

PD COOKIES CONSENT - INFORMACIÓN TÉCNICA

FUNCIONALIDAD QUE INSERTA UN PANEL DE CONSENTIMIENTO DE COOKIES Y BLOQUEA ELEMENTOS DEL DOM QUE PUDIERAN INSERTARLAS

"PD COOKIE CONSENT" es una librería JavaScript desarrollada bajo *licencia MIT* y a través de la cual se pretende facilitar el bloqueo de aquellos elementos del DOM que pudieran incrustar *cookies*.

"PD COOKIE CONSENT" sirve para que el técnico del sitio web pueda insertar una ventana emergente (banner) que avise al visitante sobre la existencia de cookies, además de facilitar un medio para la gestión de aquellos elementos que pudieran generarlas, bloqueándolos y evitando que las generen hasta el posterior desbloqueo por parte del visitante.

1. INSERTAR LA LIBRERIA EN NUESTRO SITIO WEB

IMPORTANTE:

El primer paso que debemos realizar es insertar la librería y sus estilos entre las etiquetas "**head**"..."/**head**", quedando el script de la librería por encima de cualquier otro que pudiera haber entre las etiquetas mencionadas.

Si no se cumple este primer paso tal como se describe, "PDCookieConsent" no funcionará correctamente.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>PDCookieConsent: Documentación</title>

  <!-- Otros estilos... -->
  <!-- Estilos <rel="stylesheet" href=""> de PDCookieConsent -->
  <!-- Otros estilos... -->

  <!-- Aquí insertaremos el <script src=""> de PDCookieConsent -->
  <!-- Otros scripts... -->
</head>
<body>

  <!-- Otros scripts... -->
</body>
</html>
```

NOTA:

Si nuestro sitio web ha sido desarrollado a través de un CMS como WordPress, Ghost, Shopify, Magento, Prestashop, Joomla, Drupal u otro similar, es importante asegurarse de que también se cumpla el primera paso. En caso de utilizar un CSM **la mejor opción, la cual nos permitirá cumplir con el primer paso de forma segura** es la de modificar el tema o plantillas del sitio web, por ejemplo y en el caso de WordPress, creando un "child theme".

Consulte la documentación destinada a desarrolladores que cada una de las plataformas ofrece

El contenido del Estilo (stylesheet) y del Código (script src) se facilitará aparte mediante el documento **"BANNER + Panel de configuración (PD Cookie Consent)"**.

2. CONFIGURACIÓN

La librería ofrece una clase (*PDCookieConsent*) y método (*config*) que permitirá configurar algunas partes de esta.

