# Marco de Análisis de Prompts para IA de Desarrollo Web

## Criterios de Evaluación y Sistema de Puntuación (Escala de 1 a 5)

| Categoría | Descripción | Criterios de Puntuación | Peso |
|---|---|---|---|
| Claridad | Nivel de claridad y ausencia de ambigüedad | 1: Altamente ambiguo y confuso<br>2: Varios elementos poco claros<br>3: Mayormente claro con alguna ambigüedad<br>4: Claro con mínima ambigüedad<br>5: Perfectamente claro y sin ambigüedades | 15% |
| Especificidad | Nivel de detalle y precisión en los requisitos | 1: Extremadamente vago<br>2: Sólo información básica<br>3: Detalle moderado<br>4: Requisitos detallados<br>5: Especificaciones precisas y completas | 20% |
| Precisión Técnica | Uso correcto de términos y conceptos técnicos | 1: Errores técnicos graves<br>2: Varias imprecisiones técnicas<br>3: Problemas menores técnicos<br>4: Mayormente preciso<br>5: Totalmente preciso | 15% |
| Provisión de Contexto | Información de fondo y contexto del proyecto | 1: Sin contexto<br>2: Contexto mínimo<br>3: Contexto básico relevante<br>4: Buena información contextual<br>5: Contexto completo | 10% |
| Calidad de la Estructura | Organización y flujo lógico del prompt | 1: Estructura caótica<br>2: Organización deficiente<br>3: Estructura adecuada<br>4: Bien organizado<br>5: Estructura lógica excelente | 10% |
| Calidad del Código Generado | Calidad del código resultante | 1: Código no funcional<br>2: Funcional pero de baja calidad<br>3: Calidad aceptable<br>4: Buena calidad con problemas menores<br>5: Código excelente y listo para producción | 30% |

---

# Plantilla de Documentación de Prompts

| Campo | Contenido |
|---|---|
| ID del Prompt | P0020-APIINTEGRATION |
| Fecha de Uso | 12/03/2025 |
| Modelo de IA | Claude Sonnet 3.7 |
| Categoría de la Tarea | Frontend/Backend/Autenticación |
| Tecnología | HTML/CSS/JavaScript/PHP |

| | |
|---|---|
| Texto Completo del Prompt | V.1 Prompt inicial: Dame el código para: Fase 6: Integraciones API<br>* Desarrollo de API interna<br>* Integración con API de clima<br>* Implementación de AJAX para actualización dinámica<br>1º.Iteración: Pon bien esto para tenerlo en la misma vista: <section class="mb-5"><br> <br><h2>Productos destacados</h2><br> <br> <br><form id="search-form" class="row g-3"><br> <br><label for="search-query" class="form-label">Buscar productos:</label><br><input type="text" class="form-control" id="search-query" placeholder="Nombre o descripción..."><br> <br> <br><label for="search-category" class="form-label">Categoría:</label><br><select class="form-select" id="search-category"><br><option value="">Todas las categorías</option><br><option value="frutas">Frutas</option><br><option value="verduras">Verduras</option><br><option value="lacteos">Lácteos</option><br><option value="carnes">Carnes</option><br><option value="panaderia">Panadería</option><br><option value="artesania">Artesanía</option><br><option value="conservas">Conservas</option><br><option value="otros">Otros</option><br></select><br> <br> <br><button type="submit" class="btn btn-primary w-100">Buscar</button><br> <br></form><br> <br> <br> <br> <br> <br>Cargando...<br> <br> Cargando productos... <br> <br> <br> <br> <br> <br> <br>Cargando...<br> <br> Cargando productos destacados... <br> <br> <br></section><br><br>2º.Iteración: No aparecen las tarjetas de los productos al terminar de cargar:<br>3º.Iteración: Arregla los conflictos que encuentres pues no puedo desplegar el navdropdown:<br>4º.Iteración: No se muestran los productos destacados obtenidos mediante el js:<br>5º.Iteración: Esto es el js:<br>6º.Iteración: Deja app.blade.php con lo basico y home.blade.php con lo necesario para los js y apis<br>7º.Iteración: los productos buscados no aparecen: // public/js/product-search.js<br>/**<br>* Función para buscar productos dinámicamente<br>*/<br>function searchProducts() {<br>const query = $('#search-query').val();<br>const category = $('#search-category').val();<br>// Mostrar indicador de carga<br>$('#search-results-container').html(' <span<br>class="visually-hidden">Cargando... ');<br>$.ajax({<br>url: '/api/productos/buscar',<br>method: 'GET',<br>data: {<br>query: query,<br>category: category<br>},<br>success: function(response) {<br>// Limpiar el contenedor<br>$('#search-results-container').empty();<br>if (response.length === 0) {<br>$('#search-results-container').html(' No se encontraron productos que coincidan con tu búsqueda. ');<br>return;<br>}<br>// Crear contenedor de resultados<br>const resultsGrid = $('<div class="row"></div>');<br>// Generar HTML para cada producto<br>response.forEach(function(product) {<br>const imageUrl = product.image \|\| '/img/product-placeholder.jpg';<br>const standName = product.stand ? product.stand.name : 'Vendedor independiente';<br>const productCard =<br><div class="col-md-4 col-sm-6 mb-4"><br><div class="card h-100 product-card"><br><img src="${imageUrl}" class="card-img-top product-image" alt="${product.name}"><br> |

```
                        <h5 class="card-title">${product.name}</h5>
                        <p class="card-text text-truncate">${product.description || 'Sin descripción'}</p>
                        <p class="card-text"><strong>${product.price.toFixed(2)} €</strong></p>
                        <p class="card-text small">Puesto: ${standName}</p>
                    </div>
                    <div class="card-footer d-flex justify-content-between">
                        <a href="/productos/${product.id}" class="btn btn-sm btn-outline-primary">Ver detalle</a>
                        <button class="btn btn-sm btn-success add-to-cart" data-id="${product.id}">
                            <i class="fas fa-cart-plus"></i> Añadir
                        </button>
                    </div>
                </div>
            </div>
        `;
        resultsGrid.append(productCard);
      });
      // Añadir resultados al contenedor
      $('#search-results-container').append(resultsGrid);
      // Inicializar los botones de "Añadir al carrito"
      initAddToCartButtons();
    },
    error: function(error) {
      console.error('Error al buscar productos:', error);
      $('#search-results-container').html('<div class="alert alert-danger">Error al realizar la búsqueda. Intente nuevamente más tarde.</div>');
    }
  });
}
// Inicializar eventos
$(document).ready(function() {
  // Evento de envío del formulario de búsqueda
  $('#search-form').on('submit', function(e) {
    e.preventDefault();
    searchProducts();
  });
  // Búsqueda en tiempo real (opcional, con debounce)
  let searchTimeout;
  $('#search-query, #search-category').on('input change', function() {
    clearTimeout(searchTimeout);
    searchTimeout = setTimeout(function() {
      searchProducts();
    }, 500); // Esperar 500ms después de que el usuario deje de escribir
  });
});
```

