

## **Hito 6: Acceso y Manipulación de Datos en el Ecommerce.**

Juan Carlos Castilla Guisado



# Índice

<b>Hito 6: Acceso y Manipulación de Datos en el Ecommerce.....</b>	<b>1</b>
<b>Entrega 1.....</b>	<b>4</b>
<b>1. Plan de Acción.....</b>	<b>4</b>
1.1 Tablas de la Base de Datos de WooCommerce.....	4
1.1.1 wp_posts.....	4
1.1.2 wp_postmeta.....	4
1.1.3 wp_woocommerce_order_items.....	5
1.1.4 wp_woocommerce_order_itemmeta.....	5
1.1.5 wp_users.....	5
1.1.6 wp_usermeta.....	6
1.2 Tablas Propias.....	6
1.2.1 tabla_favoritos_clientes.....	6
1.2.2 tabla_avisos_stock.....	7
1.2.3 tabla_consultas_productos.....	7
1.3 Implementación de Operaciones CRUD.....	7
1.3.1 Crear (Create).....	7
1.3.2 Leer (Read).....	8
1.3.3 Actualizar (Update).....	9
1.3.4 Eliminar (Delete).....	10
1.4 Uso de Transacciones en WooCommerce de Procesadores.....	10
<b>2. Base de Datos Inicial:.....</b>	<b>11</b>
<b>3. Revisión de Roles y Permisos:.....</b>	<b>12</b>
3.1 Descripción de Permisos.....	13
<b>Entrega 2.....</b>	<b>14</b>
<b>1. Lectura de Datos (Read):.....</b>	<b>14</b>
1.1 Código o snippets que demuestren cómo el alumno está recuperando los productos/pedidos/usuarios de la base de datos.....	14
1.1.1 wp_posts.....	14
1.1.2 wp_postmeta.....	15
1.2 Capturas o breve vídeo que muestre dónde y cómo se listan en el front-end o back-end.....	16
1.2.1 Recuperación de datos sobre phpMyAdmin (Productos).....	16
1.2.2 Recuperación de datos sobre Wordpress (Productos).....	17
<b>2. Estructura de Código:.....</b>	<b>19</b>
<b>3. Seguridad y Validaciones Básicas:.....</b>	<b>20</b>
<b>Entrega 3.....</b>	<b>22</b>



1. Actualización de Datos (Update):.....	22
2. Eliminación de Datos (Delete):.....	24
3. Uso de Transacciones:.....	26
4. Control de Permisos:.....	29



# Entrega 1

## 1. Plan de Acción

### 1.1 Tablas de la Base de Datos de WooCommerce

#### 1.1.1 wp\_posts

Esta tabla contiene información básica sobre los productos y otros elementos de contenido gestionados por WordPress. En el caso de WooCommerce, se utiliza para registrar productos con su nombre, descripción, estado (publicado o borrador), y tipo de contenido.

- **Ejemplo de contenido:**
  - Productos (post\_type = 'product').
  - Pedidos (post\_type = 'shop\_order').

Esta tabla es esencial para identificar y categorizar los productos disponibles en la tienda.

#### 1.1.2 wp\_postmeta

La tabla wp\_postmeta almacena datos adicionales relacionados con los productos y otros elementos que se registran en wp\_posts. Cada producto tiene múltiples entradas en esta tabla para incluir detalles específicos.

- **Información almacenada:**
  - Precios (regular y en oferta).
  - Inventario (cantidad disponible en stock).
  - SKU (código único de identificación).
  - Dimensiones físicas (alto, ancho, largo).
  - Imágenes asociadas a los productos.

Esta tabla es clave para acceder a información detallada que no está directamente disponible en wp\_posts.



### 1.1.3 wp\_woocommerce\_order\_items

Esta tabla registra los elementos que forman parte de cada pedido realizado en la tienda. Cada pedido tiene una o más entradas en esta tabla, dependiendo del número de productos comprados.

- **Información almacenada:**
  - Identificación del pedido.
  - Productos incluidos en el pedido.
  - Cantidad de cada producto.

Se utiliza principalmente para analizar las ventas y los productos más adquiridos.

### 1.1.4 wp\_woocommerce\_order\_itemmeta

Esta tabla complementa a wp\_woocommerce\_order\_items al agregar detalles específicos de los elementos de un pedido. Es donde se almacenan datos personalizados relacionados con cada producto en un pedido.

- **Información almacenada:**
  - Precio unitario del producto en el momento de la compra.
  - Impuestos aplicados al producto.
  - Descuentos o cupones utilizados.

Es esencial para calcular y entender los valores finales de cada pedido.

### 1.1.5 wp\_users

La tabla wp\_users contiene información sobre los usuarios registrados en el sitio web. Esto incluye tanto a los administradores del sistema como a los clientes que realizan compras.

- **Información almacenada:**
  - Nombre de usuario.
  - Dirección de correo electrónico.



- Contraseña en formato cifrado.
- Roles (como cliente, administrador o suscriptor).

Esta tabla permite identificar a los clientes y gestionar sus accesos.

### 1.1.6 wp\_usermeta

Esta tabla almacena datos adicionales sobre los usuarios registrados en wp\_users. Cada usuario puede tener múltiples metadatos asociados que se utilizan para personalizar su experiencia.

- **Información almacenada:**

- Dirección de facturación.
- Dirección de envío.
- Preferencias del usuario.
- Información de perfil adicional.

Se utiliza para obtener detalles específicos de los clientes, especialmente para gestionar pedidos y envíos.

Las tablas mencionadas proporcionan una estructura completa para manejar la información de la tienda en línea.

Las tablas principales (wp\_posts y wp\_users) se encargan de los datos básicos, mientras que las tablas complementarias (wp\_postmeta, wp\_woocommerce\_order\_items, wp\_woocommerce\_order\_itemmeta, y wp\_usermeta) permiten gestionar información detallada sobre productos, pedidos y clientes. Juntas, forman la base de datos necesaria para operar y analizar una tienda WooCommerce de manera eficiente.

