



**UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE EDUCAÇÃO BÁSICA E PROFISSIONAL
COLÉGIO TÉCNICO**



RELATÓRIO FINAL DE ATIVIDADES DE ESTÁGIO CURRICULAR OBRIGATÓRIO

Alan Borges Martins Bispo

MG-19.413.189

Curso Técnico em Informática

Ano de conclusão: 2016

Rua Icarai, 680, Caiçaras

(31) 992957388

Instituição: Colégio Técnico da UFMG

Setor de Informática

Data de início: 16/05/2016

Total de horas cumpridas no período: 264 horas

Belo Horizonte/MG - 2016

SUMÁRIO

1. Introdução	3
2. Setor de atuação e descrição da instituição	3
3. Organização das atividades e rotina de trabalho	3
4. Objetivo do estágio	4
5. Modelagem inicial do sistema	4
6. Equipamentos e softwares utilizados	5
a. MySQL Server	5
b. Composer	5
c. PHP	5
d. Slim PHP	6
e. Doctrine	7
f. Postman	7
g. Git	8
h. Atom editor	9
7. Arquitetura e fluxo do sistema	9
8. Avaliação do curso	10
9. Conclusão	11
10. Anexos	11
a. User stories	11

1. INTRODUÇÃO

Este relatório pretende apresentar e descrever as atividades realizadas por mim, Alan Borges Martins Bispo, junto ao setor de informática e a equipe composta para a realização do projeto. Os integrantes da equipe para a realização do projeto eram, além de mim, Ariel Augusto, Augusto Lopes Ferreira França e Gabriel Carneiro de Castro. Os professores João Eduardo Montandon, Virgínia Fernandes Mota e Leandro Maia foram os orientadores nossos.

No decorrer deste documento será descrito os seis meses de trabalhos no projeto nomeado como Sistema de Gerenciamento Escolar, um sistema web de notas e organização escolar. A equipe supradita trabalhou no projeto na forma que será descrita na secção três deste documento. Neste documento estará detalhado as características das tecnologias utilizadas e listadas as etapas do processo de produção do sistema web requerido.

2. SETOR DE ATUAÇÃO E DESCRIÇÃO DA INSTITUIÇÃO

O estágio foi realizado no Colégio Técnico da UFMG - COLTEC (Avenida Presidente Antônio Carlos, 6627, Pampulha). A instituição faz parte da Unidade Especial denominada Escola de Educação Básica e Profissional da UFMG e oferece cursos técnicos integrados ao Ensino Médio concomitante.

O setor relacionado ao estágio é o Setor de Informática, que dentre os docentes estão nossos orientadores. O Setor de Informática preza por utilizar o próprio material nas aulas de seus professores e no trabalho cotidiano. Por tal motivo o estágio que realizamos tinha a mesma intenção, que era produzir um sistema estudantil para a própria escola poder dele se utilizar.

3. ORGANIZAÇÃO DAS ATIVIDADES E ROTINA DE TRABALHO

A equipe formada pelo Setor de Informática se consistia em mim, Ariel Augusto dos Santos, Augusto Lopes Ferreira França e Gabriel Carneiro de Castro. A equipe foi internamente dividida para maior produtividade. Eu e meu colega, Ariel Augusto, ficamos com a parte de desenvolvimento backend, enquanto Augusto Lopes e Gabriel Carneiro ficaram com a parte de desenvolvimento frontend. Assim, tratarei neste documento sobre a produção de tudo que diz respeito ao backend desenvolvido e à integração deste com o frontend.

Foi-nos proposto a utilização de um sistema web para organização de projetos e gestão de grupos. O *RedMine* (<http://www.hostedredmine.com/>) foi o site que nós usamos para organizar as tarefas a serem feitas, registrar as horas gastas em cada tarefa, dizer se estávamos conseguindo atender as necessidades e lançar *flags* de bugs do sistema que teríamos de mais tarde discutir nas reuniões semanais para, em consenso, resolvermos. Logo a frente segue um exemplo de tela do RedMine com as tarefas.

