-se uma proposta, existem outras!



Aulas Teóricas - Programa

- O Processo de Desenvolvimento de Software — diferentes abordagens.
- Modelação de Sistemas Software em UML:
 - Visão geral — os diferentes níveis de modelação
 - Modelação comportamental:
 - ✓ Diagramas de *Use Case*;
 - ✓ Diagramas de Interacção (Sequência/Colaboração);
 - ✓ Diagramas de Estado (*Statecharts*);
 - ✓ Diagramas de Actividade;
 - Modelação estrutural:
 - ✓ Diagramas de Classe (revisão de conceitos OO);
 - ✓ Diagramas de *Package*; Diagramas de Instalação (*Deployment*).
- Padrões de software
- Mapeamento de objectos no modelo relacional



Práticas Laboratoriais - Programa

- Programação de interfaces com o utilizador em Java;
 - breve apresentação do SWING;
- Apresentação da Ferramenta de Modelação:
 - modelação UML;
 - geração de código.
- Estudos de caso:
 - pequenos exemplos para apreensão dos conceitos;
 - realização do trabalho.



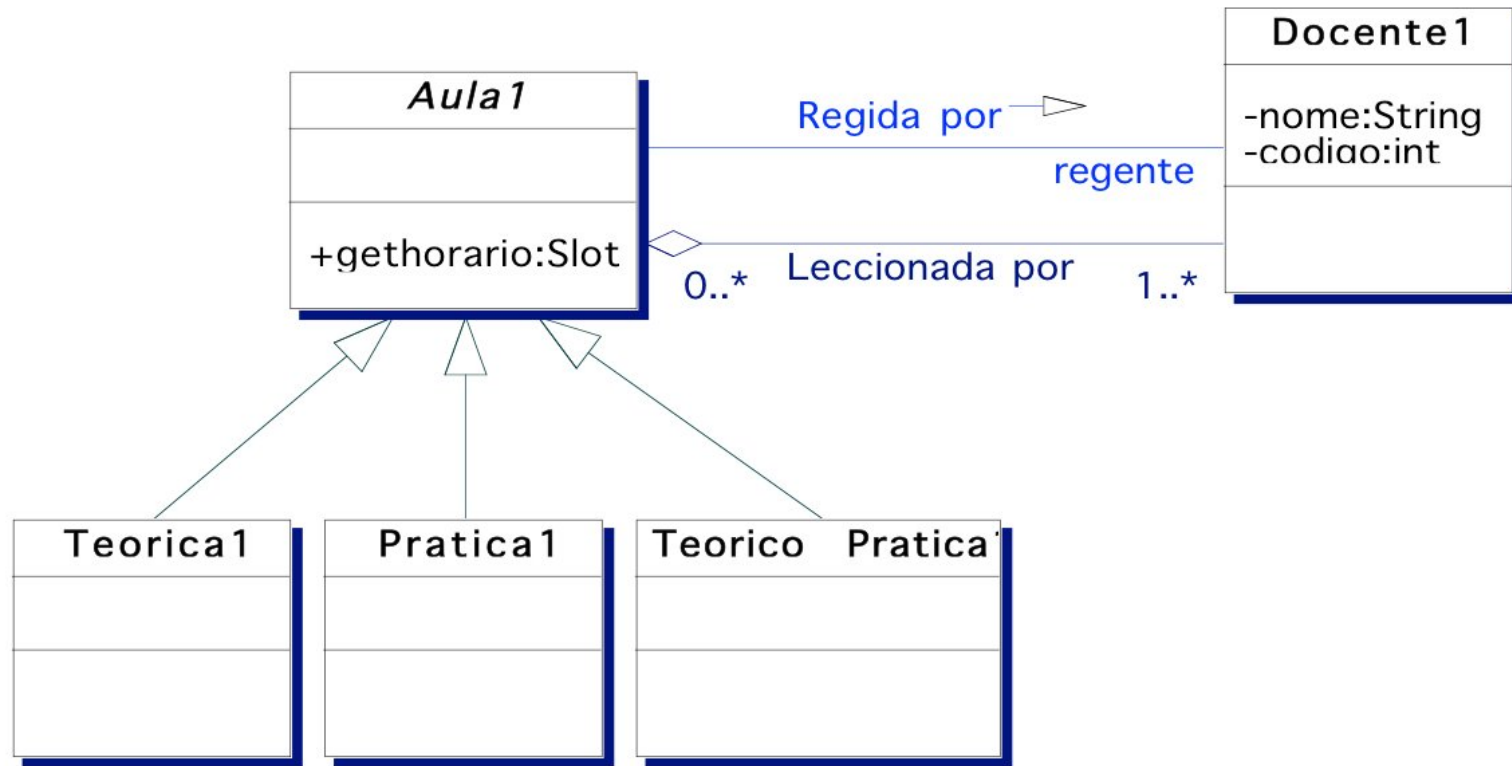
UML - *Unified Modelling Language*

(Booch, Jacobson & Rumbaugh)

- A UML é uma família de linguagens gráficas de modelação
 - *inclui (13) diagramas para as diferentes fases de desenvolvimento*
- A UML foi pensada para o desenvolvimento orientado aos objectos, mas é independente das linguagens de programação a utilizar
 - *permite explorar o paradigma OO*
- A UML possibilita trabalhar a diferentes níveis de abstracção
 - *facilita comunicação e análise*
- A UML NÃO É um processo de desenvolvimento de software, mas pode ser utilizado com diferentes processos
- A UML é uma *norma* mantida pelo OMG (Object Management Group)
 - pode também dizer-se que a norma *de facto* da indústria de software
- A UML é suportada por ferramentas (50+)
 - *Rational Rose (IBM), Together (Borland), Visual Paradigm, Poseidon, etc.*



