



El empleo
es de todos

Mintrabajo

APLICAR PROGRAMACIÓN ORIENTADA A OBJETO CON C#

José Fernando Galindo Suárez



@SENAcomunica

www.sena.edu.co



PROGRAMACIÓN ORIENTADA A OBJETO CON C#



@SENAcomunica

www.sena.edu.co

4

INTERFACES PROGRAMACIÓN ORIENTADA A OBJETO CON C#

OBJETIVOS



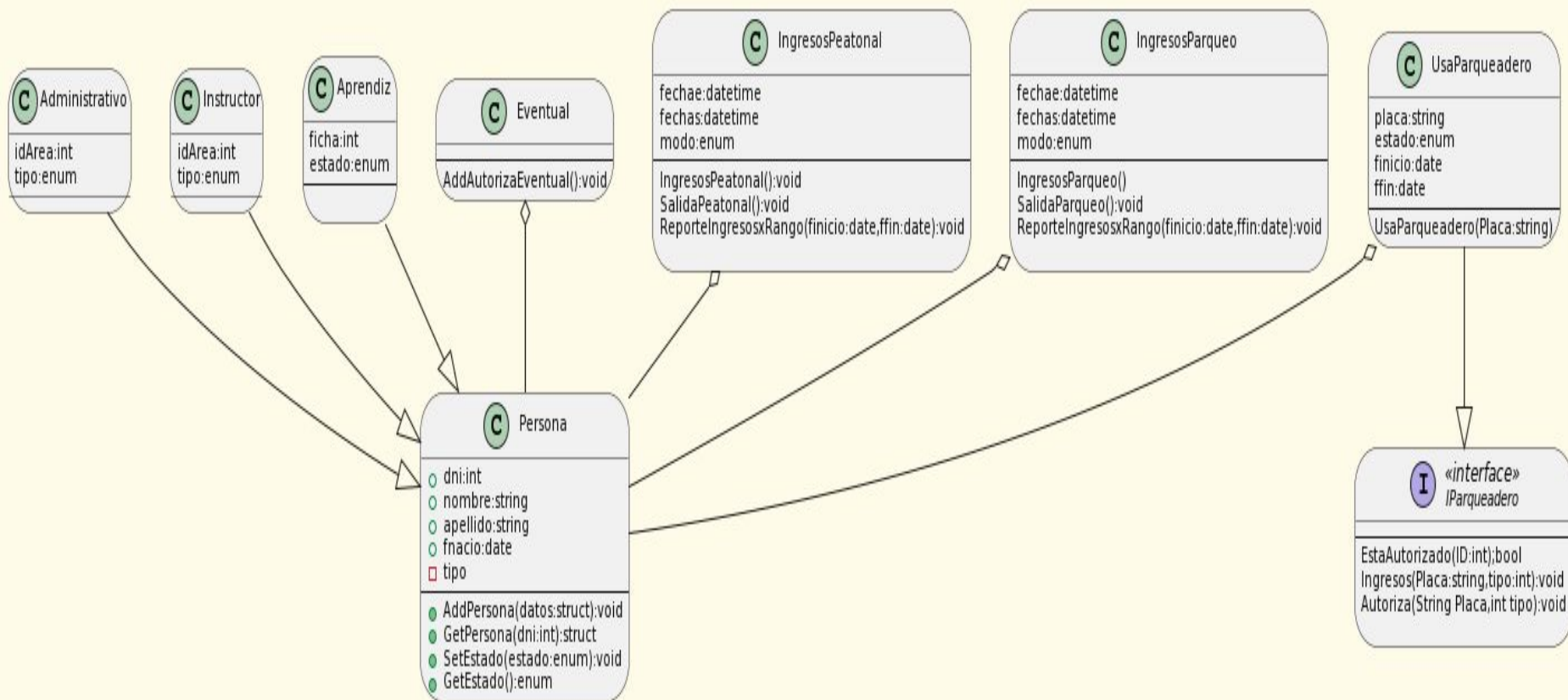
Después de completar esta lección usted estará en la capacidad de:

- **Definir los conceptos básicos de la utilización de las interfaces programación basada en objetos con C#**
- **Crear objetos heredada de una interface dada y modificar o restringir el acceso a su estado y su comportamiento.**

