

PLATAFORMA ONLINE DE VENTA DE EQUIPOS MÉDICOS: MAXISALUD

TENSÍOMETRO DIGITAL



**NEBULIZADORES PORTÁTILES
PEDIÁTRICO-ADULTO**



OXÍMETRO DIGITAL



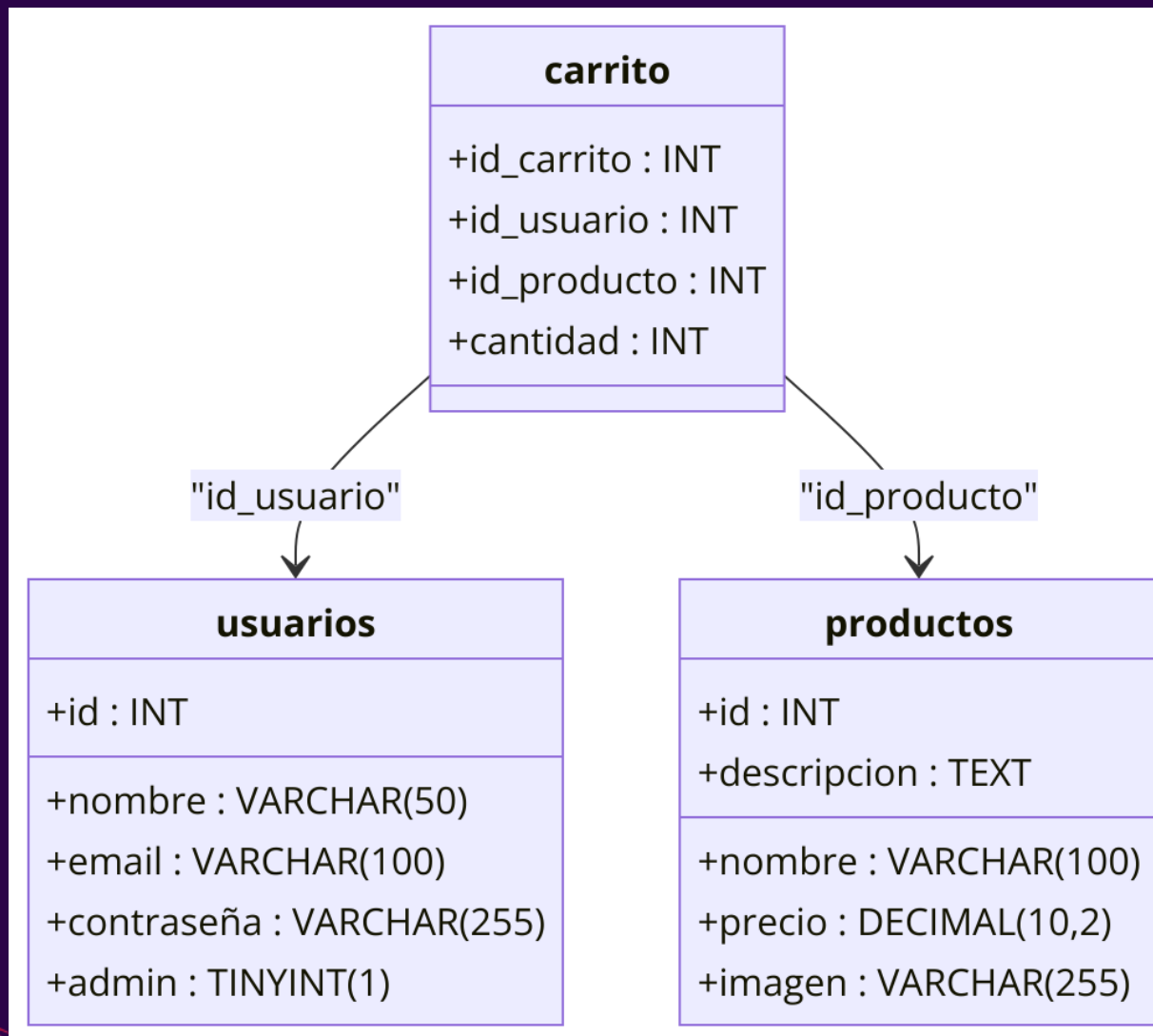
**BALONES DE
OXÍGENO PORTÁTIL**

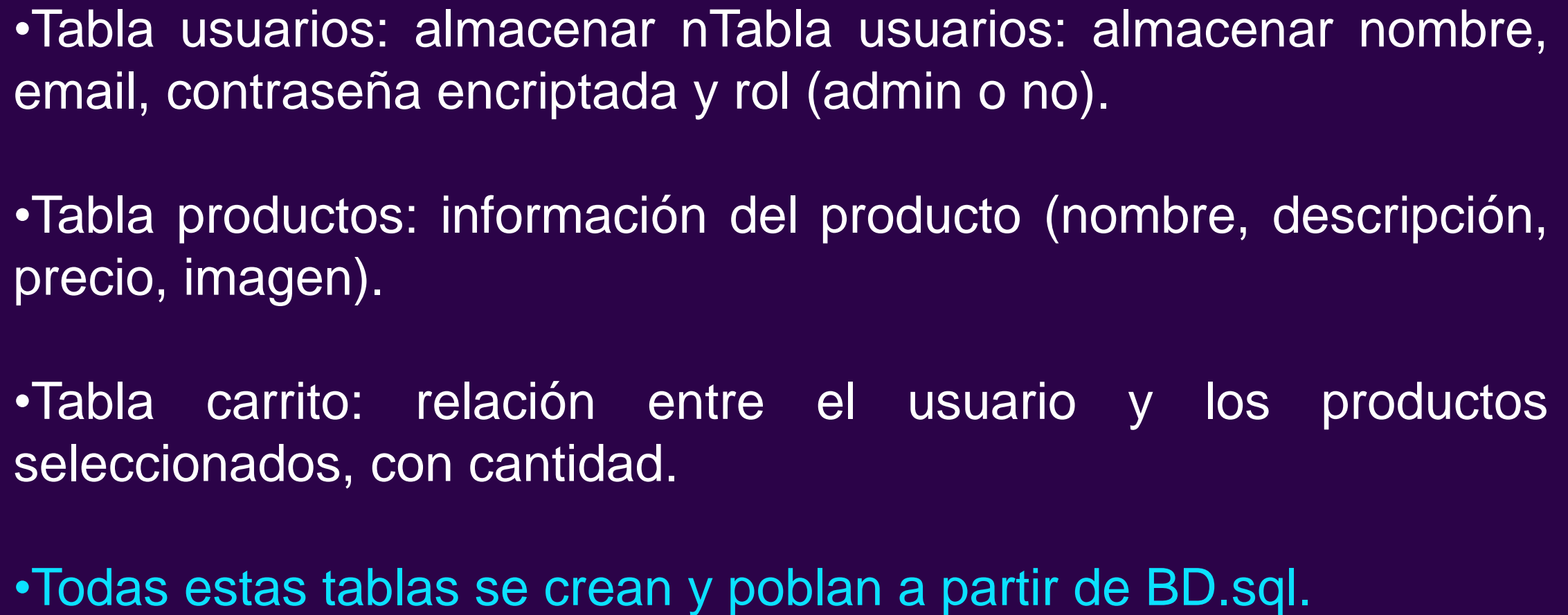
BASE DE DATOS:

- Utilicé MySQL para gestionar la información del proyecto.
- Creé tablas para usuarios, productos y un carrito de compras.
- La conexión a la base de datos se realiza a través de db.php.

```
<?php
// Declaración de las credenciales
// del servidor y base de datos
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "tienda_medica";
// Creación de la conexión a la
// base de datos usando la clase
// mysqli
$conn = new mysqli(
    hostname: $servername,
    username: $username,
    password: $password,
    database: $dbname
);
// Verificación de errores en la
// conexión
if ($conn->connect_error) {
    // Si ocurre un error, termina
    // el script y muestra un mensaje
    // de error
    die(
        "Conexión fallida: " .
        $conn->connect_error
    );
}
// Configuración del conjunto de
// caracteres para la conexión (UTF-8)
$conn->set_charset(charset: "utf8");
?>
```

TABLAS DE LA BASE DE DATOS:



- 
- Tabla usuarios: almacenar nombre, email, contraseña encriptada y rol (admin o no).
 - Tabla productos: información del producto (nombre, descripción, precio, imagen).
 - Tabla carrito: relación entre el usuario y los productos seleccionados, con cantidad.
 - Todas estas tablas se crean y poblan a partir de BD.sql.

