**01Instalando o Eclipse**

**Transcrição**

Por enquanto, temos o nosso primeiro programa Java escrito, e agora passaremos a entender como declarar variáveis, fazer if, laços e afins. Queremos um editor um pouco melhor do que o bloco de notas, de acordo com sua preferência.

A comunidade geral do Java costuma usar não um editor, mas um IDE (*Integrated Development Environment*, que em português seria algo como "Ambiente Integrado de Desenvolvimento"). Um IDE não é simplesmente um editor pois integra em um único local a linguagem, o editor, o compilador, a biblioteca e a documentação.

Os principais IDEs utilizados por quem programa em Java são: o ***NetBeans***, da própria Oracle, o ***IntelliJ IDEA***, usado como base para Android, e também conhecido por *Android Studio*, e o ***Eclipse***, projeto em código aberto absorvido pela IBM e, hoje em dia, um consórcio de muitas empresas que tomam conta do programa, que você pode baixar [aqui](https://www.eclipse.org/downloads).

O Eclipse quer te ajudar na hora de codificar, muito mais do que focar em *wizards* e na grande quantidade de opções de menu. À primeira vista, o IDE pode parecer pequeno demais (pelo peso que possui), mas é porque há muitos plugins instaláveis para se facilitar o desenvolvimento de *features* e recursos.

Quando formos instalá-lo, aparecerá uma janela perguntando o que queremos, e escolheremos "Eclipse IDE for Java Developers". A opção "Eclipse IDE for Java EE Developers" requer um conhecimento maior, e serve para desenvolvimento de aplicativos web e softwares, e poderá ser explorada futuramente.

Após instalação e durante a execução, a primeira pergunta que o Eclipse fará tem a ver com o *workspace*, o diretório a ser utilizado para guardar todos os projetos Java. Isto pode ficar a seu critério, lembrando que iremos trabalhar sempre no Eclipse, então isso acabará não sendo tão relevante, pois você não precisará mais do Prompt de Comando para acessá-lo.

É possível ter mais do que um *workspace*, um só para exercícios da Alura e outros para projetos da empresa, por exemplo.

A primeira execução trará muitas janelas diferentes, mesmo se fecharmos o "Help". Vamos maximizar o Eclipse e fechar a aba "Welcome".

O IDE, ao ser aberto, pode te assustar um pouco, mas você verá que assusta menos do que outros com muito mais janelas e perguntas de *wizards*. No centro, ficam os arquivos que queremos editar, do lado direito estão os "Task List" (Lista de Tarefas), embaixo, "Problems" (Problemas). À esquerda, há "Package Explorer" (Explorador de pacotes).

O Eclipse denomina este conjunto de janelas de **perspectiva**, e cada uma delas é uma ***view***. Então, veremos diversas *views* que irão nos ajudar em diferentes situações, tanto que se clicarmos em "Window > Show View" no menu superior, há várias opções. Não nos preocuparemos com isso agora.

No momento, queremos criar um projeto Java, e veremos poucos *wizards*! Para criarmos um projeto e uma classe Java, clicaremos em "File > New > Java Project" e, na nova janela, definiremos o projeto como "sintaxe-basica".

Provavelmente o Java já está instalado em seu computador; verifique sua versão, se é 8 ou posterior, pois utilizaremos recursos desta versão. Clicando em "Finish", o projeto é criado e aparecerá em "Package Explorer", contendo um diretório "src" (onde deve estar nosso código fonte), e "JRE System Library", uma biblioteca com tudo que temos e acessível pelo Java. Todos os comandos que utilizaremos estará nestes arquivos .jar.

Agora, queremos colocar nosso arquivo Java, o "Programa.class", no diretório de código fonte.

**02Mão na massa: instale a IDE Eclipse**

Como vimos nos vídeos, vamos fazer o download da IDE Eclipse, para isso, siga os passos abaixo!

1) Antes de tudo, acesse o site do [Eclipse](http://www.eclipse.org/) e clique no botão escrito **Download**.

2) Ao clicarmos, o site nos redirecionará para uma outra página, clique no botão **Download** abaixo do **Get Eclipse**

3) O site nos levará para o download de acordo com o seu sistema operacional, portanto, agora clique em Download e espere concluir.

4) A partir daqui, prossiga para a instalação de acordo com o seu sistema operacional.

**Windows**

Para instalar a IDE Eclipse no Windows, siga os seguintes passos:

4.1) Execute o arquivo que você acabou de baixar e aguarde o instalador

4.2) O instalador do Eclipse nos dará várias opções, para esse curso, devemos escolher a opção **Eclipse IDE for Java Developers**.

4.3) Nessa nova parte, clique em **Install** e aguarde. Quando a instalação for terminada, clique em **Launch** para abrir o Eclipse automaticamente.

**Linux**

Para instalar a IDE Eclipse no Linux, siga os seguintes passos:

4.1) Extraia os arquivos, e execute o arquivo "eclipse-inst"

4.2) O instalador do Eclipse nos dará várias opções, para esse curso, devemos escolher a opção **Eclipse IDE for Java Developers**.

