



# Módulo de análisis de ofuscación de código JavaScript para sistema de detección de ataques Drive-by-Download

Autor: Lucas Cruz Cruz

Tutor: Gabriel Maciá Fernández



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

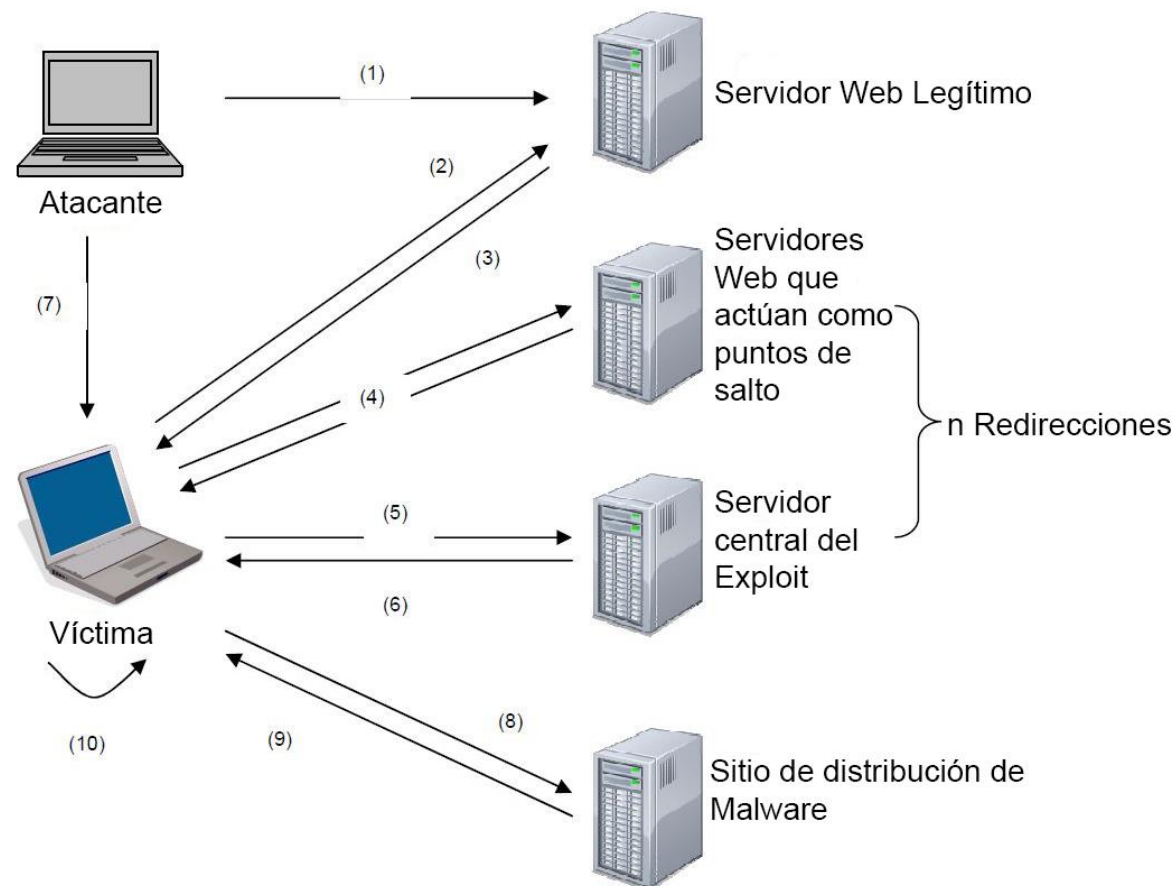
*Granada, Julio de 2016*

# INTRODUCCIÓN Y OBJETIVOS

# Ataques Drive-by-Download



# Fases de un ataque Drive-by-Download



# Ofuscación

## QUÉ

La ofuscación hace alusión al acto intencionado de modificar un código con el objetivo de que este sea **difícil de leer e interpretar a simple vista**.

