FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Spotify-ed: Music Recommendation and Discovery in Spotify

José Lage Bateira

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Fabien Gouyon

Co-Supervisor: Matthew Davies

June 23, 2014

Spotify-ed: Music Recommendation and Discovery in Spotify

José Lage Bateira

Mestrado Integrado em Engenharia Informática e Computação

June 23, 2014

Abstract

Not so long ago, before the Internet boom, listening or discovering new music was a challenge on its own. Now, with a few clicks one can have on their hands such a vast music catalogue that a human mind cannot compute it.

There are dozens of online services that offer exactly that. Some focus on creation/generation of playlists, others try to expand their music catalogue even further, but others focus more on personalized music recommendation. And these ones present their results to the user with a list or a grid of music artists, for example.

However, lists or grids do not give the user enough information about the relation between the results. One could even say that they are not related to each other, which is not true.

The relations exist and can be represented as a network of interconnected artists in a graph, where a node is a music artist, and each edge between them represents a strong connection. This is the concept that RAMA (Relational Artist MAps), a project developed at INESC Porto, uses.

From a single search, RAMA is able to draw a graph that helps the user to explore new music that might caught his/her interest in a much more natural way.

Nonetheless, when a user wants to listen to an artist's music, Youtube's stream is used. Although one can find a large catalogue of music in Youtube, this service is not Music Oriented and the sound quality is not adequate for a music streaming service.

Youtube's stream needs to be replaced, and Spotify can provide a quality stream and an accurate music catalogue.

But how can RAMA and Spotify be integrated?

This thesis proposes a Spotify App. Will a Spotify user experience a more pleasant and natural way of music discovery from this graphical representation of artist relations within Spotify, than its standard discovery more (with grids)?

That is the main question that this dissertation urges to answer.

Resumo

Bem longe vão os tempos, antes da Internet, em que ouvir e descobrir música nova era um desafio por si só. Agora, com alguns cliques, temos acesso a um catálogo de música tão grande, que o nosso cérebro não consegue processar.

Existem dezenas de serviços online que oferecem isso mesmo. Alguns especializam-se na criação/geração de playlists (que funcionam como rádios), outros em expandir o catálogo de música e outros focam-se mais na sugestão e recomendação de artistas/álbuns/músicas personalizada para os utilizadores. Estes últimos, apresentam as sugestões de conteúdo ao utilizador de uma forma rudimentar como listas ou em grelha.

No entanto, listas ou grelhas não fornecem ao utilizador qualquer tipo de informação adicional sobre a relação entre os artistas nem justificam a sua semelhança. Até fazem parecer que não existe nenhuma relação/ligação entre os artistas recomendados, o que não é verdade.

Essas relações existem e podem ser representadas como uma rede de artistas interligados num grafo, onde cada nó é um artista de música, e cada ligação entre nós representa uma ligação forte de parecença entre os artistas. Este é o conceito que o RAMA (Relational Artist MAps), projeto desenvolvido no INESC Porto, usa.

A partir de uma pesquisa de um artista de música, o RAMA cria e desenha um grafo que ajuda o utilizador a explorar música que lhe possa interessar de uma forma muito mais natural e informativa.

No entanto, quando um utilizador pretende ouvir uma música de um artista, é usado *stream* do Youtube. Apesar de este oferecer um catálogo alargado de música, o mesmo não é indicado para esta funcionalidade pois não fornece uma API nativamente orientada a música, nem a qualidade de som do *stream* é adequada.

A experiência musical do utilizador do RAMA poderá melhorar consideravelmente ao colmatar esta falha. Existe por isso uma necessidade de substituir o Youtube por outro serviço mais orientado a *streaming* de música de qualidade. O Spotify é um deles. Fornece API orientada a música, e o *streaming* é de qualidade adequada para este tipo de funcionalidade.

De que formas é que se pode integrar o RAMA e o Spotify?

