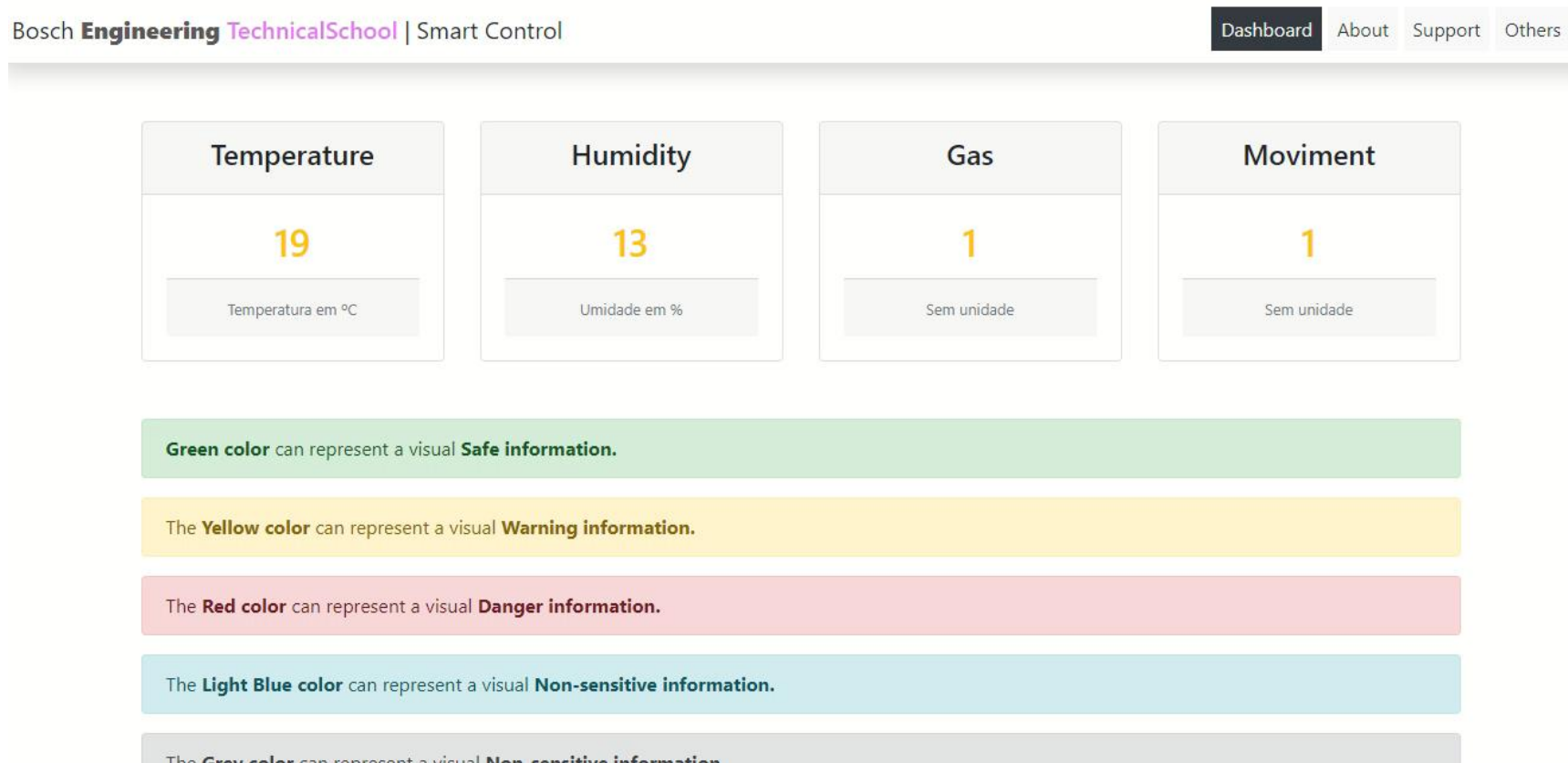


Módulo 4:

Desenvolvimento web

Aula 7

- ▶ Ao colocar o mouse sobre os cards e alertas, um efeito de sombra deve ser ativado, de forma que ressalte o elemento (use o **hover**).



Banco de Dados

Curso Web


phpMyAdmin

- ▶ Na nossa próxima etapa, para que a proposta do curso se torne cada vez mais palpável, vamos fazer a conexão da página (Cliente) para o Banco de dados, e para isso faremos o uso de um backend, que nada mais é a interface de programação que conectará nossa Dashboard ao banco de dados.
- ▶ Já ouvimos falar de banco de dados em outro momento do curso, estão já existe familiaridade com os termos.
- ▶ Com a proposta de fazer com que a interação com o banco seja mais simples, traremos um recurso que é o gerenciador de banco de dados. No passado fizemos o uso do [MySQL Workbench](#) porém agora vamos usar uma outra interface, que possui a mesma proposta, e seu nome é [phpMyAdmin](#).

