

# 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 máis empregados .....	4
1.4	Contorno de desenvolvemento libre NetBeans.....	7
1.5	Contorno de desenvolvemento libre Eclipse .....	12
1.6	Contorno de desenvolvemento Visual Studio .....	15
<b>2.</b>	<b>Actualización, personalización, automatización e xestión de módulos do contorno de desenvolvemento .....</b>	<b>18</b>
2.1	Actualizar NetBeans .....	18
2.2	Personalizar NetBeans .....	20
2.3	Automatizar en NetBeans.....	23
2.4	Xestionar complementos de NetBeans.....	26
2.5	Extensións de Visual Studio .....	27
<b>3.</b>	<b>Uso básico do contorno de desenvolvemento en operacións de edición e xeración de código.....</b>	<b>28</b>
3.1	Introdución.....	28
3.2	Proxecto C/C++ en Netbeans .....	28
3.3	Proxecto PHP en NetBeans.....	30
3.4	Proxecto Visual C++ en VisualStudio Code .....	33
<b>4.</b>	<b>Proxecto Java en Netbeans .....</b>	<b>39</b>
<b>5.</b>	<b>Anexo:Práctica PHP en Netbeans .....</b>	<b>44</b>
5.1	Descrición de la aplicación .....	44
5.2	Instalación .....	44
5.3	Modificaciónes en PHP.....	45
5.4	Modificaciónes en JavaScript .....	46
5.5	Modificaciónes en HTML y CSS .....	46
5.6	Modificaciónes en BootStrap .....	46

Traballo derivado por: Fernando Rodríguez Diéguez. rdf@fernandowirtz.com

Con 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

Licenza Creative Commons BY-NC-SA (recoñecemento - non comercial - compartir igual).

Para ver unha copia desta licenza, visitar a ligazón <http://creativecommons.org/licenses/by-nc-sa/3.0/es/>.

# 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 Desenvolvemento 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 erros, 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 máis 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 erros 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 máis 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 produto 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.
- **Construción 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 construción 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 máis 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 máis 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 máis 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 máis populares do ano 2016 (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, NetBeans e Eclipse.

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 gratuita 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.

## NetBeans

Comezou como un proxecto estudantil na República Checa en 1996 baixo a titoría da Facultade de Matemáticas e Física da Universidade Carolina en Praga e orixinalmente chamouse Xelfi. En xuño de 2000, Sun Microsystems patrocina NetBeans



e decláralo 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 utilizalo como mellor lle pareza dentro dos termos de uso que se explican na páxina oficial: <http://netbeans.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<sup>1</sup> dual CDDL (*Common Development and Distribution License*) v1.0 e GNU GPL (*General Public License*) v2 que o fai un produto de código aberto, gratuito 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. Pode ser instalado sobre calquera sistema operativo que soporte Java: Windows, Linux, MacOS X, ou Solaris.

Actualmente e segundo a páxina oficial de NetBeans, lévanse feitas máis de 18 millóns de descargas deste contorno de desenvolvemento.

## 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/>.



---

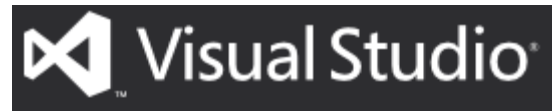
<sup>1</sup> Ver información máis detallada en <https://netbeans.org/cddl-gplv2.html>

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>2</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.

## 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 máis. 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 necesítase 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/>. Os prezos de Visual Studio 2015 con MSDN para as distintas edicións son<sup>3</sup>:

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

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

A versión gratuita Visual Studio Code estase a converter nun dos IDE máis empregados últimamente.

Mono é a plataforma de desenvolvemento de código aberto baseado no framework .NET, que permite aos desenvolvedores construír aplicacións para sistemas operativos diferentes de Windows. Foi patrocinado previamente por Novell e Xamarin, e ten unha comunidade entusiasta e activa. Mono inclúe as ferramentas de desenvolvemento e a infraestrutura necesarias para executar os clientes .NET e as aplicacións de servidor; entre elas



<sup>2</sup> Ver máis información en <https://www.eclipse.org/org/documents/epl-v10.php>

<sup>3</sup> Información extraída de <http://www.visualstudio.com/products/how-to-buy-vs> en outubro de 2018

destacan: compilador C#, Mono Runtime (que implementa o CIL- Common Language Infrastructure- e dispón de compilador JIT e as bibliotecas necesarias), biblioteca de clases base compatibles con .Net Framework classes, e biblioteca de clases Mono con funcionalidades adicionais para construír aplicacións Linux. A páxina oficial é: <http://www.mono-project.com/>.



