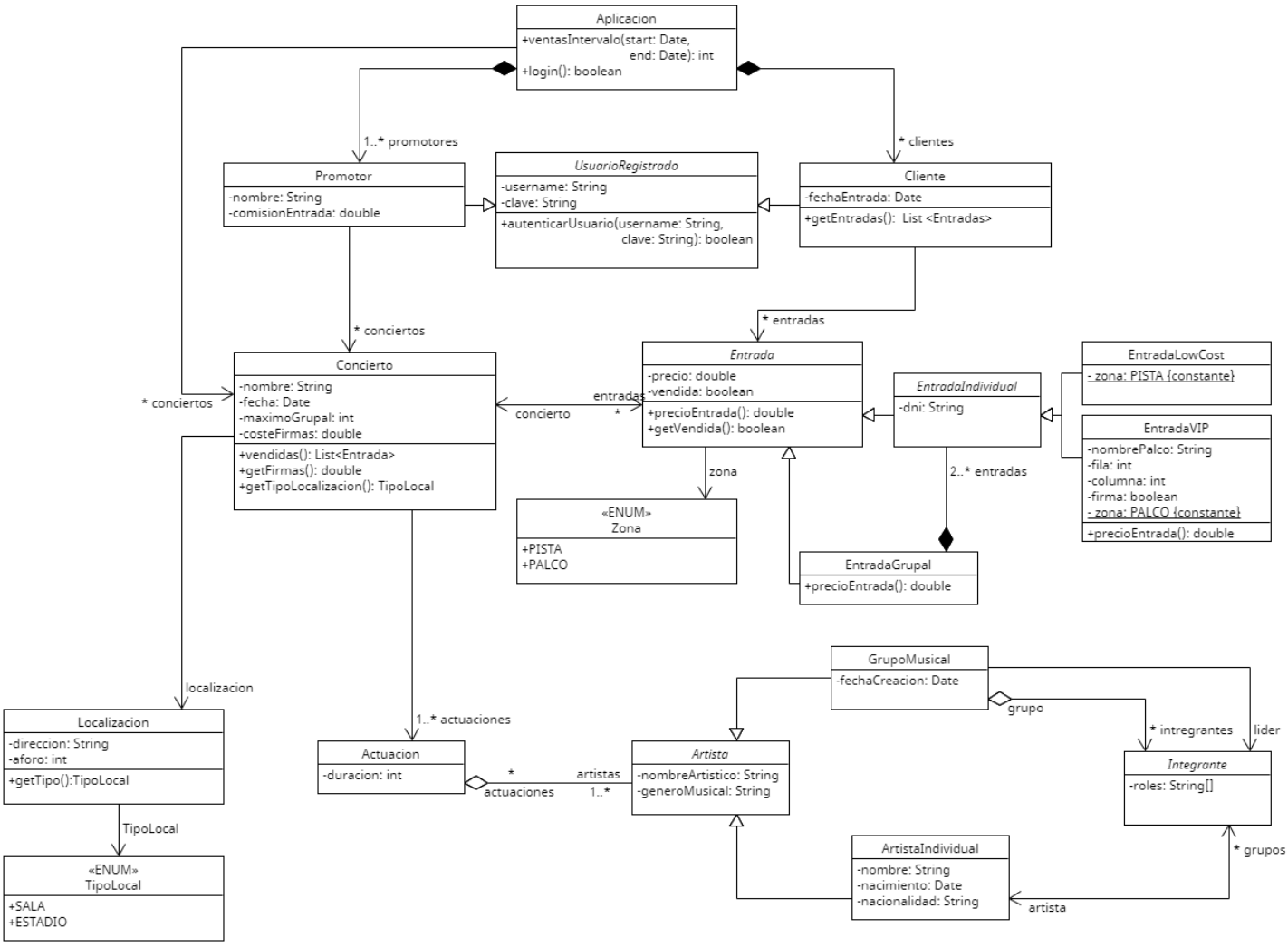

La aplicación está construida por promotores y clientes, ambos heredan de una clase UsuarioRegistrado que es la que guarda el login y contraseña de cada un de estas, además esta clase abstracta se encarga de autenticar al usuario. Los promotores están relacionados con los conciertos, ya que estos les dan de alta, proporcionando los datos de estos así como los datos relacionados con sus entradas. La localización del concierto se guarda en una clase aparte que se relaciona con este, donde se guardan todos los datos de la localización, incluyendo el TipoLocal que influye al precio de la entrada.

Los conciertos tienen al menos una actuación, la cual está formada de artistas, la clase Artista es una clase abstracta debido a que los artistas tienen que ser una de las dos subclases ArtistasIndividuales o GrupoMusical. El GrupoMusical a su vez está formado por integrantes, que son ArtistasIndividuales con uno o varios roles, así como la necesidad de tener un integrante líder, esto se marca teniendo 2 relaciones separadas, una para los integrantes normales y otra para el líder.

Por otro lado los conciertos tienen entradas, estas entradas son una clase abstracta, las entradas pueden ser individuales o grupales. Las individuales a su vez están formadas por

entradas low cost (EntradaLowCost) o entradas VIP (EntradaVIP). Por otro lado, las entradas grupales están formadas de al menos 2 entradas individuales.

> c) Proporciona pseudocódigo (o código Java) para el/los métodos que calculan el precio de las entradas (0.5 puntos).

En Entrada el código para el método que calcula el precio será:

```
public double precioEntrada()
    {
        if(concierto.getTipoLocalizacion() == ESTADIO)
            return precioBase + precioBase * 0.02;
        return precioBase;
    }
```

Donde se revisa si el concierto se realiza en un estadio para añadir un 2% extra.

En EntradaLowCost no es necesario sobrescribir dicho método, ya que no se realiza ningún cambio a los precios.

En EntradaVIP es necesario añadir un extra en caso de que se disponga de firmas de autógrafos, por tanto se sobreescribe el método de la siguiente forma

```
@Override
    public double precioEntrada()
    {
        if(firma)
            return super.precioEntrada() + super.getConcierto().getFirmas();
        return super.precioEntrada();
    }
```

Donde se utilizan getters para conseguir el extra de firmas en dicho concierto y se añade al precio original en caso de que la firma sea true. En otro caso el precio se mantiene normal.

Finalmente el precio de las entradas grupales es la suma de entradas y un descuento de entre 2 y 15 dependiendo del número de entradas. Por tanto se tiene que sobrescribir el código de cálculo del precio de la entrada.

```java
@Override
   public double precioEntrada()
   {
       double precioTotal = 0.0;
       int i = 0;
       for(EntradaIndividual entrada : entradas)
       {
           precioTotal += entrada.precioEntrada();
           i += 1;
       }

       if(i < 15)
           return precioTotal - precioTotal*i/100;
       return precioTotal - precioTotal*0.15;
   }
```

Como se observa el código para el cálculo de entrada calcula primero el precio total de sumar todas las entradas individuales y posteriormente aplicar el descuento, asegurándose de no aplicar un descuento superior de 15%.