## PARA QUÉ

En este caso, se aplica al código JavaScript en el que **se ocultan comandos** necesarios para que el ataque Drive-by-Download continúe

## CÓMO

Suelen utilizarse las propias **operaciones primitivas de JavaScript**

# Técnicas de ofuscación (1): aleatoria

## NO OFUSCADO

```
<script type="text/javascript">  
var nombre = "Lucas";  
document.write(nombre);  
alert("Hola");  
</script>
```

## OFUSCADO

```
<script type="text/javascript">  
2sd46wd562erw = base64_decode("THVjYXMNCg==");  
//gZRnOsS11eXaumft6jfRtzG3nmPly4  
document.write  
(nombre) " /*w2uyhES  
yNT0x2ZNwJVqKb39y9liUeWQpLTC XuMVkadDhQ8PX14or*/;  
alert( "Hola");  
  
</script>
```

# Técnicas de ofuscación (2): datos

NO OFUSCADO	OFUSCADO
<pre>var x=10;  function clave(){     alert("PalabraClave");     var y=x+1; }  eval(clave());</pre>	<pre>var string1="cla"; var string2="ve"; var string3="()"; var str = "PalabraClave"; var res = str.replace("PalabraClave", "OtraPalabra"); var x=eval("2*5");  function clave(){     alert(res);     var y=x+1; }  eval(string1+string2.concat(string3));</pre>

# Técnicas de ofuscación (3): codificación

## NO OFUSCADO

```
document.write('Hello world!');  
  
var nombre="Lucas";  
  
alert(nombre);
```

## OFUSCADO

```
function decode(c){  
    //FUNCIÓN DE DECODIFICACIÓN  
    return mystring;  
}  
  
var c="epdvnfou/xsjuf)(lfmmp!xpsme(*)";  
  
var mystring=decode(c);  
  
eval(mystring);  
  
var nombre="\x4C\x75\x63\x61\x73";  
  
alert(nombre);
```



# Técnicas de ofuscación (4): estructura lógica

NO OFUSCADO	OFUSCADO
<code>document.write('Hello world');</code>	<pre>var j=111;  if(j&lt;10){      alert("Warning");  }  var i=0;  for(i=0;i&lt;=10000;i++){      if(i==1){          document.write('Hello world');      }  }</pre>

# Objetivos del proyecto



Desarrollar un **módulo de clasificación** de código JavaScript que permita la detección de código ofuscado.



Descarga de sitios webs con **HTMLUnit** y extracción del código JavaScript contenido en el mismo.



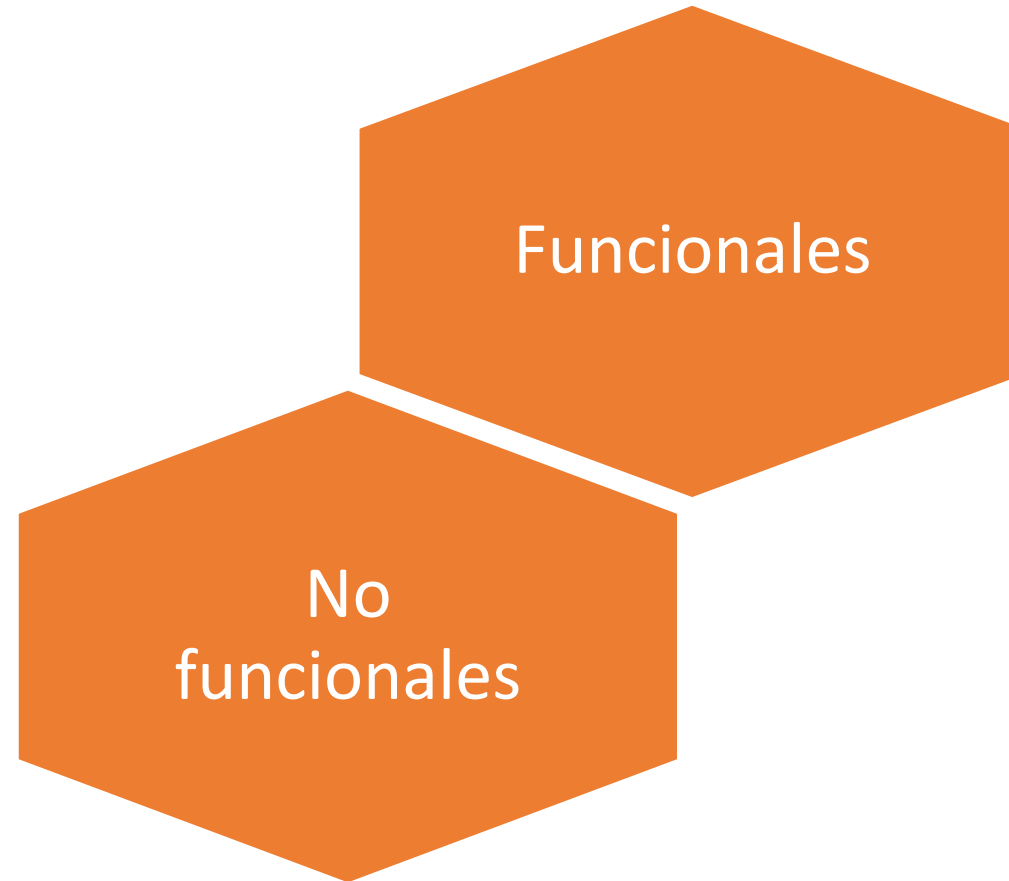
Creación de una **interfaz** para la interacción del sistema con el usuario.



