














2 Instalación e uso de contornos de desenvolvimento

Sumario

2	Instalación e uso de contornos de desenvolvemento.....	1
2.1	Convencións empregadas.....	3
2.2	Introdución.....	4
2.3	Contornos de desenvolvemento.....	4
2.3.1	Os primeiros contornos de desenvolvemento.....	5
2.3.2	Contornos de desenvolvemento actuais.....	6
2.3.2.1	NetBeans.....	6
2.3.2.2	Eclipse.....	7
2.3.2.3	Visual Studio.....	7
2.3.3	Instalación de contornos de desenvolvemento.....	8
2.3.3.1	O compilador de Java.....	8
2.3.3.2	Instalación do JDK e JRE predeterminados en Linux.....	9
2.3.3.3	Instalación de Oracle JDK 11 en Linux.....	9
2.3.3.4	Instalación de Oracle JDK 11 en Windows.....	11
2.3.3.5	Programación en Java.....	11
2.3.3.6	Instalación de Apache NetBeans en Linux.....	12
2.3.3.7	Instalación de Apache NetBeans en Windows.....	15
2.3.3.8	Javadoc.....	16
2.3.4	Instalación de plugins en NetBeans.....	16
2.3.4.1.1	Instalación desde o catálogo de plugins.....	17
2.3.4.1.2	Instalación desde arquivos .nbm.....	17
2.4	Actividades.....	18

2.1 Convencións empregadas

	Esta icona fai referencia a notas de introdución
	Esta icona indica aclaración
	Esta icona fai referencia a arquivos de configuración, de rexistro...
	Esta icona indica casos de uso
	Esta icona fai referencia a avisos o advertencias
	Esta icona indica incidencias
	Esta icona fai referencia a sección que inclúen instrucións paso a paso
	Esta icona fai referencia a sección que inclúen capturas de pantalla
	Esta icona fai referencia a actividades
	Esta icona fai referencia a documento esencial (licenza: http://www.ohmyicons.com)
	Esta icona fai referencia a enlace recomendado (licenza: http://iconleak.com)