A escolha final foi desenvolver uma aplicação (como *plugin*) para o Spotify. Será que um utilizador Spotify ao descobrir música nova de uma forma mais gráfica terá uma experiência de utilizador mais rica e natural do que o modo de descoberta *standard* do Spotify (em grelha)?

Esse é o objetivo primordial desta dissertação: Tentar descobrir se utilizadores Spotify terão uma experiência melhorada ao usar a Aplicação Spotify proposta.

Contents

1 Introduction 1
1.1 Context . . . 1
1.2 Motivation and Goals . . . 1
1.3 Project . . . 2
1.4 Dissertation Structure . . . 3

2 State of The Art 5
2.1 Introduction . . . 5
2.2 Related and Similar Services . . . 5
2.2.1 Liveplasma - liveplasma.com . . . 5
2.2.2 Tuneglue - audiomap.tuneglue.net . . . 7
2.2.3 MusicRoamer - musicroamer.com . . . 8
2.3 Conclusions . . . 9

3 Projeto 13
3.1 Spotify . . . 14
3.1.1 Ferramentas de Desenvolvimento . . . 14
3.1.2 Experimentações Feitas . . . 20
3.1.3 Conclusão . . . 23
3.2 Tecnologias . . . 23
3.2.1 *Spotify Desktop Client* . . . 23
3.2.2 Webkit Development Tools - webkit.org . . . 24
3.2.3 Npmjs - npmjs.org . . . 24
3.2.4 Gruntjs - gruntjs.com . . . 27
3.2.5 Arborjs - arborjs.org . . . 27
3.3 Resumo e Conclusões . . . 27

4 Plano de Trabalho 29
4.1 Fases do Projeto . . . 29
4.1.1 Fase 1 - Desenho da Aplicação . . . 29
4.1.2 Fase 2 - Mapeamento de Metadados Spotify em Last.fm . . . 30
4.1.3 Fase 3 - Criação e Edição do grafo . . . 30
4.1.4 Fase 4 - Reprodução de Música . . . 31
4.1.5 Fase 5 - Avaliação e Validação . . . 31
4.2 Calendarização . . . 32
4.3 Resumo . . . 32

5 Conclusões 35

CONTENTS

List of Figures

2.1 liveplasma: search result for "Amália Rodrigues"; upper left corner: artist albums; lower left corner: youtube's *mini-player* 6
2.2 liveplasma: interface to start playing tracks. *Similar* button plays tracks from similar artists, whereas, the *only* button only plays tracks from the specified artist. 6
2.3 Tuneglue: menu que aparece ao clicar num nó. 7
2.4 Tuneglue: grafo depois do primeiro nó ser expandido. 8
2.5 MusicRoamer: Search options. by artist; by keyword and by Last.fm username . 9
2.6 MusicRoamer: Visual representation of the artist graph 10
2.7 MusicRoamer: Personalizable parameters for the graph 11
2.8 MusicRoamer: The graph after expanding one node 12

3.1 Spotify: interface do modo de descoberta do *desktop client* 15
3.2 Spotify: Aplicação Last.fm aberta no *Spotify Player* 16
3.3 Spotify: *Play Button* pode ser embebido em *websites*. 17
3.4 Spotify: *Follow Button* permite seguir um artista. 17
3.5 Experiência com *Metadata API* e *Play Button Widget* (código fonte: `github.com/carsy/spotify-playground`) 21
3.6 *Website* do RAMA embebido numa Aplicação Spotify 22
3.7 Resultado do teste do elemento *canvas* 22
3.8 Menu *Develop* 24
3.9 Webkit: Vista da tab *Inspector*. Outas ferramentas disponíveis (tabs): *Resources, Network, Sources, Timeline, Profiles, Audits* e *Console*. 25
3.10 Webkit Network 25
3.11 Webkit Profile: É possível ver que a renderização do grafo é o que ocupa mais tempo de processamento como esperado. No entanto, existe uma parte de *JQuery* que ocupa 12.72% do tempo de processamento, o que pode indicar um possível ponto de melhoria de performance. 26
3.12 Webkit Audit: 96% do código *CSS* não está a ser usado, sendo por isso, um ponto de melhoria reduzir a quantidade de informação descarregada. 26
3.13 Webkit Console: Erros de *Javascript* aparecem destacados para chamar a atenção. 26

