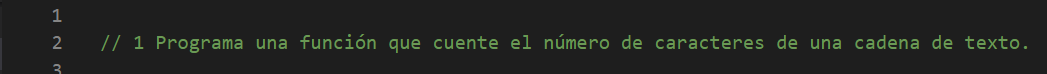
Brenda Samara Escobar Avila

Carnet: 2012-037200

Diseño de interfaces web con React JS

**TAREA 1**



// Definimos una función llamada contarCaracteres que toma una cadena como entrada.

function contarCaracteres(cadena) {

    // Usamos la propiedad length de la cadena para contar los caracteres.

    let numeroCaracteres = cadena.length;

    // Devolvemos una cadena que incluye el texto original y el número de caracteres.

    return `Texto: ${cadena}, Número de caracteres: ${numeroCaracteres}`;

  }

  // Ejemplo de uso:

  // Creamos una variable llamada cadena y le asignamos un valor "Hola Mundo".

  let cadena = "Hola Mundo";

  // Llamamos a la función contarCaracteres con la cadena como argumento y guardarmos el resultado en la variable resultado.

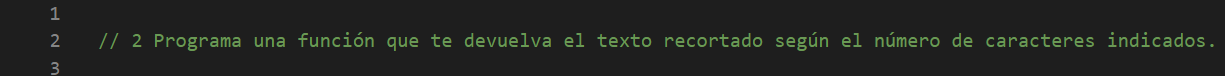
  let resultado = contarCaracteres(cadena);

   // Mostrar el resultado en la consola.

  console.log(resultado);

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente



// Definición de la función miFuncion.

function miFuncion(texto, numCaracteres) {

  //Comprobamos si el numCaracteres es mayor o igual a la longitud del texto.

    if (numCaracteres >= texto.length) {

      // Si es verdadero, devolver el texto completo.

      return texto;

    } else {

      // Si es falso, devolcer una subcadena del texto con los primeros numCaracteres caracteres.

      return texto.substring(0, numCaracteres);

    }

  }

  // Ejemplo de uso:

  let texto = "Hola Mundo";

  let numCaracteres = 4;

  // Mostrar el texto original en la consola.

  console.log("Texto original: " + texto);

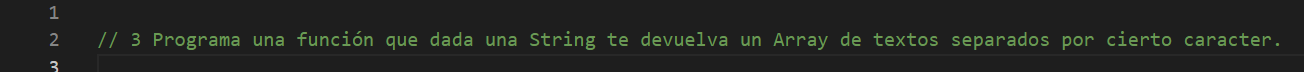
  // Mostrar el resultado en la consola.

  let resultado = miFuncion(texto, numCaracteres);

  console.log("Resultado: " + resultado);

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente



function miFuncion(cadena, caracter) {

    // Utilizamos el método split() para dividir la cadena en un array.

    // Utilizando el caracter especificado como separador.

    const arraySeparado = cadena.split(caracter);

    return arraySeparado;

  }

  // Ejemplo de uso:

  const texto = 'Hola ¿qué tal cómo estás?';

  const caracterSeparador = ' ';

  console.log("Texto original: " + texto);

  // Llamamos a la función miFuncion para separar el texto utilizando el espacio como separador.

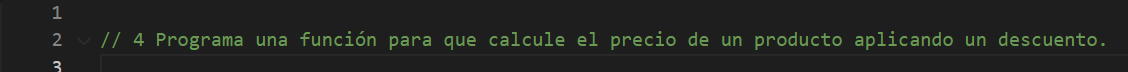
  const resultado = miFuncion(texto, caracterSeparador);

  // Mostrar el resultado separado en la consola.

  console.log(resultado);

Captura de pantalla de un celular

Descripción generada automáticamente



// Función ue calcula el precio con descuento.

function calcularPrecioConDescuento(precioOriginal, porcentajeDescuento) {

  // Verificar que el porcentaje de descuento sea válido.

  if (porcentajeDescuento < 0 || porcentajeDescuento > 100) {

    throw new Error("El porcentaje de descuento debe estar entre 0 y 100");

