Projecto TePeCe

Engenharia Web – Mestrado de Engenharia Informática

Telma Correia – PG25263

Rui Silva – A54712

Pedro Vale – PG19806

**Link para o projeto (GitHub):** <https://github.com/filbri/EAW_MEI/>

Sumário

[Introdução 3](#_Toc392368808)

[Descrição do projecto 3](#_Toc392368809)

[Contexto 3](#_Toc392368810)

[Papeis 3](#_Toc392368811)

[Modelo de Dominio 4](#_Toc392368812)

[Diagramas de UC 5](#_Toc392368813)

[Dicionario de Dados 16](#_Toc392368816)

[Modelo de Dados 17](#_Toc392368817)

[Grupos de Utilizadores 18](#_Toc392368818)

[Site Views 18](#_Toc392368819)

[Style Guidelines 21](#_Toc392368820)

[Coarse Design 23](#_Toc392368821)

[Detailed Design 25](#_Toc392368822)

[Especificações WebRatio 27](#_Toc392368823)

[Implementação 34](#_Toc392368824)

[Ruby on Rails 34](#_Toc392368825)

[Desenvolvimento do Site 34](#_Toc392368826)

[Geração de Modelos 34](#_Toc392368827)

[Geração de Controladores 35](#_Toc392368828)

[Páginas Estáticas 36](#_Toc392368829)

[Métodos 36](#_Toc392368830)

[Views 38](#_Toc392368831)

[Validações de Dados 39](#_Toc392368832)

[Autenticação e Permissões 40](#_Toc392368833)

[Layout 40](#_Toc392368834)

# Introdução

## Descrição do projecto

Na base da educação de uma criança existe um triângulo pedagógico essencial no processo da sua aprendizagem. Esse triângulo é composto pelo Encarregado de Educação, Professor e Diretores Escolares. Todos estes atores têm um papel complementar e colaborativo na construção, monitorização, avaliação de exercícios e trabalhos realizados pela criança. O Projeto “TePeCe” assenta o seu grande objetivo nesta premissa, de um maior envolvimento destes atores, utilizando a Web como meio duma união de interesses.

## Contexto

Pretendese a realização de uma Aplicação Web que terá como utilizadores alvo o aluno e os três elementos intervenientes do seu *triângulo pedagógico* (professor, enc. de educação e direção). Existem vários tipos de utilizador. A cada tipo de utilizador estará associada uma área com as funcionalidades correspondentes.

A aplicação deverá possibilitar ao professor, a inserção de fichas de trabalho/exercícios, que serão dispostos para resolução a um aluno específico ou a um conjunto de alunos. As fichas de trabalho/exercícios terão uma data limite de realização definida pelo professor.

A realização de exercícios é apenas destinada aos alunos. Terminado o exercício o aluno recebe uma nota provisoria (podendo esta ser revista pelo professor). Todos os exercícios devem ficar guardados, podendo ser consultados pelo aluno e elementos do *triângulo pedagógico*.

A aplicação deve incluir um gestor de notificações que permite o envio, consulta e eliminação de notificações a todos os utilizadores (exceto o aluno).

Aos utilizadores será mostrado um resumo estatístico de todas as operações realizadas no sistema. Este resumo será adequado a cada tipo de utilizador e aos dados a si inerentes.

A responsabilidade da gestão dos utilizadores (inserção, remoção e atualização) será da direção.

## Papeis

#### Criança/Aluno:

A criança interage com o sistema realizando os exercícios propostos pelo seu professor. No final da realização do exercício o aluno recebe uma nota provisória da ficha de trabalho/ exercício realizado, que vai ser revisto pelo professor. A criança consegue ver o seu histórico de fichas realizadas e as respetivas avaliações, incluindo a média.

#### Professor:

O Professor propõe exercícios a serem realizados pelo aluno/criança (que esteja a seu cargo) e avalia os mesmos. O Professor consegue aceder a todos os registos dos seus alunos. Após a consulta e avaliação dos trabalhos, o Professor, consegue notificar o Encarregado de Educação, em caso do Aluno/Criança não ter realizado o exercício/trabalho, e poderá encaminhar essa mesma notificação para a Direção da escola.

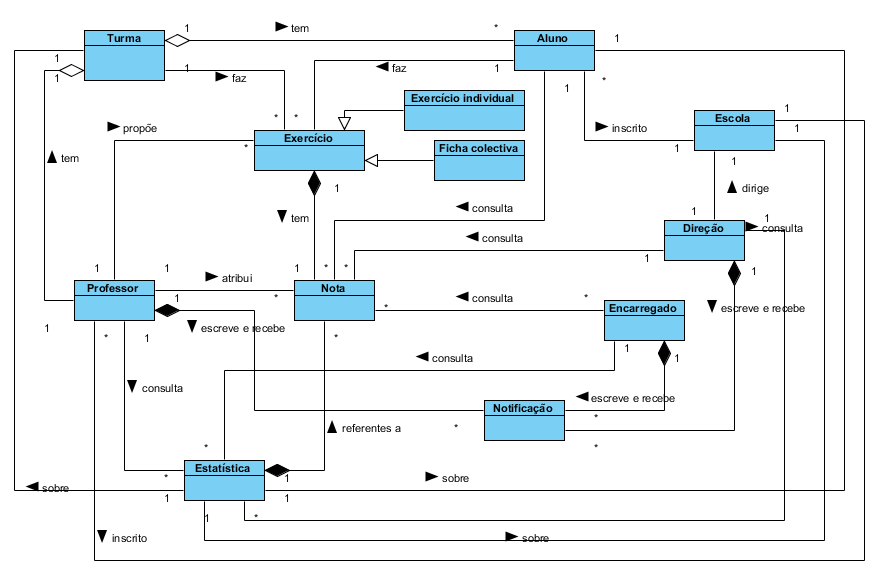
#### Encarregado de Educação:

O Encarregado de Educação poderá ver as notas e os trabalhos já avaliados do seu educando. Este poderá redigir notificações para Professores e Diretores, e responder às notificações que lhe são dirigidas.

#### Direção:

À Direção cabelhe responder às notificações dos Professores e do Encarregado de Educação, e terá acesso a estatísticas personalizadas de todas as avaliações realizadas na escola.

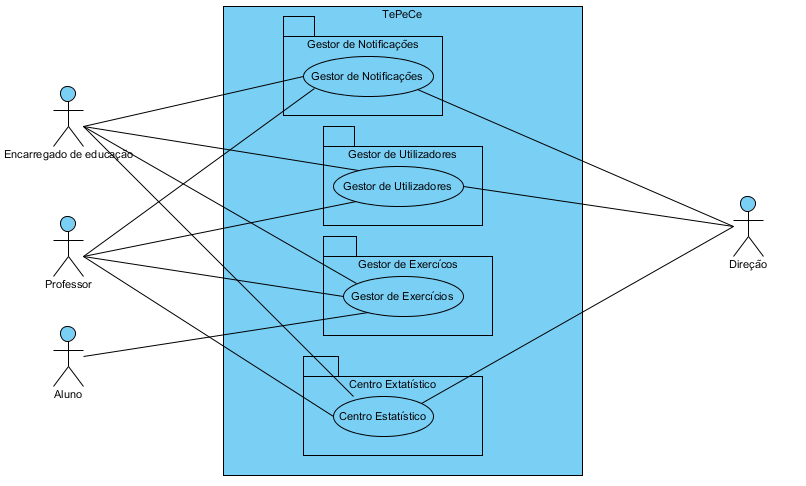
# Modelo de Dominio

Um dos primeiros passos na modelação da aplicação, foi a criação de um modelo de dominio. Este modelo indica os varios objectos e conceitos a serem utilizados na aplicação, bem como a interligação entre eles. Durante todo o processo de modelação e planeamento da aplicação este modelo foi usado como base de vários elementos. De seguida apresentamos o nosso modelo de dominio, criado utilizando a ferramenta VisualParadigm:

# Diagramas de UC

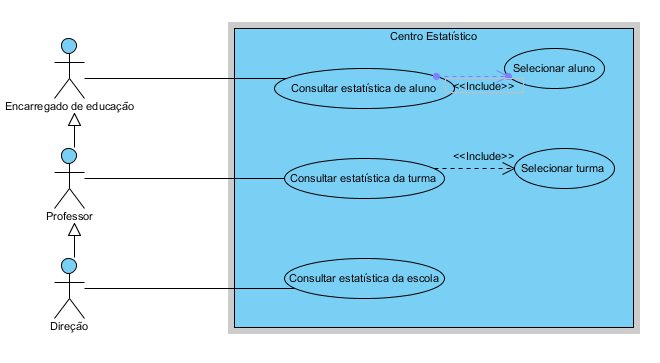
Um dos passos iniciais da modelação da aplicação, foi a especificação dos diversos actores a as interações que estes terão com o sistema. Esta especificação foi alcançada atravez da criação de diagramas de Casos de Uso, bem como a especificação textual desses casos de uso. Os actores presentes nos diagramas são os diversos tipos de utilizadores que a aplicação terá em conta: alunos, professores, encarregados e direção. Os diversos casos de uso foram repartidos por quatro packages: Gestor de Notificações, Gestor de Utilizadores, Gestor de Exercicios e Centro Estatistico. Estas diferentes packages definem as funcionalidades da aplicação de uma forma geral, enquanto que os casos de uso as definem de forma mais detalhada atravez das especificações textuais.

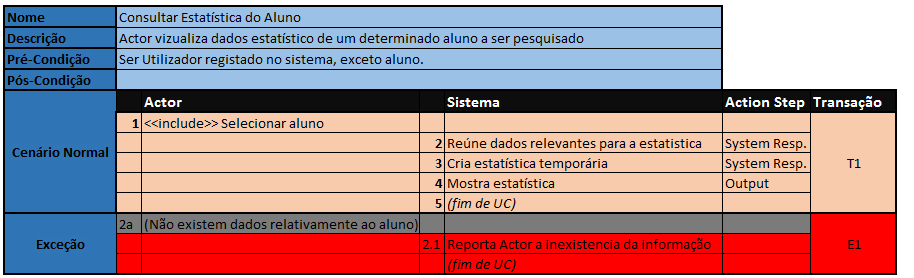
De seguida apresentamos o diagrama de casos de uso referente às diferentes packages do sistema:

