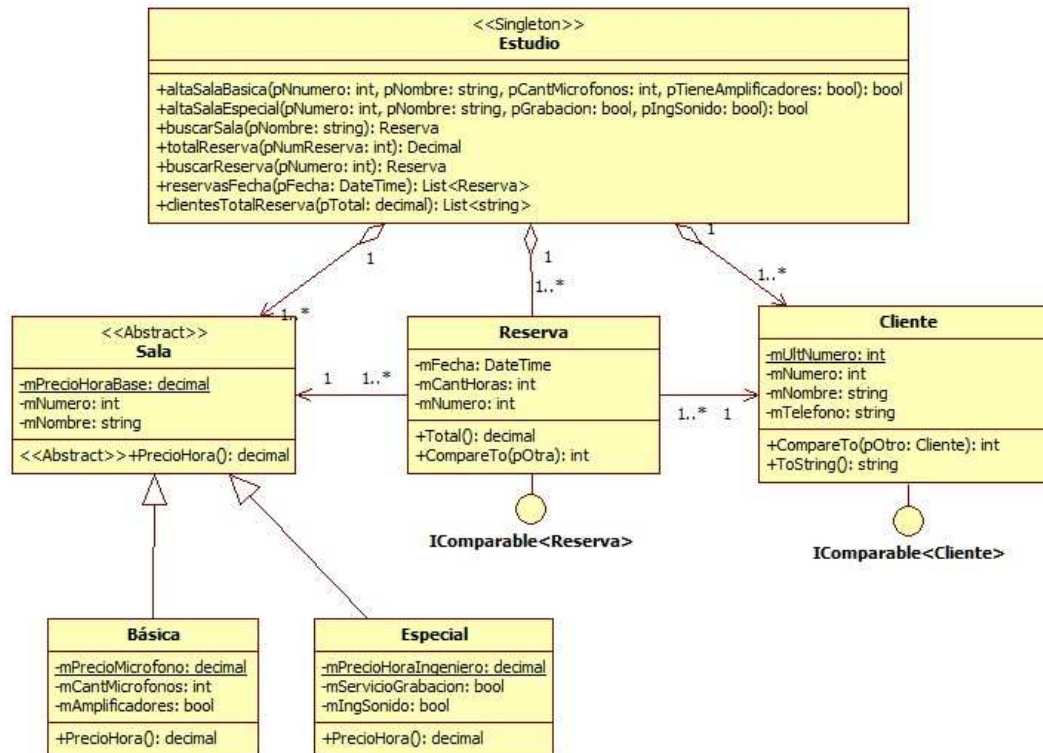


EVALUACION	<b>SOLUCION EXAMEN</b>	GRUPO	TODOS	FECHA	06/08/2013
MATERIA	PROGRAMACIÓN 2				
CARRERA	AP/ATI/APW				
CONDICIONES	<p>- Puntos: 100</p> <p>- Duración: 3 Horas</p> <p>- Sin material</p> <p>- Otros : No escriba la hoja de la letra</p> <p>Consultas solamente sobre interpretación de letra y sintaxis específica del lenguaje.</p> <p>Numerar las hojas entregadas.</p> <p><b>Indicar nombre del docente del curso en primera hoja del examen</b></p>				

**Solución:**

Parte 1:



Parte 2:

**2.b**

EN CLASE ESTUDIO:

```
public decimal totalReserva(int pNumReserva)
{
    decimal total = 0;
    Reserva aux = this.buscarReserva(pNumReserva);

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

    return total;
}

public Reserva buscarReserva(int pNumero)
{
    foreach (Reserva unaReserva in this.mColReservas)
    {
        if (unaReserva.Numero == pNumero)
        {
            return unaReserva;
        }
    }

    return null;
}
```

EN CLASE RESERVA:

```
public Decimal Total()
{
    decimal total = 0;

    if (this.Sala != null)
    {
        total = this.Sala.PrecioHora() * this.CantHoras;
    }

    return total;
}
```

EN CLASE SALA:

```
public abstract class Sala
{
    ...

    public abstract decimal PrecioHora();
}
```

EN CLASE BASICA:

```
public class Basica : Sala
{
    ...

    public Decimal PrecioHora()
    {
        decimal precio =
            Sala.PrecioHoraBase + this.CantMicrofonos * Básica.PrecioMicrofono;

        if (Amplificadores)
        {
            precio = precio * 1.10M;
        }

        return precio;
    }
}
```

EN CLASE ESPECIAL:

```
public class Especial : Sala
{
    ...

    public Decimal PrecioHora()
    {
        decimal precio = Sala.PrecioHoraBase;

        if (IngSonido)
        {
            precio += Especial.PrecioHoraIngeniero;
        }

        if (ServicioGrabacion)
        {
            precio = precio * 1.05M;
        }

        return precio;
    }
}
```

2.c:

EN CLASE ESTUDIO:

```
public List<Reserva> reervasFecha(DateTime pFecha)
{
    List<Reserva> reservas = new List<Reserva>();

    foreach (Reserva unaReserva in this.mColReservas)
    {
        if (unaReserva.Fecha.Date == pFecha.Date)
        {
            reservas.Add(unaReserva);
        }
    }

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

EN CLASE RESERVA:

```
public class Reserva : IComparable<Reserva>
{
    ...

    public int CompareTo(Reserva pOtra)
    {
        if (this.Cliente != null && pOtra.Cliente != null)
        {
            return pOtra.Cliente.Nombre.CompareTo(this.Cliente.Nombre);
        }

        return 0;
    }
}
```

2.d:

EN CLASE ESTUDIO:

```
public List<string> clientesTotalReserva(decimal pTotal)
{
    List<Cliente> clientes = new List<Cliente>();
    List<string> listado = new List<string>();

    foreach (Reserva unaReserva in this.mColReservas)
    {
        if (unaReserva.Total() >= pTotal)
        {
            if (!clientes.Contains(unaReserva.Cliente))
            {
                clientes.Add(unaReserva.Cliente);
            }
        }
    }

    clientes.Sort();

    foreach (Cliente unCliente in clientes)
    {
        listado.Add(unCliente.ToString());
    }

    return listado;
}
```

EN CLASE CLIENTE:

```
public class Cliente : IComparable<Cliente>
{
    ...

    public int CompareTo(Cliente pOtro)
    {
        return this.Numero - pOtro.Numero;
    }

    public override string ToString()
    {
        return this.Numero + " - " + this.Nombre + " - " + this.Telefono;
    }
}
```