Criação e Controle de Trabalhos Presenciais com o Framework Django

Autor: Pedro Langbecker Lima
Orientador: João Vicente Ferreira Lima

[1]Contato: plima@inf.ufsm.br, jvlima@inf.ufsm.br

1. Introdução

Quando se deseja criar um trabalho para ser realizado em aula, é geralmente difícil de controlar as submissões recebidas dos alunos presentes. Para evitar ter de usar uma plataforma unicamente para essa função, normalmente se adota um sistema simples por *email*. Infelizmente, além da visualização dos trabalhos se tornar mais penosa aos professores, a possibilidade do aluno enviar o trabalho não estando presente na aula existe. Visando solucionar este problema de uma forma prática e simples, foi modelado um sistema *web* de criação e controle de trabalhos.

O framework Django, escrito em Python, e voltado para o desenvolvimento rápido para web, forneceu as ferramentas necessárias para a construção dessa aplicação *web service*.

A partir de estudos do *framework* Django e de seguir sua documentação, foi possível implementar um sistema prático de submissões de trabalhos para os alunos de informática da Universidade Federal de Santa Maria.

2. Metodologia

2.1. Django

Foi utilizada a versão 1.10 do Django, e a versão 3.5 do Python. Com poucas modificações é possível alterar as versões utilizadas tanto do Python quanto do Django. Foi também usado uma versão *back-end* do servidor LDAP. Esse servido contém informações dos alunos e professores matriculados nos cursos de informática da Universidade Federal de Santa Maria.

2.2. Implementação

A implementação iniciou-se com a construção do modelo entidade-relacionamento do banco de dados. Além disso, para esta aplicação *web*, o banco de dados deve se conectar com o servidor LDAP, obtendo assim informações dos emails e senhas dos alunos e professores do curso. O Django possui *models* em sua estrutura para facilitar na modelagem e utilização do modelo de banco de dados. Os *forms* facilitam também ao obter as informações dadas pelo usuário na página, utilizando métodos *post* nos próprios códigos HMTL das páginas.

Sua modelagem considerou também as turmas, que devem ser previamente construídas pelo super-usuário, onde um professor é indicado como chave estrangeira da turma, a fim de possibilitar seu gerenciamento dos trabalhos.Como os alunos utilizariam o site somente em período de aula, as turmas são especificadas somente para o professor, excluindo a necessidade de informar os alunos para cada turma.

Para armazenar os arquivos enviados, tanto do professor, quanto dos alunos, os *forms* facilitaram bastante. A Figura 1 ilustra muito bem a facilidade que o Django proporciona para trabalhar com arquivos. A função salvandoInstancia é chamada no momento que o professor submete o trabalho. A variável *get('file')* pega os dados enviados para a página e o salva no banco de dados. Os outros dados como nome, descrição, ou nome do professor são retirados diretamente da página do usuário. Todos são então salvos como uma instância da *model* Trabalho, passando por parâmetro de seu construtor os dados coletados.

```
def salvandoInstancia(self, professor, nome):
        trabalho = Trabalho(nome=self.cleaned_data.get('nome'),
                descricao=self.cleaned_data.get('descricao'),
                professor=professor,
                nomeProfessor=nome,
                turma=self.cleaned_data.get('turma')
                )
        trabalho.save()
        trabalho.file=self.cleaned_data.get('file')
        trabalho.save()
        return trabalho.id
```

Figura 1. Exemplo de código

3. Cronograma

Para realizar todas tarefas descritas, foi seguido uma ordem de desenvolvimento.

A primeira parte implementada foi o banco de dados, contendo informações tanto dos trabalhos criados pelos professores, quanto das submissões dos trabalhos dos alunos.

Em seguida, a criação de um portal simplificado foi elaborado, em que tanto aluno como professor podem acessar, se estiverem cadastrados no banco de dados. Como o projeto ainda não fazia conexão com o LDAP, os alunos e professores eram ser cadastrados manualmente. Posteriormente, a confirmação do *login* utilizou os dados retirados do servidor LDAP.