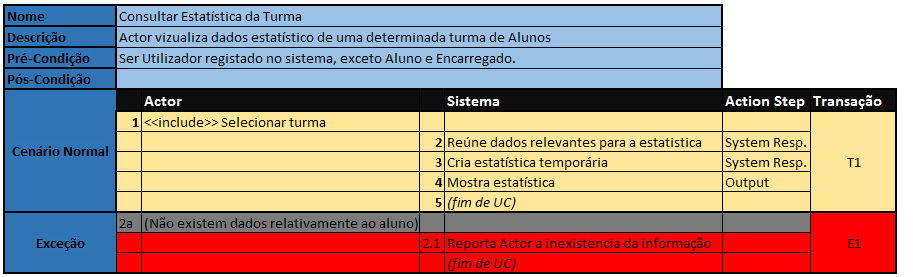


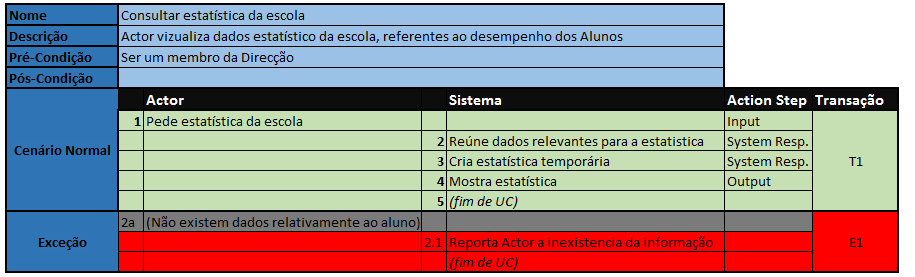
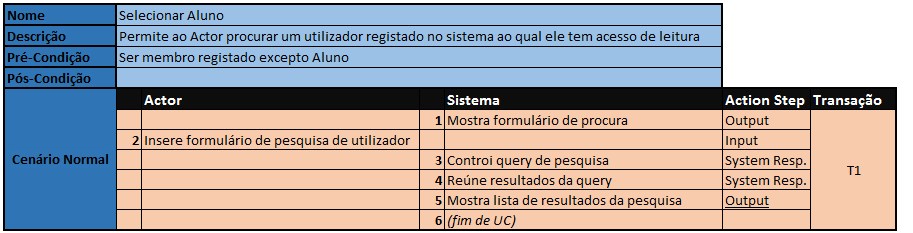
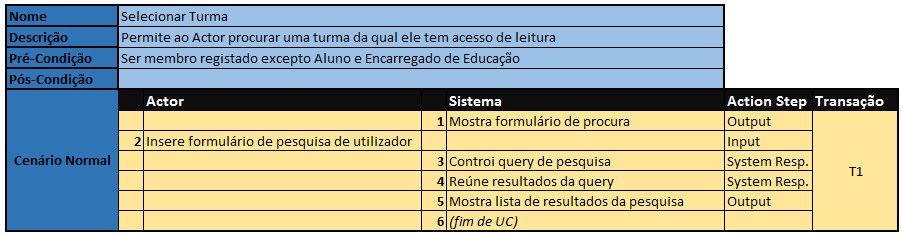
Os restantes diagramas indicam os casos de uso propriamente ditos, as acções especificas permitidas a cada tipo de utilizador, acções extra opcionais (atravez de "Extend") e acções extra obrigatórias (atravez de "Inculde"). Cada use case apresentados também terá uma especificação textual, contendo o seu nome, uma descrição, pré-condições que devem ser cumpridas, pós-condições que devem ser cumpridas e os cenários de execução (tanto os normais como excepções e alternativas) que indicam os diversos passos da interação entre o actor e o sistema.

De seguida apresentamos o diagrama referente ao centro estatistico:

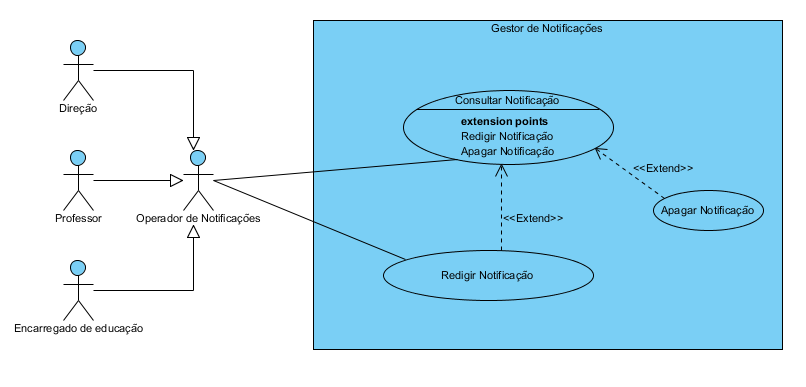


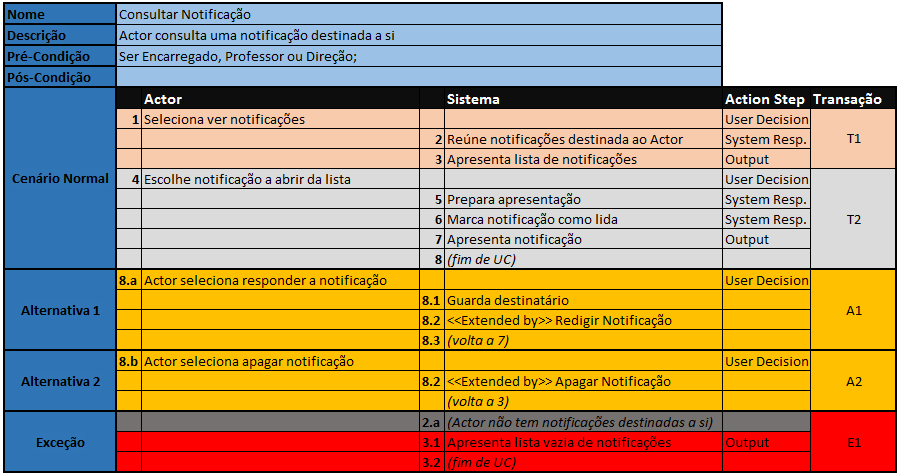
Os use cases deste diagrama contêm as seguintes especificações textuais:

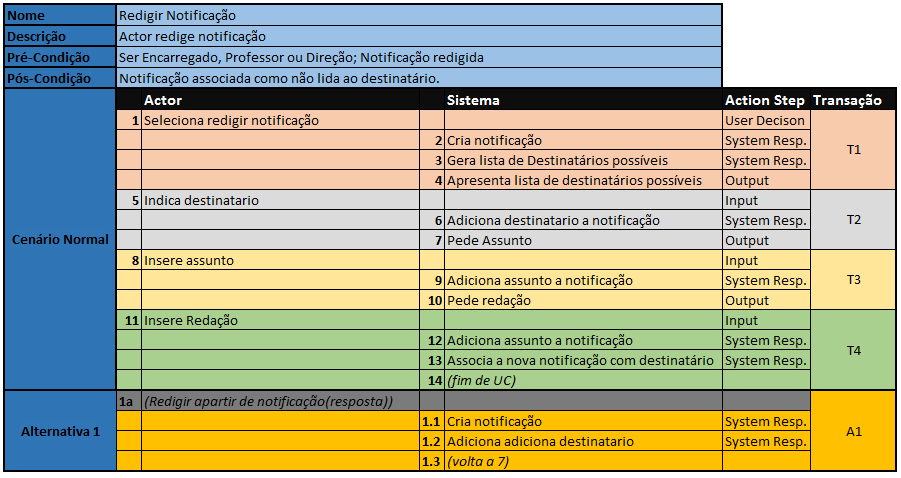


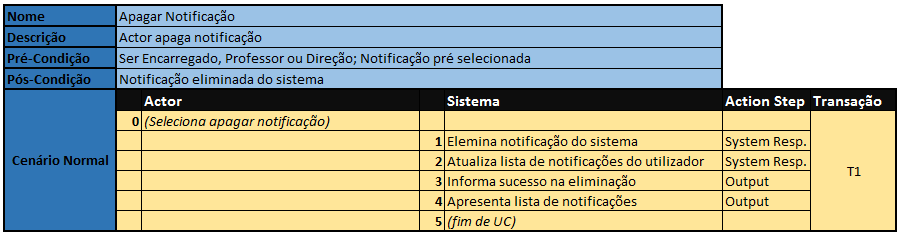


O seguinte diagrama é referente ao Gestor de Notificações:

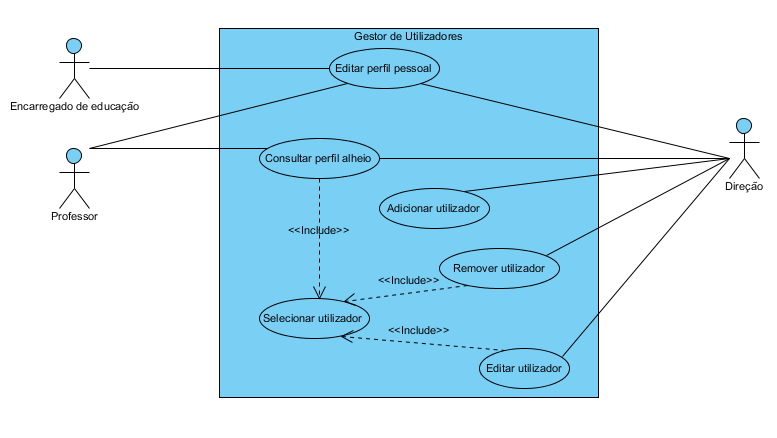


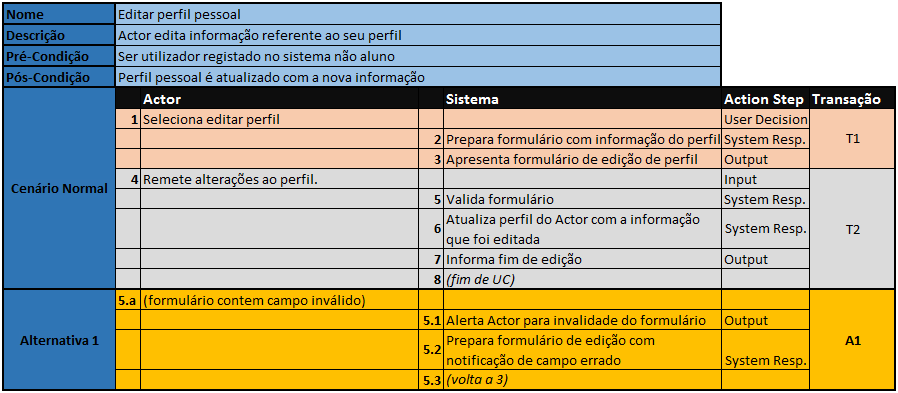
Os use cases deste diagrama contêm as seguintes especificações textuais:

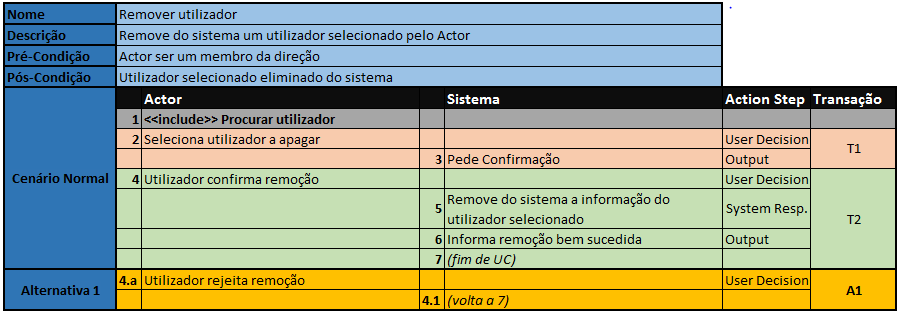
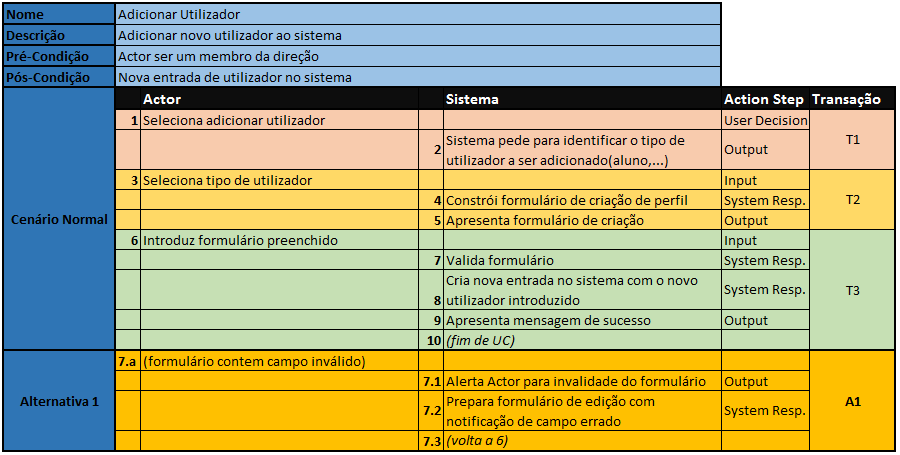
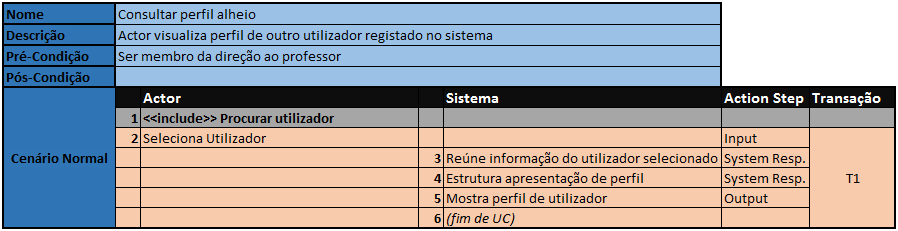


