

FUNDAÇÃO VALEPARAIBANA DE ENSINO
COLÉGIOS UNIVAP – UNIDADE CENTRO

CURSO TÉCNICO EM INFORMÁTICA

RODRIGO RODA OLIVETO ALVES

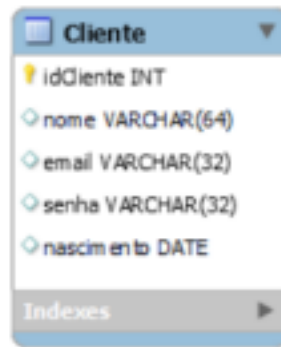
LISTA DE EXERCÍCIOS 3º BIMESTRE
PROGRAMAÇÃO AVANÇADA PARA WEB

Lista apresentada ao Curso Técnico de informática como
composição de nota.

Prof. Me. Hélio Lourenço Esperidião Ferreira

SÃO JOSÉ DOS CAMPOS
2024
PROGRAMAÇÃO AVANÇADA PARA WEB
LISTA DE EXERCÍCIOS III – PHP BÁSICO

1. Crie uma api completa no padrão REST API da tabela abaixo:



Cliente	
idCliente	INT
nome	VARCHAR(64)
email	VARCHAR(32)
senha	VARCHAR(32)
nascimento	DATE

Indexes

- Ao criar o banco de dados considere um registro inicial de administrador do sistema com: usuário: admin; senha: admin. Lembre-se de salvar a senha em md5.
- Para ser possível realizar qual operação na API o usuário deve ser autenticado.
- O acesso a todas as rotas só deve ser possível mediante validação de token JWT.
- Considere as melhores práticas de desempenho de aplicações web. e. Considere um arquivo CSV com os dados de clientes. O arquivo deve ser tratado pela sua API e os registros do arquivo CSV devem ser salvos no banco de dados. A senha do cliente deve ser o md5 do nascimento.

Banco.php:

```
<?php
class Banco{
    private static $host = 'localhost';
    private static $user = 'root';
    private static $password = '';
    private static $banco = 'clientes';
    private static $porta = 3306;
    private static $conexao = NULL;

    public static function conectar_banco(){
        if (Banco::$conexao == NULL){
            error_reporting(E_ERROR | E_PARSE);
            Banco::$conexao = new mysqli(Banco::$host, Banco::$user,
            Banco::$password, Banco::$banco, Banco::$porta);

            if (Banco::$conexao->connect_errno){
                $objResposta = new stdClass();
                $objResposta->cod = 1;
                $objResposta->erro = Banco::$conexao->connect_error;

                die(json_encode($objResposta)); // Exibe o JSON
            }
        }
    }
}
```

```

return Banco::$conexao;
}
}

```

Modelo Cliente.php:

```

<?php
require_once "Banco.php";
class Cliente implements JsonSerializable
{
    private $idCliente;
    private $nome;
    private $email;
    private $senha;
    private $nascimento;

    public function jsonSerialize()
    {
        $obj = new stdClass();
        $obj->idCliente = $this->getIdCliente();
        $obj->nome = $this->getNomeCliente();
        $obj->email = $this->getEmailCliente();
        $obj->senha = $this->getSenhaCliente();
        $obj->nascimento = $this->getNascimentoCliente();
        return $obj;
    }
    //GET E SET DOS ATRIBUTOS

    //id
    public function getIdCliente()
    {
        return $this->idCliente;
    }
    public function setIdCliente($id)
    {
        $this->idCliente = $id;
        return $this;
    }
    //nome
    public function getNomeCliente()
    {
        return $this->nome;
    }
    public function setNomeCliente($nome_cliente)
    {
        $this->nome = $nome_cliente;
    }
}

```

```

    return $this;
}
//email
public function getEmailCliente()
{
    return $this->email;
}
public function setEmailCliente($email_cliente)
{
    $this->email = $email_cliente;
    return $this;
}
//senha (usar md5!!!)
public function getSenhaCliente()
{
    return $this->senha;
}
public function setSenhaCliente($senha_cliente)
{
    $this->senha = $senha_cliente;
    return $this;
}
//nascimento
public function getNascimentoCliente()
{
    return $this->nascimento;
}
public function setNascimentoCliente($nascimento_cliente) {
    $this->nascimento = $nascimento_cliente;
    return $this;
}

public function isCliente()
{
    $conexao = Banco::conectar_banco();
    $sql = "SELECT COUNT(*) AS qtd FROM Cliente WHERE email = ?";
    $prepareSql = $conexao->prepare($sql);
    $prepareSql->bind_param("s", $this->email);
    $executou = $prepareSql->execute();
    $dados = $prepareSql->get_result();
    $valor = $dados->fetch_object();
    return $valor->qtd > 0;
}
public function create()
{
    $conexao = Banco::conectar_banco();

```

```

    $sql = "INSERT INTO Cliente (nome,email,senha,nascimento) VALUES
    (?, ?, ?, ?);";
    $prepareSql = $conexao->prepare($sql);
    $prepareSql->bind_param("ssss", $this->nome,$this->email,$this->senha,$this->nascimento);
    $executou = $prepareSql->execute();
    $idCadastrado = $conexao->insert_id;
    $this->setIdCliente($idCadastrado);
    return $executou;
}

```

```

public function readAll()
{
    $conexao = Banco::conectar_banco();
    $sql = "SELECT * FROM Cliente ORDER BY idCliente;";
    $prepareSql = $conexao->prepare($sql);
    $prepareSql->execute();
    $consulta = $prepareSql->get_result();
    $consulta = $consulta->fetch_all(MYSQLI_ASSOC);
    return $consulta;
}

```

```

public function readById()
{
    $conexao = Banco::conectar_banco();
    $sql = "SELECT idCliente,nome,email,senha,nascimento FROM Cliente WHERE
    idCliente = ?;";
    $prepareSql = $conexao->prepare($sql);
    $prepareSql->bind_param("i", $this->idCliente);
    $executou = $prepareSql->execute();
    $consulta = $prepareSql->get_result();
    $tuplas = $consulta->fetch_object();
    $cliente = new Cliente();
    $cliente->setIdCliente($tuplas->idCliente);
    $cliente->setNomeCliente($tuplas->nome);
    $cliente->setEmailCliente($tuplas->email);
    $cliente->setSenhaCliente($tuplas->senha);
    $cliente->setNascimentoCliente($tuplas->nascimento);
    $prepareSql->close();
    return $cliente;
}

