

PROJECTARY



ENGENHARIA
INFORMÁTICA

Projeto de Sistemas de Informação

Autores:

André Santos
Luís Marques
Rafael Escudeiro
Ricardo António
Rui Marques

Docentes:

Paulo Santos
Pedro Dias

Engenharia Informática - 2016/2017



Instituto Politécnico de Tomar

Resumo

Projectary resulta de um novo paradigma para Projetos Finais, resultando numa solução web para gerir Projetos Finais.

Neste projeto, desenvolveu-se uma plataforma Web que permite o acesso a Projetos Finais de licenciatura de uma determinada instituição de ensino, possibilitando assim uma gestão do lifecycle do projeto, uma entrega de enunciados, relatórios e componentes de avaliação, candidaturas a projetos, bem como a criação de perfis e de grupos.

Estiveram envolvidos neste projeto os alunos da disciplina de Projeto de Sistemas de Informação da Licenciatura em Engenharia Informática do Instituto Politécnico de Tomar, como também os docentes responsáveis pela disciplina, Paulo Santos e Pedro Dias.

No desenvolvimento deste projeto, foram utilizadas diversas tecnologias, tais como Angular, Typescript, MariaDB, MySQL, Node.js, Vagrant, entre outras.

Palavras Chave: Projetos Finais, Plataforma, Angular, MySQL, Node.js.

Glossário

Termo	Descrição
API	Application Programming Interface
Backend	Lado do servidor
Draft	Rascunho
Framework	Conjunto de bibliotecas ou classes reutilizáveis para um sistema
Freelancer	Trabalhador por conta própria
Frontend	Lado do cliente
Lifecycle	Ciclos de vida
Thread	Processo em execução
Use Cases	Casos de Uso
Script	Conjunto de instruções passíveis de serem interpretadas

Índice

Resumo	1
Glossário	2
Índice	3
Introdução	6
Panorama tecnológico atual	6
Problema em estudo	6
Benefícios e aspectos inovadores	6
Estrutura do relatório	6
Enquadramento do Projeto	8
Estado da arte	8
Tecnologias utilizadas	9
Equipas do projeto:	10
Análise de Sistemas	10
Base de Dados	10
Frontend	10
API	10
Testes	10
Análise de Sistemas	11
Base de Dados	13
MySQL	13
MariaDB	13
Frontend	15
Angular	15
TypeScript	15
API	17
Go	17
Node.js	17
Rotas	18
Vagrant	20
Testes	23
API	23
Base de Dados	23

Frontend	24
Desenvolvimentos Futuros	25
Conclusões	26
Bibliografia	27
Anexo A - Diagramas Análise de Sistemas	28
Use Cases	28
Aluno	28
Docente	28
Diagramas de Atividades	29
Registo na plataforma	29
Estado da inscrição	30
Criar enunciado para os projetos	31
Submeter ideia do projeto	32
Disponibilização de projetos na plataforma	33
Candidatura	34
Atribuição de projetos	35
Estabelecer horário de atendimento	36
Edição da página institucional por parte do alunos	37
Edição da página institucional por parte dos docentes	37
Disponibilização horário reuniões	38
Disponibilização ata final da Avaliação	39
Disponibilização de datas e notas	40
Submeter relatório final	41
Diagramas de Sequência	42
Candidatura Projetos	42
Disponibilização de informações	43
Lista de utilizadores p/ aluno	43
Lista de alunos p/ docente	44
Registo aluno	45
Registo docente	45
Visualizar perfis públicos	46
Login do aluno	47
Disponibilização dos projetos	47
Edição da página institucional p/ alunos	48
Edição da página institucional p/ docentes	48
Modelo de Dados	49
Anexo B - Levantamento de Requisitos	50
Funcionais	50
Não-Funcionais	57

Anexo C - Screenshots da Aplicação**59**

1. Introdução

1.1. Panorama tecnológico atual

A tecnologia tem sido uma forte aposta por parte da sociedade mundial nesta época a que chamamos Indústria 4.0. Isto leva a um grande desenvolvimento e modernização da indústria nacional, tornando-a mais rica, produtiva, flexível, eficiente e globalmente competitiva.

A revolução tecnológica que se tem vivido nos dias de hoje tem vindo a acentuar-se cada vez mais. Atualmente, tudo necessita de tecnologia e torna-se imprescindível arranjar novas soluções para alguns problemas, otimizando os processos e tornando-os melhores e mais eficientes.

Um dos processos que se tenta otimizar e melhorar é o processo de Projetos Finais de uma determinada licenciatura.

1.2. Problema em estudo

O problema que estamos a tentar resolver é o facto de não haver uma plataforma que funcione como uma base de dados para projetos finais realizados, bem como todas as informações alusivas a estes e onde os alunos se podem candidatar a esses projetos.

Isto leva a que a forma como as candidaturas ao projeto são feitas, a forma como esses projetos ficam disponíveis, desde que são conhecidos até que terminados, e as informações pertencentes a cada projeto sejam feitas de maneira desorganizada.

1.3. Benefícios e aspectos inovadores

Com os problemas referidos no ponto anterior 1.2, pretende-se que esta plataforma resulte num aumento da organização, coordenação e inovação relativamente à forma de inserção dos enunciados dos projetos como também ao seu arquivo. Isto leva a uma maior flexibilidade e facilidade para os orientadores e/ou docentes responsáveis da disciplina.

1.4. Estrutura do relatório

Este documento está estruturado da seguinte forma:

- Capítulo 2: Enquadramento do Projeto
 - Estado da arte

- Tecnologias utilizadas
- Capítulo 3: Equipas do Projeto
 - Análise de Sistemas
 - Base de Dados
 - Frontend
 - API
 - Testes
- Capítulo 4: Análise de Sistemas
- Capítulo 5: Base de Dados
 - MySQL
 - MariaDB
- Capítulo 6: Frontend
 - Angular
 - TypeScript
- Capítulo 7: API
 - Go
 - Node.js
 - Vagrant
- Capítulo 8: Testes
- Capítulo 9: Desenvolvimentos Futuros
- Capítulo 10: Conclusões
- Capítulo 11: Bibliografia
- Capítulo 12: Anexo A - Diagramas Análise de Sistemas
- Capítulo 13: Anexo B - Levantamento de Requisitos
- Capítulo 14: Anexo C - Screenshots da Aplicação

2. 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”, cujo nome se baseia na palavra Projeto (Project) e Biblioteca (Library), criando-se assim o conceito de biblioteca de projetos.

2.1. Estado da arte

Foi realizado um estado de arte do projeto, onde 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 consideramos que seja a que mais se enquadra com a nossa ideia.

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.

Links:

- Relatório do estado da arte do projeto:
https://github.com/iptomar/projectary-docs/blob/master/EstadoDaArte_relatorio.pdf

- Apresentação do estado da arte do projeto:
https://github.com/iptomar/projectary-docs/blob/master/EstadoDaArte_apresentacao.pdf

2.2. Tecnologias utilizadas

As tecnologias utilizadas para a realização do projeto foram as seguintes:

1. GitHub (<https://github.com/iptomar/>) - utilizado para o 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 esta e o driver do MySQL, caracterizando-se por o MySQL por não permitir o uso de multiple statements. Isto levou à substituição da linguagem devido à interação da API com a base de dados ser feita com recurso a chamadas de procedimentos;
3. MySQL (<https://www.mysql.com>) - sistema de gestão de base de dados relacionais utilizado;
4. Angular (<https://angular.io>) - framework, em TypeScript, utilizada para a realização do Frontend;
5. Node.js (<https://nodejs.org>) - 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 utilizada para testar o frontend, escrita em Selenese que permite testar várias linguagens.
12. Protractor (<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.

3. Equipas do projeto:

3.1. Análise de Sistemas

Equipa responsável pela realização do Levantamento de requisitos (funcionais e não-funcionais), realização dos diagramas de Use Cases, Modelo de Dados, Diagramas de Atividades e Sequência, realização da apresentação do projeto e relatório.

Link: <https://github.com/iptomar/projectary-as>

3.2. Base de Dados

Equipa responsável pela criação e manutenção da base de dados do projeto, procedimentos e diagrama da base de dados.

Link: <https://github.com/iptomar/projectary-bd>

3.3. Frontend

Equipa responsável pela realização da interface do projeto para o utilizador (frontend).

Link: <https://github.com/iptomar/projectary-frontend>

3.4. API

Equipa responsável pela realização do backend e do Vagrant File.

Link: <https://github.com/iptomar/projectary-api>

3.5. Testes

Equipa responsável pela realização dos testes à plataforma.

Link: <https://github.com/iptomar/projectary-tests>

4. Análise de Sistemas

O primeiro passo da equipa de Análise de Sistemas foi realizar um pequeno draft do Levantamento de Requisitos que deu origem à primeira versão do Levantamento de Requisitos.

Link:

<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.

Link: <https://github.com/iptomar/projectary-as/tree/master/Use%20Cases>

Após os Use Cases serem realizados, procedeu-se à realização dos vários Diagramas de Atividades do projeto.

Link: <https://github.com/iptomar/projectary-as/tree/master/Diagramas%20de%20Atividades>

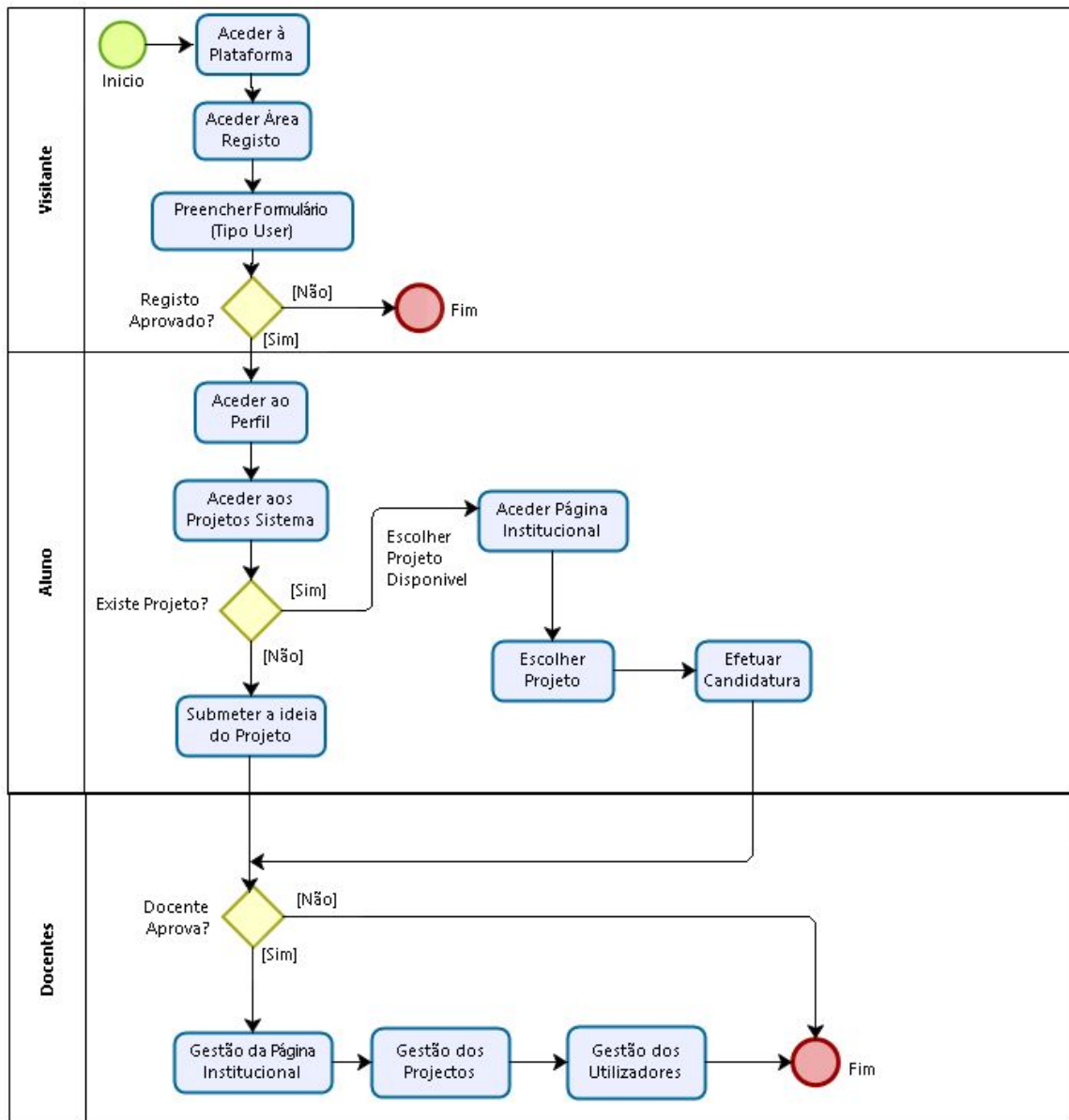
Aquando da conclusão dos Diagramas de Atividades, realizaram-se os Diagramas de Sequência.

Link:

<https://github.com/iptomar/projectary-as/tree/master/Diagramas%20de%20Sequ%C3%Aancia>

A equipa de Análise de Sistemas foi também responsável por manter os requisitos atualizados e as fases de desenvolvimento do projeto.

Na figura seguinte está representado um diagrama que representa o processo geral do funcionamento da plataforma Projectary.



5. Base de Dados

5.1. MySQL



MySQL é um sistema de gestão de base de dados que utiliza a linguagem SQL. O MySQL foi criado por um grupo de pessoas, mais especificamente David Axmark, Allan Larsson e Michael “Monty” Widenius, que trabalham juntos desde a década de 1980.

Atualmente, é desenvolvido pela Oracle Corporation, cuja versão estável é a 5.7.15 (2016). O seu grande sucesso deve-se sobretudo à integração com diversas tecnologias utilizadas nos dias de hoje.

5.2. MariaDB



MariaDB é uma base de dados lançada em 22 de janeiro de 2009. Foi desenvolvida pelo conhecido Michael ‘Monty’ Widenius, um dos fundadores do MySQL.

Em suma, MariaDB é o MySQL na forma evoluída, pois além de todos os comandos, bibliotecas, interfaces e APIs características do MySQL estarem presentes na MariaDB, a MariaDB focou-se especialmente muito na segurança, pelo que é uma das principais características da base de dados.

Eis algumas das características desta base de dados:

- Portabilidade (imensas plataformas, digamos que quase todas, suportam esta base de dados)
- Excelente estabilidade
- Magnífico desempenho
- Compatibilidade com diferentes linguagens de programação
- Utilização de vários tipos de Storage Engine
- Software livre
- Suporta Triggers, Controlo Transacional, Cursors, Functions e Stored Procedures
- Replicação configurável
- Interface gráfica

6. Frontend

6.1. Angular



Angular é uma tecnologia de frontend, open-source, baseada em TypeScript e JavaScript produzida pela Angular Team, na Google, e por comunidades que permite juntar todo o fluxo de trabalho dos vários developers enquanto estão a ser criadas aplicações muito complexas.

Esta framework foi produzida pela mesma equipa que criou o AngularJS, por isso, o AngularJS refere-se às versões desde a 1 até à 1.x e o Angular refere-se às versões 2 até, atualmente, à 4.0.

Esta framework pode ser usada para construir aplicações para web, mobile e nativos web e mobile, pois tem uma grande velocidade e performance, tirando partido de Web Workers e do render do lado do servidor.

6.2. TypeScript



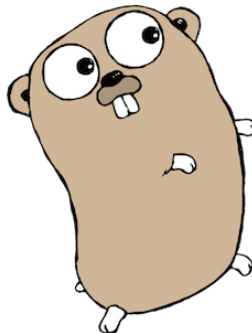
O TypeScript é uma linguagem de programação open-source desenvolvida e gerida pela Microsoft. Anders Hejlsberg, um dos principais arquitetos de C# e o criador do Delphi e Turbo Pascal, trabalhou no desenvolvimento desta linguagem.

O TypeScript é orientado para o desenvolvimento de aplicações JavaScript, tanto para o lado do cliente (client-side) como também para o lado do servidor (server-side), através do Node.js. É uma linguagem para desenvolvimento JavaScript em larga escala, com ela podemos escrever código que se caracteriza por possuir uma estrutura organizada e ter código compilado para JavaScript puro.

Tem as vantagens de ser executada em qualquer navegador, sistema operativo e de ser código aberto.

7. API

7.1. Go



Go é uma linguagem livre e open source criada pela Google para várias plataformas como o Windows, Linux, OS X e smartphones, entre outros. Esta linguagem foi criada para compensar vários erros que outras linguagens tinham, mantendo também os pontos bons dessas linguagens.

As duas maiores implementações são o compilador do Go da Google, o “gc” e o gccgo, uma GCC frontend. Esta linguagem é muito escalável para grandes sistemas, produtiva, legível e suporta muitas funcionalidades como networking e multiprocessamento, etc.

7.2. Node.js



O NodeJS é uma plataforma que foi construída sobre o motor JavaScript do Google Chrome para construir aplicações de rede rápidas e escaláveis, de forma fácil.

Utiliza um modelo de I/O direcionado a eventos não bloqueantes, tornando assim o modelo leve e eficiente, ótimo para aplicações em tempo real com uma troca imensa de dados através de dispositivos distribuídos.

Ou seja, desenvolver aplicações multi-thread e distribuídas usando as tradicionais linguagens síncronas pode ser uma tarefa complexa e desencorajadora.

Tradicionalmente, muitos problemas de desempenho das aplicações decorrem de serem dependentes I/O, no entanto, como o Node.JS foi desenhado para ser não bloqueante e orientado a eventos e não a objetos é ideal para ser distribuído em múltiplos processos e máquinas numa rede.

Nas linguagens como Java™ e PHP, cada ligação cria uma nova thread e esta tem potencialmente anexado cerca 2 MB de memória(pode variar). Um sistema que tenha 8 GB de RAM, põe o número máximo teórico de ligações concorrentes de cerca de 4.000 utilizadores. E quando o número de utilizadores aumenta têm que se aumentar os recursos ou aumentar o número de servidores.

O NodeJS resolve esta questão ao abordar a problemática de como a ligação é tratada no servidor de maneira diferente, este ao invés de criar uma nova thread do SO a cada ligação (e alocar a memória anexa a ela), cada ligação dispara um evento que é executado dentro da engine do Node.

O NodeJS tem o conceito da utilização de módulos. Há literalmente centenas de módulos para se adicionar a um projeto ou globalmente ao node, e a comunidade é bastante ativa a produzir, publicar e atualizar os módulos, sendo alguns deles até alterados diariamente. Por exemplo, neste projeto foi utilizado o módulo express.js.

7.3. Rotas

Diversas rotas foram desenvolvidas para o uso de outras aplicações e serviços, como o frontend. As rotas desenvolvidas até ao momento são as seguintes:

Utilizadores

Método	Rota	Descrição
POST	/LOGIN	Login dos utilizadores
GET	/USER	Obter a lista de utilizadores
PUT	/USER	Atualizar os dados de um utilizador
GET	/USER/PENDING	Obter utilizadores não conformados
GET	/USER/:ID	Obter os dados de um utilizador
POST	/USER/:ID/APPROVE	Aprovar um utilizador
POST	/USER	Criar um estudante
PUT	/USER/CHPASSWORD	Modificar Password

PUT	/USER/:ID/SWLOCK	Bloquear/Desbloquear Utilizador
POST	/TEACHER	Criar um professor

Atributos e Escolas

Método	Rota	Descrição
POST	/ATTRIBUTE	Criar um atributo
GET	/ATTRIBUTE	Obter um atributo
GET	/SCHOOL	Obter lista de escolas
GET	/COURSE/:ID	Obter cursos de uma escola

Projetos

Método	Rota	Descrição
POST	/PROJECT	Criar um projeto
GET	/PROJECT	Obter lista de projetos
GET	/PROJECT/:ID	Obter dados de um projeto
PUT	/PROJECT/:ID	Atualizar dados de um projeto
GET	/PROJECT/:ID/:APPLICATIONS	Obter lista de candidaturas de um projeto
GET	/PROJECT/FINISHED/:USERID/:GROUPID	Verificar se o projeto se encontra finalizado
POST	/PROJECT/FINISHED/:USERID/:STUDENTID	Definir a nota a um aluno de um projeto

Grupos

Método	Rota	Descrição
POST	/GROUP/CREATE	Criar um grupo
POST	/GROUP/JOIN	Adicionar utilizador a grupo
DELETE	/GROUP/:ID	Eliminar grupo

GET	/GROUP	Obter lista de grupos
GET	/GROUP/:ID	Obter dados de um grupo
PUT	/GROUP/:ID	Atualizar dados de um grupo

Candidaturas

Método	Rota	Descrição
GET	/APPLICATION	Obter lista de candidaturas
POST	/APPLICATION	Submeter uma candidatura
GET	/APPLICATION/:ID	Obter informação sobre uma candidatura
GET	/APPLICATION/NASSIGNED/:STATE	Obter candidaturas (não) aprovadas
POST	/APPLICATION/ACCEPT	Aprovar uma candidatura
GET	/APPLICATION/USER/:ID	Obter candidaturas de um determinado utilizador

Auxiliares

Método	Rota	Descrição
GET	/USER/:USER/RECOVER/:TOKEN	Validar a Recuperação de Password (Token enviado por Email)
POST	/USER/:USER/RECOVER	Pedir a Recuperação de Password
POST	/PHOTO	Submeter ou Alterar a Fotografia de Perfil
GET	/PHOTO/:ID	Obter a Fotografia de Perfil de um Utilizador

7.4. Vagrant

O Vagrant é uma ferramenta que permite criar um ambiente de desenvolvimento standard, possível de utilizar por todas as equipas participantes num projeto. Tem

uma biblioteca extensa de máquinas virtuais pré configuradas com o essencial, às quais depois podem ser adicionadas dependências e livrarias necessárias.