Curso Web

phpMyAdmin

► Para acessar a plataforma web, faremos os seguintes passos:

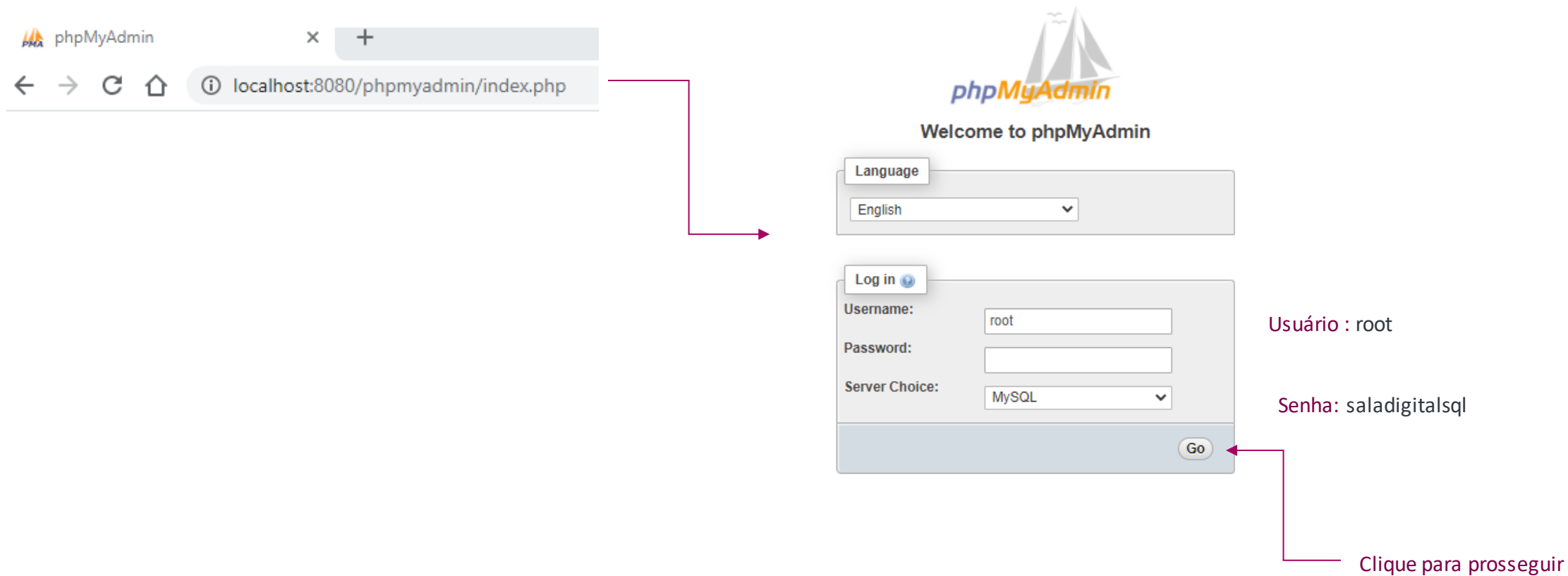
1. Primeiramente vamos iniciar o WAMPSERVER para que o servidor seja iniciado.
2. Verifiquei o ícone  para garantir que os serviços estão rodando.
3. Abra o browser do computador (Aba de internet)
4. Digite o seguinte endereço URL: localhost:8080/phpmyadmin

Em vermelho, destaco a porta de acesso ao portal, visto que alguns computadores podem estar configurados para diferentes portas, como por exemplo 8070, 8081.

Curso Web

phpMyAdmin

3. Nesta etapa você será direcionado a página inicial do gerenciador, a página de login.



Curso Web

phpMyAdmin

The screenshot displays the phpMyAdmin web interface in a browser window. The address bar shows the URL `localhost:8080/phpmyadmin/index.php`. The interface includes a top navigation bar with tabs for Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables, Charsets, Engines, and Plugins. The left sidebar shows the 'Current server: MySQL' dropdown and a tree view of databases including 'dashboard', 'energyplatform', 'information_schema', 'mysql', 'performance_schema', and 'sys'. The main content area is titled 'General settings' and contains sections for 'General settings' and 'Appearance settings'. The 'General settings' section includes a 'Change password' link and a 'Server connection collation' dropdown set to 'utf8mb4_unicode_ci'. The 'Appearance settings' section includes a 'Language' dropdown set to 'English', a 'Theme' dropdown set to 'pmahomme', and a 'Font size' dropdown set to '82%'. A 'More settings' link is also present. On the right side, there are three informational panels: 'Database server' listing server details like version (5.7.26) and user (root@localhost); 'Web server' listing Apache/PHP details; and 'phpMyAdmin' listing version (4.8.5) and links to documentation and support.

localhost:8080 / MySQL | phpMyAdmin

localhost:8080/phpmyadmin/index.php

Server: MySQL:3306

Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins

phpMyAdmin

Current server: MySQL

Recent Favorites

New

- dashboard
- energyplatform
- information_schema
- mysql
- performance_schema
- sys

General settings

Change password

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Language: English

Theme: pmahomme

Font size: 82%

More settings

Database server

- Server: MySQL (127.0.0.1 via TCP/IP)
- Server type: MySQL
- Server connection: SSL is not being used
- Server version: 5.7.26 - MySQL Community Server (GPL)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.39 (Win64) PHP/7.2.18
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 3591daad22de08524295e1bd073aceff11e6579 \$
- PHP extension: mysqli curl mbstring
- PHP version: 7.2.18

phpMyAdmin

- Version information: 4.8.5
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