```

```

public function delete()
{
    $conexao = Banco::conectar_banco();
    $sql = "DELETE FROM Cliente WHERE idCliente = ?;";
    $prepareSql = $conexao->prepare($sql);

```

```

    $prepareSql->bind_param("i", $this->idCliente);
    $executou = $prepareSql->execute();
    return $executou;
}

public function deleteAll()
{
    $conexao = Banco::conectar_banco();
    $sql = "DELETE FROM Cliente WHERE idCliente > 1;";
    $prepareSql = $conexao->prepare($sql);
    $executou = $prepareSql->execute();
    return $executou;
}

public function update()
{
    $conexao = Banco::conectar_banco();
    $sql = "UPDATE Cliente SET nome = ? WHERE idCliente = ?;";
    $prepareSql = $conexao->prepare($sql);
    $prepareSql->bind_param("si", $this->nome, $this->idCliente);
    $executou = $prepareSql->execute();
    return $executou;
}

public function login()
{
    $conexao = Banco::conectar_banco();
    $sql = "SELECT COUNT(*) AS qtd, idCliente, nome, email, nascimento FROM cliente WHERE email = ? AND senha = MD5(?);";
    $prepareSql = $conexao->prepare($sql);
    $prepareSql->bind_param("ss", $this->email, $this->senha);
    $executou = $prepareSql->execute();
    $consulta = $prepareSql->get_result();
    while($tupla = $consulta->fetch_object()){
        if($tupla->qtd == 1){
            $this->setIdCliente($tupla->idCliente); $this->
            >setNomeCliente($tupla->nome);
            $this->setEmailCliente($tupla->email); $this->
            >setNascimentoCliente($tupla->nascimento); return true;
        }
    }
    return false;
}
}

```

controle_cliente_logar.php :

```
<?php
```

```
use Firebase\JWT\MeuTokenJWT;
```

```
require_once "modelo/Cliente.php";  
require_once "modelo/MeuTokenJWT.php";  
$json = file_get_contents('php://input');  
$objJson = json_decode($json);
```

```
$cliente = new Cliente();  
$cliente->setEmailCliente($objJson->email);  
$cliente->setSenhaCliente($objJson->senha);  
$resposta = new stdClass();
```

```
if ($cliente->getEmailCliente() == ""){  
    $resposta->cod = 2;  
    $resposta->status = false;  
    $resposta->msg = "Digite seu e-mail para utilizar a API !!!"; }  
if ($cliente->getSenhaCliente() == ""){  
    $resposta->cod = 2;  
    $resposta->status = false;  
    $resposta->msg = "Digite senha para utilizar a API !!!"; }  
if ($cliente->login()){  
    $objToken = new MeuTokenJWT();
```

```
    $claims = new stdClass();  
    $claims->idCliente = $cliente->getIdCliente();  
    $claims->nome = $cliente->getNomeCliente();  
    $claims->email = $cliente->getEmailCliente(); $claims->  
    >nascimento = $cliente->getNascimentoCliente();
```

```
    $token = $objToken->gerarToken($claims);
```

```
    $resposta->cod = 1;  
    $resposta->status = true;  
    $resposta->msg = "Logado com sucesso!";  
    $resposta->cliente = $cliente;  
    $resposta->token = $token;  
}else{  
    $resposta->cod = 2;  
    $resposta->status = false;  
    $resposta->msg = "E-mail ou senha incorretos!"; }
```

```
echo json_encode($resposta);  
if ($resposta->status == true){  
    header("HTTP/1.1 200");  
}else{
```

```
header("HTTP/1.1 401");
}
```

controle_cliente_getall.php:

```
<?php
use Firebase\JWT\MeuTokenJWT;
require_once "modelo/Cliente.php";
require_once "modelo/Banco.php";
require_once "modelo/MeuTokenJWT.php";
$resposta = new stdClass();
$cliente = new Cliente();

$headers = getallheaders();
$authorization =
$headers['Authorization']; #echo
$authorization;
$token = new MeuTokenJWT();
if ($token->
validarToken($authorization)){ $dados =
$cliente->readAll();

$resposta->cod = 1;
$resposta->mensagem = "executado com sucesso";
$resposta->status = true;
$resposta->dados = $dados;
}else{
$resposta->cod = 2;
$resposta->mensagem = "Token invalido.";
$resposta->status = false;
}
echo json_encode($resposta);

header("Content-Type:
application/json"); if ($resposta->
status == true){
header("HTTP/1.1 200");
}else{
header("HTTP/1.1 401");
}
```

controle_cliente_getbyid.php:

```
<?php
use Firebase\JWT\MeuTokenJWT;
require_once "modelo/Banco.php";
require_once "modelo/Cliente.php";
require_once
```



```

"modelo/MeuTokenJWT.php"; $resposta =
new stdClass();
$cliente = new Cliente();

$headers = getAllheaders();
$authorization =
$headers['Authorization']; #echo
$authorization;
$token = new MeuTokenJWT();
if ($token-
>validarToken($authorization)){
$cliente->setIdCliente($id);
$dados = $cliente->readById();

$resposta->cod = 1;
$resposta->mensagem = "executado com sucesso";
$resposta->status = true;
$resposta->dados = $dados;
}else{
$resposta->cod = 2;
$resposta->mensagem = "Token invalido.";
$resposta->status = false;
}

echo json_encode($resposta);

header("Content-Type:
application/json"); if ($resposta-
>status == true){
header("HTTP/1.1 200");
}else{
header("HTTP/1.1 401");
}

```

controle_cliente_update.php:

```

<?php
use Firebase\JWT\MeuTokenJWT;
require_once "modelo/Banco.php";
require_once "modelo/Cliente.php";
require_once "modelo/MeuTokenJWT.php";
$json =
file_get_contents("php://input");
$objJson = json_decode($json) or die('{"mensagem" : "formato
invalido"}');

$resposta = new stdClass();
$cliente = new Cliente();

