# Universidade Federal do Rio Grande do Norte

## Instituto Metrópole Digital

**2017.2**

**Linguagem de Programação 2 (IMD0040)**

# Projeto de Programação

**Resumo.** No projeto de programação deste semestre você deverá implementar um sistema de busca indexada para organizar os arquivos no seu computador, semelhante a um motor de busca, como o Google™. Neste projeto, você irá utilizar os conhecimentos adquiridos em **LP2 e EDB2** para implementar um sistema que seja capaz de fazer uma busca eficiente em palavras chaves de uma base de arquivos.

***Inspirado no trabalho proposto pelos profs. César e Jorge, durante a disciplina de LP1*

# Introdução

Neste semestre você deverá implementar um sistema de busca por indexação em uma base de textos. O objetivo é desenvolver um sistema de médio porte, utilizando o paradigma da orientação a objetos os conhecimentos adquiridos em EDB2.

# Descrição do Problema

## O problema de busca

O problema a ser resolvido neste projeto será o de realizar uma busca por conteúdo em uma base de textos. O aluno deverá implementar um sistema que possuirá duas seções principais: **indexação**, que conterá operações relacionadas à leitura, interpretação e armazenamento de uma base de arquivos de texto; e **busca**, que conterá operações relacionadas à busca de palavras na base. Na seção **indexação**, o usuário poderá gerenciar os arquivos de texto que estão contidos na base de arquivos do sistema, sendo permitido, portanto, adicionar um arquivo a base, remover um arquivo da base e listar os arquivos inseridos na base. A seção de **busca** irá realizar a busca de acordo com um conjunto de palavras chaves e irá retornar os arquivos e o local de ocorrência da chave de busca utilizada.

Como exemplo, considere a base de busca a seguir:

Arquivo **google.txt**

```
Google Inc. é uma empresa multinacional de serviços online e software dos Estados Unidos. O Google hospeda e desenvolve uma série de serviços e produtos baseados na internet e gera lucro principalmente através da publicidade pelo AdWords. A Google é a principal subsidiária da recém-criada Alphabet Inc.

A empresa foi fundada por Larry Page e Sergey Brin, muitas vezes apelidados de "Google Guys", enquanto os dois estavam frequentando a Universidade Stanford como estudantes de doutoramento. Foi fundada como uma empresa privada em 4 de setembro de 1998 e sua oferta pública inicial foi realizada em 19 de agosto de 2004. A missão declarada da empresa desde o início foi "organizar a informação mundial e torná-la universalmente acessível e útil" e seu slogan, que foi inventado pelo engenheiro Paul Buchheit, é "Don't be evil" em inglês e "Não seja mau" em português. Em 2006, a empresa mudou-se para sua atual sede, em Mountain View, Condado de Santa Clara no estado da Califórnia. O Google é executado através de mais de um milhão de servidores em data centers ao redor do mundo e processa mais de um bilhão de solicitações de pesquisa e vinte petabytes de dados gerados por usuários todos os dias.

O rápido crescimento do Google desde sua incorporação culminou em uma cadeia de outros produtos, aquisições e parcerias que vão além do núcleo inicial como motor de buscas. A empresa oferece softwares de produtividade online, como o software de e-mail Gmail, e ferramentas de redes sociais, incluindo o Google+ e os descontinuados Google Buzz e Orkut. Os produtos do Google se estendem à área de trabalho, com aplicativos como o navegador Google Chrome, o programa de organização de edição de fotografias Picasa e o aplicativo de mensagens instantâneas Google Talk. Notavelmente, o Google também lidera o desenvolvimento do sistema operacional móvel para smartphones Android, usado em celulares como o Nexus 6, Motorola Moto X e o Samsung Galaxy S6.
```

Arquivo **apple.txt**

```
Apple Inc. é uma empresa multinacional norte-americana que tem o objetivo de projetar e
```

comercializar produtos eletrônicos de consumo, software de computador e computadores pessoais. Os produtos de hardware mais conhecidos da empresa incluem a linha de computadores Macintosh, o iPod, o iPhone, o iPad, a Apple TV e o Apple Watch. Os softwares incluem o sistema operacional Mac OS X, o navegador de mídia iTunes; a suíte de software multimídia e criatividade iLife; a suíte de software de produtividade iWork; Aperture, um pacote de fotografia profissional; Final Cut Studio, uma suíte de vídeo profissional e produtos de software; Logic Studio, um conjunto de ferramentas de produção musical; o navegador Safari; e o iOS, um sistema operacional móvel.

