**Diagrama de Clases - Spotify UCSP**

1. Clase Usuario
   1. Atributos
      1. Id\_usuario(integer) : Este atributo representa un identificador particular para cada usuario del sistema, con este se identificará al usuario en particular para tenerlo bajo métodos del sistema.
      2. contraseña(string) :Este atributo que representa una consigna privada para cada usuario del sistema, con este el usuario tendrá acceso a s cuenta particular del sistema.
      3. fecha\_creacion(date) :Este atributo indica la fecha de creación del usuario en el calendario gregoriano.

* + 1. estado(bool) :Este atributo plasma cuando algún usuario tomó la decisión de eliminar su cuenta.

* + 1. List<seguidores> :En este atributo se representa una lista de usuarios, los cuales siguen a este usuario por afinidad de gustos o razones de distinto índole.
    2. List<seguidos> :En este atributo se representa una lista de usuarios, los cuales este usuario sigue por afinidad de gustos o razones de distinto índole.
  1. Relaciones:
     1. Usuario-Persona: Persona hereda a usuario sus atributos generales.
     2. Usuario-Valoracion: Usuario puede valorar una o varias canciones, pero una valoración debe pertenecer a un Usuario.
     3. Usuario-Playlist: Usuario puede crear una varias playlist, pero una playlist pertenece a un solo Usuario.
     4. Usuario-Usuario: Usuario sigue a unos o varios usuarios.

* 1. Responsabilidades:
     1. iniciar\_sesión() :Este método está encargado de validar el ingreso a la aplicación de la Persona que quiere identificarse como Usuario.
     2. crear\_playlist() :Este método está implementada para crear una lista de canciones a gusto del Usuario.
     3. elimina\_playlist() : Este método está implementada para que el usuario elimine alguna playlist.

* + 1. desactivar\_cuenta() :Con este método el Usuario tendrá la capacidad de desactivar su cuenta.
    2. activar\_cuenta() :Con este método el Usuario tendrá la capacidad de activar su cuenta si en algun momento desactivo su cuenta.
    3. seguir\_usuario() :Con este método el Usuario podrá seguir a otros Usuarios de su preferencia.
    4. dejar\_de\_seguir\_usuario() :Con este método el Usuario podrá dejar de seguir a otros Usuarios de su preferencia.
    5. valorar\_cancion() Con este método el Usuario podrá otorgar una validación a sus canciones preferidas.