```

```

$headers = getallheaders();
$authorization = $headers['Authorization'];
#echo $authorization;
$token = new MeuTokenJWT();
if($token->validarToken($authorization)){
    $cliente->setIdCliente($id);
    $cliente->setNomeCliente($objJson->nome);
    $cliente->setEmailCliente($objJson->email);
    $cliente->setSenhaCliente($objJson->senha);
    $cliente->setNascimentoCliente($objJson->nascimento);
    if($cliente->getNomeCliente() == "" || $cliente->getEmailCliente() == ""
    || $cliente->getSenhaCliente() == "" || $cliente-
    >getNascimentoCliente() == "") {
        $resposta->cod = 2;
        $resposta->status = false;
        $resposta->mensagem = "Por favor insira dados validos"; }
    else {
        if ($cliente->update()) {
            $resposta->cod = 1;
            $resposta->status = true;
            $resposta->mensagem = "Cliente atualizado com sucesso";
            $resposta->novoCliente = $cliente;
        } else {
            $resposta->cod = 2;
            $resposta->status = false;
            $resposta->mensagem = "Erro ao atualizar cliente"; }
        }
    }else{
        $resposta->cod = 2;
        $resposta->status = false;
        $resposta->mensagem = "Token invalido!";
    }
}

header("Content-Type: application/json");
if ($resposta->status == true) {
    header("HTTP/1.1 201");
} else {
    header("HTTP/1.1 401");
}
echo json_encode($resposta);

```

controle_cliente_create.php:

```

<?php
use Firebase\JWT\MeuTokenJWT;
require_once "modelo/Banco.php";
require_once "modelo/Cliente.php";
require_once "modelo/MeuTokenJWT.php";

```

```
$json = file_get_contents("php://input");  
$objJson = json_decode($json) or die('{"mensagem":"formato invalido"}');
```

```
$resposta = new stdClass();  
$cliente = new Cliente();  
$cliente->setNomeCliente($objJson->nome);  
$cliente->setEmailCliente($objJson->email);  
$cliente->setSenhaCliente($objJson->senha);  
$cliente->setNascimentoCliente($objJson->nascimento);
```

```
if($cliente->getNomeCliente() == "" || $cliente->getEmailCliente() == ""  
|| $cliente->getSenhaCliente() == "" || $cliente->getNascimentoCliente()  
== ""){  
    $resposta->cod = 2;  
    $resposta->status = false;  
    $resposta->erro = "Impossivel cadastrar dados vazios! Digite dados  
validos.";  
}elseif($cliente->isCliente()){  
    $resposta->cod = 3;  
    $resposta->status = false;  
    $resposta->erro = "Impossivel cadastrar e-mail ja cadastrado.";  
}else{
```

```
    $headers = getallheaders();  
    $authorization = $headers['Authorization'];  
    $token = new MeuTokenJWT();  
    if ($token->validarToken($authorization)){  
        if($cliente->create()){  
            $resposta->cod = 1;  
            $resposta->status = true;  
            $resposta->erro = "Cliente cadastrado com sucesso!";  
            $resposta->novoCliente = $cliente;  
        }  
    }else{  
        $resposta->cod = 2;  
        $resposta->status = false;  
        $resposta->erro = "Token invalido!";  
    }  
}
```

```
}
```

```
header("Content-Type:  
application/json"); if ($resposta-  
>status == true) {  
    header("HTTP/1.1 201");  
} else {  
    header("HTTP/1.1 401");
```

```
}
```

```
echo json_encode($resposta);
```

controle_cliente_delete.php:

```
<?php
use Firebase\JWT\MeuTokenJWT;
require_once "modelo/Cliente.php";
require_once "modelo/Banco.php";
require_once "modelo/MeuTokenJWT.php";
$resposta = new stdClass();
$cliente = new Cliente();

$headers = getallheaders();
$authorization =
$headers['Authorization']; #echo
$authorization;
$token = new MeuTokenJWT();
if ($token->
validarToken($authorization)){
$cliente->setIdCliente($id);
if($cliente->delete()){
$resposta->cod = 1;
$resposta->status = true;
$resposta->msg = "Deletado com sucesso!";
}else{
$resposta->cod = 2;
$resposta->status = false;
$resposta->msg = "Erro ao deletar cliente!"; }
}else{
$resposta->cod = 2;
$resposta->status = false;
$resposta->msg = "Token invalido!"; }

if($resposta->status == true){
header("Content-Type: application/json");
header("HTTP/1.1 204");
}else{
header("Content-Type: application/json");
header("HTTP/1.1 404");
}
```

controle_cliente_delete_all.php:

```
<?php
use Firebase\JWT\MeuTokenJWT;
```

```

require_once "modelo/Cliente.php";
require_once "modelo/Banco.php";
require_once "modelo/MeuTokenJWT.php";
$resposta = new stdClass();
$cliente = new Cliente();

$headers = getallheaders();
$authorization =
$headers['Authorization']; #echo
$authorization;
$token = new MeuTokenJWT();
if ($token->validarToken($authorization)){
$cliente->setIdCliente($id);
if($cliente->delete()){
$resposta->cod = 1;
$resposta->status = true;
$resposta->msg = "Deletado com sucesso!";
}else{
$resposta->cod = 2;
$resposta->status = false;
$resposta->msg = "Erro ao deletar cliente!"; }
}else{
$resposta->cod = 2;
$resposta->status = false;
$resposta->msg = "Token invalido!"; }

if($resposta->status == true){
header("Content-Type: application/json");
header("HTTP/1.1 204");
}else{
header("Content-Type: application/json");
header("HTTP/1.1 404");
}
}

```

controle_cliente_create_csv.php:

```

<?php
require_once 'modelo/Cliente.php';

if (isset($_FILES['csv'])) {
$nomeArquivo = $_FILES['csv']['tmp_name'];

if (($handle = fopen($nomeArquivo, "r")) != false) {
$isFirstRow = true;
$i = 0;
$cliente = array();