## 1.2 Tablas Propias

### 1.2.1 tabla\_favoritos\_clientes

- **Propósito:** Registrar los productos que los clientes marcan como favoritos.
- **Campos principales:**



- id\_favorito: Identificador único del registro.
- id\_usuario: ID del cliente (relación con wp\_users).
- id\_producto: ID del producto marcado como favorito (relación con wp\_posts).
- fecha\_agregado: Fecha en que se agregó a favoritos.

### 1.2.2 tabla\_avisos\_stock

- **Propósito:** Llevar un registro de las solicitudes de notificación cuando un producto vuelva a estar disponible en stock.
- **Campos principales:**
  - id\_avisos: Identificador único del aviso.
  - id\_usuario: ID del cliente que solicitó el aviso (relación con wp\_users).
  - id\_producto: ID del producto solicitado (relación con wp\_posts).
  - fecha\_solicitud: Fecha en que se realizó la solicitud.

### 1.2.3 tabla\_consultas\_productos

- **Propósito:** Registrar preguntas realizadas por los clientes sobre productos.
- **Campos principales:**
  - id\_consulta: Identificador único de la consulta.
  - id\_usuario: ID del cliente que hizo la pregunta (relación con wp\_users).
  - id\_producto: ID del producto relacionado (relación con wp\_posts).
  - consulta: Texto de la pregunta realizada.
  - fecha\_consulta: Fecha en que se envió la consulta.

## 1.3 Implementación de Operaciones CRUD

### 1.3.1 Crear (Create)

Para añadir nuevos productos, pedidos o registros en tablas personalizadas:

- **Productos:** Se utilizará la función wp\_insert\_post() para crear nuevos productos en la tabla wp\_posts.



Ejemplo:

```
<?php
wp_insert_post([
    'post_title'     => 'Nuevo Producto',
    'post_content'   => 'Descripción del producto.',
    'post_status'    => 'publish',
    'post_type'      => 'product',
]);
?>
```

- **Metadatos de productos:** Para añadir detalles adicionales, como el precio o el inventario de los productos, se usará `update_post_meta()`.

Ejemplo:

```
<?php
update_post_meta($product_id, '_price', '19.99');
update_post_meta($product_id, '_stock', 100);
?>
```

### 1.3.2 Leer (Read)

Para obtener datos de la base de datos y mostrarlos en el sitio o en el panel de administración:

- **Obtener productos:** Para leer los productos y sus detalles, se utilizará `get_posts()` o `WP_Query()`.

Ejemplo:

```
<?php
$args = [
    'post_type' => 'product',
    'posts_per_page' => 10,
];
$productos = new WP_Query($args);
if ($productos->have_posts()) {
    while ($productos->have_posts()) {
        $productos->the_post();
    }
}
?>
```





- **Leer metadatos:** Para obtener los detalles específicos de un producto, como el precio o el stock, se usará `get_post_meta()`.

Ejemplo:

```
<?php
$precio = get_post_meta($product_id, '_price', true);
$stock = get_post_meta($product_id, '_stock', true);
?>
```

### 1.3.3 Actualizar (Update)

Para modificar la información existente en la base de datos:

- **Actualizar productos:** Para modificar un producto, se utilizará la función `wp_update_post()`.

Ejemplo:

```
<?php
wp_update_post([
    'ID' => $product_id,
    'post_title' => 'Producto Actualizado',
    'post_content' => 'Descripción actualizada del producto.',
]);
?>
```

- **Actualizar metadatos:** Para modificar información adicional de un producto, como el precio o el inventario, se puede usar `update_post_meta()` de nuevo.

Ejemplo:

```
<?php
update_post_meta($product_id, '_price', '29.99');
update_post_meta($product_id, '_stock', 80);
?>
```



### 1.3.4 Eliminar (Delete)

Para eliminar productos, pedidos o registros de las tablas personalizadas:

- **Eliminar productos:** Para eliminar un producto de la base de datos, se utilizará `wp_delete_post()`. Esta función también eliminará los metadatos asociados a ese producto.

Ejemplo:

```
<?php
wp_delete_post($product_id, true);
?>
```

- **Eliminar metadatos:** Para eliminar metadatos específicos de un producto, se usará `delete_post_meta()`.

Ejemplo:

```
<?php
delete_post_meta($product_id, '_price');
?>
```

## 1.4 Uso de Transacciones en WooCommerce de Procesadores

### 1.4.1 Creación de pedidos

Al crear un pedido con procesadores, se asegurará que tanto la creación del pedido en `wp_posts`, como la inserción de los productos en `wp_woocommerce_order_items` y sus metadatos, se realicen correctamente. Si algo falla, se revertirá todo el proceso.