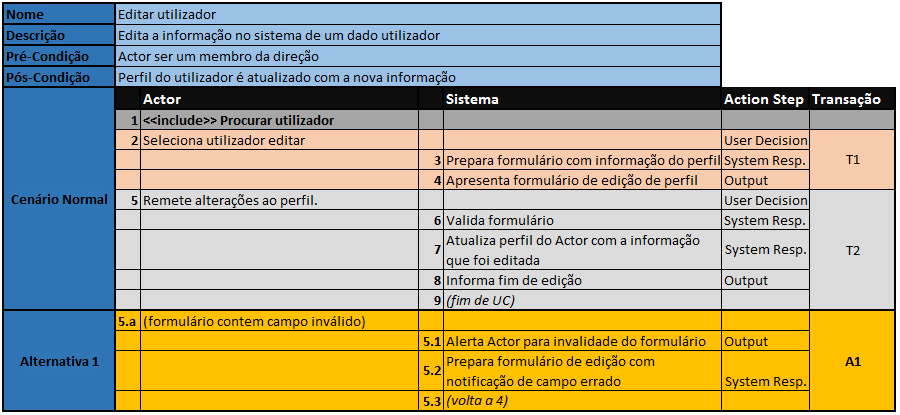
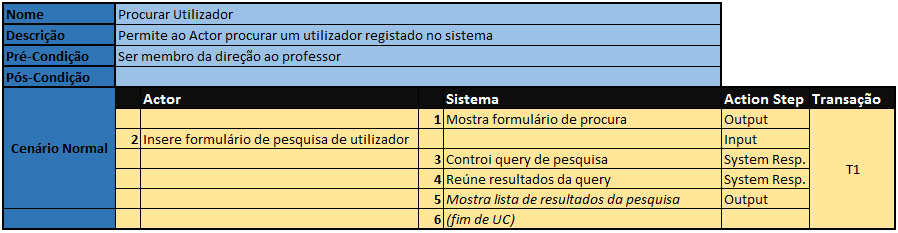


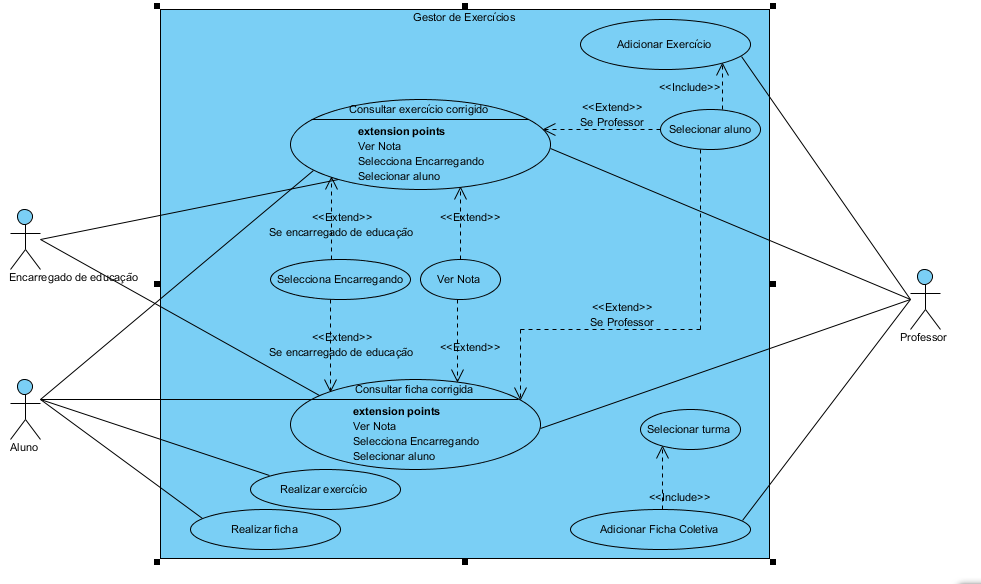
O diagrama que é referente ao Gestor de Utilizadores é o seguinte:

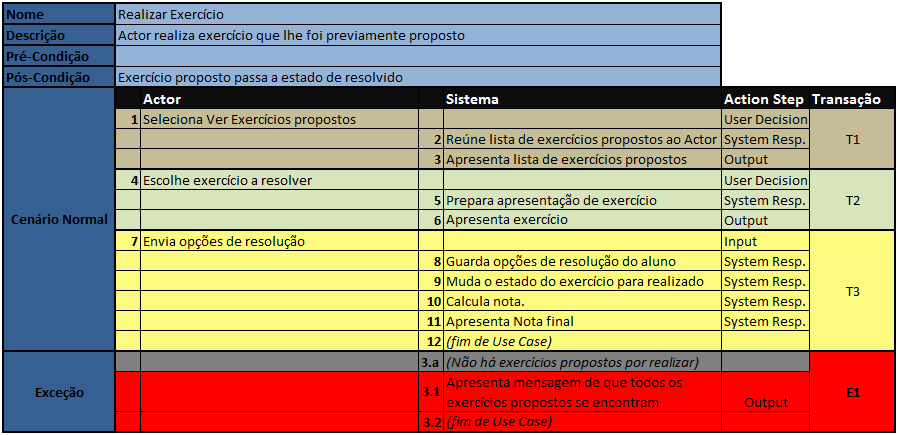


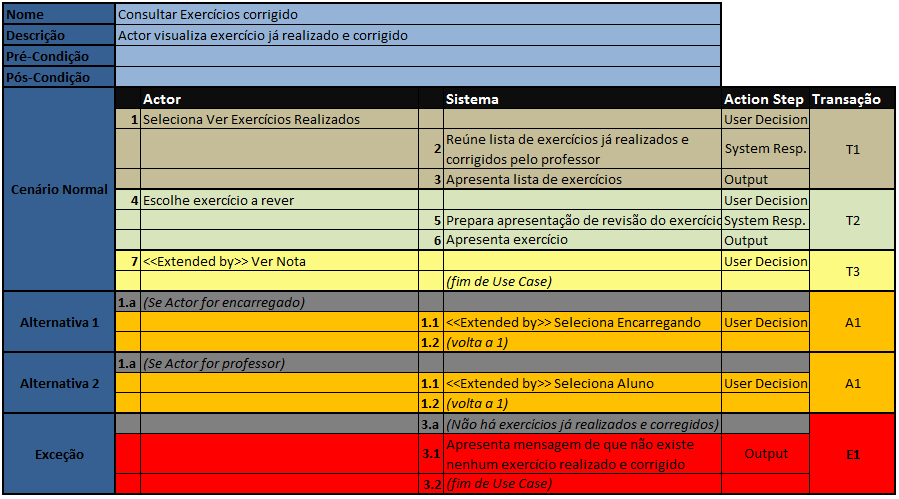
Os use cases deste diagrama contêm as seguintes especificações textuais:



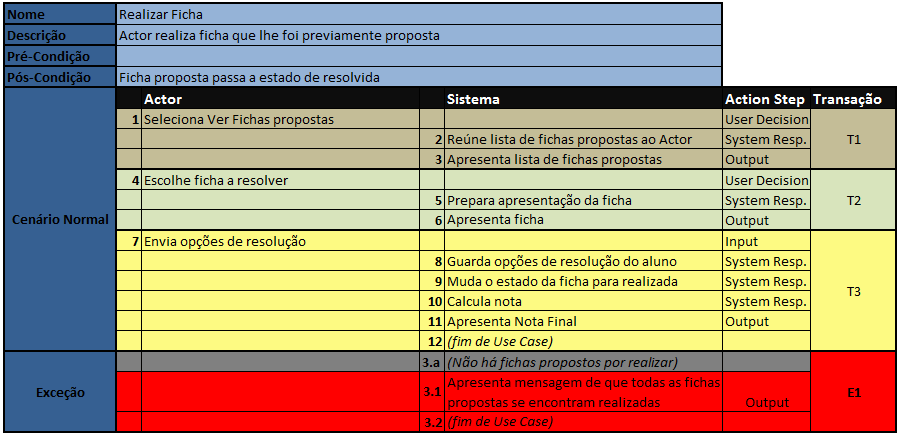
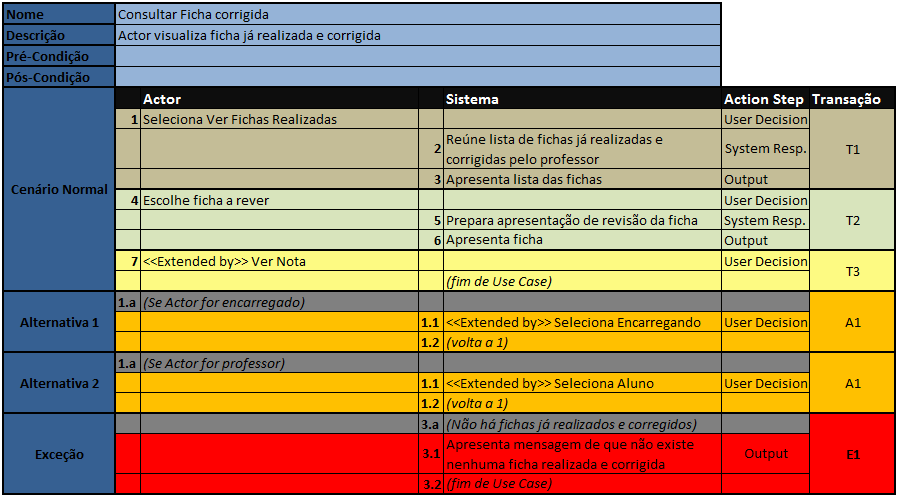


E, por fim, apresentamos o diagrama referente ao Gestor de Exercicios:

Os use cases deste diagrama contêm as seguintes especificações textuais:

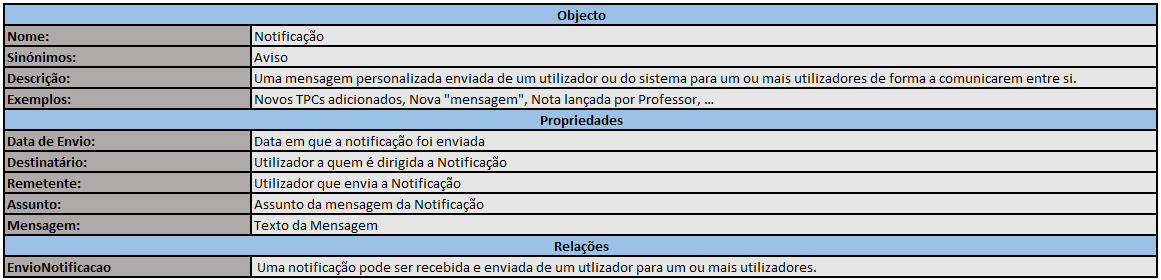
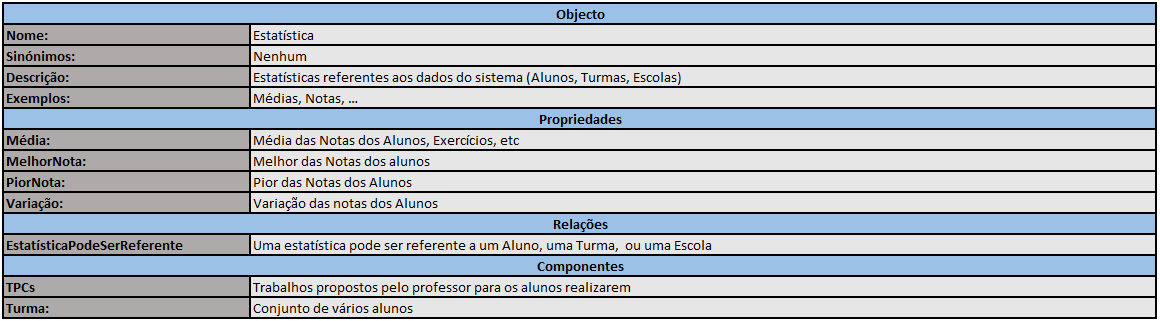