<input type="checkbox"/>	#	Tipo	Situação	Prioridade	Título	Atribuído para	Alterado em	Categoria
<input type="checkbox"/>	572353	Task	Assigned	Normal	Fazer DAOs para as classes PHP	Ariel Santos	2016-07-08 05:26	
<input type="checkbox"/>	572352	Task	In Progress	Normal	Fazer Controllers para as classes PHP	Alan Borges Martins Bispo	2016-08-30 07:15	
<input type="checkbox"/>	571573	Task	New	Normal	Ajustar o entity manager p/ singleton	Ariel Santos	2016-07-05 03:48	
<input type="checkbox"/>	569231	Bug	Resolved	High	Não há nenhuma ligação entre Atividade e Aluno	Leandro Maia	2016-06-28 03:55	
<input type="checkbox"/>	569216	Task	Resolved	High	Classes PHP para o Doctrine	Alan Borges Martins Bispo	2016-07-07 10:43	
<input type="checkbox"/>	566357	Task	Resolved	Normal	Mapeamento XML para o Doctrine	Ariel Santos	2016-07-05 03:48	
<input type="checkbox"/>	563079	Task	In Progress	Normal	Implementar Criação de nova atividade	Ariel Santos	2016-08-30 07:19	
<input type="checkbox"/>	556185	Task	Resolved	Normal	Construção do DER	Ariel Santos	2016-06-07 10:51	
<input type="checkbox"/>	556172	Task	New	Low	Implementar um sistema de build	Ariel Santos	2016-05-20 04:28	

(1-9/9)

Exportar para [Atom](#) | [CSV](#) | [PDF](#)

Dentro do período de trabalho, as 12 horas semanais, nós desenvolvíamos e testávamos os módulos produzidos pouco a pouco. Estudos necessários para o conhecimento exigido nos trabalhos envolvidos na produção do código para o backend eram feitos independentemente por cada aluno. Reuniões semanais eram realizadas para que os orientadores analisassem o trabalho feito e pudessem avaliar se estava ou não satisfatório. Um relatório de cada reunião era feito, constando nele aquilo que foi feito e está fechado e aquilo que haveria de ser completado para a próxima semana.

4. OBJETIVO DO ESTÁGIO

O objetivo do projeto era produzir um sistema estudantil que englobasse a maior parte das áreas da escola. Assim, haveria um sistema de notas robusto, um sistema de cadastro de novos alunos, professores e administradores. Além disso, turmas, aulas, disciplinas, ementas de curso fariam parte do que se poderiam criar, pesquisar, atualizar e excluir neste novo sistema. O sistema serviria para ocupar o lugar do atual sistema de notas do COLTEC, hospedado em um ambiente web onde alunos, professores e administradores poderiam ter acesso através de suas credenciais (login e senha) às informações supraditas.

5. MODELAGEM INICIAL DO SISTEMA

A modelagem inicial do sistema começou com o levantamento de necessidades. Discutimos com os orientadores qual eram as necessidades que o sistema deveria atender e, concluindo isto, o orientador modelou as entidades atuantes do sistema e o fluxo de dados entre elas. Além disso fomos orientados a fazer as *user stories* do sistema (Ver Secção 9 – subsecção a). Essas são as formas pelas quais um usuário comum gostaria de passar pelo sistema. Em geral, um software é modelado inicialmente assim: decidindo as entidades que atuam nele e definindo suas *user stories*. Ademais, nós, tendo decidido então como o usuário veria o sistema e qual era a relação entre as entidades deste, teríamos agora de escolher as tecnologias a serem usadas na produção efetiva de código.