### 1.4.2 Actualización de stock

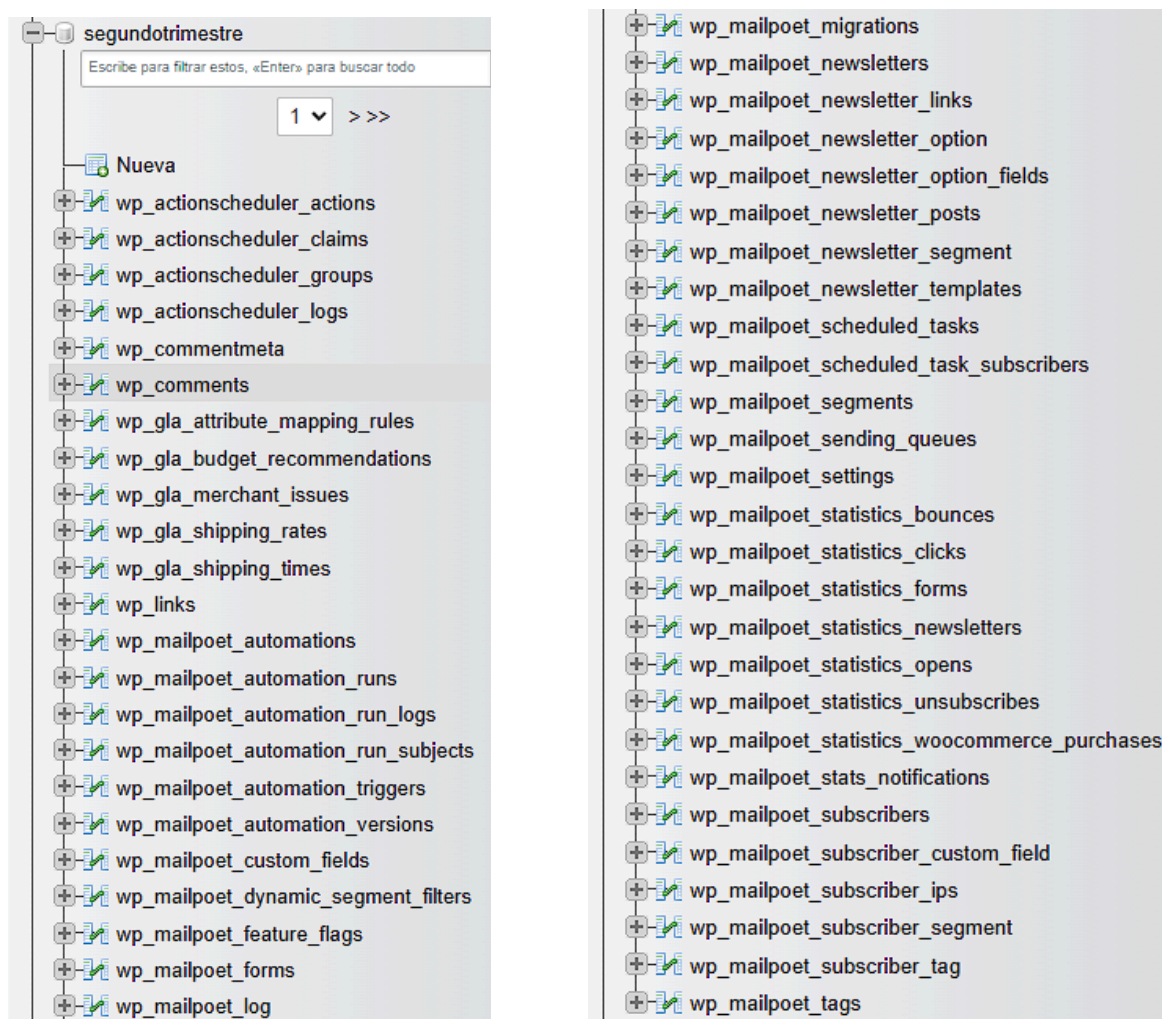
Si se actualiza el stock de procesadores en `wp_postmeta`, la transacción garantiza que tanto el precio como la cantidad de stock se actualicen correctamente o no se haga nada si falla alguna parte.



### 1.4.3 Eliminación de productos:

Al eliminar un procesador, se garantizará que tanto el producto en wp\_posts como sus metadatos en wp\_postmeta y cualquier registro relacionado en wp\_woocommerce\_order\_items sean eliminados correctamente.

## 2. Base de Datos Inicial:



Error base de datos: En mi base de datos faltan las tablas más importantes como wp\_posts u wp\_postmeta y salen tablas totalmente diferentes y ya luego al buscar la tabla en mi explorador de archivos salen pero no en mi phpMyAdmin.



Resultados de la búsqueda en xampp			wp_post	
Ordenar Ver Opciones de búsqueda Cerrar búsqueda Detalles				
wp_posts.ibd	C:\xampp\mysql\data\segundotrimestre	Tipo: Archivo IBD	Fecha de modificación: 10/01/2025 19:18 Tamaño: 288 KB	
wp_postmeta.ibd	C:\xampp\mysql\data\segundotrimestre	Tipo: Archivo IBD	Fecha de modificación: 10/01/2025 19:17 Tamaño: 128 KB	
wp_posts.frm	C:\xampp\mysql\data\segundotrimestre	Tipo: Archivo FRM	Fecha de modificación: 10/01/2025 18:46 Tamaño: 7,04 KB	
wp_postmeta.frm	C:\xampp\mysql\data\segundotrimestre	Tipo: Archivo FRM	Fecha de modificación: 10/01/2025 18:46 Tamaño: 2,95 KB	

### 3. Revisión de Roles y Permisos:

Rol	Leer Productos	Actualizar Productos	Eliminar Productos	Leer Pedidos	Actualizar Pedidos	Eliminar Pedidos	Leer Clientes	Actualizar Clientes	Eliminar Clientes
Administrador	Si	Si	Si	Si	Si	Si	Si	Si	Si
Gerente de Tienda	Si	Si	No	Si	Si	No	Si	No	No
Vendedor	Si (propios)	Si (propios)	No	No	No	No	No	No	No
Cliente	Si	No	No	Si	No	No	Si	Si (propios)	No
Invitado	Si	No	No	No	No	No	No	No	No



### 3.1 Descripción de Permisos

- **Leer Productos:** Todos los roles pueden ver los productos en la tienda, pero solo algunos roles pueden realizar modificaciones.
- **Modificar Productos:** Solo los roles con permisos de gestión (Administrador, Gerente de Tienda, Vendedor para productos propios) tienen la capacidad de editar productos.
- **Eliminar Productos:** Solo el Administrador tiene permiso para eliminar productos de la tienda.
- **Ver Pedidos:** Todos los roles que gestionan la tienda (Administrador, Gerente de Tienda) pueden ver todos los pedidos, mientras que los clientes solo pueden ver los suyos.
- **Modificar Pedidos:** Los Administradores y Gerentes de Tienda pueden actualizar pedidos, mientras que los vendedores no tienen este permiso.
- **Eliminar Pedidos:** Solo el Administrador puede eliminar pedidos de la tienda.
- **Ver Clientes:** Los Administradores y Gerentes de Tienda pueden ver los datos de todos los clientes. Los Clientes solo pueden ver su propia información.
- **Modificar Clientes:** Los Administradores pueden editar cualquier dato de los clientes, mientras que los Gerentes de Tienda solo pueden verlos, y los clientes solo pueden editar su propia información.
- **Eliminar Clientes:** Solo el Administrador tiene la capacidad de eliminar clientes de la tienda.