  }

  // Calcular el descuento.

  const descuento = (precioOriginal \* porcentajeDescuento) / 100;

  // Calcular el precio con el descuento aplicado.

  const precioConDescuento = precioOriginal - descuento;

  // Devolver el precio con descuento.

  return precioConDescuento;

}

// Ejemplo de uso:

const precioOriginal = 1800;

const porcentajeDescuento = 15;

const precioFinal = calcularPrecioConDescuento(precioOriginal, porcentajeDescuento);

//Mostrar el precio original y el porcentaje de descuento.

console.log("El precio sin descuento es: " + precioOriginal + ".")

console.log("El descuento que se aplica es de: " + porcentajeDescuento + "%.")

// Mostrar el precio con el descuento aplicado.

console.log("El precio con descuento es: " + precioFinal + ".");

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



// Función para calcular la edad a partir de una fecha de nacimiento.

function calcularEdad(fechaNacimiento) {

  const fechaActual = new Date();

  const añoNacimiento = fechaNacimiento.getFullYear(); // Obtenemos el año de nacimiento.

  const mesNacimiento = fechaNacimiento.getMonth(); // Obtenemos el mes de nacimiento.

  const diaNacimiento = fechaNacimiento.getDate(); // Obtenemos el día de nacimiento.

  const añoActual = fechaActual.getFullYear(); //Obtenemos el año actual.

  const mesActual = fechaActual.getMonth(); // Obtenemos el mes actual.

  const diaActual = fechaActual.getDate(); // Obtenemoe el día actual.

  //Calculamos la edad inicial.

  let edad = añoActual - añoNacimiento;

  // Ajustamos la edad si aún no ha cumplido años este año.

  if (mesActual < mesNacimiento || (mesActual === mesNacimiento && diaActual < diaNacimiento)) {

    edad--;

  }

  return [diaNacimiento, mesNacimiento, añoNacimiento, edad];

}

// Ejemplo de uso:

// Definimos la fecha de nacimiento: 1 de octubre de 1988.

const fechaNacimiento = new Date(1988, 9, 1);

// Calculamos la edad.

const [diaNacimiento, mesNacimiento, añoNacimiento, edad]= calcularEdad(fechaNacimiento);

// Mostramos la fecha de nacimiento.

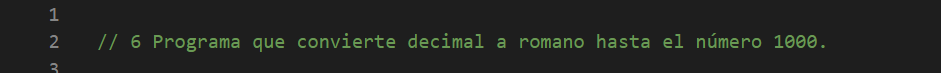
console.log("La fecha de nacimiento es: " + diaNacimiento + "/" + mesNacimiento + "/" + añoNacimiento + ".")

// Mostramos la edad en la consola.

console.log("La edad es: " + edad + " años.");

Texto

Descripción generada automáticamente



function decimalARomano(numero) {

  // Verificar si el número está fuera del rango permitido.

  if (numero < 1 || numero > 1000) {

    return "Número fuera de rango. Debe estar entre 1 y 1000.";

  }

  // Arreglos para los valores decimales y los símbolos romanos correspondientes.

  const valoresDecimales = [1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1];

  const simbolosRomanos = ["M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"];

  // Variable para almacenar la representación romana.

  let resultado = "";

  // Iterar a través de los valores decimales y construir la representación romana.

  for (let i = 0; i < valoresDecimales.length; i++) {

    while (numero >= valoresDecimales[i]) {

      resultado += simbolosRomanos[i]; // Agregar el símbolo romano al resultado.

      numero -= valoresDecimales[i]; // Restar el valor decimal del número.

    }

  }

  return resultado;

}

// Ejemplos de uso:

const numero = 95;

const conversionARomano = decimalARomano(numero);

// Mostrar el resultado en la consola.

console.log("El número " + numero + ", en romano es " + conversionARomano + ".")

Texto

Descripción generada automáticamente



// Función para convertir un número decimal en su representación binaria.

function decimalABinario(numero) {

    if (numero === 0) {

        return "0"; // Si el número es 0, su representación binaria también es 0.

