**Fundação valeparaibana de ensino**colégios univap – unidade centro

Curso técnico em informática

THIAGO CESAR CARVALHO

.

**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  
2023

**PROGRAMAÇÃO AVANÇADA PARA WEB  
LISTA DE EXERCÍCIOS II – PHP BÁSICO**

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

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

* 1. 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.
  2. Para ser possível realizar qual operação na API o usuário deve ser autenticado.
  3. O acesso a todas as rotas só deve ser possível mediante validação de token JWT.
  4. Considere as melhores práticas de desempenho de aplicações web
  5. 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.
  6. Considere tratamento de exceções e erros de conexão com banco e operações com o banco.
* **>> CÓDIGO LOGO ABAIXO COM TODAS AS QUESTÕES.**

**=-=-=-=-=-=-=-=-=-=-= INDEX =-=-=-=-=-=-=-=-=-=-=**

**<?php**

**require\_once("modelo/Router.php");**

**$roteador = new Router();**

**$roteador->get("/usuarios/readpage/(\d+)/", function($pagina){**

**require\_once("controle/controle\_usuario\_read\_page.php");**

**});**

**$roteador->get("usuarios/(\d+)", function($parametro\_idusuario){**

**require\_once("controle/controle\_usuario\_read\_by\_id.php");**

**});**

**$roteador->post("usuarios", function(){**

**require\_once("controle/controle\_usuario\_create.php");**

**});**

**$roteador->delete("usuarios/(\d+)", function($parametro\_idusuario){**

**require\_once("controle/controle\_usuario\_delete.php");**

**});**

**$roteador->put("usuarios/(\d+)", function($parametro\_idusuario){**

**require\_once("controle/controle\_usuario\_update.php");**

**});**

**$roteador->put("/usuarios/senha", function(){**

**require\_once("controle/controle\_usuario\_trocarSenha.php");**

**});**

**$roteador->post ("logar",function(){**

**require\_once("controle/controle\_usuario\_login.php");**

**});**

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

**require\_once("controle/controle\_usuario\_create\_from\_csv.php");**

**});**

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

**?>**

**=-=-=-=-=-=-=-=-=-=-= CONTROLE =-=-=-=-=-=-=-=-=-=-=**

Controle\_usuario\_Create.php

<?php

require\_once("modelo/Banco.php");

require\_once("modelo/Usuario.php");

$textorecibido =file\_get\_contents("php://input");

$objJson = json\_decode($textorecibido) or die ('{"msg":"formato incorreto!"}');;

$objResposta = new stdClass();

$objUsuario = new Usuario();

$objUsuario -> setNomeUsuario($objJson->usuario->nomeUsuario);

$objUsuario->setEmail($objJson->usuario->emailUsuario);

$objUsuario -> setSenha($objJson->usuario->senha);

$objUsuario->setNascimento($objJson->usuario->nascimento);

try{

if ($objUsuario->getNomeUsuario() == '') {

$objResposta-> status = false;

$objResposta->msg = "Erro! o nome nao pode ser vazio!";

$objResposta-> cod =1;

}else if($objUsuario->getNomeUsuario() < 3){

$objResposta-> status = false;

$objResposta->msg = "Erro! o nome nao pode ser menor que 3 caracteres!";

$objResposta-> cod =2;

}else if ($objUsuario->IsUsuario() == true) {

$objResposta-> status = false;

$objResposta->msg = "Erro! Ja existe alguem com este email";

$objResposta-> cod =4;

}else if ($objUsuario->create()== true) {

$objResposta-> status = true;

$objResposta->msg = "Cadastrado com sucesso!";

$objResposta-> cod = 3;

}else{

$objResposta-> status = false;

$objResposta->msg = "Erro ao cadastrar!";

$objResposta-> cod =5;

}

} catch (Exception $e) {

$objResposta->codigo = 1;

$objResposta->status = false;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

}

if( $objResposta->status == true) {

header('HTTP/1.1 201');

}else {

header('HTTP/1.1 200');

}

echo json\_encode($objResposta);

**?>**

**Controle\_usuario\_delete.php**

<?php

USE Firebase\JWT\MeutokenJWT;

require\_once("modelo/Banco.php");

require\_once("modelo/Usuario.php");

require\_once("modelo/MeutokenJWT.php");

$headers = apache\_request\_headers();

$objtoken = new MeutokenJWT();

$objResposta = new stdClass();

$objUsuario = new Usuario();

try{

if ($objtoken->validarToken($headers['Authorization']) == true) {

$objUsuario->setIdUsuario($parametro\_idusuario);

if ($objUsuario->delete() == true ){

header('HTTP/1.1 204');

}else{

header('HTTP/1.1 200');

header('content-type : application/json');

}

}else{

$objResposta->status = 'false';

$objResposta->codigo = 1;

$objResposta->msg= 'Falha ao excluir! Token invalido';

}

}catch (Exception $e) {

$objResposta->codigo = 1;

$objResposta->status = false;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

}

echo json\_encode($objResposta);

?>

Controle\_usuario\_update.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once("modelo/Banco.php");

require\_once("modelo/Usuario.php");

require\_once("modelo/MeutokenJWT.php");

$textorecibido =file\_get\_contents("php://input");

$objJson = json\_decode($textorecibido) or die ('{"msg":"formato incorreto!"}');;

$headers = apache\_request\_headers();

$objResposta = new stdClass();

$objUsuario = new Usuario();

$objtoken = new MeutokenJWT();

if ($objtoken->validarToken($headers['Authorization']) == true) {

$objUsuario->setIdUsuario($parametro\_idusuario);

$objUsuario -> setNomeUsuario($objJson->usuario->nomeUsuario);

$objUsuario->setEmail($objJson->usuario->emailUsuario);

$objUsuario -> setSenha($objJson->usuario->senha);

$objUsuario->setNascimento($objJson->usuario->nascimento);

try{

if ($objUsuario->verficarUsuarioSenha()==true){

if ($objUsuario->getNomeUsuario() == '') {

$objResposta-> status = false;

$objResposta->msg = "Erro! o nome nao pode ser vazio!";

$objResposta-> cod =1;

}else if($objUsuario->getNomeUsuario() < 3){

$objResposta-> status = false;

$objResposta->msg = "Erro! o nome nao pode ser menor que 3 caracteres!";

$objResposta-> cod =2;

}else if ($objUsuario->IsUsuario() == true) {

$objResposta-> status = false;

$objResposta->msg = "Erro! o email do usuario nao pode ser o mesmo";

$objResposta-> cod =4;

}else if ($objUsuario->update()== true) {

$objResposta-> status = true;

$objResposta->msg = "Atualizado com sucesso!";

$objResposta-> cod = 3;

}else{

$objResposta-> status = false;

$objResposta->msg = "Erro ao cadastrar!";

$objResposta-> cod =5;

}

}else{

$objResposta-> status = false;

$objResposta->msg = "Token invalido";

$objResposta-> cod =6;

}

if( $objResposta->status == true) {

header('HTTP/1.1 201');

}else {

header('HTTP/1.1 200');

}

} catch (Exception $e) {

// Trata a exceção

$objResposta = new stdClass();

$objResposta->cod = 1;

$objResposta->status = false;

$objResposta->erro = $e->getMessage();

//echo json\_encode($objResposta);

die(json\_encode($objResposta));

}

}

echo json\_encode($objResposta);

?>

Controle\_usuario\_createfrocsv.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once("modelo/Usuario.php");

require\_once("modelo/MeutokenJWT.php");

$headers = apache\_request\_headers();

$tokenJWT = new MeuTokenJWT();

$resposta = new stdClass();

try{

if ($tokenJWT->validarToken($headers['Authorization']) == true) {

$nomeArquivo = $\_FILES["variavelArquivo"]["tmp\_name"];

$ponteiroArquivo= fopen($nomeArquivo,"r");

$i=0; $objUsuario = array();

while ( ($linhaArquivo = fgetcsv($ponteiroArquivo,1000,";")) !== false) {

$objUsuario[$i] = new Usuario();

$objUsuario[$i]->setNomeUsuario($linhaArquivo[0]);

$objUsuario[$i]->setEmail($linhaArquivo[1]);

$objUsuario[$i]->setSenha($linhaArquivo[2]);

$objUsuario[$i]->setNascimento($linhaArquivo[3]);

if ($objUsuario[$i]->createFromCsv()==true){ $i++;}

}

$resposta->status = true;

$resposta->msg = "Cargos cadastrados com sucesso";

$resposta->usuarios = $objUsuario;

$resposta->totalUsuarios = $i;

echo json\_encode($resposta);

}else{

$resposta->status = false;

$resposta->msg = "Token invalido";

echo json\_encode($resposta);

}

} catch (Exception $e) {

$objResposta->codigo = 1;

$objResposta->status = false;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

}

?>

Controle\_usuario\_create\_logar.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once("modelo/Usuario.php");

require\_once("modelo/MeutokenJWT.php");

$textorecibido = file\_get\_contents("php://input");

$objJson = json\_decode($textorecibido);

$objResposta = new stdClass();

$objUsuario = new Usuario();

$objUsuario->setEmail($objJson->emailUsuario);

$objUsuario -> setSenha($objJson->senha);

$objResposta = array();

try{

if ($objUsuario->verficarUsuarioSenha() == true) {

$tokenJWT = new MeuTokenJWT();