```

```

while (($linhaArquivo = fgetcsv($handle, 1000, ";")) !== false) { if
($isFirstRow) {
    $isFirstRow = false;
    continue;
}
$linhaArquivo = array_map("utf8_encode", $linhaArquivo);

$cliente[$i] = new Cliente;
$cliente[$i]->setNomeCliente($linhaArquivo[0]); $cliente[$i]-
>setEmailCliente($linhaArquivo[1]); $cliente[$i]-
>setNascimentoCliente($linhaArquivo[2]); $cliente[$i]-
>setSenhaCliente(md5($linhaArquivo[2]));

#echo $linhaArquivo[0]."<br>";
#echo $linhaArquivo[1]."<br>";
#echo $linhaArquivo[2]."<br>";
#echo md5($linhaArquivo[2])."<br>";
if ($cliente[$i]->isCliente()){
    $resposta = new stdClass;
    $resposta->status = true;
    $resposta->mensagem = "Erro ao cadastrar! Cliente ja existente!
Verifique CSV.";
    $cadastrou = false;
    echo json_encode($resposta);
    break;
}else{
    if($cliente[$i]->create() == true) {
        $i++;
        $cadastrou = true;
    }
}

}

if ($cadastrou == true){
    $resposta = new stdClass();
    $resposta->status = true;
    $resposta->mensagem = 'Clientes cadastrados com sucesso!';
    $resposta->cadastrados = $i;
    echo json_encode($resposta);
}

} else {
    $resposta = new stdClass();
    $resposta->status = false;
    $resposta->mensagem = 'Não foi possível abrir o arquivo.'; echo
    json_encode($resposta);
}

```

```

    } else {
        $resposta = new stdClass();
        $resposta->status = false;
        $resposta->mensagem = 'Nenhum arquivo CSV enviado.';
        echo json_encode($resposta);
    }

    fclose($handle);

```

index.php:

```

<?php
require_once "modelo/Router.php";

$roteador = new Router();

// Rota de login (sem autenticação necessária)
$roteador->post("/login", function () {
    require_once ("controle/controle_cliente_logar.php");
});

// Cliente (rotas protegidas)
$roteador->get("/clientes", function () {
    require_once "controle/controle_cliente_getall.php";
});

$roteador->get("/clientes/(\d+)", function ($id) {
    require_once "controle/controle_cliente_getbyid.php";
});

$roteador->post("/clientes", function () {
    require_once "controle/controle_cliente_create.php";
});

$roteador ->post("/clientes/csv",function(){

```

```
require_once "controle/controle_cliente_create_csv.php";
});
```

```
$roteador->put("/clientes/(\d+)", function ($id) {
    require_once "controle/controle_cliente_update.php";
});
```

```
$roteador->delete("/clientes/(\d+)", function ($id) {
    require_once "controle/controle_cliente_delete.php";
});
```

```
$roteador->delete("/clientes", function () {
    require_once "controle/controle_cliente_delete_all.php";
});
```

```
$roteador->run();
```

- f. Considere tratamento de exceções e erros de conexão com banco e operações com o banco.
- g. Construa programas em Javascript para resolver os problemas abaixo:
 - i. Converter uma quantidade de horas digitadas pelo usuário em minutos. Informe o resultado em minutos.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial
scale=1.0">
    <title>1</title>
</head>
<body>
    <input type="number" name="" id="horas" placeholder="Horas">
    <button id="botao">Converter para minutos</button>
    <div id="resposta"></div>
</body>
```



```

<script>
  const Objhoras = document.getElementById("horas")
  const Objresposta = document.getElementById("resposta")
  const Objbotao = document.getElementById("botao")

  Objbotao.onclick = function(){
    while (Objresposta.hasChildNodes()){
      Objresposta.removeChild(Objresposta.firstChild)
    }

    const valor_horas = horas.value
    let resposta = valor_horas + " Hora(s) em minutos: " +
    valor_horas*60
    let txtresposta = document.createTextNode(resposta)
    Objresposta.appendChild(txtresposta)
  }
</script>
</html>

```

- ii. Crie um programa que dada a idade de uma pessoa calcule quantos dias, horas, minutos e segundo essa pessoa já viveu

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>2</title>
</head>
<body>
  <input type="number" name="" id="idade" placeholder="Idade">
  <button id="botao">Calcular</button>
  <div id="resposta"></div>
</body>
<script>
  const idade = document.getElementById("idade")
  const botao = document.getElementById("botao")
  const label_resposta = document.getElementById("resposta")

  botao.onclick = function(){
    //limpa div
    while (label_resposta.hasChildNodes()){
      label_resposta.removeChild(label_resposta.firstChild)
    }

    let anos = idade.value
    let dias = anos*365
    let horas = dias*24

```

```

let minutos = horas*60
let segundos = minutos*60
let resposta = "Anos : " + anos + "\nDias: " + dias + "\nHoras: " +
horas + "\nMinutos: " + minutos + "\nSegundos: " + segundos
let
txtresposta = document.createTextNode(resposta)
label_resposta.appendChild(txtresposta)
}
</script>
</html>

```

- iii. Crie um programa que leia duas notas de um aluno e apresente a média.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial
scale=1.0">
<title>3</title>
</head>
<body>
<input type="number" name="" id="n1">
<input type="number" name="" id="n2">
<button id="botao">Calcular Media</button>
<div id="resposta"></div>
</body>
<script>
const botao = document.getElementById("botao")
const label_n1 = document.getElementById("n1")
const label_n2 = document.getElementById("n2")
const label_resposta = document.getElementById("resposta")
botao.onclick = function(){
while(label_resposta.hasChildNodes()){
label_resposta.removeChild(label_resposta.firstChild) }

let n1 = Number(label_n1.value)
let n2 = Number(label_n2.value)
let media = (n1+n2)/2
console.log(media)
let resposta = "Media das notas: " + n1 + " e " + n2 + " = " + media
let txtresposta = document.createTextNode(resposta)
label_resposta.appendChild(txtresposta)
}
</script>
</html>

```

- iv. Uma imobiliária vende terrenos retangulares. Faça um programa para ler as dimensões de um terreno e depois exibir a área e comprimento.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>4</title>
</head>
<body>
  <input type="number" name="" id="largura">
  <input type="number" name="" id="comprimento">
  <button id="botao">Calcular Area e Perimetro</button>
  <div id="resposta"></div>
</body>
<script>
  const botao = document.getElementById("botao")
  const label_1 = document.getElementById("largura")
  const label_2 = document.getElementById("comprimento")  const
label_resposta = document.getElementById("resposta")

  botao.onclick = function(){
    while(label_resposta.hasChildNodes()){
      label_resposta.removeChild(label_resposta.firstChild)  }
    let largura = Number(label_1.value)
    let comprimento = Number(label_2.value)
    let area = largura*comprimento
    let perimetro = largura*2 + comprimento*2
    let resposta = "Area = " + area + " Perimetro = " + perimetro  let
txtresposta = document.createTextNode(resposta)
    label_resposta.appendChild(txtresposta)
  }
</script>
</html>
```

- v. Escreva um programa para ler o número total de eleitores de um município, o número de votos brancos, nulos e válidos. Calcular e escrever o percentual que cada um representa em relação ao total de eleitores.

