

Projectary

Projeto de Sistemas de Informação

2016/2017

Docentes:

Paulo Santos | Pedro Dias



ENGENHARIA
INFORMÁTICA
INSTITUTO POLITÉCNICO DE TOMAR

Projectary

Em que consiste:

Solução web para gerir projetos finais

Componentes do projeto:

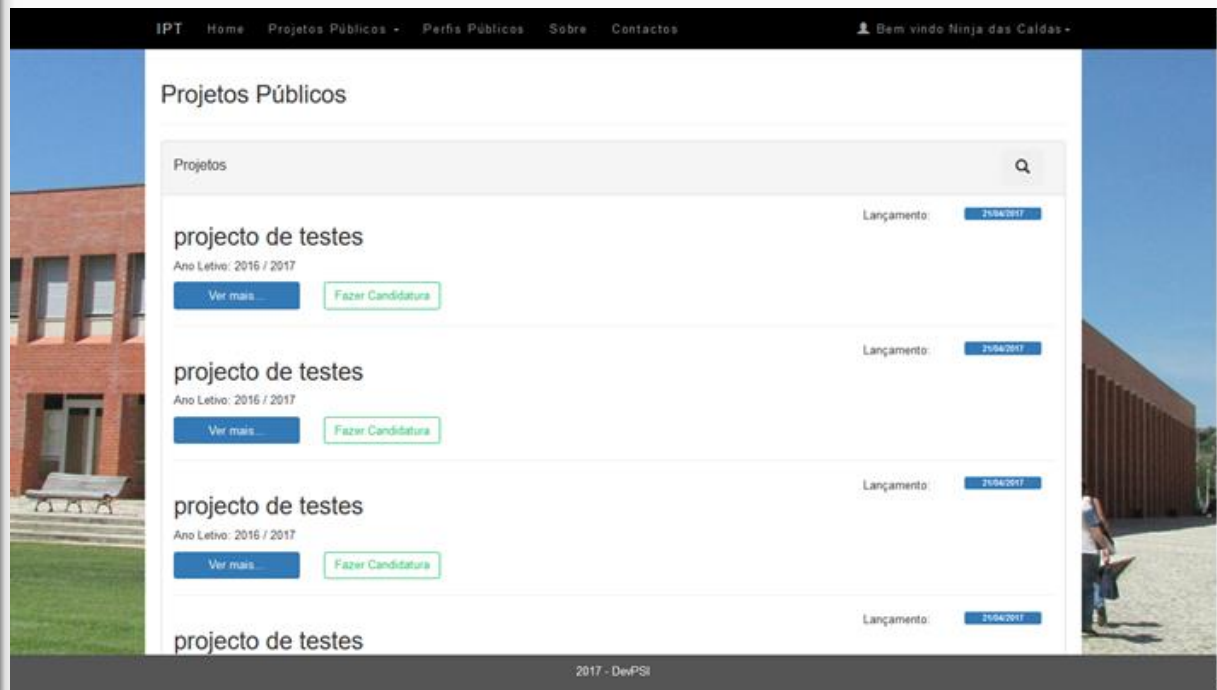
Gestão do lifecycle do projeto

Entrega de enunciados, relatórios e todas as componentes de avaliação

Candidatura a um projeto

Criação de perfil de utilizador

Criação de grupos



Áreas Funcionais

Análise de Sistemas



Base de Dados



Frontend

→ **A.S. responsável por:**

- ◆ Requisitos do sistema
- ◆ Diagramas
- ◆ Apresentação do projeto
- ◆ Relatório

→ **B.D. responsável por:**

- ◆ Desenho da BD

→ **Frontend responsável por:**

- ◆ Toda a interação direta com o utilizador

Áreas Funcionais

API



Vagrant



Testes

→ **API responsável por:**

- ◆ Todo o Back-end
- ◆ Criação de rotas
- ◆ Comunicação entre Frontend e BD

→ **Vagrant responsável por:**

- ◆ Criar ambientes de desenvolvimento
- ◆ Criar ambientes de

→ **Testes responsável por:**

- Testar as diferentes componentes criadas
- Testar a aplicação final

Análise de Sistemas





Levantamento de Requisitos

Requisitos Funcionais

Exemplo:

B. O sistema possibilita aos visitantes as seguintes funções:

- a. Visualização dos projetos públicos
- b. Registo no sistema
- c. Visualização dos perfis públicos





Levantamento de Requisitos

Requisitos Não-Funcionais

Exemplo:

I. Ao entrar nos sistema o utilizador deve deparar-se com:

- a. o logótipo da plataforma
- b. pequena mensagem de boas vindas à plataforma
- c. destaque de alguns Projetos Finais públicos finalizados que estão inseridos na plataforma





Problemas reais que o projeto visa resolver

Inexistência de uma plataforma de gestão de projetos

Ausência de uma biblioteca de projetos finais

Falta de comunicação entre docentes e alunos

Não divulgação dos projetos finais





Tipos de Utilizador



Visitante



Docente

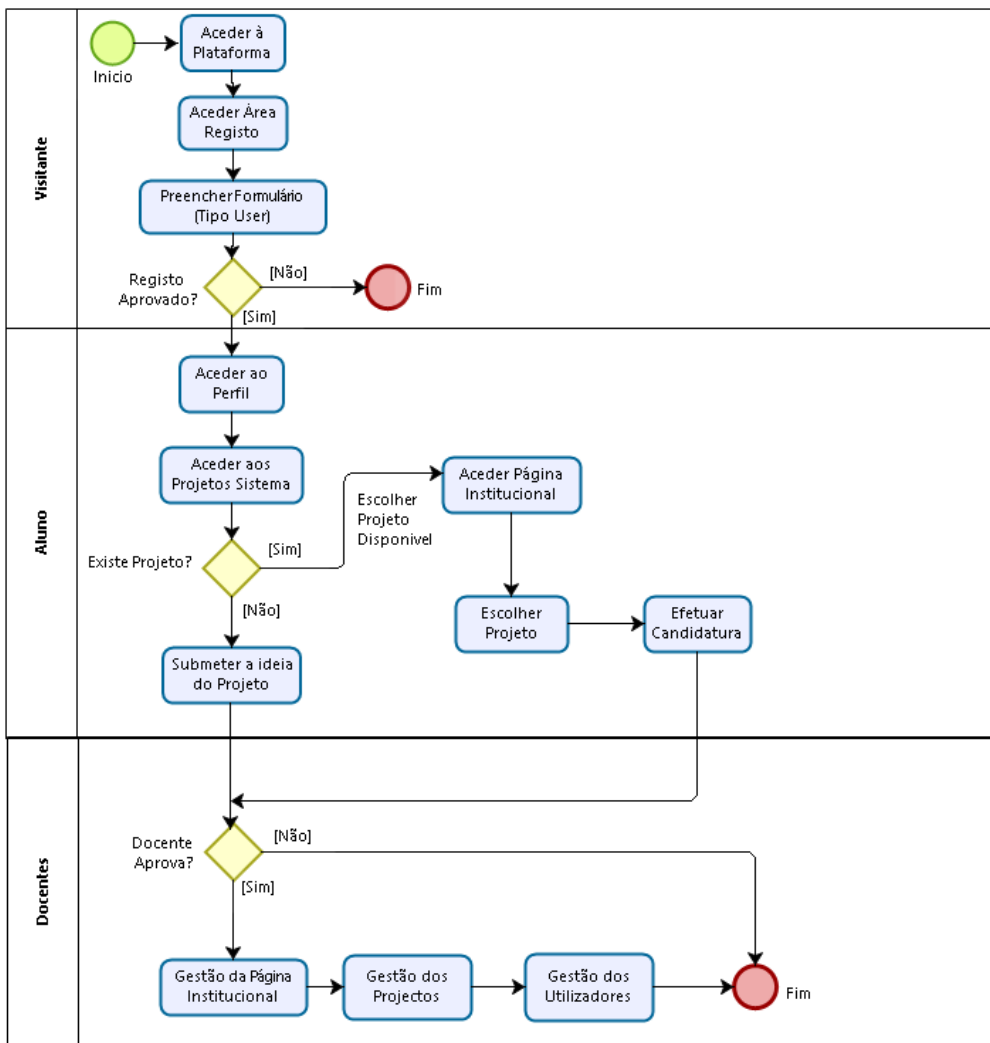


Aluno



Diagrama

Caso real



Base de Dados



MySQL

Sistema de Gestão de Base de Dados (SGBD) adoptado: MySQL

Modelo relacional

Sistema multiplataforma

Fortes capacidades de integração

Utilização sem custos

Suporta transações, cursores, stored procedures, funções, triggers

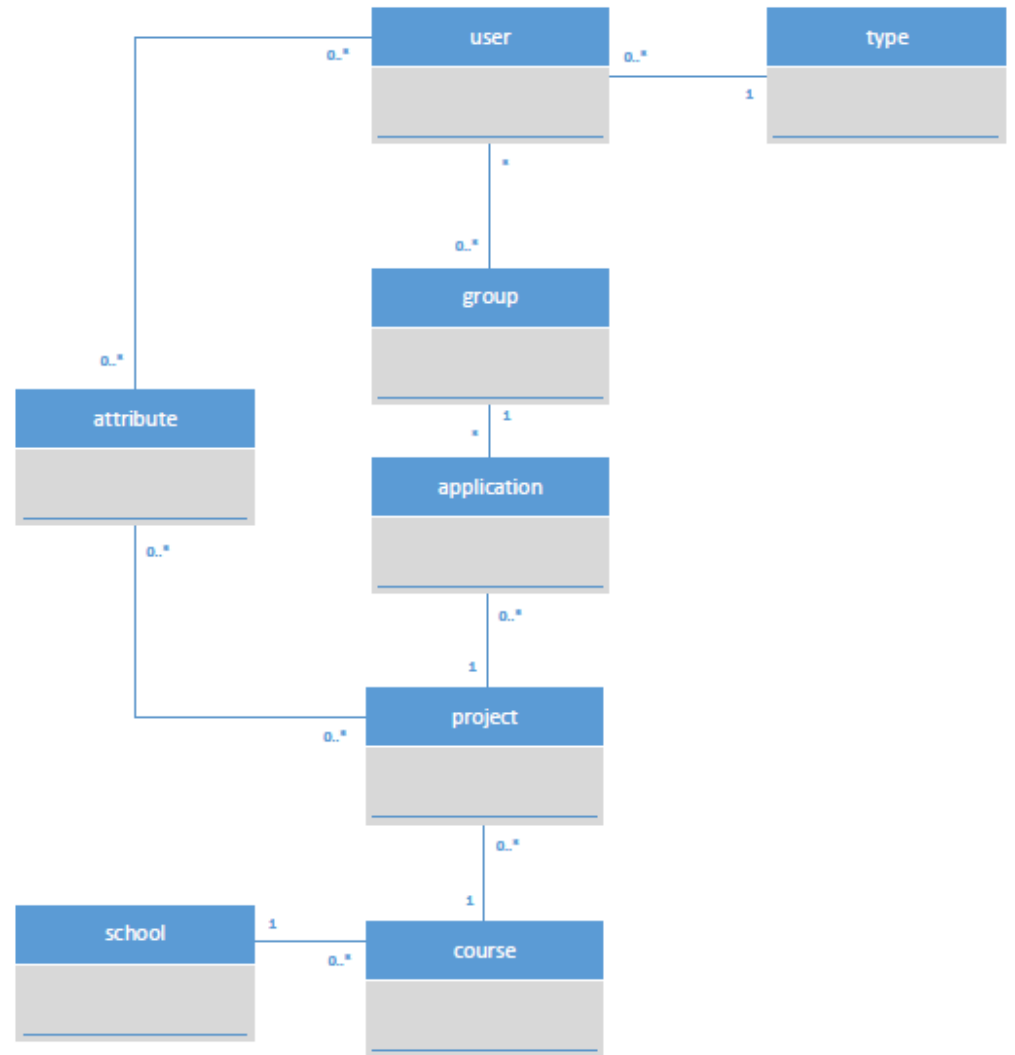
Conhecimento consolidado





Modelo de Dados

Diagrama reduzido do
