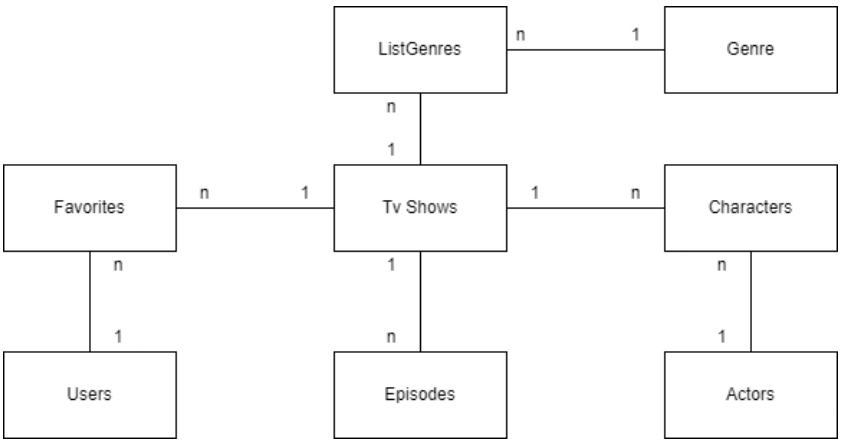
TvShows API:

The approach to this problem started with the modeling of the database for the API. Taking into account the requirements presented by the document, the database was designed as follows:



After this step I started by collecting the data to fill the database with some data. According to my interpretation of the document, I created a Python script (main.py) that makes requests to the “Episodate” API, where the script makes a request to get the TvShows present on the first page (20 TvShows) and then makes requests to obtain data related to the episodes of those TvShows. The data is saved in SQL files to be placed in the database. However, it wasn't possible to obtain the data related to the actors and characters from the TvShows in the “Episodate” API so I used the Mockaroo tool to generate some random data for the Actors and Characters tables.

Regarding the implementation of the API, I chose to use the Database First approach, from the Entity Framework, which consists of creating the database first, using the “bdSQL” file that contains the instructions to create the database and tables. After this step, I put in the database the data that are in the files present in the “dataTvShows” directory.

Using the .NET scaffolding functionality I created the model files of the database tables and using another Entity Framework functionality I created the controllers used in the API.

Regarding the features, the API allows to create users, consult all TvShows, as well as their episodes, search for TvShows by genre and consult their actors. It's also possible to create users, mark TvShows as favorites, consult favorites and remove TvShows from favorites.

The API documentation (the endpoints that can be used) can be viewed when starting the project through the Swagger UI interface. (Ctrl + F5 to run the project)

These endpoints can also be tested through the Swagger interface or through, for example, Postman.