

# Rate Me

Leonel Freitas<sup>1</sup>, Nádia Vieira<sup>1</sup>, Vitor Gavina<sup>1</sup>

<sup>1</sup> Universidade da Madeira – Faculdade de Ciências Exatas e Engenharias, Portugal

<sup>2</sup> Aplicações Centradas em Redes

**Resumo.** Um dos obstáculos do trabalho em equipa num contexto universitário ou escolar, é muitas vezes, não existir cooperação e entendimento entre os elementos de um determinado grupo. Isto acontece na medida em que podem encontrar-se em níveis dispares qualitativamente ou não existe motivação para elaborar um bom trabalho, ou ainda, por existirem áreas de interesse/prática distintas. Desta forma, uma plataforma que permita classificar e selecionar os elementos com quem trabalhamos em grupo, num contexto académico/escolar, torna-se numa ferramenta essencial a existir. Neste relatório pretende-se efetuar uma descrição da plataforma Rate Me, desde a sua conceção e motivação passando pelo seu processo de desenvolvimento no âmbito da cadeira de Aplicações centradas em Redes.

## 1 Introdução

O objetivo primordial é que, Rate Me, resulte numa plataforma online que permita classificar as pessoas com quem trabalhamos em grupo num contexto académico/escolar, e aferir quais são as instituições de ensino em que o nível qualitativo dos alunos é mais homogéneo. Esta ferramenta permitiria aos docentes gerirem de melhor forma como os grupos para trabalhos são formados, e possibilitaria aos alunos optarem melhor com quem trabalham, assim como aumentar a sua motivação para serem avaliados positivamente pelos seus colegas, e melhorarem em determinadas habilidades sociais inerentes ao trabalho em grupo.

No presente relatório procuramos, de forma explícita e clara, relatar todo o processo que envolveu a criação desta plataforma com a *framework Laravel*. Framework esta que era desconhecida para o grupo. Logo foi necessária muita investigação para apreender e dominar esta ferramenta que apresenta bastante potencial para o desenvolvimento web. Este processo revelou-se desafiante e vantajoso numa perspetiva de aprendizagem, já que o *Laravel* demonstrou ser uma ferramenta extremamente completa e eficiente.

## 2 Problema

Pretende-se colmatar uma lacuna identificada, a falta de cooperação e entendimento entre os elementos de um determinado grupo, bem como, a dificuldade para os alunos em criar grupos de trabalho em contextos académicos/escolares. Então, a ideia é que um dado grupo de utilizadores cria um perfil, por exemplo, no secundário, alunos que trabalham entre si num determinado projeto, no final desse trabalho, cada um dos elementos participantes, avalia os colegas em aspetos como: realização, compressão, empenho, eficiência e qualidade do trabalho realizado. Após esta avaliação o professor da disciplina, valida as notas atribuídas. Isto permitiria que os alunos tenham maior motivação para trabalhar ativamente e criarem o seu "portfolio" virtual de projetos, assim como de contatos. Para os docentes existe a vantagem de terem a possibilidade de gerirem melhor os trabalhos de grupo, para que os alunos com maior classificação não trabalhem só entre si. Com os dados estatísticos recolhidos seria também possível fazer varias análises importantes para o melhor funcionamento das instituições de ensino.

Desta forma, o que se ambiciona é resolver o problema da formação de grupos de forma satisfatória e racional, incentivar a motivação/empenho para a contribuição nos projetos e, proporcionar melhor controle/gestão de projetos para os docentes.

## 3 Revisão de literatura e plano de ação

### 3.1 Revisão de literatura

Em termos de documentação, o *Laravel*<sup>1</sup> encontra-se bastante bem documentado graças a comunidade extensa e ativa que possui, utilizamos primeiramente como meio de pesquisa sítios relacionados com a plataforma como o site oficial. Além da pesquisa associada a *framework*, realizou-se ainda pesquisa das tecnologias associadas ao desenvolvimento web como *Javascript*<sup>2</sup>, *Ajax*<sup>3</sup>, *css*<sup>4</sup>, *mysql*<sup>5</sup>.

### 3.2 Plano de ação

Como ponto de partida, realizamos um análise aos requisitos que a aplicação deveria possuir, e se cumpriam com o que era pedido no enunciado do trabalho.