modelo da base de dados



FrontEnd 



Tecnologias do Frontend

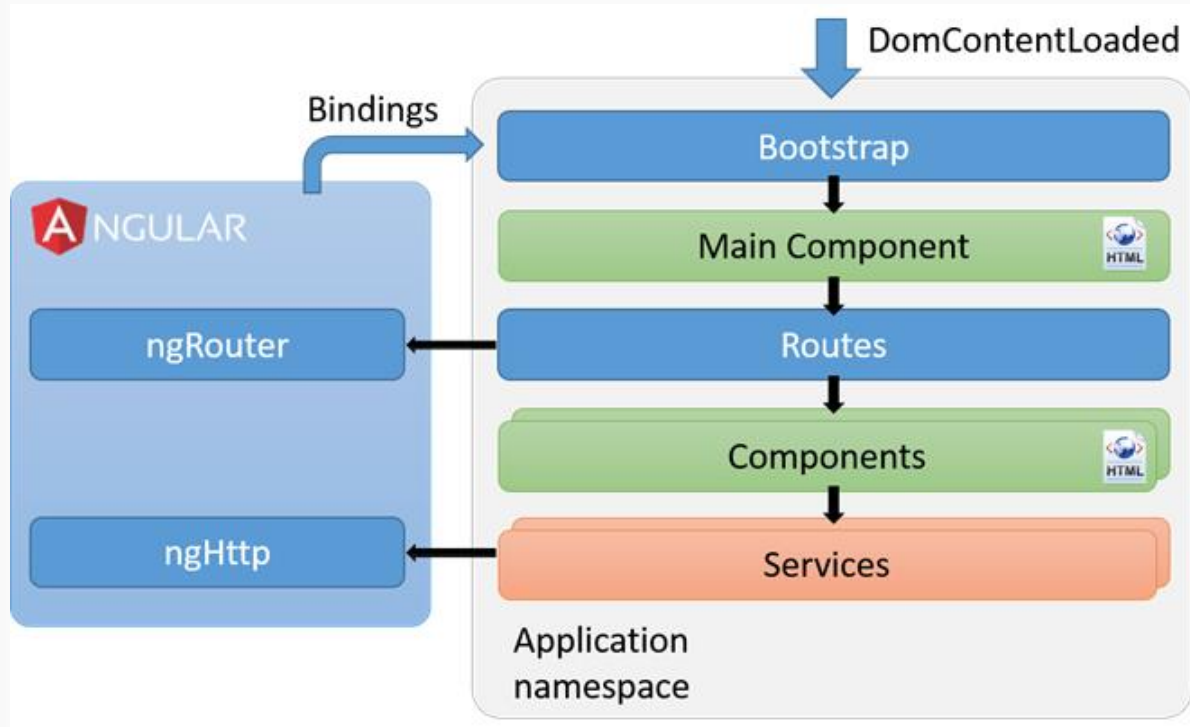
- Estudo da estado da arte
- Principais candidatas após o estudo:
 - Angular
 - React
 - Vue
- Angular foi a tecnologia escolhida





Diagrama

O Angular é normalmente utilizado com typescript e baseia-se em componentes que irão ser alimentados por um serviço, a sua estrutura de implementação pode ser representada desta forma:





Componentes & Serviços

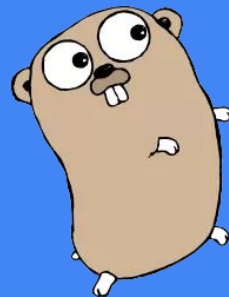
A comunicação entre o Frontend e a API é essencialmente feita através de pedidos Ajax, direcionados à API.

