Prueba Técnica para Desarrollador WordPress con WooCommerce

# **Parte 1: Configuración y Entorno**

Instalación de WordPress y WooCommerce

* ✅Instala WordPress en un entorno local o de prueba.
* ✅Instala y configura WooCommerce.
* ✅Importa algunos productos de muestra.

# **Parte 2: Desarrollo de Plugins**

Plugin de Customización de WooCommerce

* ✅Crea un plugin llamado `custom-woocommerce-features`.
* ✅Dentro del plugin, agrega una función que muestre un mensaje personalizado en la página de checkout basado en el valor del carrito. Por ejemplo: "Gracias por tu compra. ¡Obtén un descuento del 10% en tu próxima compra si gastas más de $100 hoy!".
* ✅Asegúrate de que el mensaje sea dinámico y cambie según el valor del carrito.

# **Parte 3: Desarrollo de Temas y Personalización**

Modificación de Tema

* ✅Usa un tema hijo de `Storefront` (tema oficial de WooCommerce).
* ✅Agrega una sección personalizada en la página de producto individual que muestre productos relacionados de la misma categoría en una cuadrícula de 3 columnas.

# **Parte 4: Manejo de Bases de Datos**

Consulta de la Base de Datos  
 - ✅Escribe una función que realice una consulta directa a la base de datos para obtener todos los pedidos completados en el último mes y los muestre en una página de administración personalizada dentro de WordPress.  
 - ✅La página debe mostrar el ID del pedido, el nombre del cliente, la fecha del pedido y el total del pedido.

# **Parte 5: Compartir código general en un repositorio de gitHub (incluir copia de base de datos)✅**

# **Parte 6: Evaluación y Preguntas**

Preguntas Teóricas  
✅ - **¿Cómo asegurarías que tu plugin no cause conflictos con otros plugins o temas instalados?**

Rta:  
  
Para asegurar que un plugin no cause conflictos con otros plugins o temas, se pueden seguir las siguientes prácticas:

* **Uso de prefijos**: Prefijar funciones, clases, variables, hooks y archivos del plugin es esencial para evitar colisiones con otros plugins. Por ejemplo, usar myplugin\_ como prefijo en lugar de nombres genéricos.
* **Buen uso de los hooks de WordPress**: Aprovechar las acciones y filtros de WordPress para modificar o extender funcionalidades sin alterar directamente el código de otros plugins o del tema. Además, usar correctamente los *hooks* prioritarios para asegurar que el código se ejecute en el momento adecuado.
* **Incluir dependencias correctamente**: Utilizar wp\_enqueue\_script y wp\_enqueue\_style para cargar scripts y estilos, asegurándose de que no se incluyan varias versiones de la misma librería. Además, verificar que las dependencias estén correctamente versionadas y registradas.
* **Pruebas exhaustivas**: Antes de lanzar un plugin, probarlo en un ambiente de staging con una variedad de plugins populares y diferentes temas para identificar posibles conflictos.

✅- **Explica la diferencia entre acciones y filtros en WordPress. Proporciona un ejemplo de cada uno.**

Rta:

**Acciones (Actions)**: Las acciones permiten ejecutar una función personalizada en puntos específicos del ciclo de vida de WordPress. Son eventos que desencadenan funciones cuando WordPress llega a ciertos puntos, como la carga de una página o la creación de un post.

Ejemplo de actions:

function my\_custom\_action() {

// Código que se ejecuta cuando se llama la acción

echo 'Esta es mi acción personalizada.';

}

add\_action('wp\_footer', 'my\_custom\_action');

**Filtros (Filters)**: Los filtros permiten modificar o ajustar datos antes de que se envíen al navegador o a la base de datos. Los filtros reciben un valor, lo procesan y devuelven el valor modificado.

Ejemplo de filter:  
  
function modify\_content($content) {

// Añade texto al contenido de cada post

return $content . ' - Texto agregado al final del contenido.';

}

add\_filter('the\_content', 'modify\_content');

✅ - **¿Cómo manejarías la seguridad en una tienda WooCommerce? Menciona al menos tres prácticas recomendadas.**

Rta:

* **Mantener plugins, temas y WordPress actualizados**: Asegurarse de que todos los componentes del sitio estén siempre actualizados reduce la exposición a vulnerabilidades conocidas que ya hayan sido parcheadas en versiones posteriores.
* **Uso de certificados SSL**: Implementar SSL (HTTPS) garantiza que la transmisión de datos sensibles, como información de pago y datos personales, esté cifrada y sea segura.
* **Validación y sanitización de datos**: Asegurarse de que toda la entrada de usuario, especialmente en formularios, esté debidamente validada y saneada. Esto protege contra ataques como inyecciones SQL o XSS.
* **Limitación de intentos de inicio de sesión**: Utilizar plugins de seguridad que limiten los intentos de inicio de sesión para evitar ataques de fuerza bruta. Aquí funcionarían bien Wordfence, iThemes security, All In One WP Security & Firewall y entre otros.

# **Instrucciones para el Candidato**

Tiempo Estimado: 10-16 horas.  
 Entregables:  
 - Código del plugin `custom-woocommerce-features`.  
 - Tema hijo de `Storefront` modificado.  
 - Función para consulta de la base de datos.  
 - Formularios y scripts AJAX implementados.

- Respuestas a las preguntas teóricas.  
Evaluación:  
 - Funcionalidad y cumplimiento de los requisitos.  
 - Calidad y claridad del código.  
 - Uso de buenas prácticas en desarrollo y seguridad.  
 - Innovación y creatividad en las soluciones propuestas.