6. EQUIPAMENTOS E SOFTWARES UTILIZADOS

Os equipamentos utilizados por nós foram providos pelo Setor de Informática e ficaram no Laboratório de Informática, Telecomunicações e Eletrônica (LITE). Eles foram:

- Computadores com sistema operacional Linux Ubuntu instalado;
- Periféricos necessários para o uso dos computadores;
- Cartazes e post-its;

Além disso, para a produção de código e instalação dos serviços necessários ao sistema, usamos alguns softwares. Abaixo estão descritos apenas os softwares usados na produção da parte backend do sistema:

- MySQL Server 5.6;
- Composer 1.0;
- PHP 5.0/7.0 & PHP Server;
- Slim PHP 3.0;
- Doctrine ORM 2.5;
- Git;
- Atom editor;

Seguem-se subsecções explicitando o uso e a necessidade de cada item da lista acima.

a. MySQL Server

Utilizamos o software MySQL (<https://www.mysql.com/>) para estruturar o banco de dados que serviria como camada de persistência para os dados inseridos no sistema, vindos da web. Dessa forma, estruturamos o banco de dados primeiro com um Diagrama Entidade Relacionamento para, depois de avaliado pelos orientadores, o diagrama ser transportado para a linguagem SQL que faria efetivamente o banco de dados no MySQL Server. Neste banco de dados estavam todas as tabelas que representavam as classes.

b. Composer

O Composer (<https://getcomposer.org/>) é um gerenciador de pacotes que faz para nós de forma automática o download das dependências de um projeto. Isto é, as dependências são as bibliotecas que usamos para conseguir executar certas tarefas necessárias ao sistema. Mais à frente, nas subsecções de Slim PHP e Doctrine, explicaremos mais como são importantes estas dependências para o projeto.

c. PHP 5.0/7.0 & PHP Server

PHP (<https://secure.php.net/>) é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na web. Assim, optamos por usar esta linguagem, que é um software de livre uso, justamente para podermos separar bem e distinguir a diferença entre o que é frontend e o que é backend. Dessa forma a manutenção se torna mais simples. O outro motivo foi o ótimo recurso que poderíamos usar, seu framework Slim. Falaremos a seguir.

Em alguns pontos do projeto utilizamos o PHP 5.0 e em outros utilizamos sua versão mais atual, a 7.0. Isso aconteceu por que em alguns casos era melhor testar as funcionalidades na versão mais estável da linguagem. Para fazer os testes dos códigos por nós produzidos, nós nos utilizamos do PHP Server comum, que cria um servidor no computador e que pode então receber diversas requisições HTTP.

d. Slim PHP

O Slim PHP (<https://www.slimframework.com/>) é um framework para a linguagem PHP que torna mais simples a manipulação HTTP e boa estruturação do código, isto é, obedecendo os padrões de projeto. O Slim possui várias ferramentas, mas a grande qualidade dele é seu router simples: um router literalmente roteia as requisições absorvidas pelo servidor PHP para as rotas definidas. Nas rotas criamos, portanto, os controladores para entender a requisição e fazer o que se pede no caso em questão. Abaixo segue um exemplo do roteamento do Slim na prática do projeto:

```
73
74 $app->group(pessoa, function () {
75     $this->post('/registro', '\Controller\PessoaController:create');
76     $this->delete('/{id}', '\Controller\PessoaController:delete');
77     $this->get('/{id}', '\Controller\PessoaController:get');
78     $this->patch('/{id}/{params:.*)', '\Controller\PessoaController:update');
79 });
80
81 $app->group(pessoaF, function () {
82     $this->post('/registro', '\Controller\PessoaFController:create');
83     $this->delete('/{id}', '\Controller\PessoaFController:delete');
84     $this->get('/{argumento}', '\Controller\PessoaFController:getBy');
85     $this->patch('/{id}/{params:.*)', '\Controller\PessoaFController:update');
86 });
87
88 $app->group(pessoaJ, function () {
89     $this->post('/registro', '\Controller\PessoaJController:create');
90     $this->delete('/{id}', '\Controller\PessoaJController:delete');
91     $this->get('/{id}', '\Controller\PessoaJController:get');
92     $this->patch('/{id}/{params:.*)', '\Controller\PessoaJController:update');
93 });
94
95 $app->group(professor, function () {
96     $this->post('/registro', '\Controller\ProfessorController:create');
97     $this->delete('/{id}', '\Controller\ProfessorController:delete');
98     $this->get('/{id}', '\Controller\ProfessorController:get');
99
```

