Código

1. Identar adecuadamente el Código.
2. Nombrar las variables y funciones con nombres descriptivos que representen lo que hacen, esta practica permite entender el código escrito por otra persona de forma más rápida.
3. Agregar comentarios descriptivos al código ayuda tanto a otros desarrolladores como a nosotros mismos cuando revisamos líneas de código escritas tiempo atrás.
4. Si se van a declaras varias variables que no se les va a asignar ningún valor al momento de ser declaradas, es una buena practica usar solo una vez la palabra reservada var o let, y luego colocar todas las declaraciones de variables separadas por coma, por ejemplo:
   1. Buena práctica:

var variable1, variable2, variable3;

* 1. Mala práctica:

var variable1;

var variable2, ;

var variable3;

1. Separar el código repetitivo en funciones que pueden ser reutilizadas, esto reduce las líneas de código y hace que sea más entendible, además de mejorar el rendimiento.
2. Cuando condicionales if/else que contienen una sola línea de código es una buena practica omitir los corchetes y colocar el código que se va ejecutar en la misma línea, esto con el fin de reducir la cantidad de línes de código, esto también aplica para el, por ejemplo:
   1. Buena práctica:

if(condición) función1();

else funcion2();

* 1. Mala práctica:

if(condición){

funcion1();

} else {

funcion2();

}

1. Cuando se usan condicionales if/else es una buena practica que el código a ejecutar en el else tenga el mismo formato que en el if y viceversa, por ejemplo:
   1. Buena práctica:

if(condición) función1();

else funcion2();

* 1. Mala práctica:

if(condición) funcion1();

else {

funcion2();

}

1. Es una buena mantener la misma estructura que ya se esta manejando dentro de un código que ya se encuentra escrito, por ejemplo cuando se usan comillas simples y comillas dobles en Javascript, así:
   1. Buena práctica:

var string1 = “texto1”;

var stgring2 = “texto2”;

var string3 = “texto3”;

* 1. Mala práctica:

var string1 = ‘texto1’;

var string2 = “texto2”;

var string3 = ‘texto3;

Base de Datos

1. Siempre que se crea una tabla nueva se le debe asignar una primary key.
2. La columna de una tabla que sirve para enlazar con otra tabla debe mantener el nombre que tiene la tabla principal, por ejemplo:
   1. Buena práctica:

tabla\_clientes

|  |  |  |
| --- | --- | --- |
| Id\_cliente | Cedula | nombre |
| 1 | 123425234 | Carlos Marin |
| 2 | 544353452 | Maria Cifuentes |

tabla\_pedidos

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_pedido | Id\_cliente | Cantidad | Valor |
| 1 | 1 | 5 | 10.000 |
| 2 | 2 | 8 | 15.000 |
| 3 | 1 | 10 | 80.000 |

* 1. Mala práctica:

tabla\_clientes

|  |  |  |
| --- | --- | --- |
| Id | Cedula | nombre |
| 1 | 123425234 | Carlos Marin |
| 2 | 544353452 | Maria Cifuentes |

tabla\_pedidos

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_pedido | Id\_cliente | Cantidad | Valor |
| 1 | 1 | 5 | 10.000 |
| 2 | 2 | 8 | 15.000 |
| 3 | 1 | 10 | 80.000 |

1. Nombrar las tablas con nombres descriptivos, por ejemplo:
   1. Buena práctica:

frm\_empresa1\_proceso1\_clientes

frm\_empresa1\_proceso2\_pedidos

* 1. Mala práctica:

Clientes

Tabla\_pedidos