**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Engenharia de *Software***

Projeto Integrado

Relatório Técnico

SoMix

Antonio Clovis Verza Junior

Belo Horizonte

Maio, 2023

# Projeto Integrado

**Sumário**

Projeto Integrado 3

1. Cronograma de Trabalho 4

2. Introdução 5

3. Definição Conceitual da Solução 6

3.1 Diagrama de Casos de Uso 6

3.2 Requisitos Funcionais 6

3.3 Requisitos Não-funcionais 7

4. Protótipo Navegável do Sistema 7

5. Diagrama de Classes de Domínio 8

6. Arquitetura da Solução 8

6.1 Padrão Arquitetural 8

6.2 C4 model - Diagrama de Contexto 9

7. Frameworks de Trabalho 10

8. Estrutura Base do Front End 10

9. Modelo Relacional ou Projeto de Banco de Dados NoSQL 10

10. Plano de Testes 11

11. Apropriação de Horas no Projeto 11

12. Código da Aplicação 12

13. Avaliação Retrospectiva 12

13.1 Objetivos Estimados 12

13.2 Objetivos Alcançados 12

13.2 Lições aprendidas 12

14. Referências 13

## Cronograma de Trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 08 / 05 / 23 | 12 / 05 / 23 | 1. Pesquisa e contextualização | Introdução e definição conceitual da solução |
| 15 / 05 / 23 | 19 / 05 / 23 | 2. Definir atores e casos de uso do sistema | Diagrama de casos de uso |
| 22 / 05 / 23 | 26 / 05 / 23 | 3. Levantar e definir os requisitos do sistema | Requisitos funcionais e não-funcionais |
| 29 / 05 / 23 | 02 / 06 / 23 | 4. Criação de telas de navegação baseadas nos casos de uso | Protótipo de interface navegável |
| 05 / 06 / 23 | 23 / 06 / 23 | 5. Definição das classes de domínio | Diagrama das classes de domínio |
| 10 / 07 / 23 | 14 / 07 / 23 | 6. Definição da arquitetura da aplicação | Padrão arquitetural e Arquitetura da solução |
| 17 / 07 / 23 | 21 / 07 / 23 | 7. Definição da estrutura do front end | Arquivos HTML e CSS |
| 24 / 07 / 23 | 04 / 08 / 23 | 8. Definição da persistência de dados | Projeto do banco de dados |
| 07 / 08 / 23 | 25 / 08 / 23 | 9. Definição dos testes | Casos de teste |
| 28 / 08 / 23 | 29 / 09 / 23 | 10. Desenvolvimento de uma funcionalidade | Código fonte e site da aplicação |
| 02 / 10 / 23 | 08 / 10 / 23 | 11. Revisão e fechamento | Vídeo de apresentação |

## Introdução

Aplicações de software para reprodução de música tem sigo largamente utilizadas desde a popularização da digitalização de arquivos de música pelo consumidor final, de início baseada principalmente em salvar localmente na memória do dispositivo arquivos oriundos de mídias físicas como CD-ROMs. Com o advento da internet, os arquivos antes mantidos localmente pelos usuários passaram a ser compartilhados na rede, gerando a necessidade por aplicações específicas para esse fim, algumas delas incluindo a capacidade de reprodução dos arquivos baixados da rede. Por fim, a versão mais recente das aplicações de reprodução de música foi disponibilizada ao público, a aplicação na qual o usuário não precisa manter uma cópia do arquivo a ser executado na sua memória local, ele apenas seleciona o que deseja ouvir a aplicação traz o arquivo do próprio acervo, geralmente acessado pela internet. Esse último tipo de aplicação é considerada uma aplicação de “streaming” de música. Com várias das funcionalidades baseadas nas aplicações da era inicial das músicas digitalizadas, essas aplicações ganharam proporção ao tornar possível que o usuário ouvisse músicas sem precisar tê-las salvas localmente na memória do seu aparelho.

Com a democratização do acesso à internet, tanto doméstica quanto móvel, e aos smartphones, a criação de aplicações moveis e web escalonou rapidamente e com isso também a quantidade de aaplicações de streaming de música cresceu, hoje existem mais de 15 opções para que o usuário escolha qual sua preferência com base em acervo, tipos de planos, tipos de interação com outros usuários da aplicação ou outras funcionalidades do aplicativo.

