



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Desenvolvimento de Sistemas de Software

LEI/MiEI - 3º ano / 1º semestre

Apresentação 2022/2023

José Creissac Campos
jose.campos@di.uminho.pt

António Nestor Ribeiro
anr@di.uminho.pt

<http://www.di.uminho.pt>



Desenvolvimento de Sistemas Software 2019/20

- **Equipa Docente**



José C. Campos (T/PL)

jose.campos@di.uminho.pt



António N. Ribeiro (PL)

anr@di.uminho.pt



Óscar R. Ribeiro (PL)

d2680@di.uminho.pt

- **Canais de comunicação**

- Aulas teóricas (canal principal)
- Aulas PL (turno)
- Blackboard



Dívida técnica (*technical debt*)...



Version 2 announcement:

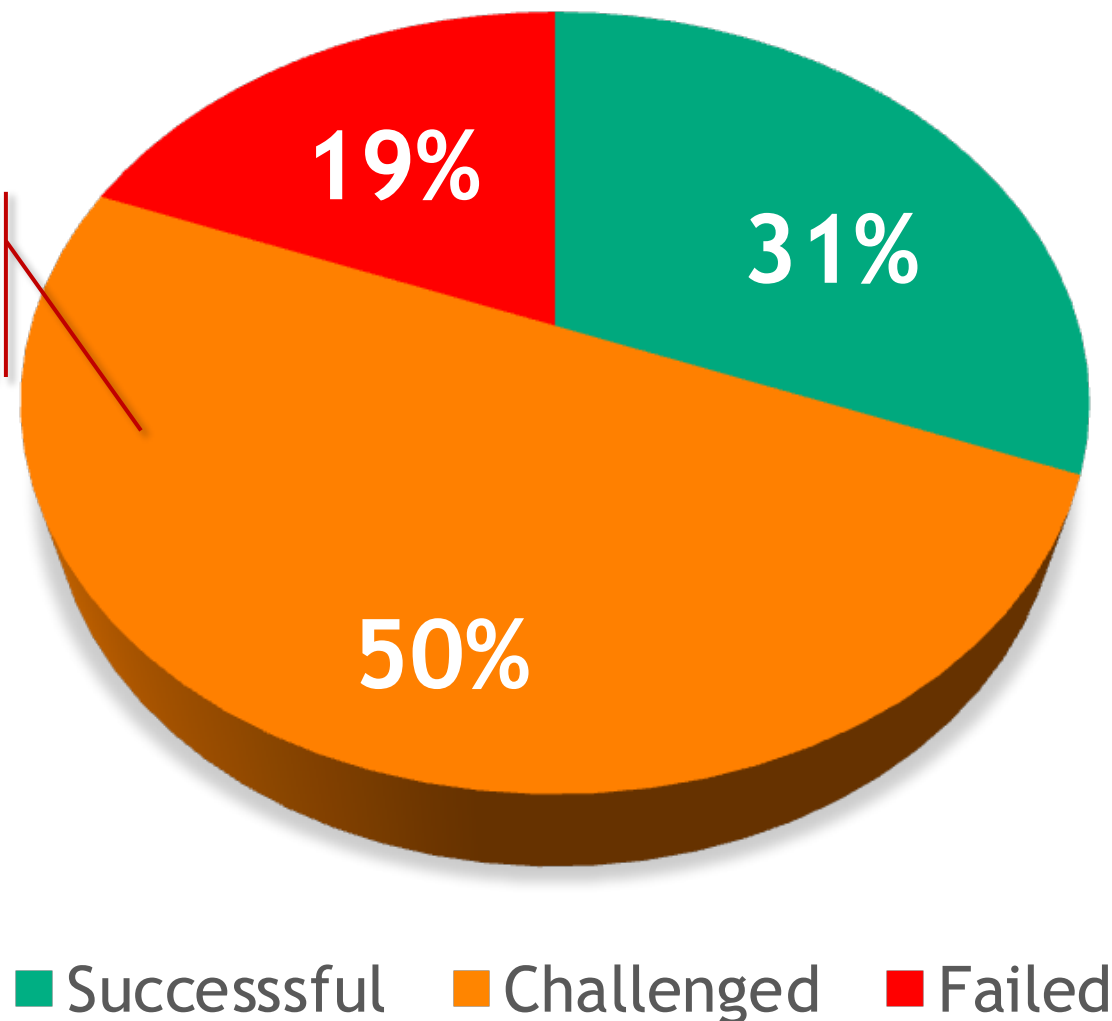
Among Us 1 (...) started as a tiny local-multiplayer-only game and **has grown and grown and grown**. (...) Because of this, it's **extremely hard to add more things** (...) **because the game is so fragile**. Fixing [it would be] (...) **harder than just making a new game**.

