

Desenvolvimento de Sistemas

Profº Mauro Ladeia

O que é IDE?

IDE, ou ambiente de desenvolvimento integrado, é um software que combina ferramentas comuns de desenvolvimento em uma única interface gráfica do usuário (GUI), facilitando o desenvolvimento de aplicações. Um IDE geralmente consiste em:

- **Editor de código-fonte:** é um editor de texto que auxilia na criação de código de software por meio de funcionalidades como destaque da sintaxe com indicadores visuais, recurso de preenchimento automático específico da linguagem e verificação de bugs durante a criação.
- **Automação de compilação local:** são utilitários que automatizam tarefas simples e repetíveis durante a criação de uma compilação local do software usada pelo desenvolvedor. São tarefas como compilação de código-fonte em código binário, criação de pacotes de código binário e execução de testes automatizados.
- **Debugger:** é um programa usado para testar outros programas e mostrar graficamente a localização do bug no código original.

Para que serve um IDE?

Os ambientes de desenvolvimento integrado ajudam os desenvolvedores a programar novas aplicações de forma rápida, já que os vários utilitários não precisam ser ajustados e integrados manualmente durante a configuração. Os desenvolvedores também não precisam passar horas aprendendo a usar cada uma das diferentes ferramentas, porque cada utilitário está localizado no mesmo workbench. Isso é especialmente útil quando há desenvolvedores novos em um projeto. Eles podem contar com o IDE para se atualizar em relação às ferramentas e fluxos de trabalho da equipe. Na verdade, o objetivo da maior parte das funcionalidades é economizar tempo: o preenchimento inteligente e a geração automática de código, por exemplo, eliminam a necessidade de digitar sequências inteiras.

Quais as vantagens de um IDE?

Outras funcionalidades comuns aos IDEs têm o objetivo de ajudar os desenvolvedores a organizar seu fluxo de trabalho e solucionar problemas. Os IDEs analisam o código no momento em que está sendo escrito. Assim, bugs causados por erro humano são identificados em tempo real. Como todos os

utilitários estão em uma única interface gráfica (GUI), os desenvolvedores podem executar as tarefas sem precisar trocar de aplicação. A maioria dos ambientes de desenvolvedor integrados também conta com destaque da sintaxe, usando indicadores visuais para diferenciá-la da gramática no editor de texto. Além disso, alguns IDEs incluem navegadores de classes e objetos, bem como diagramas de hierarquia de classes em determinadas linguagens.

É possível desenvolver aplicações sem um IDE. O desenvolvedor também pode, basicamente, compilar seu próprio IDE, integrando manualmente vários utilitários com um editor leve de textos, como Vim ou Emacs. O benefício dessa abordagem é o alto nível de personalização e controle que oferece aos desenvolvedores. No contexto empresarial, entretanto, a economia de tempo, a padronização do ambiente e as funcionalidades de automação dos IDEs modernos geralmente superam outros benefícios.

Atualmente, a maioria das equipes empresariais de desenvolvimento escolhe o IDE pré-configurado que melhor serve ao seu caso de uso. A questão, portanto, não é decidir usar ou não um IDE, e sim qual usar.

Os IDEs, dentre outras coisas, geralmente possuem:

- um editor de textos;
- um compilador (quando necessário);
- verificadores de sintaxe;
- complementos e assistentes de código;
- prompt integrado; snippets de código;
- ferramentas de auxílio para conexão a banco de dados;
- automatizador de tarefas;
- editor de interface gráfica
- Entre outros...

Quais são os tipos de IDE?

Existem inúmeros casos de usos técnicos e empresariais dos ambientes de desenvolvimento integrados, portanto são muitas as opções proprietárias e open source no mercado. De modo geral, as características mais importantes que diferenciam os IDEs são:

A quantidade de linguagens compatíveis: alguns IDEs são dedicados a uma linguagem específica, por isso acabam sendo melhores para determinado paradigma de programação. O IntelliJ, por exemplo, é conhecido principalmente como um IDE de Java. Outros IDEs suportam uma vasta gama

de linguagens. O Eclipse, por exemplo, é compatível com Java, XML e Python, dentre outras.

Quais as IDEs existentes no mercado?

Bem, é claro que existem diversos IDEs para as mais diversas linguagens de programação, além também de existirem **IDEs “híbridos”** que podem trabalhar com mais de uma linguagem. Em muitos casos, as IDEs híbridas, dependem apenas da instalação de plugins específicos. Este é o caso do famoso **Eclipse IDE**.

Entre as principais IDEs, podemos citar:

- Eclipse IDE
- JetBrains
- Visual Studio
- Embarcadero
- NetBeans
- XCode
- Android Studio

Principais IDEs

NetBeans – ambiente multiplataforma, tem como principal característica o layout simples e intuitivo. Suporta XML, C, PHP, C++, Ruby e HTML.