$novoToken = $tokenJWT->gerarToken(json\_encode($objUsuario));

$objResposta['status'] = 'true';

$objResposta['msg'] = "Login efetuado com sucesso";

$objResposta['token'] = $novoToken;

}else{

$objResposta['status'] = 'false';

$objResposta['msg'] = "Falha ao login";

}

}catch (Exception $e) {

$objResposta = new stdClass();

$objResposta->codigo = 1;

$objResposta->status = false;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

}

echo json\_encode($objResposta);

?>

Controle\_usuario\_read\_by\_id.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once "modelo/MeuTokenJWT.php";

require\_once("modelo/Banco.php");

require\_once("modelo/Usuario.php");

$headers = apache\_request\_headers();

$tokenJWT = new MeuTokenJWT();

$objResposta = new stdClass();

try {

if ($tokenJWT->validarToken($headers['Authorization']) == true) {

$objUsuario = new Usuario();

$objUsuario->setIdUsuario($parametro\_idusuario);

$objResposta->status = true;

$objResposta->msg = "executado com sucesso!";

$objResposta->dados = $objUsuario->ReadById();

echo json\_encode($objResposta);

}else{

$objResposta->status = false;

$objResposta->msg = "falhado com sucesso!";

echo json\_encode($objResposta);

}

} catch (Exception $e) {

$objResposta->codigo = 1;

$objResposta->status = false;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

}

header("HTTP/1.1 200");

// Define o tipo de conteúdo da resposta como JSON

header("Content-Type: application/json");

// Converte o objeto resposta em JSON e o imprime na saída

?>

Controle\_usuario\_read\_by\_page.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once ("modelo/MeuTokenJWT.php");

require\_once("modelo/Banco.php");

require\_once("modelo/Usuario.php");

try {

$headers = apache\_request\_headers();

$objResposta = new stdClass();

$tokenJWT = new MeuTokenJWT();

if ($tokenJWT->validarToken($headers['Authorization']) == true) {

$objUsuario = new Usuario();

$objUsuario->setIdUsuario($pagina);

$objResposta->status = true;

$objResposta->msg = "executado com sucesso!";

$objResposta->dados = $objUsuario->ReadByPage($pagina);

echo json\_encode($objResposta);

}else{

$objResposta->status = false;

$objResposta->msg = "falhado com sucesso!";

echo json\_encode($objResposta);

}

header("HTTP/1.1 200");

// Define o tipo de conteúdo da resposta como JSON

header("Content-Type: application/json");

echo json\_encode($objResposta);

} catch (Exception $e) {

$objResposta = new stdClass();

$objResposta->erro = $e->getMessage();

echo json\_encode($objResposta);

}

?>

**=-=-=-=-=-=-=-=-=-=-=MODELO-=-=-=-=-=-=-=-=-=--**

Banco.php

<?php

class Banco {

private static $HOST='127.0.0.1';

private static $USER='root';

private static $PWD='';

private static $DB='lista';

private static $PORT=3306;

private static $CONEXAO = null;

private static function conectar(){

error\_reporting(E\_ERROR | E\_PARSE);

try{

if(Banco::$CONEXAO==null){

Banco::$CONEXAO= new mysqli(Banco::$HOST,Banco::$USER,Banco::$PWD,Banco::$DB,Banco::$PORT);

if(Banco::$CONEXAO->connect\_error){

$objResposta =new stdClass();

$objResposta->cod= 1;

$objResposta->msg = 'eror ao conectar no banco!!';

$objResposta->erro = Banco::$CONEXAO->connect\_error;

die(json\_encode($objResposta));

}

}

} catch (Exception $e) {

// Em caso de qualquer outra exceção, trata normalmente

$objResposta = new stdClass();

$objResposta->cod = 1;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

} catch (Error $e) {

$objResposta = new stdClass();

$objResposta->cod = 2;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

}

}

public static function getConexao(){

if(Banco::$CONEXAO==null){

Banco::conectar();

}

return Banco::$CONEXAO;

}

}

?>

Usuario.php

<?php

require\_once("modelo/Banco.php");

class Usuario implements JsonSerializable {

private $idUsuario;

private $nomeUsuario;

private $senhaUsuario;

private $emailUsuario;

private $tokenUsuario;

private $nascimento;

public function jsonSerialize() {

$obj = new stdClass();

$obj->idUsuario= $this->getIdUsuario();

$obj->nomeUsuario = $this->getNomeUsuario();

$obj->senhaUsuario = $this->getsenha();

$obj->emailUsuario = $this->getEmail();

}

public function verficarUsuarioSenha () {

$conexao = Banco::getConexao();

$sql = "SELECT count(\*) AS qtd, emailUsuario, senhaUsuario FROM app\_user WHERE emailUsuario = ? AND senhaUsuario = md5(?) GROUP BY emailusuario, senhaUsuario";

$prepararsql = $conexao->prepare($sql);

$prepararsql->bind\_param("ss", $this->emailUsuario, $this->senhaUsuario);

$prepararsql ->execute();

$matriz = $prepararsql -> get\_result();

$objTupla = $matriz -> fetch\_object();

return $objTupla->qtd > 0;

}

public function verificarEmail () {

$conexao = Banco::getConexao();

$sql = "SELECT count(\*) AS qtd, emailUsuario FROM app\_user WHERE emailUsuario = ? GROUP BY emailusuario";

$prepararsql = $conexao->prepare($sql);

$prepararsql->bind\_param("s", $this->emailUsuario);

$prepararsql ->execute();

$matriz = $prepararsql -> get\_result();

$objTupla = $matriz -> fetch\_object();

return $objTupla->qtd > 0;

}

public function create (){

$conexao = Banco::getConexao() ;

$this->senhaUsuario = md5($this->senhaUsuario);

$sql = "insert into app\_user (nomeUsuario,emailUsuario,senhaUsuario) values (?,?,?)";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("sss",$this->nomeUsuario,$this->emailUsuario,$this->senhaUsuario);

$executou = $prepararsql->execute();

if (!$executou) {

throw new Exception("Erro ao executar a consulta SQL");

}

$idCadastrado = $conexao->insert\_id;

$this->setIdUsuario($idCadastrado);

return $executou;

}

public function createFromCsv (){

$conexao = Banco::getConexao() ;

$this->senhaUsuario = md5($this->nascimento);

$sql = "insert into usuario (nomeUsuario,emailUsuario,senhaUsuario) values (?,?,?)";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("sss",$this->nomeUsuario,$this->emailUsuario,$this->senhaUsuario);

$executou = $prepararsql->execute();

if (!$executou) {

throw new Exception("Erro ao executar a consulta SQL");

}

$idCadastrado = $conexao->insert\_id;

$this->setIdUsuario($idCadastrado);

return $executou;

}

public function IsUsuario (){

$conexao = Banco::getConexao() ;

$sql = "select count(\*) as qtd from usuario where emailUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("s",$this->emailUsuario);

$executou = $prepararsql->execute();

if (!$executou) {

throw new Exception("Erro ao executar a consulta SQL");

}

$matriz = $prepararsql->get\_result();

$objTupla = $matriz->fetch\_object();

return $objTupla->qtd > 0;

}

public function delete (){

$conexao = Banco::getConexao() ;

$sql = "delete from app\_user where idUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("i",$this->idUsuario);

return $prepararsql->execute();

}

public function update (){

$conexao = Banco::getConexao() ;

$this->senhaUsuario = md5($this->senhaUsuario);

$sql = "update app\_user set nomeUsuario=?,emailUsuario=?,senhaUsuario=? where idUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("sssi",$this->nomeUsuario,$this->emailUsuario,$this->senhaUsuario,$this->idUsuario);

return $prepararsql->execute();

}

public function updateSenha (){

$conexao = Banco::getConexao() ;

$this->senhaUsuario = md5($this->senhaUsuario);

$sql = "update app\_user set senhaUsuario = md5(?) where emailUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("ss",$this->senhaUsuario,$this->emailUsuario);

return $prepararsql->execute();

}

public function ReadById (){

$conexao = Banco::getConexao() ;

$sql = "select \* from app\_user where idUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("i", $this->idUsuario);

$prepararsql->execute();

if (!$prepararsql->execute()) {

throw new Exception("Erro ao executar a consulta SQL");

}

$matrizTuplas = $prepararsql->get\_result();

$matrizTuplas = $matrizTuplas ->fetch\_all(MYSQLI\_ASSOC);

return $matrizTuplas;

}

public function ReadByPage ($pagina){

$conexao = Banco::getConexao() ;

$itensPorPagina = 5;

$inicio = ($pagina - 1) \* $itensPorPagina;

$sql = "select \* from app\_user limit ?,?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("ii",$inicio,$itensPorPagina);

$executou = $prepararsql->execute();

if (!$executou) {

throw new Exception("Erro ao executar a consulta SQL");

}

$matriz = $prepararsql->get\_result();

$matriz = $matriz->fetch\_all(MYSQLI\_ASSOC);

return $matriz;

}

public function getTokenUsuario ( $token ){

return $this->tokenUsuario;

}

public function setTokenUsuario( $token ){

$this->tokenUsuario = $token;

}

public function setIdUsuario($idUsuario) {

$this->idUsuario = $idUsuario;

}

public function getIdUsuario() {

return $this->idUsuario;

