

PLANNING YOUR SERIES- ESTUDO COMPARATIVO ENTRE AS LINGUAGENS DE PROGRAMAÇÃO: PHP e PYTHON

Camilly do Nascimento Ghellar¹, Alice Lima dos Santos², Luiz Ricardo Uriarte³, Ricardo de La Rocha Ladeira⁴, Hylson Vescovi Netto⁵

Instituto Federal Catarinense – Campus Blumenau – Rua Bernardino José de Oliveira, 81,
Blumenau – SC – Brazil

ghellarcamily@gmail.com¹, alicelimasantos@gmail.com², luiz.uriarte@ifc.edu.br³, ricardo.ladeira@ifc.edu.br⁴,
hylson.vescovi@ifc.edu.br⁵

Tema: pesquisa e análise acerca das diferentes ferramentas que podem ser utilizadas para a criação de uma plataforma com o intuito de organizar séries.

Escopo: desenvolvimento de um estudo comparando as linguagens de programação Python e PHP, através da implementação destas linguagens em um software gerenciador de séries de TV que se organiza em três diferentes tópicos: assistidos, assistindo e quero assistir. As duas linguagens são conectadas a um mesmo front-end. O usuário poderá achar as séries através de uma barra de pesquisas, localizando-as em um banco de dados. Esse trabalho tem como objetivo comparar as funcionalidades, como desempenho na velocidade, processamento de dados e capacidade de gerenciamento do software desenvolvido. As pesquisas e o desenvolvimento do trabalho serão registrados por meio da plataforma Google Documentos.

SUMÁRIO

- 1- CRONOGRAMA
- 2- REQUISITOS FUNCIONAIS:
- 3- REQUISITOS NÃO FUNCIONAIS:
- 4- DESENVOLVIMENTO
- 5- TELAS DE ENTRADA, CADASTRO E INDEX DO SITE COM HTML E CSS
- 6- CORES PADRÃO DO LAYOUT:
- 7- TESTE DE CONTRASTE
- 8- FERRAMENTAS UTILIZADAS
- 9- RESULTADOS DOS TESTES DE DESEMPENHO
- 10- TRABALHOS CORRELATOS
- 11- REFERÊNCIAS

1- CRONOGRAMA

O que fazer	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
Definição do tema	X								
Levantamento de ideias para realizar no software	X								
Mapa mental sobre o que será desenvolvido	X								
Definição de layout	X								
Pesquisa de cores para layout	X								
Realizar protótipo de interface gráfica - inicial	X								
Criação do repositório no GitHub	X								
Levantamento de requisitos funcionais	X	X							
Levantamento de requisitos não funcionais	X	X	X						
Desenvolvimento de modelos conceituais		X							
Pesquisas de linguagens de programação		X	X						
Criação do esqueleto da plataforma em HTML e CSS		X	X	X					
Utilizando PHP									
Implementação da programação na tela de cadastro			X						
Pesquisas acerca da implementação do banco de dados na linguagem		X	X				X		
Implementação do banco de dados no cadastro			X						
Implementação da programação na tela inicial			X			X	X	X	X
Implementação do banco de dados na tela inicial			X			X	X	X	X
Utilizando Python									
Implementação da programação na tela de cadastro			X	X					
Pesquisas acerca da implementação do banco de dados na linguagem		X	X	X			X		
Implementação do banco de dados no cadastro				X	X				
Implementação da programação na tela inicial				X	X	X	X	X	X
Implementação do banco de dados na tela inicial				X	X	X	X	X	X
Utilizando JavaScript									
Escolha de qual back-end será utilizado					X				
Conexão do formulário de cadastro com o back-end e banco de dados					X				

Conexão do formulário de login com o back-end e banco de dados					X	X	X	X	X
Conexão da função listar as séries (back-end) e listar elas no front-end					X	X	X	X	X
Testes computacionais			X	X	X	X	X	X	X
Resolução de falhas			X	X	X	X	X	X	X
Testes comparativos entre as linguagens							X	X	
Reunião para análise do desenvolvimento do trabalho	X	X	X	X	X	X	X	X	X
Desenvolvimento da documentação da pesquisa no Google Documentos	X	X	X	X	X	X	X	X	X
Finalização da documentação								X	X
Finalização do software								X	X

2- REQUISITOS FUNCIONAIS:

[RF001] Ao acessar a plataforma, abrirá uma tela para o usuário clicar para escolher qual linguagem deseja executar a página, sendo as opções: PHP e Python;

[RF002] Após o clique para a escolha da linguagem, o usuário será redirecionado à tela de cadastro;

[RF003] Na página de cadastro, haverá a opção, no fim do formulário, de pular para a tela de login caso o usuário já possua uma conta;

[RF004] Para criar a conta, o usuário deverá informar seu nome, e-mail e uma senha;

[RF005] Ao criar sua conta, o usuário não entrará diretamente na conta, devendo ir até a aba “Entrar”;

[RF006] Nada aba “Entrar”, o usuário deverá informar seu e-mail e senha para ingressar na plataforma;

[RF007] Ao fazer o login, será apresentado na tela um menu lateral do perfil do usuário;

[RF008] Clicando no menu do perfil, aparecerá o espaço para que o usuário possa inserir uma imagem de perfil, o nome do mesmo, a opção de adicionar séries em cada classificação da situação atual, e o botão de sair da conta;

[RF009] Na páginas inicial, o usuário visualizará três colunas, “Quero assistir”, “Assistindo” e “Assistidos”, na qual listará as séries que o usuário deseja assistir, está assistindo no momento e as que o usuário já terminou de assistir, respectivamente;

[RF010] Ao lado de cada nome de série adicionada pelo usuário, haverá as opções de editar e excluir;

[RF011] As séries listadas na coluna “Assistindo” terão um ‘check’ para o usuário clicar quando terminar de assistir uma determinada série, fazendo com que a mesma seja encaminhada para a coluna “Assistidos”;

[RF012] Ao adicionar uma série que se interesse e deseja assistir, o usuário terá a funcionalidade “+”, que encaminhará o nome da série para a lista “Quero assistir”;

[RF013] Na coluna “Quero assistir”, o usuário terá a opção de clicar em “... ” (três pontos), que enviará o nome da série para a coluna “Assistindo”;

[RF014] Ao clicar no botão “Sair”, o usuário fará logout da sua conta no site.