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial
scale=1.0">
<title>5</title>
</head>
<body>
<input type="number" id="total"><br>
<input type="number" id="vb"><br>
<input type="number" id="vn"><br>
<input type="number" id="vv"><br>
<button id="botao">Calcular percentual</button>
<div id="resposta"></div>
</body>
<script>
const botao = document.getElementById("botao")
const label_1 = document.getElementById("total")
const label_2 = document.getElementById("vb")
const label_3 = document.getElementById("vn")
const label_4 = document.getElementById("vv")
const label_resposta = document.getElementById("resposta")

botao.onclick = function(){
while(label_resposta.hasChildNodes()){
label_resposta.removeChild(label_resposta.firstChild) }
let total = Number(label_1.value)
let vb = Number(label_2.value)
let vn = Number(label_3.value)
let vv = Number(label_4.value)

let resposta =
`
Votos Totais: ${total}
Percentual de Votos Brancos: ${(vb/total)*100}%
Percentual de Votos Nulos: ${(vn/total)*100}%
Percentual de Votos Validos: ${(vv/total)*100}% `
let txtresposta = document.createTextNode(resposta)
label_resposta.appendChild(txtresposta)
}
</script>
</html>

```

- vi. Faça um programa que calcule e mostre valores de graus Celsius e graus Fahrenheit, cujos graus variem de 40 a 70 F de 1 em 1. A conversão de graus Fahrenheit (F) para graus Celsius (C) é dada por: $C = 5/9 * (F - 32)$.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>6</title>
</head>
<body>
  <div id="resposta"></div>

  <script>
    const lbl_resposta = document.getElementById("resposta")
    let txtresposta = ""

    for(let celsius = 40; celsius <= 70; celsius++){
      let far = celsius * 1.8 + 32
      txtresposta += `${celsius} C° = ${far} F°<br>`
    }

    lbl_resposta.innerHTML = txtresposta
  </script>
</body>
</html>

```

- vii. Faça um programa que calcule e mostre a área de um trapézio.
 Sabe-se que: $A = (base\ maior + base\ menor) * altura / 2$;

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>7</title>
</head>
<body>
  <input type="number" name="bm" id="bm">
  <input type="number" name="bM" id="bM">
  <input type="number" name="h" id="h">
  <input type="submit" value="Calcular area" id="btn">
  <div id="resposta"></div>
  <script>

    const div = document.getElementById("resposta")
    const btn = document.getElementById("btn")

```

```

    btn.onclick = function(){
    let bm = Number(document.getElementById("bm").value) let bM =
    Number(document.getElementById("bM").value) let h =
    Number(document.getElementById("h").value) let area = (bm +
    bM)*h/2
    let resposta = `Area = ${area}`
    let texto = document.createTextNode(resposta)
    div.appendChild(texto)
    }

</script>
</body>
</html>

```

- viii. Calcular o salário líquido do funcionário sabendo que este é constituído pelo salário bruto mais o valor das horas extras subtraindo 8% de INSS do total. Serão lidos nesse problema o salário bruto, o valor das horas extras e o número de horas extras. Apresentar ao final o salário líquido.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial
scale=1.0">
<title>8</title>
</head>
<body>
<div>
<input type="number" name="" id="salario_bruto"> <input
type="number" name="" id="valor_horas_extras"> <input
type="number" name="" id="qtd_horas_extras"> <input
type="submit" id="btn" value="Calcular liquido"> <div
id="resposta"></div>
</div>
<script>
const btn = document.getElementById("btn")
const div = document.getElementById("resposta")

    btn.onclick = function(){
    let sb =
    Number(document.getElementById("salario_bruto").value)
    let vhe =
    Number(document.getElementById("valor_horas_extras").value
    ) let qhe =
    Number(document.getElementById("qtd_horas_extras").value)

```

```

    let liquido = (sb + (vhe*qhe))*0.92
    let resposta = `Salario Liquido: ${liquido}`
    let texto = document.createTextNode(resposta)
    div.appendChild(texto)
  }
</script>
</body>
</html>

```

- ix. Efetuar a leitura do número de quilowatts consumido e calcular o valor a ser pago de energia elétrica, sabendo-se que o valor a pagar por quilowatt é de 0,12. Apresentar o valor total a ser pago pelo usuário acrescido de 18% de ICMS.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>9</title>
</head>
<body>
  Quilowatts de consumo <input type="number" name="" id="kW"><br>
  <input type="submit" value="Calcular valor a pagar" id="btn">
<div id="resposta"></div>

  <script>
    const div = document.getElementById("resposta")
    const btn =document.getElementById("btn")

    btn.onclick = function(){
      let kW = Number(document.getElementById("kW").value)
      let valor = (0.12 * kW)*1.18
      let texto = `Valor a pagar $R${valor}`
      let resposta = document.createTextNode(texto)
      div.appendChild(resposta)
    }
  </script>
</body>
</html>

```

- x. Calcular a média de combustível gasto pelo usuário, sendo informado a quantidade de quilômetros rodados e a quantidade de

combustível.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>10</title>
</head>
<body>
  Quantidade de Km rodados <input type="number" name=""
id="qtd_km"><br>
  Quantidade de combustível [L]<input type="number" name=""
id="qtd_comb"><br>
  <input type="submit" value="Calcular Media" id="btn">
<div id="resposta"></div>
  <script>
    const div = document.getElementById("resposta")
    const btn = document.getElementById("btn")
    btn.onclick = function(){
      let qtd_km = Number(document.getElementById("qtd_km").value)
      let qtd_com =
        Number(document.getElementById("qtd_comb").value)
      let media = qtd_km/qtd_com
      let texto = `Media de combustível gasto: ${media}L`
      let resposta = document.createTextNode(texto)
      div.appendChild(resposta)
    }
  </script>
</body>
</html>
```

