

Projetos:

- [br.com.leituraEscritaEncoding](#): Alguns exemplos de como lidar com **leitura e escrita** de arquivos em java, além de passar um pouco sobre o funcionamento de **encoding's e codepoints**.

- [br.com.pacoteBasicosJava](#) : Um projeto extremamente simples que aborda conceitos **básicos dos pacotes base de java**, uma breve explicação sobre o conceito de **imutabilidade em Strings**. Temos uma **introdução ao Object** focando em métodos essenciais dele, como **o equals**. E por fim uma utilização da **classe utils** para introduzir o conceito de ordenação através da interface **Comparator** *(onde assinamos um contrato de compare que podemos usar para definir o critério de ordenação dos nossos objetos)*.

-[br.com.serializacao](#): Como serializar *(o processo de serialização é transformar um objeto em um fluxo de bits e bytes, com o propósito de salvar o estado daquele objeto)* e desserializar objetos comuns e objetos próprios, entrando um pouco na **interface de Serializable**.

- [br.com.colecoes](#): Usando ordenação de maneira mais enxuta em **listas (com lambda)**. Explorando a interface List e Set*(se aprofundando em hashCode e Equals)*. Conhecendo a "collection" de **Map**.

- [br.com.java8](#): Uma boa **introdução ao conceito de Lambdas e Streams**, onde aprendemos a base para se trabalhar com esses excelentes recursos, além de mostrar o **MethodReference** que é um complemento a esse cenário que nos auxilia a ter um código elegante. Optei por deixar a **API de Date** fora pelo fato de que ela é mto necessária no dia a dia e **merece um projeto próprio**.

- [br.com.apiDataJava8](#): Agora sim entrando com um pouco mais de detalhes na **API de data do java 8**. Conhecendo as classes que foram introduzidas para se lidar com **datas e horas**, como **calcular/armazenar o tempo entre datas e horas** e como **formatar a saída de data e hora**.

-[API Reflection Java](#): Aqui entramos em um assunto grande e avançado, sendo a base de muito framework, portanto temos um próprio índice dentro dessa pasta com seus próprios projetos.

[Processamento MultiThreading](#): Outro assunto grande que merece o próprio índice, abordando programação paralela e concorrente.