4.1 Calendarização do Plano de Trabalho 33

LIST OF FIGURES

Chapter 1

Introduction

1.1 Context

Not so long ago, before the Internet boom, listening or discovering new music was a challenge on its own. Now, with a few clicks one can have on their hands such a vast music catalogue that a human mind is not able to compute.

There is an uncountable number of online services that offer exactly that. Some focus on creation/generation of playlists, others try to expand their music catalogue even further, while others focus more on personalized music recommendations. Most of these, present their results to the user with a list or a grid of music artists, for example.

However, lists or grids do not give the user enough information about the relation between the results [1]. One could even say that they are not related to each other, which is not true.

The relations exist and can be represented as a network of interconnected artists in a graph, where a node is a music artist, and each edge between them represents a strong connection. This is the concept that RAMA (Relational Artist MAps), a project developed at INESC Porto, uses. [2] [3] [4] [5]

1.2 Motivation and Goals

From a single search, RAMA is able to draw a graph that helps the user to explore new music that might caught his/her interest in a much more natural way.

Nonetheless, when a user wants to listen to an artist's music, Youtube's stream is used. Although one can find a large catalogue of music in Youtube, this service is not music oriented and the sound quality is not adequate for a music streaming service.

Youtube's stream needs to be replaced, and Spotify[1] can provide a quality stream and an accurate music catalogue.

[1] http://spotify.com

But how can RAMA and Spotify be integrated?

Several possibilities were analysed.

Spotify Play Button[2]
A Spotify widget that can be embedded in RAMA.

Integrate Spotify User's profile data in RAMA
To help complement artist recommendations.

Spotify App[3]
A plugin to Spotify's desktop client

This dissertation proposes a Spotify App. Will a Spotify user experience a more pleasant and natural way of music discovery from this graphical representation of artist relations within Spotify, than its standard discovery mode (with lists)?

That is the main question that this dissertation urges to answer.

Moreover, to evaluate and validate the final product, end-user testing will be done to compare Spotify's user experience with and without the developed application.

1.3 Project

The app is meant to be an extra mode for discovering new music in Spotify.

This way, a visual representation of an artist network with a graph, similar to RAMA, is proposed.

The application runs inside the Spotify environment (Spotify's Desktop Client) where its main features are: visualization of relations between artists, starting by the current playing artist; ability to grow the graph by expanding nodes; visualize tags (that describe an artist) in the graph representation.

The tools used in the development of the application were:

Spotify Desktop Client
The developed application is integrated in the Spotify's desktop client.

Webkit Development Tools - webkit.org
This is the engine used to run Spotify Applications.

Npmjs - npmjs.org
Package manager for development dependencies.

Bower - bower.io
Package manager for runtime dependencies.

Gruntjs - gruntjs.com

Manager for automating tasks. Very usefull for tests, code optimization and other repetitive tasks.

Vis.js - visjs.org

Visualization framework.

1.4 Dissertation Structure

This dissertation contains four additional chapters.

In chapter 2, related works will be presented to evaluate the current state of the art.

In chapter 3, the project's details will be explained, starting with an introduction to the Spotify Apps' development environment and the role of the technologies used during the development of the prototype.

In chapter 4, a more detailed explanation about the developed prototype will be given, as well as some challenges and problems encountered during the development process.

Chapter 5 concludes this report.

Chapter 2

State of The Art

2.1 Introduction

In this chapter, the most relevant web services for this thesis will be analysed.

