

# MIGAS DE PAN



13/11/2017

Sofía Valera & Miranda Arroyo

Generador de migas de pan

# MIGAS DE PAN

SOFÍA VALERA & MIRANDA ARROYO

## Contenido

EXPLICACIÓN DEL CÓDIGO:.....	2
FUNCIONES EMPLEADAS:.....	6
IMPLEMENTACION MIGAS:.....	7

## EXPLICACIÓN DEL CÓDIGO:

Lo primero que hacemos es declarar algunas variables:

1. La variable **cadena** guardará todo el contenido URL pasado por parámetro pero en minúscula (`url.toLowerCase()`).
2. La variable **reemplazarHTTP** reemplaza dentro del contenido de **cadena** las expresiones “http” mediante una expresión regular y en su lugar dejará un espacio vacío.
3. La variable **separación\_url** almacena en cada posición de un array los elementos que se encuentran separados por “/” dentro de la variable **reemplazarHTTP**.
4. La variable **array\_Reservadas** almacena dentro de un array las palabras que no se pueden incluir en el acrónimo.
5. La variable **UrlValida** la inicializamos como un array vacío.

```
function generateBC(url, separator) {  
  
    var cadena = url.toLowerCase();  
  
    var reemplazarHTTP = cadena.replace(/(http(s?)):\:\/\/gi, "");  
    var separacion_url = reemplazarHTTP.split("/");  
    var migas = "";  
  
    var array_Reservadas = ["the", "of", "in", "from", "by", "with", "and", "or", "for",  
"to", "at", "a"];  
  
    var urlValida = [];
```

En este primer bloque comprobamos la longitud de la variable “separador\_url”, si la longitud es 0 ó 1 añadiremos a la variable “migas” la etiqueta `span` y si no añadiremos a “migas” la etiqueta `href`. Como es el principio de la dirección URL, las migas siempre comenzaran por HOME.

```
if(separacion_url.length===0 || separacion_url.length===1){  
  
    migas = '<span class="active">HOME</span>';  
}else{  
  
    migas = '<a href="/">HOME</a>'+separator;  
}
```

La función que nos separa la url en migas la llevaremos a cabo dentro de un bucle `for` teniendo en cuenta la longitud del array **separación\_url**.

```
for(let i=1; i<separacion_url.length; i++){  
  
    urlValida.push(separacion_url[i]);  
  
    if(!comprobarIndex(separacion_url[i]) &&  
!comprobarPunto(separacion_url[i])){
```

```
if(comprobarGuion(separacion_url[i])){
```

La primera comprobación que haremos será ver si dentro de la posición **i** del array **separación\_url**, **no** tiene el string “index.” Y si **no** tiene un punto llamando a las funciones **comprobarIndex** y **comprobarPunto**. Si no lo tiene, entraremos dentro de la función.

Una vez dentro, llevaremos a cabo una serie de comprobaciones:

1. Comprobaremos si el string tiene guiones y en caso de tenerlos:
  - a) Comprobaremos si el string tiene un carácter de interrogación o un carácter de almohadilla.
  - b) Si la longitud es mayor de 30 caracteres o menor o igual a 30 caracteres, dependiendo de esto realizaremos una tarea u otra.
2. Si no tiene guiones, comprobaremos la última posición del array para añadir la etiqueta **span** y nuevamente comprobaremos si tiene un carácter de interrogación o un carácter de almohadilla.

En caso de que el string contenga un carácter de almohadilla o de interrogación; crearemos la variable **reemplazarCaracteres** y cambiaremos las almohadillas o las interrogaciones con **replace** y mediante una expresión regular por ‘8’. A continuación mediante **split** quitaremos los ‘8’ creando la variable **separarCaracteres** y nos quedaremos con la primera posición del array (así quitamos todo lo que esté detrás de ‘8’).