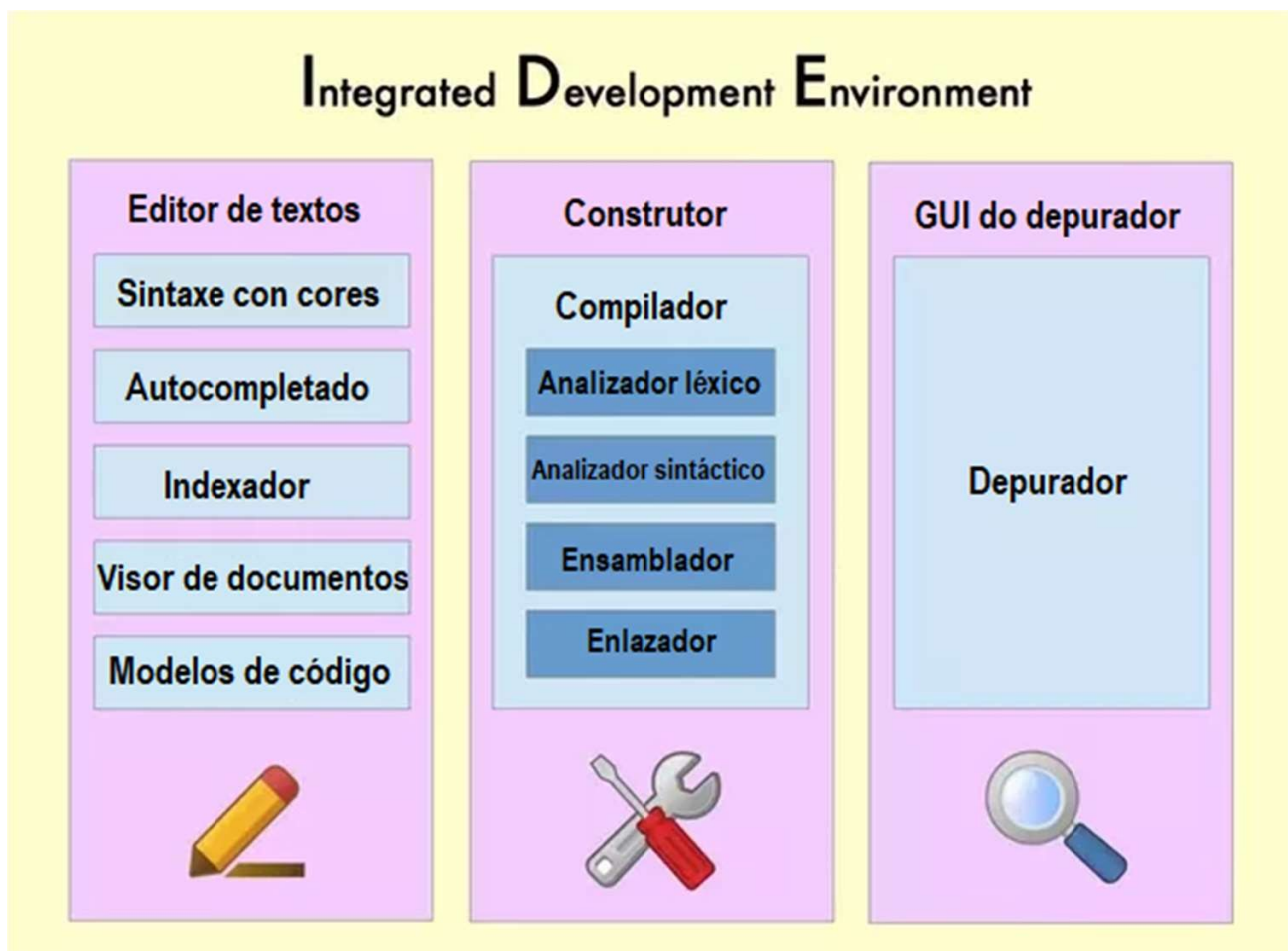
2.2 Introducción

Un contorno de desenvolvemento ou IDE (Integrated Development Environment) ou EDI (Entorno de Desarrollo Integrado) é unha aplicación informática que asiste ao técnico informático na tarefa de deseñar e codificar software mediante a inclusión de múltiples ferramentas integradas e relacionadas. Este tipo de aplicación utilízase en diferentes momentos do desenvolvemento de software: no deseño, codificación e probas

2.3 Contornos de desenvolvemento

Os contornos de desenvolvemento son as ferramentas coas que os programadores crean aplicacións. Aínda que pode programarse cun editor e un compilador, nos contornos profesionais soe utilizarse un IDE xa que axudan a aumentar a produtividade dos programadores.

Aos contornos de desenvolvemento chámaselles con frecuencia IDE (Integrated Development Environment – Contorno de Desenvolvemento Integrado) e apórtannos unha serie de ferramentas para que nos resulte máis doado programar e, normalmente, contan cun resaltado de sintaxe por defecto.



Un IDE é un contorno de programación empaquetado como un programa de aplicación e que, normalmente, consiste nun editor de texto, un compilador, un depurador e, con frecuencia, un construtor de interface gráfica.

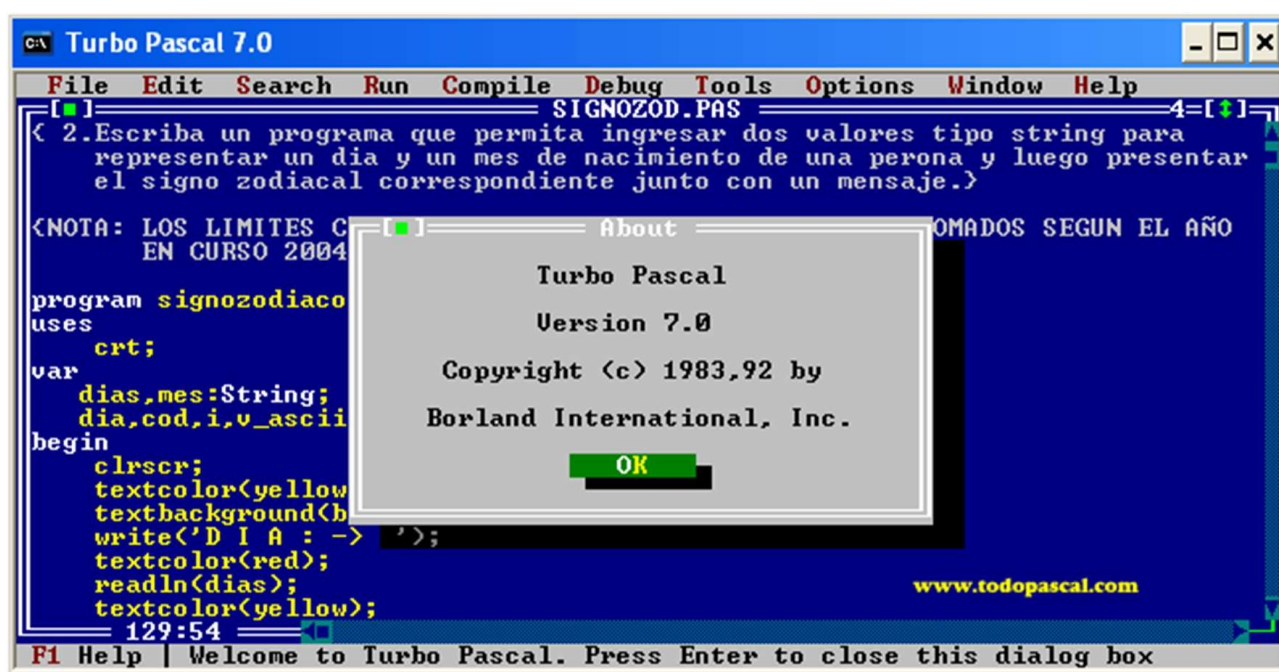
O IDE agrupa as ferramentas indicadas normalmente nun contorno visual, de forma que o programador non necesite introducir múltiples comandos para compilar, interpretar, depurar, etc. Habitualmente teñen unha avanzada interface gráfica de usuario (GUI).

Os IDE ofrecen un marco de traballo amigable para a maioría das linguaxes de programación como poden ser Java, C++, C#, Python, Delphi, etc.

Un mesmo IDE pode traballar con varias linguaxes de programación.