- xi. Crie um programa que dada a idade de uma pessoa calcule quantos dias, horas, minutos e segundo essa pessoa já viveu.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>11</title>
</head>
<body>
  Digite sua idade: <input type="number" name="" id="idade">
  <input type="submit" value="Calcular dias" id="btn"> <div>
```



```

id="resposta"></div>
<script>
const div = document.getElementById("resposta") const
btn = document.getElementById("btn")

btn.onclick = function(){
let idade =
Number(document.getElementById("idade").value)
let dias = idade*365
let horas = dias*24
let minutos = horas*60
let segundos = minutos*60
let texto = `
Anos = ${idade}
Dias = ${dias}
Horas = ${horas}
Minutos = ${minutos}
Segundos = ${segundos}
`
let resposta = document.createTextNode(texto)
div.appendChild(resposta)
}
</script>
</body>
</html>

```

- xii. Faça um algoritmo que calcule e mostre valores de graus Celsius e graus Fahrenheit, cujos graus variem de 40 a 70 F de 1 em 1. A conversão de graus Fahrenheit (F) para graus Celsius (C) é dada por: $C = 5/9 * (F - 32)$.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial
scale=1.0">
<title>12</title>
</head>
<body>
<div id="resposta"></div>

<script>
const lbl_resposta = document.getElementById("resposta") let
txtresposta = ""

```

```

for(let celsius = 40; celsius <= 70; celsius++){ let
far = celsius * 1.8 + 32
txtresposta += `${celsius} C° = ${far} F°<br>` }

lbl_resposta.innerHTML = txtresposta
</script>
</body>
</html>

```

xiii. Uma fábrica de refrigerantes Meia-Cola vende seu produto em três formatos: lata de 350 ml, garrafa de 600 ml e garrafa de 2 litros.

Se um comerciante compra uma determinada quantidade de cada formato, faça um algoritmo para calcular quantos litros de refrigerante ele comprou.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial
scale=1.0">
<title>13</title>
</head>
<body>
Garrafas de 2L <input type="number" name="" id="grande"><br>
Garrafas de 600 ml <input type="number" name="" id="media"><br>
Garrafas de 350 ml <input type="number" name="" id="pequena"><br>
<input type="submit" value="Calcular Total Litros" id="btn"> <div
id="resposta"></div>
<script>
const div = document.getElementById("resposta") const
btn = document.getElementById("btn")

btn.onclick = function(){
let grandes =
Number(document.getElementById("grande").value)
let medias =
Number(document.getElementById("media").value)
let pequenas =
Number(document.getElementById("pequena").value)
let total = (grandes*2)+(medias*0.600)+(pequenas*0.350) let texto =
`Total Litros = ${total}L`
let resposta = document.createTextNode(texto)
div.appendChild(resposta)
}
</script>

```

```
</body>
</html>
```

- xiv. Construa um programa que leia três notas de um aluno, calcule a média obtida por este aluno e no final escreva o resultado indicando se o mesmo foi **aprovado** ou **reprovado** (considere que aluno aprovado obteve Média $\geq 7,0$ e aluno reprovado Média $< 7,0$).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>14</title>
</head>
<body>
  Nota 1<input type="number" name="" id="n1" minlength="2"><br>
  Nota 2<input type="number" name="" id="n2" minlength="2"><br>
  Nota 3<input type="number" name="" id="n3" minlength="2"><br>
  <input type="submit" value=" Média" id="btn">
  <div id="resposta"></div>

  <script>
    const btn = document.getElementById("btn")
    const div = document.getElementById("resposta")

    btn.onclick = function(){
      let n1 = Number(document.getElementById("n1").value)
      let n2 = Number(document.getElementById("n2").value)
      let n3 = Number(document.getElementById("n3").value)
      let media = (n1+n2+n3)/3
      let texto = `Média das notas: ${media}`
      let resposta = document.createTextNode(texto)
      div.appendChild(resposta)
    }
  </script>
</body>
</html>
```

- xv. Construa um programa que leia nome de um funcionário, o número de horas trabalhadas, o valor que recebe por horas trabalhadas e o número de filhos, com estas informações, calcular o salário deste funcionário, considerando que o mesmo terá uma gratificação de

3% sobre o salário bruto por cada filho, caso o mesmo possua acima de três filhos. Escreva ao final, o nome do funcionário, seu respectivo salário e o acréscimo de salário, caso ela tenha tido direito a esta gratificação.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>15</title>
</head>
<body>
  Nome do funcionario <input type="text" name="" id="nome"><br>
  Numero de horas trabalhadas <input type="number" name=""
id="nht"><br>
  Valor por hora trabalhada <input type="number" name="" id="vht"><br>
  Numero de filhos <input type="number" name="" id="qtd_filhos"><br>
  <input type="submit" value="Calcular salario" id="btn"><br> <div
id="resposta"></div>