O isolamento do projeto numa máquina virtual, permite por um lado evitar conflitos desnecessários com outros serviços em funcionamento no host, essenciais para o utilizador desempenhar outras tarefas, e por outro, apurar em concreto quais as necessidades reais da aplicação a desenvolver.

Para a configuração de qualquer ambiente de desenvolvimento, é necessário em primeira mão a VirtualBox, pois é através dela que o Vagrant irá montar a máquina virtual, e depois o Cygwin com o plugin OpenSSH, que não é essencial, mas que foi a forma mais simples de configurar uma Shell ssh para aceder à máquina virtual (o putty não se revelou compatível na utilização dos certificados gerados).

Utilizamos como base de desenvolvimento uma máquina virtual em Linux Ubuntu que depois foi personalizada de acordo com as necessidades das diversas equipas de desenvolvimento.

O Vagrant recorre a um script de configuração (Vagrantfile), que ao ser executado faz o download do ficheiro da máquina virtual indicada, a partir da biblioteca de VMs, configurando a mesma e instalando as ferramentas, livrarias e dependências necessárias ao desenvolvimento e funcionamento da aplicação.

De notar que além das vantagens já descritas, outra funcionalidade inclui uma partilha da pasta onde se encontra a Vagrantfile (e onde a máquina virtual é montada), e o sistema operativo host, podendo assim sincronizar ficheiro entre as 2 máquinas.

Depois de uma primeira execução, e criação da máquina, de cada vez que é executado o script, este encarrega-se de atualizar na máquina as configurações que tenham sido alteradas.

O script de configuração utilizado por nós efetua as seguintes operações :

- Redirecionamento das portas :
 - 8080 para 8080 no host (servidor Web da aplicação)
 - 3306 para 3306 no host (acesso remoto ao MySQL)
 - 4200 para 4200 no host (utilizada pelo frontend)
 - 4201 para 4201 no host (utilizada pelo frontend)
 - 80 para 10000 no host (utilizada pelo NGinx – servidor web)
- Instalação do MySQL server
- Instalação do NodeJS
- Instalação do Python
- Instalação do yarn (alternativa ao npm do Node)
- Atualização das bibliotecas com o npm
- Instalação do Nginx

- Obtenção do script da Base de Dados a utilizar no projeto a partir o GIT e criação da mesma no MySQL
- Definição da password do utilizador 'root' do MySQL
- Configuração do bind-address do MySQL para permitir acessos remotos e assim ser possível manipular as bases de dados a partir do host com a Workbench

Comandos utilizados :

vagrant up: instala a máquina virtual se não existir, atualiza a existente se as configurações tiverem sido alteradas na Vagrantfile, ou inicializa a máquina

vagrant halt: parar a máquina virtual

vagrant destroy: elimina a máquina virtual

vagrant ssh: acesso à Shell de ubuntu da máquina virtual

Foram detetados alguns problemas na execução da máquina virtual em alguns dos postos dos membros das equipas, que foram sendo corrigidos e que em alguns casos se prendiam com as seguintes situações :

- Caminho no host para a pasta da máquina virtual, demasiado extenso ou com caracteres especiais
- Execução da linha de comandos de Windows sem ser em modo de administrador
- Incompatibilidades entre o File System de Linux e de Windows
- Versões do Vagrant instalado diferentes daquela em que o script tinha sido criado, e que não mostravam sinais de retro compatibilidade com alguns comandos

8. Testes

A realização de testes é importante para haver um controlo de qualidade de cada componente do projeto, portanto foram realizados diversos testes automáticos. Estes testes automáticos servem para a deteção de problemas anómalos ao sistema que possam contribuir negativamente para a execução de qualquer sistema.

Bastante comum num enquadramento de *Continuous Integration* (CI), esta prática de desenvolvimento requer que os desenvolvedores integrem diversos vezes ao dia código. Cada verificação é então verificada por um processo automático, permitindo às equipas a deteção de problemas com antecedência.

O objetivo principal de CI é evitar problemas de integração e é destinado a ser usado em combinação com testes unitários automatizados escritos através das práticas de Test Driven Development (TDD).

Testando as diferentes componentes do projeto, a realização destes testes tiveram como uso as seguintes tecnologias:

- Na sua generalidade o uso de Node.js e npm.
- API
 - SuperTest
 - Mocha
- Frontend
 - Protractor (Selenium)
- Base de Dados
 - MySQL - Queries, mysqltest, mysqlcheck

8.1. API

Realizados testes funcionais para o teste da API, esta foi testada enviando requests e verificando se os dados enviados para a API eram introduzidos na DB ou se eram propriamente avaliados.

Foi também verificado se a API enviava respostas com códigos corretos para as ações realizadas pela mesma.

A tecnologia usada foi o SuperTest e o Mocha.

8.2. Base de Dados

A integridade de uma base de dados é essencial para um bom funcionamento de um sistema e portanto foi realizado testes de *integridade* na base de dados, mais propriamente no dump, e nas suas tabelas e procedimentos.

Foi usado o mysqlcheck para testar a integridade do dump e a framework de testes do mysql, mysqltest, para o teste das tabelas e procedimentos. Isto foi possível com o uso em conjunto com o Node.js e diversos pacotes.

8.3. Frontend

A realização de testes no frontend requer o uso de diversas tecnologias.

Numa primeira fase do projeto foi usado o CasperJS para se realizar um teste de UI, ou como também é chamado, end-to-end. No entanto, o seu uso em ambientes é mais direcionado para ambientes estáticos e o projeto devido a usar Angular é dinâmico não deu para se usar.

A tecnologia atual usada para a realização de testes end-to-end foi o Protractor, e no fim houve também uma tentativa de se usar o Karma para a realização de testes unitários.

9. Desenvolvimentos Futuros

Futuramente, são consideradas várias hipóteses para serem implementados na continuação do projeto, entre elas:

- Notificações
- Horários/Locais disponíveis
- Upload/Download de ficheiros
- Criar projetos
- Permitir enviar notificações/mensagens, após análise das candidaturas, atribuição de projetos, informações necessárias ao projeto
- Os alunos registados deveriam receber notificações/mensagens com a lista de projetos disponíveis e projetos atribuídos pelo Docente após efectuada a análise às candidaturas
- Os locais, datas e horário disponível para reunião entre alunos e docentes responsáveis também chegariam ao respectivo grupo por mensagem
- Página personalizada para cada projeto
- Projetos criados por alunos
- Upload ou envio de assets para o projeto
- Chat

10. Conclusões

Neste projeto abordámos a criação de uma plataforma de Projetos Finais de uma determinada licenciatura e concluímos que se apostarmos nas diversas tecnologias recentes que se usam nos dias de hoje e se trabalharmos bem em equipa, tudo é possível.

Cumprimos maior parte dos objetivos que nos tínhamos proposto, porém alguns ficaram por realizar devido ao curto espaço de tempo que tínhamos disponível.

Este projeto foi muito importante para o nosso aprofundamento em Sistemas de Informação, uma vez que ficámos a conhecer muitas tecnologias que não aprendemos durante a licenciatura em Engenharia Informática, além de nos ter permitido aperfeiçoar competências de investigação, seleção, organização, comunicação da informação e sobretudo trabalho em equipa.

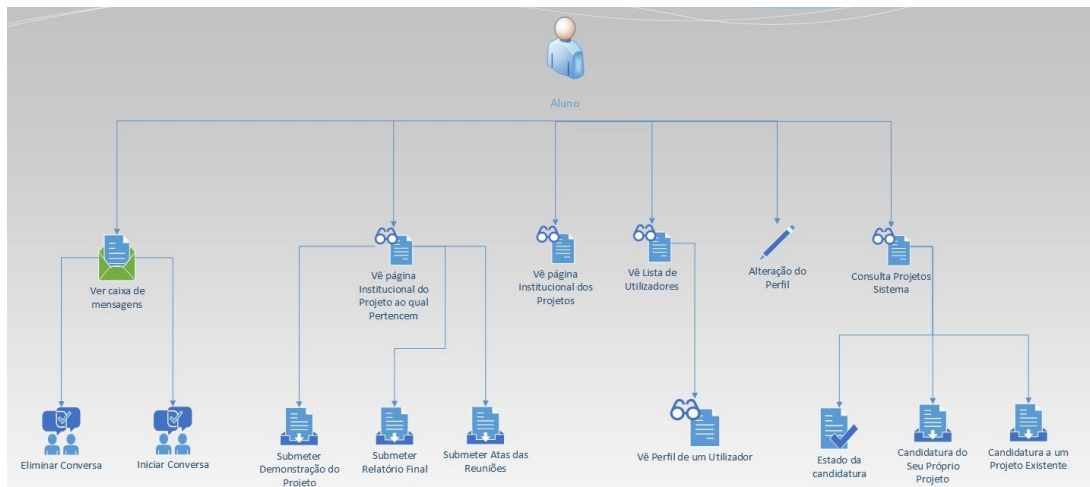
11. Bibliografia

- Estado da Arte:
 - <https://portal.ufsm.br/projetos/>
 - <https://www.upwork.com>
 - <https://peerlibrary.org>
 - <https://github.com/peerlibrary/>
- GitHub iptomar:
 - <https://github.com/iptomar/>
 - https://github.com/iptomar/projectary-docs/blob/master/EstadoDaArte_relatorio.pdf
 - https://github.com/iptomar/projectary-docs/blob/master/EstadoDaArte_apresentacao.pdf
 - <https://github.com/iptomar/projectary-as>
 - <https://github.com/iptomar/projectary-bd>
 - <https://github.com/iptomar/projectary-frontend>
 - <https://github.com/iptomar/projectary-api>
 - <https://github.com/iptomar/projectary-tests>
 - <https://github.com/iptomar/projectary-as/tree/master/Levantamento%20de%20Requisitos%20Projectary>
 - <https://github.com/iptomar/projectary-as/tree/master/Use%20Cases>
 - <https://github.com/iptomar/projectary-as/tree/master/Diagramas%20de%20Atividades>
 - <https://github.com/iptomar/projectary-as/tree/master/Diagramas%20de%20Sequencia>
- Tecnologias:
 - <https://golang.org>
 - <https://www.mysql.com>
 - <https://angular.io>
 - <https://nodejs.org>
 - <https://expressjs.com>
 - <https://www.vagrantup.com>
 - <https://dev.mysql.com/doc/mysqltest/2.0/en/mysqltest.html>
 - <https://dev.mysql.com/doc/refman/5.7/en/mysqlcheck.html>
 - <https://github.com/visionmedia/supertest>
 - <http://www.seleniumhq.org>
 - <http://www.protractortest.org>
 - <https://karma-runner.github.io/1.0/index.html>

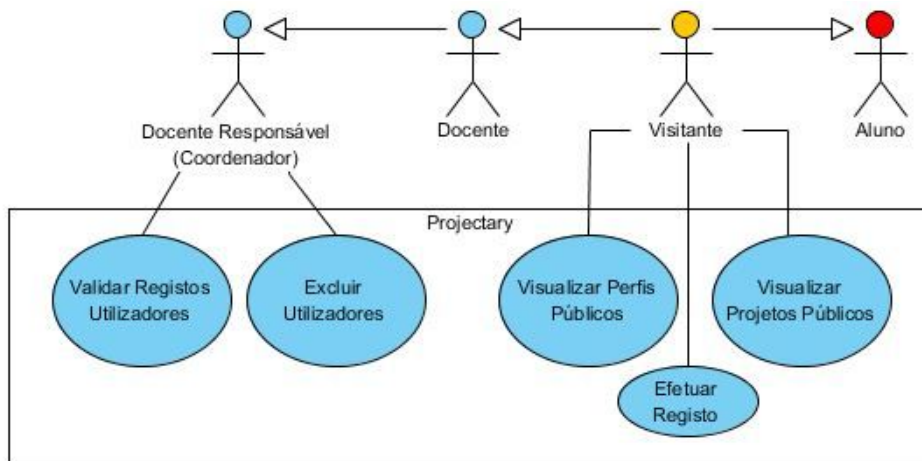
12. Anexo A - Diagramas Análise de Sistemas