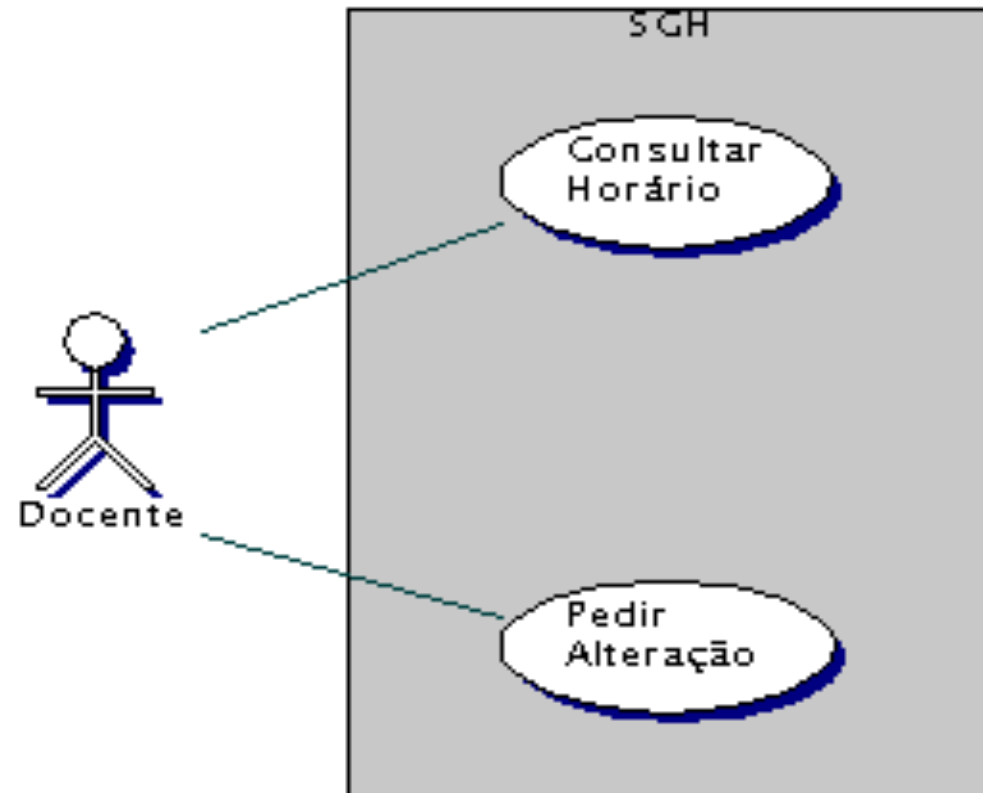
Diagrama de Classe





