

# **Proposta de trabalho**

## **Engenharia Software**

Feliz Gouveia

[fribeiro@ufp.edu.pt](mailto:fribeiro@ufp.edu.pt)

Christophe Soares

[csoares@ufp.edu.pt](mailto:csoares@ufp.edu.pt)

2015

**Universidade Fernando Pessoa**

Faculdade de Ciência e Tecnologia

# 1. Sistema para a apoio na gestão duma Universidade

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 que seja capaz de armazenar e gerir informação de uma Universidade. Em concreto, devem focar-se no planeamento, especificação, modelização, implementação, teste e documentação do projeto.

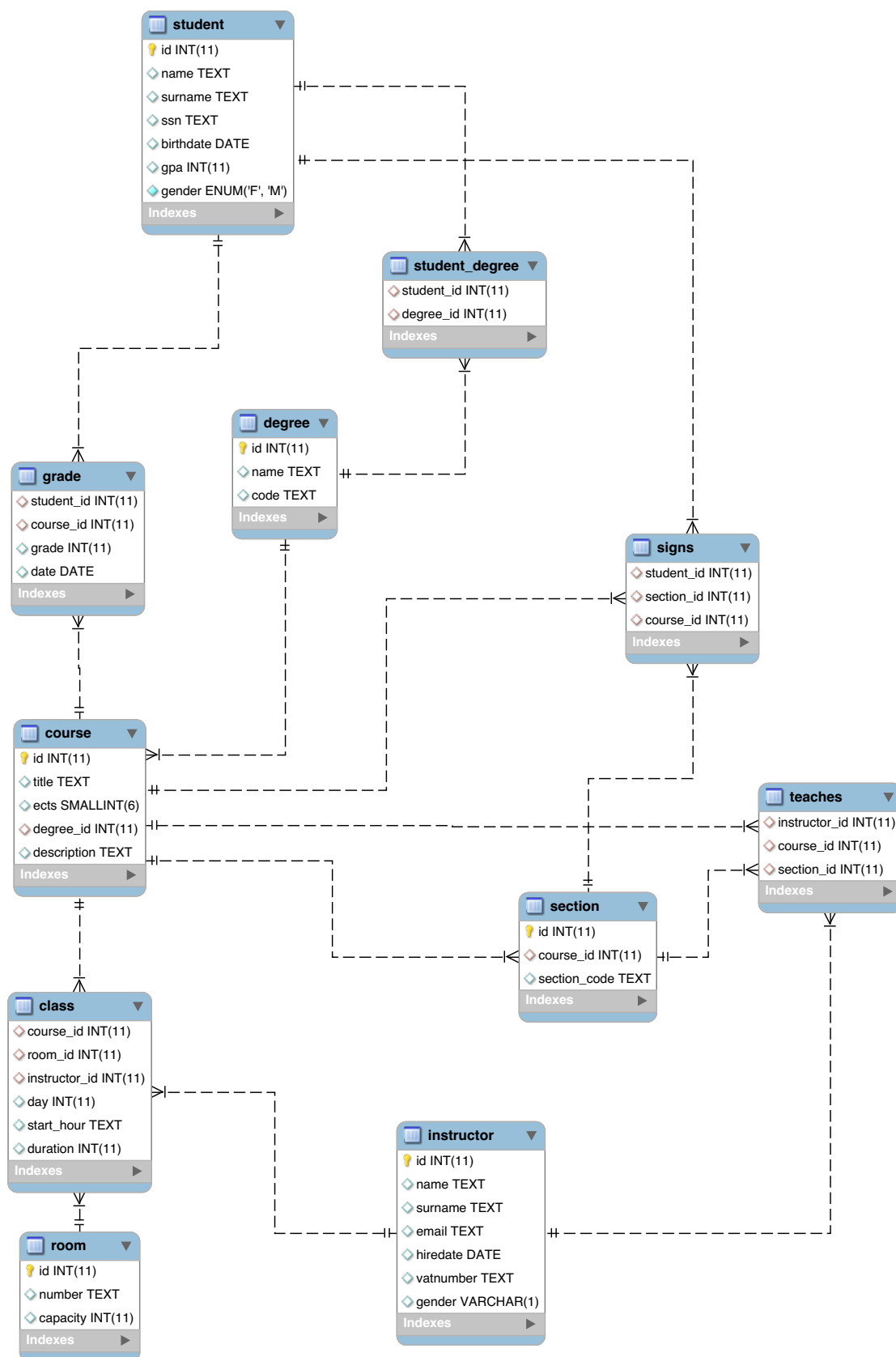
## 1.1. Objectivos do Projeto

Neste projeto pretende-se que desenvolvam um sistema capaz de automatizar a gestão de uma Universidade. Este deve permitir a existência de três perfis distintos: os alunos, os docentes e os administrativos. Todos estes perfis só poderão aceder aos serviços web com a utilização de um mecanismo de autenticação baseado em *tokens*, e terão ações distintas. Por exemplo, os alunos poderão inscrever-se em cursos e respetivas unidades curriculares, enquanto que os docentes poderão rever a lista de alunos inscritos, lançar notas, marcar faltas, *etc.* O perfil administrativo irá permitir validar as inscrições dos alunos, fazer a atribuição das unidades curriculares aos respetivos docentes, imprimir pautas das notas, definir as salas, *etc.* Estes serviços *web* irão permitir melhorar e agilizar os serviços entre os três tipos de perfis existentes. Em concreto pretende-se:

- i) Desenvolver um WS1 (*Web Service*) que permita aos estudantes: inscreverem-se em cursos, atualizar os seus dados, inscreverem-se em unidades curriculares obterem o seu horário, verificar a ocupação das salas, pesquisar se o docente se encontra disponível num determinado dia e período de tempo, consultar as suas notas. Deve elaborar uma interface na linguagem a sua escolha que permita evocar os métodos implementados e submeter e obter a respectiva informação. Este WS1 deve controlar o acesso aos dados e métodos.
- ii) Desenvolver um WS2 (*Web Service*) que permita aos docentes: consultarem o seu horário, obter a lista de presença por unidade curricular num determinado horário, lançar notas numa unidade curricular, pesquisar salas livres, atualizar os seus dados. Como no primeiro serviço *web* deve implementar uma interface para os docentes que permita testar todas os métodos implementados enviando e apresentando a informação.