Para cada agrupamento temos a função CRUD normal (Create, Retrieve, Update, Delete). Assim, podemos definir as opções do nome da rota, a qual controlador esta fará a chamada e qual o método HTTP que se suporta naquela rota. Além disso podemos também fazer a definição de cabeçalho, que diz as diretrizes de como aquele servidor PHP irá lidar com as requisições.

e. Doctrine

O Doctrine (<http://doctrine-orm.readthedocs.io/en/latest/>) é o que chamamos de ORM (Object Relational Mapper). Isto significa que ele faz o papel de mapeamento das classes entre o banco de dados e o Slim PHP. Sem ele, o que se produz no servidor, não se pode persistir no banco de dados MySQL já discutido.

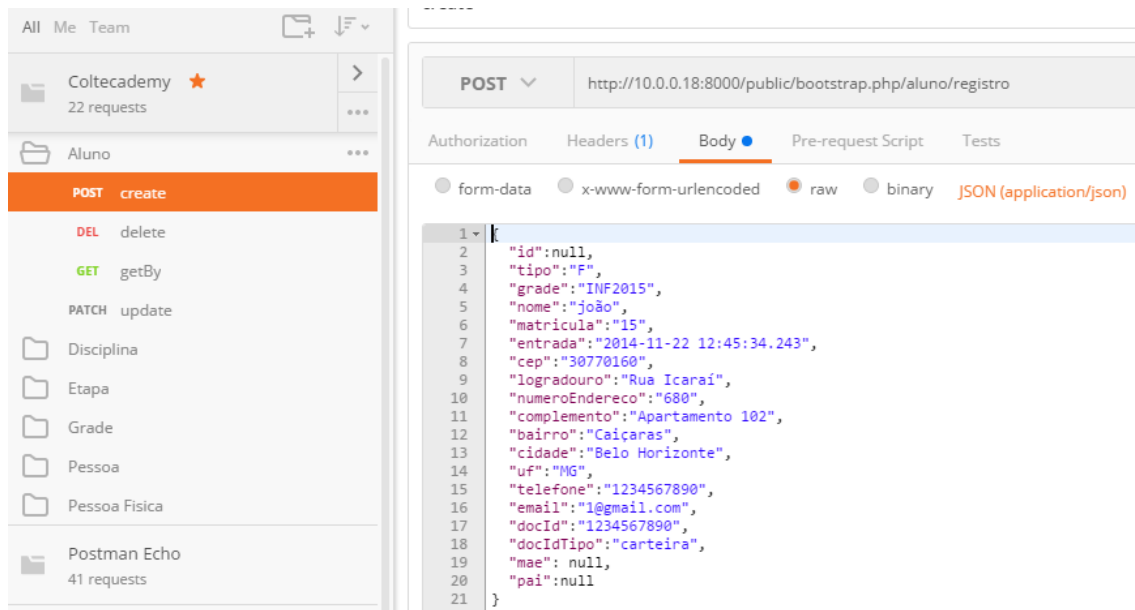
O mapeamento dele é feito em linguagem XML (<https://www.w3.org/XML/>), indicando os campos das tabelas em SQL e a relação destas com as outras tabelas. Assim, seguimos a sua documentação e produzimos o mapeamento necessário para todas as entidades do sistema. Veja abaixo um exemplo de mapeamento para a entidade Aluno:

```
Entidade.Aluno.dcm.xml
1 <doctrine-mapping xmlns="http://doctrine-project.org/schemas/orm/doctrine-mapping"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://doctrine-project.org/schemas/orm/doctrine-
4
5   <entity name="Entidade\Aluno" table="tbAluno">
6     <indexes>
7       <index name="pessoaF_idx" columns="pessoaF"/>
8       <index name="grade_idx" columns="grade"/>
9     </indexes>
10
11     <id name="matricula" type="string"/>
12
13     <field name="entrada" type="date" />
14     <field name="saida" type="date" nullable="true"/>
15     <field name="pessoaF" type="integer" />
16     <field name="grade" type="string" />
17
18     <one-to-one field="pessoaF" target-entity="PessoaFisica">
19       <cascade><cascade-remove/></cascade>
20       <join-column name="pessoaF" referenced-column-name="id" on-delete="CASCADE"/>
21     </one-to-one>
22
23     <one-to-one field="grade" target-entity="Grade">
24       <cascade><cascade-remove/></cascade>
25       <join-column name="grade" referenced-column-name="grade" on-delete="CASCADE"/>
26     </one-to-one>
27
```

