

INSTITUTO POLITÉCNICO DE TOMAR

ESCOLA SUPERIOR DE TECNOLOGIAS DE TOMAR

Repositório de Projetos do IPT

**Mestrado em Engenharia Informática – Internet das Coisas**

Engenharia de Software

2016/2017

**Autores**António Paulo Rodrigues  
*<email@ipt.pt> - nº 2208*  
Miguel Filipe Redol Cotralha Inácio Coelho  
*<email@ipt.pt> - nº 18284*  
Nelson Miguel Pereira Gomes  
*<email@ipt.pt> - nº 16037*  
Pedro André Casimiro Nunes  
*<email@ipt.pt> - nº 16811*  
Pedro Henriques Dias  
*<email@ipt.pt> - nº 17108*Pedro Miguel Pinheiro De Matos  
*<email@ipt.pt> - nº 13341*  
Vasco Manuel Joaquim Marques  
*<email@ipt.pt> - nº 11046*

**Orientador**Renato Panda  
*<renato.panda@ipt.pt>*

# Resumo

Este documento documenta o projecto “Repositório de Projectos do IPT”, realizado no âmbito da disciplina de Engenharia de Software, Mestrado de Engenharia Informática (*Internet of Things*) do Instituto Politécnico de Tomar. Nesta disciplina foram introduzidas metodologias ágeis de desenvolvimento de *software,* assim como tecnologias e conceitos novos, tais como Ruby, Rails, mapeamento de objectos relacionais (ORM), ambientes de desenvolvimento virtuais, testes automatizados e ferramentas colaborativas, que serviram de base para o desenvolvimento do projecto.

## Sumário

O objectivo deste trabalho foi criar um repositório de projectos académicos que agregue todos os trabalhos desenvolvidos pelos alunos do IPT. Pretendeu-se com isto dotar a instituição de uma plataforma web que sirva de ponto central para quem quiser conhecer o que tem sido feito dentro da instituição, aumentando assim a sua exposição para o exterior.

O trabalho foi realizado utilizando metodologias SCRUM, sendo dividido em 1+4 iterações. A primeira iteração serviu para contacto com ferramentas e linguagens novas (Ruby) e para definição da plataforma a implementar. Esta iteração deu origem a um conjunto inicial de “*user stories*” que foram sido implementadas nas quatro iterações seguintes, em conjunto com novas histórias que foram surgindo. O projecto foi desenvolvido em Ruby 2.3, usando a framework Ruby on Rails 5. Segue o paradigma Model-View-Controller, tendo recorrido às frameworks / bibliotecas bootstrap 4 e jQuery para um interface moderno e adaptado a diversos dispositivos – responsivo.

O projecto e respectivo código-fonte estão disponíveis na página deste e no repositório respectivo.

## Summary

Lorem ipsum… o mesmo mas em inglês ☺

**Palavras-chave**

Repositório, projectos, Ruby, Ruby on Rails, Bootstrap, responsive

Índice

[Resumo 2](#_Toc472457963)

[Sumário 2](#_Toc472457964)

[Summary 2](#_Toc472457965)

[Lista de Acrónimos e Abreviaturas 4](#_Toc472457966)

[1 Introdução 5](#_Toc472457967)

[1.1 Desenvolvimento Ágil 5](#_Toc472457968)

[1.2 Padrão Model-View-Controller (MVC) 7](#_Toc472457969)

[1.3 Ruby on Rails 7](#_Toc472457970)

[1.4 Bootstrap 4 e jQuery 8](#_Toc472457971)

[2 Desenvolvimento da Aplicação 9](#_Toc472457972)

[2.1 Planeamento (Iteração 0) 9](#_Toc472457973)

[2.2 Desenvolvimento (Iterações 1 a 4) 9](#_Toc472457974)

[2.2.1 Estrutura do Projecto 9](#_Toc472457975)

[2.2.2 Bibliotecas Utilizadas (Gems) 9](#_Toc472457976)

[2.2.3 Modelos 10](#_Toc472457977)

[2.2.4 Controladores 10](#_Toc472457978)

[2.2.5 Vistas 10](#_Toc472457979)

[2.2.6 Como experimentar / colaborar? 10](#_Toc472457980)

[3 Conclusão 22](#_Toc472457981)

[3.1 Trabalho futuro 22](#_Toc472457982)

[4 Referências 23](#_Toc472457983)

# Lista de Acrónimos e Abreviaturas

**IPT** - Instituto Politécnico de Tomar

**CEO** – Chief Executive Officer (Diretor Executivo)

**PO** - Product Owner

**Freeware** – Software livre de utilização

**TOP12** – Posição mais elevada numa lista / Herança / Classificação

**RoR** – Ruby on Rails

**Framework** – Estrutura que une código num projeto de software

**MVC** – Model-View-Controller

**Feeds –** Notícias novas

**Tags –** Referencias a palavras contidas em texto

**VBox** – Virtualização de sistemas operativos

**Linux** – Sistema operativo Livre

# Introdução

Nos paradigmas dos dias atuais, em que o desenvolvimento em projetos organizacionais são constantes, recai o problema de onde os guardar e como o publico alvo pode pesquisar sobre os desenvolvimentos efetuados. É neste problema que nos propusemos desenvolver uma plataforma vanguardista na resolução destas questões levantadas e observadas.

As questões mais relevantes, propõem uma oportunidade única, demonstrando o conhecimento adquirido nas escolas, e em particular no IPT, podendo ser alargada a outras escolas de ensino. Observou-se que não existe a possibilidade dos orientadores e alunos guardarem os seus trabalhos, e assim deixarem de estarem na gaveta ou numa prateleira a ganharem pó ou a serem esquecidos pela comunidade cientifica ou mesmo novos investigadores.