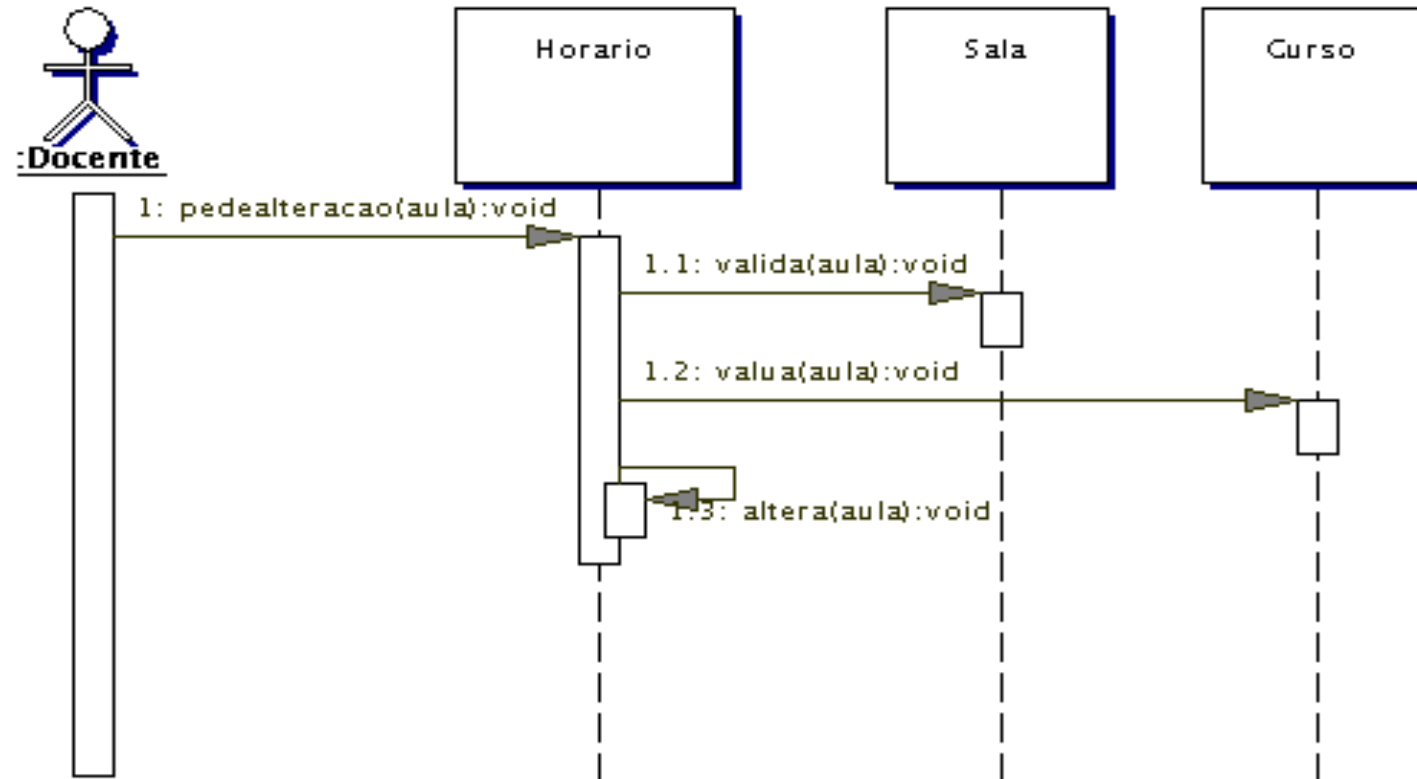
Alguns Exemplos de Diagramas UML

UML - Diagrama de Use Case