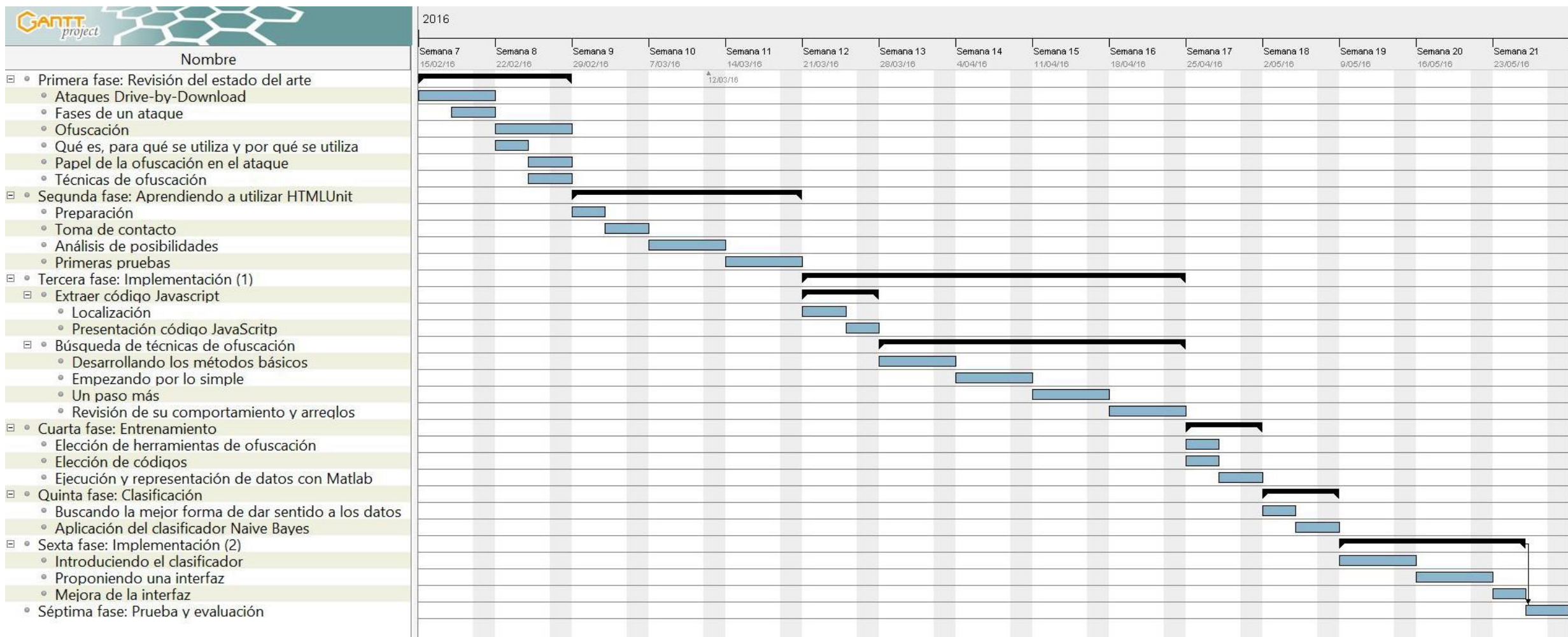
Sentar las bases de un sistema mayor que pueda hacer frente de forma eficaz a los ataques Drive-by-Download.

# ANÁLISIS Y METODOLOGÍA

# Requisitos



# Planificación



# Presupuesto

Concepto	Coste
Salario desarrollador	5.000 euros
Salario supervisor	1.000 euros
Material utilizado	167 euros
Software	42,35 euros
Gastos indirectos	1.200 euros
<b>Total:</b>	<b>7.409,35 euros</b>

# Tecnologías utilizadas: HTMLUnit

## Características

Es un navegador sin interfaz

Simular diferentes navegadores y versiones

Soporte HTTP, HTTPS, cookies, HTML, acceso al DOM...