---

<sup>1</sup> «Laravel - The PHP Framework For Web Artisans».

<sup>2</sup> «JavaScript Tutorial».

<sup>3</sup> «jQuery.ajax() | jQuery API Documentation».

<sup>4</sup> «CSS Tutorial».

<sup>5</sup> «MySQL Tutorial 1: Overview, Tables, Queries».

Apos está análise procedemos a identificação dos principais Atores (Instituições de Ensino, Alunos e Professores) a elaboração de casos de utilização onde podemos assim esquematizar as ações mais importantes da aplicação equais os utilizadores que as podem efetuar.



Figura 1 Diagrama de Casos de Utilização Rate Me

Posteriormente efetuamos o desenho do nosso esquema relacional que se encontra em anexo (Figura 4) usando a ferramenta *mysql Workbench* para servir como guia.

Concluídas estas etapas efetuamos protótipos de baixa fidelidade em papel para obtermos o layout básico do website. Findados todos estes passos procedeu-se então ao início da implementação dividindo as tarefas de acordo com os casos de utilização encontrados.

## **4 Solução Proposta**

### **4.1 Descrição geral da solução**

A solução idealizada para o(s) problemas(s) anteriormente identificados, traduz-se na criação de uma aplicação que vá de encontro às necessidades num contexto de projetos escolares/académicos, em grupo. Podemos de certa forma considerar que as proposições chave da aplicação que ajudam a resolver os problemas previamente identificados são o sistema de avaliação, e a mais fácil criação de grupo de trabalhos. Podemos definir as mais valias da nossa aplicação da perspetiva do aluno e dos professores.

#### **4.1.1 Perspetiva do aluno**

- Encontrar membros para grupo;
- Avaliar membros com quem trabalhamos;
- Melhor produção de trabalho;
- Trabalhar com pessoas com quem nos identificamos;
- Identificar membros com áreas de interesse comuns.

#### **4.1.2 Perspetiva do professor**

- Visualizar os Grupos formados nas disciplinas formadas
- Visualizar o Rating de cada grupo formado
- Visualizar os alunos que ainda não possuem grupo

## 4.2 Descrição geral técnica da solução implementada

Usando a framework Laravel como ponto de partida procedeu-se a implementação da solução, tentando respeitar sempre o modelo MVC<sup>6</sup> (Model View Controller). Na figura abaixo podemos visualizar de forma esquematizada o *workflow* do Laravel e do funcionamento geral de uma arquitetura MVC.

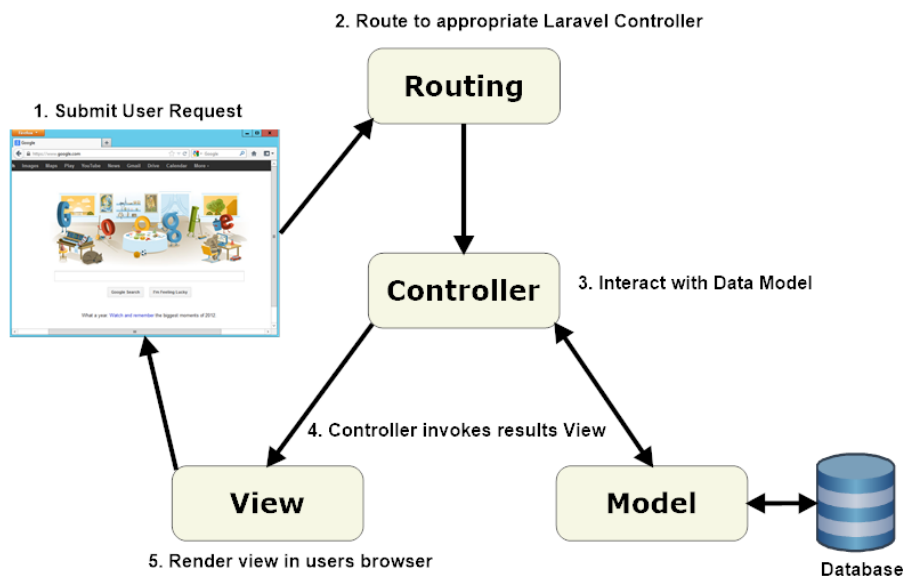


Figura 2 Workflow do Laravel em MVC fonte: <http://laravelbook.com/laravel-architecture/>

De uma forma geral, podemos descrever este fluxo de trabalho como o seguinte. O utilizador faz um pedido como por exemplo mudar de pagina, este pedido possui uma rota associada, rota essa que tem um *controller* associado com uma função para tratar diferentes tipos de pedidos. O *controller* interage com o Modelo que por sua vez interage com a Base de dados que retorna à informação pretendida. O *controller* posteriormente passa essa informação a *view* que faz a apresentação da informação pedida pelo utilizador.