Después comprobaremos si la longitud de la posición **i** del array **separación\_url** es mayor que 30, si es así:

Comprobamos la última posición del array o la posición siguiente para ver su valor, llamamos a la función **acrónimo** y añadimos a las migas la etiqueta **span** y el acrónimo, si no se trata de la última posición, añadimos a las migas la etiqueta **a href** y reemplazamos las comas por / y llamamos a la función **acrónimo**.

```
if(comprobarGuion(separacion_url[i])){

    if(comprobarInterrogacion_almohadilla(separacion_url[i])){

        var reemplazarCaracteres = separacion_url[i].replace(/[#?]/g, '8');
        var separarCaracter = reemplazarCaracteres.split('8');
        separacion_url[i] = separarCaracter[0];
    }

    if(separacion_url[i].length>30){
        if(i === (separacion_url.length)-1 || comprobarIndex(separacion_url[i+1])){
            migas += '<span class="active">' + acronimo(separacion_url[i],
                array_Reservadas) + '</span>';
        }else{
            migas += '<a
href="'+urlValida.toString().replace(/,/g, '/')+'"/>' + acronimo(separacion_url[i],
array_Reservadas) + '</a>' + separator;
        }
    }
}
```

Si la longitud es menor o igual a 30:

Comprobamos la última posición del array o la siguiente para ver su valor, si tiene “index.” Añadimos a la variable **migas** la etiqueta **span** seguido de la cadena en mayúsculas y reemplazamos los guiones por espacios en blanco.

Si no es así, añadimos a **migas** la etiqueta **span** seguido de nuestro array `urlValida`, que es convertida a `String` y con la expresión regular reemplazamos las comas por `/` seguido de la cadena en mayúsculas reemplazando los guiones por espacios en blanco.

```
}else if(separacion_url[i].length<=30){ //la longitud es menor o igual a 30

    if(i === (separacion_url.length)-1 || comprobarIndex(separacion_url[i+1])){
        migas += '<span class="active">' + separacion_url[i].toUpperCase().replace(/-/g, " ") + '</span>';
    }else{
        migas += '<a'
href="/" + urlValida.toString().replace(/,/g, "/") + "/">' + separacion_url[i].toUpperCase().replace(/-/g, " ") + '</a>' + separator; //mostramos etiquetas <a>
    }
}
```

Si la cadena no tiene guion:

Comprobamos si es la última posición del array o si la siguiente tiene "index.", si es así, comprobamos si el string tiene el carácter almohadilla o interrogación, si lo tiene lo reemplazamos por el carácter '8'

Metemos en cada posición del array **separarCaracter** mediante **Split**, la cadena sin el carácter '8' y añadimos a las migas la etiqueta **span** seguido de la primera posición del array **separarCaracter** en mayúsculas. (así eliminamos todo lo que venga detrás del carácter '8').

En caso de no tener ni almohadillas ni interrogaciones, añadimos a las migas la cadena en mayúsculas y la etiqueta **span**.

Si no la cadena no tiene "index." Añadimos a las migas la etiqueta **a href**.

```
}else { //si no tiene guion

    if(i === (separacion_url.length)-1 || comprobarIndex(separacion_url[i+1])){
        if(comprobarInterrogacion_almohadilla(separacion_url[i])){

            var reemplazarCaracteres = separacion_url[i].replace(/[#?]/g, '8');
            var separarCaracter = reemplazarCaracteres.split('8');

            migas += '<span class="active">' + separarCaracter[0].toUpperCase() +
                '</span>';
        }else{

            migas += '<span class="active">' + separacion_url[i].toUpperCase() +
                '</span>'; //mostramos las etiquetas span
        }

        }else if(!comprobarIndex(separacion_url[i])){ //si no tiene index.

            migas += '<a href="/" + urlValida.toString().replace(/,/g, "/") +
                "/">' + separacion_url[i].toUpperCase() + '</a>' + separator;

        }
    }
}
```

Si la cadena tiene punto y no tiene “index.”:

Comprobamos si la cadena tiene guiones y si tiene puntos, en caso de tenerlos creamos un array(**separarStringGuion**) que elimine los puntos con **Split** y nos quedamos con la primera posición (0) del array creado y si la longitud de la cadena es menor o igual a 30, comprobamos si se trata de la ultima posición, de ser así añadimos a migas la etiqueta **span** con la primera posición de **separarStringGuion** en mayúsculas y remplazando los guiones por espacios en blanco.