Excelente soporte para JavaScript

## Objetos y métodos

WebClient

- BrowserVersion
- getPage

HTMLPage

- getElementsByTagName

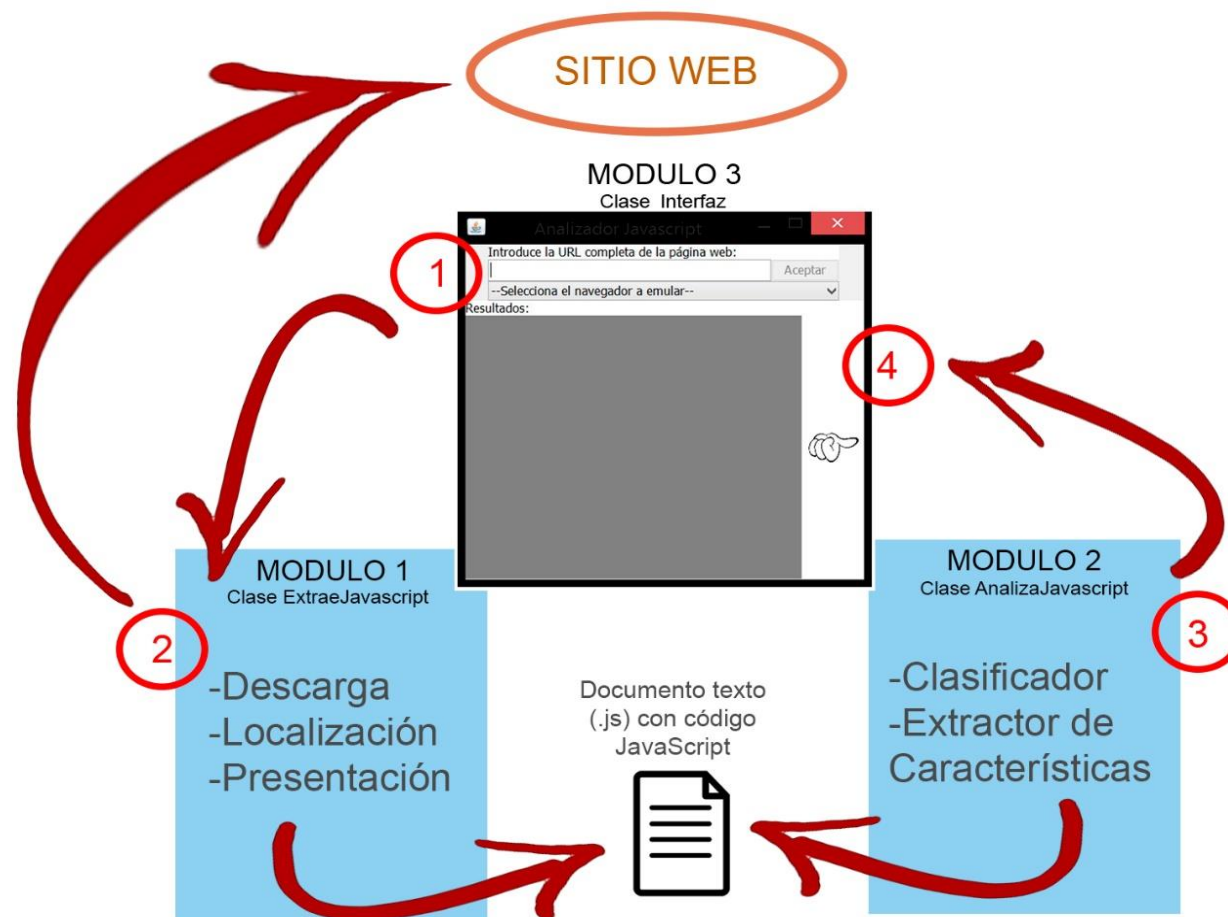
DomElement

- getTextContent
- getAttribute

# DISEÑO E IMPLEMENTACIÓN



# Arquitectura del sistema



# Extracción del código JavaScript

Descarga

Localización

Presentación

# Extracción de características



## Características implementadas

- Ratio de definiciones y usos de strings
- Número de ejecuciones de código dinámico
- Tamaño del código evaluado dinámicamente
- Cantidad de comentarios
- Cantidad de espacios en blanco
- Divisiones de strings
- Cantidad de operaciones
- Cantidad de Hexadecimales y Unicode

