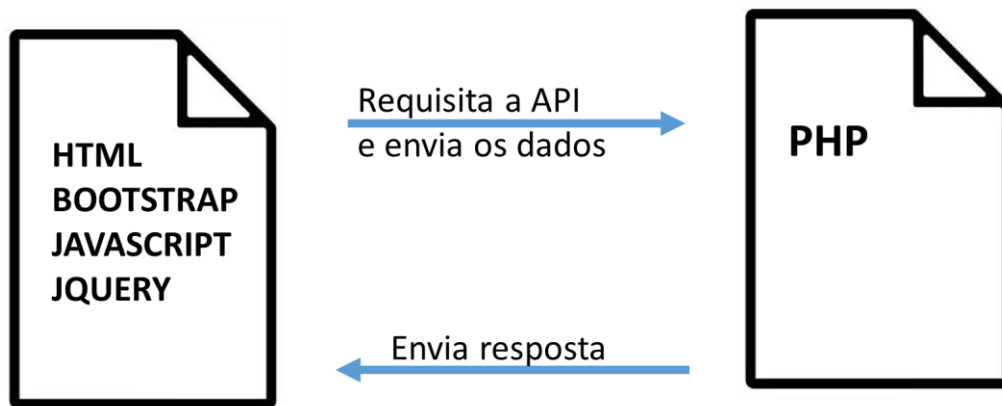


O objetivo deste trabalho é criar uma página de internet que será capaz de exibir a data e hora local, o IP do servidor e a quantidade de caracteres de uma frase qualquer digitada. A página html deverá consultar, via javascript, uma API criada em PHP que retornará os resultados desejados.



Simplificadamente, o HTML servirá para estruturar a página para o usuário (*front end*) por meio das TAGs e utilizando CSS para estilizar por meio da biblioteca (ou framework) do Bootstrap. Nesta mesma página conterá código Javascript bem como código JQuery que nada mais é uma biblioteca (ou framework) mais “amigável” para se trabalhar com javascript. O código jquery servirá para consumir (requisitar) a API (que desenvolveremos em PHP), para receber a resposta e exibir na tela.

PASSO A PASSO:

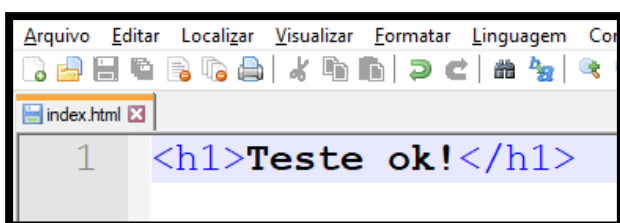
1) INFRAESTRUTURA: SIMULANDO A INTERNET LOCALMENTE

Para “rodar” o sistema, é necessário instalar um servidor de internet, o interpretador do PHP e o SGBD MySql (Maria DB) o qual utilizaremos em outra tarefa. Deste modo podemos abrir (ou rodar, executar...) o sistema em um navegador de internet local.

Como sugestão você pode baixar o WAMP – “A Windows Web development environment for Apache, MySQL, PHP databases” (<https://sourceforge.net/projects/wampserver/>). Ele instalará o servidor de internet Apache, o PHP e o MySql.

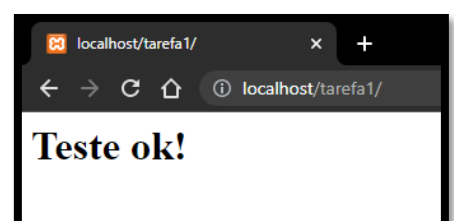
Após a instalação localize a pasta “www” dentro de “wamp64”.

É nesta pasta “www” que deverão estar os arquivos a serem exibidos no navegador (html ou php). Então, dentro desta pasta crie uma subpasta chamada “tarefa1” e nela crie o arquivo index.html contendo apenas a seguinte tag html:



Para editar os programas .html ou .php sugerimos o **Notepad++**, o **Visual Studio Code** ou outro editor que não seja o Bloco de Notas ou o Word.

