

# CONTORNOS DE DESENVOLVEMENTO

**Tema 2:  
INSTALACIÓN E USO  
DE CONTORNOS DE  
DESENVOLVEMENTO**

# Índice

---

<b>1.</b>	<b>Instalación de contornos de desenvolvemento.....</b>	<b>3</b>
1.1	Características dos contornos de desenvolvemento .....	3
1.2	Funcións, ferramentas e asistentes dun contorno de desenvolvemento .....	3
1.3	Contornos de desenvolvemento más empregados .....	4
1.4	Contorno de desenvolvemento libre NetBeans.....	8
1.5	Contorno de desenvolvemento libre Eclipse .....	14
1.6	Contorno de desenvolvemento Visual Studio Code .....	19
1.7	Contorno de desenvolvemento IntelliJ Idea .....	22
<b>2.</b>	<b>Actualizar, personalizar, automatizar e xestionar módulos do contorno .....</b>	<b>24</b>
2.1	Actualizar NetBeans .....	24
2.2	Personalizar NetBeans .....	26
2.3	Automatizar en NetBeans .....	29
2.4	Xestionar complementos de NetBeans.....	32
2.5	Extensións de Visual Studio Code .....	33
2.6	Plugins de IntelliJ Idea.....	33
<b>3.</b>	<b>Uso básico do contorno de desenvolvemento en operacións de edición e xeración de código.....</b>	<b>35</b>
3.1	Introducción.....	35
3.2	Proxecto C/C++ en Netbeans .....	35
3.3	Proxecto PHP en NetBeans.....	41
3.4	Proxecto Spring en Visual Studio Code .....	47
3.5	Proxecto Java en IntelliJ Idea .....	49
<b>4.</b>	<b>Proxecto Java en Netbeans .....</b>	<b>52</b>
<b>5.</b>	<b>Anexo:Práctica PHP en Netbeans .....</b>	<b>58</b>
5.1	Descripción de la aplicación .....	58
5.2	Instalación .....	58
5.3	Modificaciones en PHP.....	61
5.4	Modificaciones en JavaScript .....	63
5.5	Modificaciones en HTML y CSS .....	63



**Material complementario e vídeos:**

<https://wirtzIDE.blogspot.com/>



Fernando Rodríguez Diéguez  
rdf@fernandowirtz.com  
Versión 2024-11-10

Licenza Creative Commons BY-NC-SA (recoñecemento - non comercial - compartir igual) a partir de material propio e dos documentos orixinais:

© Xunta de Galicia. Consellería de Cultura, Educación e Ordenación Universitaria.  
Autores: María del Carmen Fernández Lameiro

# 1. Instalación de contornos de desenvolvemento

---

## 1.1 Características dos contornos de desenvolvemento

Un contorno de desenvolvemento ou IDE (*Integrated Development Environment*) ou EDI (*Entorno de Desenvolvimento 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.

A característica máis importante é que consta de diferentes ferramentas integradas, é dicir, os resultados dunha ferramenta poden ser utilizados para outra ferramenta e quedarán relacionados. Por exemplo, un programa Java e o seu correspondente programa de probas unitarias en JUnit estarán relacionados, de tal forma que o contorno avisará se un cambio no programa Java afecta ao programa de proba; isto non ocorrería se os programas fosen codificados nun editor de texto.

Outra característica é que se poden facer moitas operacións sen saír da mesma aplicación e seguindo sempre a mesma forma de traballar (ventás, menús, combinación de teclas, manexo da axuda,...).

Para crear un programa só é imprescindible un editor de textos sinxelo e para executalo só é imprescindible o tradutor correspondente, pero é moito mellor ter unha aplicación dende a que se poidan facer operacións relacionadas como por exemplo: deseñar os diagramas UML que servirán de base para a codificación, xerar código fonte mediante editores de texto que suxiran e completen automaticamente instrucións, analizar e depurar código para detectar errores, executar código e probalo, e todo iso dentro da mesma aplicación.

Nalgúns contornos só se pode codificar nunha linguaxe pero noutros pódese codificar en varias permitindo entón que un proxecto poida estar formado por programas en diferentes linguaxes sen ter que utilizar varias aplicacións, unha para cada linguaxe, cada unha delas coa súa propia forma de traballar.

## 1.2 Funcións, ferramentas e asistentes dun contorno de desenvolvemento

As funcións más destacadas dun contorno de desenvolvemento son:

- **Edición do código fonte.** O editor de texto realiza operacións xerais de edición como buscar, substituír, copiar, cortar, pegar, desfacer, refacer, salvar, imprimir, importar, exportar e ademais ten características deseñadas para simplificar e acelerar a escritura de código fonte como por exemplo resaltar e colorear as instrucións segundo o tipo, marcar errores sintácticos, suxerir instrución, realizar o completado automático de instrucións que se está escribindo, ou tabular automaticamente as instrucións.
- **Análise de código.** O analizador de código pode ser estático e dinámico. O analizador estático fai a **análise léxica, sintáctica e semántica do código** en tempo de codificación e detecta os fallos dese tipo. O analizador dinámico valora o comportamento do código ao mesmo tempo que se está executando e obtén información como por exemplo: tempo invertido na execución, ou ocupación de memoria que permitirá optimizar o funcionamento do código.

- **Depuración de código.** O depurador é un programa usado para eliminar erros noutros programas (o programa "obxectivo"). O código a examinar execútase nun simulador de instrucións (ISS) que permite deterse cando se verifiquen certas circunstancias establecidas polo programador. Nese momento o programador poderá conseguir máis información para localizar o erro e arranxalo. Este modo de execución do programa é más lento que a execución normal.
- **Xeración de código executable e execución** mediante as ferramentas adecuadas. Estas ferramentas dependerán do tipo de linguaxe: compilada, interpretada, de maquina virtual ou administración asistida pero en calquera caso, o contorno debe estar configurado para localizar o compilador, as librerías e os intérpretes ou máquinas virtuais que necesite. Algúns contornos teñen a posibilidade de ter compiladores cruzados, é dicir, compiladores para unha plataforma diferente da plataforma sobre a que funciona o contorno de desenvolvemento.
- **Probas.** O contorno de desenvolvemento facilita a creación de probas unitarias ou de integración, permite a execución das mesmas e emite o informe sobre o resultado.
- **Control de versións.** A ferramenta de control de versións facilita a administración das distintas versións ou especializacións dun produto. Unha versión é o estado no que se encontra un producto nun momento dado do seu desenvolvemento. O control de versións é imprescindible cando se leva a cabo un proxecto informático coa colaboración de varias persoas.
- **Construcción de interfaces gráficas de usuario.** As GUI (*Graphical User Interface*) son interfaces gráficas de usuario e proporcionan un entorno visual sinxelo para permitir a comunicación do usuario co sistema operativo mediante imaxes e obxectos gráficos que representan a información e as accións dispoñibles na interface. O contorno de desenvolvemento pode ter un construtor GUI (*Graphical User Interface*) que simplifica o proceso de deseño e construcción de GUI.
- **Xeración de documentación.** O xerador de documentación permite crear de forma automática un arquivo coa documentación do código baseándose no código fonte.
- **Refactorización de código.** O contorno de desenvolvemento ten ferramentas que facilitan a operación de refactorizar código, é dicir, cambiar o código fonte alternado a estrutura interna sen cambiar o comportamento externo. O programador indica o tipo de refactorización que se quere realizar e a ferramenta de refactorización levara a cabo en detalle e de forma segura sen a intervención do programador.

## 1.3 Contornos de desenvolvemento más empregados

Existen moitos contornos de desenvolvemento: algúns con licenza propietaria, outros con licenza libre; algúns só funcionan sobre unha plataforma, outros funcionan en varias plataformas; algúns só permiten unha linguaxe, outros permiten varias linguaxes e o número de ferramentas dispoñibles varían moito duns a outros.

Todo isto fai moi difícil ou case imposible indicar os contornos de desenvolvemento más empregados xa que ningún contorno é óptimo para todas as necesidades e por outro lado o mellor contorno para un desenvolvedor é aquel que coñece a fondo e do que polo tanto pode sacar más partido.

En [https://en.wikipedia.org/wiki/Comparison\\_of\\_integrated\\_development\\_environments](https://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments) existe unha comparativa de contornos de desenvolvemento por linguaxe sobre aspectos como licencias, plataformas, depurador, analizador, construtor GUI, refactorizador, ou completar código. Aínda que ás veces a comparativa é confusa e non actualizada pode servir de referencia.

Se revisamos esta comparativa para as linguaxes de programación más populares (C, C++, Java, C#, PHP, Python e JavaScript), a lista de contornos de desenvolvemento nos que se podería codificar no maior número desas linguaxes con case tódalas ferramentas quedaría reducido a: Visual Studio Code, a familia de IDEs de JetBrains como por exemplo IntelliJ Idea, Apache Netbeans e Eclipse, áinda que estes dous últimos moi orientados á programación en Java.

Visual Studio ten licenza propietaria de Microsoft, permite codificar en moitas linguaxes como C/ C++, C#, Java, JavaScript e Python. Está dispoñible para Windows, Linux e MacOS X, e ten unha versión gratuítia baixo o nome Visual Studio Code.

NetBeans e Eclipse teñen licencias libres, poden utilizarse nas plataformas Windows, Mac OS X e Linux, e que permiten codificar en C/ C++, Java, PHP, JavaScript e Python. Ambos teñen moitas características comúns e permiten traballar de forma moi parecida.

IntelliJ Idea pertence á familia de IDEs de JetBrains que partindo dun núcleo común, ofrece IDE especializados para distintas linguaxes, sendo IntelliJ o máis empregado na actualidade para desenvolvemento en Java. É software de pago áinda que dispón dunha versión gratuítia.

NetBeans



Comezou como un proxecto estudiantil na República Checa en 1996 baixo a tutoría da Facultade de Matemáticas e Física da Universidade Carolina en Praga e orixinalmente chamouse Xelfi. En xuño de 2000, Sun Microsystems patrocina NetBeans e declárao de código aberto. En 2010 Oracle adquire Sun; actualmente (2018) continúa patrocinando NetBeans e apostando porque sexa o IDE oficial para a plataforma Java.

O proxecto NetBeans está formado por dous produtos: o IDE e a plataforma, ambas gratuítas e de código aberto para que calquera poida utilizarlo como mellor lle pareza dentro dos termos de uso que se explican na páxina oficial: <https://netbeans.apache.org/>.

No proxecto NetBeans participan neste momento unha comunidade de máis de 800.000 desenvolvedores e xente de todo o mundo pode facer preguntas, dar consellos e en última instancia contribuír no éxito dos produtos do proxecto.

A maioría do código de NetBeans IDE está dispoñible baixo unha licenza dual CDDL (*Common Development and Distribution License*) v1.0 e GNU GPL (*General Public License*) v2 que o fai un produto de código aberto, gratuíto e sen restricións de uso.

NetBeans IDE está escrito en Java pero pode servir para programar noutras linguaxes de programación como: Java, PHP, C/C++, XML, HTML, Groovy, JavaScript e JSP (tecnoloxía Java para páxinas web) e mediante complementos pódense utilizar máis linguaxes de programación e frameworks como Spring. Pode ser instalado sobre calquera sistema operativo que soporte Java: Windows, Linux, MacOS X, ou Solaris.

Aínda que segundo a páxina oficial de NetBeans, lévanse feitas máis de 18 millóns de descargas deste contorno de desenvolvemento, nos últimos anos o seu uso descendeu notablemente xa que os seus competidores, coas extensión/plugins, permiten mellorar a nosa produtividade notablemente (exemplo Github Copilot) e, neste aspecto, Netbeans estase a quedar atrás.

## Eclipse

Eclipse foi desenvolvido orixinalmente por IBM como o sucesor da súa familia de ferramentas para VisualAge. No ano 2003 foi creada a Fundación Eclipse que é unha organización independente e sen ánimo de lucro que fomenta unha comunidade de código aberto e un conxunto de produtos complementarios, capacidades e servizos. A páxina oficial é <https://www.eclipse.org/>.



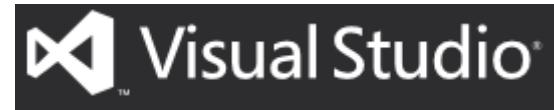
Eclipse permite codificar en Java e tamén noutras linguaxes como C, C++, Python, Perl, PHP e Ruby. Igual que NetBeans, mediante complementos pode utilizar máis linguaxes.

Eclipse está dispoñible baixo unha licenza *Eclipse Public License v 1.0*<sup>1</sup>. Esta licenza permite utilizar, modificar, copiar e distribuír as versións modificadas e nalgúns casos os cambios tamén terán que ser libres. Este tipo de licenza está aprobada pola *Free Software Foundation* e a *Open Source Initiative* como licenza de software libre.

Como xa comentamos sobre Netbeans, hai outros IDE como Visual Studio ou IntelliJ Idea, que superan a Eclipse ofrecendo unha interface máis moderna e melloras na produtividade.

## Visual Studio

Visual Studio é un contorno de desenvolvemento con licenza propietaria de Microsoft para a plataforma .NET. Permite crear aplicacións en Visual Basic , Visual C# , Visual C++ y JScript e noutras linguaxes más. A páxina oficial é: <https://visualstudio.microsoft.com/es/downloads/>



Visual Studio permite compilar o código fonte nun código intermedio denominado CIL (*Common Intermediate Language*) similar ao BYTECODE de Java. Para executar este código necesitase o framework .NET que dispón do contorno de execución CLR (*Common Language Runtime* ) que mediante un compilador JIT (*Just-In-Time*) realiza o paso a código máquina real que se executa na plataforma do cliente. Desta forma conséguese a independencia da plataforma de hardware.

Especialmente útiles para desenvolvedores de software son as subscricións MSDN anuais de Microsoft que permiten dispoñer de licencias de moitos produtos Microsoft e acceder a certos servizos a un prezo competitivo. A páxina oficial é: <http://msdn.microsoft.com/es-es/default.aspx/>. Estas son as súas versións:

- Enterprise: 5999\$. Ten todo o que ten a edición profesional e engade novas funcionalidades como ferramentas para medir código, analizalo ou facer probas unitarias.
- Professional: Licenza independente 499\$ e de subscrpción estándar 1199\$. Ferramentas e servizos de desenvolvemento profesional para desenvolvedores individuais e equipos pequenos.
- Code: Gratuita e de código aberto, que se executa en calquera parte
- Community: Para desenvolvedores que crean aplicacións non empresariais

Microsoft ofrece algunas aplicacións gratis a estudiantes sempre que estes acrediten a situación de estudiantes da maneira que esixe Microsoft e tamén ofrece a opción de descargar algunas aplicacións software con 90 días de proba.

---

<sup>1</sup> Ver máis información en <https://www.eclipse.org/org/documents/epl-v10.php>

A versión gratuita Visual Studio Code estase a converter nun dos IDE máis empregados ultimamente, xa que no elevado número de extensións permite mellorar a produtividade e adaptalo ao estilo e necesidades de cada programador.

### IntelliJ Idea (JetBrains)

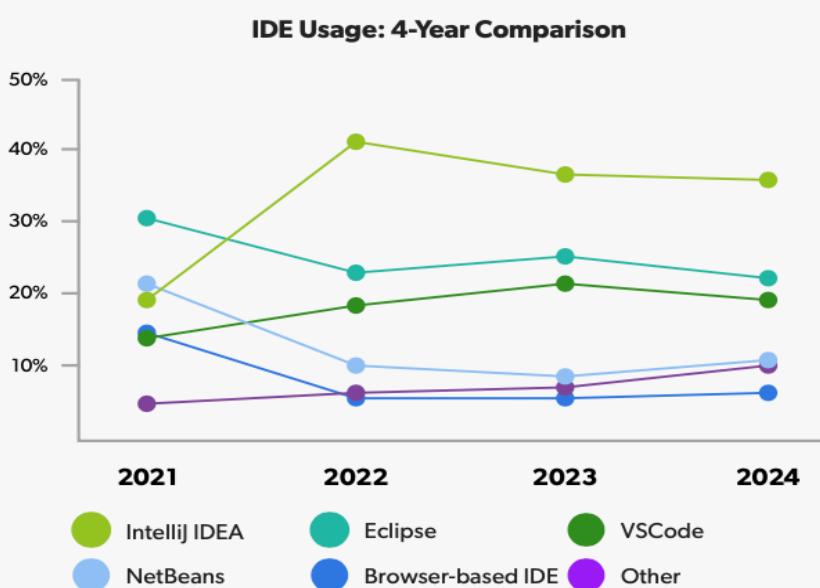
Jetbrains non é un IDE, é unha empresa que dispón dun dos IDEs máis empregados: IntelliJ IDEA, e outros especializados en distintas linguaxes: PhpStorm, PyCharm, GoLand, RubyMine, WebStorm (para JavaScript en desenvolvemento web), etc.



En realidade, todos estes IDE son o mesmo, pero con lixeiras variacións para adaptalo a cada linguaxe. Dispón de dúas licencias: "Ultimate", con toda a funcionalidade e de pago e "Community", versión gratuita pero non permite o uso comercial. Jetbrains ofrece tamén a súa versión *Ultimate* a profesores e estudantes sen custo.

#### How Many People Use IntelliJ IDEA?

The [2024 Java Developer Productivity Report](#) found that 41% of respondents are using IntelliJ IDEA as their primary IDE. That number has increased over the last four years as usage rates of other IDEs (like Eclipse) have slipped.



## IDE para desenvolvemento móbil

Para rematar este apartado, cabe destacar que hai tamén IDEs orientados a linguaxes para dispositivos multimedia. A súa principal diferenza é que inclúen un simulador de dispositivo móvil.

Para desenvolver aplicáños IOS (en linguaxe Swift ou Objective-C) o máis popular é XCode é so se pode executar en equipos Mac. Por outra banda, para desenvolvemento para aplicáños Android (en linguaxes Kotlin ou Java) é Android Studio.



**Tarefa 2.1. Buscar en internet contornos de desenvolvemento más populares nos que se poida codificar en distintas linguaxes/frameworks. Caracteriza ditos contornos de desenvolvemento: versión actual, plataformas para as que está dispoñible, tipo de licencia e prezos, linguaxes que soporta, etc.**

## 1.4 Contorno de desenvolvemento libre NetBeans

### Instalación

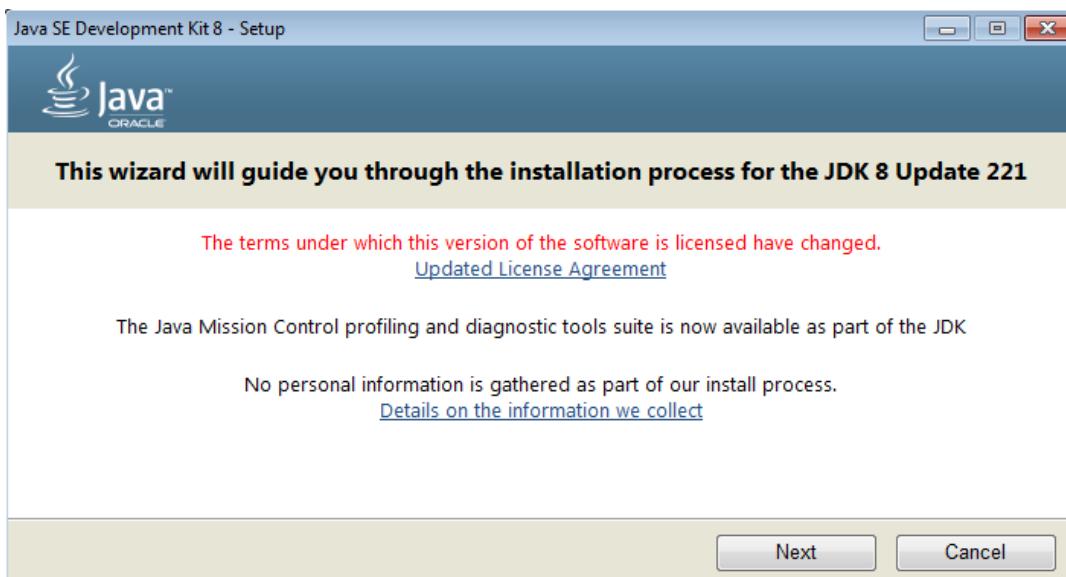
Instalarase o contorno de desenvolvemento libre NetBeans na súa versión **NetBeans 21**.

O software necesario pode descargarse da páxina oficial e as instrucións detalladas de instalación están descritas en <https://netbeans.apache.org/>.

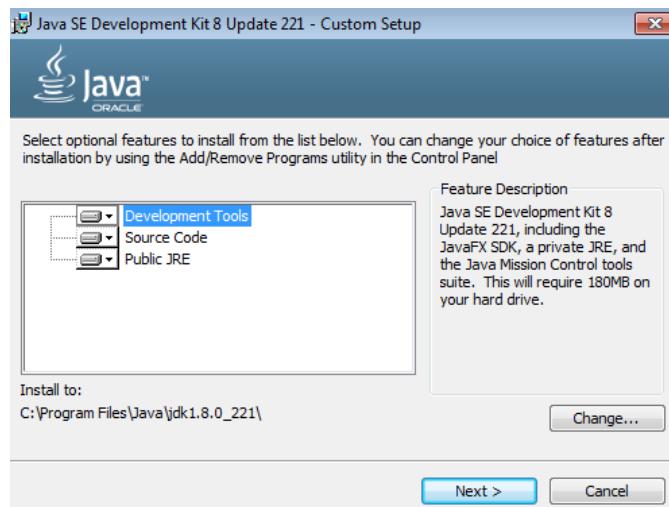
NetBeans está escrito en Java e por tanto antes de instalar NetBeans hai que ter instalado ou instalar JDK (*Java Development Kit*). Empregaremos a versión **jdk 21 de Oracle**.

JDK é un conxunto de ferramentas para desenvolver aplicáños Java que inclúe entre outros JRE e o compilador Java. JRE (*Java Runtime Environment*) é o contorno necesario para executar as aplicáños Java e inclúe entre outros a JVM (*Java Virtual Machine*) encargada de interpretar o bytecode Java. Hai a versión oficial de Oracle, con restricións de uso, e outra versión libre: OpenJDK.

A instalación de JDK consiste na descompreión do zip descargado ou ben a execución do arquivo, no caso de que descargásemos o instalador executable.



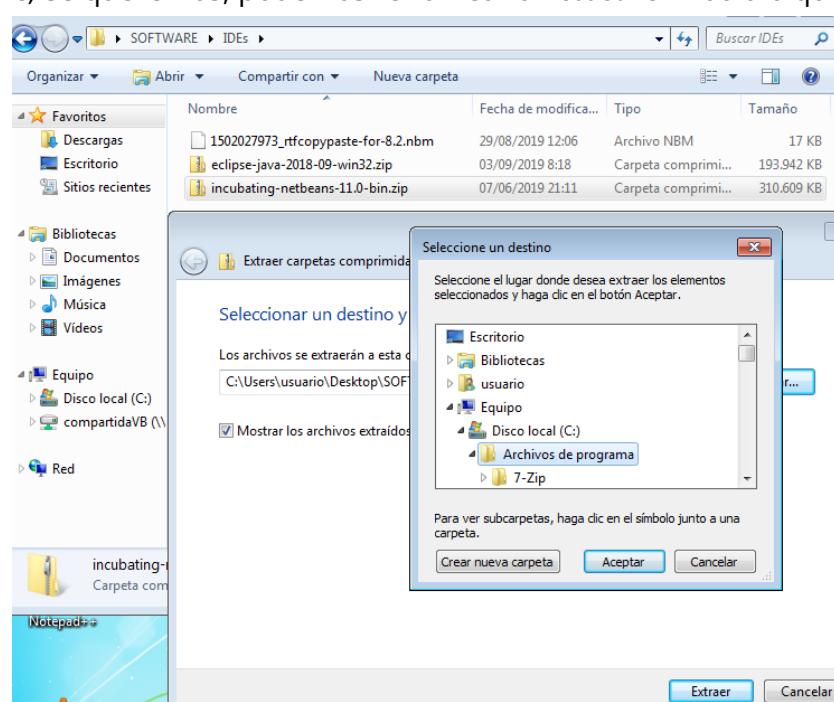
Premeríamos *Next*.



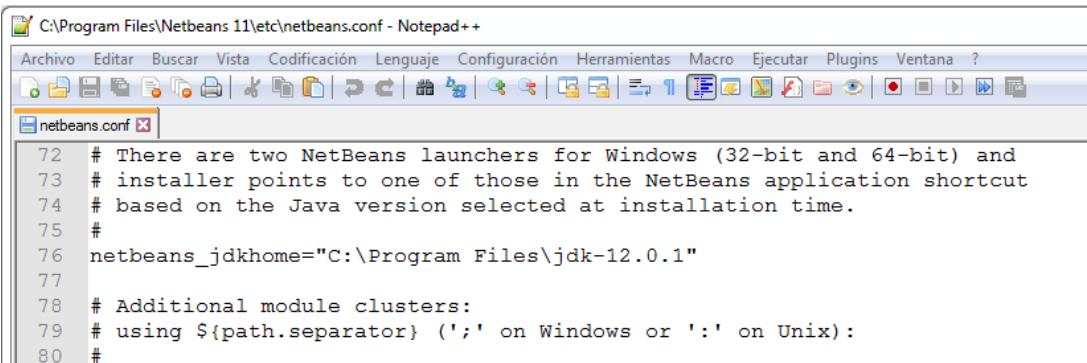
E outra vez *Next*. Deixamos a carpeta destino por defecto, e *Seguinte*, rematando a instalación na seguinte pantalla, na que premeremos *Close*.



A instalación de Netbeans consiste na descompresión do arquivo .zip descargado. En Windows podemos levar o cartafol descomprimido a c:\Archivos de Programa\ . Como o cartafol se chamará *netbeans*, se queremos, podemos renomear a *netbeans22* ou a o que queiramos.



Antes da primeira execución hai que informar a Netbeans onde está instalado Java. Para isto debemos editar o arquivo *netbeans.conf* situado no cartafol *etc*, na ruta onde estea instalado o jdk de Java. Outra opción sería configurar a variable PATH dentro das variables de contorno do noso sistema Windows para incorporar a ruta dos arquivos de java (jdk/bin).

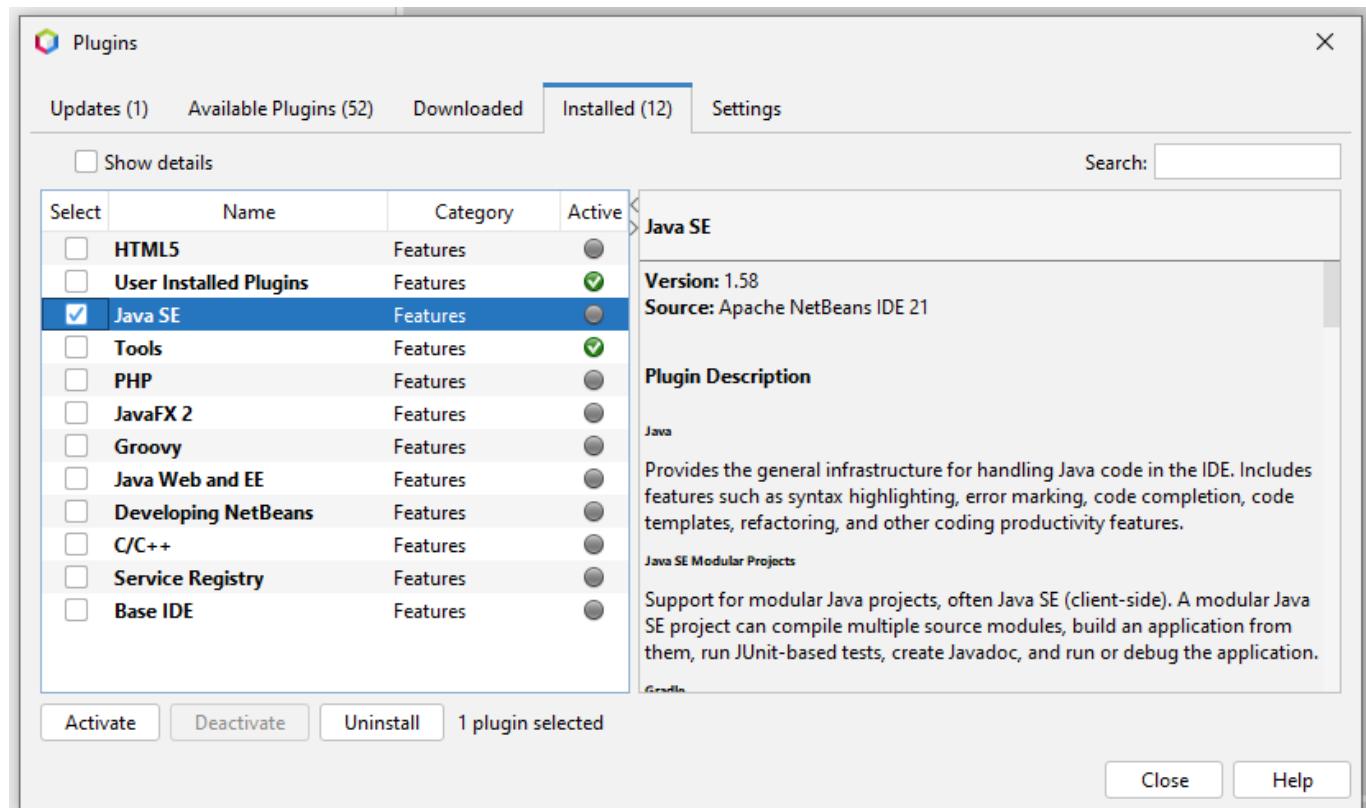


```

C:\Program Files\Netbeans 11\etc\netbeans.conf - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?
netbeans.conf x
72 # There are two NetBeans launchers for Windows (32-bit and 64-bit) and
73 # installer points to one of those in the NetBeans application shortcut
74 # based on the Java version selected at installation time.
75 #
76 netbeans_jdkhome="C:\Program Files\jdk-12.0.1"
77 #
78 # Additional module clusters:
79 # using ${path.separator} (';' on Windows or ':' on Unix):
80 #

```

Para poder desenvolver aplicacións Java, temos que activar o plugin con esa funcionalidade (Java SE) desde o menú superior *Tools > Plugins* e reiniciar Netbeans.