Curso Web

phpMyAdmin

The screenshot displays the phpMyAdmin interface in a web browser. The browser's address bar shows the URL `localhost:8080/phpmyadmin/index.php`. The interface includes a top navigation bar with tabs for **Databases**, **SQL**, **Status**, **User accounts**, **Export**, **Import**, **Settings**, **Replication**, **Variables**, **Charsets**, **Engines**, and **Plugins**. The **Databases** and **SQL** tabs are highlighted with a red box. On the left sidebar, under the **Recent** tab, a tree view shows databases: **dashboard**, **energyplatform**, **information_schema**, **mysql**, **performance_schema**, and **sys**. The **New** button and the **sys** database are highlighted with a red box. The main content area is divided into two sections: **General settings** and **Appearance settings**. The **General settings** section includes a **Change password** link and a **Server connection collation** dropdown set to `utf8mb4_unicode_ci`. The **Appearance settings** section includes a **Language** dropdown set to `English`, a **Theme** dropdown set to `pmahomme`, and a **Font size** dropdown set to `82%`. A **More settings** link is also present. The right sidebar contains three panels: **Database server**, **Web server**, and **phpMyAdmin**. The **Database server** panel lists details such as `Server: MySQL (127.0.0.1 via TCP/IP)`, `Server type: MySQL`, `Server connection: SSL is not being used`, `Server version: 5.7.26 - MySQL Community Server (GPL)`, `Protocol version: 10`, `User: root@localhost`, and `Server charset: UTF-8 Unicode (utf8)`. The **Web server** panel lists details such as `Apache/2.4.39 (Win64) PHP/7.2.18`, `Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - $Id: 3591daad22de08524295e1bd073aceeff11e6579 $`, `PHP extension: mysqli curl mbstring`, and `PHP version: 7.2.18`. The **phpMyAdmin** panel lists details such as `Version information: 4.8.5`, `Documentation`, `Official Homepage`, `Contribute`, `Get support`, `List of changes`, and `License`.

Curso Web

phpMyAdmin

localhost:8080 / MySQL | phpMyAdmin

localhost:8080/phpmyadmin/index.php

Server: MySQL:3306

Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins

Current server: MySQL

Recent Favorites

New

- dashboard
- energyplatform
- information_schema
- mysql
- performance_schema
- sys

General settings

- Change password
- Server connection collation: utf8mb4_unicode_ci

Appearance settings

- Language: English
- Theme: pmahomme
- Font size: 82%
- More settings

Database server

- Server: MySQL (127.0.0.1 via TCP/IP)
- Server type: MySQL
- Server connection: SSL is not being used
- Server version: 5.7.26 - MySQL Community Server (GPL)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.39 (Win64) PHP/7.2.18
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 3591daad22de08524295e1bd073aceff11e6579 \$
- PHP extension: mysqli curl mbstring
- PHP version: 7.2.18

phpMyAdmin

- Version information: 4.8.5
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

Nesta sessão temos a possibilidade de visualizar os bancos de dados existentes em nosso servidor, bem como criar novos bancos de dados clicando no botão New

Curso Web

phpMyAdmin

1

2

3

Server: MySQL:3306

Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins

Databases

Create database

Database name Collation Action

<input type="checkbox"/>	dashboard	latin1_swedish_ci	Check privileges
<input type="checkbox"/>	energyplatform	latin1_swedish_ci	Check privileges
<input type="checkbox"/>	information_schema	utf8_general_ci	Check privileges
<input type="checkbox"/>	mysql	latin1_swedish_ci	Check privileges
<input type="checkbox"/>	performance_schema	utf8_general_ci	Check privileges
<input type="checkbox"/>	sys	utf8_general_ci	Check privileges
Total: 6		latin1_swedish_ci	

☐ Check all With selected: [Drop](#)

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

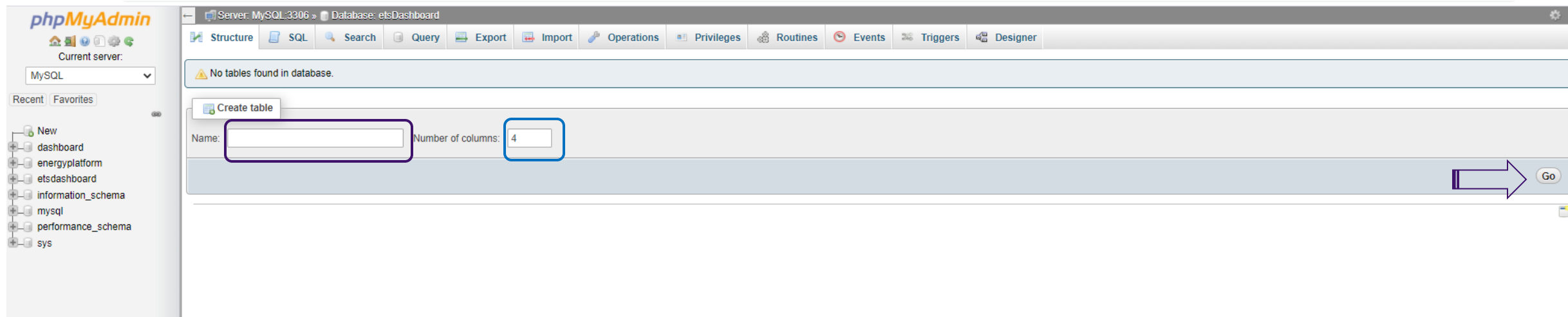
- Enable statistics

Observe que no item 2, é justamente onde iremos inserir o nome de nosso banco de dados, no meu caso o nome é **dashboard**

Curso Web

phpMyAdmin

Nosso banco de dados já foi criado, então é hora de criar nossa tabela para organizarmos os dados então precisamos dar um nome para esta tabela, no meu caso **dados_arduino**



Para a Proposta do Projeto, teremos 7 colunas, portanto altere o *“Number of columns”* para **7**