    } else {

        let binario = "";

        while (numero > 0) {

            const residuo = numero % 2; // Calcula el residuo al dividir por 2.

            binario = residuo + binario; // Agrega el residuo al principio de la cadena binaria.

            numero = Math.floor(numero / 2); // Divide el número decimal por 2 y redondea hacia abajo.

        }

        return binario; // Retorna la representación binaria.

    }

}

// Ingresa el número decimal que deseas convertir.

const numeroDecimal = parseInt(prompt("Ingresa un número decimal:"));

// Mostrar el texto ingresado por el usuario.

console.log(`Número decimal ingresado: ${numeroDecimal}`);

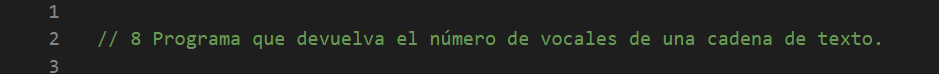
// Llama a la función para realizar la conversión e imprime el resultado.

const resultadoBinario = decimalABinario(numeroDecimal);

console.log(`El número binario equivalente es: ${resultadoBinario}`);

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



// Función para contar las vocales en una cadena de texto.

function contarVocales(cadena) {

    // Convertir la cadena a minúsculas para considerar las vocales en mayúsculas y minúsculas.

    cadena = cadena.toLowerCase();

    // Inicializar un contador para las vocales.

    let contador = 0;

    // Definir una cadena que contiene todas las vocales.

    const vocales = "aeiou";

    // Iterar a través de la cadena y contar las vocales.

    for (let i = 0; i < cadena.length; i++) {

      // Verificar si el caracter actual está incluido en la cadena de vocales.

      if (vocales.includes(cadena[i])) {

        // Si el caracter es una vocal, aumentar el contador.

        contador++;

      }

    }

    // Devolver el contador que contiene el número de vocales encontradas en la cadena de texto.

    return contador;

  }

  // Ejemplo de uso:

  const cadena = "Hola Mundo, Hoy Me Encuentro Bien.";

  // Mostrar el texto original.

  console.log("Texto original: " + cadena);

  // Llamar a la función.

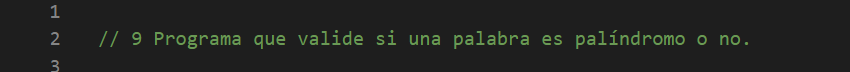
  const numVocales = contarVocales(cadena);

  // Mostrar el resultado en la consola.

  console.log("El número de vocales en la cadena es: " + numVocales + ".");

Texto

Descripción generada automáticamente



// Función para verificar si una palabra es un palíndromo.

function esPalindromo(palabra) {

  // Convierte la palabra a minúsculas para evitar problemas de mayúsculas y minúsculas.

  palabra = palabra.toLowerCase();

  // Elimina los espacios en blanco de la palabra.

  palabra = palabra.replace(/\s/g, "");

  // Comprueba si la palabra es igual a su inversa.

  return palabra === palabra.split("").reverse().join("");

}

// Ejemplo de uso:

// Solicita al usuario que ingrese una palabra.

const palabra = prompt("Ingrese una palabra:");

// Mostrar el texto ingresado por el usuario.

console.log(`Palabra ingresada: ${palabra}`);

// Llama a la función y muestra el resultado.

if (esPalindromo(palabra)) {

  console.log(`"${palabra}" es un palíndromo.`);

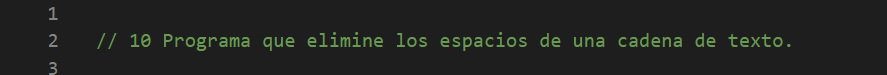
} else {

  console.log(`"${palabra}" no es un palíndromo.`);

}

Captura de pantalla de un celular

Descripción generada automáticamente



// Definimos una función llamada eliminarEspacios que toma una cadena de texto como entrada.

function eliminarEspacios(cadena) {

  // Utilizamos el método replace con una expresión regular para eliminar los espacios en blanco.

