

Reporte de Mejoras — Mi Gatito Regalón

Fecha: 2026-05-17

Branch: feature/development-ai

Alcance: Análisis de código del paquete Shop (vistas, CSS, componentes Vue) desde perspectiva de negocio y diseño UX/UI.

Resumen ejecutivo

El storefront tiene una base funcional sólida sobre Bagisto. Las personalizaciones existentes (carousel, CSS de marca, WhatsApp flotante, redes sociales en footer) están bien orientadas. Sin embargo, se identificaron **19 mejoras** agrupadas en 5 áreas: correcciones de bugs, alineación de marca, UX de conversión, campaña estacional y deuda técnica menor.

ÁREA 1 — Bugs y errores de código

M01 · Memory leak en el carousel (ALTA PRIORIDAD)

Archivo: packages/Webkul/Shop/src/Resources/views/components/carousel/index.blade.php

Problema: Los hooks `beforeUnmount` (línea 118) y `beforeDestroy` (línea 122) llaman a `this.setSliderHeight`, pero ese handler nunca se registró en `window.addEventListener`. El listener registrado en `init()` (línea 144) es `this.setPositionByIndex`. El listener de `resize` registrado en `mounted()` es `this.updateSliderHeight`. Resultado: el event listener de `resize` **nunca se elimina** al destruir el componente → memory leak en navegación SPA.

Corrección:

```
// beforeUnmount / beforeDestroy:  
window.removeEventListener('resize', this.updateSliderHeight);  
window.removeEventListener('resize', this.setPositionByIndex);
```

M02 · Método muerto `setSliderHeight` en el carousel

Archivo: packages/Webkul/Shop/src/Resources/views/components/carousel/index.blade.php
línea 287

Problema: El método `setSliderHeight` existe pero nunca es llamado por ningún listener activo (fue reemplazado por `updateSliderHeight`). Es código muerto que confunde.

Corrección: Eliminar el método `setSliderHeight` o consolidarlo con `updateSliderHeight`.

M03 · Número de WhatsApp hardcodedo en Blade

Archivo: packages/Webkul/Shop/src/Resources/views/components/layouts/footer/index.blade.php
línea 104

Problema: El número 51924687157 está directamente en el template. Cualquier cambio de número requiere un deploy.

Corrección: Mover a configuración de admin: `core()->getConfigData('general.general.whatsapp_button')` y leer desde ahí.

M04 · Script JS de WhatsApp puede fallar si el elemento no existe

Archivo: packages/Webkul/Shop/src/Resources/views/components/layouts/footer/index.blade.php líneas 128-140

Problema: El script accede a `document.getElementById("btnFooterWhatsapp")` incondicionalmente, pero el botón solo se renderiza si `show_footer` es true. Si está desactivado, el script lanza un error en consola (`Cannot read properties of null`).

Corrección: Envolver el script en `@if(core()->getConfigData('general.general.whatsapp_button.show_f` o agregar un guard `if (!buttonWhatsapp) return;`.

ÁREA 2 — Alineación de marca e identidad visual

M05 · Botón “Realizar pedido” no usa color de marca

Archivo: packages/Webkul/Shop/src/Resources/views/checkout/onepage/index.blade.php línea 110

Problema: El botón de confirmar pedido usa `bg-navyBlue` (azul marino del tema base de Bagisto), no el naranja de marca `rgb(255 152 0)`. Es el CTA más importante de toda la tienda y no sigue la identidad visual.

Corrección:

```
<!-- Cambiar: -->
class="primary-button w-max py-3 px-11 bg-navyBlue rounded-2xl ..."
<!-- Por: -->
class="primary-button w-max py-3 px-11 rounded-2xl ..."
```

Y en `custom.css` asegurar que `.primary-button` use el naranja de marca.

M06 · Botón “Iniciar sesión” en mobile no usa color de marca

Archivo: packages/Webkul/Shop/src/Resources/views/components/layouts/header/mobile/index.blade.php línea 199

Problema: El botón usa `bg-navyBlue` en lugar del naranja de marca.

Corrección: Reemplazar `bg-navyBlue` por la clase de botón primario del proyecto.

M07 · Panel mobile con color rosa sin justificación de marca

Archivo: packages/Webkul/Shop/src/Resources/assets/css/custom.css línea 48

Problema: `.content-panel-mobile { background-color: rgb(239 125 129 / 75%) }` usa un rosa semi-transparente que no pertenece a la paleta base (naranja) ni a ninguna campaña activa. Genera inconsistencia visual en el menú lateral mobile.

Corrección: Reemplazar por el naranja de marca o alinear con la campaña vigente.

M08 · CSS muerto en `.categoryTitleSlugHome`

Archivo: packages/Webkul/Shop/src/Resources/assets/css/custom.css líneas 53-62

Problema: La regla `.categoryTitleSlugHome` incluye `display: none !important` que suprime todo el bloque. Es código muerto que ocupa espacio y puede confundir.