The proposed methodology focus on how the content is presented and less on what the content is (without discarding its importance). Even so, some projects that focus on the content will be analysed.

The presented projects often use external data bases (like last.fm) to fetch metadata from. This is the preferred way, since those are the most complete sets of information.

2.2 Related and Similar Services

2.2.1 Liveplasma - liveplasma.com

liveplasma.com is a *flash*[1] application that not only it allows to see a graph of music artists, but also of books and movies.

The interaction with the graph is very faulted: no changes to the graph are allowed, and the user can easily make a mistake and perform unwanted actions like redrawing the graph with another artist as the root node.

In 2.1 one can see the search result for "Amália Rodrigues".

On the left side of the application there are some interesting elements: a grid of the artist's albums, a mini-player (stream from Youtube).

In 2.2 the user can have the choice to play tracks *only* from that artist, or play *similar* artists.

2.2.1.1 Pros

This tools, has two interesting aspects to it:

[1] http://get.adobe.com/flashplayer

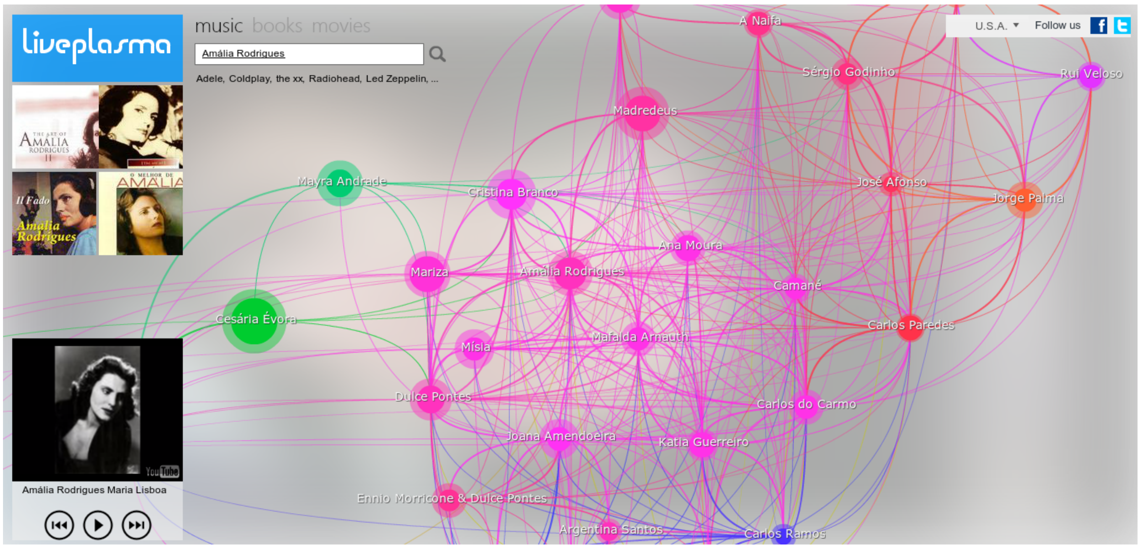

Figure 2.1: liveplasma: search result for "Amália Rodrigues"; upper left corner: artist albums; lower left corner: youtube's *mini-player*

- Links to buy albums of the artist
- Play tracks from similar artists to the search artist.

2.2.1.2 Cons

The graph drawn from this simple search, is very cluttered with edges. Two nodes can have several connections between, which seems to overload the graph and making it very confusing.

Different colours are used, but their meaning remains unknown. One can assume that they represent the similarity between artists, but that is just speculation.

It can also be assumed that the size of the nodes (radius value) can be directly proportional to the artist's popularity, but that is, again, just speculation.

