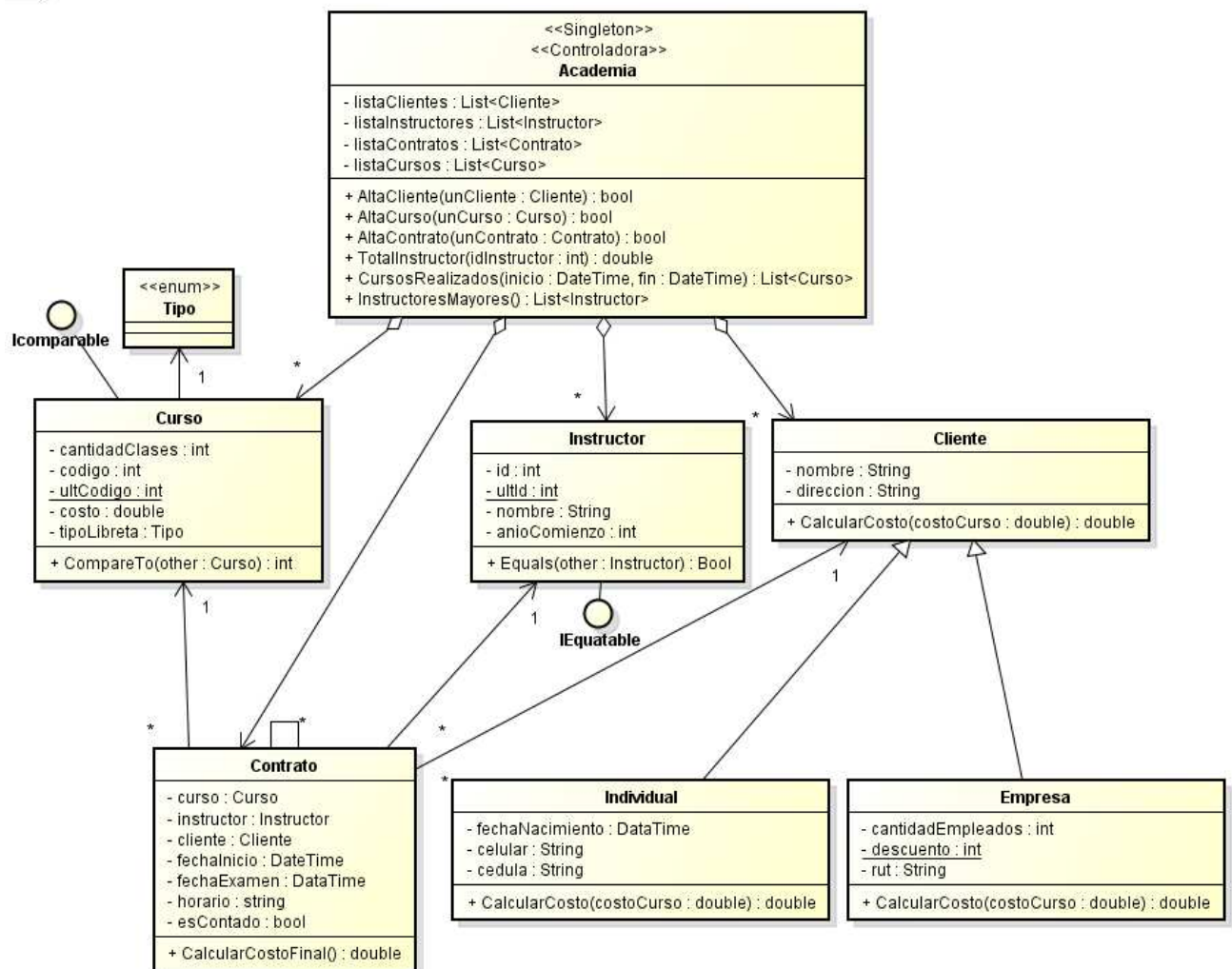


EVALUACION	<b>SOLUCION Examen</b>	GRUPO		FECHA	27/02/2018
MATERIA	PROGRAMACIÓN 2				
CARRERA	Analista Programador / Analista en Tecnologías de la Información				
CONDICIONES	<ul style="list-style-type: none"> <li>- Puntos: 100 (MÁXIMO) – 0 (MÍNIMO)</li> <li>- Duración: 3 Hrs</li> <li>- Sin material</li> <li>- No escriba la hoja de la letra</li> <li>- Consultas solamente sobre interpretación de la letra y sintaxis específica del lenguaje.</li> </ul>				

**Parte 1**  
*priv*



## Parte 2

### Punto C

```
// Academia
// punto c
public double TotalInstructor(int idInstructor)
{
    double total=0;

    foreach (Contrato unContrato in listaContratos)
    {
        if (unContrato.Instructor.Id == idInstructor)
        {
            total += unContrato.CalcularCostoFinal();
        }
    }
    return total;
}

// Contrato
// punto c
public double CalcularCostoFinal()
{
    double total = 0;
    double costo = curso.costo);

    if (esContado)
    {
        costo *= 0.85;
    }

    total = cliente.CalcularCosto(costo);
    return total;
}

// Individual
// punto c
public double CalcularCosto (double costoCurso)
{
    double costo = costoCurso;

    if (fechaNacimiento.year > 1995)
    {
        costo *= 0.75;
    }
    costo += alquilerAuto;

    return costo;
}
```

```
// Empresa
// punto c
public double CalcularCosto (double costoCurso)
{
    double costo = costoCurso;
    costo *= descuento;
    if (cantidadEmpleados > 10)
    {
        costo *= 0.70;
    }
    return costo;
}
```

**Punto D**

```
// Academia
// punto d
public List<Curso> CursosRealizados(DateTime inicio, DateTime fin)
{
    List<Curso> auxCurso = new List<Curso>();
    foreach (Contrato unContrato in listaContratos)
    {
        if (unContrato.fechaInicio >= inicio && unContrato.fechaInicio <= fin &&
            unContrato.esContado)
        {
            auxCurso.Add (unContrato.Curso)
        }
    }
    auxCurso.Sort();
    return auxCurso;
}

// Curso
// punto d
public int CompareTo(Curso other)
{
    return (other.cantidadClases - this.cantidadClases);
}
```

**Punto E**

```
// Academia
// punto e
public List<Instructor> InstructoresMayores()
{
    List<Instructor> auxInstructor = new List<Instructor>();
    int mayor = 0;
    double costo = 0;
    foreach (Contrato unContrato in listaContratos)

        costo = unContrato.CalcularCostoFinal();
        if (costo > mayor)
        {
            mayor = costo;
            auxInstructor.Clear();
            auxInstructor.Add(unContrato.Instructor)
        }
        else
        {
            if (costo == mayor)
            {
                If (!auxInstructor.Contains(unContrato.Instructor))
                {
                    auxInstructor.Add(unContrato.Instructor)
                }
            }
        }

    }
    return auxInstructor;
}

// Instructor
// punto d
public Bool Equals(Instructor other)
{
    return (other != null && this.id = other.id;)
}
```