Bibinson Celissaint - GRR20169928 | Ricardo Norio Miyata - GRR20166368

Trabalho 2

Universidade Federal do Paraná (UFPR)
Departamento de Informática
Bacharelado em Ciência da Computação (BCC)

Brasil

2019

Sumário

Introdução

O modelo UML é uma linguagem padrão para a elaboração da estrutura de projetos de software. Ela pode ser empregada para visualização, especificação, construção e documentação de artefatos que façam uso de sistemas complexos de software, isto é, é uma linguagem de modelagem *Wikipédia (2019)*. O LoopBack é uma estrutura Node.js de código aberto altamente extensível que permite: Criar APIs REST dinâmicas de ponta a ponta com poucas ou nenhuma codificação *Npm (2019)*.

Contextualização

A especificação diz respeito a um serviço de streaming de áudio. Seguindo as seguintes características:

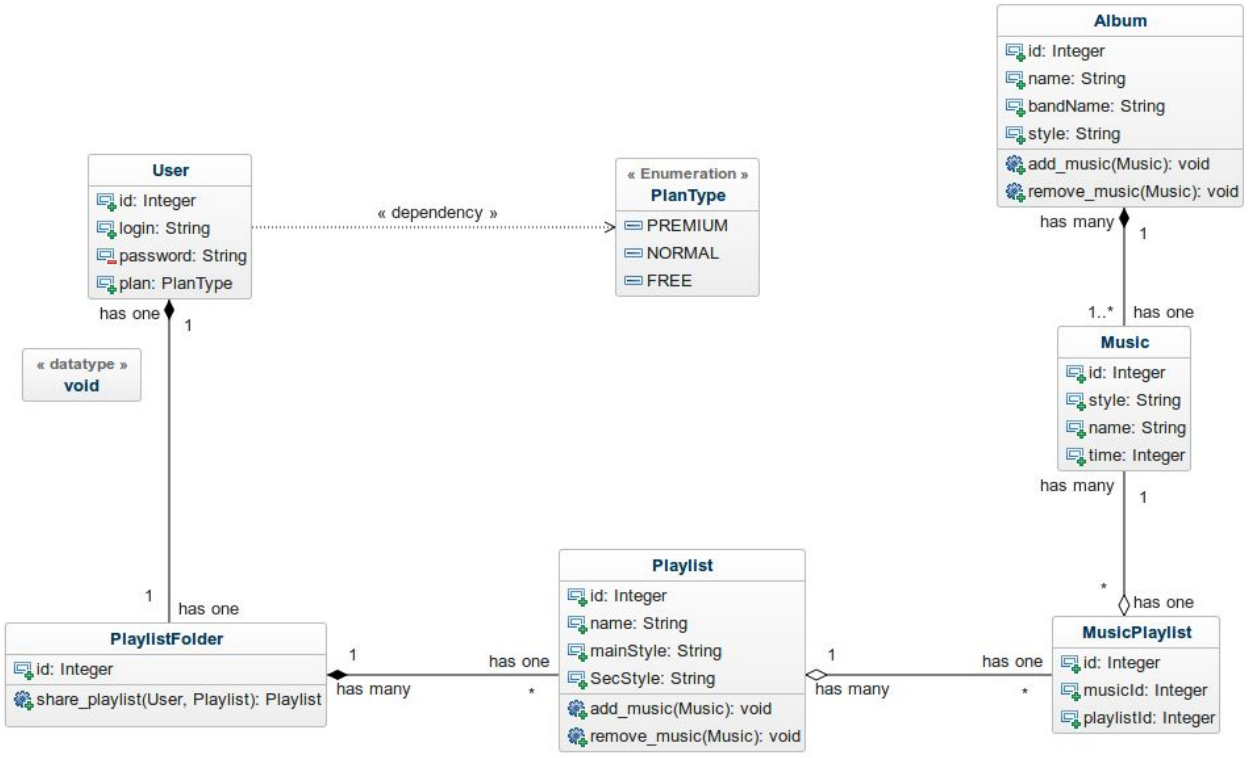
1.1) Usuários: O serviço de streaming de áudio deverá armazenar informações referentes ao login do usuário, senha, plano escolhido (premium, normal, free), e armazenar o diretório com as músicas que poderão ser tocadas. O diretório deverá ser uma adaptação do padrão Composite.