Neste sentido iremos construir uma estrutura acolhedora e simplificada ao utilizador, de modo que consiga introduzir os seus projetos, disponibilizando assim os seus conteúdos a nível mundial. A estrutura vai disponibilizar via internet, todo o estudo efetuado do projeto em causa, deixando os contatos dos desenvolvedores patentes e associados ao seu projeto. Assim propulsionamos que este estejam disponíveis para o contacto com interessados ou mesmo para serem anexados ao conhecimento e desenvolvimento dos sistemas de tecnologias.

## Desenvolvimento Ágil

Nos conceitos existentes hoje em dia, o desenvolvimento de software cada vez mais segue novos padrões, deixando de ser um desenvolvimento rígido de inicio e fim, em que o software é apresentado ao cliente como uma ferramenta completa. As tendências apontam para grandes ganhos no desenvolvimento ágil, podendo assim criar um software de inicio e fim com a modulação e estrutura em crescimento constante em cada etapa definida pela equipa de desenvolvimento e pelo seu requisitante.

Nestes termos modernos e testados é nesta base que nos propomos a desenvolver o projeto de repositório dos projetos desenvolvidos nas escolas.

O desenvolvimento ágil recai sobre um conceito flexível de desenvolvimento, más com prazos curtos de apresentação dos trabalhos desenvolvidos, no entanto mais eficaz na sua metodologia ágil, esta metodologia escolhida fui as *Scrum* que recai sobre;

O *Scrum* é sem duvida, o método ágil mais usado nos nossos dias, focando na gestão de projetos e com base no planeamento interativo e incremental, estas desdobram-se em fases em se dá o nome sprints.

O projeto desde o inicio, define-se a lista de funcionalidades a serem desenvolvidas, chamadas de “*product backlog*”, cada desta se trona uma “sprint” que cada desta passa a designar “*sprint backlog*”, são atividades distribuídas entre os membros da equipa, que as desenvolvem dentro de um prazo estabelecido. No fim de cada “Sprint” é feita uma reunião do que foi desenvolvido e o que ficou por desenvolver, esta chamada “*Sprint review meeting*”. Após esta reunião começam a planear a próxima “*Sprint*” e assim continuamente até chegar ao fim do desenvolvimento.

No desenvolvimento ágil existem três (3) papeis fundamentais;

- **Product Owner**: é um CEO do produto, este membro faz a ligação entre o cliente e “Scrum Team”, e é o responsável pelo projeto.

- **Scrum Master**: é basicamente o “Coach” da equipa, o orientador e responsável da equipa de desenvolvimento e motivador.

- **Scrum Team**: equipa de desenvolvimento do projeto.

Funcionamento do *Scrum*

***Sprint*** – são interações de 2 a 4 semanas, em cada uma delas inicia-se com uma reunião “*Sprint Planning Meeting*”, onde são decididos o que será desenvolvido, com a apresentação dos itens com maior prioridade e são divididos pela equipa.

***Scrum Team*** - todos os dias iniciam com uma reunião até 15 minutos, para sincronizar o trabalho a ser desenvolvido e informa o “*Scrum Master”* de algum impedimento no desenvolvimento.

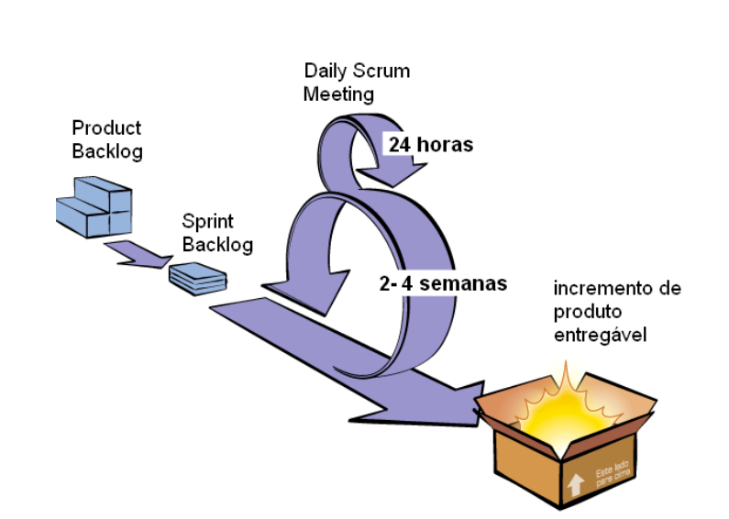


fig.1 – Processo *Scrum*

No fim de cada “*Sprint”* ocorre a “*Sprint Review Meeting”*, reunião na qual o “*Srcum Team”* apresenta para PO o que foi desenvolvido durante a “*Sprint”*.

Após esta, o *Scrum* Master efetua uma reunião com o *Scrum* Team numa ultima reunião da “*Sprint”* e faz uma respetiva do desenvolvimento (“*Retrospective Meeting*”), motivando a equipa para a próxima Sprint e melhorias a fazer.

Neste perfil, a nossa equipa segui-o a metodologia ágil apresentada.