## Métodos básicos

- Archivolenght
- Cuentapalabra
- Cuentacaracter
- Extraeargumentos

# Clasificador

## Clasificador Naive Bayes

$$P(V|A) = \frac{P(A|V)P(V)}{P(A)}$$

## Teorema de Bayes

$$V_{nb} = \operatorname{argmax}_{v_j \in V} P(v_j) \prod P(a_i|v_j) \quad (1)$$

Se estima  $P(a_i|v_j)$  como:

$$P(a_i|v_j) = \frac{n_c + mp}{n + m} \quad (2)$$

# Entrenamiento (1)

## Herramientas ofuscación

<https://www.daftlogic.com/projects-online-javascript-obfuscator.htm>

<http://www.danstools.com/javascript-obfuscate/index.php>

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

<http://javascript2img.com/>

## Muestras

<https://www.google.es>

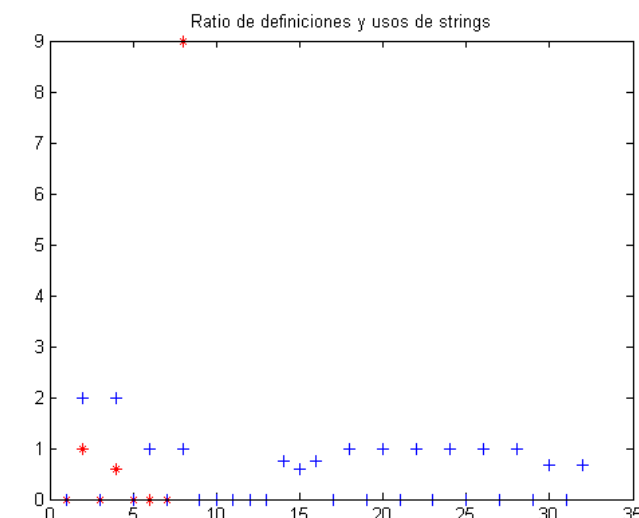
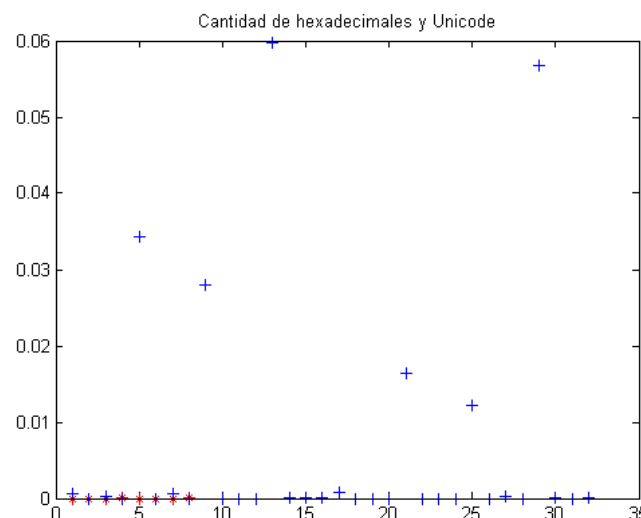
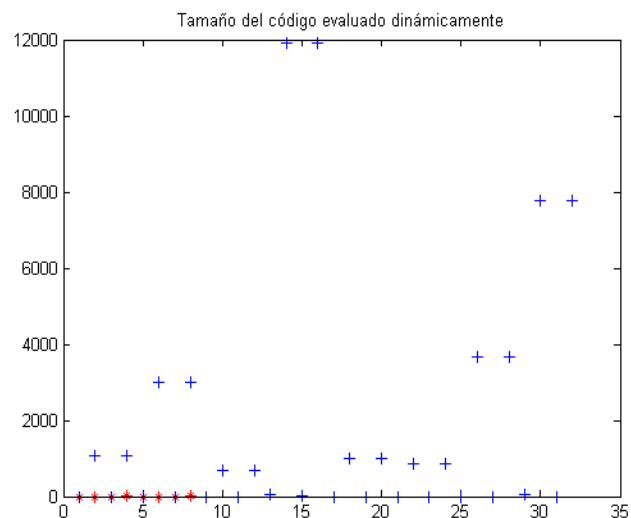
<https://es.wikipedia.org/>

