



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 14/05/2018 a 28/05/2018

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

Com a criação da tabela de dados contendo os dados dos alunos, foi criado em *HTML* uma tela de login para o aluno, na qual ele realizará a autenticação para resolver os exercícios propostos pela plataforma. Pode ser visualizado na Figura 1 a tela de *login*.

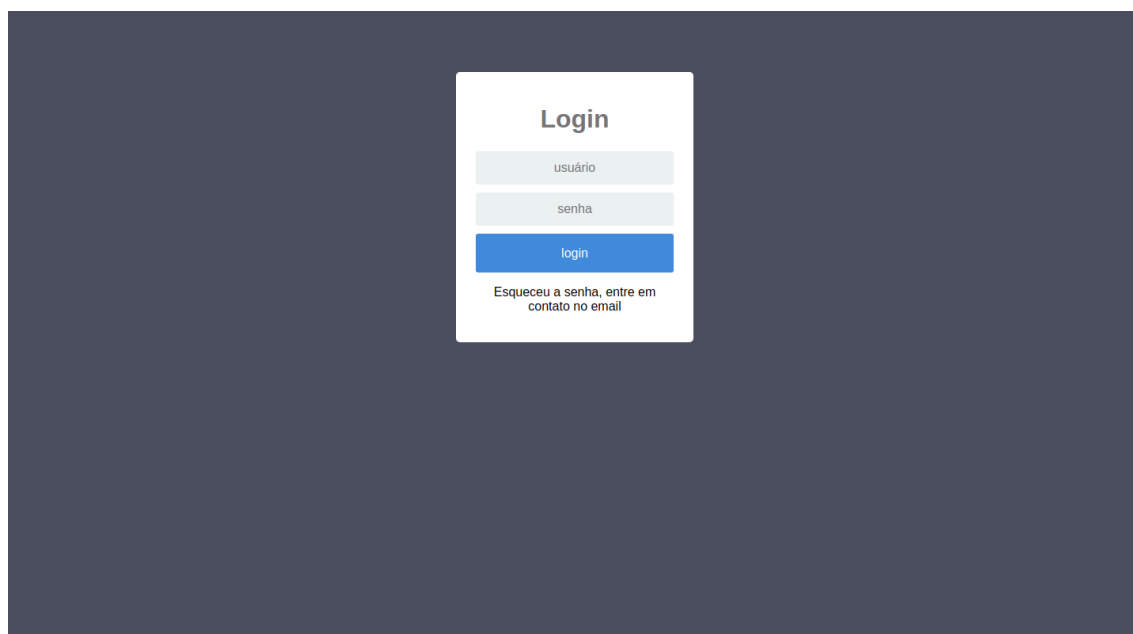


Figura 1: Tela de login visto pelo usuário

Para a criação da tela de desenvolvimento, bem como as outras telas da plataforma foi utilizado o *framework Bootstrap* que facilita o desenvolvimento do *front-end*. Esse *framework* utiliza da linguagem *HTML* e *CSS* para o seu desenvolvimento.

Com o desenvolvimento da tela de *login*, partiu-se para o desenvolvimento do *back-end* que é o local que realiza as autenticações do usuário. Como nunca havia trabalhado com *Django* houve também a dificuldade de criar essa autenticação.

Explicado nos outros relatórios o *Django* trabalha com o modelo *MVC*, onde cada parte da plataforma fica no devido local, por exemplo as tabelas são criadas no pacote *model*, todos os *templates* usados, isto é, todas as janelas que serão visíveis para o usuário são colocados neste pacote, por fim, o pacote de controle que é feito para gerir todos os acessos e autenticações, em *Django* é chamado de *urls*. Pode ser visto na Figura 2, como foi criado a tela de *login* em *HTML*.

```
1 {% extends "website-base.html" %}
2 {% block main %}
3     <div id="login">
4         <form class="form-horizontal" name="LoginForm" action="/login/" method="post">
5             {% csrf_token %}
6             {% if next %}
7                 <input type="hidden" name="next" value="{{ next }}" />
8             {% endif %}
9             <div class="control-group">
10                 <label class="control-label" for="username">Username</label>
11                 <div class="controls">
12                     <input type="text" id="username" name="username" placeholder="Username">
13                 </div>
14             </div>
15             <div class="control-group">
16                 <label class="control-label" for="password">Password</label>
17                 <div class="controls">
18                     <input type="password" name="password" id="password" placeholder="Password">
19                 </div>
20             </div>
21             <div class="control-group">
22                 <div class="controls">
23                     <button type="submit" class="btn">Login</button>
24                 </div>
25             </div>
26         </form>
27     </div>
28 {% endblock %}
```