12.1. Use Cases

12.1.1. Aluno

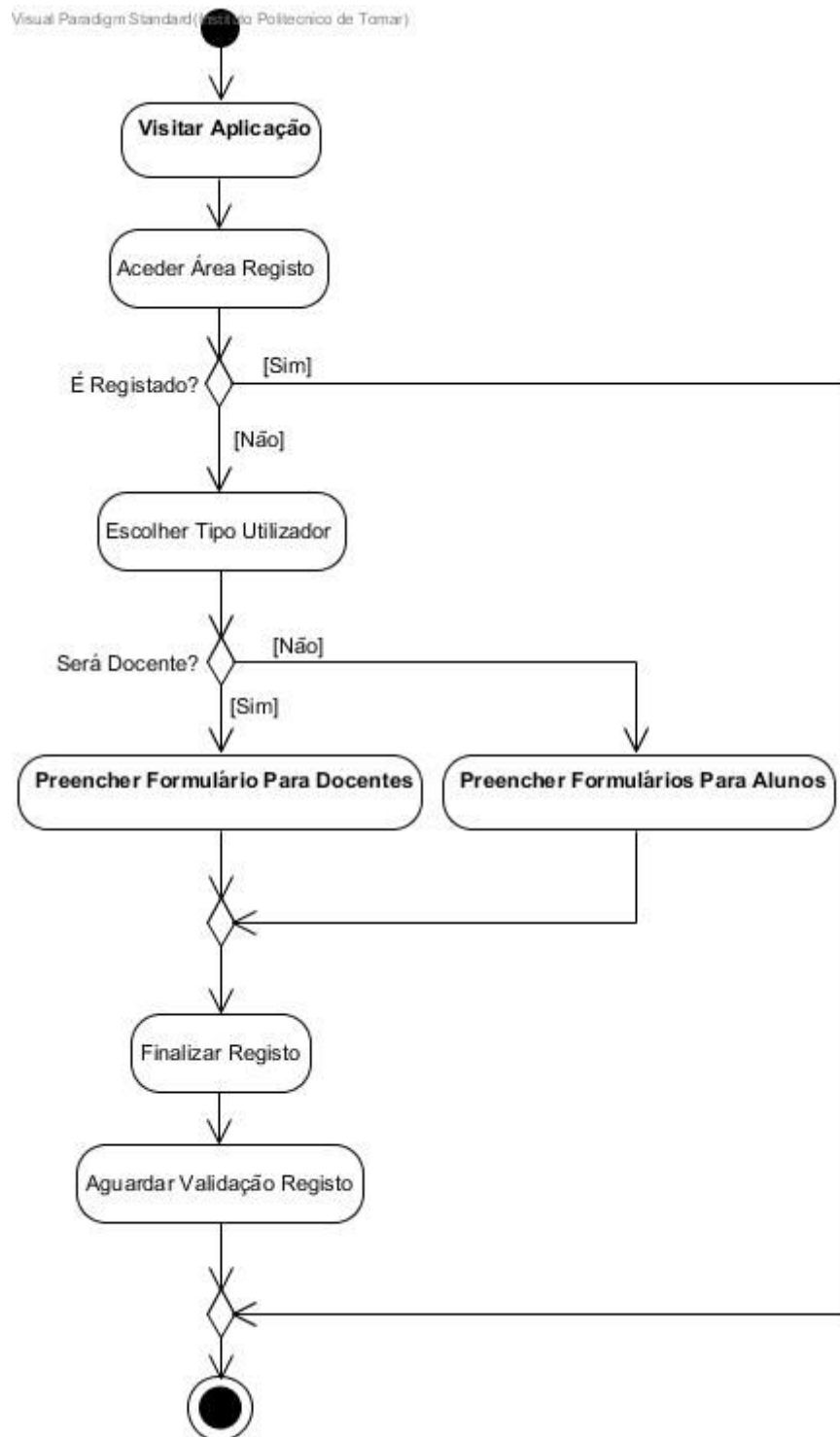


12.1.2. Docente

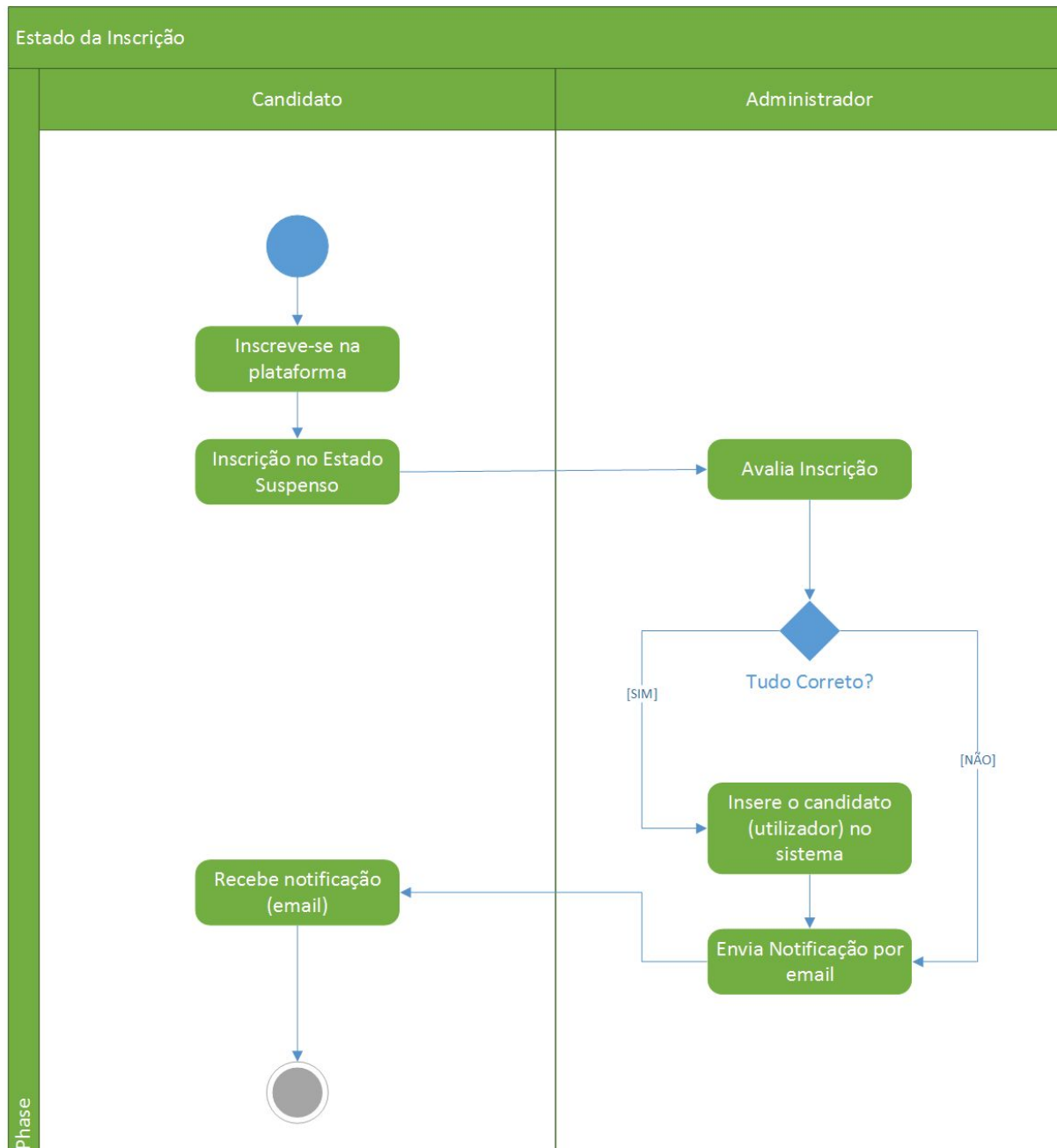


12.2. Diagramas de Atividades

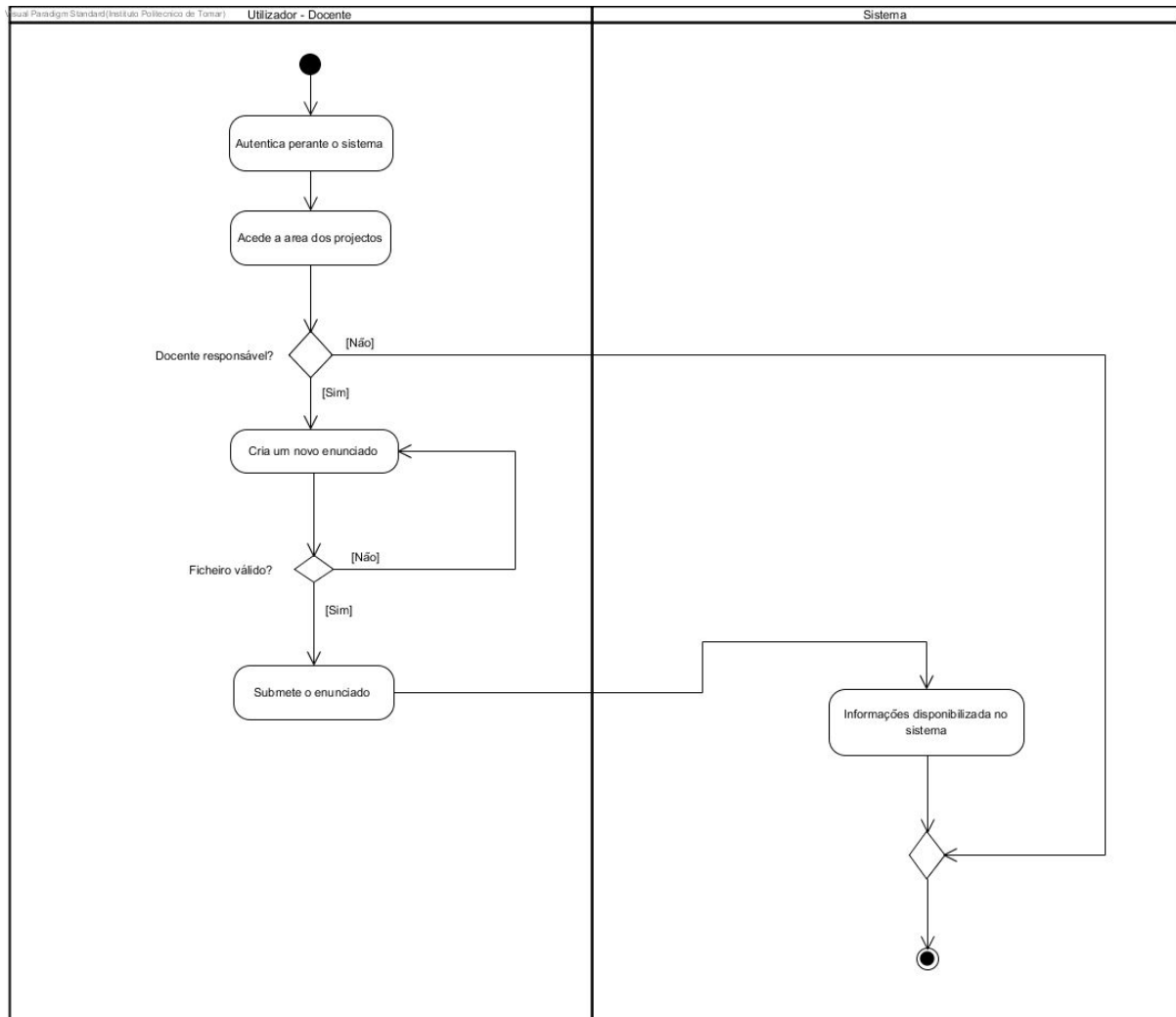
12.2.1. Registo na plataforma



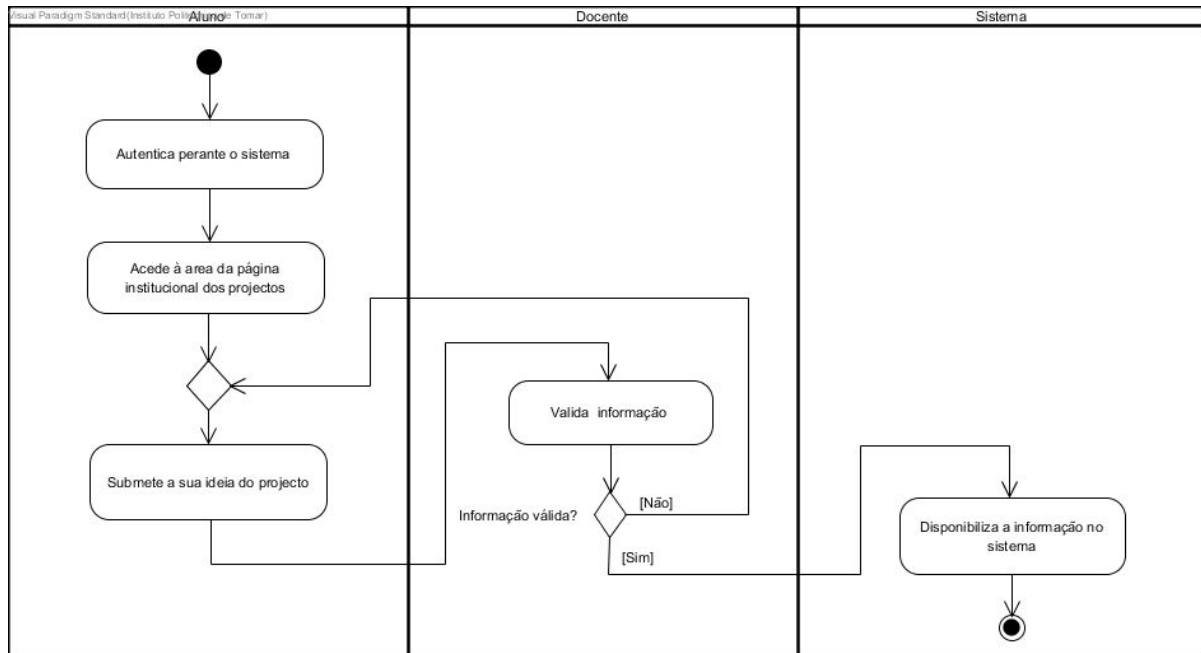
12.2.2. Estado da inscrição



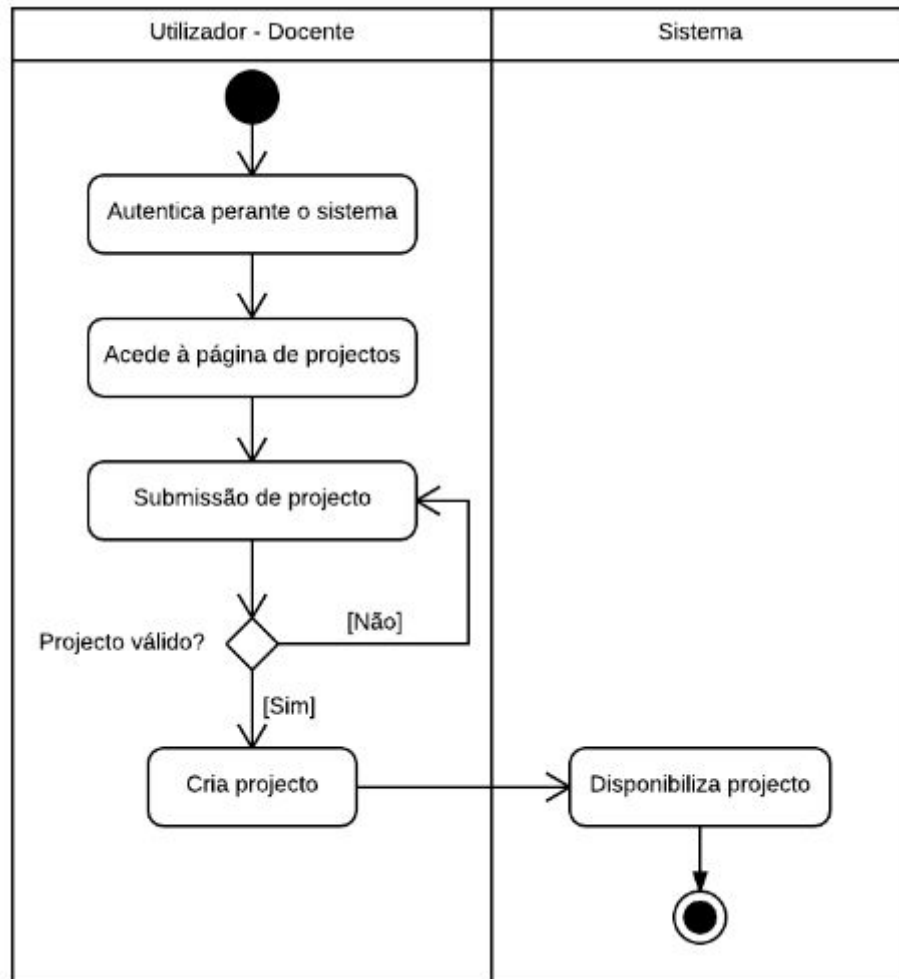
12.2.3. Criar enunciado para os projetos



12.2.4. Submeter ideia do projeto

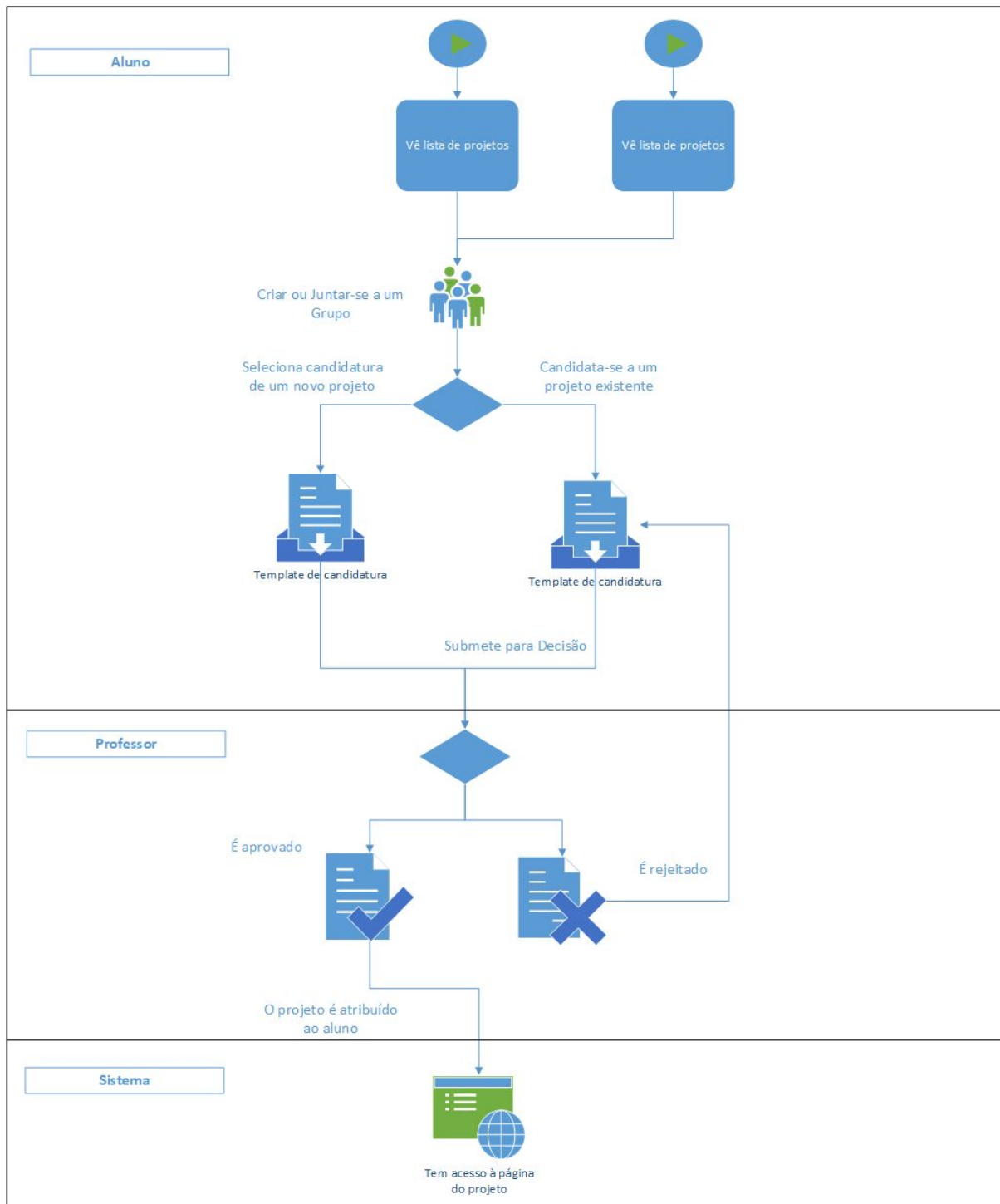


12.2.5. Disponibilização de projetos na plataforma

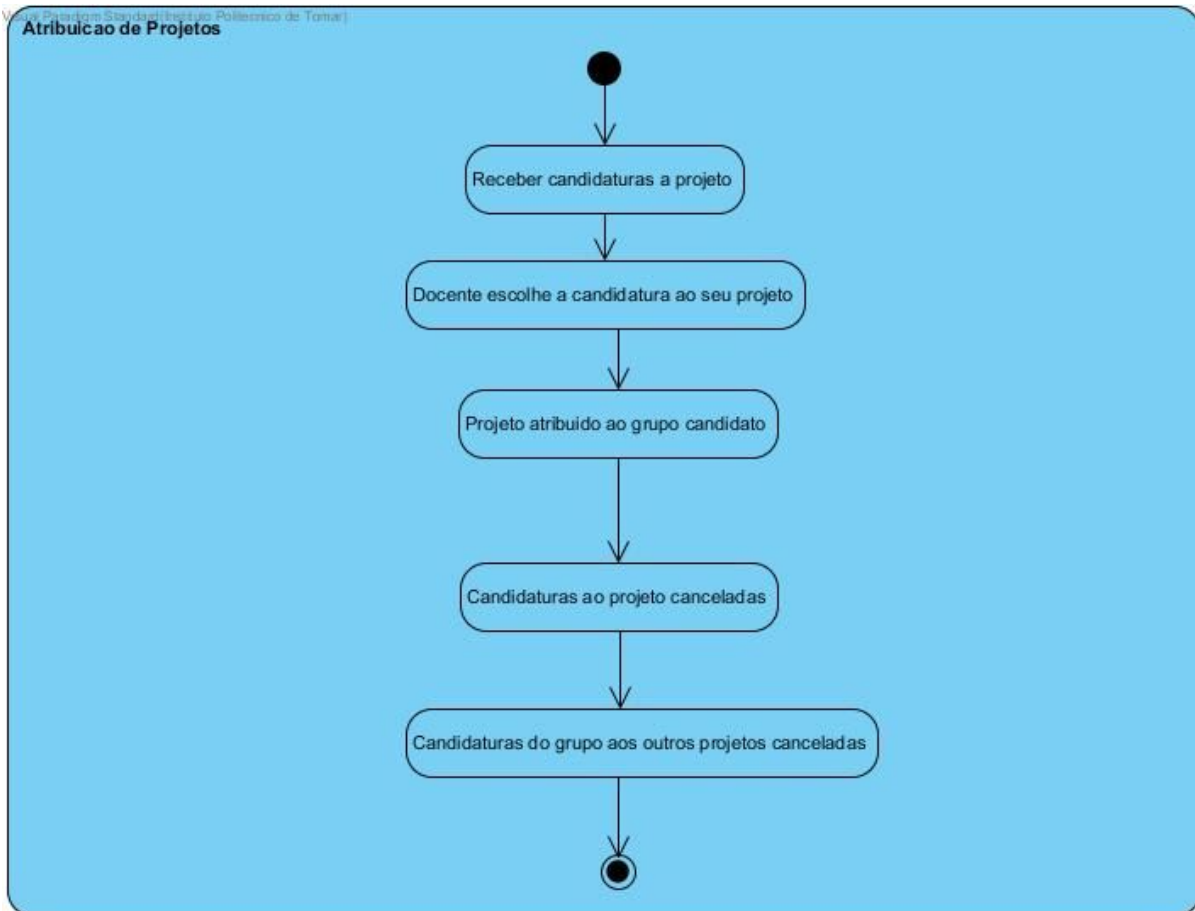


12.2.6. Candidatura

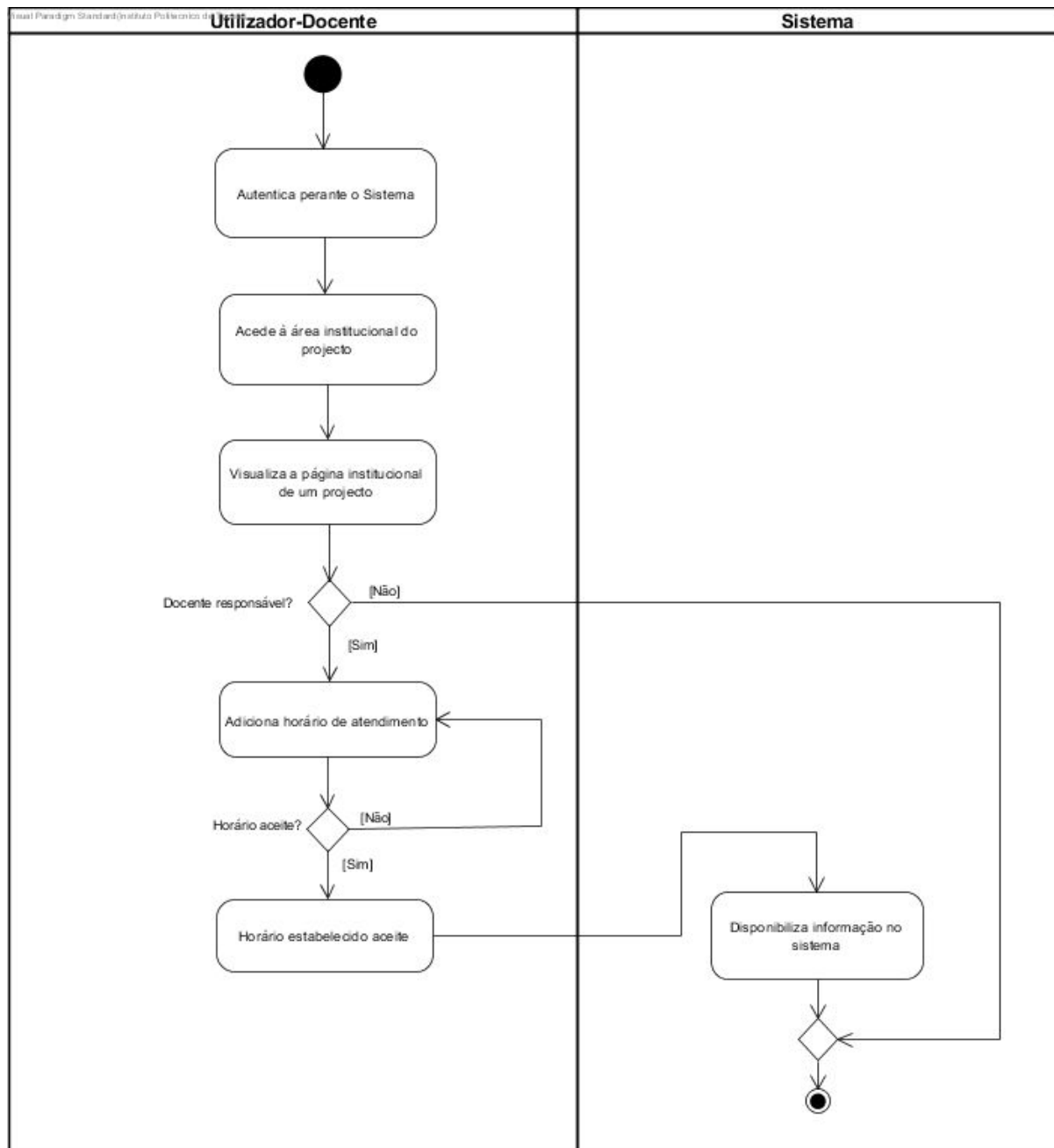
O aluno tem de estar logado para se poder juntar ou criar um grupo, apenas um elemento do grupo submete a candidatura