## Padrão Model-View-Controller (MVC)

Fddfs

Gfdgfdg

Fgd

## Ruby on Rails

No paradigma da evolução do ser humano, remete-se cada vez mais o aparecimento de novas linguagens, estas crescem entre muitas e ganham espaço próprio, é o caso do “*Ruby*”[1], uma linguagem de programação que nasceu em meados do ano 1995 no Japão, tendo o seu criador “*Yukihiro Matsumoto*”, pensado numa linguagem dinâmica em “*Script*”, inspirada em outras linguagens existente o “*Ruby”* vem mostrar o seu poder de resolução de problemas conhecidas nas existentes, tais como, todas as variáveis são objetos, os métodos de geração de código são em tempo real, utiliza “*RubyGems*” que são bibliotecas desenvolvidas para facilitar e interagir com a aplicação, utiliza herança múltipla e dinâmica. Estando esta no TOP12 das linguagens segundo índice “Tiobe”[2].

fig.2 – Logotipo Ruby

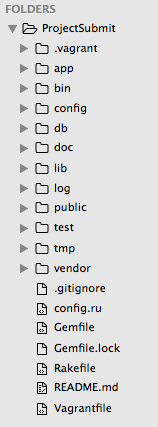
O RoR é uma “*Framework* *freeware*”, desenvolvida com base no padrão de arquitetura MVC, que assenta na linguagem “*Ruby”*, com o seu criador “*David Heinemeier Hansson*”, num projeto desenvolvido chamado “*BaseCamp*” foi lançado em miados de 2003.

Existem muitas MVC no desenvolvimento de aplicações para a internet, más o RoR, ganhou mercado na sua facilidade de desenvolvimento e na sua velocidade de desenvolvimento, com uma estrutura pré-definida, e com as bibliotecas como suporte (gems), veio trazer a esta “*Framework”* um poder enágua e uma posição vanguardista. Esta assenta em conceitos bastantes simples, “*Active Record, Action Pack, Action Mailer, Active Suport, Action WebService*”.

Tendo uma estrutura simplificada, ainda assim existe aspetos importantes que qualquer programador deve conhecer como parte importante no desenvolvimento[3];

- **app**: onde se encontra os Controllers, Models e Views

- **config**: routes, environments, initializers.

- **db**: onde se encontra tudo sobre a base de dados: migrate, seeds, schema.

- **public**: paginas gerais que mostrar erros de app e outras informações.

- **test**: onde se configura os teste da app

- **gemfile**.rb: onde colocamos as gems a serem utilizadas[7].

Existe outras pastas e ficheiros na estrutura, más consideramos para segundo plano, já que não são essenciais para o desenvolvimento propriamente dito, estás funcionam com o conjunto da MVC.

Fig.3 – Estrutura RoR

## Bootstrap 4 e jQuery

Fdsfsdfs

Dfsdf

Dfs

# Desenvolvimento da Aplicação

O desenvolvimento foi divido em 1 + 4 iterações detalhadas de seguida. A primeira iteração serviu para criar uma visão geral do projecto pretendido e planeamento do mesmo. Bla bla bla. Falar da distribuição do trabalho entre membros?

## Planeamento (Iteração 0)

Falar um pouco do que foi feito nesta iteração – levantamento de requisitos, escrita das “user stories”, os protótipos em papel (acho que fizeram 1), modelos identificados (a lista inicial), uma imagem do próprio protótipo inicial que fizeram,

## Desenvolvimento (Iterações 1 a 4)

Bla bla bla bla

O desenvolvimento decorreu durante o resto do semestre, sendo dividido em 4 iterações de cerca de duas semanas cada. Entre cada iteração foram feitas as reuniões de scrum típicas (revisão, retrospectiva, planeamento – numa linha o que se faz em cada uma delas). No início de cada aula prática foi ainda feita a típica “Daily Meeting” para cada membro informar a equipa do que fez na semana anterior e o que iria fazer na semana seguinte.

### Estrutura do Projecto

Imagem das pastas e dizer pada onde vão os controladores, o que é o Vagrantfile e .vagrant, o Gemfile, pasta doc, pasta tests, o que há dentro da pasta app (app/views, app/controllers), o que há na pasta assets (js e css), que os uploads do paperclip vão para a pasta system/p

### Bibliotecas Utilizadas (Gems)

Breve descrição das gems utilizadas mas \*bem organizada\*. Apenas as gems mais relevantes, com uma descrição do que fazem / porque foram usadas e o url.

#### Paperclip

Esta biblioteca permite gerir de forma mais fácil o envio de ficheiros (upload) pelo utilizador. Foi utilizada no modelo Person para a foto das pessoas e no modelo Project para fazer toda a gestão de documentos, slides e imagens relativas a cada projecto.

Link: www.xcxcxc.fdfd

#### Devise

Dfsfsdfdsfsd

#### Rails\_Admin

fsdfsdfsd

#### will\_paginate

fsdfdsfdsfds

etc

etc

etc

### Modelos

O que representam com pequena descrição e diagrama detalhado.

### Controladores

O que faz cada um dos controladores e o diagrama respectivo

### Vistas

Aqui falar rapidamente das vistas e tecnologia usada. Colocar 2 ou 3 imagens chave – home, ver projecto, ver pessoa, backoffice. Idealmente colocar vista também num smartphone para mostrar resultado.

### Como experimentar / colaborar?

Pode experimentar o projecto ou participar no desenvolvimento fazendo:

Instalar vagrant e virtualbox

Git clone

Cd IPT-pastaçasras

Cd …

Vagrant up

Vagrant ssh

Cd /vagrant

Rails s –b0.0.0.0

Abrir localhost:3000

\*\*\*\*\*\*\*\*\*\* aqui está o texto anterior, reorganizar e reescrever! \*\*\*\*\*\*\*\*\*\*\*\*\*\*

O nosso desenvolvimento recai sobre o modelo ágil, tendo em conta uma modulação do processo ao tempo disponível de acordo com o semestre a decorrer. Neste principio eremos modelar o nosso modelo de modo a termos uma aplicação funcional aos princípios que nos propusemos.