Em agosto de 2010, a empresa operava 301 lojas de varejo em dez países, e uma loja online onde os produtos de hardware e software são vendidos. Para além das Apple Store, a empresa possui as Apple Shops e as Apple Premium Resellers (APR's). As primeiras são pequenas áreas exclusivas à marca, devidamente sinalizadas e inseridas em operadores multimarca. As APR's são parcerias estabelecidas com empresários locais e são lojas exclusivas à marca e que disponibilizam toda a gama de produtos e serviços colocados ao dispor do cliente pela casa-mãe. Em maio de 2011, a Apple era uma das maiores empresas do mundo e a empresa de tecnologia mais valiosa do planeta, tendo ultrapassado a Microsoft. Em janeiro de 2012 a Apple passou a multinacional do petróleo ExxonMobil em valor de mercado e passa a ser a maior empresa de capital aberto do mundo.

Fundada em 1 de abril de 1976 em Cupertino, Califórnia, e incorporada 3 de janeiro de 1977, a empresa foi anteriormente denominada Apple Computer, Inc. nos seus primeiros 30 anos, mas retirou a palavra "Computer" em 9 de janeiro de 2007, para refletir a contínua expansão da empresa no mercado de eletrônicos de consumo, além de seu foco tradicional em computadores pessoais.Em setembro de 2010, a Apple tinha 46,6 mil empregados em tempo integral e 2.800 temporários empregados em tempo integral em todo o mundo e tinha vendas anuais mundiais de 65,23 bilhões de dólares.

Arquivo **microsoft.txt**

Microsoft Corporation é uma empresa transnacional americana com sede em Redmond, Washington, que desenvolve, fabrica, licencia, apoia e vende softwares de computador, produtos eletrônicos, computadores e serviços pessoais. Entre seus produtos de software mais conhecidos estão as linhas de sistemas operacionais Windows, a linha de aplicativos para escritório Office e o navegador Internet Explorer. Entre seus principais produtos de hardware estão os consoles de videogame Xbox, a série de tablets Surface e os Smartphones Microsoft Lumia, antiga Nokia. É a maior produtora de softwares do mundo por faturamento, e uma das empresas mais valiosas do mundo.

A Microsoft foi fundada por Bill Gates e Paul Allen em 4 de abril de 1975 para desenvolver e vender interpretadores BASIC para o Altair 8800. A empresa posteriormente iria dominar o mercado de sistemas operacionais de computadores pessoais com o MS-DOS, em meados da década de 1980, seguido pelo Microsoft Windows. A oferta pública inicial da empresa, em 1986, e o subsequente aumento no preço de suas ações, tornou bilionários e milionários cerca de um terço dos 12 mil funcionários da Microsoft. É considerada a terceira empresa startup de maior sucesso de todos os tempos em termos de capitalização de mercado, receita, crescimento e impacto cultural. Desde os anos 1990, tem diversificado cada vez mais o mercado de sistemas operacionais e tem feito uma série de aquisições de empresas. Em maio de 2011, a Microsoft adquiriu a Skype Technologies por 8,5 bilhões de dólares, em sua maior aquisição até aquela data. Em 2014 também finalizou a compra da fabricante de celulares Nokia.