**Tarefa 2.1.** Buscar en internet contornos de desenvolvemento nos que se poida codificar no maior número de linguaxes de entre as máis populares do ano en curso. Caracterizar os contornos de desenvolvemento diferentes aos estudados nesta actividade



**Tarefa 2.2.** Buscar en internet a última versión estable de NetBeans, Eclipse e Visual Studio Code.

## 1.4 Contorno de desenvolvemento libre NetBeans

### Instalación

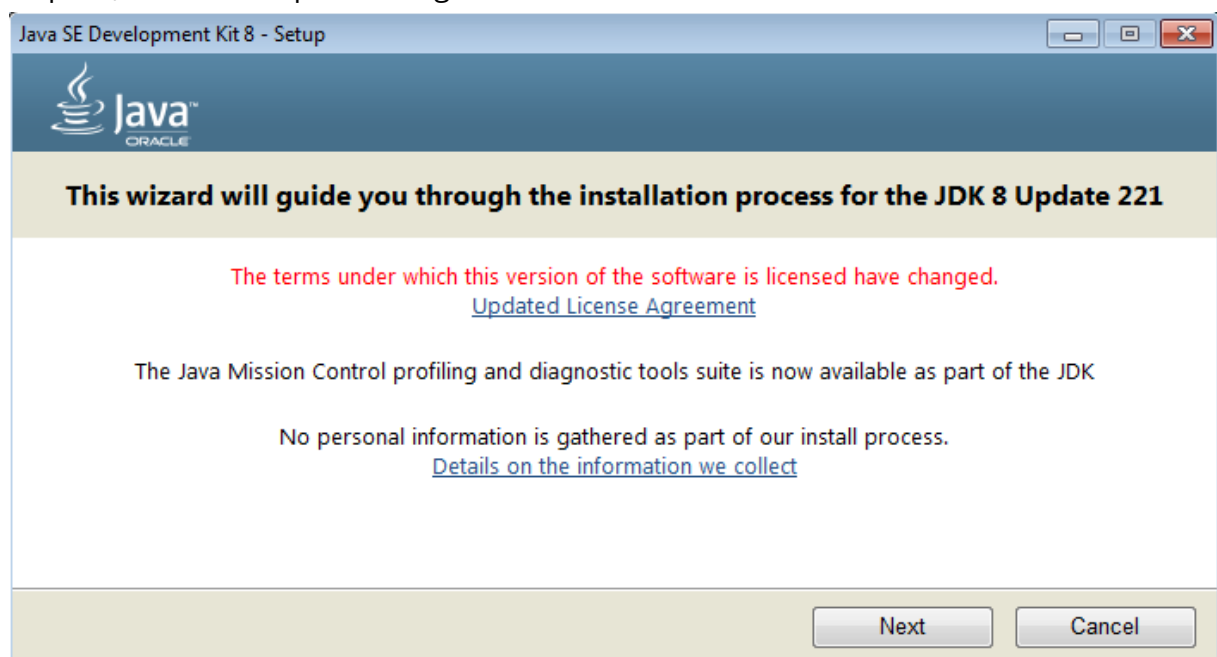
Instalarase o contorno de desenvolvemento libre NetBeans na última versión actual: NetBeans 11.0 en español.

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

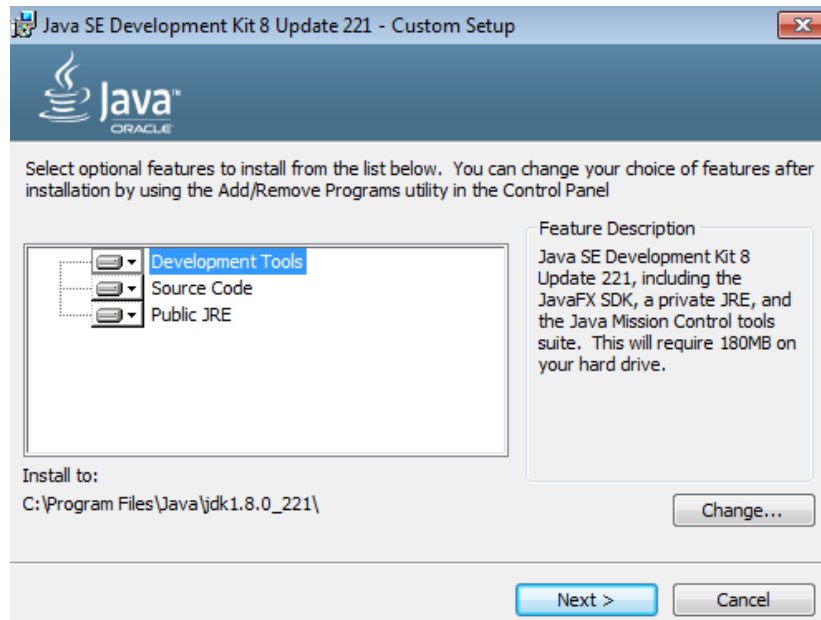
NetBeans 11 está escrito en Java e por tanto antes de instalar NetBeans hai que ter instalado ou instalar JDK 8 (*Java Development Kit*)

JDK é un conxunto de ferramentas para desenvolver aplicacións Java que inclúe entre outros JRE e o compilador Java. JRE (*Java Runtime Environment*) é o contorno necesario para executar as aplicacións Java e inclúe entre outros a JVM (*Java Virtual Machine*) encargada de interpretar o bytecode Java.

