

## Plantilla de Documentación de Prompts

Campo	Contenido
ID del Prompt	P003-Fase 3 Qwen
Fecha de Uso	23/04/2025
Modelo de IA	Qwen2.5-coder:32b
Categoría de la Tarea	Frontend/Backend/Base de Datos/Autenticación
Tecnología	HTML/CSS/JavaScript/PHP/SQL
Texto Completo del Prompt	<p>V.1 Generar el código necesario para la <b>**Fase 3: Gestión de productos**</b> en un sistema basado en Laravel.</p> <p>#### <b>**Requerimientos**</b></p> <p><b>**CRUD de productos**</b> para vendedores y proveedores.</p> <ul style="list-style-type: none"> <li>* Los vendedores pueden gestionar productos que venden directamente a los clientes.</li> <li>* Los proveedores pueden gestionar productos que venden directamente a los vendedores.</li> <li>* Validaciones necesarias en la creación y edición de productos (ejemplo: precio positivo, stock mínimo).</li> <li>* Cada usuario solo puede gestionar sus propios productos.</li> <li>* Implementar paginación en la vista de listado de productos.</li> </ul> <p>Cada vendedor y proveedor tiene un puesto (stand) al que está asociado los productos, en caso de no tener puesto añade la opción de crearlo y gestionarlo.</p> <p><b>**Gestión de stock**</b></p> <ul style="list-style-type: none"> <li>* Opción para modificar el stock de un producto.</li> <li>* Restricciones para evitar valores negativos.</li> <li>* Mensajes de alerta cuando el stock esté bajo.</li> </ul> <p><b>**Categorización y búsqueda**</b></p> <ul style="list-style-type: none"> <li>* Filtros por categoría, precio y disponibilidad.</li> <li>* Búsqueda por nombre y descripción.</li> <li>* Ordenación por precio (ascendente/descendente) y fecha de creación.</li> </ul> <p><b>**Estructura del Código**</b></p> <ul style="list-style-type: none"> <li>* <b>**Separación por archivos**</b>: Cada controlador y modelo debe estar en su propio archivo.</li> <li>* El código generado debe organizarse en <b>**múltiples archivos**</b> en lugar de incluirlo todo en un solo bloque.</li> </ul> <p><b>**Extras (Opcionales pero recomendados)**</b></p> <ul style="list-style-type: none"> <li>* Middleware para restringir acceso a vendedores/proveedores según su rol.</li> <li>* Mensajes de éxito/error en las vistas.</li> </ul> <p>#### <b>**Formato de Entrega**</b></p> <ul style="list-style-type: none"> <li>* <b>**Controladores en archivos separados**</b>, con sus respectivos métodos.</li> <li>* <b>**Modelos**</b> con sus relaciones necesarias.</li> <li>* <b>**Vistas Blade**</b> organizadas para cada funcionalidad.</li> <li>* Rutas web organizadas en web.php.</li> <li>* <b>**Migraciones y Seeders**</b> en archivos distintos.</li> <li>* <b>**Estructura de las tablas**</b></li> </ul>

**\*\*Tabla users\*\***

Esta tabla almacena la información de todos los usuarios del sistema, independientemente de su rol.

Campo

Tipo

Descripción

Restricciones

id

INT

Identificador único del usuario

PK, AUTO\_INCREMENT

name

VARCHAR(255)

Nombre completo del usuario

NOT NULL

email

VARCHAR(255)

Correo electrónico

UNIQUE, NOT NULL

password

VARCHAR(255)

Contraseña encriptada

NOT NULL

phone

VARCHAR(20)

Número de teléfono

NULL

profile\_picture

VARCHAR(255)

Ruta a la imagen de perfil

NULL

role

ENUM

Rol del usuario: 'cliente', 'vendedor', 'proveedor'

NOT NULL

remember\_token

VARCHAR(100)

Token para "recordar sesión"

NULL

created\_at

TIMESTAMP

Fecha de creación del registro

NOT NULL

updated\_at

TIMESTAMP

Fecha de última actualización

NOT NULL

**\*\*Relaciones:\*\***

- Un usuario no cliente puede tener un puesto (stand)(1:0/1)
- Un usuario no cliente puede crear múltiples productos (1:N)
- Un usuario cliente puede realizar múltiples pedidos (1:N)
- Un usuario vendedor puede recibir múltiples pedidos (1:N)
- Un usuario proveedor puede realizar múltiples ventas a vendedores (1:N)
- Un usuario vendedor puede realizar múltiples compras a proveedores (1:N)
- Un usuario puede crear múltiples foros (1:N)
- Un usuario puede publicar múltiples temas (1:N)
- Un usuario puede escribir múltiples comentarios (1:N)
- Un usuario puede enviar y recibir múltiples mensajes (1:N)

**### \*\*2.2 Tabla stands\*\***

Almacena la información de los puestos que gestionan los vendedores en el mercado.