1.2) Diretório de músicas: O diretório de músicas terá Playlists, estas com várias músicas, sendo que as músicas estar conectadas com 1 ou mais álbuns.

As PlayLists terão um nome, o estilo principal e um estilo secundário. Os álbuns terão nome, nome da banda e estilo. As músicas terão estilo, nome e duração.

1.3) Compartilhamento de listas: as Playlists poderão ser compartilhadas entre os usuários, isto é, se um usuário U1 cria uma Playlist P1, esta lista poderá ser compartilhada com usuários U2.

Modelo UML



Para o modelo UML foram definidas as classes “PlanType”, “User”, “PlaylistFolder”, “Playlist”, “MusicPlaylist”, “Music” e “Album”.

1. PlanType - Dependência da classe User.
    a. Possui identificadores de plano.
    b. Classe pertencente a User.

2. User - Possui uma PlaylistFolder e é dependente da classe PlanType.
    a. id: número identificador.
    b. login: string de identificação do User.
    c. password: string referente a senha.
    d. plan: dependência da classe PlanType.

3. PlaylistFolder - Possui um User e muitas Playlist.
    a. id: número identificador.
    b. share_playlist(): método para compartilhar uma Playlist, recebe como parâmetro a identificação do User e a Playlist, retornando, em caso de sucesso, a Playlist compartilhada.

4. Playlist - Possui uma PlaylistFolder e muitas MusicPlaylist.
    a. id: número identificador da Playlist.
    b. name: string de identificação da Playlist.
    c. mainStyle: string com a identificação do estilo principal da Playlist.
    d. secStyle: string com a identificação do estilo secundário da Playlist.
    e. add_music(): método para adicionar uma Music na Playlist.
    f. remove_music(): método para remover uma Music da Playlist.

5. MusicPlaylist - Classe “ponte” de relação muitos para muitos, possui uma Playlist e uma Music.
    a. id: número identificador do relacionamento.
    b. musicId: número identificador de User.
    c. playlistId: número identificador da Playlist.

6. Music - Possui muitas MusicPlaylist e um Album.
    a. id: número identificador da Music.
    b. style: string referente ao estilo da música (flag de estilo).
    c. name: string de nome da musica.
    d. time: integer informativo de duração de tempo da música.

7. Album - Possui muitas Music.
    a. id: número identificador.
    b. name: string de identificação do Album.
    c. bandName: string identificador da banda em que a música pertence.
    d. style: string de identificação do estilo.
    e. add_music(): método para adicionar uma Music no repositório Album.
    f. remove_music(): método para remover uma Music do repositório Album.

LoopBack

Foram usados as seguintes funcionalidades do framework LoopBack para geração de código fonte no padrão MVC para plataforma web.

Geração do aplicativo em forma de andaimes (módulos);
Adição dos modelos (models);
Adição de uma fonte de dados (datasource);
Adição de um repositório (repository);
Adição de um controlador (controller);
Junção de todas as operações supracitadas;
Geração de relacionamento de modelos (relational databases).

Conclusão

O projeto é bastante intuitivo, cada parte do projeto se seguido corretamente em etapas, acaba por resultar no uso muito facilitado do framework LoopBack, pois já possuindo o modelo UML, é questão de “copiar e colar” o que o modelo especifica nos campos da ferramenta LoopBack. Cada comando de geração de módulos no LoopBack, pede nomes chaves e estes estão visualmente notável no modelo UML. Desta forma, em conclusão, o maior desafio de todo um projeto, acreditamos nós, que seja em questão do planejamento e estruturação do deste. Pois com as atuais ferramentas (frameworks) existentes, muito do trabalho braçal que antes era necessário, hoje já não é mais, pois as ferramentas geram perfeitamente os esqueletos seguindo um modelo, no caso do LoopBack, o MVC.

Bibliografia

https://www.npmjs.com/package/loopback
https://pt.wikipedia.org/w/index.php?title=UMLoldid=54878386