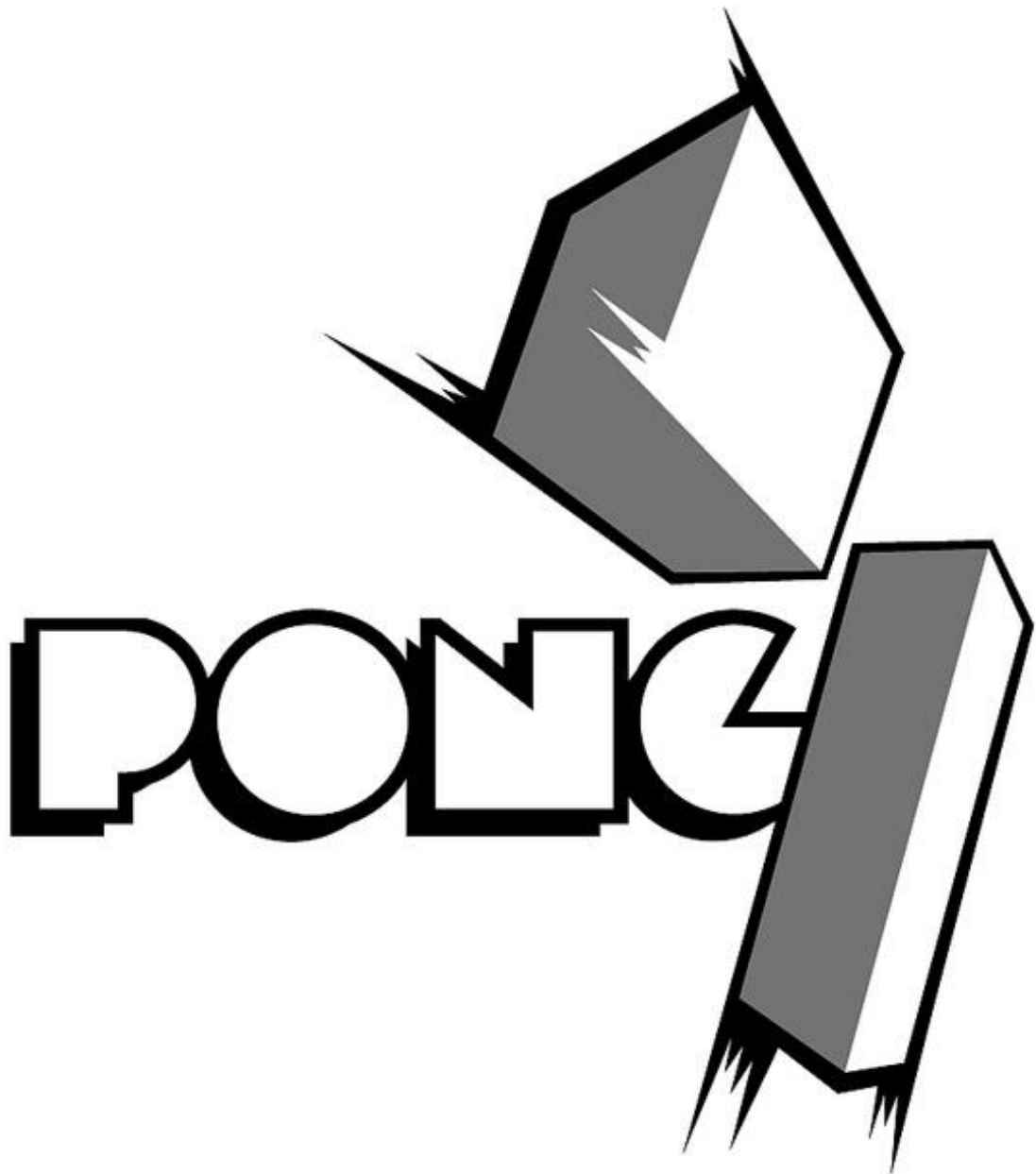


# *Normativa de código.*

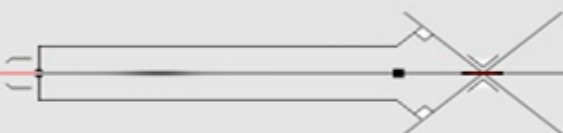
Técnicas Ágiles de Desarrollo del Software