Si la cadena no tiene punto, creamos el array **separarString** que nos meterá en una posición distinta del array la cadena, pero sin puntos y añadimos a las migas la etiqueta **span** seguido de la primera posición de **separarString** en mayúsculas.

```
if(comprobarPunto(separacion_url[i]) && !comprobarIndex(separacion_url[i])){ //
si tiene punto entra en el if

    if(comprobarGuion(separacion_url[i]) && comprobarPunto(separacion_url[i])){
//si tiene guion y punto

        var separarStringGuion = separacion_url[i].split('.');
        if(separarStringGuion[0].length<=30){
            if(i === (separacion_url.length)-1){ //comprobar la ultima posicion del
array
                migas += '<span class="active">'+
                    separarStringGuion[0].toUpperCase().replace(/-/g, " ")+</span>';
//mostramos las etiquetas span
            }
        }
    }else{
        var separarString = separacion_url[i].split('.'); //separamos en un array
ej:holiday.html --> holiday(posicion 0) y html(posicion 1)

        migas += '<span class="active">'+separarString[0].toUpperCase()+</span>';
//muestra la ultima etiqueta que es <span>
    }
}
```

Por último, si la cadena tiene “index.” y su longitud es 2:

- Añadimos a **migas** la etiqueta **span** seguida de **HOME**.

Si en la última posición se encuentra algún carácter de almohadilla, interrogación o nada:

- Añadimos a **migas** la etiqueta **span** seguida de **HOME**.

```
if(comprobarIndex(separacion_url[i]) && separacion_url.length===2){
    migas = '<span class="active">HOME</span>'; //muestra la ultima

    if(i === (separacion_url.length)-1 && (separacion_url[i]=== "#" ||
separacion_url[i]=== "?" || separacion_url[i]=== "")){
        migas = '<span class="active">HOME</span>';
    }
}
```

```
}
```

Lo que devolverá el programa serán las migas como cadena:

```
return migas;
```

## FUNCIONES EMPLEADAS:

La función comprueba si tiene "index." y si lo encuentra devuelve true.

```
function comprobarIndex(buscarCadena) {  
    if (buscarCadena.indexOf("index.") !== -1) {  
        return true;  
    }  
}
```

La función comprueba si contiene "." y si lo encuentra devuelve true.

```
function comprobarPunto(buscarCadena) {  
    if (buscarCadena.indexOf(".") !== -1) {  
        return true;  
    }  
}
```

La función comprueba si tiene un "-" y si lo contiene devuelve true.

```
function comprobarGuion(buscarCadena) {  
    if (buscarCadena.indexOf("-") !== -1) {  
        return true;  
    }  
}
```

La función comprueba la interrogación y la almohadilla, si encuentra una u otra devuelve true.

```
function comprobarInterrogacion_almohadilla(buscarCadena) {  
    if (buscarCadena.indexOf("?") !== -1 || buscarCadena.indexOf("#") !== -1) {  
        return true;  
    }  
}  
  
return migas;
```

Por último, a la función acrónimo le pasamos dos parámetros, la url separada y el array de palabras reservadas, cuando entra en la función separamos por guiones, ya que el acrónimo nos llega varias palabras separadas por guiones de las cuales algunas pueden ser reservadas.