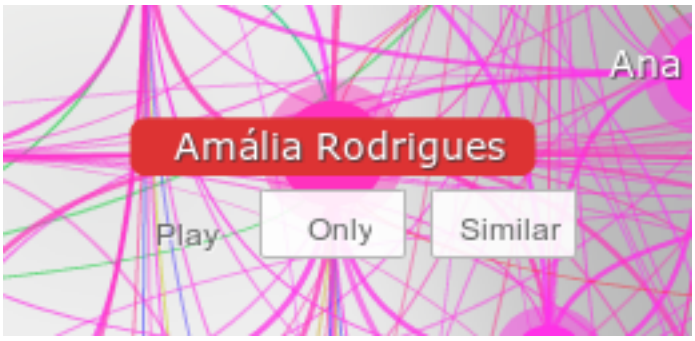

Figure 2.2: liveplasma: interface to start playing tracks. *Similar* button plays tracks from similar artists, whereas, the *only* button only plays tracks from the specified artist.

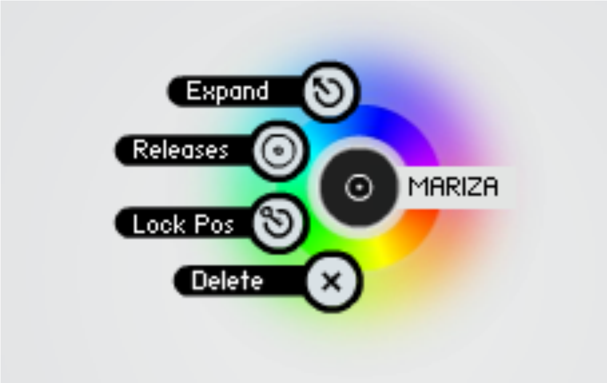

Figure 2.3: Tuneglue: menu que aparece ao clicar num nó.

One critical detail is that the user cannot visually point out the search node in the graph, given the lack of visual distinction from the other nodes of the graph 2.1.

2.2.1.3 Summary

In short, liveplasma is not very user friendly. It uses too many colours and edges, which makes the user experience of searching for new music even harder than usual.

2.2.2 Tuneglue - audiomap.tuneglue.net

Tuneglue is another flash application that tries to explore the graphic visualization of network of related artists. Last.fm's metadata API is used to retrieve artist information.

When you start Tuneglue and search for an artist, say "Mariza", the user is presented with a single-node graph. By clicking the node, the user has four options (2.3): expand, releases, lock position and delete.

When you first expand a node, you get the root node with six child nodes 2.4.

So the first feature that brings the user experience to another level (in comparison with liveplasma) is that of graph editing. The user can expand, fix and delete every single node in the graph.

2.2.2.1 Pros

Tuneglue gives control to the user. On one hand, the user is able to craft a graph and tailor it to its needs. The user feels that graph is its own creation.

2.2.2.2 Cons

On the other hand, the user has the responsibility to create the whole graph, which might be too much trouble and deteriorate the user experience.