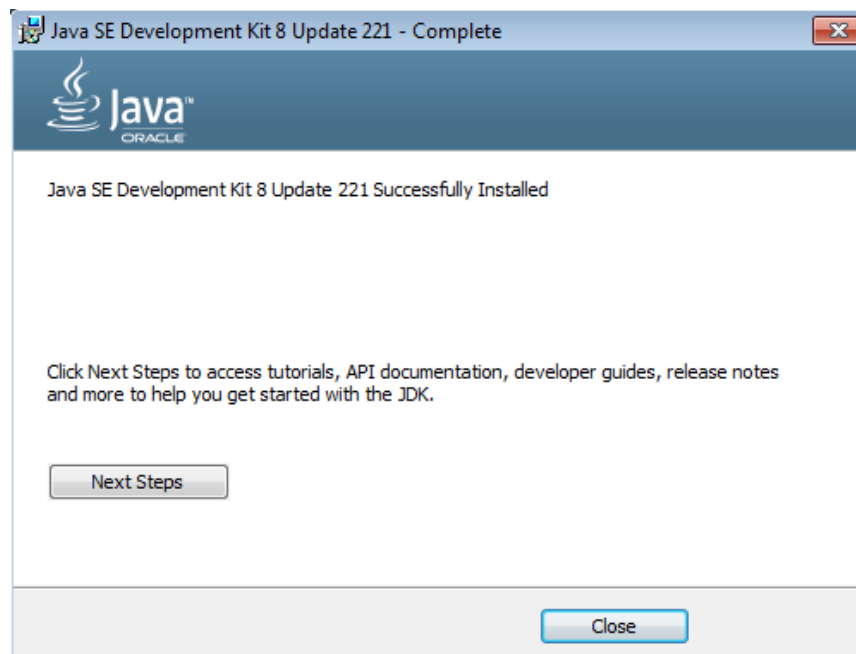
A instalación de JDK consiste na descompresión do zip descargado ou ben a execución do arquivo, no caso de que descargásemos un executable.



Prememos *Next*.

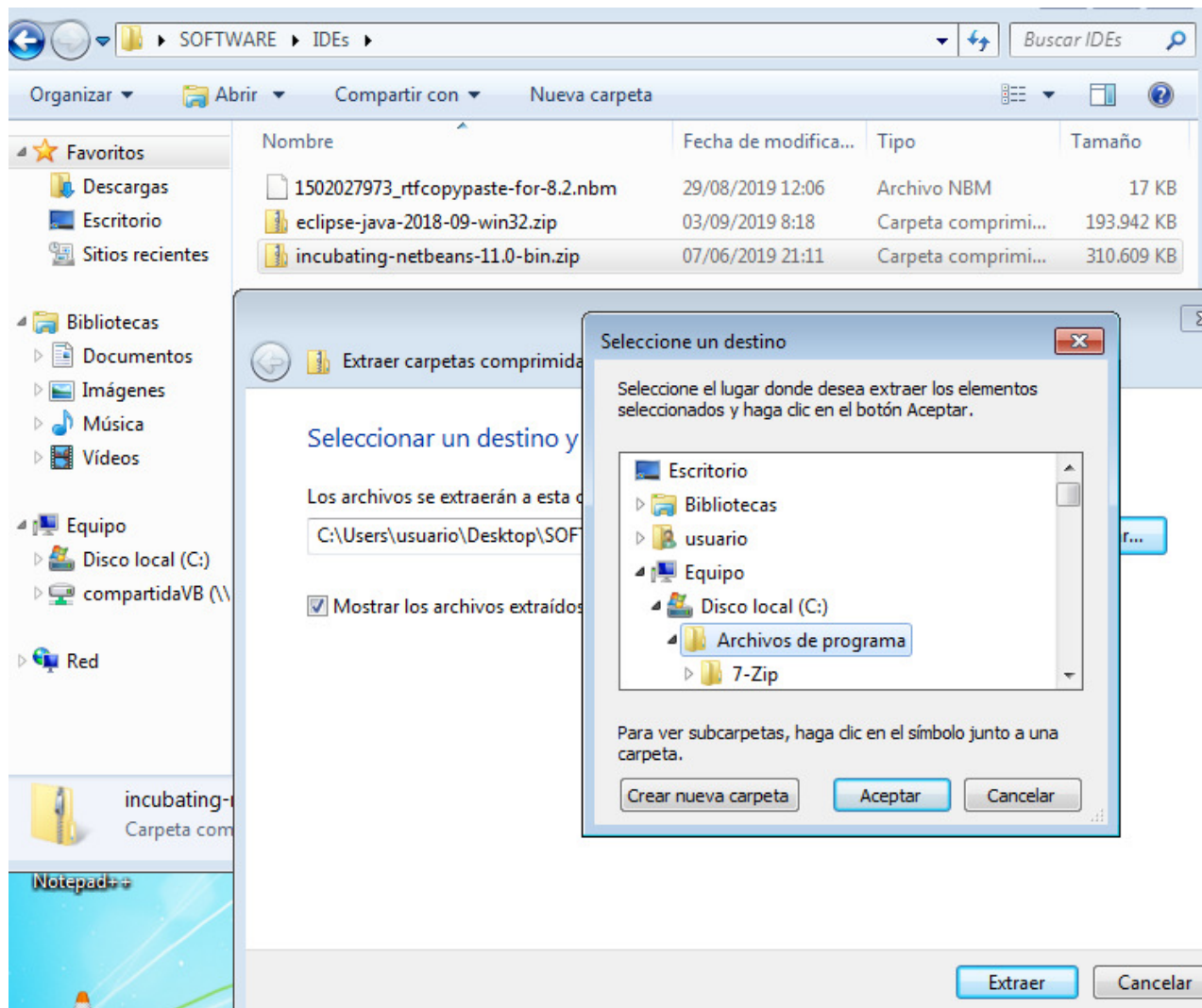


E outra vez *Next*. Deixamos a carpeta destino por defecto, e *Seguinte*, rematando a instalación na seguinte pantalla, na que premeremos *Close*.



