

Prova D3 – Desenvolvimento Web
João Gabriel Hespanhol – 11824219

1. Por meio do navegador _método get_ e frameworks como Apigee, Postman, Rest-Assured, Karate DSL, entre outros.
2. Ao usar a API fetch é provida uma interface JavaScript que vai acessar e manipular algumas partes do HTML, como requisições e respostas. Além disso, a API tem um método global que facilmente é capaz de buscar alguns recursos por meio da rede utilizando o XMLHttpRequest. O Fetch fornece um ambiente ímpar para definição de conceitos referentes ao protocolo HTTP, tais como CORS e extensões ao HTTP;
A API Fetch fornece uma interface JavaScript para acessar e manipular partes do pipeline HTTP, tais como os pedidos e respostas. Ela também fornece o método global fetch() que fornece uma maneira fácil e lógica para buscar recursos de forma assíncrona através da rede. Este tipo de funcionalidade era obtida anteriormente utilizando XMLHttpRequest. Fetch fornece uma alternativa melhor que pode ser facilmente utilizada por outras tecnologias como Service Workers. Fetch também provê um lugar lógico único para definir outros conceitos relacionados ao protocolo HTTP como CORS e extensões ao HTTP.
3. Utilizando as variações do método Get Element By (ID; Tag; Class; CSS Selectors; HTML Object Collections;) além de poder usar JQuery.
EX:

```
var myElement = document.getElementById("home");  
var x = document.getElementsByTagName("p");  
var x = document.getElementsByClassName("colunas");  
var x = document.querySelectorAll("p.home ")
```
4. Git hub
5. A- A rota é uma definição de destino para uma solicitação, como exemplo quando passamos um diretório(nome) e a aplicação, nessa rota consegue saber quem chamou, o que chamou e como chamou, por exemplo(GET, POST, PUT, etc).
B- As linguagens mais famosas utilizadas na construção do backend web são JavaScript e PHP.
C- Antes de se iniciar o projeto de construção de uma API, deve-se verificar se realmente sua construção é necessária. O primeiro ponto é que a API vai permitir que ao acessar os dados, o usuário não precisará baixar toda base de dados. Após isso, deve-se decidir se a API é pública ou privada. Posteriormente, deve-se permitir o contato com a base de dados. Ao divulgar a API, deve-se evitar as mudanças e na URL pois é o meio em que outros desenvolvedores utilizarão sua API e nunca transmitir dados sensíveis pelo url.
D – POST – Post é um método utilizado para a criação de recursos novos, contudo não é um método seguro pois altera o estado no servidor;
GET – Método utilizado para recuperar informações de um recurso, ou seja, é utilizado para efetuar a leitura de dados;

PUT – Método utilizado para atualizar ou substituir uma informação. Também não é considerado um método seguro pois muda o estado no servidor;

DELETE – Usado para excluir um recurso;

6. No PHP existe alguns termos denominados reservados ou constantes, o que traz informações sobre a execução em curso. Apesar de ser chama como constante mágica, essas constantes só são relevantes dentro do contexto que está sendo executada. Essas constantes são muito úteis quando é preciso obter alguns valores como arquivo roteiro que está em execução, a classe do objeto em questão e o diretório atual. São exemplos de constantes mágicas:

__LINE__ (linha de código)	__DIR__ (diretório do arquivo)	__CLASS__ (nome da classe)	__Method__ (nome do método)
Ex: 1 echo __LINE__; 2 3 4 echo __LINE__; 5 SAÍDA: 1 4	Ex: echo __DIR__ SAIDA: /www/prova/controller	Ex: class Pessoa{ private getCpf{ echo 'sua classe é __CLASS__'; \$pessoa = new pessoa (); \$pessoa->getCpf(); SAIDA: sua classe é Pessoa	Ex: class Pessoa{ private getCpf{ echo 'seu método é __METHOD__'; \$pessoa = new pessoa (); \$pessoa->getCpf(); SAIDA: seu método é getCpf

7- class superHeroi{

```
private $id;  
private $energy;  
private $name;  
private $poderEspecial;  
private $origem  
private $forca
```

```
public function superHeroi($id, $name, $poderEspecial,$energy,$forca,$origem)
```

```
{  
    $this->setId($id);  
    $this->setEnergy($energy);  
    $this->seName($name);  
    $this->setPoder($poderEspecial);  
    $this->setOrigem($origem);  
    $this->setForca($forca);  
}
```

```

public function getId()
{
    return $this->id;
}
public function setId($id)
{
    $this->id = $id;
}
public function setPoderEspecial($poderEspecial)
{
    $this->poderEspecial = $poderEspecial;
}
public function getPoderEspecial($poderEspecial)
{
    return $this->poderEspecial;
}
public function getName()
{
    return $this->name;
}
public function setName($name)
{
    $this->name = $name;
}
public function getEnergy()
{
    return $this->energy;
}
public function setEnergy($energy)
{
    $this->energy = $energy;
}

```

```

public function superHeroi() {
    echo superHeroi\n";
}
public function getForca()
{
    return $this->$forca;
}
public function setForca($forca)
{
    $this->forca = $forca;
}

```

```
public function CalcularpoderMedio()
{
    return $this->($energy * $forca)/2;
}
public function plusEnergy()
{
    $this->$energy= $energy * 11824219;
}
$Heroil = new SuperHeroi("21","goku"," super sayajin","30000","50000","Out Of
Space");
$Heroill = new SuperHeroi("25","harry potter","
wizard","60000","278000","Hogwarts");
$HeroiIII = new SuperHeroi("22","sirius black","
wizard","90000","35000","askaban");
$HeroiIV = new SuperHeroi("23","dubledore","
wizard","70000","34500","hogsmeade");
}
```