PROJECTARY



Projeto de Sistemas de Informação

**Autores:**

Rafael Escudeiro

Ricardo António

**Docentes:**

Paulo Santos

Pedro Dias

Engenharia Informática - 2016/2017



# Resumo

# Glossário

|  |  |
| --- | --- |
| Termo | Descrição |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Indíce

[**Resumo**](#_1cepcd971qfi) **1**

[**Glossário**](#_yb481c7fti4k) **2**

[**Indíce**](#_ycws4bns6g1z) **3**

[**Introducão**](#_cob1d2wgp6kr) **5**

[**Panorama tecnológico atual**](#_myayj1o03fe5) **5**

[**Problema em estudo**](#_bfv305g135t2) **5**

[**Benefícios e aspectos inovadores**](#_myayj1o03fe5) **5**

[**Estrutura do relatório**](#_myayj1o03fe5) **5**

[**Enquadramento do Projeto**](#_igv0h5jmtlr7) **5**

[**Estado da arte**](#_kr995f9ebwy2) **5**

[**Tecnologias utilizadas**](#_9a7q3u1ub7ja) **6**

[**Equipas do projeto:**](#_nn3k6sh6w7rs) **7**

[**Análise de Sistemas (https://github.com/iptomar/projectary-as) - equipa responsável pela realização do levantamento de requisitos, funcionais e não-funcionais, realização de diagramas de Use Cases,modelo de dados, atividades e de sequência, realização da apresentação do projeto e relatório.**](#_hfj1j93ea4rt) **7**

[**Análise de Sistemas**](#_s7cc3orddnmy) **7**

[**Frontend**](#_wmlqxwty4c7v) **8**

[**Desenvolvimentos Futuros**](#_7sjhmmrrdwz) **8**

[**Conclusões**](#_myayj1o03fe5) **8**

# Introducão

# Panorama tecnológico atual

# Problema em estudo

# Benefícios e aspectos inovadores

# Estrutura do relatório

# Enquadramento do Projeto

No âmbito da cadeira de Projeto de Sistemas de Informação foi-nos solicitado a realização de um sistema de informação que tivesse utilidade num futuro próximo e que tivesse como objetivos a aplicação dos conhecimentos adquiridos durante a licenciatura e a descoberta e utilização de novas tecnologias. Surgiu então a ideia da criação de um sistema onde fosse possível submeter os enunciados dos projetos finais e onde os alunos se poderiam candidatar para o seu desenvolvimento. Surge então o “Projectary”.

# Estado da arte

**Links:** <https://github.com/iptomar/projectary-docs/blob/master/EstadoDaArte_relatorio.pdf> , <https://github.com/iptomar/projectary-docs/blob/master/EstadoDaArte_apresentacao.pdf>

Foi realizado um estado de arte do projeto e foram encontradas vários sistemas de informação idênticos à solução que pretendemos implementar, entre elas está um portal de projetos, da Universidade Federal de Santa Maria, no Brasil, que nos pareceu bastante mais similar à nossa ideia do que as outras soluções encontradas.

Este portal de projetos da Universidade Federal de Santa Maria (<https://portal.ufsm.br/projetos/>) é uma plataforma onde os utilizadores submetem os seus projetos e, caso haja permissão, estes são apresentados numa lista pública com todos os detalhes do projeto desenvolvido.

Outro sistema de informação que nós encontramos, denominado UpWork (<https://www.upwork.com/>), é um site de freelancer que é utilizado por empresas para submeterem projetos e onde freelancers se podem candidatar a esses projetos. Este site contém também uma plataforma onde estes freelancers podem promover os seus currículos através de perfis, com algumas informações como formas de contacto, histórico de trabalhos com feedback, entre outras. Os clientes, ao submeterem os projetos/trabalhos ficam sujeitos a uma análise por parte de outros utilizadores que podem submeter uma proposta relativa a esses projetos.

Por último, o PeerLibrary (<https://peerlibrary.org/>) é um projeto open-source (<https://github.com/peerlibrary>) da UC Berkeley de procura, submissão e partilha de artigos científicos que facilita a procura de pesquisa científica semelhante. Os utilizadores podem submeter os seus artigos, partilhá-los ou colaborar na pesquisa científica dos mesmos.