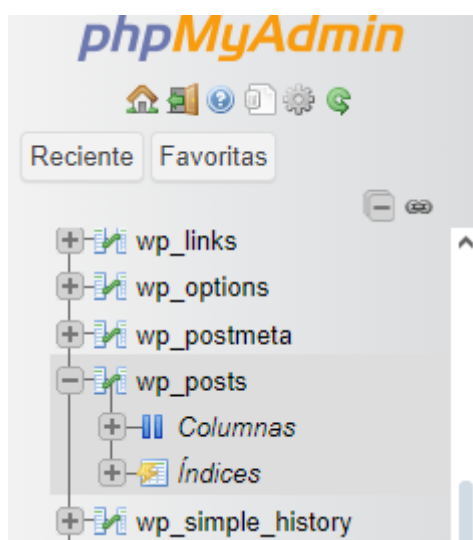


## Entrega 2

### 1. Lectura de Datos (Read):

1.1 Código o snippets que demuestren cómo el alumno está recuperando los productos/pedidos/usuarios de la base de datos.

#### 1.1.1 wp\_posts



✓ Mostrando filas 0 - 24 (total de 71, La consulta tardó 0.0015 segundos.)								
SELECT * FROM `wp_posts`								
<input type="checkbox"/> Perfilando [ <a href="#">Editar en línea</a> ] [ <a href="#">Editar</a> ] [ <a href="#">Explicar SQL</a> ] [ <a href="#">Crear código PHP</a> ] [ <a href="#">Actualizar</a> ]								
1 > >> <input type="checkbox"/> Mostrar todo   Número de filas: 25   Filtrar filas: <input type="text" value="Buscar en esta tabla"/>   Ordenar según la clave: Ninguna								
Opciones extra								
	ID	post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_stat
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	1	1	2024-10-17 14:34:08	2024-10-17 12:34:08	<!-- wp:paragraph --> <p>Te damos la bienvenida a ...	¡Hola, mundo!		publish
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	2	1	2024-10-17 14:34:08	2024-10-17 12:34:08	<!-- wp:paragraph --> <p>Esta es una página de eje...	Página de ejemplo		publish
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	3	1	2024-10-17 14:34:08	2024-10-17 12:34:08	<!-- wp:heading --> <h2 class="wp-block-heading">Q...	Política de privacidad		draft
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	4	1	2024-10-17 14:34:43	0000-00-00 00:00:00		Borrador automático		auto-draft



La tabla wp\_posts almacena la información de todo el contenido del sitio, como entradas, páginas, adjuntos y revisiones. Cada fila representa un artículo o elemento de contenido, con detalles como el título, el contenido, la fecha de publicación y el estado.

### 1.1.2 wp\_postmeta



Mostrando filas 0 - 24 (total de 198, La consulta tardó 0.0008 segundos.)

```
SELECT * FROM `wp_postmeta`
```

☐ Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

1 > >> ☐ Mostrar todo | Número de filas: 25 | Filtrar filas:  Ordenar

Opciones extra

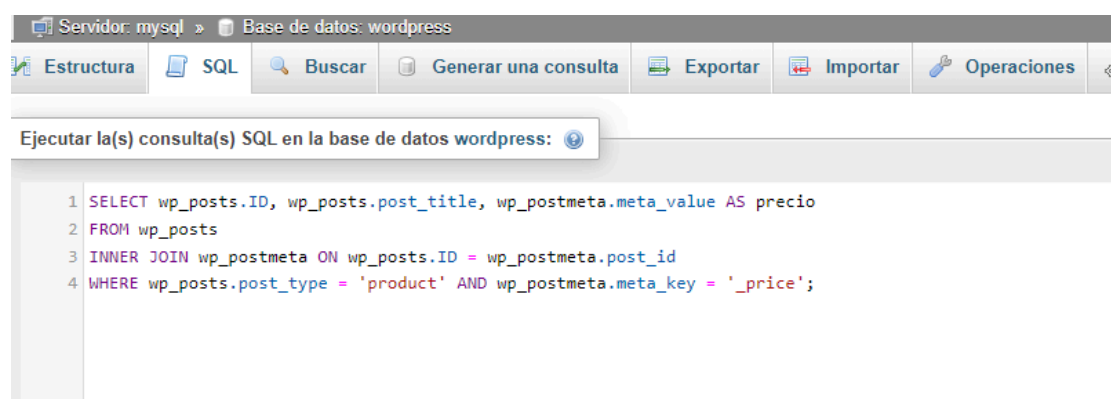
			meta_id	post_id	meta_key	meta_value
<input type="checkbox"/>	Editar	Copiar	Borrar	1	2 _wp_page_template	default
<input type="checkbox"/>	Editar	Copiar	Borrar	2	3 _wp_page_template	default
<input type="checkbox"/>	Editar	Copiar	Borrar	3	1 post_image_type	list-blog
<input type="checkbox"/>	Editar	Copiar	Borrar	4	5 _elementor_edit_mode	builder
<input type="checkbox"/>	Editar	Copiar	Borrar	5	5 _elementor_template_type	kit
<input type="checkbox"/>	Editar	Copiar	Borrar	6	7 _edit_lock	1731332589:1

La tabla wp\_postmeta en WordPress almacena metadatos adicionales relacionados con las entradas, páginas o cualquier tipo de contenido en la tabla wp\_posts. Guarda información extra, como configuraciones personalizadas o datos específicos de un plugin, asociados a cada elemento de contenido, con claves y valores.