2.3.1 Os primeiros contornos de desenvolvemento

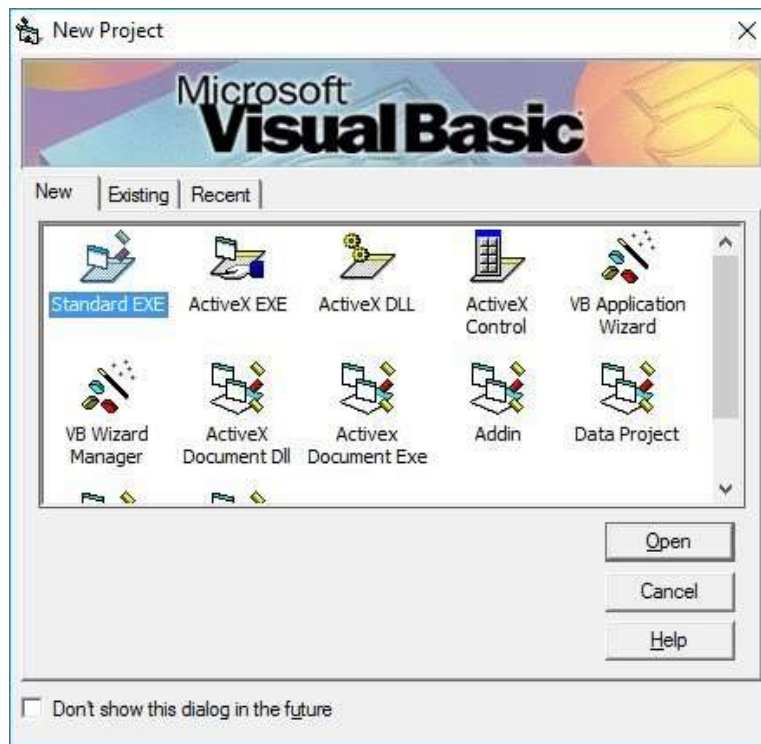
Turbo Pascal. Foi lanzado pola empresa Borland en 1983 e foi o IDE máis potente da súa época. Durante a década dos 80 e ata principios dos 90 foi moi popular debido ás súas magníficas extensións e aos tempos de compilación sumamente curtos. Foi unha revolución para a súa época. A rapidez de compilación era asombrosa. De feito, os compiladores actuais son máis lentos.



Lanzáronse sete versións e, na última, podía usarse o rato. Na versión inicial comprimía nuns 50KB un editor, un compilador e un enlazador para crear arquivos executables.

Tras o éxito desta ferramenta, Borland creou novas ferramentas como Delphi, baseada na mesma linguaxe de programación, Pascal.

Microsoft Visual Basic 6. Foi un dos IDE máis utilizados na súa época. Este novo tipo de ferramenta creou o paradigma de desenvolvemento RAD (Rapid Application Development – Desenvolvemento Rápido de Aplicacións).



No paradigma RAD desenvólvanse primeiro de maneira rápida as interfaces que se consensuaban co usuario. Cando os desenvolvedores tiñan o visto bo das interfaces, realizaban a programación. As interfaces creábanse a partir dunha serie de compoñentes que proporcionaba a propia ferramenta. VB6 só funcionaba no sistema operativo Windows.

2.3.2 Contornos de desenvolvemento actuais

2.3.2.1 *NetBeans*

NetBeans é un IDE escrito en Java. É un contorno moi utilizado polos programadores. Trátase dun contorno multilinguaxe e multiplataforma no que poder desenvolver software de calidade. Permite desenvolver aplicacións de escritorio, aplicacións móbiles e web. É un IDE libre e gratuito, sen restricións de uso.



NetBeans é un dos IDE máis famosos para programar en Java.

NetBeans comezou como proxecto estudantil de estudantes universitarios de matemáticas e física en 1996 na república Checa. Sun Microsystems adquiriu NetBeans no ano 2000. Posteriormente, NetBeans pasou a ser da compañía Oracle cando Oracle comprou en 2010 Sun Microsystems. No ano 2016 Oracle enviou á incubadora de proxectos da organización sen fins de lucro Apache Software Foundation (ASF) o proxecto NetBeans. Desenvolvéronse 3 versións de NetBeans desde que Oracle donou NetBeans á ASF. A versión 11.0 de Apache NetBeans foi lanzado a principio do mes de abril de 2019.

A linguaxe que mellor soporta é Java, xa que inicialmente foi creado para ser o IDE de Java. Non obstante é multilinguaxe xa que soporta JavaScript, HTML5, PHP, C/C++, etc. Ten bastantes módulos (plugins) que permiten ampliar as súas funcionalidades.

É multiplataforma xa que está dispoñible para un grande número de sistemas operativos como Windows, Linux ou MAC OS X.

2.3.2.2 Eclipse

Eclipse é un dos contornos de desenvolvemento máis coñecidos e utilizados polos programadores xa que se trata dun contorno de desenvolvemento de código aberto e multiplataforma. Eclipse creouse a partir do Visual Age de IBM pero agora manteno a Fundación Eclipse, que é independente e sen ánimo de lucro. Está soportado por unha comunidade de usuarios o que fai que teña moitos módulos (plugins) que permiten que sirva para case calquera linguaxe de programación. Neste aspecto, é un dos mellores. Utilízase para Java, C++, PHP, Perl e moitas linguaxes máis.



Hai moitos tutoriais na rede tanto de instalación como de utilización.

O seu punto débil é que está lonxe de ser un contorno no que sexa sinxelo iniciarse e, polo tanto, para usuarios pouco experimentados pode ser un pouco duro.

