

# **Gerência de Projetos de Software**

## **Termo de Encerramento do Projeto: *MyTrackList***

### **Componentes da Equipe:**

*Bruna Anselmi, Cauã Diniz, David Nunes, Lucca Pellegrini*

### **1. Objetivo do projeto**

Desenvolver um site para avaliar, organizar e compartilhar músicas, permitindo que usuários registrem suas preferências e descubram novas faixas.

### **2. Resumo do projeto**

No projeto MyTrackList, iniciamos o desenvolvimento com a configuração do serviço de API utilizando Axios, que seria a base para a comunicação entre o frontend e o backend. Em seguida, criamos a página de detalhes da música, onde os usuários poderiam visualizar informações específicas sobre cada faixa.

Após isso, implementamos um componente de Rota Protegida, garantindo que apenas usuários autenticados pudessem acessar determinadas áreas da aplicação. A criação do contexto de autenticação (AuthContext) foi um passo crucial, permitindo que gerenciassemos o estado do usuário e suas sessões de maneira eficiente. Em paralelo, integrarmos o formulário de login com a API, permitindo que os usuários se autenticassem e acessassem suas listas de reprodução. O próximo passo foi a integração do formulário de cadastro com a API, possibilitando que novos usuários se registrassem na plataforma. Uma parte importante do projeto foi a implementação da lógica para obter o token de acesso do Spotify, essencial para que os usuários pudessem interagir com a biblioteca de músicas da plataforma.

Ao longo do desenvolvimento, realizamos testes de integração e unitários para assegurar a qualidade do código e a funcionalidade da aplicação. O resultado foi uma solução robusta e funcional, que proporciona aos usuários uma experiência agradável ao gerenciar suas músicas e lista de reprodução.

### **3. Artefatos entregues**

Entregamos uma aplicação frontend desenvolvida em React 19, utilizando Vite, Material-UI, Axios e React Router. Essa aplicação inclui componentes

como ProtectedRoute, RatingInput e SearchBar, além de páginas como HomePage, LoginPage, ProfilePage, RegisterPage, SearchResultsPage e TrackDetailsPage. Também implementamos o contexto AuthContext, juntamente com serviços para a API, reviews e Spotify.

No backend, desenvolvemos uma API REST utilizando Node.js/Express com TypeScript, aproveitando o Bun como runtime. A estrutura do backend conta com controllers para Admin, Review e User, um banco de dados SQLite com factories e interfaces, e serviços para integração com o Spotify. Implementamos autenticação através de sessions, rate limiting e *helmet*, além de scripts de administração para registro e listagem de usuários e hashes. Também realizamos testes de API e integração, com as devidas configurações em TypeScript e dotenv.

#### 4. Conclusões

O projeto alcançou os objetivos definidos inicialmente, que incluíam a criação de uma aplicação web funcional para gerenciamento de playlists e a integração com a API do Spotify. Através de um desenvolvimento ágil e colaborativo, conseguimos entregar uma solução que não apenas atende às necessidades do cliente, mas também proporciona uma experiência intuitiva e agradável. Os resultados obtidos demonstraram a eficácia da comunicação entre o front-end e o back-end, a segurança na autenticação de usuários e a integração fluida com a biblioteca de músicas.

As principais contribuições do projeto incluem a criação de uma interface amigável e responsiva, a implementação de um sistema de autenticação robusto e a capacidade de buscar e gerenciar músicas diretamente do Spotify. Além disso, o projeto serviu como um excelente exercício de aplicação das tecnologias aprendidas, como React, Node.js, o runtime Bun, SQLite, e a integração de APIs.

Sobre as lições aprendidas, destacaram-se:

- a. **Importância do Planejamento:** O planejamento inicial e a definição clara dos objetivos foram fundamentais para manter o foco durante o desenvolvimento e garantir que todas as funcionalidades necessárias fossem implementadas no tempo previsto.
- b. **Integração e Comunicação:** A integração entre o front-end e o back-end é crucial. Aprendemos que uma boa comunicação entre as equipes de desenvolvimento é necessária para resolver problemas rapidamente e garantir que as funcionalidades se alinhem.
- c. **Segurança em Primeira Mão:** A implementação de um sistema de autenticação robusto destacou a importância da segurança em aplicações web, especialmente ao lidar com dados e credenciais de usuários.

- d. **Testes Contínuos:** A realização de testes de integração e unitários ao longo do desenvolvimento ajudou a identificar e prevenir problemas, melhorando a qualidade do código e a confiabilidade da aplicação.
- e. **Adaptação e Flexibilidade:** O projeto exigiu adaptação a novas tecnologias e ferramentas. Aprendemos a ser flexíveis e a buscar soluções criativas para os desafios que surgiram durante o desenvolvimento. Por exemplo—a escolha do runtime Bun, com sua forte integração nativa ao SQLite e desempenho elevado—ou o emprego do Axios para fazer as requisições à API REST do back-end.