

Supuestos Entrega 3

Supuestos:

- El admin no tiene acceso a la contraseña de un usuario.
- El Admin tiene su propio menú(Usuario: Admin Contraseña: Admin123) y no es considerado como un usuario.
- El Usuario al registrarse automáticamente es gratuito y público. Esto lo puede editar en su perfil si quiere ser premium o privado.
- Todos los usuarios premium tienen la posibilidad de ser artistas de la app, es decir, tendrán su perfil de artista en donde puede subir contenido y ver su álbum. Además los usuarios pueden seguir artistas (**Idea Innovadora**)
- El álbum de los artistas es donde se muestra el contenido que a subido a la aplicación.
- El Profile Free tendrá acceso al contenido de manera limitada, es decir, no podrá ser artista.
- El usuario tendrá una lista para con las canciones/vídeos más escuchadas/vistos en la plataforma (**Idea innovadora**)
- Cada canción y video tiene un nombre único que permite identificarlo.
- Para reproducir contenido tiene que ser mediante el la búsqueda(Artista o media) o desde una playlist. En caso que sea Artista podría reproducir canciones/Videos desde su álbum.
- Al eliminar un Artista que también es un usuario, de la lista de seguidos del usuario. Se eliminan ambos.
- Las listas de “Música favorita” y “Videos Favoritos” serán creados por default y con la propiedad de ser privados.
- Al buscar las playlist de otro usuario solo se mostrarán las playlist que han sido creadas de forma pública.
- La privacidad de la playlist se define solo en el momento de ser creadas, osea que una vez creada no es posible cambiar su privacidad.
- El usuario que sea privado no podrá crear playlist públicas.

Clases

Model

- Artist:
 - Es objeto Artista, con todos los métodos y atributos correspondientes.
 - Este en comparación con la entrega anterior, fue modificado por completo dado que fue utilizado por las clases song, video y playlist. Lo cual para esta entrega esperamos todo lo anteriormente mencionado a sus propias clases.
- PlaylistSong:
 - En la entrega anterior el usuario guardaba sus canciones en una lista de listas.
 - Esta clase fue creada para que el objeto usuario tenga una lista de objetos PlaylistSong.
 - Esta clase guarda una lista de canciones y el nombre de la playlist.

- PlaylistVideo:
 - En la entrega anterior el usuario guardaba sus canciones en una lista de listas.
 - Esta clase fue creada para que el objeto usuario tenga una lista de objetos PlaylistVideo.
 - Esta clase guarda una lista de videos y el nombre de la playlist.
- Song:
 - Clase encargada de mantener toda la información de una canción (reproducciones, calificaciones, cantante, entre otros), no hubo cambios con la tarea pasada en esta clase.
- User:
 - Es el objeto usuario con todos los métodos y atributos correspondientes.
 - En comparación con la entrega anterior no tuvo cambios.
- Video:
 - Clase encargada de mantener toda la información de un video (reproducciones, calificaciones, actores, entre otros), no hubo cambios con la tarea pasada en esta clase.

●

Controllers:

En comparación con la entrega 2, los controllers reemplazan a la clase "AppSystem" la cual era la encargada de la relacionarse con todas las clases y el usuario.

- ArtistController:
 - Clase creada para usarse como conexión entre la vista (AppForm) de un programa y los modelos como el objetos "artistas", "User" creados. Encargada básicamente de controlar la información de un artista. Realizándose la unión por medio de eventos y delegados.
 - Contiene todos los métodos para buscar artistas.
- SongController:
 - Clase creada para usarse como conexión entre la vista (AppForm) de un programa y los modelos como los objetos "Song" creados. Encargada básicamente de controlar la información de una canción. Realizándose la unión por medio de eventos y delegados.
 - Contiene todos los métodos para buscar canciones
- VideoController:
 - Clase creada para la interacción entre la clase de AppForm y la clase Video mediante eventos
 - Los métodos de Vídeo se utiliza mucho en esta parte ya que aquí se encontraba una lista con todas las canciones.
- Usercontroller:
 - Clase creada para la interacción entre la clase de AppForm y la clase User mediante eventos

- Tiene una lista con todos los usuarios por lo que muchos métodos de User se utilizan en esta clase al activar ciertos eventos.

Proyecto15WF:(View)

Ambas clases no existían en la entrega anterior, y son los pilares del programa winforms. Por otro lado rempazan a AppSystem la cual una de sus funciones era la interacción con el usuario.

- AppForm:
 - Controladora del diseño
- AppForm[Diseño]:
 - Diseño gráfico de la aplicación

CostumEventsArgs:

Librería de clases encargada del contenido enviado en los eventos.

- AddplaylistEventsArgs:
- ArtistInfoEventsArgs:
- ChangelImageEventsArgs:
- ChangePasswordEventsArgs:
- GetArtistEventsArgs:
- GetUserPlaylistEventsArgs:
- LodingEventsArgs:
- MailEventsArgs:
- MandarclasificacionEventsArgs:
- RegisterEventsArgs:
- ReturnsongEventsArgs:
- ReturnVideoEventsArgs:
- SearchingSongorVideoEventsArgs:
- SearchUserEventsArgs:
- SelectSongEventsArgs:
- SelectVideoEventsArgs:
- SendingArtistInfoEventsArgs:
- SendingsongcharacteristicEventsArgs
- SendingSongEventsArgs
- SendingMupleFiltersEventsArgs
- SendingtyoeaccountEventsArgs
- SendingVideocaracteristicEventsArgs
- SendingVideoEventsArgs
- SongExistEventsArgs
- VideoExistEventsArgs