}

public function setNomeUsuario($nome) {

$this->nomeUsuario = $nome;

}

public function getNomeUsuario() {

return $this->nomeUsuario;

}

public function setSenha($senha) {

$this->senhaUsuario = $senha;

}

public function getSenha() {

return $this->senhaUsuario;

}

public function setEmail($email) {

$this->emailUsuario = $email;

}

public function getEmail() {

return $this->emailUsuario;

}

public function setNascimento($nascimento) {

$this->nascimento = $nascimento;

}

public function getNascimento() {

return $this->nascimento;

}

}

?>

MeuTokenJWT.php

<?php

namespace Firebase\JWT;

use stdClass;

use Firebase\JWT\Key;

use Firebase\JWT\JWT;

use DomainException;

use Exception;

use InvalidArgumentException;

use UnexpectedValueException;

use Firebase\JWT\SignatureInvalidException;

use Firebase\JWT\ExpiredException;

require\_once "jwt/JWT.php";

require\_once "jwt/Key.php";

require\_once "jwt/SignatureInvalidException.php";

require\_once "jwt/ExpiredException.php";

class MeuTokenJWT

{

//chave de criptografia, defina uma chave forte e a mantenha segura.

private $key = "x9S4q0v+V0IjvHkG20uAxaHx1ijj+q1HWjHKv+ohxp/oK+77qyXkVj/l4QYHHTF3";

//algoritmo de criptografia para assinatura

//Suportados: 'HS256' , 'ES384','ES256', 'ES256K', ,'HS384', 'HS512', 'RS256', 'RS384'

private $alg = 'HS256';

private $type = 'JWT';

private $iss = '<http://localhost>'; //emissor do token

private $aud = '<http://localhost>'; //destinatário do token

private $sub = "acesso\_sistema"; //assunto do token

private $iat = ""; //momento de emissão

private $exp = ""; //momento de expiração

private $nbf = ""; //não é válido antes do tempo especificado

private $jti = ""; //Identificador único

private $payload; //claims

//tempo de validade do token

private $duracaoToken = 3600 \* 24 \* 30; //3600 segundos = 60 min

public function gerarToken($parametro\_claims)

{

// Criação dos headers como objeto da classe stdClass

$objHeaders = new stdClass();

$objHeaders->alg = $this->alg;

$objHeaders->typ = $this->type;

// Criação do payload como objeto da classe stdClass

$objPayload = new stdClass();

//Registered Claims

$objPayload->iss = $this->iss; // emissor do token

$objPayload->aud = $this->aud; // destinatário do token

$objPayload->sub = $this->sub; // assunto do token

$objPayload->iat = time(); // momento de criação do token

$objPayload->exp = time() + $this->duracaoToken; // momento de expiração = tempo atual + duração

$objPayload->nbf = time(); // momento em que o token torna-se valido.

$objPayload->jti = bin2hex(random\_bytes(16)); // gera um valor aleatório para jti;

//Public Claims

$objPayload->emailUsuario = $parametro\_claims->emailUsuario;

$objPayload->senhaUsuario = $parametro\_claims->senhaUsuario;

//Private claims

// Utiliza a biblioteca do Firebase para gerar o token com os parâmetros

$token = JWT::encode((array) $objPayload, $this->key, $this->alg, null, (array) $objHeaders);

return $token;

}

public function validarToken($stringToken)

{

if (isset($stringToken)) {

if ($stringToken == "") {

return false;

} else {

$remover = ["Bearer ", " "];

$token = str\_replace($remover, "", $stringToken);

try {

$payloadValido = JWT::decode($token, new Key($this->key, $this->alg));

$this->setPayload($payloadValido);

return true;

} catch (SignatureInvalidException $e) {

// A assinatura do token é inválida.

//error\_log("Invalid token signature: " . $e->getMessage());

return false;

} catch (BeforeValidException $e) {

// O token não é válido ainda (antes do tempo 'nbf').

//error\_log("Token not valid yet: " . $e->getMessage());

return false;

} catch (ExpiredException $e) {

// O token expirou.

// error\_log("Token expired: " . $e->getMessage());

return false;

} catch (InvalidArgumentException $e) {

// Argumento inválido passado.

//error\_log("Invalid argument: " . $e->getMessage());

return false;

} catch (DomainException $e) {

// Exceção de domínio genérica.

//error\_log("Domain exception: " . $e->getMessage());

return false;

} catch (UnexpectedValueException $e) {

// Valor inesperado encontrado.

//error\_log("Unexpected value: " . $e->getMessage());

return false;

} catch (Exception $e) {

// Qualquer outra exceção genérica.

//error\_log("General exception: " . $e->getMessage());

return false;

}

}

}

return false;

}

/\*\*

\* Get the value of payload

\*/

public function getPayload()

{

return $this->payload;

}

/\*\*

\* Set the value of payload

\*

\* @return self

\*/

public function setPayload($payload)

{

$this->payload = $payload;

return $this;

}

/\*\*

\* Get the value of alg

\*/

public function getAlg()

{

return $this->alg;

}

/\*\*

\* Set the value of alg

\*

\* @return self

\*/

public function setAlg($alg)

{

$this->alg = $alg;

return $this;

}

}

* 1. Construa programas em Javascript para resolver os problemas abaixo:
     1. 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>Document</title>

</head>

<body>

<div id="divResposta"> </div>

<input type="text" id="txtnum">

<button id="btn"> Clique aqui!</button>

<style>

body{

font-family: Arial, Helvetica, sans-serif;

display: flex;

align-items: center;

justify-content: center;

background-color: lavender;

}

#divResposta {

width: 450px;

height: 100px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex; /\* Inicialmente escondido \*/

justify-content: center;

text-align: center; /\* Centraliza o texto \*/

margin-top: 40px;

padding-top: 80px;

}

</style>

</body>

<script>

let txtnum = document.getElementById("txtnum");

let divResposta = document.getElementById ("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function (){

divResposta.innerHTML = "";

let minutos = (txtnum.value) \* 60;

let resp = "o numero de " + txtnum.value + " horas correspondem em minutos:" + minutos;

let texto = document.createTextNode(resp);

divResposta.appendChild(texto);

}

</script>

</html>

* + 1. 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>Document</title>

</head>

<body>

<div id="divResposta"> </div>

<input type="text" id="txtnum">

<button id="btn"> Clique aqui!</button>

<style>

body{

font-family: Arial, Helvetica, sans-serif;

display: flex;

align-items: center;

justify-content: center;

background-color: lavender;

}

#divResposta {

width: 450px;

height: 100px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex; /\* Inicialmente escondido \*/

justify-content: center;

text-align: center; /\* Centraliza o texto \*/

margin-top: 40px;

padding-top: 80px;

}

</style>

</body>

<script>

let txtnum = document.getElementById("txtnum");

let divResposta = document.getElementById ("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function (){

divResposta.innerHTML = "";

let dias = "o numero de dias vividos:" + ( (txtnum.value) \* 365) ;

let horas = "o numero de horas vividos:" + ( (txtnum.value) \* 365 \* 24);

let minutos = "o numero de minutos vividos:" + ( (txtnum.value) \* 365 \* 24 \* 60);

let segundos = "o numero de segundos vividos:" + ( (txtnum.value) \* 365 \* 24 \* 60 \* 60);

let resp = dias + horas + minutos + segundos ;

let texto = document.createTextNode(resp);

divResposta.appendChild(texto);

}

</script>

</html>

* + 1. 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>Document</title>

</head>

<body>

<div id="divResposta"> </div>

<input type="number" id="txtnum">

<input type="number" id="txtnum2">

<button id="btn"> Clique aqui!</button>

<style>

body{

font-family: Arial, Helvetica, sans-serif;

display: flex;

align-items: center;

justify-content: center;

background-color: lavender;

}

#divResposta {

width: 450px;

height: 100px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex; /\* Inicialmente escondido \*/

justify-content: center;

text-align: center; /\* Centraliza o texto \*/

margin-top: 40px;

padding-top: 80px;

}

</style>

</body>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let divResposta = document.getElementById ("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function (){

divResposta.innerHTML = "";

let media = (parseFloat(txtnum.value) +parseFloat(txtnum2.value) ) / 2 ;

let resp = "A media do aluno: " + media ;

let texto = document.createTextNode(resp)

divResposta.appendChild(texto);

}

</script>

</html>

* + 1. 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>Document</title>

</head>

<body>

<div id="divResposta"> </div>

<input type="number" id="txtnum" placeholder="largura">

<input type="number" id="txtnum2" placeholder="altura">

<button id="btn"> Clique aqui!</button>

<style>

body{

font-family: Arial, Helvetica, sans-serif;

display: flex;

align-items: center;

justify-content: center;

background-color: lavender;

}

#divResposta {

width: 450px;

height: 100px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex; /\* Inicialmente escondido \*/

justify-content: center;

text-align: center; /\* Centraliza o texto \*/

margin-top: 40px;

padding-top: 80px;

}

</style>

</body>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let divResposta = document.getElementById ("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function (){

divResposta.innerHTML = "";

let medidas = (parseFloat(txtnum.value) \* parseFloat(txtnum2.value) ) ;

let resp = "A area do terreno " + medidas + " O comprimento do terrreno " +parseFloat(txtnum2.value) ;

let texto = document.createTextNode(resp)

divResposta.appendChild(texto);