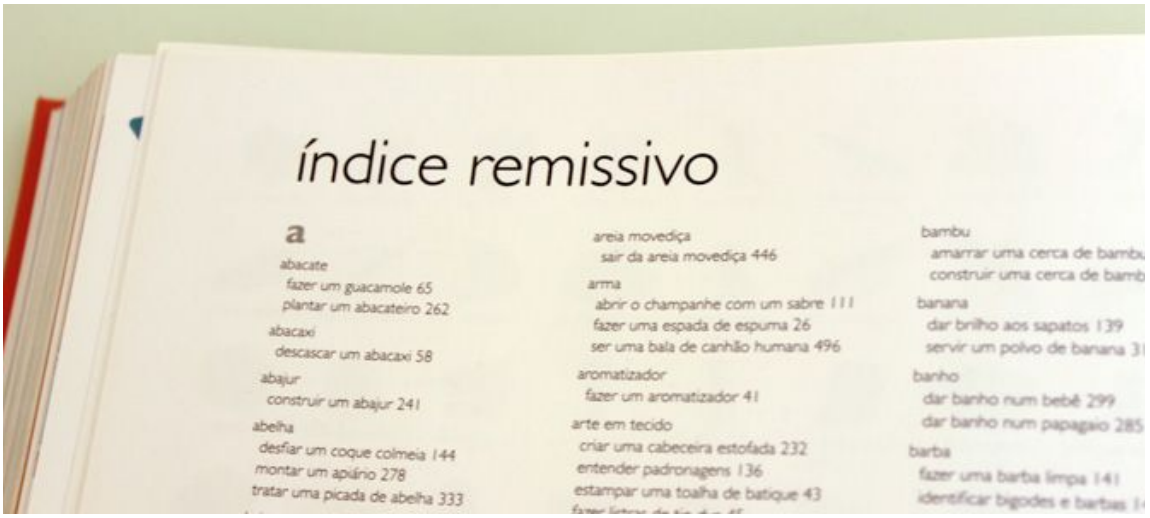
Em 2014, a Microsoft é dominante tanto em sistemas operacionais IBM PCs compatíveis quanto em programas para escritório (este último com o Office). A empresa também produz uma grande variedade de outros softwares para desktops e servidores e é ativa em áreas como pesquisa na internet (com o Bing), indústria de videogames (com os consoles Xbox), mercado de serviços digitais (através do MSN) e de telefones celulares (através da Nokia e do Windows Phone). Em junho de 2012, a Microsoft entrou pela primeira vez no mercado de produção de computadores pessoais, com o lançamento do Surface, uma linha de computadores tablet.

Neste exemplo, a base de dados deve ser criada a seção do **indexação** do sistema, indicando a adição dos arquivos *google.txt*, *apple.txt* e *microsoft.txt*. Após esta ação, uma busca pode ser realizada na seção **busca** indicando uma palavra chave, como, por exemplo, "*computador*". Esta operação deve retornar informações sobre a ocorrência da palavra "*computador*" na base de dados, como exibido abaixo.

apple.txt: linha 1
microsoft.txt: linha 1

## Busca indexada

Um conceito importante a ser aplicado no projeto é o da indexação da base de arquivos. A indexação é um processo de no qual se constrói um índice do conteúdo da base de busca para facilitar a busca. Este índice é equivalente a um índice remissivo em um livro de receitas. Se você quer saber em que página existe uma receita que use abacate, é mais rápido buscar no índice remissivo a página em que a palavra "*abacate*" aparece do que folhear todo o livro na busca da palavra abacate. O conceito a ser aplicado no projeto é equivalente. Os arquivos da base de dados devem ser pre-processados de maneira a criar um índice (ou *index*) com a ocorrência das palavras que constituem a base de dados.



Índice remissivo: ao final de um livro, existe uma seção em que palavras chaves são organizadas em ordem alfabética e são listadas as páginas do livro em que cada página ocorre.