*TADS-LEGANES//GRUPO3*

# Normativa de código.

Introducción .....	3
Definiciones .....	3
Idioma elegido para la codificación .....	4
JavaScript .....	4
Comentarios .....	4
Nombres de fichero .....	5
Estándar para Nombres y Variables .....	5
Clases y Miembros de Clases .....	5
Visibilidad .....	6
Indentación y llaves .....	6
HTML .....	7
Comentarios. ....	7
Nombres de ficheros. ....	7
Escritura de etiquetas.....	7



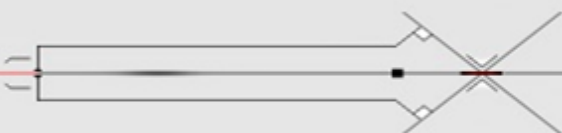
## INTRODUCCIÓN

---

Una de las debilidades que tiene la generación de código es la falta de un estándar reconocido que facilite la comprensión entre distintos desarrolladores. Por ello, se ha establecido en el grupo 3 de *Técnicas Ágiles de Desarrollo Software* un estándar de codificación, aceptado e implantado por todo el equipo, para asegurarse de que todos los programadores del grupo trabajen de forma coordinada y a su vez, facilitar la comprensión.

### Definiciones

- **Sensitive-case:** expresión en inglés que se refiere a la sensibilidad a mayúsculas y minúsculas.
- **Indentar:** mover un bloque de texto hacia la derecha insertando espacios o tabuladores, para así separarlo del margen izquierdo y mejor distinguirlo del texto adyacente
- **HTML:** sigla en inglés de **HyperText Markup Language**. Hace referencia al lenguaje de marcado para la elaboración de páginas web.



## IDIOMA ELEGIDO PARA LA CODIFICACIÓN

Toda la codificación del proyecto deberá hacerse en inglés, así como los identificadores, nombres de ficheros, variables, constantes, métodos y clases. Los comentarios se realizarán en castellano para facilitar la comprensión del código desarrollado.

## JAVASCRIPT

### Comentarios

En cada una de los ficheros se va a incluir las siguientes líneas de código:

En los ficheros con situados en la carpeta **src** irán definidas estas líneas antes del código, mientras que los ficheros situados en la carpeta **spec** irán definidas las líneas después del *describe*.

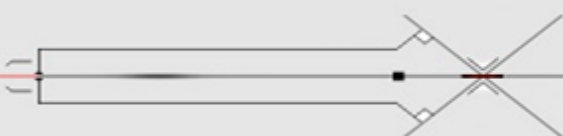
- *Fichero de ejemplo situado en la carpeta **src**.*

```
/**
Descripción:
@author: TADS-LEGANES//GRUPO3
@subject: Técnicas Ágiles de Desarrollo de Software.
@year: 2017
*/
var modo = false;
```

- *Fichero de ejemplo situado en la carpeta **spec**.*

```
describe('DrawEngine test suite', function() {
/**
Descripción:
@author: TADS-LEGANES//GRUPO3
@subject: Técnicas Ágiles de Desarrollo de Software.
@year: 2017
*/
```

Los comentarios se podrán realizar tras una declaración de variable o después de un bucle. Además, los comentarios que ocupen más de 100 caracteres se escribirán en múltiples líneas.



## Nombres de fichero

Los ficheros se van a nombrar en minúsculas para evitar error con posibles plataformas **sensitive-case**. Además, los ficheros tienen que finalizar con la extensión .js y no deben contener ningún signo de puntuación.

## Estándar para Nombres y Variables

### Clases y Miembros de Clases

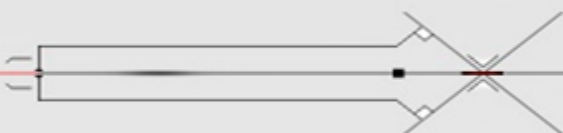
Las clases han de ser nombradas con sustantivos identificativos e unívocos. Los sustantivos deben estar en minúscula excepto su primer carácter que ha de estar en mayúscula. El primer sustantivo del nombre de la clase, el primer carácter del sustantivo debe ir en mayúsculas. Deben ser descriptivos en concordancia a la funcionalidad de la clase. Cada clase, además, se realizará en un fichero distinto.