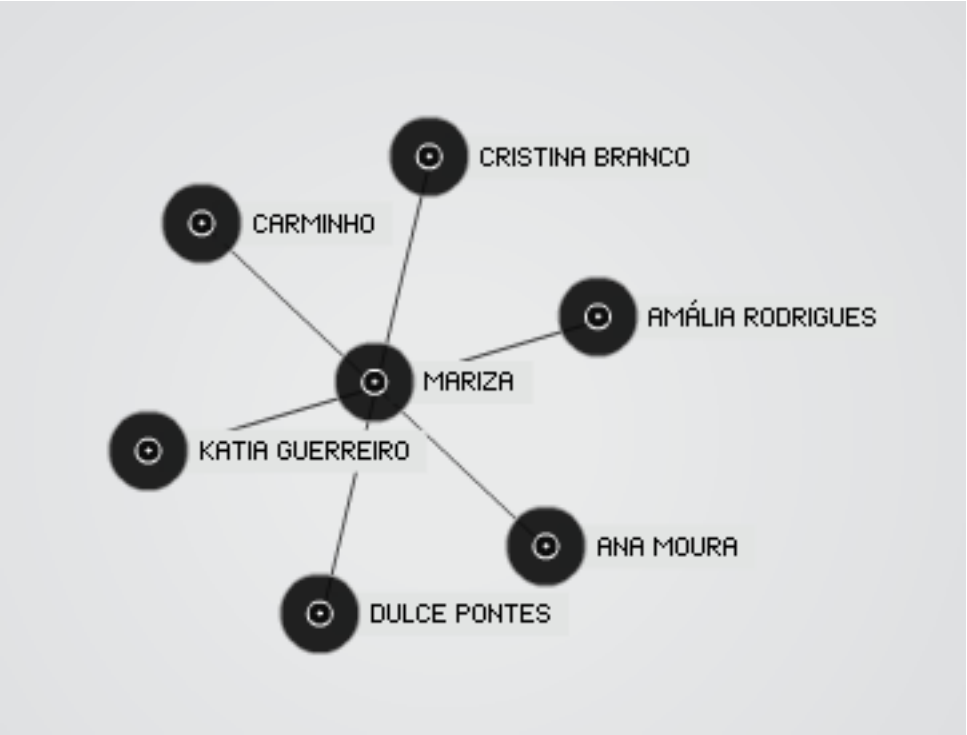

Figure 2.4: Tuneglue: grafo depois do primeiro nó ser expandido.

Again the root node is not highlighted, which might leave the user lost when the graph gets more and more complex.

2.2.2.3 Summary

Tuneglue takes the approach to give the user the power to create what he wants. But with no limit, the user can easily create a very complex graph that deteriorates the user experience.

2.2.3 MusicRoamer - musicroamer.com

MusicRoamer is yet another flash application. Although it is similar to Tuneglue when it allows the user to expand the graph further and further, it also imposes some limits to the user to avoid getting the graph confusing.

2.2.3.1 Pros

There are three types of search 2.5):

Artist Search
The most used one.

Keyword Search
Search using keywords like genres and tags

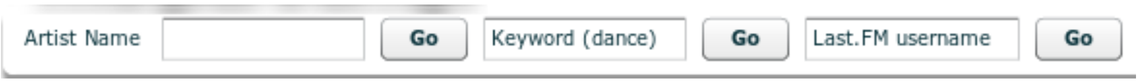

Figure 2.5: MusicRoamer: Search options. by artist; by keyword and by Last.fm username

Last.fm user search

The search result generates several graphs with the top artists of the user as the root nodes.

Independently of the search form used, the result will always be one (or more) graphs where the nodes are music artists.

MusicRoamer is worth mentioning because of the way it shows the graph. In 2.6 one can see the search result for "Mariza".

The images of the music artists are used to represent the nodes. This way, the user has a more friendly mind map of the resulting graph.

There is also some parameters (2.7) that the user can personalize to change the appearance of the graph: zoom; repulsion force between the nodes; size of the artist's images and the number of artist to be used as the branching value.

2.2.3.2 Cons

MusicRoamer is a flash application which makes the interface less natural and fluid to a website user.

Another problem occurs when the user starts to expand more and more nodes. The graph starts to get confusing (2.8), the edges are drawn over the images, the artist's names start to get mixed in the images.

2.2.3.3 Summary

Although the MusicRoamer user has a lot freedom when creating the graph, the graph presentation if weak and not very æsthetically pleasant.

2.3 Conclusions

There is an uncountable number of services to discover new music. The ones presented in the previous examples have a visual representation in graph.