## 1.2 Capturas o breve vídeo que muestre dónde y cómo se listan en el front-end o back-end.

### 1.2.1 Recuperación de datos sobre phpMyAdmin (Productos)



Este código obtiene el ID, el título y el precio de los productos en una tienda de WordPress. Para hacerlo, une la tabla de productos (wp\_posts) con la tabla de metadatos (wp\_postmeta), donde se guarda el precio de cada producto. Solo selecciona los productos cuyo precio esté registrado, mostrando el título y el valor del precio para cada uno.





Mostrar ventana de consultas SQL

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

✓ Mostrando filas 0 - 6 (total de 7, La consulta tardó 0.0005 segundos.)

```
SELECT wp_posts.ID, wp_posts.post_title, wp_postmeta.meta_value AS precio FROM wp_posts INNER JOIN wp_postmeta ON wp_posts.ID = wp_postmeta.post_id WHERE wp_posts.post_type = 'product' AND wp_postmeta.meta_key = '_price';
```

☐ Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

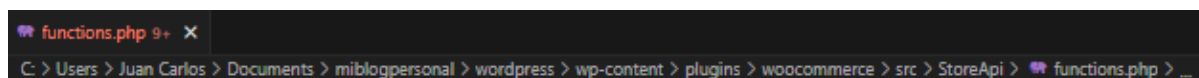
☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

ID	post_title	precio
19	Retro Glass Jug (330 ml)	115
20	Hi-Fi Headphones	125
21	3-Speed Bike	115
22	Black and White	115
23	Leather-Clad Leisure Chair	249
24	Vintage Typewriter	90
69	Procesador AMD Ryzen 5	225

El resultado es una tabla creada con los productos de mi woocommerce.

### 1.2.2 Recuperación de datos sobre Wordpress (Productos)



Esta es la ruta de mi functions.php donde he utilizado la función para mostrar mis productos de mi WooCommerce y lo utilizo con un shortcode en una nueva página creada solamente para el listado de productos.

```
// Función para mostrar productos de WooCommerce
0 references
function mostrar_productos_woocommerce_shortcode(): bool|string {
    ob_start(); // Comienza la captura de la salida HTML

    // Consulta para obtener los productos de WooCommerce
    $args = array(
        'post_type' => 'product',
        'posts_per_page' => -1 // Recupera todos los productos
    );

    $query = new WP_Query($args);

    // Verifica si hay productos
    if ($query->have_posts()) {
        echo '<div class="productos-lista">';
        while ($query->have_posts()) {
            $query->the_post();
            $product_id = get_the_ID();
            $product_title = get_the_title();
            $product_price = get_post_meta($product_id, '_price', true);

            // Muestra el producto con título y precio
            echo '<div class="producto-item">';
            echo '<h2>' . $product_title . '</h2>';
            echo '<p>Precio: ' . $product_price . '</p>';
            echo '</div>';
        }
        echo '</div>';
        wp_reset_postdata();
    } else {
        echo 'No se encontraron productos.';
    }

    return ob_get_clean(); // Devuelve la salida capturada
}

// Registra el shortcode [productos]
add_shortcode('productos', 'mostrar_productos_woocommerce_shortcode');
```

Este código define una función que muestra todos los productos de WooCommerce con su título y precio. Usa WP\_Query para obtener los productos y recorre cada uno para mostrar su título y precio en un formato HTML. Si no se encuentran productos, muestra un mensaje indicando que no hay productos disponibles. Al final, la función se registra como un shortcode [productos], lo que permite agregar fácilmente la lista de productos en cualquier página o entrada de WordPress.



Juan Carlos Proyecto

[Carrito](#)

[Cuestionario](#)

[Finalizar compra](#)

[Mi cuenta](#)

[Página de ejemplo](#)

[Productos](#)

[Tienda](#)

[User](#)

[Shopping Cart](#)

## Productos

### Procesador AMD Ryzen 5

Precio: 225€

### Black and White

Precio: 115€

### Leather-Clad Leisure Chair

Precio: 249€

### Vintage Typewriter

Precio: 90€

### Retro Glass Jug (330 ml)

Precio: 115€

### Hi-Fi Headphones

Precio: 125€

### 3-Speed Bike

Precio: 115€

Juan Carlos Proyecto

[Acerca de](#)

[Equipo](#)

[Historia](#)

[Careras](#)

[Privacidad](#)

[Política de privacidad](#)

[Términos y condiciones](#)

[Contacta con nosotros](#)

[Social](#)

[Facebook](#)

[Instagram](#)

[Twitter/X](#)

Desarrollado con [WordPress](#)

Este es el resultado de mostrar los productos de mi página donde se ve la página que he creado solamente para esto y donde se conecta a las demás páginas de mi WooCommerce.

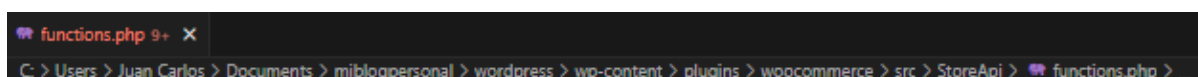
## 2. Estructura de Código:

He colocado el código en el archivo functions.php de mi plugin de WooCommerce para que la funcionalidad esté asociada directamente al plugin y no al tema. Esto me permite mostrar





los productos de WooCommerce con su título y precio a través de un shortcode, lo que hace que sea fácil insertarlos en cualquier página o entrada. Al hacerlo dentro del plugin, aseguro que esta funcionalidad sea independiente del tema, lo que significa que si cambio de tema en el futuro, no perderé la funcionalidad. Así, todas las personalizaciones relacionadas con el plugin permanecen dentro de él, sin depender del tema activo.