  // Con una cadena vacía ("") en toda la cadena de texto de entrada.

  return cadena.replace(/\s/g, "");

}

// Ejemplo de uso:

const textoConEspacios = "Hola Mundo, Hoy Es Un Día Lluvioso";

// Llamamos a la función eliminarEspacios con la cadena de texto.

const resultado = eliminarEspacios(textoConEspacios);

// Mostrar el texto original.

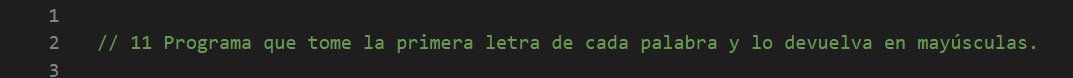
console.log("Texto original: " + textoConEspacios + ".");

// Mostramos el resultado en la consola.

console.log("Resultado: " + resultado + ".");

Texto

Descripción generada automáticamente



// Función que toma una frase y devuelve las iniciales en mayúsculas de cada palabra de la cadena de texto.

function obtenerInicialesMayusculas(frase) {

    // Dividir la frase en palabras.

    const palabras = frase.split(' ');

    // Inicializar una cadena vacía para almacenar las iniciales en mayúsculas.

    let iniciales = '';

    // Iterar a través de cada palabra y obtener la primera letra en mayúsculas.

    palabras.forEach(palabra => {

        // Verificar si la palabra no está vacía.

        if (palabra) {

            // Obtener la primera letra de la palabra en mayúsculas y agregarla a 'iniciales'.

            iniciales += palabra[0].toUpperCase();

        }

    });

    // Devolver las iniciales en mayúsculas.

    return iniciales;

}

// Ejemplo de uso:

const frase = "Brenda Samara Escobar Avila";

const resultado = obtenerInicialesMayusculas(frase);

// Mostrar el texto original en la consola.

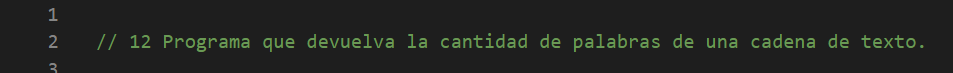
console.log("Texto original: " + frase + ".");

// Mostrar el resultado en la consola.

console.log("Resultado: " + resultado + ".");

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



// Definir una función para contar palabras en una cadena de texto.

function contarPalabras(cadena) {

    // Divide la cadena en palabras utilizando espacios como separadores.

    let palabras = cadena.split(/\s+/);

    // Filtra las palabras vacías.

    palabras = palabras.filter(function(palabra) {

        return palabra.trim() !== '';

    });

    // Devuelve la cantidad de palabras.

    return palabras.length;

}

// Solicita al usuario que ingrese una cadena de texto.

let texto = prompt("Ingrese una frase: ");

// Mostrar el texto ingresado por el usuario.

console.log("Texto ingresado: " + texto);

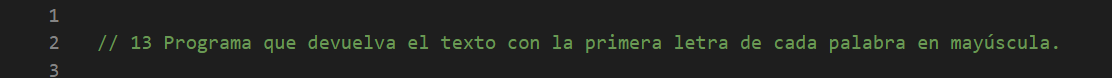
// Llama a la función contarPalabras y muestra el resultado en la consola.

var cantidadPalabras = contarPalabras(texto);

console.log("La cantidad de palabras es: " + cantidadPalabras);

Texto

Descripción generada automáticamente



// Función para capitalizar la primera letra de cada palabra en una cadena de texto.

function capitalizarPalabras(texto) {

    // Dividir el texto en palabras.

    let palabras = texto.split(" ");

    // Capitalizar la primera letra de cada palabra.

    for (var i = 0; i < palabras.length; i++) {

        palabras[i] = palabras[i].charAt(0).toUpperCase() + palabras[i].slice(1);

        // Obtener la primera letra de la palabra, capitalizarla y luego concatenar el resto de la palabra sin modificarla.

    }

    // Unir las palabras capitalizadas nuevamente en un solo texto.

    let textoCapitalizado = palabras.join(" ");

    return textoCapitalizado;

}

// Ejemplo de uso:

let texto = "hola mundo, hoy tengo ganas de hacer un rico pastel de chocolate";

let textoCapitalizado = capitalizarPalabras(texto);

// Mostrar el texto original en la consola.

