

# Engenharia de Software: Proposta de Trabalho

Ivo Pereira  
ivopereira@ufp.edu.pt

Alessandro Moreira  
afmoreira@ufp.edu.pt

2020

Universidade Fernando Pessoa  
Faculdade de Ciência e Tecnologia

## 1 Sistema de suporte a gestão de projetos

Pretende-se que os alunos apliquem na prática as várias ferramentas de engenharia de software abordadas nas aulas, no desenvolvimento de todas as etapas de um projeto de software cuja finalidade é suportar a gestão de projetos. Em concreto, devem focar-se no planeamento, especificação, modelização, implementação, teste e documentação deste projeto.

### 1.1 Objetivos do Projeto

Neste projeto pretende-se que desenvolvam um Web Service REST que suporte a gestão de projetos que uma empresa executa para seus clientes. Um determinado cliente pode ter muitos projetos executados pela empresa e cada projeto é constituído por muitas tarefas que necessitam de um determinado número de horas para serem executadas. Cada tarefa é executada por um empregado da empresa e que possuem um dos cargos listados a seguir, com o respectivo valor-hora:

Cargo	Valor-hora
Desenvolvedor Júnior	10 Euros/hora
Analista Júnior	20 Euros/hora
Desenvolvedor Sénior	40 Euros/hora
Analista Sénior	80 Euros/hora

Table 1: Cargos e valor-hora

O cliente, numa primeira iteração, deverá ser capaz de usar o Web Service para consultar informações sobre os seus projetos, especialmente tempo e valor estimados para a conclusão. O Web Service deve ser capaz de gerir as operações necessárias para as funcionalidades descritas.

Numa segunda iteração, a aplicação deverá ser capaz de representar a execução efetiva das tarefas. Os empregados devem incluir períodos de tempo que dedicaram à tarefa e o gestor do projeto deve atribuir um percentual de conclusão à tarefa. O cliente deverá poder consultar o estado do projeto com base na comparação das estimativas iniciais e no progresso da tarefa efetiva.

### 1.2 Endpoints mínimos

O conjunto mínimo de endpoints a serem implementados serão os seguintes:

- POST /tarefa
- POST /projeto
- POST /empregado
- PATCH /projeto/tarefa/{idProjeto}
- PATCH /tarefa/empregado/{idTarefa}

- GET /projeto/{idProjeto}/valor
- GET /projeto/{idProjeto}/tempo

### 1.3 Metodologia

Este projeto deverá contemplar todas as fases de gestão e desenvolvimento do processo de engenharia de software e não deverá focar-se estritamente na implementação. Assim, o trabalho deve cobrir todas as fases do processo de engenharia de software e ser suportado pelos modelos e ferramentas abordados nas aulas.

O(s) responsável(eis) pelo projeto deve(m) começar por fazer uma análise do problema e delinear a execução do projeto, organizando-o em várias tarefas com prazos e custos envolvidos.

Tanto a análise inicial de requisitos e respectivos cenários de utilização, como o desenho da arquitetura e da implementação, devem basear-se em padrões existentes. Toda a modelização deve ser suportada e documentada por diagramas UML adequados.

Devem ainda documentar a implementação efectuada (ajustada à metodologia e ferramentas escolhidas), bem como o conjunto de testes delineados para avaliar o protótipo. Devem ainda gerar a documentação adequada ao software desenvolvido.

### 1.4 Componentes

O projeto deverá ser realizado por no mínimo 2 e no máximo 3 membros, sem exceções.

## 2 Relatórios

O projeto proposto deve ser planeado, modelizado, implementado, testado e documentado com recurso às várias ferramentas abordadas nas aulas. Nomeadamente, toda a documentação deve ser criada seguindo as recomendações ECSS-E-ST-40C da ESA, conforme apresentado nas aulas teóricas. A documentação inclui: Software requirements specification (SRS), Software design document (SDD) e Software (unit-integration) test plan (SUITP).

O SRS deverá ser entregue até a 6ª semana de aulas juntamente com um relatório intercalar em formato electrónico (PDF). Este relatório deverá incluir todo o trabalho desenvolvido até a data, i.e.:

- planeamento do projeto
- análise de requisitos
- arquitetura da aplicação

A modelização deverá ser suportada pelos diagramas UML necessários, que devem ser entregues juntamente com o relatório. O relatório deve identificar as tarefas a realizar, sua calendarização e qual o membro do grupo responsável pela sua implementação. O relatório final deve ser entregue em formato electrónico (PDF) através do sistema de e-learning até ao dia definido na plataforma. O relatório final deverá atualizar o intercalar e descrever os testes desenvolvidos. Devem ainda entregar em separado o código desenvolvido, em um ficheiro zip, bem como a documentação gerada. Por fim, o protótipo desenvolvido deverá ser apresentado aos docentes da disciplina.