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**.



Como vai o desenvolvimento de Software?

Standish Group CHAOS Report 2020*





Standish Group CHAOS Report 2020

- Top 5 Factors in **Successful** IT Projects

1. 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
2. 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



Aulas Teóricas - Programa

- O ciclo de vida do Desenvolvimento de Software
- **Desenvolvimento de Software baseado em modelos:**
análise e concepção orientada aos objectos, com UML
 - Modelação de Domínio
 - Análise e modelação de requisitos
 - Concepção da solução
 - Refinamento e implementação da solução - mapeamento de objectos no modelo relacional
 - Modelação de comportamento
- Pré-requisito (oficioso): POO!

☐ *Semana 1*

☐ *Semana 2*

☐ *Semanas 3-4*

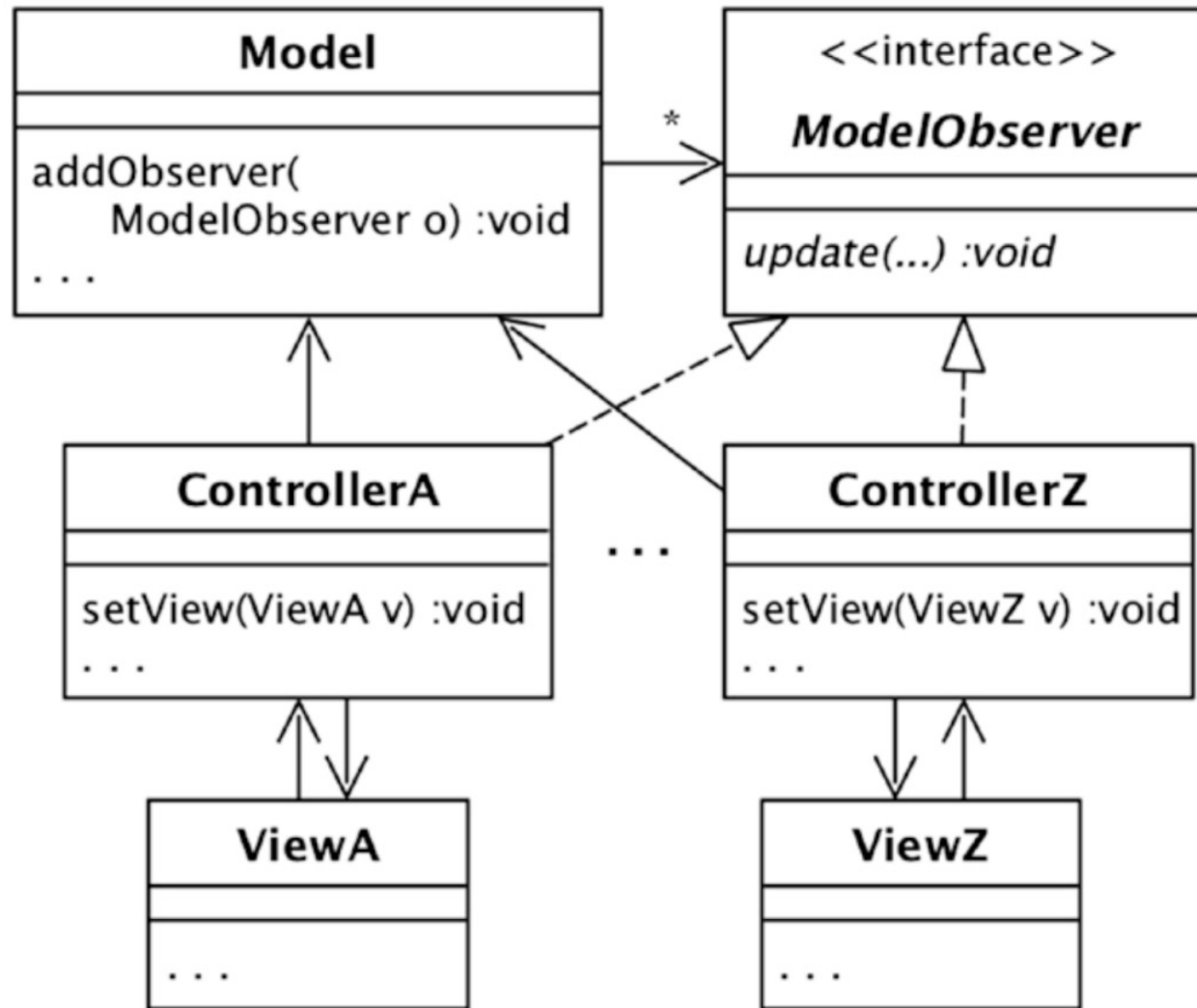
☐ *Semanas 5-8*

☐ *Semanas 10-12*

☐ *Semanas 9-13*



UML - Diagramas de Classe