O problema identificado fui o não existir uma aplicação publica que desse acesso as pessoas a pesquisa e o conhecimento dos projetos desenvolvidos no IPT ou em outras escolas que queram participar neste invento.

*4.1 - Nesta identificação definimos*

**Projetos**: recipiente para guardar a informação em pdf, fotos e os colaboradores.

**Escolas**: Registar escolas na plataforma para associar aos projetos.

**Cursos e Disciplinas**: Registar cursos referentes às escolas.

**Participantes**: Registar alunos e professores referentes aos projetos.

**Tags**: Poder-se pesquisar por palavras pretendidas.

**Administração**: Poder-se administrar facilmente todos os registos e utilizadores.

**Divulgação**: Poder-se ter um sistema de “***feeds***” noticias, quando é registado um novo projeto.

**Base de dados**: Guardar toda a informação estruturada e com referencias.

**Github** – Repositório partilhado de desenvolvimento.

**Vagrant** – Utilização *VBox* com sistema linix.

**Trello** – Controlo das interações e tarefas a executar pela equipa.

*4.2 – Desenvolvimento*

No desenvolvimento ágil (“*Scrum*”), definimos as nossas interações (“*Sprint”).*

*Requisitos 1ºSprint:*

Eu quero ver os projetos sem estar registado.

Eu quero um sistema de inserir projetos fácil e simples.

Eu quero que sejam apresentados na página inicial os projetos mais recentes.

Eu quero poder fazer logout.

Eu quero como utilizador registado poder criar novos projetos.

Eu como utilizador registado quero poder modificar projetos.

Eu como utilizador devo poder pesquisar projetos independentemente de este estar registado ou não.

Eu como utilizador normal quero ver as páginas sem login.

***1º - Sprint:***

- Desenvolver / estudar a base de dados em papel.

- Criar sistema de Virtualização com VBox e Vagrant.

- Instalar o RoR

- Criação da estrutura de projetos

- Criação da estrutura escolas

- Criação da estrutura cursos e Disciplinas

- Criação da estrutura users

*Requisitos 2ºSprint*

Eu como utilizador registado quero poder recuperar a password da minha conta.

Eu como utilizador registado quero poder alterar os meus dados pessoais.

Eu como utilizador quero ver os projetos mais recentes primeiro, na página inicial.

Eu como utilizador quero poder ver as classificações dados pelas pessoas aos projetos.

Eu como utilizador quero poder ver as classificações dados pelas pessoas aos projetos.

Eu como utilizador registado quero poder alterar a minha password.

Eu como utilizador registado quero poder criar projetos.

Eu como utilizador registado quero poder adicionar vídeos e fotografias aos projetos.

Eu como utilizador quero poder pesquisar por curso.

Eu como utilizador quero poder pesquisar por tag´s.

Eu como utilizador quero poder pesquisar pela data ou ano do projeto.

Eu como utilizador registado quero poder eliminar projetos.

Eu como utilizador registado quero poder modificar projetos inseridos.

Eu como utilizador registado quero poder eliminar projetos.

Eu como utilizador quero um design atrativo.

Eu como utilizador quero poder pesquisar por projetos na página principal.

***2º - Sprint***

- Design da página inicial

- Criação da estrutura participantes

- Criação da estrutura de inserir files

- Criar o sistema Tag

- Criar o sistema de pesquisa

*Requisitos 3ºSprint*

Adicionar mensagem flash de log in / Log out do utilizador.

Ver integração de facebook e twitter com rails

O projecto é o principal do nosso site, é aquilo para o qual ele foi feito. Deve mostrar toda a informação e as associações existentes.

O utilizador \*\*não registado\*\* deve ser capaz de listar todos os projectos existentes, vendo primeiro os projetos mais recentes.

O utilizador \*\*não registado\*\* deve ser capaz de, a qualquer altura, pesquisar por um projeto.

O utilizador \*\*não registado\*\* deve ser capaz de ver os detalhes de um projeto.

O utilizador \*\*não registado\*\* deve, nos detalhes do projeto, conseguir ver a que escola, curso e disciplina pertence o projeto, sendo capaz de navegar para detalhes de cada um desses campos.

O utilizador \*\*não registado\*\* deve, nos detalhes do projeto, ser capaz de ver as pessoas associadas ao projeto (equipa e orientadores).

O utilizador \*\*não registado\*\* deve, nos detalhes do projeto, ser capaz de ver informação da apresentação (se houve ou não), o dia, hora, sala e os júris.

O utilizador \*\*não registado\*\* deve, nos detalhes do projeto, ser capaz de ver as várias imagens associadas a este.

O utilizador \*\*não registado\*\* deve, nos detalhes do projeto, ser capaz de ver as tags associadas ao projeto.

O utilizador \*\*registado\*\* deve ser capaz de adicionar ou editar um projeto, sendo obrigado a escolher o curso / disciplina a que está associado

O utilizador \*\*registado\*\* deve ser capaz de apagar um projeto, apagando os objetos associados a este (p. ex. apresentações).

O utilizador \*\*registado\*\* deve ser capaz de adicionar uma ou várias imagens a um projeto.

O utilizador \*\*registado\*\* deve ser capaz de adicionar e remover pessoas (equipa, orientadores) ao projeto.

O utilizador \*\*não registado\*\* deve conseguir ver informação da disciplina tal como o ano e semestre desta, a que curso pertence os últimos projetos desta.

O utilizador \*\*registado\*\* deve ser obrigado a escolher o curso a que a disciplina pertence ao criar/editar.

O utilizador \*\*registado\*\* deve ser obrigado a confirmar que quer apagar uma disciplina. Ao apagar a disciplina, os seus projetos também terão que ser apagados.

