



## 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 04/02/2018 a 18/02/2018

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

Na primeira semana foi realizado reuniões com o supervisor Rogério Aparecido Gonçalves para coletar os requisitos iniciais do projeto.

A primeira reunião fizemos a ideia geral do projeto que consiste em uma plataforma web para suporte a cursos de computação paralela. Essa ferramenta consiste em um ambiente online onde o aluno que estivesse cadastrado no curso de computação paralela possa estar desenvolvendo seus códigos e executando sem necessariamente precisar que seu computador tenha suporta a programação paralela. O primeiro esboço pode ser visualizado na Figura 1.

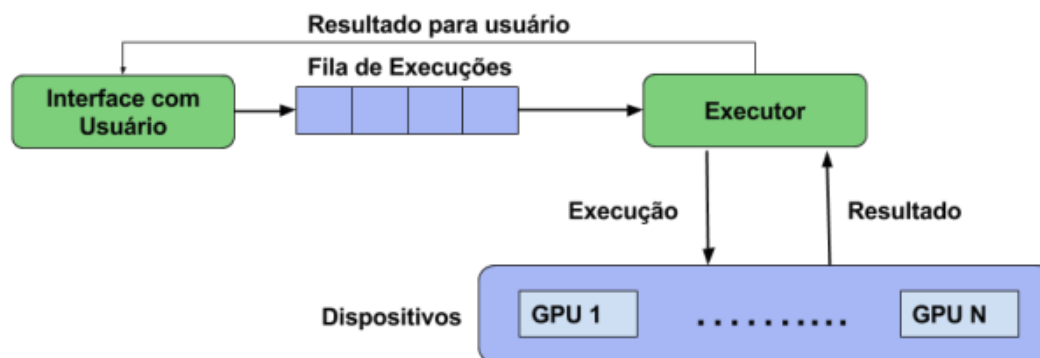


Figura 1: Funcionamento da Plataforma

O funcionamento do projeto consiste primeiro em uma interface com usuário que pode ser visto na Figura 2, em que o aluno estaria realizando os exercícios propostos pelo curso e a partir que seu código estivesse completo, realiza uma chamada enviando seu código para alguma GPU que esta em espera e o resultado obtido é retornado para a interface com o usuário.