3- REQUISITOS NÃO FUNCIONAIS:

[RNF001] A conta e as informações de um usuário não poderão ser acessadas por outros, exceto se possuir as informações de cadastro;

[RNF002] O sistema será implementado na primeira versão através da linguagem de programação PHP;

[RNF003] O sistema será implementado na segunda versão através da linguagem de programação Python;

[RNF004] O sistema terá também códigos na linguagem de programação JavaScript;

[RNF005] Em JavaScript, será utilizado o mesmo código de conexão para Python e para PHP;

[RNF006] Juntamente com as linguagens de programação, todas as versões contarão com a linguagem de marcação HTML e estilização CSS;

[RNF007] O usuário deverá ser identificado a partir do e-mail informado no cadastro;

[RNF008] O usuário deverá estar em sua conta para modificar, adicionar e excluir as séries;

[RNF009] As séries adicionadas ficarão expostas em colunas na tela principal.

[RNF010] Os registros feitos utilizando Python ou PHP serão salvos em um mesmo banco de dados.

4- DESENVOLVIMENTO

O trabalho será desenvolvido em duas etapas. A primeira será a realização da tela de cadastro e a implementação do banco de dados na mesma, seguido do começo do desenvolvimento da tela inicial do site em todas as linguagens. No final dele, ou seja, na segunda etapa, haverá a retomada de cada código resolvendo possíveis falhas, inserindo melhorias e implementando as séries, através do banco de dados. Haverá duas fases, utilizando uma linguagem de programação em cada, sendo estas, respectivamente: PHP, Python, e ambas juntamente com JavaScript.

Na primeira fase da primeira etapa foi utilizada a linguagem de programação PHP, a linguagem de marcação HTML estilizada com CSS.

O HTML (Hypertext Mark-up Language), que foi utilizado na criação do front-end, é a linguagem de marcação padrão em páginas web. É um padrão que descreve a estrutura de páginas web usando tags, instruindo o navegador a como exibir o conteúdo. Sem alguma programação no back-end, é considerado um esqueleto ou o layout da página. Sendo o parte essencial da rede mundial de computadores, possui algumas versões, tendo a HTML5 como a mais recente.

Foi utilizado o CSS para estilizar as páginas web criadas com o HTML. O CSS (Cascading Style Sheets) é a linguagem que descreve o layout e a formatação de cada elemento da página, como cores e fontes, também ajudando a renderizar a página de forma responsiva em dispositivos de diferentes tamanhos. Pode-se utilizar o CSS em três estilos de formatação, o chamado estilo inline, o interno e o externo, sendo que todos os estilos foram utilizados no presente projeto.

Utilizando o HTML e o CSS foram elaborados os formulários de cadastro e login de usuário, e edição e inserção da série. Da mesma forma, foi construída a página principal, onde aparece as colunas das séries adicionadas e a aba lateral de perfil do usuário.

Houve o uso da linguagem interpretada livre PHP (Hypertext Preprocessor) por esta ser uma linguagem de prévio conhecimento das desenvolvedoras, facilitando o desenvolvimento e aplicação do mesmo. Esta linguagem é muito adequada para o desenvolvimento web pois pode ser embutida dentro do HTML. O PHP (extensão *php*) foi utilizado para fazer o cadastro e o login do usuário. Ao usuário se cadastrar com seu nome, e-mail e senha, as informações são enviadas ao banco de dados MySQL no PHPMyAdmin com a configuração e a programação feitas por meio da linguagem PHP, e neste banco ficam armazenados os registros. Com as informações de cadastro registradas no banco de dados, o usuário pode efetuar seu login, que também é feito por meio do PHP, onde consulta-o e confirma se os dados digitados por quem deseja acessar a conta já existem. Para realizar os códigos, houveram estudos em cima de vídeos sobre a linguagem.

```

1  <?php
2      include("conexao.php");
3      echo "inciando cadastro";
4      $nome = $_GET['nome'];
5      $email = $_GET['email'];
6      $senha = md5($_GET['senha']);
7
8      $sql = "INSERT INTO usuario(nome, email, senha)
9      VALUES ('$nome', '$email', '$senha')";
10
11     if(mysqli_query($conexao, $sql)){
12         $parte1 = array('resultado' => 'ok');
13         $parte2 = array('detalhes' => 'ok');
14         $resposta = json_encode(
15             array_merge(
16                 $parte1,
17                 $parte2
18             )
19         );
20         echo $resposta;
21     }
22
23     else{
24         $parte1 = array('resultado' => 'erro');
25         $parte2 = array('detalhes' => 'não foi possível realizar o cadastro');
26         $resposta = json_encode(
27             array_merge(
28                 $parte1,
29                 $parte2
30             )
31         );
32         echo $resposta;
33     }
34     echo "terminou cadastro";
35
36     mysqli_close($conexao);
37
38  ?>

```

Figura 1- imagem do código PHP que realiza o cadastro.

	id	nome	email	senha
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	3	camilly	camilly@gmail.com	202cb962ac59075b964b07152d234b70
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	4	ana	ana@gmail.com	5390489da3971cbbcd22c159d54d24da
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	5	Alice	alice@gmail.com	123
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	6	Camilly	camilly@gmail.com	456
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	7	Alice	alice@gmail.com	123
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	8	Camilly	camilly@gmail.com	456
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	9	Alice	alice@gmail.com	123
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	10	Camilly	camilly@gmail.com	456
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	11	uriarte	uriarte@gmail.com	fb73bb87a71c2d05c38acfd5f0509836
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	12	Uriarte	uriarte@gmail.com	uriarte123
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	13	professor	camilly@gmail.com	456

Figura 2- imagem do banco de dados utilizado juntamente com seus testes.