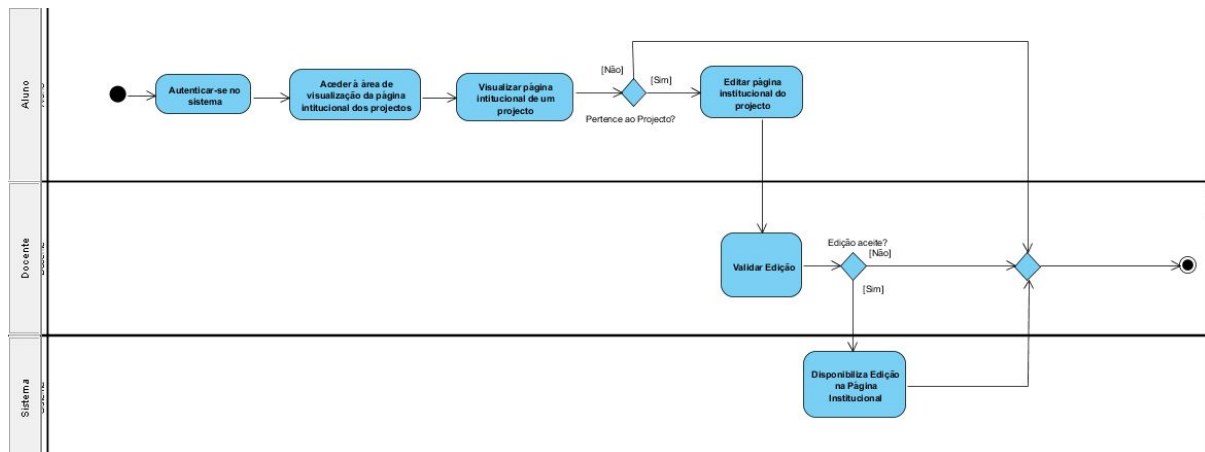
12.2.7. Atribuição de projetos



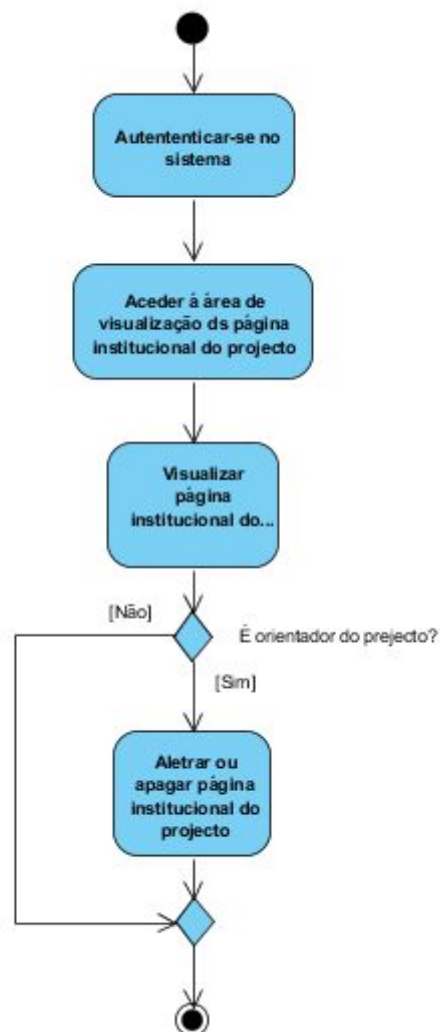
12.2.8. Estabelecer horário de atendimento



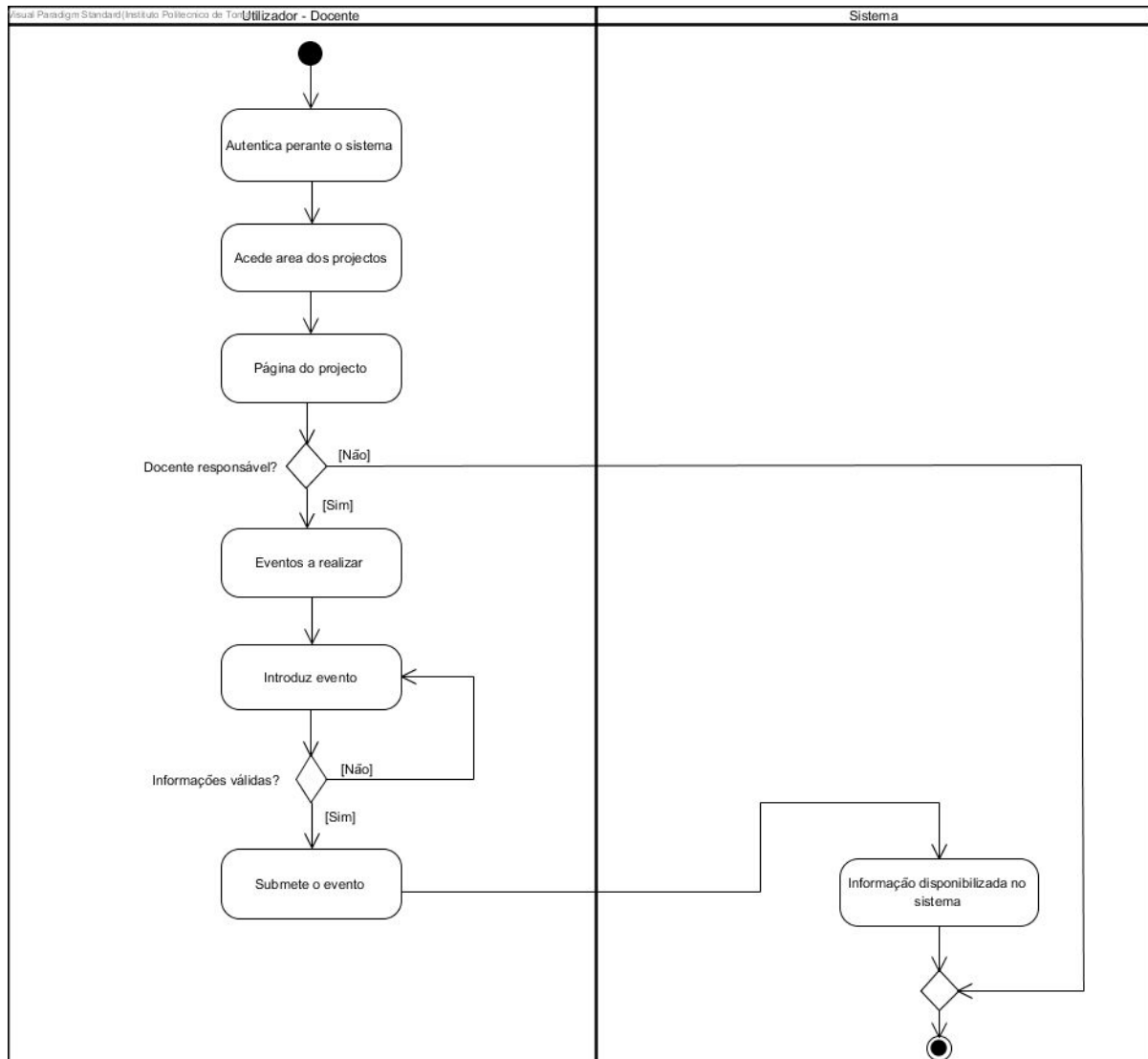
12.2.9. Edição da página institucional por parte do alunos



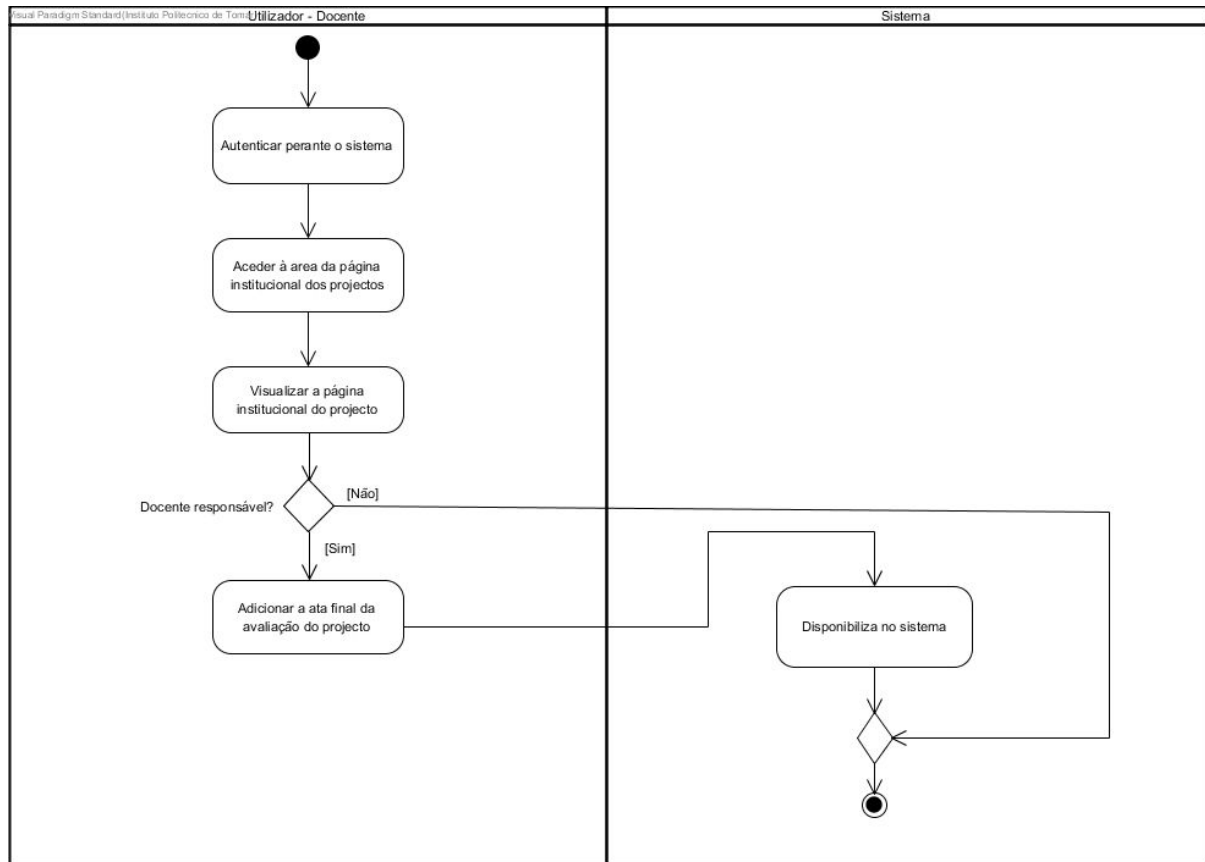
12.2.10. Edição da página institucional por parte dos docentes



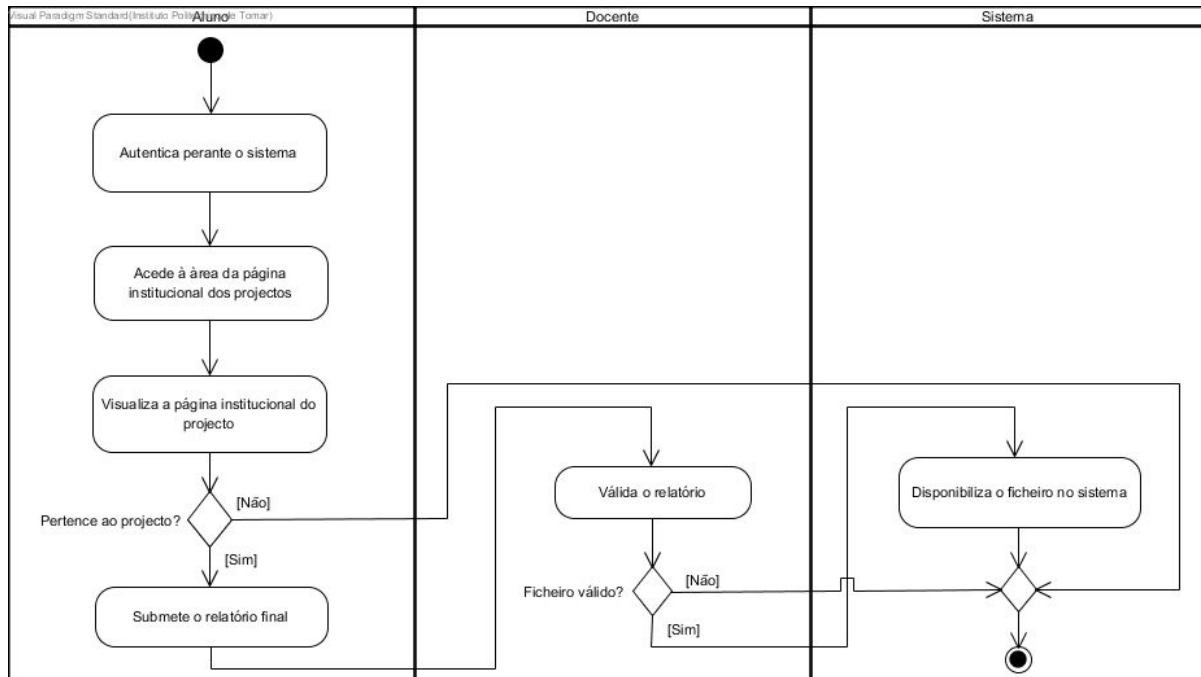
12.2.11. Disponibilização horário reuniões



