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

Plantilla de Documentación de Prompts

Campo	Contenido
ID del Prompt	P0011-RefactoringCode
Fecha de Uso	05/03/2025
Modelo de IA	Claude Sonnet 3.7
Categoría de la Tarea	Backend/Autenticación
Tecnología	HTML/CSS/JavaScript/PHP/SQL/etc.

Texto Completo del Prompt	V.1 Refactoriza si es posible:
	`// Get the authenticated user`
	`$user = Auth::user();`
	`// If no user is logged in, return Vendedores products`
	`if (!$user) {`
	`return $query->whereHas('stand', function($q) {`
	`$q->whereHas('user', function($userQ) {`
	`$userQ->where('role', 'vendedor');`
	`});`
	`});`
	`}else{`
	`switch ($user->role) {`
	`case 'cliente':`
	`// Clientes can see products from all vendedores`
	`return $query->whereHas('stand', function($q) {`

```
                $q->whereHas('user', function($userQ) {
                    $userQ->where('role', 'vendedor');
                });
            });
        case 'vendedor':
            // Vendedores can see:
            // 1. Their own products
            // 2. Products from other vendedores
            // 3. Products from proveedores
            return $query->whereHas('stand', function($q) use ($user) {
                $q->where(function($subQ) use ($user) {
                    $subQ->whereHas('user', function($userQ) use ($user) {
                        $userQ->where('role', 'vendedor')
                            ->orWhere('role', 'proveedor')
                            ->orWhere('id', $user->id);
                    });
                });
            });
        case 'proveedor':
            // Proveedores can see only their own products
            return $query->whereHas('stand', function($q) use ($user) {
                $q->whereHas('user', function($userQ) use ($user) {
                    $userQ->where('id', $user->id);
                });
            });
        default:
            // For any other role or unexpected scenario, return empty result
            return $query->whereRaw('1 = 0');
    }
  }
}
```

V.1.1 Refactoriza el siguiente código de Laravel para mejorar su legibilidad, reducir la repetición y optimizar su estructura manteniendo su funcionalidad.

Requisitos de la refactorización:

El código debe seguir las mejores prácticas de Laravel y PHP.

Debe ser más compacto sin perder claridad ni modificar su lógica central.

Se debe evitar la repetición de código mediante estructuras más eficientes.

Se recomienda el uso de funciones modernas de PHP (como arrow functions si es aplicable).

La solución debe ser fácil de extender en caso de que se necesiten agregar más roles o modificar reglas de visibilidad.

Código a refactorizar:

```
// Get the authenticated user
$user = Auth::user();

// If no user is logged in, return Vendedores products
if (!$user) {
    return $query->whereHas('stand', function($q) {
        $q->whereHas('user', function($userQ) {
            $userQ->where('role', 'vendedor');
        });
    });
} else {
    switch ($user->role) {
        case 'cliente':
            // Clientes can see products from all vendedores
            return $query->whereHas('stand', function($q) {
                $q->whereHas('user', function($userQ) {
                    $userQ->where('role', 'vendedor');
                });
            });
        case 'vendedor':
            // Vendedores can see:
            // 1. Their own products
            // 2. Products from other vendedores
            // 3. Products from proveedores
            return $query->whereHas('stand', function($q) use ($user) {
                $q->where(function($subQ) use ($user) {
                    $subQ->whereHas('user', function($userQ) use ($user) {
                        $userQ->where('role', 'vendedor')
                            ->orWhere('role', 'proveedor')
                            ->orWhere('id', $user->id);
                    });
                });
            });
        case 'proveedor':
            // Proveedores can see only their own products
            return $query->whereHas('stand', function($q) use ($user) {
                $q->whereHas('user', function($userQ) use ($user) {
                    $userQ->where('id', $user->id);
                });
            });
        default:
            // For any other role or unexpected scenario, return empty result
            return $query->whereRaw('1 = 0');
    }
}
```

Importante:

Si la refactorización incluye cambios estructurales significativos, explica las mejoras realizadas y justifica las decisiones tomadas.

Long. del Prompt	V.1 2080 caracteres	V.1.1 2508 caracteres
Documentos incluidos	No	
Ejemplos Incluidos	No	
Calidad de la Respuesta	V.1 5	V.1.1 5
P. Ponderada	V.1 4,17	V.1.1 4,83
Prob. de Código	Ninguno	
Iteraciones	0	
Mejoras Recomendadas	Mayor especificación sobre los objetivos de la refactorización, y una descripción más detallada del contexto del código en la aplicación.	

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	4	5	3	4	5	4,17	-
1.1	Se añadieron requisitos específicos para orientar la refactorización.	5	5	5	4	5	5	4,83	0,67

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