Passemos então a definir para a nossa aplicação quais os componentes mais importantes, dos Modelos, Views e Controllers.

<sup>6</sup> «Understanding MVC: Model-View-Controller».

### 4.3 Models:

Os modelos são o nível mais baixo da arquitetura MVC ficam encarregues de efetuar a conexão com a base de dados. No laravel<sup>7</sup> usa-se o padrão *Active records*<sup>8</sup> cada base de dados possui um modelo que é usado para interagir com a tabela em questão o comando *php artisan make model:"nome\_do\_modelo"*.

#### Alguns dos Modelos Utilizados:

- Student\_grades
- Discipline
- Institution
- User
- Group
- Perfil

### 4.4 Views

As Views correspondem ao *front-end* do website, são responsáveis pela apresentação da informação ao utilizador. O nosso site faz igualmente uso de uma das *features* do Laravel, os *blade Templates*<sup>9</sup> que é um *templating engine*<sup>10</sup> que permite que se herde e reutilize secções de paginas.

No caso da nossa aplicação a secção principal que se repete em todas as *views* que consiste na barra de navegação o *footer* e o background da pagina, é reutilizada pelas diferentes *views*. As principais *views* implementadas na nossa aplicação são as seguintes:

- **Perfil:** apresenta as informações pessoais, rating de avaliação e avatar do utilizador
- **Gerir Disciplinas e Grupo:** apresenta informações relacionadas com a subscrição, de disciplinas, e criação de grupos de trabalho
- **Procurar Grupos:** apresenta os grupos formados na disciplina escolhida pelo utilizador assim como os alunos que não possuem grupos
- **Avaliar Grupos** diz respeito a apresentação de quais os grupos aos quais o aluno pertence, cujo os alunos são passíveis de serem avaliados no momento

---

<sup>7</sup> «Eloquent: Getting Started - Laravel - The PHP Framework For Web Artisans».

<sup>8</sup> «Working with Databases: Active Record | The Definitive Guide to Yii | Yii PHP Framework».

<sup>9</sup> «Laravel - The PHP Framework For Web Artisans».

<sup>10</sup> «Blade Templates - Laravel - The PHP Framework For Web Artisans».

## 4.5 Controllers

Os controllers fazem a interligação entre as *views* e os modelos<sup>11</sup>, no fundo controlam o comportamento da aplicação. No Laravel para criar um *controller* podemos utilizar o comando ***php artisan make controller: "nome\_do\_controller"***. Na nossa aplicação realizamos um abordagem de um *controller* por *view* ou por cada duas *views*. É possível que pudéssemos ter reduzido o numero de *controllers* visto alguns comportamentos serem comuns a varias *views*, porém este tipo de abordagem também traz a vantagem de separar as responsabilidades<sup>12</sup> de *views* diferentes por *controllers* distintos, assim caso ocorra algum problema com um *controller* apenas uma *view* é afetada. Os principais *controllers* implementados são os que se seguem:

- **GerirDiscpGroupController:** Este *controller* trata das funcionalidades referentes a gestão das disciplinas e grupos do utilizador, ou seja, subscrição/anulação de disciplinas do curso do utilizador, e ainda a criação de grupos para as disciplinas em que o aluno se encontra inscrito.
- **PerfilController:** trata das funcionalidades referentes aos dados necessários para a visualização dos dados do utilizador, tais como nome e perfil (Lider, comunicativo, etc) a avaliação e rating atuais do utilizador.
- **ProcurarGrupoController:** Trata das funções referentes a procura de grupos, tais como procurar alunos sem grupo, grupos formados, e inscrição em grupos já formados.
- **Avaliar Controller:** É responsável pela logica de avaliar utilizadores, interagindo com os modelos de user , group , student\_grade , e as views avaliar.
- **ApiController:** disponibiliza as funções que permitem a consulta de dados por intermedio de aplicações externas, que podem usar esses dados para integrar com outra aplicação. São disponibilizadas de momento as funções de api para listar Instituições, procurar cursos numa data Instituição e disciplinas associadas, visualizar quais os utilizadores da aplicação professores ou alunos e ainda visualizar o numero de grupos criados na aplicação ate ao momento.