Além disso, a outra parte que compunha as tarefas do Doctrine no sistema é o Entity Manager. O EM, como é conhecido, é a entidade que faz o gerenciamento das outras entidades no sistema, isto é, é ele que pode persistir, buscar, atualizar ou apagar informações do bando de dados. O EM é responsável por todas as transações entre o Slim e o MySQL. Portanto, o EM se vale dos mapeamentos para acessar o database requerido e nele fazer as modificações específicas de cada caso.

f. Postman

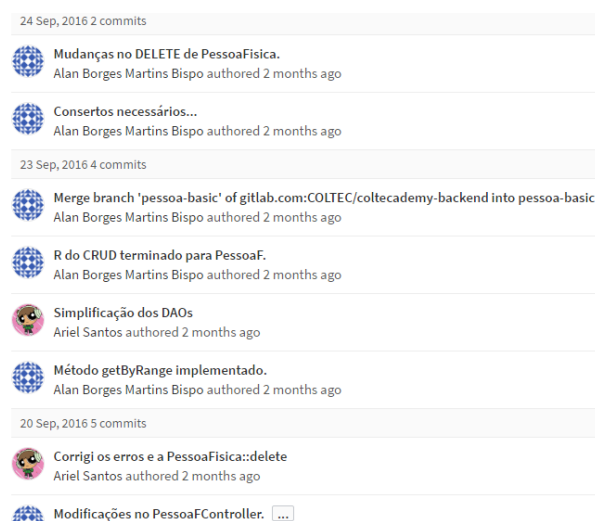
O Postman é um software usado para emular requisições web. Assim, sem precisar que a requisição venha do frontend, é possível testar a qualidade do código produzido, e se está atendendo a necessidade em questão. As requisições do Postman são feitas usando a linguagem de marcação JSON (<http://www.json.org/>), por onde são passadas as informações para o backend. Veja abaixo o exemplo de requisição no Postman:



No campo da direita temos o JSON para a criação do Aluno, enquanto à esquerda temos a biblioteca de testes desenvolvida exclusivamente por mim. Nesta se encontram todos os testes para as classes do sistemas, testando a funcionalidade do seu CRUD.

g. Git

O Git (<https://git-scm.com/>) é um de servidor de arquivos que nos possibilita guardar nossos projetos online. Ele é uma ferramenta muito interessante para se usar em grupo de trabalho, como é nosso caso, pois desta forma podemos gerenciar o que cada um fez. Ao término do dia de trabalho nós sempre fazíamos a operação de *push*, que é basicamente mandar para a internet os arquivos que eu criei, modifiquei ou de lá eu excluí. Fica gravado num esquema de árvore todas as mudanças feitas por cada usuário. Assim também, os orientadores puderam averiguar as coisas que produzíamos, os nossos *commits*, que são as vezes que fazemos o upload dos arquivos para internet. Veja abaixo o histórico de *commits* do mês de setembro:



O exemplo acima é do servidor web GitLab (<https://gitlab.com/>). Para conseguirmos mandar commits para este servidor temos que ter uma conta no site. Assim, como visto na imagem acima, o histórico tem o commit feito e quem o fez.