Uma funcionalidade que os aplicações de streaming de música herdaram das que executavam apenas arquivos locais é a criação de listas de reprodução. Nessa funcionalidade o usuário cria listas de músicas de sua escolha, nas gerações anteriores a funcionalidade se limitava a tocar, de maneira ordenada ou aleatória, as músicas que o usuário previamente adicionou à lista ao criá-la, musicas estas que obrigatoriamente deveriam estar gravadas localmente na memória do aparelho utilizado. Já nas aplicações de streaming de música a versão básica se manteve e foram incrementadas outras opções de lista de reprodução, geralmente listas já prontas oferecidas ao usuário agrupando músicas principalmente pelos temas de gênero e época (e.g. Sucessos dos anos 90, Rock e Pop-Rock, Rock anos 70, etc.). Além dessas funcionalidades, várias aplicações oferecem também listas personalizadas com base no que o usuário mais ouve e alguns ainda oferecem a funcionalidade de reproduzir uma lista de músicas relacionadas a um artista ou música de escolha do usuário.

Uma breve pesquisa realizada pelo autor com usuários de aplicaçẽos de streaming de música, quase 80% do grupo abordado usa listas de reprodução baseadas em gênero já prontas ou montam suas próprias listas com base no mesmo tema, temas subsequentes na lista de classificação de preferência dos usuários foram: artista, ambiente, época e evento.

O relacionamento dos usuários das aplicações de streaming de música com artistas específicos é algo a ser considerado quando se pensa em novas funcionalidades para aplicativos do tipo. A maioria desses usuários tem um ou mais artistas preferidos, dos quais nem sempre conhece todas as músicas mas tem um apreço especial por alguma caracterísita musical desse artista, seja voz, instrumentos utilizados, letras das músicas ou outros elementos da sonoridade empregada pelo artista. A mesma pesquisa citada anteriormente aponta que, do grupo abordado, mais de 90% gostaria de poder criar listas de reprodução com base nos artistas favoritos apenas selecionando os artistas, evitando ao usuário ter de construir a lista selecionando as músicas uma a uma. Outro resultado mostrado pela pesquisa é que quase 90% do grupo abordado gostaria também de poder criar listas com base nos seus álbuns favoritos apenas selecionando os álbuns.

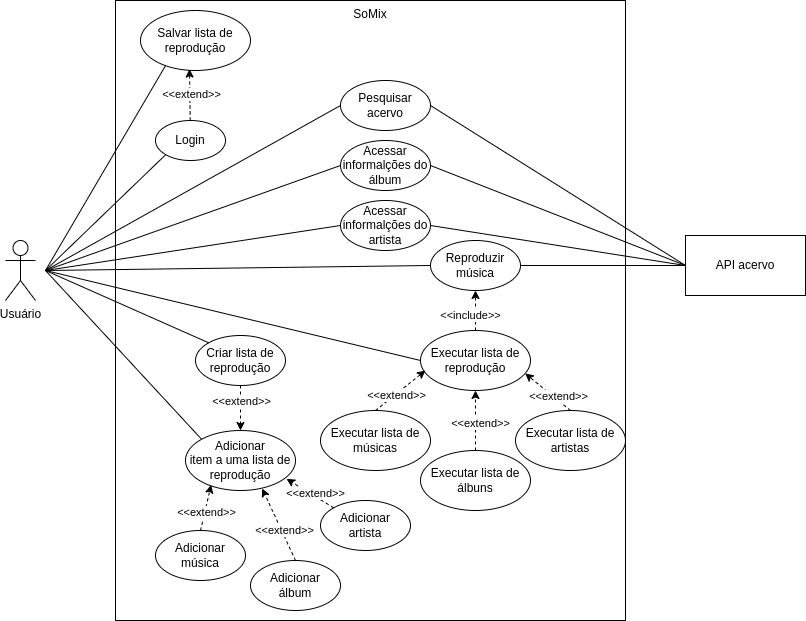
Dessa forma, o objetivo geral desse trabalho é apresentar a descrição de um projeto de uma aplicação de streaming de música que disponibilize ao usuário novas maneiras de criar listas de reprodução de músicas, bem como implementar ao menos uma funcionalidade da mesma aplicação.