Após o *login*, a criação dos trabalhos foi produzida. É dado ao professor a possibilidade de enviar seu trabalho com nome, arquivo e descrição, bem como escolher uma das turmas que leciona para enviar o trabalho. O envio de arquivos foi implementado posteriormente, após uma maior compreensão do sistema de *media* do Django. Usando *forms* do Django foi possível extrair as informações dos trabalhos criados, transportando-os para uma instância em um modelo de banco de dados.

Posteriormente, a *home* foi modelada, aceitando tanto aluno como professor, mas com diferenças em sua interface. Para os professores, é possível criar um trabalho, indo para a página mostrada na Figura 2. Também é possível alterar o estado do trabalho. Os estados implementados foram: *Não enviado*, *Em execução* e *Pronto*.

No estado *Não enviado*, é permitido ao professor modificar o trabalho, ou removê-lo. Na *execução*, os trabalhos não podem ser editados pelos professores, mas aparecem nas *homes* dos alunos. Lá eles podem acessar o trabalho e enviá-los aos professores. Já no

estado *pronto*, os trabalhos não podem mais ser submetidos pelos alunos, mas o professor tem acesso aos que foram enviados.

Junto com este sistema, foi implementado também uma chave, que é gerada aleatoriamente quando o trabalho é submetido. Essa chave deve ser escrita pelo aluno quando for tentar acessar o trabalho. Como a ideia do projeto consiste no aluno realizar a tarefa em aula, esse método impede que alunos que não estejam na aula submetam o trabalho se não estiverem presentes.

Figura 2. Criação de Trabalho

4. Resultados Obtidos

Após fazer o login, o usuário (professor ou aluno) pode visualizar os trabalhos, bem como interagir com eles. Aos professores, é permitido tanto a criação como a modificação dos trabalhos existentes. Já aos alunos, como a Figura 4 demonstra, é permitido somente interagir com os trabalhos que estiverem em andamento.

Figura 3. Home dos Professores

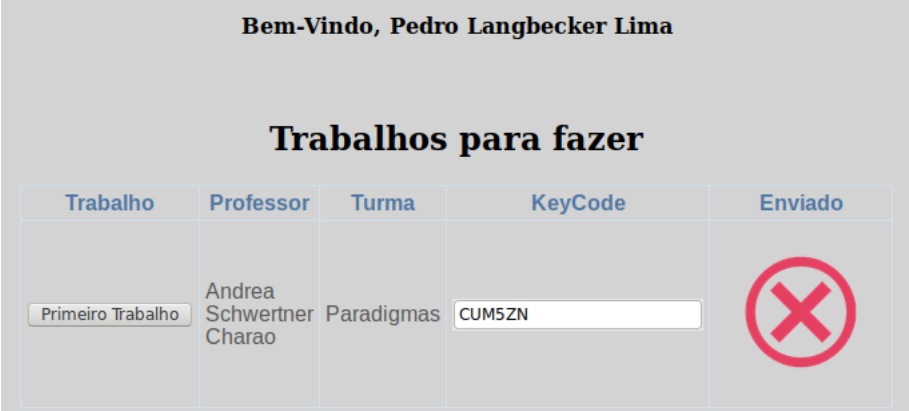

Figura 4. Home do Aluno - Trabalho Em Execução

Na área de criação de trabalhos, é disponível ao professor a escolha de envios tanto textual e por arquivo (atualmente, sem suporte para múltiplos arquivos para um único trabalho, mas arquivos comprimidos solucionam a limitação). Para finalizar, é necessário escolher uma turma na qual o professor leciona.

Então, quando finalizado, o professor pode acessar os trabalhos recebidos, identificando o usuário pelo nome, seguido pelo envio textual e um possível arquivo.

Na situação atual, é possível enviar somente um arquivo por vez. Convenientemente, pode-se submeter vários arquivos compactados em um único arquivo.

5. Conclusão

O sistema simples de submissão de trabalhos foi implementado, e já está pronto para estar ativo. Como ideias futuras do trabalho, visasse arrumar a organização dos arquivos submetidos pelos usuários, bem como melhorar a interface visual utilizando *css* para formatar as páginas web.

6. Referências

Django Software Foundation (2005-2017). The Web framework for perfectionists with deadline. Disponível em: https://www.djangoproject.com/.

Python Software Foundation. Welcome to Python. Disponível em: https://www.python.org/.