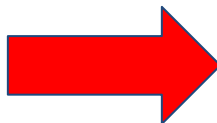
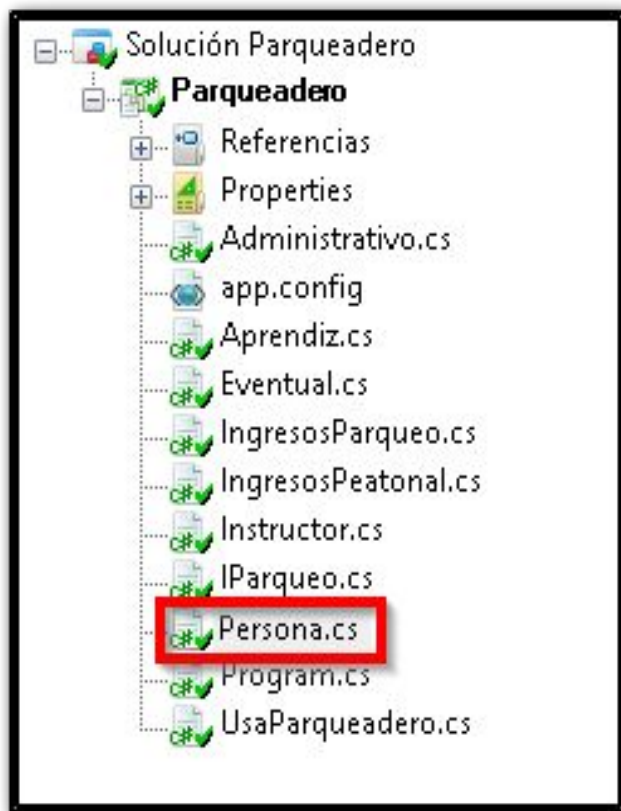
Una interfaz (interface) es sintácticamente similar a una clase abstracta, en la que puede especificar uno o más métodos que no tienen cuerpo (`{}`). Esos métodos deben ser implementados por una clase para que se definan sus acciones.

Por lo tanto, una interfaz especifica qué se debe hacer, pero no cómo hacerlo. Una vez que se define una interfaz, cualquier cantidad de clases puede implementarla. Además, una clase puede implementar cualquier cantidad de interfaces.

INTERFACES - CASO DE ESTUDIO



INTERFACES



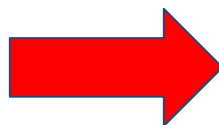
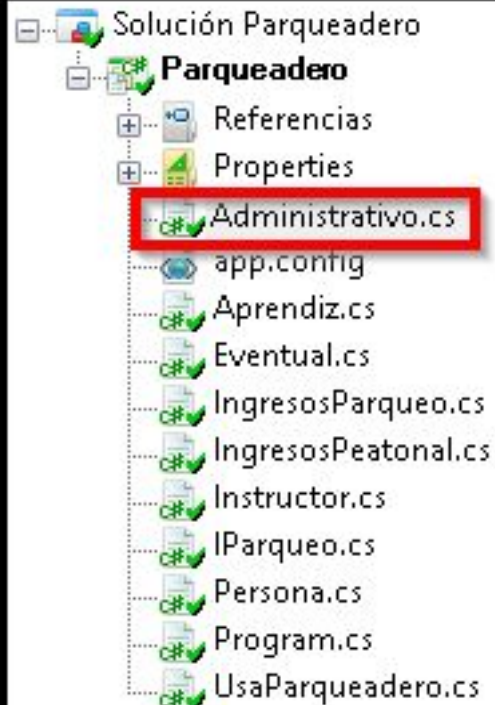
```
using System;

namespace Parqueadero
{
    /// <summary>
    /// Description of Persona.
    /// </summary>
    public class Persona
    {
        public enum Etipo{ 1
            InstructorPlanta,
            InstructorContrato,
            AdministrativoPlanta,
            AdministrativoContratista,
            Aprendiz,
            Egresado
        }

        public struct DatosPer{ 2
            public int dni;
            public string nombre;
            public string apellido;
            public DateTime fnacio ;
            public Etipo tipo;
        }

        public DatosPer DPesonales; 3
        public void AdicionPersona(DatosPer datos) 4
        {
        }
    }
}
```