A instalación de Netbeans consiste na descompresión do arquivo .zip descargado. En Windows podemos descomprimilo en c:\Archivos de Programa\. Crearase un cartafol chamado *netbeans*, que se queremos podemos renomear a *netbeans11* 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).

```

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 #

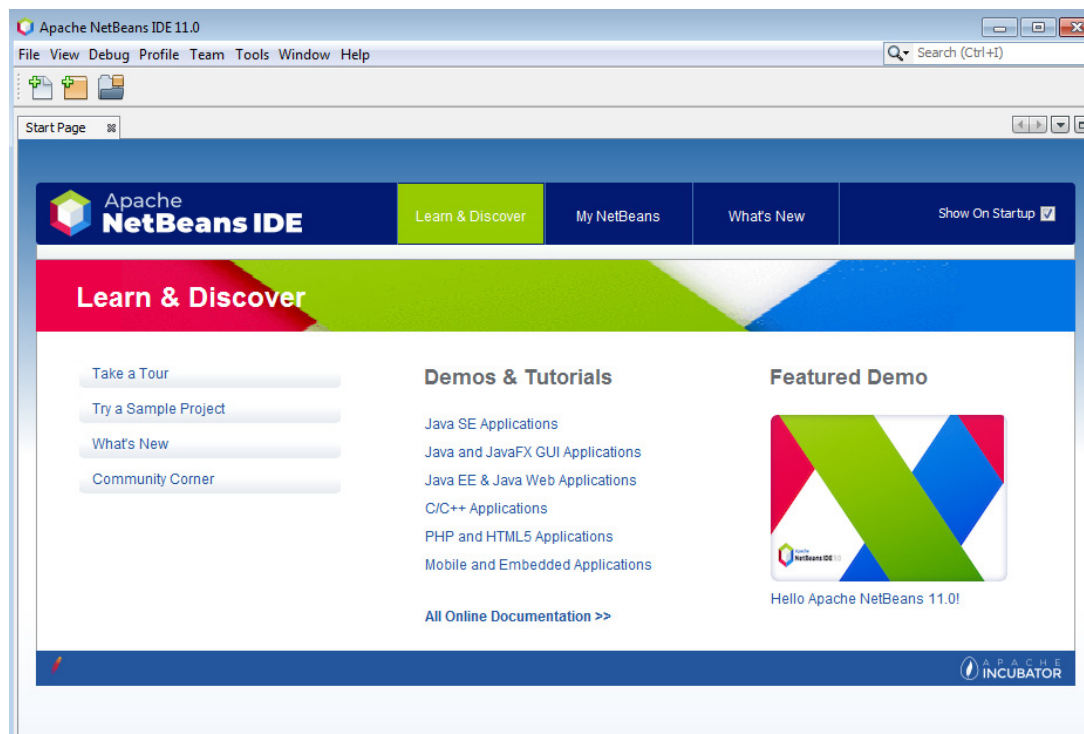
```

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

## Inicio

A forma máis 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 eliminala 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

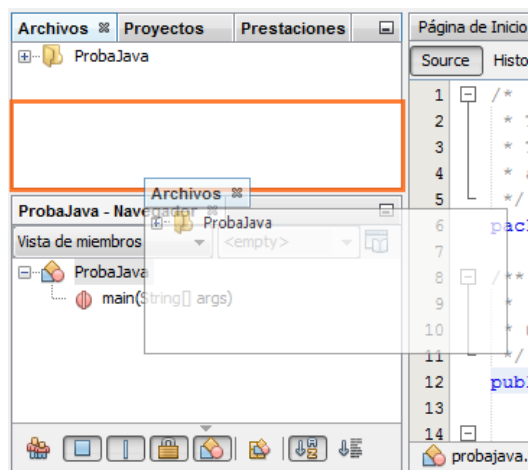
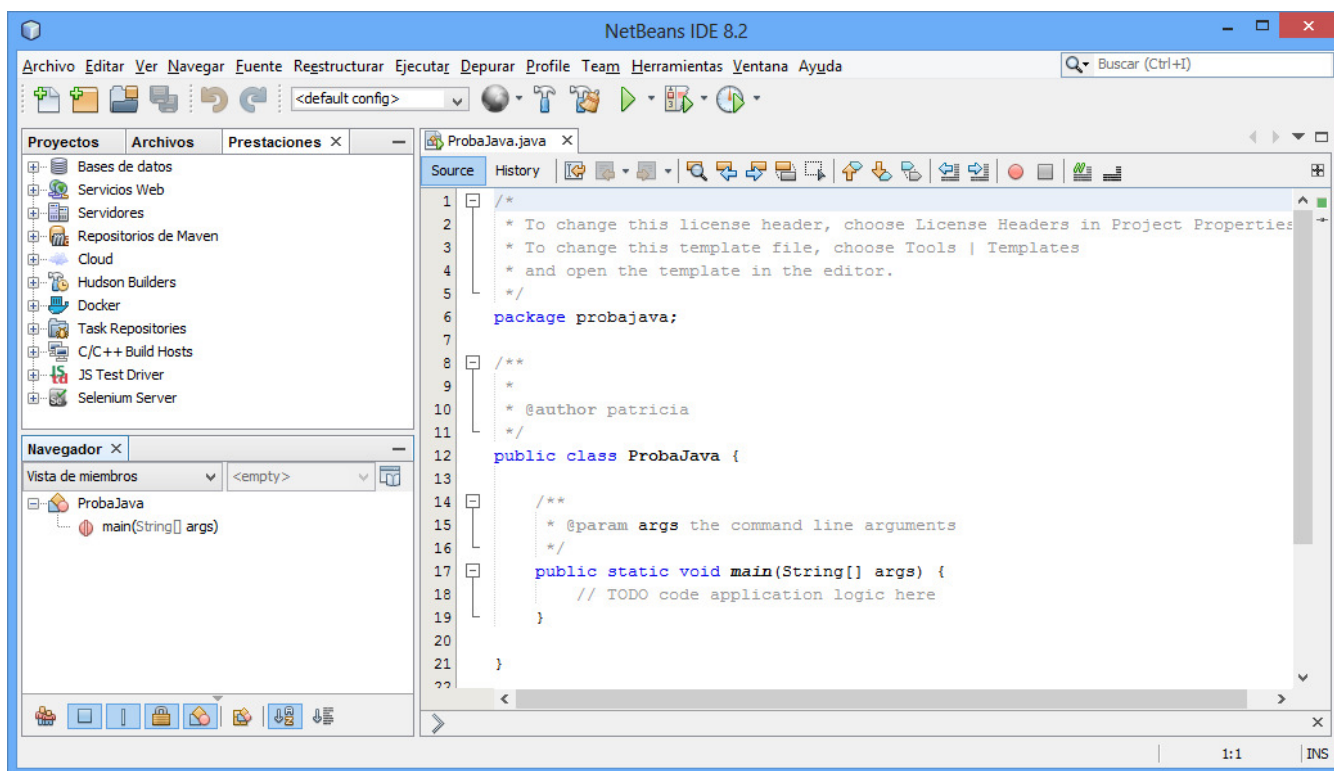
NetBeans permite editar un arquivo independente pero onde se saca todo o proveito ao IDE é utilizando un proxecto, é dicir, unha carpeta cunha estrutura controlada por NetBeans que pode ter arquivos de diferentes tipos relacionados que darán lugar a unha aplicación.