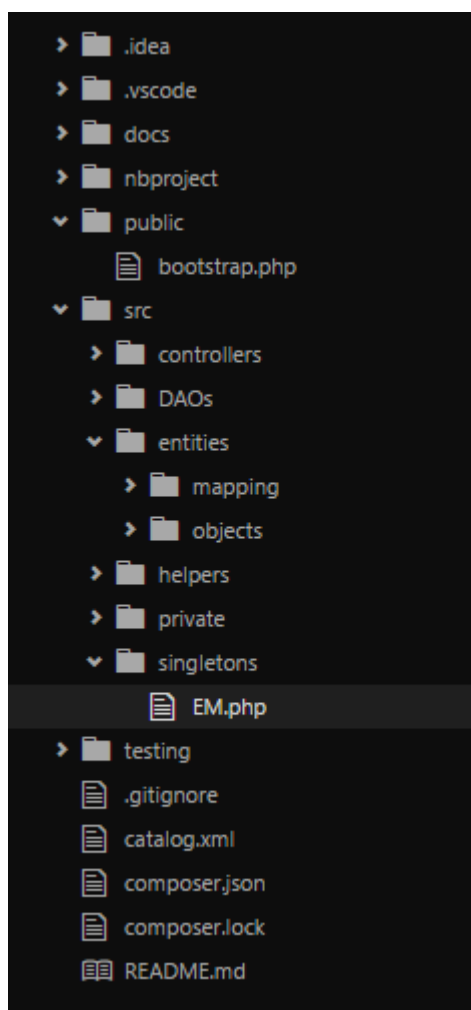
h. Atom editor

Para a produção de código nós usamos o Atom editor (<https://atom.io/>). Ele é um editor comum de texto, muito bom pois possui integração com o Git, mostrando os arquivos modificados e criados.

7. ARQUITETURA E FLUXO DO SISTEMA

A arquitetura de projeto que usamos foi baseada em alguns padrões de projeto renomados. Tentamos isolar todas as classes diferentes em pacotes diferentes, deixando simples a manutenção e o futuro entendimento do projeto. Veja abaixo o esquema de pastas:

Figura 7.1 - Pacotes do sistema



Perceba na figura 7.1 que o que está na pasta *public* é tudo aquilo que é acessado externamente através de requisições. Os outros pacotes estarão fechados para o que é externo, podendo serem acessados só pelo *bootstrap.php*. Na pasta *src* temos o núcleo do sistema, onde estão as classes que se relacionam, tanto a nível de PHP, quanto com o EM, ou seja, o banco de dados.