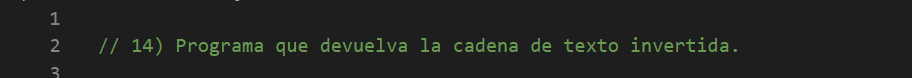
console.log("Texto original: " + texto + ".");

// Mostrar el resultado en la consola.

console.log("Resultado: " + textoCapitalizado + ".");

Texto

Descripción generada automáticamente



// Definir una función llamada invertirCadena.

function invertirCadena(cadena) {

  // Dividimos la cadena en un arreglo de caracteres.

  // Invertimos el arreglo.

  // Unimos la cadena nuevamente.

  return cadena.split('').reverse().join('');

}

// Ejemplo de uso:

let texto = "¡¡¡Hola Mundo!!!, ¿Quieres jugar una partida de ajedréz?";

// Llamamos a la función invertirCadena con el texto.

let textoInvertido = invertirCadena(texto);

// Mostrar el texto original en la consola.

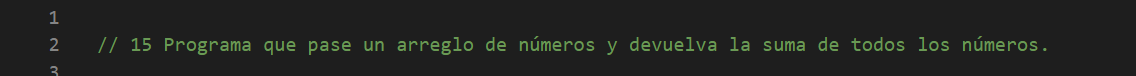
console.log("Texto original: " + texto);

// Mostrar el resultado en la consola.

console.log(textoInvertido);

Captura de pantalla de un celular

Descripción generada automáticamente



// Definimos una función llamada suma que toma un número variable de argumento.

function suma(...numeros) {

  // Utilizamos el método "reduce" para sumar todos los números en el arreglo.

  // El segundo argumento de "reduce" (0) es el valor inicial de la suma.

  const total = numeros.reduce((sum, num) => sum + num, 0);

  // Creamos una cadena de texto que muestra el resultado.

  const resultado = `La suma de todos los números es ${total}.`;

  // Devolvemos el resultado.

  return resultado;

}

// Ejemplo de uso:

const numeros = [15, 27, 38, 90, 65];

const resultado = suma(...numeros);

// Mostrar el texto original en la consola.

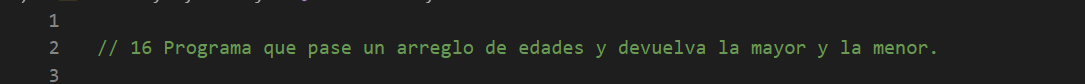
console.log("El arreglo de números es: " + numeros + ".")

// Mostrar el resultado en la consola.

console.log(resultado);

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente



// Definir una función que encuentra la edad mayor y la edad menor.

function encontrarMayorMenorEdad(...edades) {

  // Verificar si no se proporcionaron edades.

  if (edades.length === 0) {

    return "No hay edades para comparar.";

  }

  // Inicializar las edades mayor y menor con el primer elemento del arreglo.

  let mayor = menor = edades[0];

  // Iterar a través de las edades restantes para encontrar la mayor y la menor edad.

  for (let i = 1; i < edades.length; i++) {

    // Comprobar si la edad actual es mayor que la edad mayor actual.

    if (edades[i] > mayor) {

      mayor = edades[i];

      // Comprobar si la edad actual es menor que la edad menor actual.

    } else if (edades[i] < menor) {

      menor = edades[i];

    }

  }

  // Devolver el resutlado en forma de cadena de texto.

  return `La edad mayor es ${mayor} y la edad menor es ${menor}`;

}

// Ejemplo de uso:

const edades = [5, 10, 20, 25, 30, 40, 45, 50, 60, 65, 70];

const resultado = encontrarMayorMenorEdad(...edades);

// Mostrar el arreglo en la consola.