Curso Web

phpMyAdmin

phpMyAdmin

Current server: MySQL

Recent Favorites

- New
- dashboard
- energyplatform
- etsdashboard
- information_schema
- mysql
- performance_schema
- sys

Server: MySQL:3306 » Database: etsDashboard » Table: Dashboard

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table name: Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments	Virtuality	Move column
	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>			
	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>			
	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>			
	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>			

Table comments: Collation: Storage Engine: MyISAM

PARTITION definition:

Partition by: (Expression or column list)

Partitions:

Preview SQL Save

Hora de dar as colunas seus respectivos nomes, teremos as 7 delas sendo: id ,id_arduino,movimento, temperatura, umidade,gas, data_hora

← → ↻ 🏠 localhost:8080/phpmyadmin/server_databases.php?server=1 🔍 ☆ 🌐

phpMyAdmin

Current server: MySQL

Recent Favorites

- New
- dashboard
- New
- dados_arduino
 - Columns
 - New
 - data_hora
 - gas
 - id
 - id_arduino
 - movimento
 - temperatura
 - umidade
 - Indexes
 - energyplatform
 - etsdashboard
 - information_schema
 - mysql
 - performance_schema
 - sys

Server: MySQL:3306 Database: etsDashboard Table: DashBoard

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table name: Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I.	Comments	Virtuality	Move column
id_arduino	INT	4	None			<input type="checkbox"/>	---	<input type="checkbox"/>			
temperatura	INT	4	None			<input type="checkbox"/>	---	<input type="checkbox"/>			
movimento	BOOLEAN		None			<input type="checkbox"/>	---	<input type="checkbox"/>			
umidade	INT	4	None			<input type="checkbox"/>	---	<input type="checkbox"/>			
gas	BOOLEAN		None			<input type="checkbox"/>	---	<input type="checkbox"/>			
id	INT	50	None			<input type="checkbox"/>	PRIMARY	PRIMARY <input checked="" type="checkbox"/>			
data_hora	TIMESTAMP		None			<input type="checkbox"/>	---	<input type="checkbox"/>			

Table comments: Collation: Storage Engine: MyISAM

PARTITION definition: (Expression or column list)

Partitions:

Preview SQL Save

Curso Web phpMyAdmin

localhost:8080 / MySQL / etsDash

localhost:8080/phpmyadmin/server_databases.php?server=1

phpMyAdmin

Current server: MySQL

Recent Favorites

New dashboard
New dados_arduino
Columns
New data_hora
gas
id
id_arduino
movimento
temperatura
umidade

Indexes
energyplatform
etsdashboard
information_schema
mysql
performance_schema
sys

Table name: Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_1	Comments	Virtuality	Move column
id_arduino	INT	4	None			<input type="checkbox"/>	---	<input type="checkbox"/>			
temperatura	INT	4	None			<input type="checkbox"/>	---	<input type="checkbox"/>			
movimento	BOOLEAN		None			<input type="checkbox"/>	---	<input type="checkbox"/>			
umidade	INT	4	None			<input type="checkbox"/>	---	<input type="checkbox"/>			
gas	BOOLEAN		None			<input type="checkbox"/>	---	<input type="checkbox"/>			
id	INT	50	None			<input type="checkbox"/>	---	<input checked="" type="checkbox"/>			
data_hora	TIMESTAMP		None			<input type="checkbox"/>	---	<input type="checkbox"/>			

Table comments: Collation:

PARTITION definition:

Partition by: (Expression or column list)

Partitions:

Add index

Index name: PRIMARY

Index choice: PRIMARY

+ Advanced Options

Column	Size
id [int]	

Go Cancel

Preview SQL Save

Curso Web

phpMyAdmin

Current server: MySQL

Recent Favorites

New

dashboard

New

dados_arduino

Columns

New

data_hora

gas

id

id_arduino

movimento

temperatura

umidade

Indexes

energyplatform

etsdashboard

information_schema

mysql

performance_schema

sys

Server: MySQL:3306 » Database: etsDashboard » Table: dados_arduino

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Triggers

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id_arduino	int(4)		No	None			Change Drop More
<input type="checkbox"/>	2	temperatura	int(4)		No	None			Change Drop More
<input type="checkbox"/>	3	movimento	tinyint(1)		No	None			Change Drop More
<input type="checkbox"/>	4	umidade	int(4)		No	None			Change Drop More
<input type="checkbox"/>	5	gas	tinyint(1)		No	None			Change Drop More
<input type="checkbox"/>	6	id	int(50)		No	None	AUTO_INCREMENT		Change Drop More
<input type="checkbox"/>	7	data_hora	timestamp		No	None			Change Drop More

☐ Check all With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#)

[Print](#) [Propose table structure](#) [Move columns](#) [Normalize](#)

[Add](#) column(s) [after data_hora](#) [Go](#)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	0	A	No	

Create an index on columns [Go](#)

Partitions

No partitioning defined!

