

# Engenharia de Software: Proposta de Trabalho

Feliz Gouveia  
fribeiro@ufp.edu.pt

Alessandro Moreira  
afmoreira@ufp.edu.pt

2018

Universidade Fernando Pessoa  
Faculdade de Ciências e Tecnologia

## 1 Sistema de suporte a explicações

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 explicações no âmbito do ensino superior. 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 Web Services que suportem explicações a conteúdos programáticos de cadeiras do ensino superior. Deverá haver pelo menos dois tipos de utilizadores, alunos que buscam por explicações e os explicadores. Os explicadores deverão dar explicações a uma ou várias cadeiras de um determinado curso e deverão informar as disponibilidades que possuem em período de tempo (início e fim) para cada dia. O explicador deverá ainda, informar os idiomas em que consegue ministrar as explicações.

O aluno que busca explicação deverá poder pesquisar explicadores pelos mais variados critérios, sendo que, no mínimo por cadeiras, dias, períodos de tempo, idiomas, separadamente ou por conjunção destes critérios (filtros de pesquisa).

Por fim, o sistema deverá permitir que um aluno realize uma marcação de explicação, com 1 hora de duração, a um dado explicador. Um explicador não pode atender mais que um aluno simultaneamente. O modelo deverá contemplar a modelização de uma universidade do ponto de vista de suas faculdades e dos cursos e respectivas cadeiras. O projeto deverá ser implementado em duas fases:

- Fase 1: Desenvolver um WS1 (Web Service) que permite fazer a gestão dos recursos do sistema: criar, atualizar e listar os explicadores, faculdades, cursos e cadeiras e criar atendimentos. Um WS1 é uma instância de uma universidade.
- Fase 2: Desenvolver um WS2 que permita fazer pesquisas em diversas instâncias de WS1 (mínimo 2). Os serviços ofertados pelo WS2 deverão utilizar os serviços expostos pelas diferentes instâncias de WS1 registadas.

**Todo o projeto deve ser implementado com a linguagem Java e utilizando a *framework Spring*.**

### 1.2 Endpoints

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

Fase 1

- POST /explicador
- POST /faculdade
- POST /curso/{faculdade}
- POST /cadeira/{curso}

- PUT /explicador/{curso}
- PUT /explicador (deverá ser utilizado para definir as disponibilidades do explicador)
- GET /explicador
- GET /explicador/{nome\_explicador}
- GET /explicador?curso={curso}&dia={dia}&inicio={hora\_inicio}&fim={hora\_fim}
- POST /atendimento

#### Fase 2

- POST /explicador/{universidade}
- PUT /explicador (deverá ser utilizado para definir as disponibilidades do explicador)
- PUT /explicador/{universidade}/{curso}
- GET /explicador?curso={curso}&dia={dia}&inicio={hora\_inicio}&fim={hora\_fim}
- GET /explicador/{universidade}?curso={curso}&dia={dia}&inicio={hora\_inicio}&fim={hora\_fim}
- POST /atendimento/{universidade}

### 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.

**O projeto deverá ser realizado por no mínimo 2 e no máximo 3 componentes, sem exceções. O grupo deverá indicar, no Documento de Início do Projeto, quais tarefas cada componente irá realizar. Um trabalho em grupo é a soma dos trabalhos individuais de cada componente. A avaliação irá refletir o contributo de cada integrante no resultado final.**

## 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 PSS-05 da ESA, conforme apresentado nas aulas teóricas. A documentação inclui:

- Documento de Início de Projeto (PID)
- Requisitos Utilizador (URD)
- Projeto de Arquitetura (ADD)
- Projeto Detalhado (DDD)
- Transferência (STP)

Deve ser dada especial atenção à definição dos testes e procedimentos de instalação e verificação do software.

O PID deverá ser entregue até 10 dias após a publicação deste documento na plataforma de elearning. Deverá ser entregue um relatório intercalar em formato electrónico (PDF), através do sistema de elearning, até o dia definido no referido sistema. Este relatório deverá ter 2 a 3 páginas (máximo) e incluir todo o trabalho desenvolvido até a data, i.e.:

- planeamento do projeto
- análise de requisitos
- arquitetura

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 o dia definido na plataforma. O relatório final deverá atualizar o intercalar, conter de 4 a 6 páginas, no máximo, e descrever os testes desenvolvidos. Devem ainda entregar em separado o código desenvolvido, bem como a documentação gerada. Por fim, o protótipo desenvolvido deverá ser apresentado em sala de aula aos docentes da disciplina. Em alternativa ao tema acima apresentado poderão os alunos elaborar uma nova proposta de tema que deverá ser entregue, aos respetivos docentes, no prazo de 15 dias após a divulgação deste documento. Esta proposta após análise poderá ser considerada válida e irá substituir o tema acima proposto; contudo os módulos de avaliação e os requisitos de avaliação acima mencionados serão mantidos, alterando-se apenas o caso de uso do projeto.

### 3 Avaliação

Cada projeto será avaliado com base na tabela 1:

Table 1: Avaliação

Fase 1	19.44%
Fase 2	9.72%
Padrões	8.33%
Testes	16.67%
Logging	4.17%
Codacy	4.17%
Git	4.17%
UML	16.67%
Planeamento	16.67%
Total	100.00%

Cada item da avaliação será avaliado com base nos critérios definidos na tabela 2 abaixo:

Table 2: Critérios de avaliação

Critério de avaliação	Suficiente	Bom	Excelente
Fase 1	Deverá conter as entidades do modelo de dados, bem como todos os endpoints definidos como mínimos no enunciado	Deverá incluir mais 2 entidades do modelo que suportem pesquisas adicionais OU mais 3 endpoints	Deverá incluir mais 2 entidades do modelo que suportem pesquisas adicionais E mais 3 endpoints
Fase 2	Deverá conter as entidades do modelo de dados, bem como todos os endpoints definidos como mínimos no enunciado	Deverá permitir inclusão da informação de novas instâncias WS1 (url, identificador) via endpoint, bem como fazer cache destas instâncias (via Map ou Ehcache)	Deverá permitir inclusão da informação de novas instâncias WS1 (url, identificador) via endpoint, bem como fazer cache destas instâncias (via Ehcache) e obter outras informações (p. ex. morada da universidade) via API de terceiros (p. ex. Open Street View, Google Maps)
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%	O projeto deverá ter cobertura de testes de 80%	O projeto deverá ter cobertura de testes de 95%
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
Qualidade de Código	O projeto deverá possuir pelo menos ranking C no Codacy	O projeto deverá possuir pelo menos ranking B no Codacy	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 sem necessidade de utilizar múltiplas branches	O grupo deverá demonstrar utilização consistente da ferramenta com existência de commits ao longo das semanas com a utilização de múltiplas branches	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

Para os critérios Bom e Excelente da fase 1, o grupo deverá demonstrar o valor acrescentado aos utilizadores pelas inclusões implementadas.

A tabela 3 mostra os valores que cada critério irá possuir para cada item de avaliação.

Table 3: Valoração dos critérios

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

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 ( $20 \cdot 10\% = 2 \cdot 100\% = 2$ ). Se tivesse obtido um Suficiente, o valor deste item seria 1 ( $20 \cdot 10\% = 2 \cdot 50\% = 1$ ).