

Proyecto TypeScript:

Trabajo con clases

Aplicación de gestión de un equipo de carreras



Contenido

1. Contexto:.....	3
2. Definición de clases:	3
3. Estructura de los datos:.....	4
4. Funciones de la Aplicación:	6

1. Contexto:

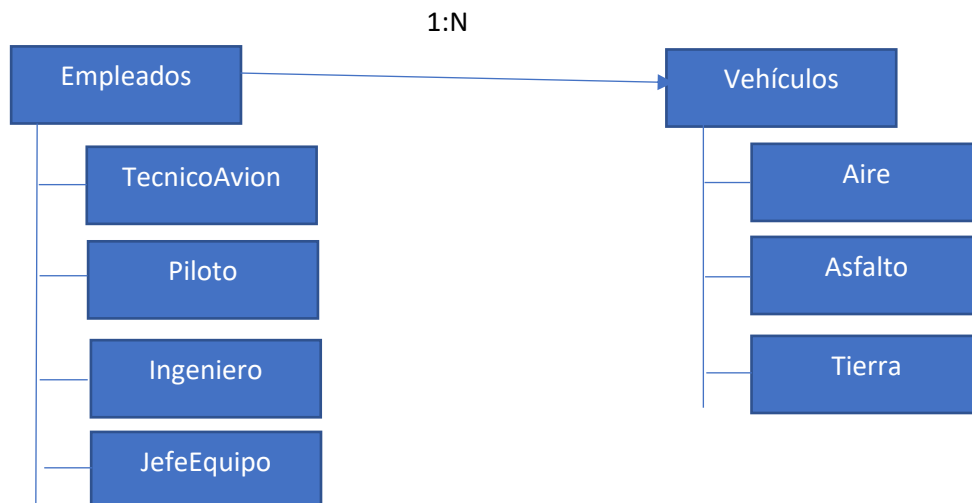
Para elaborar el proyecto se ha elegido de ejemplo la organización del equipo multidisciplinar **Repsol**

Este equipo tiene una sección de Motos, Aviones y Coches de Rally. En el pasado también tenían una sección de Camiones, y es posible que en el futuro se pueda expandir a otras categorías.



2. Definición de clases:

Principalmente tendremos 2 superclases, de las que extenderán cada una de sus subclases.



La relación entre clases se dará por medio de un campo “**_numeroSerie**” común entre ambos.

3. Estructura de los datos:

Empleados:

```
export type iIngeniero = { //Schema tipo ingeniero
  _dni: string | null,
  _nombre: string | null,
  _puesto: string | null,
  _salario: number | null,
  _numeroSerie: string | null,
  _experiencia: number | null,
  _estudios: string | null,
  _idiomas: string[] | null,
}
```

```
export type iPiloto = { //Schema tipo piloto
  _dni: string | null,
  _nombre: string | null,
  _puesto: string | null,
  _salario: number | null,
  _numeroSerie: string | null,
  _descalificaciones: number | null,
  _podios: number | null,
  _mundiales: number | null,
  _poles: number | null,
  _victorias: number | null,
}
```

```
export type itecnicoAvion = { //Schema tipo tecnico de avion
  _dni: string | null,
  _nombre: string | null,
  _numeroSerie: string | null,
  _puesto: string | null,
  _salario: number | null,
  _modelo: string | null,
  _horasDeVuelo: number | null,
  _horasDeMantenimiento: number | null,
}

export type iJefeEquipo = { //Schema tipo jefe de equipo
  _dni: string | null,
  _nombre: string | null,
  _puesto: string | null,
  _salario: number | null,
  _numeroSerie: string | null,
  _proyectos: string[] | null,
  _posicion1: number | null,
  _cilindraje: string | null,
  _fechaContratacion: Date | null,
  _trial: boolean | null,
}
```

Vehículos:

```
export type iAire = {
  _tipoVehiculo: string | null,
  _numeroSerie: string | null,
  _modelo: string | null,
  _presupuesto: number | null,
  _posicionMundial: number | null,
  _motoresUsados: number | null,
}

export type iAsfalto = {
  _tipoVehiculo: string | null,
  _numeroSerie: string | null,
  _modelo: string | null,
  _presupuesto: number | null,
  _posicionMundial: number | null,
  _unidadesPotencia: number | null,
  _ruedas: number | null,
  _cilindraje: string | null,
  _trial: boolean | null
}

export type iTierra = {
  _tipoVehiculo: string | null,
  _numeroSerie: string | null,
  _modelo: string | null,
  _presupuesto: number | null,
  _posicionMundial: number | null,
  _Dakar: boolean | null,
  _siniestros: number | null,
  _dias: Date | null,
}
```

4. Funciones de la Aplicación:

```
1.- Nuevo Vehiculo
2.- Nuevo Empleado
3.- Presupuesto
4.- Sueldos por Vehiculo
5.- Subida de datos de prueba
6.- Modificar Empleado
7.- Eliminar
8.- Crear tabla de Resultados
9.- Horas de vuelo
10.-Comparar Antigüedad de Jefes
opción: : ^C
D:\ASTBsegundo\proyectoPD\podo_dict
```

- 1- Guardado de vehículos en todas sus variantes **(CRUD)**
- 2- Guardado de empleadas en todas sus variantes **(CRUD)**
- 3- Presupuesto por vehículo **(polimorfismo)**
- 4- Se calcula el sueldo de todos los empleados ligados a un vehículo **(polimorfismo)**
- 5- Subida de banco de datos
- 6- Modificación de la asociación entre empleados y vehículos **(persistencia)**
- 7- Eliminar un registro **(CRUD)**
- 8- Creación de una nueva colección de datos derivados de las dos clases principales **(persistencia)**
- 9- Obtención de horas de vuelo por avión
- 10- Comparar antigüedad jefes **(métodos)**