INTERFACES

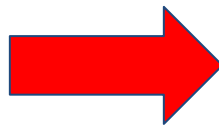
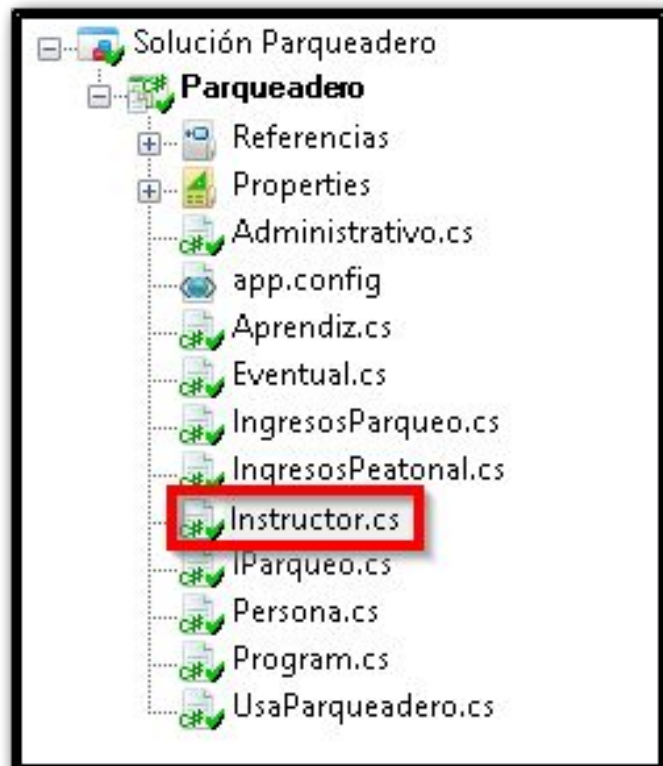


```
using System;

namespace Parquadero
{
    /// <summary>
    /// Description of Administrativo.
    /// </summary>
    public class Administrativo:Persona
    {
        enum Etipo{ 1
            Planta,
            Contrato
        }

        int idarea; 2
        Etipo tipo;
        public Administrativo() 3
        {
        }
    }
}
```


INTERFACES

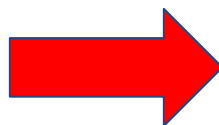
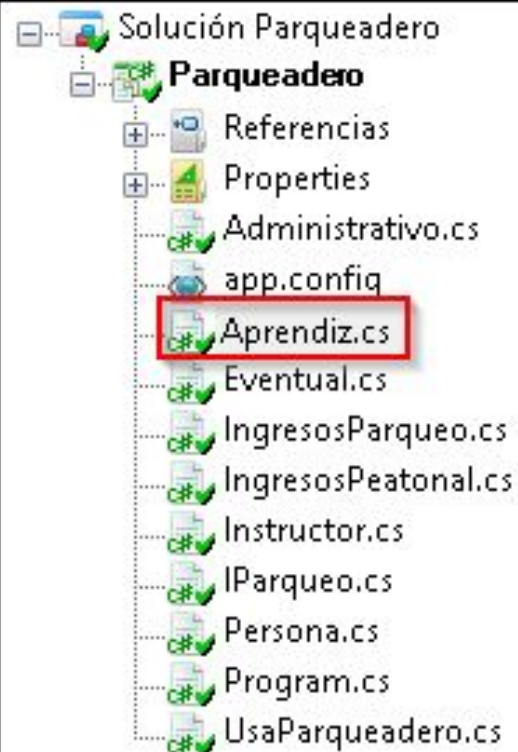


```
using System;

namespace Parqueadero
{
    /// <summary>
    /// Description of Instructor.
    /// </summary>
    public class Instructor:Persona
    {
        enum Etipo { 1
            Planta,
            Contrato
        }

        int idarea; 2
        Etipo tipo; 2
        public Instructor() 3
        {
        }
    }
}
```

