Manual de buenas practicas

Se procede a declarar en el siguiente documento, un conjunto flexibles de reglas a modo de recomendación para establecer un estándar que facilite la comprensión sintáctica de todo código generado en el proyecto a fines de mejor la manutención del mismo.

La implementación actual consiste de las siguientes tecnologías:

JavaScript (javascript, json, jQuery)  
 CSS (bootsrap)  
 HTML5

**Reglas de nombrado**

Todo elemento, ya sea variable, función o etiqueta, debe tener un ID o nombre, que siga la notación CAMEL CASE.

Los siguientes son ejemplos válidos de su aplicación:

* producto
* precioVenta
* comprobanteVenta
* montoCliente
* precioLista

**Creación y declaración de variables**

Toda declaración y / o creación de variable debe ser realizada en una única línea, a menos que crearla, implique el uso de arreglos o estructuras mas complejas.

Ejemplos:

var fieldMonto = document.getElementById("fieldMonto");

var fieldTarjeta = document.getElementById("fieldTarjeta");

**Separación**

Es importante, particularmente para funciones y estructuras de control (for, while, etc) que una vez declarada dicha estructura, se deje un espacio de al menos un renglón a la hora de abrir los brackets que definan el contexto.

Por ejemplo:

if (foo == true) {do something}  
 else {do something else}

Si bien el ejemplo es completamente funciona, es importante en este caso, por cuestiones de claridad, que el contexto de ejecución de cada estructura esté claramente delimitado, de la siguiente manera:

If (foo == true) {

Do something  
}

Else {

Do something else  
}

Identacion

En caso de que se utilicen herramientas que no identen nuestro código automáticamente, es más que una recomendación tabular correctamente el inicio y fin de cada bloque, de tal manera que los símbolos de cierre correspondan con los inicios de las líneas declarativas de tal bloque, obteniendo como resultado:  
  
function disableTimeField() {

if (document.getElementById("datepicker").disabled) {

document.getElementById("datepicker").disabled = false;

document.getElementById("timepicker").disabled = false;

}

else {

document.getElementById("datepicker").disabled = true;

document.getElementById("timepicker").disabled = true;

}

}

**Nota importante:** evitar la tabulación lograda con espacios.

**Estilo (CSS)**

Se recomienda no utilizar estilos incrustados (inner css) como ser en los siguientes casos:

<label **style="padding-right:50px"** class="form-check-label">

A menos que sea necesario para dar una estructura correcta.

Es siempre recomendable aplicar estilos a través de archivos, y relacionar los mismos con el stylesheet de la hoja HTML.

<link rel="stylesheet" href="css/bootstrap.css">

<link rel="stylesheet" href="css/estilos.css">