

Busca de expressões em músicas usando Perl

Alunos: Krishynan Shanty, Nicholas Quagliani

25 de novembro de 2016

1 Introdução

A linguagem Perl oferece muitos recursos para manipulação de arquivos e expressões regulares. Utilizamos disso para construção de um módulo com funções para extrair informações de interesse de um modelo de arquivo de música e para formulação de uma função de busca entre todas músicas em um diretório.

2 Definição do Modelo

O modelo de uma música deve ser, onde a terceira linha é a data de lançamento da música:

```
Autor:Exemplo de Autor
Titulo:Exemplo de Titulo
01/01/0000
Letra:Exemplo de Letra
Permite novas linhas

E tambem ate UMA linha vazia
```

A primeira função foi feita para extrair o autor do arquivo de texto. A função recebe como parâmetro um escalar contendo todas as linhas do arquivo. Todas as funções utilizam a comparação sem se importar com maiúsculas ou minúsculas.

```
sub EncontraAutor{
    my $arquivo = $_[0];
    my ($autor);

    ($autor) = ($arquivo =~ /^Autor:([\n\r]*)/i); #No inicio do arquivo,captura tudo ate o pula
    linha
    return $autor;
}
```

Analogamente,a segunda função foi feita para extrair o titulo do arquivo de texto. A função recebe como parâmetro um escalar contendo todas as linhas do arquivo.

```

sub EncontraTitulo{
    my $arquivo = $_[0];
    my ($titulo);

    ($titulo) = ($arquivo =~ /\n\r]Titulo:([\n\r]*)/i); #Dessa vez esta apos um pula linha, captura
    tudo ate o proximo pula linha
    return $titulo;
}

```

A função da data basta pegar o padrão de dígitos e barras pois não ocorrerá em outro local do arquivo. Ela devolve o dia,mês e ano como 3 escalares separados.

```

sub encontraData{
    my $arquivo = $_[0];
    my ($dia,$mes,$ano);

    ($dia, $mes, $ano) = ($arquivo =~ /(\d\d)\/(\d\d)\/(\d\d\d\d)/);
    #Dois digitos seguidos de barra seguido de dois digitos seguido de barra seguido de quatro digitos
    return ($dia,$mes,$ano);
}

```

A função de encontrar letra ou trecho recebe então dois argumentos, o escalar com as linhas do arquivo e um segundo argumento com a palavra ou trecho a ser procurado. A função retira os caracteres pula linha para garantir que o trecho será encontrado mesmo que na organização da letra, estejam em linhas diferentes.

```

sub EncontraLetra{
    my $arquivo = $_[0];
    my $trecho = $_[1];
    my $arquivoUmaLinha = $arquivo;
    my $trechoCompleto;

    $arquivoUmaLinha =~ tr{\n\n}{ } ;
    $arquivoUmaLinha =~ tr{\n}{ } ;
    if ($arquivoUmaLinha =~ /\sLetra:(.*$trecho.*)/i) {
        ($trechoCompleto) = $1 =~ /((?:['\w]+(?:,?\s)){0,10}$trecho(?:,?\s)['\w]+){0,10}/i;
        #regex bolado pra copiar um trecho com ate 10 palavras ao redor da palavra achada
        podendo ter virgula
        return $trechoCompleto;
    }
    return 0;
}

