

07

Criando o próprio template

Uma das estruturas comuns durante o desenvolvimento de um projeto é justamente a de um **logger**. Uma das implementações mais comuns é algo similar ao código abaixo:

```
final static Logger logger = Logger.getLogger(Principal.class);
```

Veja que criamos uma constante que pega a classe na qual vai ser feito o logger.

Adicionando a lib do Log4j no projeto

Baixe a [lib do Log4 \(https://github.com/alexfelipe/intellij-idea-truques-para-aumentar-sua-produtividade/raw/aula-5/lib/log4j-1.2.17.jar\)](https://github.com/alexfelipe/intellij-idea-truques-para-aumentar-sua-produtividade/raw/aula-5/lib/log4j-1.2.17.jar) e adicione no diretório **lib** do projeto que já está configurado para ser reconhecido no classpath do projeto.

Salvando um Live Template

Agora adicione o código abaixo como uma constante da classe `Principal`:

```
final static Logger logger = Logger.getLogger(Principal.class);
```

Note que o IntelliJ não faz o import da classe inicial, portanto utilize as **sugestões do IntelliJ (Alt + Enter)** e escolha o **Log4j** como opção de import. Agora já temos a nossa estrutura inicial selecione todo o código que faz a constante, use o **find action (Ctrl + Shift + A / CMD + Shift + A)** e escreva "**save as live template**".

Repara que vai aparecer uma única opção, escolha ela. Nesse instante o IntelliJ Já criou um template pra nós no grupo **user** com o seguinte valor , isso significa que precisamos atribuir os seguintes valores pra ele:

- **Abbreviation:** forma como o template vai ser chamado
- **Description:** breve descrição quando o template aparecer

Deixe o **Abbreviation** como "logger" e o **Description** como "Implementação padrão para logger". Note que o código gerado dentro do template teve o seguinte resultado:

```
final static org.apache.log4j.Logger logger =
    org.apache.log4j.Logger.getLogger(br.com.alura.bytebank.Principal.class);
```

Em outras palavras, ele já é escrito com o nome composto da classe (pacotes.NomeDaClasse) justamente para que no momento que for usado já seja feito o import se possível.

Após realizar todos os ajustes, clique em **OK**.

Testando o template

Para testar, primeiro remova a estrutura do logger do código, então, utilize o atalho **Ctrl + Alt + O** para ajustar os imports, ou seja, remover imports que não estão sendo mais utilizados.

Então, vá no corpo da classe `Principal`, isto é, fora de qualquer método e escreva `logger`, note se aparece o template, caso aparecer, tecle **Enter**. Repare se o template implementou o `Logger` do Log4j.