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.



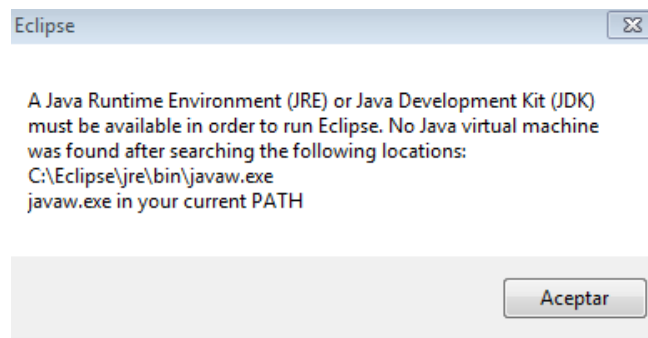
### Tarefa 2.3. Instala Netbeans e crea un proxecto novo Java.

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, sobre o paquete > New > Java Main Class

## 1.5 Contorno de desenvolvemento libre Eclipse

### Instalación

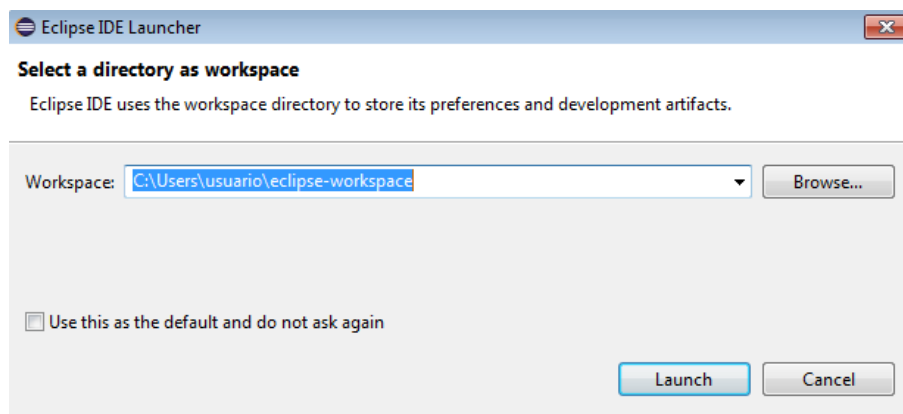
Descargamos o arquivo comprimido e o descomprimos no nos disco, obtendo un cartafol chamado Eclipse, e dentro del o executable *eclipse.exe*. Se o executamos obteremos o seguinte erro:



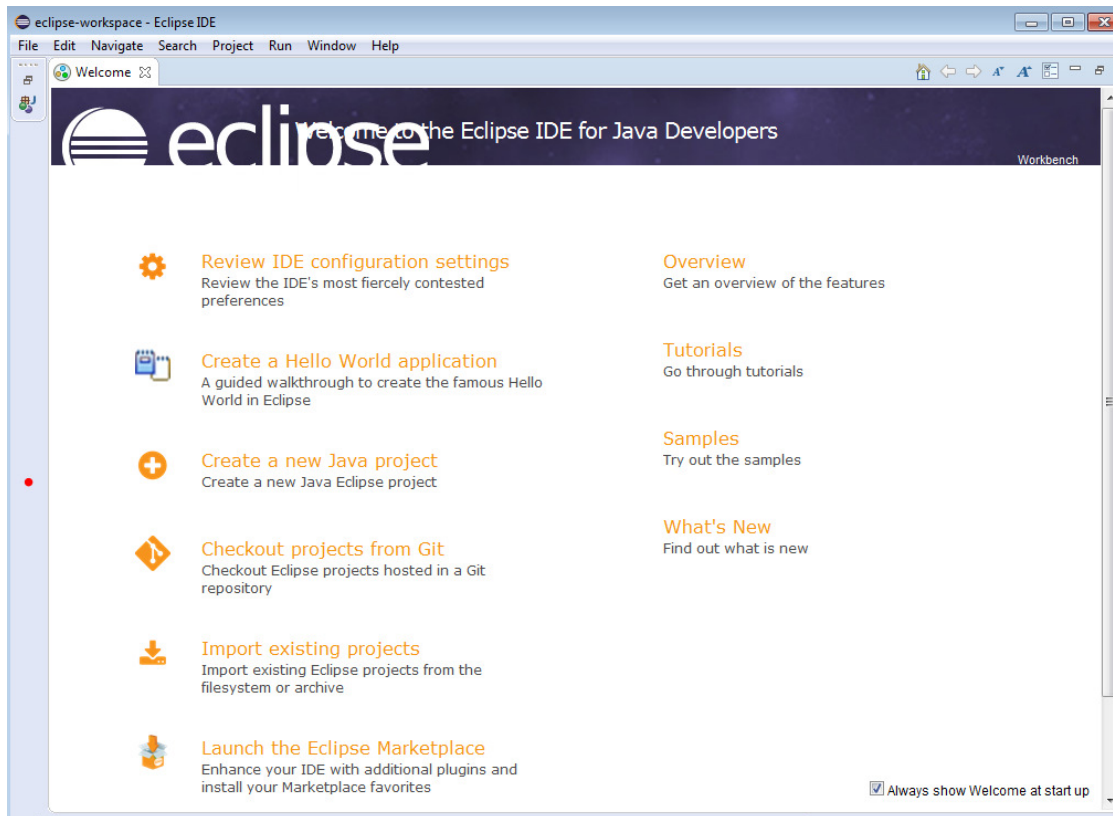
Polo que primeiro instalaremos o jdk de Java, ben a partires dun zip ou dun executable, depende o formato no que teñamos descargado Java. **Unha vez instalado Java, actualizaremos a variable de contorno PATH** (Panel de Control > Sistema > Configuración avanzada). Se a instalación de Java fixémosla mediante unha versión executable, xa non será necesario este paso.

### Inicio

Na primeira execución, pídenos o cartafol orixe para gardar os nosos proxectos. Podemos deixar o valor por defecto e premer *Launch*.



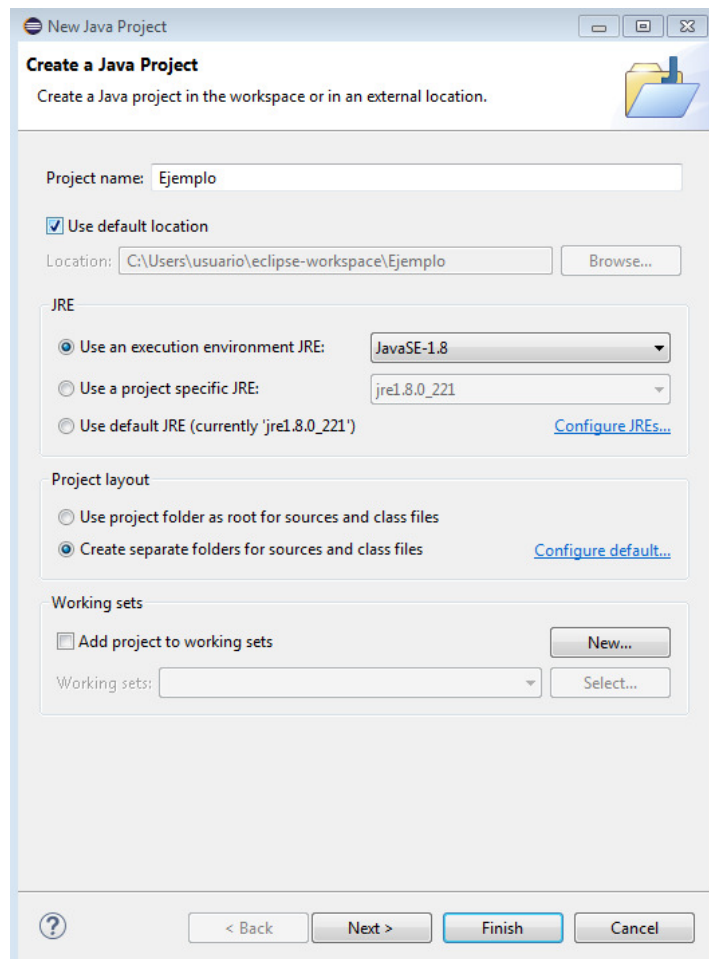
Chegando a pantalla de inicio.



