



Ejercicios JavaScript (for, while y do-while)

//Ejercicios (for)

//Ejercicio 1

/*

Una persona desea invertir \$1000 en un banco, el cual le otorga un 2% de interés mensual ¿Cuál será la cantidad de dinero que esta persona tendrá al cabo de un año?

En el primer mes tendrá acumulado \$1000 más \$20 de interés (2% de 1000). En el segundo mes se le sumará un 2% a la base de \$1020 del mes anterior y así sucesivamente.

*/

//Ejercicio 2

/*

Crear un programa que ingrese una oración y muestre cuál es el carácter que más se repite.

Consideraciones

- No se debe contar el espacio en blanco.
- La oración a ingresar no debe estar vacía.
- Si hay más de un carácter que sea el que más se repita, se tomará en cuenta al primero que aparezca.

*/

//Ejercicios (while)

//Ejercicio 1

//aclaraciones previas:

//un número aleatorio entero se genera de la siguiente manera en JS

//Entero entre 0 y 9

//let enteroAleatorio = Math.floor(Math.random() * 10);

//Math.random() genera un número decimal entre 0 y 1 (sin incluirlo)

//al multiplicar, incremento el rango

//Math.floor() redondea el número decimal al entero más cercano

/*

Crear una aplicación que valide el ingreso a una
plataforma Online Banking a través de una clave Token.

Se debe tener en cuenta lo siguiente:

- La Clave Token debe ser un número aleatorio de 6 dígitos.
- El cliente debe ingresar los campos Usuario, Contraseña y Clave Token (todos obligatorios).
- El campo Usuario no distingue minúsculas o mayúsculas.
- El campo Contraseña es sensible a las minúsculas y mayúsculas.
- La clave Token aleatoria se le informa al usuario al pedirle que ingrese las credenciales.
- El cliente solo posee 3 intentos de logueo.
- Si alcanza los 3 intentos fallidos de forma consecutiva, la aplicación deberá informar al usuario que debe dirigirse a la sucursal del banco más cercana para poder desbloquear sus credenciales.
- Por cada intento fallido, la aplicación debe preguntar al cliente si desea continuar colocando las credenciales de manera correcta.

- Si el cliente coloca las credenciales de forma correcta, deberá informar que ha ingresado correctamente al Online Banking.

*/

//Ejercicios (do-while)

//Ejercicio 1

/*

Implementar un juego de adivinanza: el programa genera un número aleatorio entre 1 y 100 y le pide al usuario que lo adivine, indicando “más alto” o “más bajo” hasta que acierte. Cuando el usuario acierta, debe ver por pantalla cuál fue el número adivinado y cuántos intentos utilizó para adivinarlo.

*/