Figura 2: Código realizado para a tela de Login

Como o *Django* trabalha com a linguagem de programação *Python*, todos os métodos *GET* e *POST* devem ser feitos em *Python* o que ocorreu uma outra dificuldade e um tempo gasto para aprendizado. Primeiro nas linhas 1-2, é criado um *annotation*, isto é, é especificado que aquela parte do código é para ser executado da seguinte forma. Em *Python* o método *POST* é feito especificando primeiro na *class* do form e logo após é criado variáveis que realizarão o envio e recebimento dos valores, este valor enviado é especificado na linha 7 - `{{ next }}`.

Criado a tela de *login* no *view*, deve-se especificar as *urls*, isto é, o *controller*, onde fica especificado qual o caminho que se deve realizar após o acesso do usuário. Essa especificação é feita por expressão regular, o código é mostrado na Figura 3.

```
1 from django.conf.urls.defaults import patterns, include, url
2
3 urlpatterns = patterns('',
4     url(r'^main/$', 'example.views.main'),
5     (r'^login/$', 'example.views.login_user'),
6     (r'^exercicios/$', 'example.views.exercicios')
7 )
```

Figura 3: Especificação do caminho do usuário no arquivo *urls.py*

Por fim, é implementado a *view* que fica responsável para a requisição com banco de dados e autenticação do usuário. Na Figura 4 pode ser visualizado a implementação da *view*.

```

1 from django.http import *
2 from django.shortcuts import render_to_response, redirect
3 from django.template import RequestContext
4 from User.models import *
5 from django.contrib.auth.decorators import login_required
6 from django.contrib.auth import authenticate, login, logout
7
8 def login_user(request):
9     logout(request)
10    username = password = ''
11    if request.POST:
12        username = request.POST['username']
13        password = request.POST['password']
14
15        user = authenticate(username=username, password=password)
16        if user is not None:
17            if user.is_active:
18                login(request, user)
19                return HttpResponseRedirect('/main/')
20    return render_to_response('login.html', context_instance=RequestContext(request))
21
22 @login_required(login_url='/login/')

```

Figura 4: Implementação da view para o login do aluno

Para o desenvolvimento da view hoje também um tempo para o aprendizado do funcionamento do método *authenticate*, bem como entender os métodos *GET* e *POST* em *Python*, pois até o momento tinha feito esses métodos em *PHP*, que é muito diferente. Contudo com o aprendizado a implementação se torna rápida e tranquila de se fazer, sendo até melhor que o desenvolvimento em *PHP*.

Uma outra dificuldade interessante é que os arquivos *HTML* não funciona os códigos em CSS se não estiver dentro do mesmo arquivo *.html*, para resolver o problema foram criados os CSS dentro do arquivo *.html*, contudo, o código fica desorganizado e poluído.

A partir da autenticação do usuário, foi criado a tela de exercícios, onde o aluno estará desenvolvendo os algoritmos e também realizando a compilação nas placas de vídeo que estarão instanciadas para a plataforma.

A implementação da tela de exercícios é comparável a tela de login, contudo sua diferença é que o aluno desenvolverá o código na plataforma, quando ele clicar no botão **Compilar**, este salvará o código desenvolvido na base de dados, para que haja um controle e estatística por parte do professor e depois requisitará à placa de vídeo o resultado da execução do código. Quando o aluno verificar que está correto seu código, ele pode ir para os próximos exercícios que estarão no conto direito da tela Na Figura 5 é mostrado a tela de exercícios.