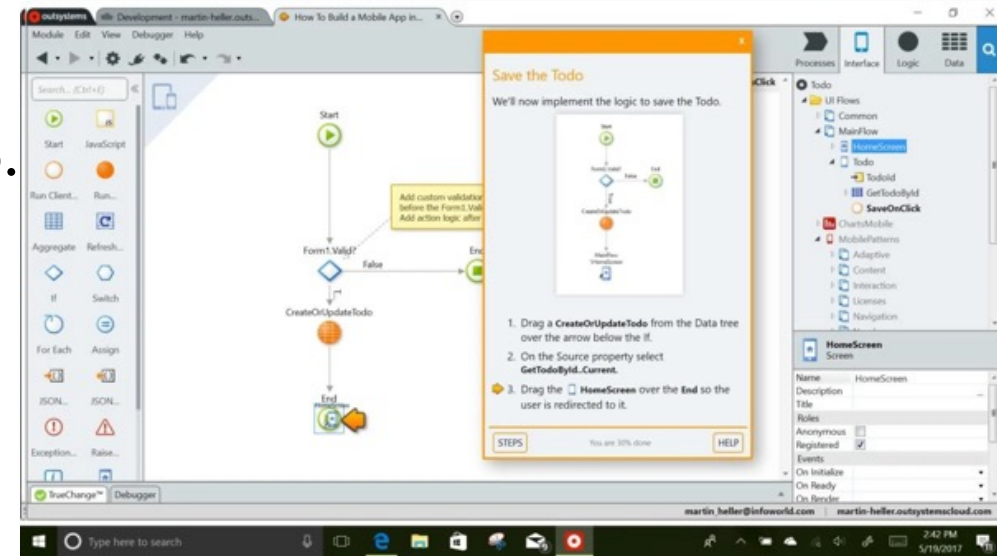


(*) retirado de Java Program Design, E. Sciore, 2019



Low-code development

- Desenvolvimento com recurso a linguagens visuais, minimizando a necessidade de escrever código.
- The 9 Essential Principles
 1. **Model-driven development**
 2. **Collaboration**
 3. **Agility**
 4. **The cloud**
 5. **Openness**
 6. **Multi-User Development**
 7. **Experimentation & Innovation**
 8. **Governance & Control**
 9. **Community**



 outsystems

 mendix

 PowerApps

 salesforce

 servicenow



Práticas Laboratoriais - Programa

- Estudos de caso:
 - pequenos exemplos para aprendizagem dos conceitos
 - realização do projecto
- Apresentação da Ferramenta de Modelação Visual Paradigm:
 - modelação em UML (Unified Modelling Language)
 - geração de código





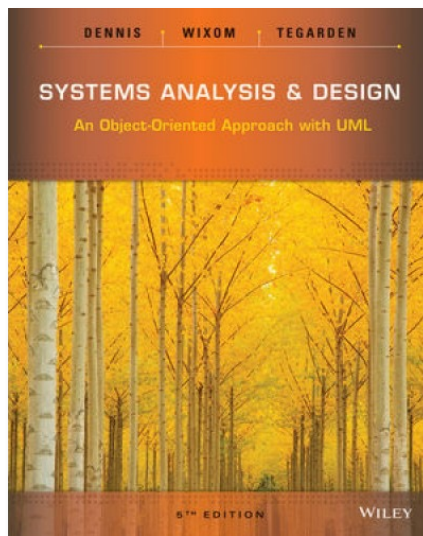
Objectivos de aprendizagem

No final da UC os alunos deverão ser capazes de:

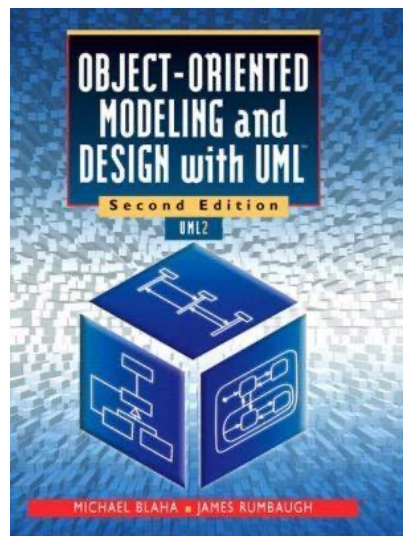
1. Caracterizar as fases típicas de um processo de desenvolvimento de software
2. Interpretar os diferentes tipos de diagramas da UML
3. Avaliar qual o diagrama mais adequado em função das diferentes necessidades de modelação
4. Avaliar criticamente modelos (de requisitos / estruturais / comportamentais)
5. Conceber sistemas de software utilizando modelos
6. Implementar sistemas de software a partir de modelos



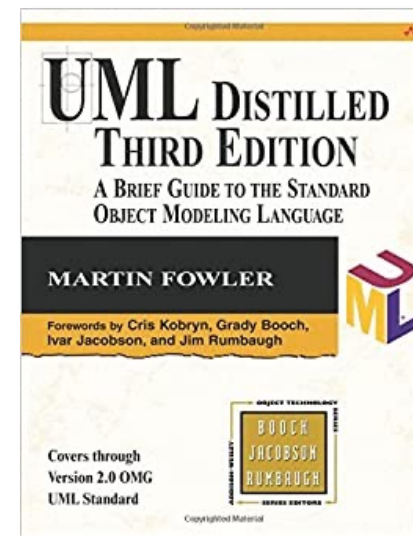
Bibliografia



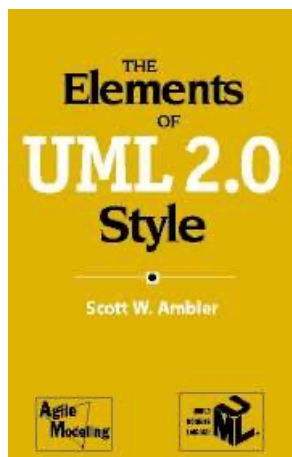
Dennis, Wixom & Tegarden. *System Analysis & Design* (5th ed). Wiley, 2015.



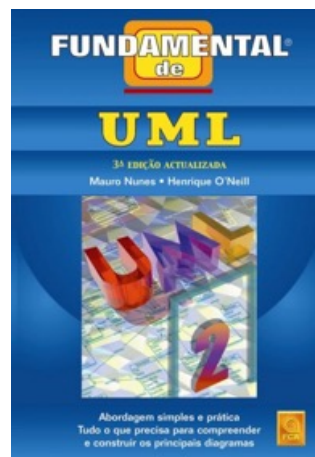
Blaha & Rumbaugh. *Object-Oriented Modeling and Design with UML* (2nd ed). Prentice Hall, 2005.



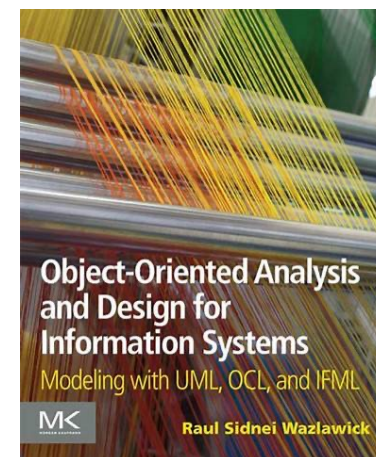
Fowler. *UML Distilled* (3rd ed). Addison-Wesley, 2004.



Wempler, *The Elements of UML 2.0 Style*. Cambridge University Press, 2005.