Para rodar esta página, abra seu navegador (Google Chrome) e digite na barra de endereço “localhost/tarefa1”. Deverá aparecer a tela mostrada aqui à direita:



O nome “index.html” tem um motivo específico para se chamar assim, por padrão, o servidor de internet localizará a existência do arquivo “index.html” ou “index.php” e o “executará” na tela.

2) O que é uma API?

API - Application Programming Interface, de forma simplificada, é um programa que geralmente fica no servidor (back end), escrita em PHP, ASP, JAVA,... e que pode ser “chamada” por meio do front end por alguma linguagem compatível: javascript, react, java, ... afim de retornar alguma informação.

Veja este exemplo de API que pode ser solicitado via endereço de internet:

<https://viacep.com.br/ws/13201160/json/>

Esta API (ou Webservice de um modo geral) retorna informações do CEP contido no endereço, neste caso, o CEP da Fatec Jundiaí.

3) O Front End

Altere agora o código **index.html** para o seguinte conteúdo:

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>Tarefa 1</title>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css">
  </head>

  <body>
    <div class="container">

      <div class="h1 text-primary">Tarefa 1 - Teste de API</div>
      <hr>

      <div class="row">

        <div class="col-12">
          <label>Digite um texto qualquer:</label>
          <input type="text" id="um_texto_qualquer" class="form-control col-4">
        </div>

        <div class="col-12 mt-4">
          <div class="btn btn-lg btn-primary">Consultar API</div>
        </div>

        <div class="col-md-6 col-sm-12 mt-4">
          <div class="h4">Resultados:</div>
          <table class="table table-hover">
            <tr>
              <td>Data da consulta</td>
              <td><div id="div_resposta_data" class="badge badge-info"></div></td>
            </tr>
            <tr>
              <td>IP do servidor</td>
              <td><div id="div_resposta_ip" class="badge badge-info"></div></td>
            </tr>
            <tr>
              <td>Qtde de caracteres</td>
              <td><div id="div_resposta_qtde" class="badge badge-info"></div></td>
            </tr>
          </table>
        </div>