}

</script>

</html>

* + 1. 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>Document</title>

</head>

<body>

<div id="divResposta"> </div>

<input type="number" id="txtnum" placeholder="votos brancos">

<input type="number" id="txtnum2" placeholder="votos nulos">

<input type="number" id="txtnum3" placeholder="votos validos">

<button id="btn"> Clique aqui!</button>

<style>

body{

font-family: Arial, Helvetica, sans-serif;

display: flex;

align-items: center;

justify-content: center;

background-color: lavender;

}

#divResposta {

width: 450px;

height: 100px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex; /\* Inicialmente escondido \*/

justify-content: center;

text-align: center; /\* Centraliza o texto \*/

margin-top: 40px;

padding-top: 80px;

}

</style>

</body>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let txtnum3 = document.getElementById("txtnum3");

let divResposta = document.getElementById ("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function (){

divResposta.innerHTML = "";

let soma\_votos = (parseFloat(txtnum.value) + parseFloat(txtnum2.value) + parseFloat(txtnum3.value));

let vtBrancos = (parseFloat(txtnum.value)/ soma\_votos )\*100;

let vtNulos = (parseFloat(txtnum2.value)/ soma\_votos) \*100;

let vtValidos = (parseFloat(txtnum3.value) / soma\_votos)\*100;

let resp1 = "O percentual de votos brancos:" + vtBrancos + "%" ;

let resp2 = "O percentual de votos nulos:" + vtNulos +"%" ;

let resp3 = "O percentual de votos validos:" + vtValidos + "%" ;

let texto = document.createTextNode(resp1);

let texto2 = document.createTextNode(resp2);

let texto3 = document.createTextNode(resp3);

divResposta.appendChild(texto);

divResposta.appendChild(texto2);

divResposta.appendChild(texto3);

}

</script>

</html>

* + 1. 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>Document</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 700px;

margin: 0;

}

#divResposta {

display: flex;

flex-wrap:wrap;

justify-content: center;

margin-top: 20px;

}

.bloquinho {

width: 150px; /\* Ajuste a largura conforme necessário \*/

background-color: white;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

margin: 10px;

padding: 10px;

text-align: center;

}

button {

margin-top: 20px;

}

</style>

</head>

<body>

<div id="divResposta"></div>

<button id="btn">Clique aqui!</button>

<script>

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function(){

divResposta.innerHTML = ''; // Limpa o conteúdo antes de adicionar novas respostas

for (let i = 40; i <= 70; i++) {

let celsius = 5 / 9 \* (i - 32);

let celsiusArredondado = celsius.toFixed(2);

let resp1 = `Temperatura em Fahrenheit: ${i} ---> ${celsiusArredondado} Celsius`;

let bloquinho = document.createElement("div");

bloquinho.className = "bloquinho";

bloquinho.textContent = resp1;

divResposta.appendChild(bloquinho);

}

};

</script>

</body>

</html>

* + 1. 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>Document</title>

</head>

<body>

<div id="divResposta"> </div>

<input type="number" id="txtnum">

<input type="number" id="txtnum2">

<input type="number" id="txtnum3">

<button id="btn"> Clique aqui!</button>

<style>

body{

font-family: Arial, Helvetica, sans-serif;

display: flex;

align-items: center;

justify-content: center;

background-color: lavender;

}

#divResposta {

width: 450px;

height: 100px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex; /\* Inicialmente escondido \*/

justify-content: center;

text-align: center; /\* Centraliza o texto \*/

margin-top: 40px;

padding-top: 80px;

}

</style>

</body>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let txtnum3 = document.getElementById("txtnum3");

let divResposta = document.getElementById ("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function (){

divResposta.innerHTML = "";

let area = (parseFloat(txtnum.value) +parseFloat(txtnum2.value) ) \* parseFloat(txtnum3.value)/ 2 ;

let resp = "A area do trapezio: " + area ;

let texto = document.createTextNode(resp)

divResposta.appendChild(texto);

}

</script>

</html>

* + 1. 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>Document</title>

</head>

<body>

<div id="divResposta"> </div>

<input type="number" id="txtnum" placeholder="salaario bruto">

<input type="number" id="txtnum2" placeholder="valor da horas extras">

<input type="number" id="txtnum3" placeholder="quantidade de horas extras trabalhadas">

<button id="btn"> Clique aqui!</button>

<style>

body{

font-family: Arial, Helvetica, sans-serif;

display: flex;

align-items: center;

justify-content: center;

background-color: lavender;

}

#divResposta {

width: 450px;

height: 100px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex; /\* Inicialmente escondido \*/

justify-content: center;

text-align: center; /\* Centraliza o texto \*/

margin-top: 40px;

padding-top: 80px;

}

</style>

</body>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let txtnum3 = document.getElementById("txtnum3");

let divResposta = document.getElementById ("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function (){

divResposta.innerHTML = "";

let salario = parseFloat(txtnum.value) + (parseFloat(txtnum2.value) \* parseFloat(txtnum3.value));

let sal\_liq = salario - ( salario \* 0.08);

let resp1 = "Seu salario liquido corresponde a " + sal\_liq ;

let texto = document.createTextNode(resp1);

divResposta.appendChild(texto);

}

</script>

</html>

* + 1. 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>Document</title>

</head>

<body>

<div id="divResposta"> </div>

<input type="number" id="txtnum" placeholder="quilowats gastados">

<button id="btn"> Clique aqui!</button>

<style>

body{

font-family: Arial, Helvetica, sans-serif;

display: flex;

align-items: center;

justify-content: center;

background-color: lavender;

}

#divResposta {

width: 450px;

height: 100px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex; /\* Inicialmente escondido \*/

justify-content: center;

text-align: center; /\* Centraliza o texto \*/

margin-top: 40px;

padding-top: 80px;

}

</style>

</body>

<script>

let txtnum = document.getElementById("txtnum");

let divResposta = document.getElementById ("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function (){

divResposta.innerHTML = "";

let custo = parseFloat(txtnum.value) \*0.12;

let custoFinal = custo + ( custo \* 0.18);

let resp1 = "Seu custo final corresponde a " + custoFinal ;

let texto = document.createTextNode(resp1);

divResposta.appendChild(texto);

}

</script>

</html>

* + 1. 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>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px; /\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px; /\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start; /\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0; /\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="txtnum" placeholder="quantidade de combustivel rodadp">  
 <input type=”text” id=”textnum2” placeholder=”quantidade de km rodado”>

<button id="btn">Clique aqui!</button>

</div>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function () {

divResposta.innerHTML = "";

Let media = textnum2.value / textnum .value

Let msg = ‘ a media entre as grandezas sao ${media}

let p = document.createElement("p");

p.textContent = msg;

divResposta.appendChild(p);

}

</script>

</body>

</html>

* + 1. 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>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px; /\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px; /\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start; /\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0; /\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="txtnum" placeholder="Insira sua idade">

<button id="btn">Clique aqui!</button>

</div>

<script>

let txtnum = document.getElementById("txtnum");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function () {

divResposta.innerHTML = "";

let anos = parseInt(txtnum.value);

let dias = `O número de dias vividos: ${anos \* 365}`;

let horas = `O número de horas vividos: ${anos \* 365 \* 24}`;

let minutos = `O número de minutos vividos: ${anos \* 365 \* 24 \* 60}`;

let segundos = `O número de segundos vividos: ${anos \* 365 \* 24 \* 60 \* 60}`;

let respostas = [dias, horas, minutos, segundos];

respostas.forEach(function(resp) {

let p = document.createElement("p");

p.textContent = resp;

divResposta.appendChild(p);

});

}

</script>

</body>

</html>

* + 1. 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>Document</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 700px;

margin: 0;

}

#divResposta {

display: flex;

flex-wrap:wrap;

justify-content: center;

margin-top: 20px;

}

.bloquinho {

width: 150px; /\* Ajuste a largura conforme necessário \*/

background-color: white;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

margin: 10px;

padding: 10px;

text-align: center;

}

button {

margin-top: 20px;

}

</style>

</head>

<body>

<div id="divResposta"></div>

<button id="btn">Clique aqui!</button>

<script>

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

btn.onclick = function(){

divResposta.innerHTML = ''; // Limpa o conteúdo antes de adicionar novas respostas

for (let i = 40; i <= 70; i++) {

let celsius = 5 / 9 \* (i - 32);

let celsiusArredondado = celsius.toFixed(2);

let resp1 = `Temperatura em Fahrenheit: ${i} ---> ${celsiusArredondado} Celsius`;

let bloquinho = document.createElement("div");

bloquinho.className = "bloquinho";

bloquinho.textContent = resp1;

divResposta.appendChild(bloquinho);

}

};

</script>

</body>

</html>

* + 1. 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>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px; /\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px; /\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start; /\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0; /\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="txtnum" placeholder="350ml ">

<input type="text" id="txtnum2" placeholder="600ml ">

<input type="text" id="txtnum3" placeholder="2LITROS">

<button id="btn">Clique aqui!</button>

</div>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let txtnum3 = document.getElementById("txtnum3");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

let msg = "Digite a quantidade de cada quantia de coca-cola:";

let txt = document.createTextNode(msg)

divResposta.appendChild(txt);

btn.onclick = function () {

divResposta.innerHTML = "";

let anos = parseInt(txtnum.value);

let qntLitros = (parseFloat(txtnum.value) \* 0.35 )+ (parseFloat(txtnum2.value) \* 0.6 )+ (parseFloat(txtnum.value) \* 2 ) + "litros";

let respostas = qntLitros;

let p = document.createElement("p");

p.textContent = respostas;

divResposta.appendChild(p);