Os dados são enviados e recebidos em formato JSON.



API 

Desenvolvimento Inicial



API desenvolvida inicialmente em Go.

O Go é linguagem compilada, focada na produtividade e programação concorrente.

O Node.js foi posteriormente adotado devido ao Go não permitir a execução de procedimentos e 'multiple statements'.





Node.js

Modelo assíncrono

I/O orientado a eventos

Reutilização da mesma linguagem do frontend

Rápido desenvolvimento

Comunidade alargada





Arquitetura REST

A arquitetura utilizada para a API baseia-se na filosofia **RESTful**

CREATE
READ
UPDATE
DELETE

As respostas são dadas pelo protocolo HTTP tendo em conta os códigos de erro do mesmo, sendo o corpo de dados normalmente em JSON.

{ REST }



Rotas Principais

Método	Rota	Descrição
POST	/LOGIN	Login dos utilizadores
GET	/USER	Obter a lista de utilizadores
POST	/TEACHER	Criar um professor
GET	/SCHOOL	Obter lista de escolas
GET	/COURSE/:ID	Obter cursos de uma escola
GET	/PROJECT	Obter lista de projetos
GET	/PROJECT/:ID	Obter dados de um projeto

Vagrant





VAGRANT

- Criar e manter ambientes de desenvolvimento virtuais portáteis
- Replicação de ambientes de desenvolvimento
- Padronização de aplicações



Testes





Tecnologias Usadas

NodeJS

API:

Supertest

Mocha

Frontend:

Protractor (Selenium)

Base de Dados:

MySQL (Queries, mysqltest e mysqlcheck)



Tipos de Testes

Testes de *Integridade* (Base de Dados)

Testes E2E (Frontend)

Testes Funcionais (API)



Conclusão



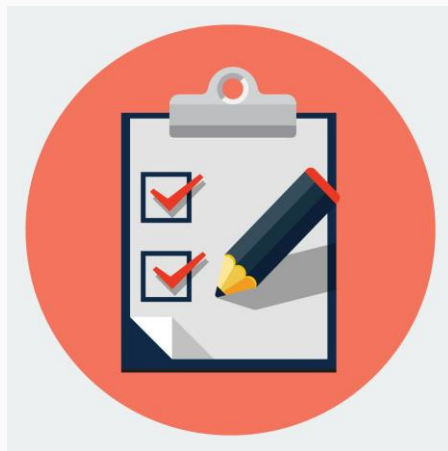
O que ficou por fazer

Notificações

Horários/Locais disponíveis

Upload/Download de ficheiros

Criar projetos





Trabalho Futuro

Página personalizada para cada projeto

Projetos criados por alunos

Upload ou envio de assets para o projeto

Chat



Curiosidades



GitHub



github.com/iptomar/projectary-bd/issues/132

Remove MD5 #132



asantos3 opened this issue 21 hours ago · 4 comments



asantos3 commented 21 hours ago

Member



Curiosidades

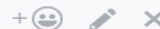


GitHub



razoes commented 21 hours ago • edited

Member



I'll remove it as soon as the API team agrees with it and do the hashing. Or else meanwhile the passwords would be in plain text.



This was referenced 21 hours ago

Plain password on groups dump #133

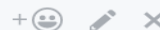
Add removed MD5 hashing iptomar/projectary-api#60

Closed

Open



MarkRijckenberg commented an hour ago



I suggest reading this:

https://en.wikipedia.org/wiki/Secure_Hash_Algorithms

MD5, SHA-0 and SHA1 are all vulnerable to collision attacks.

So if you need to hash, I suggest replacing md5 with sha256 or sha512.

Regards,

Mark Rijckenberg

Projects



None yet

Milestone



No milestone

Notifications

Subscribe

You're not receiving notifications from this thread.

3 participants



Lock conversation

Projectary

Projeto de Sistemas de Informação

2016/2017

Docentes:

Paulo Santos | Pedro Dias



ENGENHARIA
INFORMÁTICA
INSTITUTO POLITÉCNICO DE TOMAR