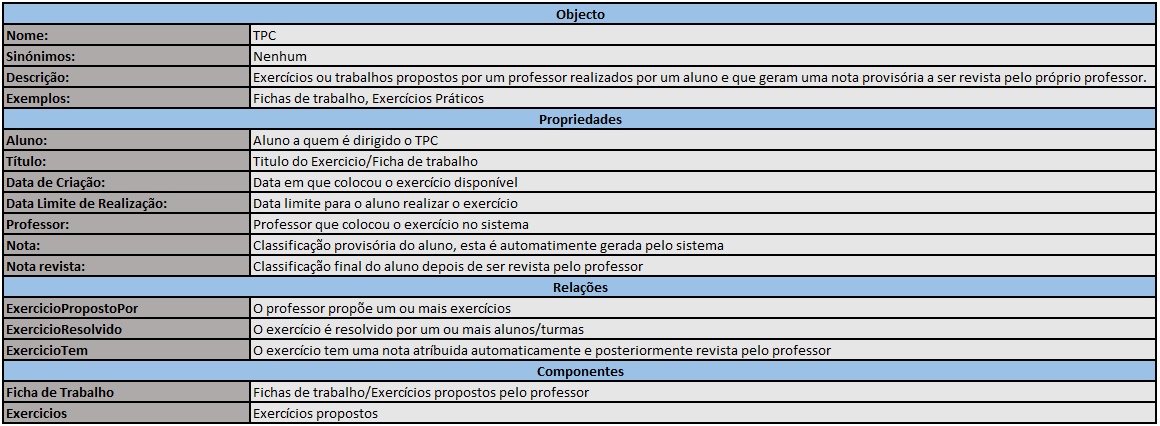
# 

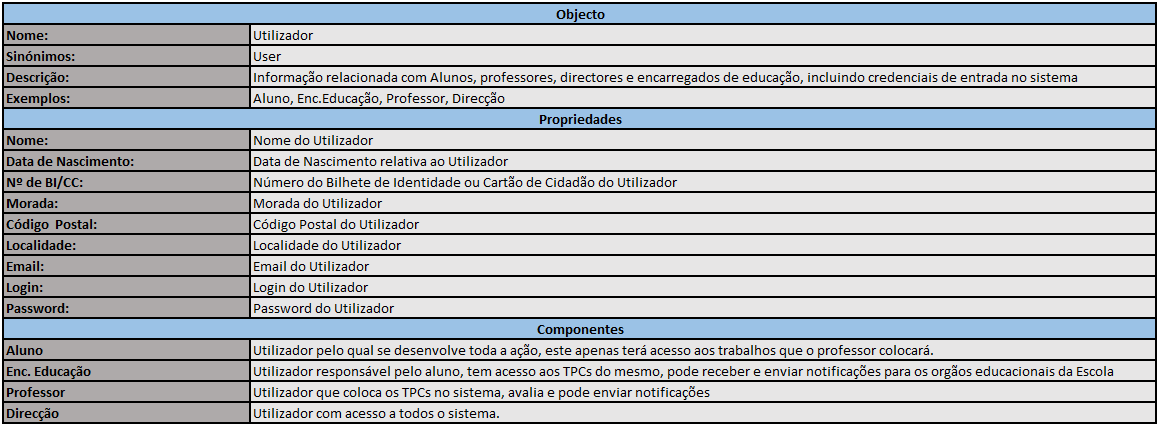


# 

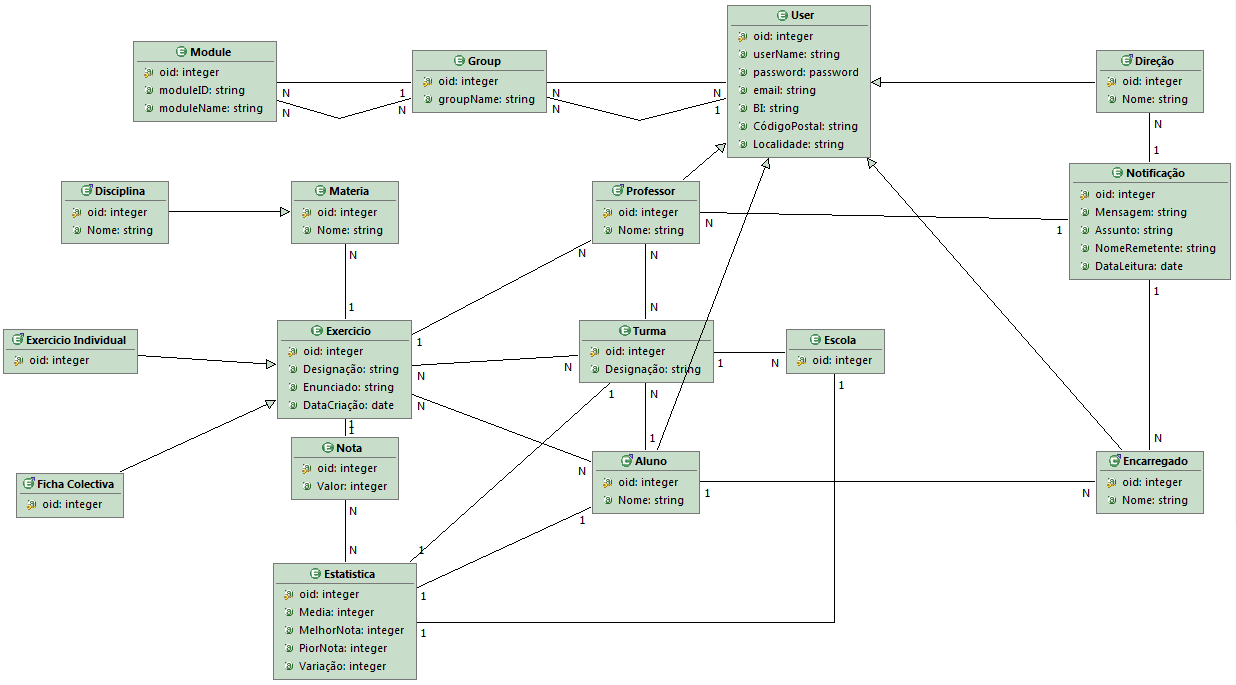
# Dicionario de Dados

Para representar os dados manuseados pela aplicação, foi criado um dicionario de dados que contem informação diversa sobre os objectos de dados principais que a aplicação deveria ter em conta. Este dicionario detalha as relações, propriedades e componentes, bem como outros dados, sobre os objectos que a aplicação irá possuir e manusear. Este dicionário foi usado como base de diversos outros componentes do planeamento e modelação da aplicação, como o modelo de dados. De seguida apresentamos as tabelas que representam o nosso dicionario de dados:

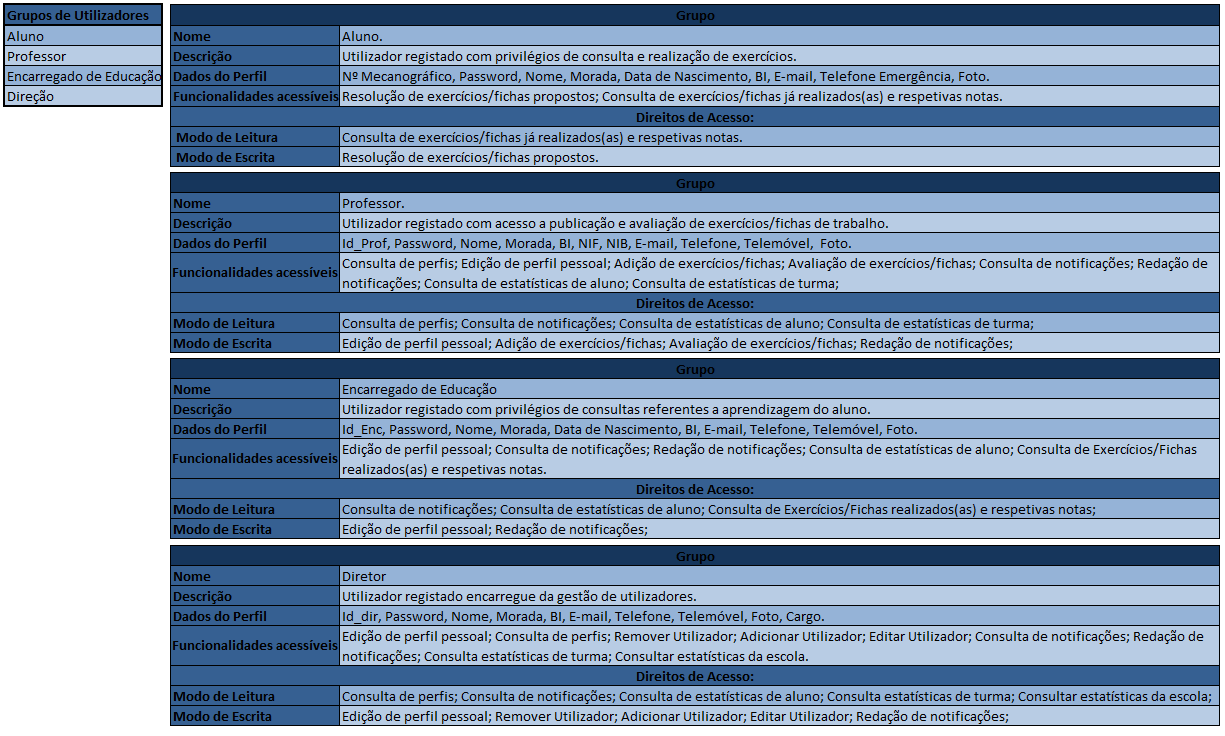




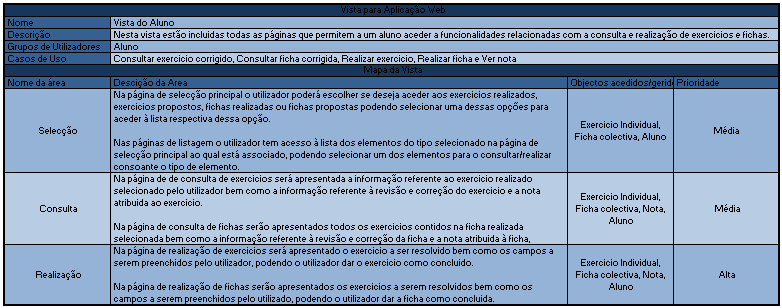
# Modelo de Dados

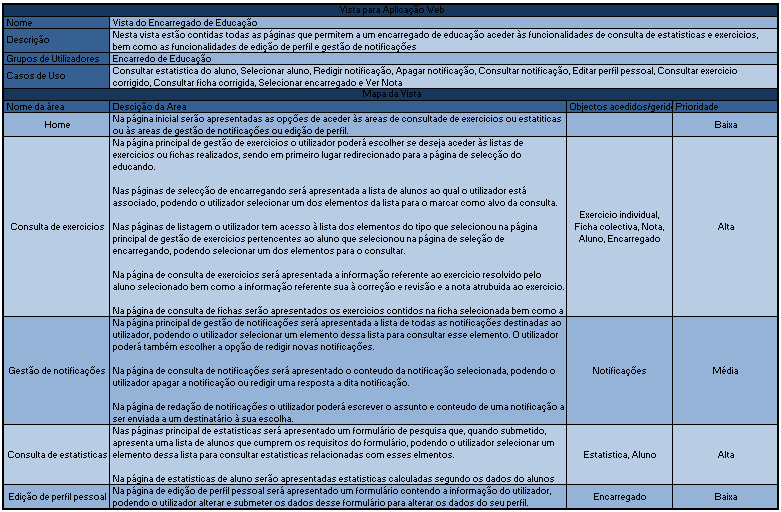
Durante a fase de modelação da aplicação, criamos um Modelo de Dados, que contem todos os objectos e dados que foram mais tarde utilizados na implementação da aplicação e na modelação das páginas da mesma. Este modelo toma em conta vários elementos da modelação da aplicação, tal como o dicionário de dados e modelo de dominio. Neste modelo WebML estão representados os diversos objectos de dados que a aplicação utiliza, os seus componentes e as interligações entre esses diversos objectos. De seguida apresentamos o nosso modelo de dados, criado utilizando a ferramenta WebRatio:

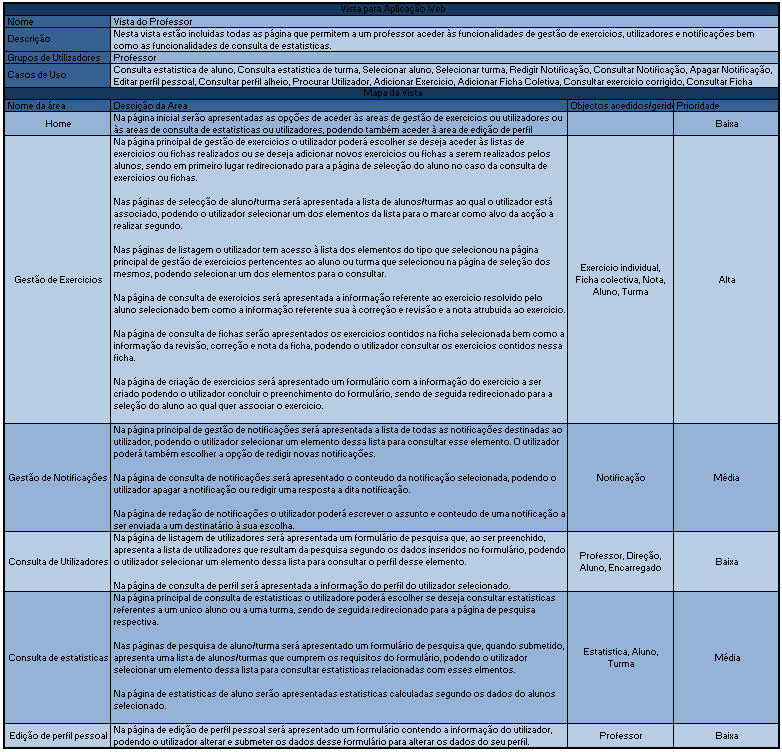
# Grupos de Utilizadores

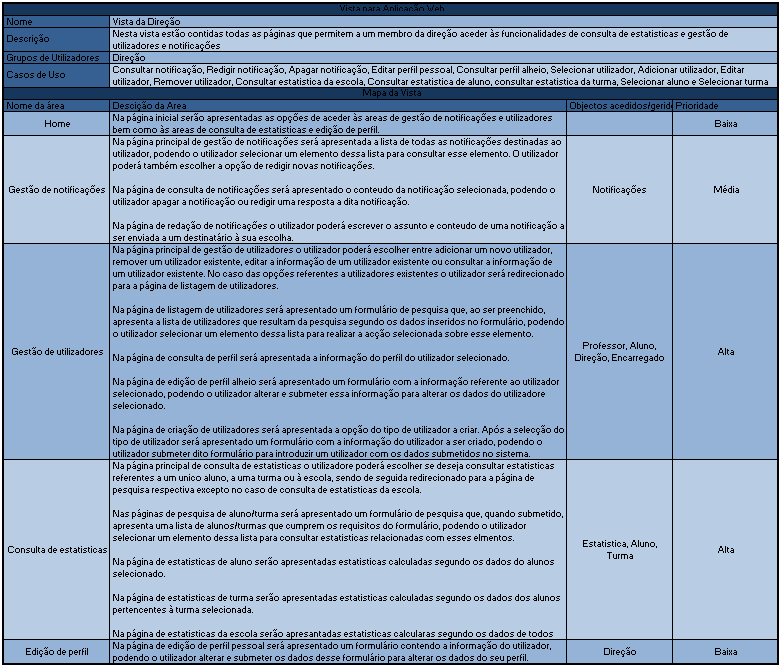
Durante o processo de planeamento e modelação da aplicação, foi necessário determinar e detalhar os grupos de utilizadores que terão acesso à mesma. Como tal, determinamos três grupos de utilizadores: Alunos, Professores, Encarregados e Educação e a Direção. Os alunos tratam-se do alvo principal da aplicação, no entanto ele possuem acesso reduzido em comparação com os restantes utilizadores dado que não devem possuir controlo sobre os dados e irão, principalmente, realizar exercicios e consultar correções. Os professores terão um nivel de controlo superior, sendo capazes de criar e avaliar exercicios, bem como consultar uma diversidade de dados. Os encarregados não terão acesso a nenhuma funcionalidade de controlo, podendo apenas consultar dados de diversos tipos. A direção poderá consultar qualquer tipo de dados e terá controlo sobre os restates utilizadores. Em seguida apresentamos as tabelas que contêm os detalhes da cada um dos grupos de utilizadores:

# Site Views

Cada grupo de utilizadores terá acesso a uma Site View que determina as páginas a que cada utilizador terá acesso. Cada view contem um conjunto de àreas que, por sua vez, contêm um conjunto de páginas. Cada àrea representa o conjunto de páginas que permitem ao utilizador aceder um conjunto de funcionalidades como manipulação/consulta de dados de utilizadores. Cada página permite ao utilizador aceder a funcionalidades especificas como inserção ou remoção de utilizadores. De seguida apresentamos um conjunto de tabelas que especificam e detalham cada Site View, bem como as suas àreas e páginas:

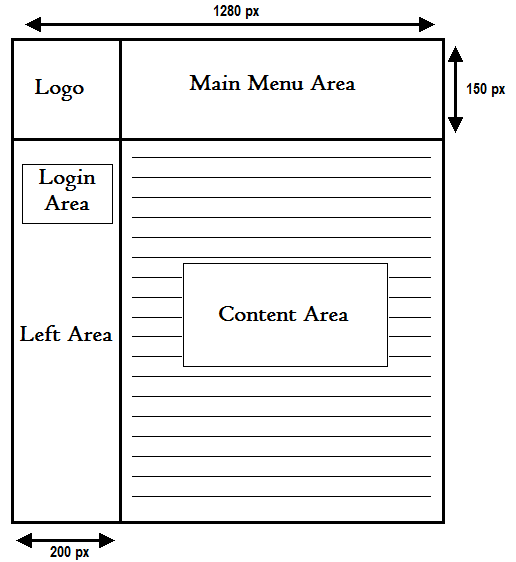


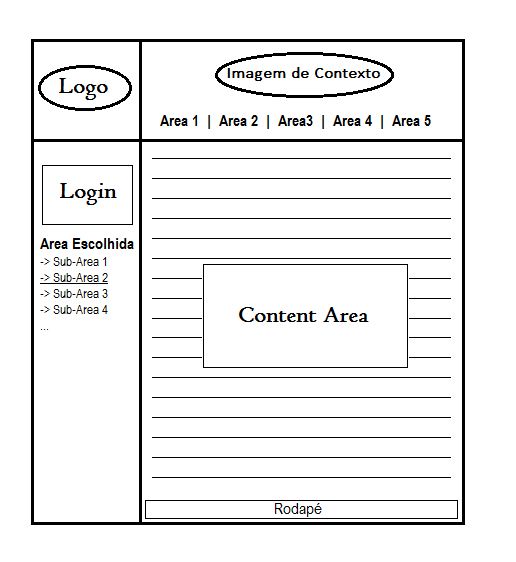




# Style Guidelines

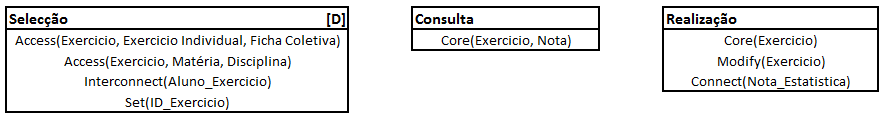
Numa fase inicial foram definidos alguns detalhes visuais que iriam servir como "guias" para a construção das páginas que constituem a aplicação. O propósito destas Style Guidelines è criar um modelo base que serviria como exemplo do aspecto visual que as páginas seguiriam durante a implementação da aplicação. Estas guideline definem alguns detalhes, como tamanho e posicionamento de determinadas secções ou menus. Embora o aspecto final das páginas seja diferente dos exemplos projectados inicialmente, estes continuaram a ser usados como exemplos base na sua criação. De seguida apresentamos as imagens representantes dessas Style Guidelines:

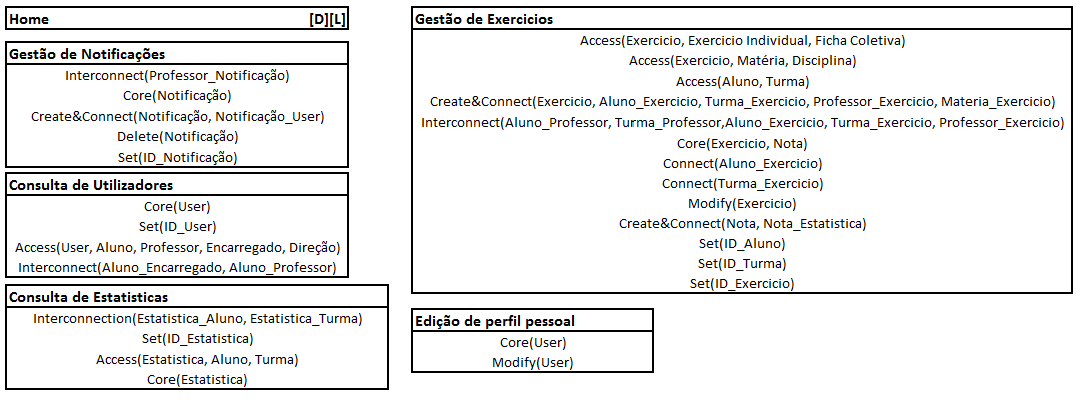


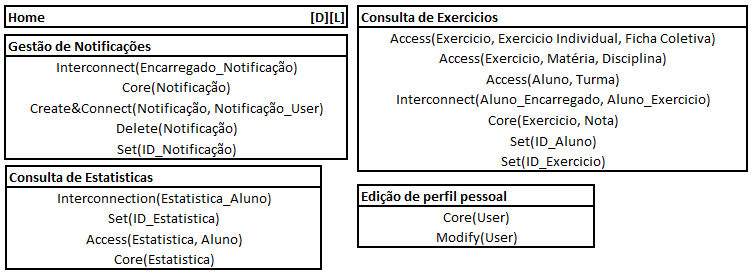


# Coarse Design

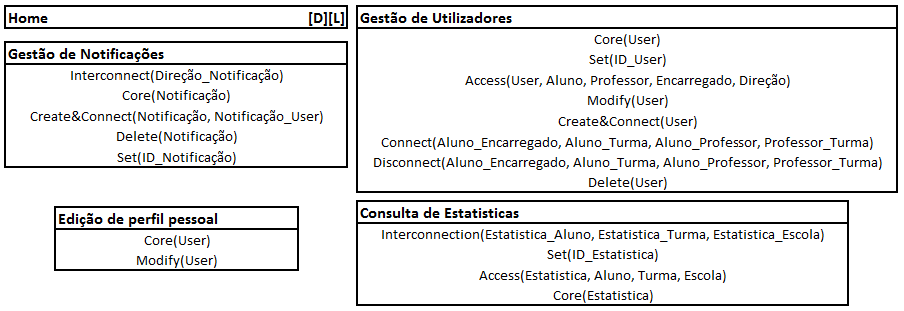
Um dos passos iniciais do design da aplicação, foi a criação de um Coarse Design, um rascunho inicial do funcionamento das areas de cada vista dos utilizadores. Este coarse design descreve varias funcionalidades, acessos a dados, interligações a dados, etc. . O coarse design foi utilizado como a base do detailed design, bem como outros aspectos do design da aplicação. Por sua vez, o coarse design foi baseado em informação estabelecida durante a modelação da aplicação como, por exemplo, a modelação das Site Views.