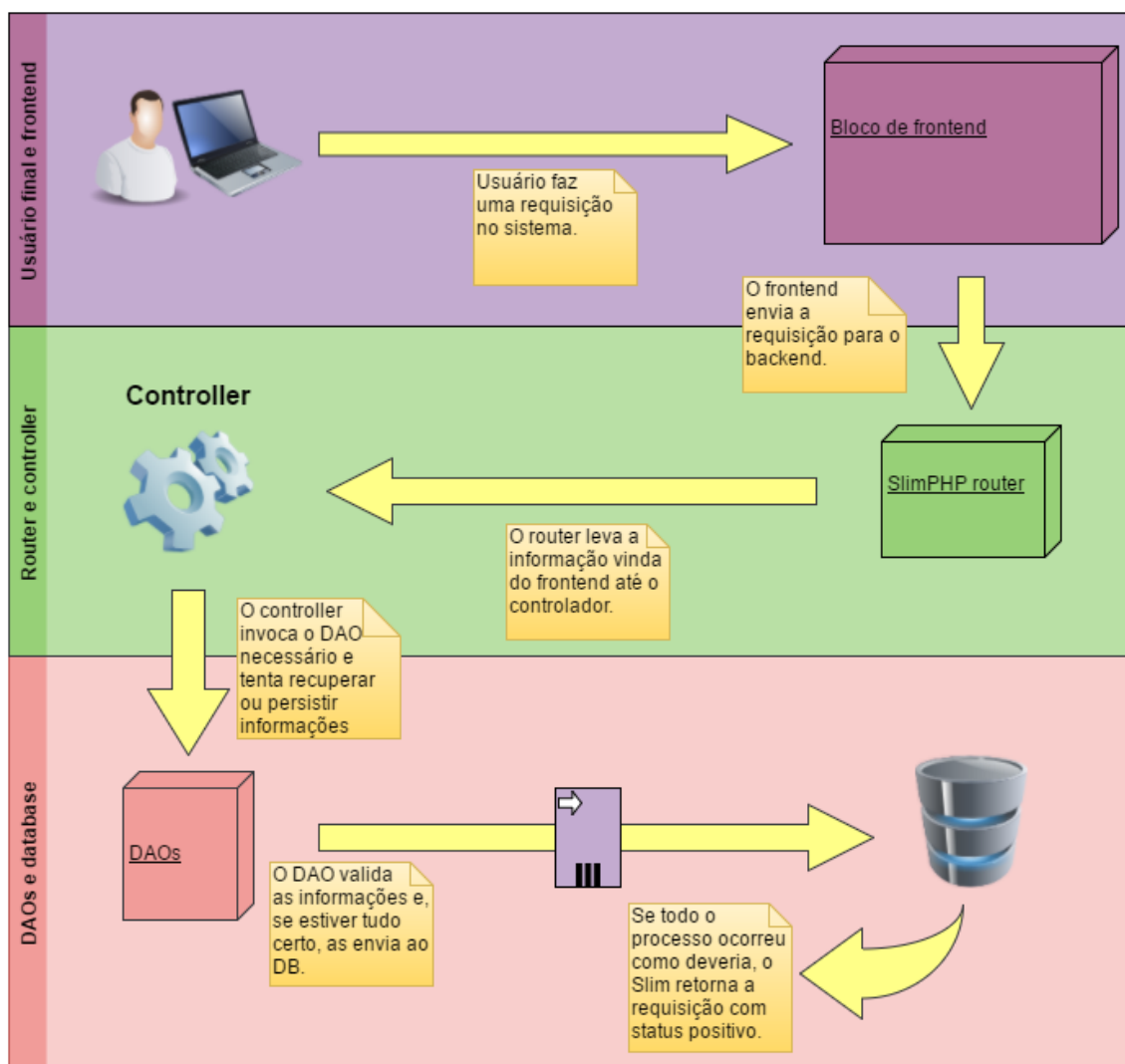
A pasta *controllers* possui os controladores do CRUD de cada entidade. Isto é, ele relaciona as entidades que têm dependência e persiste as mudanças no banco de dados. Para isso, os controladores fazem acesso ao pacote *DAOs*, onde estão os Data Access Object para cada objeto. São os *DAOs* que conversam com o EM para a persistência.

No pacote *entities* estão todas as classes PHP das entidades e todos os seus respectivos mapeamentos XML para o Doctrine. Assim, também, a pasta *singletons* guarda o singleton do sistema, que é o próprio EM. Este é um padrão de projeto muito requerido.

Além disso temos as pastas auxiliares, a *docs*, que guarda os documentos referentes ao sistema, como o DER; e também os *helpers*, que são funções auxiliares. As outras pastas, como as ocultas ou *testing*, são geradas no projeto para auxílio do programador.

