

# PROJETO – TuneCatch

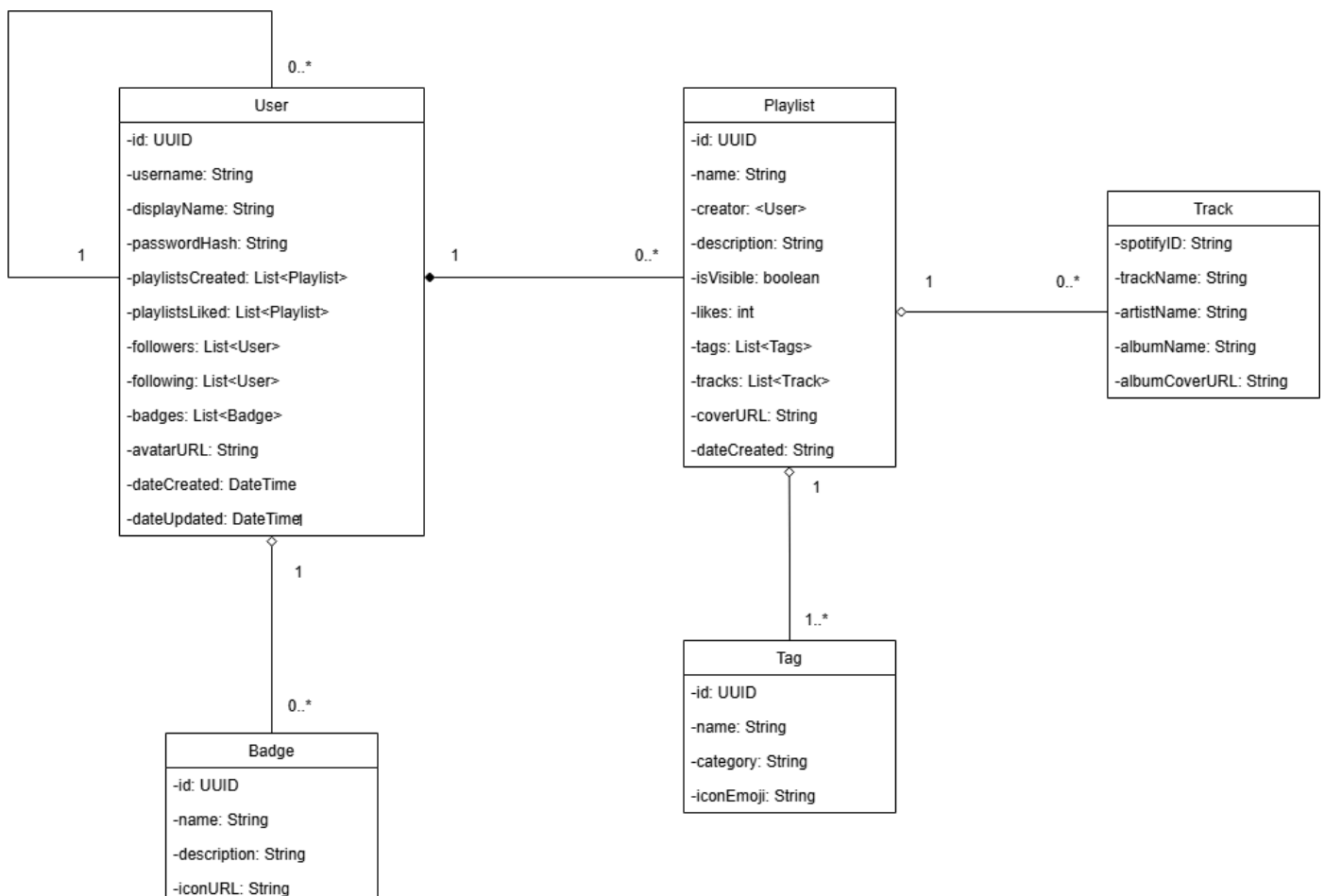
## Elementos do TuneCatch –

O TuneCatch estrutura-se em torno de dois pilares fundamentais: **usuários** e **playlists**.

Os *usuários* são agentes ativos da plataforma – são os perfis que interagem com o conteúdo e entre si. As *playlists*, por sua vez, são as unidades as quais os usuários organizam e compartilham as músicas com base nos contextos emocionais, temporais ou situacionais, atribuindo a elas as *tags*.

No que diz respeito às *músicas*, em vez de armazenar arquivos de áudio ou metadados, cada faixa é referenciada por meio do seu *Spotify Track ID* — um identificador único fornecido pela API do Spotify, o que facilita não apenas o armazenamento, mas também o acesso ao catálogo de músicas disponíveis e sua apresentação para o usuário do sistema.