Para fazer a ligação do front-end em HTML com o back-end em Python ou em PHP, foi utilizado a linguagem JavaScript. O JavaScript (extensão *js*) é uma linguagem de

programação leve, interpretada e baseada em objetos com funções de primeira classe. É considerada a principal linguagem de script para páginas Web. Utilizando esta, foi feita a seleção da linguagem escolhida pelo usuário para rodar na página, o cadastro, o login e a confirmação do login para entrar na página principal da plataforma.

Para executar a plataforma, o JavaScript guarda na sessão a linguagem de programação selecionada pelo usuário, e esta é referenciada em todos os arquivos .js. Para fazer o cadastro do usuário, o JavaScript é responsável por localizar a linguagem que será usada, mapear o “click” no botão de confirmação do cadastro, pegar os dados da tela, enviar os dados do cadastro para o backend em formato *json*, e dar um retorno para o usuário após o cadastro, que pode ser uma mensagem de erro, ou redirecionar o usuário para a página de login. No login, o JavaScript faz praticamente as mesmas funções que no cadastro, localiza a linguagem usada, mapeia o “click” no botão de confirmação, pega os dados da tela, envia os dados para o backend em formato *json*, e dá um retorno para o usuário após o login, que pode ser uma mensagem de erro, ou redirecionar o usuário para a página principal da sua conta. O código de checar o login confere se o usuário acessou sua conta. Se sim, o usuário tem acesso a sua conta, juntamente com uma mensagem de “Bem vindo”, do contrário, aparecerá a mensagem “Você ainda não fez login. Faça agora”, sendo este ‘Faça agora’ um link clicável para a tela de login.

```
1  $(function () { // escolher a linguagem do site
2
3      $(document).on("click", "#bt_php", function () {
4          //guarda na sessão que a linguagem escolhida é o php
5          sessionStorage.setItem('linguagem_usada', 'php');
6
7          // encaminha para a página de cadastro
8          window.location = 'cadastrar.html'; });
9
10     $(document).on("click", "#bt_flask", function () {
11         //guarda na sessão que a linguagem escolhida é o flask
12         sessionStorage.setItem('linguagem_usada', 'flask');
13
14         // encaminha para a página de cadastro
15         window.location = 'cadastrar.html'; });
16 });
```

Figura 3- imagem do código responsável pela escolha de linguagem feita pelo usuário.

```

1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4     <title>PYS</title>
5
6     <meta charset="UTF-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <meta http-equiv="X-UA-Compatible" content="ie=edge">
9     <meta name="refresh" content="2000">
10    <meta name="description" content="Entrar no site Planning Your Series."/>
11
12    <link rel="stylesheet" href="estilo.css">
13
14 </head>
15
16 <body>
17     <div class="conteudo" >
18         <h1>ESCOLHA A LINGUAGEM QUE DESEJA UTILIZAR:</h1>
19         <br>
20
21         <div class="caixa">
22             <form>
23
24                 <button type="submit" style="font-size: 15px;" id="bt_php">PHP</button>
25                 <br>
26                 <br>
27                 <button type="submit" style="font-size: 15px;" id="bt_flask">PYTHON</button>
28
29             </form>
30         </div>
31     </div>
32     <script src="js/escolher_linguagem.js"></script>
33 </body>
34 </html>

```

Figura 4- imagem do código HTML que mostra ao usuário a tela de escolha de linguagem que faz ligação ao código da figura 3.

Na segunda fase da primeira etapa foi utilizada a linguagem de programação Python e novamente a linguagem de marcação HTML e estilizado com CSS.

O Flask -micro-framework web escrito em Python- foi escolhido por ser um modelo mais rápido, simples e leve, porém ao mesmo tempo que suprisse a necessidade de obter ferramentas que o desenvolvimento do site exige. Mesmo que básica, essa linguagem ainda conta com o auxílio de bibliotecas que podem ser utilizadas ao longo do trabalho. Essas extensões servem para casos mais complexos, como a conexão do site ao banco de dados.

Com o Python (extensão *py*), foi desenvolvido o back-end, as tabelas do banco de dados e definida as rotas de login, logout e listagem das séries, que aparecerão no perfil do usuário quando forem designadas para uma das três opções: assistido, assistindo e assistir, além dos testes de funcionamento de cada um.


```

1  # importações
2  from geral import *
3  from modelo.usuario import *
4
5  @app.route("/login", methods=['POST'])
6  def login():
7      resposta = jsonify({"resultado": "ok", "detalhes": "ok"})
8
9      # receber as informações do novo objeto
10     dados = request.get_json()
11     login = dados['login']
12     senha = dados['senha']
13
14     if login == 'mylogin' and senha == '123':
15         # armazenar sessão
16         session[login] = "OK"
17     else:
18         resposta = jsonify({"resultado": "erro", "detalhes": "login e/ou senha inválido(s)"})
19     resposta.headers.add("Access-Control-Allow-Origin", meuservidor)
20     return resposta

```

Figura 5- imagem do código Python que realiza o login no site.

Para a segurança da senha do usuário, foi utilizado o algoritmo MD5 (Message-Digest algorithm 5). Ele é um algoritmo de hash usado por softwares para autenticar mensagens e verificar conteúdos e assinaturas digitais. Ele converte os dados informados para uma *string* de 32 caracteres. Esses caracteres são padrão, ou seja, todas as vezes que a palavra “olá” for informada, resultará no mesmo código (ab11fa17ff3337a200cb2714dbc2318c). Por esse motivo, o método MD5 não é considerado o mais seguro a ser utilizado atualmente, porém como o intuito deste trabalho é apenas a comparação entre as duas linguagens anteriormente citadas, ele cumpriu com sua finalidade.