---

<sup>11</sup> «Understanding MVC: Model-View-Controller».

<sup>12</sup> Makabee, «Separation of Concerns».

#### 4.6 Descrição de aspetos particulares importantes

Foram vários os aspetos tidos em conta para a implementação da nossa aplicação. Um dos aspetos considerados, foi no que respeita às decisões de aspeto visual, *CSS*. Optamos sempre por criar classes e ids ao longo da elaboração de código e formatar o aspeto visual completo num único ficheiro externo (*homepage.css*). Assim, o nosso site preza por um visual uniforme, constante e funcional para uma navegação intuitiva e fácil, dado que um dos objetivos iniciais quando a aplicação foi concebida era de também ser facilmente escalável fora do contexto académica, ou seja, pudesse ser utilizável noutros moldes pelos alunos do secundário, ou 3º ciclo, daí que o layout tente ser o intuitivo possível.

A autenticação<sup>13</sup> do nosso website foi implementada de forma automática pelo Laravel fazendo uso do comando *php artisan make:auth*, isto gera toda a lógica e apresentação (*models views e controllers*), relacionados com a autenticação e registo de utilizadores. O que foi feito pois necessitamos de mais informação por parte do utilizador aquando do seu registo, tal como a sua instituição de ensino, o seu curso, foi alterar estes modelos, *controllers*, e *views pre-feitos* de forma a satisfazer os requisitos da nossa aplicação.

#### 4.7 Dependências Externas

Introduzimos ainda uma *api* externa neste caso o *Recaptcha*<sup>14</sup> da Google de forma a proteger o site de possíveis *bots* e *spam* e assim cumprir com um dos requisitos do projeto de utilizar uma aplicação externa. Esta funcionalidade foi implementada no registo de utilizador e no login. Para facilitar a interação com as imagens fez-se uso da livreria *Image intervention*<sup>15</sup>, que facilita a manipulação de imagem, neste caso usamos esta biblioteca para implementar a imagem de utilizador a qual chamamos de avatar. O utilizador ao criar a conta fica com um avatar predefinido neste caso uma foto de perfil de utilizador comum descaracterizada, que pode posteriormente alterar no seu perfil, para uma foto que o identifique, deste modo é mais fácil aos outros utilizadores identificarem o utilizador em questão, e transmite-se um sentimento de posse do utilizador em relação a aplicação. Foi ainda realizada algumas tentativas para implementar validação *cliente side*<sup>16</sup> por meio de uma biblioteca externa porem não foi possível coloca-la em funcionamento, na discussão abordamos mais em detalhe esse problema.

---

<sup>13</sup> «Authentication - Laravel - The PHP Framework For Web Artisans».

<sup>14</sup> «reCAPTCHA: Easy on Humans, Hard on Bots».

<sup>15</sup> «Intervention Image - Introduction».

<sup>16</sup> «web development - What are the differences between server-side and client-side programming?»



#### 4.8 Opções de Implementação

Durante o desenvolvimento da aplicação optou-se por utilizar as potencialidades do Laravel, para a implementação de formulários. O Laravel permite através dos seus *Forms e Html* <sup>17</sup>, limitar a repetição de código ao implementar formulário e código html desses formulários com poucas linhas de código. Esta forma torna na nossa opinião o código mais legível e permite uma maior produtividade, embora por vezes seja menos flexível do que a implementação tradicional.

### 5 Utilização da aplicação

Apos chegar ao site o utilizador visualiza a pagina de entrada, a partir daqui caso possua uma conta, pode efetuar o *login*, caso contrario proceder ao registo de utilizador, introduzir os seus dados, o seu tipo de utilizador, aluno ou professor, qual a instituição de ensino a que pertence, e o seu perfil, em baixo apresentamos os perfis que se encontram disponíveis. A ideia é que os grupos possuam uma heterogeneidade de perfis e os utilizadores possam diferenciar as suas diferentes valias e preferência relativas ao trabalho em grupo.