[Partition table](#)

Information

Space usage		Row statistics	
Data	0 B		
Index	0 B		
Overhead			
Effective	0 B		
Total	0 B		

Módulo 4:

Backend - PHP

- ▶ PHP é uma linguagem interpretada livre de script, usada originalmente para aplicações atuantes do lado do servidor.
- ▶ Pode ser inserida facilmente em documentos HTML através das instruções “<?php” e “?>”.
- ▶ O que distingue o PHP de algo como o JavaScript no lado do cliente é que o código é executado no servidor, gerando o HTML que é então enviado para o navegador. O navegador recebe os resultados da execução desse script, mas não sabe qual era o código fonte.
- ▶ Para realizar a conexão com o banco de dados, vamos criar um arquivo chamado [app.php](#) e dentro dele montar 3 classes:
- ▶ A primeira classe define as variáveis que serão utilizadas dashboard.

```
<?php
//Classe Dashboard
class Dashboard{
    public $temperature;
    public $humidity;
    public $gas;
    public $moviment;

    public function __get($atributo){
        return $this->$atributo;
    }

    public function __set($atributo, $valor){
        $this->$atributo = $valor;
        return $this;
    }
}
```

- ▶ A segunda classe configura as variáveis de conexão com o banco de dados e define a função para conectar no banco.
- ▶ Não se esqueça de modificar o **host** e o **dbname** conforme você configurou anteriormente.

```
class Conexao{
    private $host = 'localhost:3309';
    private $dbname = 'ets_db3';
    private $user = 'root';
    private $pass = '';

    public function conectar(){
        try{
            $conexao = new PDO(
                "mysql:host=$this->host;dbname=$this->dbname",
                "$this->user",
                "$this->pass"
            );
            $conexao->exec("set charset set utf8");
            return $conexao;
        } catch (PDOException $e){
            throw new PDOException($e);

            //echo é como um print
        }
    }
}
```

- ▶ Para a terceira classe, ela terá o intuito de definir a maneira de conexão com o banco de dados e define a função para recuperar dados armazenados no banco.

```
class Bd{
    private $conexao;
    private $dashboard;

    public function __construct(Conexao $conexao, Dashboard $dashboard){
        $this->conexao=$conexao->conectar();
        $this->dashboard=$dashboard;
    }

    public function getDados(){
        $query = 'SELECT dados_arduino.temperatura, dados_arduino.umidade, dados_arduino.gas,
        dados_arduino.movimento FROM dados_arduino WHERE id_arduino=1';

        $stmt = $this->conexao->prepare($query);
        //$stmt->bindValue();
        $stmt->execute();
        return $stmt->fetch(PDO::FETCH_OBJ);
    }
}
```

- Por fim, definimos as variáveis que serão instâncias das classes definidas e realizarão a conexão com o banco de dados.

```
$dashboard = new Dashboard();  
  
$conexao = new Conexao();  
  
$bd = new Bd($conexao, $dashboard);  
  
echo json_encode($bd->getDados());  
  
?>
```


Módulo 4:

JavaScript

Curso Web

JavaScript

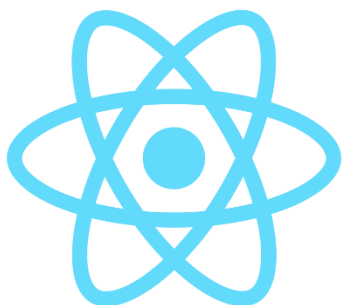
- ▶ JavaScript (frequentemente abreviado como JS) é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma (protótipos, orientado a objeto, imperativo e, funcional). Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web. JavaScript permite páginas da Web interativas e, portanto, é uma parte essencial dos aplicativos da web. A grande maioria dos sites usa, e todos os principais navegadores têm um mecanismo JavaScript dedicado para executá-lo.



Curso Web

JavaScript

- ▶ JavaScript, é uma linguagem que possui extensa abrangência, pode ser aplicado na parte cliente-side, ou seja no próprio browser, a qual será nossa maior aplicação no curso, em desenvolvimento server-side (Servidor) e até mesmo em aplicações Desktop
- ▶ Vale lembrar inclusive que JavaScript é base de diversas ferramentas de desenvolvimento, como por exemplo:



Curso Web

JavaScript

Porém vamos reforçar que apesar de semelhança de nomes, JavaScript **não** é o também famoso Java.

Algumas diferenças:

- ▶ Java é uma linguagem de programação OOP, ao passo que Java Script é uma linguagem de scripts OOP.
- ▶ Java cria aplicações executadas em uma máquina virtual ou em um browser, ao passo que o código JavaScript é executado apenas em um browser.
- ▶ O código Java precisa ser compilado, ao passo que os códigos JavaScript estão totalmente em texto

Curso Web

JavaScript

- ▶ Para iniciar o contato com o JavaScript, vamos aprender como acoplá-lo em nossa página HTML. Para sua perfeita execução, é necessário informar a página onde existe conteúdo JavaScript e o método é o seguinte:

```
<script type="text/javascript">  
    //Conteúdo JavaScript  
</script>
```

Toda a codificação interna às tag <script> serão interpretadas como JavaScript. Recomenda-se não colocar códigos JavaScript no meio do conteúdo HTML, para manter a organização do documento. Fazemos a inclusão antes da fechamento da tag </body> para manter o sistema de ordem de precedência

Curso Web

JavaScript

- ▶ Declaração de variáveis.
- ▶ Em Java Script, ou JS, como você preferir, a declaração de variáveis é feita da seguinte forma:
`var nome`
`var _idade`
`var endereco`
- ▶ Outro ponto bastante importante é ressaltar que o Javascript é uma linguagem do tipo Case Sensitive ou seja, faz diferenciação entre variáveis maiúscula e minúscula.