12.2.12. Disponibilização ata final da Avaliação

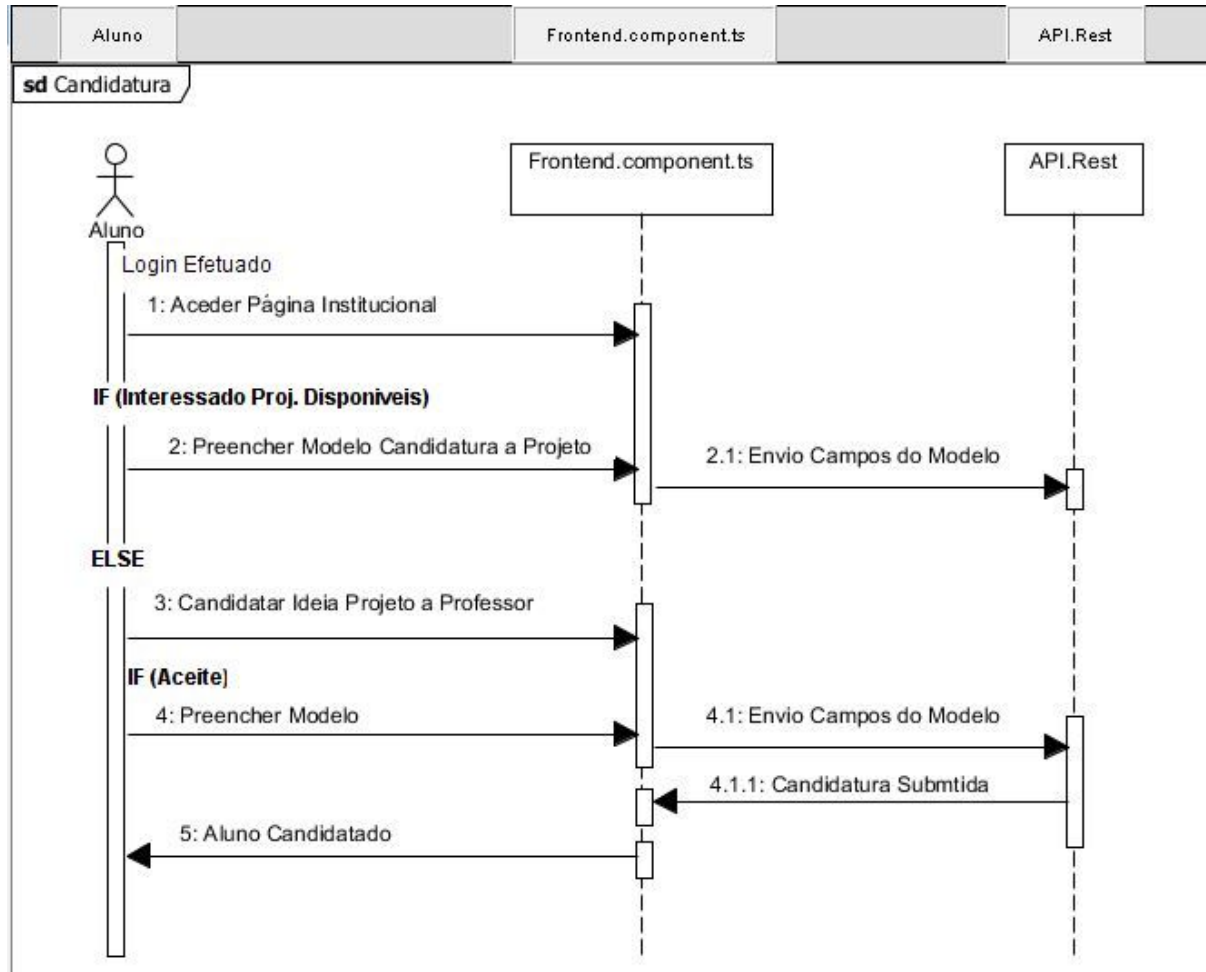


12.2.14. Submeter relatório final

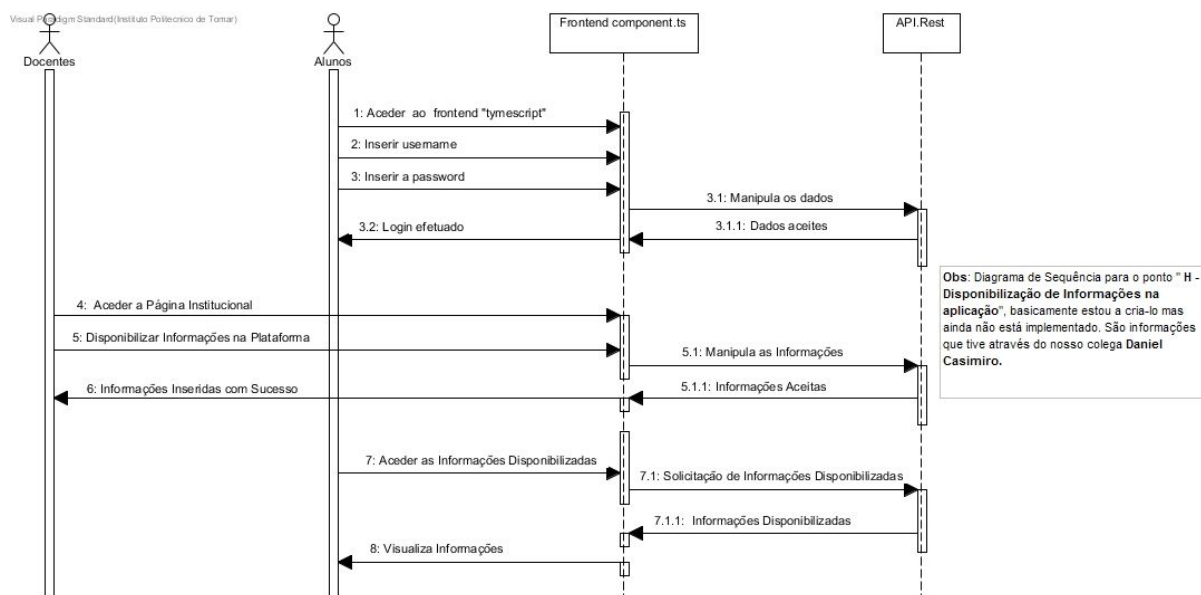


12.3. Diagramas de Sequência

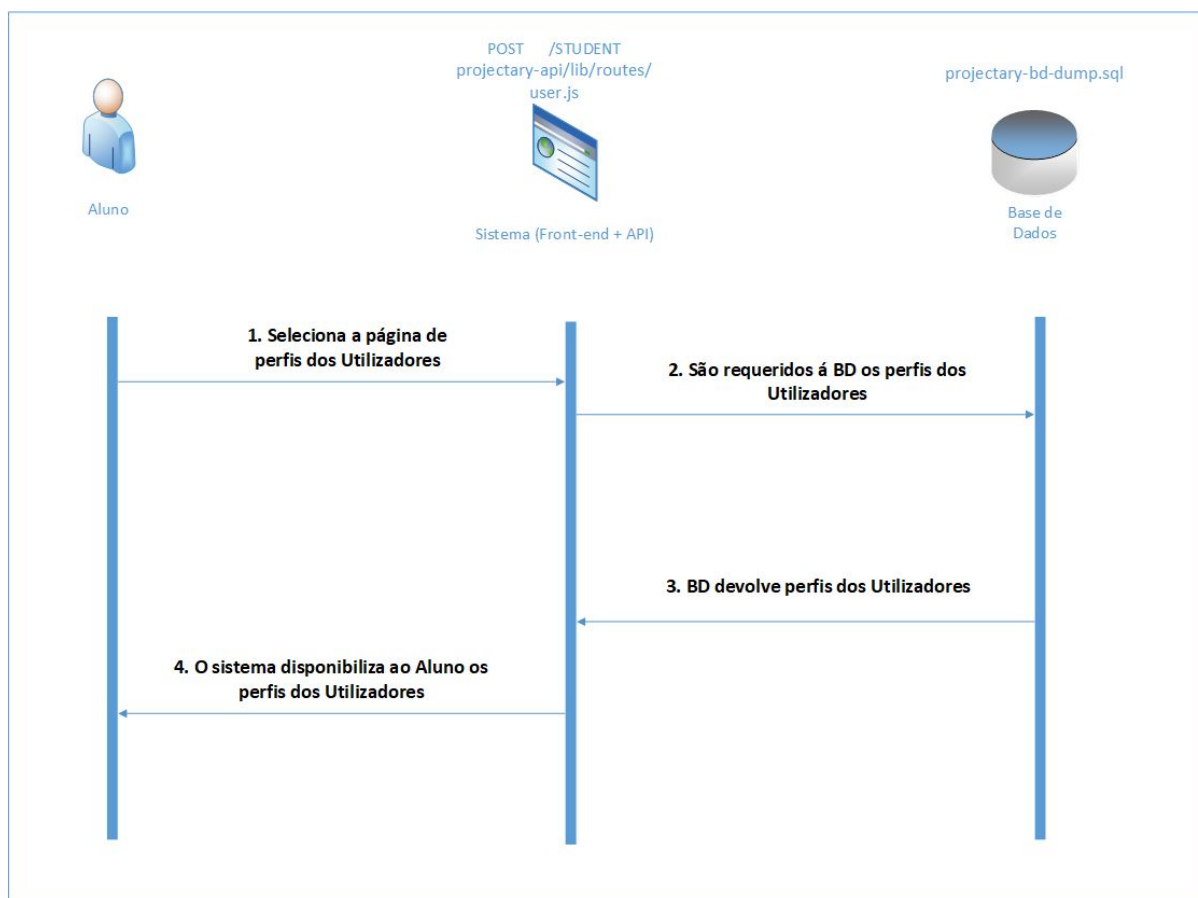
12.3.1. Candidatura Projetos



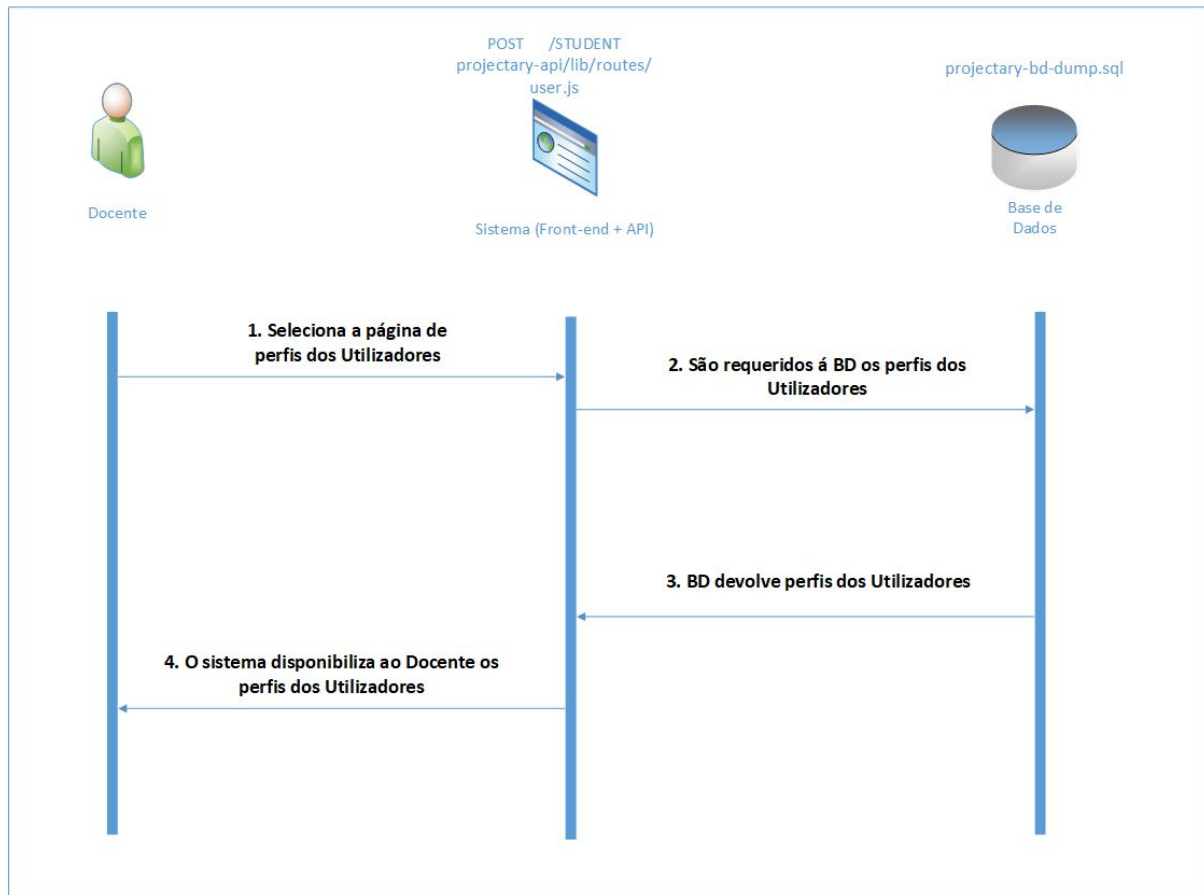
12.3.2. Disponibilização de informações



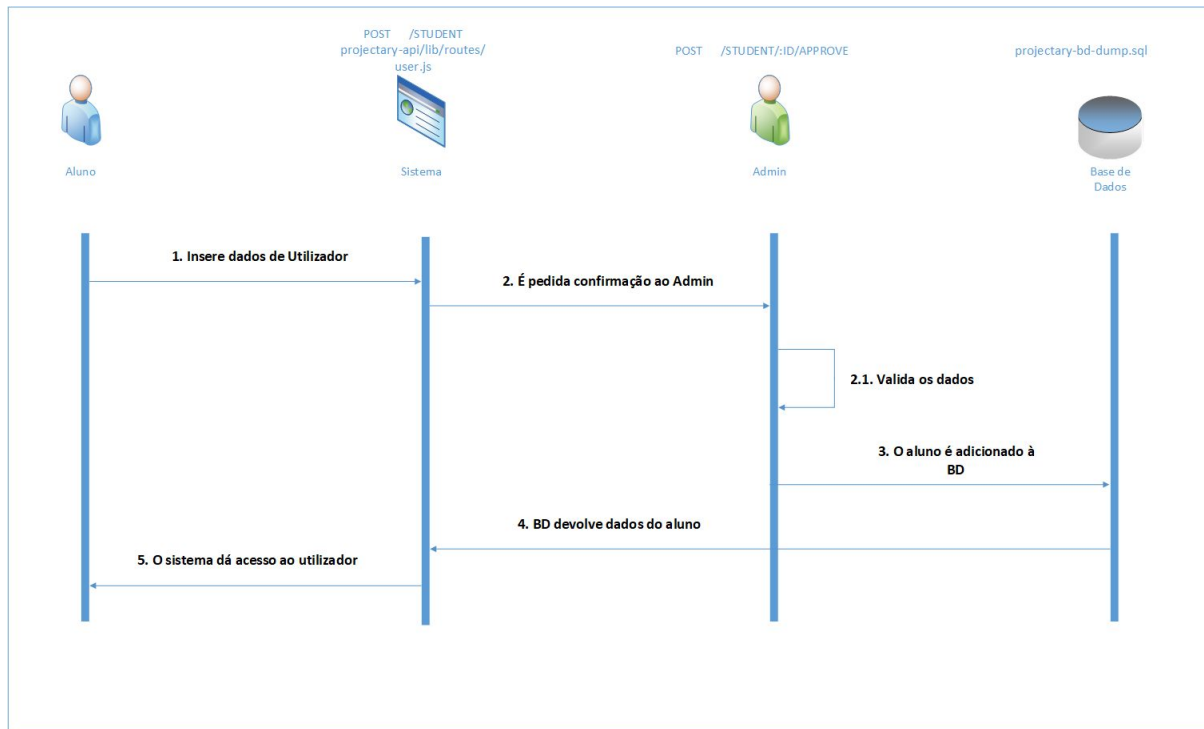
12.3.3. Lista de utilizadores p/ aluno



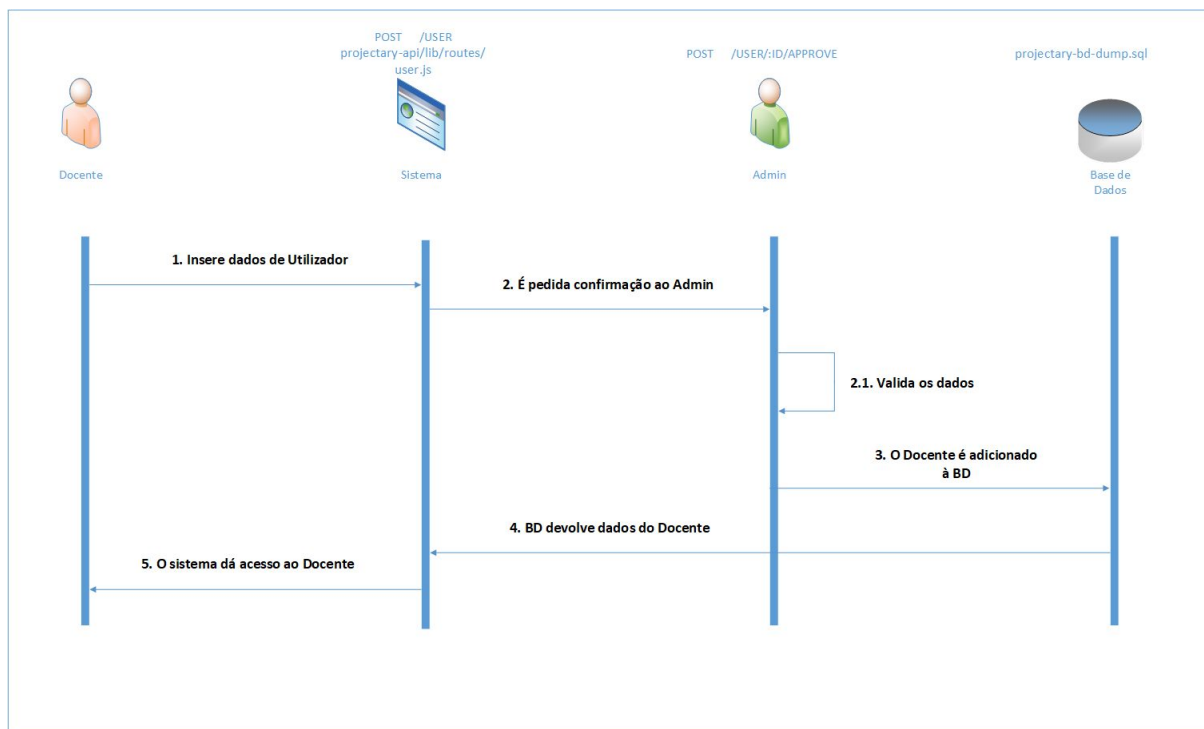
12.3.4. Lista de alunos p/ docente



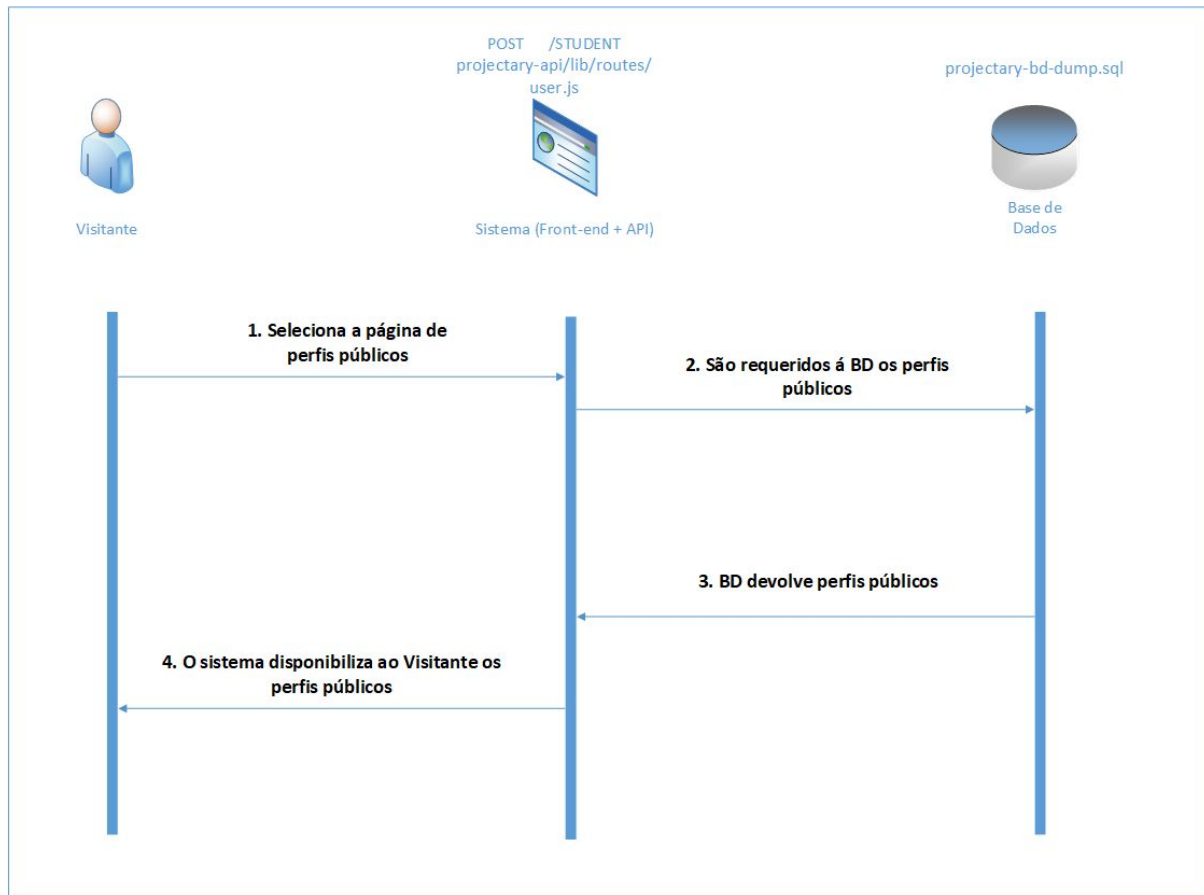
12.3.5. Registo aluno



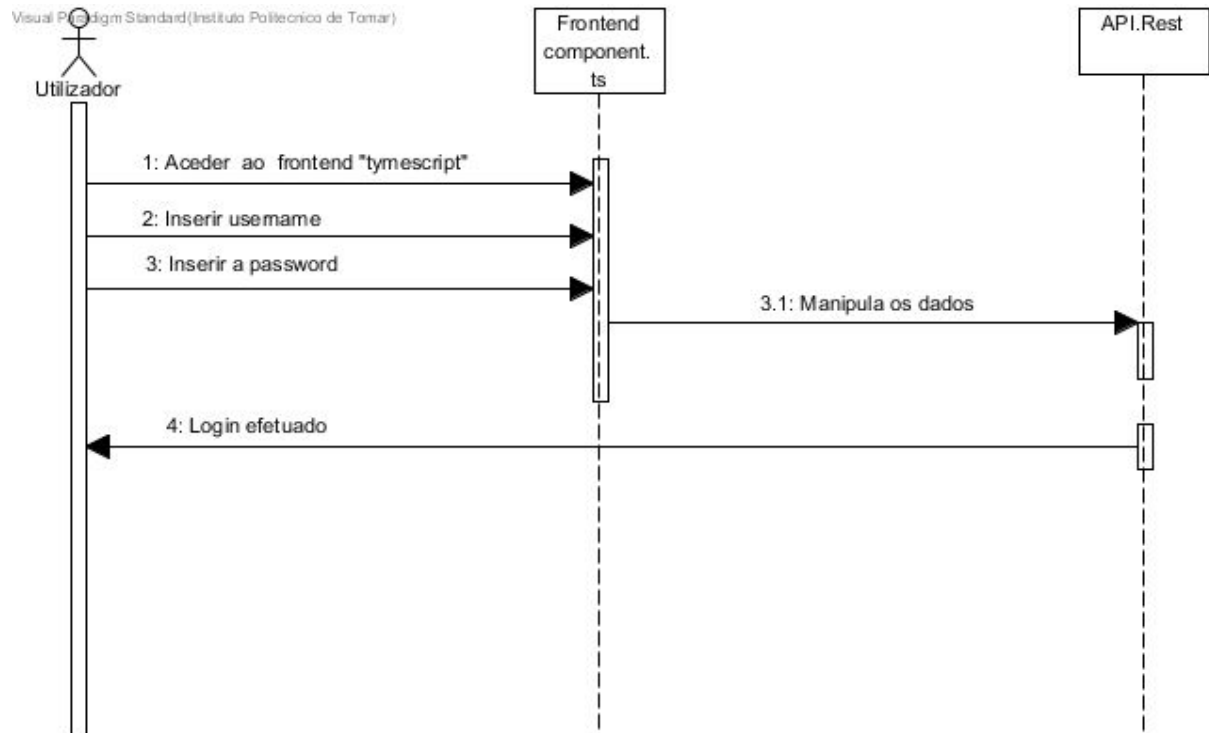
12.3.6. Registo docente



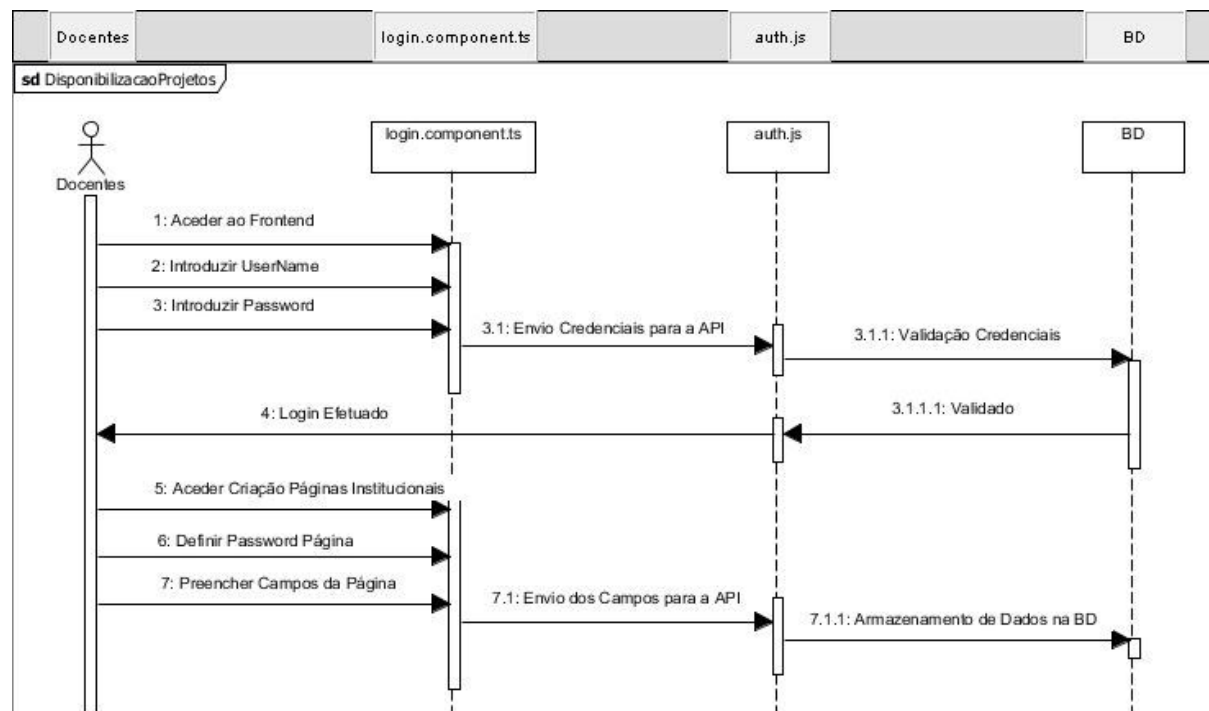
12.3.7. Visualizar perfis públicos



12.3.8. Login do aluno

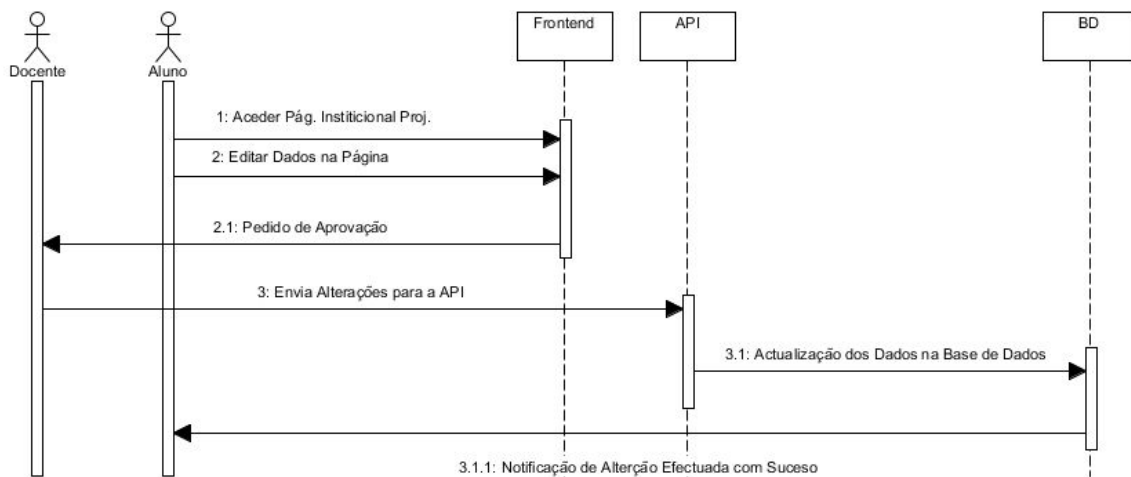


12.3.9. Disponibilização dos projetos



12.3.10. Edição da página institucional p/ alunos

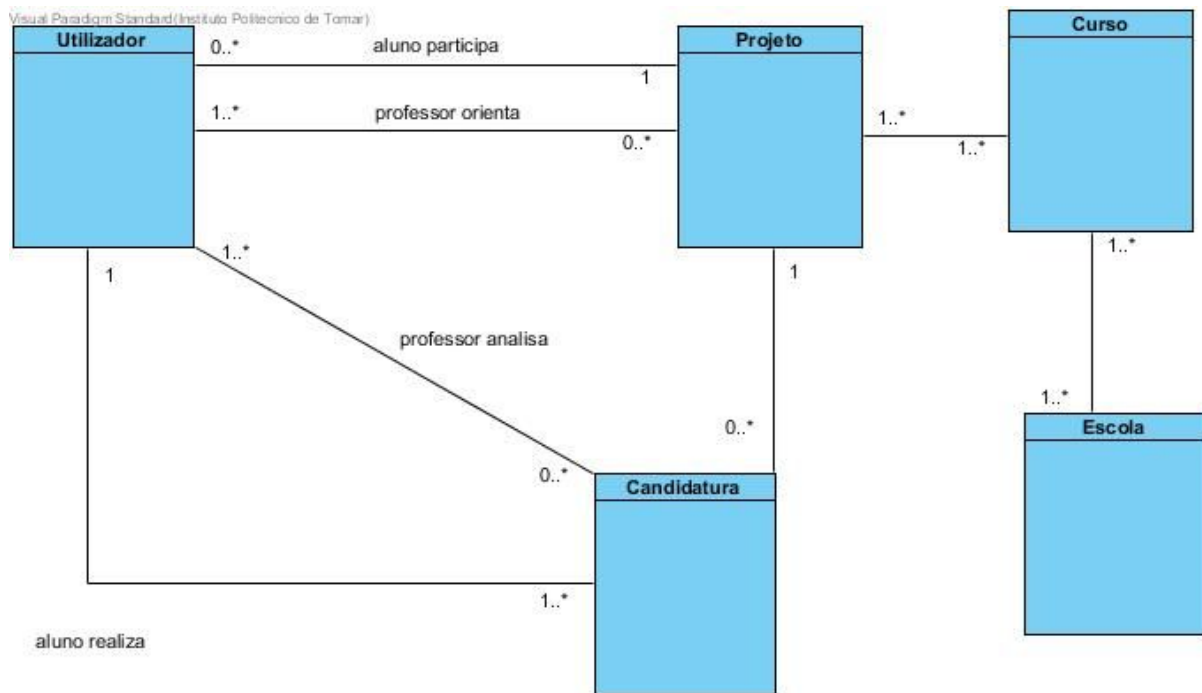
sd Edição Pág Inticional Alunos



12.3.11. Edição da página institucional p/ docentes



12.4. Modelo de Dados



13. Anexo B - Levantamento de Requisitos

13.1. Funcionais

Identificação e caracterização dos utilizadores do sistema:

- Docente: utilizador que atribui, cria, avalia e disponibiliza projetos; consulta de todos os perfis dos utilizadores e projetos registados na aplicação; estabelecer horário de atendimento para o projeto; alteração do seu perfil; cria enunciados para os projetos; troca de mensagens pessoais com utilizadores;
- Aluno: utilizador que se candidata aos projetos existentes ou submete a sua ideia de projeto para avaliação; consulta dos perfis de todos os utilizadores e projetos registados na aplicação; troca de mensagens pessoais com utilizadores; criação de perfil e sua alteração; submissão pública ou privada do relatório final e demonstração do projeto na aplicação;
- Docente Responsável: utilizador que tem acesso a todas as funcionalidades dos docentes e é responsável pela cadeira de Projeto Final de cada licenciatura; aprova ou rejeita os registos dos utilizadores candidatos ao sistema; excluir utilizadores;
- Visitante: utilizador que visualiza os projetos e perfis públicos na aplicação, regista-se no sistema;

Requisitos:

A. O sistema possibilita a criação de novos utilizadores, dos seguintes tipos:

- a. Docente
- b. Aluno
- c. Docente Responsável
- d. Visitante (consideram-se todos os utilizadores que não se encontram registados na aplicação)

B. O sistema possibilita aos visitantes as seguintes funções:

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

C. Registo na aplicação

- a. No registo da aplicação, o utilizador deverá preencher um formulário relacionado com o tipo de utilizador;
- b. Apenas o administrador poderá criar utilizadores do tipo docentes, onde deverá preencher um formulário com os seguintes campos:
 - i. Nome Completo
 - ii. Password
 - iii. Foto de Perfil
 - iv. Link Pessoal Institucional
 - v. Unidade Departamental

- vi. Email
- vii. Áreas Funcionais
- viii. Perfil público/privado
- c. Caso pretenda ser Aluno, o utilizador deverá preencher o formulário com os seguintes campos:
 - i. Nome Completo
 - ii. Password
 - iii. Número de Aluno
 - iv. Foto de Perfil
 - v. Escola a que pertence
 - vi. Cursos
 - vii. Email
 - viii. Áreas de Interesse
 - ix. Perfil público/privado

D. Estado da inscrição

- a. Pendente para aprovação
 - i. As inscrições aguardam a aprovação de um administrador (estado suspenso)
- b. Aceitação / Reprovação no sistema
 - i. Depois de avaliado, o candidato receberá uma notificação por email com o resultado da avaliação do seu registo;
 - ii. Caso seja aceite, o administrador deverá inserir no sistema esse utilizador

E. Disponibilização dos projetos na aplicação

- a. Docentes:
 - i. Criação da página institucional do projeto, de acordo com o seguinte modelo:
 - 1. Título
 - 2. Enunciado
 - 3. Campo de submissão de candidaturas ao projeto
 - 4. Estado do projeto
 - a. Por atribuir
 - b. Atribuído
 - c. Cancelado
 - 5. Campo de submissão das atas de reunião
 - 6. Campo de submissão do relatório final
 - 7. Campo de submissão da demonstração
 - 8. Campo de submissão da ata final de avaliação do projeto
 - ii. Os campos de submissão referidos no ponto i) deverão estar protegidos por uma password definida pelo docente ao criar a página institucional;

F. Candidatura

- a. Os alunos poderão candidatar-se a projetos disponibilizados na aplicação;
 - i. Caso o aluno não esteja interessado em nenhum projeto disponibilizado ou tenha uma ideia, o aluno deverá candidatar a sua ideia a um professor responsável;
- b. Para se candidatar ao projeto o aluno deverá criar um grupo com uma determinada password e preenchendo os seguintes campos, referentes ao grupo:
 - i. Nome do grupo;
 - ii. Password;
- c. Após a criação do grupo, o aluno que criou o grupo deverá utilizar a página institucional para submeter a sua candidatura, utilizando um modelo disponibilizado pela aplicação com os seguintes tópicos:
 - i. Escola
 - ii. Licenciatura
 - iii. Ano Letivo
 - iv. Constituição do grupo
 - 1. Número de aluno
 - 2. Nome
 - 3. Crédito ECTS
 - 4. Média das disciplinas
 - v. Ordem de preferência dos projetos:
 - 1. Título do projeto
 - vi. Local e data da submissão deste documento
 - vii. Grupo de alunos candidatos
 - 1. Rúbricas
- d. Os alunos pertencentes ao grupo que foi criado para a submissão da candidatura deverão juntar-se ao grupo utilizando a password escolhida para o efeito.
- e. Cada aluno, inscrito na aplicação, só poderá ter um grupo durante o decorrer do ano letivo.
- f. Para fazer a candidatura a projetos, qualquer elemento do grupo poderá submeter a candidatura, pelo grupo.
- g. Um aluno ingresso num grupo tem a possibilidade de abandonar o grupo, porém, se for o único elemento do grupo o grupo será eliminado.
- h. O aluno ingresso num grupo, poderá eliminar o grupo, apenas se o grupo só ser constituído por ele.

G. Atribuição de projetos

- a. O administrador e docente deverão ter acesso a todas as candidaturas aos projetos em que está envolvido;
- b. O docente, depois de analisar as candidaturas, irá atribuir um projeto aos alunos requerentes, enviando uma notificação por email a esses alunos e

alterando o estado na página institucional para o estado a que se encontra;

- c. Ao ser atribuído a um projeto, todas as outras candidaturas (se existirem), às quais o aluno pertence, são canceladas;
- d. Quando um projeto é atribuído, todas as outras candidaturas a esse mesmo projeto são canceladas;

H. Disponibilização de informações na aplicação

- a. O docente administrador do sistema deverá ter um local onde irá disponibilizar informações para os utilizadores da plataforma;
- b. Os alunos deverão ter acesso às datas e notas disponibilizadas pelo docente responsável;