## 8º.Iteración: Haz que estos dos js sean compatibles en la misma vista:

```
// public/js/product-search.js
/**
 * Función para buscar productos dinámicamente
 */
function searchProducts() {
  const query = $('#search-query').val();
  const category = $('#search-category').val();
  // Mostrar indicador de carga
  $('#search-results-container').html('<div class="text-center"><div class="spinner-border text-primary" role="status"><span
class="visually-hidden">Cargando...</span></div></div>');
  $.ajax({
    url: '/api/productos/buscar',
    method: 'GET',
    data: {
      query: query,
      category: category
    },
    success: function(response) {
      // Limpiar el contenedor
      $('#search-results-container').empty();
      if (response.length === 0) {
        $('#search-results-container').html('<div class="alert alert-info">No se encontraron productos que coincidan con tu búsqueda.</div>');
        return;
      }
      // Crear contenedor de resultados
      const resultsGrid = $('<div class="row"></div>');
      // Generar HTML para cada producto
      response.forEach(function(product) {
        const imageUrl = product.image || '/img/product-placeholder.jpg';
        const standName = product.stand ? product.stand.name : 'Vendedor independiente';
        const productCard =
          <div class="col-md-4 col-sm-6 mb-4">
            <div class="card h-100 product-card">
              <img src="${imageUrl}" class="card-img-top product-image" alt="${product.name}">
              <div class="card-body">
                <h5 class="card-title">${product.name}</h5>
                <p class="card-text text-truncate">${product.description || 'Sin descripción'}</p>
                <p class="card-text"><strong>${product.price.toFixed(2)} €</strong></p>
                <p class="card-text small">Puesto: ${standName}</p>
              </div>
              <div class="card-footer d-flex justify-content-between">
                <a href="/productos/${product.id}" class="btn btn-sm btn-outline-primary">Ver detalle</a>
                <button class="btn btn-sm btn-success add-to-cart" data-id="${product.id}">
                  <i class="fas fa-cart-plus"></i> Añadir
                </button>
              </div>
            </div>
          </div>
        `;
        resultsGrid.append(productCard);
      });
      // Añadir resultados al contenedor
      $('#search-results-container').append(resultsGrid);
      // Inicializar los botones de "Añadir al carrito"
      initAddToCartButtons();
    },
    error: function(error) {
      console.error('Error al buscar productos:', error);
      $('#search-results-container').html('<div class="alert alert-danger">Error al realizar la búsqueda. Intente nuevamente más tarde.</div>');
```

```
        }
    });
}
// Inicializar eventos
$(document).ready(function() {
    // Evento de envío del formulario de búsqueda
    $('#search-form').on('submit', function(e) {
        e.preventDefault();
        searchProducts();
    });
    // Búsqueda en tiempo real (opcional, con debounce)
    let searchTimeout;
    $('#search-query, #search-category').on('input change', function() {
        clearTimeout(searchTimeout);
        searchTimeout = setTimeout(function() {
            searchProducts();
        }, 500); // Esperar 500ms después de que el usuario deje de escribir
    });
});
```

9º.Iteración: Modifica home.blade.php para que no salga el circulo de cargando en la seccion de buscar productos, pues este se queda infinitamente hasta que le das al boton de buscar

V.1.1 Objetivo**:** Implementar la fase 6 del proyecto, centrada en integraciones API y actualización dinámica de la interfaz con AJAX en Laravel.

