

# Relatório de Erros Corrigidos - Projeto TocaToca

## Introdução

Este relatório detalha a identificação e correção de três erros propositalis no projeto TocaToca, uma API REST para gerenciamento de artistas e músicas. Os erros foram corrigidos para garantir conformidade com boas práticas REST, robustez e integridade dos dados.

## Erro 1: Rota DELETE /artist/:id Associada ao Controlador Errado (Erro Teórico)

- **Local:** Arquivo routes.ts.
- **Descrição:** A rota DELETE /artist/:id foi associada ao controlador getArtistByIdController, que apenas busca um artista por ID, em vez de excluí-lo.
- **Justificativa:** Isso viola o padrão REST, pois o verbo DELETE deve excluir um recurso, não buscar. O comportamento enganoso (retornar 200 ou 404) não atende à expectativa de exclusão.
- **Correção:** Alterada a rota para GET /artist/:id, já que o controlador getArtistByIdController foi projetado para buscar um artista específico por ID, alinhando o verbo HTTP com a funcionalidade.
- **Impacto:** A API agora suporta a busca de um artista por ID corretamente, seguindo as convenções REST, embora a funcionalidade de exclusão não tenha sido implementada inicialmente (não era obrigatória pelo projeto).

## Erro 2: Bug na Geração de IDs no artistService.ts (Erro Lógico)

- **Local:** Arquivo artistService.ts, função createArtist.
- **Descrição:** A função subtraía 1 do newId (id: newId - 1), causando IDs duplicados (ex.: se o último ID era 3, o novo também seria 3).
- **Justificativa:** IDs duplicados quebram a unicidade dos recursos, o que pode levar a erros em operações como busca ou atualização.

- **Correção:** Removida a subtração, usando diretamente id: newId para garantir IDs únicos.
- **Impacto:** Novos artistas agora recebem IDs únicos, evitando conflitos e garantindo a integridade dos dados.

## Erro 3: Validação de Gênero Incorreta no musicService.ts (Erro Lógico/Regra de Negócio)

- **Local:** Arquivo musicService.ts, função createMusic.
- **Descrição:** A mensagem de erro para gêneros inválidos era enganosa (Artista inválido: \${genre}) e faltava validação de tipo para o parâmetro genre.
- **Justificativa:** Uma mensagem errada confunde os usuários, e uma validação insegura poderia levar a erros se o gênero não fosse uma string.
- **Correção:** Corrigida a mensagem para Invalid Genre: \${genre} e adicionada validação para garantir que genre seja uma string presente no enum MusicGenre.
- **Impacto:** A API agora retorna mensagens de erro claras e valida gêneros de forma segura, melhorando a experiência do usuário.

### Observações Gerais:

- O processo de correção foi facilitado pelo uso de TypeScript, que ajudou a identificar problemas de tipo (ex.: no caso do parâmetro genre) antes mesmo da execução.
- Durante a correção, enfrentamos um problema inicial de compilação devido à exportação inadequada da variável musics no musicService.ts. Isso foi resolvido exportando a variável corretamente, o que também foi necessário para as pendências posteriores.
- A abordagem iterativa (corrigir um erro por vez, testar, e prosseguir) garantiu que as mudanças fossem controladas e não introduzissem regressões.

## Considerações Finais

Os três erros foram corrigidos com sucesso, melhorando a robustez e a conformidade do projeto com boas práticas de desenvolvimento. A API agora funciona de forma mais consistente, e a funcionalidade de exclusão de artistas foi implementada posteriormente como parte das pendências, completando o ciclo de CRUD. O

processo de correção destacou a importância de validações rigorosas, mensagens de erro claras e conformidade com padrões REST.