

La **estructura switch** es una alternativa a la estructura **if...else** y se utiliza para evaluar una expresión y ejecutar un bloque de código **dependiendo del valor de esa expresión**.

Aquí están los pasos para usar switch en JavaScript:

Iniciar la estructura switch

La sintaxis básica para iniciar una estructura switch es la siguiente:

La expresión es aquello que se va a evaluar. Normalmente, es una variable o una expresión que se resuelve a un valor único.

```
switch(expresión) {  
    // casos aquí  
}
```

Definir los casos

Dentro de la estructura switch, **debes definir los diferentes casos** que se evaluarán. Cada caso se define con la palabra clave **case** seguida del valor que se desea evaluar y dos puntos (:). Después de los dos puntos, se coloca el código que se ejecutará si la expresión coincide con el valor del case.

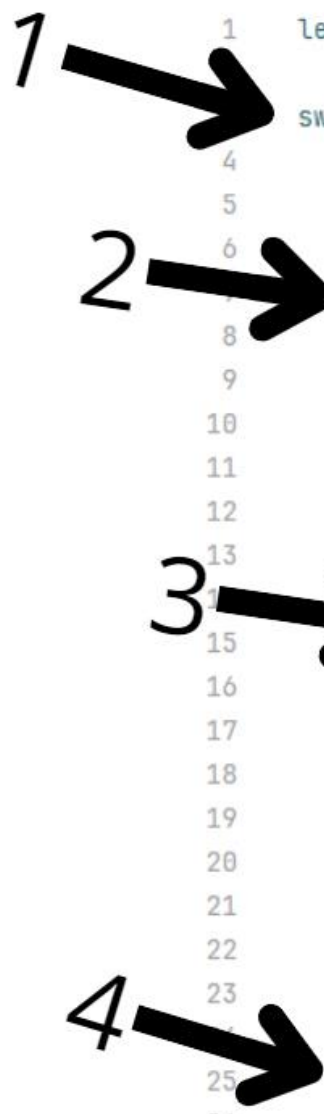
```
switch(expresión) {  
    case valor1:  
        // código a ejecutar si la expresión es valor1  
        break;  
    case valor2:  
        // código a ejecutar si la expresión es valor2  
        break;  
    // más casos aquí...  
    default:  
        // código a ejecutar si ninguno de los casos coincide  
}
```

Usar la sentencia `break`

La sentencia `break` es opcional, pero muy recomendable. Indica al intérprete que salga de la estructura `switch` una vez que se ha ejecutado el código correspondiente a un caso. Si no se incluye `break`, el intérprete continuará ejecutando el código de los casos siguientes, lo cual rara vez es el comportamiento deseado.

Caso `default`

El caso `default` es opcional, pero también se recomienda incluirlo. Se ejecutará si ninguno de los casos coincide con el valor de la expresión. Se coloca al final de la estructura `switch`.



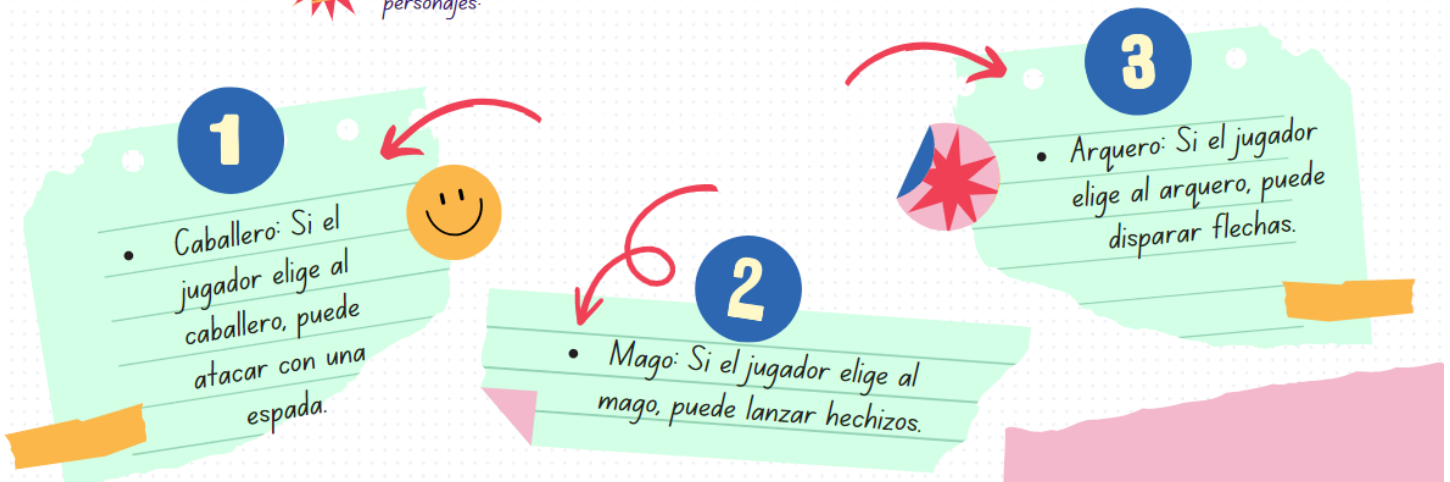
```
1  let dia = 3;
2
3  switch (dia) {
4      case 1:
5          console.log("Lunes");
6          break;
7      case 2:
8          console.log("Martes");
9          break;
10     case 3:
11         console.log("Miércoles");
12         break;
13     case 4:
14         console.log("Jueves");
15         break;
16     case 5:
17         console.log("Viernes");
18         break;
19     case 6:
20         console.log("Sábado");
21         break;
22     case 7:
23         console.log("Domingo");
24         break;
25     default:
26         console.log("Día inválido");
27 }
```