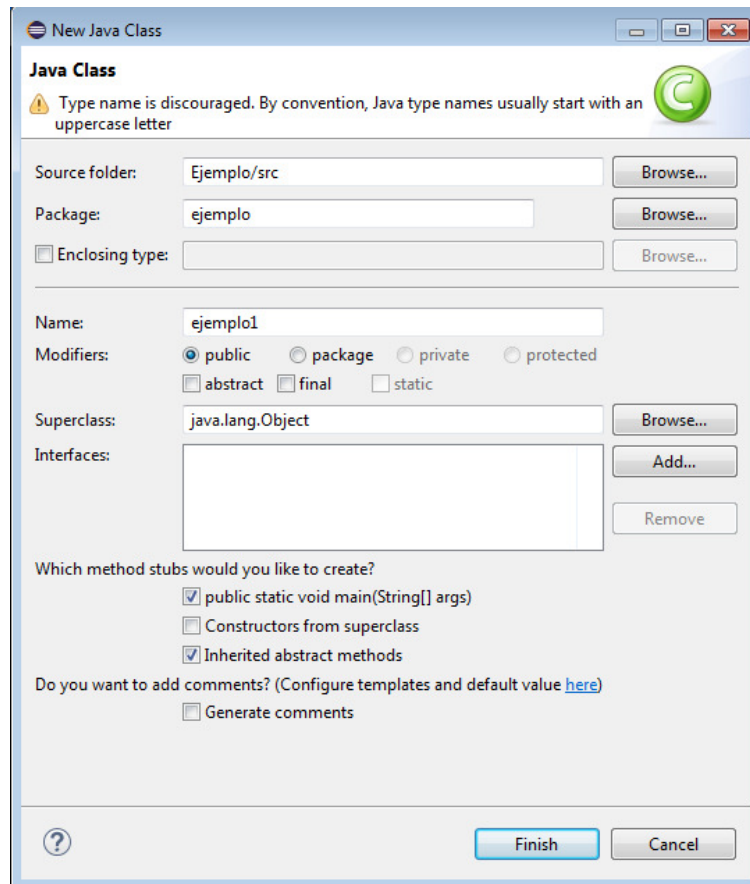
Na que podemos selección “*Create a new Java project*”

## Primeiros pasos

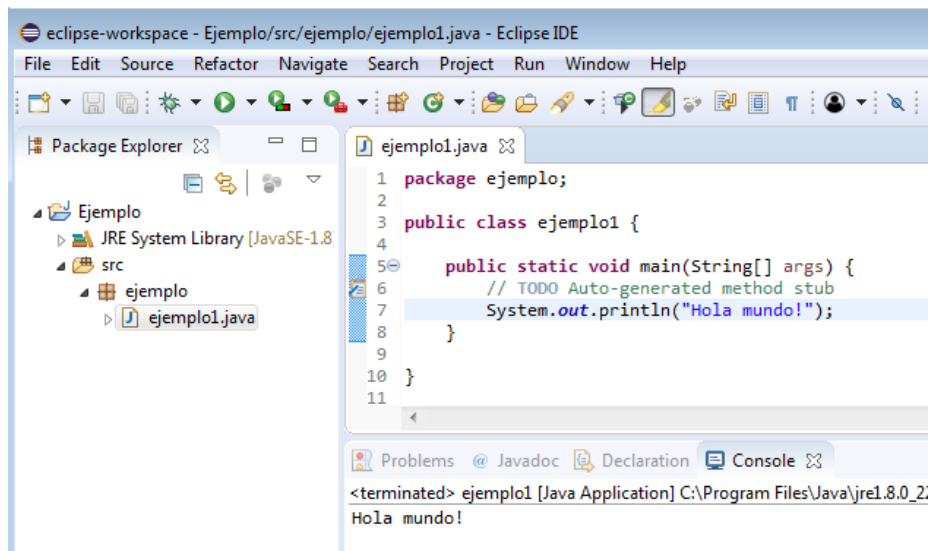
Creamos o novo proxecto dándolle un nome e deixando o resto de parámetros por defecto.



No proxecto, seleccionamos *New Java Class*, indicándolle *package*, *name* e marcando o check *public static void main*.