```
PDCookieConsent.config({
  "defaultLang" : "es",
  "brand"       : {
    "dev"       : true,
    "name"      : "DEMO SL",
    "url"       : "https://demo.com",
    "websiteOwner" : "Nombre Sitio Web"
  },
  "cookiePolicyLink": "https://sitioweb.com/politica-cookies",
  "hideModalIn": [
    "https://sitioweb.com/politica-cookies",
    "https://sitioweb.com/aviso-legal",
    "https://sitioweb.com/politica-privacidad"
  ],
  "showBadges": true,
  "styles": {
    "primaryButton": {
      "bgColor" : "#fab36b",
      "txtColor" : "#000000"
    },
    "secondaryButton": {
      "bgColor" : "#bfbfbf",
      "txtColor" : "#000000"
    },
    "cookieButton": {
      "activatedColor" : "#f88410",
      "disabledColor" : "#fbcc9d"
    }
  }
});
```

"config": Se trata de un objeto que contendrá distintas propiedades destinadas a personalizar los siguientes aspectos de la librería *"PDCookieConsent"*.

"defaultlang" String	Determina el idioma por defecto del panel para cuando no exista definido en la etiqueta "..." o bien no se corresponda a uno de los 5 posibles : "es" Español, (se establece como valor por defecto si no se asigna a través de la configuración), "ca" (catalán), "en" (Inglés), "eu" ((eusquera), y "gl" (gallego). La librería detecta los idiomas en código ISO639-1 ("es") y en caso de que estos estén agrupados por familias ("es-ES") el idioma se establecerá por los dos primeros caracteres (para el caso de "es-ES" será "es").
--------------------------------	---

"brand"	Alterar cualquiera de los predefinidos de este campo que se entregan en el documento BANNER + Panel de configuración (PD Cookies consent), provocará un mal funcionamiento de la utilidad	
	Pie de página	Es obligatorio el uso de la propiedad "dev" para que se visualicen los datos que deben aparecer. Si no se utiliza dicha propiedad no se visualizará el pie de página y si cualesquiera de los valores de las propiedades son incorrectos, tampoco se visualizará este. "dev" . (Boolean). Su valor es "true" . "name" (String). Su valor es "Consultancy X3 S.L." "url" . (String). Su valor es "https://grupox3.es"
	Nombre del sitio web en la lista de la vista "Configurar"	"websiteOwner" . (String). Define el nombre que reemplazará el dominio de nuestro sitio web en la lista del panel "PDCookieConsent" , vista "Configurar" . Si no se define se mostrará el dominio. Se trata del nombre del sitio web.
"cookiePolicyLink" (String)	URL que establece el enlace hacia la Política de Cookies. Este aparece en el panel, dentro del texto que informa al visitante del sitio web sobre las cookies, antes de los botones "Aceptar" y "Configurar" .	
"hideModalIn" (Array de String)	Nos permite definir aquellas URLs en las que no debe aparecer el panel de forma automática.	
"showBadges" (Boolean)	Si su valor es "true" , aparecerá un globo con un número encima de cualquier elemento que contenga la clase "pdcc-open-modal" . Es útil para informar al visitante del sitio web sobre nuevos elementos del DOM bloqueados. Los "badges" no se visualizarán si el visitante ha aceptado todas las cookies del sitio ("Aceptar").	
"styles"	Permite definir los colores de los botones del panel y del mensaje. Todos los colores deben ser <i>Hexadecimales</i> .	
	"primaryButton"	Permite definir los colores de los botones "Aceptar" y "Aceptar todas" (panel y mensaje). "bgColor" : Color de fondo. "textColor" : Color del texto.
	"secondaryButton"	Permite definir los colores de los botones "Configurar" , "Rechazar todas" y "Guardar configuración" (panel y mensaje).
	"cookieButton"	Permite definir los colores de los conmutadores que son utilizados para activar o desactivar los elementos del DOM. Estos aparecen en la vista "Configurar". "activeColor" : Color para cuando esté activo. (Cuando esté inactivo siempre será gris). "disabledColor" : Color del conmutador cuando esté deshabilitado y activo. (útil para que el visitante no pueda manipular su estado y pueda ver fácilmente que además está activo).

IMPORTANTE:

Todos los datos introducidos son analizados. Se puede consultar la consola del navegador para visualizar alertas sobre los posibles errores existentes en los valores dados a las propiedades.

Las propiedades no admiten comillas simples (') para sus nombres o valores, deben usarse comillas dobles ("). Tampoco debe insertarse una coma (",") en el último elemtno de las listas de propiedades.

NOTA:

El idioma por defecto del panel **"PDCookieConsent"** es el español. Es necesario hacer uso de la propiedad **"defaultLang"** para poder establecer un idioma por defecto.

3. BLOQUEAR ELEMENTOS DEL DOM PARA EVITAR QUE GENEREN COOKIES

Mediante "PDCookieConsent" se bloquean todos los elementos "iframe", "script" y "noscript" que contengan una cadena de texto (una palabra, un fragmento de una URL o de un texto) no permitida. La razón de no bloquear todos los elementos por defecto es la de evitar que se rompan funcionalidades del sitio web que ya de por sí no generen cookies.

Las cadenas de texto que tiene definidas la librería son las siguientes:

"adwords", "analytics", "doubleclick", "facebook.", "google.com/maps", "google.com/recaptcha", "googleadservices", "googlesyndication", "googletagmanager", "googletagservices", "googletraveladservices", "googleusercontent", "gstatic", "linkedin.", "maps.google.com", "maps.googleapis", "twitter.", "vimeo.", "youtube.", "ytimg", "urchin".
Obsérvese que algunas de las cadenas de texto son palabras con un punto detrás (p.ej. "facebook.").

Además de las cadenas de texto no permitidas, se han definido unos nombres que serán dados a todos los elementos del DOM bloqueados que las contengan, agrupándolos y permitiendo su activación/desactivación por grupo:

- Google: "google", "google.com/maps", "googleusercontent", "googleusercontent", "maps.google.com" y "maps.googleapis".
- Google Ads: "adwords", "doubleclick", "googleadservices", "googlesyndication", "googletagservices" y "googletraveladservices".
- Google Analytics: "googletagmanager", "urchin" y "analytics".
- Facebook: "facebook".
- LinkedIn: "linkedin".
- Twitter: "twitter".
- Vimeo: "vimeo".
- You Tube: "youtube" e "ytimg".

También podemos bloquear los elementos DOM que no hayan sido tenidos en cuenta por la librería:

Opción 1: Bloquear "script src" e "iframe" analizando su URL (propiedad "domain"):