The following services are that have some sort of mechanism to present the users with new music (not necessarily with visual tools):

- liveplasma.com
- audiomapa.tuneglue.net
- musicroamer.com

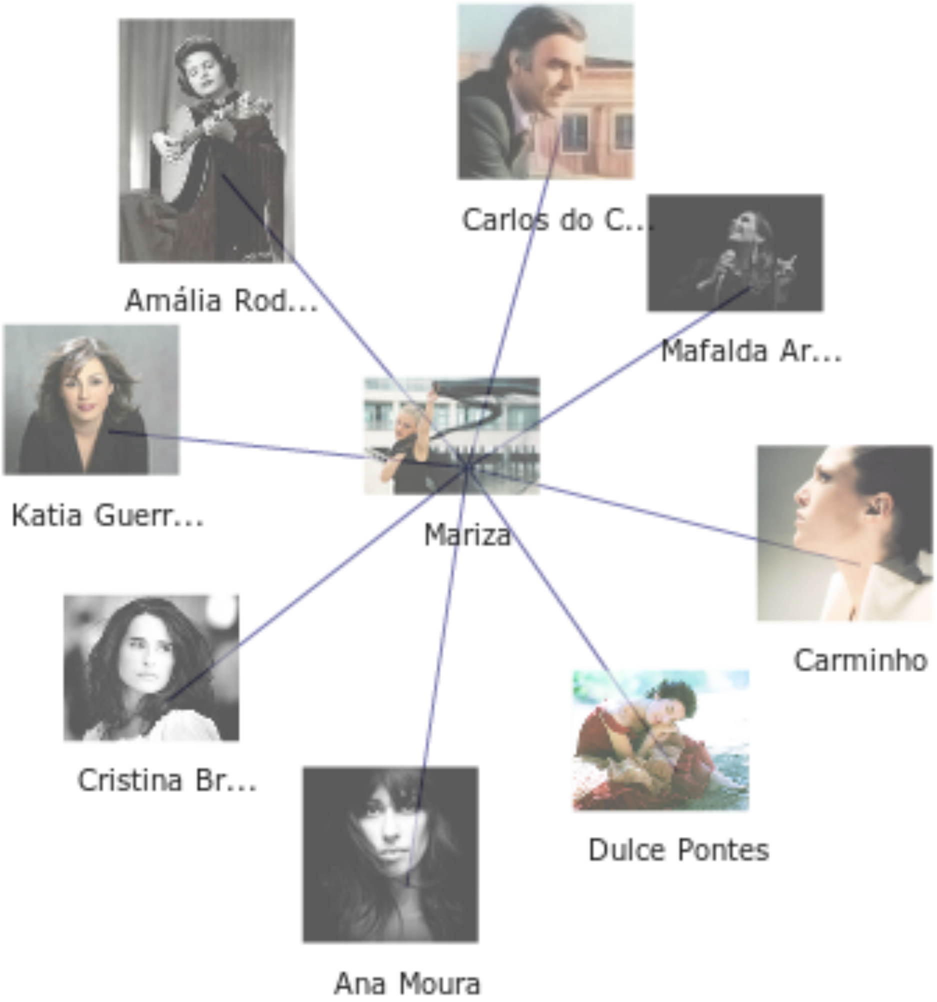

Figure 2.6: MusicRoamer: Visual representation of the artist graph

- discovr.info
- ifyoudig.net
- pitchfork.com
- hypem.com
- awdio.com
- 8tracks.com
- tastekid.com

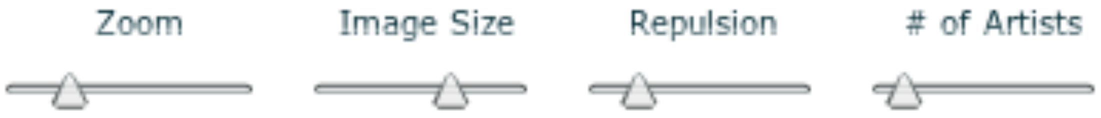

Figure 2.7: MusicRoamer: Personalizable parameters for the graph

- songza.com
- thesixtyone.com
- mog.com
- stereogum.com
- gigfi.com
- jango.com
- soundcloud.com
- grooveshark.com

The most important aspect to retain from the previous examples is that the bigger the branching value of the graph, the more confusing and cluttered the graph becomes. One could say that the visual tool loses its initial purpose to help the user to discover new music.

