



# Engenharia de Software

Realizado por:  
Tiago Cardoso Nº 28106

Docentes:  
Feliz Ribeiro Gouveia  
Christophe Soares

Data de entrega : 12 de Janeiro de 2017

# Introdução

Este projeto tem como objetivo a criação de um sistema que faça a gestão de uma universidade e que seja automatizada.

Neste sistema vai haver 3 perfis diferentes:

- Alunos;
- Docentes;
- Administrativos;

Para ajudar os serviços entre os perfis vai-se desenvolver 3 Web Services diferentes, um para cada perfil.

# Descrição Geral

O Web Service 1 deve permitir aos alunos 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.

O Web Service 2 deverá permitir 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.

O Web Service 3 deve permitir aos administradores 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.

Nestes 3 Web Services deve de ser criada uma interface que permite que aos utilizadores testar todos os métodos implementados.

## Modelização

A modelização, que está disponível no ficheiro ArgoUML, foi feita durante o início do projeto tendo sido atualizada e alterada no decorrer do desenvolvimento do projeto.

## Arquitetura

Neste projeto foi implementada uma arquitetura MVC, que é um padrão de arquitetura de software que separa a aplicação em 3 partes:

- View;
- Model;
- Controller;

Essencialmente foquei-me em fazer o Web Service na arquitetura REST.

- <https://pt.wikipedia.org/wiki/REST>

## Framework

Neste projeto foi usado a Micro Framework do Lumen, da empresa do Laravel na qual facilitou bastante em termos de rotas e de gerenciamento da base de dados.

## Análise de Requisitos

Requisitos funcionais:

- Operações que os Web Services permitem aos utilizadores.
  - Web Service 1 deverá permitir um conjunto de interações específicas aos alunos na interação com o modelo de dados de uma Universidade.
  - Web Service 2 deve permitir aos docentes realizarem operações relativamente às Unidades Curriculares que lecionam, como lançar notas, verificar o horário e aceder à lista de alunos presentes
  - Web Service 3 deve permitir aos administrativos supervisionar as ações dos alunos, e gerir as turmas que compõem o modelo de dados da Universidade.
- A interface disponível ao utilizador deverá permitir realizar estas operações.

Requisitos não funcionais:

- A aplicação deve de ter pelo menos idioma de Inglês.
- A aplicação deve de apresentar informação específica.
- A aplicação deve ser intuitiva.

# Testes Desenvolvidos

Foram desenvolvidos vários testes através de dados fictícios gerados pela biblioteca “Faker” na qual foi utilizado para testar as rotas, modelo relacional e as query’s.

## Controlo de Versões

Todo o desenvolvimento a nível de código pode ser encontrado no meu repositório Github que a partir da entrega do relatório final e do projeto passa a ser publico até a nota ser lançada, após será removido ou movido novamente para privado.

- [https://github.com/tiagocardosoweb/engsoft\\_university](https://github.com/tiagocardosoweb/engsoft_university)

## Documentação

A documentação foi gerada através do ApiGen que a meu ver é muito simples de usar, clean e utilizável, a documentação pode ser encontrada no directório “documentos/api”.

- <http://www.apigen.org>

## Interfaces

Como o projeto trata-se de um Web Service preferi não criar interfaces, preferi usar requisições em cURL usando o Postman.

- <https://www.getpostman.com>

A coleção que criei está também disponível na pasta documentos.

- Engenharia Software - Project.postman\_collection

# Requisitos Implementados

Os requisitos implementados estão disponíveis no ficheiro markdown “readme.md”.