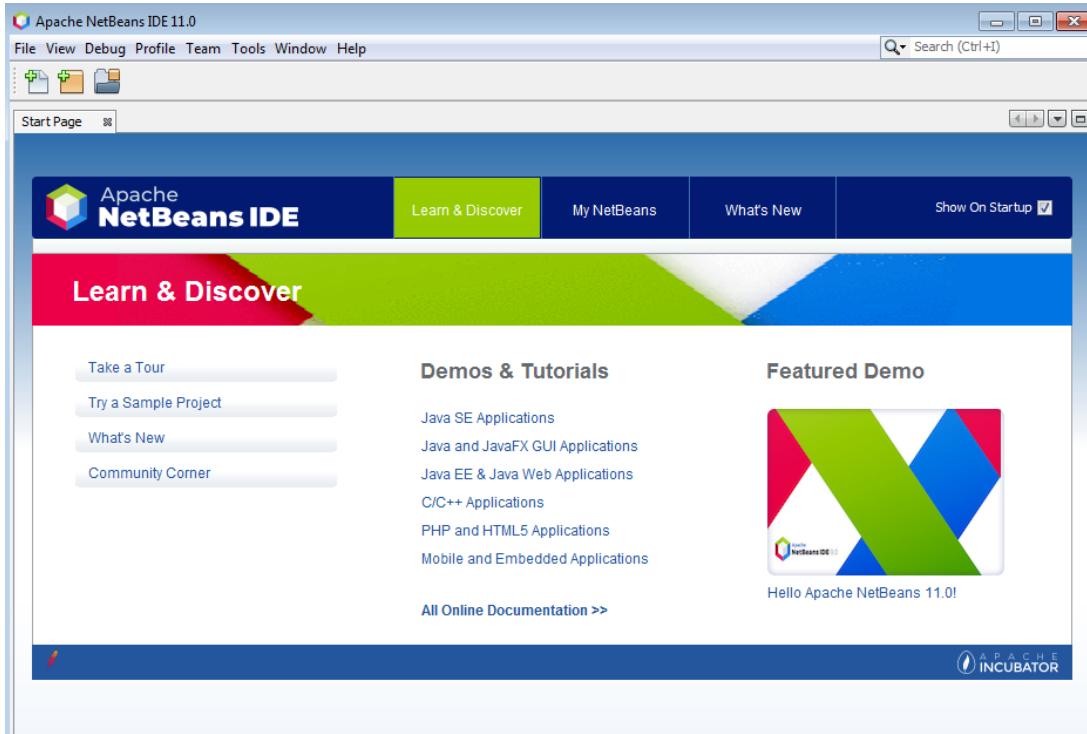


Como derradeiro paso, podemos crear no escritorio unha icona do executable situado en *c:\Archivos de Programa\Netbeans\bin*.

## Início

A forma más sinxela de iniciar NetBeans é facer dobre clic sobre a icona do escritorio. O primeiro inicio pode levar máis tempo.

O aspecto de NetBeans ten é moi austero a primeira vez que se executa despois da instalación pero a medida que se vaian abrindo e creando novos proxectos, vanse incorporando novos elementos e este aspecto inicial vai cambiando.



A páxina de inicio ten enlaces a diferentes informacións relacionadas con NetBeans e ten accesos rápidos aos últimos proxectos manipulados e á xestión de complementos. Pode obterse máis información sobre NetBeans utilizando a opción *Ayuda* do menú principal ou premendo a tecla F1. Esta ventá inicial podemos eliminarla para seguintes execucións, desmarcando *Show on Startup*.

En xeral, consta de: barra de menús, caixa de buscar, iconas de ferramentas, e pestana coa páxina de inicio. O contorno péchase dende o menú principal seleccionando *Archivo>Salir*.

## Primeiros pasos

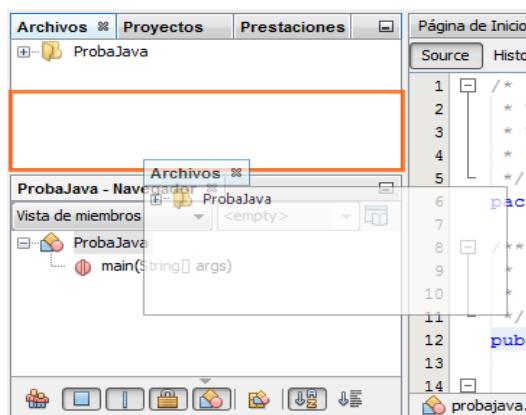
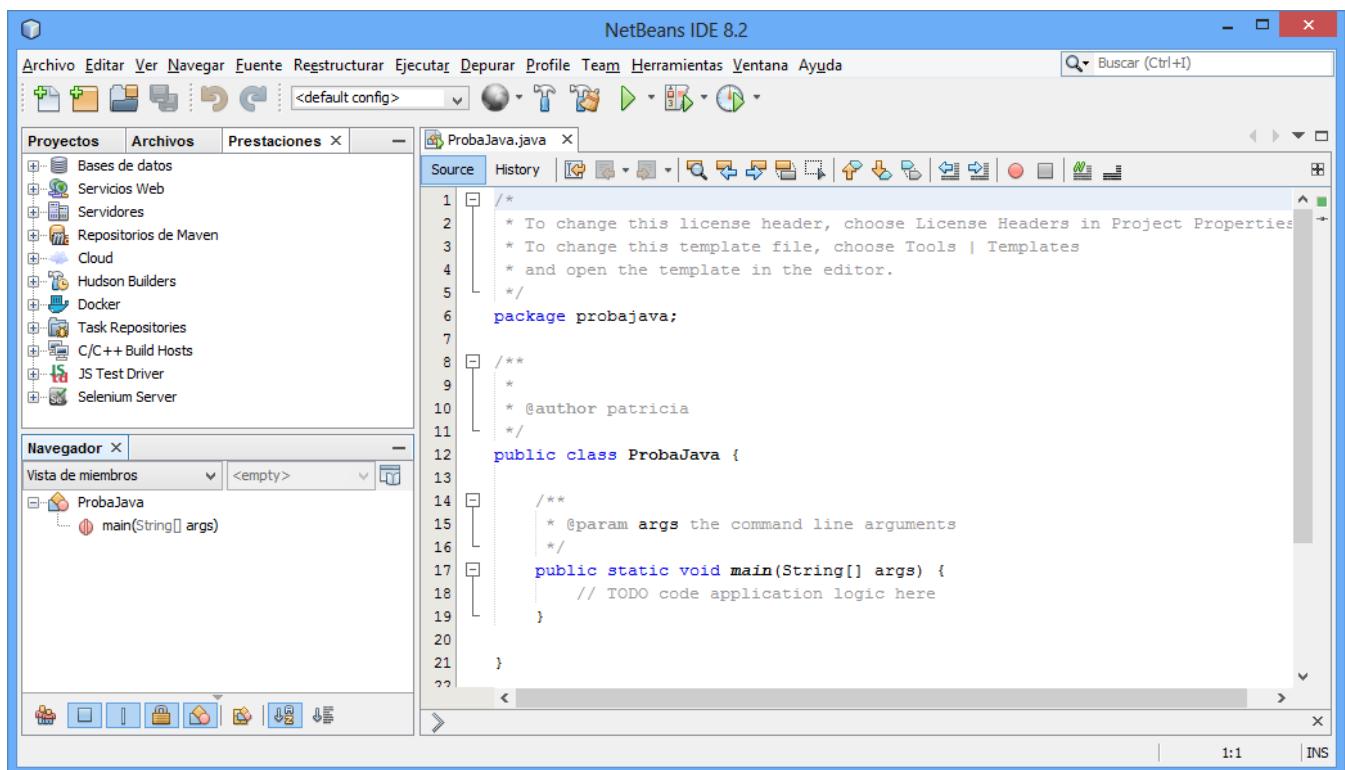
NetBeans permite editar un arquivo independente pero onde se saca todo o proveito ao IDE é utilizando un proxecto, é dicir, unha carpeta cunha estrutura controlada por NetBeans que pode ter arquivos de diferentes tipos relacionados que darán lugar a unha aplicación.

Para crear un proxecto novo hai que elixir *Archivo>Proyecto Nuevo* no menú principal, na ventá *Proyecto Nuevo* que hai que indicar o tipo de proxecto ou o exemplo desexado e na seguinte ventá hai que indicar polo menos o nome do proxecto e a localización. As primeira vez que creamos un proxecto dunha nova linguaxe pode pedirnos descargar e instalar módulos específicos.

O aspecto inicial de NetBeans cambia e aparecen novas ventás e pestanas dependendo do tipo de proxecto creado. Normalmente aparecen as ventás *Proyectos*, *Prestaciones* e *Archivos* que inicialmente están solapadas formando un grupo, a ventá *Navegador* que inicialmente está debaixo e o arquivo principal editado nunha pestana da ventá de edición.

- A ventá *Proyectos* permite ver os proxectos abertos e a súa estrutura dende o punto de vista de NetBeans que non é a mesma que a que se pode ver dende o sistema operativo.
- A ventá *Archivos* permite ver as carpetas e arquivos dos proxectos abertos tal e como se ven no sistema operativo. A carpeta *nbproject* sempre aparece e contén información que necesita NetBeans para xestionar o proxecto.
- A ventá *Prestaciones/Servizos* permite enlazar con servizos.

- A ventá *Navegador* visualiza a estrutura en forma de árbore dos elementos que forman o código editado na ventá de edición.



Normalmente, o arquivo principal aparece editado na pestana de edición co código que NetBeans ten establecido como modelo para este tipo de arquivos, pero se non é así, faise dobre clic sobre o nome do arquivo na ventá de *Proyecto* para editalo.

NetBeans dispón dun sistema de axuda en liña para escribir código adaptado á linguaxe que se estea utilizando na ventá de edición.

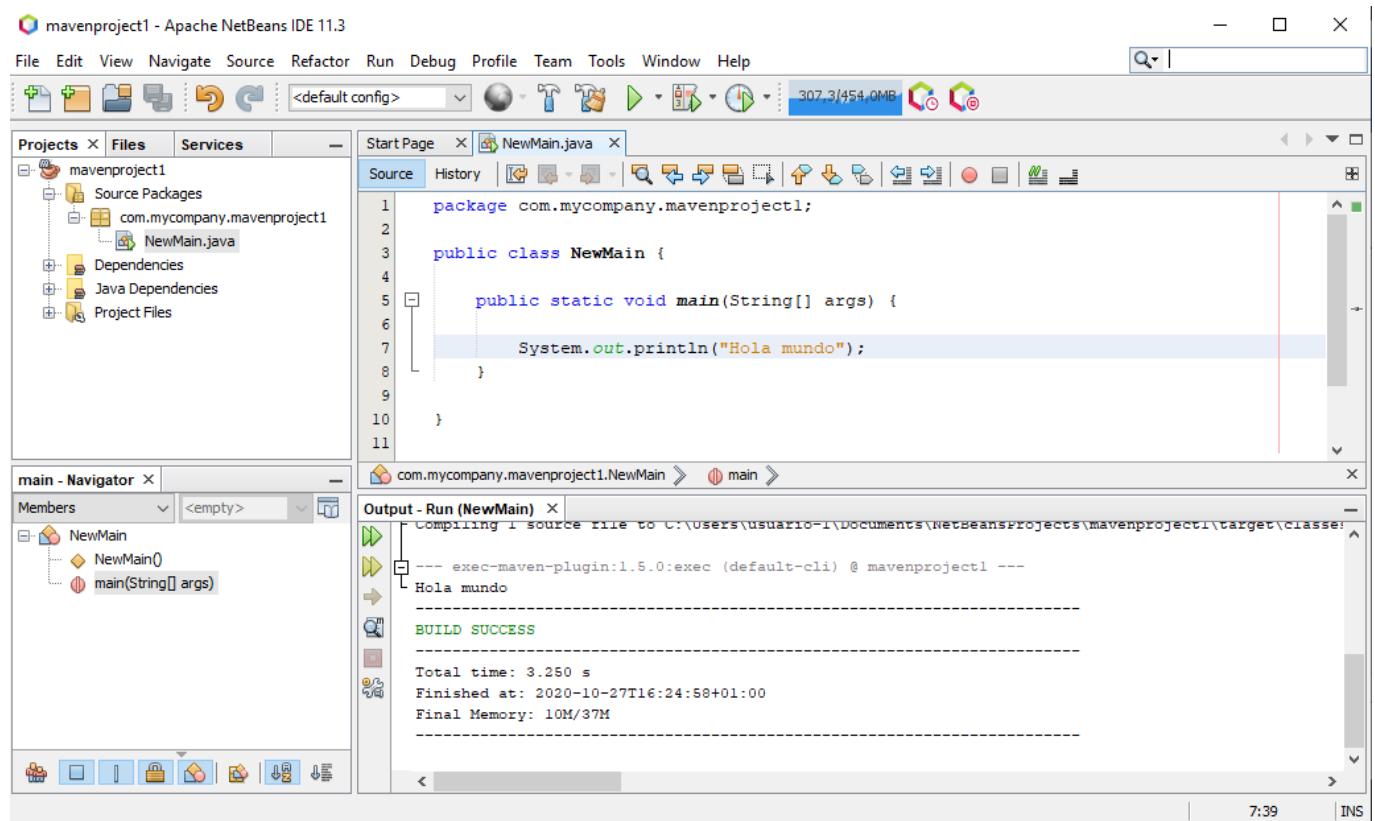
O peche dun proxecto pode facerse iluminando o proxecto na ventá *Proyecto*, clic dereito e *Cerrar*.

Pode haber varios proxectos abertos ao mesmo tempo pero o principal é o que estea seleccionado ou sobre o que se estea traballando.

Cando abrimos un novo proxecto Java temos que indicar se é Maven, Gradle ou Ant. A diferenza é a xestión de dependencias. Se o creamos Ant xa crea unha clase principal co método main().

Se a creamos Maven, deberemos facer nós. Na ventá de proxectos > Source Packages > Nome do paquete > Botón dereito > New > Java Main Class

A primeira vez descargará e instalará os plugins precisos. Podes compilalo e executalo premendo F6.



### Tarefa 2.2. Instala Netbeans na súa versión actual e crea un proxecto novo Java.

A entrega da práctica debe cubrir os seguintes aspectos:

- 1) Descarga de jdk: De onde? Que versión? Tipo de licencia
- 2) Como instalo o jdk e onde (descomprimir, instalador....)
- 3) Descarga de netbeans: De onde? Que versión? Tipo de licencia
- 4) Pasos da instalación Netbeans e onde (descomprimir, instalador....)
- 5) Configuración final (ver etc/netbeans.conf)
- 6) Pasos finais para crear un proxecto

Podes respostar a estes apartados con texto e incorporar algunha captura de pantalla dalgún aspecto concreto para que quede más clara a explicación.

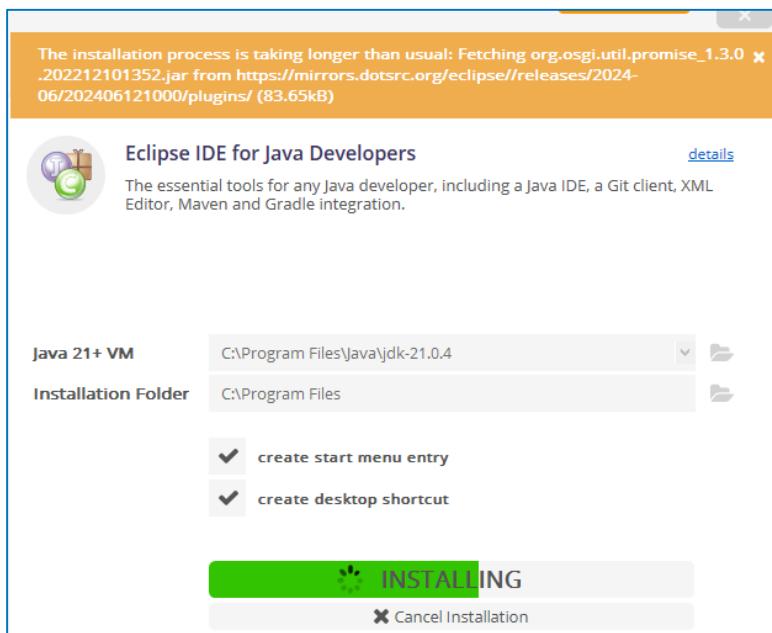
## 1.5 Contorno de desenvolvemento libre Eclipse

### Instalación

Descargamos o arquivo de instalación desde <https://www.eclipse.org/downloads/> e executamos. Na instalación, eliximos a primeira opción (Java Developers):

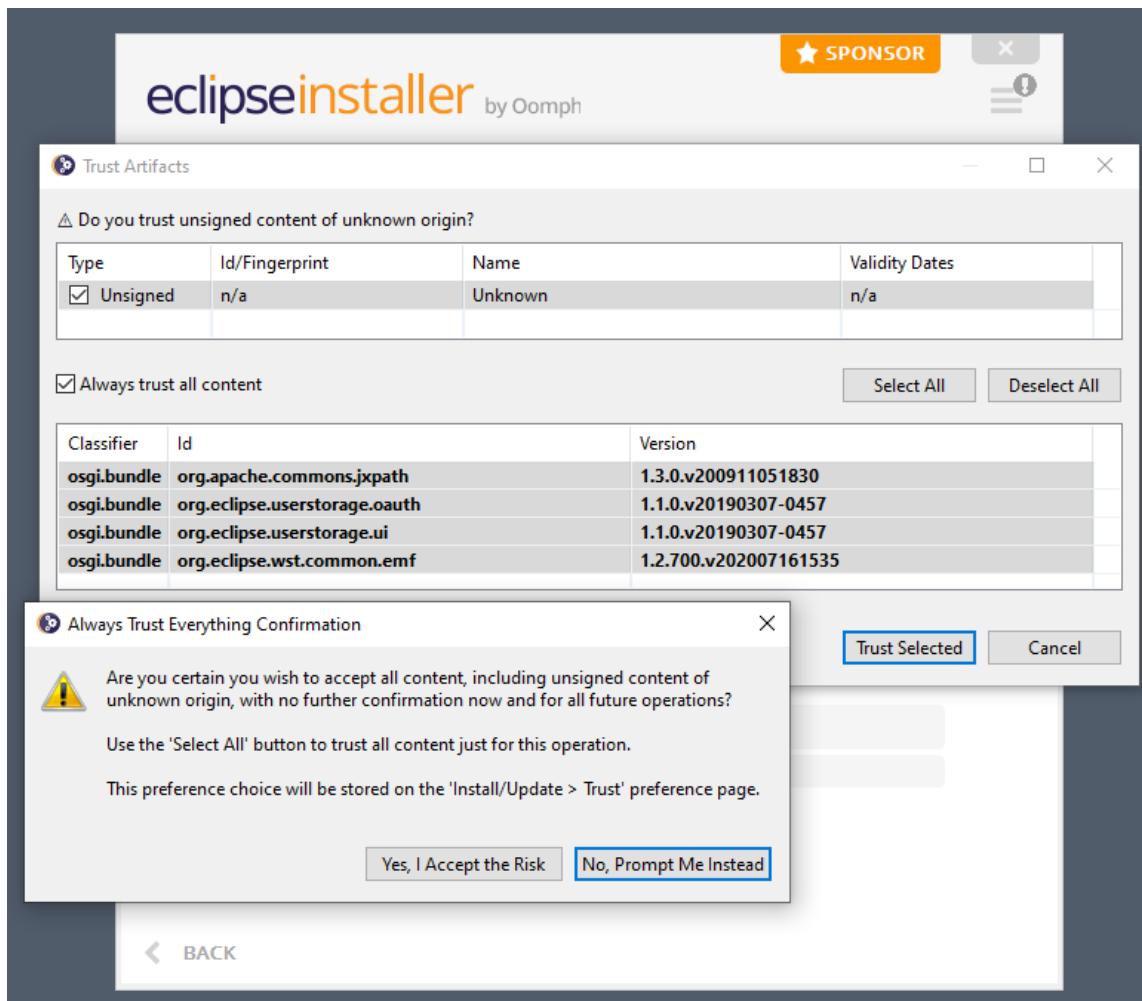


Seleccionamos a ubicación do jdk, que teremos instalado previamente, da instalación de Netbeans (en realidad solicita o jre, pero xa sabemos que o jdk inclúe o jre) e a ubicación onde instalará Eclipse (se queremos elixir *Archivos de Programa* tiñamos que executar o instalador en modo administrador)



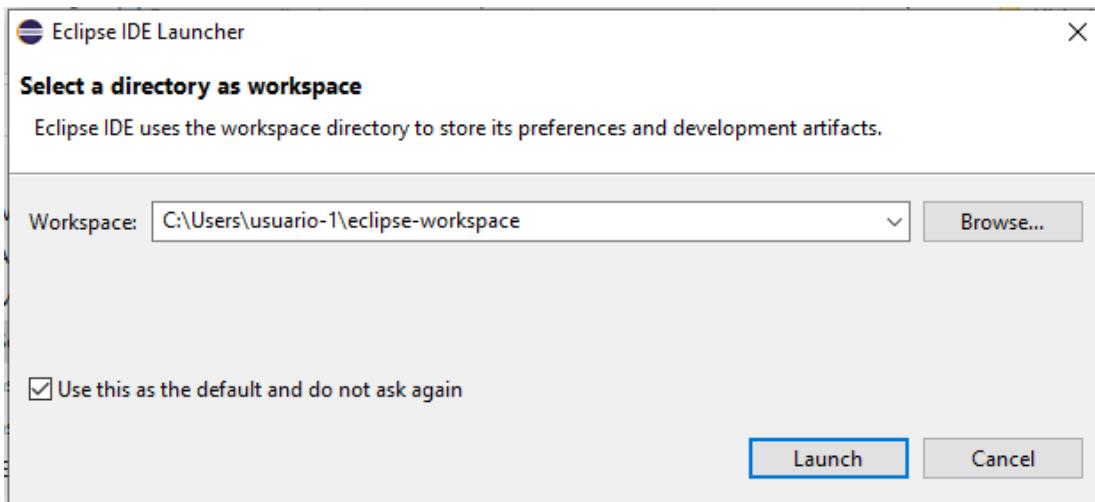
Durante a instalación, pediranos confirmación para a instalación de compoñentes de orixe descoñecido. Marcamos o check: *Always trust all content* e confirmamos no diálogo emerxente. Premendo *Trust selected* continua a instalación ata o final.

Na versión actual, Eclipse precisa da versión 21 de Java. Se máis adiante queremos modificar a versión do jre: menú window > Preferences > Java > Installed jre's. En versións anteriores era preciso incluir a ruta do jdk na variable de entorno path.

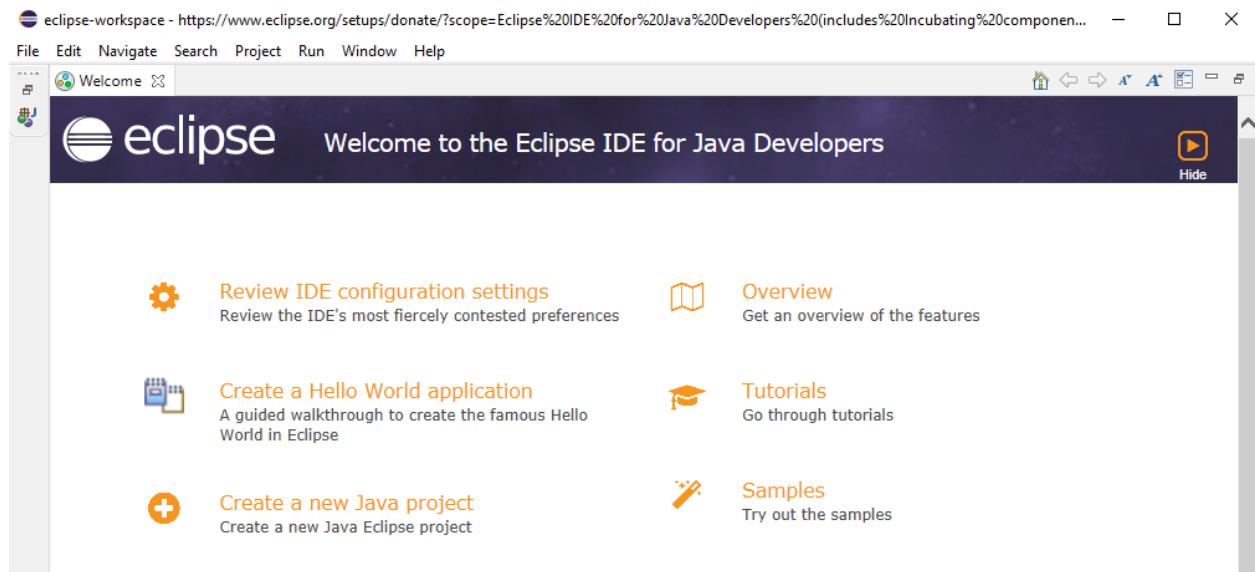


## Início

Na primeira execución, pídenos o cartafol orixe para gardar os nosos proxectos. Podemos deixar o valor por defecto, marcar o check para que non volva a preguntar e premer *Launch*.



Chegando a pantalla de inicio.

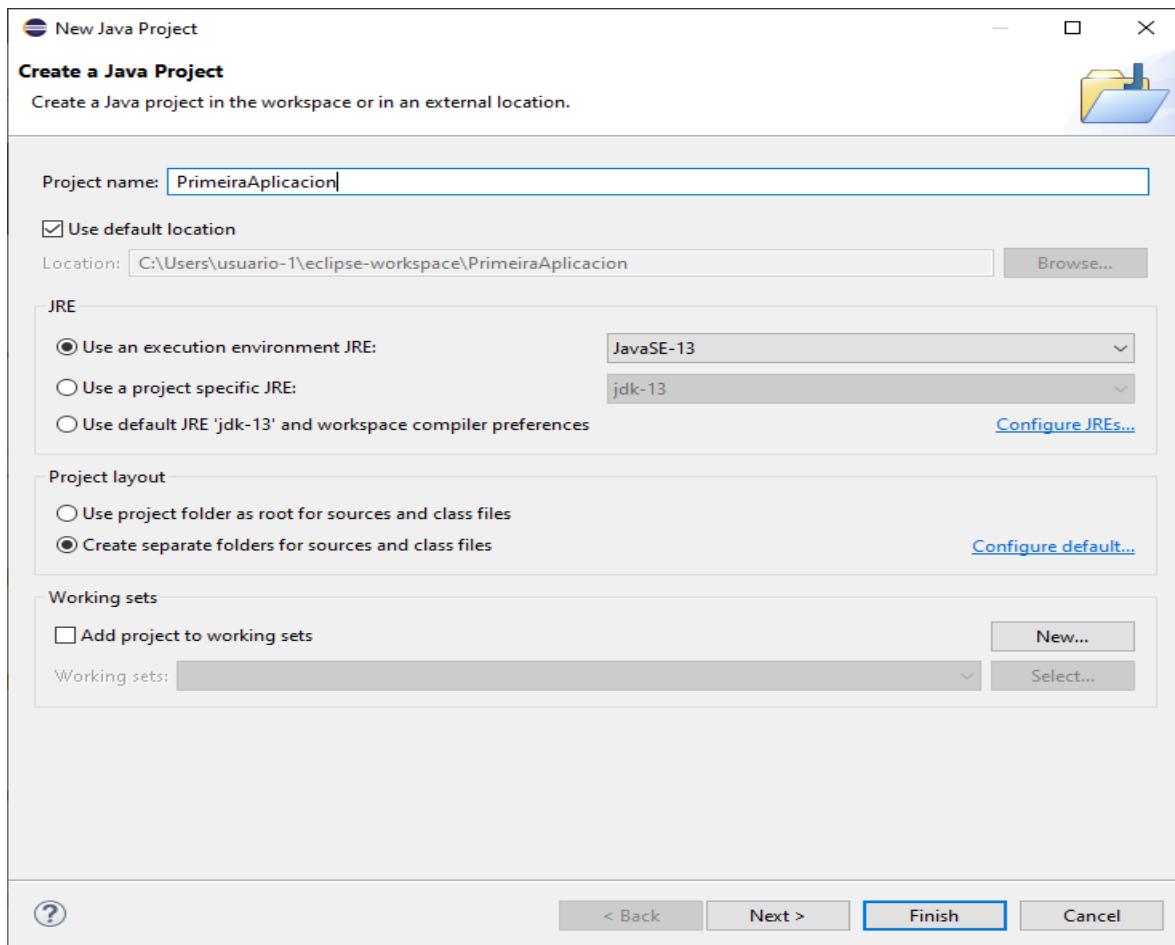


Na que podemos selección “Create a new Java project”

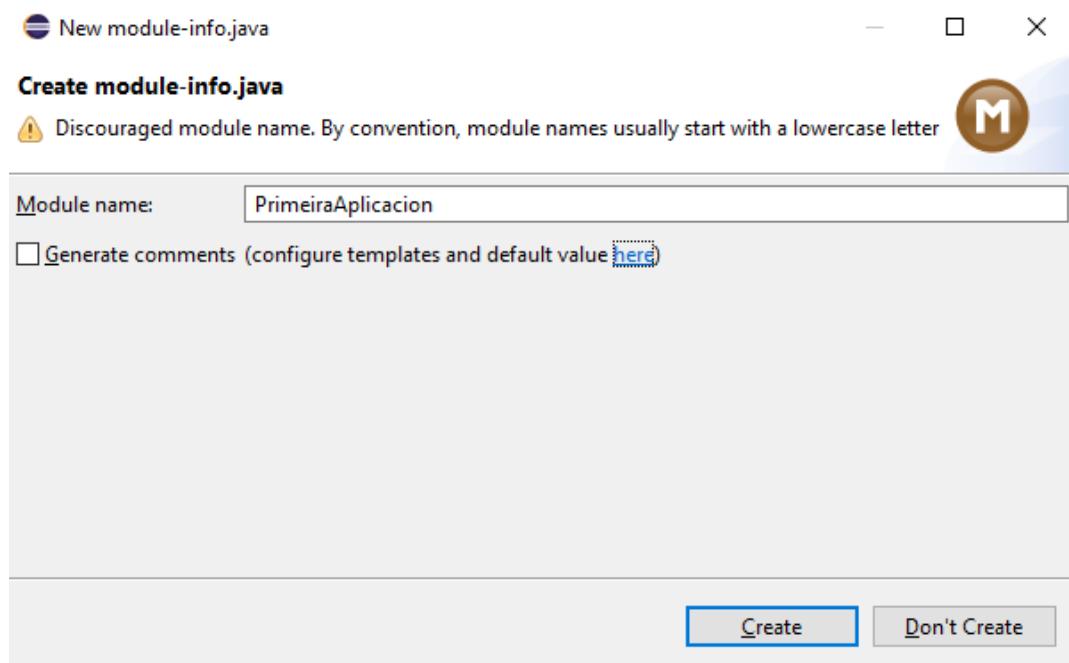
En versións anteriores, ás veces, é preciso actualizar a variable de contorno PATH engadindo a ruta onde está situados os executables de Java (...\\jdk21\\bin) (Panel de Control > Sistema > Configuración avanzada).