2.3.2.3 Visual Studio

Foi deseñado por Microsoft é trátase dun dos mellores contornos de programación que existen, sempre e cando se utilicen linguaxes de Microsoft. Microsoft ten creado o Visual Studio Community que é moi parecido ao Visual Studio de pagamento só que está soportado pola comunidade. Este contorno permite facer aplicación web e de escritorio e axuda moito ao programador. O inconveniente que ten é que só é válido para linguaxes de Microsoft tales como Visual C++, Visual C#, ASP.NET e Visual Basic.NET. Actualmente téñense desenvolvido as extensións para moitas outras.



Visual Studio permite as desenvolvedores crear aplicacións, sitios e aplicacións web, así como servizos web en calquera contorno que soporte a plataforma .NET.

2.3.3 Instalación de contornos de desenvolvemento

2.3.3.1 O compilador de Java

O compilador de Java, tamén chamado javac, encapsúlase dentro dun paquete de desenvolvemento chamado JDK (Java Development Kit – Equipo de Desenvolvemento de Java).

Para programar en Java necesítase o compilador e, polo tanto, haberá que instalar o seu JDK na máquina onde vaia facerse o desenvolvemento.

Para executar os programas desenvolvidos en Java, o sistema onde se execute deberá ter un JRE (Java Runtime Environment – Contorno de Execución de Java) que conterá a JVM (Java Virtual Machine – Máquina Virtual de Java). Java é multiplataforma, polo tanto, non hai que compilar cada programa para cada sistema operativo, xa que, cando se compila un programa, funcionará en calquera sistema operativo, sempre e cando teña instalada a JVM correspondente.

Para saber se temos instalada a JVM no noso sistema podemos usar o seguinte comando na liña de comandos:

```
C:\>java -version
java version "1.8.0_221"
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)
Java HotSpot(TM) Client VM (build 25.221-b11, mixed mode, sharing)
```

Para saber se temos instalado o JDK podemos usar o seguinte comando:

```
C:\>javac -version
javac 1.8.0_91
```

Hai 3 edicións de Java:

- Java Standard Edition (Java SE)
- Java Enterprise Edition (Java EE)
- Java Micro Edition (Java ME).

Java SE é unha plataforma que permite a execución e despregamento de aplicacións Java en computadoras de escritorio e servidores. Os dous produtos de Oracle que implementan a plataforma Java SE son o JDK (Java SE Development Kit) e o JRE (Java Runtime Environment).

Java EE é unha plataforma que está construída sobre a plataforma JSE. Esta plataforma subministra APIs adicionais e o contorno para o desenvolvemento e execución de aplicacións empresariais (as que requiren operacións a grande escala, nunha topoloxía distribuída, escalable, confiable e segura).

Java ME é o subconxunto mínimo de tecnoloxías e especificacións de Java para poder executar aplicacións en dispositivos pequenos.

2.3.3.2 *Instalación do JDK e JRE predeterminados en Linux*

Para a instalación eliximos un servidor Ubuntu 18.04.

A opción máis sinxela para instalar Java en Ubuntu é utilizar a versión que forma parte do paquete Ubuntu. Por defecto, Ubuntu 18.04 inclúe OpenJDK, que é unha variante de código aberto de JRE e JDK de Oracle (licenza GPL) e que é máis que suficiente para que funcione o 99% do software que require Java. Para os programadores, o principal problema desta versión de Java é que non permite a creación de aplicacións comerciais aínda sendo un software de código aberto.

Para instalar o JRE executamos o seguinte comando:

```
sudo apt install default-jre
```

Con este comando instálase o Java Runtime Environment (JRE). Isto permite executar case todo o software de Java.

Tras a instalación debe verificarse que se instalou correctamente.

```
java -version
openjdk version "11.0.9" 2020-10-20
OpenJDK Runtime Environment (build 11.0.9+11-Ubuntu-
Oubuntu1.18.04.1) OpenJDK 64-Bit Server VM (build 11.0.9+11-
Ubuntu-Oubuntu1.18.04.1,mixed mode, sharing)
```

Para instalar o JDK, debe executarse o seguinte comando:

```
sudo apt install default-jdk
```

Tras a instalación debe verificarse que se instalou correctamente o compilador de Java.

```
javac -version
javac 11.0.9
```

A instalación do JDK instala tamén o JRE no caso de que non se instalara previamente.

No caso de que queiramos desinstalar o OpenJDK incluíndo as súas dependencias e os ficheiros de configuración usaremos o seguinte comando no terminal:

```
sudo apt-get purge --auto-remove openjdk*
```

2.3.3.3 *Instalación de Oracle JDK 11 en Linux*

Oracle JDK 11 é o primeiro kit de desenvolvemento LTS de Java. JDK 11 terá soporte comercial ata setembro de 2026.

O acordo de concesión de licenzas para Java de Oracle só permite a instalación automática mediante os administradores de paquetes para a última versión de Java.

Na páxina de descargas de Oracle:

<https://www.oracle.com/es/java/technologies/javase-downloads.html>

podemos ver que a última versión de Java é Java SE 15.

Para poder descargar o JDK de Oracle é necesario crear unha conta de Oracle e descargar manualmente o JDK que queiramos utilizar.

A versión de JDK de Oracle que deberemos descargar debe coincidir coa versión da secuencia de comandos do instalador.

Para saber a versión necesaria podemos consultar a páxina: <https://launchpad.net/~linuxuprising/+archive/ubuntu/java/+packages>

Tras consultar esta páxina, vemos que a versión a instalar é a Java 11.0.9. Descargamos da web de Oracle a versión tar.gz para Linux.

Unha vez descargada debemos subila ao servidor Ubuntu.