No entanto, isso não significa que elas serão as únicas classes envolvidas no sistema - haverá classes que as “orbitam” – além da classe `User` e `Playlist`, há as classes `Track`, `Tag` e `Badge`:



A partir deste diagrama de classes, é possível visualizar de forma preliminar como os seus objetos seriam representados dentro sistema da aplicação:

```
user {
  "id": "550e8400-stargazer",
  "username": "@stargazer",
  "displayName": "marco",
  "passwordHash": "$senha$455sadforte$18$as3aEcXYZ",
  "playlistsCreated": ["6ba7b-night-chill-driving", "3c4a5-afternoon-studies"],
  "playlistsLiked": ["1a2b3-warrior-mix"],
  "followers": ["a1b2c3d4-kateologia", "x8l5aad4-sentry"],
  "following": ["d4c3b2a1-kateologia"],
  "badges": ["founder", "imaginative"],
  "avatarURL": "https://armazenamento/avatars/stargazer.jpg",
  "dateCreated": 15-02-2025,
  "dataUpdated": 10-04-2025
}

playlist {
  "id": "6ba7b-night-chill-driving",
  "name": "night chill driving",
  "creator": "550e8400-stargazer",
  "description": "midnight rides",
  "isVisible": true,
  "likes": 1783,
  "tags": ["night", "cold", "drive", "chill"],
  "tracks": ["spotify:track:saigon698", "spotify:track:babyblue145"],
  "coverImageURL": "https://armazenamento/covers/night-chill-11d1.jpg",
  "dateCreated": 25-04-2025
}

track {
  "spotifyID": "spotify:track:saigon698",
  "trackName": "Saigon",
  "artistName": "Luke Hemmings",
  "albumName": "When Facing the Things We Turn Away From",
  "albumCoverURL": "https://spotify-api/image/luke-hemming-album1-cover.jpg"
}

tag {
  "id": "night",
  "name": "Noite",
  "category": "Ocasão",
  "iconEmoji": "🌙"
}

badge {
  "id": "founder",
  "name": "Fundador",
  "description": "Fez parte dos 10 primeiros usuários",
  "iconeURL": "https://armazenamento/badge/founder-5dsp.jpg"
}
```

Das classes citadas, seriam armazenadas no banco de dados todos os objetos criados a partir de cada uma - com exceção daqueles criados a partir da classe *Track*.

A classe *Track* cria instâncias temporárias das músicas com o objetivo de exibir suas informações na aba de pesquisas e nas playlists – as quais armazenam somente seus identificadores (IDs). A partir desses IDs, a classe *Track* permite que todos os dados relevantes das músicas sejam apresentados ao usuário, sem a necessidade de armazená-las no banco, já que são obtidas dinamicamente de fontes externas (API do Spotify).

Estas classes irão se relacionar entre si de três maneiras:

### 1) Autoassociação

- a) Classe *User* associa consigo mesma
  - i) Como um usuário pode seguir outros usuários, é necessário armazenar referências tanto daqueles que ele segue quanto dos que o seguem.

### 2) Agregação

- a) Classe *Badge* é agregada à *Usuário*
  - i) Um perfil de usuário pode possuir *badges*, mas essas *badges* não dependem do ciclo de vida do usuário. Ou seja, mesmo que um usuário seja excluído, as *badges* permanecem no sistema, pois não pertencem exclusivamente a ele – outros usuários ainda podem obtê-las.
- b) Classe *Tag* é agregada à *Playlist*
  - i) Um *playlist* deve ter *tags*, no entanto, essas *tags* não dependem do ciclo de vida da *playlist*. Ou seja, mesmo que uma *playlist* seja excluída, as *tags* permanecem no sistema, pois não pertencem exclusivamente a ela – outras *playlists* ainda podem usá-las.
- c) Classe *Track* é agregada à *Playlist*
  - i) As faixas (*tracks*) fazem parte de uma *playlist*, mas não pertencem exclusivamente a ela e não têm seu ciclo de vida controlado por ela – uma mesma *track* pode ser referenciada em múltiplas *playlists*, e continua existindo independentemente da existência ou exclusão de qualquer uma delas

### 3) Composição –

- a) Classe *Playlist* é parte do *Usuário*
  - i) *Playlists* são associadas diretamente com a classe usuário – seu criador. Essa relação indica que cada *playlist* pertence a um único usuário e sua existência está atrelada à existência do usuário e seu ciclo de vida, ou seja, caso ele seja deletado, a *playlist* atrelada a ele também será.

Por fim, por meio do diagrama, é possível verificar quantas instâncias de uma classe se relaciona com outras por meio da cardinalidade dos vínculos entre elas:

- Um único usuário se relaciona com 0 ou mais outros usuários.
  - Ele pode seguir (ou ser seguido) por nenhum ou por muitos usuários;
- Um único usuário se relaciona com 0 ou mais *playlists*.
  - Ele pode ter em seu perfil nenhuma (caso ele só queira procurar novas *playlists*) ou várias *playlists*;
- Um único usuário se relaciona com 0 ou mais *badges*
  - Ele pode ter conquistado para seu perfil nenhuma ou várias *badges*;
- Uma única *playlist* se relaciona com 1 ou mais *tags*
  - Uma *playlist* deve obrigatoriamente ter no mínimo uma *tag*, mas pode ter inúmeras à gosto do seu criador.
- Uma única *playlist* se relaciona com 0 ou mais faixas musicais
  - Uma *playlist* não terá nenhuma música (recém-criada) ou várias.