



Executando o modo debug

Uma das técnicas comuns para verificarmos o comportamento do nosso código é por meio da execução em modo **debug**. Nela temos a capacidade de inspecionar o nosso código a ponto de verificar o resultado de execução linha a linha.

Preparando a execução em debug

Quando queremos executar nesse modo, precisamos indicar o momento do nosso código que queremos realizar a inspeção. Para isso fazemos uso de **breakpoints**. Para marcar o breakpoint clique ao lado direito do número da linha do código que deseja inspecionar com o botão esquerdo do mouse.

Então, a partir do momento que executar em modo debug, observe que o IntelliJ apresenta uma view de debug.

Executando o modo debug

Para executar o modo debug fazemos da mesma forma que o **Run...**, porém mudamos o atalho final, portanto use o seguinte atalho **Alt + Shift + F9 / Ctrl + Shift + D** e escolha a classe `Principal`.

Observe que agora o IntelliJ parou exatamente no ponto do breakpoint, portanto podemos começar a usar a funcionalidades disponíveis para este modo.

Nevegando em modo debug

Para prosseguir neste modo, temos algumas opções disponíveis que permitem vasculhar o nosso programa de forma detalha.

Step Into

A primeira delas é justamente o **Step Into** que executa trecho a trecho do código, para usá-lo basta apenas usar o atalho **F7**.

A partir desse modo temos a capacidade de acompanhar detalhadamente o a execução do nosso programa, por exemplo neste trecho:

```
new OpcoesPagamento().mensagemDeBoasVindas();
```

Considerando o código acima, inicialmente o **Step Into** vai realizar a instância, usando novamente o **Step Into**, ele vai entrar no método `mensagemDeBoasVindas()`.

Perceba que esse modo é muito eficaz no ponto do código que queremos entender detalhe por detalhe do que está acontecendo.

Step Over

Embora sejamos capaz de navegar por todo o projeto apenas usando o **Step Into** em alguns momentos ele pode não ser tão desejado, pois durante a execução do modo debug, o código que estamos inspecionando não é de nosso interesse saber

detalhe por detalhe.

Sendo assim, também temos a capacidade de executar a linha inteira do código com o **Step Over** por meio do atalho **F8**, vejamos o exemplo do trecho abaixo:

```
LeitorCSV leitor = new LeitorCSV();  
apresentaArquivosDisponiveis(leitor);
```

Se estivermos nesse trecho de código e usarmos duas vezes o **Step Over** ele já vai executar as duas linhas de código sem entrar dentro da execução de ambas como é feito no **Step Into**.

O **Step Over** é útil justamente nos pontos de códigos que queremos apenas avançar sem querer saber os detalhes internos dos mesmos.

Considerando apenas esses dois modos de navegação, tente chegar até o ponto do código em que apresenta o problema de execução.

Ponto de atenção

Lembre-se que o programa durante sua execução espera uma resposta no console quando chega nesse ponto do código:

```
int codigoDoArquivo = pedeCodigo();
```

Portanto, se o debug parar durante a navegação é porque precisa interagir com o console enviando o código do arquivo que deseja executar, no caso, o de junho que está apresentando o problema.

