

A continuación te explico paso a paso cómo crear una **pantalla de ingreso y consulta de inventario**, con frontend en **HTML + JavaScript** y backend en **PHP**



Objetivo

Tener una página web con:

1. **Formulario de ingreso** de productos, incluyendo campos generales y específicos (como vRAM, frecuencia, etc.).
2. **Tabla de consulta** de inventario, con filtros por categoría y marca, y orden por precio o stock.

Estructura general del proyecto

TIENDA_PROYECTO_PHP

- categorias.php
- conectar.php
- conectar_mysql.php
- conectar_pg.php
- exportar_historial_csv.php
- index.html
- index.php
- info.php
- ingresar.php
- insertar_producto.php
- insert_datatienda_inventario.sql
- inventario.php
- inventario_schema_create_mysql.sql
- marcas.php
- obtener_categorias.php
- obtener_marcas.php
- obtener_productos.php
- productos.php
- README.md
- script.js
- tienda_inventario.sql

- css
 - estilo.css

- img
 - imagen.jpg

Paso 1: Base de datos (PostgreSQL)

Ya la tienes: tablas **producto**, **categoria**, **marca**, **cpu**, **gpu**, etc. Asegúrate de tener al menos unas pocas filas en cada una.

✓ Paso 2: Backend en PHP

backend/conectar.php (conexión básica)

php

```
<?php
$host = "localhost";
$dbname = "tienda";
$user = "postgres";
$pass = "tu_contraseña";

try {
    $pdo = new PDO("pgsql:host=$host;dbname=$dbname", $user, $pass);
} catch (PDOException $e) {
    echo "Error de conexión: " . $e->getMessage();
    exit;
}
?>
```

backend/insertar_producto.php (recibe datos del formulario)

php

```
<?php
require 'conectar.php';

$categoria = $_POST['categoria'];
$marca = $_POST['marca'];
$nombre = $_POST['nombre'];
$precio = $_POST['precio'];
$stock = $_POST['stock'];

// Datos específicos
$nucleos = $_POST['nucleos'] ?? null;
$hilos = $_POST['hilos'] ?? null;

$pdo->beginTransaction();

try {
    // Insertar producto general
    $stmt = $pdo->prepare("INSERT INTO producto (nombre, precio, stock, id_categoria,
id_marca)
VALUES (?, ?, ?, (SELECT id_categoria FROM categoria WHERE nombre = ?),
(SELECT id_marca FROM marca WHERE nombre = ?))
RETURNING id_producto");

    $stmt->execute([$nombre, $precio, $stock, $categoria, $marca]);
    $id_producto = $stmt->fetchColumn();
}
```

```

        // Si es CPU, insertar en la tabla cpu
        if ($categoria === 'CPU') {
            $stmtCpu = $pdo->prepare("INSERT INTO cpu (id_producto, nucleos, hilos) VALUES (?, ?,
?)" );
            $stmtCpu->execute([$id_producto, $nucleos, $hilos]);
        }

        $pdo->commit();
        echo "Producto insertado";
    } catch (Exception $e) {
        $pdo->rollBack();
        echo "Error: " . $e->getMessage();
    }
?>

```

backend/obtener_datos.php (para mostrar inventario)

php

```

<?php
require 'conectar.php';

$query = "SELECT p.nombre, c.nombre AS categoria, m.nombre AS marca, p.precio, p.stock
        FROM producto p
        JOIN categoria c ON p.id_categoria = c.id_categoria
        JOIN marca m ON p.id_marca = m.id_marca
        ORDER BY p.nombre";

$data = $pdo->query($query)->fetchAll(PDO::FETCH_ASSOC);
echo json_encode($data);
?>

```

Paso 3: Frontend en HTML + JS

index.html

html

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Inventario Tienda</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<h2>Ingreso de Producto</h2>
<form id="formProducto">
    <label>Nombre: <input type="text" name="nombre" required></label><br>
    <label>Precio: <input type="number" name="precio" required></label><br>
    <label>Stock: <input type="number" name="stock" required></label><br>
    <label>Categoría:
        <select name="categoria" id="categoria">

```

```

        <option value="CPU">CPU</option>
        <option value="GPU">GPU</option>
    </select>
</label><br>
<label>Marca:
    <select name="marca">
        <option value="Intel">Intel</option>
        <option value="AMD">AMD</option>
    </select>
</label><br>

<div id="camposEspecificos">
    <!-- Campos dinámicos según categoría -->
</div>