Os objetivos específicos são:

* Elaborar diagramas de Casos de Uso, Classes de domínio, modelo de contexto C4 e modelo relacional do banco de dados;
* Listar requisitos funcionais e não-funcionais;
* Elaborar Wireframes e protótipo navegável;
* Definir a arquitetura da solução e as tecnologias a serem aplicadas;
* Desenvolver ao menos uma funcionalidade da aplicação de forma a atender a ao menos um dos casos de uso;
* Elaborar um plano inicial de testes da aplicação;

## Definição Conceitual da Solução

## Diagrama de Casos de Uso

Figure 1: Diagrama de casos de uso - Somix

## Requisitos Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O sistema deve prover todas as suas funcionalidades, com exceção da persistência das listas criadas, ao usuários que optarem por não fazer login | B | A |
| RF02 | O sistema deve prover todas as suas funcionalidades, inclusive a persistência das listas criadas, aos usuários que optarem por fazer login | A | M |
| RF03 | O usuário deve ser capaz de encontrar itens do acervo através de um campo de busca. A busca deve trazer resultados de músicas, álbuns e artistas com base nos dados de entrada informados pelo usuário no mesmo campo | B | A |
| RF04 | O sistema deve fornecer ao usuário a funcionalidade de executar qualquer música disponível no acervo | M | A |
| RF05 | A interface de uso deve apresentar uma barra de progresso da música que seja interativa, dando ao usuário a capacidade de avançar e voltar o cursor de execução na barra para refletir o mesmo efeito no progresso da música | M | B |
| RF06 | O usuário deve ser capaz de criar listas de reprodução de músicas, atribuindo um título a cada uma delas. O atributo de título da lista de reprodução não pode ser nulo e o sistema não deve prosseguir com a criação da lista enquanto o usuário não definir um título para a mesma | B | A |
| RF07 | O usuário deve ser capaz de criar 3 (três) tipos de listas de reprodução de músicas, sendo mandatório selecionar o tipo de lista de reprodução ao criá-la. Os tipos disponíves devem ser Lista de Reprodução de Músicas, Lista de Reprodução de Álbuns e Lista de Reprodução de Artistas | B | A |
| RF08 | O acervo musical do sistema deve ser fornecido por uma API externa | M | A |
| RF09 | O usuário deve ser capaz de acessar todas as informações disponíveis no acervo sobre os itens da categoria “álbum” ao selecionar um, incluindo a lista de músicas deste | B | M |
| RF10 | O usuário deve ser capaz de acessar todas as informações disponíveis no acervo sobre os itens da categoria “artista” ao selecionar um | B | M |
| RF11 | O usuário deve ser capaz de adicionar qualquer item do acervo categorizado como “música”, e apenas esse tipo de item, a listas de reprodução do tipo “Lista de Reprodução de Músicas” | M | A |
| RF12 | O usuário deve ser capaz de adicionar qualquer item do acervo categorizado como “álbum”, e apenas esse tipo de item, a listas de reprodução do tipo “Lista de Reprodução de Álbuns” | M | A |
| RF13 | O usuário deve ser capaz de adicionar qualquer item do acervo categorizado como “artista”, e apenas esse tipo de item, a listas de reprodução do tipo “Lista de Reprodução de Artistas” | M | A |
| RF14 | As listas de reprodução de todos os tipos devem prover as possibilidades de reproduzir seus itens em ordem sequencial de adição ou aleatória | A | M |
| RF15 | As listas de reprodução de álbuns devem sempre executar as músicas de um mesmo álbum na sequência já definida nas propriedades do mesmo sem deixar de respeitar a sequência entre álbuns, definida pela lista de reprodução criada pelo usuário | B | A |
| RF16 | As listas de reprodução de artistas devem sempre acessar músicas aleatórias do artista sem deixar de respeitar a sequência entre artistas, definida pela lista de reprodução criada pelo usuário | A | A |

\* B = Baixa, M = Média, A = Alta.

