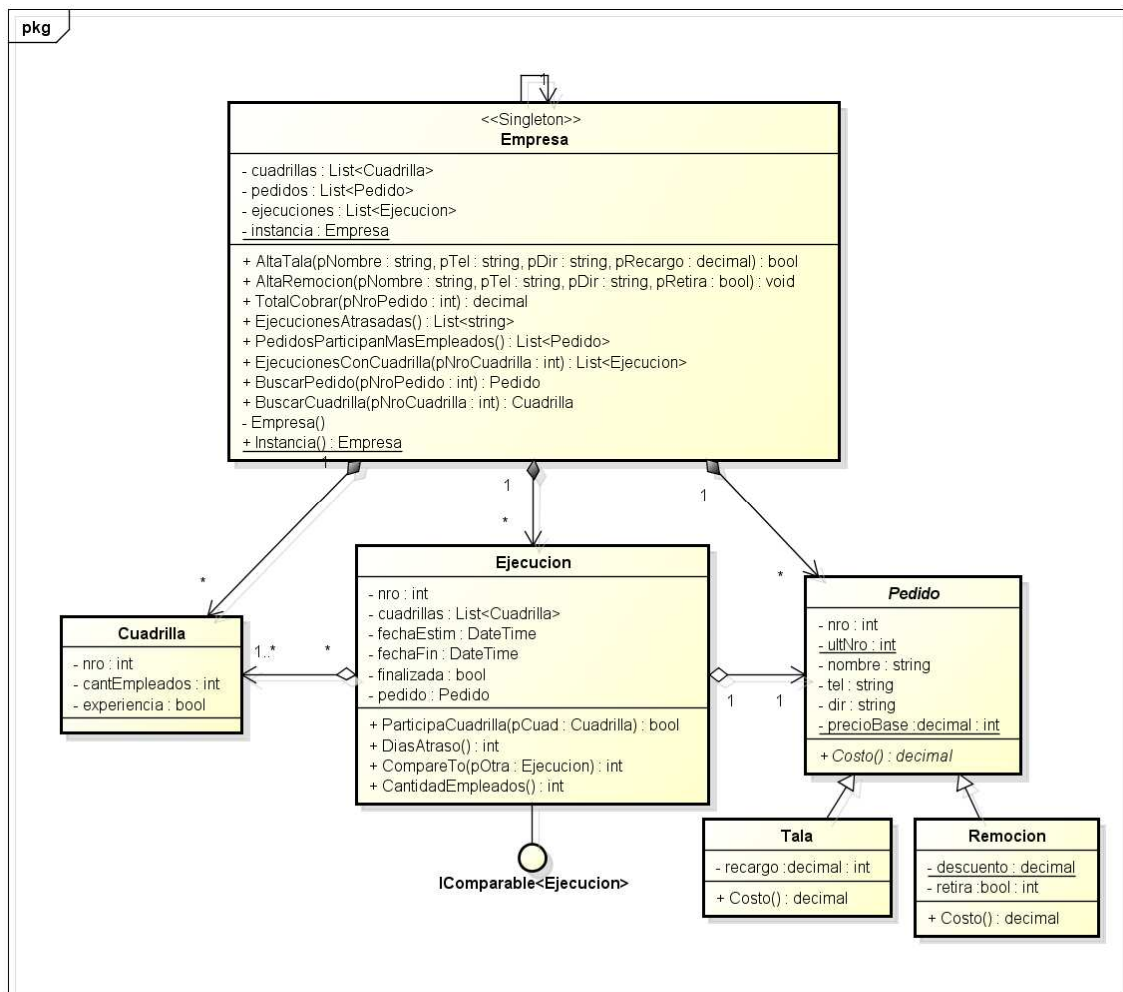


EVALUACION	SOLUCION EXAMEN	GRUPO	TODOS	FECHA	07/11/2014
MATERIA	PROGRAMACIÓN 2				
CARRERA	AP/ATI/APW				
CONDICIONES	<p>- Puntos: 100</p> <p>- Duración: 3 Horas</p> <p>- Sin material</p> <p>- Otros : No escriba la hoja de la letra</p> <p>Consultas solamente sobre interpretación de letra y sintaxis específica del lenguaje.</p> <p>Numerar las hojas entregadas.</p> <p>Indicar nombre del docente del curso en primera hoja del examen</p>				

Solución Examen Noviembre 2014

1. Diagrama UML:



2. Implementación:

b. (20 puntos):