Esta es exactamente la ruta donde he modificado el código para los listados creados anteriormente.

### 3. Seguridad y Validaciones Básicas:

```
C:\Users\Juan Carlos\Documents\miblogpersonal\wordpress\wp-content\plugins\woocommerce\src\StoreApi\functions.php > PHP > mostrar_producto_en_pie_pagina()
129 //
130 // Función para mostrar un producto en el pie de página
131 //
132 // ID del producto que quieres mostrar (cambia este ID por el correcto)
133 $producto_id = 69; // Cambia este valor por el ID de tu producto
134
135 // Validar que el ID del producto sea un número entero válido
136 $producto_id = intval(value: $producto_id); // Convertir a entero
137
138 // Comprobar si el ID es válido (mayor que 0)
139 if ($producto_id <= 0) {
140     echo 'Por favor, ingresa un ID de producto válido.';
141     return;
142 }
143
144 // Realizar la consulta para obtener el producto
145 $args = array(
146     'post_type' => 'product', // Asegura que solo buscamos productos
147     'p' => $producto_id, // Filtra por el ID del producto
148     'post_status' => 'publish' // Asegura que solo se traigan productos publicados
149 );
150
151 $query = new WP_Query($args);
152
153 // Verificar si el producto existe
154 if ($query->have_posts()) {
155     while ($query->have_posts()) {
156         $query->the_post();
157         $product_title = get_the_title(); // Título del producto
158         $product_price = get_post_meta(get_the_ID(), '_price', true); // Precio del producto
159         $product_link = get_permalink(); // Enlace al producto
160
161         // Mostrar el producto en el pie de página
162         echo '<div class="producto-en-pie-pagina">';
163         echo '<h2>' . esc_html($product_title) . '</h2>';
164         echo '<p>Precio: ' . esc_html($product_price) . '</p>';
165         echo '<a href="' . esc_url($product_link) . '" target="_blank">Ver Producto</a>';
166         echo '</div>';
167     }
168 } else {
169     echo 'Producto no encontrado. Verifica que el ID del producto sea correcto.';
170 }
171
172 wp_reset_postdata(); // Resetear los datos del loop
173
174 // Enganchar la función al hook wp_footer
175 add_action('wp_footer', 'mostrar_producto_en_pie_pagina');
```



Este código se utiliza para mostrar un producto específico en el pie de página de tu tienda WooCommerce. Lo que hace es usar el ID de un producto (que puedes modificar) para buscar ese producto en la base de datos de WordPress. Una vez encontrado, muestra su título, precio y un enlace al producto para que los usuarios puedan verlo. Si el ID del producto es incorrecto o el producto no existe, muestra un mensaje de error indicándolo. Todo esto se realiza automáticamente en todas las páginas del sitio gracias al hook `wp_footer`, lo que asegura que el producto se muestre en el pie de página sin necesidad de intervenir en cada página manualmente.

## Procesador AMD Ryzen 5



Precio: 225€

[Ver Producto](#)

Este es el resultado que sale en el pie de página de mi WooCommerce donde he elegido que salga mi producto Ryzen 5 utilizando su id y donde debajo dando click a ver producto puedes ver toda la información de este y por siguiente añadirlo al carrito y comprarlo.



Juan Carlos Proyecto

[Carrito](#) [Cuestionario](#) [Finalizar compra](#) [Mi cuenta](#) [Página de ejemplo](#) [Productos](#) [Tienda](#)  

[Inicio](#) / [Sin categorizar](#) / [Procesador AMD Ryzen 5](#)

¡Oferta!



## Procesador AMD Ryzen 5

~~230,00 €~~ 225,00 €

Procesador AMD Ryzen 5 2600X 3.6 Ghz

1

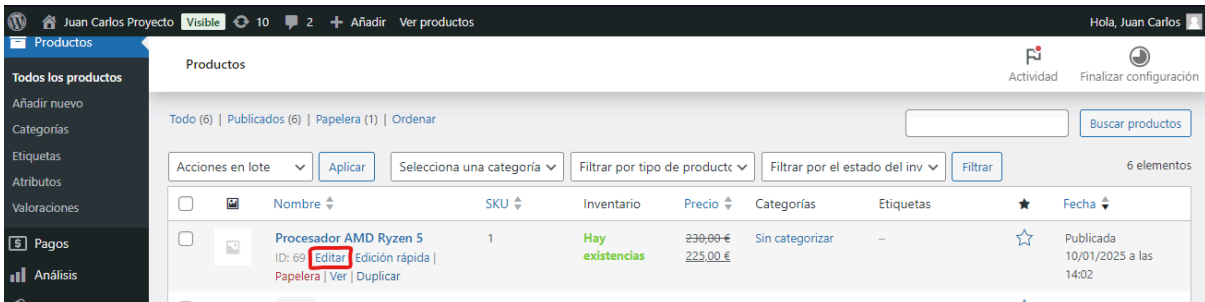
Añadir al carrito

SKU: 1 Category: Sin categorizar

Esto es lo que sale tras hacer click en ver producto donde sale la información de mi producto.

## Entrega 3

### 1. Actualización de Datos (Update):



Productos

Actividad Finalizar configuración

Todo (6) | Publicados (6) | Papelera (1) | Ordenar

Acciones en lote Aplicar Selecciona una categoría Filtrar por tipo de producto Filtrar por el estado del inv Filtrar 6 elementos

	Nombre	SKU	Inventario	Precio	Categorías	Etiquetas	Fecha
<input type="checkbox"/>	Procesador AMD Ryzen 5 ID: 69 <a href="#">Editar</a> Edición rápida   <a href="#">Papelera</a>   <a href="#">Ver</a>   <a href="#">Duplicar</a>	1	Hay existencias	<del>230,00 €</del> 225,00 €	Sin categorizar	—	Publicada 10/01/2025 a las 14:02



Para modificar el stock debemos entrar en los productos de mi woocommerce y dentro de uno darle a editar.