The diagram illustrates the execution flow of the provided JavaScript code. Four large black arrows, numbered 1 through 4, point to specific lines in the code:

- Arrow 1 points to line 1: `let dia = 3;`
- Arrow 2 points to line 4: `case 1:`
- Arrow 3 points to line 13: `case 4:`
- Arrow 4 points to line 25: `default:`



Imagina que tienes un juego donde el jugador puede elegir entre tres personajes:



La variable personaje guarda el valor elegido por el jugador

```
1 let personaje: string = "Caballero";
2
3 switch (personaje) {
4   case "Caballero":
5     console.log("El caballero ataca con su espada!");
6     break;
7   case "Mago":
8     console.log("El mago lanza un hechizo!");
9     break;
10  case "Arquero":
11    console.log("El arquero dispara una flecha!");
12    break;
13  default:
14    console.log("No has elegido un personaje válido.");
15 }
```

La estructura switch evalúa el valor de la variable personaje.

Si el valor coincide con un "caso", se ejecuta la instrucción dentro de ese caso.

La palabra clave break se usa para salir de la estructura switch.

El caso default se ejecuta si el valor de la variable no coincide con ningún caso anterior.

variable Booleana.)

```
1 let doorSensor: boolean = true;
2
3 let doorState: string = 'cerrada';
4
5 switch(doorSensor) {
6   case true:
7     if(doorState === 'cerrada') {
8       console.log('Abriendo puerta');
9       doorState = 'abierta';
10    } else {
11      console.log('La puerta ya está abierta');
12    }
13    break;
14   case false:
15     if(doorState === 'abierta') {
16       console.log('Cerrando puerta');
17       doorState = 'cerrada';
18     } else {
19       console.log('La puerta ya está cerrada');
20     }
21    }
22    break;
23 }
```

Estado de la puerta (abierta/cerrada)

Si el sensor detecta presencia

Y la puerta está cerrada

Se abre la puerta

Si ya estaba abierta, no hacemos nada

Si el sensor NO detecta presencia

```

1  let day : number = new Date().getDay();
2
3  switch(day) {
4      case 0:
5          console.log("Domingo");
6          break;
7      case 1:
8          console.log("Lunes");
9          break;
10     case 2:
11         console.log("Martes");
12         break;
13     case 3:
14         console.log("Miércoles");
15         break;
16     case 4:
17         console.log("Jueves");
18         break;
19     case 5:
20         console.log("Viernes");
21         break;
22     case 6:
23         console.log("Sábado");
24         break;
25     default:
26         console.log("Día inválido");
27 }

```

Se obtiene el día actual con `new Date().getDay()`

Se compara ese valor en el switch

Según el caso, se imprime el nombre del día correspondiente

El caso default se ejecuta si no coincide ningún caso

```

1  const opciones : string = `
2  1. Comenzar programa
3  2. Configuración
4  3. Salir
5  `
6
7  console.log(opciones);
8
9  const opcionElegida : number = parseInt(prompt(message: 'Elige una opción (1-3)'));
10
11 switch(opcionElegida) {
12     case 1:
13         console.log('Iniciando programa...');
14         break;
15
16     case 2:
17         console.log('Abriendo configuración...');
18         break;
19
20     case 3:
21         console.log('Saliendo...');
22         break;
23
24     default:
25         console.log('Opción inválida. Debe ingresar un número entre 1 y 3.');
```