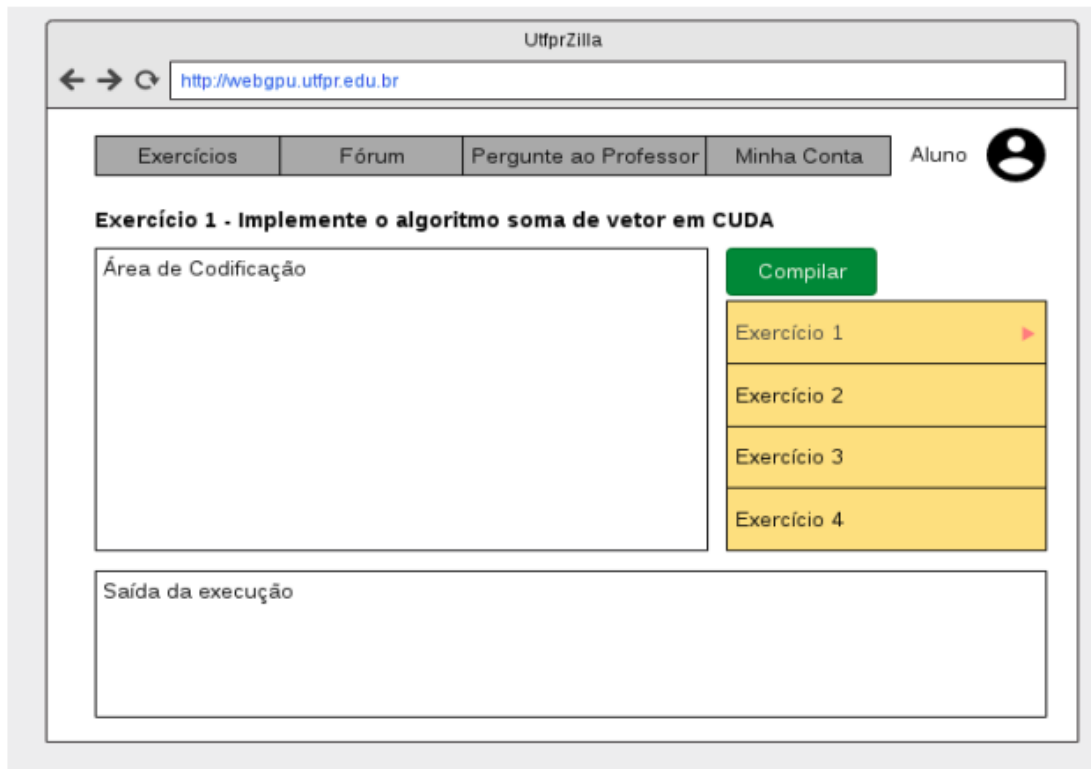


Figura 2: Modelo de Interface para a ferramenta

Assim que criamos este esboço, na segunda reunião realizamos a pesquisa de quais ferramentas poderiam ser usadas, essa escolha foi feita a critério do aluno com base em seus conhecimentos de programação, principalmente nas linguagens de programação que as ferramentas utilizam, outro critério que levaria em conta é se a ferramenta teria uma fila de execuções já implementada o que diminui o tempo de desenvolvimento do projeto.

A partir destes critérios foram escolhidas duas ferramentas *Flask* e *Django* ambas ferramentas desenvolvidas na linguagem de programação *Python* e que possui suporte a uma fila de execuções.

Com a escolha da ferramenta foi usado de um tempo para conhecimento de como a ferramenta funciona, também ficou escolhido em reunião a utilização do banco de dados *MariaDB* para implementação das tabelas que seriam utilizados, sendo que foi criado três tabelas uma do Administrador que pode estar visualizando as respostas dos alunos bem como estar escrevendo novos exercícios, uma tabela Usuário que consiste nos dados do aluno e uma tabela Exercícios que contém os exercícios propostos, a configuração, bem como os relacionamentos entre as tabelas pode ser visto na Figura 3.

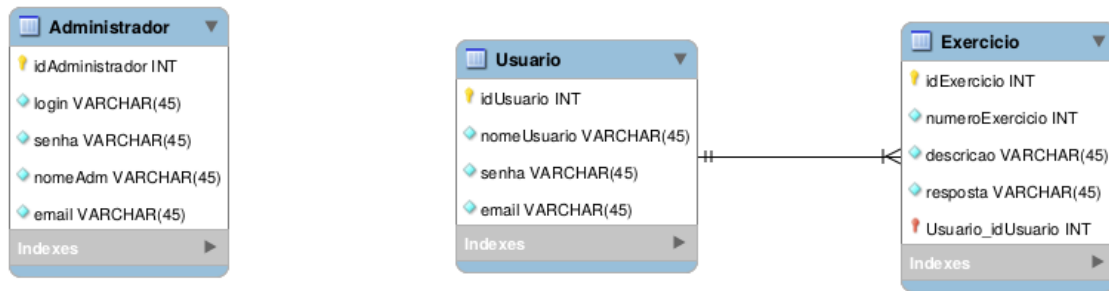


Figura 3: Tabela e os respectivos Relacionamentos

A partir da criação das tabelas foi desenvolvido um primeiro esboço com a utilização da ferramenta *Flask* para a união entre as tabelas e a ferramenta bem como criar uma tela inicial. O código a seguir mostra o esboço inicial para o funcionamento do website.

```

1  import os, subprocess, sys, time
2  from subprocess import check_output
3  import flask
4  from flask import Flask, render_template, request, redirect, url_for, send_from_directory
5  from werkzeug import secure_filename
6
7  app = Flask(__name__)
8
9  @app.route('/')
10 def index():
11     return render_template('index.html')
12
13
14 if __name__ == '__main__':
15     app.run(
16         host="localhost",
17         port=int("5000"),
18         debug=True
19     )

```

Código 1: Esboço inicial do projeto Web

O *Flask* é uma ferramenta dividida em três pilares que deve-se conhecer, primeiro o **Werkzeug** que é uma biblioteca para desenvolvimento de *apps* WSGI que é a especificação universal de como deve ser a interface entre um *app* Python e um *web server*. Ele possui implementado de padrão controle de cache, *cookies*, *status HTTP*, roteamento de *URLs* e também conta com uma poderosa ferramenta de debug.