De seguida apresentamos as tabelas que definem o coarse design da vista dos alunos:

Apresentamos também as tabelas que definem o coarse desgin da vista dos professores:

O coarse design da vista dos encarregados é apresentado de seguida:

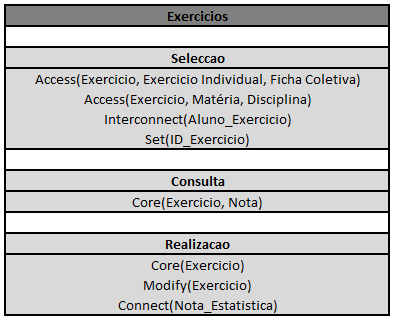
E, por fim, apresentamos o coarse design da vista da direção:

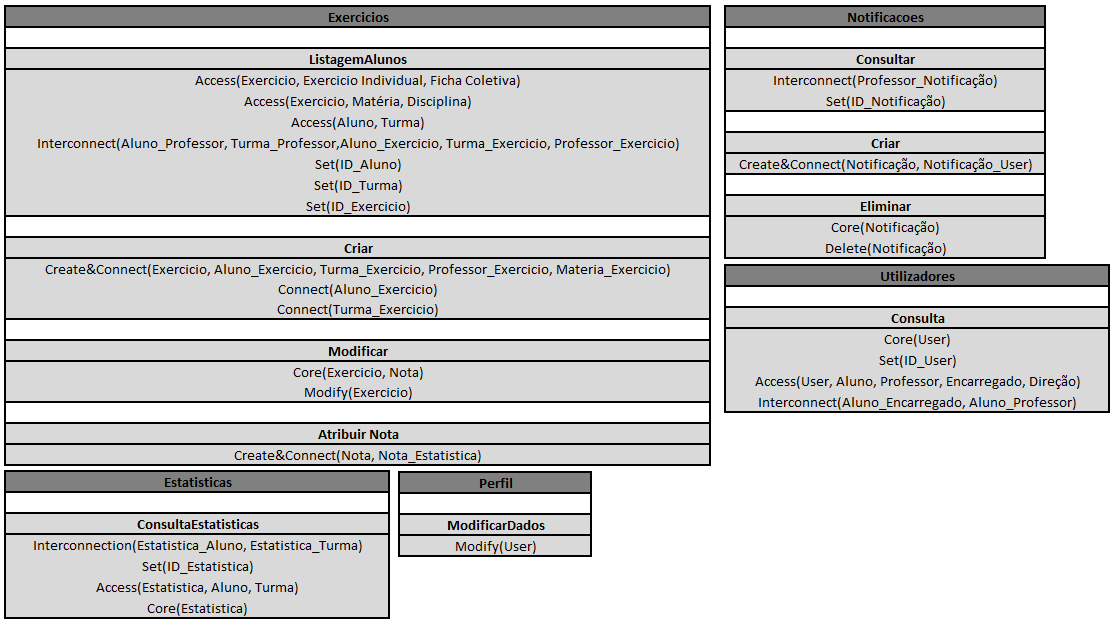


# Detailed Design

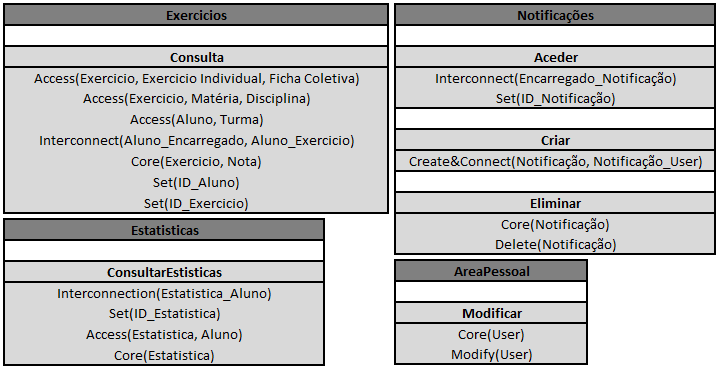
Após a criação de um Coarse Design, procedemos à criação de um Detailed Design, uma versão mais detalhada do coarse desgin que indica o funcionamento de cada página pertencete a cada área, repartindo as funcionalidades indicadas no coarse design pelas várias páginas que cada área possui. Deste processo resulta um conjunto de dados sobre cada página que pode ser utilizado directamente na criação de um modelo IFML.

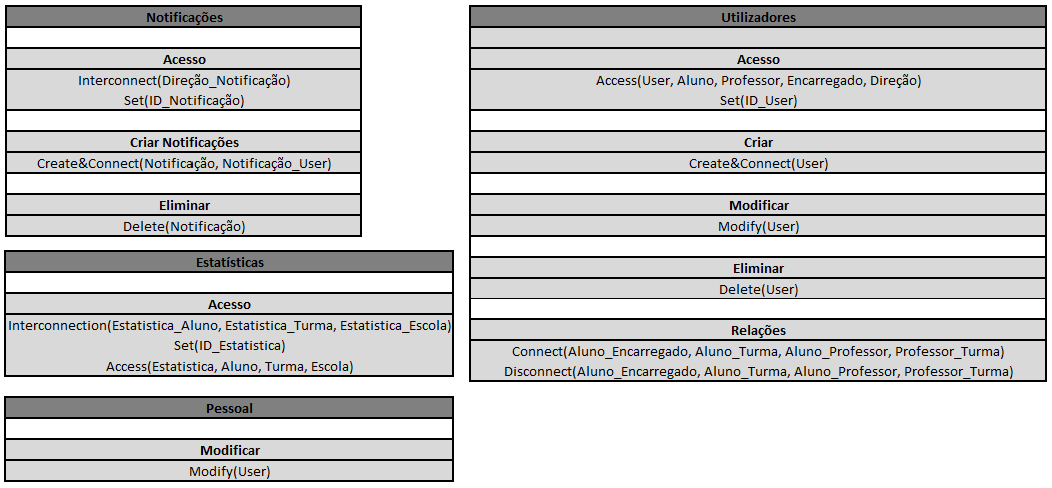
De seguida apresentamos os dados do detailed design referentes à vista de alunos:



Abaixo apresentamos tambem os dados do detailed design referentes à vista de professores:

Apresentamos tambem os dados do detailed design referentes à vista de encarregado de educação:

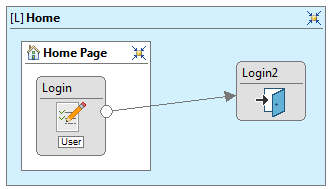


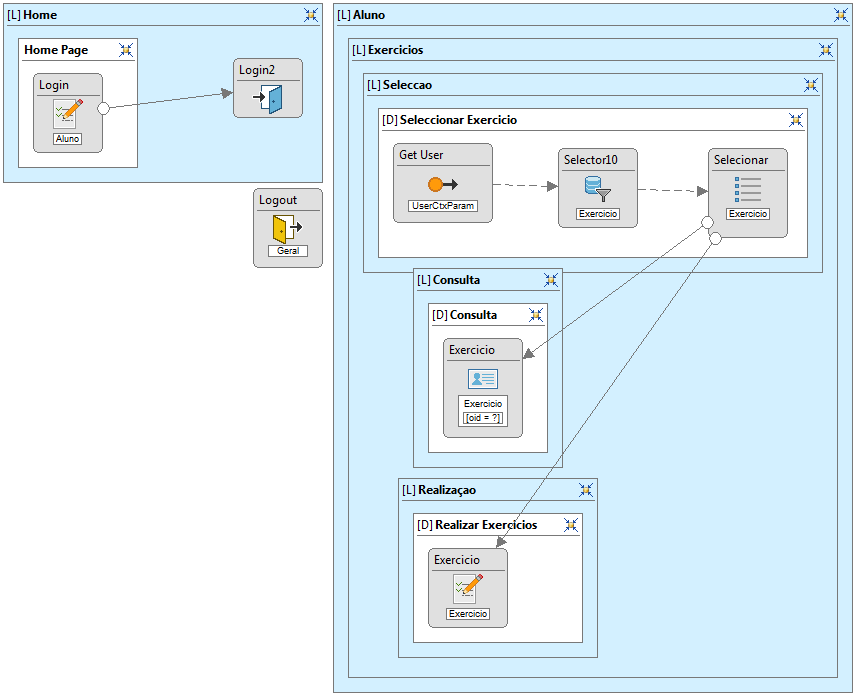
E, por fim, apresentamos os dados do detailed design referentes à vista da direção:

# Especificações WebRatio

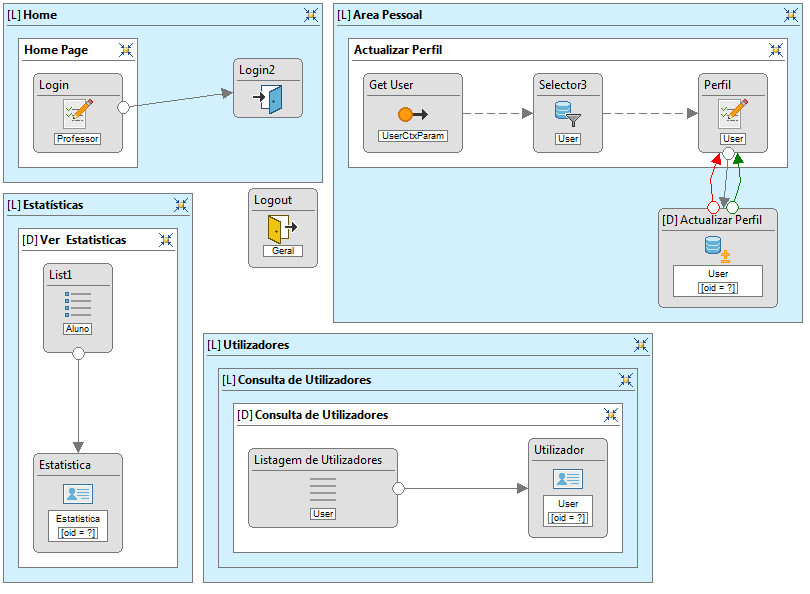
Um dos passos finais antes de procedermos à implementação da aplicação, foi a criação de especificações de cada site view utilizando a ferramenta WebRatio. Estas especificações são realizadas na forma de diagramas que apresentam as diveras áreas, as suas páginas, os componentes que determinam as interações entre os utilizadores, o sistema e a base de dados e os fluxos entre todos esses elementos que determinam as ligações entre componentes e a navegabilidade da aplicação. Cada site view apresentada abaixo contem todas as àreas, páginas e funcionaldidades descritas na descrição das site views, bem como no detailed design

