

# HITO 2 DEL 2º TRIMESTRE DE PROGRAMACIÓN

Rodrigo García Ortiz

14 - 2 – 2025

# Índice

## Contenido

conexión.php.....	1
index.php.....	1
class_usuario.php.....	2
usuariosController.php .....	2
registrar_usuario.php.....	3
login.php.....	4
class_tarea.php .....	6
tareasController.php .....	7
lista_tareas.php.....	8
alta_tarea.php.....	9
completar_tarea.php .....	11
eliminar_tarea.php.....	11
logout.php.....	11
tareas.sql .....	11
Webgrafía .....	13

## conexión.php

```
<?php
class Conexion {
    private $servidor = 'localhost';
    private $usuario = 'root';
    private $password = 'curso';
    private $base_datos = 'tareas';
    public $conexion;

    public function __construct() {
        $this->conexion = new mysqli($this->servidor, $this->usuario,
$this->password, $this->base_datos);

        if ($this->conexion->connect_error) {
            die("Error de conexión: " . $this->conexion->connect_error);
        }
    }

    public function cerrar() {
        $this->conexion->close();
    }
}
?>
```

## index.php

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min
.css">
</head>
<body>
    <h1>Inicio</h1>

    <div class="container mt-4">

        <div class="mb-3">
            <a href="login.php" class = "btn btn-primary">Iniciar
sesión</a> <!-- Elige con que usuario iniciar sesión -->
        </div>
    </div>
```

```

        <a href="vista/registrar_usuario.php" class = "btn btn-
primary">Registrarse</a> <!-- Registra un nuevo usuario -->
    </div>

</div>
</body>
</html>

```

## class\_usuario.php

```

<?php
require_once '../config/conexion.php';

class Usuario {
    private $conexion;

    public function __construct() {
        $this->conexion = new Conexion();
    }

    public function registrarUsuario($nombre, $correo, $contrasena) {
        //Guarda los datos del usuario en la base de datos
        $query = "INSERT INTO usuarios (nombre, correo, contrasena)
VALUES (?, ?, ?)";
        $stmt = $this->conexion->conexion->prepare($query);
        $stmt->bind_param("sss", $nombre, $correo, $contrasena);

        if ($stmt->execute()) {
            header("Location: ../index.php"); //Al terminar regresa al
indice
        } else {
            echo "Error al registrar usuario: " . $stmt->error;
        }

        $stmt->close();
    }
}
?>

```

## usuariosController.php

```

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

class UsuariosController {
    private $modelo;

```

```

    public function __construct() {
        $this->modelo = new Usuario();
    }

    public function registrarUsuario($nombre, $correo, $contrasena) {
        //Guarda los datos del usuario en la base de datos
        $this->modelo->registrarUsuario($nombre, $correo, $contrasena);
    }
}
?>

```

## registrar\_usuario.php

```

<?php
require_once '../controlador/usuariosController.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') { //Guarda los datos recibidos
    en la base de datos
    $controller = new UsuariosController();
    $controller->registrarUsuario(
        $_POST['usuario'],
        $_POST['correo'],
        password_hash($_POST['contrasena'], PASSWORD_DEFAULT),
    );
    exit();
}
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min
.css">
    <title>Registrar Usuario</title>
</head>
<body>
    <div class="container mt-4">
        <h1>Registrar Usuario</h1>
        <form method = "POST">

            <div class="mb-3">
                <label for="usuario" class="form-label">Usuario</label>
<!-- Guarda el nombre del usuario -->

```

```

        <input type="text" class="form-control" id="usuario"
name="usuario" required>
    </div>

    <div class="mb-3">
        <label for="correo" class="form-label">Correo</label> <!--
- Guarda el correo del usuario -->
        <input type="email" class="form-control" id="correo"
name="correo" required>
    </div>

    <div class="mb-3">
        <label for="contrasena" class="form-
label">Contraseña</label> <!-- Guarda la contraseña del usuario -->
        <input type="password" class="form-control"
id="contrasena" name="contrasena" required>
    </div>

    <div class="mb-3">
        <label><h5>He leído y acepto las políticas de
privacidad</h5></label><br> <!-- Acepta las políticas de privacidad -->
        <label>Sí</label>
        <input type="radio" id="privacidad" name="privacidad"
value="si">
        <br>
        <label>No</label>
        <input type="radio" id="privacidad" name="privacidad"
value="no">
    </div>

    <button type="submit" class="btn btn-
primary">Registrar</button>
</form>
</div>
</body>
</html>

```

## login.php

```

<?php
session_start();
require_once 'config/conexion.php';
$conexion = new Conexion();
$query = "SELECT * FROM usuarios";
$resultado = $conexion->conexion->query($query);
$usuarios = [];
while ($fila = $resultado->fetch_assoc()) {
    $usuarios[] = $fila;
}

```

