

Herencia y Composición

Código

```
interface Persona
{
    nombre: string;
    apellidos: string;
    genero: string;
    telefono: string;
    direccion: Direccion;
}

interface Direccion
{
    calle: string;
    codigoPostal: string;
    localidad: string;
    provincia: string;
}

interface Jugador extends Persona
{
    categoria: number;
}

interface Arbitro extends Persona
{
    numeroColegiado: string;
}

class JugadorImpl implements Jugador
{
    constructor(
        public nombre: string,
        public apellidos: string,
        public genero: string,
        public telefono: string,
        public direccion: Direccion,
        public categoria: number
    )
    {}
}
```

```

class ArbitroImpl implements Arbitro
{
    constructor(
        public nombre: string,
        public apellidos: string,
        public genero: string,
        public telefono: string,
        public direccion: Direccion,
        public numeroColegiado: string
    )
    {}
}

class Equipo
{
    constructor(public jugador1: Jugador, public jugador2: Jugador)
    {}

    getNombreEquipo(): string
    {
        return `${this.jugador1.apellidos}-${this.jugador2.apellidos}`;
    }
}

class Set
{
    constructor(public juegosEquipo1: number, public juegosEquipo2: number)
    {}
}

class Partido
{
    private sets: Set[] = [];

    constructor(
        public equipo1: Equipo,
        public equipo2: Equipo,
        public fecha: Date,
        public recinto: string,
        public pista: string,
        public arbitro: Arbitro
    )
    {}
}

```

```

    agregarSet(juegosEquipo1: number, juegosEquipo2: number): void
    {
        this.sets.push(new Set(juegosEquipo1, juegosEquipo2));
    }

    obtenerResultado(): string
    {
        let setsGanadosEquipo1 = 0;
        let setsGanadosEquipo2 = 0;

        for (const set of this.sets)
        {
            if (set.juegosEquipo1 > set.juegosEquipo2)
            {
                setsGanadosEquipo1++;
            }
            else if (set.juegosEquipo2 > set.juegosEquipo1)
            {
                setsGanadosEquipo2++;
            }
        }

        const equipoGanador = setsGanadosEquipo1 > setsGanadosEquipo2 ?
this.equipo1 : this.equipo2;
        const resultado = `${equipoGanador.getNombreEquipo()} ha ganado por
${setsGanadosEquipo1 > setsGanadosEquipo2 ? setsGanadosEquipo1 :
setsGanadosEquipo2} sets a ${setsGanadosEquipo1 > setsGanadosEquipo2 ?
setsGanadosEquipo2 : setsGanadosEquipo1}.`;
        return resultado;
    }
}

class Calendario
{
    private partidos: Partido[] = [];

    agregarPartido(partido: Partido): void
    {
        this.partidos.push(partido);
    }
}

```

```

    mostrarCalendario(): void
    {
        for (const partido of this.partidos)
        {
            console.log(`Partido entre ${partido.equipo1.getNombreEquipo()}
y ${partido.equipo2.getNombreEquipo()}`);
            console.log(`Fecha: ${partido.fecha}`);
            console.log(`Recinto: ${partido.recinto}, Pista: $
{partido.pista}`);
            console.log(`Árbitro: ${partido.arbitro.nombre} $
{partido.arbitro.apellidos}`);
            console.log(`Resultado: ${partido.obtenerResultado()}`);
            console.log('_____');
        }
    }
}

const direccion1: Direccion = { calle: "Calle 1", codigoPostal: "12345",
localidad: "Ciudad A", provincia: "Provincia A" };
const direccion2: Direccion = { calle: "Calle 2", codigoPostal: "67890",
localidad: "Ciudad B", provincia: "Provincia B" };

const jugador1 = new JugadorImpl("Juan", "López", "M", "111-111-111",
direccion1, 3);
const jugador2 = new JugadorImpl("Pedro", "Martín", "M", "222-222-222",
direccion2, 4);
const jugador3 = new JugadorImpl("Luis", "Carrasco", "M", "333-333-333",
direccion1, 5);
const jugador4 = new JugadorImpl("Carlos", "Beltrán", "M", "444-444-444",
direccion2, 2);
const jugador5 = new JugadorImpl("Pepe", "García", "M", "555-555-555",
direccion1, 1);
const jugador6 = new JugadorImpl("Teresa", "Murillo", "F", "666-666-666",
direccion2, 6);

const equipo1 = new Equipo(jugador1, jugador2);
const equipo2 = new Equipo(jugador3, jugador4);
const equipo3 = new Equipo(jugador5, jugador6);

const arbitro1 = new ArbitroImpl("Ana", "Gómez", "F", "777-777-777",
direccion1, "ARB001");
const arbitro2 = new ArbitroImpl("María", "Pérez", "F", "888-888-888",
direccion2, "ARB002");

const calendario = new Calendario();

```

```

const partido1 = new Partido(equipo1, equipo2, new Date("2023-10-01T10:00:00"), "Polideportivo A", "Pista 1", arbitro1);
partido1.agregarSet(6, 4);
partido1.agregarSet(3, 6);
partido1.agregarSet(5, 7);

const partido2 = new Partido(equipo2, equipo3, new Date("2023-10-05T10:00:00"), "Polideportivo B", "Pista 2", arbitro2);
partido2.agregarSet(3, 6);
partido2.agregarSet(4, 6);

const partido3 = new Partido(equipo1, equipo3, new Date("2023-10-08T10:00:00"), "Polideportivo A", "Pista 1", arbitro1);
partido3.agregarSet(6, 7);
partido3.agregarSet(7, 5);
partido3.agregarSet(6, 3);

calendario.agregarPartido(partido1);
calendario.agregarPartido(partido2);
calendario.agregarPartido(partido3);

calendario.mostrarCalendario();

```

Resultado

Partido entre López-Martín y Carrasco-Beltrán	di312.ts:126
Fecha: Sun Oct 01 2023 10:00:00 GMT+0200 (hora de verano de Europa central)	di312.ts:127
Recinto: Polideportivo A, Pista: Pista 1	di312.ts:128
Árbitro: Ana Gómez	di312.ts:129
Resultado: Carrasco-Beltrán ha ganado por 2 sets a 1.	di312.ts:130
-----	di312.ts:131
Partido entre Carrasco-Beltrán y García-Murillo	di312.ts:126
Fecha: Thu Oct 05 2023 10:00:00 GMT+0200 (hora de verano de Europa central)	di312.ts:127
Recinto: Polideportivo B, Pista: Pista 2	di312.ts:128
Árbitro: María Pérez	di312.ts:129
Resultado: García-Murillo ha ganado por 2 sets a 0.	di312.ts:130
-----	di312.ts:131
Partido entre López-Martín y García-Murillo	di312.ts:126
Fecha: Sun Oct 08 2023 10:00:00 GMT+0200 (hora de verano de Europa central)	di312.ts:127
Recinto: Polideportivo A, Pista: Pista 1	di312.ts:128
Árbitro: Ana Gómez	di312.ts:129
Resultado: López-Martín ha ganado por 2 sets a 1.	di312.ts:130
-----	di312.ts:131