4.3) Nessa nova parte, clique em **Install** e aguarde. Quando a instalação for terminada, clique em **Launch** para abrir o Eclipse automaticamente.

**Mac**

4.1) Extraia os arquivos, e execute o instalador (eclipse).

4.2) O instalador do Eclipse nos dará várias opções, para esse curso, devemos escolher a opção **Eclipse IDE for Java Developers**.

4.3) Nessa nova parte, clique em **Install** e aguarde. Quando a instalação for terminada, clique em **Launch** para abrir o Eclipse automaticamente.

**Opinião do instrutor**

Conseguiu? Não? Tire suas dúvidas no fórum!

**03Sobre IDEs e editores**

Podemos programar em Java utilizando editores de textos e IDEs. Nesse contexto, temos as seguintes afirmativas:

**a)** Um IDE é um ambiente integrado de desenvolvimento que centraliza em um único lugar o compilador da linguagem utilizada, editor de texto, documentação e tudo que gira em torno da criação de aplicações.

**b)** Para quem está começando, o Eclipse IDE for Java EE Developers é o mais recomendado.

**c)** NetBeans e Intellij são outros IDEs famosos do mundo Java.

**d)** Um editor de texto possui tudo o que um IDE oferece.

Marque a opção verdadeira a respeito das afirmações anteriores:

Parte superior do formulário

* C e D são verdadeiras.
* Alternativa correta

A, C são verdadeiras.

Correto! *Um IDE é um ambiente integrado de desenvolvimento que centraliza em um único lugar*, e *NetBeans e Intellij são outros exemplos famosos de IDEs Java.*

* Alternativa correta

A, D são verdadeiras.

Parte inferior do formulário

Parabéns, você acertou!

Este conteúdo foi útil para o seu aprendizado?

Sim

Não muito

**04Eclipse Workspace**

O que sabemos sobre o workspace?

Parte superior do formulário

* Sabemos que não é possível ter diferentes workspaces no Eclipse.
* Alternativa correta

Cada projeto do Eclipse fica dentro de um workspace.

Alternativa correta, um projeto sempre fica dentro de um workspace.

* Alternativa correta

Um workspace é a pasta padrão que será utilizado para armazenar todos os projetos criados com a IDE Eclipse.

Alternativa correta, um workspace é como uma pasta para vários projetos.

Parte inferior do formulário

Parabéns, você acertou!

**05Nosso programa rodando no Eclipse**

**Transcrição**

Vamos colocar o código do programa no Eclipse, utilizando o editor mais poderoso, o ***IDE***. Para isto, há várias opções: acessando "File > New > Class", ou clicando com o lado direito do mouse em "src" e selecionando "New > Class", o que abrirá um *wizard* mas, como dito anteriormente, focaremos mais no código em si do que nas "mágicas" que os editores fazem para nós.

O nome desta classe será "Programa", porém não se preocupe ainda com as diversas opções que aparecem nesta janela. Há até um checkbox para o caso de querermos public static void main(String[] args), o que não é o caso, pois por ora queremos praticar bastante e escrever um código básico.

Clicando-se em "Finish", teremos o programa simples que escrevemos antes:

public class Programa {

public static void main(String[] args){

System.out.println("ola mundo");

}

}Copiar código

Conforme vamos digitando o código, o programa vai tentando completar, para nos ajudar. Na lateral esquerda, a bolinha vermelha com "**x**" indica erro de compilação em determinadas linhas.

O asterisco (**\***) ao lado do nome do arquivo indica que ele não foi salvo!

Para executarmos o código, basta acessarmos "Run > Run As > Java Application", o que abre uma *view* para Console, que abre e executa, no caso, o javaw.exe, uma versão do Java que não abre no Prompt do MS-DOS, utilizado internamente pelo Eclipse para chamar o nosso programa, que é o que gostaríamos de fazer neste ambiente.

Por enquanto, não há tantas vantagens em relação ao Notepad, e não é à toa que a Microsoft, quando lançou o Visual Studio, foi atrás dos recursos apresentados pelo Eclipse, com o *ReSharper*, contratando o Erich Gamma, autor de *Design Patterns: Elements of Reusable Object-Oriented Software*, para trabalhar com o IDE deles. O Eclipse foi uma inspiração para muitos, e é considerado uma ferramenta incrível.

Mas onde se encontra o "Programa.class", o *bytecode* que a *virtual machine* entende?

O "Package Explorer", que é uma *view*, esconde arquivos e diretórios que julga não serem relevantes. E faz sentido, pois no momento estamos focados no programa Java. Acessando-se "Window > Show View > Navigator", ele irá mostrar o *File System*.

Clicando em "sintaxe-basica" para abri-lo, além do "src", existem outros diretórios e arquivos, dentre os quais "Programa.class" na pasta "bin". ".classpath" e ".project" são arquivos de configuração utilizados pelo Eclipse para obter informações sobre seu projeto. Eles não devem ser editados diretamente e, clicando-se na aba "Source" na parte inferior da interface, você verá que trata-se de um .xml.