## Requisitos Não-funcionais

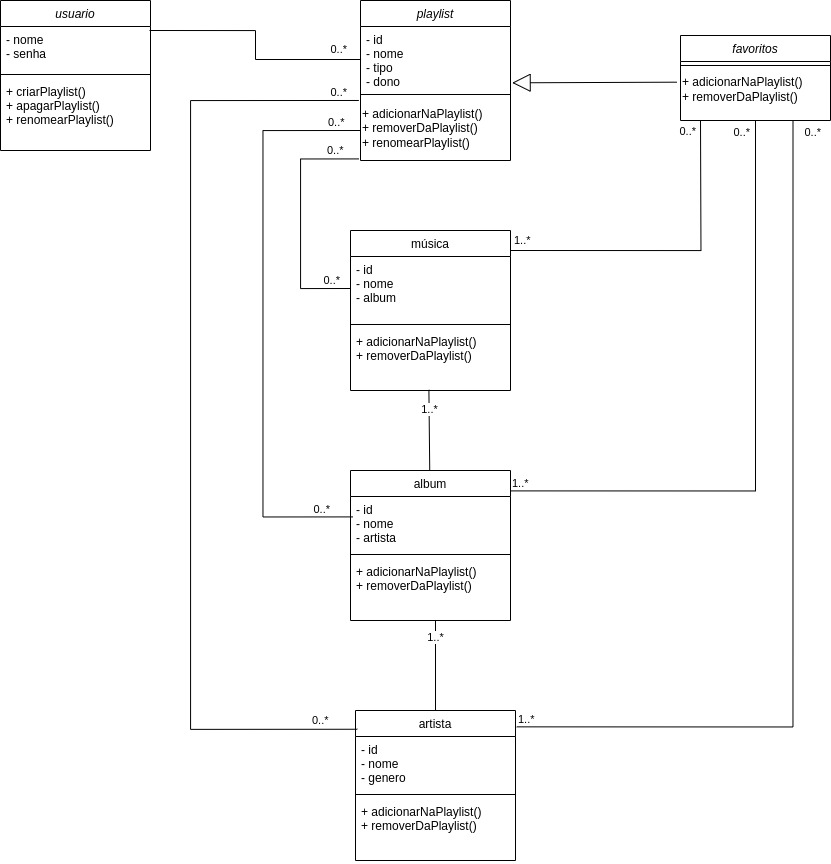
|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | A aplicação deve ser estruturada de maneira a prover fácil manutenção e expansão | A |
| RNF02 | A aplcação deve ser desenvolvida tendo como plataforma primária a web e possibilitar na medida do possível uma expansão futura para plataforma móvel | M |

\* B = Baixa, M = Média, A = Alta.

## Protótipo Navegável do Sistema

1. Vídeo de apresentação dos três principais casos de uso com o protótipo navegável disponível no link: [Vídeo apresentação](https://drive.google.com/drive/folders/11GXaIef_Tl8BVprtBEZTAeIBsa2ISdtT?usp=drive_link)
2. Protótipo navegável disponível no link: [Protótipo Navegável](https://www.figma.com/proto/SA9QLwDh3gDTw4NcEDECBG/SoMix?type=design&node-id=0-1&t=xkaPVQe5bXbOrKrR-0&scaling=min-zoom&starting-point-node-id=110%3A27)
3. Repositório Github do projeto: [Github](https://github.com/verzajr/SoMix)

## Diagrama de Classes de Domínio

Figure 2: Diagrama de classes de domínio - Somix

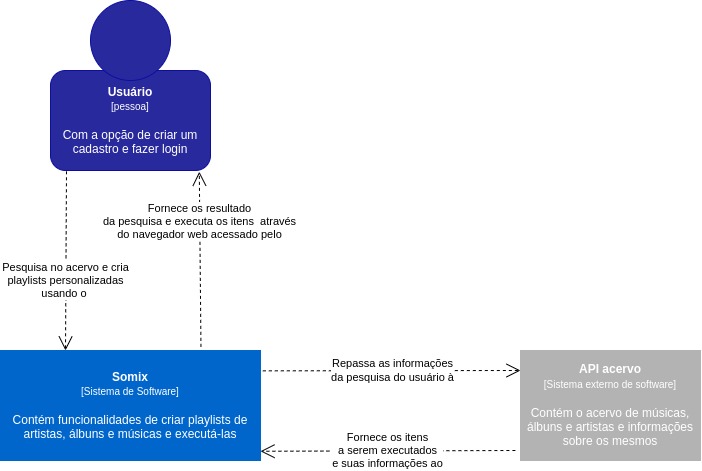
## Arquitetura da Solução

## Padrão Arquitetural

O Padrão arquitetural escolhido para o desenvolvimento da aplicação é o padrão de microsserviços, onde tanto back-end quanto front-end terão suas funções e funcionalidades construídas em pequenas partes com uma única responsabilidade buscando otimização e reuso de código.