Opciones del menú

Imprimimos las opciones

Entrada del usuario

Switch para manejar la opción

Pedimos de nuevo al usuario que ingrese opción

Funciones o acciones para buscar un número aleatorio

```

1  const dado1 : number = Math.floor(x: Math.random() * 6) + 1;
2  const dado2 : number = Math.floor(x: Math.random() * 6) + 1;
3
4  console.log(`Dado 1: ${dado1}`);
5  console.log(`Dado 2: ${dado2}`);
6
7  const suma : number = dado1 + dado2;
8
9  switch(suma) {
10     case 2: console.log('Serpiente ojos 🐍'); break
11     case 3: console.log('Tres en raya!'); break;
12     case 4: console.log('Mar - al aire libre 🌊'); break;
13     case 5: console.log('Estudiantes 2024 - casi llegas!'); break;
14     case 6: console.log('Seis - perfecto!'); break;
15     case 7: console.log('Siete ascendente 📈'); break;
16     case 8: console.log('Ocho de frente 🎲'); break;
17     case 9: console.log('Nueve es mío 🍷'); break;
18     case 10: console.log('Doble cinco 🎲🎲'); break;
19     case 11: console.log('Seis cinco, más alto que nadie 🏎️'); break;
20     case 12: console.log('Doble seis - suerte suprema! 🎲🎲'); break;
21 }

```

Imprimimos valor de dados

sumamos dados

Switch para manejar la opción

RECORRIDO DE ESCRITORIO!

Línea	Variable	Valor antes	Valor después	Observación
1	day	-	2 (asumiendo hoy es martes)	Se obtiene el día de la semana (2 para martes).
2	switch(day)	-	-	Evalúa el valor de day.
3	case 0	-	-	No se ejecuta porque day es 2 y no 0.
4	case 1	-	-	No se ejecuta porque day es 2 y no 1.
5	case 2	-	-	Se ejecuta porque day es 2.
6	console.log("Martes")	-	-	Imprime "Martes".
7	break	-	-	Sale del switch.
8-12	case 3 hasta default	-	-	No se ejecutan porque ya se encontró un caso coincidente.

RECORRIDO DE ESCRITORIO!



Paso	Variable	Valor antes	Valor después	Observación
1	<code>doorSensor</code>	-	<code>true</code>	Se asume que el sensor está activado.
2	<code>doorState</code>	-	<code>'cerrada'</code>	Se asume que la puerta está cerrada.
3	<code>switch(doorSensor)</code>	-	-	Se evalúa el estado del sensor.
4	<code>case true</code>	-	-	Se ejecuta el caso <code>true</code> porque el sensor está activado.
5	<code>if(doorState === 'cerrada')</code>	-	-	Se evalúa el estado de la puerta.
6	<code>console.log('Abriendo puerta')</code>	-	-	Se imprime el mensaje "Abriendo puerta".
7	<code>doorState</code>	<code>'cerrada'</code>	<code>'abierta'</code>	Se cambia el estado de la puerta a "abierta".
8	<code>break</code>	-	-	Se sale del <code>switch</code> .
9-13	<code>case false</code> hasta <code>break</code>	-	-	No se ejecutan porque el sensor está activado.

RECORRIDO DE ESCRITORIO!

Paso	Variable	Valor antes	Valor después	Observación
1	<code>dado1</code>	-	Número aleatorio entre 1 y 6	Se genera un número aleatorio para el primer dado.
2	<code>dado2</code>	-	Número aleatorio entre 1 y 6	Se genera un número aleatorio para el segundo dado.
3	<code>console.log(...)</code>	-	-	Se muestran los resultados de los dados.
4	<code>suma</code>	-	<code>dado1 + dado2</code>	Se calcula la suma de los dados.
5	<code>switch(suma)</code>	-	-	Se evalúa la suma de los dados.



Ejercicio 1:

Enunciado: Se requiere desarrollar un sistema que permita a los clientes de una lavandería seleccionar y cotizar diferentes servicios de lavado de ropa.

Requisitos:

- El sistema debe presentar al usuario las siguientes opciones de servicios:
 - Lavado básico
 - Lavado premium (incluye secado y planchado)
- Cada tipo de servicio tiene un precio diferente por pieza de ropa:
 - Lavado básico: \$2000
 - Lavado premium: \$4000
- El usuario debe seleccionar el tipo de servicio deseado.
- Basado en la selección del usuario, el sistema debe calcular un estimado del costo total del servicio.
- El usuario ingresa la cantidad de piezas de ropa para lavar.
- El sistema presenta el costo total como resultado.