Nunes & O'Neill. *Fundamental de UML* (3ª ed). FCA, 2007.



Wazlawick. *Análise e Design Orientados a Objetos para Sistemas de Informação* (3ª ed). Grupo GEN, 2014.



Funcionamento

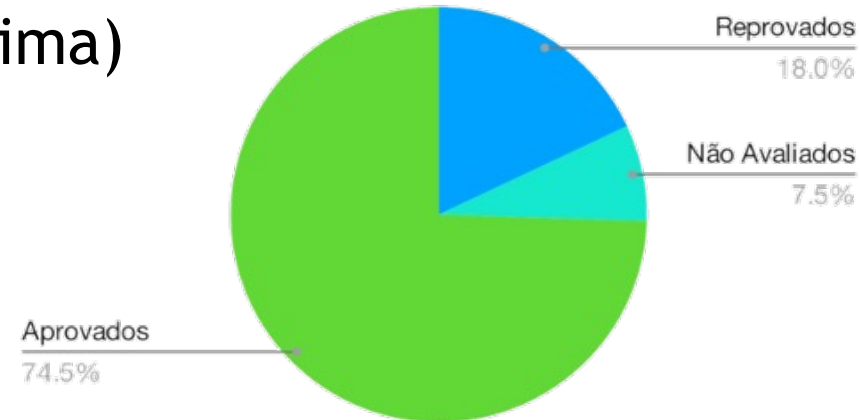
- Aulas T (1h+1h)
 - Apresentação e discussão da matéria
 - Interação no Blackboard e VoxVote
- Aulas PL (2h)
 - Realização de exercícios e **esclarecimento de dúvidas** (aplicação da matéria leccionada nas aulas T)
 - Fichas práticas disponibilizadas com antecedência
- Horário de atendimento
 - Sextas 15:30-17:30
 - Horário flexível (**marcar com antecedência**)





Avaliação

- Trabalho Prático (≥ 10.0)
 - projecto de análise e desenvolvimento de software, em grupo
- Teste/Exame (≥ 8.0)
 - prova individual escrita
- Avaliação Contínua (s/ nota mínima)
 - participação nas aulas
- Classificação Final (≥ 10.0)
 - .45 Exame + .45 Trabalho + .10 Avaliação continua