```

    }

foreach ($usuarios as $usuario):

    if ($_POST['usuario'] == $usuario['nombre']) { //Si la contraseña y
el usuario están en la base de datos permite acceder a la página

        if ($_POST['contrasena'] == $usuario['contrasena']){

            $_SESSION['usuario'] = $usuario['nombre']; //El usuario
introducido se vuelve el usuario de la sesión
            $_SESSION['id'] = $usuario['id_usuario']; //El id del usuario
introducido se vuelve el id de la sesión
            header("Location: vista/lista_tareas.php");

        } else {

            echo "Contraseña incorrecta.";
        }
    } else {

        echo "Usuario incorrecto.";
    }
}

endforeach
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min
.css">
</head>
<body>
    <div class="container mt-4">
        <h1>Iniciar sesión</h1>
        <form method = "POST">

            <div class="mb-3">
                <label for="usuario" class="form-label">Usuario</label>
<!-- Recibe el nombre del usuario a utilizar -->
                <input type="text" class="form-control" id="usuario"
name="usuario" required>
            </div>

```

```

        <div class="mb-3">
            <label for="contrasena" class="form-
label">Contraseña</label> <!-- Recibe la contraseña del usuario a
utilizar -->
            <input type="password" class="form-control"
id="contrasena" name="contrasena">
        </div>

        <button type="submit" class="btn btn-primary">Iniciar
sesión</button>
    </form>
</div>
</body>
</html>

```

## class\_tarea.php

```

<?php
require_once '../config/conexion.php';

class Tarea {
    private $conexion;

    public function __construct() {
        $this->conexion = new Conexion();
    }

    public function agregarTarea($usuario, $nombre, $estado) { //Guarda
los datos de la tarea en la base de datos
        $query = "INSERT INTO tarea (usuario, nombre, estado) VALUES (?,
?, ?)";
        $stmt = $this->conexion->conexion->prepare($query);
        $stmt->bind_param("sss", $usuario, $nombre, $estado);

        if ($stmt->execute()) {
            header("Location: ../vista/lista_tareas.php"); //Al terminar
regresa a la lista de tareas
        } else {
            echo "Error al agregar tarea: " . $stmt->error;
        }

        $stmt->close();
    }

    public function obtenerTareas($usuario) { //Recoge los datos de todas
las tareas del usuario en uso
        $query = "SELECT * FROM tarea where usuario = ?";
        $stmt = $this->conexion->conexion->prepare($query);

```



```

        $stmt->bind_param("i", $usuario);
        $stmt->execute();
        $resultado = $stmt->get_result();
        $tareas = [];
        while ($fila = $resultado->fetch_assoc()) {
            $tareas[] = $fila;
        }
        return $tareas;
    }

    public function completarTarea($id_tarea, $estado) { //Marca la tarea
    como completada
        $query = "UPDATE tarea SET estado = ? WHERE id_tarea = ?";
        $stmt = $this->conexion->conexion->prepare($query);
        $stmt->bind_param("si", $estado, $id_tarea);
        if ($stmt->execute()) {
            header("Location: ../vista/lista_tareas.php"); //Al terminar
    regresa a la lista de tareas
        } else {
            echo "Error al actualizar la tarea: " . $stmt->error;
        }

        $stmt->close();
    }

    public function eliminarTarea($id_tarea) { //Elimina los datos de la
    tarea indicada
        $query = "DELETE FROM tarea WHERE id_tarea = ?";
        $stmt = $this->conexion->conexion->prepare($query);
        $stmt->bind_param("i", $id_tarea);

        if ($stmt->execute()) {
            header("Location: ../vista/lista_tareas.php"); //Al terminar
    regresa a la lista de tareas
        } else {
            echo "Error al eliminar la tarea: " . $stmt->error;
        }

        $stmt->close();
    }
}
?>

```

## tareasController.php

```

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

```

```

class TareasController {
    private $modelo;

    public function __construct() {
        $this->modelo = new Tarea();
    }

    public function agregarTarea($usuario, $nombre, $estado) { //Guarda
los datos de la tarea en la base de datos
        $this->modelo->agregarTarea($usuario, $nombre, $estado);
    }

    public function obtenerTareas($usuario) { //Recoge los datos de las
tareas del usuario indicado
        return $this->modelo->obtenerTareas($usuario);
    }

    public function completarTarea($id_tarea, $estado) { //Marca la tarea
como completada
        $this->modelo->completarTarea($id_tarea, $estado);
    }

    public function eliminarTarea($id_tarea) { //Elimina los datos de la
tarea indicada
        $this->modelo->eliminarTarea($id_tarea);
    }
}
?>

```

## lista\_tareas.php

```

<?php
session_start();
if ($_SESSION['usuario'] == "") {

    header("Location: ../index.php");
}

$usuario = $_SESSION['id'];
require_once '../controlador/tareasController.php';
$controller = new TareasController();
$tareas = $controller->obtenerTareas($usuario);
?>