## Primeiros pasos

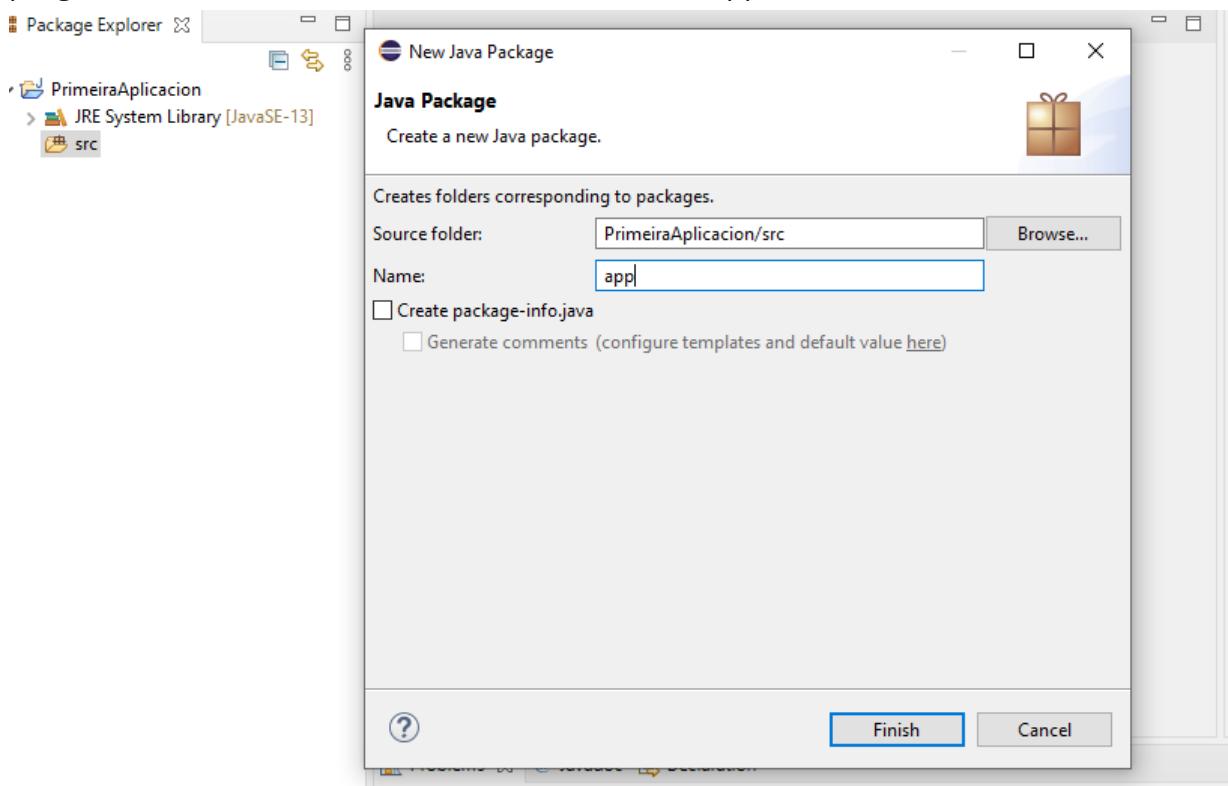
Creamos o novo proxecto dándolle un nome e deixando o resto de parámetros por defecto.



Prememos [Finish] e lle dicimos que non cree o módulo: [Don't Create]

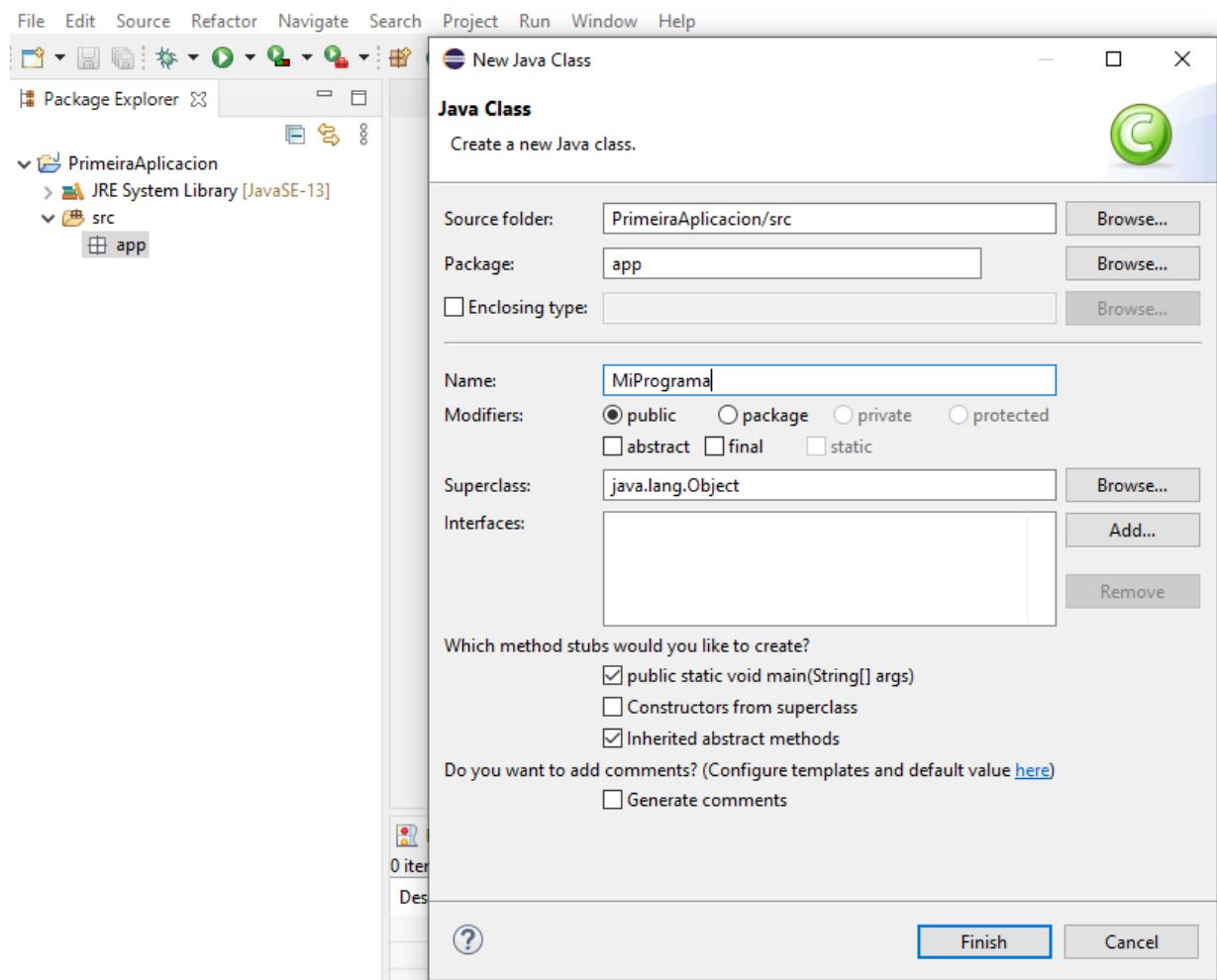


No proxecto, sobre o cartafol 'src' creamos un novo paquete que conterá as clases Java (os programas tamén son clases). Podemos chamarlle 'app':

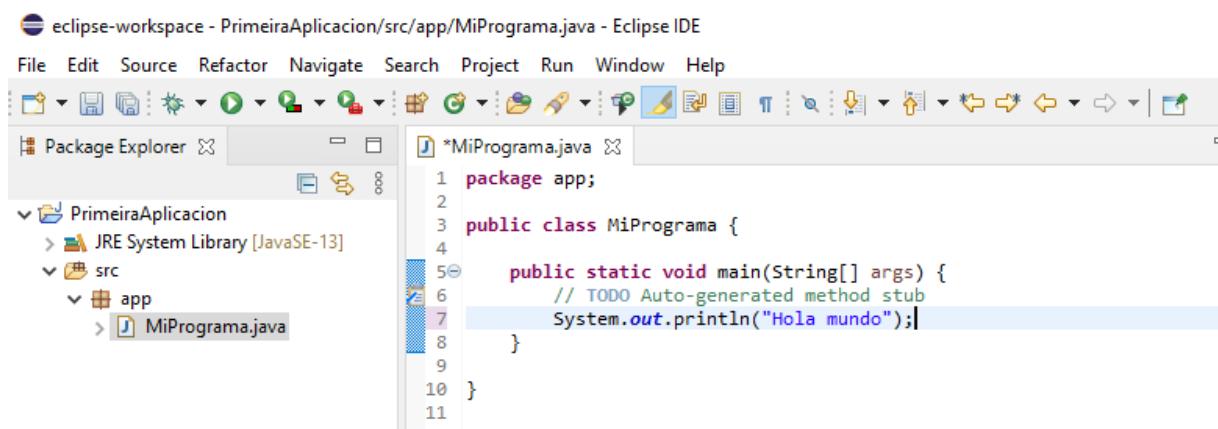


Sobre o paquete creado > New > Class.

Chamámoslle "MiPrograma" e marcamos o check: *public static void main...*



Agora xa podemos introducir o código e executalo (ctrl+F11).



**Tarefa 2.3. Instala Eclipse e crea un proxecto novo Java.**

# 1.6 Contorno de desenvolvemento Visual Studio Code

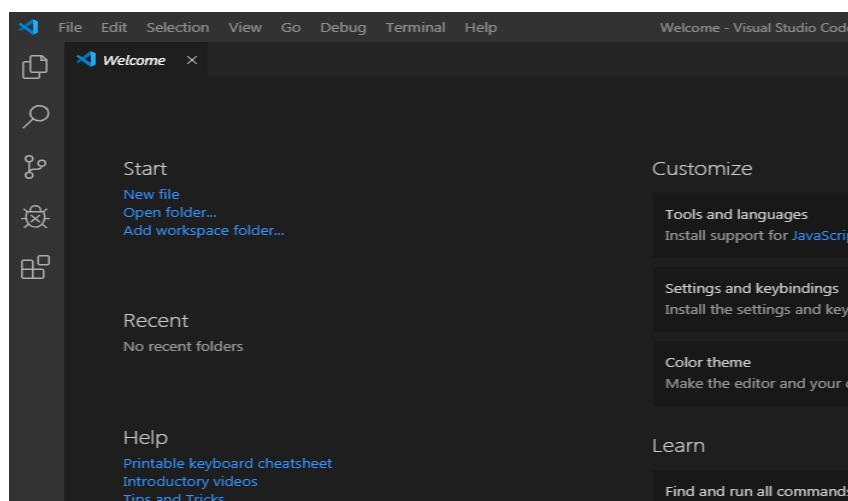
## Instalación

Instalarase a última versión de Visual Studio Code. Esta versión está disponible para OS X, Windows e Linux, sendo preciso o .NET Framework (versión 4.5.2 ou superior). Pódese consultar os requisitos en <https://code.visualstudio.com/docs/supporting/requirements>.

Descargamos a versión System Installer x64 (para todos os usuarios do ordenador) e executamos o instalador sen cambiar ningún parámetro.

## Inicio

Executamos e chegamos á pantalla de benvida.

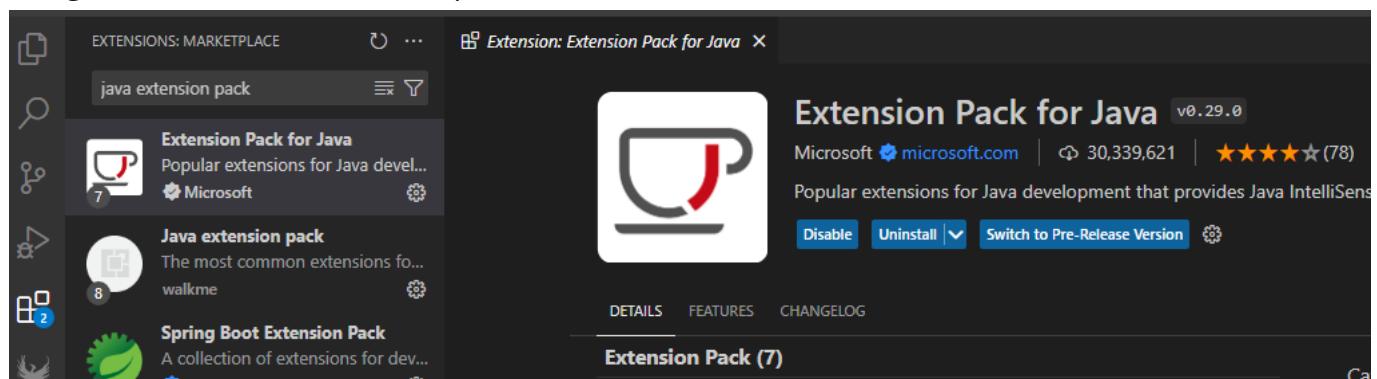


A páxina de inicio ten enlaces con diferentes informacións relacionadas con Visual Studio e no botón apartado *Reciente* ten accesos rápidos aos últimos proxectos utilizados. Pódese obter máis axuda no menú *HELP*.

O contorno péchase dende o menú principal seleccionando *Archivo>Salir*.

Hai versións orientadas a distintas linguaxes, polo que se descargamos a versión de Java xa poderíamos traballar directamente, pero esas versións non están dispoñibles para todos os versións operativos. No noso caso, veremos os pasos as seguir no caso que a nosa versión non teña soporte para Java. Visual Studio ten un *market* de extensións que permite engadir novas funcionalidades ao programa.

Dirixímonos ao xestor de extensións mediante o botón lateral ou ben premendo *Ctrl+Maiusc+X*, buscamos "Extension Pack for Java" e instalámolo se non está instalada. Este "pack" instala varias extensións necesarias para codificar Java (debugger, completador de código, test runner, xestor de dependencias Maven, etc.)



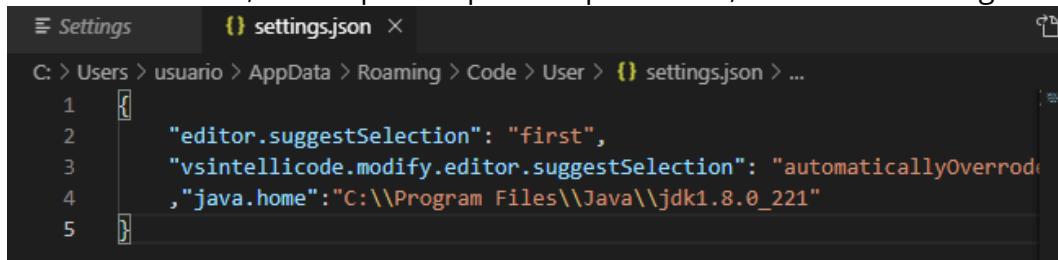
O seguinte paso da instalación sería a configuración na que indicamos onde se atopa o jdk. Para iso modificamos a variable de contorno PATH engadindo a ruta do JDK/bin.

Logo tamén facemos a mesma configuración xa no propio IDE. Para iso:

- Abrimos a paleta de comandos: Menú View > Command Palette (Ctrl + Shift + P)
- Escribimos: Preferences > Open Settings (JSON)
- No arquivo que aparece escribimos a ubicación do jdk. Por exemplo:

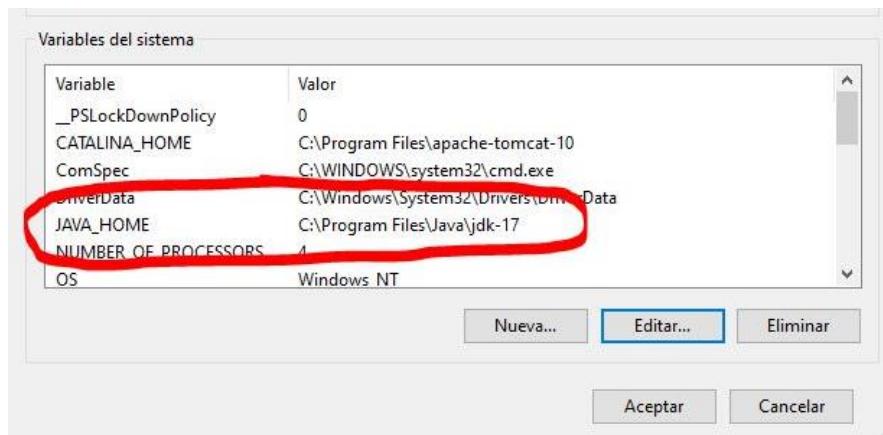
```
"java.home" : "C:\\Java\\jdk-21" (deprecated)  
"java.jdt.ls.java.home" : "C:\\Java\\jdk-21"
```

Se hai outras variables, teñen que ir separadas por comas, como se ve na seguinte imaxe:



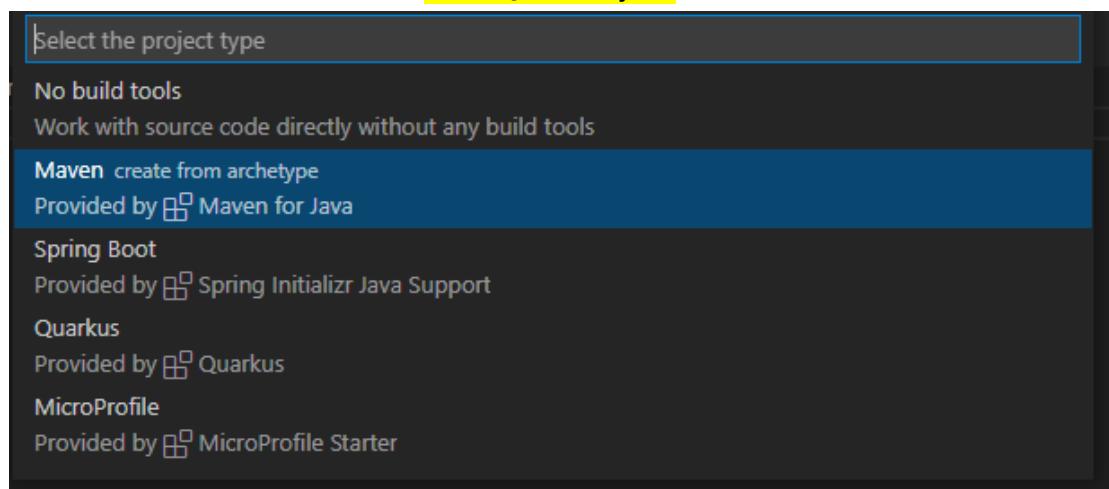
```
Settings settings.json  
C: > Users > usuario > AppData > Roaming > Code > User > settings.json > ...  
1 {  
2     "editor.suggestSelection": "first",  
3     "vsintellicode.modify.editor.suggestSelection": "automaticallyOverrodeDefaultValue",  
4     "java.home": "C:\\Program Files\\Java\\jdk1.8.0_221"  
5 }
```

- Pechamos a pestana *settings.json* gardando os cambios.
- Para proxectos Maven e imprescindible ter definida a variable de contorno no sistema operativo: JAVA\_HOME, e debe conter a ruta do jdk, polo debemos definir esta variable de contorno:



## Primeiros pasos

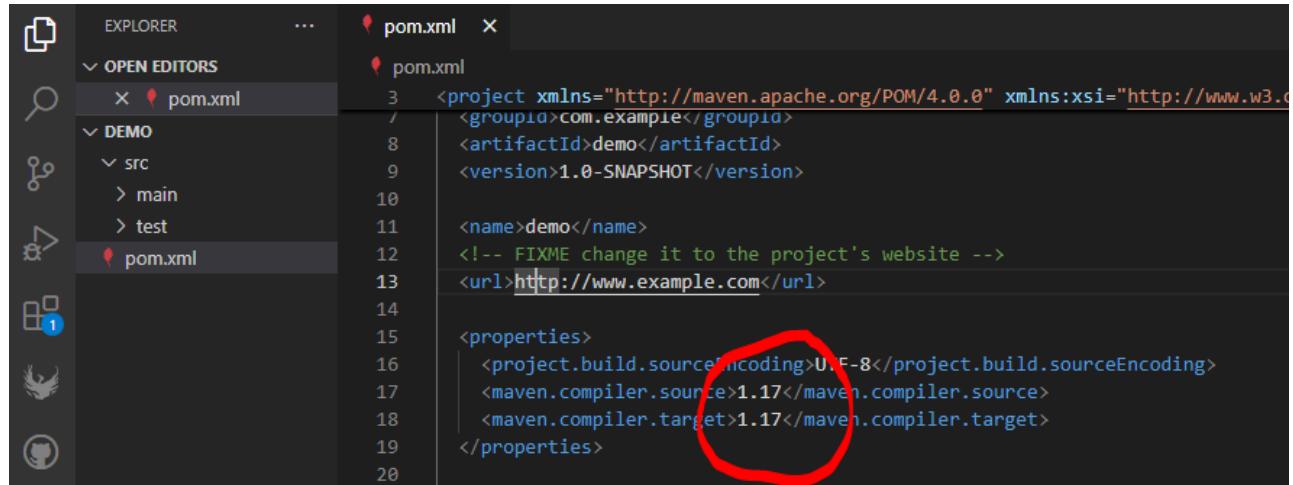
Para crear un proxecto Java a opción más rápida é abrir a paleta de comandos (menú View > Command Palette) e tecleando *Create Java Project*.



No tipo de proxecto: *Maven > maven-archetype-quickstart*. Logo deixamos os seguintes parámetros por defecto (*versión, group id*) e dámosselle o nome que queremos ao *artifact Id*, que será o nome do noso programa. Finalmente, pídenos o nome do cartafol no que queremos xerar todo o proxecto e fainos na consola unhas preguntas finais:

- Define value for property 'version' 1.0-SNAPSHOT. > Prememos [Enter]
- Confirm properties configuration: Y: > Prememos [Enter]

Automaticamente crearase a estrutura de cartafoles necesarios, entre eles *src > app* e alí o arquivo *App.java*. Antes de traballar con el, debemos facer un cambio no arquivo *pom.xml* que vemos na raíz do proxecto xerado, para fixar a versión do JDK empregado, no noso caso a 17:

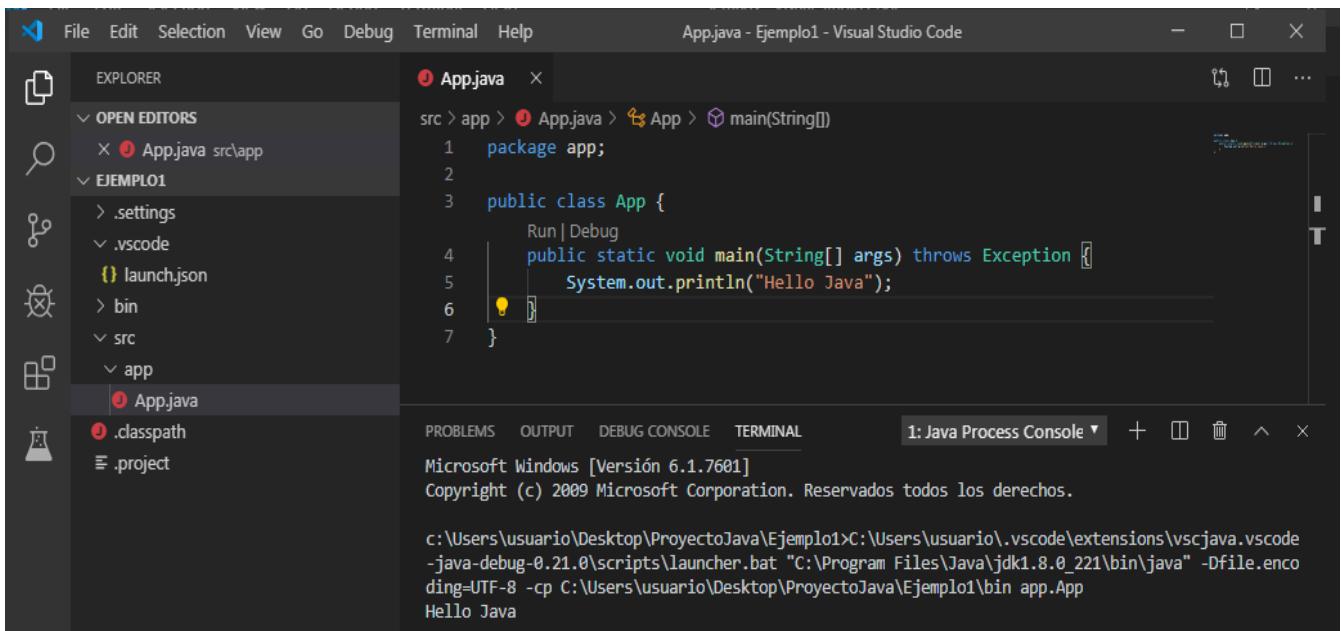


```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <groupId>com.example</groupId>
    <artifactId>demo</artifactId>
    <version>1.0-SNAPSHOT</version>
    <name>demo</name>
    <!-- FIXME change it to the project's website -->
    <url>http://www.example.com</url>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>1.17</maven.compiler.source>
        <maven.compiler.target>1.17</maven.compiler.target>
    </properties>

```

Agora xa poderemos editar o proxecto finalmente executalo, ben con *Ctrl+F5*, ben coa ligazón *Run* que vemos na clase que contén o método *main*.



```

package app;

public class App {
    public static void main(String[] args) throws Exception {
        System.out.println("Hello Java");
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
1: Java Process Console