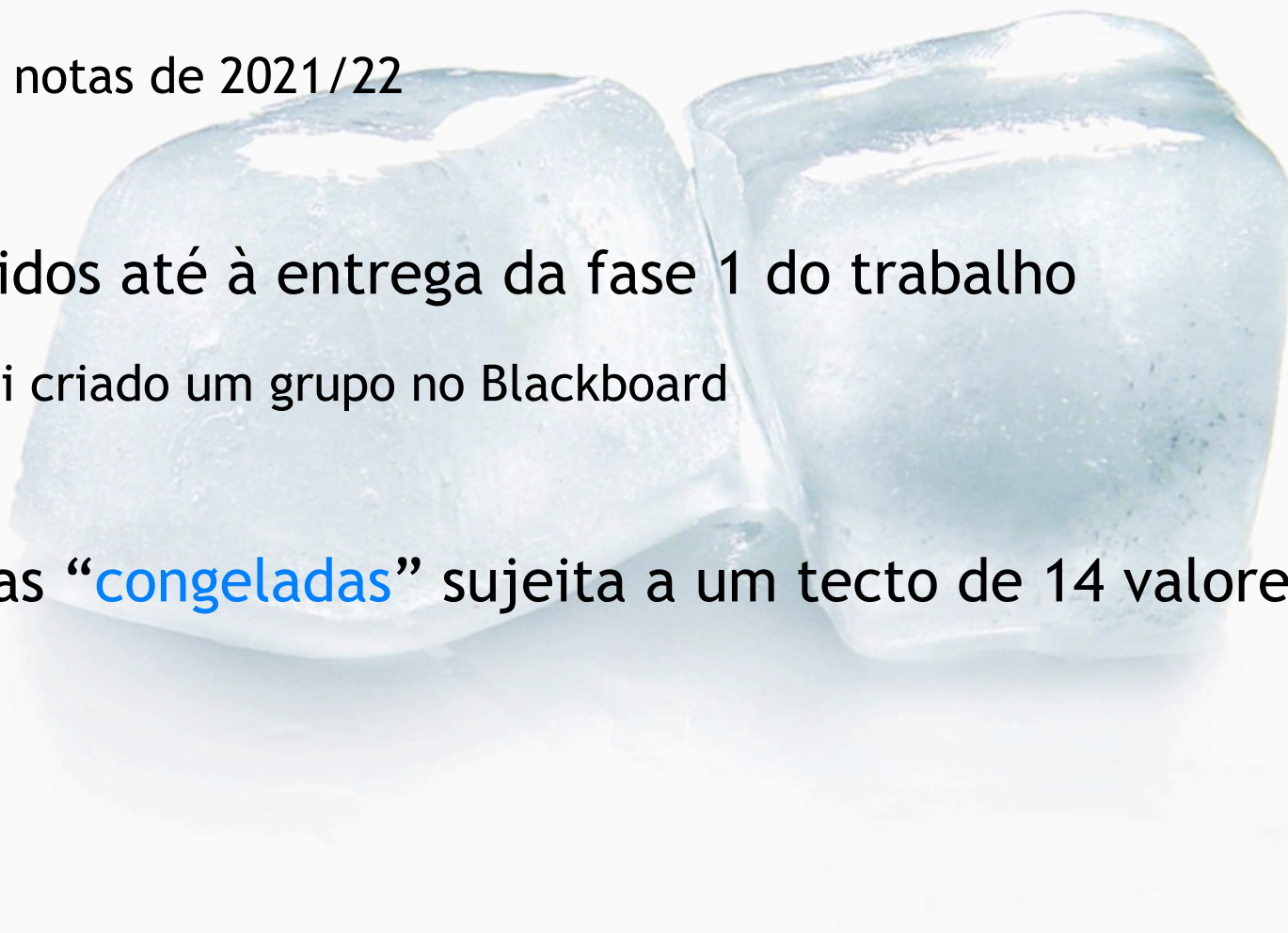


IMPORTANTE:
Mínimos são condição necessária,
mas não suficiente, para garantir
aprovação à UC.



Congelamentos de nota prática

- É possível reutilizar a nota do trabalho do ano passado
 - Só notas de 2021/22
- Pedidos até à entrega da fase 1 do trabalho
 - Foi criado um grupo no Blackboard
- Notas “congeladas” sujeita a um tecto de 14 valores





Teste / Exame

- Individual
- Com consulta de apontamentos
 - Podem levar **duas folhas** com a informação que considerarem relevante
- No exame aplicam-se as mesma regras de cálculo da nota que para o teste
- Datas
 - Teste: 6 de Janeiro
 - Exame de recurso: 30 de Janeiro



Trabalho Prático

- Grupos de 3-5 elementos
 - Discussão de soluções alternativas
- A realizar **durante** o semestre - 5 momentos relevantes:
 - 1) Apresentação do enunciado - aula T de **26 de Setembro**
 - 2) Entrega intermédia 1 - até às 24h00 de **20 de Outubro** (3.5 semanas)
 - Análise de requisitos - 1ª parte do relatório
 - 3) Entrega intermédia 2 - até às 24h00 de **17 de Novembro** (5 semanas)
 - Concepção da solução - 2ª parte do relatório
 - 4) Entrega final - até às 24h00 de **7 de Janeiro** (4 semanas)
 - 3ª parte do relatório + sistema desenvolvido
 - 5) Apresentação e discussão: **terceira semana de Janeiro** de 2023
 - Todos os elementos do grupo presentes; avaliação será individualizada



Sobre a quantidade trabalho...

- ECTS - European Credit Transfer System
 - 1 ECTS = 28h de trabalho
 - 5 ECTS = 140h
 - 15 semanas -> $140/15 = 9h20m/semana$ (para DSS!)
 - Aulas = 4h/semana (na verdade são só 13 semanas!)
 - Trabalho autónomo = $5h20m/semana = \sim 1h/dia$ útil
 - 1h por dia = 65h em 13 semanas → **~50h para o Trabalho Prático!**
- Calendário

2023-01-02	2023-01-08			SD CC						IA						DSS			DSS
2023-01-09	2023-01-15			SD						CC			CP			CP			LI4
2023-01-16	2023-01-22			DSS, IA, CC, SD				DSS, IA, CC, SD		DSS, IA, CC, SD			DSS, IA, CC, SD, CP			DSS, IA, CC, SD, CP			



