



# **TECNOLÓGICO NACIONAL DE MÉXICO**

**Instituto Tecnológico de Nogales**

**Ingeniería en Sistemas Computacionales**

Materia:

***Tec. De Prog. Web avanzada***

Docente:

***Ing. Fabián Vásquez Sesteaga***

Estudiante:

**Freyde Moreno Miguel Ángel**

**Investigación 4.1.**

**H. Nogales, Sonora**

## Ofuscación de código

La ofuscación se refiere a encubrir el significado de una comunicación haciéndola más confusa y complicada de interpretar.

En computación, la ofuscación se refiere al acto deliberado de realizar un cambio no destructivo, ya sea en el código fuente de un programa informático, en el código intermedio (bytecodes) o en el código máquina cuando el programa está en forma compilada o binaria. Es decir, se cambia el código se "enrevesa" manteniendo el funcionamiento original, para dificultar su entendimiento. De esta forma se dificulta los intentos de ingeniería inversa y desensamblado que tienen la intención de obtener una forma de código fuente cercana a la forma original.

A las herramientas que realizan procesos de ofuscación se les llama ofuscadores. Los ofuscadores pueden actuar sobre el código fuente, sobre el código objeto o sobre ambos.

Como un efecto lateral, la ofuscación, en ocasiones, hace que los programas resultantes sean más pequeños (aunque puede hacer que los programas sean más grandes en otros casos).

Algunos lenguajes tienden más a la ofuscación que otros. C, C++ y Perl son los más citados como fácilmente ofuscables. Las macros de preprocesador son usadas a menudo para crear código complicado de leer enmascarando la gramática y sintaxis estándar del lenguaje del cuerpo principal de código.

Es posible ofuscar el código sin la necesidad de ofuscadores con las siguientes técnicas:

Eliminar comentarios, espacios en blanco y saltos de línea.

Modificar los nombres de las variables y funciones para que no proporcionen información sobre su objetivo.

Complicar artificialmente el control de flujo del programa (usar gotos en lugar de bucles y condicionales).

Alterar la distribución del código dentro del proyecto (es decir, partir funciones, mover código de un fichero a otro o copiarlo)

Añadir código innecesario (condiciones o parámetros que no se utilizan o cumplan alguna función).

Cifrar partes del código para que deban ser descifrados al momento de la ejecución.

Existen ofuscadores en línea como por ejemplo: <https://javascriptobfuscator.com/>

Te permite subir un archivo js, y automáticamente te permite ver el código ofuscado y de igual forma descargarlo.

Código js.

```
s.ajax({
  type: "POST",
  url: strUrl,
  data: "{" + strParametros + "}",
  contentType: "application/json; charset=utf-8",
  dataType: "json",
  async: bolAsync,
  success: function (objResult) {
    if (fnSuccess !== undefined) {
      fnSuccess(objResult.d);
    }
    return false;
  },
  error: function (XMLHttpRequest, textStatus, errorThrown) {
    if (errorThrown !== "") {
      alert("Error en Post: " + errorThrown);
    }
  }
});
return false;
}
```



Codigo js ofuscado.

```
var _0xf8f3=
["\x50\x4f\x53\x54","\x7b","\x7d","\x61\x70\x70\x6c\x69\x63\x61\x74\x69\x6f\x6e\x2f\x6a\x73\x6f\x6e\x3b\x20\x63\x68\x61\x72\x73\x65\x74\x3d\x75\x74\x66\x2d\x38","\x6a\x73\x6f\x6e","\x64":"","\x45\x72\x72\x6f\x72\x20\x65\x6e\x20\x50\x6f\x73\x74\x3a\x20","\x61\x6a\x61\x78"];function GetAjax(_0xab74x2,_0xab74x3,_0xab74x4,_0xab74x5){$_0xf8f3[8]]
({type:_0xf8f3[0],url:_0xab74x2,data:_0xf8f3[1])+ _0xab74x3+
_0xf8f3[2],contentType:_0xf8f3[3],dataType:_0xf8f3[4],async:_0xab74x4,succes:function(_0xab74x6){if(_0xab74x5!= undefined)
{_0xab74x5(_0xab74x6[_0xf8f3[5]]);return false},errorfunction(_0xab74x7,_0xab74x8,_0xab74x9){if(_0xab74x9!= _0xf8f3[6])
{alert(_0xf8f3[7]+ _0xab74x9)}}});return false}
```



Para ofuscar código HTML lo podemos minimizar.

Minimizar código HTML:

Lo que debemos hacer, es agregar el siguiente código en el inicio del documento, en la primera línea, php de la página que queremos ofuscar. Debe ser archivo php.

```
<?php ob_start("comprimir_pagina"); ?>
```

Después el siguiente código al finalizar el documento.

```
<?php
// Una vez que el búfer almacena nuestro contenido utilizamos "ob_end_flush"
para usarlo y deshabilitar el búfer
ob_end_flush();
// Función para eliminar todos los espacios en blanco
function comprimir_pagina($buffer) {

    $search = array(
        '>[^\S ]+/s', // elimina espacios en blanco después de las etiquetas,
excepto el espacio
        '/[^\S ]+</s', // elimina en blanco antes de las etiquetas, excepto el
espacio
        '/(\s)+/s', // Acortar múltiples secuencias de espacios en blanco.
        '/<!--(.\s)*?-->/' // Borrar comentarios html
    );

    $replace = array(
        '>',
        '<',
        '\\1',
        ''
    );

    $buffer = preg_replace($search, $replace, $buffer);

    return $buffer;
}
?>
```

Con esto, ya tendríamos nuestro código html ofuscado y como mencionaba, una carga más rápida del archivo.

## Conclusiones

El ofuscamiento de código es una práctica que puede ayudarnos a proteger nuestro código fuente, ya que, prácticamente lo volvemos ilegible.

Uno de los motivos para ofuscar el código es tener protección intelectual, puesto que al ser ilegible no es sencillo de crear una copia pirata.

En internet podemos encontrar una gran cantidad de ofuscadores, algunos de paga y otros de licencia libre; incluso existen páginas las cuales pueden ofuscar una parte de nuestro código.

## Fuentes

<https://javascriptobfuscator.com/Javascript-Obfuscator.aspx>

<https://es.wikipedia.org/wiki/Ofuscaci%C3%B3n#Motivaci%C3%B3n>

<http://informatica.blogs.uoc.edu/2015/10/26/ofuscacion-de-codigo-te-reto-a-que-entiendas-este-programa/#:~:text=Seg%C3%BAn%20el%20Diccionario%20de%20la,hacer%20m%C3%A1s%20compleja%20su%20comprensi%C3%B3n.>

<https://wjharil.com/blog/ofuscar-html-minify/>