<https://www.youtube.com/>

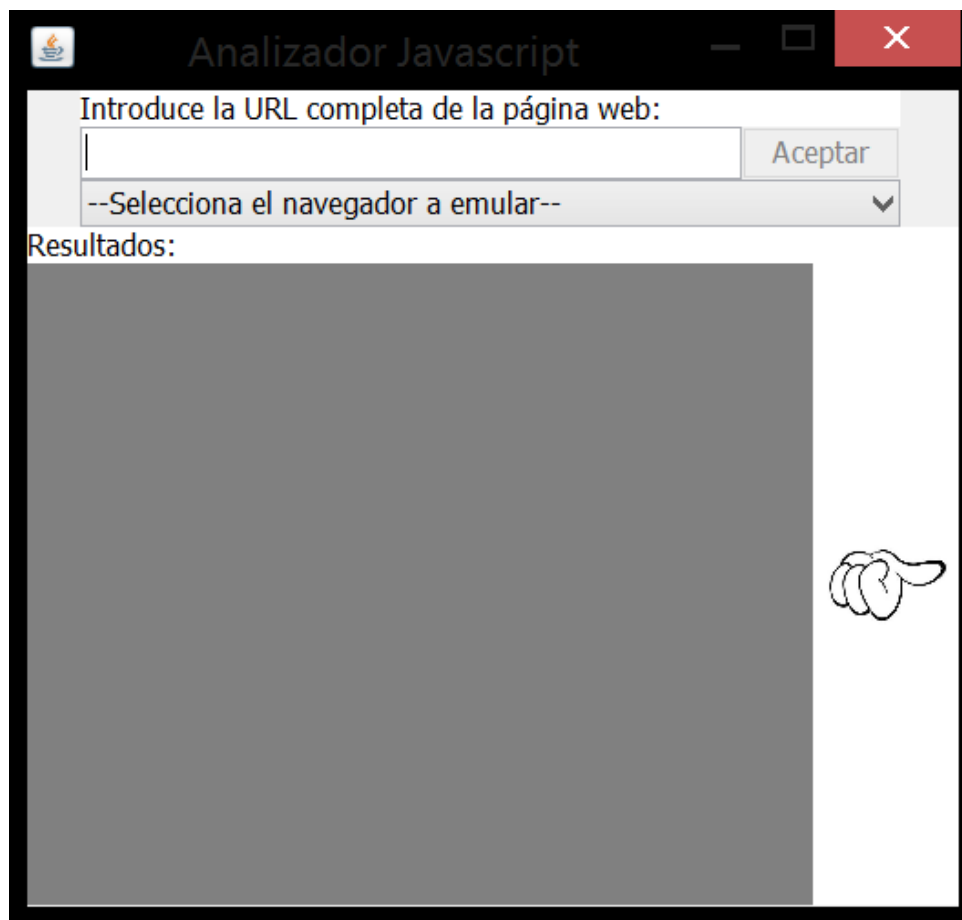
<https://www.uv.es/jac/guia/jscript/javascript.htm>

# Entrenamiento (2)

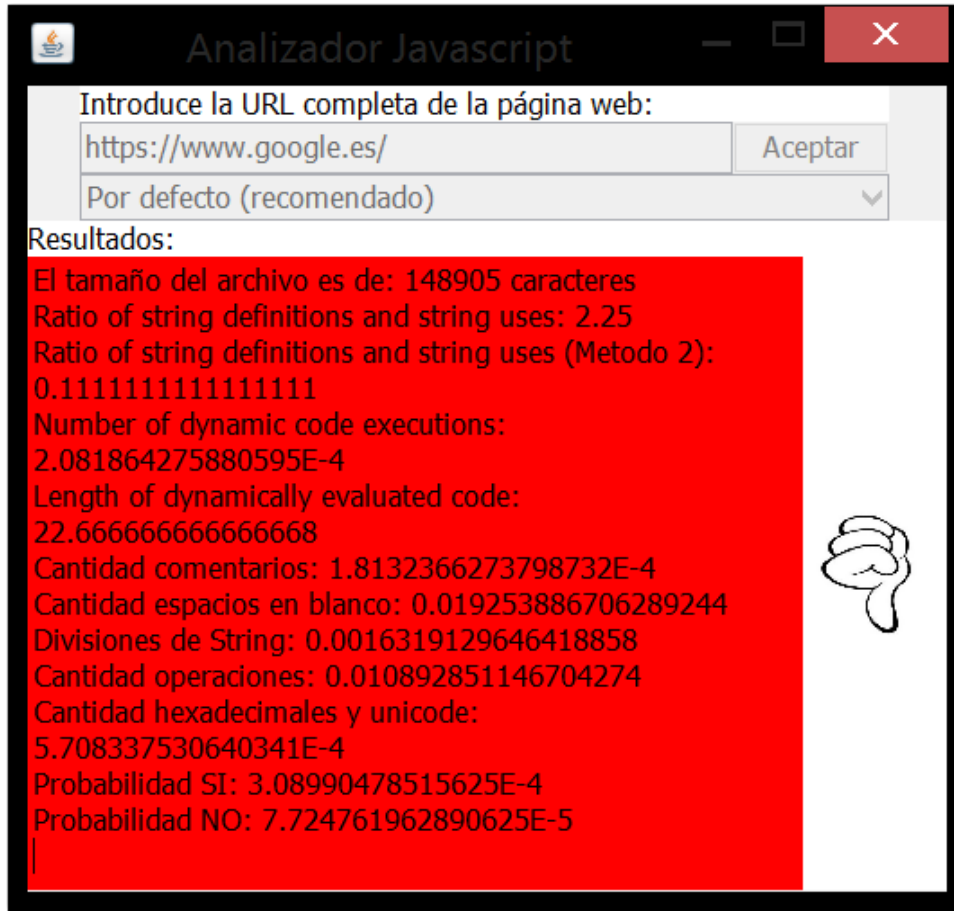
	Ratio >0	NODCE >0,0005	LDEC >0	CC <0,001	CEB <0,04	CO >0,015	DS >0,0015	CHU >0	Total
SI	0,5	0,3125	0,625	0,9375	0,937	0,5	0,1875	0,5	0,8
NO	0,375	0,25	0,125	0,5	0,375	0,25	0,125	0,12	0,2