- c. Os docentes deverão ter um local onde irão disponibilizar o horário das reuniões a serem realizadas;

I. Edição da página institucional

- a. Docentes:
 - i. Deverão ter permissões de visualização a todos os projetos, edição e remoção relativamente à página institucional do projeto que orienta;
- b. Alunos:
 - i. Deverão ter permissões de visualização sobre as páginas institucionais;
 - ii. Deverão ter permissões de edição sobre os campos da página institucional, do projeto ao qual pertencem. Estas edições só serão validadas depois de aceites pelo docente;

J. O sistema deverá possibilitar aos docentes as seguintes funções:

- a. Criação da página institucional do projeto, de acordo com o seguinte template:
 - i. Enunciado
 - ii. Campo de submissão das atas de reunião
 - iii. Campo de submissão do relatório final
 - iv. Campo de submissão da demonstração
 - v. Campo de submissão da ata final de avaliação do projeto
- b. Alteração da página institucional do projeto
- c. Eliminação da página institucional do projeto
- d. Disponibilizar enunciado na página institucional do projeto
- e. Avaliar candidaturas dos alunos de temas para o Projeto Final
- f. Alteração do perfil pessoal
- g. Listar todos os alunos do sistema
- h. Consulta de todos os perfis dos utilizadores do sistema
- i. Consulta de todos os projetos do sistema
- j. Estabelecer horário de atendimento para o projeto

- k. Alterar horário de atendimento para o projeto
- l. Tem acesso a um template de enunciados de Projeto Final, com as seguintes características:
 - i. Escola
 - ii. Licenciatura
 - iii. Ano lectivo
 - iv. Título
 - v. Sumário
 - vi. Número de alunos a envolver
 - vii. Objetivos específicos
 - viii. Pré-requisitos
 - ix. Orientador(es)
- m. Disponibilização da ata final da avaliação de projeto na página institucional do projeto

K. O sistema deverá possibilitar aos alunos as seguintes funções:

- a. Alteração do perfil pessoal
 - b. Consulta dos perfis dos utilizadores
 - c. Consulta dos projetos do sistema
 - d. Candidatura a um projeto existente segundo o seguinte template:
 - i. Escola
 - ii. Licenciatura
 - iii. Ano letivo
 - iv. Constituição do grupo
 - 1. Número de aluno
 - 2. Nome
 - 3. Crédito ECTS
 - 4. Média das disciplinas
 - v. Ordem de preferência dos projetos:
 - 1. Título do projeto
 - vi. Local
 - vii. Data da submissão deste documento
 - viii. Grupo de alunos candidatos
 - 1. Rúbricas
- e. Candidatura do seu próprio projeto segundo o seguinte template:
 - i. Escola
 - ii. Licenciatura
 - iii. Ano letivo
 - iv. Título do projeto
 - v. Sumário
 - vi. Número de alunos a envolver
 - vii. Objetivos específicos
 - viii. Pré-requisitos
 - ix. Orientador(es)
 - x. Proponentes
 - 1. Rúbricas

f. Submeter as atas de reunião na página institucional com o seguinte template:

- i. Escola
 - ii. Licenciatura
 - iii. Número de ata
 - iv. Data
 - v. Nome do projeto
 - vi. Pessoas presentes na reunião
1. Nome
 2. Rúbrica
- vii. Documentos da reunião
 - viii. Ordem de trabalho
 - ix. Ata da reunião

g. Submissão do relatório final na página institucional com o seguinte template:

- i. Escola
 - ii. Título do projeto
 - iii. Ano letivo
 - iv. Número do(s) aluno(s)
 - v. Nome do(s) aluno(s)
 - vi. Licenciatura
 - vii. Título do projeto
 - viii. Número do(s) aluno(s)
 - ix. Nome do(s) aluno(s)
 - x. Mês de apresentação
 - xi. Ano de apresentação
 - xii. Orientador(es)
1. Nome(s)
- xiii. Dedicatória
 - xiv. Agradecimentos
 - xv. Resumo
 - xvi. Palavras Chave
 - xvii. Índice
 - xviii. Introdução
 - xix. Enquadramento do Projeto
 - xx. Trabalho de Projeto
 - xxi. Conclusões
 - xxii. Bibliografia
 - xxiii. Anexos:
1. Enunciado do projeto
 2. Atas de reunião de projeto
 3. Outros anexos

h. Submissão da demonstração do projeto na página institucional

DEFINIÇÃO DE PRIORIDADES:

Primeira Fase (Desenvolvimento)	Requisitos: <ul style="list-style-type: none"> · A (todas as alíneas); · B (alínea a),b)); · C (todas as alíneas); · D (todas as alíneas); · E (todas as alíneas); · J (alíneas a),b),c),d),i),l); · K (alínea c));
Segunda Fase (Conclusão)	Requisitos: <ul style="list-style-type: none"> · F (todas as alíneas); · G (todas as alíneas); · I (todas as alíneas); · J (alínea e),g),m)); · K (alínea d),e),f),g),h));
Terceira Fase (Extras)	Requisitos: <ul style="list-style-type: none"> · B (alínea c)); · H (todas as alíneas); · J (alíneas f),h),i),j),k)); · K (alíneas a),b));

NOTA: as alíneas têm a denotação de numAlínea).

13.2. Não-Funcionais

- A. O fundo da interface inicial deve ser composto pelo logo da plataforma e uma imagem elucidativa à mesma como background;
- B. O sistema deverá correr em ambiente web;
- C. Deverá ser usado um design responsivo nas interfaces gráficas;
- D. O sistema deverá ser compatível com os diversos browsers que existem de momento no mercado, como por exemplo Chrome, Firefox e Edge;
- E. A base de dados do sistema deverá ser bem estruturada e eficiente;
- F. O sistema deverá oferecer métodos flexíveis que lidam com situações diferentes e inesperadas;
- G. A privacidade e segurança da plataforma são duas das grandes prioridades da plataforma;
- H. O design da interface deverá ser simples e apelativo;
- I. Ao entrar no 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;
 - d. uma barra de navegação que permita navegar nas diferentes componentes de sistema (por predefinição, a componente inicial do website é a componente Início):
 - i. Início
 - 1. Lista com todas as últimas publicações sobre a plataforma e notificações relevantes
 - ii. Projetos Públicos
 - 1. Permite visualizar todos os projetos públicos inseridos na plataforma
 - iii. Perfis Públicos
 - 1. Permite visualizar todos os projetos públicos inseridos na plataforma
 - iv. Sobre
 - 1. Mais informações acerca da plataforma
 - v. Contactos
 - 1. Informação sobre os diferentes modos de contactar com a plataforma
- J. Na página de registo:
 - a. Registo para alunos, deverá aparecer um formulário com os seguintes campos:
 - i. Nome Completo
 - ii. Username

- iii. Password
 - iv. Número de Aluno
 - v. Foto de Perfil
 - vi. Escola a que pertence
 - vii. Cursos
 - viii. Email
 - ix. Áreas de Interesse
 - x. Perfil público/privado
- b. Registo para docentes (este registo é apenas visível/possível para o administrador da plataforma), deverá aparecer um formulário com os seguintes campos:
- i. Nome Completo
 - ii. Username
 - iii. Password
 - iv. Foto de Perfil
 - v. Link Pessoal Institucional
 - vi. Unidade Departamental
 - vii. Email
 - viii. Áreas Funcionais
 - ix. Perfil público/privado

14. Anexo C - Screenshots da Aplicação

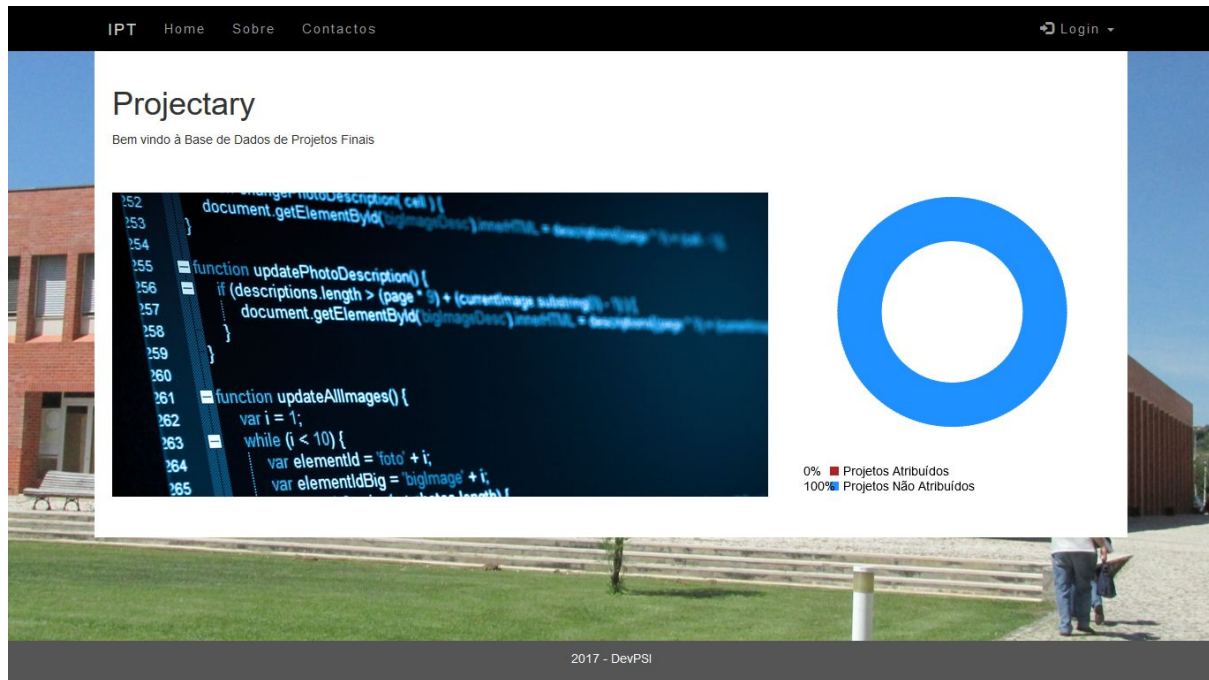


Figura 14.1 - Página inicial, sem autenticação efetuada.