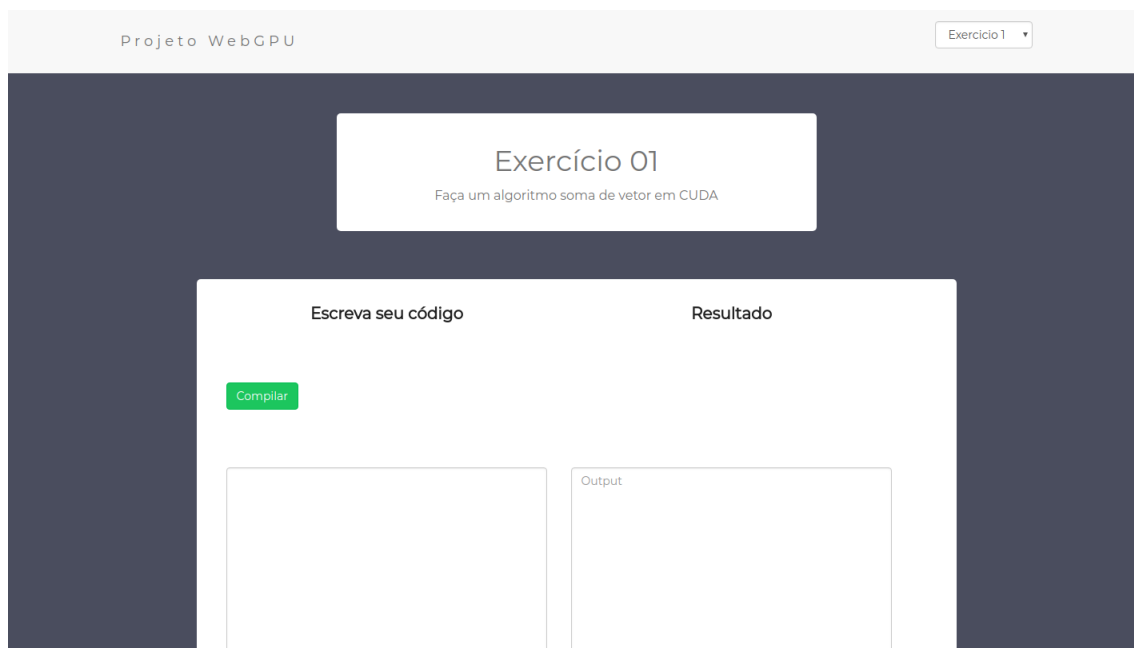


Figura 5: Tela de exercícios onde o aluno poderá desenvolver seus códigos CUDA

Outra dificuldade encontrada nesta parte do desenvolvimento foi entender como que havia o controle de concorrência por parte do *Python* na utilização da placa de vídeo. Até o momento está um pouco abstrato o seu funcionamento, para realizar a chamada para a placa de vídeo é usado a biblioteca *sys* que não tem nenhuma documentação que explica como que esse controle funciona.

Também houve a dificuldade no conceito do aluno desenvolver o código na plataforma, pois para enviar para a placa de vídeo deve ter um ajuste no código, a primeiro momento o ajuste foi gerar um arquivo no formato *.cu* (da linguagem CUDA) nos diretórios da máquina que está instanciada a placa de vídeo e esta faz a compilação e execução.

A plataforma ainda não esta em funcionando, para instanciar deve-se realizar modificações na biblioteca CUDA e também atualização do sistema operacional (Debian), estas modificações e atualizações serão feitas após o uso das máquinas para o Trabalho de Conclusão de Curso do aluno João Martins.

A partir de instanciada será realizado modificações e manutenções para que a plataforma fique apto a ter alunos e ser usado para o minicursos e projetos voltado a educação para desenvolvimento de código para plataformas heterogêneas.

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