

## ThesisMan: Gestão de Teses de Mestrado

A Faculdade de Ciências da Universidade de Lisboa, tal como muitas outras universidades que leccionam o segundo ciclo, tem a necessidade de gerir o estado dos mestrados dos seus alunos, desde a submissão de temas por parte de docentes e das empresas, até à marcação das salas a usar.

### Contextualização

Em cada ano lectivo, existe um processo que começa em Maio do ano lectivo anterior onde tanto os docentes, como as empresas podem submeter propostas de temas para teses (título, descrição, mestrados compatíveis e remuneração mensal em euros).

Por volta de Junho, abre o período de candidaturas em que os alunos escolhem até 5 temas em que estão interessados. Quando este período fecha, os temas são atribuídos aos alunos por ordem descendente da média. Alunos que fiquem sem tema são resolvidos manualmente pelo administrador, embora se saiba que é raro acontecer.

Com essa atribuição começa a tese, que pode ser uma de duas modalidades: Dissertação ou Projecto. Ambas as modalidades são orientadas sempre por um docente, mas o Projecto tem também um orientador externo da empresa.

Por volta de Janeiro, os alunos têm de entregar um documento que é a proposta de tese. Têm também de realizar uma defesa (presencial ou remota) desse documento. No caso de ser presencial terá de ser numa sala que não esteja ocupada nesse dia/hora. A defesa tem a duração de 1 hora. A essa defesa é atribuída uma nota pelo júri. O júri é composto pelo orientador interno, bem como um outro docente (arguente). A inserção da nota é da responsabilidade do orientador interno. No caso da nota ser inferior a 10 valores, o aluno poderá entregar um novo documento e será marcada uma nova defesa da proposta.

Por volta de Junho, os alunos entregam uma versão final do documento e é marcada a defesa final. Nesta vez existe um terceiro elemento que é o presidente do júri, responsável este pela inserção da nota final. No caso do aluno não ter nota igual ou superior a 10, fica reprovado à tese, e terá de realizar outra no ano lectivo seguinte. Tal como a defesa da proposta, pode ser presencial ou remota, tendo a mesma restrição da sala estar livre, mas com uma duração de 90 minutos.

## 1 Casos de Uso

Numa versão inicial pretendemos implementar alguns casos de uso, deixando outros para segunda fase. Os casos de uso assinalados com  $F_1$  foram identificados para esta primeira fase. No entanto, apresenta-se uma lista mais completa de casos de uso, de forma a que a equipa tenha uma noção do desenvolvimento futuro.

- A. Login com autenticação da universidade (Nota: vamos fazer mock desta funcionalidade e qualquer palavra passe será aceite)
- B. Registo de utilizadores empresariais

- C. Login de utilizadores empresariais
- D. Submissão de temas por parte dos docentes
- E. Submissão de temas por parte dos utilizadores empresariais
- F. Listar os temas disponíveis neste ano lectivo, por parte dos alunos
- G. Candidatura a um tema (limite de 5), por parte dos alunos
- H. Cancelamento da candidatura a um tema
- I. Atribuição dos temas aos alunos (da parte do administrador)
- J. Submissão do documento de proposta de tese, por parte dos alunos
- K. Marcação da Defesa da proposta de tese, por parte do orientador da tese
- L. Registo da nota da defesa da proposta de tese, por parte do orientador da tese
- M. Submissão do documento final de tese, por parte dos alunos
- N. Marcação da Defesa de tese, por parte do orientador da tese, incluindo a nomeação do júri.
- O. Registo da nota da defesa da proposta de tese, por parte do presidente do júri.
- P. Recolha de estatísticas sobre taxa de sucesso dos alunos.

## 2 Requisitos Não Funcionais

O projecto tem ainda os seguintes requisitos:

- A autenticação deverá ser feita com o sistema de autenticação da FCUL.
- Toda a informação deverá ser armazenada numa base de dados relacional.
- A plataforma deverá ser implementada em Spring Boot, de forma a ter um custo de desenvolvimento baixo, e a usar a linguagem Java, que é a linguagem para a qual é mais fácil de contratar engenheiros.
- A camada de dados deverá usar JPA e Spring Data.
- A aplicação deverá lidar com pedidos concorrentes, sem criar inconsistências.
- A camada de negócios deverá usar o Domain Model, com meta-dados suficientes para que o JPA faça o mapeamento para a base de dados.
- A interface deverá ser através da web, num momento inicial.
- Deverá ser exposta uma API REST que os clientes (web ou mobile) poderão usar para interagir com a aplicação.
- O repositório deverá aceitar apenas código com o nível de qualidade aceite pela equipa (usando o pre-commit).
- O projecto deverá correr dentro de um docker, que será a forma como vai ser deployed no servidor de produção.
- O controlo da participação de cada membro da equipa será feita através da actividade no repositório git.
- As decisões técnicas deverão estar justificadas num relatório técnico.

### 3 Tarefas da Fase 1

A equipa deverá:

- Fazer fork do repositório original (<https://git.alunos.di.fc.ul.pt/css000/thesisman>), para a conta de um dos elementos do grupo.
- Dar acesso a todos os membros do grupo, como *Maintainer*.
- Dar acesso à conta css000, como *Reporter*.
- Alterar o Readme, para identificar a equipa.
- Criar uma pasta *docs*, onde colocará um único documento PDF, com todos os desenhos resultantes do projecto.
- Desenhar o modelo de domínio que modela o problema e colocar na pasta *docs*.
- Desenhar o SSD para o caso de uso K.
- Desenhar um esboço do diagrama de classes que mostre a divisão em camadas.
- Implementar o código correspondente ao diagrama.
- Anotar as classes relevantes com as anotações JPA, de forma a fazer o melhor mapeamento possível.
- Justificar no relatório o porquê de cada decisão tomada neste mapeamento.
- Gerar a base de dados a partir das anotações escritas.

Nesta entrega não se pretende ainda a realização dos casos de uso, nem nenhuma interface de interação. Deverão, no entanto, verificar que a base de dados está correcta, de acordo com os princípios que aprenderam na cadeira de Bases de Dados e em Construção de Sistemas de Software.

### 4 Como entregar

Para entregarem o trabalho, basta identificarem a tag da fase1, e colocá-la no servidor. O repositório deve ter o código fonte do projecto, os ficheiros necessários para o correr no docker, bem como um único documento PDF na pasta docs, com todos os documentos necessários.

```
git tag fase1
git push origin fase1
```

Devem confirmar que o vosso projecto está acessível à conta CSS000 na tag fase1. Caso contrário, terão 0 nesta entrega.

### 5 Critérios de Avaliação

Esta entrega corresponde a 30% da nota do projecto. Os restantes 70% serão avaliados na segunda meta. Para perceberem onde se devem focar, segue uma lista de critérios em que serão avaliados:

#### 5.1 Processo de Desenvolvimento

- Granularidade dos commits
- Qualidade das mensagens de commit
- Qualidade do código presente no documento

## 5.2 Requisitos

- Em que medida os requisitos funcionais conseguem ser cumpridos com este modelo de domínio.

## 5.3 Relatório

- Descrição da arquitectura em camadas.
- Qualidade do Modelo de Domínio.
- Qualidade do Diagrama de Classes.
- Qualidade do SSD pedido.
- Qualidade do Mapeamento JPA
- Qualidade da justificação das decisões de mapeamento, de acordo com os padrões descritos na aula.