

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>clase 17.6 JAVASCRIPT</title>
  <script>
    // Ejercicio 6 - Billetera Virtual
    class BilleteraVirtual {
      //Propiedades
      #monto
      #operaciones

      constructor(montoInicial) {
        this.#monto = montoInicial
        //Los ingresos van en la 1ra lista de operaciones
        // Los gastos van en la 2da lista de operaciones
        this.#operaciones = []
      }

      //Métodos
      agregarGasto(gasto) {
        this.#operaciones.push(gasto)
        console.log(`Gasto agregado`)
      }

      agregarGanancia(ganancia) {
        this.#operaciones.push(ganancia)
        console.log(`Ganancia agregado`)
      }

      obtenerGastosPorMes(mes) {
        let gastos = this.#operaciones.filter((operacion) => operacion.descripcion == 'Gasto')
        let gastosMes = gastos.filter((gasto) => gasto.fecha.includes(mes))
        console.log(`Los gastos del mes de '${mes}' han sido:`)
        for (let operacion of gastosMes) {
          console.log(operacion)
        }
        return gastosMes;
      }

      obtenerGananciasPorMes(mes) {
        let ganancias = this.#operaciones.filter((operacion) => operacion.descripcion ==
'Ganancia')
        let gananciasMes = ganancias.filter((ganancia) => ganancia.fecha.includes(mes))
        console.log(`Las ganancias del mes de '${mes}' han sido:`)
        for (let operacion of gananciasMes) {
          console.log(operacion)
        }
        return gananciasMes;
      }

      obtenerGastosPorCategoria(categoria) {
        let gastos = this.#operaciones.filter((operacion) => operacion.descripcion == 'Gasto')
        let gastosCategoria = gastos.filter((gasto) => gasto.categoria == categoria)
```

```

        if (gastosCategoria.length == 0) {
            console.log(`No existe la categoria '${categoria}' dentro de Gastos`)
        } else {
            console.log(`Los gastos de la Categoria '${categoria}' son:`)
            for (let gasto of gastosCategoria) {
                console.log(gasto)
            }
        }
        return gastosCategoria;
    }

    obtenerGananciasPorCategoria(categoria) {
        let ganancias = this.#operaciones.filter((operacion) => operacion.descripcion ==
'Ganancia')

        let gananciasCategoria = ganancias.filter((ganancia) => ganancia.categoria == categoria)
        if (gananciasCategoria.length == 0) {
            console.log(`No existe la categoria '${categoria}' dentro de Ganancias`)
        } else {
            console.log(`Las ganancias de la categoria '${categoria}' son:`)
            for (let ganancia of gananciasCategoria) {
                console.log(ganancia)
            }
        }
        return gananciasCategoria;
    }

    calcularTotalPorMes(mes) {
        let gananciasMes = this.obtenerGananciasPorMes(mes)
        let gastosMes = this.obtenerGastosPorMes(mes)

        let totalGanancias = gananciasMes.map(op => op.cantidad).reduce((total, cantidad) => total
+ cantidad, 0);
        let totalGastos = gastosMes.map(op => op.cantidad).reduce((total, cantidad) => total +
cantidad, 0);
        let saldo = totalGanancias - totalGastos;

        console.table({
            "Mes": mes,
            "Ganancia Total": totalGanancias,
            "Gasto Total": totalGastos,
            "Saldo": saldo
        });
    }

    obtenerMonto() {
        let ganancias = this.#operaciones.filter((operacion)=>operacion.descripcion == 'Ganancia')
        let gastos = this.#operaciones.filter((operacion)=>operacion.descripcion == 'Gasto')

        let totalGanancias = ganancias.map(op=>op.cantidad).reduce((total,op)=>total+op,0)
        let totalGastos = gastos.map(op=>op.cantidad).reduce((total,op)=>total+op,0)
        let saldo = totalGanancias - totalGastos + this.#monto

        console.table({
            'Monto inicial': this.#monto,
            'Ganancias Totales': totalGanancias,

```

```

        'Gastos Totales': totalGastos,
        'Monto final': saldo,
    })
}
}

```

```

class Operacion {
    //propiedades
    descripcion
    cantidad
    fecha
    categoria

    constructor(descripcion, cantidad, fecha, categoria) {
        this.descripcion = descripcion
        this.cantidad = cantidad
        this.fecha = fecha
        this.categoria = categoria
    }
}

```

```

// Prueba de ejecución

```

```

let billeteraBirtual_01 = new BilleteraVirtual(100)

```

```

let operacion_01 = new Operacion('Gasto', 20, '05-marzo-25', 'Comida')
let operacion_02 = new Operacion('Ganancia', 200, '30-marzo-25', 'Inversiones')
let operacion_03 = new Operacion('Gasto', 30, '04-abril-25', 'Transporte')
let operacion_04 = new Operacion('Gasto', 35, '10-abril-25', 'Comida')
let operacion_05 = new Operacion('Ganancia', 500, '17-abril-25', 'Venta')
let operacion_06 = new Operacion('Gasto', 25, '24-abril-25', 'Comida')
let operacion_07 = new Operacion('Ganancia', 100, '30-abril-25', 'Inversiones')
let operacion_08 = new Operacion('Gasto', 30, '04-mayo-25', 'Transporte')
let operacion_09 = new Operacion('Gasto', 30, '07-mayo-25', 'Comida')
let operacion_10 = new Operacion('Ganancia', 150, '10-mayo-25', 'Venta')

```

```

billeteraBirtual_01.agregarGasto(operacion_01)
billeteraBirtual_01.agregarGanancia(operacion_02)
billeteraBirtual_01.agregarGasto(operacion_03)
billeteraBirtual_01.agregarGasto(operacion_04)
billeteraBirtual_01.agregarGanancia(operacion_05)
billeteraBirtual_01.agregarGasto(operacion_06)
billeteraBirtual_01.agregarGanancia(operacion_07)
billeteraBirtual_01.agregarGasto(operacion_08)
billeteraBirtual_01.agregarGasto(operacion_09)
billeteraBirtual_01.agregarGanancia(operacion_10)

```

```

billeteraBirtual_01.obtenerGastosPorMes('abril')
billeteraBirtual_01.obtenerGananciasPorMes('mayo')

```

```

billeteraBirtual_01.obtenerGastosPorCategoria('Transporte')
billeteraBirtual_01.obtenerGastosPorCategoria('Juguetes')

```

```
        billeteraBirtual_01.obtenerGananciasPorCategoria('Inversiones')

        billeteraBirtual_01.calcularTotalPorMes('abril')

        billeteraBirtual_01.obtenerMonto()

    </script>
</head>
<body>
    <h1>Ejercicio 6 - Clase 17 JAVASCRIPT</h1>
    <h2>Billetera Virtual</h2>
</body>
</html>
```