|  |  |  |
| --- | --- | --- |
|  | **Ministério da Educação**  **Universidade Tecnológica Federal do Paraná**  ***Diretoria De Graduação e Educação Profissional***  ***Coordenação do Curso Técnico Integrado em Informática*** |  |

FORMULÁRIO DE AVALIAÇÃO PERIÓDICA DO ESTÁGIO

Nome do Estagiário: João Martins de Queiroz Filho RA: 1552260

Nome da Empresa: Desenvolvimento de uma plataforma web para suporte a cursos de computação paralela

Nome do Supervisor (Empresa): Rogério Aparecido Gonçalves

Nome do Orientador (UTFPR): Marcos Silvano Almeida

Período avaliado: de 12/03/2018 a 26/03/2018

Atividades Desenvolvidas (inclua as dificuldades, os conhecimentos e experiências adquiridas no período):

Com o aprendizado através dos erros da ferramenta *Flask* foi utilizado um tempo maior para leitura da documentação da ferramenta *Django,* a principio seu funcionamento é parecido, contudo há criações automáticas como a união com o banco de dados (pode ser escolhido entre o *Sqlite* ou *MariaDB*), também há criação a partir de linha de comando que já cria o projeto com uma tela de administração.

Depois da instalação do ***Django*** na máquina foi necessário criar o projeto, para criar foi utilizado o seguinte comando em terminal:

|  |
| --- |
| *$ django-admin startproject webgpu* |

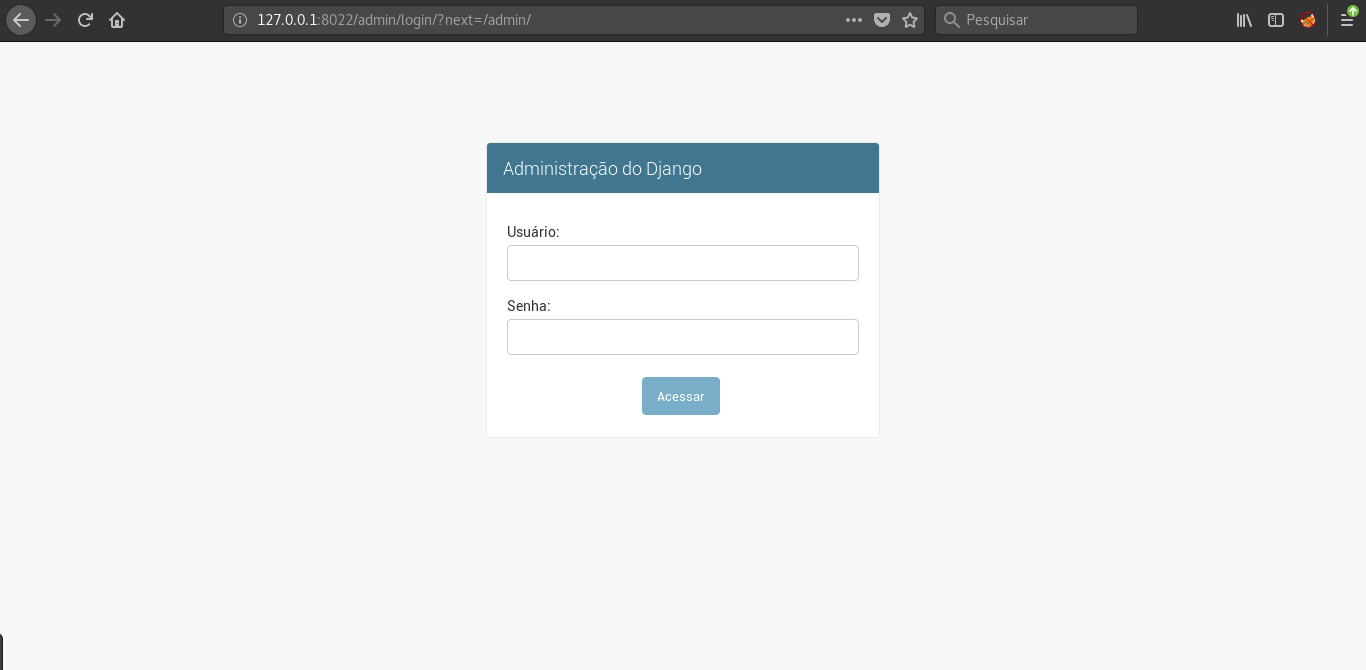
Desta forma, é criado um diretório contendo arquivos em *python* para controle do projeto, abaixo é mostrado como fica o diretório e uma breve descrição de sua função