Não precisaremos nos preocupar com estes arquivos, pois é muito raro termos que mexer neles. Há muito tempo, só existia esta *view*, o "Navigator" (similar ao Windows Explorer). O "Package Explorer" surgiu para ajudar quem trabalha com Java, e o "Navigator", para quem trabalha com tudo, de forma geral. Na maioria das vezes, usaremos o "Package Explorer".

Se clicarmos em "src > Programa.java" com o lado direito do mouse, e em seguida em "Properties", veremos a localização exata do arquivo, possível de ser confirmado por meio do Prompt de Comando.

A partir de agora deixaremos de utilizar o prompt, pois faremos tudo no Eclipse. De qualquer forma, é importante lembrarmos que o Java pode ser usado por linhas de comando, se você preferir, o que acaba sendo até necessário em alguns casos.

**06Mão na massa: Rodando nosso programa no Eclipse**

Chegou a hora de rodarmos nosso código no Eclipse, siga os passos abaixo!

1) Ao inicializarmos o Eclipse, ele nos pedirá para selecionarmos um "workspace", que é onde ficarão armazenados seus projetos. Caso você queira personalizar isso, clique em **Browse...** e selecione sua pasta. Nesse curso usaremos a própria pasta **workspace**.

2) Feche a página de Welcome e vamos iniciar o Eclipse da maneira que encontramos ele normalmente.

3) Clique em **File** e coloque seu cursor sobre a linha **new** e veja que o Eclipse nos dará algumas opções. Escolha **Java project**.

4) Nessa janela nova, nomearemos nosso projeto, escolhemos o nome **sintaxe-basica**, lembre-se de verificar a versão do Java utilizada.

5) Repare que temos um novo projeto na **View** Package Explorer, guardaremos nosso projeto dentro desta pasta **src** que foi criada.

6) No diretório **src**, clique com o botão direito do mouse, coloque o cursor em **new** e escolha **class**

7) Nessa nova janela, criaremos a classe com o nome Programa, nesse momento não se preocupe com as diversas opções que o Eclipse nos oferece. Clique em **finish** e teremos nossa classe Programa.

8) Antes de colocar o tal do System.out.println, escreva o método main. Fique tranquilo, entenderemos melhor o main no decorrer do curso. Nosso código ficará assim:

public class Programa{

public static void main(String[] args){

System.out.println("ola mundo");

}

}Copiar código

9) Repare que durante a escrita, o Eclipse tenta inferir algumas coisas, além de fechar as aspas automaticamente.

10) Para executarmos nosso código, na parte de cima do Eclipse, clique em **Run**, e depois em **Run As** e escolha **Java Application**.

**Opinião do instrutor**

Tudo dando certo!? Qualquer dúvida crie um tópico no fórum!

**07Projeto Java**

Marque as alternativas verdadeiras:

Parte superior do formulário

* Por mais que estejamos usando uma IDE, ainda precisaremos verificar as mensagens de erro do nosso programa no terminal do nosso sistema operacional.
* Alternativa correta

Executamos nosso programa no Eclipse através do menu *Run -> Run as -> Java Application*.

Exato! Inclusive há uma tecla de atalho, exibida durante a opção que varia de acordo com seu sistema operacional.

* Alternativa correta

Dentro de um projeto Java, criamos uma nova classe através da opção do menu *File -> New -> Class*.

Correto! Caso o projeto não seja um projeto Java, a opção Class não estará disponível, fique atento!

* Alternativa correta

A saída do nosso programa executado pelo Eclipse é feita através da **view** console.

Correto! A **view** console mostra a saída do programa.

Parte inferior do formulário

Parabéns, você acertou!

**08Sobre a View Navigator**

Marque a alternativa correta a respeito da View Navigator:

Parte superior do formulário

* É parecida com o Windows Explorer do Windows ou o Finder do MAC.

Exato. Ela nos permite ver o diretório do projeto com seus arquivos na integra.

* Alternativa correta

Mostra menos arquivos e pastas do que o Package Explorer.

* Alternativa correta

Podemos selecioná-la através do menu Window -> Navigator.

O correto é Windows -> Show View -> Navigator.

Parte inferior do formulário

Parabéns, você acertou!

Este conteúdo foi útil para o seu aprendizado?

Sim

Não muito

**09O que aprendemos?**

Este capítulo apresentou:

* O papel de um IDE e sua diferença de um editor
* Como fazer o download do Eclipse IDE
* Para que serve um workspace
* O conceito de perspectiva
* Como criar um projeto Java, inclusive classes e como executá-lo.
* Como exibir diferentes views

**10Arquivos do projeto atual**

No link abaixo, você encontra o projeto até o momento atual do curso.

<https://github.com/alura-cursos/Curso-Java-parte-1-Primeiros-passos/archive/capitulo3.zip>

**Opinião do instrutor**

Lembre-se de usar o fórum em caso de dúvidas.