FUNCIONES DE SEGURIDAD Y VALIDACIÓN (SEGURIDAD.PHP):

- Filtrado de datos con htmlspecialchars() y filter_var() para prevenir inyecciones XSS.
- filtrarTexto(), filtrarEmail(), filtrarEntero(), filtrarDecimal() aplicadas antes de interactuar con la BD.
- (Se aplica en controladores al recibir datos por POST/GET)

```
function filtrarTexto($campo): string {  
    return htmlspecialchars(  
        string: trim(string: $campo),  
        flags: ENT_QUOTES,  
        encoding: 'UTF-8');  
}
```


- Utilicé la API de Mailjet en CarritoController::confirmarCompra().
- Se genera el contenido del correo (recibo de compra) y se envía directamente al email del usuario.

```
$apiKeyPublic = '...';  
$apiKeyPrivate = '...';  
$data = [  
    'Messages' => [  
        [  
            'From' => ['Email' => '...', 'Name' =>  
                'Plataforma Médica'],  
            'To' => [['Email' => $email, 'Name' =>  
                $_SESSION['nombre_usuario']],  
            'Subject' => 'Recibo de tu Compra',  
            'HTMLPart' => $contenido,  
        ],  
    ],  
];
```

PATRÓN MVC APLICADO:

- Modelos (Models): Acceden a la BD (ej. ProductModelos (Models): Acceden a la BD (ej. ProductoModel.php, CarritoModel.php).
- Controladores (Controllers): Gestionan la lógica de negocio y llamadas a modelos (ej. ProductoController.php, CarritoController.php).
- Vistas (Views): Presentan datos al usuario (ej. Views/productos/lista.php, Views/carrito/ver.php).

EJEMPLO DE CONTROLADOR (CARRITOCONTROLLER):

- ver() obtiene el carrito del usuario llamando al modelo y luego carga la vista carrito/ver.php.
- agregar() filtra el ID del producto, llama al modelo y redirige al carrito.

```
public function ver() { Unexpe
    $this->checkLogin();
    $id_usuario =
    $_SESSION['id_usuario'];
    $carrito =
    $this->carritoModel->
    obtenerCarrito($id_usuario);
    require __DIR__ .
    '../Views/carrito/ver.php';
}
```


VALIDEZ DE DATOS EN FORMULARIOS (EJ. REGISTRO DE USUARIO):

- En UsuarioController.php, antes de crear un usuario, se filtra el email y se encripta la contraseña con password_hash().
- Se comprueba si el email ya existe en la BD.
- Si se marca el checkbox de admin, se pide contraseña fija de administración.

```
public function procesarLogin(): void {
    $email = filtrarEmail(campo: $_POST['email'] ?? '');
    $contraseña = $_POST['contraseña'] ?? '';

    $user = $this->usuarioModel->obtenerPorEmail(email: $email);
    if ($user && password_verify(password: $contraseña, hash: $user['contraseña'])) {
        session_start();
        $_SESSION['id_usuario'] = $user['id'];
        $_SESSION['nombre_usuario'] = $user['nombre'];
        $_SESSION['admin'] = $user['admin']; // Guardamos el rol en la sesión
        header(header: "Location: index.php");
    } else {
        $error = "Email o contraseña incorrectos.";
        require __DIR__ . '/../Views/usuarios/login.php';
    }
}
```

CONEXIÓN A LA BASE DE DATOS (DB.PHP):

- Uso de mysqli para la conexión.
- Se configura el charset UTF-8.
- El objeto \$conn se pasa a los controladores para inicializar los modelos.

```
$conn = new mysqli($servername, $username, $password, $dbname);  
$conn->set_charset("utf8");
```

MÁS SOBRE EL ENVÍO DE CORREO CON MAILJET:

- Se utiliza curl para realizar la petición POST a la API de Mailjet.
- Después de enviar el correo, se vacía el carrito, confirmando la compra.

```
$ch = curl_init('https://api.mailjet.com/v3.1/send');  
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);  
curl_setopt($ch, CURLOPT_USERPWD, "$apiKeyPublic:$apiKeyPrivate");  
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
```

GRACIAS POR
SU ATENCIÓN

FIN.

