

# Ambientes e Qualidades de IDE

## Gramática na Compreensão de Software

### Universidade do Minho

João Gomes  
MIEI  
a74033

Tiago Fraga  
MIEI  
a74092

João Grenhas  
MEI  
pg33874

(14 de Outubro de 2018)

#### Resumo

*IDE* é o acrónimo de *Integrated Development Environment*, que na língua portuguesa pode ser entendido como *Ambiente de Desenvolvimento Integrado*. Um *IDE* é um programa que reúne características e ferramentas de apoio ao desenvolvimento de *Software*. Atualmente, está comprovado que o uso deste tipo de ferramenta a nível empresarial ajuda no processo de criação de software, bem como no aumento da produtividade, diminuição dos gastos e aumento do desempenho.

## 1 Introdução

Ao longo do presente artigo iremos elaborar uma contextualização histórica (secção 2), explicando o porquê da criação destes programas, bem como certas características (3).

Após esta breve contextualização e caracterização geral, iremos aprofundar e abordar três casos de estudo (4) efetuados pelos elementos do nosso grupo, sendo eles a plataforma **NetBeans**(4.1), o **Eclipse** (4.2) e por fim o **IntelliJ** (4.3).

Neste capítulo, iremos enumerar um conjunto de características específicas e as vantagens que cada um oferece à comunidade informática.

Em seguida, faremos uma comparação (4.4) entre eles, de forma a sintetizar todo o conteúdo estudado, terminando com as conclusões (5) acerca do artigo, onde oferecemos a nossa opinião pessoal sobre a utilização destes ambientes de desenvolvimento.

## 2 Contextualização Histórica

Antes dos ambientes de desenvolvimento integrado, os programadores só podiam escrever o seu código em editores de texto. O único meio possível de criar uma aplicação era:

1. escrever no editor de texto, gravar o ficheiro;
2. compilar no terminal e pôr a correr;
3. anotar os erros;
4. voltar ao editor para corrigir os erros.

Com o aparecimento dos *IDE's*, a criação de aplicações ficou mais facilitada, embora ainda seja utilizado o método de editor e terminal.

Em 1983, a *Borland Ltd*, adquiriu um compilador de *Pascal* e lançou-o como *TurboPascal*, que apresentou, pela primeira vez, um editor e compilador integrados.

Embora o *TurboPascal* tenha lançado a ideia de um ambiente de desenvolvimento integrado, muitos acreditam que o *Visual Basic (VB) da Microsoft*, lançado em 1991, foi o primeiro *IDE* real. O *Visual Basic* foi construído na linguagem *BASIC* mais antiga, que era uma linguagem de programação popular nos anos 80. Com o surgimento do *Visual Basic*, a programação poderia ser pensada em termos gráficos, e benefícios significativos de produtividade começaram a surgir.

## 3 Características Gerais

Após algum estudo, podemos referir que todos os ambientes de desenvolvimento integrados possuem um conjunto de características que são transversais a todos, sendo elas as seguintes:

- **Editor** - edita o código-fonte do programa escrito na(s) linguagem(ns) suportada(s) pela IDE;
- **Compilador** (*compiler*) - compila o código-fonte do programa, editado em uma linguagem específica e transforma-a em linguagem máquina;
- **Linker** - liga os vários "*pedaços*" de código-fonte, compilados em linguagem máquina, em um programa executável que pode ser executado em um computador ou em outro dispositivo computacional;
- **Depurador** (*debugger*) - auxilia no processo de encontrar e corrigir defeitos no código-fonte do programa, na tentativa de aprimorar a qualidade de *software*;

## 4 Casos de Estudo

Assim como referido no capítulo 1, os ambientes de desenvolvimento integrado que escolhemos estudar foi o **NetBeans**, o **Eclipse** e o **IntelliJ**.

Decidimos abordar estes três *IDE's* visto que são dedicados ao desenvolvimento de *software* na linguagem *Java*, facilitando assim a comparação entre eles face ao objetivo comum que os três apresentam.

Além disto, segundo alguns estudos, são também os ambientes de desenvolvimento integrado mais usados pela comunidade de programadores mundial.

### 4.1 NetBeans

O **NetBeans** é um *ambiente de desenvolvimento integrado* gratuito de código aberto para programadores de *Software Java*, *JavaScript*, *HTML5*, *PHP*, *C/C++*, *Groovy*, *Ruby*;

As características fundamentais deste *IDE* são:

- Melhor suporte para tecnologias *Java* mais recentes;
- Edição de código de forma rápida e inteligente;
- Gerir projetos de forma fácil e eficiente;
- Rápido desenvolvimento da interface do utilizador;
- Escrever código sem erros (auxílio do *IDE*);
- Suporte a várias linguagens e suporte entre plataformas;
- Vasto conjunto de *plug-ins* fornecidos pela comunidade.

### 4.2 Eclipse

O ambiente de desenvolvimento integrado **Eclipse** foi desenvolvido pela *VisualAge*, uma empresa pertencente à gigante americana *IBM*. Originalmente foi criado com o suporte da linguagem de programação *SmallTalk*, no entanto, devido a variados problemas que foram surgindo com o desenvolvimento das tecnologias de *software*, a *IBM* decidiu avançar para a sua reestruturação com o auxílio da linguagem *Java*.

O *Eclipse* além de suportar a linguagem *Java*, através da instalação de *plug-ins* é capaz de integrar no seu código linguagens como *C/C++*, *PHP*, *ColdFusion*, *Python*, *Scala* e *Android*.