A inserção do MD5 na linguagem PHP se deu nos arquivos “*cadastro.php*”, ao incluir `$senha = md5($_GET['senha'])` na linha de código 6, e “*verificacaoLogin.php*”, com `senha = md5('$senha')`, na linha de código 11, juntamente com a consulta no banco de dados através do código `$sql = "SELECT * FROM cadastro_usuario WHERE email = '$email' and senha = md5('$senha')";`.

```

include("conexao.php");
echo "inciando cadastro";
$nome = $_GET['nome'];
$email = $_GET['email'];
$senha = md5($_GET['senha']);

```

Figura 6- imagem do código que armazena as informações de cadastro.

```

// Acessa o sistema
include_once('conexao.php');
$email = $_POST['email'];
$senha = $_POST['senha'];

$sql = "SELECT * FROM cadastro_usuario WHERE email = '$email' and senha = md5('$senha')";

```

Figura 7- imagem do código que realiza o login e acessa o sistema.

5- TELAS DE ENTRADA, CADASTRO E INDEX DO SITE COM HTML E CSS

The login screen features a light gray background. At the top center, the word "ENTRAR" is displayed in a bold, black, sans-serif font. Below this, a darker gray rectangular box contains the login form. Inside this box, there are two white input fields: the top one is labeled "E-mail" and the bottom one is labeled "Senha". Below these fields is a purple button with the word "ENTRAR" in white, uppercase letters.

Figura 8- imagem da tela de entrada do PYS.

The registration screen has a light gray background. At the top center, the words "CRIAR CONTA" are displayed in a bold, black, sans-serif font. Below this, a darker gray rectangular box contains the registration form. Inside this box, there are three white input fields: the top one is labeled "Nome de usuário", the middle one is labeled "E-mail", and the bottom one is labeled "Senha". Below these fields is a purple button with the words "CADASTRAR-SE" in white, uppercase letters.

Figura 9- imagem da tela de cadastro do PYS.

6- CORES PADRÃO DO LAYOUT:

preto: #000000

cinza claro: #DCDCDC

cinza médio: #c0c0c0

cinza escuro: #393939

roxo escuro: #4B0082

7- TESTE DE CONTRASTE

Ao definir a tabela de cores do site, foi realizado o teste de contraste com as cores que estão mais frequentemente em contraste, roxo com cinza médio e cinza claro com cinza escuro, para a acessibilidade do site na plataforma WebAIM, e foi obtido como resultado:

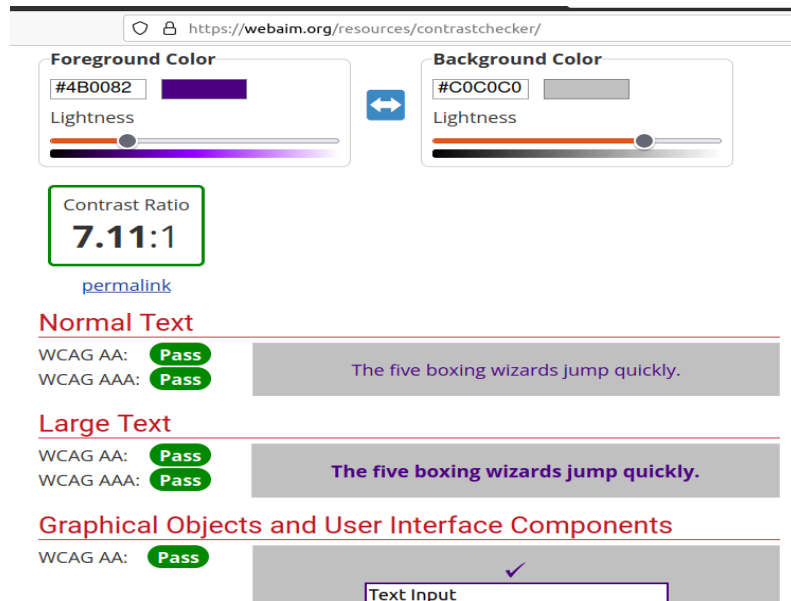


Figura 10- imagem da nota de contraste entre roxo com cinza médio.

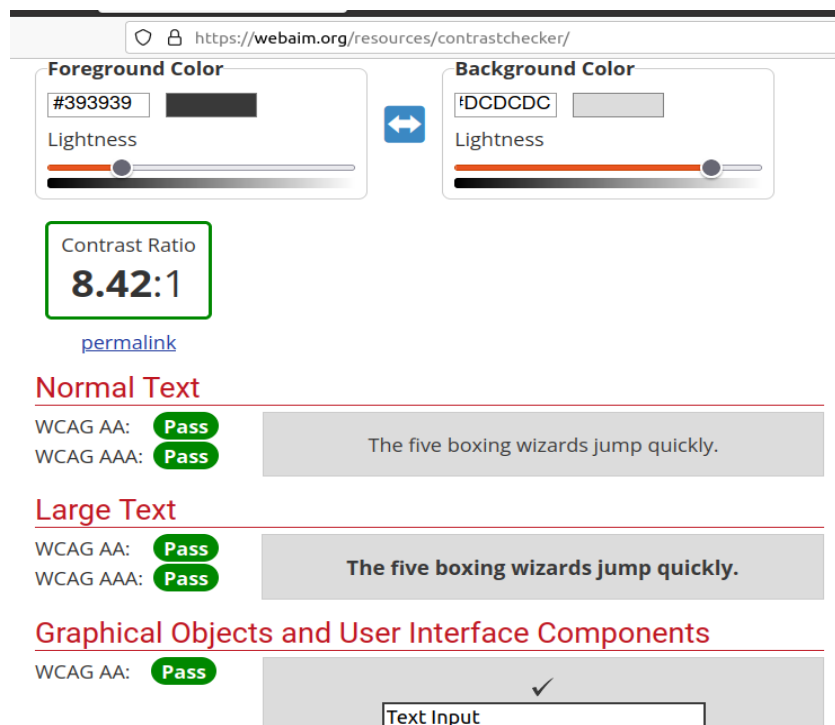


Figura 11- imagem da nota de contraste entre cinza claro com cinza escuro.