Reglas de negocio:

- Solo se permite seleccionar una opción de servicio por transacción
- La cantidad mínima de piezas por transacción es de 1
- La cantidad máxima de piezas por transacción es 20
- Si el usuario selecciona una opción inválida, se debe mostrar un mensaje de error

Ejercicio 2:

Enunciado: Se requiere un sistema que permita a los estudiantes pagar su matrícula en la universidad, seleccionando formas de pago y visualizando el valor total.

Requisitos:

- El sistema debe presentar las siguientes opciones de pago:
 - Pago completo (5% de descuento)
 - Pago en 2 cuotas
 - Pago en 3 cuotas
- El valor base de la matrícula es de \$7.000.000
- El estudiante debe seleccionar la opción de pago deseada.

- Según la opción elegida, el sistema debe:
 - Si es pago completo, aplicar 5% de descuento
 - Si es en 2 cuotas, dividir el pago en partes iguales
 - Si es en 3 cuotas, dividir el pago en partes iguales
- Mostrar al estudiante el valor total a pagar según la opción elegida.

Reglas de negocio:

- Solo permite seleccionar una opción de pago por transacción
- El número máximo de cuotas es 3
- Si se elige una opción inválida, mostrar un mensaje de error
- Los pagos en cuotas tienen un interés del 2% mensual sobre el saldo.

Ejercicio 3:

Enunciado: Se requiere un sistema que permita gestionar la venta de zapatos deportivos en una tienda, seleccionando modelos, tallas y calculando precios.

Requisitos:

- El sistema debe permitir seleccionar entre los siguientes modelos de tenis:
 - Clásicos
 - Running
 - Basketball
- Para cada modelo hay tallas desde 35 a 44
- Los precios base por modelo son:
 - Clásicos: \$500.000
 - Running: \$800.000
 - Basketball: \$1.000.000
- El usuario debe seleccionar: modelo de tenis, talla y cantidad a comprar
- El sistema debe calcular el total a pagar incluyendo descuentos

Reglas de negocio:

- Existe un 10% de descuento para compras de 3 pares o más
- No se puede seleccionar una talla fuera del rango permitido
- La cantidad mínima que se puede comprar es de 1 par
- La cantidad máxima es de 5 pares por transacción
- Si se ingresa una opción inválida se debe mostrar un mensaje al usuario

Ejercicio 4:

Enunciado: Se requiere un sistema para una fábrica de escobas que permita seleccionar modelos, materiales y opciones para cotizar el producto según las especificaciones del cliente.

Requisitos:

- El usuario debe poder seleccionar el modelo de escoba entre:
 - Escoba dura
 - Escoba suave
 - cepillo
- Para la fabricación se utilizan dos tipos de materiales:
 - Cerdas sintéticas
 - Cerdas naturales
- El usuario puede escoger opciones adicionales como mango largo o gancho en la punta
- El sistema debe calcular el precio de cada configuración según las siguientes tarifas:
 - Escoba dura sintética: \$8.000
 - Escoba dura natural: \$10.000
 - cepillo sintético: \$12.000
 - cepillo natural: \$15.000
 - Mango largo: \$2.000 adicionales
 - Gancho en punta: \$500 adicionales
- El usuario indica la cantidad de escobas a cotizar.

Reglas de negocio:

- Solo se permite un tipo de material por producto
- No se pueden combinar mango largo y gancho en punta
- La cantidad mínima de fabricación es 3 escobas, la máxima 30
- En caso de selección inválida mostrar mensaje de error.

Ejercicio 5:

Enunciado: Se requiere un sistema que permita gestionar los préstamos de libros en una biblioteca pública. Los usuarios podrán seleccionar libros a solicitar y el sistema calculará las fechas y condiciones de préstamo.

Requisitos:

- El sistema clasifica los libros en tres categorías:
 - Bestsellers (novedades)

- Literatura
- Académicos
- Cada categoría tiene una tarifa de préstamo por día diferente:
 - Bestsellers: \$500 por día
 - Literatura: \$100 por día
 - Académicos: préstamo gratuito
- El usuario selecciona los libros a solicitar, ingresando título y categoría
- El usuario indica la cantidad de días que necesita el préstamo
- El sistema calcula el costo del préstamo basado en:
 - Categoría del libro
 - Número de días solicitado
- Para préstamos mayores a 10 días se aplica 10% de descuento

Reglas de negocio:

- El préstamo puede ser de 1 a 30 días
- Se pueden solicitar máximo 5 libros por préstamo
- Académicos siempre son gratuitos independiente de los días

Ejercicio 6:

Enunciado: Se requiere un sistema para un servicio de entrega a domicilio que permita a los clientes pedir comida de diferentes restaurantes, seleccionar opciones y calcular el valor total del pedido.