Segundo vários artigos, o *Eclipse* é o ambiente de desenvolvimento integrado de *Java* mais usado em todo o mundo. Uma das causas para este título é o facto de que a sua única versão da plataforma ser *open-source*, não havendo deste modo diferenciação entre versões livres e pagas como iremos assistir no próximo capítulo 4.3.

De salientar que, neste momento, o utilizador do *Eclipse* também pode usufruir das características gerais de um ambiente de desenvolvimento integrado que foram referidas no capítulo 3.

Por fim, este *IDE* possui uma característica marcante em relação aos demais, uma vez que utiliza uma interface gráfica baseada em *SWT* em vez de *SWING*.

### 4.3 IntelliJ

O *IntelliJ IDEA* está disponível em duas edições: *Ultimate* e *Community*.

A versão *Community* é um projeto *open-source* e gratuita, mas possui menos características.

A versão *Ultimate* é comercial, sendo anunciada pelo fabricante como disponibilizando um "leque excepcional de ferramentas e características".

*IntelliJ IDEA* é um *IDE* orientado para desenvolvimento em *Java*, tendo como objetivo o desenvolvimento para a *Web*, aplicações móveis e empresariais. Mas não só: veja-se a vasta possibilidade de *plug-in*, como por exemplo o *AntLR*.

Este *IDE* orienta-se por projetos. A interface é gráfica e interativa. A disposição por painéis revela diversos itens associados ao projeto em linha, como a lista de classes implementadas.

O editor de código fonte vai fornecendo "*Code Completion*", pistas, sugestões, opções. É possível realizar saltos no código, por exemplo entre implementação e declaração. O código é colorido em função da sintaxe, destacando as coisas mais relevantes. Os erros e "suspeitas" de sintaxe também são realçados.

Além de permitir compilar e correr, mostrando o *output* numa janela de consola, disponibiliza, como seria de esperar, um *debugger*.

### 4.4 Comparação

A maior vantagem do *Eclipse* é o facto de possuir uma infinidade de *plugins*, tornando-o versátil e altamente personalizável.

A grande diferença do *NetBeans* sobre o *Eclipse* reside no facto de que o primeiro inclui integrações a bases de dados com *drivers* para *Java DB*, *MySQL*, *PostgreSQL* e *Oracle*. O seu *Database Explorer* permite criar, modificar e apagar facilmente tabelas e bases de dados no *IDE*. Outrora visto no passado como uma espécie de sombra do *Eclipse*, o *NetBeans* surgiu como um concorrente formidável, com capacidade para destronar o *Eclipse* como *IDE* mais usado pela comunidade *JAVA*.

A versão comercial *Ultimate* do *IntelliJ*, que tem como alvo principalmente o sector empresarial, tem como grande vantagem a capacidade para suportar *SQL*, *ActionScript*, *Ruby*, *Python* e *PHP*. A versão 12 dessa plataforma também possui uma nova interface *Android* para desenvolvimento de aplicações *Android*. Apesar desta versão ter uma excelente apreciação global, alguns programadores, principalmente os independentes, consideram que o seu custo é uma despesa desnecessária.

Em suma, todos os *IDEs*’ descritos em cima vêm com as suas próprias vantagens. Embora o *Eclipse* ainda seja o *IDE* mais usado, com o passar do tempo o NetBeans está a aumentar a popularidade junto da comunidade *open-source*.

## 5 Conclusão

No presente artigo, estudamos e aprofundamos o conhecimento sobre os ambientes de desenvolvimento integrado que, ao longo desta etapa no ensino superior, têm sido plataformas que recorrentemente necessitamos de utilizar.

Em primeiro lugar, procuramos explicar qual foi o objetivo e como aparecerem os primeiros *IDE*’s na história. Percebendo esta parte, avançamos para uma caracterização global dos ambientes de desenvolvimento que estão disponíveis no mercado atual.

Estabelecidos estes conhecimentos, avançamos para o estudo de casos particulares, tendo sido eles o *NetBeans*, o *Eclipse* e o *IntelliJ*, onde aprofundamos as características individuais de cada um, bem como as suas vantagens e desvantagens, estabelecendo uma comparação geral entre os três.

Em suma, podemos apreciar que cada ambiente de desenvolvimento integrado tem as suas capacidades e atributos, e a escolha de cada um deles depende do gosto pessoal do utilizador, portanto aconselhamos ensaiar os três *IDE*’s abordados no artigo, de forma a escolher o que seja da preferência do programador.

## Referências

- [1] IntelliJ Página do Fabricante - <https://www.jetbrains.com/idea/>. Consultado a 12-10-2018.
- [2] NetBeans Página do Fabricante- <https://netbeans.org/> Consultado a 11-10-2018.
- [3] NetBeans Wikipédia - <https://pt.wikipedia.org/wiki/NetBeans> Consultado a 11-10-2018.
- [4] Eclipse Wikipédia - [https://pt.wikipedia.org/wiki/Eclipse\\_\(software\)](https://pt.wikipedia.org/wiki/Eclipse_(software)) Consultado a 11-10-2018.
- [5] Eclipse Pagina do Fabricante- <https://www.eclipse.org/ide/> Consultado a 11-10-2018
- [6] Comparação de IDE’s- <https://www.lifewire.com/comparing-java-ides-eclipse-vs-netbeans-vs-intellij-2373152> Consultado a 10-10-2018