A etapa de pre-processamento deve ser aplicada durante a **indexação** na adição de um arquivo à base de arquivos. A indexação consiste na criação de um índice de ocorrência de uma palavra na base de dados, como por exemplo considerando os arquivos google.txt, apple.txt e microsoft.txt:

computador apple.txt:1 ; microsoft.txt:1;
gmail google.txt:3;
mercado apple.txt:2; apple.txt:3; microsoft.txt:2; microsoft.txt:3;

## Estrutura de Dados

À medida que os arquivos forem processados na **indexação** o programa deverá construir uma árvore balanceada em memória, contendo em cada um dos seus nós um registro com uma palavra e os arquivos associadas a ela (tal qual foi descrito na seção anterior). Após o término do processamento os registros da árvore deverão ser persistido em um arquivo de

texto. Assim, sempre que for iniciado, o programa irá recriar a árvore à partir do arquivo que representa a sua base, permitindo que operações na seção de **busca** realizem uma busca na árvore para localizar as palavras de interesse. A escolha da árvore ficará por conta dos alunos.

## Especificação do Projeto

Seu projeto será um programa com uma interface gráfica, contendo duas seções: **indexação** e **busca**. O módulo de **indexação** permitirá gerenciar quais arquivos compõem a base de arquivos e irá criar o índice de busca. O módulo de **busca** irá realizar uma busca de uma ou múltiplas palavras chaves na base de busca utilizando o índice.

### A interface com o usuário

#### indexação

O programa deverá permitir a adição e remoção de arquivos na base, ou ainda atualização dos índices associados aos arquivos que compõem a base de dados, recriando assim o índice; esta opção é útil no caso do conteúdo de um arquivo seja modificado.

O programa também deverá permitir visualizar a base de arquivos através de opções de listar os arquivos que compõem a base de dados em ordem alfabética, indicando o nome do arquivo e a quantidade de palavras contidas nele.

É um requisito ainda do sistema, que seja lido um arquivo de configuração com uma lista negra (blacklist) de palavras. Esta lista pode conter desde palavras comuns na língua portuguesa (como artigos, preposições, etc) a palavras indesejadas (como palavras de baixo calão).

#### busca

O programa **busca** irá realizar a busca em si. Ele terá dois modos de funcionamento: <OR> ou <AND>. No modo <OR>, o programa deverá retornar todas as ocorrências em que pelo menos uma das palavras-chave seja encontrada. No modo <AND>, o programa deverá retornar todas as ocorrências nos arquivos em que todas as palavras chaves sejam encontradas. Como resultado da busca, o seu programa deverá apresentar uma lista de arquivos, com as respectivas linhas em que a linha as palavras aparecem. Os arquivos devem ser ordenados de acordo com a quantidade de linhas em que as palavras aparecem; aqueles que possuem mais linhas relacionadas, devem aparecer primeiro. Caso tenham a mesma quantidade de linhas, o desempate será de acordo com a sua ordem lexicográfica.
Por exemplo, ao buscar pela palavra “computador”, teríamos como resposta:

```
microsoft.txt: 1 - “softwares de computador, produtos eletrônicos”
apple.txt: 1 - “software de computador e computadores”
```

Enquanto que buscando pelas palavras “computador microsoft” com a opção AND, teríamos como resposta:

**microsoft.txt: 1 - "softwares de computador, produtos eletrônicos"**
**microsoft.txt: 1 - "Microsoft Corporation é"**
**microsoft.txt: 1 - "os Smartphones Microsoft Lumia, antiga"**
**microsoft.txt: 2 - "A Microsoft foi fundada"**

### Desafio

**Adicionalmente, seu módulo de busca poderá utilizar a distância de *Levenshtein* para permitir que a busca ainda retorne algo mesmo quando o usuário digitar a palavra de forma incorreta. Esta é uma técnica utilizada em processadores de texto para sugerir correções (o famoso "auto-correct") para palavras com erros de digitação. Tal algoritmo normalmente sugere palavras com a menor distância da palavra digitada.**
**Mais informações sobre a distância de Levenshtein podem ser encontradas nos links abaixo:**

- **http://www.cut-the-knot.org/do_you_know/Strings.html**
- **http://www-igm.univ-mlv.fr/~lecroq/seqcomp/node2.html**

## Entrega e Avaliação

### Entrega

Seu grupo deverá submeter um arquivo compactado contendo os seguintes entregáveis:

- Código fonte do software desenvolvido, incluindo um documento README.TXT contendo instruções de como se pode compilar o código fonte.
- Manual de usuário mostrando como instalar, executar e utilizar seu programa.
- Relatório técnico.