```
PDCookieConsent.blockList([
{
  "domain" : "google.com/recaptcha",
  "name" : "Google reCaptcha",
  "report" : true
}
]);
```

Opción 2: Bloquear "script...texto.../script" y "noscript.../noscript" analizando el TEXTO que contiene (propiedad "contain"):

```
PDCookieConsent.blockList([
{
  "contain" : "recaptcha",
  "name" : "Google reCaptcha",
  "report" : true
}
]);
```

blockList: Contendrá un Array de objetos "{...}" que representarán la cadena de texto no permitida, además de ofrecer la posibilidad de definir algunas características.

"domain" (opción 1) (String)	URL que establece el enlace hacia la Política de Cookies. Este aparece en el panel, dentro del texto que informa al visitante del sitio web sobre las cookies, antes de los botones "Aceptar" y "Configurar". Determina que la cadena de texto que toma como valor se va a usar para analizar la URL de los "script src" e "iframe src". No se puede usar al mismo tiempo que "contain".
"contain" (opción 2) (String)	Determina que la cadena de texto que toma como valor se va a usar para analizar los textos de los "iframe...texto.../iframe". No se puede usar al mismo tiempo que "domain".
"name" (String)	Define el nombre al que pertenecerá todo elemento del DOM que contenga la cadena de texto no permitida. Puede usarse el mismo para un solo elemento o para más de uno (agrupar). Si más de un elemento del DOM tiene definido el mismo "name", estos se agruparán y, en consecuencia se bloquearán y desbloquearán en grupo.
"report" (Boolean)	Hacer que la librería tome en cuenta el elemento para poder gestionarlo a través del mensaje que informa sobre los elementos bloqueados de la página visitada (consultar el apartado 6. Funcionamiento de PDCookieConsent de este mismo documento).

IMPORTANTE:

Todos los datos introducidos son analizados. Se puede consultar la consola del navegador para visualizar alertas sobre los posibles errores existentes en los valores dados a las propiedades.

Las propiedades no admiten comillas simples (') para sus nombres o valores, deben usarse comillas dobles ("). Tampoco debe insertarse una coma (",") en el último elemtno de las listas de propiedades.

Además de la posibilidad de bloquear aquellos elementos del DOM que no hayan sido tenidos en cuenta por la librería (opciones 1 y 2 anteriores), también existe la posibilidad de **forzar el bloqueo de aquellos elementos "script" que, a pesar de ser detectados por la librería, siguen insertando sus cookies**. Ante esta situación la única opción posible es hacer un **cambio manual** en estos elementos del DOM:

1. Buscaremos todas aquellas "script" que o bien fueron detectadas por la librería o bien fueron tenidas en cuenta a través de nuestro método "PDCookieConsent.blockList()", pero que aún siguen insertando sus cookies.
2. Insertaremos a cada uno de los "script" el tipo MIME "text/plain". para ello tan solo deberemos modificar el valor del atributo "type" ya existente o, en el caso de que este atributo no existiera, lo insertaremos junto al valor mencionado.

