**PlaylistShare**

(i) Descrição do produto:

PlaylistShare é uma aplicação web que funciona como uma biblioteca pública de playlists de música, onde os usuários podem compartilhar suas playlists favoritas com outros usuários e descobrir novas músicas para escutar.

(ii) Objetivos:

Criar um hub capaz de receber as playlists de música inseridas pelos usuários cadastrados e listá-las, junto com suas respectivas músicas, em uma página web de fácil visualização.

O projeto deve poder se integrar com a API dos principais aplicativos de música e obter as playlists de usuários selecionados.

(iii) Público Alvo:

Pessoas interessadas em compartilhar suas músicas favoritas ou conhecer novos estilos.

(iv) Funcionalidade principais:

* Cadastro e Edição de usuários (Usuários e visitantes).
* Remoção de usuários (administrador)
* Inserção, edição e remoção de playlist e suas músicas em uma database. (usuários)
* Exibição das playlist salvas em uma lista na página principal. (Acesso a todos)
* Exibição das músicas da playlists, junto com seus dados, em um modal. (Acesso a todos)
* Usuários podem favoritar playlists e exibi-las na página de favoritos. (Acesso exclusivo do usuário)

**Release 1:**

Em nosso primeiro release do PlaylistShare foram feitas 4 páginas web usando HTML e CSS. Nelas utilizamos os conhecimentos de HTML para criar, dividir e organizar os elementos da página e suas respectivas classes, fazendo uso das diferentes Tags disponíveis. Em relação ao CSS, fizemos o uso de diversos elementos de estilo para criar o design das páginas e do o modelo de flexbox para garantir o posicionamento adequado de todos os elementos.

Foram criadas duas versões diferentes de CSS, uma colorida com cores vibrantes e adequadas para usuários daltônicos e outra com alto contraste, utilizando uma mistura de branco e preto para facilitar a vista.

O trabalho da equipe foi dividida da seguinte forma:

* **Átila:**

Desenvolveu o arquivo HTML index e conta e da folha de estilo style.css. Criou a página principal do site, junto com o cabeçalho que deve estar presente em todas as páginas, e a página de conta onde deve aparecer todos os dados do usuário e suas playlists adicionadas.

* **Gustavo:**

Desenvolveu o arquivo HTML login e cadastro e as folhas de estilo conta.css e login.css. Criou essas duas telas contendo os formulários de preenchimento adequados como nome de usuário, senha e email que serão utilizados futuramente para autenticar e registrar os usuários. Também se estruturou formas de padronizar e organizar os arquivos CSS para releases futuros.

* **Lucas:**

Contribuiu na fase de planejamento do site e acompanhou o processo de desenvolvimento das páginas.

* **Marciano:**

Contribuiu na fase de planejamento do site e acompanhou o processo de desenvolvimento das páginas.

Todos os requisitos genéricos foram cumpridos, o site foi feito usando dialeto HTML5, características de formatação foram feitas exclusivamente dentro de HTMLs externos, foram apresentados dois estilos de CSS e todos os arquivos passaram sem erros nos valiadores.

**Release 2:**

* **Átila:**

Desenvolveu a estrutura do servidor node.js, configurou a model engine do EJS e as rotas padrões do Express. Criou a estrutura do banco de dados relacional PostgreSQL, junto com as funções para obter e inserir dados.  
Desenvolveu e organizou os templates do EJS com base os HTMLs feitos anteriormente e inseriu os dados nas páginas dinamicamente com base em dados recebidos do banco de dados e da API do spotify.

Em resumo foi feito também:

* Alternância de estilo CSS através de scripts que salvam dados do stylesheet em cookies do navegador.
* Integração com a API do spotify.
* Sistema de autenticação que registra dados do usuário em variáveis globais para serem inseridas nas páginas HTML (EJS).
* Criação de requisições assíncronas com FETCH API.
* Sistema de autenticação de formulários.
* Estrutura de tratamento de sucesso e erro em diversas partes da aplicação.
* Reorganização dos arquivos CSS.

**Release 3:**

* **Átila:**

Implementou a lógica base do controle de acesso de visitantes, usuários e administradores, com os Endpoints devidamente protegidos e as páginas geradas dinamicamente para os diferentes tipos de usuários logados.

Implementou os Endpoints para cadastro do usuário, edição de senha, exclusão de conta, cadastro de playlists e exibição do conteúdo de forma dinâmica de acordo com o usuário logado, junto com a lógica para salvar tudo no banco de dados.

Na página de dados das contas, o usuário pode ver todas as informações do perfil logado, pode excluir sua conta ou alterar sua senha e também visualiza as playlists que foram inseridas por ele.

* **Gustavo:**

Desenvolveu o menu de acessibilidade com o design dos botões e as funções de aumentar e diminuir o tamanho da fonte do site.

Implementou, na página inicial do site, modais que são exibidos ao clicar em uma playlist inserida, onde contem de forma mais nítida os dados da playlists, junto com um botão capaz de redirecionar ao site original em que ela se encontra.

Criou um endpoint para excluir qualquer playlists, o qual possui um controle de acesso onde apenas um administrador autenticado pode utilizar, que pode ser executado ao clicar em um botão no modal com os detalhes da playlist selecionada, botão esse que só aparece se o administrador estiver logado no site.

**Release 4:**

* **Átila:**

Planejou e implementou a separação da aplicação em duas camadas diferentes. Configurou o servidor da API, criou as rotas para serem acessadas, que por sua vez realizam as diversas alterações na database e implementou as diversas mensagens de resposta e os tratamentos de erro.

No servidor intermediário, corrigiu as rotas para se comunicarem com a API, realizando os request HTTP de get, post, delete e put.

Realizou diversas correções de erros encontrados e melhorias no fluxo de dados do site.

* **Gustavo:**

Planejou e implementou a separação da aplicação em duas camadas diferentes. Contribuiu na configuração do servidor da API e abriu um serviço de host no servidor intermediário para abrir acesso ao público.

Adicionou a função do usuário excluir playlists tanto na página de contas como diretamente da página inicial, desenvolveu a rota de acesso para isso e nela implementou uma lógica de autenticação.

Modificou e melhorou o sistema de cookies do site com um prazo para expirar e criou os cookies para salvar o tamanho da fonte escolhida pelo usuário. Também implementou um sistema de busca simples de playlists por nome na página inicial do site.

Realizou diversas correções de erros encontrados e melhorias no fluxo de dados do site.