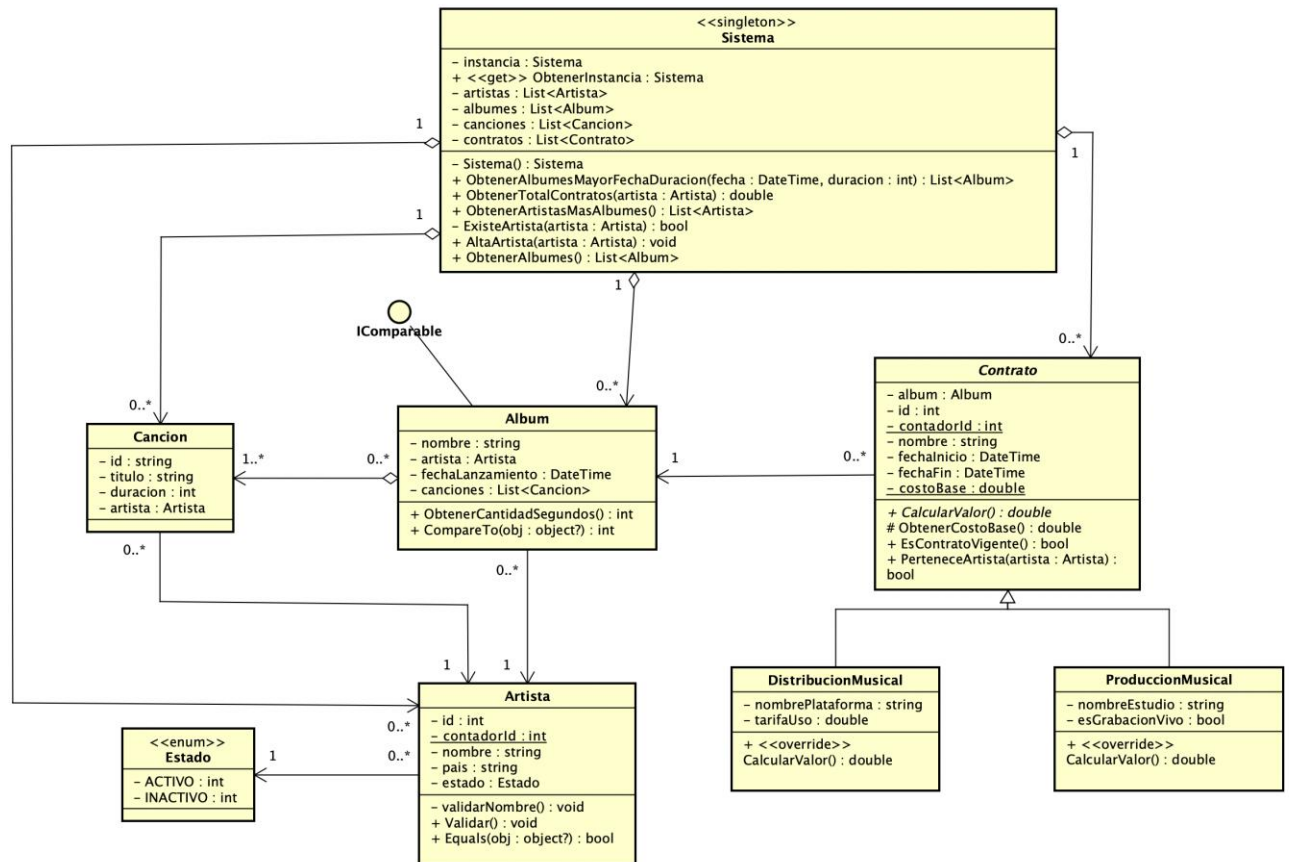


1)



2 a)

//En clase Sistema

```

public List<Album> ObtenerAlbumesMayorFechaDuracion(DateTime fecha, int duracion) {
    List<Album> retorno = new List<Album>();
    foreach(Album a in Albumes) {
        if (a.FechaLanzamiento > fecha && a.ObtenerCantidadSegundos() > duracion) {
            retorno.Add(a);
        }
    }
    retorno.Sort();
    return retorno;
}

```

```
}
```

**// En clase Album**

```
public class Album : IComparable {  
    public int CompareTo(object? obj) {  
        Album comparar = (Album) obj;  
        return comparar.FechaLanzamiento.CompareTo(this.FechaLanzamiento);  
    }  
}
```

**2 b)**

**// En clase Sistema**

```
public double ObtenerTotalContratos(Artista artista) {  
    If (!ExisteArtista(artista)) {  
        throw new Exception("El artista no existe en el sistema.");  
    }  
    double total = 0;  
    foreach(Contrato c in Contratos) {  
        if (c.PerteneceArtista(artista) && c.EsContratoVigente()) {  
            total += c.CalcularValor();  
        }  
    }  
    return total;  
}  
  
private bool ExisteArtista(Artista artista) {  
    return Artistas.Contains(artista);  
}
```