Notepad++ – se destaca pelo recurso que permite a identificação da linguagem que está sendo usada. Leve, tem como ponto negativo estar disponível apenas para Windows. Suporta Assembly, Ruby C, HTML, PHP, JavaScript, ASP, SQL, C++, Java, C#, XML, Objective-C, CSS, Pascal, Perl, Python e Lua.

Sublime Text – um dos mais populares, tem como principal recurso a possibilidade de instalar plugins de acordo com a necessidade do usuário. Suporta C, C++, C#, CSS, HTML, Haskell, Java, Latex, PHP, Ruby, SQL, XML, JavaScript e Groovy.

Porque usar IDEs

Bom, praticamente já comentei sobre as vantagens de usar um IDE nos tópicos anteriores. Contudo, nesse tópico, vamos reforçar e mostrar mais alguns detalhes:

- Um único ambiente para trabalhar com determinada linguagem de programação;
- Integração com as ferramentas (frameworks) que cercam determinada linguagem;
- Facilidade para o desenvolvedor escrever código, visto que o IDE muitas vezes já “escreve por si só” trechos de códigos repetitivos;
- Facilidade de configuração, visto que ela tenta entregar ao desenvolvedor tudo de modo instantâneo;
- Integração com documentação, facilitando o desenvolvedor pesquisar temas técnicos na documentação oficial da linguagem.

Porque “não” usar IDEs

Claro! Nem tudo é um mar de rosas. Como sempre, existem desvantagens também. E é aqui que as disputas e discussões se iniciam. Mas tudo bem, vamos listar algumas desvantagens e aproveitar para comentar sobre tais aspectos.

- **O IDE geralmente é um ambiente lento.**
Por se tratar de um programa que vai unir todo ambiente necessário para um desenvolvedor, o IDE muitas vezes torna-se lento, visto que ele facilmente gerencia dezenas de aplicativos que estendem suas funcionalidades. Nesse caso, a lentidão acaba prejudicando o desenvolvedor;
- **“Acostuma mal” o desenvolvedor na hora de programar.**
Esse motivo talvez seja um dos aspectos que gerem mais discussões, visto que um IDE pode facilmente fazer um trabalho que exigiria um pouco mais de conhecimento por parte do desenvolvedor. Dessa forma, agiliza bastante o desenvolvimento, mas por outro lado faz com que o conhecimento em questão seja deixado de lado e, muitas vezes, esquecido.
- **É uma camada a mais de configurações para atingir um determinado fim.**
Quando vamos usar um IDE primeiramente devemos configurá-lo para deixar tudo em harmonia. Essa configuração inicial pode ser bem desgastante, principalmente se o desenvolvedor não estiver em seu computador pessoal, ou seja, mudar de estação de trabalho pode ser um problema.
- **De novo, “acostuma mal” o desenvolvedor no que tange a ferramenta.**
É muito comum desenvolvedores acostumados a usar uma versão X, de um determinado IDE, reclamar que a versão Y não possui tal funcionalidade. Ou ainda, o desenvolvedor depender de um IDE para realizar tarefas que são possíveis sem o IDE. Em outras palavras, o desenvolvedor acaba por ficar dependendo do IDE.

E aí... devo usar ou não!?

Bom, a resposta para essa pergunta não é tão simples, mas vou listar algumas opiniões pessoais no intuito de que você perceba algumas vantagens e desvantagens a partir de um outro ângulo. Vamos lá:

1 – Existem situações em que um IDE é essencial, com por exemplo, **projetar interfaces gráficas**. Em outras palavras, quando você precisa criar janelas onde os usuários vão interagir. Nesse caso, usar um IDE será uma grande vantagem, visto que para criar interfaces visuais usando apenas um editor de texto é um trabalho imenso. Enquanto, com uma IDE, é possível criar janelas, botões e entre outros elementos com pouquíssimos cliques.

2 – Quando você já domina uma determinada linguagem e consegue fazer qualquer coisa **SEM o uso do IDE**. Bem, isso quer dizer que um IDE trará muita produtividade para você.

3 – Muitos desenvolvedores se acostumam tanto com o uso de um IDE que já ouvi casos de candidatos, em entrevistas de emprego, que não conseguiram criar um simples algoritmo em uma determinada linguagem de programação, pois o recrutador não disponibilizou o IDE, mais precisamente, a funcionalidade de **‘Assistente de Código’**.

4 – Já ouvi relatos também de pessoas que resolverem problemas que se arrastavam por semanas, em poucos minutos, pois dominavam um IDE que praticamente solucionou o problema em poucos cliques.

Bibliografia

<https://www.redhat.com/pt-br/topics/middleware/what-is-ide>

<https://www.psafe.com/blog/o-que-serve-ide/>

<https://becode.com.br/ides-usar-ou-nao-usar/>