### Avaliação

A avaliação para a disciplina de LP2 levará em conta os seguintes aspectos:

- Acompanhamento do projeto
- Relatório técnico
- Funcionamento do software da maneira solicitada, atendendo a todos os requisitos.
- Qualidade do projeto desenvolvido
- Apresentação em formato PDF

## Acompanhamento do projeto

Ao longo do andamento do projeto serão definidas datas de checkpoint onde cada grupo deverá fazer uma breve apresentação de seu progresso. Cada checkpoint terá um entregável esperado.

| | |
|---|---|
| Checkpoint 1: Tarefas a serem desenvolvidas pelo grupo para o andamento do projeto com alocação de responsáveis por cada tarefa. | 07/11/2017 |
| Checkpoint 2: Projeto OO (diagrama de classes) inicial da solução. | 09/11/2017 |
| Checkpoint 3: Implementação - fase 1 | 16/11/2017 |
| Checkpoint 4: Implementação - fase 2 | 28/11/2017 |
| Checkpoint 5: Entrega final | 04/12/2017 |

**Atenção!!!! A apresentação será realizada com o arquivo submetido no dia 04/12/2017.**

## Relatório

**O** relatório deverá ser feito seguindo o template da Sociedade Brasileira de Computação (SBC) que pode ser encontrado no seguinte endereço:

http://www.sbc.org.br/index.php?option=com_jdownloads&Itemid=195&task=view.download&cid=38

O texto do relatório técnico deverá ser coerente, coeso e objetivo, contemplando as informações suficientes e necessárias para o entendimento do software desenvolvido, e contendo pelo menos **as seguintes seções:**

a. **Introdução : contendo uma visão geral do sistema;**
b. **Funcionalidades: descrevendo as operações do sistema e a utilização de sua interface gráfica;**
c. **Descrição da solução: apresentando um diagrama de classes do sistema e as principais decisões de projeto.**
d. **Estrutura de dados: explicando a utilização da árvore no sistema, e a sua representação no arquivo, assim como uma análise de complexidade da solução adotada.**
e. **Reflexão: Esta seção deve ainda trazer alguns comentários sobre os seguintes quesitos:**
    i. **Qualidade de código: quais considerações foram levantadas acerca de questões relacionadas a acoplamento, coesão, design baseado em responsabilidade, manutenabilidade, etc, durante o projeto e implementação de seu trabalho.**
    ii. **Bugs conhecidos ou problemas**
    iii. **O que foi aprendido: uma discussão sobre o que foi aprendido ao longo da disciplina e deste projeto, como seu grupo aprendeu, o que deu certo, o que deu errado. O que você faria da mesma maneira, e o que você faria de maneira diferente.**

**Apresentação**

Cada grupo terá 20 minutos para fazer sua apresentação. Nesta apresentação, os grupos deverão demonstrar o funcionamento do programa desenvolvido. Além disso, deverão estar aptos a responder questões sobre o desenvolvimento do projeto.

Importante: Não será dada uma única nota ao grupo. Cada componente do grupo receberá uma nota de acordo com seu desempenho durante a apresentação.

O programa será avaliado como um todo, ou seja, os requisitos não receberão pontuações individualmente. Dessa forma, a falta de um ou mais requisitos acarretará na perda de pontos, que poderá ser compensada (não totalmente, claro) através de outros componentes bem desenvolvidos.

Componentes adicionais serão muito bem vistos, desde que implementados de maneira racional. Lembre-se de usar o bom senso para não transformar criatividade em bagunça.