```

Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

c:\Users\usuario\Desktop\ProyectoJava\Ejemplo1>C:\Users\usuario\.vscode\extensions\vscjava.vscode
-javadebug-0.21.0\scripts\launcher.bat "C:\Program Files\Java\jdk1.8.0_221\bin\java" -Dfile.encoding=UTF-8 -cp C:\Users\usuario\Desktop\ProyectoJava\Ejemplo1\bin app.App
Hello Java

```

Outras características que debemos coñecer no noso primeiro contacto coa ferramenta son:

- Na ventá dereita "*Java Dependencies*" podemos ver estrutura de cartafoles completa do proxecto, de forma similar a Netbeans.
- Configuración: está no menú *File > Preferences > Settings*, e proporciona multitud de parámetros para personalizar.
- A *paleta de comandos*: pulsando "Cntrl" + "May" + "P" chegamos a ela e podemos escribir o que queres facer.



**Tarefa 2.4. Instala Visual Studio Code e crea un proxecto novo Java.** Fai unha captura de pantalla da estrutura de arquivos do proxecto, modifica o tamaño da letra (Settings), e executa algún comando desde a paleta.

## 1.7 Contorno de desenvolvemento IntelliJ Idea

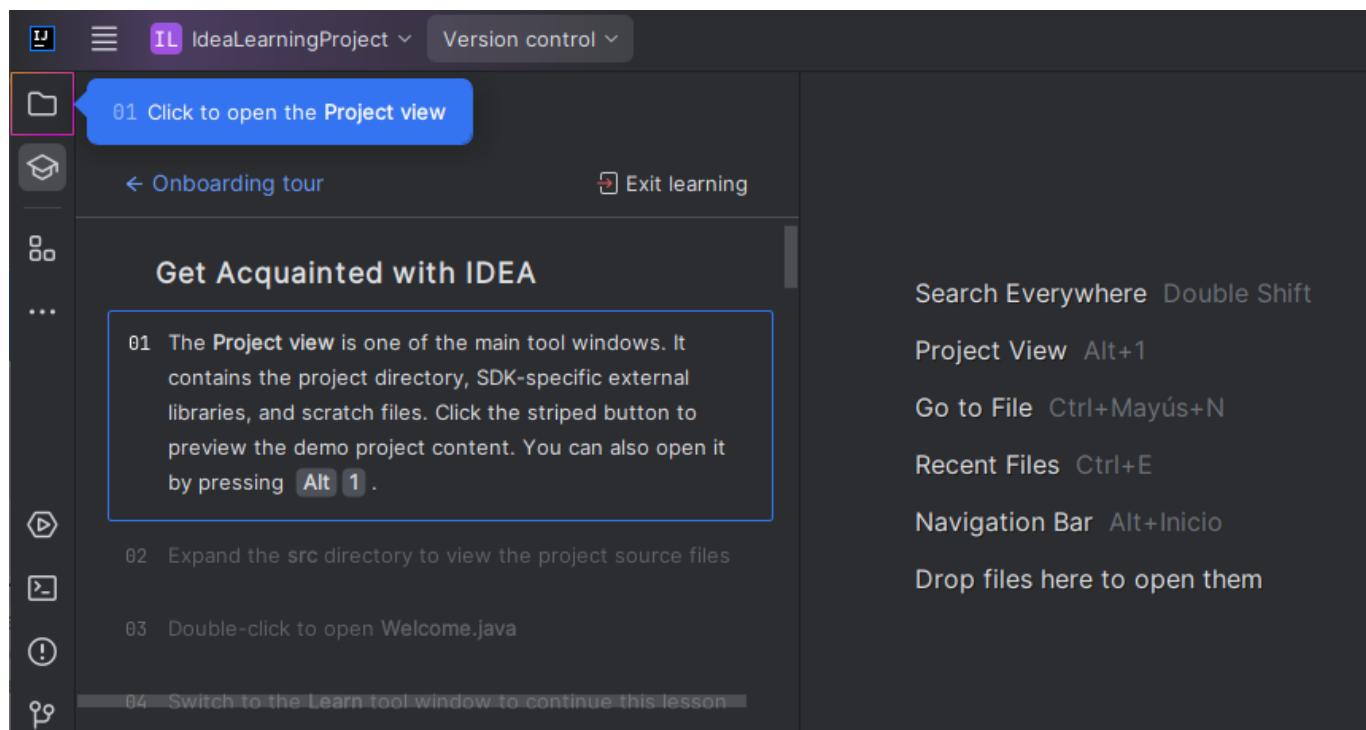
### Instalación

Na páxina <https://www.jetbrains.com/idea/download> podemos descargar a versión de pago (Ultimate) para distintos sistemas operativos. Premendo na ligazón “Other versions” diríxenos á versión gratuítia para uso non comercial (Community). No noso caso, descargamos o instalador para Windows: “Windows x64 (exe)”.

Seguimos a instalación sen ningunha configuración particular.

### Configuración

A primeira vez que o executamos, ofrécenos a posibilidade de importar a configuración de Visual Studio Code, en caso de que o tiveramos instalado previamente no noso ordenador, e de seguido, un “on boarding tour” para ver as súas funcionalidades más importantes.



A aparencia é similar a Visual Studio Code pero sen a paleta de comandos e cun menú superior máis completo.

Un dos aspectos más importantes dos IDEs, non so de este, é coñecer as súas teclas rápidas, isto é, os comandos que podemos executar mediante unha combinación del teclas, sen ter que ir co rato a buscar a opción de menú para facer a tarefa. Ao principio custa un pouco aprenderla, pero coa práctica, o seu uso aumenta notablemente a nosa produtivididade.

Na seguinte ligazón tes un resumo das más importantes en IntelliJ:

<https://ruleoftech.com/2015/essential-intellij-idea-keyboard-shortcuts>

aínda que este IDE permite cambiaras para empregar as mesmas que outros IDE (menú: *File > Settings > KeyMap*). Isto pode ser útil cando comezamos a usalo e vimos doutro IDE.

### IntelliJ IDEA Default Keymap



Editing	
<b>Ctrl + Space</b>	Basic code completion (the name of any class, method or variable)
<b>Ctrl + Shift + Space</b>	Smart code completion (filters the list of methods and variables by expected type)
<b>Ctrl + Shift + Enter</b>	Complete statement
<b>Ctrl + P</b>	Parameter info (within method call arguments)
<b>Ctrl + Q</b>	Quick documentation lookup
<b>Shift + F1</b>	External Doc
<b>Ctrl + mouse over code</b>	Brief info
<b>Ctrl + F1</b>	Show descriptions of error or warning at caret
<b>Alt + Insert</b>	Generate code... (Getters, Setters, Constructors, hashCode>equals, toString)
<b>Ctrl + O</b>	Override methods
<b>Ctrl + I</b>	Implement methods
<b>Ctrl + Alt + T</b>	Surround with... (if, else, try, catch, for, synchronized, etc.)
<b>Ctrl + /</b>	Comment/uncomment with line comment
<b>Ctrl + Shift + /</b>	Comment/uncomment with block comment
<b>Ctrl + W</b>	Select successively increasing code blocks
<b>Ctrl + Shift + W</b>	Decrease current selection to previous state
<b>Alt + Q</b>	Context info
<b>Alt + Enter</b>	Show intention actions and quick-fixes
<b>Ctrl + Alt + L</b>	Reformat code
<b>Ctrl + Alt + O</b>	Optimize imports
<b>Ctrl + Alt + I</b>	Auto-indent line(s)
<b>Tab / Shift + Tab</b>	Indent/unindent selected lines
<b>Ctrl + X or Shift + Delete</b>	Cut current line or selected block to clipboard
<b>Ctrl + C or Ctrl + Insert</b>	Copy current line or selected block to clipboard
<b>Ctrl + V or Shift + Insert</b>	Paste from clipboard
<b>Ctrl + Shift + V</b>	Paste from recent buffers
<b>Ctrl + D</b>	Duplicate current line or selected block
<b>Ctrl + Y</b>	Delete line at caret
<b>Ctrl + Shift + J</b>	Smart line join
<b>Ctrl + Enter</b>	Smart line split
<b>Shift + Enter</b>	Start new line
<b>Ctrl + Shift + U</b>	Toggle case for word at caret or selected block
<b>Ctrl + Shift + J / [</b>	Select till code block end/start
<b>Ctrl + Delete</b>	Delete to word end
<b>Ctrl + Backspace</b>	Delete to word start
<b>Ctrl + NumPad+/-</b>	Expand/collapse code block
<b>Ctrl + Shift + NumPad+</b>	Expand all
<b>Ctrl + Shift + NumPad-</b>	Collapse all
<b>Ctrl + F4</b>	Close active editor tab
Search/Replace	
<b>Double Shift</b>	Search everywhere
<b>Ctrl + F</b>	Find
<b>F3</b>	Find next
<b>Shift + F3</b>	Find previous
<b>Ctrl + R</b>	Replace
<b>Ctrl + Shift + F</b>	Find in path
<b>Ctrl + Shift + R</b>	Replace in path
<b>Ctrl + Shift + S</b>	Search structurally (Ultimate Edition only)
<b>Ctrl + Shift + M</b>	Replace structurally (Ultimate Edition only)

[www.jetbrains.com/idea](http://www.jetbrains.com/idea)

### IntelliJ IDEA Default Keymap



Usage Search	
<b>Alt + F7 / Ctrl + F7</b>	Find usages / Find usages in file
<b>Ctrl + Shift + F7</b>	Highlight usages in file
<b>Ctrl + Alt + F7</b>	Show usages
Compile and Run	
<b>Ctrl + F9</b>	Make project (compile modified and dependent)
<b>Ctrl + Shift + F9</b>	Compile selected file, package or module
<b>Alt + Shift + F10</b>	Select configuration and run
<b>Alt + Shift + F9</b>	Select configuration and debug
<b>Shift + F10</b>	Run
<b>Shift + F9</b>	Debug
<b>Ctrl + Shift + F10</b>	Run context configuration from editor
Debugging	
<b>F8</b>	Step over
<b>F7</b>	Step into
<b>Shift + F7</b>	Smart step into
<b>Shift + F8</b>	Step out
<b>Alt + F9</b>	Run to cursor
<b>Alt + F8</b>	Evaluate expression
<b>F9</b>	Resume program
<b>Ctrl + F8</b>	Toggle breakpoint
<b>Ctrl + Shift + F8</b>	View breakpoints
Navigation	
<b>Ctrl + N</b>	Go to class
<b>Ctrl + Shift + N</b>	Go to file
<b>Ctrl + Alt + Shift + N</b>	Go to symbol
<b>Alt + Right/Left</b>	Go to next/previous editor tab
<b>F12</b>	Go back to previous tool window
<b>Esc</b>	Go to editor (from tool window)
<b>Shift + Esc</b>	Hide active or last active window
<b>Ctrl + Shift + F4</b>	Close active run/messages/find/.. tab
<b>Ctrl + G</b>	Go to line
<b>Ctrl + E</b>	Recent files popup
<b>Ctrl + Alt + Left/Right</b>	Navigate back/forward
<b>Ctrl + Shift + Backspace</b>	Navigate to last edit location
<b>Alt + F1</b>	Select current file or symbol in any view
<b>Ctrl + B or Ctrl + Click</b>	Go to declaration
<b>Ctrl + Alt + B</b>	Go to implementation(s)
<b>Ctrl + Shift + I</b>	Open quick definition lookup
<b>Ctrl + Shift + B</b>	Go to type declaration
<b>Ctrl + U</b>	Go to super-method/super-class
<b>Alt + Up/Down</b>	Go to previous/next method
<b>Ctrl + J / {</b>	Move to code block end/start
<b>Ctrl + F12</b>	File structure popup
<b>Ctrl + H</b>	Type hierarchy
<b>Ctrl + Shift + H</b>	Method hierarchy
<b>Ctrl + Alt + H</b>	Call hierarchy
<b>F2 / Shift + F2</b>	Next/previous highlighted error
<b>F4 / Ctrl + Enter</b>	Edit source / View source
<b>Alt + Home</b>	Show navigation bar
<b>F11</b>	Toggle bookmark
<b>Ctrl + F11</b>	Toggle bookmark with mnemonic
<b>Ctrl + # [0-9]</b>	Go to numbered bookmark
<b>Shift + F11</b>	Show bookmarks

[blogs.jetbrains.com/idea](http://blogs.jetbrains.com/idea)

### IntelliJ IDEA Default Keymap



Refactoring	
<b>F5</b>	Copy
<b>F6</b>	Move
<b>Alt + Delete</b>	Safe Delete
<b>Shift + F6</b>	Rename
<b>Ctrl + F6</b>	Change Signature
<b>Ctrl + Alt + N</b>	Inline
<b>Ctrl + Alt + M</b>	Extract Method
<b>Ctrl + Alt + V</b>	Extract Variable
<b>Ctrl + Alt + F</b>	Extract Field
<b>Ctrl + Alt + C</b>	Extract Constant
<b>Ctrl + Alt + P</b>	Extract Parameter
VCS/Local History	
<b>Ctrl + K</b>	Commit project to VCS
<b>Ctrl + T</b>	Update project from VCS
<b>Alt + Shift + C</b>	View recent changes
<b>Alt + BackQuote ('')</b>	'VCS' quick popup
Live Templates	
<b>Ctrl + Alt + J</b>	Surround with Live Template
<b>Ctrl + J</b>	Insert Live Template
<b>iter</b>	Iteration according to java SDK 1.5 style
<b>inst</b>	Check object type with instanceof and downcast it
<b>itco</b>	Iterate elements of java.util.Collection
<b>itit</b>	Iterate elements of java.util.List
<b>itli</b>	public static final
<b>psf</b>	throw new
General	
<b>Alt + # [0-9]</b>	Open corresponding tool window
<b>Ctrl + S</b>	Save all
<b>Ctrl + Synchronize</b>	Synchronize
<b>Ctrl + Shift + F12</b>	Toggle maximizing editor
<b>Alt + Shift + F</b>	Add to Favorites
<b>Alt + Shift + I</b>	Inspect current file with current profile
<b>Ctrl + BackQuote (`)</b>	Quick switch current scheme
<b>Ctrl + Alt + S</b>	Open Settings dialog
<b>Ctrl + Alt + Shift + S</b>	Open Project Structure dialog
<b>Ctrl + Shift + A</b>	Find Action
<b>Ctrl + Tab</b>	Switch between tabs and tool window

To find any action inside the IDE use  
Find Action (Ctrl+Shift+A/⌘ A)

Enter action or option name: <input type="text" value="doc"/>	<input checked="" type="checkbox"/> Include non-menu actions (Ctrl+Shift+A)
<b>doc</b>	
External Documentation (Shift+F10)	
Generate GroovyDoc	Code View Actions
Quick Documentation (Ctrl+Q)	Tools
Convert Schema...	
Collapse doc comments	Web
XML Actions	
Folding	

@intellijidea

JETBRAINS

11/2013 © JETBRAINS. Copyright © JetBrains. 2000-2013. All Rights Reserved.

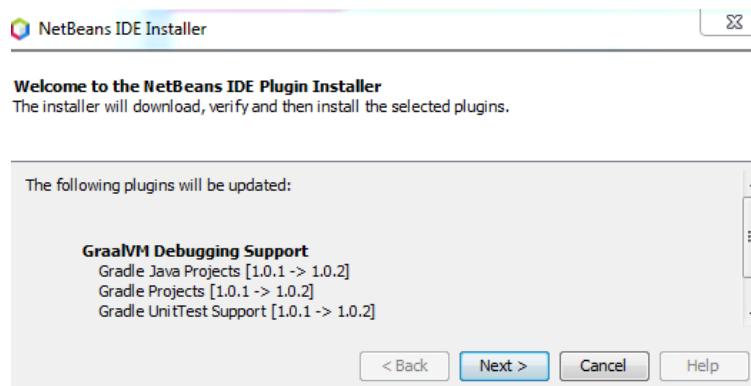
## Establecendo o JDK

Este IDE detecta os *jdk* instalados no sistema, pero no caso de ter varios, podemos indicarle cal é o que queremos que empregue cando creamos un novo proxecto desde o menú superior: *File > Project Structure > Project Settings > Project > SDK*. Incluso podemos descargar un novo *jdk* desde esa opción.

## 2. Actualizar, personalizar, automatizar e xestionar módulos do contorno

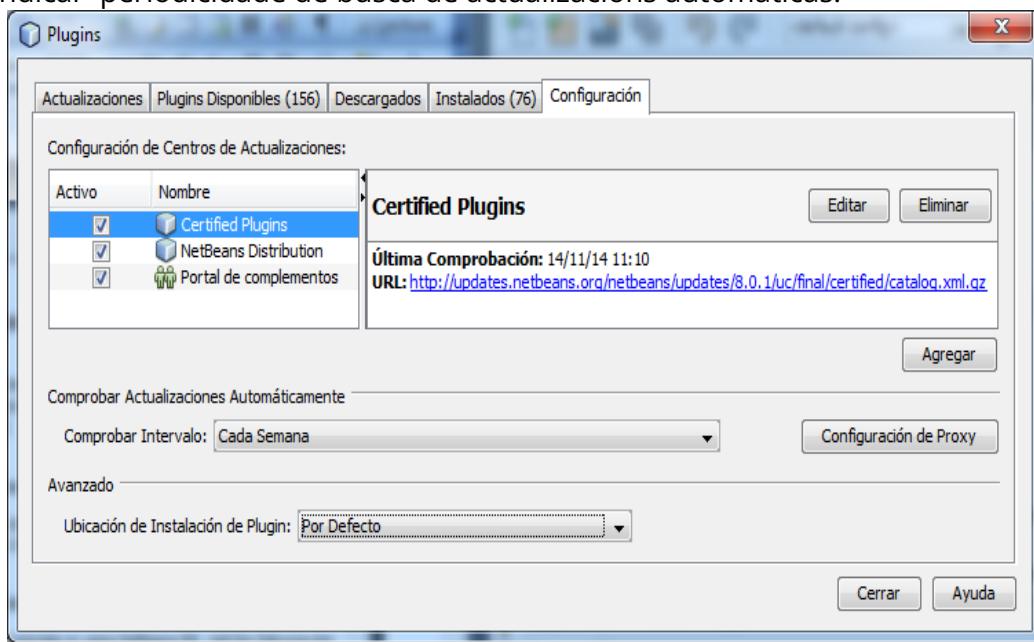
### 2.1 Actualizar NetBeans

A xestión do proceso de actualización faise na ventá Plugins á que se pode acceder dende a opción *Herramientas>Plugins* do menú principal de NetBeans o desde el menú principal: *Ayuda > Comprobar actualizaciones*.

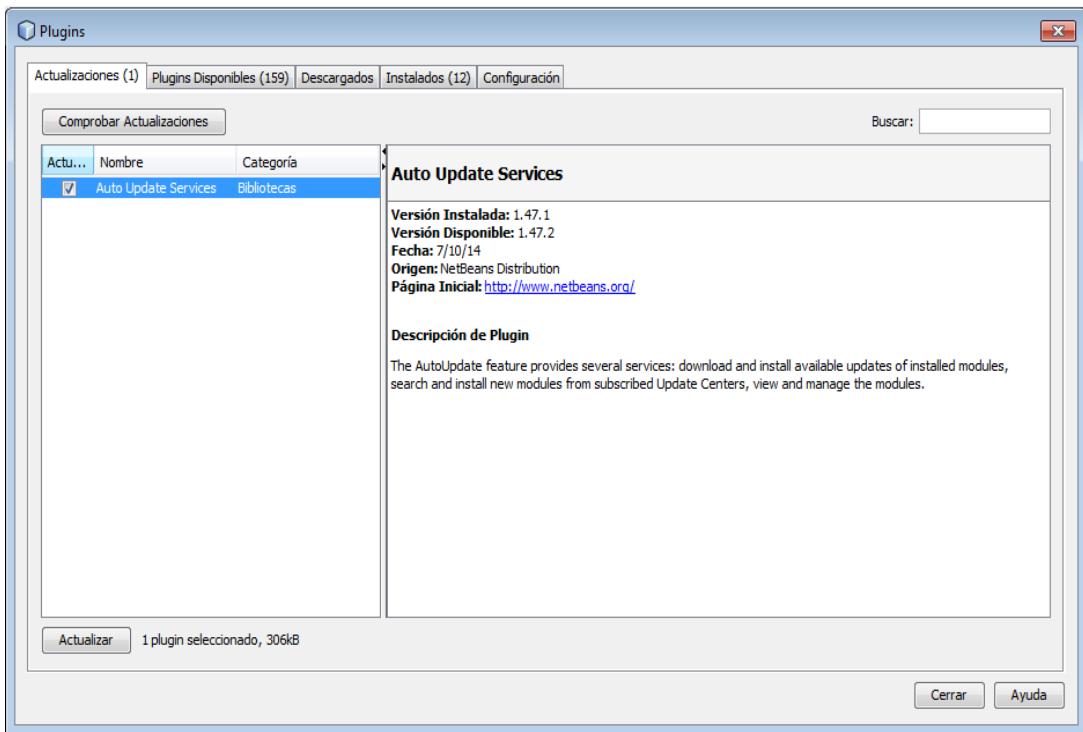


Un complemento ou *plugin* é un grupo de módulos dependentes. Estes módulos poden ser utilizados por varios complementos polo que cando se actualiza un complemento poden actualizarse algúns dos módulos que o compón e por tanto afectar a outros complementos.

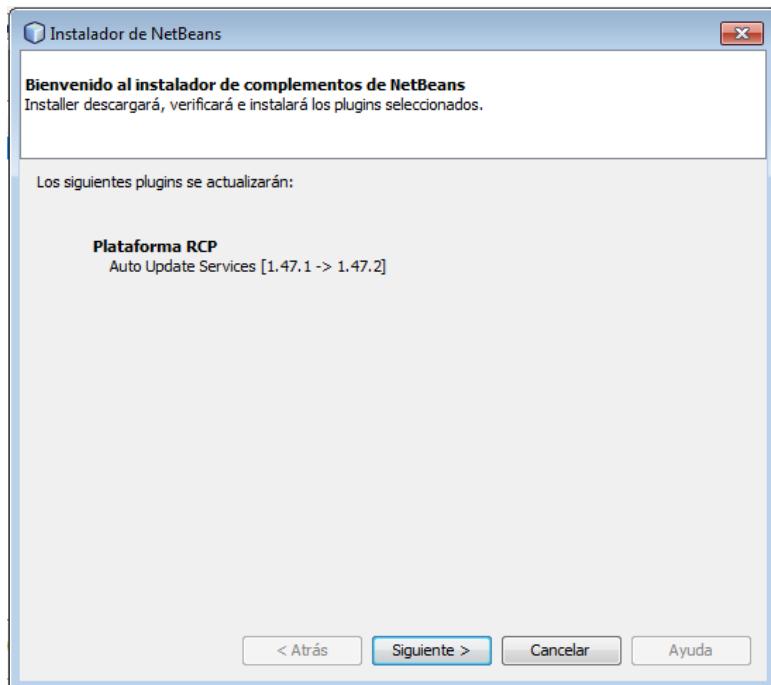
A pestana **Configuración (Settings)** da ventá *Plugins* mostra as características do centro de actualizacións. Nela aparece por defecto 3 url fontes nas que conseguir actualizacións e novos complementos e pódese engadir, activar, desactivar, editar ou eliminar urls. Na mesma pestana pódese indicar periodicidade de busca de actualizacións automáticas.



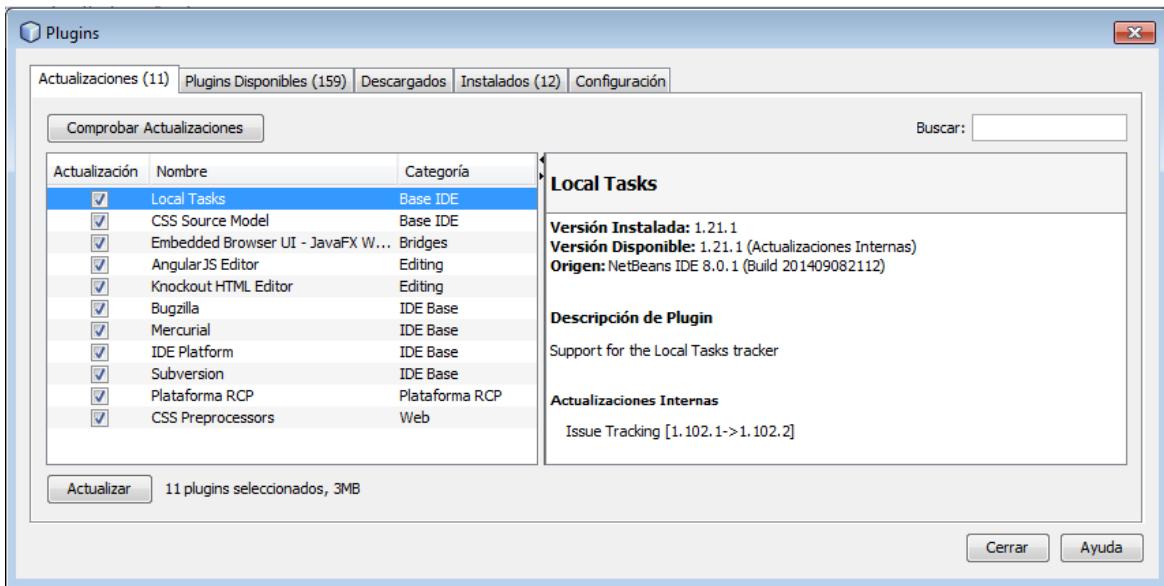
Para comprobar actualizacións en calquera momento débese utilizar a pestana *Actualizaciones (Updates)* da ventá *Plugins* e premer no botón *Comprobar Actualizaciones (Check for updates)*. Pode ocorrer que segundo a versión instalada, non teña instalado o complemento *Auto Update Service* para facilitar as actualizacións polo que esa pode ser a primeira actualización que aparezca:



De premer no botón Actualizar, o instalador informará, como fai sempre, do plugin que se vai a actualizar, e haberá que confirmar a actualización. A actualización terminará de facerse cando se reinicie o IDE.



Despois de instalar a actualización automática, pódese volver a comprobar actualizacións para que NetBeans comprobe se existen actualizacións dos complementos instalados buscando nas url declaradas no centro de actualizacións; no caso de encontrarlas, aparecerá a relación delas e haberá que seleccionar o complemento ou complementos que se queiran actualizar e premer no botón *Actualizar*. NetBeans avisará no caso de que haxa que reiniciar o contorno para completar o proceso de actualización dalgún complemento.



**Tarefa 2.5. Xestión de actualizacións.** Primeiro consulta o estado de actualización de Netbeans e actualiza os plugins. Logo engade a url <https://github.com/stapler/netbeans-stapler-plugin> ao centro de actualizacións como una nova URL para novos complementos ou actualizacións, e configura as actualizacións para que non as busque de forma automática. Este complemento ofrece compatibilidade co framework Stapler Web.



Nota: Inclúe capturas de pantalla e indica os pasos polos menús ata chegar a estas opcións.

## 2.2 Personalizar NetBeans

A personalización de NetBeans poden facer máis cómodo e amigable o traballo no IDE pero tamén hai que ter en conta que se garante máis a comunicación con outros usuarios de NetBeans se os valores por defecto se modifican moi pouco. A personalización almacénase en: C:\Users\usuario\AppData\Roaming\NetBeans\version\

A operación de personalizar o contorno pode facerse basicamente dende dúas opcións do menú principal: Ver e Herramientas>Opciones (Tools > Options).

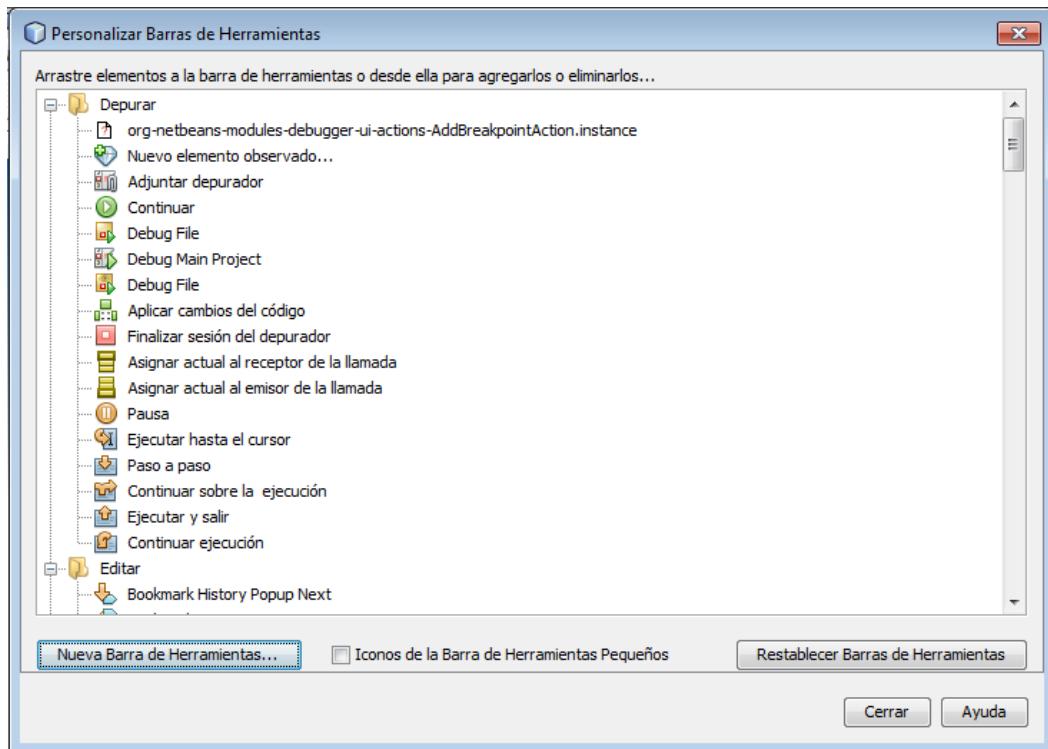
### Opción Ver

Ten moitas posibilidades para personalizar tal e como se ve ao despregar o menú Ver. O nome dalgunha das posibilidades xa indican claramente o efecto; outras precisan explicación extra como por exemplo a personalización das barras de ferramentas.

Unha barra de ferramentas está formada por un grupo de botóns asociados cada un a unha operación do contorno. Sobre unha barra de ferramentas pode:

- Deixar o ratoriba da zona punteada da esquerda da barra ata que apareza unha cruz e entón poderá ver o nome da barra.
- Deixar o ratoriba da zona punteada da esquerda da barra ata que apareza unha cruz e arrastrar (mantendo pulsado o botón esquierdo do rato) ata chegar á localización onde se quere colocar a barra onde se soltará (deixando de premer o botón esquierdo do rato).
- Mostrar ou ocultar as barras de ferramentas sen máis que activala ou desactivala no menú Ver: móstrase se non estaba visible e ocúltase se estaba visible.

- En Ver>Barra de Herramientas>Personalizar pode verse a relación de iconas asociadas a operacións do IDE.
- Mover un botón dentro dunha barra de ferramentas arrastrándoo e soltándoo no sitio desexado.



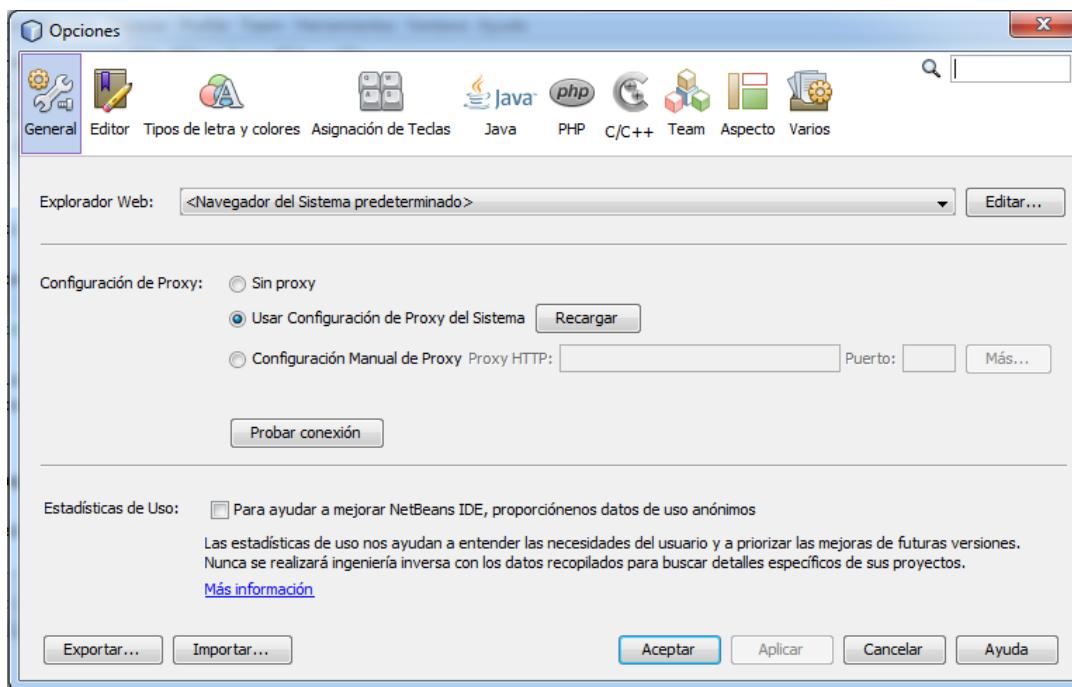
Con esta ventá aberta pódese:

- Crear unha nova barra de ferramentas premendo no botón *Nueva Barra de Herramientas*. Neste caso aparece unha nova ventá na que se debe de dar un nome á nova barra; cando se acepta, aparece a nova barra na zona de barra de ferramentas. A continuación hai que arrastrar e soltar iconas dende a ventá *Personalizar Barras de Herramientas* ata a nova barra coidando de poñelas seguidas.
- Engadir botóns a unha barra existente arrastrando o botón dende a ventá *Personalizar Barras de Herramientas* e soltándoo na barra de ferramentas.
- Quitar botóns arrastránndoos fóra da barra sempre que se teña aberta a ventá *Personalizar Barras de Herramientas*.
- Ver os botóns das barras de ferramentas en tamaño máis pequeno seleccionando o checkbox *Iconos de la Barra de Herramientas Pequeños*.
- Restablecer as barras de ferramentas eliminando as barras creadas polo usuario premendo no botón *Restablecer Barras de Herramientas*.

 **Tarefa 2.6. Personalizar barra de ferramentas.** A tarefa vai consistir en realizar de forma secuencial as seguintes operacións: crear unha nova barra de ferramentas chamada Proba con dous botóns *+Aumentar Zoom e - Disminuir Zoom*, quitar un dos botóns e eliminar a barra Proba.

## Opción Herramientas > Opciones

Permite realizar unha gran variedade de personalizacións tanto xerais como específicas dunha linguaxe tal e como se pode ver na ventá *Opciones*.



Unha mostra das diferentes personalizacións permitidas dende esta ventá pode ser:

- No botón *General* pódese por exemplo elixir o navegador que se vai a usar dende NetBeans e incluso editar as particularidades dese uso.
- No botón *Editor* pódense personalizar os detalles da edición para tódalas linguaxes ou para algunha en particular. Por exemplo en *Formato, Categoría: Tabuladores y sangrados*, pódese cambiar o número de espazos no sangrado ou o número de caracteres por liña para imprimir.
- No botón *Tipos de letra y colores* pódense personalizar as cores de primeiro plano e fondo para a sintaxe, os resaltados, as anotacións, as diferenzas ou o control de versións para tódalas linguaxes ou para só algunas. Por exemplo en *Sintaxis, Idioma: C, Categoría: Number, Fondo: Heredado* pódese cambiar a cor de fondo dos números en C e ver en *Vista previa* como quedaría.
- No botón *Asignación de Teclas* pódese ver a lista de accións posibles e a combinación de teclas asignada por defecto a cada acción se é que existe; tamén se pode editar algunha combinación.
- Hai un botón por cada linguaxe instalada, dende onde se poden personalizar os detalles do desenvolvemento nesa linguaxe. Nalgúns casos, haberá que activar o soporte para esa linguaxe premendo no botón *Activate*.

In order to use this functionality, support for php must be activated.

[Activate](#)

- No botón *Team* pódese personalizar o versionado para traballo en equipo.
- No botón *Aspecto* pódese personalizar os separadores de documentos ou as ventás.

- O botón *Varios* é o caixón de xastre onde poden facerse variadas personalizacións.
- O botón *Exportar* permite gardar a personalización actual nun arquivo zip e o botón *Importar* permite recuperar unha personalización gardada nun arquivo zip de personalización perdendo a personalización actual. Como medida de precaución antes dunha importación recoméndase gardar a personalización actual por se é necesario recorrer a ela.

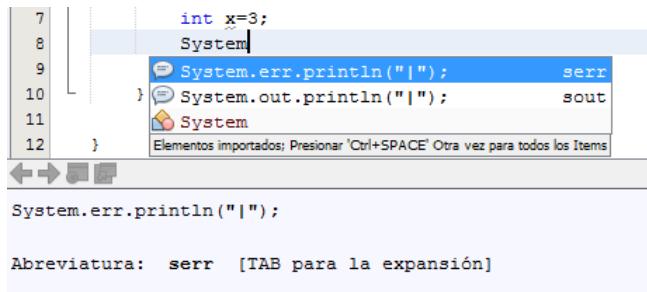
 **Tarefa 2.7. Exportar e importar personalización.** A tarefa vai consistir en realizar de forma secuencial as seguintes operacións: exportar a personalización actual a un arquivo denominado *personaliza1*, definir o navegador que vai a utilizar NetBeans de entre os instalados na máquina, poñer a cor marela como fondo para o código fonte de calquera linguaxe de programación, e finalmente importar a personalización *personaliza1* e comprobar que se fixo correctamente.

## 2.3 Automatizar en NetBeans

NetBeans dispón de mecanismos para programar máis rápido e cómodo como o completado de código, os consellos (*hints*), as macros e os modelos de código (*code templates*). As dúas últimas permiten respectivamente automatizar certos eventos de teclado e rato e asociar un anaco de código a un modelo.

### Completado de código

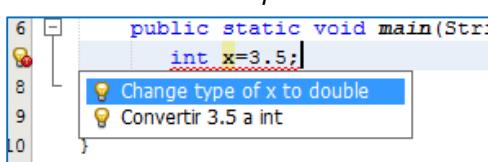
O completado de código depende da linguaxe de programación e da personalización de NetBeans para esa linguaxe. Terase unha axuda automática para completar código se en *Herramientas>Opciones>Editor>Finalización de código (Code completion)* así se reflicte. De todas formas pódese pulsar unha vez Ctrl+barra espazadora para abrir unha caixa con suxestións das que elixir unha coa que completar o código. Poden volver a pulsarse para ter a suxestión más detallada.



 **Tarefa 2.8. Activar completado de código.** Enumera as distintas opcións de completado de código que presenta a configuración de Netbeans.

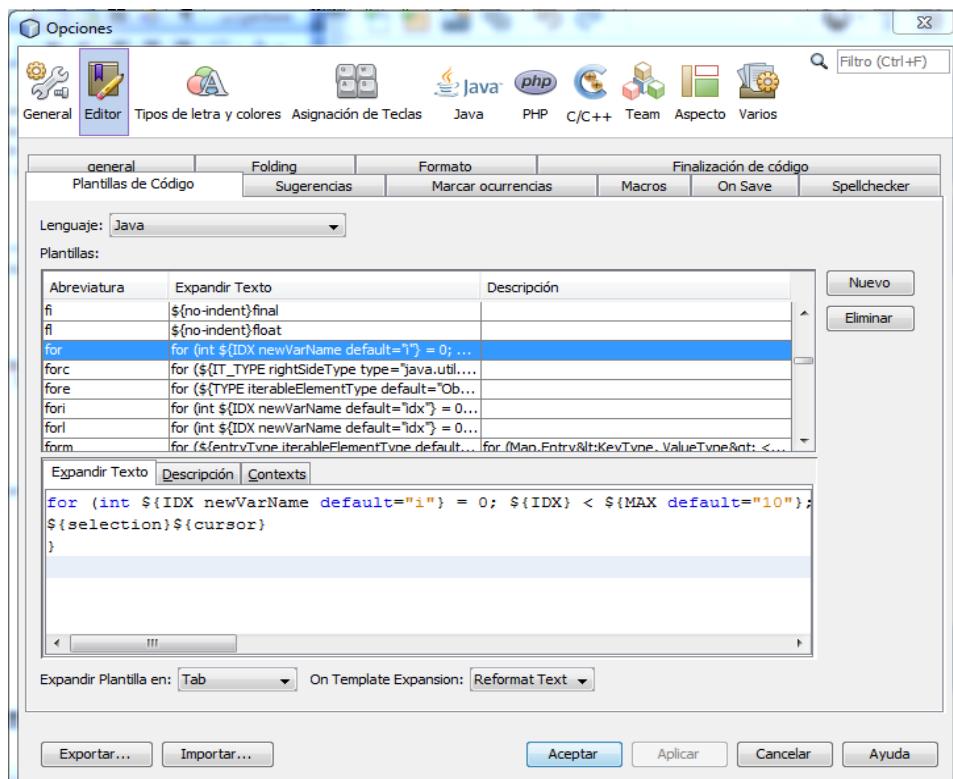
### Suxestións

As suxestións ou consellos ou *hints* vense pulsando Alt+Enter cando aparece un erro , unha advertencia  ou unha combinación delas na columna de numeración de liñas. Poden personalizarse as suxestións en *Herramientas>Opciones>Editor>Sugerencias (Hints)*



## Modelos de código (Plantillas de código)

NetBeans dispón de moitos modelos de código (*Code templates*) que poden verse en **Herramientas>Opciones>Editor>Plantillas de Código**. Nesta pestana pódese elixir a lingua e editar, eliminar e crear novos modelos.



O modelo *for* resaltado na imaxe anterior está definido para poder utilizarse nun sitio do código Java adecuado escribindo *for* e premendo na tecla *[Tab]* (*Expandir Plantilla en: Tab*) para expandirse. A expansión consiste en escribir o texto da pestana *Expandir Texto*, permitir cambiar as variables (\$ seguido de maiúsculas) por defecto que aparecerán en caixas azuis.

Os modelos de código aparecerán identificados co símbolo na lista de opción do completado de código.

**Tarefa 2.9. Modelos de código.** Explica que ocorre se nun programa Java tecleas “for” + *[TAB]*, e tamén se tecleas “sout” + *[TAB]*. Logo crea un modelo de código para HTML chamado js que se expanda con *js* +*[TAB]* que permita inserir as etiquetas de apertura e peche dun script JavaScript dentro do código HTML e deixar o cursor dentro do script. O código para o modelo pode ser: `<script type="text/javascript">${cursor}</script>`. Proba o novo modelo creado e outros como *ol* +*[TAB]* e *tab* +*[TAB]*

## Macros

Unha macro en NetBeans é a gravación dunha secuencia de eventos de rato e teclado cun nome á que se asocia unha combinación de teclas para que poida ser executada en calquera momento. Un procedimiento para a creación da macro realizase durante a edición de código e consiste en:

- Iniciar a gravación da macro coa icona *Iniciar gravación de macro*
- Todas as pulsacións de teclado ou de rato que se fagan a partir deste momento serán gravadas na macro.

- Finalizar a gravación da macro coa icona *Detener grabación de macro* .
- Asociar unha combinación de teclas para a activación da macro.  
Recoméndase editar o código da macro despois de creala (*Herramientas>Opciones>Editor>Macros*) e revisar o seu código para engadir os caracteres que non se tiveron que teclear porque NetBeans completou o código automaticamente durante a creación pero que se necesitan cando se execute a macro.

 **Tarefa 2.10. Crear macro.** A tarefa consiste en crear unha macro asociada á combinación de teclas *[Alt]+i* que actúa sobre unha variable numérica enteira seleccionada e escribe o código `if (variable!=0) {} else {}` deixando unha liña en branco entre as chaves e o cursor ao principio das primeiras. Para probar esta macro pode utilizarase o código fonte Java seguinte:

```
import java.util.Scanner;
public class parimpar {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.println("Teclee o número enteiro:");
        int n = teclado.nextInt();
        if(n%2!=0) {
            System.out.println(n + " é impar");
        }
        else {
            System.out.println (n + " é par");
        }
        System.out.println();
    }
}
```

### Macros vs. Modelos de Código

La macro tiene una funcionalidad parecida a los modelos de código, pero la creamos de otra forma y tiene más potencia.

Básicamente podríamos decir que es una "grabadora" de todas las pulsaciones que hagamos. Cuando empezamos a grabar una macro va guardando todo lo que vamos haciendo hasta que paremos la grabación. A todo eso que hemos hecho le asignamos una combinación de teclas y la podemos invocar cuando queramos.

Imáginate que estás en un programa Java y quieres declarar una variable Scanner. Harías: `Scanner teclado = new Scanner (System.in);` y luego pulsas botón derecho y eliges "Fix imports" para que te incorpore `import java.util.Scanner;` arriba. (Seguro que lo has hecho muchas veces!!)

Pues puedes empezar a grabar una macro, hacer esos pasos, es decir escribir: `Scanner teclado = new Scanner (System.in);` y luego pulsar botón derecho > "Fix imports" y parar la grabación. Le asignas un nombre y una tecla aceleradora, por ejemplo ALT + T

A partir de ese momento en cualquier programa, al pulsar ALT + T te hará lo grabado, tanto escribir `Scanner.... como lo del botón derecho "Fix imports". Ahorran mucho tiempo!!`

En cuanto a la forma de grabarlas, lo más fácil es pulsando el botón rojo que aparece en la barra de menú de la pestaña del programa, y para pararla con el cuadrado gris que hay al lado. También se puede mediante el menú superior *Edit > Start macro recording*. La principal diferencia entre la grabación de macros y los modelos de código es que las macros graban también cosas que hagamos con el ratón, por ejemplo por los menús. Los modelos de código solo valen para abreviar texto de nuestro código. En el ejemplo anterior, en un modelo de código no podría reflejar lo de botón derecho > Fix Imports.

## 2.4 Xestionar complementos de NetBeans

Un complemento ou *plugin* está formado por un grupo de módulos (arquivos con extensión *nbm*) dependentes; pode ocorrer que algún módulo sexa utilizado por varios complementos. En NetBeans os complementos están agrupados formando unha característica. Estas conexións entre módulos, complementos e características fan que a xestión sobre un deles poida afectar aos demais; por exemplo ao actualizar un complemento pode actualizarse algún módulo que tamén se utiliza noutro complemento e por tanto este último está afectado pero NetBeans irá avisando destas conexións.

Todas as operacións de xestión de complementos fanse dende a ventá *Plugins* á que se accede dende a opción *Herramientas>Plugins* do menú principal de NetBeans. Esta ventá ten as pestanas *Plugins Disponible*, *Descargados* e *Instalados* dende as que se pode instalar, desinstalar, activar e desactivar complementos.

A pestana **Plugins Disponibles** (*número deles*) mostra os complementos dispoñibles nas url existentes no centro de actualizacións que non están instalados e que poderían instalarse.

A pestana *Instalados* (*número deles*) mostra as características instaladas. De seleccionar *Mostrar Detalles* mostraránse os complementos instalados. Nesta pestana pódese desinstalar, activar ou desactivar características ou módulos.

A pestana *Descargados* (*número deles*) mostra os complementos descargados e non instalados. As operacións de activación e desactivación só poden facerse despois de estar instalado un complemento. Para minimizar o tempo de inicio do contorno e aforrar memoria pódense desactivar complementos que non se necesiten e activalos máis tarde cando se necesiten sen necesidade de volver a instalalos.

A operación de desinstalación faise cando un complemento non se vaia a utilizar máis. Se nalgún momento se volvera a necesitar, habería que volver a instalalo.

NetBeans utiliza uns pequenas imaxes para representar graficamente o estado do complemento que son:



Complemento instalado e activado



Complemento que necesita que o IDE se volva a iniciar



ou Complemento instalado e desactivado

### Instalar

Se o complemento a instalar está dispoñible nas *URL* fonte, hai que seleccionalo na pestana *Plugins Disponibles* e premer no botón *Instalar*. Na ventá *Plugins Disponibles* poden aparecer moitos complementos e pode ser difícil dar co que se desea polo que se recomenda utilizar a caixa de texto *Buscar* ou ordenar alfabeticamente os complementos facendo clic na cabeceira da columna *Nombre* para localizar rapidamente un módulo. Pode observarse que cando se fai clicriba dun nome de complemento aparece na parte dereita da ventá información detallada sobre el.



**Tarefa 2.11. Instalar complemento dispoñible no IDE: Rainbow braces.** Este complemento, unha vez instalado, dálle cor aos corchetes, parénteses e chaves, de forma que é más sinxelo identificar os elementos de cerre cos correspondentes de apertura.

Se o complemento a instalar non está dispoñible nas URL fonte, hai que descargar o complemento de onde sexa necesario, ir á pestana *Descargados* e engadir o arquivo *nbm* para que sexa descargado. Cando finalice a descarga haberá que seleccionalo na ventá *Descargados* e premer no botón *Instalar*.

 **Tarefa 2.12. Desde a páxina de plugins de Netbeans, fai una lista cos más descargados, explicando a súa funcionalidade.**

 **Tarefa 2.13. Instala un complemento non dispoñible no IDE:** Descarga do portal de plugins o plugin NB SpringBoot e, unha vez instalado, na opción de crear novo proxecto (Java > Maven) ofrece a posibilidade de crear proxectos SpringBoot.

### Activar/Desactivar

A operación de activar ou desactivar un complemento faise na ventá *Instalados* e consiste en seleccionar o complemento e premer no botón correspondente, é dicir, se o complemento está activado aparece o botón *Desactivar* e se o complemento está desactivado aparece o botón *Activar*. Hai que premer en "Ver detalles" para velos un a un.

### Desinstalar

Para desinstalar un módulo hai que ir á pestana *Instalados*, seleccionar o complemento a desinstalar e premer no botón *Desinstalar*.

## 2.5 Extensións de Visual Studio Code

Do mesmo xeito que en Netbeans falamos dos plugins, en VS falamos das extensións, xa que en ambos casos son complementos que dotan de nova funcionalidade ao IDE.

Podemos destacar extensións de **idioma** (como o Spanish Language Pack), extensións de **soporte de distintas linguaxes de programación** (como o Java Extension Pack xa visto) ou extensións que **melloran a produtividade**.

Xa describimos o proceso na instalación do Java Extension Pack. Agora debemos investigar cales son as que precisamos para o noso traballo.

 **Tarefa 2.14. Instalar extensión en Visual Studio.** A tarefa consiste en instalar estas dúas extensións e comprobar o seu funcionamento:

- Bracket Pair Colorizer: asigna diferentes cores a cada chave de apertura e peche {} para ver mellor como van emparelladas.
- Live Server: Abre un navegador cun servidor web interno para visualizar as páxinas estáticas (html) e ten a interesante función de autocarga, de xeito que se facemos cambios na páxina, ao gardala, xa se actualiza automaticamente no navegador.

## 2.6 Plugins de IntelliJ Idea

Como no resto dos IDEs, os plugins ou extensións engaden funcionalidades interesantes o soporte para outras linguaxes de programación. Para instalalos, menú superior *File > Settings > Plugins*.

Lista de plugins interesantes:

- Lombok: soporte desta da libraría Java do mesmo nome.
- GitToolBox : optimiza o traballo con Git e os repositorios remotos.
- Copilot : soporte para a IA de axuda á programación de Github. O plugin é gratuito pero a funcionalidade de Copilot só é gratuita para estudiantes e profesores.
- Atom Material Icons: cambia as iconas dos arquivos na estrutura do proxecto.

### 3. Uso básico do contorno de desenvolvemento en operacións de edición e xeración de código

#### 3.1 Introdución

As operacións básicas realizadas nesta actividade serán as de edición, compilación, enlace e execución de código en distintas linguaxes. As aplicacións resultantes en todos os casos serán aplicacións de consola sinxelas para non confundir con demasiado código ao alumnado cando está iniciándose na programación informática.

#### 3.2 Proxecto C/C++ en Netbeans

##### Instalación do compilador C/C++ e configuración de NetBeans

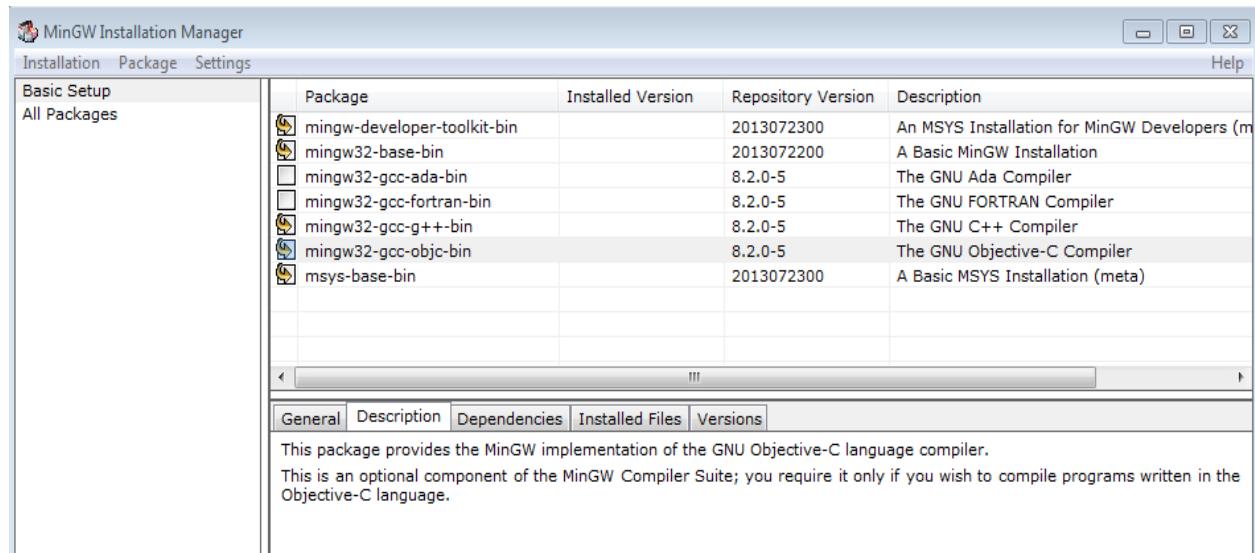
Para que se poida editar, compilar, enlazar e executar código C++ en NetBeans é preciso cumplir tres condicións:

- Empregar unha versión de Netbeans 21 ou inferior.
- Empregar a versión 14 do JDK ou inferior.
- Ter instalado un compilador C/C++ e NetBeans estea configurado para poder utilizalo.

Os pasos a seguir serán:

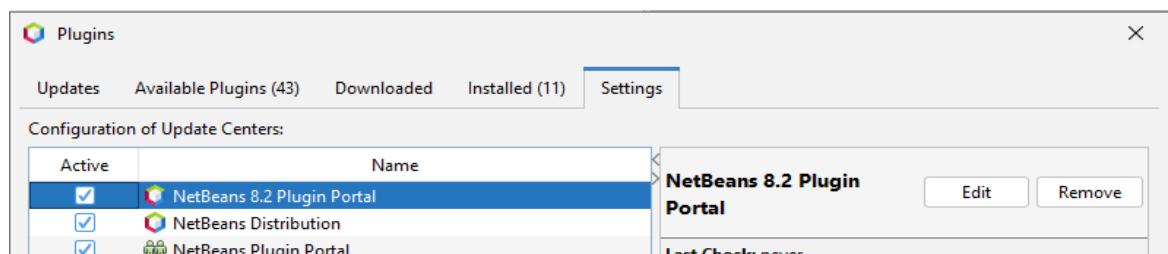
1.- Descargar o compilador: Podemos descargalo desde a súa páxina oficial en sourceforge: <https://sourceforge.net/projects/mingw/files/Installer/>

2.- Instalar o compilador. É sinxelo, so hai que seguir os pasos. Ao rematar hai que marcar os paquetes que se mostran na seguinte figura, e engadilos desde o menú *Installation > Apply changes*.



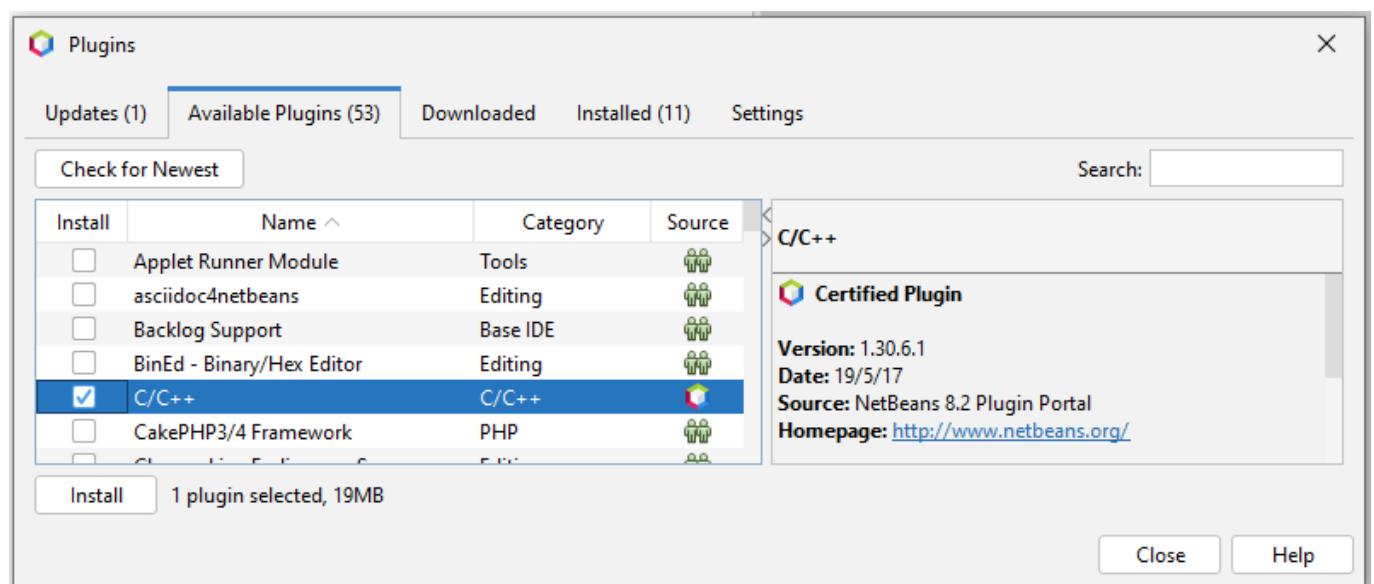
Debemos ter na variable PATH do sistema a ruta dos executables instalados, por defecto C:\MinGW\bin.

3.-Xa en Netbeans, hai que ir ao menú superior *Tools > Plugins >> Settings* e marcar como activo o portal de plugins *Netbeans 8.2. Plugin Portal*.

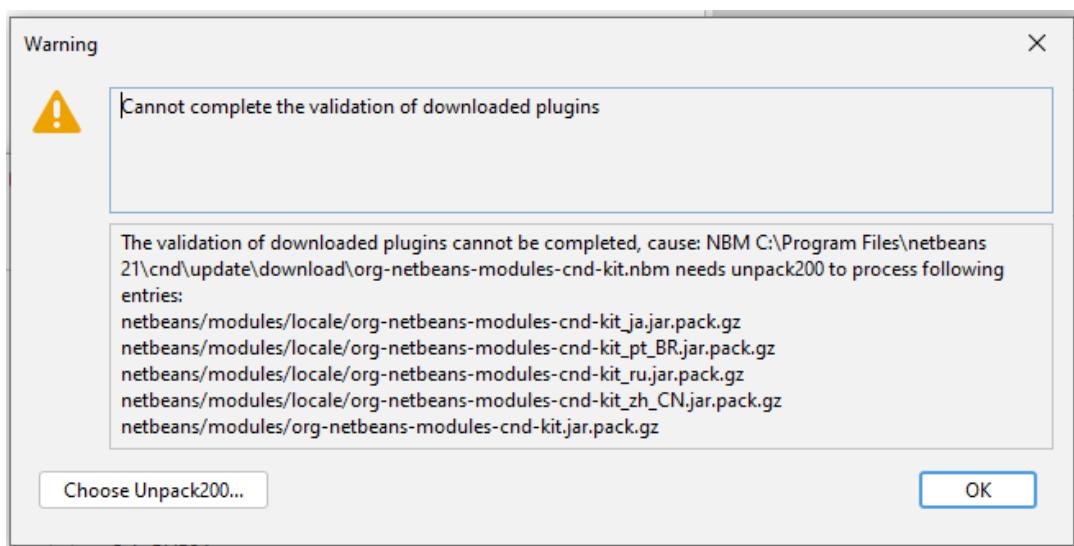


Hai que destacar que na versión 22 de Netbeans neste momento (agosto 2024) non é posible facer este paso, polo que non é posible instalar o plugin de C/C++

4.- En *Available Plugins*, prememos en [*Check for Newest*] para que incorpore os plugins do portal activado no punto anterior. Seleccionamos "C/C++" e prememos [*Install*] e seguimos o proceso de instalación aceptando os termos de licenza.

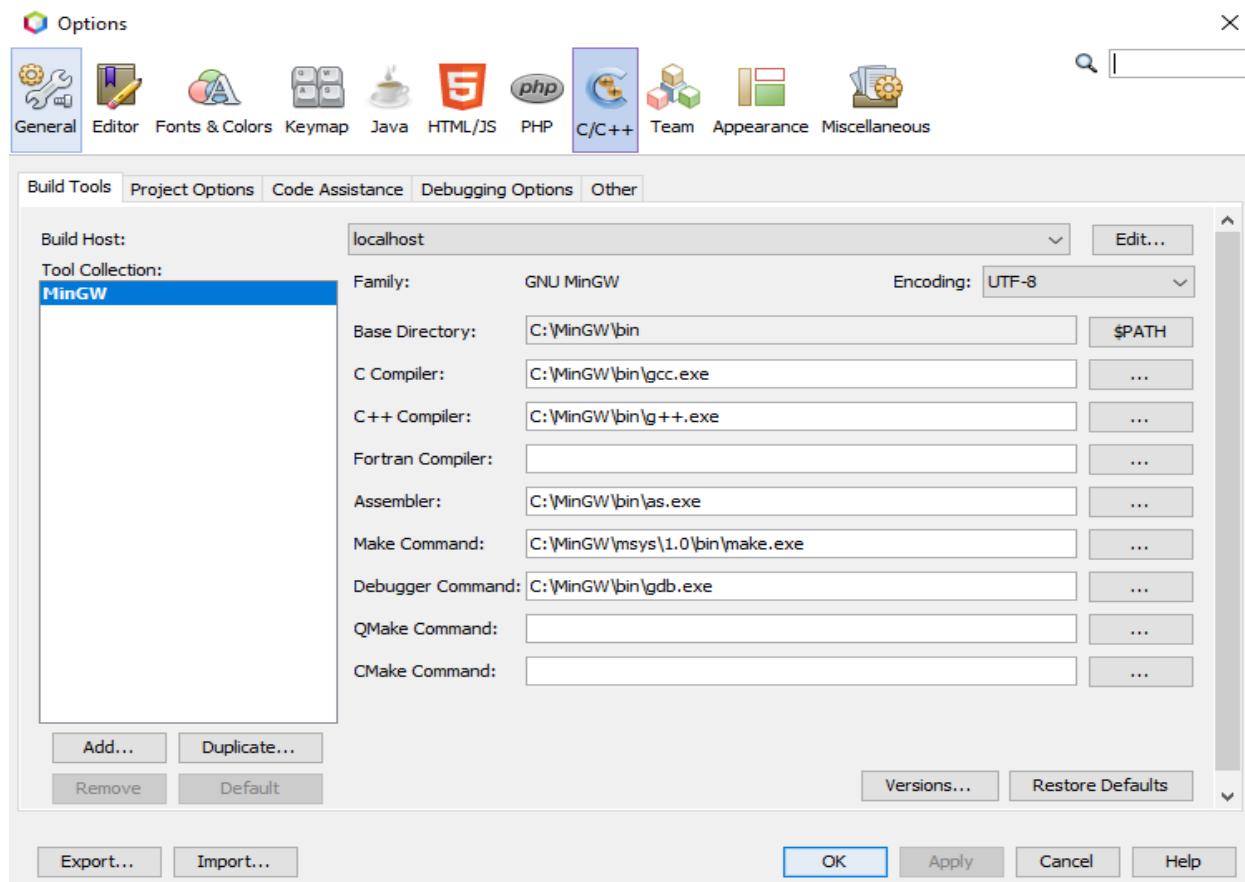


Se estamos traballando cunha versión de jdk superior á versión 14, aparece un *warning* por problemas na validación do plugin.



Para continuar premeríamos [Choose Unpack200] e pídenos que seleccionamos o arquivo *unpack200.exe* do cartafol */bin* de Java. O problema é que este arquivo so está dispoñible ata a versión 14 do jdk. Debemos entón obter este arquivo desde unha versión anterior e copialo no cartafol */bin*. En resumo, mellor traballar cunha versión do jdk 14 ou menor como comentamos ao comezo deste apartado.

5.- No menú superior *Ferramentas > Opcións >* na pestana **[C/C++]** verificamos ou engadimos a ruta do compilador de C. So temos que indicar a ruta base, o resto de parámetros cóbreñense automaticamente.



Se non cubre o campo C++ compiler é debido á instalación de minGw, que non se instalou o paquete: mingw32-gcc-g++-bin. Hai que reinstalar minGw.

## Desenvolvemento do proxecto C/C++ en NetBeans

A xeración de código C++ en NetBeans farase utilizando un proxecto C++, é dicir, un conxunto de carpetas e arquivos xestionados por NetBeans que facilita e automatiza as operacións a realizar sobre o código. O proxecto de NetBeans non é o único procedemento para realizar as operacións básicas xa que a edición podería facerse cun editor de texto calquera, a compilación e enlace poderían facerse utilizando a liña de comandos, o compilador e a ferramenta make e a execución podería facerse directamente no sistema operativo, pero a utilización do proxecto é más cómoda e permite realizar máis operacións que as básicas.



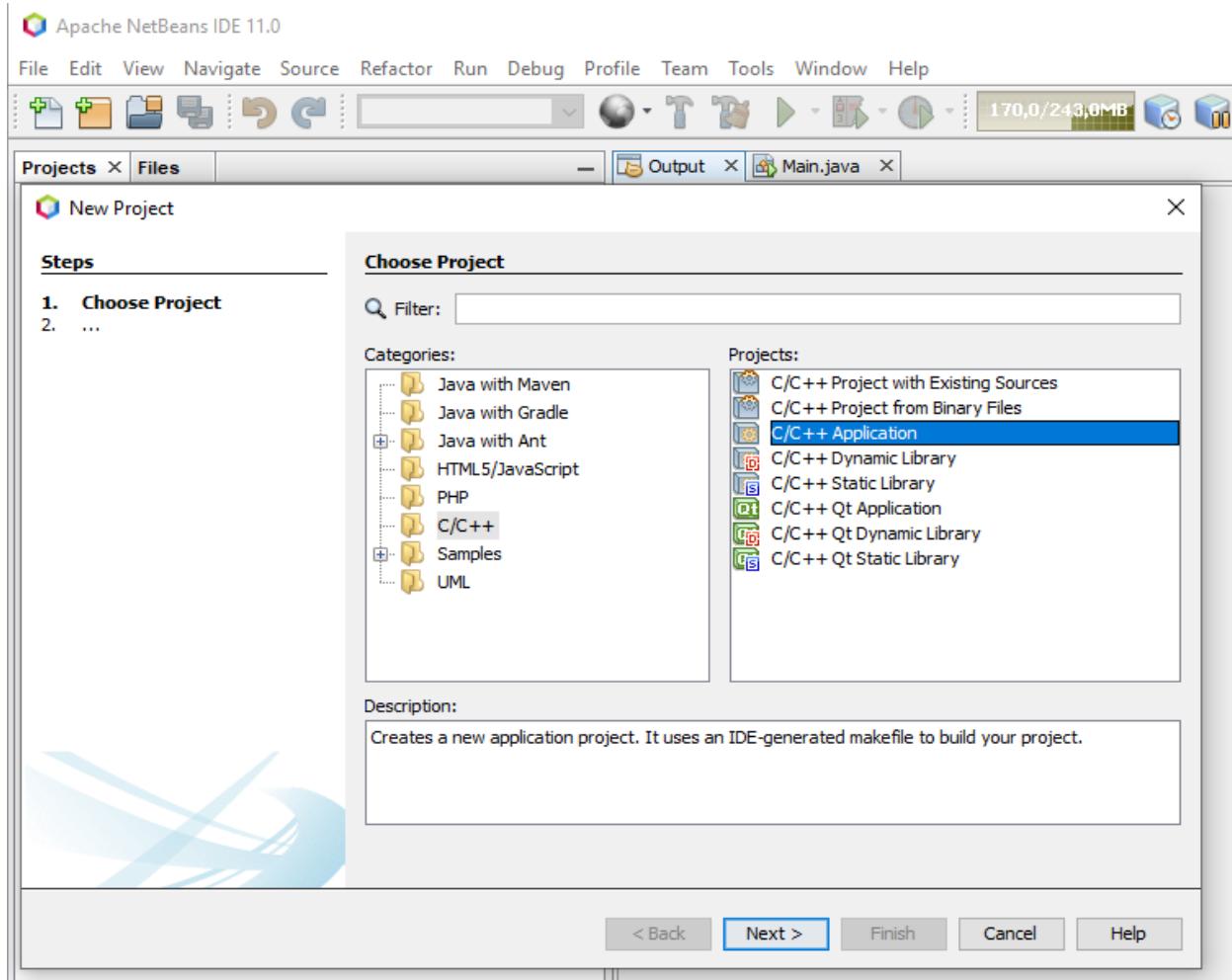
## Tarefa 2.15. Instalar compilador C/C++ e configurar NetBeans para que poida crear proxecto en C++.

Descargar MinGW e instalalo en c:\ seguindo as instrucións descritas. Lembra engadir á variable de entorno PATH a ubicación do cartafol "bin" do compilador.

En Netbeans realizar os pasos indicados previamente para a instalación e configuración do plugin.

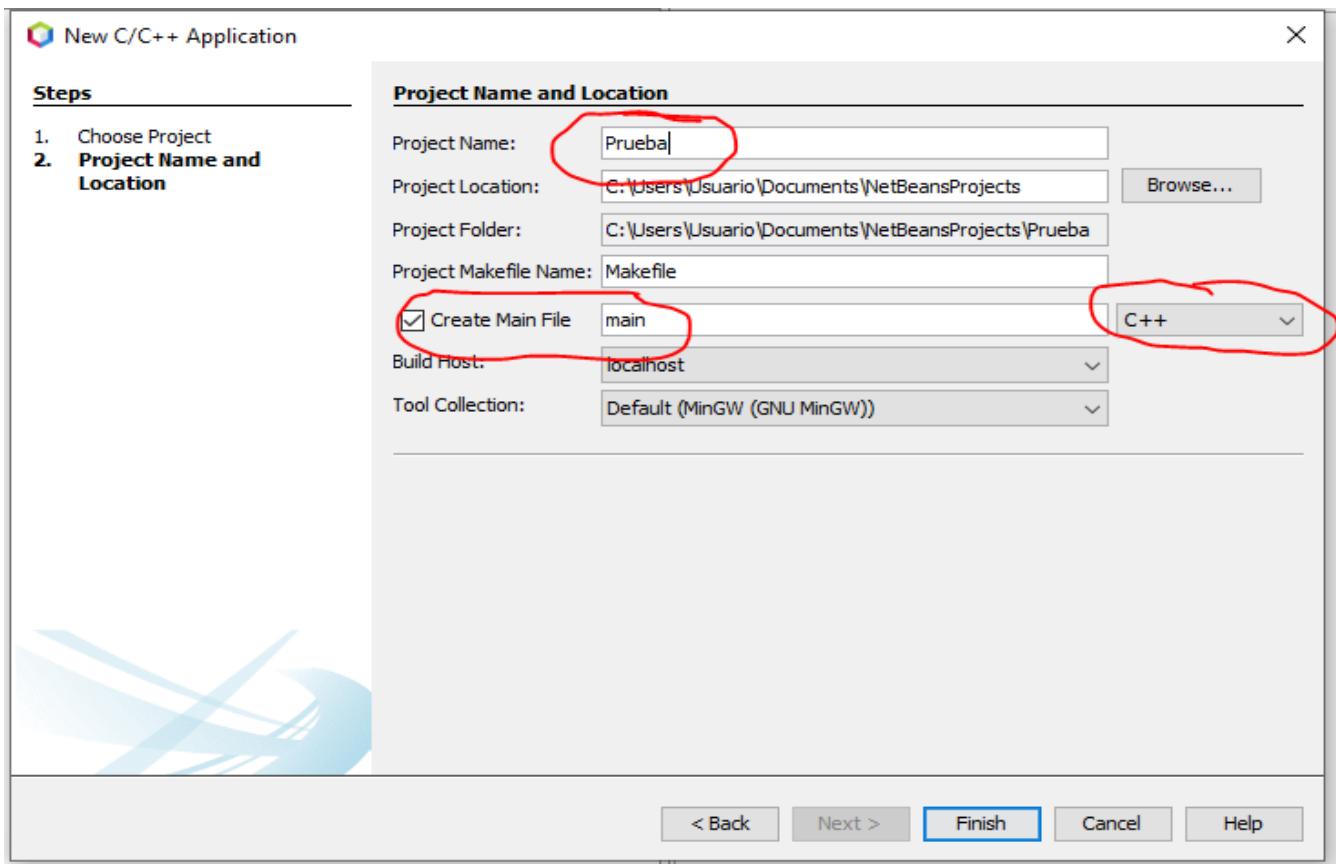
### Crear proxecto

- Para comezar un proxecto, menú File > New Project > Grupo C/C++ > C/C++ Application.

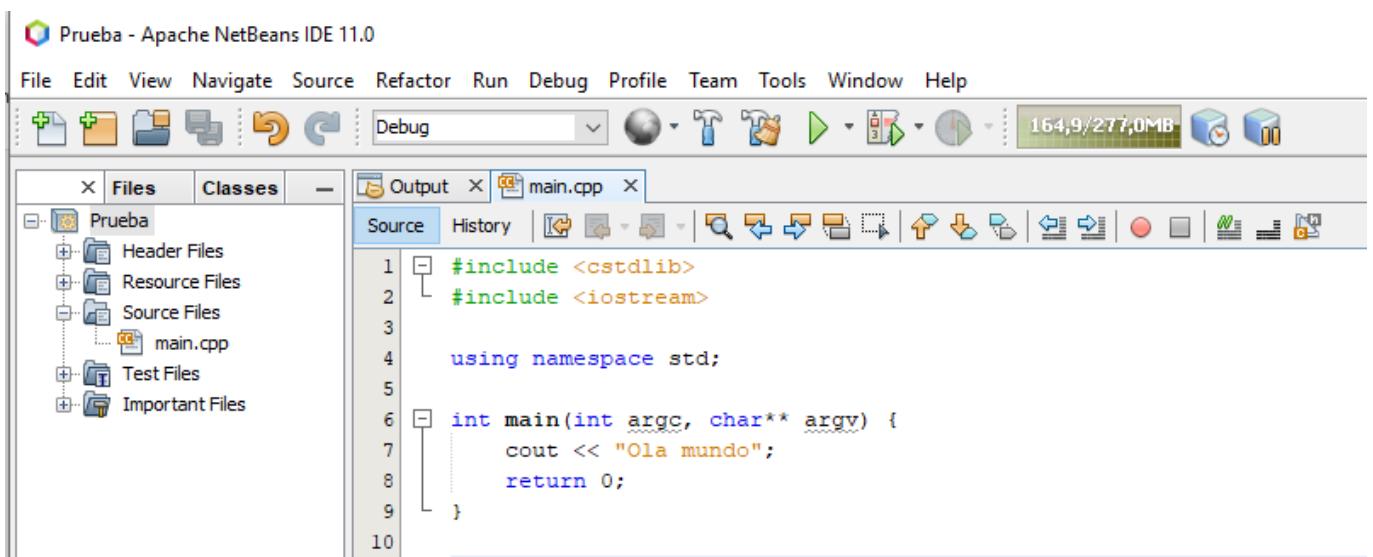


Pode observarse que tamén é posible definir un proxecto para crear unha biblioteca estática ou dinámica ou con arquivos fontes ou binarios xa existentes.

- Dáse nome ao proxecto, elíxese unha localización e a linguaxe: C (extensión c) ou C++ (extensión cpp). Pódese indicar tamén se deseja crear o arquivo principal de forma automática có código por defecto que teña NetBeans para esa linguaxe.



- Pode verse na ventá *Proyectos* a vista do proxecto e na ventá *Archivos* a estrutura de carpetas do proxecto. En calquera das dúas ventás pode editarse o arquivo principal facendo dobre clic sobre o seu nome, se é que existe.



- Poden verse ou modificarse as características do proxecto iluminando o nome do proxecto na ventá *Proyecto*, clic dereito, *Propiedades*.

 **Tarefa 2.16 Crear proxecto C++ en NetBeans** de nome *sumatecleados* que dará lugar a unha aplicación de consola sinxela. A Tarefa consiste en editar o arquivo *sumatecleados.cpp* para obter un código que permita sumar números que se van tecleando na ventá de comandos ata que se teclea o número 0; momento no que se visualiza a suma e finaliza o programa.

Este pode ser o código da actividade:

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    float i;
    float suma = 0;
    cout << "Sumar números tecleados ata que se teclee 0\n";
    do {
        cout << "Teclea un número (Finalizar con 0)...";
        cin >> i;
        suma = suma + i;
    } while (i);
    cout << "Suma=" << suma << "\n";
    return 0;
}
```