|  |
| --- |
| **Webgpu/** → contêiner para o projeto  **manage.py**  → Um utilitário de linha de comando que permite a você interagir com o projeto Django de várias maneiras  **webgpu/** → é o pacote Python para o seu projeto  **\_\_init\_\_.py** → Um arquivo vazio que diz ao *Python* que este diretório deve ser considerado um pacote *Python*  **settings.py →** Configurações para este projeto Django  **urls.py** → As declarações de URLs para este projeto Django; um “índice” de seu site movido a Django  **wsgi.py** → Um ponto de integração para servidores WEB compatíveis com WSGI usado para servir o projeto |

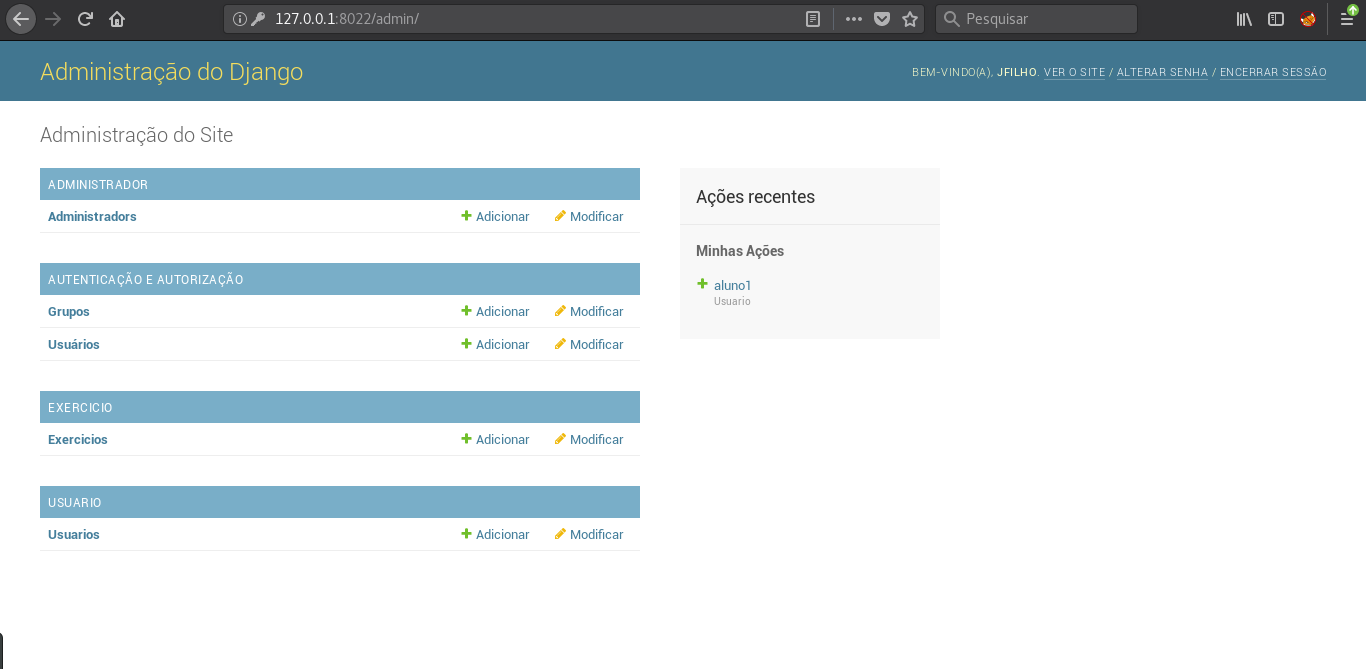
Com os diretórios criados foi utilizado o comando que cria um superusuário para o projeto, sendo que o *Django* automaticamente cria uma tela de admin, o comando utilizado é apresentado abaixo.