Para poder subila ao servidor Ubuntu instalado nunha máquina virtual (Virtual Box) necesitamos poder enviar arquivos á máquina virtual. Coa máquina parada entramos na configuración de rede NAT e en opcións avanzadas pulsamos o botón Reenvío de Portos (Port Forwarding) para engadir a seguinte regra:

- Nome: SSH
- Protocolo: TCP
- Host Port: 22
- Guest Port: 22

Para transferir o arquivo ao servidor usamos o comando scp e enviamos o arquivo ao directorio principal do usuario

Unha vez cargado o arquivo no servidor, actualizamos o repositorio:

```
sudo add-apt-repository ppa:linuxuprising/java
```

Actualizamos a lista de paquetes para que o novo software estea dispoñible para a instalación:

```
sudo apt update
```

O instalador buscará o JDK de Oracle que descargamos en /var/cache/oracle-jdk11-installer-local.

Creamos este directorio e copiamos alí o arquivo do JDK de Oracle:

```
sudo mkdir -p /var/cache/oracle-jdk11-installer-local/  
sudo cp jdk-11.0.9_linux-x64_bin.tar.gz /var/cache/oracle-jdk11-installer-local/
```

Por último instálase o paquete:

```
sudo apt install oracle-java11-installer-local
```

O instalador solicita aceptar o acordo de concesión de licenzas de Oracle. Debe aceptarse o acordo para poder continuar coa instalación.

Tras a instalación, debe comprobarse a mesma.

Se queremos desinstalar o Oracle JDK, primeiro debemos comprobar os paquetes que están instalados:

```
sudo dpkg --get-architecture | grep -i jdk
```

Posteriormente borramos Oracle Java 11 completamente:

```
sudo apt-get purge oracle-java11-installer-local  
sudo apt-get autoremove
```

Podemos comprobar que os paquetes se borraron:

```
sudo dpkg --get-architecture | grep -i jdk
```

2.3.3.4 *Instalación de Oracle JDK 11 en Windows*

Para instalar o Oracle JDK 11 en Windows é necesario descargalo da web de Oracle:

<https://www.oracle.com/java/technologies/javase-downloads.html>

Eliximos o instalador para Windows e dámoslle á ligazón para descargar. Para poder descargar é necesario aceptar as condición da licenza e, tras premer o botón de descarga é necesario utilizar as credencias de Oracle para que se inicie a descarga do produto. Procedemos á instalación unha vez temos descargado o executable. Comprobamos que foi correcta na liña de comandos.

2.3.3.5 *Programación en Java*

Podemos crear un arquivo de nome Saudo.java co seguinte contido:

```
public class Saudo {  
    public static void main(String args[]) {  
        System.out.println("Java en Debian Linux"); }  
}
```

O nome da clase sempre ten que ser igual ao nome do arquivo sen ter en conta a extensión.

Para compilar este arquivo en liña de comandos temos que facer:

```
javac Saudo.java
```

Tras compilalo, se non hai erros de compilación, no cartafol onde se creou Saudo.java debe aparecer o arquivo Saudo.class.

Para executar o arquivo Saudo.class facemos:

```
java Saudo
```

Como resultado debe verse na pantalla a mensaxe: Java en Debian Linux.

Se queremos realizar un programa que colla os datos da liña de comandos coma en Produto2.java:

```
public class Produto2 {  
    public static void main(String args[]) {  
        if (args.length>0){  
            int num=Integer.parseInt(args[0]);  
            System.out.println("O dobre de "+num+" e: "+(num*2));  
        }else{  
            System.out.println("Programa executado");  
        }  
    }  
}
```

```
}  
}
```

Para compilar Produto2.java facemos:

```
javac Produto2.java
```

Se non hai ningún erro de compilación, executamos facendo:

```
java Produto2 12
```

O resultado desta execución será: O dobre de 12 e: 24.

Se queremos facer un programa interactivo como Produto2Interactivo.java facemos:

```
import java.util.Scanner;  
public class Produto2Interactivo {  
    public static void main(String args[]) {  
        Scanner in=new Scanner(System.in);  
        System.out.println("Introduce un numero:");  
        int num=in.nextInt();  
        System.out.println("O dobre de "+num+" e: "+(num*2));  
    }  
}
```

Para compilalo facemos:

```
javac Produto2Interactivo.java
```

Para executalo, tras compilalo sen erros facemos:

```
java Produto2Interactivo
```

Se, durante a execución, indicamos que o número é 114 o resultado será: O dobre de 114 e: 228.

Se non queremos indicar de cada vez os datos da execución do programa, sobre todo se estamos depurando e non damos atopado o erro, podemos crear un arquivo de texto co nome que queiramos como pode ser datos.txt para gardar eses datos de proba. Por exemplo, o noso arquivo datos.txt pode ter gardado o número 80. Se o programa pide máis dun número, poden poñerse estes números separados por espazos no ficheiro datos.txt.

Para traballar co arquivo datos.txt indicamos o seguinte na liña de comandos:

```
java Produto2Interactivo < datos.txt
```

O resultado da execución debe ser a mensaxe: O dobre de 80 e: 160.

2.3.3.6 *Instalación de Apache NetBeans en Linux*

Para instalar Apache NetBeans 12.0 en Linux é necesario instalar previamente o Java JDK. OpenJDK é a versión aberta e libre de Java que moitas distribución Linux teñen nos seus repositorios e que na maioría dos desenvolvementos é mais que suficiente para que funcione o 99% do software que require Java.