```

A função de encontrar trecho duplo recebe um argumento a mais sendo o segundo trecho. Além disso, há uma lógica para definir o trecho retornado. Se a quantidade de caracteres entre os dois trechos digitados for menor que 40 ela retorna o trecho iniciando com a primeira palavra e terminando com a segunda. Caso contrário, faz como a EncontraLetra e envia os trechos separadamente mas menores entre cada trecho.

```

sub EncontraLetraDupla{
    my $arquivo = $_[0];
    my $trecho1 = $_[1];
    my $trecho2 = $_[2];
    my $arquivoUmaLinha = $arquivo;
    my $trechoCompleto;
    my $aux;

    $arquivoUmaLinha =~ tr{\n\n}{ } ;
    $arquivoUmaLinha =~ tr{\n}{ } ;
    if ($arquivoUmaLinha =~ /\sLetra:.*($trecho1.*)$trecho2).*/i) {
        if (length ($2) < 40) { #envia o trecho entre as palavras procuradas se nao for grande
            return $1;
        }
        ($trechoCompleto) = $arquivoUmaLinha =~
            /((?:[^\w]+(?:,?\s)){0,6}$trecho1(?::(?:,?\s)[^\w]+){0,6})/i; #se nao procura ate mais 6
            palavras entre o trecho
        $trechoCompleto .= "_(...)_" ;
        ($aux) = $arquivoUmaLinha =~ /((?:[^\w]+(?:,?\s)){0,6}$trecho2(?::(?:,?\s)[^\w]+){0,6})
            /i;
        $trechoCompleto .= $aux;
        return $trechoCompleto;
    }
    return 0;
}

```

A função final é a de pesquisa global. Para escolher o tipo de pesquisa, o primeiro argumento é um escalar representando uma string. A função aceita que as palavras chave (autor,titulo,lancamento,data,pedaco e dupla) sejam escritas por extenso ou apenas a primeira letra. Ela percorre todos os arquivos de texto na pasta songs e faz a pesquisa nos válidos. No caso de autor, se for encontrado o nome exato do artista, todas músicas daquele artista são impressas. Caso o trecho seja uma expressão não exata de outros artistas, o nome deles será sugerido garantindo que não há repetição. No título, se for exato imprime na tela todas informações (o arquivo inteiro) senão imprime os títulos possíveis. Para data está apenas implementado imprimir o título das músicas que tem o ano igual ao sugerido. Por último, na pesquisa por trecho ou trechos imprime o título das músicas e o trecho encontrado para ajudar no reconhecimento.

```

sub PesquisaGlobal{
    my $tipoPesquisa = $_[0];
    my $procurado = $_[1];
    my $termo1 = $_[2];
    my ($achados,$atual,$dia,$mes,$ano);
    $achados = "";

    foreach my $arquivos (glob("songs/*.txt")) {
        open FILE, $arquivos or die "Couldn't open file:$_!";
        my $arquivo = do {local $_/<FILE>};
    }
}

```

```

if (!ValidaArquivo($arquivo)){
    if ($tipoPesquisa =~ /a(?:utor)?/i) {
        $atual = EncontraAutor($arquivo);
        if ($atual =~ /^$procurado$/i) { #se for exato, devolve todas musicas que
            forem com o autor exato
            $achados .= "Musica_dele:";
            $achados .= EncontraTitulo($arquivo);
            $achados .= "\n";
        }
        else {
            if ($atual =~ /[\\w\\s]*$procurado[\\w\\s]*/i) { #se nao for exato
                mas ainda acha
                if (!$achados =~ /[\\w\\s]*$atual[\\w\\s]*/i){ #confere se
                    ja nao sugeriu esse artista antes
                    $achados .= "Artista_possivel:$atual\n";
                }
            }
        }
    }
    if ($tipoPesquisa =~ /t(?:itulo)?/i) {
        $atual = EncontraTitulo($arquivo);
        if ($atual =~ /^$procurado$/i) { #exato mostra a musica inteira
            $achados .= $arquivo;
            $achados .= "\n";
        }
        else {
            if ($atual =~ /[\\w\\s]*$procurado[\\w\\s]*/i) { #nao exato retorna
                o titulo
                $achados .= "Musica_possivel:$atual\n";
            }
        }
    }
    if ($tipoPesquisa =~ /p(?:edaco)?/i) {
        $atual = EncontraLetra($arquivo,$procurado);
        if ($atual) {
            $achados .= "Musica:";
            $achados .= EncontraTitulo($arquivo);
            $achados .= "\nTrecho:\n$atual\n";
        }
    }
    if ($tipoPesquisa =~ /d(?:upla)?/i) {
        $atual = EncontraLetraDupla($arquivo,$procurado,$termo1);
        if ($atual) {
            $achados .= "Musica:";
            $achados .= EncontraTitulo($arquivo);
            $achados .= "\nTrecho:\n$atual\n";
        }
    }
    if ($tipoPesquisa =~ /L(?:ancamento)?/i) {
        ($dia,$mes,$ano) = EncontraData($arquivo); #nao uso o dia e o mes pra
            nada ainda, fica pro futuro
    }
}