<!DOCTYPE html>
<html lang="es">
<head>

```

```

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min
.css">
    <title>Listado de Tareas</title>
</head>
<body>
    <h1>Tareas</h1>
    <table border="1" class = "table table-striped"> <!-- Muestra las
tareas del usuario elegido -->
        <tr>
            <th>Usuario</th>
            <th>Nombre</th>
            <th>Estado</th>
            <th>Opciones</th>
        </tr>
        <?php foreach ($tareas as $tarea): ?>
            <tr>
                <td><?= $tarea['usuario'] ?></td>
                <td><?= $tarea['nombre'] ?></td>
                <td><?= $tarea['estado'] ?></td>
                <td>
                    <?php if ($tarea['estado'] == "incompleta") { ?>
                        <a href="completar_tarea.php?id=<?=
$tarea['id_tarea'] ?>" class = "btn btn-primary">Marcar como
completada</a>
                    <?php } else {} ?>
                        <a href="eliminar_tarea.php?id=<?= $tarea['id_tarea']
?>" class = "btn btn-primary">Eliminar</a>
                    </td>
            </tr>
        <?php endforeach; ?>
    </table>
    <br>
    <a href="alta_tarea.php" class = "btn btn-primary">Agregar una nueva
tarea</a>
    <a href=" ../logout.php" class = "btn btn-primary">Cerrar sesión</a>
</body>
</html>

```

## alta\_tarea.php

```

<?php
session_start();
require_once ' ../controlador/tareasController.php';

```

```

if ($_SERVER['REQUEST_METHOD'] == 'POST') { //Guarda la nueva tarea en la
base de datos
    $controller = new TareasController();
    $controller->agregarTarea(
        $_SESSION['id'],
        $_POST['nombre'],
        $_POST['estado'],
    );
    exit();
}
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min
.css">
    <title>Agregar Tarea</title>
</head>
<body>
    <div class="container mt-4">
        <h1>Agregar Tarea</h1>
        <form method = "POST">

            <div class="mb-3">
                <label for="nombre" class="form-label">Nombre</label> <!--
- Guarda el nombre de la tarea -->
                <input type="text" class="form-control" id="nombre"
name="nombre" required>
            </div>

            <div class="mb-3">
                <label for="estado" class="form-label">Estado</label> <!--
- Guarda el estado de la tarea -->
                <select class="form-control" id="estado" name="estado">
                    <option value="completada">Completada</option>
                    <option value="incompleta">Incompleta</option>
                </select>
            </div>

            <button type="submit" class="btn btn-
primary">Agregar</button>
        </form>
    </div>
</body>

```

```
</html>
```

### completar\_tarea.php

```
<?php
session_start();

require_once '../controlador/tareasController.php';
$controller = new TareasController();

$id_tarea = $_GET['id']; //Obtiene el id de la tarea
$estado = "completada";
$resultado = $controller->completarTarea($id_tarea, $estado); //Cambia el
estado de la tarea a completado
?>
```

### eliminar\_tarea.php

```
<?php
session_start();

require_once '../controlador/tareasController.php';
$controller = new TareasController();

$id_tarea = $_GET['id']; //Obtiene el id de la tarea
$controller->eliminarTarea($id_tarea) //Elimina la tarea indicada
?>
```

### logout.php

```
<?php //Cierra sesión y vuelve al inicio
session_start();
session_unset();
session_destroy();
header("Location: index.php");
?>
```

### tareas.sql

```
create database tareas;
use tareas;

create table usuarios(
id_usuario int primary key auto_increment,
nombre varchar(50),
correo varchar(50) unique,
```

```
contrasena varchar(100)
);

create table tarea(
id_tarea int primary key auto_increment,
usuario int,
nombre varchar(50),
estado varchar(50),
foreign key (usuario) references usuarios (id_usuario)
);

insert into usuarios (nombre, correo, contrasena) values ('Rodrigo',
'rodrigo@ejemplo.com', '1234');
insert into usuarios (nombre, correo, contrasena) values ('Juan',
'juan@ejemplo.com', 'abcd');

insert into tarea (usuario, nombre, estado) values (1, 'EJ1',
'incompleta');
insert into tarea (usuario, nombre, estado) values (2, 'EJ2',
'incompleta');
```

## Webgrafía

PHP crypt(), password\_hash() Functions. (2018, febrero 20). GeeksforGeeks.  
[https://www.geeksforgeeks.org/php-crypt-password\\_hash-functions/](https://www.geeksforgeeks.org/php-crypt-password_hash-functions/)

HTML input type="radio". (s/f). W3schools.com. Recuperado el 10 de febrero de 2025,  
de [https://www.w3schools.com/tags/att\\_input\\_type\\_radio.asp](https://www.w3schools.com/tags/att_input_type_radio.asp)

<https://github.com/Rodrigo-Garcia-Ortiz/Programacion.git>