Para instalar NetBeans, primeiro descargamos o arquivo binario:

```
wget https://www-us.apache.org/netbeans/netbeans/12.0/netbeans-12.0-bin.zip
```

Unha vez descargado, descomprimos o arquivo descargado:

```
unzip netbeans-12.0-bin.zip
```

Tras descomprimir movemos o cartafol de netbeans ao cartafol /opt que é o cartafol que se usa en Linux para instalar as aplicacións de terceiros:

```
sudo mv netbeans/ /opt/
```

O executable de Netbeans está en /opt/netbeans/bin/netbeans. É necesario engadir o seu directorio pai ao \$PATH para que podamos lanzar o programa sen especificar a ruta absoluta ao executable.

Cando tentamos executar un comando, o sistema busca o ficheiro executable nunha lista de directorios que está almacenada na variable de contorno chamada \$PATH.

Para actualizar a variable \$PATH do usuario de linux o que facemos é modificar o arquivo .bashrc deste usuario. O arquivo .bashrc é un script que se executa cada vez que un usuario inicia sesión nun terminal de Linux.

Abrimos o arquivo ~/.bashrc:

```
nano ~/.bashrc
```

Engadimos a seguinte liña ao final do ficheiro:

```
export PATH="$PATH:/opt/netbeans/bin/"
```

Para poder lanzar NetBeans sen reiniciar a shell usamos o comando:

```
source ~/.bashrc
```

Para crear un arquivo .desktop que permita que a aplicación NetBeans apareza no menú de aplicacións creamos o ficheiro netbeans.desktop en

```
/usr/share/applications/:
```

sudo nano /usr/share/applications/netbeans.desktop No ficheiro escribimos os seguintes datos:

```
[Desktop Entry]
Name=Netbeans IDE
Comment=Netbeans IDE
Type=Application
Encoding=UTF-8
Exec=/opt/netbeans/bin/netbeans
```

```
Icon=/opt/netbeans/nb/netbeans.png  
Categories=GNOME;Application;Development;  
Terminal=false  
StartupNotify=true
```

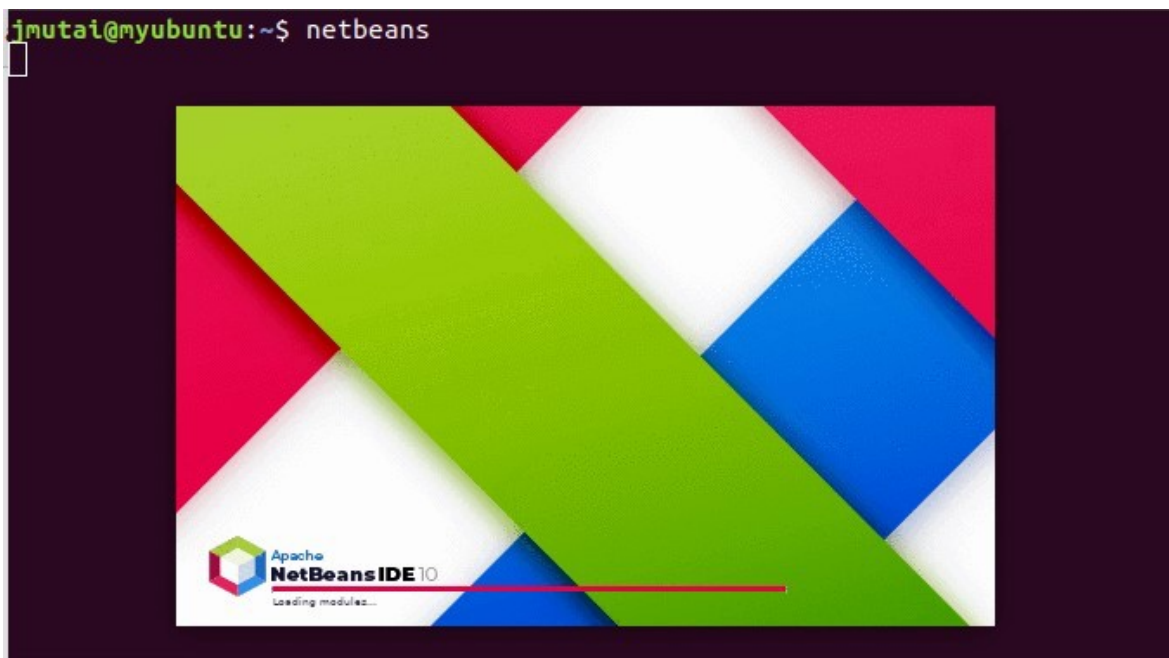
Tras isto, xa temos instalado o IDE NetBeans e podemos lanzalo desde o menú de aplicación