## C4 model - Diagrama de Contexto

A Figura 1 mostra o diagrama no nível de Contexto do modelo C4 para o sistema SoMix. O usuário interage com o sistema SoMix através de um navegador web onde ele pode executar todos os casos de uso explicitados nos capítulos anteriores. O sistema SoMix usa como suporte para o acervo uma API externa e gratuita que fornecerá todas as informações e arquivos que serão apresentados e executados através do navegador web para cumprir os casos de uso. O sistema SoMix também cuida da persistência de dados, usando dados informados tanto pelo usuário quanto pela API para salvar as playlists criadas pelos usuários de forma que eles possam acessá-las no futuro ao fazer login no sistema.

Figure 3: Diagrama de contexto conforme molde C4

## Frameworks de Trabalho

As tecnologias escolhidas para o desenvolvimento da aplicação foram:

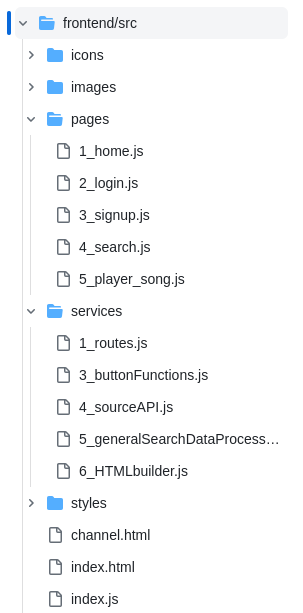
1. Front-end: JavaScript, CSS e HTML sem frameworks adicionais
2. Back-end: JavaScript com Node.js e acesso a banco de dados com SQL

## Estrutura Base do Front End

Como pode ser observado no link do Github mencionado no capítulo 4, a estrutura do frontend se divide em pastas para ícones, imagens, páginas, serviços e estilos, estando todas elas abaixo da pasta “src” onde ficam os arquivos inde.html e index.js que fazem a inicialização da aplicação.

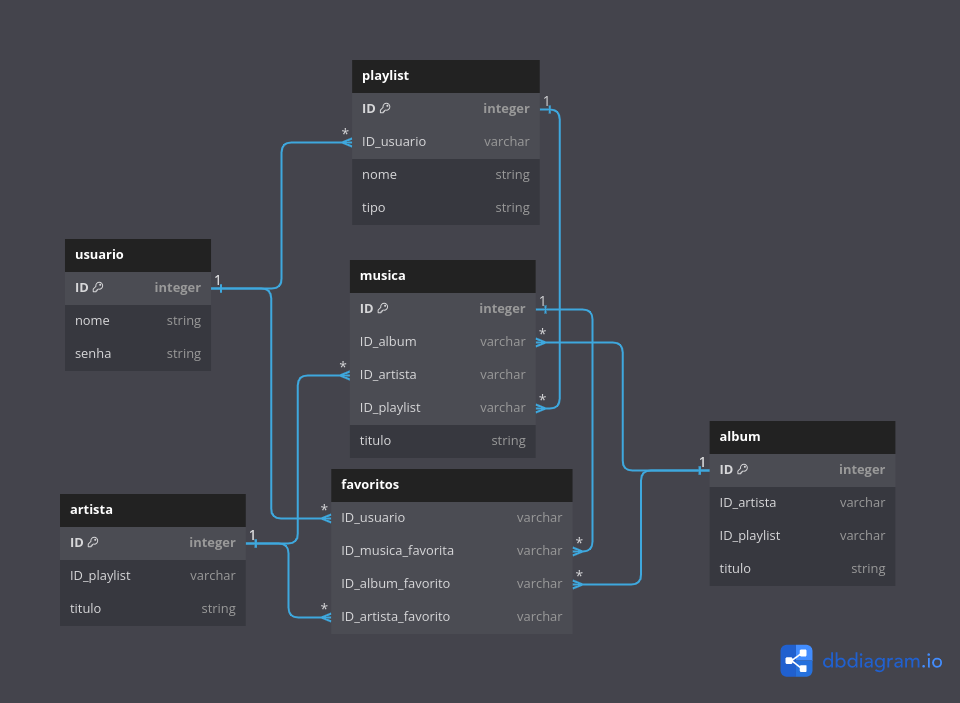
Seguindo a arquitetura de microsserviços, a pasta “services” engloba componentes com funções específicas e que podem ser alterados ou substituídos de maneira independente, facilitando alterações, correções e reuso.

A pasta de “pages” engloba componentes Javascript que constroem elementos HTML que são usados posteriormente na renderização de cada uma das páginas, na maioria dos casos importando funções e dados dos componentes de serviços.

