

Nombre \_\_\_\_\_ Apellidos \_\_\_\_\_

**1) EJERCICIO: objetos definidos por el usuario: (2,5P)**

Necesitamos almacenar en un programa todos los discos de música que tenemos en casa. Ahora que sabemos crear nuestros propios objetos es el mejor modo de guardar esta información.

Crea un objeto “**Disco**” que almacene la siguiente información: (1P)

- Nombre del disco. (propiedad **nombreDisco**)
- Grupo de música o cantante.(propiedad **grupoCantante**)
- Año de publicación. (propiedad **anioPublicacion**)
- Tipo de música (podrá ser “rock”, “pop”, “punk” o “indie”); (propiedad **tipoMusica**)
- Localización: almacenará un número de estantería. ( propiedad **localización**)
- Prestado: almacenará un valor booleano. Por defecto será false. ( propiedad **prestado**)

Además tendrá los siguientes **métodos**:

- Construir el objeto con parámetros (se pasarán solo las cinco primeras propiedades; la propiedad **prestado** será false). (0,25P)
- Un método que permitirá cambiar el número de estantería en la localización. (0,25P)
  - **cambiarDeEstanteria(nuevaPosicion)**
- Un método que permitirá cambiar la propiedad Prestado. (0,12P)
  - **cambiarEstadoPrestado(prestado)**
- Un método que muestre toda la información de un disco. (0,12P)
  - **muestraInfoDisco()** – mostrará la información de las propiedades del disco devolviendo una cadena
- Crear los métodos setters y getters para las propiedades del disco
- Guarda todo el código anterior en un archivo llamado **disco.js**
- Realiza un ejemplo donde: (guardar código en **pruebaDisco.js**)
  - Crees 3 discos de ejemplo. ( crear un array de discos llamado **misDiscos** ) (0,2P)
  - Mostrar la información del primer disco (0,2P)
  - Cambiar de estantería el primer disco (0,2P)
  - Cambiar el estado de prestado del segundo disco (0,2P)
  - Mostrar la información de todos los discos del array **misDiscos** (0,2P)

## 2) EJERCICIO: math (objetos predefinidos javascript) (2P)

Crea un programa que pida al usuario un número sobre el que realizar operaciones matemáticas junto con la operación a realizar en un formulario. Crear un **formulario** con un campo input con **id="numeroUsuario"** donde el usuario introducirá el número para realizar las operaciones matemáticas de potencia, raíz, redondeo o trigonometría según elija que tipo de operación realizar (creará un select con estas opciones). Crear un **botón llamado con id="btnOperacionMatematica"** que al hacer click sobre él, realizará la operación matemática seleccionada sobre dicho número introducido por el usuario.

Crear un div con **id="mensajePantalla"** donde se mostrarán el resultado de la operación matemática seleccionada por el usuario. (div donde se mostrarán los resultados calculados)

Crear un select (html) con las distintas opciones que puede elegir el usuario. (**utilizar el selectedIndex de dicho select**)

1. Potencia (elevado siempre a 3, es decir, calcular la potencia de 3)
2. Raíz cuadrada.
3. Redondeo.
4. Trigonometría.

Si el usuario selecciona la **opción Potencia** del select, el usuario debe introducir en el campo de texto con **id="numeroUsuario"** el valor sobre el que se calcula la potencia (valor de X), y se mostrará el resultado en pantalla (La potencia de X elevado a 3 es: ) (**0,5P**)

- Crear la función **opcionPotencia(datoCalcular)** que devuelve el cálculo

Si el usuario introduce la **opción Raíz** del select, se le mostrará el resultado en pantalla (La raíz de X es: ) (**0,5P**)

- Crear la función **opcionRaiz(datoCalcular)** que devuelve el cálculo

Si el usuario introduce la **opción Redondeo** del select, el usuario debe introducir un número decimal en el campo de texto con **id="numeroUsuario"** el valor sobre el que se calcula la potencia (valor de X), y se mostrará el redondeo al : (**mostrar estas tres opciones**) (**0,5P**)

- a) entero más próximo,
  - b) entero más alto (al alta)
  - c) entero a la baja.
- Crear la función **opcionRedonde(datoCalcular)** que devuelve un array con el valor de los cálculos

Si el usuario introduce **opción Trigonometría** del select, el usuario deberá introducir un ángulo (entre 0 y 360) y se le mostrarán por pantalla los valores trigonométricos del seno, coseno y tangente. (mostrar estas tres cálculos) (**0,5P**)

- Crear la función **opcionTrigonometria(datoCalcular)** que devuelve un array con los tres valores calculados de seno, coseno y tangente.