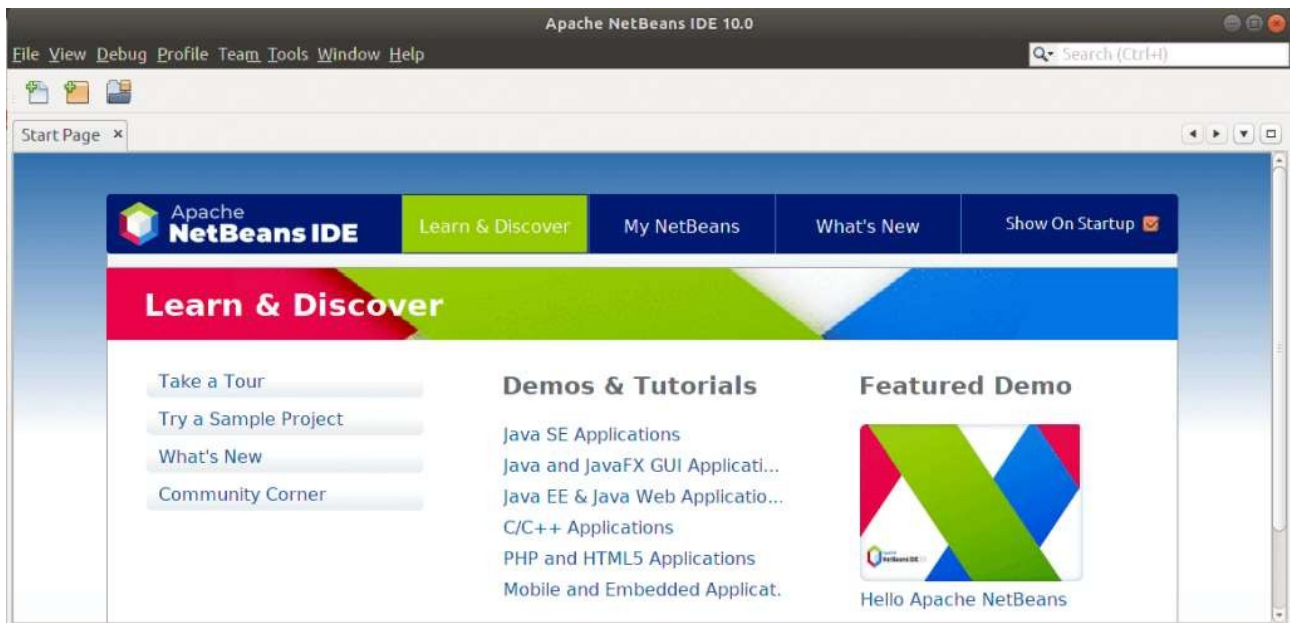
Tamén podemos lanzalo desde o terminal usando o comando netbeans:

```
netbeans
```

Debe mostrarse unha pantalla como a seguinte



Cando a aplicación se teña iniciado, debería verse a páxina de benvinda por defecto de NetBeans.



A partir deste momento, xa podemos crear os nosos proxectos Java con NetBeans. Temos que ter en conta que, os IDE, non traballan con arquivos e cartafol, usan o concepto de proxecto en vez do de cartafol. Un proxecto lóxicamente é un cartafol no disco duro, pero nun IDE créanse arquivos adicionais ao código para optimizar a experiencia dos usuarios. Nestes arquivos pode estar configuración de execución, despregamento, tipo de proxecto, etc.

Tras crear un proxecto, podemos querer utilizar argumentos da liña de comandos. Para facelo imos a File eliximos Project Properties e en categories eliximos run. Aparece unha caixa de texto para indicar os argumentos.

Se queremos coller a entrada estándar dun ficheiro, é dicir, facer o equivalente a:

```
java Produto2Interactivo < datos.txt
```

podemos facelo engadindo un obxectivo propio de execución ao arquivo build.xml do proxecto. Por exemplo:

```
<target name="run" depends="jar">
  <exec dir="${work.dir}" executable="java" input="${work.dir}/datos.txt">
    <arg value="-jar"/>
    <arg file="${dist.jar}"/>
  </exec>
</target>
```

Hai que ter en conta que os comandos como Run e Debug só usan o build.xml personalizado se nas propiedades do proxecto de Compiling en Build non está seleccionado Compile on Save.

2.3.3.7 Instalación de Apache NetBeans en Windows

Na URL <https://netbeans.apache.org/download/index.html> pode descargarse Apache Netbeans.

Eliximos a última versión LTS (Apache Netbeans 12.0).

Descargamos o instalador para Windows e instalamos.

2.3.3.8 Javadoc

Javadoc é unha ferramenta do SDK que permite documentar, dunha maneira rápida e sinxela, as clases e métodos que se crean, sendo de grande utilidade para a comprensión do desenvolvemento.

A linguaxe Java soporta tres tipos de comentarios:

`/* texto */`. O compilador ignora todo o que estea comprendido entre `/*` e `*/`.

`// texto`. O compilador ignora todo o que estea a partir de `//` ata a fin da liña.