Datos del producto — Producto simple ? ☐ Virtual ☐ Descargable ^ v ▲

- General
- Inventario**
- Envío
- Productos relacionados
- Atributos
- Avanzado

SKU  ?

GTIN, UPC, EAN, or ISBN  ?

Gestión de inventario ☒ Hacer seguimiento de la cantidad de inventario de este producto

Cantidad  ?

¿Permitir reservas?

- ☐ No permitir
- ☒ Permitir, pero se avisará al cliente
- ☐ Permitir

Umbral de pocas existencias  ?

Vendido individualmente ☐ Limitar compras a 1 artículo por pedido ?

Desde aquí podemos cambiar el stock de cada producto modificando todo lo que tengas que cambiar para el nuevo stock de la tienda.



**Publicar** ^ v ▲

Vista previa de los cambios

Estado: **Publicada** [Editar](#)

Visibilidad: **Pública** [Editar](#)

Publicado el: 10 Ene 2025 a las 14:02  
[Editar](#)

Visibilidad catálogo: En la tienda y en los resultados de búsqueda [Editar](#)

[Copiar a un nuevo borrador](#)  
[Mover a la papelera](#) **Actualizar**

Para finalizar la actualización de stock debemos darle a actualizar y publicar para que se modifique todo el nuevo stock.

## Formulario

Nuevo Precio:  Nuevo Stock:

Nueva Descripción:  **Actualizar Producto**

Desde este formulario podemos hacer lo mismo que hemos hecho antes pero mucho más rápido donde se actualiza al poner los nuevos datos que queremos en nuestro producto.

**Código:**





```
function product_update_form_shortcode($atts): bool|string {
    ob_start();
    ?>
    <form action="<?php echo esc_url(admin_url('admin-post.php')); ?>" method="POST">
        <input type="hidden" name="action" value="update_product_data">
        <input type="hidden" name="product_id" value="69"> <!-- ID del producto -->

        <label for="price">Nuevo Precio:</label>
        <input type="number" name="price" value="20.00" step="0.01">

        <label for="stock">Nuevo Stock:</label>
        <input type="number" name="stock" value="100">

        <label for="description">Nueva Descripción:</label>
        <textarea name="description">Descripción actualizada del producto.</textarea>

        <button type="submit">Actualizar Producto</button>
    </form>
    <?php
    return ob_get_clean();
}

add_shortcode('product_update_form', 'product_update_form_shortcode');
```

## 2. Eliminación de Datos (Delete):

Descripción de cómo y cuándo se permite eliminar un producto o pedido.

Nombre	SKU	Inventario	Precio	Categorías	Etiquetas	Fecha
Procesador AMD Ryzen 5		1	239,00 € 225,00 €	Sin categorizar	—	Publicada 10/01/2025 a las 14:02
Simple Vintage Typewriter		—	90,00 €	Sin categorizar	—	Publicada 10/01/2025 a las 14:00
Simple Retro Glass Jug (330 ml) ID: 19   Editar   Edición rápida   <b>Enviar a la papelera</b> Ver   Duplicar		—	115,00 €	Sin categorizar	—	Publicada 10/01/2025 a las 14:00
Simple Hi-Fi Headphones		—	125,00 €	Sin categorizar	—	Publicada 10/01/2025 a las 14:00
Simple 3-Speed Bike		—	115,00 €	Sin categorizar	—	Publicada 10/01/2025 a las 14:00

Para eliminar un producto desde wordpress debemos enviar a la papelera el producto que no tenemos ya en stock enviándolo a la papelera.



<input type="checkbox"/>	Nombre	SKU	Inventario	Precio	Categorías	Etiquetas	★	Fecha
<input type="checkbox"/>	Simple <b>Leather-Clad Leisure Chair</b>	—	Hay existencias	249,00 €	Sin categorizar	—	☆	Última modificación 10/01/2025 a las 14:00
<input type="checkbox"/>	Simple <b>Black and White</b>	—	Hay existencias	115,00 €	Sin categorizar	—	☆	Última modificación 10/01/2025 a las 14:00
<input type="checkbox"/>	Nombre	SKU	Inventario	Precio	Categorías	Etiquetas	★	Fecha
Acciones en lote <input type="button" value="Aplicar"/> <input type="button" value="Vaciar papelera"/>								
2 elementos								

Después debemos entrar en la papelera y vaciarla para eliminar del todo estos productos que ya no tenemos en stock

✓ Mostrando filas 0 - 5 (total de 6, La consulta tardó 0.0009 segundos.)

SELECT \* FROM `wp\_wc\_product\_meta\_lookup`

☐ Perfilando

[ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

☐ Mostrar todo

Número de filas: 25

Filtrar filas:

Ordenar según la clave: Ninguna

Opciones extra

			product_id	sku	global_unique_id	virtual	downloadable	min_price	max_price	onsale	stock_quantity	stock_status	rating
<input type="checkbox"/>	Editar  Copiar  Borrar		19			0	0	115.0000	115.0000	0	NULL	instock	
<input type="checkbox"/>	Editar  Copiar  Borrar		20			0	0	125.0000	125.0000	0	NULL	instock	
<input type="checkbox"/>	Editar  Copiar  Borrar		21			0	0	115.0000	115.0000	0	NULL	instock	
<input type="checkbox"/>	Editar  Copiar  Borrar		24			0	0	90.0000	90.0000	0	NULL	instock	
<input type="checkbox"/>	Editar  Copiar  Borrar		69	1	18598489584848	0	0	225.0000	225.0000	1	25	instock	

Aquí podemos ver en nuestra base de datos que solo están los productos que tenemos en stock y los que hemos eliminado ya no están en nuestra base de datos.