Tendo em vista que as normas da WCAG 2.0 nível AA exigem uma nota de contraste de no mínimo 4.5:1 para um texto normal e 3:1 para um texto grande, as notas de contraste das cores do PYS estão dentro das normas de acessibilidade WCAG 2.0 AA.

8- FERRAMENTAS UTILIZADAS

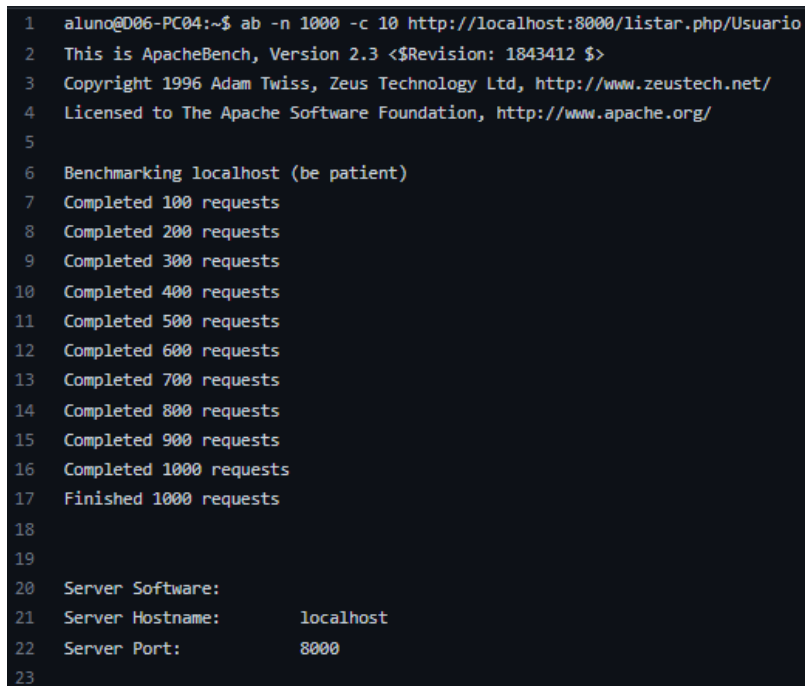
- GitHub
- HTML
- CSS
- PHP
- Banco de dados SQL e MySQL - PHPMyadmin no localhost
- Python
- JavaScript
- LibreOffice Calc
- Google Docs

9- RESULTADOS DOS TESTES DE DESEMPENHO

Após o desenvolvimento e os testes realizados, conclui-se que a melhor linguagem no geral é a PHP. Foram realizados 3 (três) testes em cada uma das linguagens em 3 (três) máquinas diferentes, sendo duas delas no sistema operacional Windows e uma no Linux. Para análise, todas as abas dos computadores estavam sem abas externas abertas e com a rede internet desligada.

Além disso, foram testados com 10 conexões simultâneas e 1000 requisições. Com os resultados obtidos, foi realizada a média e o desvio padrão dos tempos. Dessa forma, pôde ser notado que a linguagem PHP é a mais rápida e eficiente quando comparadas ao Python, tendo notável desempenho nos testes de carga e análise de performance. Esse resultado não foi interferido mesmo com o uso de diferentes sistemas operacionais, pois a diferença do tempo foi mínima.

- Linguagem PHP
 - Teste no Linux (Ubuntu)



```
1 aluno@006-PC04:~$ ab -n 1000 -c 10 http://localhost:8000/listar.php/Usuario
2 This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
3 Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
4 Licensed to The Apache Software Foundation, http://www.apache.org/
5
6 Benchmarking localhost (be patient)
7 Completed 100 requests
8 Completed 200 requests
9 Completed 300 requests
10 Completed 400 requests
11 Completed 500 requests
12 Completed 600 requests
13 Completed 700 requests
14 Completed 800 requests
15 Completed 900 requests
16 Completed 1000 requests
17 Finished 1000 requests
18
19
20 Server Software:
21 Server Hostname: localhost
22 Server Port: 8000
23
```

Figura 12- imagem da primeira parte do teste realizado no Linux.

```

24 Document Path:      /listar.php/Usuario
25 Document Length:    177 bytes
26
27 Concurrency Level:   10
28 Time taken for tests: 0.881 seconds
29 Complete requests:   1000
30 Failed requests:     0
31 Total transferred:   340000 bytes
32 HTML transferred:   177000 bytes
33 Requests per second: 1135.44 [#/sec] (mean)
34 Time per request:    8.807 [ms] (mean)
35 Time per request:    0.881 [ms] (mean, across all concurrent requests)
36 Transfer rate:       377.00 [Kbytes/sec] received
37
38 Connection Times (ms)
39           min  mean[+/-sd] median  max
40 Connect:     0    0   0.0    0    1
41 Processing:   2    9   1.1    8   13
42 Waiting:     2    9   1.1    8   13
43 Total:       2    9   1.1    8   14
44
45 Percentage of the requests served within a certain time (ms)
46  50%    8
47  66%    9
48  75%    9
49  80%    9
50  90%   10
51  95%   11
52  98%   12
53  99%   12
54 100%   14 (longest request)

```

Figura 13- imagem da segunda parte do teste realizado no Linux.