- iii) Desenvolver um WS3 (*Web Service*) que permita aos administrativos validar as matrículas feitas pelos alunos, criar/inserir unidades curriculares, associar docentes a unidades curriculares, obter as listas de alunos inscritos, criar o horário para as unidades curriculares, definir salas para as unidades curriculares, consultar a listas das notas de um determinado aluno, obter a lista de todas as salas livres num determinado período de tempo. A semelhança dos outros serviços web deve criar uma interface que permita testar todos os métodos implementados enviando e apresentando a respetiva informação.

Resumidamente, neste projeto deverá desenvolver três WS e as suas respectivas interfaces para permitir a sua utilização. O WS1 deverá permitir um conjunto de interações específicas aos alunos na interação com o modelo de dados de uma Universidade. O WS2 deve permitir aos docentes realizarem operações relativamente as Unidades Curriculares que leccionam, como lançar notas, verificar o horário e aceder a lista de alunos presentes. Por fim, o WS3 deve permitir aos administrativos supervisionar as ações dos alunos, e gerir as turmas que compõe o modelo de dados da Universidade.

Este trabalho deve ser realizado reutilizando o modelo de dados estudados na cadeira de Base de dados, sendo que o respetivo diagrama ER se encontra abaixo representado. Poderão alterar o mesmo modelo para que este se adeque às necessidades identificadas. Poderão de mesmo acrescentar novas funcionalidades aos WS, caso identifiquem novos requisitos nos perfis. Por fim, podem inserir dados sintéticos na base de dados de forma a testar os seus WS. Podem colocar dados reais, consultando os cursos e unidades curriculares existentes na UFP através da seguinte ligação: <http://fct.ufp.pt>.



**Figura 1 - Modelo de dados genérico de uma Universidade**

## 1.2. Metodologia e Avaliação

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 por toda a panóplia de modelos e ferramentas abordadas 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.

## 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.

O **relatório intercalar** deve ser entregue em formato electrónico (PDF) através do sistema de *elearning* até ao dia **definido no elearning**. Este relatório deverá ter 3 a 4 **páginas** (máximo) e incluir todo o trabalho desenvolvido até a data, i.e., i) planeamento do projeto, ii) análise de requisitos e iii) arquitetura. A modelização deverá ser suportada pelos diagramas UML necessários que devem ser entregues juntamente com o relatório.

O **relatório final** deve ser entregue em formato electrónico (PDF) através do sistema de *elearning* até ao dia **definido no elearning**. O relatório final deverá atualizar o relatório intercalar, conter 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 aulas aos respectivos docentes da disciplina.