Corrección: Eliminar el bloque completo si no se usa, o remover el `display: none !important` si se planea activar.

M09 · Reglas CSS vacías en breakpoint 1022px

Archivo: `packages/Webkul/Shop/src/Resources/assets/css/custom.css` líneas 187-193

Problema: El bloque `@media(max-width: 1022px)` contiene dos reglas sin declaraciones (`.header-categories {}` y `.header-categories a {}`). Son residuos de desarrollo que aumentan el tamaño del CSS compilado.

Corrección: Eliminar las reglas vacías.

ÁREA 3 — UX de conversión (impacto directo en ventas)

M10 · No hay CTA sticky en mobile en la página de producto

Contexto de negocio: 70 %+ del tráfico es móvil. El arquetipo primario (mujeres 25-40) toma decisiones rápidas desde el teléfono.

Problema: La ficha de producto no tiene un botón “Agregar al carrito” o “Personalizar ahora” fijo en la parte inferior del viewport mobile. El usuario debe hacer scroll hasta encontrar el botón.

Archivo a crear: Override de `resources/themes/default/views/products/view.blade.php`

Corrección propuesta: Añadir un `<div class="fixedbottom-0 left-0 w-full z-50 p-4 bg-white shadow-lg lg:hidden">` con el CTA visible en todo momento en mobile.

M11 · Campo de cupón visible por defecto en carrito

Archivo: `packages/Webkul/Shop/src/Resources/views/checkout/cart/coupon.blade.php`

Problema: El campo de cupón está desplegado por defecto. Mostrar este campo genera “abandono de esperanza de descuento” en usuarios que no tienen cupón, aumentando la tasa de abandono del carrito.

Corrección: Colapsar por defecto el campo y mostrarlo solo al hacer click en “¿Tienes un código de descuento?”.

M12 · No hay badges de producto personalizados

Archivo: `packages/Webkul/Shop/src/Resources/views/components/products/card.blade.php` líneas 42-50

Problema: Solo existen badges “Sale” y “New”. Para una tienda de regalos personalizados son más relevantes badges como “Personalizable”, “Más vendido”, “Edición especial”, “Agotándose”.

Corrección: Agregar lógica para leer atributos o etiquetas del producto y renderizar badges adicionales en la card. Requiere coordinación con configuración de productos en Admin.

M13 · Texto del CTA en card de producto no es emocional

Archivo: `packages/Webkul/Shop/src/Resources/views/components/products/card.blade.php` línea 94

Problema: El botón dice `@lang('shop::app.components.products.card.add-to-cart')` que en español es “Añadir al carrito” — frío y transaccional.

Corrección: Cambiar el string de traducción para productos personalizados a “Personalizar ahora” o “Quiero este detalle”. Esto puede hacerse en el archivo de traducciones `packages/Webkul/Shop/src/Resources/lang/es/app.php`.

M14 · No hay sellos de confianza en el checkout

Contexto de negocio: El arquetipo primario no es necesariamente técnico y puede dudar antes de introducir datos de pago.

Archivo: packages/Webkul/Shop/src/Resources/views/checkout/onepage/summary.blade.php

Problema: No hay sellos visuales de seguridad (SSL, métodos de pago aceptados, política de cambios) cerca del botón de confirmar pedido.

Corrección: Agregar debajo del botón de pago un bloque con íconos de métodos de pago aceptados y un texto breve de garantía (“Compra 100% segura · Cambios sin complicaciones”).

ÁREA 4 — Campañas estacionales (San Valentín / Día de las Madres)

M15 · No existe estructura de landing para campañas estacionales

Contexto de negocio: San Valentín y Día de las Madres son los dos picos de ventas del año.

Problema: No existe ninguna vista, componente o sección dedicada a campañas estacionales. Actualmente dependería de contenido estático administrado desde el panel (tipo `STATIC_CONTENT`), lo cual es limitado y no escala visualmente para un hero banner con countdown.

Corrección propuesta: - Crear un componente `<x-shop::campaign-banner>` que lea una configuración de “campaña activa” desde el admin. - Incluir: hero full-screen, texto de campaña, cuenta regresiva (JS puro), enlace a categoría de campaña. - Paleta: rojos/rosas para San Valentín; lavanda/dorado para Día de las Madres, controlada por configuración.

M16 · No hay sección fija de campaña sobre el fold en home

Archivo: packages/Webkul/Shop/src/Resources/views/home/index.blade.php

Problema: La home itera sobre personalizaciones del admin, pero no tiene lógica para posicionar una sección de campaña **obligatoriamente sobre el fold** durante la vigencia de una campaña. El carousel ocupa ese espacio pero no comunica urgencia.