}

</script>

</body>

</html>

* + 1. 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 >= 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>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px; /\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px; /\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start; /\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0; /\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="txtnum" placeholder="nota ">

<input type="text" id="txtnum2" placeholder="nota2 ">

<input type="text" id="txtnum3" placeholder="nota3">

<button id="btn">Clique aqui!</button>

</div>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let txtnum3 = document.getElementById("txtnum3");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

let msg = "Digite as tres notas do aluno correspondente";

let txt = document.createTextNode(msg)

divResposta.appendChild(txt);

btn.onclick = function () {

divResposta.innerHTML = "";

let media = (parseFloat(txtnum.value)+ parseFloat(txtnum2.value) + parseFloat(txtnum3.value))/3;

let resp = `A média do aluno corresponde a ${media.toFixed(1)}`;

let respostas = resp;

let p = document.createElement("p");

p.textContent = respostas;

divResposta.appendChild(p);

}

</script>

</body>

</html>

* + 1. 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>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px; /\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px; /\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start; /\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0; /\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="txtnum" placeholder="nome ">

<input type="text" id="txtnum2" placeholder="horas trabalhadas ">

<input type="text" id="txtnum3" placeholder="valor horas trabalhadas">

<input type="text" id="txtnum4" placeholder="quantia de filhos">

<button id="btn">Clique aqui!</button>

</div>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let txtnum3 = document.getElementById("txtnum3");

let txtnum4 = document.getElementById("txtnum4");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

let msg = "Digite o seu nome, o número de horas trabalhadas, o valor que recebe por horas trabalhadas e o número de filhos";

let txt = document.createTextNode(msg)

divResposta.appendChild(txt);

btn.onclick = function () {

divResposta.innerHTML = "";

let sal = parseFloat(txtnum2.value)\*parseFloat(txtnum3.value);

if (parseFloat(txtnum4.value) >= 3){

sal += 0.03 \* parseFloat(txtnum4.value) ;

}

let resp = `O salario do ${txtnum.value} corresponde a R$ ${sal.toFixed(2)}`;

let p = document.createElement("p");

p.textContent = resp;

divResposta.appendChild(p);

}

</script>

</body>

</html>

* + 1. 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>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px;

/\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px;

/\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start;

/\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0;

/\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="txtnum" placeholder="lado 1 ">

<input type="text" id="txtnum2" placeholder="lado 2 ">

<input type="text" id="txtnum3" placeholder="lado 3">

<button id="btn">Clique aqui!</button>

</div>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let txtnum3 = document.getElementById("txtnum3");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

let msg = "Digite a medida de cada lado do triangulo para verificar o tipo dele:";

let txt = document.createTextNode(msg)

divResposta.appendChild(txt);

let resp = '';

btn.onclick = function () {

divResposta.innerHTML = "";

let num1 = txtnum.value;

let num2 = txtnum2.value;

let num3 = txtnum3.value;

if (num1 == num2 && num1==num3) {

resp = 'equilatero';

}else if (num1 == num2 || num1 == num3 || num2== num3 ){

resp = 'isoceles';

}else{

resp = 'escaleno';

}

let p = document.createElement("p");

p.textContent = resp;

divResposta.appendChild(p);

}

</script>

</body>

</html>

* + 1. 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

Texto

Descrição gerada automaticamente

Escreva na tela de acordo com as seguintes situações:

Tabela

Descrição gerada automaticamente

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px;

/\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px;

/\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start;

/\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0;

/\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="txtnum" placeholder="peso ">

<input type="text" id="txtnum2" placeholder="altura ">

<button id="btn">Clique aqui!</button>

</div>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

let msg = "Digite a sua altura e peso para classificalo na tabela do imc:";

let txt = document.createTextNode(msg)

divResposta.appendChild(txt);

let resp = '';

btn.onclick = function () {

divResposta.innerHTML = "";

let num1 = txtnum.value;

let num2 = txtnum2.value;

$imc = num1 /( num2\*num2);

switch (true){

case ($imc < 20):

resp = 'abaixo do peso';

break;

case ($imc <25 && $imc>=20):

resp = 'peso normal';

break;

case ($imc >=25 && $imc<30):

resp = 'sobre peso';

break;

case ($imc >=30 && $imc<35):

resp = 'obeso';

break;

default:

resp = 'obeso morbido';

break;

}

let p = document.createElement("p");

p.textContent = resp;

divResposta.appendChild(p);

}

</script>

</body>

</html>

* + 1. 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>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px;

/\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px;

/\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start;

/\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0;

/\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="txtnum" placeholder="idade ">

<button id="btn">Clique aqui!</button>

</div>

<script>

let txtnum = document.getElementById("txtnum");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

let msg = "Digite a sua idade para classifica-lo";

let txt = document.createTextNode(msg)

divResposta.appendChild(txt);

let resp = '';

btn.onclick = function () {

divResposta.innerHTML = "";

let num1 = parseFloat(txtnum.value);

switch (true){

case (num1 <5):

resp = 'Sem classificação ';

break;

case (num1 >=5 && num1<=7):

resp = 'Infantil A';

break;

case (num1 >=8 && num1<=11):

resp = 'Infantil B ';

break;

case (num1 >=12 && num1<=13):

resp = 'Juvenil A';

break;

case (num1 >=14 && num1<=17):

resp = 'Juvenil B ';

break;

default:

resp = 'Adultos ';

break;

}

let p = document.createElement("p");

p.textContent = resp;

divResposta.appendChild(p);

}

</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 pelo usuário: ax2 + bx + c = 0, Δ= b2 - 4ac ,



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px;

/\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px;

/\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start;

/\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0;

/\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="a" placeholder="b ">

<input type="text" id="b" placeholder="a ">

<input type="text" id="c" placeholder="c ">

<button id="btn">Clique aqui!</button>

</div>

<script>

let text1= document.getElementById("a");

let text2= document.getElementById("b");

let text3= document.getElementById("c");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

let msg = "Digite os valores para a,b,c para o calculo de bhaskara:";

let txt = document.createTextNode(msg)

divResposta.appendChild(txt);

let resp = '';

btn.onclick = function () {

divResposta.innerHTML = "";

let a= text1.value;

let b=text2.value;

let c=text3 .value;

let delta = ((b\*b) - (4 \* a \* c))\*\*(1/2)

let x1 = (-b + delta ) / 2

let x2 = (-b - delta ) / 2

let msg = "as raizes dessa equacao correspondem a ${x1} e ${x2}";

}

let p = document.createElement("p");

p.textContent = resp;

divResposta.appendChild(p);

}

</script>

</body>

</html>

* + 1. 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:
  1. Para homens: (72.7h \* h) – 58
  2. 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>at helio</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: lavender;

height: 400px;

margin: 0;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px;

/\* Espaço entre o input/botão e a divResposta \*/

}

button {

margin-left: 10px;

/\* Espaço entre o input e o botão \*/

}

#divResposta {

width: 450px;

background-color: white;

border-radius: 9px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

align-items: flex-start;

/\* Alinha à esquerda \*/

justify-content: center;

margin-top: 40px;

padding: 20px;

}

p {

margin: 3px 0;

/\* Espaço entre os parágrafos \*/

}

</style>

</head>

<body>

<div id="divResposta"></div>

<div class="input-container">

<input type="text" id="txtnum" placeholder="sexo ">

<input type="text" id="txtnum2" placeholder="altura ">

<button id="btn">Clique aqui!</button>

</div>

<script>

let txtnum = document.getElementById("txtnum");

let txtnum2 = document.getElementById("txtnum2");

let divResposta = document.getElementById("divResposta");

let btn = document.getElementById("btn");

let msg = "Digite a sua altura e sexo para descobrir seu peso ideal:";

let txt = document.createTextNode(msg)

divResposta.appendChild(txt);

let resp = '';

btn.onclick = function () {

divResposta.innerHTML = "";

let num1 = txtnum.value;

let num2 = parseFloat(txtnum2.value);

let peso = 0;

if (num1.toLowerCase() === 'masculino') {

peso = (72.7 \* num2) - 58;

} else {

peso = (62.1 \* num2) - 44.7;

}

resp = `peso calculado e aproximadamente ${peso.toFixed(2)}kg`;

let p = document.createElement("p");

p.textContent = resp;

divResposta.appendChild(p);

}

</script>

</body>

</html>

=-=-=-=-=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=-=

=-=-=-=-=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=-=

* 1. Construa uma interface gráfica do usuário que acessa API com as seguintes páginas:
     1. Login do usuário.
     2. Interface para trocar senha.
     3. CRUD completo de usuário.