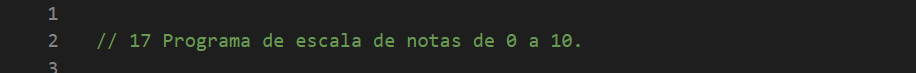
console.log("El arreglo es: " + edades + ".");

// Mostrar el resultado en la consola.

console.log(resultado);

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



// Función para obtener la calificación según la nota.

function obtenerCalificacion(nota) {

    if (nota < 0 || nota > 10) {

        return "Nota fuera de rango";

    } else if (nota >= 0 && nota < 6) {

        return "Mal estudiante";

    } else if (nota >= 6 && nota < 8) {

        return "Estudiante regular";

    } else if (nota >= 8 && nota < 10) {

        return "Buen estudiante";

    } else {

        return "Excelente estudiante";

    }

}

// Solicita al usuario que ingrese una nota.

let nota = parseFloat(prompt("Ingrese la nota (0-10): "));

// Verificar si la entrada del usuario es un número válido.

if (!isNaN(nota)) {

    // Llamar a la función obtenerCalificacion para obtener la calificación.

    let resultado = obtenerCalificacion(nota);

    // Mostrar la nota ingresada por el usuario.

    console.log("La nota ingresada es: " + nota + ".");

    // Mostrar el resultado en la consola.

    console.log(`La calificación es: ${resultado}`);

} else {

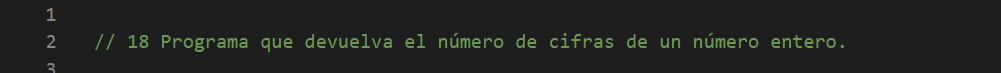
    // Mostrar un mensaje de error si la entrada de la nota no es válida.

    console.log("Ingrese una nota válida (número entre 0 y 10)");

}

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente



// Función para contar las cifras de un número entero.

function contarCifras(numero) {

  // Convierte el número a una cadena para facilitar el conteo de cifras.

  const numeroStr = numero.toString();

  // Usa la propiedad length para contar las cifras en la cadena.

  const cifras = numeroStr.length;

  return cifras;

}

// Ingresa un número entero.

const numero = parseInt(prompt("Ingresa un número entero: "));

// Llama a la función para contar las cifras y muestra el resultado.

const resultado = contarCifras(numero);

// Mostrar el entero ingresado por el usuario.

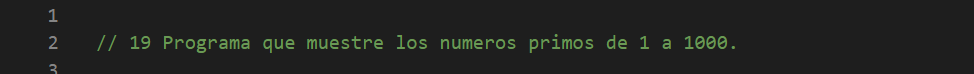
console.log("El número entero ingresado es: " + numero + ".");

// Mostrar el resultado en la consola

console.log(`${numero} tiene ${resultado} cifras.`);

Texto

Descripción generada automáticamente



// Función para verificar si un número es primo.

function esPrimo(numero) {

  if (numero <= 1) { // Si el número es menor o igual a 1, no es primo.

    return false;

  }

  if (numero <= 3) { // 2 y 3 son primos.

    return true;

  }

  if (numero % 2 === 0 || numero % 3 === 0) { // Si es divisible por 2 o 3, no es primo.

    return false;

  }

  let i = 5;

  while (i \* i <= numero) { // Verificamos divisibilidad hasta la raíz cuadrada del número.

    if (numero % i === 0 || numero % (i + 2) === 0) { // Si es dividible, no es primo.

      return false;

    }

    i += 6; // Optimización: saltamos de 6 en 6 para reducir las verificaciones innecesarias.

  }

  return true; // Si no se encontraron divisores, es primo.

}

// Iteramos a través de los números del 1 al 1000 y mostramos los números primos.

for (let numero = 1; numero <= 1000; numero++) {

  if (esPrimo(numero)) {

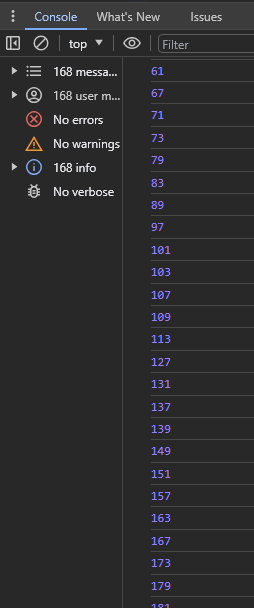
    // Mostramos el número primo en la consola.

