**PESQUISA SOBRE SDI (SINGLE DOCUMENT INTERFACE) E MDI (MULTIPLE DOCUMENT INTERFACE)**

**MDI – MULTIPLE DOCUMENT INTERFACE**

**Definição**

MDI (Multiple Document Interface) é um tipo de interface gráfica em que uma janela principal (container) pode conter várias janelas internas (filhas). Essas janelas internas ficam organizadas dentro do mesmo espaço de trabalho, permitindo ao usuário abrir e manipular vários documentos ao mesmo tempo.

**Características**

* Uso de um JFrame principal no Java Swing;
* Dentro dele, há um JDesktopPane (plano de fundo), que serve para conter várias JInternalFrame (janelas internas);
* O usuário pode minimizar, maximizar, mover e fechar essas janelas internas.

**Exemplos do Mundo Real**

* Adobe Photoshop: permite abrir várias imagens dentro da mesma interface;
* Microsoft Excel antigo (2003): cada planilha aberta ficava dentro da janela principal;
* IDEs como NetBeans e Eclipse: permitem a abertura de diversos arquivos em janelas internas ou abas dentro da mesma interface.

**Vantagens**

* Organização centralizada: tudo fica dentro de uma única janela;
* Facilidade para comparar e manipular vários documentos simultaneamente;
* Proporciona uma experiência de usuário eficiente em aplicações complexas, como sistemas de edição, CAD, IDEs e ERPs.

**Desvantagens**

* É mais complexo de programar e manter;
* Pode causar poluição visual quando muitas janelas internas estão abertas;
* Está em desuso atualmente, pois muitas aplicações migraram para a interface baseada em abas (tabbed interface).

**SDI – SINGLE DOCUMENT INTERFACE**

**Definição**

SDI (Single Document Interface) é um tipo de interface gráfica onde cada documento ou conteúdo é aberto em uma janela separada e independente. Não há uma janela "principal" que contenha outras; cada instância é autônoma.

**Características**

* Cada documento ou janela funciona de forma isolada, geralmente em sua própria instância da aplicação;
* Não utiliza JDesktopPane ou JInternalFrame, como ocorre no MDI – em vez disso, são utilizadas janelas independentes (JFrame, Stage, entre outras);
* A interface tende a ser mais simples, com menos elementos para gerenciar internamente.

**Exemplos do Mundo Real**

* Bloco de Notas do Windows: cada arquivo aberto aparece em uma janela separada;
* Visualizadores de imagem simples: cada imagem é aberta em sua própria janela;
* Google Chrome em janelas separadas (sem o uso de abas): cada site é exibido em uma janela independente;
* Aplicativos modernos do Microsoft Office (Word, Excel, PowerPoint): cada documento abre em sua própria janela.

**Vantagens**

* Desenvolvimento mais simples e fácil de manter;
* Isolamento de documentos, o que reduz o risco de um erro afetar todos os conteúdos abertos;
* Melhor compatibilidade com múltiplos monitores, permitindo distribuir janelas entre diferentes telas;
* Interface mais limpa e menos sobrecarregada.

**Desvantagens**

* Menor organização: a dispersão de documentos em várias janelas pode dificultar a navegação e o controle;
* Pode consumir mais recursos do sistema, especialmente com várias instâncias abertas;
* Apresenta dificuldade para comparação lado a lado dentro da mesma janela, pois não há suporte interno para isso.

**REFERÊNCIAS**

* ECKEL, Bruce. *Thinking in Java*. 4. ed. São Paulo: Bookman, 2006.
* ORACLE. *JDK Documentation - Java Swing*. Disponível em: https://docs.oracle.com/javase/tutorial/uiswing/. Acesso em: 08 set. 2025.
* MICROSOFT. *User interface design basics*. Disponível em: https://learn.microsoft.com/. Acesso em: 08 set. 2025.