



Universidad de
los Andes

Modelos de Clases

Grupo 2

Programación Orientada a objetos.

Integrantes:

Antonia Carrizo

Álvaro Céspedes

Matías Cornejo

Profesores:

Andrés Howard

Carlos Díaz

9 de junio, 2020

Clase Actor:

- Atributos:
 - `List<Video> movies` : Contiene todas las películas en las que ha participado el actor.
 - `List<string> awards` : Contiene los premios que ha ganado el actor.
 - `int yearsActive`: años que lleva el actor activo
- Métodos:
 - No tiene métodos.

Clase Álbum:

- Atributos:
 - `string nameAlbum`: Nombre del álbum.
 - `string gender`: género al que pertenece el álbum.
 - `string producer`: productor que participó en el álbum.
 - `List<Songclass> listSongs`: contiene las canciones del álbum.
 - `Singer singer`: cantante del álbum.
 - `DateTime publicationDate`: año de lanzamiento del álbum.
- Métodos:
 - No tiene métodos.

Clase DataBase:

- Atributos:
 - No tiene atributos.
- Métodos:
 - `ChangePassword`: Para cambiar la contraseña
 - `ChangePlan`.
 - `ChangeTargetPay`.
 - `Login`.
 - `AddUser`.
 - `GetData`.

Clase Director:

- Atributos:
 - `List<Video> movies`: contiene las películas que ha realizado.
 - `List<string> awards`: contiene los premios que ha ganado.
 - `int yearsActive`: años de carrera activo.
- Métodos:
 - No tiene métodos.

Clase Person

- Atributos:
 - `string name`: nombre.
 - `int age`: edad.
 - `lastname`: apellido.
 - `gender`: género.
 - `nationality`: nacionalidad.
 - `occupation`: ocupación.
- Métodos:
 - No tiene métodos.

Clase PlaylistSpotifai

- **Atributos:**
 - `string nombre`;
 - `List<SongClass> cancionesPlaylist`;
 - `string privacidad`;
- **Métodos:**
 - No tiene.

Clase PlaylistVideoEmptyClass

- **Atributos:**
 - `string name`;
 - `List<Video> videosPlaylist`;
 - `string privacidad`;
- **Métodos:**
 - No tiene métodos.

Clase Profile

- **Atributos:**

```

string nameProfile;
string profileType;
List<string> pleasuresMusic;
List<string> pleasuresMovies;
List<PlaylistSpotifai> playlistCanciones;
List<PlaylistVideoEmptyClass> playlistVideos;
List<SongClass> playlistFavoritasCanciones;
List<Video> playlistFavoritasVideos;

```

- **Métodos:**

Clase ProfileManagment

- **Atributos:**

No tiene atributos.

- **Métodos:**

```

public bool DeleteProfile():
public bool ChangeProfile():
public bool AddProfile():
public bool SeeProfile()

```

Clase ProfileUser

- **Atributos:**

```

List<ProfileUser> followedProfile;
List<ProfileUser> followers;
List<PlaylistVideoEmptyClass> followedPlaylistVideo;
List<PlaylistSpotifai> followedPlaylistSongs;
List<Singer> followedSingers;
List<Actor> followedActors;
List<Director> followedDirectors;
List<Album> followedDisk;
List<SongClass> downloadsSongs;
List<SongClass> likesSongs;
List<Video> likesVideos;

```

- **Métodos:**

No tiene métodos.

Clase Serialization

- **Atributos:**

No tiene.

- **Métodos:**

```

public void Serialize<Object>(Object list, Stream stream)
public Object Deserialize<Object>(Stream stream) where Object : new()

```

private static Object CreateInstance<Object>() where Object : new()

Clase Singer

- **Atributos:**

- List<SongClass> songs;
 - List<Album> album;
 - List<string> awards;
 - string voiceType;
 - string genders;
 - int yearsActive;
 - List<string> discographic;

- **Métodos:**

- No tiene métodos.

Clase SongClass

- **Atributos:**

- private string composer;
 - private Singer singer;
 - private Album album;
 - private string format;
 - private List<SongClass> canciones;
 - string gender;
 - string publicationYear;
 - string title;
 - double duration;
 - int memory;
 - string study;
 - string keyword;
 - int evaluation;
 - int resolution;
 - string url;
 - int nReproduction;
 - int likes;

- **Métodos:**

Clase User:

- **Atributos:**

- string mail;
 - string paymentInfo;
 - string plan;
 - DateTime registrationDate;

```
string nameUser;  
string password;  
int numPhone;  
List<User> usuarios;  
List<Profile> profiles;
```

- **Métodos:**

```
public override string ToString(): Para que cuando la clase sea llamada se  
muestra por orden de:  
Title + " " + Composer + " " + Gender + " " + PublicationYear + " " + Duration +  
" " + Memory + " " + Study + " " + Keyword + " " + Singer.Name + " " +  
Album.NameAlbum;
```

Clase Video:

- Atributos:

```
string description: Descripcion del video  
Actor mainActor: Actor del video  
Director director: Director del video  
string format: formato, '.mp4'  
List<Video> videos: lista de videos  
string gender: Categoría a la que pertenece  
string publicationYear:  
string title:  
double duration:  
int memory:  
string study:  
string keyword:  
int evaluation:  
int resolution:  
string url:  
int nReproduction:  
int likes:
```

- Métodos:

```
public override string ToString(): Para que cuando la clase sea llamada se  
muestra por orden de:  
Title + " " + Director.Name + " " + Gender + " " + PublicationYear + " " +  
Duration + " " + Memory + " " + Study + " " + Keyword;
```

Cambios realizados:

- Eliminamos las clases: Server, SearchClass, RegisterEventArgs, PrintAndReceive, MediaPlayer, ChangePasswordEventArgs.
Motivo: No las utilizamos en el programa.
- Agregamos la clase Serialization, donde se incluyen los métodos para serializar y deserializar las clases necesarias.
Motivo: llamando a esta clase nos ahorramos el proceso de escribir las funciones para serializar y deserializar cada vez que se requiera.
- No pondremos la confirmación del mail ni telefono.
Motivo: No podemos confirmar realmente si el usuario ingresa un teléfono y email real.
- La nacionalidad del usuario será solo de América Latina.
Motivo: Al pedir la nacionalidad en el registro, nos ahorramos poner todos los países existentes.
- En la información de pago no confirmaremos si la tarjeta es real.
Motivo: No tenemos cómo confirmar si la tarjeta realmente existe.

- En vez de 5 perfiles que tiene el plan familiar, ahora tendra solo 4.

Motivo: Simplificar la interfaz del usuario.

- Incluimos las siguientes funcionalidades:

- Se pueden importar canciones y videos.
- Se guarda la información de los archivos multimedia.
- Es posible hacer búsquedas considerando uno o más criterios.
- Se pueden crear playlist de videos o música considerando el nivel de visibilidad de estas.

Motivo: No las implementamos en la entrega 3.