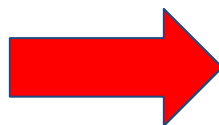
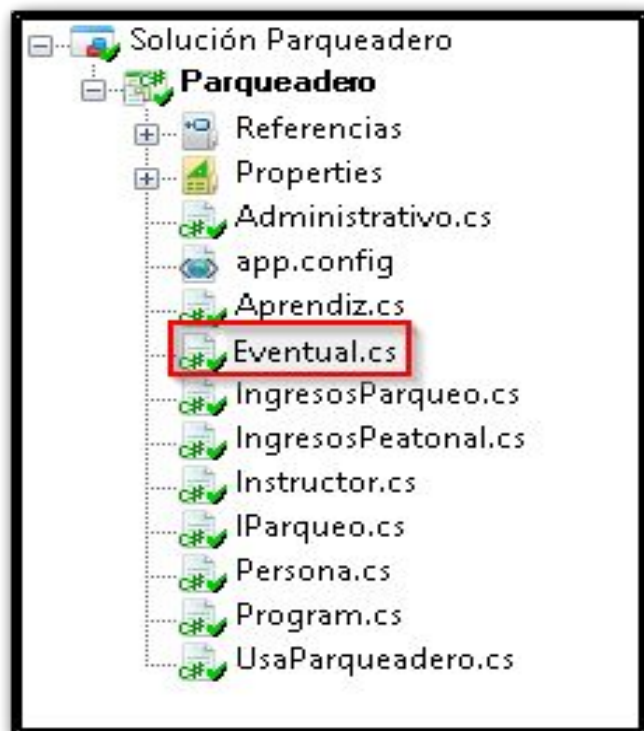
INTERFACES



```
using System;

namespace Parqueadero
{
    /// <summary>
    /// Description of Aprendiz.
    /// </summary>
    public class Aprendiz:Persona
    {
        enum EEstado{
            Inactivo, 1
            Activo
        }
        int ficha;
        EEstado estado{get;set;} 2
        public Aprendiz() 3
        {
        }
    }
}
```

INTERFACES

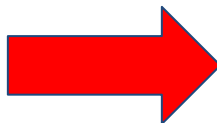
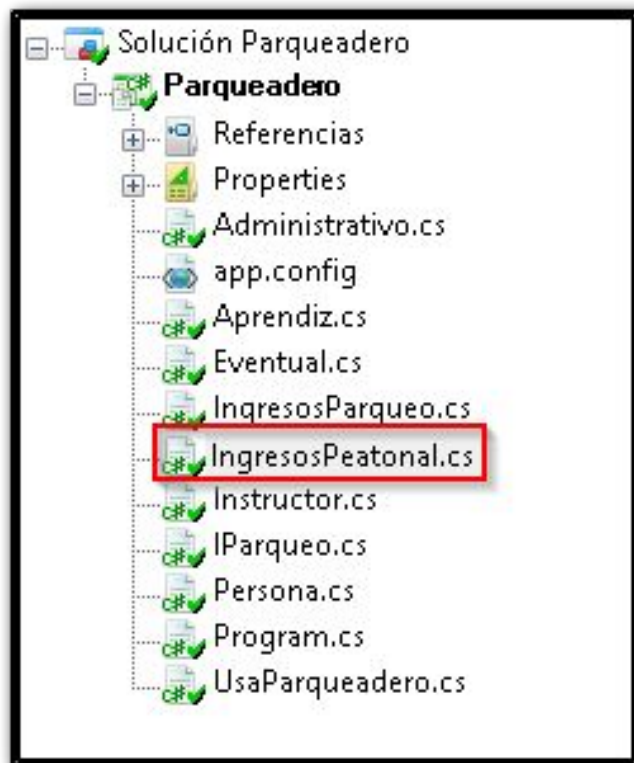


```
using System;

namespace Parquadero
{
    /// <summary>
    /// Description of Eventual.
    /// </summary>
    public class Eventual:Persona
    {
        public void AutorizaEventual()
        {
        }
    }
}
```

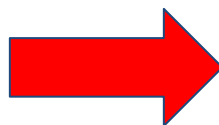
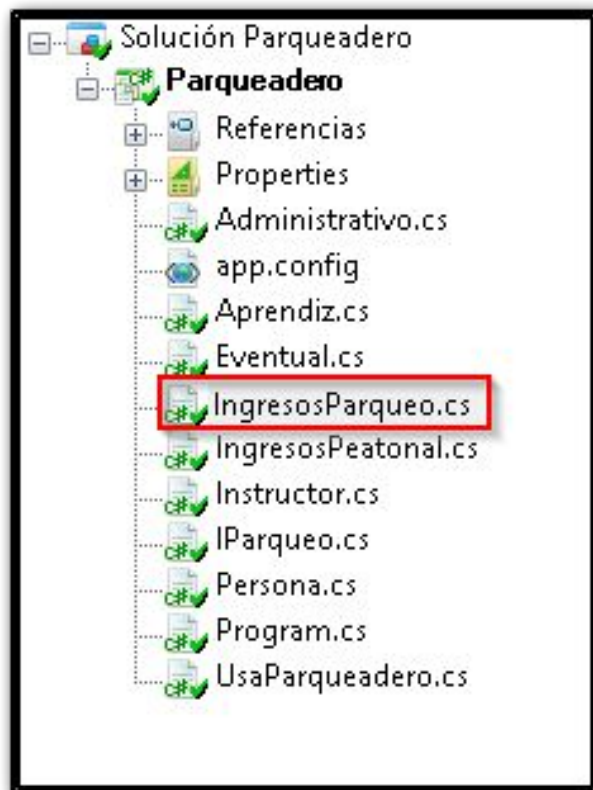
1

