

Atividade 01 - Segundo Bimestre

Alunos

André Luis Jacob Roman RA: 22014551-2

Gabriel Dezembro RA: 22201764-2

Gabriel Marassi RA: 22302091-2

Willyan Tomaz RA: 22014128-2

- **Estrutura:**
 - Contextualização do problema
 - Muitas linhas de código
 - Dificuldade de leitura e entendimento da função da classe
 - Gasto de tempo escrevendo linhas de código que podem ser simplificadas
 - Solução
 - Utilização das anotações do Lombok
 - Simplificação das classes
 -
 - Plano de ação ou venda do produto
 - Demonstração de código extenso sem as anotações do lombok, contendo getters, setters e construtores
 - Apresentação das utilidades de anotações para simplificação de classes e métodos
 - Exemplos de anotações:
 - @Getter
 - @Setter
 - @AllArgsConstructor
 - @NoArgsConstructor
 - @SneakyThrows
 - @Slf4j
 - @ToString

Fontes de sua pesquisa, vídeos do youtube e documentações:

<https://projectlombok.org/>

<https://www.dio.me/articles/como-usar-o-lombok-em-projetos-java>

<https://medium.com/javarevisited/all-the-16-lombok-annotations-explained-in-a-4-minute-article-926f71934ec6>

Roteiro

Iremos apresentar sobre uma biblioteca que soluciona o problema de gastar muitas linhas de código, e consequentemente tempo, para coisas simples, porém necessárias no Java, o Lombok. Essa biblioteca faz de forma automática, por meio de anotações simples, o trabalho de escrever getters, setters e construtores mais simples, diminuindo a verbosidade do código. Isso além de otimizar tempo por não escrever código similar repetidas vezes, facilita a leitura de métodos que podem ter uma complexidade e importância maior naquela classe, despoluindo e simplificando a estrutura da classe. Outra vantagem do Lombok é a facilidade de utilização nas principais IDEs, por exemplo no IntelliJ apenas baixando a extensão do Lombok e tendo a dependência no projeto já é possível utilizar as anotações dessa biblioteca. Além dessas anotações apresentadas o Lombok possui diversas outras que podem ser utilizadas, por exemplo a anotação de Log, que gera um logger estático chamado de log que pode ser usado nos métodos da classe que possui essa anotação, também a `@SneakyThrows` que pode ser usada em algum método para automaticamente “embrulhar” ele em um try catch sem a necessidade de escrever isso dentro do método, outra anotação que pode ser utilizada nos métodos é a `@NonNull`, que é utilizada para indicar que o parâmetro que segue essa anotação não pode ser passado como nulo para aquele método. Podemos concluir então que a utilização desta biblioteca trás diversos benefícios para o projeto que ela for utilizada, otimizando a produtividade, limpando o código e facilitando o entendimento das funções de cada método e classe.