

Neste projeto, o nosso grupo tomou a decisão de indicar que uma música ou playlist tendo o `selected != -1`, ou seja, quando o `selected` é `-1` nada se encontra selecionado.

Foi criada uma classe abstrata chamada `SongHandler`, onde a pasta `songs` é lida e onde são criadas `songs` a partir das músicas que se encontram nessa mesma pasta. Deste modo, ao serem criadas as músicas são criados também os meta dados. Decidimos que esta classe seria abstrata para ir ao encontro do conceito de reutilização.

Nas classes que estendem a `SmartPlaylist`, foi decidido que o número máximo de músicas em cada playlist seria de 5, para que o output do nosso programa correspondesse ao output do `simpleClient`.

Visto que todas as playlists que estendem a `SmartPlaylist` não permitem que músicas sejam adicionadas, removidas ou manipuladas manualmente, foi decidido que os métodos `add` e `remove` da classe `AbsPlaylist`, fossem overridden para que nada fosse feito e estes retornassem falso, caso fossem chamados manualmente (fora dos eventos).

Na classe `MostLikedSongsPlaylist`, no método `organizeMostLikedSong`, foi decidido organizar a `library` (sort), dentro de um array, dando `clear` à playlist. Caso a `library` tenha mais músicas que o máximo permitido na playlist, vão ser colocadas apenas as com maior `rate`. Caso a `library` tenha menos músicas que o máximo permitido na playlist, apenas se vão adicionar as músicas presentes. De ressaltar que vão ser adicionadas apenas músicas com `rate` superior a 0 (para ir ao encontro do output esperado).