Por ello recorreremos nuestro array de **separarGuiones** y dentro hace la comprobación de si no está incluida en nuestro array reservadas, nuestro array de separar guiones. Una vez dentro de nuestro `if` cogemos la primera letra y al final de la función devolvemos esa letra en mayúsculas. Así con todas las palabras, hasta formar el acrónimo (ej: PIPER ).

```
function acronimo(separacion_url, array_Reservadas){

    var separarGuiones = separacion_url.split('-');
    var acronimos = "";

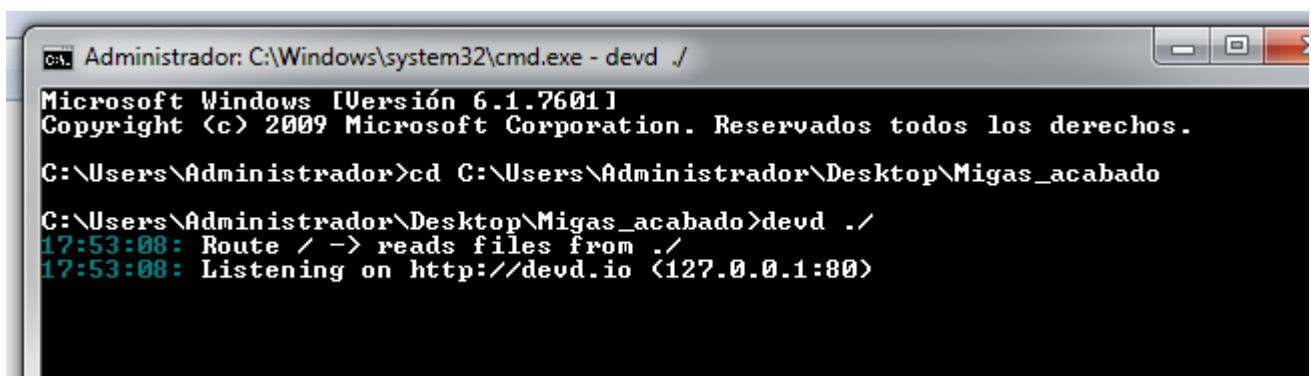
    for(let i=0; i<separarGuiones.length; i++){

        if(!array_Reservadas.includes(separarGuiones[i])){
            acronimos = acronimos+separarGuiones[i].charAt(0);
        }
    }

    return acronimo_mayusculas = acronimos.toUpperCase();
}
```

## IMPLEMENTACION MIGAS:

Arrancamos el servidor:



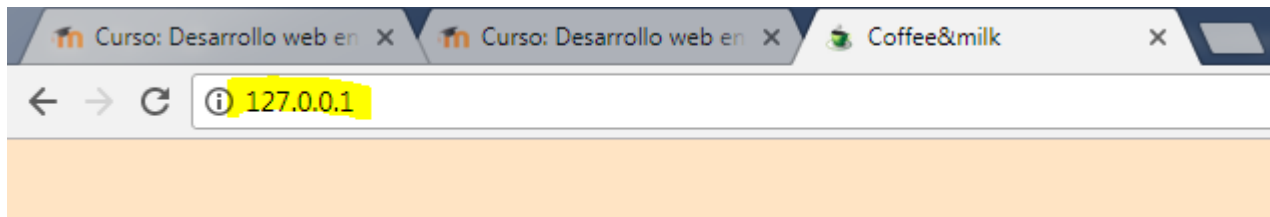
```
Administrator: C:\Windows\system32\cmd.exe - devd ./
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Administrador>cd C:\Users\Administrador\Desktop\Migas_acabado

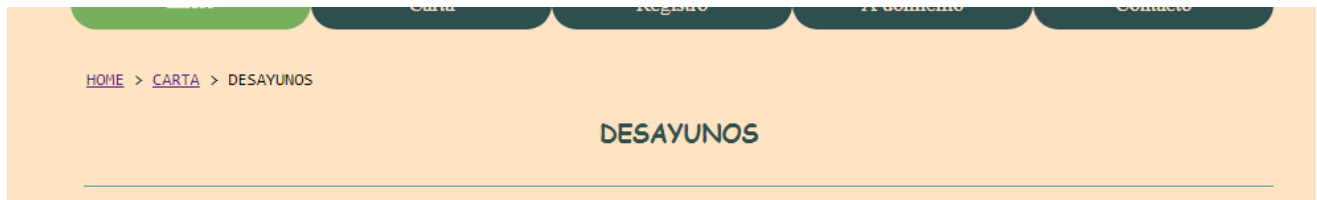
C:\Users\Administrador\Desktop\Migas_acabado>devd ./
17:53:08: Route / -> reads files from ./
17:53:08: Listening on http://devd.io (127.0.0.1:80)
```



Vemos el resultado en la página:



Miga con tres niveles:



Para cambiar 127.0.0.1 por, [www.coffee&milk.com](http://www.coffee&milk.com) → modificar en Windows el archivo hosts.