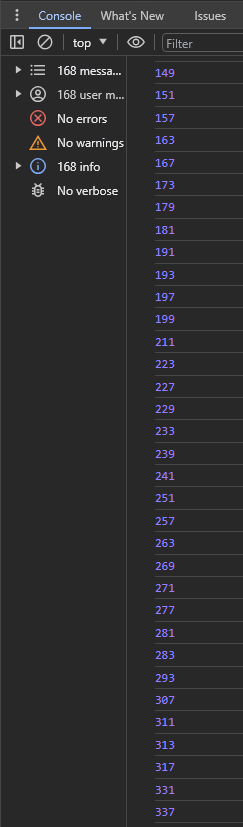
    console.log(numero);

  }

}

Imagen que contiene Gráfico

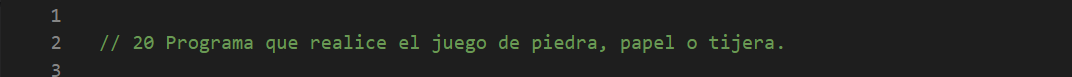
Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media



function juegoPiedraPapelTijera() {

  const opciones = ["Piedra", "Papel", "Tijera"];

  while (true) {

    // El jugador elige una opción.

    const jugador = prompt("Elige Piedra, Papel o Tijera (o pulsa Cancelar para salir): ");

    if (jugador === null) {

      alert("¡Adiós!");

      break;

    }

    // Asegurarse de que la entrada del jugador esté en formato capitalizado.

    const jugadorCapitalizado = jugador.charAt(0).toUpperCase() + jugador.slice(1).toLowerCase();

    // Verificar si la opción del jugador es válida.

    if (!opciones.includes(jugadorCapitalizado)) {

      alert("Opción no válida. Por favor, elige Piedra, Papel o Tijera.");

      continue;

    }

    // La computadora elige una opción al azar.

    const computadora = opciones[Math.floor(Math.random() \* opciones.length)];

    // Determina el ganador.

    if (jugadorCapitalizado === computadora) {

      alert(`Tú elegiste ${jugadorCapitalizado}\nLa computadora eligió ${computadora}\n¡Es un empate!`);

    } else if (

      (jugadorCapitalizado === "Piedra" && computadora === "Tijera") ||

      (jugadorCapitalizado === "Tijera" && computadora === "Papel") ||

      (jugadorCapitalizado === "Papel" && computadora === "Piedra")

    ) {

      alert(`Tú elegiste ${jugadorCapitalizado}\nLa computadora eligió ${computadora}\n¡Ganaste!`);

    } else {

      alert(`Tú elegiste ${jugadorCapitalizado}\nLa computadora eligió ${computadora}\nLa computadora gana.`);

    }

  }

}

// Inicia el juego.

juegoPiedraPapelTijera();

Interfaz de usuario gráfica, Aplicación

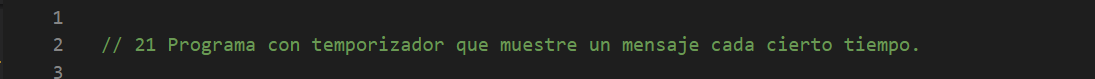
Descripción generada automáticamenteTexto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente con confianza baja



// Función que muestra el mensaje.

function mostrarMensaje() {

  console.log("Hola Mundo");

  // Agrega un salto de linea después de cada mensaje.

  console.log("\n");

}

// Intervalo en milisegundos (en este caso, 3000 ms que equivalen a 3 segundos).

const intervalo = 3000;

// Configurar un temporizador que llame a la función mostrarMensaje cada intervalo de tiempo

const temporizador = setInterval(mostrarMensaje, intervalo);

// Definir una variable para llevar el registro del número de mensajes mostrados.

let contadorMensajes = 0;

// Puedes detener el temporizador después de un cierto número de repeticiones o cuando sea necesario:

// clearInterval(temporizador);

Interfaz de usuario gráfica

Descripción generada automáticamente

**ENLACE DE GITHUB**