Alejandro Pacheco Torres

1. ¿Para poder utilizar cualquier nombre de variable con seguridad en nuestros ficheros .JS que estrategia podemos adoptar?

Debemos adoptar la estrategia de espacio de nombres, donde se definirán las variables en un entorno único de ese programa. De esta manera evitamos conflictos con nombres de variables iguales o solapamientos de información erronea.

2. ¿Por qué razón funciona esa estrategia?

Esta estrategia es segura ya que llegados a este punto donde hay tantas variables que pueden llamarse igual, podemos acceder a la variable que deseamos sin temor a equivocarnos. Además es una práctica segura ya que el codigo queda más limpio de cara a otros programadores que vengan después de nosotros.

3.Da un **ejemplo práctico** en el que se explique el **conflicto y su solución**

Llegados a un punto donde la aplicación sea de un tamaño consistente, es posible que repitamos variables que por algún caso tengan un nombre en común. Uno de los conflictos más comunes que nos podemos llegar a encontrar es el solapamiento de la información, ya que supongamos que tenemos una variable llamada **X** y otro programador en otra sección del programa decida crear una variable **X,** lo más seguro que o su variable o la mía cambie de información, todo dependerá de qué variable sea declarada primero, por lo que se generaría un bug que podría dar inconsistencias en el programa. La solución sería que cada programador cree las variables en su propio espacio de nombres, de esta manera se evitará el conflicto de nombres iguales.

4. Llegir aquesta http://notasjs.blogspot.com.es/2012/04/el-patron-modulo-en-javascript.html del blog JavaScript a boli explicar que és el patró de [**Christian Heilmann**].

El patrón de Christian Heilman, denominado **Modulo Revelado** se simplifica y solucionan algunos problemas del patrón original, las propiedades y funciones privadas se declaran de forma diferente de las públicas. Ahora todas las propiedades se declaran igual. Al final, en el *return*, 'revelamos' sólo los métodos/propiedades que queremos hacer públicas. Haciendo las cosas de esta forma, facilita a la hora de convertir metodos o variables en publicas o privadas.