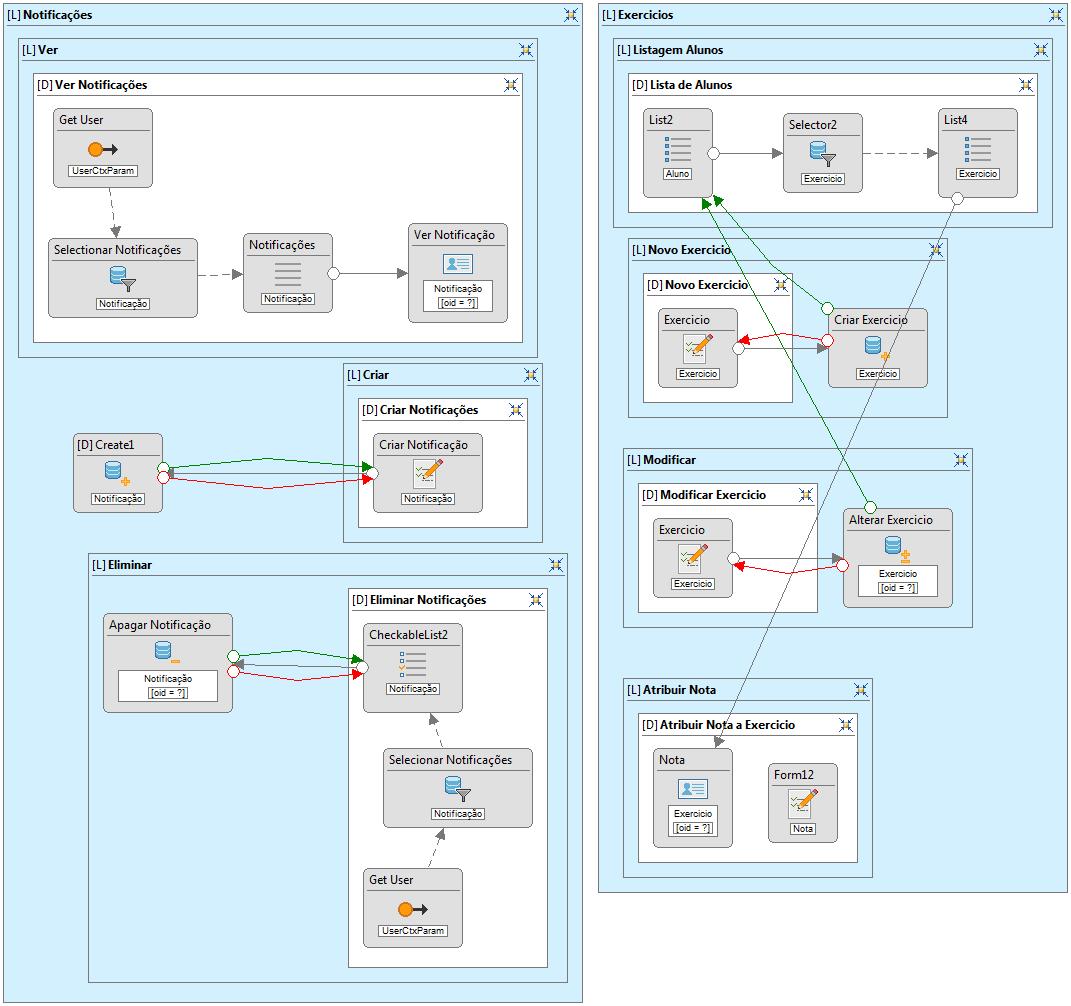
Em primeiro lugar apresentamos a site view geral, que é apresentada ao utilizador não registado e que lhe permite fazer login usando as suas credenciais:



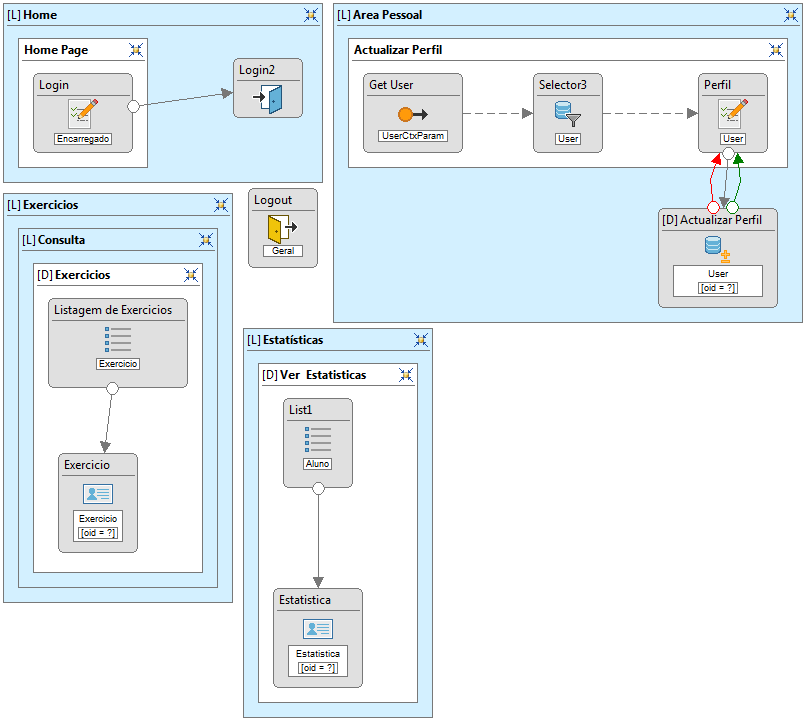
De seguida apresentamos a site view dos alunos, que contem as àreas home e exercicios:

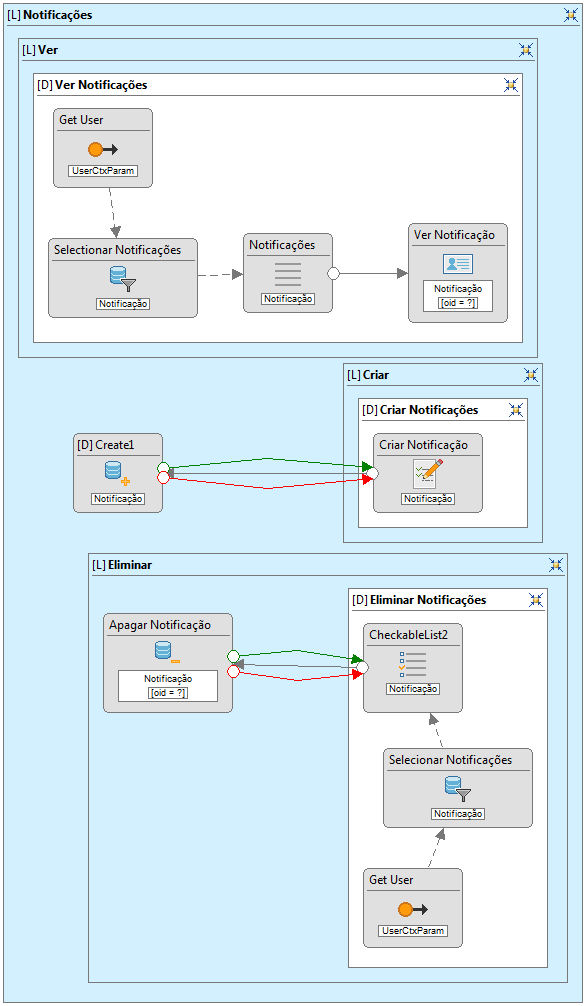
A site view dos professores contem as àreas home, exercicios, àrea pessoal, estatisticas, utilizadores, e notificações e è apresentada de seguida:



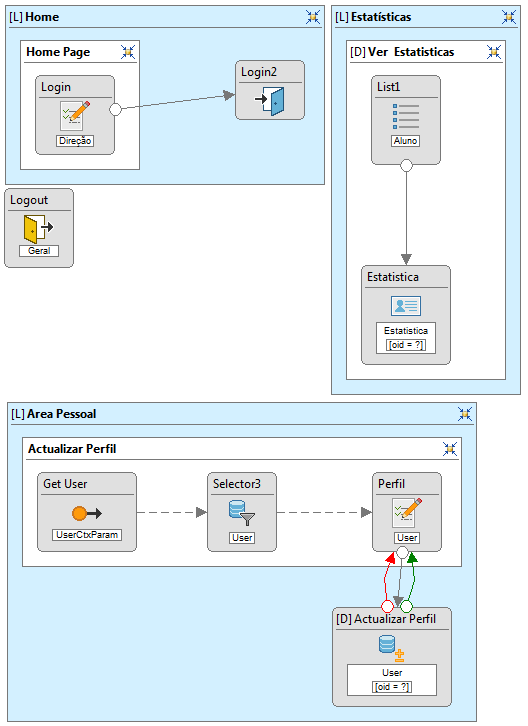


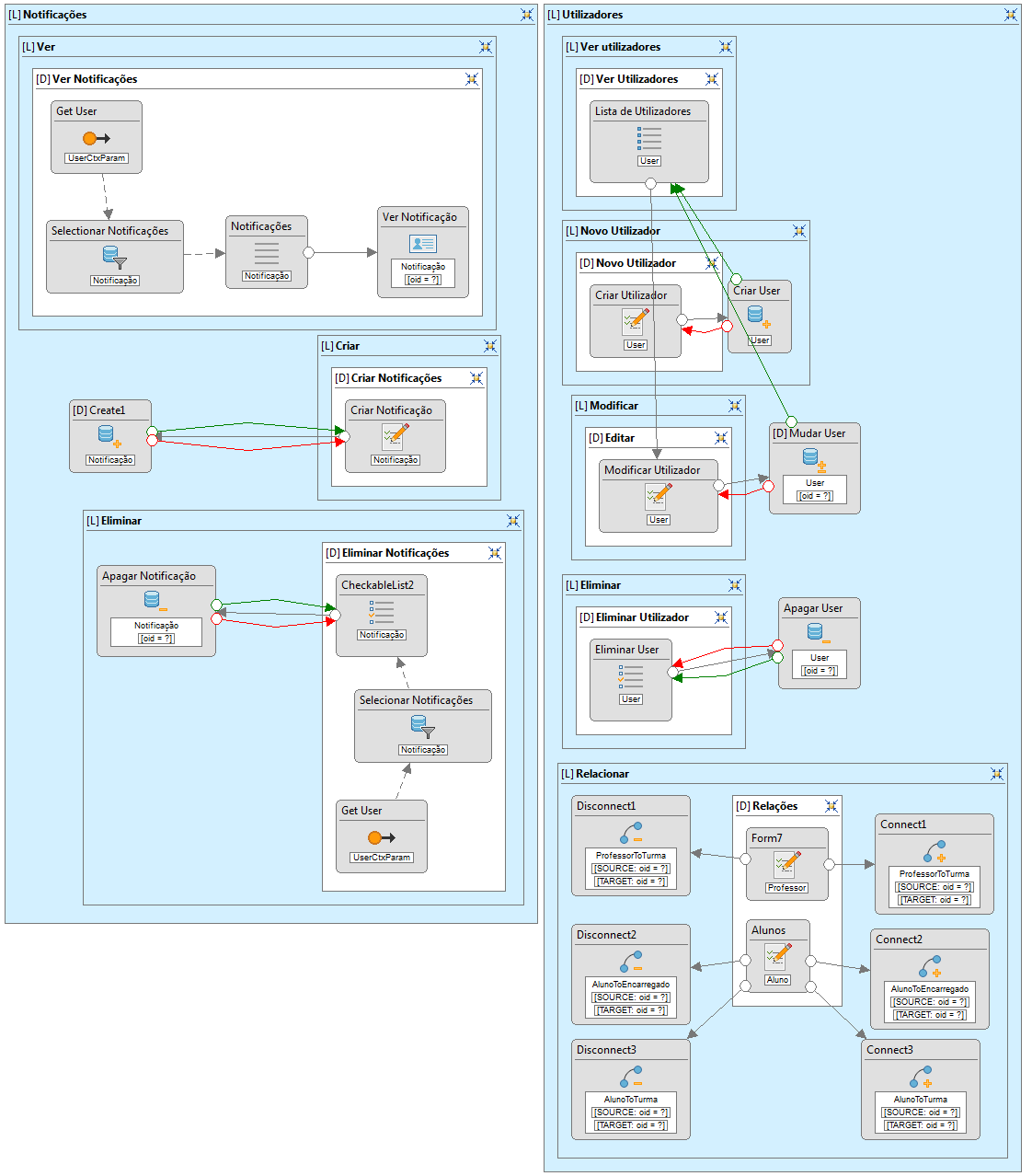
Apresentamos de seguida a site view dos encarregados que contem as àreas home, notificações, estatisticas, exercicios e àrea pessoal:





Por fim, apresentamos a site view da direcao que contem as àreas home, notificações, estatisticas, utilizadores e area pessoal:





Estas site views, em conjunto com o modelo de dados também criado com a ferramenta WebRatio, foi-nos possivel gerar uma aplicação que serviu de base para a implementação da aplicação final.

# Implementação

Após a modelação, foi decidida a linguagem que seria usada para a implementação. Depois alguma análise, o grupo decidiu utilizar a ferramenta Ruby on Rails, que como o nome sugere tem como base a linguagem Ruby. A escolha da mesma deveu-se a alguma experiencia anterior de alguns elementos do grupo com a ferramenta, e também a facilidade de aprendizagem que esta apresenta.