## Compilar, enlazar e executar

A forma más rápida de compilar, enlazar e executar é premer F6 ou facer clic en  na barra de ferramentas; se hai varios proxectos abertos, executarase o proxecto principal.

Ábrese a ventá *Salida* coa pestana de construcción e execución (*Build, Run*) na que se irán executando as operacións definidas no arquivo *makefile* que NetBeans crea automaticamente coa información do proxecto e a ferramenta *make* e que teñen como obxectivo a creación do executable.

Se hai errores no proceso de compilación ou enlace, non se xerará o executable e a información sobre eles aparecen na pestana de construcción e execución.

Se non hai errores no proceso de compilación ou enlace, xeraranse os arquivos obxecto resultantes da compilación con extensión .o e o arquivo executable resultante do enlace con extensión .exe. Estes arquivos poden verse na ventá *Archivos* despregando as carpetas *build* e *dist* respectivamente. Se o proxecto é unha aplicación de consola, verase o resultado da execución na pestana execución (*Run*) da ventá *Salida* como se fose a ventá de comandos do sistema operativo.

A opción *Ejecutar>Generar Project* do menú principal (ou a pulsación da tecla F11) permite realizar a compilación e o enlace pero non permite executar.

A execución do arquivo resultante tamén pode facerse:

- Dende o sistema operativo facendo doble clic sobre o executable. Se a aplicación é de consola, abrirase a ventá de comandos ou consola do sistema e executaranse as instruccións do código fonte; cando a execución finalice, pecharase a consola.
- Dende a ventá de comandos. Hai que acceder á ventá de comandos do sistema, moverse pola árbore de directorios co comando *cd* ata chegar á localización do executable e escribir o seu nome.

Se ao executar o proxecto en Netbeans, non nos deixa introducir valores por consola hai que ir ás propiedades do proxecto > Run > Console Type :Standar output



**Tarefa 2.17 Compilar, enlazar e executar o proxecto C++ en NetBeans.** Executar tamén dende fóra de NetBeans. A tarefa consiste en compilar o arquivo fonte *sumatecleados.cpp* da tarefa anterior para obter o arquivo obxecto *sumatecleados.o*, e enlazalo para xerar o executable *sumatecleados.exe*. Este último arquivo executarase dende NetBeans e dende o sistema operativo.

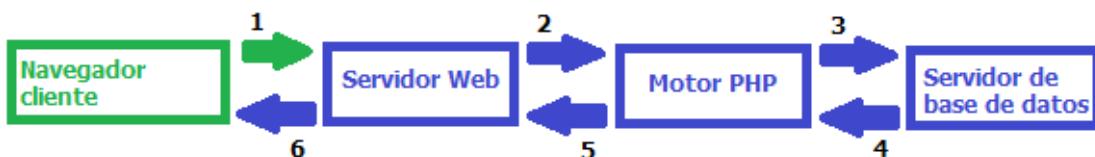
### 3.3 Proxecto PHP en NetBeans

Unha páxina web dinámica que teña scripts PHP ten que estar aloxada nun servidor web que teña intérprete de PHP e servidor de base de datos se é que o código PHP manexa bases de datos.

Un proxecto PHP en NetBeans permitirá editar o código fonte, detectar erros de sintaxe e executalo. Polo tanto necesítase un servidor web que cumpla os requisitos anteriores e necesítase que NetBeans poida executar o proxecto utilizando ese servidor.

Durante o proceso de desenvolvemento do sitio web PHP é conveniente ter instalado un servidor Web local sobre o que se farán as probas do código. Cando o sitio estea rematado e probado, pode facerse a publicación nun servidor Web adecuado en Internet.

Graficamente o proceso de transformación do código é o seguinte:



- Paso 1: O navegador dun usuario envía una petición http de una páxina web determinada a un servidor web.
- Paso 2: O servidor web recibe a petición e vai procesando as liñas de códigos que se necesite executar (non teñen por que ser todas nin teñen que procesarse de forma secuencial). Se detecta código PHP, envíao ao motor PHP para que o procese e espera os resultados.
- Paso 3: O motor PHP analiza a secuencia de instrucións recibidas, e se hai algunha petición de conexión cunha base de datos, abre una conexión co servidor de bases de datos, envíalle as instrucións SQL e espera os resultados.
- Paso 4: O servidor de bases de datos recibe as instrucións, procésalas, e envía os resultados ao motor de PHP, ou un código de error, se é que non pudo realizar a operación.
- Paso 5: O motor de PHP termina de executar as instrucións e devolve os resultados ao servidor web.
- Paso 6: O servidor web devolve o código HTML ao navegador do usuario que fixo a petición.

#### Servidor local

O entorno máis popular para executar PHP está composto por un servidor Web Apache, servidor de base de datos MySQL e intérprete de PHP. Existen varios programas libres baixo licenza GNU GPL que permiten instalar un entorno deste tipo como por exemplo:

- XAMPP (X=calquera sistema operativo, A=Apache, M=MySQL, P=PHP, P=Perl).
- WAMP, (W=Windows, A=Apache, M=MySQL, P=PHP).
- LAMP igual que WAMP pero para Linux.
- MAMP igual que WAMP pero para Mac OS X.

Existe unha versión "Lite" que dispón dunha instalación portable, más sinxela q que permite instalar o servidor nun dispositivo extraíble e executalo en distintos equipos.

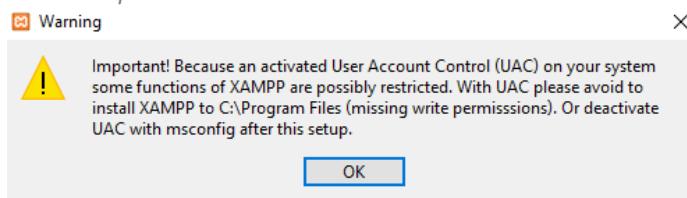


## Tarefa 2.18. Instalar un servidor XAMPP en local.

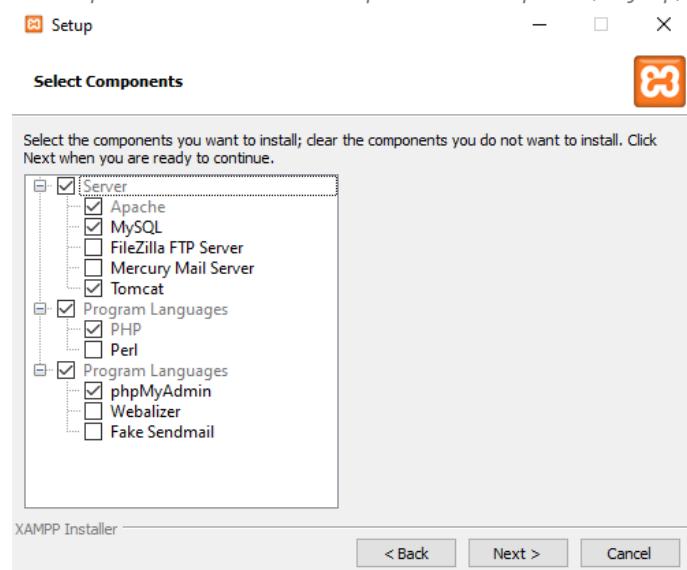
- Podemos descargar XAMPP en 32bit ou 64bits e para distintos sistemas operativos desde <https://www.apachefriends.org/es/download.html>, pero no noso caso, para simplificar descargamos a versión "Lite" na súa variante portable desde <https://sourceforge.net/projects/xampplite/>. Descomprimimos o zip no mesmo cartafol de descargas e movemos o contido do cartafol "App" a un cartafol novo ao que chamaremos "xampp" na raíz dun dispositivo extraíble, sen ningunha instalación adicional.

DATA (D:) > xampp >				
		Ordenar	Ver	...
Nombre		Fecha de modificación	Tipo	Tamaño
apache		15/08/2024 17:44	Carpeta de archivos	
App		15/08/2024 17:52	Carpeta de archivos	
mysql		15/08/2024 17:45	Carpeta de archivos	
php		15/08/2024 17:52	Carpeta de archivos	
phpMyAdmin		15/08/2024 17:45	Carpeta de archivos	
tmp		15/08/2024 17:54	Carpeta de archivos	
www		15/08/2024 17:45	Carpeta de archivos	
portable.lock		15/08/2024 17:28	Archivo LOCK	0 KB
readme_en		15/08/2024 17:28	Documento de te...	1 KB
xampp-control		15/08/2024 17:28	Aplicación	4.351 KB
xampp-control		15/08/2024 17:56	Opciones de confi...	1 KB

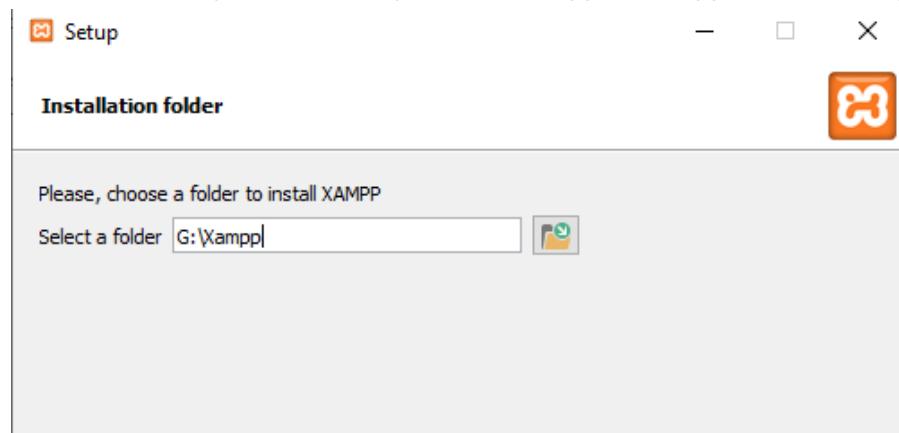
Se houberamos elixido a versión estándar (non Lite) seguiríamos os pasos de instalación. Primeiro infórmanos que se o instalamos sobre o disco c: podemos ter problemas de permisos. Podes instalalo mellor sobre outro disco ou nun pendrive.



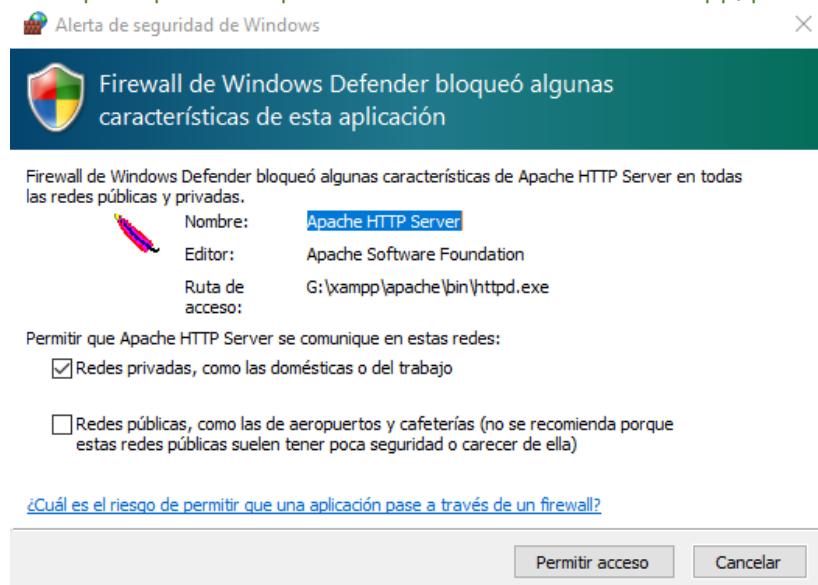
Logo seleccionaríamos os componentes a instalar. So precisamos Apache, MySQL, PHP e phpMyAdmin.



E o cartafol de instalación (ten que ser un cartafol raíz, c:\xampp, d:\xampp, etc. e mellor que non sexa c:)

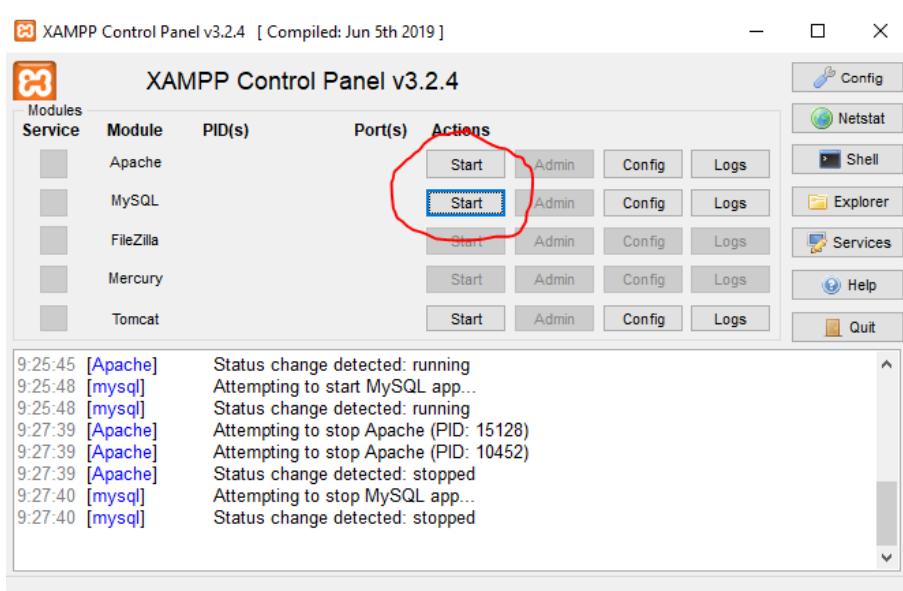


- O firewall de Windows pode pedirnos que confirmemos o acceso de Xampp, permitimos o acceso.