EN CLASE EMPRESA:

```
public decimal TotalACobrar(int pNroPedido)
{
    decimal total = 0;

    Pedido aux = this.BuscarPedido(pNroPedido);

    if (aux != null) {
        total = aux.Costo();
    }

    return total;
}

public Pedido BuscarPedido(int pNroPedido) {
    Pedido encontrado = null;

    int i = 0;
    while (i < this.pedidos.Count && encontrado == null)
    {
        Pedido aux = this.pedidos[i];

        if (aux.Nro == pNroPedido)
        {
            encontrado = aux;
        }
    }

    return encontrado;
}
```

EN CLASE PEDIDO:

```
public abstract class Pedido
...
...
public abstract decimal Costo();
```

EN CLASE TALA:

```
public class Tala : Pedido
...
...
public override decimal Costo()
{
    decimal costo = this.PrecioBase;
    costo += costo * this.Recargo / 100;
    return costo;
}
```

EN CLASE REMOCION:

```
public class Remocion : Pedido
...
...
public override decimal Costo()
{
    decimal costo = this.PrecioBase;

    if (Retira == true)
    {
        costo -= costo * Remocion.Descuento / 100;
    }

    return costo;
}
```

c. (20 puntos):

EN CLASE EMPRESA:

```
public List<Ejecucion> EjecucionesAtrasadas()
{
    List<Ejecucion> retorno = new List<Ejecucion>();

    foreach (Ejecucion unaEjec in this.ejecuciones)
    {
        if (unaEjec.DiasAtraso() > 0)
        {
            retorno.Add(unaEjec);
        }
    }

    retorno.Sort();
    return retorno;
}
```

EN CLASE EJECCION:

```
public class Ejecucion : IComparable<Ejecucion>
...
...
public int CompareTo(Ejecucion pOtra)
{
    int retorno = 0;

    if (pOtra != null)
    {
        retorno = pOtra.DiasAtraso().CompareTo(this.DiasAtraso());
    }

    return retorno;
}
```

EN CLASE EJECUCION (CONT):

```
public int DiasAtraso()
{
    int dias = 0;

    if (this.Finalizada)
    {
        dias = (this.FechaFin - this.FechaEstim).Days;
    }

    return dias;
}
```

d. (15 puntos):

EN CLASE EMPRESA:

```
public List<Pedido> PedidosParticipanMasEmpleados()
{
    List<Pedido> retorno = new List<Pedido>();
    int maxCant = 0;

    foreach (Ejecucion aux in this.ejecuciones)
    {
        int cantEmp = aux.CantidadEmpleados();
        if (cantEmp >= maxCant)
        {
            if (cantEmp > maxCant)
            {
                maxCant = cantEmp;
                retorno.Clear();
            }

            retorno.Add(aux.Pedido);
        }
    }

    return retorno;
}
```

EN CLASE EJECUCIÓN:

```
public int CantidadEmpleados() {
    int cant = 0;

    foreach (Cuadrilla unaCuad in this.Cuadrillas)
    {
        cant += unaCuad.CantEmpleados;
    }

    return cant;
}
```

e. (15 puntos):

EN CLASE EMPRESA:

```
public List<Ejecucion> EjecucionesConCuadrilla(int pNroCuadrilla)
{
    List<Ejecucion> retorno = new List<Ejecucion>();

    Cuadrilla aux = this.BuscarCuadrilla(pNroCuadrilla);

    if (aux != null)
    {
        foreach (Ejecucion unaEjec in this.ejecuciones)
        {
            if (unaEjec.ParticipaCuadrilla(aux))
            {
                retorno.Add(unaEjec);
            }
        }
    }

    return retorno;
}

public Cuadrilla BuscarCuadrilla(int pNroCuadrilla)
{
    Cuadrilla encontrada = null;

    int i = 0;
    while (i < this.cuadrillas.Count && encontrada == null)
    {
        Cuadrilla aux = this.cuadrillas[i];

        if (aux.Nro == pNroCuadrilla)
        {
            encontrada = aux;
        }
    }

    return encontrada;
}
```

EN CLASE EJECUCION:

```
public bool ParticipaCuadrilla(Cuadrilla pCuad)
{
    bool retorno = false;

    int i = 0;
    while (i < this.Cuadrillas.Count && retorno == false)
    {
        if (this.Cuadrillas[i] == pCuad) retorno = true;
    }

    return retorno;
}
```