

Brasilify: um site de divulgação da cultura brasileira

**Gabriel Lucas Silva Machado, Livia Almeida Barbosa, Tarsila Bessa Nogueira
Assunção**

**Departamento de Ciência da Computação – Universidade Federal de Minas Gerais,
Brasil**

`gabriellucas1366@gmail.com, liviaab@dcc.ufmg.br, tataah.b@gmail.com`

***Abstract.** This paper presents Brasilify, a web-application made to divulge the brazilian culture. It is made a brief introduction about the theme, what is the motivation, what is the application focus, how the data was obtained, what tools were used and how they were used. In this paper, it's also demonstrated how Brasilify works.*

***Resumo.** Este artigo, apresenta Brasilify, um site de divulgação da cultura brasileira. É feita uma breve introdução à proposta do trabalho, qual a motivação, qual o foco da aplicação, como os dados foram coletados, quais ferramentas foram utilizadas e como a aplicação funciona.*

1. Introdução

Nos últimos anos, a popularidade de gêneros musicais no Brasil tem dado espaço para novos ritmos, tendo como os principais representantes desses, o sertanejo universitário e o funk. Apesar disso, a cultura musical brasileira não se resume a tais ritmos, visto que existem diversos artistas bem-sucedidos representando gêneros, como Bossa Nova, Samba, MPB, entre outros.

Devido à popularização da internet, e consequentemente, a facilidade de se acessar uma informação, a divulgação da cultura acabou se adaptando à nova realidade, tendo como principal alvo, a internet. Porém, mesmo com a melhoria do acesso às informações, muitos brasileiros, com destaque aos jovens, acabam conhecendo melhor a música popular nacional mais recente e também a internacional, deixando de conhecer grandes artistas brasileiros que marcaram suas épocas.

Em resposta ao problema, decidiu-se implementar uma aplicação web, denominada *Brasilify*, cujo objetivo é divulgar a cultura musical nacional, de uma maneira direcionada, onde os artistas são classificados pelos seus respectivos gêneros musicais, e seus álbuns pelas décadas em que se destacaram. Para construir tal aplicação, foi criado um banco de dados, cujos dados foram coletados de API's, como Spotify e MusicGraph.

2. Decisões de projeto

2.1. Escolha das APIs

Existem diversas ferramentas disponíveis para a manipulação de dados musicais, que lidam tanto com informações essenciais, como nome da música, artista, do álbum e época de lançamento, como com dados mais específicos, como tom, timbre e outras informações mais específicas. Cada uma possui seu diferencial que será primordial no desenvolvimento da aplicação. Analisando as necessidades, escolheu-se trabalhar com os serviços MusicGraph e com o Spotify para recuperar as informações fundamentais.

A escolha do MusicGraph ocorreu devido à necessidade de uma aplicação que fornecesse os artistas de acordo com um determinado país. A aplicação também retorna dados como as principais décadas de atuação do artista ou da banda e o identificador do artista no Spotify, o que facilitou, respectivamente, para a separação dos artistas por décadas e para a pesquisa de outros dados do artista na API do Spotify.

Dados como popularidade atual, artistas relacionados, álbuns, músicas, estão disponíveis na API do Spotify, os quais foram considerados como básicos para construir a aplicação Brasilify. Visto que os dados disponíveis atendiam à demanda e partindo do pressuposto que essa primeira versão da aplicação seria simples para focar na aprendizagem do processo de sua construção, considerou-se que a utilização desta API seria suficiente.

Cogitou-se a possibilidade de realizar a coleta de dados utilizando o servidor de dados do MusicBrainz, o qual é vasto e tem uma gama muito maior de informações sobre artistas. Entretanto, existiram dificuldades durante a aprendizagem de manipulação do servidor e preferiu-se utilizar as APIs citadas anteriormente.

2.2. Desenvolvimento

As APIs se baseiam nos princípios REST e retornam os metadados no formato JSON. A recuperação dos dados fornecidos pelas APIs foi feita através de scripts escritos na linguagem Python. A escolha da linguagem deve-se à quantidade de bibliotecas disponíveis para manipular dados json, arquivos csv, além da possibilidade da utilização do Spotify, uma biblioteca para o Spotify Web API.