Figure 4: Estrutura do frontend - Somix

## Modelo Relacional do Banco de Dados

Abaixo representação do modelo relacional das tabelas do banco de dados, contanto com chaves primárias e chaves estrangeiras para a relação entre as principais classes de domínio da aplicação.

Figure 5: Modelo relacional do banco de dados

## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | Pesquisar acervo | Verificar que a busca ao acervo provê resultados independente de o termo de busca ser em caixa alta, caixa baixa ou uma combinação dos dois. | Buscar pela palavra “eminem” usando diferentes combinações de letras maiúsculas e minúsculas na escrita palavra | A busca trás resultados normalmente, principalmente na seção artista onde é sabido que devemos ter ao menos um resultado. |
|  | Reproduzir música | Verificar que os dados fornecidos pela API podem ser reproduzidos em diferentes navegadores | Buscar uma música no acevo e solicitar a reprodução dela usando diferentes navegadores web (Ex: Google Chrome, Microsoft Edge, Safari, etc.) | Todos os navegadores web são capazes de reproduzir a música selecionada. |
|  | Criar Lista de Reprodução | Verificar que é possível criar listas de reprodução dos diferentes tipos: Artistas, Álbuns e Músicas. | Navegue até a página do usuário e use as opções na tela para criar uma nova playlist; Crie três playlists diferentes, uma de cada um dos três tipos disponíveis. | As playlists são criadas com sucesso e aparecem como opção ao clicar no botão “Adicionar à playlist” na tela de resultados de busca. |
|  | Adicionar música à playlist | Verificar que elementos do tipo “música” só podem ser adicionados a playlists do tipo “música” | Busque por uma música e selecione a opção “Adicionar à playlist” na tela de resultados de busca. | Apenas playlists do tipo “música” aparecem na lista de playlists alvo. |
|  | Adicionar álbum à playlist | Verificar que elementos do tipo “álbum” só podem ser adicionados a playlists do tipo “álbum” | Busque por um álbum e selecione a opção “Adicionar à playlist” na tela de resultados de busca. | Apenas playlists do tipo “álbum” aparecem na lista de playlists alvo. |
|  | Adicionar artista à playlist | Verificar que elementos do tipo “artista” só podem ser adicionados a playlists do tipo “artista” | Busque por um artista e selecione a opção “Adicionar à playlist” na tela de resultados de busca. | Apenas playlists do tipo “artista” aparecem na lista de playlists alvo. |
|  | Executar playlist de músicas | Validar a execução da playlist de músicas | Selecione uma playlist do tipo “música” para reprodução | Todas as músicas podem ser carregadas pelo navegador e executadas; Todas as músicas da playlist são executadas sem repetições de itens já executados e sem deixar de executar nenhum item. |
|  | Executar playlist de álbums | Validar a execução da playlist de álbums | Selecione uma playlist do tipo “álbum” para reprodução | Todas as músicas de cada álbum podem ser carregadas pelo navegador e executadas; Todas os albums da playlist são executados sem repetições de itens já executados e sem deixar de executar nenhum item; Todas as músicas de um álbum são executadas antes de iniciar o próximo da lista. |
|  | Executar playlist de artistas | Validar a execução da playlist de artistas | Selecione uma playlist do tipo “artistas” para reprodução | As músicas podem ser carregadas pelo navegador e executadas; As músicas de cada artista são reproduzidas de maneira aleatória; Apenas uma música do artista é executada antes de mover para o próximo artista da lista; É executada uma música de cada artista da lista sem repetição do mesmo artista antes de passar por todos os outros da lista; |
|  | Acessar informações do artista | Verificar os dados trazidos da API acervo ao solicitar informações do artista | Busque por um artista e clique na imagem ou no nome do mesmo | A página de informações do artista traz informações de biografia, álbums e músicas do artista. |
|  | Acessar informações do álbum | Verificar os dados trazidos da API acervo ao solicitar informações do álbum | Busque por um album e clique na imagem ou no nome do mesmo | A página de informações do álbum traz todas as músicas contidas no mesmo bem como o nome do artista ao qual o álbum pertence. |