1. Clase Administrador
   1. Atributos:
      1. Id\_administrador(integer) :Con este atributo se representa un identificador particular para cada Administrador del sistema, con este se identificará al Administrador en particular para tenerlo bajo métodos del sistema.
      2. contraseña(string) :Este atributo que representa una consigna privada para cada Administrador del sistema, con este el usuario tendrá acceso a su cuenta particular de la aplicación.
      3. fecha\_creación(date) :Este atributo indica la fecha de creación del Administrador en el calendario gregoriano.
      4. estado(bool) :Este atributo plasma cuando algún usuario tomó la decisión de eliminar su cuenta.
   2. Relaciones:
      1. Administrador-Persona: Persona hereda a Administrador sus atributos generales.
   3. Responsabilidades:
      1. login() :Este método está encargado de validar el ingreso a la aplicación de la Persona que quiere identificarse como Administrador.
      2. habilitar\_cancion() :Con este método el Administrador podrá habilitar canciones subidas por el Artista.
      3. habilitar\_album() :Con este método el Administrador podrá habilitar Albums subidas por el Artista.
      4. desactivar\_usuario() :Con este método el Administrador podrá deshabilitar Usuarios.
      5. crear\_ranking() :Con este método el Administrador podrá crear o modificar el ranking de cansiones.
      6. desactivar\_cancion() :Con este método el Administrador podrá deshabilitar canciones.
      7. desactivar\_album() Con este método el Administrador podrá deshabilitar Albums.
      8. habilitar\_usuario() Con este método el Administrador podrá habilitar Usuarios.
2. Clase Artista
   1. Atributos:
      1. Id\_artista(integer): Este atributo representa un identificador particular para cada artista del sistema, con este se identificará al artista en particular para tenerlo bajo métodos del sistema.
      2. nombre(string): Este atributo representa un nombre en particular que tiene cada artista.
      3. estado(bool):: Este atributo representa el estado del artista y que si en ese momento está activo o no en el sistema.
      4. fecha(date) Este atributo nos indica la fecha del cual el artista ingresó al sistema.
      5. contraseña(string): Este atributo que representa una consigna privada para cada artista del sistema, con este el artista tendrá acceso a s cuenta particular del sistema.
   2. Relaciones:
      1. Artista-Persona: Persona hereda sus atributos generales.
      2. Artista-Álbum: Artista puede tener uno o más álbumes, pero un Álbum solo puede pertenecer a un Artista.
   3. Responsabilidades:
      1. crear\_album(): Este método está encargado de la creación de álbumes por parte del artista.
      2. cargar\_canciones(): Este método está encargado de la carga de canciones a un álbum determinado por parte del artista.
3. Clase Álbum
   1. Atributos:
      1. Id\_album(integer): Este atributo representa un identificador particular para cada álbum del sistema, con este se identificará al álbum en en particular para tenerlo bajo métodos del sistema.
      2. nombre(string): Este atributo representa un nombre en particular que tiene cada Álbum.
      3. Id\_artista(integer): Este atributo que representa una consigna privada para cada usuario del sistema, con este el usuario tendrá acceso a s cuenta particular del sistema.
      4. estado(bool): Este atributo plasma cuando algún álbum está disponible para reproducir o no.
      5. fecha\_de\_creación(date): Este atributo indica la fecha de publicación del álbum.
   2. Relaciones:
      1. Albun-Artista: Un Álbum pertenece a un solo Artista
   3. Responsabildades:
      1. mostrar\_canciones(): En este método nos muestra las canciones que contiene determinado album.
4. Clase Valoración
   1. Atributos:
      1. Id\_valoracion(integer): Este atributo representa un identificador particular para valorar una canción, con este se identificará a una valoración en particular para tenerlo bajo métodos del sistema.
      2. Id\_usuario(integer): Este atributo representa un identificador particular para cada usuario del sistema, con este se identificará al usuario en particular para tenerlo bajo métodos del sistema.
      3. Id\_cancion(integer): Este atributo representa un identificador particular para cada canción en particular, con este se identificará a una canción en particular para tenerlo bajo métodos del sistema.
      4. num\_reproducciones(integer): Este atributo nos muestra las veces que fue reproducida al seleccionar una canción en particular.
   2. Relaciones:
      1. Valoración-Usuario: Una valoración pertenece a un solo usuario
      2. Valoración-Cancion: Una valoración califica a una o varias canciones
   3. Responsabilidades:
      1. actualizar(): En este método actualiza el registro de las canciones reproducidas con sus respectivas valoraciones.
5. Clase Playlist
   1. Atributos:
      1. Id\_playlist:integer Este atributo representa un identificador particular para cada PlayList de la aplicación musical creada por diferentes personas, con este se identificará al usuario en particular para tenerlo bajo métodos del sistema.
      2. nombre:integer Con este atributo la PlayList se identificara hacia las diferentes entidades de la aplicación.
      3. Id\_usuario:integer Con este atributo se identificará la procedencia de la PlayList.
      4. estado:bool Con este atributo se indicará si la PlayList está borrada o activa.
      5. fecha:date En este atributo mostrará la fecha de creación realizada por el Usuario o Administrador.
      6. List<Cancion> Este atributo almacenara la lista de canciones que contendrá la PlayList.
   2. Relaciones:
      1. Plasylist-Usuario: Una playlist pertenece a un solo solo usuario
      2. Plasylist-Canción: Una playlist contiene ninguna o varias canciones
   3. Responsabilidades:
      1. mostrar\_canciones() Con este método se mostrará el contenido musical que contendrá la PlayList hacia todas las entidades de la aplicación.
      2. reproducir() Este método tendrá la capacidad reproducir el contenido musical almacenada en la PlayList.
      3. agregar\_canciones() Con este método se podrá agregar canciones a la PlayList por parte del Usuario o Administrador que crearon dicha PlayList.
      4. eliminar\_canciones() Con este método se podrá descartar las canciones que no se deseen en la PlayList.
6. Clase Persona
   1. Atributos:
      1. Id\_pesona:integer Este atributo representa un identificador particular para cada Persona del sistema, con este se identificará a la Persona en particular para tenerlo bajo métodos del sistema.
      2. email:string Este atributo almacenara el correo electrónico de la Persona.
      3. telefono:varchar Este atributo almacenara el teléfono de la Persona.
   2. Relaciones:
      1. Persona-Administrador: Persona hereda sus atributos a administrador
      2. Persona-Usuario: Persona hereda sus atributos a Usuario.
      3. Persona-Arista: Persona hereda sus atributos a artista
7. Clase Canción
   1. Atributos:
      1. Id\_cancion:integer Este atributo representa un identificador particular para cada Cancion del sistema, con este se identificará a la Canción en particular para tenerlo bajo métodos del sistema.
      2. nombre:string Con este atributo la Canción se identificara hacia las diferentes entidades de la aplicación.
      3. estado:bool Con este atributo se indicará si la Canción está borrada o activa.
      4. fecha\_de\_creación(date): Este atributo indica la fecha de publicación de la Canción.
      5. valoración(date): Este atributo indica el nivel de aceptación hacia todas las entidades de la aplicación.
   2. Relaciones:
      1. Cancion-Album: Relación de composición entre la clase Album y Cancion
      2. Cancion-Valoración: Canción tiene una o muchas valoraciones
      3. Canción-Playlist: Canción pertenece a una o varias playlist.
   3. Responsabilidades:

* + 1. reproducir() Este método tendrá la capacidad reproducir el contenido musical de la canción.
    2. obtener\_informacion() Con este método se mostrará la información inscrita en la canción.