Após a coleta dos dados, para utilizá-los na aplicação, foi necessário criar um banco de dados para comportá-los. Esse banco de dados foi feito seguindo o modelo relacional, segue as definições das tabelas e uma breve descrição das mesmas.

2.2.1. Tabela Artistas

No banco criado, foi feita a tabela “Artistas”, que possui o formato abaixo:

```
CREATE TABLE Artistas(  
    spotify_id VARCHAR(100) NOT NULL,  
    name VARCHAR(200) NOT NULL,  
    genero VARCHAR(10),  
    popularidade INTEGER,  
    imagem TEXT,  
    PRIMARY KEY (spotify_id)  
);
```

Essa tabela foi criada para comportar os dados referentes a todos os artistas presentes na aplicação.

2.2.2. Tabela ArtistasSimilares

No banco criado, foi feita a tabela “ArtistasSimilares”, que possui o formato abaixo:

```
CREATE TABLE ArtistasSimilares(  
    spotify_id VARCHAR(100) NOT NULL,  
    similar_id VARCHAR(100) NOT NULL,  
    PRIMARY KEY (spotify_id, similar_id)
```

```

    artista1 VARCHAR(100) NOT NULL,
    artista2 VARCHAR(100) NOT NULL,
    PRIMARY KEY (artista1, artista2),
    FOREIGN KEY (artista1) REFERENCES Artistas(spotify_id),
    FOREIGN KEY (artista2) REFERENCES Artistas(spotify_id)
);

```

Essa tabela foi criada para comportar os dados referentes aos artistas que são similares. Foi necessário a criação de uma tabela específica, pois trata-se de um atributo multi-valorado.

2.2.3. Tabela Decadas

No banco criado, foi feito a tabela “Decadas”, que possui o formato abaixo:

```

CREATE TABLE Decadas(
    artista VARCHAR (100) NOT NULL,
    decada VARCHAR(5) NOT NULL,
    FOREIGN KEY(artista) REFERENCES Artistas(spotify_id),
    PRIMARY KEY (artista, decada)
);

```

Essa tabela foi criada para comportar os dados referentes às décadas em que os artistas se destacaram. Foi necessário a criação de uma tabela específica, pois trata-se de um atributo multi-valorado.

2.2.4. Tabela ArtistaGeneroMus

No banco criado, foi feito a tabela “ArtistaGeneroMus”, que possui o formato abaixo:

```

CREATE TABLE ArtistaGeneroMus(
    artista VARCHAR (100) NOT NULL,
    generomusical VARCHAR(100) NOT NULL,
    FOREIGN KEY(artista) REFERENCES Artistas(spotify_id),
    PRIMARY KEY (artista, generomusical)
);

```

Essa tabela foi criada para comportar os dados referentes aos gêneros musicais em que cada artista pode ser classificado. Foi necessário a criação de uma tabela específica, pois trata-se de um atributo multi-valorado.

2.2.5. Tabela Albuns

No banco criado, foi feito a tabela “Albuns”, que possui o formato abaixo:

```

CREATE TABLE Albuns(
    artista VARCHAR (100) NOT NULL,
    id_album VARCHAR(200) NOT NULL,

```

```

    nome_album VARCHAR(100) NOT NULL,
    lancamento VARCHAR(50),
    imagem TEXT,
    popularidade INTEGER,
    FOREIGN KEY(artista) REFERENCES Artistas(spotify_id),
    PRIMARY KEY (artista, id_album)
);

```

Essa tabela foi criada para comportar os dados referentes aos álbuns gravados por cada artista . Foi necessário a criação de uma tabela específica, pois trata-se de um atributo multi-valorado.