      </div> <!-- /row -->
    </div> <!-- /container -->
    <script src="https://code.jquery.com/jquery-3.6.0.slim.js"></script>
  </script>
</body>
</html>
```

A seguinte tela deverá surgir:

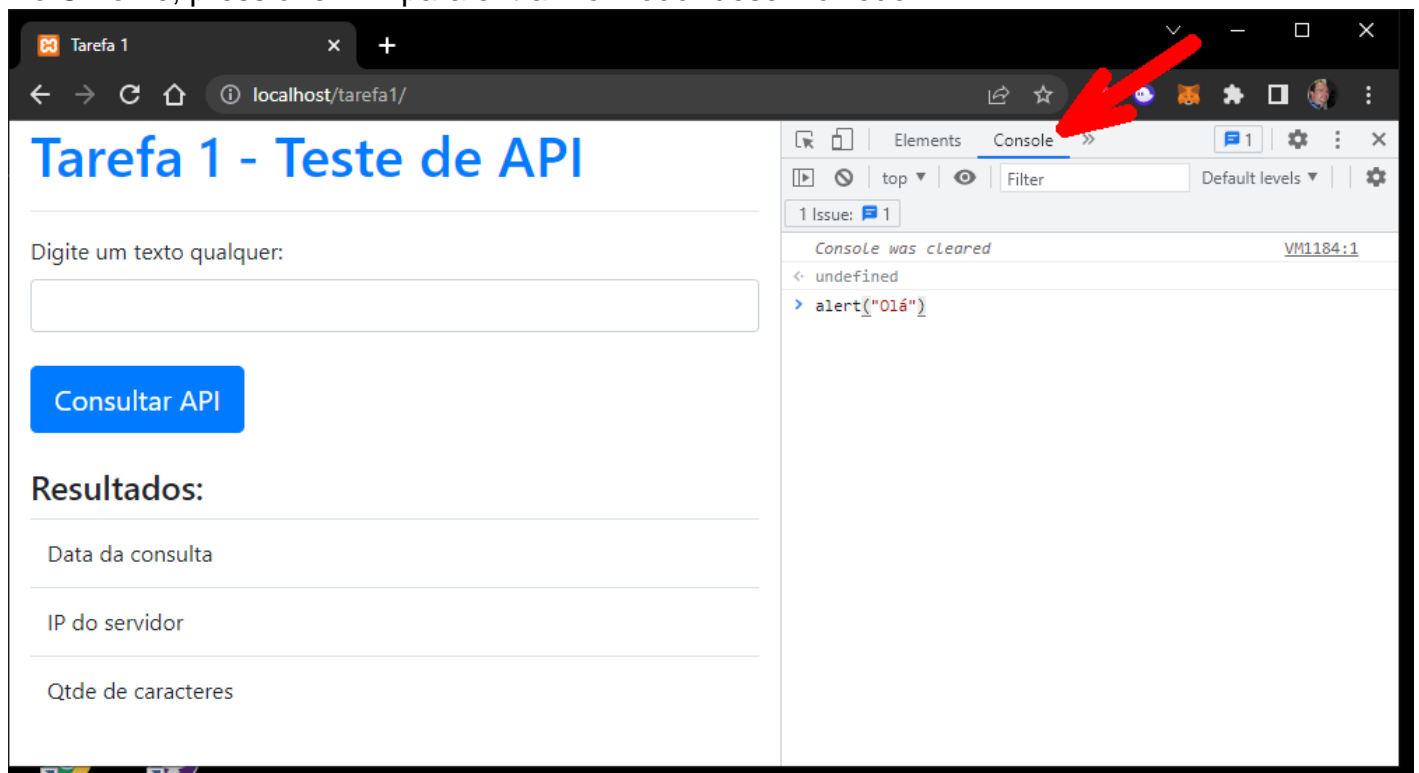


4) Testando o JQuery

O JQuery é uma biblioteca de funções amigáveis para interagir com javascript.

Vamos testar manualmente alguns comandos diretamente no navegador.

No Chrome, pressione F12 para entrar no modo “desenvolvedor”:



No console, teste digitando o comando javascript: alert("Olá") e dê enter.

Continue testando com os seguintes comandos:

`$("#um_texto_qualquer").val("Esta é a tarefa 1 com jQuery")`

O comando JQuery acima seleciona o elemento html cujo ID é "um_texto_qualquer" e atribui o valor descrito acima. Copie e cole no Console para testar.

O equivalente em javascript seria:

`document.getElementById("um_texto_qualquer").value = " Esta é a tarefa 1 com javascript"`

Ambos os comandos atribuem um valor à TAG INPUT cujo ID="um_texto_qualquer".

Teste com os seguintes comandos:

```
$("#div_resposta_data").html("01/04/2020");  
$("#div_resposta_ip").html ("111.222.333.444");  
$("#div_resposta_qtde").html ("123");
```

Como agora estamos atribuindo um valor a uma tag DIV não usamos mais o “val()” para tags do tipo INPUT e sim html().

Agora altere a seguinte linha de index.html

```
<div class="btn btn-lg btn-primary">Consultar API</div>
```

Para:

```
<div onClick="enviar_api()" class="btn btn-lg btn-primary">Consultar  
API</div>
```

O código em vermelho acima executará a função “enviar_api()” quando clicar no botão.

Entre as tags <script> e </script> acrescente o código:

```
<script>  
    function enviar_api() {  
        alert("teste");  
    }  
</script>
```

Atualize a página com F5 e teste clicar no botão.

Na próxima tarefa criaremos a API em PHP.