Campo

Tipo

Descripción

Restricciones

id

	<p>INT</p> <p>Identificador único del puesto</p> <p>PK, AUTO_INCREMENT</p> <p>user_id</p> <p>INT</p> <p>ID del usuario vendedor propietario</p> <p>FK (users.id), NOT NULL</p> <p>name</p> <p>VARCHAR(255)</p> <p>Nombre del puesto</p> <p>NOT NULL</p> <p>description</p> <p>TEXT</p> <p>Descripción detallada del puesto</p> <p>NULL</p> <p>location</p> <p>VARCHAR(255)</p> <p>Ubicación física dentro del mercado</p> <p>NOT NULL</p> <p>category</p> <p>VARCHAR(255)</p> <p>Categoría del puesto (ej. frutas, artesanía)</p> <p>NULL</p> <p>stand_picture</p> <p>VARCHAR(255)</p> <p>Ruta a la imagen de perfil</p> <p>NULL</p> <p>created_at</p> <p>TIMESTAMP</p> <p>Fecha de creación del registro</p> <p>NOT NULL</p> <p>updated_at</p> <p>TIMESTAMP</p> <p>Fecha de última actualización</p> <p>NOT NULL</p> <p><b>**Relaciones:**</b></p> <ul style="list-style-type: none"> <li>- Un puesto pertenece a un único usuario vendedor/proveedor (N:1)</li> <li>- Un puesto puede tener múltiples productos (1:N)</li> </ul> <p><b>### **2.3 Tabla products**</b></p> <p>Contiene todos los productos disponibles en el sistema, tanto de vendedores como de proveedores.</p> <p>Campo</p> <p>Tipo</p> <p>Descripción</p> <p>Restricciones</p> <p>id</p> <p>INT</p> <p>Identificador único del producto</p> <p>PK, AUTO_INCREMENT</p> <p>user_id</p> <p>INT</p> <p>ID del usuario que creó el producto</p> <p>FK (users.id), NOT NULL</p> <p>stand_id</p> <p>INT</p> <p>ID del puesto al que pertenece (si aplica)</p> <p>FK (stands.id), NULL</p> <p>name</p> <p>VARCHAR(255)</p> <p>Nombre del producto</p> <p>NOT NULL</p> <p>description</p> <p>TEXT</p> <p>Descripción detallada del producto</p> <p>NULL</p>
--	--

	<p>price DECIMAL(8,2) Precio del producto NOT NULL</p> <p>stock INT Cantidad disponible NOT NULL, DEFAULT 0</p> <p>image VARCHAR(255) Ruta a la imagen del producto NULL</p> <p>category VARCHAR(255) Categoría del producto NULL</p> <p>is_available BOOLEAN Indica si el producto está disponible NOT NULL, DEFAULT TRUE</p> <p>created_at TIMESTAMP Fecha de creación del registro NOT NULL</p> <p>updated_at TIMESTAMP Fecha de última actualización NOT NULL</p> <p><b>**Relaciones:**</b></p> <ul style="list-style-type: none"> <li>- Un producto pertenece a un único usuario (N:1)</li> <li>- Un producto puede pertenecer a un puesto (N:0/1)</li> <li>- Un producto puede estar en múltiples items de pedido (1:N)</li> <li>- Un producto puede estar en múltiples items de venta de proveedor (1:N)</li> </ul> <p>1ª Iteración: Dame las vistas, controladores y rutas para la gestión del puesto (stand) de los usuarios con rol de vendedor o proveedor:</p> <p><b>**Tabla stands**</b></p> <p>Almacena la información de los puestos que gestionan los vendedores en el mercado. Campo Tipo Descripción Restricciones id INT Identificador único del puesto PK, AUTO_INCREMENT user_id INT ID del usuario vendedor propietario FK (users.id), NOT NULL name VARCHAR(255) Nombre del puesto NOT NULL description TEXT Descripción detallada del puesto location VARCHAR(255) Ubicación física dentro del mercado NOT NULL category VARCHAR(255) Categoría del puesto (ej. frutas, artesanía) NULL stand_picture VARCHAR(255) Ruta a la imagen de perfil NULL created_at TIMESTAMP Fecha de creación del registro NOT NULL updated_at TIMESTAMP Fecha de última actualización NOT NULL</p> <p><b>**Relaciones:**</b></p> <ul style="list-style-type: none"> <li>* Un puesto pertenece a un único usuario vendedor/proveedor (N:1)</li> <li>* Un puesto puede tener múltiples productos (1:N)</li> </ul> <p>2ª Iteración: Modifica la función para que cree el puesto(stand) tomando el id del usuario autenticado:</p> <pre> ```php public function store(Request \$request) {     \$validatedData = \$request-&gt;validate([         'name' =&gt; 'required string max:255',         'description' =&gt; 'nullable string',         'location' =&gt; 'required string max:255',         'category' =&gt; 'nullable string max:255',         'stand_picture' =&gt; 'nullable image mimes:jpeg,png,jpg,gif,svg max:2048',     ]);     if (\$request-&gt;hasFile('stand_picture')) {         \$path = \$request-&gt;file('stand_picture')-&gt;store('images/stands', 'public');         \$validatedData['stand_picture'] = \$path;     } } </pre>
--	--