2.2.6. Tabela AlbumMusicas

No banco criado, foi feito a tabela “AlbumMusicas”, que possui o formato abaixo:

```

CREATE TABLE AlbumMusicas(
    artista VARCHAR (100) NOT NULL,
    album VARCHAR(200) NOT NULL,
    id_musica VARCHAR(200) NOT NULL,
    numero INTEGER,
    nome_musica VARCHAR(200),
    preview_url TEXT,
    FOREIGN KEY(artista) REFERENCES Artistas(spotify_id),
    FOREIGN KEY(album) REFERENCES Albuns(id_album),
    PRIMARY KEY (artista, album, id_musica)
);

```

Essa tabela foi criada para comportar os dados referentes às músicas de cada álbum . Foi necessário a criação de uma tabela específica, pois trata-se de um atributo multi-valorado.

Após a definição do esquema do banco de dados, primeiro testou-se sua implementação localmente, a fim de detectar possíveis erros básicos. Posteriormente, migrou-se para o MySQL da plataforma phpMyAdmin, disponibilizado pelo servidor do Departamento de Ciência da Computação - UFMG .

A partir daí, foi possível desenvolver a aplicação Web que interage com o banco de dados criado, utilizando a linguagem PHP. Foram utilizados, também, alguns recursos do framework Bootstrap para construir, por exemplo, a página de contatos.

3. Aplicação

Como dito anteriormente, Brasilify é uma aplicação web. Para acessar a página principal, clique [aqui](#).

Segue abaixo algumas imagens demonstrando o funcionamento da aplicação.

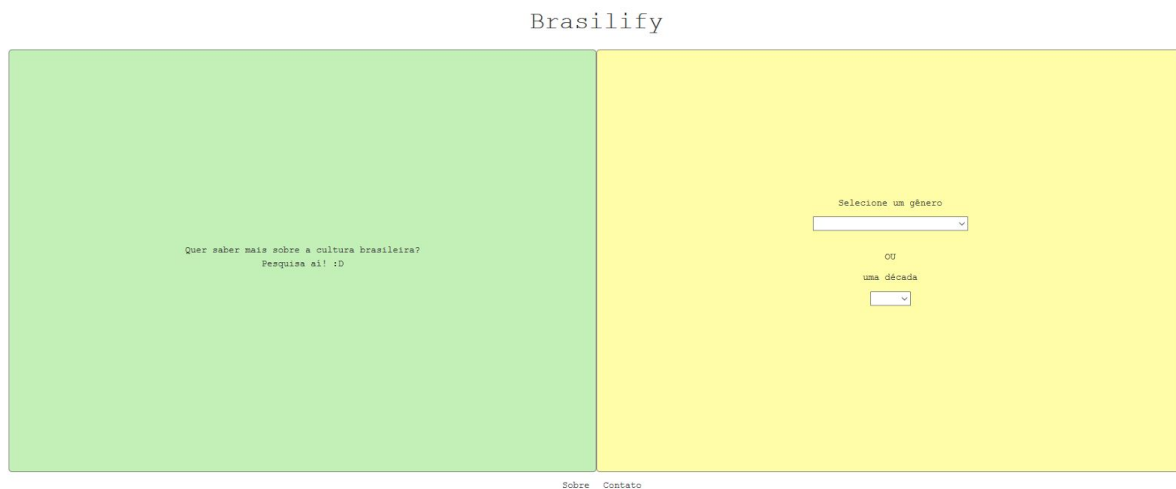


Figura 1. Página inicial da aplicação.

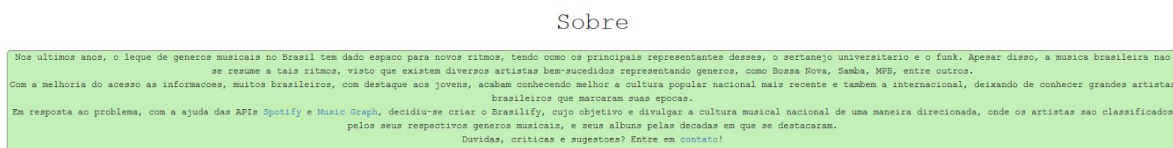


Figura 2. Página para o link “Sobre”.

