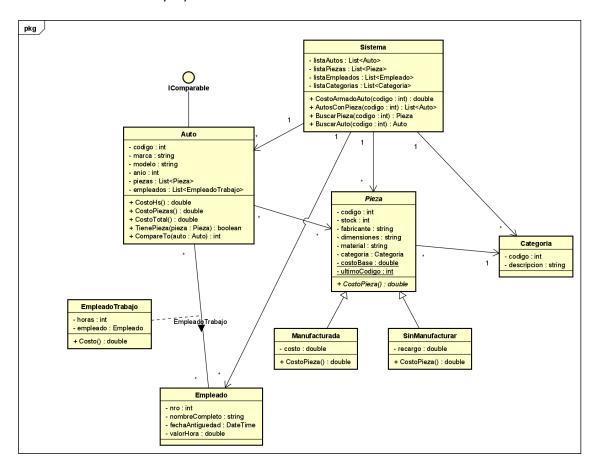
## Solución examen P2 – 20/07/2022



# Punto 1

```
Clase Pieza
```

```
public abstract class Pieza
{
      public abstract double CostoPieza();
Clase Manufactura
public class Manufacturada : Pieza
       public override double CostoPieza()
          return CostoBase + Costo;
Clase SinManufacturar
public class SinManufacturar : Pieza
      public override double CostoPieza()
            double costo = CostoBase;
            if (Categoria.Descripcion.Trim().ToUpper() != "TORNILLOS")
                costo += Recargo;
            return costo;
        }
Clase EmpleadoTrabajo
public class EmpleadoTrabajo
{
       public double Costo()
         return Horas * Empleado.ValorHora;
```

### Clase Auto

```
public class Auto
   public double CostoPiezas()
   {
       double total = 0;
       foreach(Pieza pieza in piezas)
           total += pieza.CostoPieza();
       return total;
   }
  public double CostoHs()
      double total = 0;
      foreach(EmpleadoTrabajo empleadoTrabajo in empleados)
         total += empleadoTrabajo.Costo();
      return total;
   }
  public double CostoTotal()
      return CostoHs() + CostoPiezas();
Clase Sistema
public class Sistema
       public Auto BuscarAuto(int codigo)
        {
            Auto auto = null;
            int index = 0;
            while (auto == null && index < listaAutos.Count)</pre>
                if (listaAutos[index].Codigo == codigo)
                    auto = listaAutos[index];
                index++;
            }
            return auto;
        }
        public double CostoArmadoAuto(int codigo)
            double costo = 0;
            Auto auto = BuscarAuto(codigo);
            if (auto != null)
                costo = auto.CostoTotal();
            return costo;
        }
```

## Punto 2

```
public abstract class Pieza
{
    public override bool Equals(object obj)
    {
        return ((Pieza)obj).Codigo == this.Codigo;
    }

Clase Auto

public class Auto:IComparable<Auto>
    {
        public bool TienePieza(Pieza pieza)
        {
            return Piezas.Contains(pieza);
        }

        public int CompareTo([AllowNull] Auto other)
        {
            return this.Marca.CompareTo(other.Marca);
        }
}
```

### Clase Sistema

## Punto 3

#### Vista

```
@model WebApplication1.Models.Empleado
<h1>Alta de empleado</h1>
<hr />
<div>
    @ViewBag.Mensaje
</div>
<form action="/Empleado/Registar" method="post">
      <label>Valor hora</label>
      <input type="number" value="@Model.ValorHora" name="ValorHora"/>
  </div>
  <div>
       <label>Nro empleado</label>
       <input type="number" value="@Model.Nro" name="Nro" />
  </div>
  <div>
       <label>Nombre completo</label>
       <input value="@Model.NombreCompleto" name="NombreCompleto" />
  </div>
  <div>
       <label>Fecha antiguedad</label>
       <input type="date" value="@Model.FechaAntiguedad"</pre>
name="FechaAntiguedad"/>
  </div>
  <input type="submit" value="Registrar" />
</form>
```

## Controlador

```
public class EmpleadoController : Controller
    Sistema sistema = new Sistema();
    public ActionResult Index()
        return View(new Empleado());
    }
    [HttpPost]
    public ActionResult Registar(Empleado empleado)
        if (sistema.AltaEmpleado(empleado))
            ViewBag.Mensaje = "Empleado registrado";
            return View("Index",empleado);
        }
        else
        {
            ViewBag.Mensaje = "No se pudo registrar el empleado";
            return View("Index", new Empleado());
        }
    }
}
```