	<pre> Stand::create(\$validatedData); return redirect()-&gt;route('stands.index')-&gt;with('success', 'Stand created successfully.');</pre> <p>...</p> <p>3ª Iteración: He creado una función llamada verpropio para que el usuario pueda ver solo su propio puesto pero me da: <b>**Function () does not exist cada vez que intento entrar.**</b></p> <p><b>**Función**</b></p> <pre> ```php public function verpropio() {     \$stands = Stand::where('user_id', auth()-&gt;user()-&gt;id)-&gt;get();     return view('stands.index', compact('stands')); } ... <b>**ruta**</b> ```php Route::get('/stand/propio', [StandController::class, 'verpropio']-&gt;middleware(['auth', 'checkUserRole'])-&gt;name('stand.propio');</pre> <p>...</p> <p>4ª Iteración: Como puedo hacer que se muestre el enlace si solo el usuario no tiene puesto:</p> <pre> ```php @if (empty(auth()-&gt;user()-&gt;Stand()))     &lt;a href="{{ route('stands.create') }}" class="btn btn-primary mb-3"&gt;Create New Stand&lt;/a&gt; @endif ... 5ª Iteración: modifica el checkbox para que devuelva boolean: ```php &lt;div class="mb-3 form-check"&gt;     &lt;input type="checkbox" name="is_available" id="is_available" class="form-check-input @error('is_available') is-invalid @enderror" {{ old('is_available', true) ? 'checked' : " " }}&gt;     &lt;label for="is_available" class="form-check-label"&gt;Available&lt;/label&gt;     @error('is_available')         &lt;span class="invalid-feedback"&gt;{{ \$message }}&lt;/span&gt;     @enderror &lt;/div&gt; ... </pre>
--	--



## Objetivo General

Desarrollar un módulo completo en Laravel que permita a los usuarios con roles de **vendedor** y **proveedor** gestionar productos y su disponibilidad, garantizando control de stock, categorización, búsqueda avanzada y seguridad basada en roles.



## Requerimientos Funcionales

### 1. Gestión de Productos (CRUD)

- Cada **usuario autenticado** podrá gestionar únicamente sus productos.
- Soporte para creación, visualización, edición y eliminación de productos.
- Asociación de productos a un puesto (**stand**), si el usuario tiene uno.
  - Si no tiene stand, se mostrará la opción para crearlo.

### 2. Validaciones

- El **precio** debe ser mayor que cero.
- El **stock** debe ser mayor o igual a cero.
- Todos los campos deben ser validados en la creación y edición.

### 3. Gestión de Stock

- Opción para modificar stock de productos.
- Restricción para evitar valores negativos.
- Alertas visuales cuando el stock esté por debajo de un umbral (e.g. < 5 unidades).




### 4. Categorización y Búsqueda

- Filtros por **categoría**, **precio**, **disponibilidad**.
- Búsqueda por **nombre** y **descripción**.
- Ordenación por **precio** (asc/desc) y **fecha de creación**.



## Requerimientos Técnicos

- Separar la lógica en:
  - Modelos (**Product**, **Stand**, **User**)
  - Controladores (**ProductController**)
  - Rutas (**web.php**)
  - Vistas Blade (**products/**)
- Implementar paginación en el listado de productos.