Index.php

<?php

require\_once("modelo/Router.php");

$roteador = new Router();

$roteador->get("/usuarios/readpage/(\d+)/", function($pagina){

require\_once("controle/controle\_usuario\_read\_page.php");

});

$roteador->get("usuarios/(\d+)", function($parametro\_idusuario){

require\_once("controle/controle\_usuario\_read\_by\_id.php");

});

$roteador->post("usuarios", function(){

require\_once("controle/controle\_usuario\_create.php");

});

$roteador->delete("usuarios/(\d+)", function($parametro\_idusuario){

require\_once("controle/controle\_usuario\_delete.php");

});

$roteador->put("usuarios/(\d+)", function($parametro\_idusuario){

require\_once("controle/controle\_usuario\_update.php");

});

$roteador->put("/usuarios/senha", function(){

require\_once("controle/controle\_usuario\_trocarSenha.php");

});

$roteador->post ("logar",function(){

require\_once("controle/controle\_usuario\_login.php");

});

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

require\_once("controle/controle\_usuario\_create\_from\_csv.php");

});

$roteador->run();

?>

**======== Controle ==========**

Controle\_usuario\_create

<?php

require\_once ("modelo/Banco.php");

require\_once ("modelo/MeuTokenJWT.php");

require\_once("modelo/Usuario.php");

$textoRecebido = file\_get\_contents("php://input");

$objJson = json\_decode($textoRecebido) or die ('{"msg": "formato incorreot!"}');

$objResposta = new stdClass();

$objUsuario = new Usuario();

$objUsuario->setNomeUsuario($objJson->nomeUsuario);

$objUsuario->setEmail($objJson->emailUsuario);

$objUsuario->setSenha($objJson->senhaUsuario);

try{

if ($objUsuario->getNomeUsuario()==''){

$objResposta->status = false;

$objResposta->msg = 'Nome esta em branco!';

}else if (strlen($objUsuario->getNomeUsuario()) < 4){

$objResposta->status = false;

$objResposta->msg = 'o nome nao pode ser menor que 4 caracterees!';

}else if ($objUsuario->create()==true){

$objResposta->status = true;

$objResposta->msg = 'sucesso!';

}else{

$objResposta-> status = false;

$objResposta->msg = "Erro ao cadastrar!";

}

}catch (Exception $e){

$objResposta-> status = false;

$objResposta->error =$e->getMessage();

die(json\_encode($objResposta));

}

if( $objResposta->status == true) {

header('HTTP/1.1 201');

}else {

header('HTTP/1.1 200');

}

echo json\_encode($objResposta);

?>

Controle\_usario\_delete.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once ("modelo/Banco.php");

require\_once ("modelo/MeuTokenJWT.php");

require\_once("modelo/Usuario.php");

$headers = apache\_request\_headers();

$objResposta = new stdClass();

$objToken = new MeuTokenJWT();

try{

if ($objToken->validarToken($headers['Authorization']) == true){

$objUsuario = new Usuario();

$objUsuario->setIdUsuario($parametro\_idusuario);

if ($objUsuario->delete()==true){

header('HTTP/1.1 204');

}else{

header('HTTP/1.1 200');

header('content-type : application/json');

}

}

}catch (Exception $e){

$objResposta->status = false;

$objResposta->error =$e->getMessage();

die(json\_encode($objResposta));

}

echo json\_encode($objResposta);

Controle\_usario\_login.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once ("modelo/Banco.php");

require\_once ("modelo/MeuTokenJWT.php");

require\_once("modelo/Usuario.php");

$textoRecebido = file\_get\_contents("php://input");

$objJson = json\_decode($textoRecebido) or die ('{"msg": "formato incorreot!"}');

$objResposta = new stdClass();

$objUsuario = new Usuario();

$objUsuario->setEmail($objJson->emailUsuario);

$objUsuario->setSenha($objJson->senhaUsuario);

try{

if ($objUsuario->verficarUsuarioSenha() == true){

$tokenjwt = new MeuTokenJWT();

$objClaimsToken = new stdClass();

$objClaimsToken->emailUsuario = $objUsuario->getEmail();

$objClaimsToken->senhaUsuario = $objUsuario->getSenha();

$novoToken = $tokenjwt->gerarToken($objClaimsToken);

$objResposta->status = true;

$objResposta->msg = 'Login efetuado com sucesso!';

$objResposta->token = $novoToken;

}else{

$objResposta->status = false;

$objResposta->msg = 'Usuario inexistente!';

}

}catch (Exception $e){

$objResposta-> status = false;

$objResposta->error =$e->getMessage();

die(json\_encode($objResposta));

}

echo json\_encode($objResposta);

Controle\_usuario\_read\_by\_id.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once ("modelo/Banco.php");

require\_once ("modelo/MeuTokenJWT.php");

require\_once("modelo/Usuario.php");

$headers = apache\_request\_headers();

$objResposta = new stdClass();

$objToken = new MeuTokenJWT();

try{

if ($objToken->validarToken($headers['Authorization']) == true){

$objUsuario = new Usuario();

$objUsuario->setIdUsuario($parametro\_idusuario);

$objResposta->status = true;

$objResposta->msg = 'executado com sucesso!';

$objResposta->dados = $objUsuario->ReadById();

}else{

$objResposta->status = false;

$objResposta->msg = 'token invalido';

}

}catch (Exception $e){

$objResposta->status = false;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

}

echo json\_encode($objResposta);

header("HTTP/1.1 200");

// Define o tipo de conteúdo da resposta como JSON

header("Content-Type: application/json");

Controle\_usuario\_readPage.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once ("modelo/Banco.php");

require\_once ("modelo/MeuTokenJWT.php");

require\_once("modelo/Usuario.php");

$headers = apache\_request\_headers();

$objResposta = new stdClass();

$objToken = new MeuTokenJWT();

try{

if ($objToken->validarToken($headers['Authorization'])== true){

$objUsuario = new Usuario();

$objResposta->status = true;

$objResposta->msg = 'executado com sucesso!';

$objResposta->dados = $objUsuario->ReadbyPage($pagina) ;

}else {

$objResposta->status = false;

$objResposta->msg = 'token invalido';

}

}catch (Exception $e){

$objResposta->status = true;

$objResposta->msg = 'executado com sucesso!';

$objResposta->erro = $e->getMessage();

}

echo json\_encode($objResposta);

header("HTTP/1.1 200");

// Define o tipo de conteúdo da resposta como JSON

header("Content-Type: application/json");

Controle\_usuario\_Trocarsenha.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once ("modelo/Banco.php");

require\_once ("modelo/MeuTokenJWT.php");

require\_once("modelo/Usuario.php");

$textoRecebido = file\_get\_contents("php://input");

$objJson = json\_decode($textoRecebido) or die ('{"msg": "formato incorreot!"}');

$objResposta = new stdClass();

$objToken = new MeuTokenJWT();

try{

$objUsuario = new Usuario();

$objUsuario->setEmail($objJson->emailUsuario);

$objUsuario->setSenha($objJson->senhaUsuario);

if ($objUsuario->verificarEmail() == true){

if ($objUsuario->updateSenha()==true){

$objResposta->status = true;

$objResposta->msg = 'sucesso ao trocar a senha!';

}else {

$objResposta->status = false;

$objResposta->msg = 'email invalido';

}

}

}catch (Exception $e){

$objResposta-> status = false;

$objResposta->error =$e->getMessage();

die(json\_encode($objResposta));

}

echo json\_encode($objResposta);

Controle\_usuario\_update.php

<?php

use Firebase\JWT\MeuTokenJWT;

require\_once ("modelo/Banco.php");

require\_once ("modelo/MeuTokenJWT.php");

require\_once("modelo/Usuario.php");

$textoRecebido = file\_get\_contents("php://input");

$objJson = json\_decode($textoRecebido) or die ('{"msg": "formato incorreot!"}');

$headers = apache\_request\_headers();

$objResposta = new stdClass();

$objUsuario = new Usuario();

$objToken = new MeuTokenJWT();

$objUsuario->setIdUsuario($parametro\_idusuario);

$objUsuario->setNomeUsuario($objJson->nomeUsuario);

$objUsuario->setEmail($objJson->emailUsuario);

$objUsuario->setSenha($objJson->senhaUsuario);

try{

if ($objToken->validarToken($headers['Authorization'])== true){

if ($objUsuario->getNomeUsuario()==''){

$objResposta->status = false;

$objResposta->msg = 'Nome esta em branco!';

}else if ($objUsuario->getNomeUsuario() < 4){

$objResposta->status = false;

$objResposta->msg = 'o nome nao pode ser menor que 4 caracterees!';

}else if ($objUsuario->Update()==true){

$objResposta->status = true;

$objResposta->msg = 'sucesso!';

}else{

$objResposta-> status = false;

$objResposta->msg = "Erro ao atualizar!";

}

}else{

$objResposta->status = false;

$objResposta->msg = 'token invalido';

}

}catch (Exception $e){

$objResposta-> status = false;

$objResposta->error =$e->getMessage();

die(json\_encode($objResposta));

}

if( $objResposta->status == true) {

header('HTTP/1.1 201');

}else {

header('HTTP/1.1 200');

}

echo json\_encode($objResposta);

?>

**============== Modelo ===============**

Banco.php

<?php

class Banco {

private static $HOST='127.0.0.1';

private static $USER='root';

private static $PWD='root123';

private static $DB='app\_user';

private static $PORT=3306;

private static $CONEXAO = null;

private static function conectar(){

error\_reporting(E\_ERROR | E\_PARSE);

try{

if(Banco::$CONEXAO==null){

Banco::$CONEXAO= new mysqli(Banco::$HOST,Banco::$USER,Banco::$PWD,Banco::$DB,Banco::$PORT);

if(Banco::$CONEXAO->connect\_error){

$objResposta =new stdClass();