  <script>
    const div = document.getElementById("resposta")
    const btn = document.getElementById("btn")

    btn.onclick = function(){
      let nome = document.getElementById("nome").value let nht =
      Number(document.getElementById("nht").value) let vht =
      Number(document.getElementById("vht").value) let qtd_filhos =
      Number(document.getElementById("qtd_filhos").value)
      let salario = nht*vht

      let resposta = `Nome:${nome} - Salario:${salario}` if
      (qtd_filhos > 3){
        let percentual = 0.03*qtd_filhos
        let acrescimo = percentual*salario
        resposta += `Acrescimo:${ acrescimo}`
      }
      let texto = document.createTextNode(resposta)
      div.appendChild(texto)
    }
  </script>
</body>
</html>
```

- xvi. Construa programa que leia três lados de um triângulo, verifique e escreva que tipo de triângulo eles formam (considere **triângulo equilátero** com três lados iguais, **triângulo isósceles** com dois lados iguais e **triângulo escaleno** com todos os lados diferentes).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>16</title>
</head>
<body>
  Lado 1<input type="number" name="" id="l1"><br>
  Lado 2<input type="number" name="" id="l2"><br>
  Lado 3<input type="number" name="" id="l3"><br>
  <input type="submit" value="" id="btn">
  <div id="resposta"></div>

  <script>
    const div = document.getElementById("resposta")
    const btn = document.getElementById("btn")

    btn.onclick = function(){
      let l1 = Number(document.getElementById("l1").value) let l2 =
Number(document.getElementById("l2").value) let l3 =
Number(document.getElementById("l3").value)

      if(l1 == l2 && l1 == l3){
        classificacao = "equilatero"
      }else if(l1 != l2 && l2 != l3 && l1 != l3){
        classificacao = "escaleno"
      }else{
        classificacao = "isocetes"
      }
      let resposta = `Classificacao: ${classificacao}` let texto
= document.createTextNode(resposta) div.appendChild(texto)
    }
  </script>
</body>
</html>
```

- xvii. Construa programa para determinar se o indivíduo esta com um peso favorável. Essa situação é determinada através do IMC (Índice de Massa Corpórea), que é definida como sendo a relação

entre o peso (PESO) e o quadrado da Altura (ALTURA) do indivíduo. Ou seja

$$IMC = \frac{PESO}{ALTURA^2}$$

Escreva na tela de acordo com as seguintes situações:

Condição	Situação
IMC abaixo de 20	Abaixo do peso
IMC de 20 até 25	Peso Normal
IMC de 25 até 30	Sobre Peso
IMC de 30 até 40	Obeso
IMC de 40 e acima	Obeso Mórbito

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>17</title>
</head>
<body>
  Altura : <input type="number" name="" id="h"><br>
  Peso : <input type="number" name="" id="peso"><br>
  <input type="submit" value="Calcular IMC" id="btn">
  <div id="resposta"></div>

  <script>
    const div = document.getElementById("resposta")
    const btn = document.getElementById("btn")

    btn.onclick = function(){
      let h = Number(document.getElementById("h").value) let peso =
Number(document.getElementById("peso").value) let imc = peso/(h*h)
      let texto = ""
      if (imc < 20){
        texto = "Abaixo do Peso"
      }else if(imc >=20 && imc <25){
        texto = "Peso Normal"
      }else if(imc >=25 && imc <30){
        texto = "Sobre Peso"
      }else if(imc >= 30 && imc <40){
        texto = "Obeso"
      }else if(imc >= 40){
        texto = "Obeso Morbido"
      }
      let resposta = document.createTextNode(texto)
      div.appendChild(resposta)
```

```

}
</script>
</body>
</html>

```

- xviii. Construa programa que leia três lados de um triângulo, verifique e escreva que tipo de triângulo eles formam (considere triângulo equilátero com três lados iguais, triângulo isósceles com dois lados iguais e triângulo escaleno com todos os lados diferentes).

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial
scale=1.0">
<title>16</title>
</head>
<body>
Lado 1<input type="number" name="" id="l1"><br>
Lado 2<input type="number" name="" id="l2"><br>
Lado 3<input type="number" name="" id="l3"><br>
<input type="submit" value="" id="btn">
<div id="resposta"></div>

<script>
const div = document.getElementById("resposta")
const btn = document.getElementById("btn")

btn.onclick = function(){
let l1 = Number(document.getElementById("l1").value) let l2 =
Number(document.getElementById("l2").value) let l3 =
Number(document.getElementById("l3").value)

if(l1 == l2 && l1 == l3){
classificacao = "equilatero"
}else if(l1 != l2 && l2 != l3 && l1 != l3){
classificacao = "escaleno"
}else{
classificacao = "isocetes"
}

let resposta = `Classificacao: ${classificacao}` let texto
= document.createTextNode(resposta) div.appendChild(texto)
}
</script>
</body>
</html>

```

xix. Elabore um algoritmo que dada a idade de um nadador classifique o em uma das seguintes categorias:

Infantil A = 5 a 7 anos

Infantil B = 8 a 11 anos

Juvenil A = 12 a 13 anos

Juvenil B = 14 a 17 anos

Adultos = Maiores de 18 anos

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>19</title>
</head>
<body>
  Idade <input type="number" name="" id="idade"><br>
  <input type="submit" value="Classificar" id="btn">
  <div id="resposta"></div>
  <script>
    const div = document.getElementById("resposta")
    const btn = document.getElementById("btn")

    btn.onclick = function(){
      let idade = Number(document.getElementById("idade").value)
      let classificacao = ""
      if (idade >=5 && idade <=7){
        classificacao = "Infantil A"
      }else if(idade >=8 && idade <=11){
        classificacao = "Infantil B"
      }else if(idade >= 12 && idade <=13){
        classificacao = "Juvenil A"
      }else if(idade >= 14 && idade <=17){
        classificacao = "Juvenil B"
      }else if(idade >= 18){
        classificacao = "Adultos"
      }
      let resposta = document.createTextNode(classificacao)
      div.appendChild(resposta)
    }
  </script>
</body>
</html>
```


Elabore um algoritmo para calcular a equação do segundo grau e imprima as raízes da equação na tela, os valores a, b e c são inseridos

- $\pm \Delta$

pelo usuário: ax^2

+ bx + c = 0, $\Delta = a^2 - 4ac$,
xx.

$\frac{-b \pm \sqrt{\Delta}}{2a}$

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial
scale=1.0">
<title>20</title>
</head>
<body>
Valores para equacao de segundo grau<br>
Digite A <input type="number" name="" id="a"><br> Digite
B <input type="number" name="" id="b"><br> Digite C <input
type="number" name="" id="c"><br> <input type="submit"
value="Calcular raizes" id="btn">
<div id="resposta"></div>
<script>
const div = document.getElementById("resposta")
const btn = document.getElementById("btn")

btn.onclick = function(){
let a = Number(document.getElementById("a").value) let b =
Number(document.getElementById("b").value) let c =
Number(document.getElementById("c").value)
let texto = bhaskara(a,b,c)
let resposta = document.createTextNode(texto)
div.appendChild(resposta)
}

function calcularDelta(a,b,c){
return (b*b)-(4*a*c)
}

function bhaskara(a,b,c){
let delta = calcularDelta(a,b,c)
if(delta < 0){
return "A equacao nao possui raizes" }else
if(delta == 0){
let x = (-b + Math.pow(delta,0.5))/2*a return "Raiz
= " + x
}else if(delta>0){
```

```

    let x1 = (-b + Math.pow(delta,0.5))/2*a    let x2 = (-b
- Math.pow(delta,0.5))/2*a    return `Raiz 1 = ${x1} e
Raiz 2 = ${x2}`  }
}
</script>
</body>
</html>