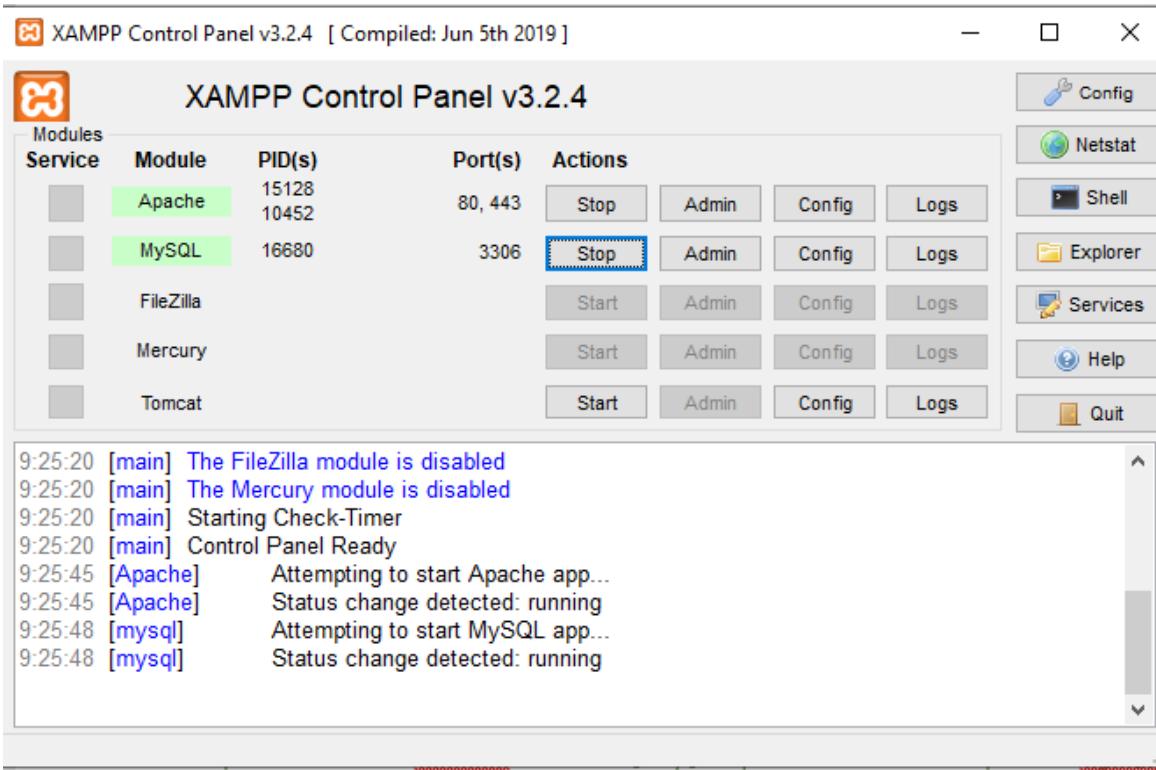
- Para arrincar Xampp, prememos en: `xampp-control.exe`.

- Iniciamos/Apagamos Apache e MySQL premendo no botón [Start/Stop].



- Se non arranca poder ser que teñas o porto ocupado con VMWare, Skype ou outra aplicación de rede. Se todo foi ven, os botóns de [Start] agora mostrarán [Stop] e indicará o porto no que está correndo.

Podemos cambiar os portos nos que se executa Apache: para o porto 80 modificando o arquivo: \xampp\apache\conf\httpd.conf, e para o porto 443 modificando o arquivo: \xampp\apache\conf\extra\httpd-ssl.conf. Ambos na liña que pon "Listen" e o número de porto. Se os cambiamos debemos reiniciar Apache.

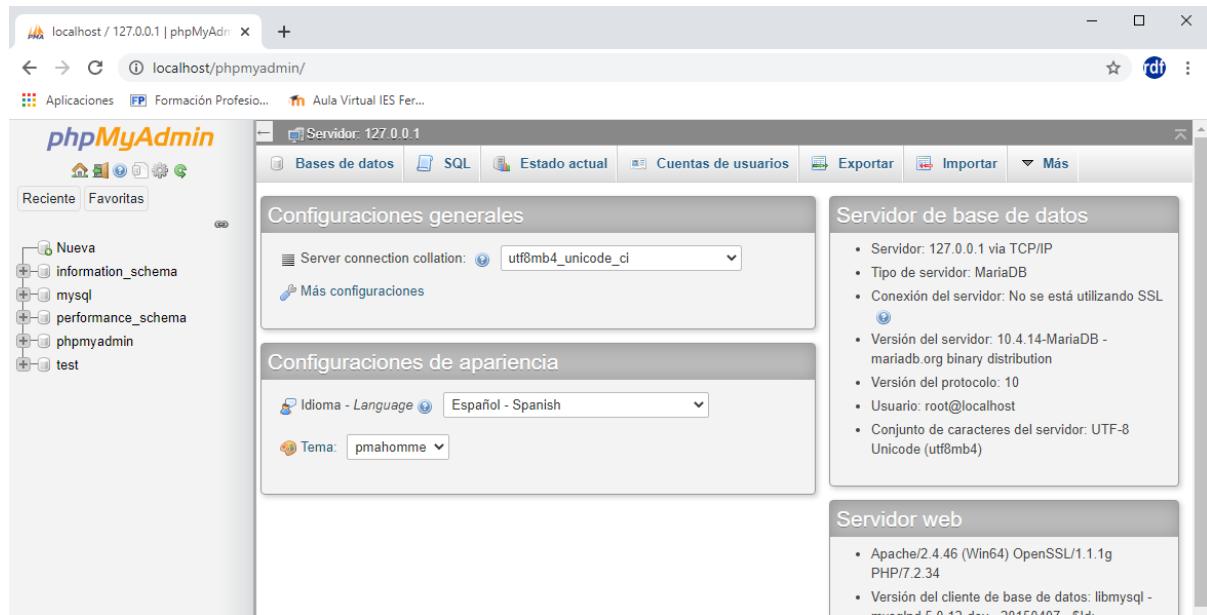


- Na propia máquina podemos abrir nun navegador: <http://localhost> para comprobar que funciona.

The screenshot shows a web browser window titled 'Welcome to XAMPP'. The address bar shows 'localhost/dashboard/'. The page content includes the XAMPP logo and the text 'XAMPP Apache + MariaDB + PHP + Perl'. Below this is a heading 'Welcome to XAMPP for Windows 7.2.34'. A paragraph of text follows, stating: 'You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.' At the bottom of the page, there is a note about security: 'XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the FAQ to learn how to protect your site. Alternatively you can use WAMP.'

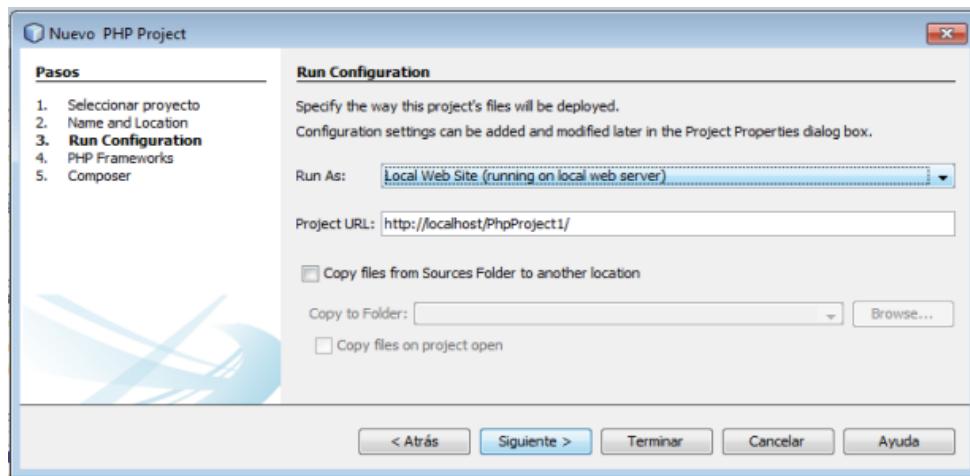
- Os archivos de configuración son /apache/conf/httpd.conf /php/php.ini e mysql/my.ini para Apache, PHP e Mysql respectivamente.
- Investiga para que serven os parámetros de httpd.conf : Listen, DocumentRoot, DirectoryIndex

- Investiga para que serven os parámetros do *php.ini*: *max\_execution\_time*, *memory\_limit*, *display\_errors*.
- Para parar o servidor, prememos os *botóns [Stop]* no panel de control, en Apache e MySQL e *[Quit]*. Ollo: o botón de peche de ventá non pecha a aplicación, minimizaa na area de notificacións.
- O cartafol onde depositaremos os documentos php,html, etc. é *www* (nalgúnha instalación este cartafol pode chamarse *htdocs* ou *public\_html* ou similar).
- Ademais do servidor, dispoñemos da ferramenta de xestión da BD: <http://localhost/phpmyadmin>



## Crear proxecto

A creación dun proxecto PHP é similar á dun proxecto C/C++ ou Java coas diferenzas derivadas da propia linguaxe. Por exemplo existe o paso *Run Configuration* no que se indican as condicións para a execución do proxecto.



A execución pode facerse de varias maneiras pero só nos centraremos na utilización dun servidor web local, polo que indicarase a url para acceder ao servidor local (<http://localhost>) e ao arquivo principal que se vai a executar (dentro da carpeta *www* do servidor). Algo moi útil na fase de desenvolvemento do código PHP é indicar que NetBeans faga unha copia dos arquivos fontes ao servidor web local cada vez que se salven modificacións nos arquivos fontes do proxecto. A continuación pode premerse no botón *Terminar* xa que as opcións restantes superan o nivel básico de xeración de código en NetBeans obxectivo desta actividade.

Aparece o arquivo *index.php* na ventá *Archivos* e na ventá *Proyectos* e editado na pestana de edición co código por defecto que ten NetBeans para PHP e que consiste nunha páxina HTML cun script PHP incrustado no corpo da páxina.

As propiedades do proxecto poden cambiarse igual que nos proxectos C++ ou Java. Por exemplo, se se desea cambiar o nome a *index.php* e que ese sexa o arquivo principal habería que ir ás propiedades do proxecto e no paso *Run Configuration* cambiar o *Index File*.

## Editar código fonte

A edición ten as mesmas características que nun proxecto C/C++ ou Java coa diferenza de que ademais de escribir código PHP normalmente tamén se escribe código HTML e CSS para dar estilo ao código HTML.

## Executar proxecto PHP

A forma máis rápida é executar o proxecto principal dende NetBeans facendo clic en  na barra de ferramentas. NetBeans abrirá o navegador cliente por defecto coas características indicadas na ventá *Run Configuration*.

O programador ten que asegurarse de que o código fonte PHP estea na localización adecuada dentro da carpeta *www* (ou *htdocs*) colocándoo manualmente ou configurando o proxecto para que NetBeans o faga automaticamente. Tamén se ten que asegurar de que as bases de datos necesarias estean funcionando correctamente.



**Tarefa 2.19. Crear proxecto PHP e configuralo para que se poida executar dende NetBeans utilizando o servidor web local instalado na tarefa anterior.** (Ver anexo ao final do tema).

- Hai que descomprimir a aplicación PHP entregada polo profesor no cartafol de documentos do servidor Apache (*www* ou *htdocs*) e seguir as instrucións para, desde *phpmyadmin*, crear a base de datos e táboas necesarias en MySQL.
- Comprobar que funciona correctamente executándoa desde o navegador.
- En Netbeans, crea un novo proxecto en PHP (fontes existentes) indicando a ruta onde descargaches a aplicación e execútala.



**Tarefa 2.20. Editar proxecto PHP en NetBeans.** (Ver anexo ao final do tema).

- Modifica o contido e a aparencia da aplicación incorporando melloras no HMTL e CSS.
- **(Opcional)** Realiza os cambios precisos no código PHP da aplicación entregada para que o contrasinal almacenado no ficheiro estea encriptado.
- **(Opcional)** Realiza os cambios precisos no código PHP para que o usuario poda cambiar o contrasinal.
- **(Opcional)** Crea código JavaScript para que, cando o usuario introduza o novo contrasinal dúas veces, ambos sexan iguais e teñan unha lonxitude mínima de 4 caracteres.

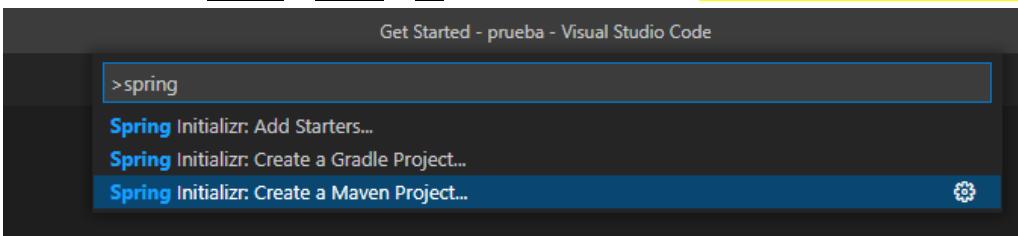
## 3.4 Proxecto Spring en Visual Studio Code

Para a edición e execución de proxectos baixo Spring Framework (co módulo SpringBoot) desde Visual Studio Code e preciso facer a configuración necesaria para programar en Java, como xa se comentou ao comezo deste capítulo, e instalar unha extensión adicional. Estes serían os pasos:

- a) Instalar o JDK como xa se describiu, isto é, descargar o zip da páxina de Oracle, e descomprimilo, por exemplo, en c:/archivos de programa/java.
- b) Configurar Visual Studio Code para vincular o JDK instalado:
  - Abrimos a paleta de comandos: Menú View > *Command Palette* (Cntrl + Shift + P) e escribimos: *Preferences > Open Settings (JSON)*
  - No arquivo que aparece escribimos a ubicación do jdk. Por exemplo:  
**"java.jdt.ls.java.home" : "C:\\Archivos de Programa\\Java\\jdk-21.0.1"**
- c) Dirixímonos ao xestor de extensións mediante o botón lateral  ou ben premendo *Cntrl+Maiusc+X*, buscamos **"Java Extension Pack"** e instalámolo se non está instalada. Facemos o mesmo coa extensión: **"SpringBoot Extension Pack"**.

### Crear e executar proxectos

Para crear un proxecto SpringBoot, abrimos aa paleta de comandos (*Menú View > Command Palette* ou ben **Cntrl + May + P**) e seleccionamos **"Spring Initializr: Create a Maven Project"**



Seguimos os pasos que nos vai indicando:

- **Versión de Spring:** deixamos a que nos indica por defecto
- **Lenguaje:** Java
- **Group id:** esta será a raíz dos paquetes do noso proxecto, comunmente é *com.nomeOrganizacion*, pero podemos deixalo co seu valor por defecto.
- **Artifact id:** será o nome do programa.
- **Packaging type:** jar. Ao xerar un arquivo jar, o propio programa terá un servidor web embebido polo que poderemos executalo de forma independente. O empaquetado "war" sería para ubicar o proxecto nun servidor web externo.
- **Versión de Java:** Deberá coincidir con algún de los jdk que teñamos instalado.
- **Dependencias:** marcaremos aquí las dependencias (librerías) que sexan necesarias para o noso proxecto. Para unha aplicación web ás mínimas será *starter-Web* e un motor de plantillas html como *Thymeleaf*. Prememos *[Enter]* cando non queiramos introducir máis.

Finalmente, pregúntanos polo cartafol no que gardar os proxectos. Visual Studio creará un cartafol co nome que lle dimos ao *artifact id*, coa estrutura de arquivos e cartafoles dun proxecto Spring.

## Edición de código

Este IDE ofrece moitas utilidades para facer o traballo de desenvolvemento máis rápido, sobre todo empregando as teclas aceleradoras. A continuación mostramos as más frecuentes.

- **Comentar/Descomentar liñas:** Primeiro seleccionamos as liñas e prememos “**Ctrl+K**”. Isto fai que VS quede agardando a seguinte operación: comentalas premendo “**Ctrl+C**” ou descomentalas premendo “**Ctrl+U**”. Podemos facer as dúas operacións tamén con “**Ctrl+ç**”
- **Buscar e abrir arquivos:** “**Ctrl + T**”
- **Quick Fix.** “**Ctrl + .**” Para resolver errores no código, por exemplo facer as importacións das clases en Java.  
*Java import: cando escribimos a clase, VS incorpora o import de forma automática.*
- **Ver varias xanelas á vez:** sobre a solapa da pestana do código, onde está o nome do arquivo ou programa, con botón derecho, temos opción de partit a xanela en horizontal e vertical.
- **Pechar xanelas:** Cntrl+ W pecha a pestana de código actual, Cntrl + K Cntrl + W pecha todas.
- **Mover a liña actual:** situado enriba dela, Alt + frecha arriba ou frecha abaxo. Se tes un bloque seleccionado, move todo o bloque.
- **Clonar a liña actual:** situado enriba dela, May + Alt+ frecha arriba ou frecha abaxo. Se tes un bloque seleccionado, copia todo o bloque. Cntrl + C e Cntrl+V tamén funciona, e non e preciso seleccionar a liña.
- **Formatea** o documento para que quede ben indentado: May +Alt + F.
- **Engadir os imports** necesarios das clases empregadas no arquivo: May + Alt + O
- **Renomear** arquivos, variables, etc. F2
- **Eliminar liña actual:** Cntrl + May + K
- **Múltiples cursores:** Alt + clic, cada clic crea un cursor. Logo podemos escribir en todos á vez. “Esc” para volver ao cursor simple.
- **Eliminar tabuladores e espazos del final:** Un dos trucos más fáceis, sinxelos, rápidos e eficaces que podes aplicar para eliminar espazos, tabulacións ou líñas en branco que polo xeral son innecesarios, é premendo “**Ctrl+E**” ou “**Ctrl+O**”. Isto evitará que teñas que premer unha e outra vez a tecla “**Supr**” para borralos.  
Asemade, podes combinar e usar teclas tales como: “**Ctrl+E, S**” para poder verificar e visualizar se existen espazos en branco onde hai códigos; o que lograrás con isto é substituír dítos espazos por puntos.
- **Seleccionar rectángulos de texto:** En vez de seleccionar liñas, podemos seleccionar zonas rectangulares de varias liñas premendo “**May”+ “Alt**” e arrastrando o rato.
- **Xerar código** (como getters, setter, etc...): botón derecho sobre o código > Source Action
- Buscar en varios arquivos: “**Ctrl+F**” ou “**Ctrl+Shift + F**” incluindo todas os cartafoles e subcartafoles do proxecto.
- **Snippets:** Nome snippet + “Tab” + “Tab”, por exemplo: For + “Tab” + “Tab” mostrará unha estrutura de exemplo dun bucle for. Podemos crear os nosos snippets de forma sinxela, en formato JSON.
- **Contraer ou expandir un fragmento de código:** “**Ctrl+M+M**” para expandir ou contraer. Pódese ocultar un código para que apareza debaixo dun signo “**+**”. Do mesmo modo, pódese contraer se se fai dobre clic nunha liña da rexión. Logo, para ver o contido, hai que colocar o rato sobre esta para que mostre a información.

- **Zoom in/out:** "Cntrl" + "+" / "Cntrl" + "-" respectivamente.
- **Mostrar na carpeta contenedora:** Na ventá da dereita, sobre o arquivo, botón dereito podemos facelo directamente (en Netbeans era preciso un plugin)
- **Crear archivos referenciados:** Por exemplo, se estamos facendo unha páxina HTML e temos unha referencia <a href="cartafol1/arquivo1.html"> con CRTL+clic sobre o arquivo referencia, dirá que non existe ese arquivo1.html, e ofrécenos a posibilidade de crealo nese momento. Esta é unha forma rápida de ir construíndo a estrutura na nosa web.



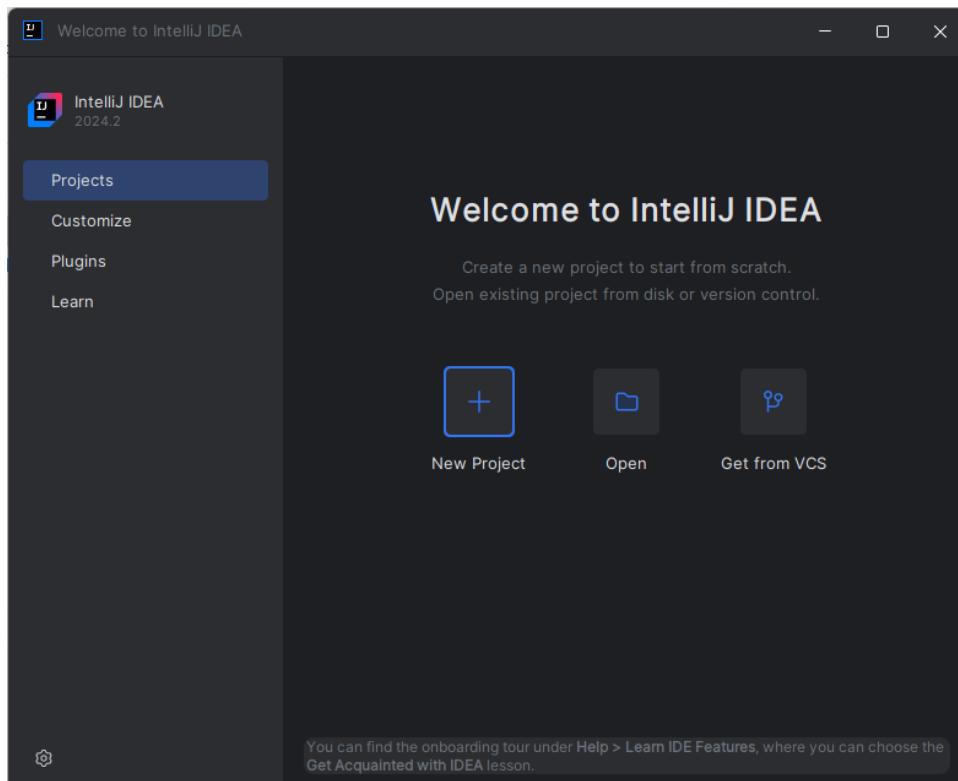
**Tarefa 2.21. Crear proxecto Spring en Visual Studio Code.** A tarefa consiste en preparar un videotutorial que capture a túa pantalla e inclúa as seguintes operacións:

- Configuración Visual Studio para o seu uso con SpringBoot como se mostra nestes apuntamentos.
- Abrir o proxecto proporcionado polo profesor e identificar o cartafolios fundamentais (o dos archivos java, os archivos html, imaxes, etc).
- Podes modificar os archivos html.
- Identificar o dinamismo no html (investiga que é o motor de plantillas Thymeleaf).
- Mostrar as utilidades de edición que ofrece o IDE descritas previamente como clonar e mover liñas, selección rectangular, múltiples cursores, etc.
- Compilar e executar o proxecto en Visual Studio Code (jar vs war, proba nun navegador mediante http://localhost:8080).
- Para gravar e editar o vídeo podes emplegar unha ferramenta gratuita como Active Presenter.

## 3.5

## Proxecto Java en IntelliJ Idea

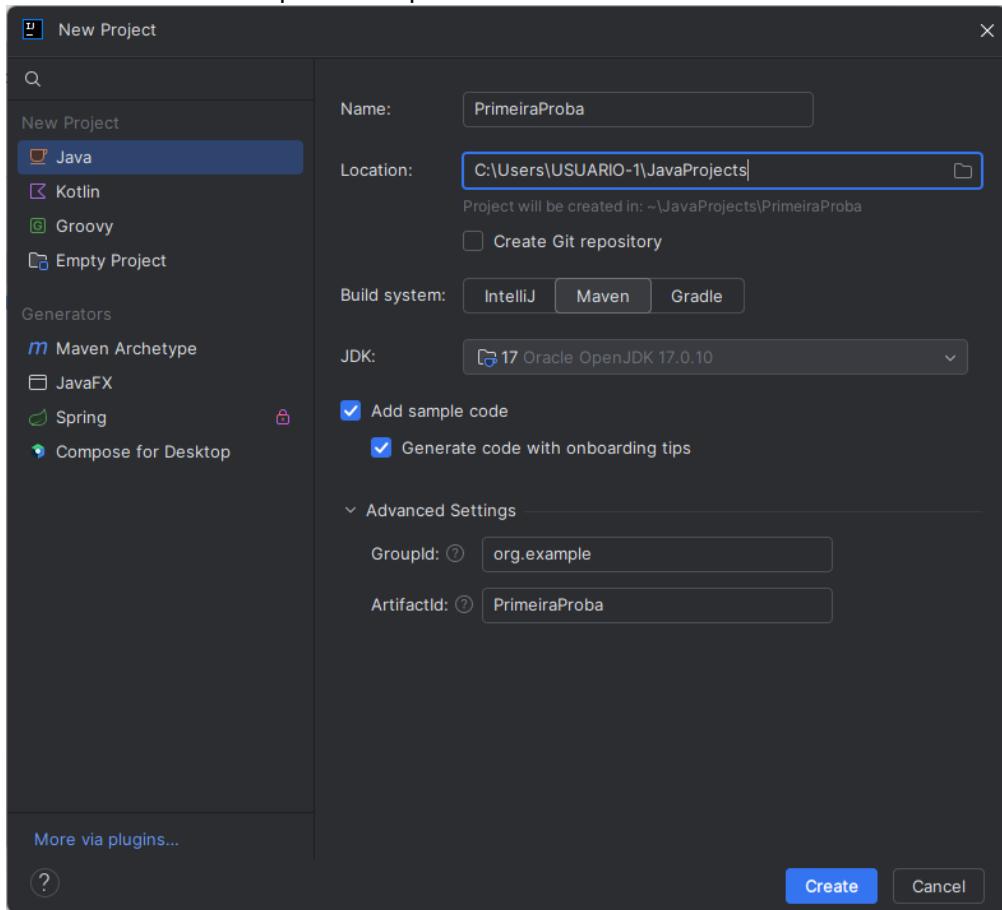
Unha vez instalado e configurado o JDK como vimos en apartados anteriores, podemos crear proxectos Java en IntelliJ Idea directamente sen tarefas adicionais. Desde a ventá de inicio: *New Project*



Na creación do proxecto permítéños seleccionar o JDK a empregar, entre os instalados, proxecto, incluso descargar un novo JDK nese momento. Seleccionamos o nome da aplicación e a súa

ubicación, o sistema de empaquetado (en xeral empregarmos Maven) e tamén podemos indicar o nome do paquete raíz.

O IDE xerará toda a estrutura do proxecto por nós.



A partires dese momento, na ventá lateral “Project” podemos ver esta estrutura, navegar por ela e ir engadindo e editando os arquivos da aplicación.

Alt + Insert = Novo arquivo (nova clase Java) e podemos indicarle o paquete no que creala (os paquetes podemos consideralo como cartafolos que nos permiten organizar todos os arquivos Java que conforman a aplicación).

Para escribir o código do noso primeiro programa, non crearemos nin arquivos nin paquetes, xa que podemos introducir todo o código no arquivo Main.java que se creou no proxecto.

Agora deberemos introducir o código que desexemos. Aquí son moi interesantes as “plantillas activas de código” coas que, escribindo unhas soas letras dunha instrución, a modo de abreviatura, nos suxire a instrución completa. Por exemplo, se escribo “sout”..... e pulso TAB suxire..... que a premer ENTER quedará introducida no código ou ESC para descartalo. Con Control + J podemos ver todas as plantillas de código dispoñibles nese escenario (poderíamos engadir as nosas propias plantillas)

Unha vez rematada a introducción de código, podemos executar a aplicación de distintos xeitos:

- Premendo a icona do triángulo verde na clase que contén o método “main”.
- Premendo a icona do triángulo verde no método “main”.
- Alt + F10
- Etc...

A execución pode ser completa ou ben paso a paso (debug):

Main.java

```
package org.example;
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello and welcome!");
    }
}
```

Run

```
"C:\Program Files\Java\jdk-17.0.10\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Hello and welcome!
Process finished with exit code 0
```

## 4. Proxecto Java en Netbeans

Un proxecto Java en NetBeans permitirá editar o código fonte, compilalo para xerar o *bytecode* e executalo utilizando a máquina virtual Java. O compilador e a máquina virtual Java están instalados na máquina obrigatoriamente porque se necesitaron para a instalación de NetBeans.

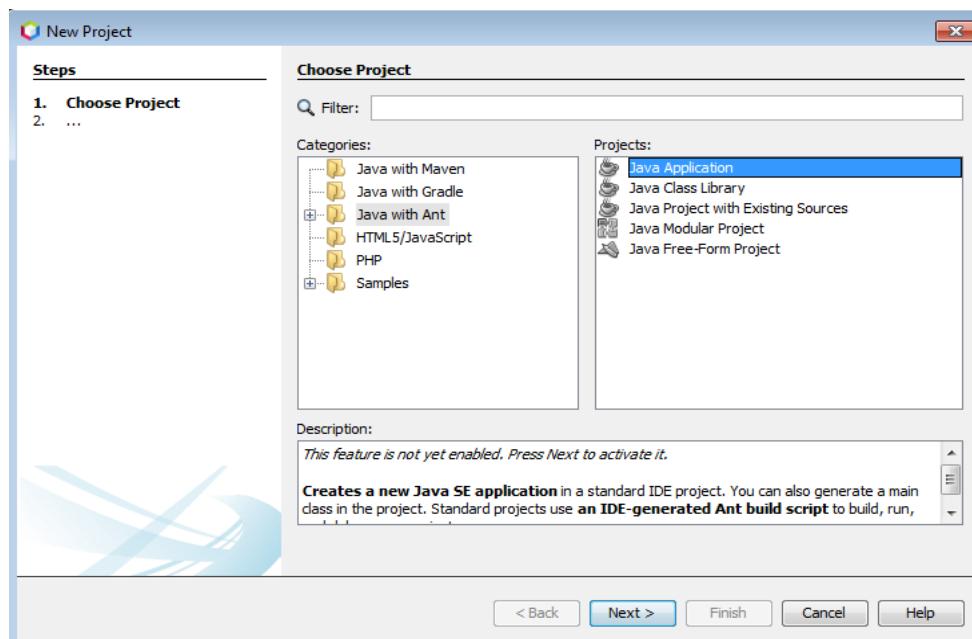
NetBeans facilita e automatiza as operacións a realizar sobre o código Java pero este non é o único procedemento para realizaras xa que a edición podería facerse cun editor de texto calquera, a compilación podería facerse dende a liña de comandos co compilador *javac* e as ferramentas de construcción Java e a execución podería facerse directamente no sistema operativo mediante a máquina virtual Java.