```
<script type="text/plain"></script>
|
<script type="text/plain" src=""></script>
```

Una vez realizados los dos pasos anteriores, los "script" seguirán siendo detectados por la librería porque en su contenido o URL seguirán existiendo cadenas de texto no permitidas, pero ya de por sí, al tener definido el tipo MIME "text/plain", estos no serán tratados como un código ejecutable sino como texto plano (no ejecutable).

IMPORTANTE:

Es necesario que los "..." que se bloqueen de forma manual sean detectados por la librería para que la activación de estos, a través del panel "PDCookieConsent", sea efectiva.

Si no fuesen detectados por la librería quedarían siempre como un código de tipo MIME "text/plain" (no ejecutable) ya que la librería modifica el código a "application/javascript" cuando el visitante activa el elemento en cuestión.

Para entender un poco mejor el manejo de los elementos del DOM bloqueados, a continuación, se explica cómo actúa la librería ante los elementos que detecta y bloquea:

- Cuando se accede al sitio web se carga el contenido de este y ante tal carga se observa cada uno de los elementos del DOM que se van generando.
- Cuando el elemento del DOM sea un "*script*" o un "*noscript*" o un "*iframe*", la librería procede a su análisis con la intención de detectar si existe una cadena de texto no permitida en su URL o en su contenido.
- Si el elemento del DOM contuviera una cadena de texto no permitida se procede a bloquearlo:
 - "*iframe*": su etiqueta HTML se comenta e inserta en el inicio de este un código que permite informar al desarrollador que ha sido bloqueado por la librería.
 - "*script*" y "*noscript*": se procede del mismo modo que en el caso del "*iframe*" pero, además, se inserta, o modifica, el atributo "*type*" poniéndole como valor el tipo MIME "*text/plain*".

A continuación, se puede observar una imagen que muestra el "*script*" de Google Analytics comentado y con el código que informa que ha sido bloqueado por la librería:

```

▶ <script type="text/javascript">...</script>
  <!-- [pd:Google Analytics] <script type="text/plain">
    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
      (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
      m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
    })(window,document,'script','https://www.google-
    analytics.com/analytics.js','ga');

    ga('create', 'UA-68824762-1', 'auto');
    ga('send', 'pageview');

  </script>-->

```

(pd:Nombre): Código que informa, al desarrollador, que el elemento del DOM ha sido bloqueado por la librería "*PDCookieConsent*". Siempre aparecerá dentro de un comentario HTML.

"pd": Identificador de la librería.

"Nombre": Es el nombre al que pertenece el elemento del DOM bloqueado. El "*Nombre*" es el que aparece en la lista de la vista "*Configurar*" del panel "*PDCookieConsent*", también es el nombre que definimos a través de la propiedad "*name*" de cada uno de los objetos insertados en "*PDCookieConsent.blockList()*".

NOTA:

Para poder visualizar los elementos bloqueados por la librería es necesario abrir nuestras herramientas para desarrolladores que nos ofrece el navegador ("*Inspeccionar Código*").

Si no fuesen detectados por la librería quedarían siempre como un código de tipo MIME "*text/plain*" (no ejecutable) ya que la librería modifica el código a "*application/javascript*" cuando el visitante activa el elemento en cuestión.

4. DESBLOQUEAR ELEMENTOS QUE HAN SIDO BLOQUEADOS POR LA LIBRERIA

Tal y como se menciona en el apartado "**3. Bloquear elementos del DOM para que no generen cookies**", la librería bloquea de forma automática aquellos elementos que cumplen unos requisitos y, a veces, ello acarrea el problema de que se bloquee algún elemento que no era necesario bloquear.

Para poder desbloquear el elemento tan solo deberemos tener en cuenta si se ha bloqueado este por contener una

cadena de texto no permitida en su URL o bien entre el texto que pudiera contener. Una vez sepamos el porqué de su bloqueo tan solo deberemos seleccionar un fragmento de su URL o texto que permita a la librería reconocer el elemento y desbloquearlo.

Opción 1: La librería ha bloqueado un elemento por su URL (propiedad "domain").

Elemento bloqueado por la librería:

```
<iframe src="https://www.youtube.com/embed/kaVg70p2sS4" ></iframe>
```

Código a utilizar para desbloquear el elemento bloqueado:

```
PDCookieConsent.blockList([  
  {  
    "domain" : "youtube.com/embed/kaVg70p2sS4",  
    "name" : "YouTube",  
    "actived" : true,  
    "editable": false,  
    "visible" : false  
  }  
]);
```

Opción 2: La librería ha bloqueado un elemento por contener una cadena de texto no permitida entre el texto que contiene (propiedad "contain"). Esta situación nada más afecta a los "script" y "noscript".

Elemento "script" bloqueado por la librería:

```
<script>  
  // En mi contenido existe una cadena de texto no permitida: gstatic.  
  document.cookie = "una_cookie=privacydriver; samesite=strict; secure;";  
</script>
```

Código a utilizar para desbloquear el elemento bloqueado:

```
PDCookieConsent.blockList([  
  {  
    "contain" : ": gstatic",  
    "name" : "Nombre Grupo",  
    "actived" : true,  
    "editable" : false,  
    "visible" : false  
  }  
]);
```

blocklist(): Contendrá un Array de objetos "[...]" que representarán la cadena de texto no permitida, además de ofrecer

la posibilidad de definir algunas características.

"domain" (opción 1) (String)	Determina que la cadena de texto que toma como valor se va a usar para analizar la URL de los <i>"script src"</i> e <i>"iframe src"</i> . No se puede usar al mismo tiempo que <i>"contain"</i> .
"contain" (opción 2) (String)	Determina que la cadena de texto que toma como valor se va a usar para analizar el contenido de los <i>"script...contenido.../script"</i> . No se puede usar al mismo tiempo que <i>"domain"</i> .
"name" (String)	Define el nombre de grupo al que pertenecerá el elemento DOM que contenga la cadena de texto definida en <i>"domain"</i> o <i>"contain"</i> . Puede usarse el mismo para un solo elemento o para más de uno.
"editable" (Boolean)	Deshabilita el botón que permite la activación/desactivación del elemento del DOM. Si el valor es <i>"false"</i> , el visitante del sitio web no podrá activar el <i>"iframe"</i> o <i>"script"</i> .
"visible" (Boolean)	Determina si el elemento va a aparecer en la lista del panel o no. Si se define con el valor <i>"false"</i> el visitante no podrá ni activar ni desactivar el elemento ni visualizarlo en la lista.

Para entender un poco mejor el manejo de los elementos del DOM desbloqueados, a continuación, se explica cómo actúa la librería ante los elementos que detecta y desbloquea:

- Cuando se accede al sitio web se carga el contenido de este y ante tal carga se observa cada uno de los elementos del DOM que se van generando.
- Cuando el elemento del DOM sea un *"script"* o un *"noscript"* o un *"iframe"*, la librería procede a su análisis con la intención de detectar si existe una cadena de texto no permitida en su URL o en su contenido.
- Si el elemento del DOM contuviera una cadena de texto no permitida se procede a analizar si la cadena de texto encontrada ha sido ya permitida por el visitante a través de la vista "Configurar" del panel y, en caso afirmativo, se ignora el bloqueo del elemento:
 - *"iframe"*: su etiqueta HTML no se comenta y se permite su carga con normalidad.
 - *"script"* y *"noscript"*: se procede del mismo modo que en el caso del *"iframe"* pero, además, se inserta, o modifica, el atributo *"type"* poniéndole como valor el tipo MIME *"application/javascript"*.

NOTA:

El tipo MIME *"application/javascript"* actúa igual que el tipo *"text/javascript"* pero con la diferencia de que este último está obsoleto y se recomienda el uso del primero mencionado.

5. FUNCIONAMIENTO DE "PD COOKIE CONSENT"

Cuando un visitante accede a nuestro sitio wweb, la librería *"PDCookieConsent"* observa cada uno de los elementos existentes en la página visitada y si alguno de estos cumple los requisitos necesarios para ser bloqueado, se bloquea.

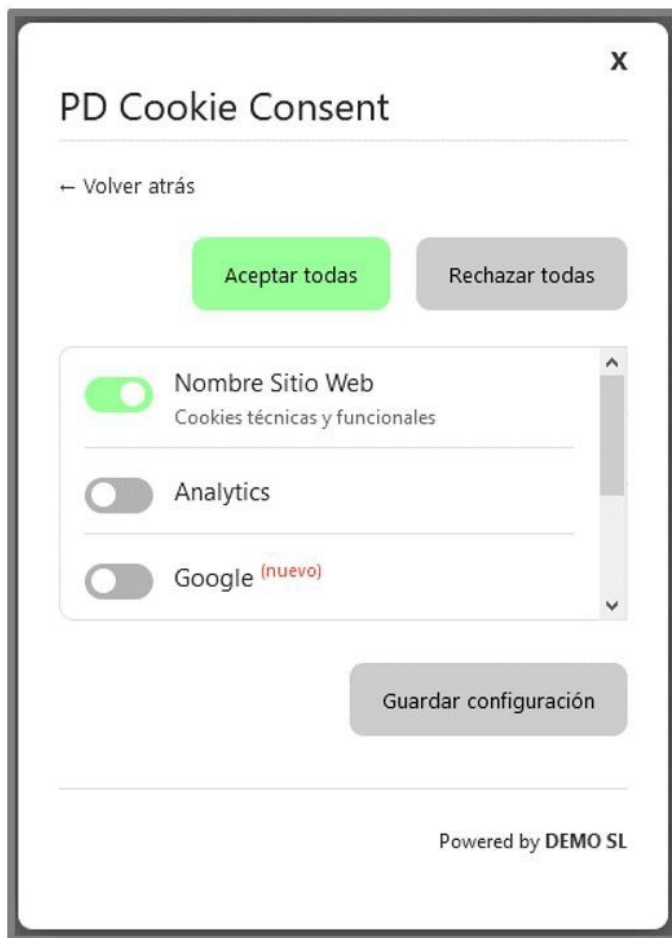
Una vez observados todos los elementos de la página, la librería mostrará el panel *"PDCookieConsent"* y generará una cookie de sesión nombrada *"pd_cc"*, la cual es necesaria para el correcto funcionamiento de esta.



En la primera vista del panel, el visitante del sitio web podrá:

1. Leer el aviso que informa sobre las cookies.
2. hacer clic en el enlace de la Política de Cookies (si previamente se ha definido en la propiedad "***cookiePolicyLink***" de la configuración "***PDCookieConsent.config()***").
3. Aceptar todas las cookies del sitio web haciendo clic en "***Aceptar***".
4. Ir a la vista de Configuración haciendo clic en "***Configurar***".
5. Cerrar el panel.

Si el visitante del sitio web pulsa el botón "Configurar" obtendremos una nueva vista:



En la vista de configuración el visitante podrá:

1. Volver a la vista anterior pulsando el enlace "Volver atrás".
2. Visualizar la lista con todos los elementos no permitidos, pudiendo:
 1. Aceptarlos de forma independiente.
 2. Rechazarlos de forma independiente.
 3. Visualizar aquellos que son nuevos y no se encontraban en la lista en una antigua visualización. (Muestra el texto ("nuevo") a su lado.
3. Aceptar todos los dominios a la vez pulsando el botón "Aceptar todas".
4. Rechazar todos los dominios a la vez pulsando el botón "Rechazar todas".
5. Guardar los cambios realizados mediante el botón "Guardar configuración".

(Es necesario guardar los cambios para que los puntos 2, 3 y 4 sean efectivos)

Una vez cerrado el panel existe la posibilidad de volverlo a abrir de forma manual.

La librería ofrece una propiedad CSS para dicho propósito y cualquier elemento que la contenga actuará como un botón que abrirá el panel. Tan solo debemos escoger un elemento del DOM cualquiera como por ejemplo un enlace e insertarle la propiedad (class="pdcc-open-modal")