INTERFACES



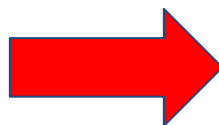
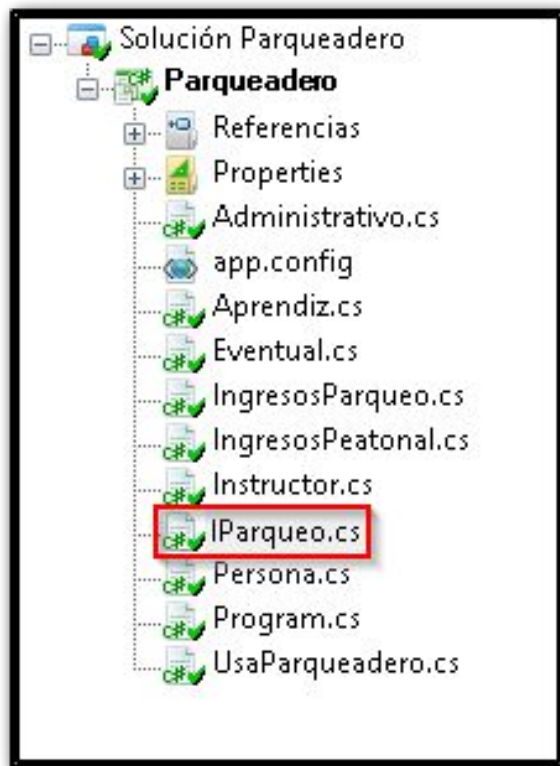
```
using System;
namespace Parquadero
{
    public class IngresosPeatonal:Persona
    {
        public enum Emodo{
            entra, 1
            sale
        }
        public struct Datos{
            public DateTime fechaE; 2
            public DateTime fechas;
            public Emodo modo{get;set;}
        }
        public Datos DatosE; 3
        public IngresosPeatonal() 4
        {
            DatosE.fechaE=DateTime.Now;
            DatosE.modo=Emodo.entra;
        }
        public void SalidaPeatonal(){ 5
            DatosE.fechas=DateTime.Now;
            DatosE.modo=Emodo.sale;
        }
        6 public void ReporteIngresosxRango(DateTime fechaI,DateTime fechas){
        }
    }
}
```

INTERFACES



```
using System;
namespace Parquadero
{
    public class IngresosParqueo
    {
        public enum Emodo{
            entra, 1
            sale
        }
        public struct Datos{
            public DateTime fechaE; 2
            public DateTime fechaS;
            public Emodo modo{get;set;}
        }
        public Datos DatosE; 3
        public IngresosParqueo()
        {
            DatosE.fechaE=DateTime.Now; 4
            DatosE.modo=Emodo.entra;
        }
        public void SalidaParqueo(){
            DatosE.fechaS=DateTime.Now; 5
            DatosE.modo=Emodo.sale;
        }
        6 public void ReporteIngresosxRango(DateTime fechaI,DateTime fechaS){
        }
    }
}
```