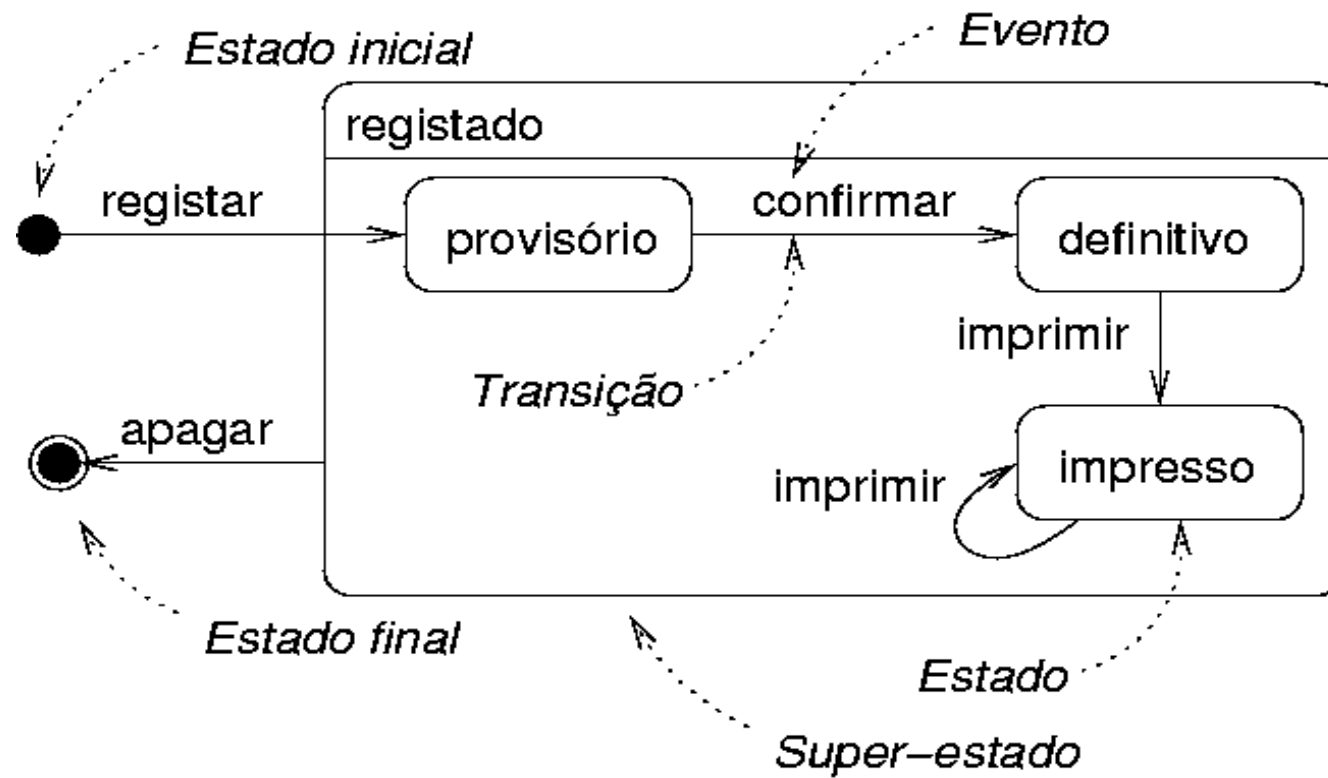


UML - Diagrama de Interacção