Trabalho Prático - Avaliação por Pares

- Avaliação dos estudantes pelos seus colegas
- Objectivos
 - **Responsabilizar** o grupo pelo seu desempenho colectivo/de cada elemento
 - **Classificar** individualmente cada aluno de acordo com a sua contribuição e empenho, tal como precepcionada pelo grupo (incluindo o próprio aluno)
 - Avaliação como parte do processo de **aprendizagem** (*feedback*)
 - **Maior comprometimento e autonomia**



Peer Assessment allows students to make judgements about others work while reflecting on their own.



Avaliação por pares

- Realizada três vezes ao longo do semestre
 - 24-28 de Outubro após entrega 1 (peso: 20%)
 - 21-25 de Novembro após entrega 2 (peso: 40%)
 - 10-11 de Janeiro após entrega final (peso: 40%)
- Preenchimento online de uma grelha de avaliação multi-critério
 - Para cada critério há $p*n$ pontos a distribuir (sendo p o peso do critério e n o número de elementos do grupo - $\sum p/n=100$)
- CAIG - Coeficiente de Avaliação Intra-Grupo
 - O resultado da avaliação por pares
 - CAIG final - média ponderada dos CAIG de cada etapa



Avaliação por pares - Exemplo

Avaliação feita pelo Aluno 1

Critério	Aluno 1	Aluno 2	Aluno 3
Ideias	24	20	20
Realização	26	18	20
Empenho	24	24	20
Assiduidade	23	23	20
Relações	22	22	20
CAIG (parcial)	1.19	1.07	1.00

Avaliação feita pelo Aluno 2

Critério	Aluno 1	Aluno 2	Aluno 3	Total
Ideias	20	20	20	60
Realização	22	20	18	60
Empenho	24	20	16	60
Assiduidade	22	20	18	60
Relações	22	22	16	60
CAIG (parcial)	1.10	1.02	0.88	

0.74

Avaliação feita pelo Aluno 3

Critério	Aluno 1	Aluno 2	Aluno 3
Ideias	20	20	20
Realização	20	20	20
Empenho	20	20	20
Assiduidade	20	20	20
Relações	20	20	20
CAIG (parcial)	1.00	1.00	1.00

Final	Aluno 1	Aluno 2	Aluno 3
CAIG	1.097	1.030	0.873
Trabalho: 11	12.07	11.33	9.60
Trabalho: 14	15.36	14.42	12.22
Trabalho: 18	19.75	18.54	15.71



Avaliação por pares

- A avaliação individual será: $\text{avaliação_do_proj} * \text{CAIG}$
 - $\text{CAIG} = 1$ - a nota individual será a da avaliação do trabalho
 - $\text{CAIG} < 1$ - a nota individual será inferior à avaliação do trabalho
 - $\text{CAIG} > 1$ - a nota individual será superior à avaliação do trabalho
- $\text{CAIG} < 0.75$ - não é atribuída nota prática
- $\text{CAIG} > 1.00$ - subida de nota limitada a dois valores
- Atenção
 - Fundamental realizar a avaliação com seriedade
 - Não se estão a avaliar amigos mas o contributo de cada um para o grupo
 - Fundamental acordo quanto a critérios de avaliação e sua ponderação



Avaliação por pares

- Critérios propostos

Critério		Peso	Descrição
C1	Ideias para o trabalho	20%	Contribuição com ideias e sugestões para a condução dos trabalhos e resolução de problemas
C2	Realização do trabalho	30%	Qualidade da execução das tarefas necessárias à realização do trabalho
C3	Empenho no trabalho	25%	Esforço realizado para que o trabalho seja entregue no prazo
C4	Assiduidade	15%	Contribuição para a marcação de reuniões do grupo e comparência nas mesmas
C5	Relações interpessoais	10%	Contribuição para a motivação, manutenção de bom ambiente e para a resolução de eventuais conflitos

- Estes critérios estarão à discussão até 10 de Outubro
 - enviem comentários e sugestões!!



Desenvolvimento de Sistemas de Software

Programming is fun,
but developing **quality** software **is hard**.

(Philippe Kruchten)