○ Teste no Windows

```

1 C:\Users\Acer>ab -n 1000 -c 10 http://localhost/pys_dp/backend/php/listar.php/Usuario
2 This is ApacheBench, Version 2.3 <$Revision: 1901567 $>
3 Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
4 Licensed to The Apache Software Foundation, http://www.apache.org/
5
6 Benchmarking localhost (be patient)
7 Completed 100 requests
8 Completed 200 requests
9 Completed 300 requests
10 Completed 400 requests
11 Completed 500 requests
12 Completed 600 requests
13 Completed 700 requests
14 Completed 800 requests
15 Completed 900 requests
16 Completed 1000 requests
17 Finished 1000 requests
18
19
20 Server Software:      Apache/2.4.51
21 Server Hostname:      localhost
22 Server Port:          80
23

```

Figura 14- imagem da primeira parte do teste realizado no Windows.

```

24 Document Path:      /pys_dp/backend/php/listar.php/Usuario
25 Document Length:    416 bytes
26
27 Concurrency Level:   10
28 Time taken for tests: 0.448 seconds
29 Complete requests:   1000
30 Failed requests:     0
31 Total transferred:   635000 bytes
32 HTML transferred:    416000 bytes
33 Requests per second: 2230.80 [#/sec] (mean)
34 Time per request:    4.483 [ms] (mean)
35 Time per request:    0.448 [ms] (mean, across all concurrent requests)
36 Transfer rate:       1383.36 [Kbytes/sec] received
37
38 Connection Times (ms)
39           min  mean[+/-sd] median   max
40 Connect:     0    0   1.0     0     8
41 Processing:   0    4   4.4     1    26
42 Waiting:     0    4   4.4     1    25
43 Total:       0    4   4.4     8    26
44
45 Percentage of the requests served within a certain time (ms)
46  50%    8
47  66%    8
48  75%    8
49  80%    8
50  90%    9
51  95%    9
52  98%    9
53  99%   17
54 100%   26 (longest request)

```

Figura 15- imagem da segunda parte do teste realizado no Windows.

- Linguagem Python Flask
 - Teste no Linux (Ubuntu)

```

1  aluno@006-PC04:~$ ab -n 1000 -c 10 http://localhost:5000/listar/Usuario
2  This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
3  Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
4  Licensed to The Apache Software Foundation, http://www.apache.org/
5
6  Benchmarking localhost (be patient)
7  Completed 100 requests
8  Completed 200 requests
9  Completed 300 requests
10 Completed 400 requests
11 Completed 500 requests
12 Completed 600 requests
13 Completed 700 requests
14 Completed 800 requests
15 Completed 900 requests
16 Completed 1000 requests
17 Finished 1000 requests
18
19
20 Server Software:      Werkzeug/2.1.2
21 Server Hostname:      localhost
22 Server Port:          5000
23

```

Figura 16- imagem da primeira parte do teste realizado no Linux.

```

24 Document Path:      /listar/Usuario
25 Document Length:    214 bytes
26
27 Concurrency Level:   10
28 Time taken for tests: 4.669 seconds
29 Complete requests:   1000
30 Failed requests:     0
31 Total transferred:   430000 bytes
32 HTML transferred:    214000 bytes
33 Requests per second: 214.16 [#/sec] (mean)
34 Time per request:    46.693 [ms] (mean)
35 Time per request:    4.669 [ms] (mean, across all concurrent requests)
36 Transfer rate:       89.93 [Kbytes/sec] received
37
38 Connection Times (ms)
39           min  mean[+/-sd] median  max
40 Connect:     0    0  0.0      0    1
41 Processing:   6   46 10.8     45   86
42 Waiting:     5   43 10.3     42   86
43 Total:       6   46 10.8     45   86
44
45 Percentage of the requests served within a certain time (ms)
46 50%    45
47 66%    49
48 75%    53
49 80%    55
50 90%    61
51 95%    66
52 98%    72
53 99%    77
54 100%   86 (longest request)

```

Figura 17- imagem da segunda parte do teste realizado no Linux.

- Teste no Windows

```

1 C:\Users\Acer>ab -n 1000 -c 10 http://localhost:5000/listar/Usuario
2 This is ApacheBench, Version 2.3 <$Revision: 1901567 $>
3 Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
4 Licensed to The Apache Software Foundation, http://www.apache.org/
5
6 Benchmarking localhost (be patient)
7 Completed 100 requests
8 Completed 200 requests
9 Completed 300 requests
10 Completed 400 requests
11 Completed 500 requests
12 Completed 600 requests
13 Completed 700 requests
14 Completed 800 requests
15 Completed 900 requests
16 Completed 1000 requests
17 Finished 1000 requests
18
19
20 Server Software:      Werkzeug/2.1.0
21 Server Hostname:      localhost
22 Server Port:          5000
23

```

Figura 18- imagem da primeira parte do teste realizado no Windows.


```

24 Document Path:      /listar/Usuario
25 Document Length:    214 bytes
26
27 Concurrency Level:   10
28 Time taken for tests: 4.363 seconds
29 Complete requests:   1000
30 Failed requests:     0
31 Total transferred:   411000 bytes
32 HTML transferred:    214000 bytes
33 Requests per second: 229.21 [#/sec] (mean)
34 Time per request:    43.627 [ms] (mean)
35 Time per request:    4.363 [ms] (mean, across all concurrent requests)
36 Transfer rate:       92.00 [Kbytes/sec] received
37
38 Connection Times (ms)
39      min  mean[+/-sd] median  max
40 Connect:    0    0   1.0    0    8
41 Processing:  0   23   5.4   24   64
42 Waiting:    0   22   5.4   24   64
43 Total:      0   23   5.4   24   64
44
45 Percentage of the requests served within a certain time (ms)
46  50%    24
47  66%    24
48  75%    24
49  80%    25
50  90%    26
51  95%    32
52  98%    33
53  99%    48
54 100%    64 (longest request)

```

Figura 19- imagem da primeira parte do teste realizado no Linux.