```

xxi. Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um programa que calcule seu peso ideal, utilizando as seguintes fórmulas:

h. Para homens: $(72.7h * h) - 58$

i. Para mulheres: $(62.1 * h) - 44.7$

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
scale=1.0">
  <title>21</title>
</head>
<body>
  Altura <input type="number" name="" id="h"><br>
  Sexo <select name="" id="sexo">
    <option value="m">Masculino</option>
    <option value="f">Feminino</option>
  </select><br>
  <input type="submit" value="Calcular Peso Ideal" id="btn"><br>
  <div id="resposta"></div>
  <script>
    const div = document.getElementById("resposta")
    const btn = document.getElementById("btn")

    btn.onclick = function(){
      let altura = Number(document.getElementById("h").value)    let sexo
= document.getElementById("sexo").value    let peso_ideal

      if(sexo === "m"){
        peso_ideal = (72.7*altura)-58
      }else if(sexo === "f"){
        peso_ideal = (62.1*altura)-44.7
      }
    }
  </script>

```

```

}
let texto = `Peso Ideal = ${peso_ideal}`
let resposta = document.createTextNode(texto)
div.appendChild(resposta)
}
</script>
</body>
</html>

```

- j. Construa uma interface gráfica do usuário que acessa API com as seguintes páginas:
- i. Login do usuário.
 - ii. Interface para trocar senha.
 - iii. CRUD completo de usuário.

login.html:

```

<!DOCTYPE html>
<html lang="pt-Br">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login</title>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<link rel="stylesheet" href="../style/style.css">
</head>

<body>
<form method="post">
<a class="voltar" href="bem_vindo.html">Voltar</a>
<h3>Entrar</h3>
<div id="resposta" style="color:red"></div>

<div class="textbox">
<input type="email" name="email" id="email" required> <label
for="email">E-mail</label>
</div>

<div class="textbox">
<input type="password" name="senha" id="senha" required> <label
for="senha">Senha</label>
</div>

```

```

<input class="button" type="submit" name="submit" value="Fazer Login"
id="btn">
</form>
<script>
const btn = document.getElementById("btn")
const lbl_email = document.getElementById("email")
const lbl_senha = document.getElementById("senha")

btn.onclick = function () {
event.preventDefault()
document.getElementById("resposta").textContent = ""
fazerLogin()
}

function fazerLogin() {
let email = lbl_email.value
let senha = lbl_senha.value

const objJson = {
"email": email,
"senha": senha
}
//console.log(objJson)
const uri = "/logar"
fetch(uri, {
method: "POST",
body: JSON.stringify(objJson)
}).then((resposta_backEnd) => {
return resposta_backEnd.text()
}).then((texto_resposta_backEnd) => {
console.log(texto_resposta_backEnd)
processarResultados(texto_resposta_backEnd)
}).catch((error) => {
console.log(error)
})
}

function processarResultados(txt_json) {
const obj_json = JSON.parse(txt_json);
if (obj_json.status == true) {
localStorage.setItem("dados", txt_json);
window.location.href = "usuarios.html"
} else {
//alert("login invalido")
const lblResposta = document.createTextNode("Usuario ou senha
invalidos!")
const divResposta =

```

```
document.getElementById("resposta").appendChild(lblResposta) }
}
```

```
<!--
```

```
video base para criacao do sistema:
```

```
https://www.youtube.com/watch?v=15CXhnQkily&list=PLSHNk\_yA5FNjoIRNHV3FprsN3NWpcnnK&index=4
```

```
-->
```

```
</body>
```

```
</html>
```

usuarios.html:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Document</title>
```

```
<link rel="stylesheet" href="../style/style.css">
```

```
</head>
```

```
<body>
```

```
<form class="form">
```

```
<h3>Usuarios</h3>
```

```
<a type="button" class="button" href="">Adicionar Usuário</a> <!-------
```

```
-----Carregando usuarios do Banco-----
```

```
<!-------
```

```
----->
```

```
<div id="textbox" class="textbox">
```

```
<input type="text" name="usuario" placeholder="Usuario" required>
```

```
<input type="password" name="senha" id="senha" placeholder="Nova Senha"
required>
```

```
<input type="password" name="confirmar_senha" id="confirmar_senha"
placeholder="Confirmar Senha" required> <div class="resposta"
```

```
style="color:red"></div> <button type="submit" class="botao">Confirmar
mudança</button> <a style="width: 120px; margin-top:50px" id="botao"
```

```
class="input" href=""
```

```
onclick="ocultaCampo(event)">Cancelar</a> </div>
```

```
<a id="voltar" class="voltar" href="bem_vindo.html">Sair</a>
```

```
</form>
```

```
</body>
<script>
  let JsonDados = JSON.parse(localStorage.getItem("dados"))
  let token = JsonDados.token
  const headers = new Headers(options.headers || {});
  headers.append('Authorization', `Bearer ${token}`);
  console.log(token)
```

```
  carregarUsuarios()
  function carregarUsuarios() {
    const uri = '/clientes_a'
    fetch(uri, {
      method: "GET",
      headers: headers
    }).then((resposta_backEnd) => {
      return resposta_backEnd.text()
    }).then((texto_resposta_backEnd) => {
      console.log(texto_resposta_backEnd)
      //processarResultados(texto_resposta_backEnd)
    }).catch((error) => {
      console.log(error)
    })
  }
</script>
```

```
</html>
```