Requisitos:

- El usuario puede elegir restaurantes asociados como:
 - Hamburguesas
 - Pizzería
 - Comida China
- Cada restaurante tiene varias opciones de menú con precios establecidos. Por ejemplo:
 - Hamburguesa simple (\$25000)
 - Pizza familia (\$85000)
 - Fideos con pollo (\$18000)
- El usuario selecciona el restaurante, el menú deseado y la cantidad para cada opción.
- El sistema calcula el subtotal por ítems y el total del pedido incluyendo:
 - Costo de los artículos

- Costo de envío (\$5000)
- Se muestra al usuario el resumen del pedido y valor total a pagar.

Reglas de negocio:

- Solo se puede pedir de un restaurant a la vez.
- Mínimo de compra \$23000 por pedido.
- Tiempo de delivery 45 minutos a 1 hora.
- El pago solo puede ser en efectivo o con tarjeta débito/crédito.

Ejercicio 7:

Enunciado: Se requiere un sistema de venta de boletas para un concierto que permita seleccionar la ubicación en el escenario y calcular los costos de acuerdo con la localidad.

Requisitos:

- El escenario tiene las siguientes localidades:
 - General
 - Preferencial
 - VIP
- Los precios para cada localidad son:
 - General: \$50000
 - Preferencial: \$150000
 - VIP: \$300000
- El usuario selecciona:
 - Cantidad de boletas
 - Localidad deseada
- El sistema calcula el total a pagar incluyendo adicionales:
 - Impuestos del 10% sobre el precio
 - Cargo por servicio por boleta de \$5000
- El usuario visualiza el desglose de cobros y total a pagar.

Reglas de negocio:

- Las localidades tienen capacidad limitada de entradas disponibles
- Los descuentos y promociones no aplican para la localidad VIP
- Solo se vende un máximo de 5 boletas por transacción
- La compra puede ser pagada en efectivo o con tarjeta de crédito

Ejercicio 8:

Enunciado: Se requiere un sistema para que los usuarios puedan apostar en carreras de carros, seleccionando pilotos, montos y tipos de apuestas.

Requisitos:

- El usuario debe poder seleccionar carreras disponibles con sus respectivos pilotos y cuotas. Por ejemplo:
 - Carrera 1: (Piloto A (1.5), Piloto B (2.3), Piloto C(1.8))
- El usuario puede hacer dos tipos de apuestas:
 - Ganador de la carrera
 - Posiciones exactas (seleccionar 1ro y 2do puesto)
- El sistema debe permitir ingresar el monto a apostar y registrar la apuesta.
- Una vez terminada la carrera, el sistema determina si el usuario ganó y calcula las ganancias según las cuotas.

Reglas de negocio:

- El usuario solo puede apostar en carreras habilitadas por el sistema.
- El monto mínimo por apuesta es \$10000
- El monto máximo por evento es \$1000.000
- Solo mayores de edad pueden apostar según su número de identificación.

Solución ejercicio 1

```
1  const PRECIO_BASICO = 2000;
2  const PRECIO_PREMIUM = 4000;
3
4  const OPCION_BASICO = 1;
5  const OPCION_PREMIUM = 2;
6
7  let costoUnitario;
8  let piezas;
9
10 let opcionSeleccionada = parseInt(prompt("Seleccione tipo de servicio: \n1. Básico ($2000 c/u) \n2. Premium ($4000 c/u)"));
11
12
13 if(opcionSeleccionada === OPCION_BASICO) {
14     costoUnitario = PRECIO_BASICO;
15     piezas = parseInt(prompt("Ingrese cantidad de piezas (1-20):"));
16 } else if(opcionSeleccionada === OPCION_PREMIUM) {
17     costoUnitario = PRECIO_PREMIUM;
18     piezas = parseInt(prompt("Ingrese cantidad de piezas (1-20):"));
19     alert(piezas)
20 } else {
21     console.log("Opción inválida");
22 }
```

```
23
24  if(piezas < 1 || piezas > 20 ) {
25      console.log("Cantidad de piezas inválida");
26  }else{
27
28      console.log("Costo total: $" + (costoUnitario * piezas));
29  }
```

Se cumple con:

- Presentar opciones de servicio y sus precios
- Permitir seleccionar una opción
- Validar la opción seleccionada
- Solicitar cantidad de piezas
- Validar cantidad mínima y máxima
- Calcular y mostrar el costo total
- Manejar casos inválidos