### 3. Uso de Transacciones:

## Transacciones

ID del Producto:  Nuevo Precio:   
Nuevo Stock:    
Producto actualizado exitosamente.

Aquí podemos ver que hemos creado una transacción para actualizar nuestros productos donde solo poniendo su id podemos cambiarle el precio y la cantidad de artículos que tenemos disponibles en stock, al pulsar el botón se actualiza instantáneamente.

		product_id	sku	global_unique_id	virtual	downloadable	min_price	max_price	onsale	stock_quantity	stock_status
<input type="checkbox"/>	Editar										
<input type="checkbox"/>	Editar	19			0	0	115.0000	5.0000	0	0	instock
<input type="checkbox"/>	Editar	20			0	0	125.0000	125.0000	0	NULL	instock
<input type="checkbox"/>	Editar	21			0	0	115.0000	115.0000	0	NULL	instock
<input type="checkbox"/>	Editar	24			0	0	90.0000	90.0000	0	NULL	instock
<input type="checkbox"/>	Editar	69	1	18598489584848	0	0	225.0000	299.9900	1	200	instock

Aquí podemos ver como nuestro producto con id 69 que es el que hemos modificado se ha cambiado de precio y salen con los nuevos datos que hemos añadido en la transacción.

Codigo:

```
function product_update_transaction_shortcode() {  
    ob_start();  
    // Formulario para actualizar producto  
    ?>  
    <form id="product_update_form" method="post">  
        <label for="product_id">ID del Producto:</label>  
        <input type="number" name="product_id" required>  
        <label for="new_price">Nuevo Precio:</label>  
        <input type="number" name="new_price" required step="0.01">
```



```
<label for="stock_quantity">Nuevo Stock:</label>
<input type="number" name="stock_quantity" required>
<button type="submit" name="submit_update_product" onclick="return
confirmTransaction()">Actualizar Producto</button>
</form>
<script type="text/javascript">
    // Función para mostrar alerta de confirmación antes de enviar el formulario
    function confirmTransaction() {
        var confirmUpdate = confirm("¿Estás seguro de que deseas actualizar el producto?
Esta acción no se puede deshacer.");
        if (confirmUpdate) {
            return true; // Si el usuario confirma, el formulario se enviará
        } else {
            return false; // Si el usuario cancela, el formulario no se enviará
        }
    }
</script>
<?php
if (isset($_POST['submit_update_product'])) {
    // Obtener los datos del formulario
    $product_id = intval($_POST['product_id']);
    $new_price = floatval($_POST['new_price']);
    $stock_quantity = intval($_POST['stock_quantity']);
    // Llamar a la función que maneja la transacción
    update_product_with_transaction($product_id, $new_price, $stock_quantity);
}
return ob_get_clean();
}
add_shortcode('update_product_form', 'product_update_transaction_shortcode');
```



```
function update_product_with_transaction($product_id, $new_price, $stock_quantity) {
    global $wpdb;
    // Iniciar la transacción
    $wpdb->query('START TRANSACTION');
    try {
        // Actualizar el precio del producto en la tabla wp_wc_product_meta_lookup
        $price_updated = $wpdb->update(
            $wpdb->prefix . 'wc_product_meta_lookup',
            array('max_price' => $new_price), // El campo a actualizar (max_price)
            array('product_id' => $product_id), // Condición para el ID del producto
            array('%f'), // Tipo de dato para max_price
            array('%d') // Tipo de dato para product_id
        );

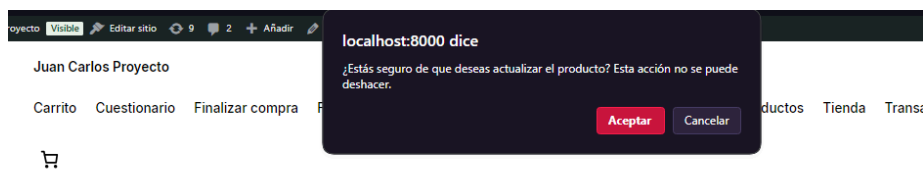
        // Actualizar el stock del producto en la misma tabla
        $stock_updated = $wpdb->update(
            $wpdb->prefix . 'wc_product_meta_lookup',
            array('stock_quantity' => $stock_quantity), // El campo a actualizar (stock_quantity)
            array('product_id' => $product_id), // Condición para el ID del producto
            array('%d'), // Tipo de dato para stock_quantity
            array('%d') // Tipo de dato para product_id
        );

        if ($price_updated !== false && $stock_updated !== false) {
            // Si ambas operaciones fueron exitosas, confirmamos la transacción
            $wpdb->query('COMMIT');
            echo '<p>Producto actualizado exitosamente.</p>';
        } else {
            // Si alguna operación falla, revertimos todas las operaciones
            $wpdb->query('ROLLBACK');
            echo '<p>Error al actualizar el producto. Los cambios han sido revertidos.</p>';
        }
    }
}
```



```
}  
} catch (Exception $e) {  
    // Si ocurre un error, revertimos todas las operaciones  
    $wpdb->query('ROLLBACK');  
    echo '<p>Error al realizar la actualización: ' . $e->getMessage() . '</p>';  
    error_log('Error al actualizar producto: ' . $e->getMessage());  
}  
}
```

## 4. Control de Permisos:



### Transacciones

ID del Producto:  Nuevo Precio:   
Nuevo Stock:    
Producto actualizado exitosamente.

Para crear un seguro de vida por si nos equivocamos en los datos y añadimos unos datos que no son añadimos un mensaje al hacer click al botón que hace que nos envíen una alerta para confirmar si son los datos correctos los que hemos añadido para que se modifiquen ya instantáneamente después de darle a aceptar en la alerta.