O utilizador \*\*não registado\*\* deve ser capaz de listar as instituições existentes.

O utilizador \*\*não registado\*\* deve ser capaz de ver os detalhes de uma instituição.

O utilizador \*\*registado\*\* deve conseguir adicionar uma nova instituição.

O utilizador \*\*registado\*\* deve conseguir editar uma nova instituição.

O utilizador \*\*registado\*\* deve ser capaz de apagar uma instituição, apagando assim tudo o que lhe está associado.

O utilizador \*\*registado\*\* deve receber mensagens de erro claras ao tentar adicionar / alterar uma instituição com dados inválidos.

O utilizador \*\*não registado\*\* ao ver os detalhes de uma escola deve conseguir ver a que instituição pertence.

O utilizador \*\*registado\*\* ao adicionar/editar uma escola terá que escolher a instituição a que este pertence.

O utilizador \*\*não registado\*\*, ao ver os detalhes de uma escola, deve ver a lista de cursos (degrees) agrupados por tipo (degree type).

O utilizador \*\*não registado\*\*, ao ver os detalhes de uma escola, deve ver também uma lista dos projetos dessa escola, ordenados por data (os últimos primeiro)" \*\*deixar esta para depois, não prioritária, seria com :through\*\*.

***3º - Sprint***

- Melhorar o sistema Tags

- Criação da estrutura administração

- Melhorar a estrutura de inserir files (projetos)

- Recuperação da password por email

*Requisitos 4ºSprint*

Style Password recuperação via email

Auto complete tags e search

Funcionalidades users vs admins

Recolher informação real sobre cursos, escolas, disciplinas?

Testes UX com users tipo.

Eliminar possíveis efeitos cansativos, nº passos para fazer certas ações, teste diferentes Resoluções, otimização do fundo, tipos de letra, tamanhos, menus, vistas de cada uma das coisas.

Integração social e rss de projetos

Search a funcionar

Layout verificar altura máxima das fotos página principal

Projetos Relacionados, onde está visível por tag's colocar em vista como na 1ª página

Escolas, mostrar só cursos com projetos

Teste do app.