Los parámetros deberán estar definidos en minúsculas a excepción de estar compuestos por varios sustantivos, en tal caso el primer sustantivo debe ser íntegramente en minúsculas y los demás únicamente podrá estar en mayúsculas el primer carácter de cada sustantivo.

Las variables globales y locales seguirán el mismo criterio que los parámetros. Cada variable global deberá estar acompañada de un comentario indicando su uso.

Las constantes estarán definidas íntegramente en mayúsculas. En caso de estar compuestas de varios sustantivos, cada sustantivo se separa con el carácter “\_” (barra baja) pero siempre que no comience y acabe por ello. Y al igual que las variables globales, deberán estar acompañadas de un comentario indicando su uso.

Para las posibles cadenas de texto, se usarán comillas simples (‘’) en vez de comillas dobles (‘‘’’). Las cadenas de texto con una longitud mayor a 100 caracteres deben ser escritas en múltiples líneas usando concatenación.



## Visibilidad

El código deberá seguir una estructura que pueda ser leída fácilmente atendiendo a los siguientes criterios:

Al comienzo del código se definirán las variables que se usarán a lo largo del código del método o clase, sin embargo, están permitidas las declaraciones de variables que se usen cerca de la sección de código que ha sido declarada.

Por cada línea irá únicamente una sola línea de código, estando prohibidas las sucesivas declaraciones en la misma línea, por ejemplo:

```
if (player == true ) {
    if (idPausaP1.style.visibility == 'visible') {
        idPausaP1.style.visibility = 'hidden';
```

Para distinguir entre variables locales de una función y los campos propios de una clase se añade **this** a los nombres de los campos.

En el caso de que se utilicen bucles que requieran la inicialización de una variable para usar como índice, se creará previamente, nunca dentro.

## Indentación y llaves

La **indentación** es un proceso que nos sirve para una mejor lectura y mantenimiento de nuestro código. En este caso, para no tener problemas con los distintos editores de texto, se va a **indentar** con cuatro espacios en vez de tabulaciones.

A su vez, las variables deben ser declaradas siempre en una nueva línea, y no ir separadas por comas. No se deben alinear los nombres de las variables, aunque esto no es una regla sino una sugerencia.

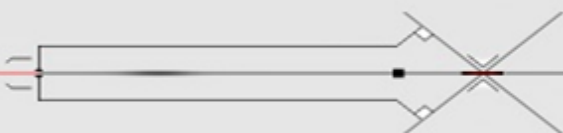
El uso de puntos y comas es obligatorio.

En los condicionales y bucles se debe dejar espacios después de una palabra clave y antes del corchete.

```
if (player == true ) {
```

Sin embargo, en las funciones solo se colocará un espacio entre el paréntesis de cierre y el corchete final.

```
pauseGame: function(player) {
```



## HTML

---

### Comentarios.

En cada uno de los ficheros HTML vamos a incluir las siguientes líneas comentadas:

```
<!--  
Descripcion:  
@Author: TADS-LEGANES//GRUPO3  
@Subject: Técnicas Ágiles de Desarrollo de Software  
@year: 2017  
-->
```

### Nombres de ficheros.

A la hora de nombrar ficheros y carpetas que contengan HTML se van a seguir las siguientes normas:

- Todo el texto se escribirá en minúsculas.
- Aunque una palabra tenga acento, se escribirá sin él.
- El primer carácter o signo que se escriba no debe ser un número, ya que puede traer consigo una mala interpretación del nombre.
- No utilizar ningún otro carácter especial a excepción de guion bajo “\_”.

### Escritura de etiquetas.

La mayoría de instrucciones en HTML tienen etiqueta de apertura y etiqueta de cierre. Es necesario poner las dos en los extremos del elemento que queremos ver afectado. A pesar de que en algunos casos no es necesario incluir la etiqueta de cierre, para facilitar la lectura se van a incluir.

En HTML es indiferente escribir las palabras clave de las etiquetas en mayúsculas, minúsculas, o una combinación de ambas. Podemos escribir por ejemplo `<br>` o `<BR>` o incluso `<bR>`. En nuestro caso, solo vamos a utilizar las etiquetas en minúsculas.