## 3 Avaliação

Cada projeto será avaliado com base na tabela 2:

Requisitos	Valor
Implementação	6
Padrões	2
Testes	4
Logging	0.5
Análise Estática do Código	0.5
Git	3
UML	2
Planeamento	2
Total	20

Table 2: Avaliação

Cada item da avaliação será avaliado com base nos critérios definidos na tabela 3 abaixo:

Critério de avaliação	Suficiente	Bom	Excelente
Implementação	Deverá conter as entidades do modelo de dados, bem como todos os endpoints definidos como mínimos no enunciado para a primeira iteração	-	Deverá conter as entidades do modelo de dados, bem como todos os endpoints necessários para a segunda iteração da aplicação
Padrões	Deverá utilizar pelo menos 1 padrão de projeto	Deverá utilizar pelo menos 2 padrões de projeto	Deverá utilizar pelo menos 4 padrões de projeto
Testes	O projeto deverá ter cobertura de testes de 60% das linhas do projeto	O projeto deverá ter cobertura de testes de 80% das linhas do projeto	O projeto deverá ter cobertura de testes de 95% das linhas do projeto
Logging	O grupo deverá utilizar logging de forma consistente em um ficheiro único	O grupo deverá utilizar logging de forma consistente com pelo menos 2 níveis em um ficheiro	O grupo deverá utilizar logging de forma consistente com 4 níveis e diversos ficheiros
Análise Estática do Código	-	-	O projeto deverá possuir ranking A no Codacy
Git	O grupo deverá demonstrar utilização consistente da ferramenta com existência de commits ao longo das semanas	-	O grupo deverá demonstrar uso consistente da ferramenta e utilização de um workflow à escolha
UML	O grupo deverá entregar, sem graves erros, os diagramas de caso de uso e de classes	O grupo deverá entregar, sem graves erros, os diagramas de caso de uso e de classes e de sequência	O grupo deverá entregar, sem graves erros, os diagramas de caso de uso e de classes, de sequência e mais outro à escolha
Planeamento	O grupo deverá apresentar o planeamento do projeto com as atividades atribuídas para cada componente e o respectivo cronograma	O grupo deverá apresentar o planeamento do projeto (atividades e cronograma) e representar corretamente uma metodologia de desenvolvimento	O grupo deverá apresentar o planeamento do projeto (atividades e cronograma), representar corretamente uma metodologia de desenvolvimento e produzir os documentos adequados em cada etapa

Table 3: Critérios de avaliação

A tabela 4 mostra os valores que cada critério irá possuir para cada item de avaliação.

Insuficiente	0%
Suficiente	50%
Bom	75%
Excelente	100%

Table 4: Valoração dos critérios

A nota final do projeto será o somatório de todos os itens avaliados com os valores dados pelos critérios. Por exemplo, se um dado item possui 10% da nota final e o aluno obteve um critério Excelente, então ele irá obter 2 valores neste item ( $20 \times 10\% = 2 \times 100\% = 2$ ). Se tivesse obtido um Suficiente, o valor deste item seria 1 ( $20 \times 10\% = 2 \times 50\% = 1$ ).

## 4 Extras

Independente do mencionado na secção anterior, serão atribuídos entre 1 a, no máximo, 2 valores para cada um dos desafios propostos abaixo:

- Utilização de técnicas de CI/CD, especialmente integração entre Jenkins e Github para atualização e implantação da aplicação
- Utilização de serviços cloud (Microsoft Azure, Amazon AWS ou Google Cloud) para hospedagem da aplicação
- Utilização de containers para desenvolvimento e implantação da aplicação. Será muito valorizada a separação dos módulos em diferentes containers com a utilização de docker-composer

Os desafios extras estão elegíveis aos projetos que cumpriram, pelo menos, o critério Suficiente para os requisitos de Implementação, Testes, UML e Planeamento. Deverão submeter um documento à parte que explique o que foi feito para o cumprimento do desafio, para além da apresentação que demonstre o funcionamento. A qualidade deste documento será determinante no valor a ser atribuído.