# Interfaz y manual de uso



# Pruebas



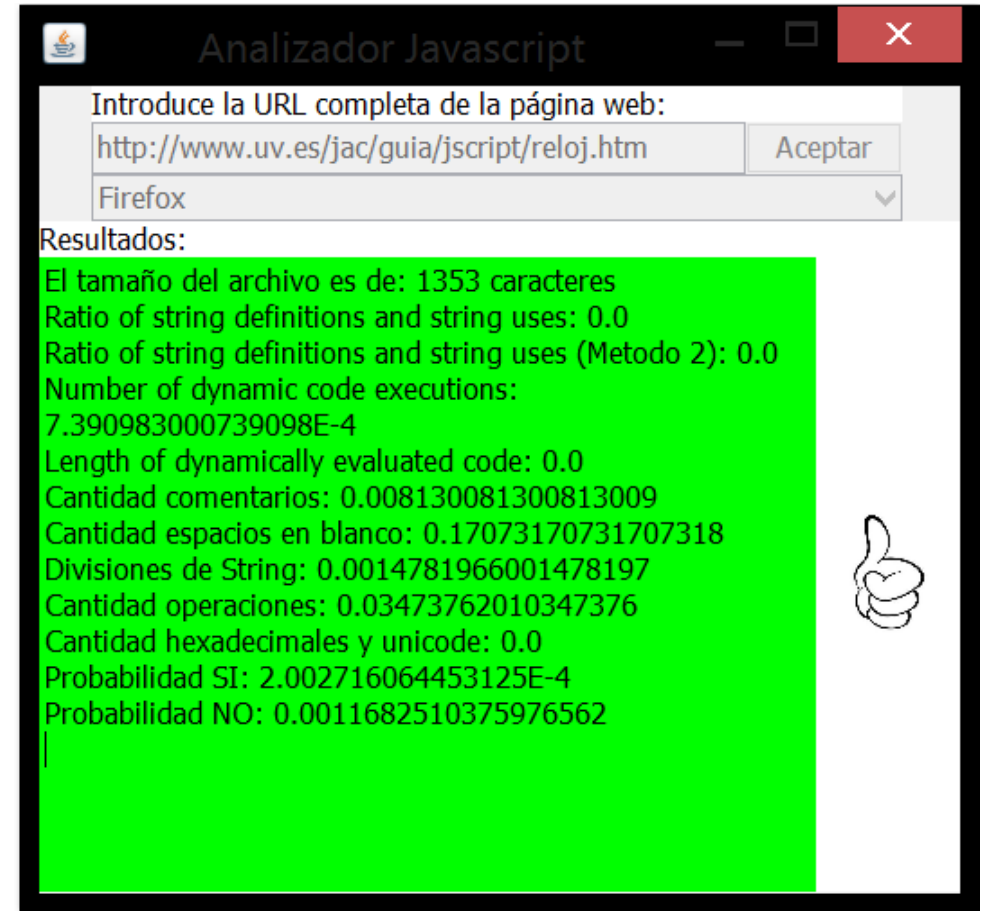
Analizador Javascript

Introduce la URL completa de la página web:

Por defecto (recomendado) ▼

Resultados:

El tamaño del archivo es de: 148905 caracteres  
Ratio of string definitions and string uses: 2.25  
Ratio of string definitions and string uses (Metodo 2): 0.1111111111111111  
Number of dynamic code executions: 2.081864275880595E-4  
Length of dynamically evaluated code: 22.666666666666666  
Cantidad comentarios: 1.8132366273798732E-4  
Cantidad espacios en blanco: 0.019253886706289244  
Divisiones de String: 0.0016319129646418858  
Cantidad operaciones: 0.010892851146704274  
Cantidad hexadecimales y unicode: 5.708337530640341E-4  
Probabilidad SI: 3.08990478515625E-4  
Probabilidad NO: 7.724761962890625E-5



Analizador Javascript

Introduce la URL completa de la página web:

Firefox ▼

Resultados:

El tamaño del archivo es de: 1353 caracteres  
Ratio of string definitions and string uses: 0.0  
Ratio of string definitions and string uses (Metodo 2): 0.0  
Number of dynamic code executions: 7.390983000739098E-4  
Length of dynamically evaluated code: 0.0  
Cantidad comentarios: 0.008130081300813009  
Cantidad espacios en blanco: 0.17073170731707318  
Divisiones de String: 0.0014781966001478197  
Cantidad operaciones: 0.03473762010347376  
Cantidad hexadecimales y unicode: 0.0  
Probabilidad SI: 2.002716064453125E-4  
Probabilidad NO: 0.0011682510375976562



# CONCLUSIONES Y TRABAJO FUTURO

# Conclusiones y líneas de desarrollo futuras



# Referencias (1)

- Marco Cova, Christopher Kruegel, and Giovanni Vigna. *Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code*.
- Aikaterinaki Niki. *Drive-by-Download attacks: effects and detection methods*.
- Wei Xu, Fangfang Zhang and Sencun Zhu. *The Power of Obfuscation Techniques in Malicious JavaScript Code: A Measurement Study*.

# Referencias (2)

- Zorabedian, J. (2014). How malware works: Anatomy of a drive-by download web attack (Infographic) | Sophos Blog. [online] Blogs.sophos.com. Disponible en: <https://blogs.sophos.com/2014/03/26/how-malware-works-anatomy-of-a-drive-by-download-web-attack-infographic/>
- Bowler, M., Guillemot, M. and Ashour, A. (2016). HtmlUnit – Welcome to HtmlUnit. [online] Htmlunit.sourceforge.net. Disponible en: <http://htmlunit.sourceforge.net/>
- Eclipse.org. (2016). [online] Disponible en: <https://eclipse.org/>
- Junit.org. (2016). JUnit. [online] Disponible en: <http://junit.org/>
- Oracle.com. (2016). Software Java | Oracle España. [online] Disponible en: <https://www.oracle.com/es/java/index.html>
- Scribd. (2016). Características del lenguaje Java VENTAJAS Y DESVENTAJAS. [online] Disponible en: <https://es.scribd.com/doc/165321281/Caracteristicas-del-lenguaje-Java-VENTAJAS-Y-DESVENTAJAS>
- Text Analysis blog | Aylien. (2015). Naive Bayes for Dummies; A Simple Explanation. [online] Disponible en: <http://blog.aylien.com/post/120703930533/naive-bayes-for-dummies-a-simple-explanation>
- Inf.u. (2016). [online] Disponible en: <http://www.inf.u-szeged.hu/~ormandi/ai2/06-naiveBayes-example.pdf>
- Es.wikipedia.org. (2016). JavaScript. [online] Disponible en: <https://es.wikipedia.org/wiki/JavaScript>
- Es.wikipedia.org. (2016). Java (lenguaje de programación). [online] Disponible en: [https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
- Es.wikipedia.org. (2016). MATLAB. [online] Disponible en: <https://es.wikipedia.org/wiki/MATLAB>
- Es.wikipedia.org. (2016). HTML. [online] Disponible en: <https://es.wikipedia.org/wiki/HTML>
- Es.wikipedia.org. (2016). Document Object Model. [online] Disponible en: [https://es.wikipedia.org/wiki/Document\\_Object\\_Model](https://es.wikipedia.org/wiki/Document_Object_Model)
- Htmlunit.sourceforge.net. (2016). HtmlUnit 2.22 API. [online] Disponible en: <http://htmlunit.sourceforge.net/apidocs/index.html?com/gargoylesoftware/htmlunit/>
- GitHub. (2016). Build software better, together. [online] Disponible en: <https://github.com/>
- Paramio, C. (2011). Conociendo GitHub, el servicio donde alojar tus repositorios Git (como el nuestro). [online] Genbetadev.com. Disponible en: <http://www.genbetadev.com/sistemas-de-control-de-versiones/conociendo-github-el-servicio-donde-alajar-tus-repositorios-git-como-el-nuestro>