# Tecnologias utilizadas

As tecnologias utilizadas para a realização do projeto foram:

1. GitHub (<https://github.com/iptomar/>) - utilizado para controlo de versões e para a colaboração de todos os intervenientes do projeto;
2. Go (<https://golang.org/>) - primeira linguagem utilizada para a realização da API, sendo substituída posteriormente por existir uma incompatibilidade entre o driver do mySQL e o mySQL por não permitir o uso de multiple statements. Isto levou à substituição da linguagem devido à interação da API com a bd ser feita com recurso a chamadas de procedimentos;
3. mySQL (<https://www.mysql.com/>) - sistema de gestão de base de dados relacionais utilizada;
4. Angular (<https://angular.io/>) - framework, em TypeScript, utilizada para a realização do Frontend;
5. Node.js (<https://nodejs.org/en/>) - linguagem utilizada depois da exclusão do Go para a realização da API e realização de testes;
6. Express.js (<https://expressjs.com/>) - framework para Node.js utilizada para a realização da API;
7. Vagrant (<https://www.vagrantup.com/>) - utilizado para criar uma plataforma de desenvolvimento homogénea, escrita na linguagem Ruby;
8. mysqltest (<https://dev.mysql.com/doc/mysqltest/2.0/en/mysqltest.html>) - utilizado para testar um server de mysql e opcionalmente comparar os outputs com um ficheiro de resultado;
9. mysqlcheck (<https://dev.mysql.com/doc/refman/5.7/en/mysqlcheck.html>) - utilizado para verificar a integridade das tabelas ao importar o dump no GitHub;
10. Supertest (<https://github.com/visionmedia/supertest>) - utilizado para verificar se a API está corretamente implementada, por exemplo, verificar se as rotas estão a retornar a informação correta para ser consumida ou se a API está preparada para certos inputs do cliente;
11. Selenium (<http://www.seleniumhq.org/>) - framework utlizada para testar o frontend, escrita em Selenese que permite testar várias linguagens.
12. Protactor (<http://www.protractortest.org/#/>) - utilizado para testar o frontend, simulando a interação entre a aplicação e a maneira como um utilizador iria utilizar;
13. Karma (<https://karma-runner.github.io/1.0/index.html>) - utilizado para testar os métodos do frontend.

# Equipas do projeto:

# Análise de Sistemas (<https://github.com/iptomar/projectary-as>) - equipa responsável pela realização do levantamento de requisitos, funcionais e não-funcionais, realização de diagramas de Use Cases,modelo de dados, diagramas de atividades e sequência, realização da apresentação do projeto e relatório.

* 1. Base de Dados (<https://github.com/iptomar/projectary-bd>) - equipa responsável pela criação e manutenção da base de dados do projeto, procedimentos e diagrama da base de dados
  2. Frontend (<https://github.com/iptomar/projectary-frontend>) - equipa responsável pela realização da interface do projeto para o utilizador.
  3. API (<https://github.com/iptomar/projectary-api>) - equipa responsável pela realização do backend e do Vagrant File;
  4. Testes (<https://github.com/iptomar/projectary-tests>) - equipa responsável pela realização dos testes à aplicação;

# Análise de Sistemas

A equipa de análise de sistemas começou por realizar um pequeno draft do levantamento de requisitos que deu origem depois à primeira versão do levantamento de requisitos (<https://github.com/iptomar/projectary-as/tree/master/Levantamento%20de%20Requisitos%20Projectary>). Depois de analisado o levantamento de requisitos foram realizados diversos diagramas de Use Cases onde estão representados os autores do sistema e as ações que cada um pode fazer (<https://github.com/iptomar/projectary-as/tree/master/Use%20Cases>). De seguida foram realizados vários diagramas de atividades do projeto (<https://github.com/iptomar/projectary-as/tree/master/Diagramas%20de%20Atividades>) e de sequência (<https://github.com/iptomar/projectary-as/tree/master/Diagramas%20de%20Sequ%C3%AAncia>). Esta foi a equipa também responsável por manter os requisitos atualizados e as fases de desenvolvimento do projeto.

# Frontend

A equipa de frontend começou por fazer um pequeno estudo sobre a framework a utilizar, o Angular. A linguagem que utilizaram foi o TypeScript, pois não existia documentação para JavaScript. Atualmente, passados alguns meses do início da realização projeto, já existe documentação para Angular em JavaScript, pelo que é muito tarde para a migração. Para a utilização do frontend é necessário iniciar um server com a API. Depois disto é necessário fazer o download do projeto e instalar os Node Modules (npm install) na pasta do projeto e, se necessário mudar o IP na pasta localizada no src/main.ts (<https://github.com/iptomar/projectary-frontend/blob/master/src/main.ts>) para o IP da API.

O frontend da aplicação está a consumir os dados enviados pela API e a fazer o render no frontend. O frontend é composto por uma pasta src, que contém todo o código (<https://github.com/iptomar/projectary-frontend/tree/master/src>).

# Desenvolvimentos Futuros

# Conclusões