▶ `var Nome` \neq `var nome`

Curso Web

JavaScript

- ▶ Para colocar na prática, vamos criar um documento na pasta do projeto chamado “varJavaScript.html”
- ▶ E vamos então escrever as seguintes instruções:



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JavaScript</title>
</head>
<body>
  <h1>Variáveis</h1>
</body>
</html>

<script type="text/javascript">
  //string
  var texto = 'bosch'
  //numeros
  var numerosInteiro = 1234
  var numeroFracionarios = 123.45
  //booleanos
  var booleanoTrue = true
  var booleanoFalse = false

</script>
```

Curso Web

JavaScript

- ▶ É muito provável que vocês tenham observado que na declaração das variáveis, nós usamos a palavra reservada `Var` para executar a declaração e não especificamos o tipo de variável a qual ela se destina. Isso acontece pois o JavaScript possui uma característica de tipagem dinâmica, que nada mais é do que a habilidade da linguagem de observar o valor que está sendo atribuído à variável e estipular o tipo dela.

Resumindo:

Se você atribuir `var numero = 10`, a linguagem vai observar que para a variável `numero` foi atribuído um valor inteiro, portanto, o tipo da variável será inteiro. Observe que quem definiu isso foi a própria linguagem

Curso Web

JavaScript

- ▶ Agora, abra sua página “varJavaScript.html” com o Google Chrome, e veja o que acontece.
- ▶ Note que as atribuições aconteceram, mas nada foi mostrado na tela. Isso se dá ao fato de que as atribuições apesar de terem sido feitas, as respectivas variáveis não foram utilizadas em lugar algum, mas pelo menos foram carregadas pelo navegador.
- ▶ Para que possamos ver o valor dessas variáveis, vamos fazer o uso de um recurso bastante interessante do JavaScript que é a função nativa **alert()**
- ▶ Essa função faz com que uma caixa de diálogo apareça na tela com um botão de confirmação, essa função é normalmente usada para alertar o usuário de algum evento.

Curso Web

JavaScript

- ▶ Agora, abra sua página “varJavaScript.html” com o Google Chrome, e veja o que acontece.
- ▶ Note que as atribuições aconteceram, mas nada foi mostrado na tela. Isso se dá ao fato de que as atribuições apesar de terem sido feitas, as respectivas variáveis não foram utilizadas em lugar algum, mas pelo menos foram carregadas pelo navegador.
- ▶ Para que possamos ver o valor dessas variáveis, vamos fazer o uso de um recurso bastante interessante do JavaScript que é a função nativa **alert()**
- ▶ Essa função faz com que uma caixa de diálogo apareça na tela com um botão de confirmação, essa função é normalmente usada para alertar o usuário de algum evento.
- ▶ Dentro do nosso <script> vamos chamar a função alert()

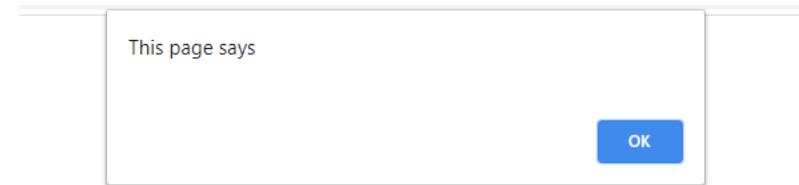
Curso Web

JavaScript

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JavaScript</title>
</head>
<body>
  <h1>Variáveis</h1>
</body>
</html>

<script type="text/javascript">
  //string
  var texto = 'bosch'
  //numeros
  var numerosInteiro = 1234
  var numeroFracionarios = 123.45
  //booleanos
  var booleanoTrue = true
  var booleanoFalse = false

  alert()
</script>
```

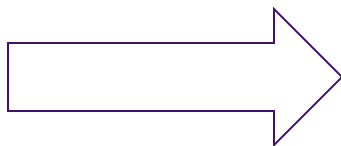


Curso Web

JavaScript

- ▶ Note que Logo que você abriu o browser, antes mesmo de aparecer a palavra “Variáveis”, uma caixa de diálogo apareceu, isso se deve ao fato do JavaScript carregar logo que o browser inicia, carregando todas as variáveis.
- ▶ Observe também que na caixa de diálogo, não existe informação, temos apenas uma caixa em branco com um botão de confirmação.
- ▶ Nossa próxima etapa é colocar dentro desta caixa de diálogo, as informações de nossas variáveis para que possamos vê-las em nossa interface.

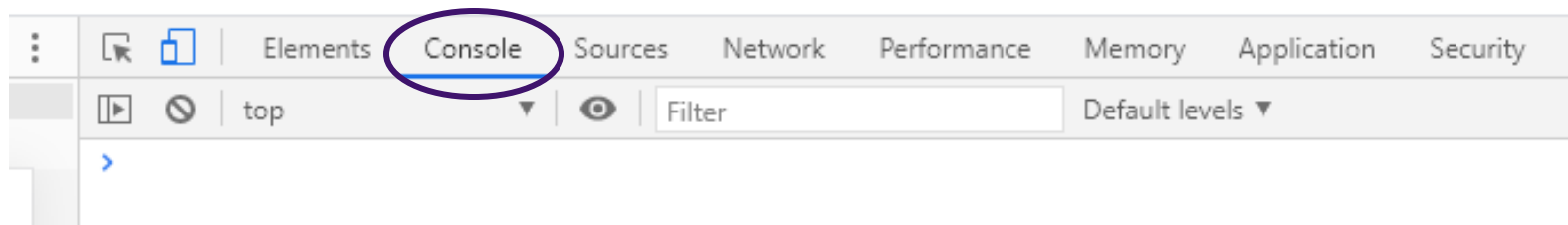
```
<script type="text/javascript">  
  //string  
  var texto = 'bosch'  
  //numeros  
  var numerosInteiro = 1234  
  var numeroFracionarios = 123.45  
  //booleanos  
  var booleanoTrue = true  
  var booleanoFalse = false  
  alert(numeroFracionarios)  
</script>
```