|  |
| --- |
| $ python3 manage.py createsuperuser |

Usando este comando é necessário colocar o nome completo do administrador, nome de usuário, senha e e-mail, após o cadastro desses dados foi feito o teste, que é feito colocando o */admin* na frente da *url* do projeto, pode ser visto na Figura 1.

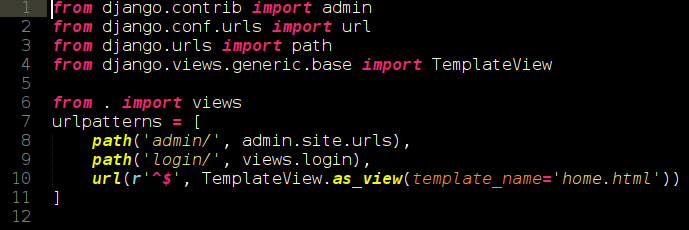
  
Figura 1: Tela de login administrador

Após a autenticação é redirecionado para a tela de administração do projeto, esta tela pode ser vista na Figura 2.

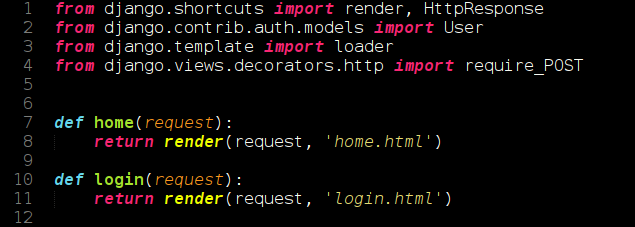
  
Figura 2: Tela inicial do administrador

A partir desta tela o administrador pode estar adicionando novos exercícios, bem como usuários e exercícios que deve ser realizado pelos alunos.

Assim que foi criado o superusuário, foi criado a tela principal *home.html,* esta tela foi reutilizada da versão *flask,* também foi reutilizada a tela de login (*login.html).* Uma semelhança entre o *flask* e *django* é que ambos trabalham com a ideia de rotas padrões, desta forma é usado rotas na *url* que o usuário deve seguir, contudo, uma diferença que essas rotas devem ser especificadas no arquivo *urls.py.* No código abaixo pode ser visualizado essas rotas.

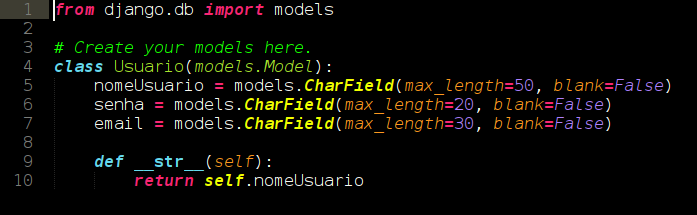
  
Código 1: Arquivo com as urls

E a partir que você especifica uma rota, deve-se criar uma função no arquivo *views.py* que especifica o funcionamento daquela rota, no Código 2 há a função *home* que especifica a pagina inicial do projeto e a função *login* que especifica a pagina de autenticação do usuário.

