

Documento de Integración del Meta Pixel "Lead"

1. Objetivo

Implementar el evento estándar **Lead** de Meta (Facebook) en el formulario de consulta de paquetes, de modo que:

- Meta Ads detecte cada envío exitoso como conversión.
 - Puedas hacer remarketing a quienes consultan sin reservar.
 - Midas el rendimiento de tus campañas de generación de consultas.
-

2. Resumen de Cambios Realizados

Paso	Archivo	Cambio realizado
1	<code>server.php</code>	Se añadió la constante <code>FACEBOOK_PIXEL_ID</code> para centralizar el Pixel ID y permitir su reutilización en todo el sistema.
2	<code>mkte_agencia/clases/Web.php</code>	Se creó el método <code>getFacebookPixel()</code> que, cuando detecta la constante definida, devuelve automáticamente el bloque completo del Meta Pixel.
3	<code>mkte_agencia/views/themes/agencia.tpl</code>	En el bloque <code>MODEL</code> se insertó la llamada a <code>Web::getFacebookPixel()</code> y su impresión condicional, garantizando la inyección del Pixel en todas las vistas.
4	<code>mkte_agencia/views/templates/gracias.tpl</code>	Se ubicó el disparo del evento Lead dentro del bloque <code>TEMPLATE_JS</code> , de modo que solo se envíe a Meta cuando el usuario llega a la página de "Gracias" tras un envío exitoso del formulario.

3. ¿Cómo funciona la integración?

La integración del evento "Lead" del Meta Pixel se compone de cuatro fases clave que aseguran su carga automática en todo el sitio y el correcto registro de conversiones al enviarse el formulario de consulta.

1. Definición del Pixel ID

- En `server.php` se declara una constante global:

```
/** Meta Pixel */
define('FACEBOOK_PIXEL_ID', 'TU_PIXEL_ID');
```

- Esta constante estará disponible en todas las capas de PHP y Smarty, evitando duplicación o hard-coding.

2. Inyección automática del Pixel en el <head>

- La clase `Web` implementa un método estático `getFacebookPixel()` que:
 1. Comprueba si `FACEBOOK_PIXEL_ID` está definido y no está vacío.
 2. Devuelve un bloque HTML que incluye el script asíncrono de Meta Pixel y la etiqueta `<noscript>` para usuarios sin JS.
- En `mkte_agencia/views/themes/agencia.tpl`, dentro del bloque `MODEL`, se asigna e imprime de esta forma:

```
{* Facebook Pixel *}
{assign value=Web::getFacebookPixel() var=facebookPixelHTML}
{if $facebookPixelHTML}
  <div class="hidden">{$facebookPixelHTML}</div>
{/if}
```

- Como resultado, **cada** página carga automáticamente:

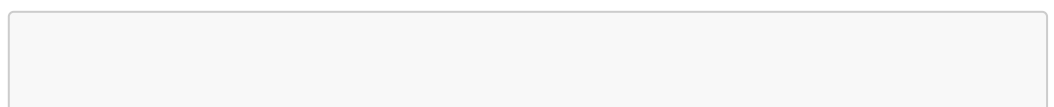
```
<!-- Meta Pixel Code -->
<script>
  !function(f,b,e,v,n,t,s){...}(window,
document,'script','https://connect.facebook.net/en_US/fbevents.js');
  fbq('init','TU_PIXEL_ID');
  fbq('track','PageView');
</script>
<noscript>
  
</noscript>
<!-- End Meta Pixel Code -->
```

3. Seguimiento de PageView y persistencia del Pixel

- En cuanto el DOM esté listo, el Pixel inicializado ya habrá enviado un evento `PageView` a Meta.
- Gracias a la inyección centralizada, no es necesario agregar manualmente este script en cada plantilla: basta con definir la constante en `server.php`.

4. Disparo del evento "Lead" en el formulario de consulta

- Cuando un usuario envía el formulario, el flujo puede ser:
 - **Redirección a "Gracias":**
 1. El controller PHP procesa datos y redirige a `gracias.tpl`.
 2. En esa plantilla, dentro de `TEMPLATE_JS`, se ejecuta:



```
document.addEventListener('DOMContentLoaded', function(){
  if (typeof fbq === 'function') {
    fbq('track','Lead');
  }
});
```

3. El navegador, al cargar “Gracias”, lanza el evento **Lead** que Meta registra como conversión.

- **AJAX + SweetAlert2** (opcional):

1. El formulario se envía por AJAX desde `consultar.tpl`.
2. En el `.done()` del request:

```
fbq('track','Lead');
Swal.fire('¡Gracias por consultar!', '', 'success')
  .then(() => window.location.href =
    'agencia/gracias/reservar');
```

3. Se dispara inmediatamente el evento y luego muestra la alerta.

5. Resumen del flujo de datos

```
[Usuario carga página]
  ↓
Web::getFacebookPixel() → inyección de Pixel + PageView
  ↓
[Usuario completa formulario]
  ↓
Envío (AJAX o POST) → procesamiento PHP → redirección
  ↓
“Gracias.tpl” carga → fbq('track','Lead')
  ↓
Meta Ads recibe evento “Lead” → registro de conversión
```

4. Pasos para replicar con otro cliente

1. Obtener el Pixel ID

- En [Meta Events Manager](#), crear o seleccionar el Pixel y copiar su ID.

2. Actualizar `server.php` - Cambiar la línea:

```
define('FACEBOOK_PIXEL_ID', 'NUEVO_PIXEL_ID');
```

3. Limpiar cache

- Borrar el cache en el admin y Ctrl + F5.

4. Probar el flujo

- Envía el formulario, observa en la consola o con Meta Pixel Helper que se ejecuten:
`fbq('track', 'PageView')` al cargar y
`fbq('track', 'Lead')` en la página de gracias.
-

5. Recomendaciones y Buenas Prácticas

- **Herramientas de depuración:**
Instala **Meta Pixel Helper** para verificar en tiempo real los eventos que Meta recibe.
- **Monitoreo continuo:**
Revisa en Ads Manager el rendimiento de las conversiones y ajusta tus campañas.