## Requerimientos Técnicos

1. Desarrollo de API interna
    - Crear un controlador en Laravel (`ApiController`) que proporcione los siguientes endpoints:
        - `GET /api/productos/destacados`: Obtener los productos destacados con relación a su puesto de venta.
        - `GET /api/puestos/cercanos`: Listar los puestos de venta cercanos, ordenados alfabéticamente.
        - `GET /api/productos/buscar?query=&category=`: Permitir búsqueda por nombre y categoría de productos.
        - `GET /api/mercado/estadisticas`: Devolver estadísticas del mercado (cantidad de vendedores, productos y puestos).
    - Implementar los modelos y relaciones necesarios en Laravel para estas funcionalidades.
    - Asegurar que las respuestas sean en formato JSON estructurado correctamente.
2. Integración con API de Clima (OpenWeatherMap)
    - Implementar un método en `ApiController` (`GET /api/clima`) que recupere los datos climáticos de San Mateo, Gran Canaria.
    - Utilizar la API de OpenWeatherMap con una clave de API configurable en `.env`.
    - Implementar caché en Laravel para almacenar los datos por 30 minutos y reducir llamadas innecesarias.
    - El resultado debe formatearse en JSON, incluyendo temperatura, condición climática, humedad y velocidad del viento.
3. Implementación de AJAX para actualización dinámica
    - Implementar en JavaScript (jQuery) las siguientes funcionalidades con AJAX:
        - Widget de clima: Recuperar datos desde `/api/clima` y actualizar dinámicamente la interfaz sin recargar la página.
        - Carga de productos destacados: Cargar dinámicamente los productos destacados desde `/api/productos/destacados`.
        - Búsqueda de productos: Implementar un formulario de búsqueda interactivo que use AJAX para consultar `/api/productos/buscar`.
        - Botón "Añadir al carrito": Implementar eventos dinámicos en botones de productos destacados y buscados para enviar la información a `/carrito/agregar` sin recargar la página.
    - Asegurar que no haya conflictos entre los scripts de `featured-products.js` y `product-search.js`.
    - Manejo adecuado de errores y estados de carga en la interfaz.
4. Estructura de Vistas (Blade Templates en Laravel)
    - app.blade.php (Layout principal):
        - Debe incluir la carga correcta de Bootstrap, jQuery y otros scripts de terceros en el orden adecuado.
        - Mantener una estructura limpia sin carga redundante de librerías.
    - home.blade.php (Página de inicio):

| | | |
|---|---|---|
| | ■ Incluir el widget de clima, la sección de productos destacados y el formulario de búsqueda de productos en la misma vista.<br>■ Asegurar que el indicador de carga solo aparezca cuando una búsqueda esté activa.<br>■ Solo cargar `featured-products.js` y `product-search.js` cuando la página de inicio esté activa.<br><br>Consideraciones Adicionales<br>● Depuración: Incluir mensajes de consola (`console.log`) en cada función AJAX para facilitar la identificación de errores.<br>● Seguridad: Utilizar tokens CSRF en las solicitudes `POST`.<br>● Optimización: Minimizar el número de llamadas API innecesarias utilizando caché y eventos bien gestionados en JavaScript. | |
| Long. del Prompt | V.1 Prompt inicial: 158 caracteres<br>Junto iteraciones: 10343 caracteres | V.1.1 3018 caracteres |
| Documentos incluidos | Sí, dos documentos con la información de la base de datos y las especificaciones del proyecto | |
| Ejemplos Incluidos | No | |
| Calidad de la Respuesta | V.1 4 | V.1.1 5 |
| P. Ponderada | V.1 3,67 | V.1.1 5 |
| Prob. de Código | Conflictos entre códigos javascript | |
| Iteraciones | 9 | |
| Mejoras Recomendadas | Mayor precisión de los requerimientos en el prompt para evitar iteraciones adicionales. | |

---

## Tabla de Comparación de Versiones de Prompt

| Versión | Camb. Clave | Cla. | Esp. | Prec. Téc. | Ctx. | Estruc. | Cal. Cód. | Punt. Pond. | Mej. |
|---|---|---|---|---|---|---|---|---|---|
| 1.0 | Inicial | 4 | 3 | 4 | 3 | 4 | 4 | 3,67 | - |
| 1.1 | Se agregaron especificaciones técnicas | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 1,33 |

**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