# Ruby on Rails

Antes de mais iremos apresentar o Ruby on Rails, a sua estrutura e conceitos.

O Ruby on Rails é uma ferramenta dedica ao desenvolvimento de aplicações web orientadas a base de dados, e tem como principal característica a facilidade no desenvolvimento de sites. O Rails tem uma estrutura e conceitos muito bem definidos para tal usa a Arquitetura MVC (Model-View-Controller) para definir as suas aplicações, ou seja, uma arquitetura que se baseia em Modelos, que serão as nossas classes, as Views que serão as nossas páginas HTML e os Controladores, que responsáveis pelos métodos e gerir todos os pedidos específicos da aplicação.

É de referir que esta ferramenta tem o seu próprio servidor, sendo que para poder simular a aplicação apenas temos de escrever no browser localhost:3000.

# Desenvolvimento do Site

## Geração de Modelos

Depois de instalada a ferramenta, foram feitas algumas experiencias de forma a familiarizarmos com a mesma.

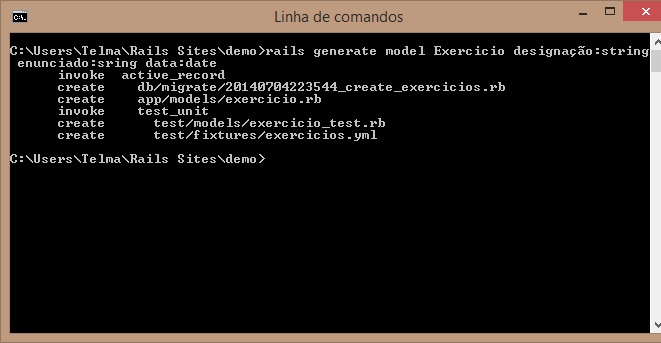
O primeiro passo para começar com a aplicação foi definir os modelos, que foram anteriormente visto no modelo de dados.

Para gerar um modelo apenas é necessário executar na linha de comandos o seguinte comando:

*- rails generate model Exercicio designação:string enunciado:sring data:date*

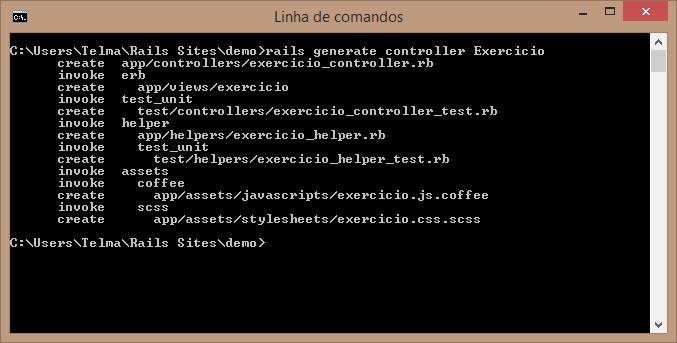
Através deste comando foi gerado o model Exercicio, com os campos, **designacao** e **enunciado** ambas do tipo string e a **data,** com o tipo date.

Aquando deste comando são gerados os ficheiros representados na imagem seguinte.



## Geração de Controladores

De seguida foi feita a geração de um controller para o model através do seguinte comando: - *rails generate controller Exercicio* que por sua vez gera os seguinte ficheiros.



Ambos os comandos foram executados para definir os todos as restante classes:

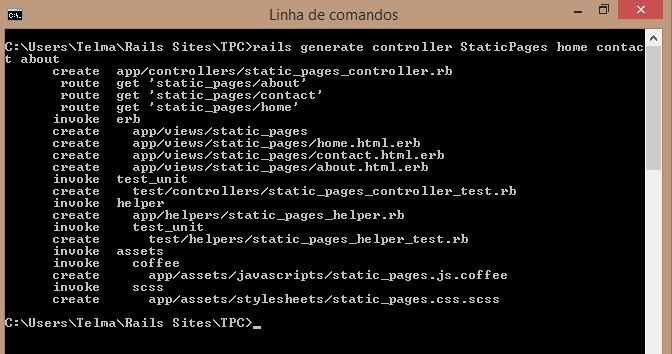
* Professores
* Alunos
* Direção
* Encarregados de Educação
* Mensagens
* Exercícios
* Nota
* Turma

## Páginas Estáticas

De seguida foi gerado um controlador para as páginas estáticas, que estarão sempre presentes seja qual for o conteúdo da página e que permitem um acesso rápido a outras “zonas” do site, foram definidas as páginas Home, Sobre e Contactos.

Para isso foi executado o seguinte comando:

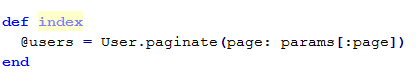
*- rails generate controller StaticPages home contact about*



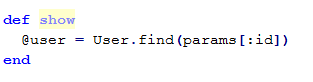
## Métodos

O Passo seguinte foi definir os métodos necessários de cada controlador e views.

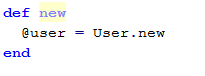
O primeiro modelo a ser definido, e que servirá de exemplo ao longo do relatório foi o de Utilizador, para que depois se possa registar utilizadores e fazer login e logout na aplicação. De seguida estão dos métodos implementados no controlador **user\_controller.rb**:

**Index** – lista todos os utilizadores

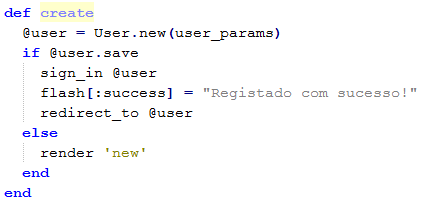
**Show** – Mostra um utilizador



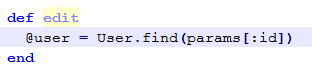
**New** – Pagina para fazer novo utilizador (Registo)



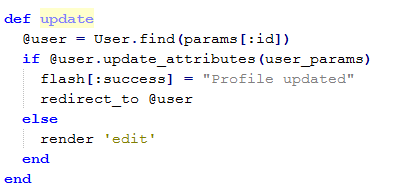
**Create** – Cria novo User



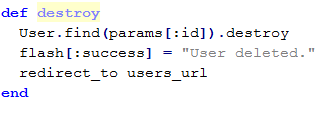
**Edit** – Editar Utilizador



**Update** – Atualiza Utilizador

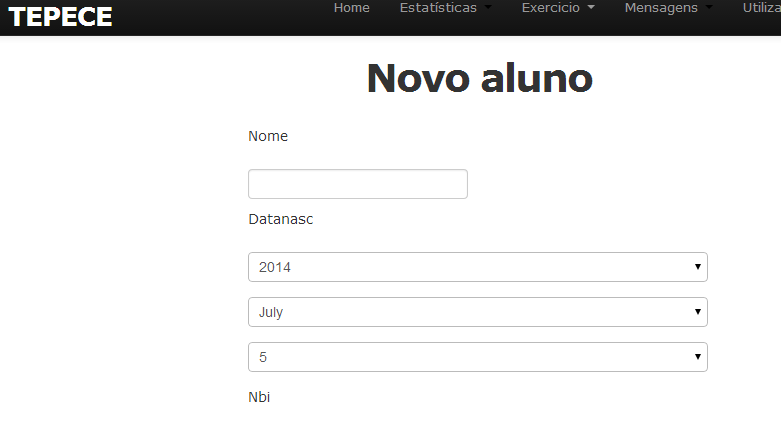


**Destroy** – Apaga Utilizador

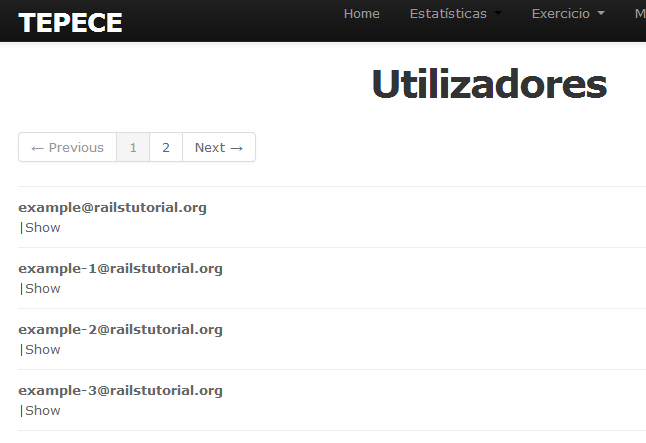


## Views

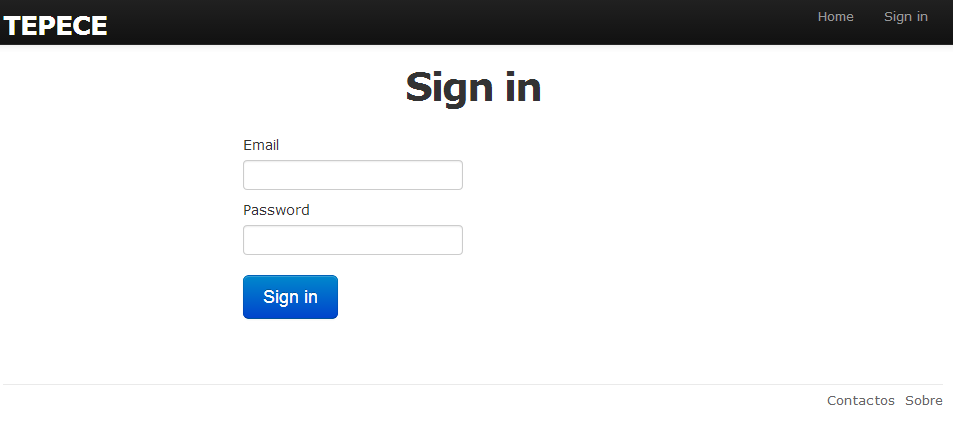
Depois de definidos todos os métodos, foram desenvolvidas as Views para cada página. Neste caso cada método definido até agora origina uma página. A imagem seguinte simula a página de inserção, que é idêntica à de edição, de um novo Aluno. O Processo é igual para os restantes utilizadores.



Página de Listagem de Utilizadores



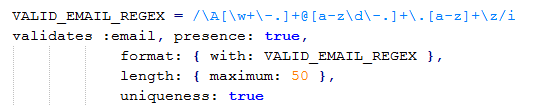
Depois de registados os Utilizadores poderão fazer Login.



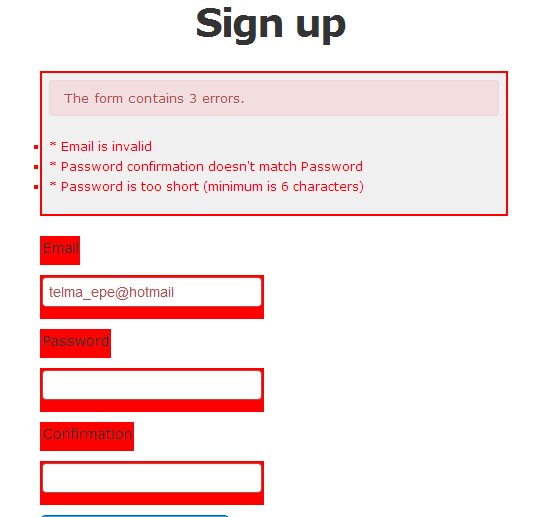
## Validações de Dados

Foram implementadas algumas estratégias de validação para que todos os campos onde possam ser inseridos dados sejam validados. Em alguns casos foram apenas foram validados a presença de dados, e noutro extremo é exigido que sigam uma forma específica.

Na seguinte imagem é mostrada como é feira é validação para um email.



Depois das validações caso, os dados não cumpram os requisitos os erros são mostrados no ecran da seguinte forma:



## Autenticação e Permissões

Os utilizadores apenas terão acesso a outras partes do site depois de se autenticarem, e cada tipo de utilizador terá as suas permissões.

## Layout

Para a apresentação do site foi usada a gem do bootstrap, que nos fornece vários templates possíveis.