**Critérios de avaliação e pontuação (incluindo Funcionamento e Qualidade do projeto desenvolvido)**

Neste aspecto serão considerados os diversos conceitos abordados ao longo do semestre. Cada critério abaixo tem uma pontuação máxima de 10 pontos.

Critérios a serem considerados

| **Funcionamento do software** | |
|---|---|
| **Módulo de Indexação** | |
| Gerência da base de busca (CRUD) | 20 |
| Suporte  a blacklist | 10 |
| **Módulo de Busca** | |
| Busca simples | 10 |
| Busca <AND> | 10 |
| Busca <OR> | 10 |
| Distância de Levenshtein | 20 |
| | |
| **Estrutura de dados** | |
| Uso adequado de uma estrutura | 20 |
| Persistir/carregar estrutura em/de arquivo | 10 |

| | |
|---|---|
| **Interface Gráfica de Usuário** | |
| Seção de Indexação | 10 |
| Seção de Busca | 10 |
| **Modelagem OO** | |
| Uso adequado de Herança | 10 |
| Uso adequado de Polimorfismo | 10 |
| Coesão e Acoplamento | 10 |
| Uso adequado de Tratamento de exceções | 10 |
| Documentação (Javadoc) | 10 |
| Uso adequado de padrões de projeto | 10 |
| **Relatório** | |
| Descrição do projeto e diagramas | 10 |

Acarretarão diminuição na nota:

- Presença de erros de compilação e/ou execução.
- Falta de análise de complexidade dos algoritmos implementados.
- Falta de documentação do programa em JavaDoc.
- Entrega incompleta.
- Mal uso da norma culta da Língua Portuguesa.

**Dúvidas e Dicas**

Se durante o desenvolvimento do projeto tiver alguma dúvida sobre a tarefa solicitada tente duas possibilidades. Primeiro leia este documento, ou novamente ou pela primeira vez. Segundo, procure na Internet por mais informações. Por último, pergunte ao professor. Se a sua dúvida for interessante ela e a resposta serão acrescentadas neste documento.

**Autoria, Política de Colaboração, Plágio e Duplicação de Material**

Este trabalho poderá ser desenvolvido em grupos com três ou, excepcionalmente, quatro integrantes. Eventualmente, alguns grupos poderão ser convocados para uma entrevista. O objetivo de tal entrevista é comprovar a verdadeira autoria do código entregue. Assim, qualquer um dos componentes do grupo deve ser capaz de explicar qualquer trecho do código do projeto.

O trabalho em cooperação entre alunos da turma é estimulado. Porém, esta interação não deve ser entendida como permissão para utilização de código ou parte de código de outras equipes, o que pode caracterizar a situação de plágio. Trabalhos plagiados receberão nota ZERO automaticamente, independente de quem seja o verdadeiro autor dos trabalhos infratores.

- Um dos motivos mais comuns para problemas de plágio em trabalhos de programação é deixar para fazer o trabalho de última hora. Evite isso, e tenha certeza de descobrir o que você tem que fazer (que não significa necessariamente como fazer) o mais cedo o possível. Em seguida, decida o que você precisará fazer para completar o trabalho. Isto provavelmente envolverá alguma leitura e prática de programação. Se estiver em dúvida sobre o que foi pedido pelo trabalho, pergunte ao professor da disciplina.
- Outra razão muito comum é trabalhar em conjunto com outros alunos da disciplina. Não faça trabalhos de programação em conjunto, ou seja, não utilizem um único PC, ou sentem lado a lado, principalmente, digitando código ao mesmo tempo. Discutam as diversas partes do trabalho, mas não submetam o mesmo código.
- Não é aceitável a submissão de código com diferenças em comentários e nomes de variáveis, por exemplo. É muito fácil para nós detectar quando isso for feito, e verificaremos esse caso.
- Nunca deixe outra pessoa ter uma cópia de seu código, não importando o quão desesperado eles possam estar. Sempre aconselhe alguém nesta situação a buscar ajudar com o professor da disciplina.