	<ul style="list-style-type: none"><li>• Middleware que verifique el <b>rol del usuario</b> antes de permitir acceso a ciertas rutas.</li><li>• Mostrar mensajes de éxito/error al usuario.</li></ul>	
	<div> <b>Estructura de Base de Datos</b></div> <div><b>Tabla <span>users</span></b><ul style="list-style-type: none"><li>• Contiene campos como <span>id</span>, <span>name</span>, <span>email</span>, <span>role</span> (cliente, vendedor, proveedor), etc.</li><li>• Relación: un usuario puede tener un <b>stand</b>, y puede tener múltiples <b>productos</b>.</li></ul></div> <div><b>Tabla <span>stands</span></b><ul style="list-style-type: none"><li>• Relacionado con usuarios no clientes.</li><li>• Atributos: <span>user_id</span>, <span>name</span>, <span>location</span>, <span>category</span>, <span>stand_picture</span>.</li></ul></div> <div><b>Tabla <span>products</span></b><ul style="list-style-type: none"><li>• Atributos: <span>user_id</span>, <span>stand_id</span>, <span>name</span>, <span>description</span>, <span>price</span>, <span>stock</span>, <span>image</span>, <span>category</span>, <span>is_available</span>.</li><li>• Relación: pertenece a un <span>user</span>, y puede estar asociado a un <span>stand</span>.</li></ul></div>	
	<div> <b>Formato de Entrega</b></div> <ul style="list-style-type: none"><li>• <b>Migraciones</b> y <b>Seeders</b> separados.</li><li>• <b>Modelos</b> con relaciones correctamente definidas.</li><li>• <b>Controladores</b> con métodos bien definidos para CRUD.</li><li>• <b>Vistas Blade</b> por funcionalidad (<span>index</span>, <span>create</span>, <span>edit</span>, <span>show</span>).</li><li>• <b>Rutas</b> definidas en <span>web.php</span>.</li></ul>	
	<div> <b>Extras (Opcionales pero Recomendados)</b></div> <ul style="list-style-type: none"><li>• Middleware para control de acceso basado en roles.</li><li>• Alertas en la interfaz cuando un producto tiene bajo stock.</li><li>• Upload y visualización de imágenes de producto y stand.</li></ul>	
Long. del Prompt	V.1 sin iteraciones: 5240 con iteraciones: 8344	V.1.1 2468
Documentos incluidos	Sí, específicos de la base de datos y especificaciones	
Ejemplos Incluidos	Sí, estructuras de la base de datos y código que requiere de modificaciones	
Calidad de la Respuesta	V.1 5	V.1.1 5

P. Ponderada	V.1 5	V.1.1 5
Prob. de Código	no	
Iteraciones	5	
Mejoras Recomendadas	Mayor especificación para evitar iteraciones	
Resultados de aprendizaje vinculados	V1 R3: Conoce, distingue y utiliza tecnologías, lenguajes y estándares propios de la computación web. R4: Diseña, implementa, valida y ofrece servicios web. R15: Aplica las competencias adquiridas en la realización de una tarea de forma autónoma. R12: Utiliza recursos para buscar información necesaria y cita adecuadamente. V1.1 R3: Conoce y usa tecnologías y estándares web. R4: Diseña e implementa servicios web funcionales. R1: Comprende relaciones entre entidades del sistema. R15: Aplica conocimientos de forma autónoma.	
Competencias relacionadas	V1 CETM-6: Capacidad de diseñar arquitecturas de redes y servicios telemáticos. CETM-7: Programación de servicios y aplicaciones en red. CG-5: Capacidad para diseñar y usar aplicaciones en el entorno de las telecomunicaciones. CB-2 / CB-5: Aplicación práctica de conocimientos y autonomía en el aprendizaje. CT-1 / CT-2: Comunicación adecuada y trabajo colaborativo en entornos técnicos. V1.1 CETM-6: Diseño de arquitecturas de redes y servicios telemáticos. CETM-7: Programación de servicios web. CG-5: Desarrollo de aplicaciones en telecomunicaciones. CT-1 y CT-2: Comunicación efectiva y trabajo colaborativo. CB-2 y CB-5: Aplicación profesional y autonomía de aprendizaje.	
Objetivos de la asignatura	V1 OBJ-1: Aprender tecnologías web del lado del cliente y servidor. OBJ-2: Desarrollar un sistema web completo con base de datos. OBJ-3: Uso de herramientas de desarrollo para una tienda de comercio electrónico. V1.1 OBJ-1: Aprender tecnologías web (cliente/servidor). OBJ-2: Implementar un sistema web completo. OBJ-3: Conocer herramientas para desarrollo de e-commerce.	

## Tabla de Comparación de Versiones de Prompt

Versión	Camb. Clave	Cla.	Esp.	Prec. Téc.	Ctx.	Estruc.	Cal. Cód.	Punt. Pond.	Mej.
1.0	Inicial	5	5	5	5	5	5	5	-
1.1	Se agregaron especificaciones técnicas	5	5	5	5	5	5	5	0

### Leyenda de Abreviaturas

- **Camb. Clave** = Cambios Clave
- **Cla.** = Claridad
- **Esp.** = Especificidad
- **Prec. Téc.** = Precisión Técnica
- **Ctx.** = Contexto
- **Estruc.** = Estructura
- **Cal. Cód.** = Calidad del Código
- **Punt. Pond.** = Puntuación Ponderada
- **Mej.** = Mejora