Curso Web

JavaScript

- ▶ Faça o mesmo procedimento para as demais variáveis, ou seja, crie alertas para todas as variáveis declaradas para exercitar este conceito.
- ▶ Após a execução vamos conhecer uma outra função do JavaScript, a qual usamos de forma recorrente, que é a `console.log()`. Essa função permite que os valores passados como argumento para essa função sejam mostrados no console do browser.
- ▶ O Console do browser é uma ferramenta bastante usada para debug e verificação de execução, podemos acessar essa ferramenta através do atalho `ctrl + shift + i`



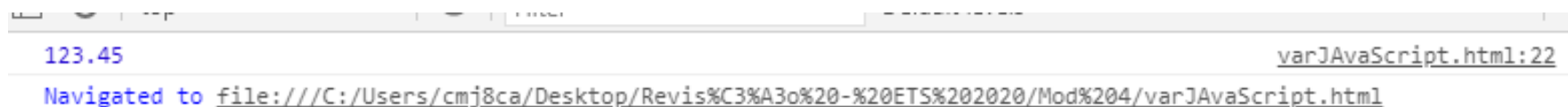
Curso Web

JavaScript

- Agora, no lugar do alert(), vamos colocar o console.log() e passar a variável como parâmetro.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JavaScript</title>
</head>
<body>
  <h1>Variáveis</h1>
</body>
</html>

<script type="text/javascript">
  //string
  var texto = 'bosch'
  //numeros
  var numerosInteiro = 1234
  var numeroFracionarios = 123.45
  //booleanos
  var booleanoTrue = true
  var booleanoFalse = false
  console.log(numeroFracionarios)
</script>
```



123.45 varJavaScript.html:22
Navigated to file:///C:/Users/cmj8ca/Desktop/Revis%C3%A3o%20-%20ETS%202020/Mod%204/varJavaScript.html

Crie console para as outras variáveis e verifique a ocorrência no console do browser.

Curso Web

JavaScript

► Funções:

No JavaScript, assim como em muitas outras linguagens, temos a possibilidade de criar funções para executar tarefas definidas.

► A sintaxe para este elabora é dada da seguinte forma:

```
Function nomeDaFunção (Parâmetros){
```

Lógica da função

Return valorASerRetornado

```
}
```

Return e parâmetros são campos opcionais, dependendo da aplicação a qual a função está submetida.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Função JS</title>
</head>
<body>

</body>
</html>

<script type="text/javascript">
  function calculaNotas(mensal1, mensal2){

    notaBimestral = (mensal1 + mensal2)/2

    return notaBimestral

  }
</script>
```

Curso Web

JavaScript - Exercício

- ▶ Como já aprendemos a estrutura de uma função, agora é hora de você criar sua própria função, a qual irá retornar alguma valor esperado. Em seguida fazer uma chamada desta função com a proposta de utilizá-la da mesma forma como fizemos a chamada da função `alert()`, faça a chamada da função e o seu retorno deverá ser mostrado no console do navegador.
- ▶ Tempo: 10 ~15 min

Curso Web

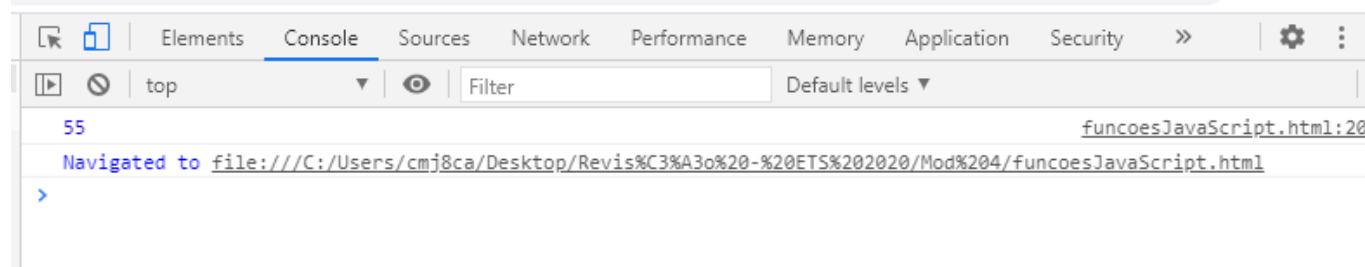
JavaScript - Resolução

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Função JS</title>
</head>
<body>

</body>
</html>

<script type="text/javascript">
  function calculaNotas(mensal1, mensal2){
    notaBimestral = (mensal1 + mensal2)/2
    return notaBimestral
  }

  console.log(calculaNotas(10,100))
</script>
```



Curso Web

JavaScript

- ▶ Para deixar a aplicação ainda mais realista, vamos fazer o seguinte:

Crie um botão em sua página usando os recursos do bootstrap:

<https://getbootstrap.com/docs/4.5/getting-started/introduction/#starter-template>

Crie um arquivo chamado eventosJavaScript.html e cole o template dentro do arquivo, em seguida salve!

- ▶ Dentro do body, insira um botão:

<https://getbootstrap.com/docs/4.5/components/buttons/#outline-buttons>

Curso Web

JavaScript

- A partir de agora você deverá ter uma página semelhante a esta:

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

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="
sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpsSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <button type="button" class="btn btn-outline-danger">Danger</button>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/
zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="
sha384-9/reFTGAW83EW2RDu2S0VKA1Zap3H66lZ81PoY1FhbgGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="
sha384-B4gt1jrGC7Jh4AgTPSdu0Bvf08shuf57BaghqFfPLYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>
  </body>
</html>
```

Curso Web

JavaScript

- Nossa tarefa agora, é fazer com que uma função do JavaScript seja acionada toda vez que o botão foi clicado.
Podemos usar a mesma função feita na exercício anterior para a execução desta atividade.