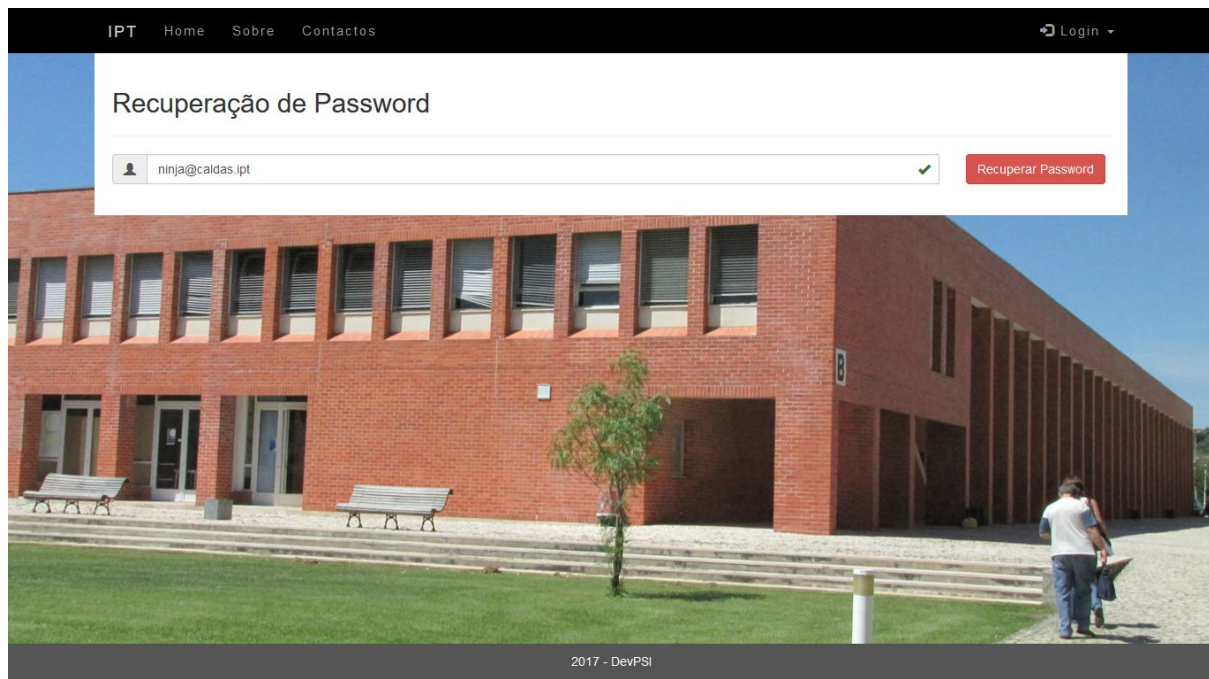


Figura 14.2 - Página de recuperação de password.

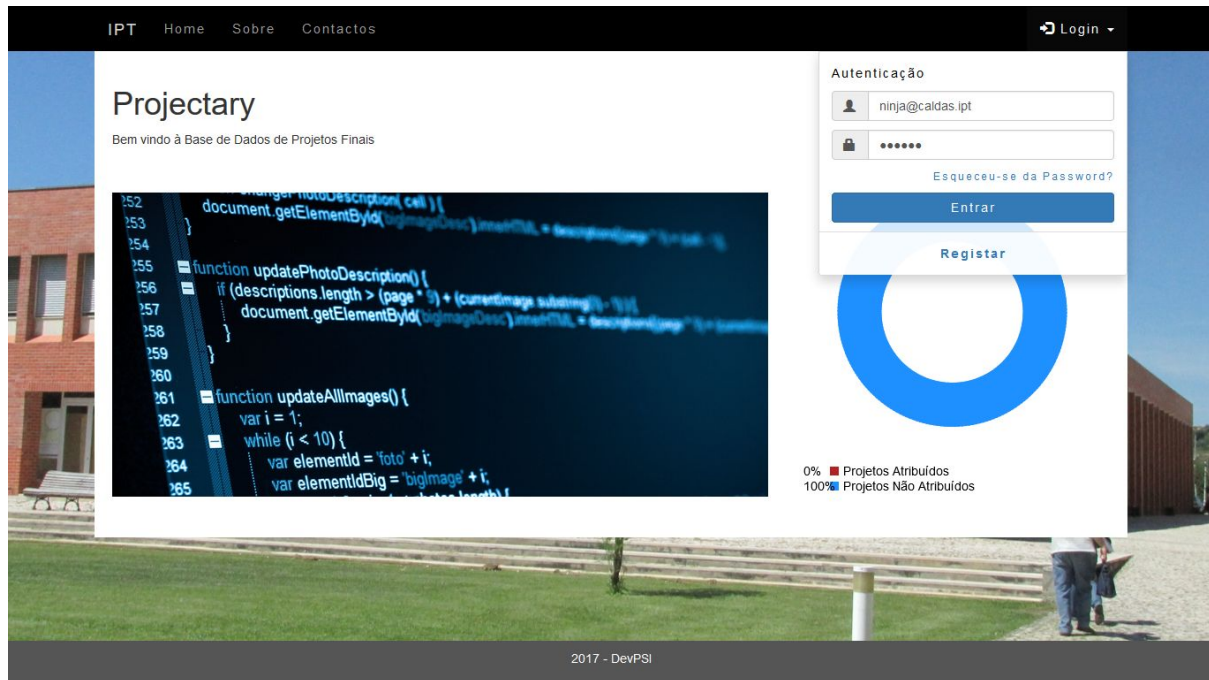


Figura 14.3 - Página de inicial ao se clicar no link “Login”.

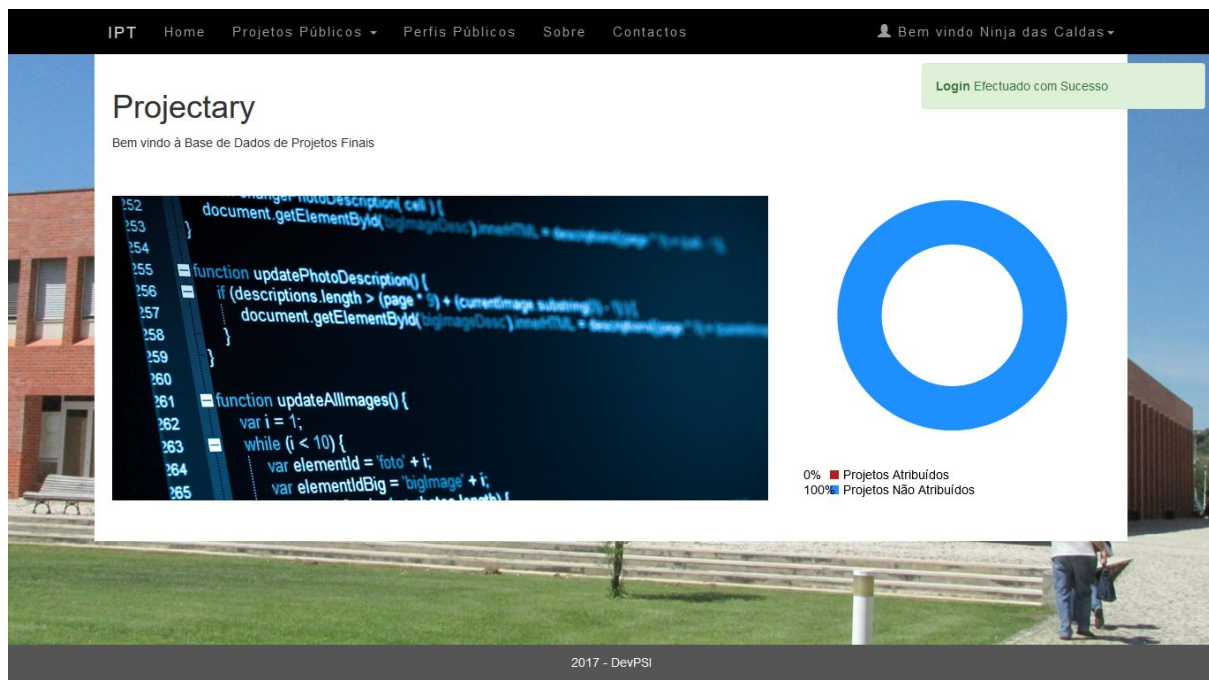


Figura 14.4 - Página de inicial com autenticação efetuada.

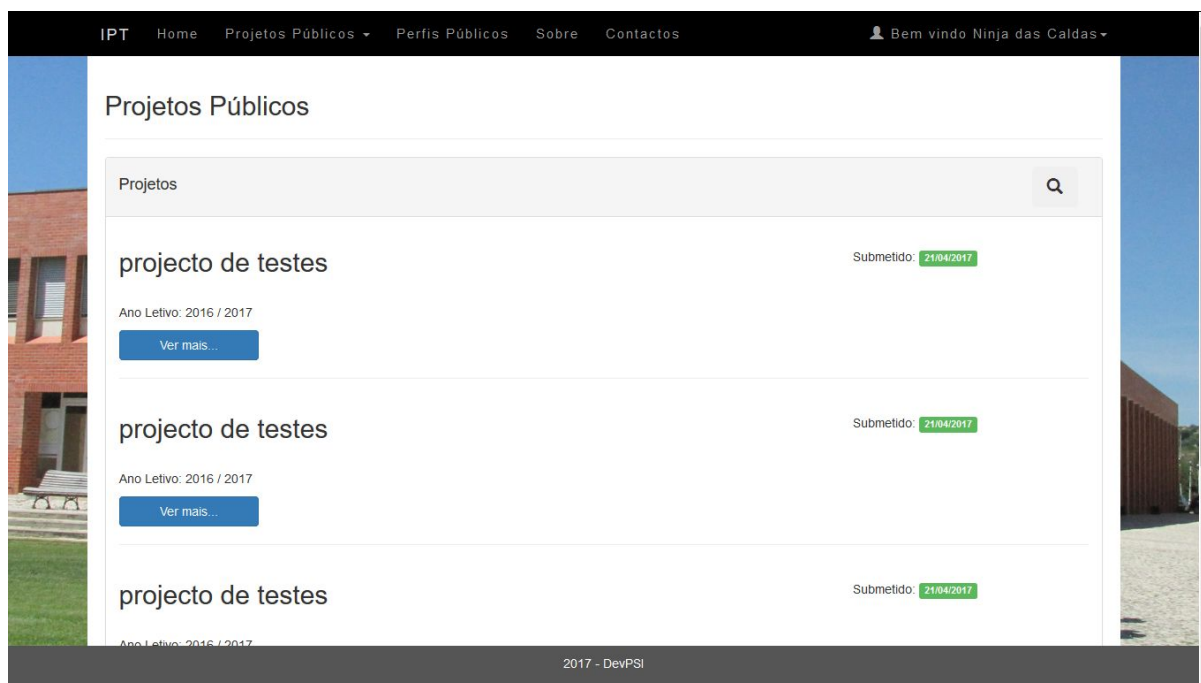


Figura 14.5 - Página de listagem de projetos.



Figura 14.6 - Página de listagem de um projeto.

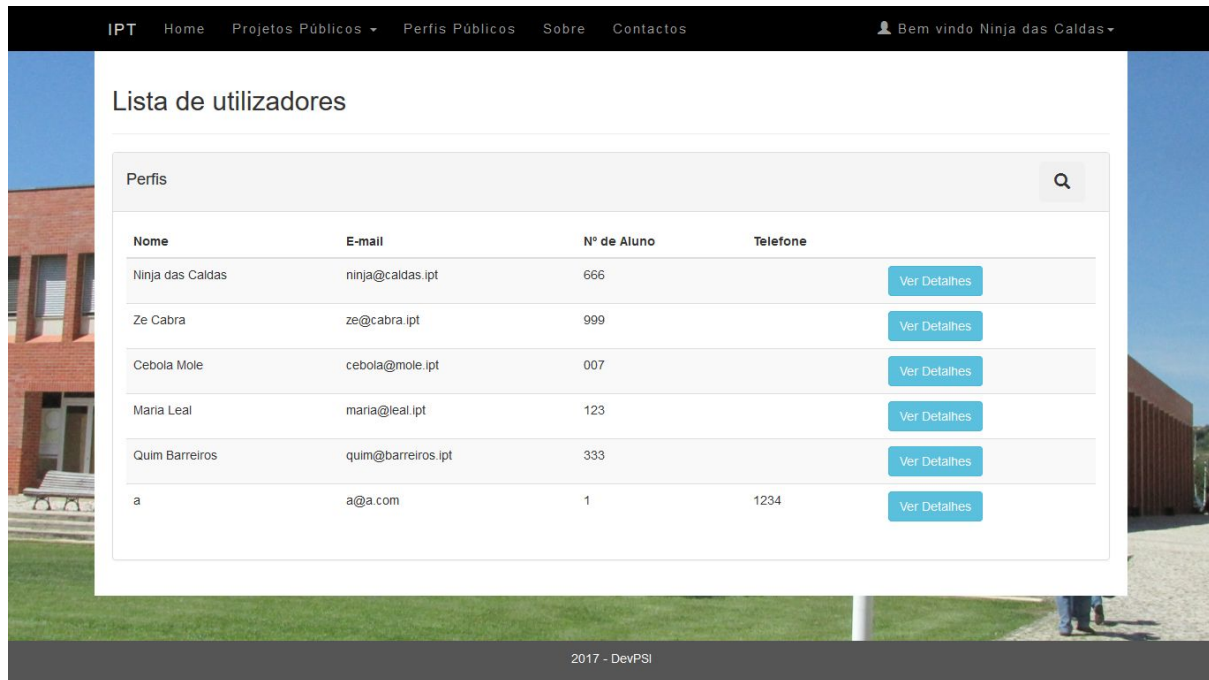


Figura 14.7 - Página de perfis públicos.

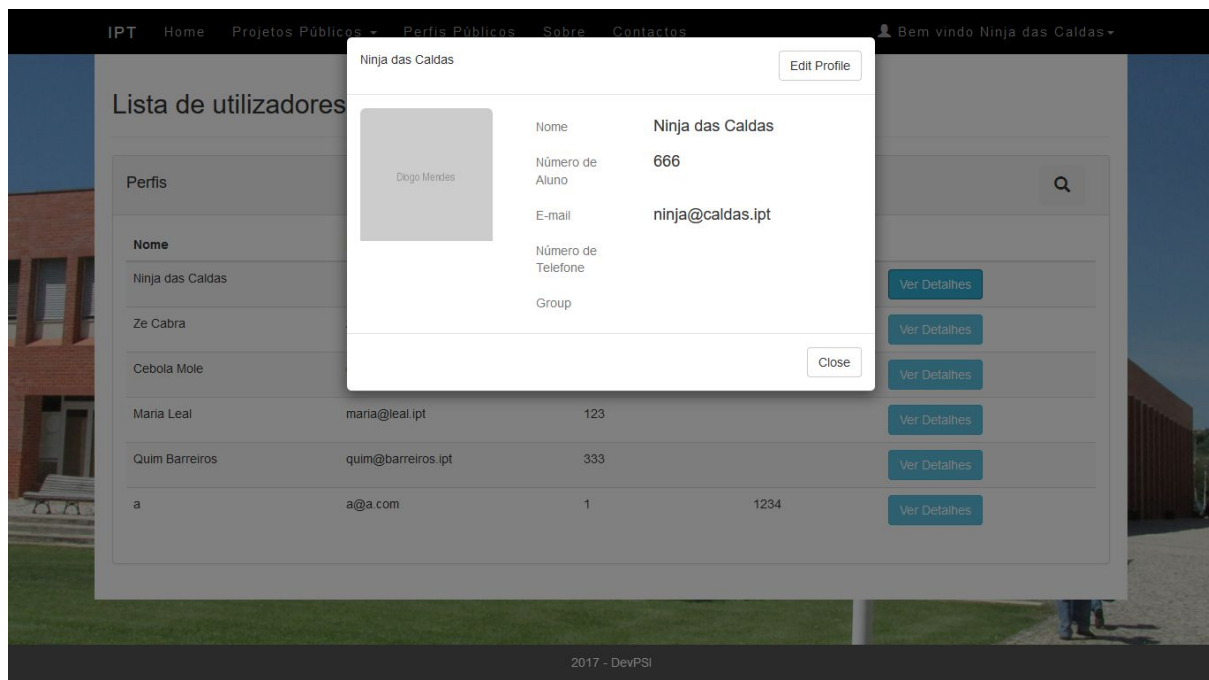


Figura 14.8 - Página de perfis públicos quando se clica no botão “Ver Detalhes” de um perfil.

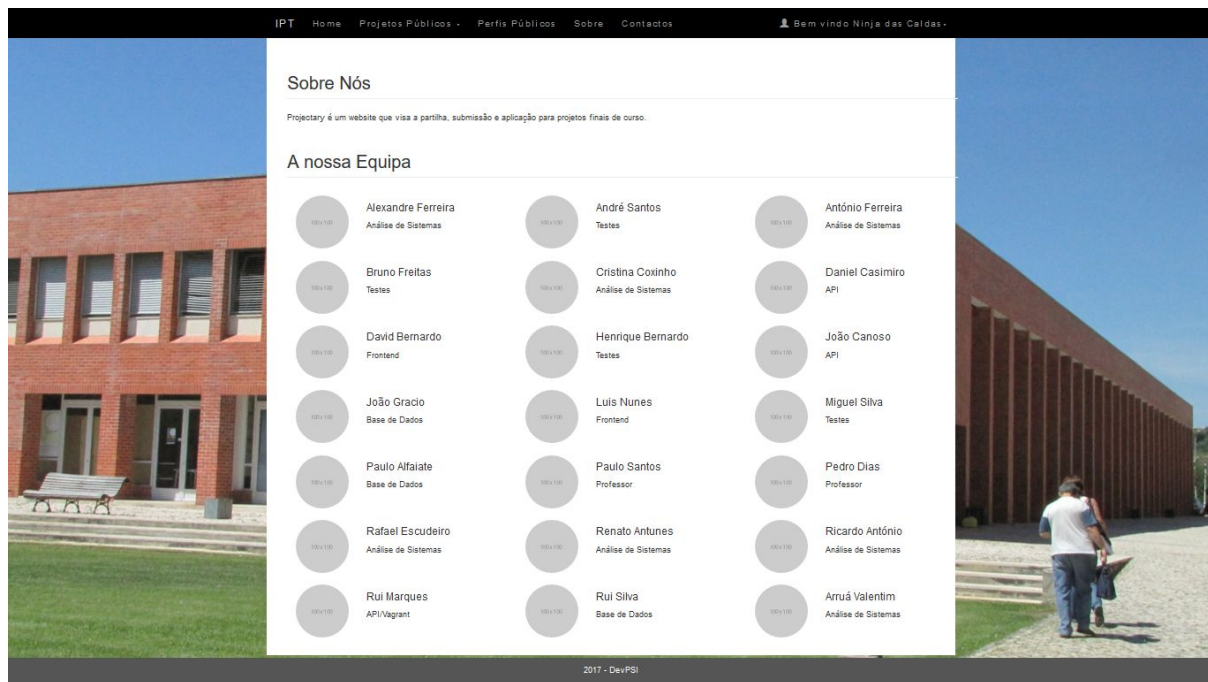


Figura 14.9 - Página “Sobre”.