**Líder:** corresponde a um utilizador que no decorrer dos trabalhos em grupo assume um papel de líder.

**Comunicador:** um utilizador que se sente confortável em realizar apresentações e se expressar oralmente.

**Bastidor:** um utilizador que prefere trabalhar sem assumir um papel de líder ou de mais relevância limitando-se a realizar o trabalho para o qual é destacado.

**Criativo:** adequa-se aos tipos de utilizador que gosta de tentar abordagens diferenciadas e tentar caminhos alternativos ao que normalmente é tido como padrão.

---

<sup>17</sup> «Forms & HTML - Laravel - The PHP Framework For Web Artisans».

## Home Page

Apos efetuar o login com sucesso o utilizador encontra a *home page*, onde podemos visualizar o menu de navegação vertical com as varias opções disponíveis do utilizador. A aplicação encontra-se bastante intuitiva de utilizar dai que as varias paginas sigam uma abordagem muito similar de forma a que a apresentação seja consistente e mais fácil de utilizar e memorizar para o utilizador.

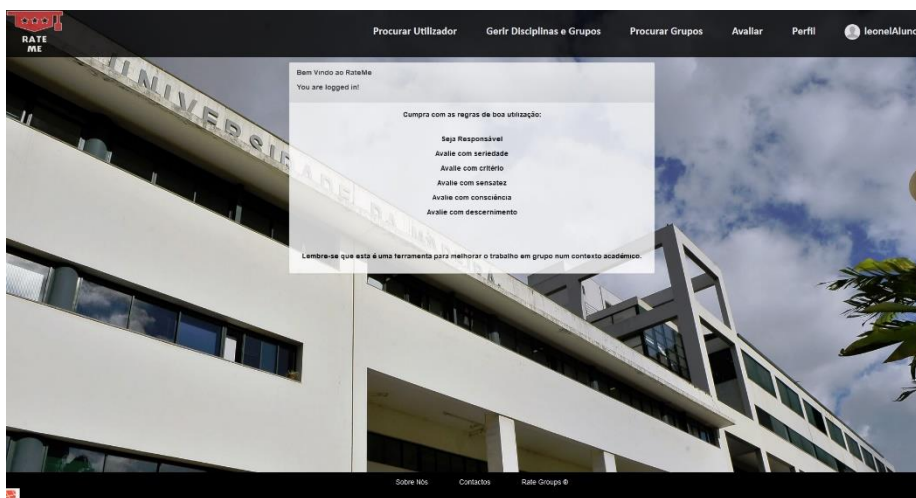


Figura 3 Navegação Rate Me

Abaixo segue-se o Mapa de Navegação, detalhando melhor a utilização da nossa aplicação.