### Crear proxecto

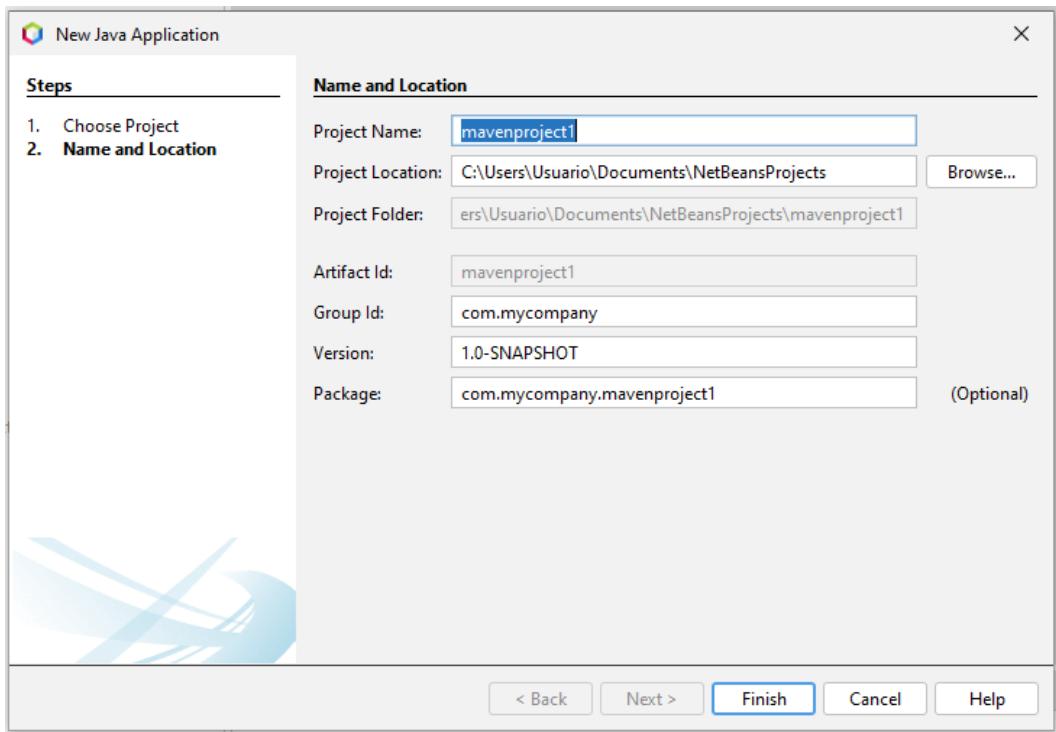
A creación dun proxecto Java que dará lugar a unha aplicación sinxela de consola é similar á creación dun proxecto noutra linguaxe. Unha diferenza propia de Java é por exemplo a utilización de paquetes: por defecto crearase un paquete Java co mesmo nome do proxecto e a clase principal chamarase como o proxecto pero empezando por maiúscula aínda que todo isto pode cambiarse na creación do proxecto.

Para crear o proxecto: Menú File > New Project.

Aquí temos varias opcións, en función de como desexamos que se xestionen as dependencias (proxecto Maven, Gradle ou Ant). Por agora, non é importante para nós, podemos elixir *Java with Maven > Java Application*. *Lembra que tes que ter activado o plugin Java SE.*



Asignámosselle un nome ao proxecto e fixámonos na súa ubicación.



Chegando á pantalla na que desenvolvermos o proxecto.

Na ventá lateral esquerda podemos ver a estrutura do proxecto. Por defecto crea un paquete co mesmo nome que o proxecto (en minúsculas). O paquete é un “contedor” cartafol de clases relacionadas. Ven sendo un concepto similar ao dun cartafol e o seu uso para ordenar os nosos arquivos. Un proxecto pode ter todos os “contedores” que queiramos e, por suposto, cada “contedor” pode ter as clases que queiramos.

Cada clase pública terá o seu correspondente arquivo .java. Todo proxecto terá unha clase especial, cun método chamado *main*, que conterá o código que executará o programa.

```

Mavenproject1.java x
Source History | I O V S M F D C E L T P G R B H
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-d...
3   */
4
5  package com.mycompany.mavenproject1;
6
7  /**
8   *
9   * @author Usuario
10  */
11 public class Mavenproject1 {
12
13     public static void main(String[] args) {
14         System.out.println("Hello World!");
15     }
16 }
17

```

## Editar código fonte

NetBeans dispón de axudas en liña para escribir código como por exemplo: completado de código, consellos en liña e suxestións. Recoméndase utilizarlas para minimizar a xeración de erros.

Outra axuda que ten NetBeans para a edición de código é que vai mostrando os distintos elementos do código coas cores e as tabulacións correspondentes a esta linguaxe. Isto facilita moito a lectura e comprensión do código.

Nalgúns casos, NetBeans fará advertencias ou avisará de erros mediante iconas que coloca sobre o número de liña (columna á esquerda). Os erros terán que ser arranxados porque se non a compilación fallará.

**Tarefa 2.22. Crear proxecto Java en NetBeans (Maven >> Java Application) de nome EsPrimo** que dará lugar a unha aplicación sinxela de consola. Editar o código fonte para que permita teclear un número entero pequeno e visualice se o número é primo ou non. Concretamente:

- Primeiro haberá unha clase que teña un método que calcula se un número é primo ou non, podería ser algo así:

```
package com.mycompany.esprimo;
public class Calculadora {
    public static boolean esPrimo(int n) {
        if (n<=2) return false;
        for (int i = 2; i <= n/2; i++) {
            if (n % i ==0) return false;
        }
        return true;
    }
}
```

- Logo, no mesmo paquete, teríamos unha clase co main (que é o programa en si) que pide que introduzamos un número e invoca ao método *esPrimo()* da outra clase. Sería así:

```
package com.mycompany.esprimo;
import java.util.Scanner;

public class NewMain {

    public static void main(String[] args) {
        Scanner teclado = new Scanner (System.in);
        System.out.println("Introduza un número entero: ");
        int num = teclado.nextInt();
        boolean resultado = Calculadora.esPrimo(num);
        if (resultado == true)
            System.out.println("El número introducido es primo");
        else
            System.out.println("El número introducido no es primo");
    }
}
```

- Sobre o código anterior, proba os diferentes atallos de teclado que ten Netbeans.

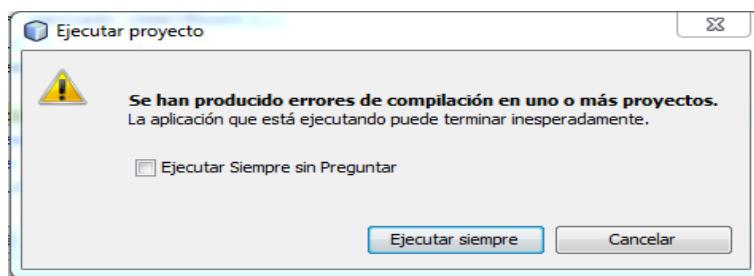
Zoom in e out	Alt + Roda rato.
Indentación do código	Alt + May + F - Botón dereito do rato > Format
Xerar código	Alt + Insert
Busca onde se usa unha determinada clase	Alt + F7
Engadir comentarios e viceversa na liña actual	Ctrl + May + C (ou bloque seleccionado)
Arranxar todas as importacións de clases	Ctrl - May - I - Botón dereito do rato > Fix Imports
Move liñas arriba/ abaixo	Alt + May + ↑ / Alt + Shift + ↓
Clonar liñas	Ctrl + May + ↑ ou Ctrl + May + ↓
Eliminar liña actual	Ctrl + E
Alternar selección rectangular	Ctrl + May + R
Inspeccionar membros	Ctrl + F12
Inspeccionar xerarquía	Alt + F12
Código Completo	Ctrl + Espazo

## Compilar e executar

A forma más rápida de compilar para xerar o código intermedio e executar utilizando a máquina Java é premer F6 ou facer clic en  na barra de ferramentas; se hai varios proxectos abertos, executarase o proxecto que estamos editando. Pódese definir tamén un proxecto principal desde o menú *Run > Set main project* e sería este o proxecto que executaría.

Se tiveramos varios archivos (clases) cun método executable (main) no mesmo proxecto, por exemplo, se estamos facendo exercicios ou probas, premeríamos [MAIÚSCULAS] + [F6] e se executaría soamente ese arquivo. Se prememos soamente [F6] executaría o main da clase especificada nas propiedades do proxecto, na sección *Run*.

NetBeans avisará dos erros producidos na compilación mediante unha ventá emergente. De elixir na ventá anterior que se quiere continuar coa execución, aparece na ventá *Salida* a información sobre os errores encontrados.



Se non hai errores de compilación, executarase o proxecto e aparece o arquivo .class resultante da compilación na ventá *Archivos* despregando a carpeta *build*. Se o proxecto é unha aplicación de consola, verase o resultado da execución na ventá *Salida* como se fose a ventá de comandos do sistema operativo.

A execución do arquivo resultante tamén pode facerse dende a ventá de comandos. Para iso hai que acceder á ventá de comandos do sistema, moverse pola árbore de directorios co comando *cd* ata chegar á localización da carpeta *classes* e executar:

```
java nomepaquete.nomearquivoclass
```

## Exportar un proxecto. Arquivos "jar".

Unha vez rematado o noso proxecto (deseño, codificación e probas) procederemos ao seu empaquetado para distribuír aos usuarios. Para construír o “executable” que poderá interpretar o *jre*, dirixímonos a ventá de proxectos, e sobre o nome do proxecto, botón dereito > *Clean and Build* ou directamente > *Build* (a primeira opción borra clases de compilacións previas).

Este proceso xerará un cartafol */target* (se é un proxecto Maven) ou */dist* (se é proxecto Ant). Nese cartafol teremos un arquivo .jar que é o produto final que se poderá executar en calquera equipo coa Máquina Virtual Java instalada.

O arquivo jar contén .class (as clases compiladas dos .java) pero tamén iconas, imaxes, etc. O arquivo jar é un arquivo comprimido, que se poderá descomprimir con 7zip ou similar aínda que non é preciso. O entorno de execución de Java pode executalo directamente con comando: *java -jar arquivo.jar*

Se probas a executar, nun proxecto Maven, directamente o jar co comando `java -jar arquivo.jar` informará do erro “*no main manifest attribute*”. Para arranxalo hai que engadir ao arquivo `pom.xml` as seguintes liñas ao final:

```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-jar-plugin</artifactId>
            <version>2.4</version>
            <configuration>
                <archive>
                    <manifest>
                        <addClasspath>true</addClasspath>
                        <mainClass>app.Inicio</mainClass>
                    </manifest>
                </archive>
            </configuration>
        </plugin>
    </plugins>
</build>
```

Sendo `app.Inicio` o paquete.clase que contén o main. Facemos o *Build* de novo e xa poderemos executar o .jar.

Finalmente, comentar que existe utilidades como *Install4j* e *InnoSetup* que permiten envolver os jar en executables típicos do sistema operativo, con pasos que o usuario debe seguir, aceptar o contrato de licenza, etc.



### Tarefa 2.23. Realizar as seguintes operacións co proxecto anterior:

- Compilar o proxecto anterior Java en NetBeans e execútalo no contorno Netbeans.**
- Executar desde fóra de NetBeans (desde a liña de comandos, sobre o arquivo .class).**
  - Para isto debemos abrir a aplicación “Símbolo do sistema” e movernos ata o cartafol que contén as clases compiladas do noso proxecto. Probablemente:  
`cd C:\Users\NombreUsuario\Documents\NetbeansProjects\EsPrimo\target\classes`
  - Nesa ubicación xa podemos executar:  
`java com.mycompany.esprimo.NewMain`
- Xerar un .jar, modificar o pom.xml e execútalo tamén desde liña de comandos.**
  - Para isto, de novo debemos abrir a aplicación “Símbolo do sistema” e movernos ata o cartafol que contén o “jar” do proxecto. Probablemente:  
`cd C:\Users\NombreUsuario\Documents\NetbeansProjects\EsPrimo\target\`
  - Nesa ubicación xa podemos executar:  
`java -jar EsPrimo-1.0-SNAPSHOT.jar`
  - Lembrade que hai que modificar o pom.xml para que este paso funcione.

## Importar clases doutros proxectos no noso proxecto

En moita ocasións, pode ser necesario empregar clases feitas noutros proxectos, sexan proxectos executables ou simplemente librerías de clase. Poden ser librerías nosas ou de terceiros.

Se o que precisamos é una clase soamente, dun proxecto noso, podemos simplemente copiala e pegala áínda que este é unha forma un pouco "de andar por casa".

Cando o que queremos é empregar son varias clases dun proxecto, o proceso sería o seguinte:

1. Se o proxecto orixinal e noso debemos **facer o Build para obter o "jar"** xa que este será o arquivo que engadiremos ao noso proxecto. Se é unha librería externa, deberemos obter o arquivo jar.
2. Se o noso proxecto destino (ao que lle queremos incorporar as clases) é Maven, imos ao cartafol **Dependencies > Add Dependency**. Pestana "Open projects" e incluímos proxecto coas clases que queremos empregar.
3. Xa no código fonte importamos as clases que precisemos do jar:  
`import paquete.clase;`  
e podemos usar as clases de jar orixe.

- **Tarefa 2.24. Crea un novo proxecto e incorpora o .jar do proxecto anterior e usa no main do novo proxecto algunha das clases que teña o proxecto anterior.**

- *Para esta práctica partimos das dúas anteriores, que inclúen a clase Calculadora co seu método estático: esPrimo().*
- *O que tedes que fazer é crear un novo proxecto, que se pode chamar, por exemplo: Tarefa2\_24 e que terá un programa (main class) que empregará o método esPrimo() do outro proxecto.*
- *Poderíamos pensar nun programa que mostrase por pantalla os números primos comprendidos entre 1 e 100.*
- *Sobre o proxecto debemos engadir a dependencia, e o código da main class sería algo así:*

```
package com.mycompany.tarefa2_24;
import com.mycompany.esprimo.Calculadora;

public class NewMain {
    public static void main(String[] args) {
        for (int i = 1; i <= 100; i++) {
            if (Calculadora.esPrimo(i)) System.out.println(i);
        }
    }
}
```

- *Ao rematar, bótalle unha ollada ao **pom.xml** e comenta os cambios que atopes ao engadir a dependencia.*

# 5. Anexo: Práctica PHP en Netbeans

Para as tarefas 2.19 e 2.20

## 5.1 Descripción de la aplicación

En la aplicación que os entrega el profesor, el usuario deberá inicialmente introducir una palabra clave para identificarse (en este momento es: 1234). Esta clave está almacenada en el servidor, en un archivo llamado *password.txt*.

Una vez identificado correctamente se le presentará un menú con las siguientes opciones:

1. Consultar el listado completo de preguntas con sus posibles respuestas.
2. Añadir nuevas preguntas (con sus respuestas posibles) e indicando cual es la correcta
3. Eliminar preguntas, seleccionándolas de una lista.
4. Generar un examen en PDF con el número de preguntas que se deseen. (El programa las elegirá al azar)
5. Desconectarse.

Se emplean los siguientes aspectos PHP:

1. Lectura de ficheros (para comparar la clave introducida por el usuario con la requerida)
2. Sesiones. La palabra clave de acceso será una variable de sesión que se comprobará en todas las páginas antes de hacer cualquier operación. Al desconectarse se destruye la variable de sesión.
3. Accesos a base de datos: altas, bajas y consultas de la base de datos.
4. Generación de archivos PDF.
5. Estructuras y algoritmia como el cálculo de números aleatorios sin repetidos.
6. Con las modificaciones solicitadas, se emplean sistemas de encriptado/desencriptado.

## 5.2 Instalación

Para instalar la aplicación descomprimir el .zip entregado sobre la carpeta raíz de documentos (*www*, *htdocs*, *public\_html*, etc), en una subcarpeta de la misma (por ejemplo: *www\appPHP\*)

A continuación, es necesario crear en MySQL la base de datos “practicas”, un usuario llamado ‘user1’ (contraseña ‘pass1’), darle permisos a ese usuario sobre esa BD y finalmente crear la tabla en la que se almacena toda la información de la aplicación. Proceso a seguir:

1. Conectarse a <http://localhost/phpmyadmin>
2. Pestana *Base de datos*. *Nombre de la base de datos*: ‘practicas’. Botón *Crear*.
3. Pestana *Privilegios* (si no ves esta pestaña, estará oculta debajo de “Más”).

The screenshot shows the 'Users' section of the phpMyAdmin interface for the 'practicas' database. It lists three users: 'root' with IP '127.0.0.1', 'root' with IP '::1', and 'root' with IP 'localhost'. All three users have 'global ALL PRIVILEGES' and 'Sí' checked for 'Conceder'. There are 'Editar privilegios' and 'Exportar' buttons for each user. At the bottom, there are 'Nuevo' and 'Agregar cuenta de usuario' buttons.

Nombre de usuario	Nombre del servidor	Tipo	Privilegios	Conceder	Acción
root	127.0.0.1	global	ALL PRIVILEGES	Sí	
root	::1	global	ALL PRIVILEGES	Sí	
root	localhost	global	ALL PRIVILEGES	Sí	

4. Agregar cuenta de usuario: Nombre de usuario: **user1**. Nombre de Host: **localhost**. Contraseña: **pass1**. Dejamos marcado el campo: "Otorgar todos los privilegios para la base de datos prácticas".  
 Este usuario/contraseña es un usuario "interno", con el que la aplicación se conecta a la base de datos, pero no son para que los usuarios finales de la aplicación se autentifiquen.

Y pulsamos el botón de *[Continuar]* abajo del todo de la página.

5. Menú lateral > **BD practicas**. Pestana: *Importar*. Seleccionar el archivo **crear\_tabla\_y\_datos.sql** (viene en el zip entregado) para crear la tabla e insertar los datos, y pulsar *Continuar*.

Obteniendo:

The screenshot shows the phpMyAdmin interface for the 'practicas' database. The left sidebar lists databases: Nueva, information\_schema, mysql, performance\_schema, phpmyadmin, practicas, test, and preguntas. The 'practicas' database is selected. The main area shows the results of an import operation: 'Importación ejecutada exitosamente, 2 consultas ejecutadas. (\_crear\_tabla\_y\_datos.sql)'. It also displays the creation of the 'preguntas' table and the insertion of 5 rows of data. The SQL code for creating the table and inserting data is visible.

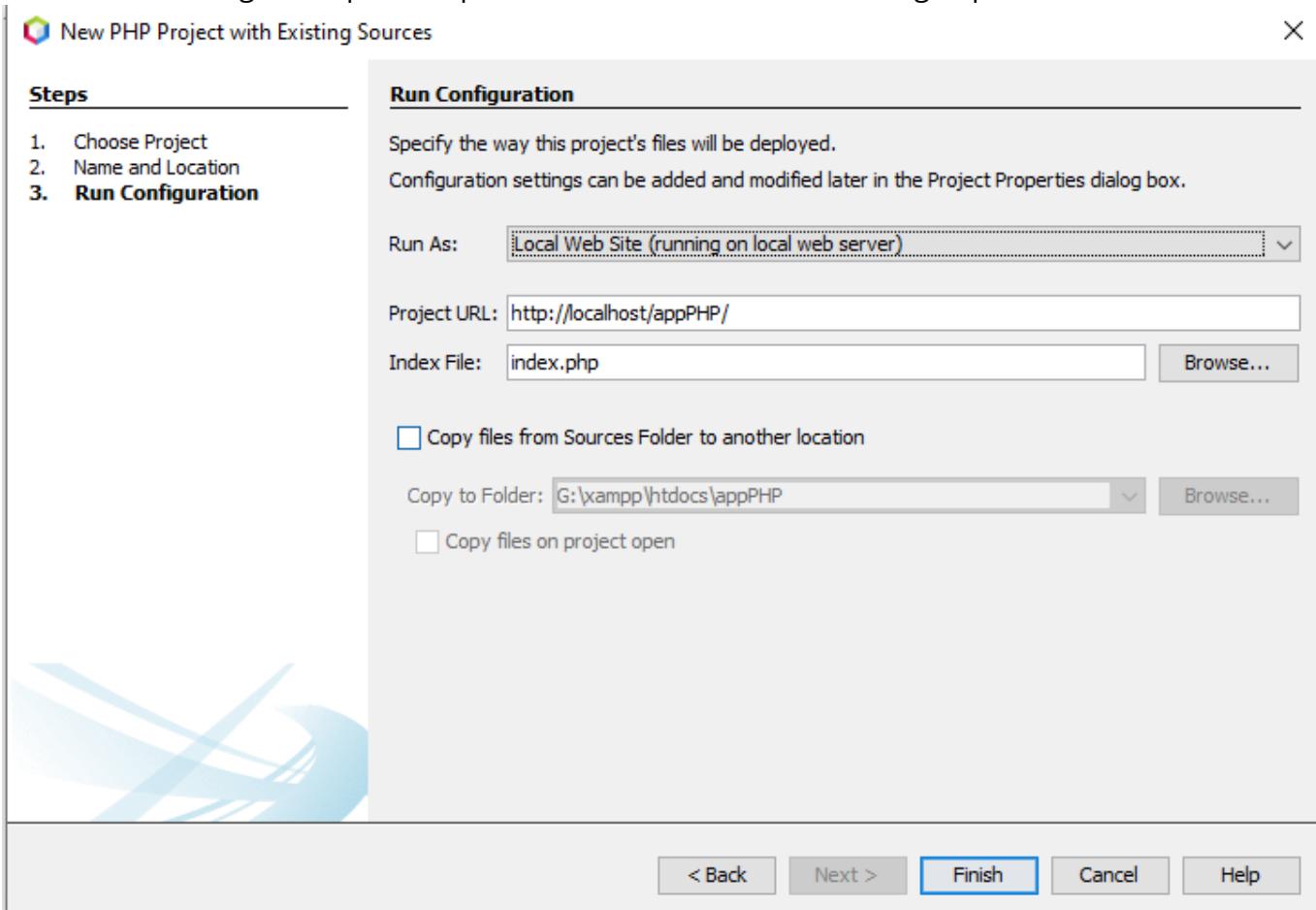
6. Ahora, en el navegador, accediendo a: <http://localhost/appPHP/> accedemos a la aplicación con contraseña 1234:

The screenshot shows a browser window with the URL 'localhost/appPHP/'. The page title is 'Gestión de examen tipo test'. It has a form with a placeholder 'Introduzca la clave de acceso' and a 'Entrar' button.

7. El último paso es asociarla a Netbeans. Para ello, desde NetBeans, menú *File > New Project > PHP > PHP Application With Existing Sources*:  
Seleccionamos la carpeta donde está nuestro proyecto.

The screenshot shows the 'New PHP Project with Existing Sources' dialog in NetBeans. It is on the 'Name and Location' step. The 'Sources Folder' field is empty and has a note: 'Sources Folder must be specified.' The 'Project Name' is 'appPHP', 'PHP Version' is 'PHP 7.2', and 'Default Encoding' is 'UTF-8'. A checkbox 'Put NetBeans metadata into a separate directory' is checked, and the 'Metadata Folder' is 'C:\Users\Usuario\Documents\NetBeansProjects\appPHP'. A 'Select Project Folder' dialog is open, showing the file tree under 'Look in: htdocs'. The 'appPHP' folder is selected. The 'File name:' field is 'G:\xampp\htdocs\appPHP' and the 'Files of type:' is 'All Files'. Buttons at the bottom include '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Y en la siguiente pantalla pulsamos *Finalizar* sin cambiar ningún parámetro:



Si ahora, en Netbeans, seleccionamos en el menú *Ejecutar Proyecto* (o F6) nos abrirá el navegador por defecto del sistema ejecutando la aplicación web. Sería lo mismo que si nosotros desde un navegador, pusiésemos la URL: <http://localhost/appPHP/>

## 5.3 Modificaciones en PHP

La modificación consistirá en permitir al usuario que pueda cambiar la contraseña y que esté encriptada en el archivo *password.txt*. Los pasos a seguir serán los siguientes.

1. Copiar todos los archivos de la aplicación sobre una nueva carpeta, por ejemplo *appPHP2*, también en la carpeta raíz de documentos (*www*) , y trabajar sobre la nueva carpeta para poder, en caso de problemas, volver a la situación inicial.
2. Repetiremos el último punto del apartado anterior, para crear un nuevo proyecto PHP a partir de la carpeta *appPHP2* recién creada. menú *File > New Project > PHP > PHP Application With Existing Sources*:
3. Sobre el proyecto, editamos el archivo *2\_menu.php* para añadirle la nueva opción de menú, para cambiar la contraseña:

```
<a href="#">./3_cambiarpass.php">Cambiar Contraseña</a>
```

4. Crear un nuevo archivo (*Source Files > New > PHP Web Page*) con el formulario para cambiar la contraseña. Le llamaremos `3_cambiarpass.php` en el que el usuario introduce la contraseña (dos veces, luego la validaremos con JavaScript) y llame al proceso que efectivamente cambia la contraseña:

```
<?php include '1_validar_sesion.php';?>
<html>
<body>
<h1>Cambiar contraseña</h1>
<form name="formu" action=".//4_FileCambiarPass.php" method="post">
    Nueva contraseña<input type="password" name="pass1" required/><br/>
    Repetir contraseña<input type="password" name="pass2" required/><br/>
    <input type="submit" value="Cambiar Contraseña" />
</form>
</body>
```

5. De la misma forma que los dos pasos anteriores, crear el archivo `4_FileCambiarPass.php` que actualiza la contraseña:

```
<?php include '1_validar_sesion.php';?>
<html>
<meta charset="UTF-8"/>
<body>
<?php
$pass1=$_POST['pass1']; $pass2=$_POST['pass2'];
if ($pass1!=$pass2) { Header ("Location:5_error1.html"); exit; }
$cryptpass = password_hash($pass1, PASSWORD_DEFAULT);
$f1=fopen("password.txt", "w");
fwrite = fwrite($f1, $cryptpass);
if (fwrite == false) {fclose($f1); Header ("Location:5_error1.html"); exit;}
fclose($f1);
$_SESSION['pass']=$pass1;
?>
<p> Contraseña guardada correctamente </p>
<a href=".//2_menu.php">Volver al menu</a><br />
</body>
</html>
```

6. Ahora hay que cambiar la verificación de la contraseña (ahora hay que desencriptarla después de leerla del archivo). Modificar el archivo: `1_validar_sesion.php` en la línea 9 cambiando:

```
if ($_SESSION['pass']!=$pass) Header ("Location:5_error1.html");
```

por:

```
if (password_verify ($_SESSION['pass'] , $pass ) == false)
    Header ("Location:5_error1.html");
```

7. Ahora ya lo podríamos ejecutar, pero tenemos un problema. El código está preparado para que lea del archivo una contraseña encriptada, pero si vemos el archivo `password.txt`, la contraseña no está encriptada. Para solucionar esto, tenemos que deshabilitar la verificación de contraseña y cambiarle la contraseña para que la escriba encriptada en el archivo `password.txt`. Para ello:

- Editamos el archivo `1_validar_sesion.php` y ponemos `//` al principio de las líneas introducidas en el paso 5. Así no las tendrá en cuenta. Guardamos el archivo.

```
// if (password_verify ($_SESSION['pass'] , $pass ) == false)
//Header ("Location:5_error1.html");
```

- Ejecutamos la aplicación, menú *Run > Run Project*, o bien pulsando F6 o directamente desde el navegador: <http://localhost/appPHP2> o como le hubiésemos llamado a la carpeta de la aplicación en www (o *htdocs*).
- Metemos una contraseña cualquiera (no la está verificando ahora) y nos dirigimos a la opción de menú “Cambiar contraseña” y le introducimos una nueva contraseña (puede ser también 1234 como antes)
- Comprobar que acepta la nueva contraseña y verificar que la contraseña está encriptada en el archivo *password.txt*.
- Quitarle las // que acabamos de meter en *1\_validar\_sesion.php* y guardamos el archivo.

8. Ahora ya podemos trabajar con la aplicación normalmente.

## 5.4 Modificaciones en JavaScript

La modificación consistirá en validar que, en el cambio de contraseña, las dos contraseñas son iguales y de una longitud mayor o igual a 4 caracteres. Para ello, editamos el documento con el formulario *3\_cambiarpass.php* añadiendo en el *<head>* justo, después de la etiqueta *<html>* la función JavaScript de validación.

```

<head>
<script>
function validarFormulario () {
    if (document.forms["formu"]["pass1"].value.length < 4 || 
        document.forms["formu"]["pass2"].value.length < 4) {
        window.alert("La contraseña debe ser de 4 o mas caracteres");
        document.forms["formu"]["pass1"].value="";
        document.forms["formu"]["pass2"].value="";
        document.forms["formu"]["pass1"].focus();
        return (false);
    }
    if (document.forms["formu"]["pass1"].value != 
        document.forms["formu"]["pass2"].value ) {
        window.alert("Las contraseñas no coinciden");
        document.forms["formu"]["pass1"].value="";
        document.forms["formu"]["pass2"].value="";
        document.forms["formu"]["pass1"].focus();
        return (false);
    }
    return (true);
}
</script>
</head>

```

Y en la definición del *<form>* la llamada a la función. El envío solo se realizará si la función devuelve *true*.

```

<form name="formu" action="./4_FileCambiarPass.php"
      onsubmit="return (validarFormulario());" method="post">

```

## 5.5 Modificaciones en HTML y CSS

En esta sección podrás hacer los cambios que deseas en HTML y CSS en base a tus conocimientos previos en estas materias.

Por ejemplo, en HTML podrías añadir una tabla para presentar los resultados de “*Ver preguntas*”, y en CSS cambiar el color de fondo de la página o los formatos de los textos.