O teste realizado na terceira máquina se encontra no repositório de uma das desenvolvedoras (https://github.com/camily-ghellar/pys_dp), juntamente com mais detalhes sobre as informadas anteriormente.

Observamos, também, através do cálculo de média e desvio padrão que este último na linguagem Python é o dobro da linguagem PHP. Mais detalhes se encontram nas figuras abaixo:

Teste 01		Dados da máquina	
Teste no PC do IFC		Nome do dispositivo A04-PC10	
python- rota listar		Processador AMD® Ryzen 5 pro 2400g with radeon vega graphics × 8	
4,662	1º teste	Memória RAM utilizável 15,5 GiB	
4,677	2º teste	Tipo de sistema Ubuntu 20.04.4 LTS	
4,669	3º teste	64 bits	
4,669333333333333	média	Versão do GNOME 3.36.8	
0,00750555349946496	desvio padrão	Capacidade de disco 1,0 TB	
php- rota listar			
0,881	1º teste		
0,883	2º teste		
0,847	3º teste		
0,870333333333333	média		
0,0202319878739914	desvio padrão		

Teste 02		Dados da máquina	
Teste no notebook da Alice		Nome do dispositivo LAPTOP-ALICE	
python- rota listar		Processador Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz	
4,363	1º teste	RAM instalada 8,00 GB (utilizável: 7,83 GB)	
4,327	2º teste	Tipo de sistema Sistema operacional de 64 bits, processador baseado em x64	
4,349	3º teste	ID do dispositivo AAC614F1-5256-471C-892E-B3E00AC33878	
4,346333333333333	média	Edição Windows 11 Home Single Language	
0,0181475434517548	desvio padrão	Versão 21H2	
php- rota listar		Instalado em 28/12/2021	
0,448	1º teste	Compilação do SO 22000.1219	
0,475	2º teste	Experiência Pacote de Experiência de Recursos do Windows 1000.22000.1219.0	
0,473	3º teste		
0,465333333333333	média		
0,0150443787951957	desvio padrão		

Teste 03		Dados da máquina	
Teste no notebook do Lemuel		Nome do dispositivo manske-lap	
python- rota listar		Processador 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz	
6,86	1º teste	RAM instalada 8,00 GB (utilizável: 7,80 GB)	
6,947	2º teste	Tipo de sistema Sistema operacional de 64 bits, processador baseado em x64	
6,943	3º teste	ID do dispositivo 043B624A-DCF4-46FA-B924-6616FEE9C038	
6,916666666666667	média	Edição Windows 11 Home Single Language	
0,0491155101096722	desvio padrão	Versão 22H2	
php- rota listar		Instalado em 18/10/2022	
0,525	1º teste	Compilação do SO 22621.819	
0,491	2º teste	Número de série PE07F1EV	
0,476	3º teste	Experiência Windows Feature Experience Pack 1000.22636.1000.0	
0,497333333333333	média		
0,0251064400768674	desvio padrão		

GERAL PYTHON	
média	5,3107777777778
desvio padrão	0,0216165654

GERAL PHP	
média	0,611
desvio padrão	0,005031842762

Figura 20- Resultado das comparações entre as médias e desvios padrão entre as linguagens PHP e Python.

10- TRABALHOS CORRELATOS

KIANTO - SITE PARA ORGANIZAÇÃO DE ESTUDOS

Foi utilizado como base o modelo de código PHP do “KIANTO” na programação do sistema de login e de cadastro.

```
1  <?php
2      session_start();
3
4      if(isset($_POST['submit']) && !empty($_POST['email']) && !empty($_POST['senha']))
5      {
6          // Acessa o sistema
7          include_once('conexao.php');
8          $email = $_POST['email'];
9          $senha = $_POST['senha'];
10
11          $sql = "SELECT * FROM cadastro_usuario WHERE email = '$email' and senha = md5('$senha')";
12
13          $result = $conexao->query($sql);
14
15          if(mysqli_num_rows($result)<1){
16              unset($_SESSION['email']);
17              unset($_SESSION['senha']);
18
19              header('Location: entrar.php');
20          }
21
22          // Entrou no site
23          else{
24              $_SESSION['email'] = $email;
25              $_SESSION['senha'] = $senha;
26
27              header('Location: ../../index_principal.php');
28          }
29      }
30      else{
31          header('Location: entrar.php');
32      }
33
34  ?>
```

Figura 21- Imagem do código PHP verifica se o login foi confirmado, baseado no código do KIANTO.

GHELLAR, Camily do Nascimento, dos SANTOS, Alice Lima. **KIANTO - SITE PARA ORGANIZAÇÃO DE ESTUDOS**. Anais da XIV Mostra Nacional de Iniciação Científica e Tecnológica Interdisciplinar (MICTI), 2021. Disponível em: <https://bit.ly/3M1fhvu>. Acesso em: 01/06/2022.

COMPARATIVE STUDY ON PYTHON WEB FRAMEWORKS: FLASK AND DJANGO.

Utilizou-se como base seus registros sobre HTML, CSS e sua organização textual de estudo comparativo.

GHIMIRE, Devndra. **Comparative study on Python web frameworks: Flask and Django**. Metropolia University of Applied Sciences, 2020. Disponível em: <https://bit.ly/3NHMv48>. Acesso em: 01/06/2022.

NETFLIX

O layout da plataforma de *streaming* “Netflix” foi utilizado como base para a criação do front-end desenvolvido.