$objResposta->cod= 1;

$objResposta->msg = 'eror ao conectar no banco!!';

$objResposta->erro = Banco::$CONEXAO->connect\_error;

die(json\_encode($objResposta));

}

}

} catch (Exception $e) {

// Em caso de qualquer outra exceção, trata normalmente

$objResposta = new stdClass();

$objResposta->cod = 1;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

} catch (Error $e) {

$objResposta = new stdClass();

$objResposta->cod = 2;

$objResposta->erro = $e->getMessage();

die(json\_encode($objResposta));

}

}

public static function getConexao(){

if(Banco::$CONEXAO==null){

Banco::conectar();

}

return Banco::$CONEXAO;

}

}

?>

Usuario.php

<?php

require\_once("modelo/Banco.php");

class Usuario implements JsonSerializable {

private $idUsuario;

private $nomeUsuario;

private $senhaUsuario;

private $emailUsuario;

private $tokenUsuario;

private $nascimento;

public function jsonSerialize() {

$obj = new stdClass();

$obj->idUsuario= $this->getIdUsuario();

$obj->nomeUsuario = $this->getNomeUsuario();

$obj->senhaUsuario = $this->getsenha();

$obj->emailUsuario = $this->getEmail();

}

public function verficarUsuarioSenha () {

$conexao = Banco::getConexao();

$sql = "SELECT count(\*) AS qtd, emailUsuario, senhaUsuario FROM app\_user WHERE emailUsuario = ? AND senhaUsuario = md5(?) GROUP BY emailusuario, senhaUsuario";

$prepararsql = $conexao->prepare($sql);

$prepararsql->bind\_param("ss", $this->emailUsuario, $this->senhaUsuario);

$prepararsql ->execute();

$matriz = $prepararsql -> get\_result();

$objTupla = $matriz -> fetch\_object();

return $objTupla->qtd > 0;

}

public function verificarEmail () {

$conexao = Banco::getConexao();

$sql = "SELECT count(\*) AS qtd, emailUsuario FROM app\_user WHERE emailUsuario = ? GROUP BY emailusuario";

$prepararsql = $conexao->prepare($sql);

$prepararsql->bind\_param("s", $this->emailUsuario);

$prepararsql ->execute();

$matriz = $prepararsql -> get\_result();

$objTupla = $matriz -> fetch\_object();

return $objTupla->qtd > 0;

}

public function create (){

$conexao = Banco::getConexao() ;

$this->senhaUsuario = md5($this->senhaUsuario);

$sql = "insert into app\_user (nomeUsuario,emailUsuario,senhaUsuario) values (?,?,?)";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("sss",$this->nomeUsuario,$this->emailUsuario,$this->senhaUsuario);

$executou = $prepararsql->execute();

if (!$executou) {

throw new Exception("Erro ao executar a consulta SQL");

}

$idCadastrado = $conexao->insert\_id;

$this->setIdUsuario($idCadastrado);

return $executou;

}

public function createFromCsv (){

$conexao = Banco::getConexao() ;

$this->senhaUsuario = md5($this->nascimento);

$sql = "insert into usuario (nomeUsuario,emailUsuario,senhaUsuario) values (?,?,?)";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("sss",$this->nomeUsuario,$this->emailUsuario,$this->senhaUsuario);

$executou = $prepararsql->execute();

if (!$executou) {

throw new Exception("Erro ao executar a consulta SQL");

}

$idCadastrado = $conexao->insert\_id;

$this->setIdUsuario($idCadastrado);

return $executou;

}

public function IsUsuario (){

$conexao = Banco::getConexao() ;

$sql = "select count(\*) as qtd from usuario where emailUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("s",$this->emailUsuario);

$executou = $prepararsql->execute();

if (!$executou) {

throw new Exception("Erro ao executar a consulta SQL");

}

$matriz = $prepararsql->get\_result();

$objTupla = $matriz->fetch\_object();

return $objTupla->qtd > 0;

}

public function delete (){

$conexao = Banco::getConexao() ;

$sql = "delete from app\_user where idUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("i",$this->idUsuario);

return $prepararsql->execute();

}

public function update (){

$conexao = Banco::getConexao() ;

$this->senhaUsuario = md5($this->senhaUsuario);

$sql = "update app\_user set nomeUsuario=?,emailUsuario=?,senhaUsuario=? where idUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("sssi",$this->nomeUsuario,$this->emailUsuario,$this->senhaUsuario,$this->idUsuario);

return $prepararsql->execute();

}

public function updateSenha (){

$conexao = Banco::getConexao() ;

$this->senhaUsuario = md5($this->senhaUsuario);

$sql = "update app\_user set senhaUsuario = md5(?) where emailUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("ss",$this->senhaUsuario,$this->emailUsuario);

return $prepararsql->execute();

}

public function ReadById (){

$conexao = Banco::getConexao() ;

$sql = "select \* from app\_user where idUsuario = ?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("i", $this->idUsuario);

$prepararsql->execute();

if (!$prepararsql->execute()) {

throw new Exception("Erro ao executar a consulta SQL");

}

$matrizTuplas = $prepararsql->get\_result();

$matrizTuplas = $matrizTuplas ->fetch\_all(MYSQLI\_ASSOC);

return $matrizTuplas;

}

public function ReadByPage ($pagina){

$conexao = Banco::getConexao() ;

$itensPorPagina = 5;

$inicio = ($pagina - 1) \* $itensPorPagina;

$sql = "select \* from app\_user limit ?,?";

$prepararsql = $conexao->prepare($sql);

if (!$prepararsql) {

throw new Exception("Erro ao preparar a consulta SQL");

}

$prepararsql->bind\_param("ii",$inicio,$itensPorPagina);

$executou = $prepararsql->execute();

if (!$executou) {

throw new Exception("Erro ao executar a consulta SQL");

}

$matriz = $prepararsql->get\_result();

$matriz = $matriz->fetch\_all(MYSQLI\_ASSOC);

return $matriz;

}

public function getTokenUsuario ( $token ){

return $this->tokenUsuario;

}

public function setTokenUsuario( $token ){

$this->tokenUsuario = $token;

}

public function setIdUsuario($idUsuario) {

$this->idUsuario = $idUsuario;

}

public function getIdUsuario() {

return $this->idUsuario;

}

public function setNomeUsuario($nome) {

$this->nomeUsuario = $nome;

}

public function getNomeUsuario() {

return $this->nomeUsuario;

}

public function setSenha($senha) {

$this->senhaUsuario = $senha;

}

public function getSenha() {

return $this->senhaUsuario;

}

public function setEmail($email) {

$this->emailUsuario = $email;

}

public function getEmail() {

return $this->emailUsuario;

}

public function setNascimento($nascimento) {

$this->nascimento = $nascimento;

}

public function getNascimento() {

return $this->nascimento;

}

}

?>

MeuTokenJWT.php

<?php

namespace Firebase\JWT;

use stdClass;

use Firebase\JWT\Key;

use Firebase\JWT\JWT;

use DomainException;

use Exception;

use InvalidArgumentException;

use UnexpectedValueException;

use Firebase\JWT\SignatureInvalidException;

use Firebase\JWT\ExpiredException;

require\_once "jwt/JWT.php";

require\_once "jwt/Key.php";

require\_once "jwt/SignatureInvalidException.php";

require\_once "jwt/ExpiredException.php";

class MeuTokenJWT

{

//chave de criptografia, defina uma chave forte e a mantenha segura.

private $key = "x9S4q0v+V0IjvHkG20uAxaHx1ijj+q1HWjHKv+ohxp/oK+77qyXkVj/l4QYHHTF3";

//algoritmo de criptografia para assinatura

//Suportados: 'HS256' , 'ES384','ES256', 'ES256K', ,'HS384', 'HS512', 'RS256', 'RS384'

private $alg = 'HS256';

private $type = 'JWT';

private $iss = '<http://localhost>'; //emissor do token

private $aud = '<http://localhost>'; //destinatário do token

private $sub = "acesso\_sistema"; //assunto do token

private $iat = ""; //momento de emissão

private $exp = ""; //momento de expiração

private $nbf = ""; //não é válido antes do tempo especificado

private $jti = ""; //Identificador único

private $payload; //claims

//tempo de validade do token

private $duracaoToken = 3600 \* 24 \* 30; //3600 segundos = 60 min

public function gerarToken($parametro\_claims)

{

// Criação dos headers como objeto da classe stdClass

$objHeaders = new stdClass();

$objHeaders->alg = $this->alg;

$objHeaders->typ = $this->type;

// Criação do payload como objeto da classe stdClass

$objPayload = new stdClass();

//Registered Claims

$objPayload->iss = $this->iss; // emissor do token

$objPayload->aud = $this->aud; // destinatário do token

$objPayload->sub = $this->sub; // assunto do token

$objPayload->iat = time(); // momento de criação do token

$objPayload->exp = time() + $this->duracaoToken; // momento de expiração = tempo atual + duração

$objPayload->nbf = time(); // momento em que o token torna-se valido.