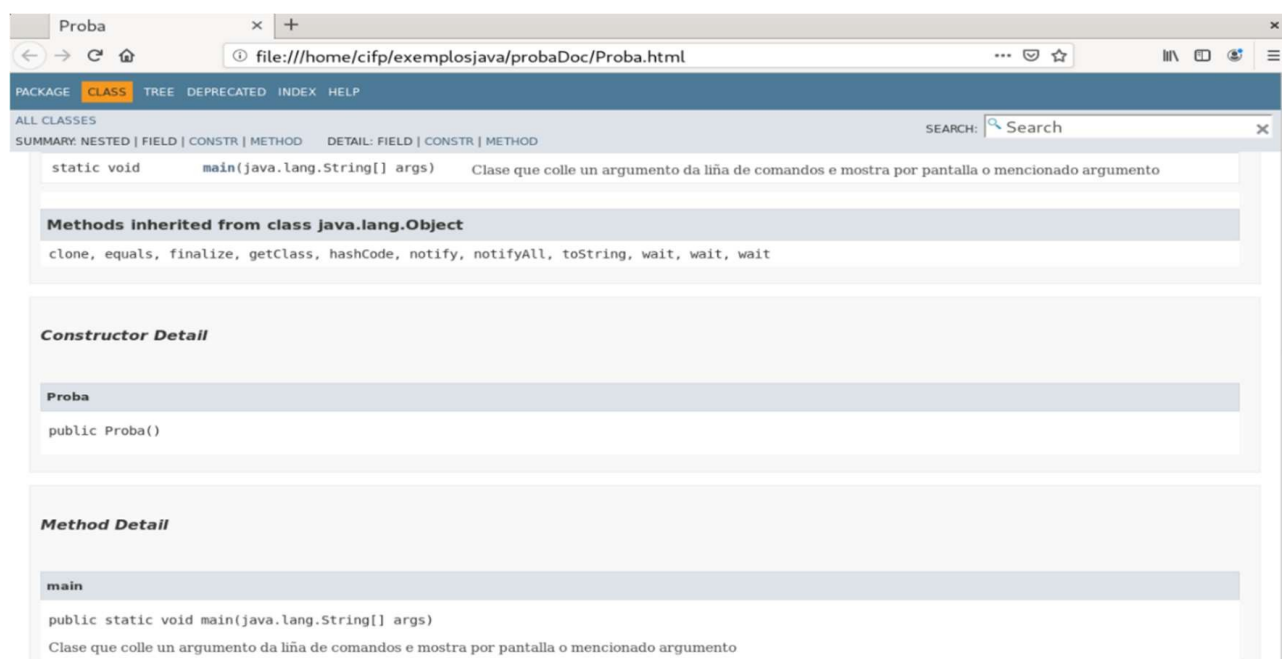
`/** documentacion */`. Este é un comentario de documentación. A ferramenta javadoc do JDK usa os comentarios de documentación para preparar a documentación que se xera automaticamente.

Para usar en liña de comandos javadoc hai moitos parámetros posibles. Se queremos crear un cartafol de nome probaDoc e almacenar nel a documentación sobre o arquivo de código proba.java podemos facer:

```
javadoc -d ./probaDoc/ Proba.java
```

Para visualizar a documentación xerada podemos facer:

```
firefox ./probaDoc/index.html
```



2.3.4 Instalación de plugins en NetBeans

Unha vez baixada e instalada a versión básica do IDE, podemos baixar outros paquetes ou plugins conforme os vaíamos necesitando.

Os plugins para NetBeans poden instalarse desde dúas fontes:

1. O catálogo de plugins de NetBeans que podemos consultar desde o IDE (instalación online).

2. Archivos .nbm (NetBeans Module) que podemos atopar nalgúnhas páxinas de internet, principalmente desde o portal de plugins de NetBeans (<http://plugins.netbeans.org/PluginPortal/> - instalación offline).

2.3.4.1.1 Instalación desde o catálogo de plugins

Esta forma de instalación é a máis simple e directa. Permítenos desde a mesma interface de NetBeans, revisar un catálogo de plugins e descargalos. Tamén é a máis segura xa que no caso de que o plugin que estamos tratando de instalar necesite outros plugins, estes plugins tamén son descargados e instalados. Ademais, podemos estar seguros de teñen pasado por un proceso de verificación e non debe haber ningún problema ao instalalos.

Para acceder a este catálogo, dirixímonos ao menú Tools -> Plugins. Con isto, aparecerá a ventá de Plugins. Esta ventá ten 5 pestanas:

- As actualizacións atopadas dos plugins que xa temos instalados.
- Os plugins dispoñibles e que non temos instalado.
- Os plugins que descargamos (archivos .nbm) que queremos instalar.
- Os plugins que xa temos instalado.

Podemos ver de instalar desta forma o plugin NB-ChangeCase que cambia o texto seleccionado de todo maiúsculas a todo minúsculas entre outros cambios posibles.

Para descargar e instalar os plugins seleccionados, facemos click no botón Install co que se nos amosará a lista de plugins que serán instalados. Facemos click en Next e mostrará un acordo de licenza. Se aceptamos o acordo, comezará a descarga e, posteriormente, realizarase a instalación. Posteriormente, comprobaremos que xa podemos usar o novo plugin.

2.3.4.1.2 Instalación desde arquivos .nbm

Esta forma de instalación ten a vantaxe de que non é necesario que esteamos conectados a internet para realizala xa que toda a información do plugin está no arquivo .nbm.

Non obstante, se un plugin necesita doutros para funcionar, será necesario que nos conectemos a internet, ou ben, que contemos co arquivo .nbm dese outro plugin.

Estes arquivos poden descargarse desde diversos sitios de internet. Normalmente faremolo desde o portal de plugins de NetBeans. Neste portal podemos ver os últimos plugins subidos, os plugins máis populares, os mellor clasificados, por categorías ou podemos buscalos por algún criterio.

Para facer probas, podemos instalar easyUML. Para isto, imos á páxina de plugins de NetBeans e descargamos este plugin. Descomprimos o arquivo descargado usando o comando unzip e a continuación o nome do arquivo descargado.

Posteriormente, desde o NetBeans, accedemos ao cartafol de descargas e seleccionamos os arquivos .nbm obtidos tras a descompresión (no caso de easyUML son 10 arquivos). Unha vez seleccionados os arquivos xa podemos realizar a instalación.

Tras reiniciar o IDE, xa podemos facer proxectos UML e probar o plugin.

2.4 Actividades



2.1

Solución