A way to avoid that problem would be to force limits in the graph creation process.

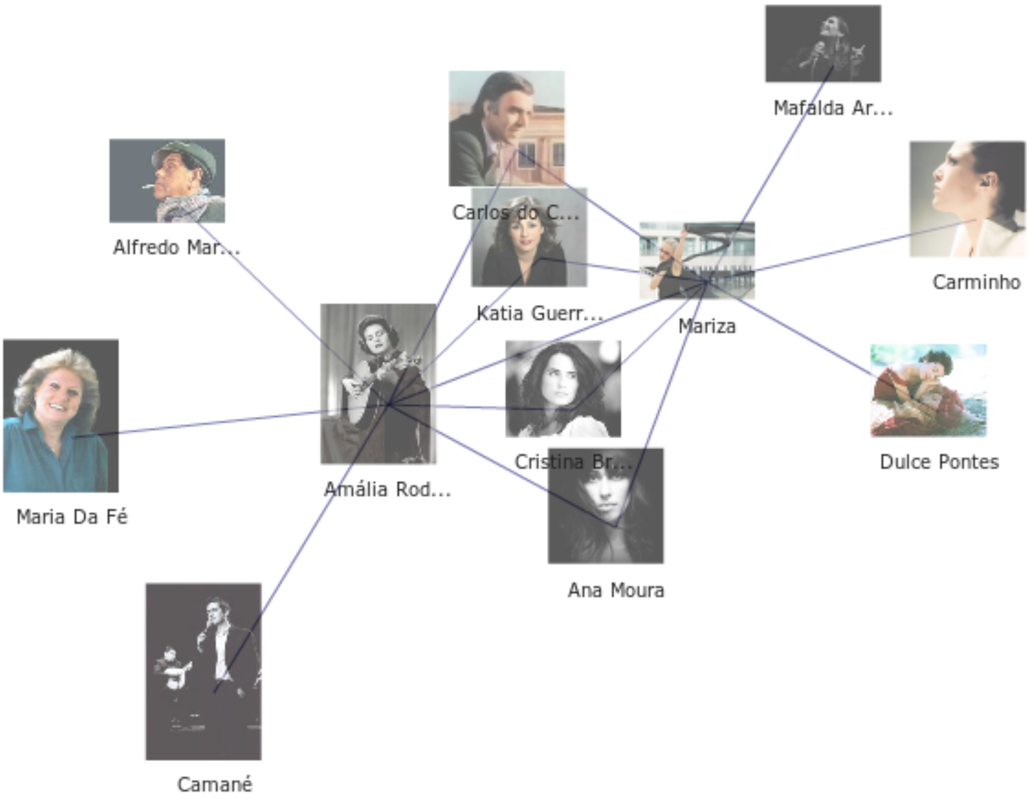

Figure 2.8: MusicRoamer: The graph after expanding one node

References

[1] P. Lamere. Creating transparent, steerable recommendations. 2008.

[2] BG Costa, Fabien Gouyon, e L Sarmento. A Prototype for Visualizing Music Artist Networks. 2008. URL: http://www.inescporto.pt/~fgouyon/docs/CostaGouyonSarmento_ARTECH2008.pdf.

[3] L Sarmento e EC Oliveira. Visualizing networks of music artists with rama. *International Conference on Web ...*, 2009. URL: http://repositorio-aberto.up.pt/bitstream/10216/15194/2/18675.pdf.

[4] Diogo Costa, Luis Sarmento, e Fabien Gouyon. RAMA : An Interactive Artist Network Visualization Tool. (i):2, 2009. URL: http://ismir2009.ismir.net/proceedings/LBD-2.pdf.

[5] Fabien Gouyon, Nuno Cruz, e Luis Sarmento. A last.fm and youtube mash-up for music browsing and playlist edition. 2011.