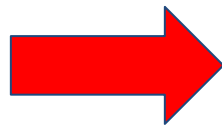
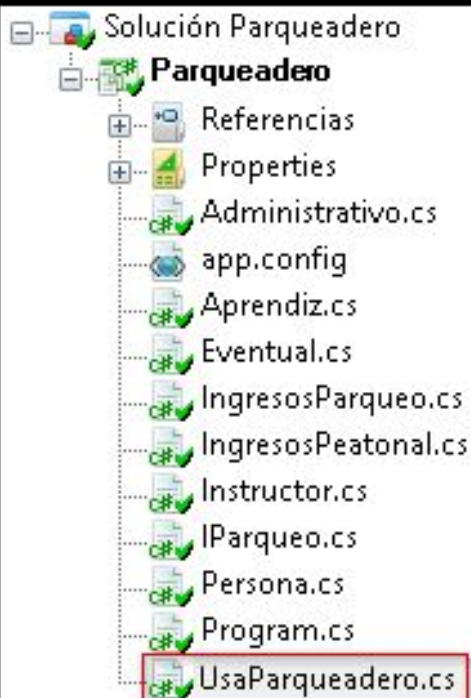
INTERFACES



```
using System;

namespace Parquadero
{
    public interface IParqueo 1
    {
        2 bool EstaAutorizado(int ID);
        void Ingresos(string Placa,int tipo); 3
        4 void Autoriza(string Placa,int tipo);
    }
}
```


INTERFACES



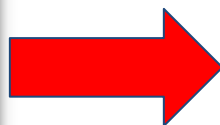
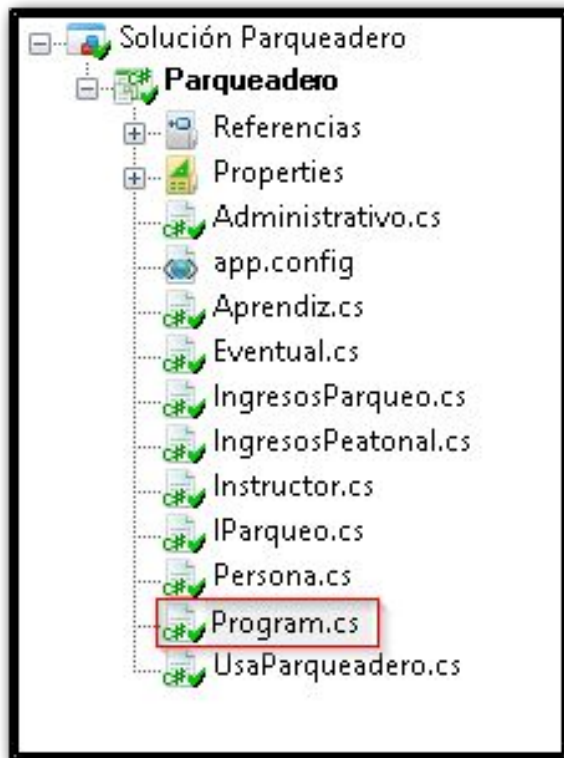
```
using System;
namespace Parquadero
{
    public class UsaParquadero: 1Persona, 2IParqueo
    {
        public enum EEstado{Inactivo,Activo}

        public struct SDatos{
            public string placa;
            public DateTime fechai;
            public DateTime fechaf;
            public EEstado estado;
        }

        4 public SDatos Datos;
        public UsaParquadero(5string Placa)
        {
            Datos.placa=Placa;
            Datos.fechai=DateTime.Now;
        }

        6 public bool EstaAutorizado(int ID){return true;}
        public void Ingresos(string Placa,int tipo){} 7
        8 public void Autoriza(string Placa,int tipo){}
    }
}
```

INTERFACES



```
using System;
namespace Parquadero
{
    class Program
    {
        public static void Main(string[] args)
        {
            1 var i=new Instructor();
              i.DPesonales.dni=51693445;
              i.DPesonales.nombre="Zoila";
              i.DPesonales.apellido="Vaca";
              i.DPesonales.tipo=0;
            // TODO: Implement Functionality Here
            Console.WriteLine("DNI={0}\nNombre={1} {2}\n",i.DPesonales.dni,i.DPesonales.nombre,i.DPesonales.apellido);
            3 Console.Write("Press any key to continue . . .");
            Console.ReadKey(true);
        }
    }
}
```

DNI=51693445
Nombre=Zoila Vaca

Press any key to continue . . . _

DESCARGAR ARCHIVOS



COMPILADOR SHARPDEV

VERSIÓN PDF





GRACIAS

Línea de atención al ciudadano: 018000 910270
Línea de atención al empresario: 018000 910682



@SENAcomunica

www.sena.edu.co

CRÉDITOS



Realizado por el instructor José Fernando Galindo Suárez
jgalindos@sena.edu.co 2022