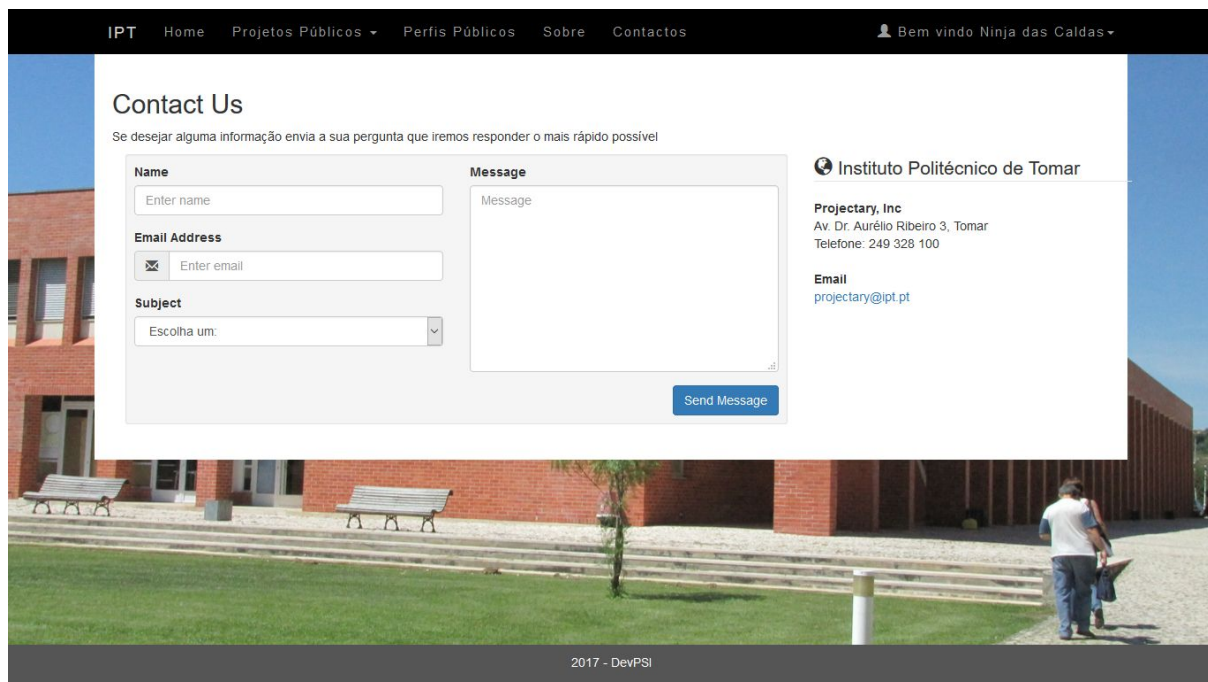


Figura 14.10 - Página “Contactos”.

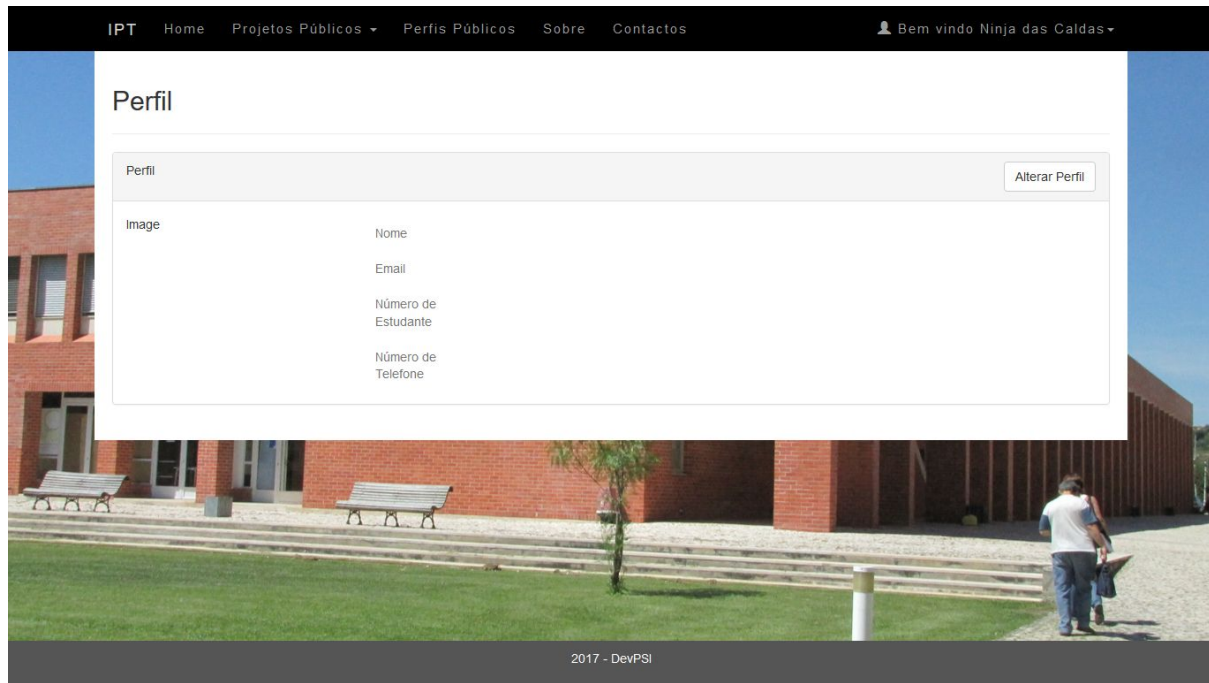


Figura 14.11 - Página de perfil.

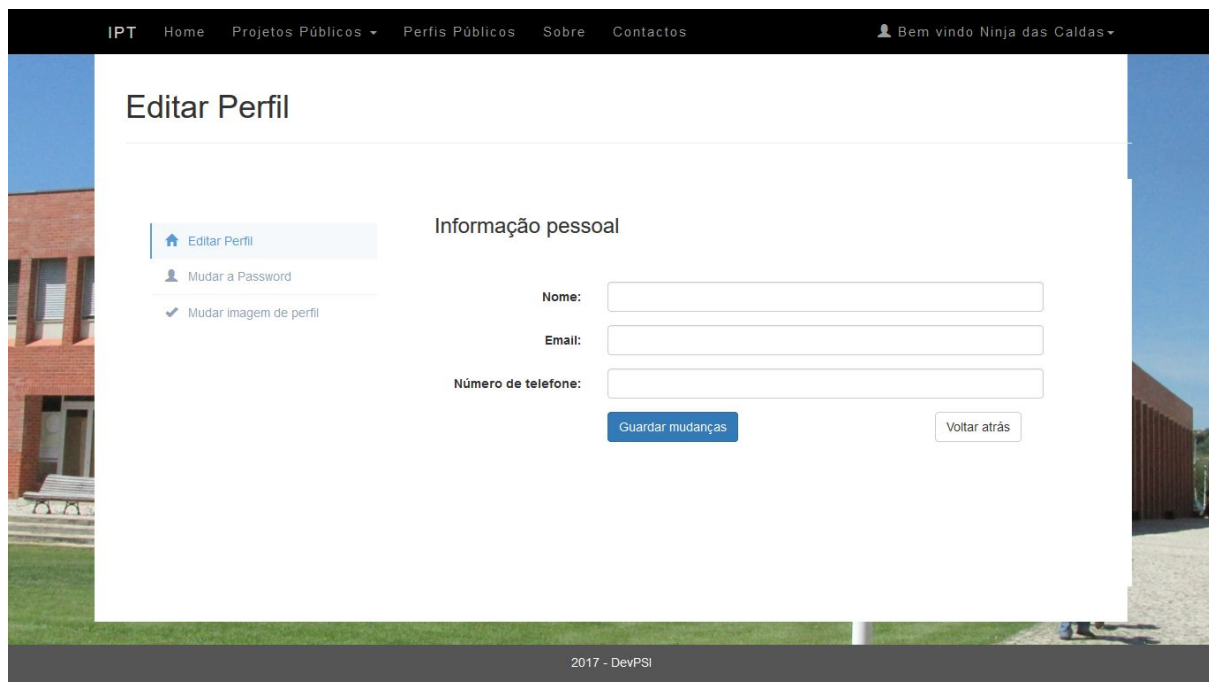


Figura 14.12 - Página de alterar perfil.

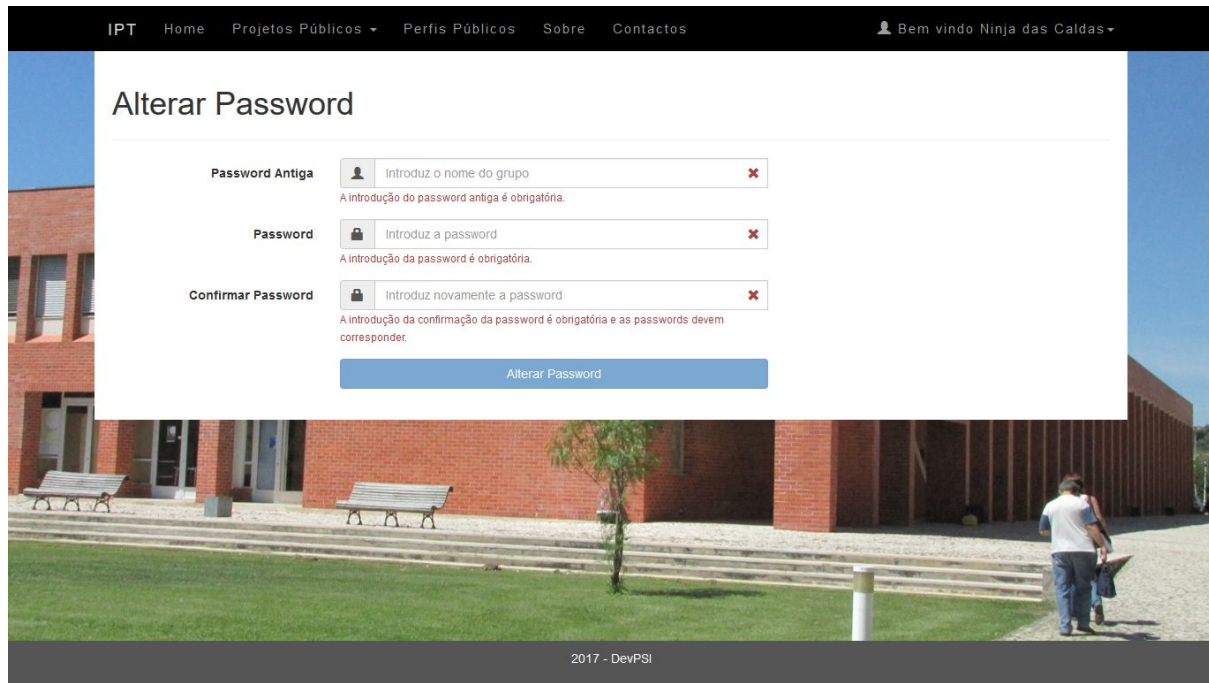


Figura 14.13 - Página de mudar password.

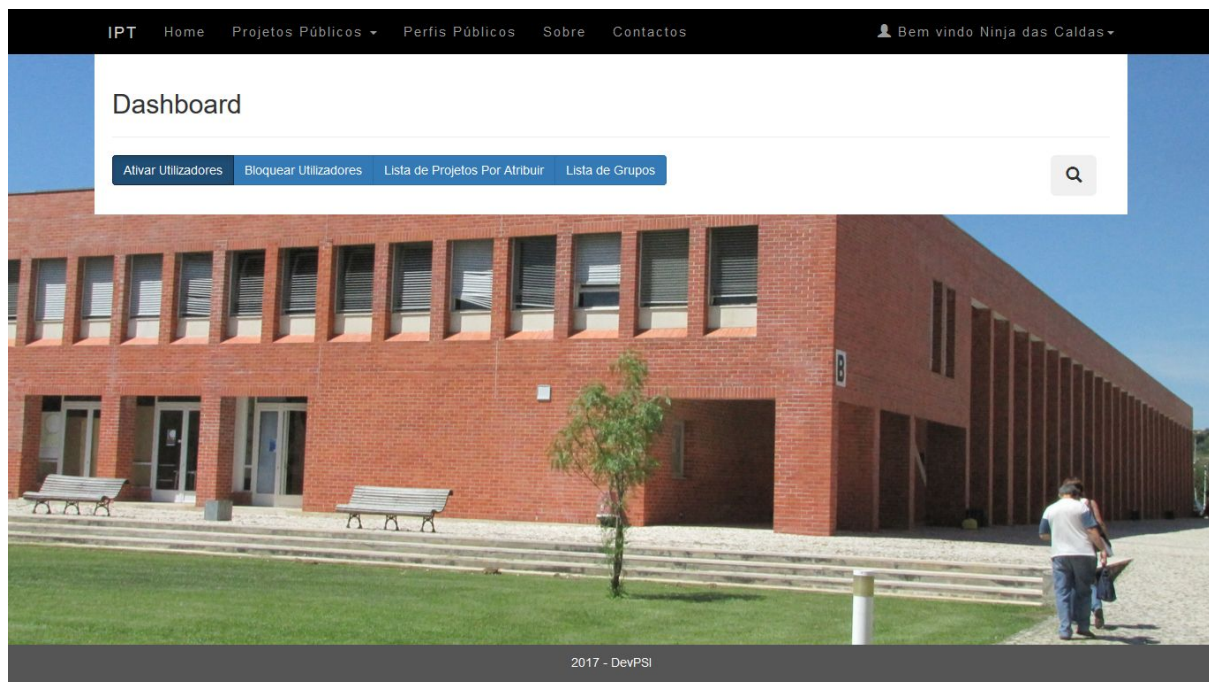


Figura 14.14 - Página de dashboard - ativar utilizadores - usado por administradores.

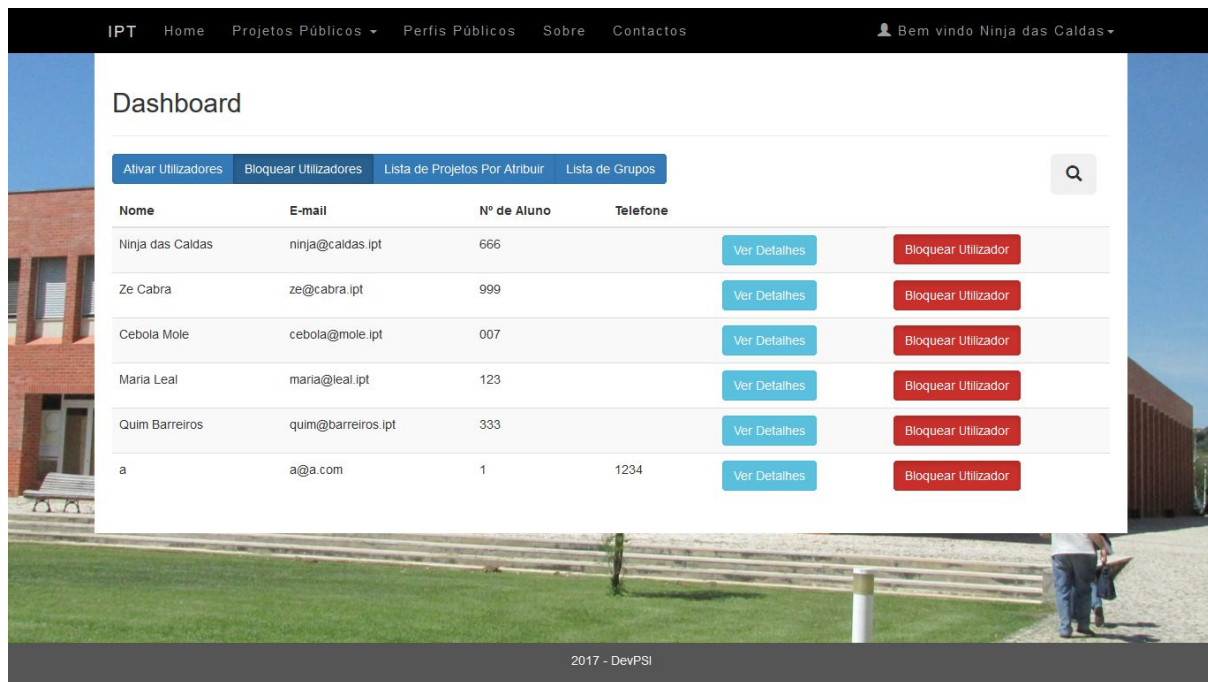


Figura 14.15 - Página de dashboard - bloquear utilizadores.

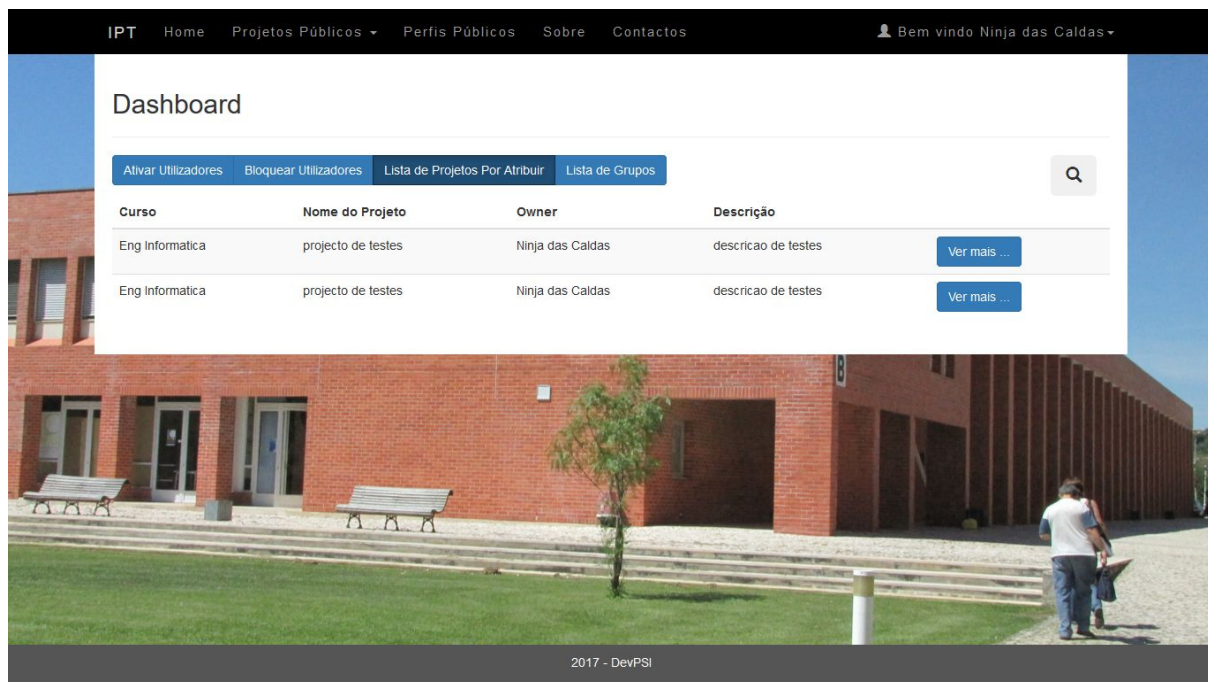


Figura 14.16 - Página de dashboard - lista de projetos por atribuir.

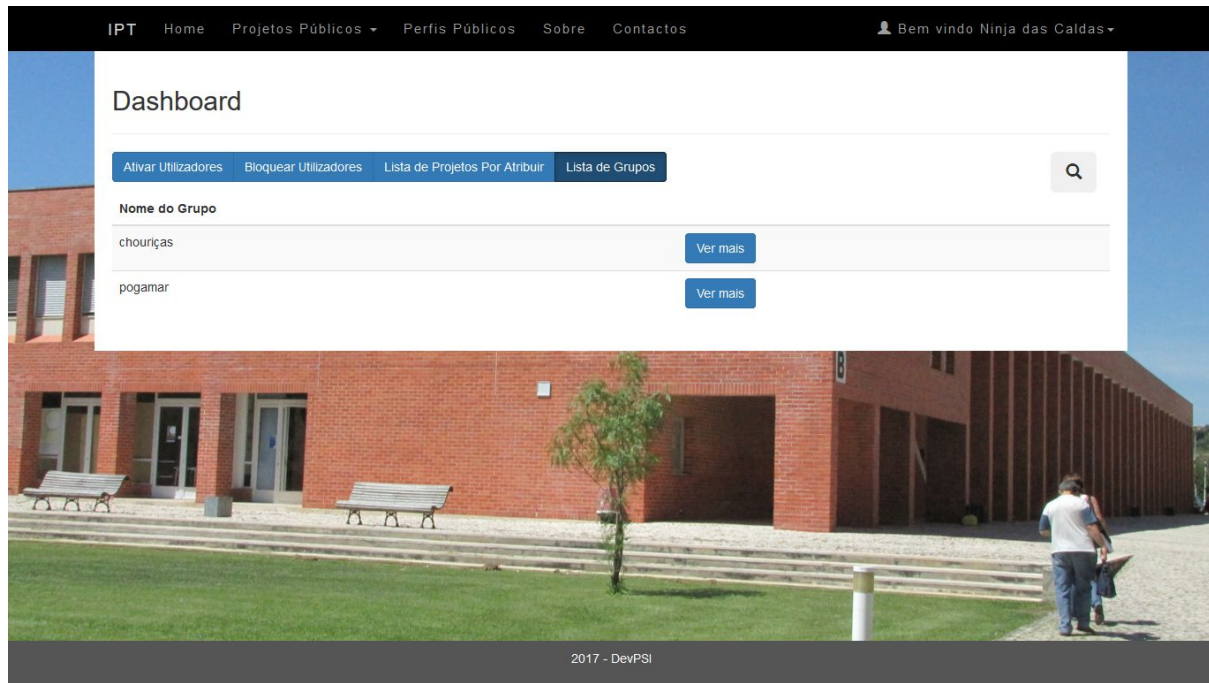


Figura 14.17 - Página de dashboard - lista de grupos.