**Guardar dicho ejercicio en el archivo "calculosMatematicos.js"**

### 3) EJERCICIO: funciones y obj. Predefinidos (2,5P)

Crea un programa que pida al usuario el valor del radio y muestre por pantalla:

Crear un formulario para introducir el valor del radio (input **con id="radio"**) y un botón con **id="btnCalcula"**. Al hacer click en dicho botón, cogeremos los datos de dicho radio y realizaremos las operaciones matemáticas pedidas referentes a dicho radio, para el cálculo del diámetro, perímetro, área del círculo, área de la esfera, volumen de la esfera. Crear el manejador de eventos de dicho botón. **(0,5P – manejador eventos)**

Crear las siguientes funciones: ( devolver el resultado con 2 decimales )

- **perimetroCircunferencia(radio)** – Devuelve el perímetro de la circunferencia **(0,5P)**
  - Fórmula:  $2 \times \text{PI} \times \text{radio}$
- **areaCirculo(radio)** – Devuelve el área del círculo **(0,5P)**
  - Fórmula:  $\text{PI} \times \text{radio}^2$
- **areaEsfera(radio)** – Devuelve el área de la esfera **(0,5P)**
  - Formula:  $4 \times \text{PI} \times \text{radio}^2$
- **volumenEsferaa(radio)** – Devuelve el volumen de la esfera **(0,5P)**
  - Formula:  $\frac{4}{3} \times \text{PI} \times \text{radio}^3$

Crear un elemento div con el id="infoCalculos" donde mostraremos la siguiente información:

1. El valor del radio.
  2. El valor del diámetro.
  3. El valor del perímetro de la circunferencia.
  4. El valor del área del círculo.
  5. El valor del área de la esfera.
  6. El valor del volumen de la esfera.
- El valor de Pi debes obtenerlo del objeto Math, no introducirlo manualmente.
  - Debes escribir al lado si son cm, o cm2, o cm3.
  - Como datos de muestra, si metes 5, deberías obtener aproximadamente: 5 / 10 / 31,41 / 78,54 / 314,15 / 523,59. (Probar con estos valores y mostrar los resultados obtenidos)

Crear un función que devuelva la cadena de dicha información calculada llamada **muestraResultados()**. Los resultados se mostrarán en el div con id="infoCalculos"

**Guardar dicho ejercicio en el archivo "calculoCirculo.js"**

#### 4) EJERCICIO: arrays (3P):

Trabajaremos con la información de diferentes países.

Ejemplo: Alemania, Finlandia, Irlanda, Reino Unido, Francia, Italia, Portugal, Checoslovaquia.

Crear un archivo **datosPaíses.js** donde incluyas un array llamado `arrPaíses` con los datos de los países anteriores. **(0,6P)**

Vamos a gestionar una lista de países haciendo uso de Arrays.

Para ello necesitarás crear un archivo **arrays.js** que incluya las siguientes funciones:

- Mostrar el número de elementos del array. **(0,3P)**
  - `numeroPaíses(arrDatos)`
- Mostrar todos los elementos del array. **(0,3P)**
  - `mostrarPaíses(arrDatos)`
- Muestra los elementos del array en sentido inverso. **(0,3P)**
  - `paísesInvertidos(arrPaíses)`
- Muestra los elementos del array ordenados alfabéticamente (pero no los ordena). **(0,3P)**
  - `paísesOrdenados(arrDatos)`
- Añadir un elemento al principio del array. **(0,3P)**
  - `anadeInicio(arrDatos)`
- Añadir un elemento al final del array. **(0,3P)**
  - `anadeFinal(arrDatos)`
- Borrar un elemento al principio del array (y devolver cuál se ha borrado). **(0,3P)**
  - `borraPrincipio(arrDatos)`
- Borrar un elemento al final del array (y devolver cuál se ha borrado). **(0,3P)**
  - `borraFinal(arrDatos)`