**Breakpoint**

No desenvolvimento de software, um **ponto de parada** ou **ponto de interrupção** (breakpoint) é um ponto intencional de pausa num programa de computador durante a depuração (debug). De forma geral, ele representa a habilidade de adquirir conhecimento sobre um programa durante sua execução. Durante a interrupção da execução, o programador pode inspecionar o ambiente que cerca o programa, como logs, memória, arquivos, para determinar se o programa está sendo executado como o esperado. O ponto de parada é uma ferramenta importante de depuração pois permite especificar o momento da suspensão da execução, automatizando o processo, ao invés da execução passo a passo do código. O ponto de parada também é útil para monitorar regiões limitadas de código, mesmo em projetos de grande escala.

A forma mais comum de ponto de parada é quando a execução do programa é interrompida antes de uma instrução especificada pelo programador. Entretanto, outros tipos de pontos de parada podem ser usados, como a leitura ou escrita dum endereço ou área de memória específicos, um ponto específico do tempo, um evento específico (a chegada duma mensagem na rede) ou uma expressão específica (centésima passagem pela linha X do código).

Diversos processadores incluem suporte em hardware para ponto de parada, e podem impor restrições de uso quanto ao local do ponto de parada. Mesmo sem suporte em hardware, os depuradores podem implementar pontos de parada em software. Para pontos de parada em instruções é uma tarefa simples substituir a instrução na localização do ponto de parada. Entretanto, implementar em software pontos de parada para áreas de memória pode impactar no desempenho da aplicação sendo depurada.

## **Para que serve o debug (depuração) ?**

Seu objetivo é encontrar erros que podem impedir que os códigos funcionem adequadamente. Com ele, é possível determinar o que está ocorrendo dentro do código-fonte e **obter sugestões de ações de melhorias**.

Cada instrumento do desenvolvimento possui sua própria ferramenta para debugar um código. Através delas, você poderá determinar os pontos de parada, conhecidos como Break Points, para verificar o estado atual da aplicação, por exemplo.

Ou, ainda, **acompanhar o conteúdo de uma determinada variável**. Tudo isso serve para facilitar o trabalho cansativo de encontrar o bug em seu sistema.

## **Qual a importância do debug?**

O debug melhora a gestão de tempo e produtividade do profissional. Afinal, corrigir bugs manualmente pode ser uma tarefa demorada e frustrante.

Quando isso é feito por uma ferramenta, **economiza-se tempo**. Principalmente porque ela não precisa ler o código integralmente para identificar as falhas, indo direto ao ponto crítico. Com a redução do tempo, os programadores conseguem se dedicar mais aos processos de melhoria e inovação.

Além disso, como os **erros são detectados logo no início**, evita-se que grandes projetos tenham que ser refeitos ou mesmo jogados fora. Isso porque, dependendo do estágio do trabalho, principalmente quando se encontra em nível avançado, é ainda mais difícil encontrar a causa-raiz do problema e solucioná-la.