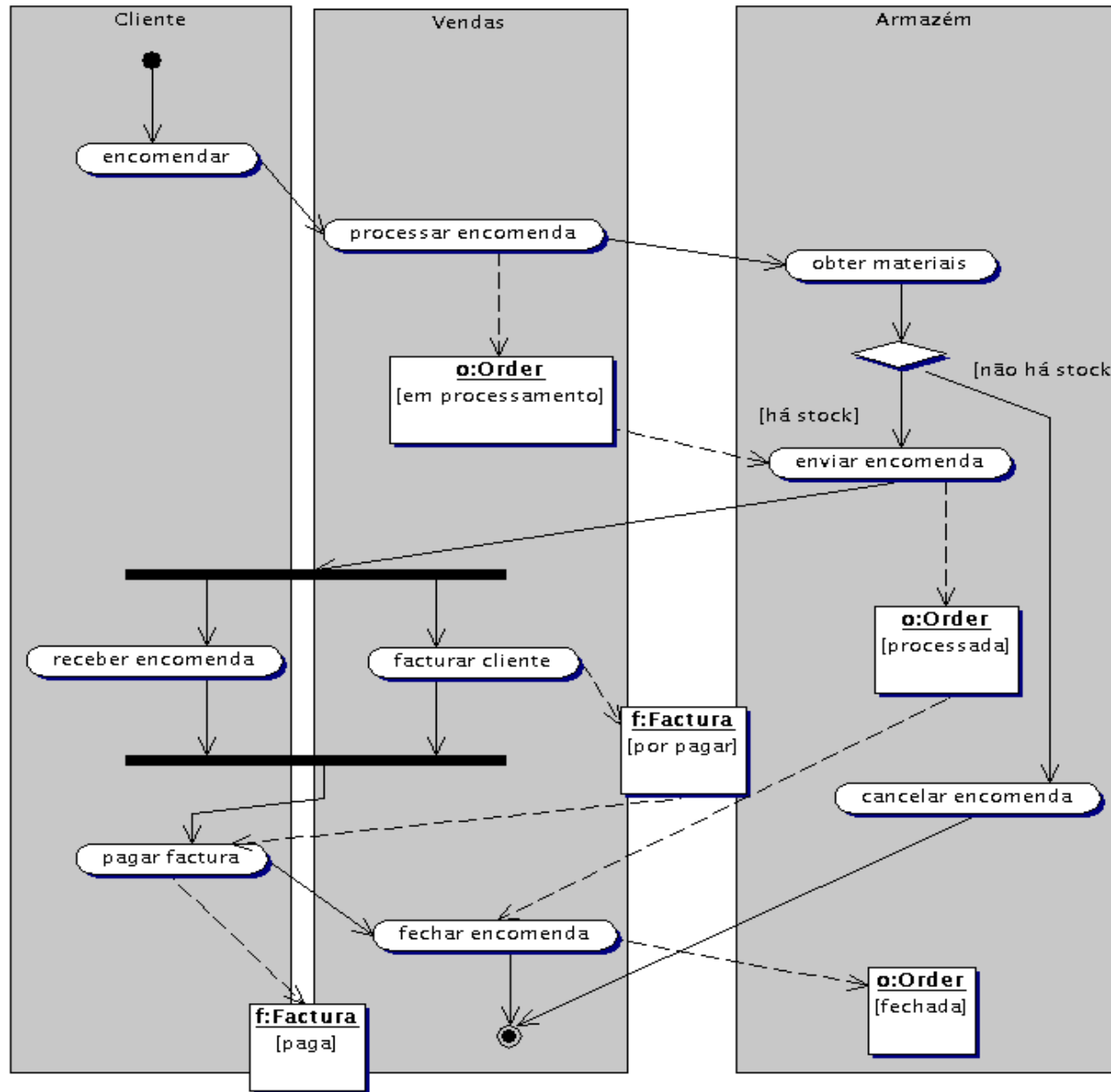


UML - Diagrama de Estado (*Statechart*)