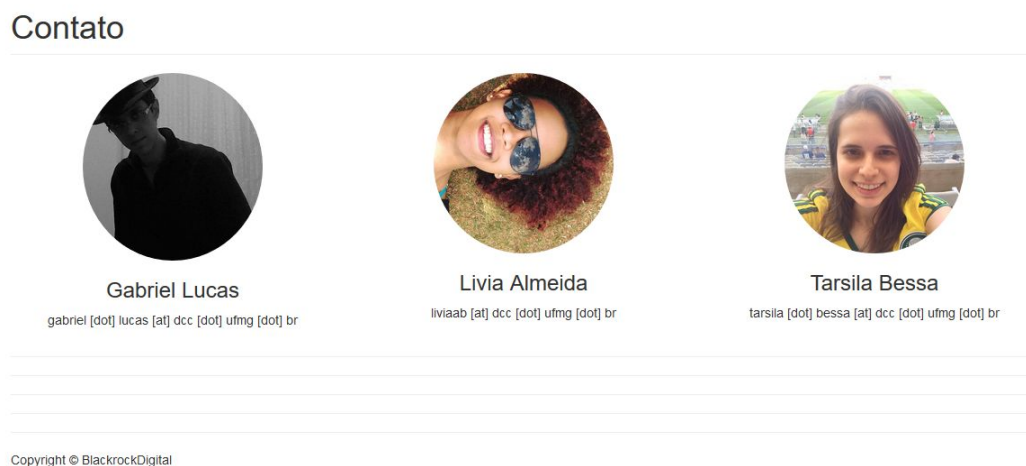


Figura 3. Página para o link “Contato”.

samba

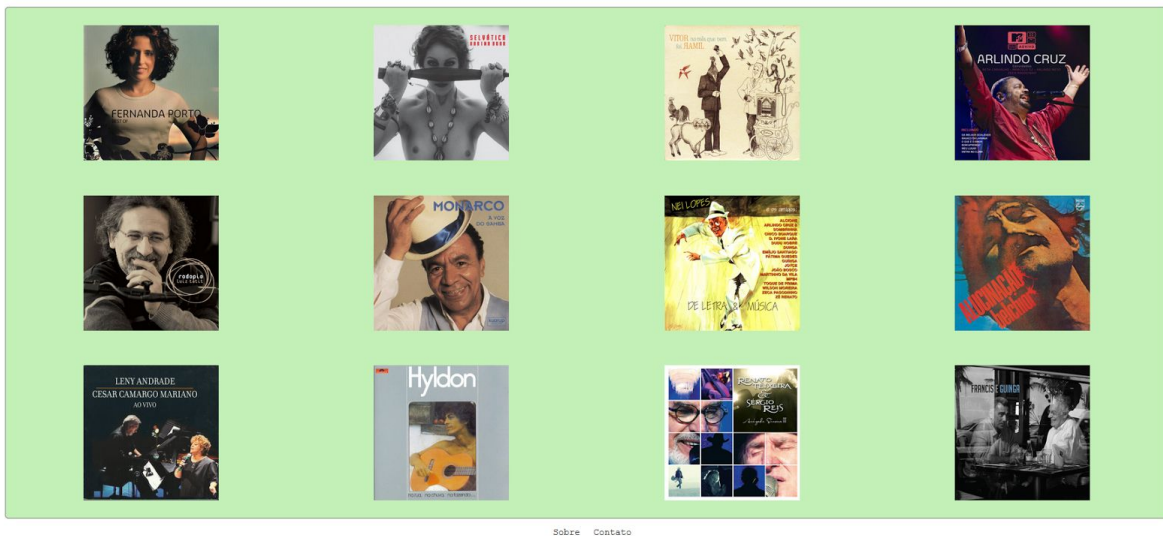


Figura 4. Modelo de página para uma pesquisa por gênero.