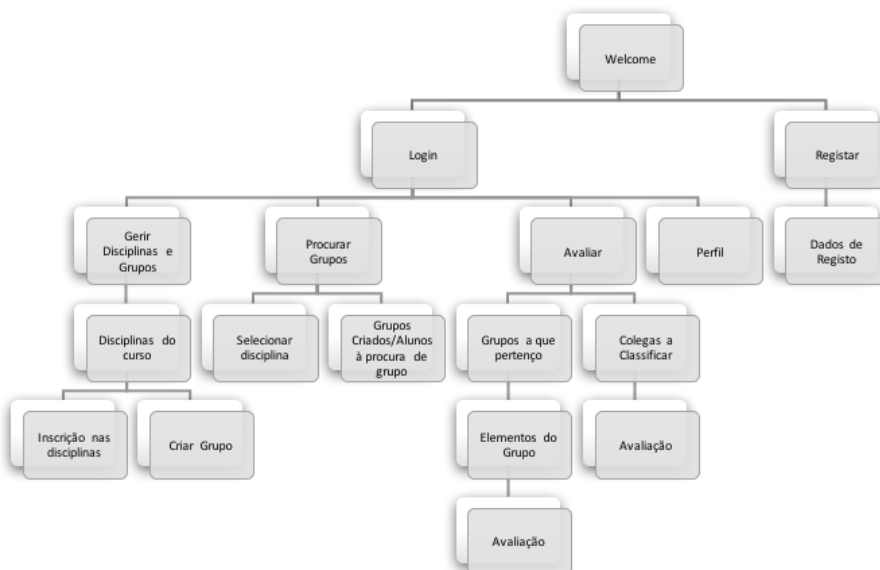


Figura 4 Mapa de Navegação da Aplicação Rate Me

## 6 Discussão

A solução proposta implementa todas as funcionalidades básicas que constituíam o “core” da aplicação ou seja avaliar os alunos, procurar grupos, visualizar alunos sem grupos e juntar-se a grupos. As alterações realizadas a nível da validação de notas apesar de não ter sido implementada totalmente, são um fator importante a destacar pois ao invés de se depender de um professor para validar as notas, este compromisso é feito inteiramente pelos alunos ao definirem a data de fim do grupo. Quando o tempo de duração do grupo que normalmente corresponde a duração do projeto de determinada cadeira chega ao fim, os alunos ficam então elegíveis para procederem às respetivas avaliações. Ao impor este mecanismo de avaliação a dependência da adesão dos professores a aplicação para a utilização da mesma por parte dos alunos torna-se nula.

Tentamos utilizar todas as tecnologias no desenvolvimento da aplicação, ou seja, fazer uso de tudo o que foi falado no decorrer das aulas teóricas, esse objetivo foi conseguido pois foi possível implementar funcionalidades que utilizassem *pedidos* Ajax no caso de procurar utilizador, implementado assim uma pesquisa *live search*<sup>18</sup> em que aparecem as sugestões do nome de utilizador. Quanto a utilização de *javascript* e *jquery* o mesmo foi utilizado para fazer o nosso background dinâmico, tentou-se ainda implementar validações *cliente side* utilizado uma biblioteca externa chamada *Laravel jsValidation*<sup>19</sup> que reutiliza as regras da validação *server side* que foram implementadas usando a validação do *Laravel*, reduzindo assim a repetição de código, porém devido a um erro relacionado com o *import do jquery (evaluation is not a function)*, esta funcionalidade não ficou a funcionar, embora todas as dependências estejam implementadas.

Num nível mais técnico a implementação do website tentou seguir as funções e as funcionalidades do *Laravel*, como o *Eloquent ORM*<sup>20</sup>, o que constituiu uma dificuldade inicial, pois a experiência do grupo a utilizar Modelos e *Active Records* para a comunicação com a base de dados era inexistente, o que conduziu a que se tenham despendido algumas dezenas de horas em investigação e aprendizagem deste tipo de funcionalidades do *Laravel*.

---

<sup>18</sup> «Ajax Live Search - PHP Tutorial».

<sup>19</sup> «proengsoft/laravel-jsvalidation: Laravel 5 Javascript Validation».

<sup>20</sup> «Eloquent ORM - Laravel - The PHP Framework For Web Artisans».

## 6.2 Vantagens

As vantagens da nossa aplicação são, permitir uma melhor gestão dos grupos de trabalho em contexto académico/escolar, e permitir aos alunos ter um rating que os motive a trabalhar e a participar ativamente nas atividades de grupo.

O site encontra-se implementado de forma a que seja relativamente fácil realizar as operações pretendidas sem muitas opções adicionais que desviem a atenção do utilizador do que é essencial, daí que muitas das seleções que poderiam ter sido feitas por meio de formulários, foram substituídas por sequência de ações utilizando botões, isto permite que a aplicação no futuro possa ser direcionada para um público mais abrangente (como por exemplo alunos do 2º e 3º ciclo) devido à maior facilidade de utilização.

## 6.3 Desvantagens

As desvantagens desta forma de aplicação algo simplista onde os “estados” de utilização se encontram bem determinados, é que condicionada um pouco a liberdade do utilizador, e pode conduzir a uma aplicação pouco elástica, assim como a pouca utilização de *client-side-scripting* que melhoraram a experiência de utilização.