## Apropriação de Horas no Projeto

|  |  |  |
| --- | --- | --- |
| **Histórico de apropriação de horas** | | |
| **Data do registro** | **Atividade** | **Quantidade de horas** |
| 08/05/2023 | Pesquisa e contextualização | 2 |
| 10/05/2023 | Pesquisa e contextualização | 1 |
| 13/05/2023 | Registro em relatório do contexto do projeto | 4 |
| 16/052023 | Definição de requisitos | 2 |
| 18/05/2023 | Definição de requisitos | 1 |
| 20/05/2023 | Definição de requisitos | 3 |
| 26/05/2023 | Registro em relatório dos requisitos funcionais e não-funcionais | 2 |
| 06/06/2023 | Definição do diagrama de casos de uso | 3 |
| 07/06/2023 | Registro do diagrma dos casos de uso no relatório do projeto | 1 |
| 15/06/2023 | Definição da ferramenta para protótipo UX | 2 |
| 01/07/2023 | Criação de telas e protótipo navegável | 4 |
| 04/07/2023 | Criação de telas e protótipo navegável | 3 |
| 09/07/2023 | Criação de telas e protótipo navegável | 6 |
| 13/07/2023 | Criação de telas e protótipo navegável | 3 |
| 29/07/2023 | Criação de telas e protótipo navegável | 5 |
| 05/08/2023 | Criação de telas e protótipo navegável | 2 |
| 06/08/2023 | Gravação do vídeo de apresentação do protótipo navegável | 3 |
| 12/08/2023 | Definição de padrão arquitetural, modelo C4 e atualização do relatório | 4 |
| 14/08/2023 | Criação da estrutura e dos arquivos base do frontend | 4 |
| 16/08/2023 | Criação das primeiras telas de navegação e dos estilos base (CSS) para a aplicação | 5 |
| 19/08/2023 | Criação do “router” para renderização das telas conforme mudanças na rota da aplicação | 5 |
| 23/08/2023 | Criação de telas e arranjo dos componentes de serviços | 3 |
| 05/07/2023 | Definição de API acervo e implementação da conexão de testes com a mesma | 7 |
| 12/07/2023 | Criação da tela de resultados de busca | 3 |
| 17/07/2023 | Implementação dos serviços de criação de elementos HTML baseados nos resultados da API | 5 |
| 05/10/2320 | Tela de execução (player), definição do ambiente de hospedagem para produção | 5 |
| 12/10/2023 | Estilização dos comandos do player | 3 |
| 14/10/2023 | Conexão com a API para ambiente de produção e implementação da autenticação do usuário junto à API | 8 |
| 15/10/2023 | Atualização do relatório e gravação do video de apresentação | 4 |

## Código da Aplicação

**Repositório público de código:** [**Github**](https://github.com/verzajr/SoMix)

**Endereço do site da aplicação:** [**Somix**](https://somix.vercel.app/)

**Vídeo de apresentação do projeto:** [**Google Drive**](https://drive.google.com/file/d/1HxflLIor40TYchBRmVGxKsUD3zEpUBRd/view?usp=drive_link)

Nota: Ao acessar o site, será solicitado o login junto à API acervo. Esse login serve apenas para autenticar o acesso ao acervo da API, sem prover nenhuma persistência de dados do uso da aplicação Somix. O mesmo login não implica em nenhum tipo de assinatura da API acervo.

## Avaliação Retrospectiva

O desenvolvimento do projeto Somix me proporcionou a oportunidade de aprender mais sobre ferramentas de design de software e seus diferentes diagramas assim como também me ajudou a entender e usar melhor as principais ferramentas e práticas de desenvolvimento de software.

## Objetivos Estimados

Ter os casos de uso de Pesquisar Acervo, Reproduzir Música, Criar Lista de Reprodução e Executar Lista de Reprodução finalizados até a data final de entrega do trabalho e do relatório.

## 13.2 Objetivos Alcançados

Implementação dos casos de uso Pesquisar Acervo e Reproduzir Música.

## Lições aprendidas

|  |  |  |
| --- | --- | --- |
|  | **Retrospectiva (Lições Aprendidas)** | |
|  | **Descrição da Lição** | **Classificação** |
| 1 | Existem várias ferramentas gratuitas para a criação de diagramas e modelos de design de software | Positiva |
| 2 | O uso de “Grids” no posicionamento dos elementos HTML junto com a definição dos estilos com base nas dimensões dinâmicas da porta de visualização contribuem muito na responsividade da aplicação | Positiva |
| 3 | Priorize sempre o uso de APIs com documentação extensiva | Negativa |
| 4 | O uso de fóruns abertos pode ser um bom aliado na resolução de problemas e uso de novas tecnologias | Positiva |