***4º - Sprint***

- Melhorar o design

- Melhorar a pagina de inserir projetos

- Melhorar o sistema de pesquisa

- Criar sistema feed noticias

- Testar

- Mover a estrutura RoR para o servidor

- Testar

---- ainda falta ver melhor as interações ----

*Aplicação final:*

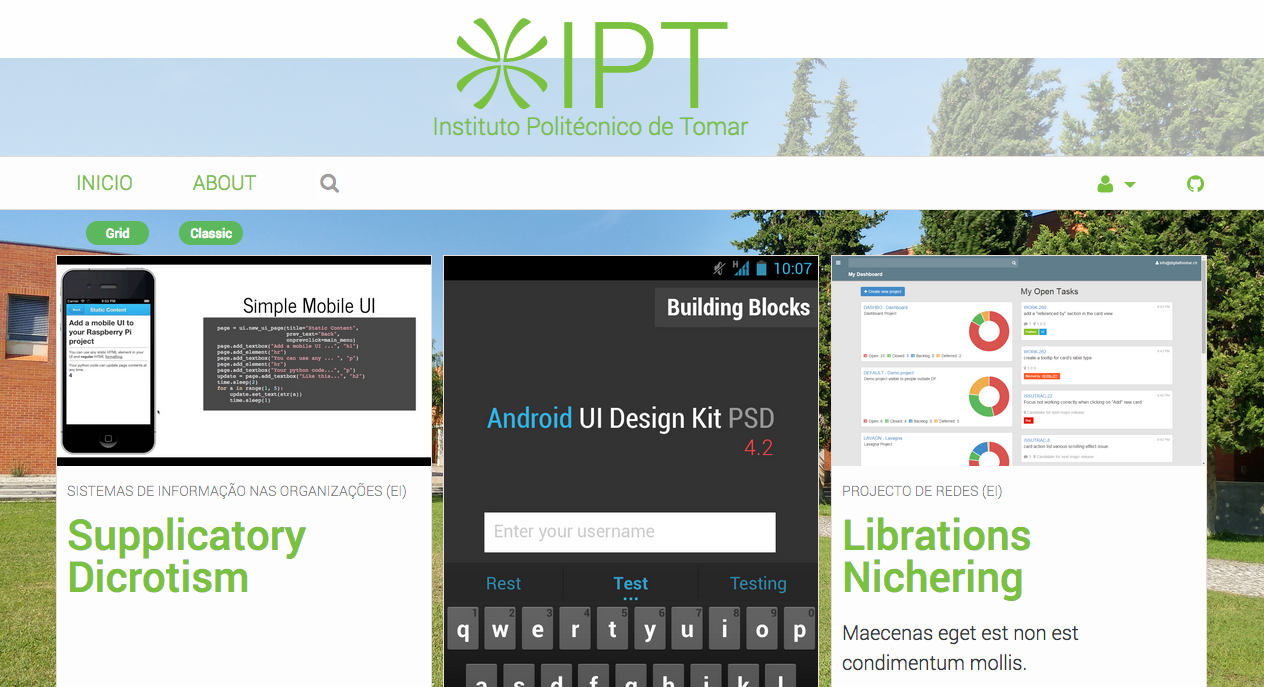


fig.4 – Design da pagina inicial

*Login:*

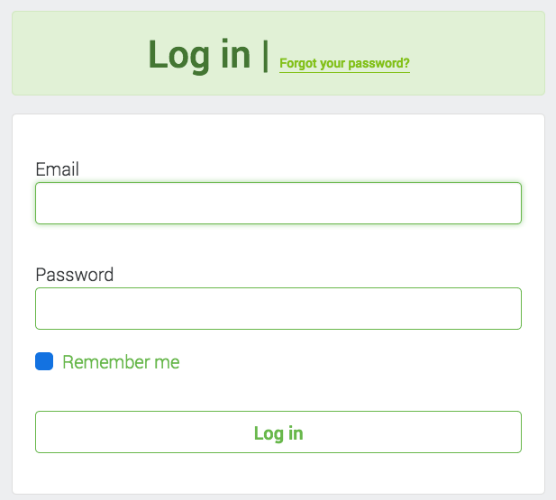


fig.5 – login de orientadores

*Novo Projeto:*

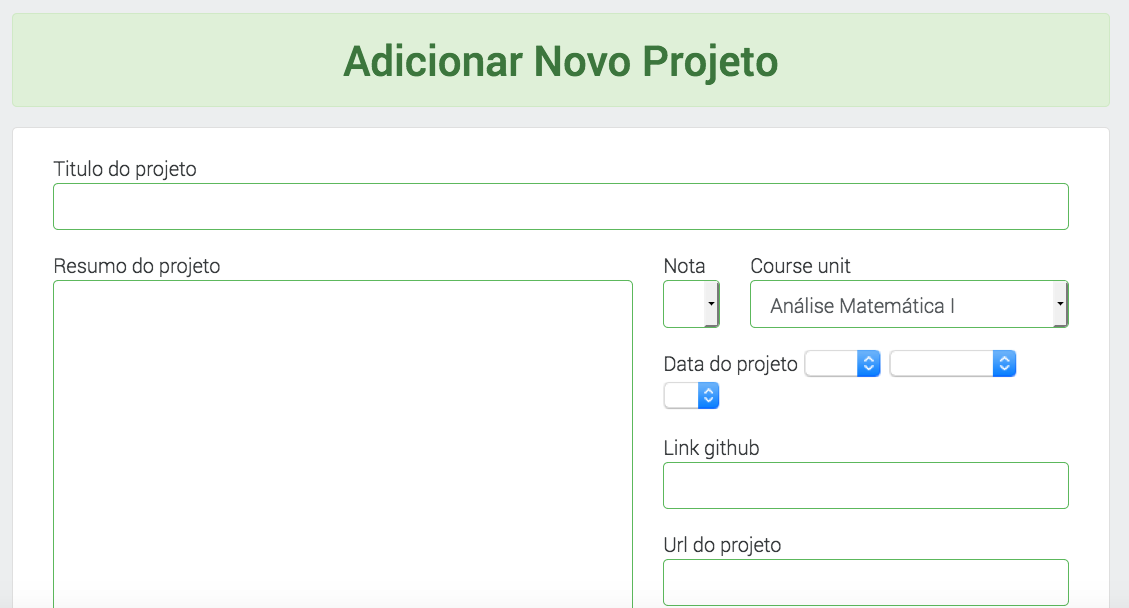


fig.6 – Adicionar novo projeto

*Administração:*

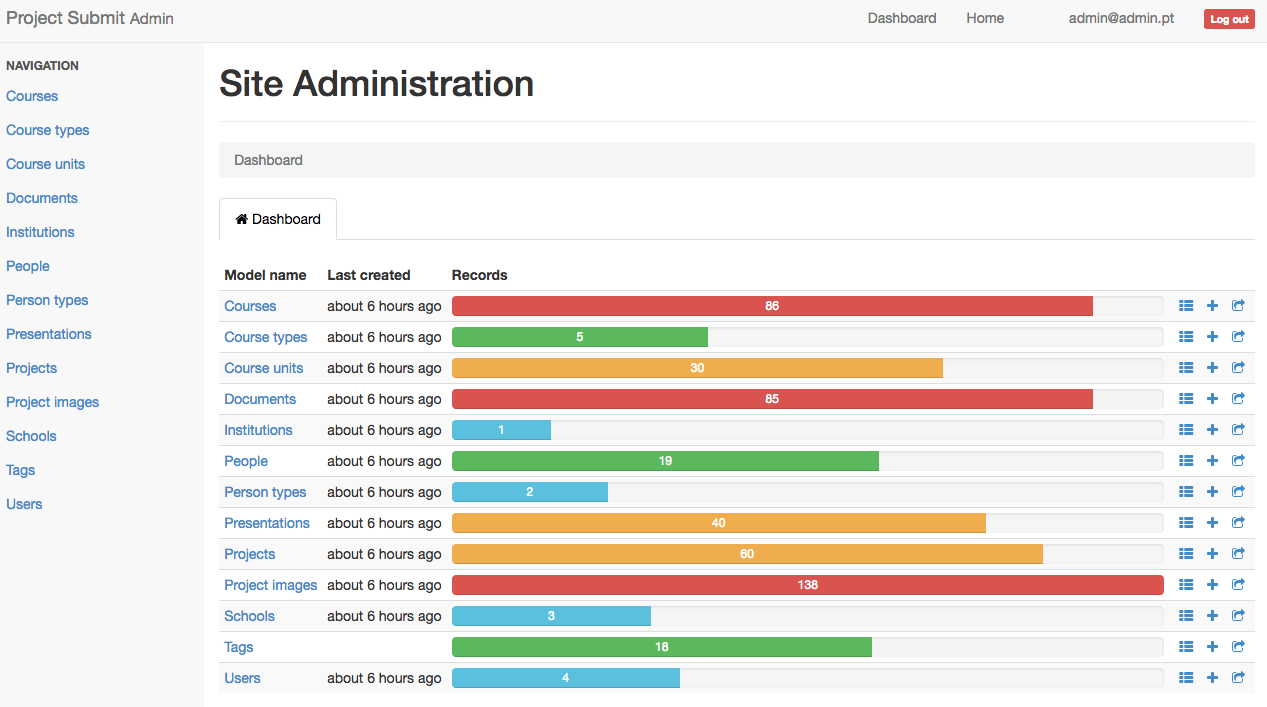


fig.7 - Administração

*Escolas:*

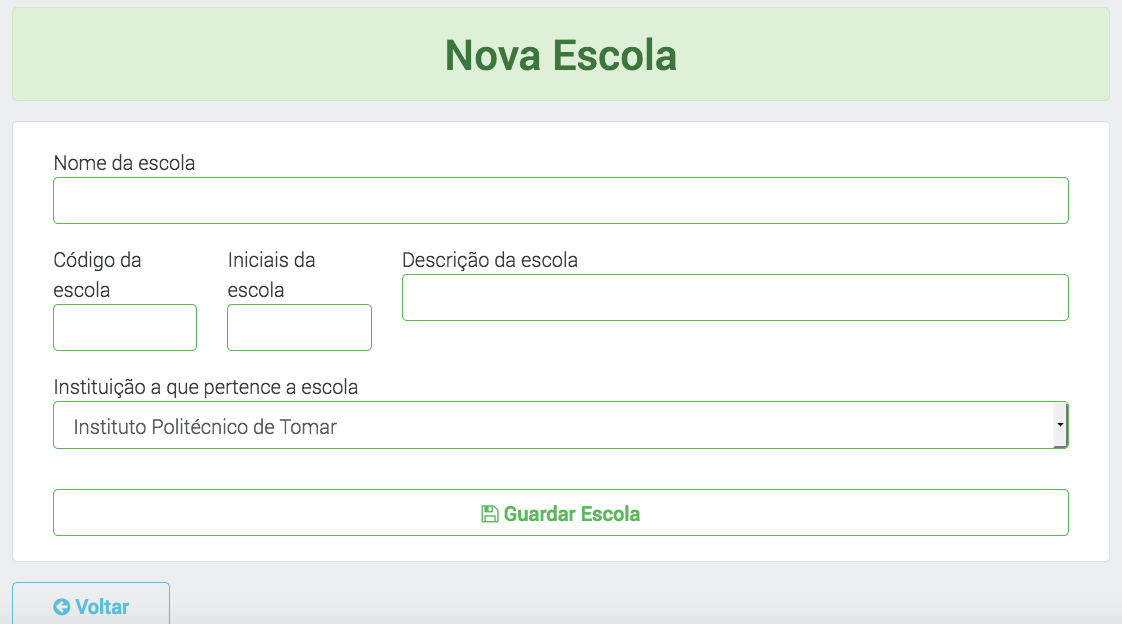


fig.8 – Nova escola

*Cursos / Disciplina:*

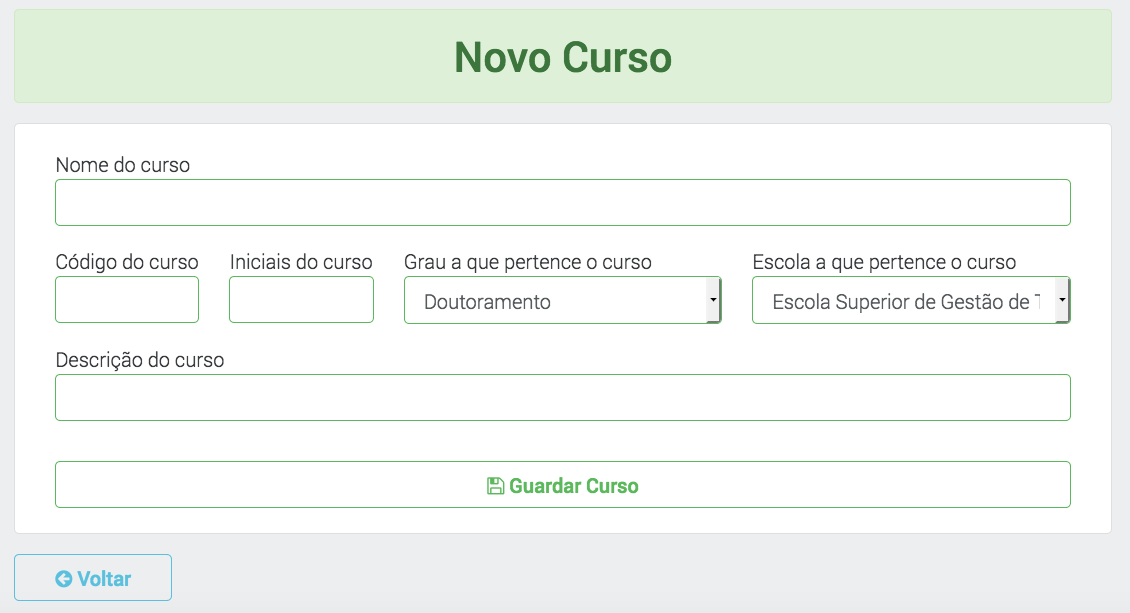


fig.9 – Novo Curso

*Gems utilizadas:*

source 'http://rubygems.org'

gem 'kaminari', '~> 0.17.0'

gem 'rails\_admin', '~> 1.1', '>= 1.1.1'

gem 'devise', '~> 4.2'

gem 'rails', '~> 5.0.0', '>= 5.0.0.1'

gem 'sqlite3'

gem 'puma', '~> 3.0'

gem 'sass-rails', '~> 5.0'

gem 'uglifier', '>= 1.3.0'

gem 'coffee-rails', '~> 4.2'

gem 'paperclip', '~> 5.0.0'

gem 'jquery-rails'

gem 'turbolinks', '~> 5'

gem 'jbuilder', '~> 2.5'

gem 'font-awesome-sass', '~> 4.7.0'

gem 'bootstrap', '~> 4.0.0.alpha6'

gem 'bootstrap-select-rails'

gem 'will\_paginate-bootstrap'

gem 'filesize'

group :development, :test do

gem 'byebug', platform: :mri

gem 'railroady'

end

group :development do

gem 'web-console'

end

gem 'tzinfo-data', platforms: [:mingw, :mswin, :x64\_mingw, :jruby]

source 'https://rails-assets.org' do

gem 'rails-assets-tether', '>= 1.1.0'

end

gem 'acts-as-taggable-on', '~> 4.0'