```

```

        if ($procurado==$ano) {
            $sachados .= EncontraTitulo($arquivo);
            $sachados .= "\n";
        }
    }
}
print $sachados;
}

```

3 Casos de uso

As seguintes músicas foram adicionadas na pasta songs para testar as diferentes funcionalidades, as letras não estão completas para caber aqui. Os nomes de arquivo não importam:

Autor:Bee Gees
 Titulo:Stayin' Alive
 13/12/1977
 Letra:Well, you can tell by the way I use my walk
 I'm a woman's man, no time to talk.
 Music loud and women warm,
 I've been kicked around since I was born.
 And now it's all right, it's okay,
 you may look the other way.
 We can try to understand
 the New York Times' effect on man.

Whether you're a brother or whether you're a
 mother
 you're stayin' alive, stayin' alive.
 Feel the city breakin' and everybody shakin'
 and you're stayin' alive, stayin' alive.

Autor:Bee Gees
 Titulo:How Deep is Your Love
 01/10/1977
 Letra:I know your eyes in the morning sun
 I feel you touch me in the pouring rain
 And the moment which you wander far from me
 I wanna feel you in my arms again

And you come to me on a summer breeze
 Keep me warm in your love then you softly leave
 And it's me you need to show
 How deep is your love

Autor:Michael Jackson
 Titulo:Beat it
 30/11/1982
 Letra:They told him don't you ever come around
 here
 Don't wanna see your face, you better disappear
 The fire's in their eyes and their words are really
 clear
 So beat it, just beat it.

You better run, you better do what you can
 Don't wanna see no blood, don't be a macho man
 You wanna be tough, better do what you can

Autor:Michael Buble
 Titulo:Everything
 23/05/2007
 Letra:You're a falling star
 You're the getaway car
 You're the line in the sand
 When I go too far
 You're the swimming pool
 On an august day
 And you're the perfect thing to say
 And you play it coy but it's kinda cute
 Oh when you smile at me

Temos duas músicas de artistas com Michael no nome e duas músicas do artista Bee Gees. Foi também adicionado um arquivo que não era válido. O seguinte programa simples foi feito para gerar os argumentos para a função PesquisaGlobal:

```
use warnings;
use strict;
use songmanager 'all'; #adiciona o modulo criado com as funcoes

my ($tipoPesquisa,$procurado,$termo1);
print "Bem vindo ao Song Manager.\n";
print "A(utor)-> Pesquisa por um autor. Retorna todas musicas do autor se for exato\n";
print "T(itulo)-> Pesquisa por um titulo. Retorna a musica inteira se for exato\n";
print "L(ancamento)-> Pesquisa por um ano de lancamento. Retorna os nomes das musicas\n";
print "P(edaco)-> Pesquisa por um trecho. Retorna o titulo e um pedaco
o do trecho achado\n";
print "D(upla)-> Pesquisa por DOIS trechos desconectados. Retorna o titulo e um pedaco do trecho
achado\n";
print "Digite por extenso ou a primeira letra da opcao desejada:\n";

$tipoPesquisa = <>;

if (!$tipoPesquisa =~ /^[atlpd]/i) {
    print "Opcao invalida";
}
if ($tipoPesquisa =~ /d/i) {
    print "Digite o primeiro trecho pra buscar:";
    chomp($procurado = <STDIN>);
    print "Digite o segundo trecho pra buscar:";
    chomp($termo1 = <STDIN>);
    PesquisaGlobal(substr($tipoPesquisa, 0, 1),$procurado,$termo1);
    #so mandei a primeira letra que ai a pessoa pode digitar errado. so pra testar mesmo
}
else {
    print "Digite o que ira buscar:";
    $procurado = <>;
    PesquisaGlobal(substr($tipoPesquisa , 0, 1),$procurado);
}
```