Corrección: Agregar antes del loop de personalizaciones una verificación de campaña activa y renderizar el banner de campaña con prioridad sobre cualquier otro contenido.

ÁREA 5 — Flujo de personalización (diferenciador del negocio)

M17 · No hay formulario de personalización visible en el código del paquete Shop

Contexto de negocio: El diferenciador clave del negocio es la personalización con foto. Sin embargo, en toda la revisión del código del paquete `Shop` no se encontró ningún componente, vista o lógica de carga de fotografía del cliente.

Problema: Si la personalización se resuelve únicamente por WhatsApp tras la compra, existe un gap de UX significativo: el cliente no tiene certeza de cómo o cuándo enviar su foto, lo que genera fricción post-compra y potenciales abandonos por incertidumbre.

Corrección propuesta: - Agregar en la ficha de producto un campo de tipo “custom attribute” o una nota de pedido visible donde el cliente pueda ingresar la URL de su foto, subir un archivo, o confirmar que la enviará por WhatsApp. - Como mínimo: agregar texto

informativo en la ficha de producto que explique el proceso de personalización antes del botón de compra.

M18 · El botón de WhatsApp no incluye el producto en el mensaje automático

Archivo: packages/Webkul/Shop/src/Resources/views/components/layouts/footer/index.blade.php líneas 124-140

Problema: El script construye el mensaje de WhatsApp con el título de la página y la URL actual. Esto funciona en la ficha de producto, pero en el carrito o checkout el mensaje no incluye los productos del pedido.

Corrección: En la ficha de producto, enriquecer el href del botón con el nombre del producto y precio directamente (posiblemente mediante un data-* attribute inyectado desde la vista del producto).

ÁREA 6 — Rendimiento y SEO

M19 · Pixel de Facebook con trackCustom en cada carga de página

Archivo: packages/Webkul/Shop/src/Resources/views/components/layouts/footer/index.blade.php líneas 124-126

Problema: Se llama fbq("trackCustom", "pageView", ...) en cada renderizado del footer. El pixel de Facebook ya tiene un evento PageView estándar que se dispara automáticamente. Este trackCustom duplica el tracking y puede inflar las métricas de eventos en Meta Ads, generando audiencias sucias.

Corrección: Verificar si el pixel base ya dispara PageView automáticamente. Si sí, eliminar este trackCustom o convertirlo en un evento semántico real (ej. ViewContent con el producto cuando aplique).

Matriz de prioridad

ID	Mejora	Impacto	Esfuerzo	Prioridad
M01	Memory leak carousel	Alto (estabilidad)	Bajo	URGENTE
M05	Botón pedido sin color de marca	Alto (conversión)	Bajo	URGENTE
M03	Teléfono WhatsApp hardcodeado	Medio (mantenibilidad)	Bajo	Alta
M04	Script WhatsApp puede fallar	Medio (estabilidad)	Bajo	Alta
M10	Sin CTA sticky mobile	Alto (conversión móvil)	Medio	Alta

ID	Mejora	Impacto	Esfuerzo	Prioridad
M17	Sin flujo de personalización	Alto (diferenciador negocio)	Alto	Alta
M11	Cupón visible por defecto	Medio (conversión)	Bajo	Media
M14	Sin sellos de confianza checkout	Medio (conversión)	Bajo	Media
M13	CTA no emocional en card	Medio (conversión)	Bajo	Media
M15	Sin landing de campañas	Alto (ventas estacionales)	Alto	Media
M16	Sin sección campaña sobre fold	Alto (ventas estacionales)	Medio	Media
M02	Método muerto carousel	Bajo (deuda técnica)	Bajo	Baja
M06	Botón login mobile sin naranja	Bajo (marca)	Bajo	Baja
M07	Panel mobile color rosa	Bajo (marca)	Bajo	Baja
M08	CSS muerto categoryTitle	Bajo (deuda técnica)	Bajo	Baja
M09	Reglas CSS vacías	Bajo (deuda técnica)	Bajo	Baja
M12	Sin badges personalizados	Medio (conversión)	Medio	Baja
M18	WhatsApp sin datos de producto	Bajo (UX)	Bajo	Baja
M19	Pixel FB duplicado	Medio (analytics)	Bajo	Baja

Próximos pasos sugeridos

1. **Sprint inmediato (sin riesgo de deploy):** M01, M02, M04, M07, M08, M09 — son correcciones de bugs y CSS que no cambian el flujo de negocio.
2. **Sprint de conversión:** M05, M06, M11, M13, M14 — mejoras visuales y de copy de bajo esfuerzo con impacto directo en tasa de conversión.
3. **Sprint de estrategia:** M10, M15, M16, M17 — funcionalidades nuevas que requieren mayor planificación. Deben estar listas **antes** de la próxima campaña estacional.
4. **Backlog:** M03, M12, M18, M19 — mejoras de mantenibilidad y analítica que no bloquean ventas.