**// En clase Contrato abstracta**

```

public abstract class Contrato {

    public abstract double CalcularValor();

    protected double ObtenerCostoBase() {

        return this.Album.ObtenerCantidadSegundos() * 0.05 + CostoBase;

    }

    public bool EsContratoVigente() {

        DateTime fechaAhora = DateTime.Now();

        return fechaAhora >= this.FechaInicio && fechaAhora < this.FechaFin;

    }

    public bool PerteneceArtista(Artista artista) {

        return this.Album.Artista.Equals(artista);

    }

}

```

#### // En clase ProduccionMusical

```

public class ProduccionMusical : Contrato {

    public override double CalcularValor() {

        double total = ObtenerCostoBase();

        if (this.EsGrabacionVivo) {

            total *= 1.2;

        }

        return total;

    }

}

```

#### // En clase DistribucionMusical

```

public class DistribucionMusical : Contrato {

    public override double CalcularValor() {

        return ObtenerCostoBase() + this.TarifaUso;

    }

}

```

**// En clase Artista**

```
public override bool Equals(object? obj) {  
    Artista comparar = (Artista) obj;  
    return this.Id == comparar.Id;  
}
```

**2 c)**

**// En clase Sistema**

```
public List<Artista> ObtenerArtistasMasAlbumes() {  
    List<Artista> retorno = new List<Artista>();  
    int maximo = 0;  
    foreach(Artista artista in Artistas) {  
        int cantidadAlbumesArtista = 0;  
        foreach(Album album in Albumes) {  
            if (album.Artista.Equals(artista)) {  
                cantidadAlbumesArtista++;  
            }  
        }  
        if (cantidadAlbumesArtista > maximo) {  
            retorno.Clear();  
            retorno.Add(artista);  
            maximo = cantidadAlbumesArtista;  
        }  
        if (cantidadAlbumesArtista == maximo) {  
            retorno.Add(artista);  
        }  
    }  
    return retorno;  
}
```

```
}
```

### 3 a)

**// En clase ArtistaController, carpeta Controllers**

```
public IActionResult MostrarFormularioAlta() {  
    return View();  
}  
[HttpPost]  
public IActionResult Alta(Artista artista) {  
    try {  
        Sistema.ObtenerInstancia.AltaArtista(artista);  
        TempData["Ok"] = true;  
    } catch (Exception e) {  
        TempData["ErrorMensaje"] = e.Message;  
    }  
    return RedirectToAction("MostrarFormularioAlta");  
}
```

**// En vista MostrarFormularioAlta, carpeta Views, subcarpeta Artista**

@using Dominio

<h1>Alta de Artistas</h1>

<form action="/Artista/Alta" method="post">

<div class="form-group">

<label class="form-label">Nombre</label>

<input type="text" id="nombre" class="form-control" name="Nombre" />

</div>

<div class="form-group">

<label class="form-label">Pais</label>

```

        <input type="text" id="pais" class="form-control" name="Pais" />
    </div>
    <div class="form-group">
        <label class="form-label">Estado</label>
        <select id="estado" name="Estado" class="form-control">
            @foreach (var estado in Enum.GetValues(typeof(Estado)))
            {
                <option value="@estado">@estado</option>
            }
        </select>
    </div>
    <div class="form-group">
        <button type="submit">Crear</button>
    </div>
</form>
@if (TempData["Ok"] != null)
{
    <div class="alert alert-success" role="alert">
        Artista dado de alta correctamente
    </div>
}
@if (TempData["Error"] != null)
{
    <div class="alert alert-danger" role="alert">
        @TempData["Error"]
    </div>
}

```

**//En clase Sistema**

```

public void AltaArtista(Artista artista) {
    artista.Validar();
}

```

```

        if (ExisteArtista(artista)) {
            throw new Exception("El artista ya se encuentra ingresado en el sistema");
        }
        Artistas.Add(artista);
    }

```

**//En clase Artista**

```

public Artista() {
    this.Id = Contador++;
    //demás asignaciones
}

```

**3 b)**

**// En clase AlbumController, carpeta Controllers**

```

public IActionResult ObtenerTodos() {
    List<Album> albumes = Sistema.ObtenerInstancia.ObtenerAlbumes();
    return View(albumes);
}

```

**// En vista ObtenerTodos, carpeta Views, subcarpeta Album**

@using Dominio

@model List<Album>

<h1>Lista de albumes</h1>

<table class="table">

<thead>

<tr>

<th>Nombre</th>

<th>Fecha de lanzamiento</th>

```

        <th>Nombre del artista</th>
    </tr>
</thead>
<tbody>
    @{
        foreach (Album a in Model)
        {
            <tr>
                <th>@a.Nombre</th>
                <td>@a.FechaLanzamiento</td>
                <td>@a.Artista.Nombre</td>
            </tr>
        }
    }
</tbody>
</table>

```

**//En clase Sistema**

```

public List<Album> ObtenerAlbumes() {
    return Albumes;
}

```