2000

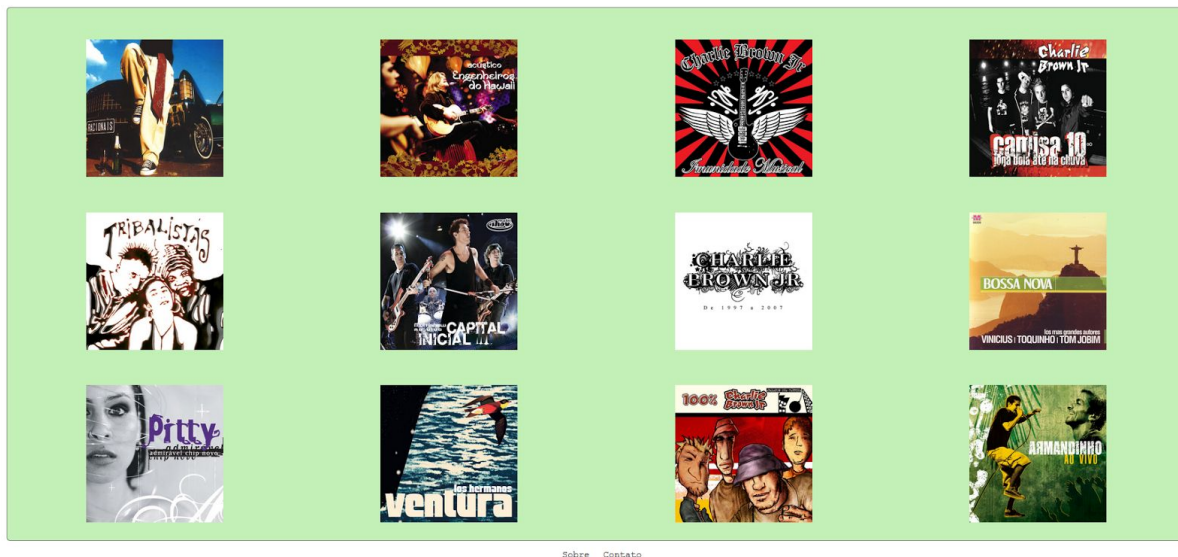


Figura 5. Modelo de página para uma pesquisa por Década..

Tanto na página de gênero quanto na página de década, são mostrados álbuns selecionados de acordo com o critério de pesquisa, com a diferença que consideração a exibição por década, também é considerada a popularidade do álbum como critério de ordenação para a exibição.

4. Conclusão

O presente artigo apresenta a aplicação destinada à divulgação da cultura brasileira denominada *Brasilify*. Apesar da ferramenta não inovar em termos de ser um banco de dados

de artistas, a mesma inova em relação ao destaque à artistas brasileiros, de acordo com suas épocas de auge, e seus estilos musicais.

Trabalhos futuros incluem a adição de novos dados ao banco, novas funcionalidades e melhoria na exibição da aplicação. Em relação aos dados, é interessante adicionar o estado de origem dos artistas e biografias dos mesmos, visto que as que algumas destas que foram coletadas continham informações de má qualidade. Já em relação às novas funcionalidades, pode-se adicionar uma ferramenta que possibilite a um usuário contribuir com novos dados, de maneira semelhante à Wikipedia. Uma outra funcionalidade interessante é um sistema de recomendação de artistas/álbuns, que leve em consideração o histórico de pesquisa de um usuário, entretanto considera-se que tal funcionalidade está além do escopo desse trabalho.

É válido ressaltar que, devido à inexperiência prática dos desenvolvedores da aplicação e da falta de um objetivo claro no início da definição do projeto, a proposta dessa aplicação foi alterada a fim de simplificá-la e de definir um objetivo claro para a mesma. Apesar das simplificações feitas, ainda teve-se uma série de dificuldades durante o desenvolvimento do trabalho, sendo estas principalmente voltadas à manipulação de dados, coleta de dados via API's e da criação de páginas web. Logo, mesmo que o trabalho aparenta ser simples, o mesmo foi o resultado de muita pesquisa e aprendizado sobre novas tecnologias.

Referências Bibliográficas:

Documentação Spotipy. Disponível em <<https://spotipy.readthedocs.io/en/latest/>>, Acesso Novembro/2016;

Spotify Web API. Disponível em <<https://developer.spotify.com/web-api/>>, Acesso Novembro/2016;

Music Graph API Documentation. Disponível em <<https://developer.musicgraph.com/>>, Acesso Novembro/2016;