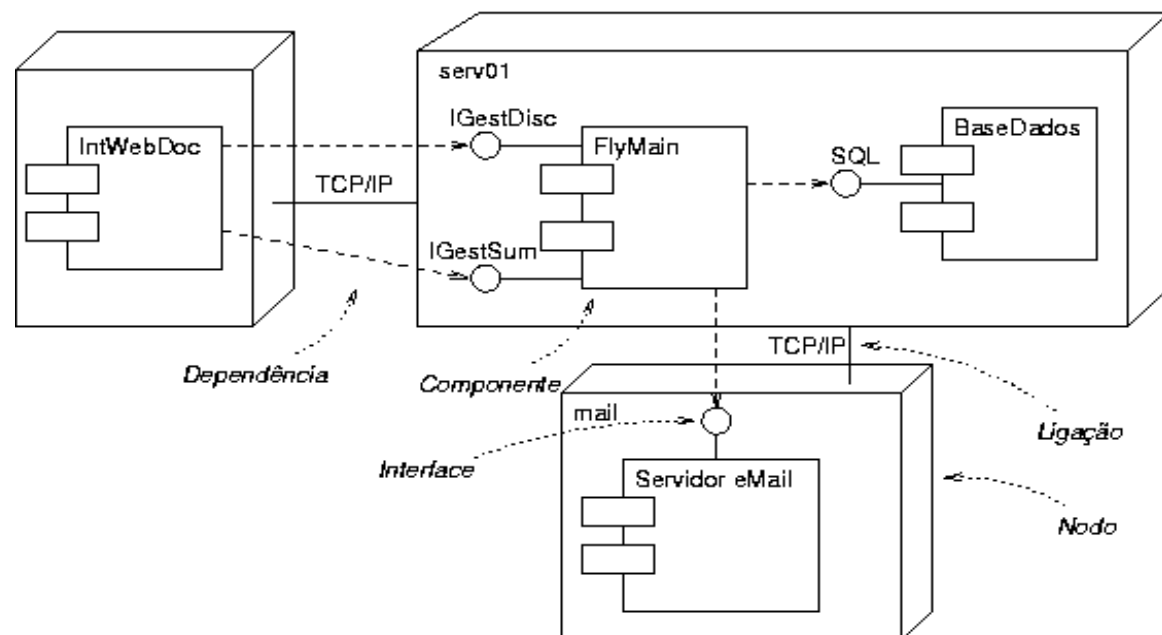


UML - Diagrama de Actividade





UML - Diagrama de Instalação





Bibliografia

- J. Arlow, I. Neustadt. *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd edition)*. Addison-Wesley Professional, 2005.
- D. Pilone, N. Pitman. *UML 2.0 in a Nutshell (2nd edition)*. O'Reilly Media, 2005.
- Martin Fowler. *UML Distilled (third edition)*. Addison-Wesley, 2004.
(bom livro!)
- Scott W. Wembler, The Elements of UML 2.0 Style, Cambridge University Press, 2005.
- R. Pressman. Engenharia de Software, 6th. Ed., McGraw Hill, 2005.

Em português:

- M. Nunes & H. O'Neill. *Fundamental do UML*, 5^a edição. FCA. 2007.
- Apontamentos de suporte às aulas teóricas
(irão sendo disponibilizados ao longo do semestre).



Sumário

- Apresentação da UC
 - Avaliação
 - Trabalho
- Motivação
 - Alguns dados sobre desenvolvimento de software
 - Necessidade de um método de desenvolvimento
 - Necessidade de uma linguagem de modelação
- UML
- Bibliografia



A fazer...

- Inscrição nos turnos PL (de hoje até segunda)
 - Senha de pré-inscrição: dss201112
 - Turnos disponíveis a partir das 14h
 - PL2 - 2ª 14h-16h
 - PL4 - 2ª 16h-18h
 - PL1 - 6ª 10h-12h
 - PL5 - 6ª 14h-16h
 - PL3 - 6ª 16h-18h
- Organizarem-se em grupos de 3 a 5 elementos
 - Inscrição será feita no BB dentro de uma semana.