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

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JbzKZLeqZ/CBS87/COS3vgySeIQCzaySeHn1e67WA2ZlwSqYui8P5T9943a0Ry6Z/" crossorigin="anonymous">

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <button type="button" class="btn btn-outline-danger">Danger</button>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-JbzKZLeqZ/CBS87/COS3vgySeIQCzaySeHn1e67WA2ZlwSqYui8P5T9943a0Ry6Z/" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/re/VGIC+b364oA00l6VKgmqbw4deAOhgffxAapVdQWvSxDkd85CxvPh+VwhGV9ce/unwUHdu9-qxjW3wY8cR+Zl8vOz3Z/Ox87NEqZ" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-91C66o7MKqJf6aY9860Bg28ZA6D8gFL9fbdH+YVj2RhEHUiaRPwVOf4q84pj/z41R8q64rWqX4a6R/KIhzo" crossorigin="anonymous"></script>

    <script type="text/javascript">
      function calculaNotas(mensal1, mensal2){

        notaBimestral = (mensal1 + mensal2)/2

        return notaBimestral

      }

      console.log(calculaNotas(10,100))
    </script>
  </body>
</html>
```

Curso Web

JavaScript

- ▶ Com a página pronta e com o JavaScript já inserido no contexto, será necessário fazer com que a cada click no botão essa função seja executada. No HTML temos um recurso chamado de **Eventos**. Podemos ter eventos de click, os quais são acionados através do **onclick** do próprio HTML. Este evento é acionado toda vez que o usuário clica no elemento que o contém.



```
<button onclick="myFunction()">Click me</button>
```

- ▶ Nesse contexto, toda vez que o botão for clicando, a função `myFunction()` será acionada.

Então vamos inserir o evento de clique em nosso botão pra que a função seja executada a cada evento.

Curso Web JavaScript

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

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-..."
    anonymous">

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <button onclick="console.log(calculaNotas(10,100))" type="button" class="btn btn-outline-danger">Danger</button>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-...">
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-...">
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-...">
  </body>
</html>

<script type="text/javascript">
  function calculaNotas(mensal1, mensal2){

    notaBimestral = (mensal1 + mensal2)/2

    return notaBimestral
  }
</script>
```

Curso Web

Eventos de temporização do JavaScript

- ▶ O Javascript pode ser executado em intervalos de tempo. Isso é chamado de eventos de temporização.
- ▶ Os dois métodos principais a serem usados são:
 - ▶ `setTimeout(função, milissegundos)`- Executa uma função, depois de aguardar um número específico de tempo.
 - ▶ `window.setTimeout(função, milissegundos)`;
 - ▶ Pode ser gravado sem o prefixo `window`.
 - ▶ `setInterval(função, milissegundos)` – o mesmo que `setTimeout()`, mas repete a execução da função continuamente.
- ▶ Crie um nome programa HTML e faça com que ao clicar no botão espera por 5 segundos e a página cria um alerta.

```
<body>

<button class="btn btn-outline-dark mt-4 ml-4" onclick="setTimeout(myfunction, 5000)"> Clique aqui</button>

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKA1Zap3H66lZ81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPSdUt0Bvf08shuf57BaghqFfPLYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>
</body>

<script type="text/javascript">

    function myfunction(){
        alert('ALERTA!!!!!!');
    }
</script>
```


Curso Web

Eventos de temporização do JavaScript

- ▶ O resultado deve ser:

This page says
Alerta!!!!

OK

- ▶ Para parar a execução devemos usar o método **clearTimeout()** interrompe a execução da função especificada em **setTimeout()**.
 - ▶ **window.clearTimeout(variável)**
- ▶ Este método usa a variável retornada em **setTimeout()**:
 - ▶ **minhaVariável: setTimeout(função, milissegundos)** ;
 - ▶ **clearTimeout(minhaVariável);**
 - ▶ Se a função não tiver sido executada, você pode interromper a execução chamando o método **clearTimeout()**.

Curso Web

Eventos de temporização do JavaScript

- Crie um botão para que realize este clearTimeout:

```
<body>

<button class="btn btn-outline-dark mt-4 ml-4" onclick= "myvar = setTimeout(myfunction, 5000)"> Clique aqui</button>
<button class="btn btn-outline-danger mt-4 ml-4" onclick="clearTimeout(myvar)"> Cancelar</button>

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFV" >
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RD" >
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4A" >
  >
</body>

<script type="text/javascript">

  function myfunction(){
    alert('ALERTA!!!!!!');
  }
</script>
```

Curso Web

Eventos de temporização do JavaScript

- ▶ O método de `setInterval()` – repete uma determinada função a cada intervalo de tempo
 - ▶ **`Window.setInterval(função, milissegundos);`**

Clique aqui

```
<body>

<button class="btn btn-outline-dark mt-4 ml-4" onclick="setInterval(myfunction, 5000)"> Clique aqui</button>

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy40" >
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTG" >
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1" >
  >
</body>

<script type="text/javascript">

  function myfunction(){
    alert('ALERTA!!!!!!');
  }
</script>
```

Treinamento IoT