# Conclusão

O desenvolvimento ágil foi uma novidade entre o grupo, no entanto todos tiveram uma adaptação fácil. Compreendeu-se as importâncias das interações, que se realizavam semanalmente (aulas), onde ficava notoriamente vincado o desenvolvimento do trabalho por parte da equipa, ainda se nutava o espirito de grupo e a entreajuda entre membros na resolução de problemas. A equipa conclui que este método é bastante pertinente no desenvolvimento de app para a web, e que se nutou um desenvolvimento mais rápido e conciso.

No que respeita ao desenvolvimento do projeto, considera-se que é um avanço no mundo tecnológico, já que se passa a disponibilizar os projetos online dos estudantes. Assim poderão ser evoluídos e ou mesmo aproveitados para a investigação de novos conceitos, dando a estes um paradigma diferente do que se passa no momento atual, em que se encontrão escondidos / guardados algures, onde ninguém os pode consultar porque não sabe onde consultar.

Em sumo poderemos dirigir-mos ao conjunto do nosso projeto e poderemos dar nota positiva, tanto na aprendizagem como no desenvolvimento com o Ruby on Rails, dois sistemas novos, com uma grande potencia de mercado e ágil no seu desenvolvimento. Considerando-se um desenvolvimento rápido e eficaz, esta será uma tag a ter em conta nas nossas pesquisas para elaboração de novos trabalho.

## Trabalho futuro

O que ficou por fazer

# Referências

[1] Ruby, Ruby. Disponível em: [http://www.ruby-lang.org/pt/](http://www.ruby-lang.org/pt/%20) . Acesso em 17 de janeiro de 2017.

[2] TIOBE Index for January 2017, TIOBE. Disponível em : <http://www.tiobe.com/tiobe-index/>. Acesso em 17 de Janeiro de 2017.

[3] Ruby S., Thomas D., Hansson D.,H.,(2016). *Agile Web Development with Rails 5*. ISBN-13:978-1-68050-171-1. Raleigh, North Caroline.

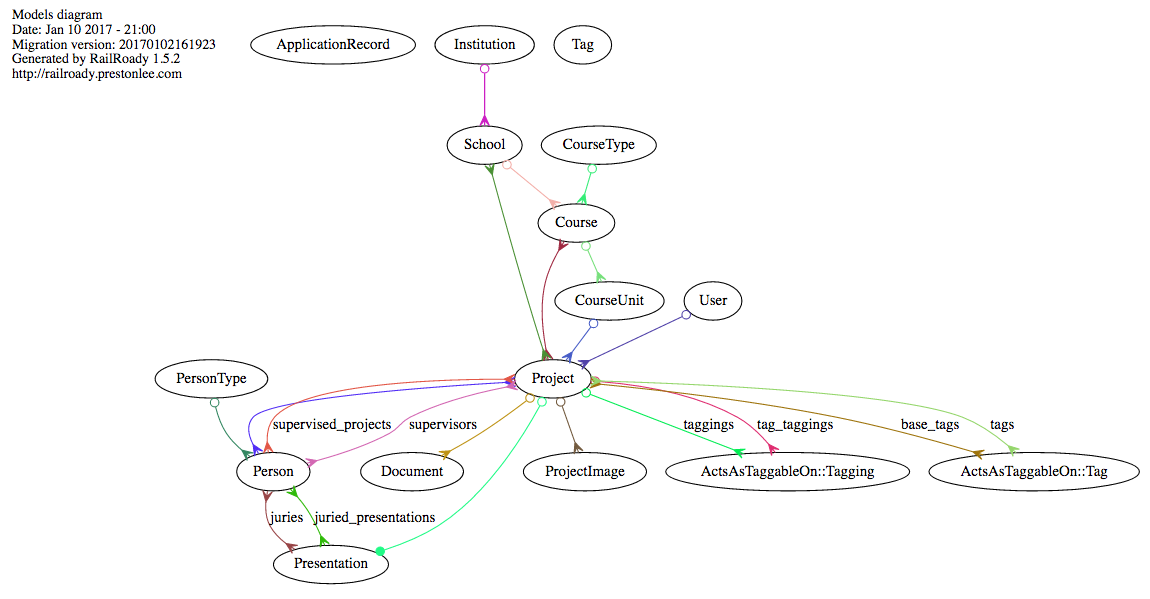
[4] Vieira N., (2012). *Howto | Conhecendo Ruby*. 1ª Versão, Brasil.

[5] Oliveira Jr. E., R.,(2016). *Conhecendo Ruby*. Victoria, British Columbia, Canada.

[6] Fowler C.,(2016). *Rails Recipes*. ISBN 0-9776166-0-6. Dallas, Texas.

[7] RubyGems. Disponivel em : <https://rubygems.org>. Acesso em 09 de janeiro de 2017.

Anexo 1 – Modelo Diagrama



Anexo 2 – Modelo Diagrama 2