<button type="submit">Insertar Producto</button>
</form>

<h2>Inventario</h2>
<table border="1" id="tablaInventario">
    <thead>
        <tr><th>Nombre</th><th>Categoría</th><th>Marca</th><th>Precio</th><th>Stock</th></tr>
    </thead>
    <tbody></tbody>
</table>

<script src="script.js"></script>
</body>
</html>

```

script.js

javascript

```

document.addEventListener("DOMContentLoaded", () => {
    const form = document.getElementById("formProducto");
    const categoria = document.getElementById("categoria");
    const camposEspecificos = document.getElementById("camposEspecificos");

    // Mostrar campos según categoría
    categoria.addEventListener("change", () => {
        camposEspecificos.innerHTML = "";
        if (categoria.value === "CPU") {
            camposEspecificos.innerHTML = `
                <label>Núcleos: <input type="number" name="nucleos"></label><br>
                <label>Hilos: <input type="number" name="hilos"></label><br>
            `;
        }
    });

    // Enviar formulario
    form.addEventListener("submit", e => {
        e.preventDefault();
        const datos = new FormData(form);
        fetch("backend/insertar_producto.php", {

```

```

        method: "POST",
        body: datos
    })
    .then(res => res.text())
    .then(msg => {
        alert(msg);
        form.reset();
        cargarInventario();
    });
});

// Cargar tabla de inventario
function cargarInventario() {
    fetch("backend/obtener_datos.php")
    .then(res => res.json())
    .then(data => {
        const tbody = document.querySelector("#tablaInventario tbody");
        tbody.innerHTML = "";
        data.forEach(prod => {
            tbody.innerHTML += `
                <tr>
                    <td>${prod.nombre}</td>
                    <td>${prod.categoria}</td>
                    <td>${prod.marca}</td>
                    <td>${prod.precio}</td>
                    <td>${prod.stock}</td>
                </tr>
            `;
        });
    });
}

// Cargar al inicio
cargarInventario();
});

```

Sí, puedes usar **Visual Studio Code (VS Code)** para editar archivos PHP como **backend/conectar.php**, pero **no puedes ejecutar directamente PHP desde el navegador solo con VS Code**. Necesitas tener un **servidor web con PHP instalado** para que funcione correctamente.

✓ Paso a paso para trabajar con PHP en Visual Studio Code:

🧱 1. Instala PHP en tu sistema

♦ En Windows:

- Descarga PHP desde <https://windows.php.net/download/>.
- Extrae y agrega la ruta a `php.exe` en las **variables de entorno**.

♦ O mejor aún, instala un servidor local como:

- XAMPP ✓

Este viene con Apache + PHP + PhpMyAdmin y son ideales para trabajar localmente.

🔧 3. Prueba tu archivo `conectar.php`

Ejemplo básico de `conectar.php`:

```
php
<?php
$host = "localhost";
$dbname = "tienda";
$user = "postgres";
$pass = "admin123"; // tu contraseña real

try {
    $pdo = new PDO("pgsql:host=$host;dbname=$dbname", $user, $pass);
    echo "✓ Conexión exitosa a PostgreSQL.";
} catch (PDOException $e) {
    echo "✗ Error de conexión: " . $e->getMessage();
    exit;
}
?>
```

🌐 4. Cómo ejecutarlo

1. Asegúrate de que tu servidor esté encendido (ej. Apache en XAMPP).

Abre tu navegador y escribe:

http://localhost/mi_proyecto/backend/conectar.php

2. Si la conexión es correcta, verás:

✓ **Conexión exitosa a PostgreSQL.**

Si hay un error, te mostrará el mensaje.

5. Opcional: Extensiones en VS Code


Instala estas extensiones en Visual Studio Code para mejorar tu experiencia:

- **PHP Extension Pack**
- **PHP Intelephense**
- **PHP Server** (si no usas XAMPP)

✓ **Opción 2: Instalar PHP de forma manual (más avanzada)**

Si **NO** usas **XAMPP**, puedes instalar solo PHP y configurarlo tú mismo.

Pasos:

1. **Descarga PHP para Windows:**
 <https://windows.php.net/download/>
2. Descarga la versión **Thread Safe** (ZIP) — la versión más estable para desarrollo.

Extrae el ZIP en una carpeta, por ejemplo:

makefile

C:\php

3. Agrega **C:\php** al **PATH** de Windows:
 - Panel de Control > Sistema > Configuración avanzada del sistema > Variables de entorno.

En "Path", haz clic en **Editar** y añade:

```
makefile  
C:\php\
```

○

Abre terminal o CMD y verifica:

```
nginx  
php -v
```

4. Si ves algo como **PHP 8.2.10**, ¡todo está listo!

⚠ Esta opción **no trae servidor web**, así que tendrías que instalar Apache o usar algo como PHP's built-in server:

```
bash  
php -S localhost:8000
```

Sí, una vez que **configuras e instalas PHP correctamente**, ya podrás ejecutar tus **scripts PHP del backend** desde Visual Studio Code, **pero necesitas un servidor web para verlos funcionar en el navegador**. Aquí te explico todo paso a paso:

✓ 1. Qué necesitas para ejecutar PHP desde Visual Studio Code

🧱 Requisitos:

1. ✓ PHP instalado correctamente en tu sistema (confirmado con **php -v** en terminal).
2. ✓ Un servidor para ejecutar los archivos (puede ser):
 - **XAMPP** (Apache + PHP) – Recomendado para desarrollo web.
 - O usar el **servidor interno de PHP** para pruebas rápidas.

Usar servidor interno de PHP

Abre terminal en la carpeta de tu proyecto:

```
bash  
cd C:\Users\fanny\Desktop\tienda_proyecto_php
```



Ejecuta:

```
bash  
php -S localhost:8000
```



Abre tu navegador:

```
bash  
http://localhost:8000/index.php
```

3. Desde Visual Studio Code puedes:

-  **Editar** tus archivos `.php`.
 -  **No puedes hacer doble clic para ejecutar .php directamente** → debes usar un **servidor** (como XAMPP o el built-in de PHP).
-

Extras recomendados en VS Code:

-  **PHP Extension Pack** → sintaxis, autocompletado, errores.
-  **PHP Server** → inicia un servidor PHP local desde VS Code con un clic.

Descarga:

Ve a:

 <https://windows.php.net/download>

Haz clic en:

```
java  
CopiarEditar  
VS16 x64 Thread Safe (ZIP)
```

¿Qué hacer después de descargarlo?

 **Paso 1: Extrae el archivo ZIP**

Extrae el ZIP en una carpeta, por ejemplo:

```
makefile  
C:\php\
```

Paso 2: Configura el PATH de Windows

Abre:

```
mathematica  
Panel de control → Sistema → Configuración avanzada del  
sistema → Variables de entorno
```

Busca la variable **Path** y haz clic en **Editar**.

Agrega esta nueva línea:
makefile

```
C:\php\
```

Acepta todo y cierra.

Paso 3: Verifica la instalación

1. Abre CMD o PowerShell.

```
bash  
php -v
```

2. Si ves algo como:

```
scss  
PHP 8.x.x (cli) (built: Mar 11 2025)
```

¡Perfecto! PHP ya está listo en tu sistema.

Pasos para instalar PHP en Windows:

1. Descargar y extraer PHP:

Ir al sitio oficial de PHP para Windows

Entra a esta URL:

👉 <https://windows.php.net/download/>

Encontrar la versión estable actual

Busca la sección que dice **PHP 8.2 (8.2.28)** (o la versión más reciente).

- Descarga el archivo ZIP mencionado anteriormente.
- Extrae el contenido en una carpeta, por ejemplo: **C:\php**.

2. Configurar el archivo **php.ini**:

- Dentro de la carpeta **C:\php**, renombra el archivo **php.ini-development** a **php.ini**.
- Abre el archivo **php.ini** con un editor de texto y realiza las siguientes configuraciones:
 - Descomenta (elimina el punto y coma al inicio de la línea) y configura las extensiones necesarias, por ejemplo:

■ **extension=mysqli**

■ **extension=pdo_mysql**

Si prefieres una solución más sencilla que incluya PHP, Apache y MySQL, puedes considerar instalar **XAMPP**, que es un paquete todo-en-uno para desarrollo web en Windows.

Usar el servidor interno de PHP

PHP viene con un servidor web simple que puedes usar para desarrollo rápido.

- Abre la terminal integrada en VS Code:
Menú → **Terminal** → **Nuevo terminal** (o usa Ctrl + ``).
- Navega a la carpeta raíz de tu proyecto si no estás ya ahí, por ejemplo:

bash

cd C:\ruta\a\tu\proyecto

- Ejecuta el siguiente comando para iniciar el servidor PHP en el puerto 8000 (puedes cambiar el puerto si quieres):
- Abre el CMD clásico (no PowerShell):

bash

php -S localhost:8000

y no te dio ningún error, eso significa que tu servidor PHP local ya está funcionando 🎉.

Acceder a tu aplicación

- Abre tu navegador y ve a:

http://localhost:8000

- Aquí podrás acceder a tus archivos PHP como si estuvieran en un servidor real.

Opcional: Extensiones recomendadas para VS Code

- PHP Server: te permite lanzar un servidor PHP directamente desde VS Code con clic derecho.