## 7 Trabalho futuro

Numa possível continuidade deste projeto, era necessário implementar funções de gestão de projetos tais como, agendamento de reuniões de grupo entre outras funcionalidades para que a utilização da aplicação não seja “sazonal”, ou seja é utilizada no início e final de cada semestre ou ano letivo para os alunos encontrarem grupos e se avaliarem e depois disso, pouca utilidade teria. Uma *feature* interessante que o site poderia também vir a incluir era a possibilidade de os professores de uma dada disciplina realizarem questionários para *peer review* por parte dos alunos de forma customizável e mais rápida, do que os métodos normalmente utilizados para obter este tipo de informação, como preenchimento de tabelas, *google surveys*, desta forma o professor poderia ter rapidamente os resultados das *peer review*, ficando a cargo da aplicação toda a parte de processamento de dados.

## 8 Conclusão

Este projeto exigiu uma carga de trabalho elevado, mas possibilitou que se desenvolvessem competências na área do desenvolvimento web e serviu como ponte de contacto com as diversas tecnologias envolvidas nesse mesmo desenvolvimento. A *framework* escolhida, revelou-se uma escolha sagaz, pois pelo feedback<sup>21</sup> obtido da comunidade de desenvolvedora web, esta encontra-se em franca expansão. Relativamente as maiores dificuldades com este projeto a nível de conceção, devem-se sobretudo ao facto de como a ideia é original e diferenciada não existe propriamente uma linha guia de como resolver determinados problemas de ordem conceptual do funcionamento da aplicação, o que levou a que se despende-se mais algum tempo na fase inicial do projeto no levamento de requisitos e de funcionamento, ao mesmo tempo esta acaba por ser uma das vantagens desta aplicação e do tema escolhido pois permite que se tenha uma maior liberdade criativa do que costuma ser habitual em projetos de âmbito académico.

Em relação aos objetivos, as *core feature* da aplicação encontram-se implementadas cumprindo assim os objetivos primordiais da aplicação, como referido na secção anterior (7) existe ainda uma quantidade substancial de trabalho a realizar especialmente se se pretender lançar a aplicação online. Em suma o projeto foi concluído de forma muito satisfatório e contribui-o para o desenvolvimento de competências nas áreas das aplicações centradas em redes.

---

<sup>21</sup> you?, «The Business Case for Laravel».

## 8 Referências

- «Ajax Live Search - PHP Tutorial». *finalwebsites.com*. Acedido 2 de Fevereiro de 2017. <http://www.finalwebsites.com/jquery-ajax-live-search/>.
- «Authentication - Laravel - The PHP Framework For Web Artisans». Acedido 2 de Fevereiro de 2017. <https://laravel.com/docs/5.4/authentication>.
- «Blade Templates - Laravel - The PHP Framework For Web Artisans». Acedido 2 de Fevereiro de 2017. <https://laravel.com/docs/5.4/blade>.
- «CSS Tutorial». Acedido 1 de Fevereiro de 2017. <http://www.w3schools.com/css/default.asp>.
- «Eloquent: Getting Started - Laravel - The PHP Framework For Web Artisans». Acedido 2 de Fevereiro de 2017. <https://laravel.com/docs/5.4/eloquent>.
- «Eloquent ORM - Laravel - The PHP Framework For Web Artisans». Acedido 2 de Fevereiro de 2017. <https://laravel.com/docs/4.2/eloquent>.
- «Forms & HTML - Laravel - The PHP Framework For Web Artisans». Acedido 1 de Fevereiro de 2017. <https://laravel.com/docs/4.2/html>.
- «Intervention Image - Introduction». Acedido 2 de Fevereiro de 2017. <http://image.intervention.io/>.
- «JavaScript Tutorial». Acedido 1 de Fevereiro de 2017. <http://www.w3schools.com/js/>.
- «jQuery.ajax() | jQuery API Documentation». Acedido 1 de Fevereiro de 2017. <http://api.jquery.com/jquery.ajax/>.
- «Laravel - The PHP Framework For Web Artisans». Acedido 1 de Fevereiro de 2017. <https://laravel.com/>.
- Makabee, Hayim. «Separation of Concerns». *Effective Software Design*, 5 de Fevereiro de 2012. <https://effectivesoftwaredesign.com/2012/02/05/separation-of-concerns/>.
- «MySQL Tutorial 1: Overview, Tables, Queries». Acedido 1 de Fevereiro de 2017. <http://arachnoid.com/MySQL/>.
- «proengsoft/laravel-jsvalidation: Laravel 5 Javascript Validation». Acedido 2 de Fevereiro de 2017. <https://github.com/proengsoft/laravel-jsvalidation>.
- «reCAPTCHA: Easy on Humans, Hard on Bots». Acedido 2 de Fevereiro de 2017. <https://www.google.com/recaptcha/intro/>.
- «Understanding MVC: Model-View-Controller». Acedido 1 de Fevereiro de 2017. <https://www.lynda.com/CakePHP-tutorials/Understanding-MVC-Model-View-Controller/315196/378333-4.html>.
- «web development - What are the differences between server-side and client-side programming? - Software Engineering Stack Exchange». Acedido 2 de Fevereiro de 2017. <http://softwareengineering.stackexchange.com/questions/171203/what-are-the-differences-between-server-side-and-client-side-programming>.
- «Working with Databases: Active Record | The Definitive Guide to Yii | Yii PHP Framework». Acedido 2 de Fevereiro de 2017. <http://www.yiiframework.com/doc/guide/1.1/en/database.ar>.
- you?, Laravel Application Design and Development: SiteRocket Labs, Boston, MA How can we help. «The Business Case for Laravel». Acedido 2 de Fevereiro de 2017. <https://www.siterocket.com/blog/2015/08/11/business-case-for-larav>