  
Código 2: Arquivo contendo as funções de views

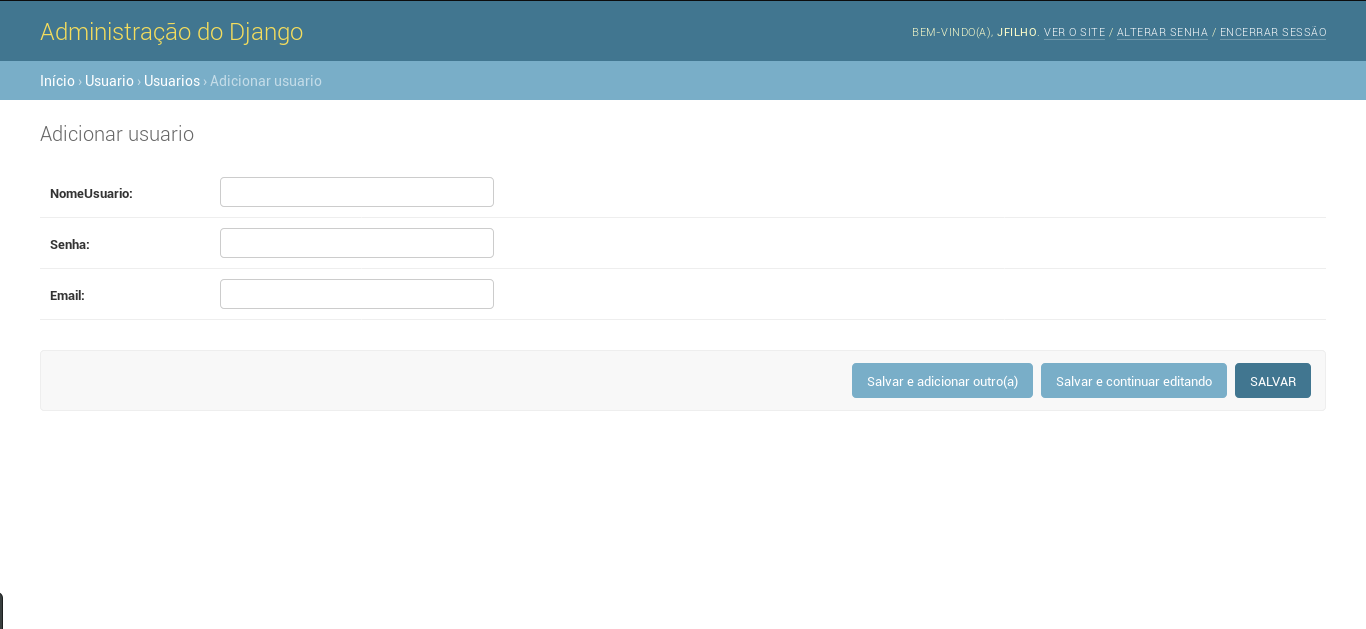
Outra diferença interessante entre o *flask* e *django* é por parte da modelagem do banco de dados, no *flask* deve-se realizar toda a criação e relacionamento no banco de dados escolhido (por exemplo neste projeto no *MariaDB*) e realizar as conexões, no *Django* é necessário somente especificar qual o nome da base de dados que ele se encarrega das conexões, os campos da tabela é especificado no arquivo *models.py,* no Código 3 há o modelo para os usuários.

Após ter criado todos os campos da tabela, é necessário somente realizar os seguintes comandos.

  
Código 3: Tabela usuários implementada no Django

|  |
| --- |
| $ python3 manage.py migrate  $ python3 manage.py makemigrations |

O primeiro comando especifica que está migrando todas as modificações que houve no arquivo *models* para o banco de dados que foi especificado, após ter realizado este comando é feito o segundo que aplica essas modificações no banco, na tela de administrador é apresentado esses campos da seguinte forma Figura 3.

  
Código 4: Adicionar usuário na tela de Adminstrador

Observações: Esse formulário deverá ser preenchido periodicamente (a cada 2 semanas de estágio) e assinado pelos responsáveis indicados abaixo. Arquive-os e, ao final do estágio, todos os formulários deverão ser anexados ao Relatório Final de estágio.

Assinaturas:

|  |  |  |
| --- | --- | --- |
| Aluno \_\_\_/\_\_\_/\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Supervisor \_\_\_/\_\_\_/\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Orientador \_\_\_/\_\_\_/\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Autor: Lucio Valentin