O segundo pilar é o **Jinja2** que é um *template* escrito em *Python*, onde os *templates* utilizando marcações como `{{ nome_da_variavel }}` ou `{% for nome in lista_de_nomes %} Hello {{nome}}!! {% endfor %}` e o Jinja se encarrega de renderizar este template, ou seja, ele substitui os placeholders pelo valor de suas variáveis. O Jinja2 já vem com a implementação da maioria das coisas necessárias na construção de templates html.

O terceiro pilar é denominado de **Good Intentions** que tem os padrões de projeto e extensões como *Blueprints* para que você reaproveite os módulos que desenvolver. Tem uma comunidade bastante ativa que compartilha muitos projetos de extensões open-source como o *Flask Admin*, *Flask-Cache*, *Flask-Google-Maps*, *Flask-Mongoengine*, *Flask-SQLAlchemy*, *Flask-Login*, *Flask-Mail*.

Neste primeiro esboço foi criado somente a tela inicial do projeto, o Código 1 mostra que o *Flask* utiliza de rotas pré estabelecidas sendo que a rota principal é o `/` sendo assim quando o usuário digitar o endereço da página no navegador irá aparecer o template denominado *index.html* que pode ser visto na Figura 4.

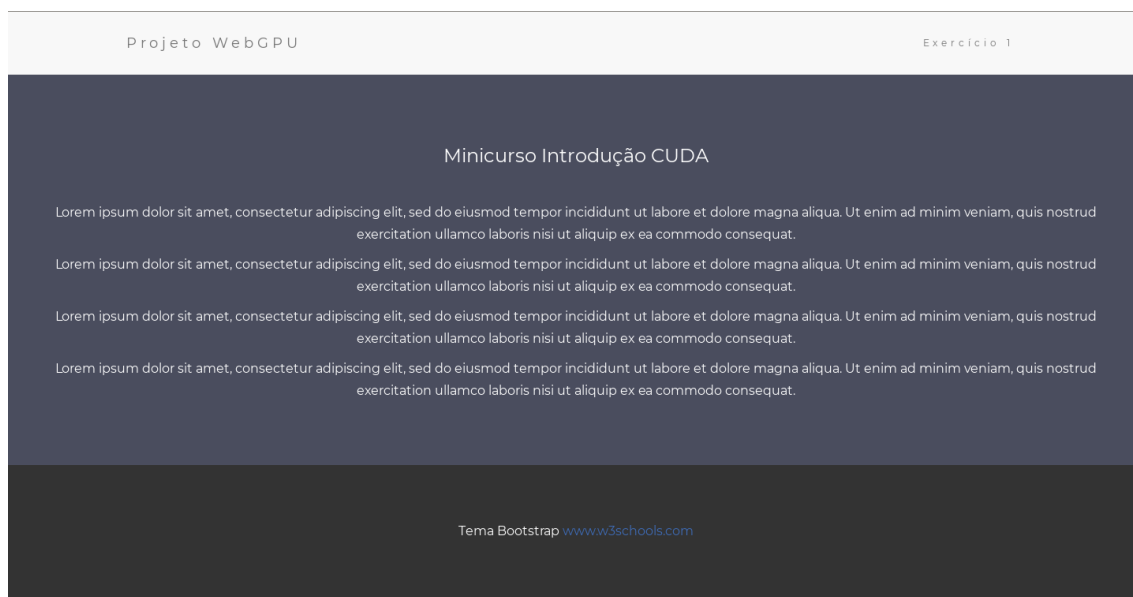


Figura 4: Tela Inicial do projeto WebGPU

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      ____/____/____
<hr/>	<hr/>	<hr/>

Autor: Lucio Valentin