## 9 Anexos

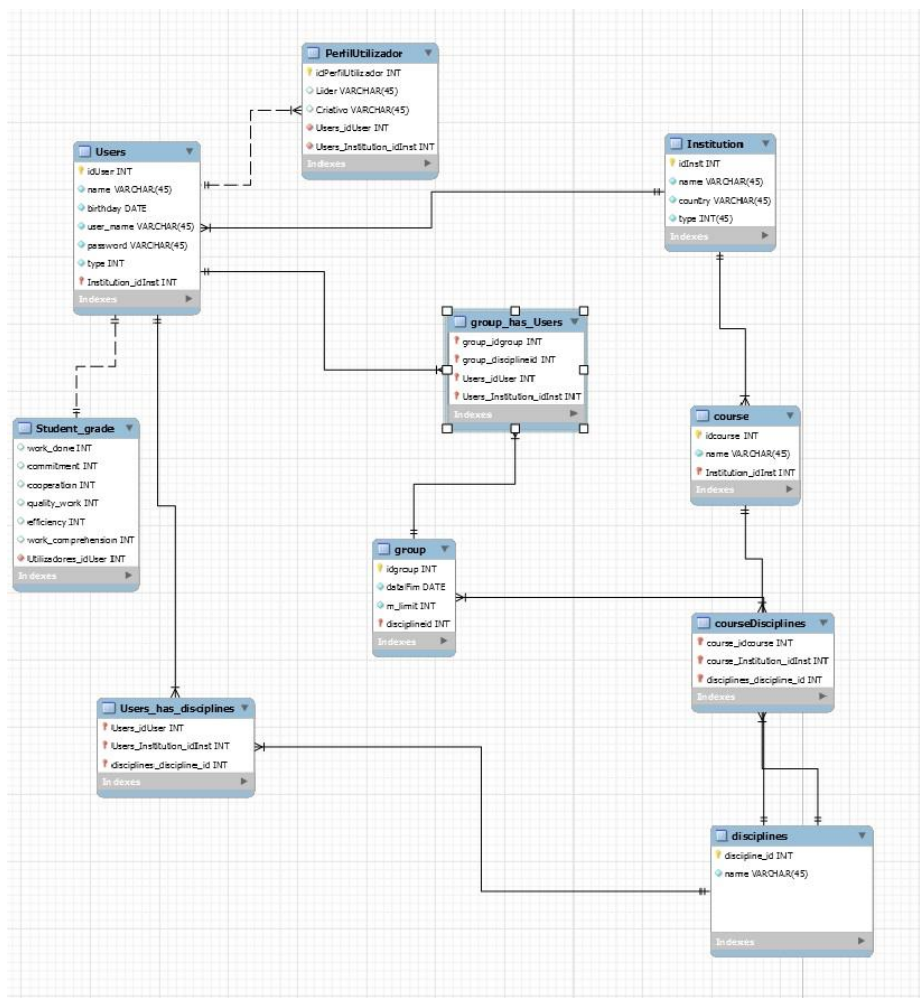


Figura 5 Esquema Relacional aproximado da aplicação