$objPayload->jti = bin2hex(random\_bytes(16)); // gera um valor aleatório para jti;

//Public Claims

$objPayload->emailUsuario = $parametro\_claims->emailUsuario;

$objPayload->senhaUsuario = $parametro\_claims->senhaUsuario;

//Private claims

// Utiliza a biblioteca do Firebase para gerar o token com os parâmetros

$token = JWT::encode((array) $objPayload, $this->key, $this->alg, null, (array) $objHeaders);

return $token;

}

public function validarToken($stringToken)

{

if (isset($stringToken)) {

if ($stringToken == "") {

return false;

} else {

$remover = ["Bearer ", " "];

$token = str\_replace($remover, "", $stringToken);

try {

$payloadValido = JWT::decode($token, new Key($this->key, $this->alg));

$this->setPayload($payloadValido);

return true;

} catch (SignatureInvalidException $e) {

// A assinatura do token é inválida.

//error\_log("Invalid token signature: " . $e->getMessage());

return false;

} catch (BeforeValidException $e) {

// O token não é válido ainda (antes do tempo 'nbf').

//error\_log("Token not valid yet: " . $e->getMessage());

return false;

} catch (ExpiredException $e) {

// O token expirou.

// error\_log("Token expired: " . $e->getMessage());

return false;

} catch (InvalidArgumentException $e) {

// Argumento inválido passado.

//error\_log("Invalid argument: " . $e->getMessage());

return false;

} catch (DomainException $e) {

// Exceção de domínio genérica.

//error\_log("Domain exception: " . $e->getMessage());

return false;

} catch (UnexpectedValueException $e) {

// Valor inesperado encontrado.

//error\_log("Unexpected value: " . $e->getMessage());

return false;

} catch (Exception $e) {

// Qualquer outra exceção genérica.

//error\_log("General exception: " . $e->getMessage());

return false;

}

}

}

return false;

}

/\*\*

\* Get the value of payload

\*/

public function getPayload()

{

return $this->payload;

}

/\*\*

\* Set the value of payload

\*

\* @return self

\*/

public function setPayload($payload)

{

$this->payload = $payload;

return $this;

}

/\*\*

\* Get the value of alg

\*/

public function getAlg()

{

return $this->alg;

}

/\*\*

\* Set the value of alg

\*

\* @return self

\*/

public function setAlg($alg)

{

$this->alg = $alg;

return $this;

}

}

**============== View ===============**

AlterarSenha.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>

<style>

body{

background-color: #363333;

display: flex;

flex-direction: column;

font-family: Arial, Helvetica, sans-serif;

justify-content: center;

align-items: center;

height: 100%;

margin: 0;

}

#divResposta{

background-color:transparent;

border-radius: 9px;

max-width: 450px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

margin-top: 40px;

padding: 30px;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px; /\* Espaço entre o input/botão e a divResposta \*/

flex-direction: column;

}

.input-container input {

margin: 5px 0; /\* Espaço entre os inputs \*/

padding: 10px;

width: 400px; /\* Largura dos inputs \*/

border: 1px solid #ccc;

border-radius: 5px;

}

h1 {

color : white;

}

.input-container button {

margin-top: 10px; /\* Espaço acima do botão \*/

padding: 10px 20px;

border: none;

border-radius: 20px;

color: white;

font-size: 16px;

background-color: rgb(111, 163, 163);

cursor: pointer;

}

p{

color: rgb(193, 180, 180);

}

</style>

</head>

<body>

<div id="divResposta">

<h1>Recuperacao de senha</h1>

<div class="input-container">

<input type="text" id="txtEmail" placeholder="Email">

<input type="password" id="txtSenha" placeholder="Senha">

<button id="bntCadastrar">Cadastrar!</button>

</div>

</div>

</body>

<script>

let txtEmail = document.getElementById('txtEmail');

let txtSenha = document.getElementById('txtSenha');

let btnOk = document.getElementById("bntCadastrar");

btnOk.onclick = function () {

alterarSenha ();

}

function alterarSenha (){

let emailUsuario = txtEmail.value;

let senhaUsuario = txtSenha.value;

let json = {

"emailUsuario":emailUsuario,

"senhaUsuario":senhaUsuario

}

let uri = "/usuarios/senha";

fetch (uri,{

method: 'put',

body: JSON.stringify(json)

}).then ( (response) =>{

return response.text();

}).then((data) =>{

console.log(data)

}).catch((error)=>{

console.error("Error", error)

});

}

</script>

</html>

Cadastro.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>

<style>

body {

background-color: #363333;

display: flex;

flex-direction: column;

font-family: Arial, Helvetica, sans-serif;

justify-content: center;

align-items: center;

height: 100%;

margin: 0;

}

#divResposta {

background-color: transparent;

border-radius: 9px;

max-width: 450px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

margin-top: 40px;

padding: 30px;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px;

/\* Espaço entre o input/botão e a divResposta \*/

flex-direction: column;

}

.input-container input {

margin: 5px 0;

/\* Espaço entre os inputs \*/

padding: 10px;

width: 400px;

/\* Largura dos inputs \*/

border: 1px solid #ccc;

border-radius: 5px;

}

h1 {

color: white;

}

.input-container button {

margin-top: 10px;

/\* Espaço acima do botão \*/

padding: 10px 20px;

border: none;

border-radius: 20px;

color: white;

font-size: 16px;

background-color: rgb(111, 163, 163);

cursor: pointer;

margin-bottom: 30px;

}

p {

color: rgb(193, 180, 180);

}

</style>

</head>

<body>

<div id="divResposta">

<h1>Create account</h1>

<div class="input-container">

<input type="text" id="txtNome" placeholder="Nome">

<input type="text" id="txtEmail" placeholder="Email">

<input type="password" id="txtSenha" placeholder="Senha">

<button id="bntCadastrar">Cadastrar!</button>

<a href="login.html"> Ja possui um cadastro? </a>

</div>

</div>

</body>

<script>

const txtNome = document.getElementById("txtNome");

const txtEmail = document.getElementById("txtEmail");

const txtSenha = document.getElementById("txtSenha");

const btnOk = document.getElementById("bntCadastrar");

btnOk.onclick = function(){

cadastrar() ;

}

function cadastrar() {

const nome = txtNome.value;

const email = txtEmail.value;

const senha = txtSenha.value;

const jsonEnvio = {

"nomeUsuario": nome,

"emailUsuario": email,

"senhaUsuario": senha

}

let uri = "/usuarios";

fetch(uri, {

method: "post",

body: JSON.stringify(jsonEnvio)

}).then((response) => {

return response.text();

}).then((data) => {

console.log(data)

}).catch((error) => {

console.error("Error:", error);

});

}

</script>

</html>

Login.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>

<style>

body {

background-color: #363333;

display: flex;

flex-direction: column;

font-family: Arial, Helvetica, sans-serif;

justify-content: center;

align-items: center;

height: 100%;

margin: 0;

}

#divResposta {

background-color: transparent;

border-radius: 9px;

max-width: 450px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

margin-top: 40px;

padding: 30px;

}

.input-container {

display: flex;

align-items: center;

margin-top: 20px;

/\* Espaço entre o input/botão e a divResposta \*/

flex-direction: column;

}

.input-container input {

margin: 5px 0;

/\* Espaço entre os inputs \*/

padding: 10px;

width: 400px;

/\* Largura dos inputs \*/

border: 1px solid #ccc;

border-radius: 5px;

}

h1 {

color: white;

}

.input-container button {

margin-top: 10px;

/\* Espaço acima do botão \*/

padding: 10px 20px;

border: none;

border-radius: 20px;

color: white;

font-size: 16px;

background-color: rgb(111, 163, 163);

cursor: pointer;

margin-bottom: 30px;

}

p {

color: rgb(193, 180, 180);

}

</style>

</head>

<body>

<div id="divResposta">

<h1>Login</h1>

<div class="input-container">

<input type="text" id="txtEmail" placeholder="Email">

<input type="password" id="txtSenha" placeholder="Senha">

<button id="bntCadastrar">Login!</button>

<a href="cadastro.html"> Nao possui um cadastro? </a>

<p> ================== ou ==================</p>

<a href="alterarSenha.html"> Esqueceu a senha? </a>

</div>

</div>

</body>

<script>

const txtEmail = document.getElementById("txtEmail");

const txtSenha = document.getElementById("txtSenha");

const btnOk = document.getElementById("bntCadastrar");

btnOk.onclick = function(){

cadastrar() ;

}

function cadastrar() {

const email = txtEmail.value;

const senha = txtSenha.value;

const jsonEnvio = {

"emailUsuario": email,

"senhaUsuario": senha

}

let uri = "/logar";

fetch(uri, {

method: "post",

body: JSON.stringify(jsonEnvio)

}).then((response) => {

return response.text();

}).then((data) => {

console.log(data)

}).catch((error) => {

console.error("Error:", error);

});

}

</script>

</html>

**============== Docs ===============**

Banco.sql

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema app\_user

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema app\_user

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `app\_user` DEFAULT CHARACTER SET utf8mb4 ;

USE `app\_user` ;

-- -----------------------------------------------------

-- Table `app\_user`.`app\_user`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `app\_user`.`app\_user` (

`idUsuario` INT(11) NOT NULL AUTO\_INCREMENT,

`nomeUsuario` VARCHAR(65) NULL DEFAULT NULL,

`emailUsuario` VARCHAR(65) NULL DEFAULT NULL,

`senhaUsuario` VARCHAR(56) NULL DEFAULT NULL,

PRIMARY KEY (`idUsuario`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;