```
<a href="#" class="pdcc-open-modal">Panel Cookies</a>
```

Una vez el visitante pulse en el enlace, o en otro elemento que contenga la propiedad ya mencionada, se abrirá la vista de Configuración del panel "*PDCookieConsent*".

Además del panel "*PDCookieConsent*" la librería también proporciona un mensaje que aparecerá en la parte superior del sitio web para informar al visitante de los elementos que hayan sido bloqueados y afecten sobre el aspecto visual del sitio o funcionalidades necesarias para la interacción con este.

Se ha ocultado el contenido porque tiene las cookies bloqueadas. Permitir cookies de: YouTube. [Aceptar todas](#) [Configurar](#)

Ante este mensaje el visitante podrá:

1. Informarse sobre el contenido bloqueado y que no se está visualizando, incluyendo la lista de los elementos bloqueados (en el caso de la imagen proporcionada, *You Tube*).
2. En relación con la lista de elementos bloqueados podrá realizar las siguientes acciones:
 1. Aceptar todos los elementos bloqueados de la página visitada haciendo clic en "*Aceptar todas*", quedando estos activos en el resto del sitio web (si existiesen). Una vez pulsado el botón se ocultará el mensaje.
 2. Abrir la vista de configuración pulsando el botón "*Configurar*", pudiendo realizar cualesquiera de las acciones ya mencionadas en este mismo apartado del documento. El mensaje se ocultará cuando se muestre el panel "*PDCookieConsent*" y, también, una vez pulsado el botón "*Guardar configuración*", siempre y cuando se hayan aceptado todos los elementos listados en el mismo mensaje.

IMPORTANTE:

El "mensaje" principalmente enuncia los "*bloqueados*". Para que sean tenidos en cuenta los "..." es necesario definir la propiedad "*report*" con el valor "*true*" en "*PDCookieConsent.blockList()*" (Obsérvese el apartado 3. Bloquear elementos del DOM para evitar que generen cookies").