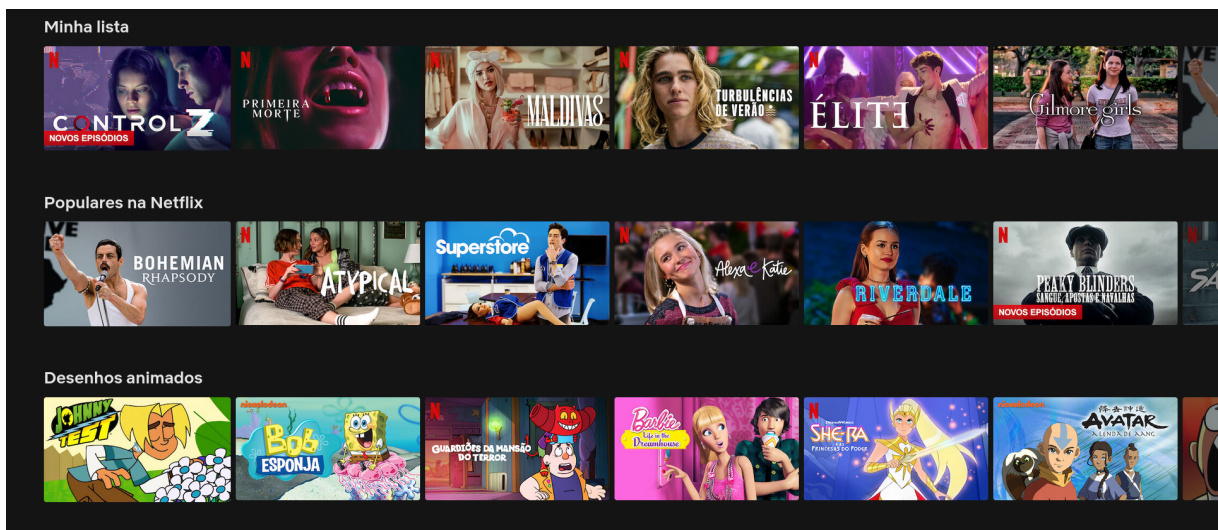


Figura 22- Imagem do layout da plataforma Netflix.

Netflix. Disponível em: <https://bit.ly/3B5S21F>. Acesso em: 06/07/2022.

ANÁLISE COMPARATIVA DE LINGUAGENS DE PROGRAMAÇÃO A PARTIR DE PROBLEMAS CLÁSSICOS DA COMPUTAÇÃO

Utilizou-se como base os métodos de avaliação -qualitativos e quantitativos- de desempenho das linguagens.

SEABRE, Rodrigo Duarte, DRUMMOND, Isabela Neves, GOMES, Fernando Coelho. **Análise Comparativa de Linguagens de Programação a partir de Problemas Clássicos da Computação.** Universidade Federal de Itajubá, Minas Gerais - Brasil, 2018. Disponível em: <https://bit.ly/3of5BEI>. Acesso em: 20/07/2022.

11- REFERÊNCIAS

- WebAIM. **web accessibility in mind**. Institute for Disability Research, Policy, and Practice. Disponível em: <https://bit.ly/3ainQET>. Acesso em: 01/06/2022.
- Edilson. **Tutorial: Menu lateral retrátil com CSS3 - N° 1 (sem javascript)**. Sala de Estudo, 2016. Disponível em: <https://bit.ly/3xar1aE>. Acesso em: 01/06/2022.
- OLIVEIRA, Anderson. **Como Dividir uma DIV em 2, 3, 4 ou mais colunas**. <BLOGSON/>: blog do prof. Anderson, 2019. Disponível em: <https://bit.ly/3x7jxoR>. Acesso em: 01/06/2022.
- NEITZKE, Gustavo. **Aprenda como ligar um FORMULÁRIO com BANCO DE DADOS #01**. Youtube, 30 mai. 2021. Disponível em: <https://bit.ly/3PUZiSK>. Acesso em: 01/06/2022.
- NEITZKE, Gustavo. **Aprenda como criar uma TELA de LOGIN com HTML e CSS**. Youtube, 11 fev. 2021. Disponível em: <https://bit.ly/3aBY2E4>. Acesso em: 01/06/2022.
- NEITZKE, Gustavo. **Como criar um SISTEMA DE CADASTRO com LOGIN, FORMULÁRIO e SESSÕES #02**. Youtube, 28 jun. 2021. Disponível em: <https://bit.ly/3aC14Iv>. Acesso em: 01/06/2022.
- NETTO, Hylson Vescovi. **Códigos Flask Python**, no GITHUB [hvescovi/dw2edfund/webap/monopy](https://github.com/hvescovi/dw2edfund/webap/monopy). Disponível em: <https://bit.ly/3usxqMU>. Acesso em: 06/07/2022.
- NETTO, Hylson Vescovi. **Códigos JavaScript**, no GITHUB [hvescovi/dw2edfund/webap/monopy](https://github.com/hvescovi/dw2edfund/webap/monopy). Disponível em: <https://bit.ly/3usxqMU>. Acesso em: 06/07/2022.
- CodeOfaNinja. **How To Create A Simple REST API in PHP? Step By Step Guide!** Disponível em: <https://bit.ly/3nFIK4n>. Acesso em: 06/07/2022.
- Mozilla. **JavaScript**. Disponível em: <https://mzl.la/3AvlpKs>. Acesso em: 06/07/2022.
- **BDS**. Disponível em: <https://bit.ly/3RQYBuC>. Acesso em: 20/07/2022.
- **Flask (framework web)**. Wikipedia, 2022. Disponível em: <https://bit.ly/3zlA6yx>. Acesso em: 20/07/2022.
- ANDRADE, Ana Paula de. **O que é Flask?** Disponível em: <https://bit.ly/3RRYWgE>. Acesso em: 20/07/2022.
- MD5 Hash Generator. Disponível em: <https://bit.ly/2RZmhkt>. Acesso em: 17/11/2022.
- FREDÁ, Anthony. **O que é e como funciona o algoritmo de hash MD5?** Avast, 2022. Disponível em: <https://bit.ly/3EDAMCb>. Acesso em: 17/11/2022.
- Christiano. **Criptografia MD5**. DEVMEDIA, 2007. Disponível em: <https://bit.ly/2DKsxUS>. Acesso em: 17/11/2022.