Exemplos de pesquisa, o símbolo >> representa que o usuário digitou o que está a seguir:

```
C:\Users\Krishynan\Desktop\perl-songmanager>perl testaModulo.pl
Bem vindo ao Song Manager.
A(utor)-> Pesquisa por um autor. Retorna todas musicas do autor se for exato
T(itulo)-> Pesquisa por um titulo. Retorna a musica inteira se for exato
L(ancamento)-> Pesquisa por um ano de lancamento. Retorna os nomes das musicas
P(edaco)-> Pesquisa por um trecho. Retorna o titulo e um pedaco do trecho achado
D(upla)-> Pesquisa por DOIS trechos desconectados. Retorna o titulo e um pedaco do trecho achado
Digite por extenso ou a primeira letra da opcao desejada:
>>A
Digite o que ira buscar:>>MICHAEL
Artista possivel:Michael Jackson
Artista possivel:Michael Buble
```

```
C:\Users\Krishynan\Desktop\perl-songmanager>perl testaModulo.pl
Bem vindo ao Song Manager.
A(utor)-> Pesquisa por um autor. Retorna todas musicas do autor se for exato
T(itulo)-> Pesquisa por um titulo. Retorna a musica inteira se for exato
L(ancamento)-> Pesquisa por um ano de lancamento. Retorna os nomes das musicas
P(edaco)-> Pesquisa por um trecho. Retorna o titulo e um pedaco do trecho achado
D(upla)-> Pesquisa por DOIS trechos desconectados. Retorna o titulo e um pedaco do trecho achado
Digite por extenso ou a primeira letra da opcao desejada:
>>a
Digite o que ira buscar:>>bee gees
Musica dele:Stayin' Alive
Musica dele:How Deep is Your Love
```

```
C:\Users\Krishynan\Desktop\perl-songmanager>perl testaModulo.pl
Bem vindo ao Song Manager.
A(utor)-> Pesquisa por um autor. Retorna todas musicas do autor se for exato
T(itulo)-> Pesquisa por um titulo. Retorna a musica inteira se for exato
L(ancamento)-> Pesquisa por um ano de lancamento. Retorna os nomes das musicas
P(edaco)-> Pesquisa por um trecho. Retorna o titulo e um pedaco do trecho achado
D(upla)-> Pesquisa por DOIS trechos desconectados. Retorna o titulo e um pedaco do trecho achado
Digite por extenso ou a primeira letra da opcao desejada:
>>T
Digite o que ira buscar:>>stayin' Alive
Autor:Bee Gees
Titulo:Stayin' Alive
13/12/1977
Letra:Well, you can tell by the way I use my walk
I'm a woman's man, no time to talk.
Music loud and women warm,
I've been kicked around since I was born.
And now it's all right, it's okay,
you may look the other way.
We can try to understand
```

```
C:\Users\Krishynan\Desktop\perl-songmanager>perl testaModulo.pl
Bem vindo ao Song Manager.
A(utor)-> Pesquisa por um autor. Retorna todas musicas do autor se for exato
T(itulo)-> Pesquisa por um titulo. Retorna a musica inteira se for exato
L(ancamento)-> Pesquisa por um ano de lancamento. Retorna os nomes das musicas
P(edaco)-> Pesquisa por um trecho. Retorna o titulo e um pedaco do trecho achado
D(upla)-> Pesquisa por DOIS trechos desconectados. Retorna o titulo e um pedaco do trecho achado
Digite por extenso ou a primeira letra da opcao desejada:
>>p
Digite o que ira buscar:>>love
Musica:How Deep is Your Love
Trecho:
me on a summer breeze Keep me warm in your love then you softly leave And it's me you need to
Musica:Everything
Trecho:
We'll see it through And you know that's what our love can do
```

4 Conclusão

As funções foram implementadas com sucesso, permitindo a análise de vários arquivos no diretório. Foi decidido a metodologia da comparação exata ou não para evitar que imprimisse todas as músicas possíveis e as letras iriam poluir toda a informação. Da forma implementada, o usuário confirma como é o nome exato da música e basta digitá-lo inteiro para acessar a letra.