Ademais, para entendermos o fluxo de dados mais amplo possível, temos um diagrama para explicitar os movimentos que o sistema faz e quais componentes estão

presentes. Abaixo está o diagrama de fluxo de dados, que inicia desde o usuário final fazendo qualquer requisição, até o *database*, de menor abstração.



8. AVALIAÇÃO DO CURSO

O curso técnico de Informática ofertado pelo Setor de Informática da UFMG foi de extrema importância na formação da minha capacidade, não só técnica, mas também de conseguir trabalhar sem precisar necessariamente de um tutor. Os docentes do curso são muito bem informados e possuem muito conhecimento técnico na área. As aulas ministradas por eles são de uma qualidade altíssima em termos de conteúdo.

No entanto, muitas vezes, tive dúvidas de alguns termos durante meu período de estágio obrigatório. Acredito que estas dúvidas nasceram porque, ainda que os professores sejam muito bons, alguns deles não são muito didáticos, o que dificulta a aprendizagem em muitos casos. Além disso, o que desenvolvemos durante o estágio foi usando uma tecnologia que inicialmente não conhecíamos, o que tirou grande parte do nosso tempo e concentração para adquirir conhecimento nessas tecnologias.

O curso dá uma ótima noção de algoritmos e estrutura de dados, que se fez essencial para a aprendizagem de qualquer linguagem ou método que eu vim a aprender durante o

estágio. Simplesmente não seria possível ter aprendido tudo que aprendi se antes eu não tivesse tido uma base tão boa em algoritmos. Mas acredito que o Setor de Informática peca em focar muito no estudo de algoritmos durante o primeiro ano de curso técnico, porque assim nós sabemos muito pouco conteúdo, ainda que tenhamos um alto potencial em aprender as linguagens modernas.

9. CONCLUSÃO

Ao final dos seis meses de projeto que eu desenvolvi junto a minha equipe e ao Setor de Informática do Coltec vejo que esta experiência foi ótima para mim, me engrandeceu tanto no campo do trabalho em equipe quanto no campo técnico. Ainda que o projeto não tenha sido completado, tendo apenas a base sua feita, considero que a experiência foi ótima para mim e para o grupo.

Este trabalho está com suas bases completas e funcionais, e nós iniciamos com pouco conhecimento. Adquiri desde o início muito conhecimento no ramo da produção de sistemas e aplicativos web, assim como o funcionamento das linguagens PHP e SQL, e também a forma pela qual trabalha o framework Slim PHP. Estas tecnologias são essenciais para quem quer espaço no mercado de produção backend para sistemas web.

10. ANEXOS

a. User stories

A seguir temos as user stories feitas para a modelagem do sistema. Usando o RedMine nós fizemos as user stories.

<input type="checkbox"/>	561173	User Story	New	Normal	Usuário professor: editar atividades		2016-06-02 04:30	
<input type="checkbox"/>	561169	User Story	New	Normal	Usuário professor: visualização das atividades		2016-06-02 04:25	
<input type="checkbox"/>	561164	User Story	New	Normal	Usuário professor: submeter/editar notas de alunos para atividades		2016-06-02 04:25	
<input type="checkbox"/>	560359	User Story	New	Normal	Usuário professor: criação de atividades		2016-08-30 07:19	
<input type="checkbox"/>	560327	User Story	New	Normal	Usuário administrador: acesso a página principal		2016-05-31 01:15	
<input type="checkbox"/>	560294	User Story	New	Normal	Usuário professor: acesso a página principal		2016-05-31 03:04	
<input type="checkbox"/>	558650	User Story	Assigned	Normal	Usuário aluno: acesso a página principal		2016-05-31 02:02	
<input type="checkbox"/>	558643	User Story	Assigned	Normal	Usuário geral: login		2016-05-31 02:02	

Estagiário (Alan Borges Martins Bispo)

Orientador (Virgínia Fernandes Mota)

Coordenador (Leandro Maia)