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



## Tarefa 2.4. Instala Eclipse e crea un proxecto novo Java.



## 1.6 Contorno de desenvolvemento Visual Studio

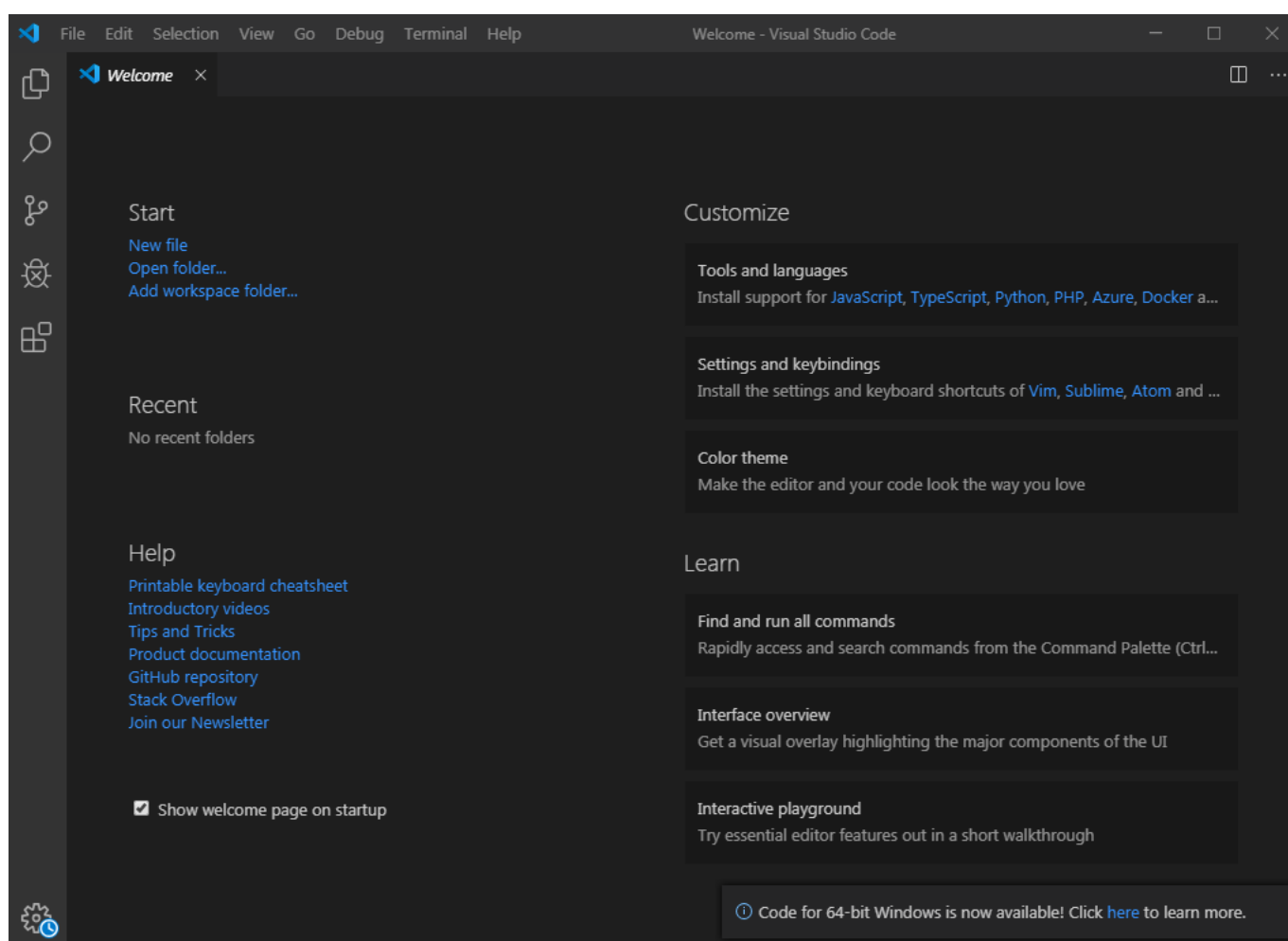
### Instalación

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

Descargamos 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 a 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 excepcións mediante o botón lateral  ou ben premendo *Cntrl+Maiusc+X*, buscamos "Java Extension Pack" e instalámolo se non está instalada. Este

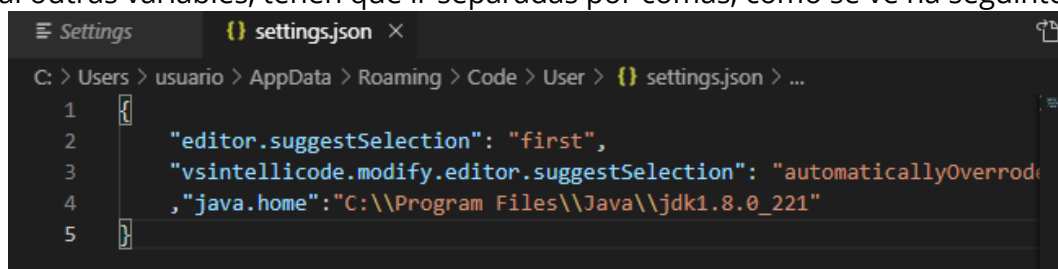
“pack” instala varias extensións necesarias para codificar Java (debugger, completador de código, test runner, xestor de dependencias Maven, etc.)



O derradeiro paso da instalación sería a configuración na que indicamos onde se atopa o *jdk*, Para iso:

- Menú *File > Preferences > Settings*.
- Na xanela *Search settings* escribimos *java.home* e prememos en *Edit in settings.json*
- No arquivo que aparece escribimos a ubicación do jdk. Por exemplo:  
`"java.home": "C:\\\\Java\\\\jdk-13"`

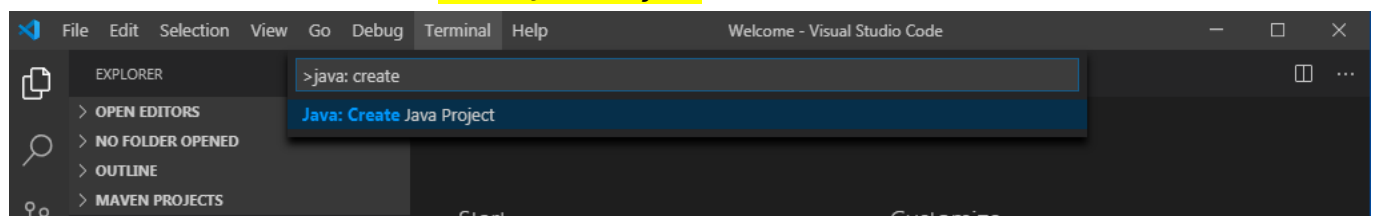
Se hai outras variables, teñen que ir separadas por comas, como se ve na seguinte imaxe:



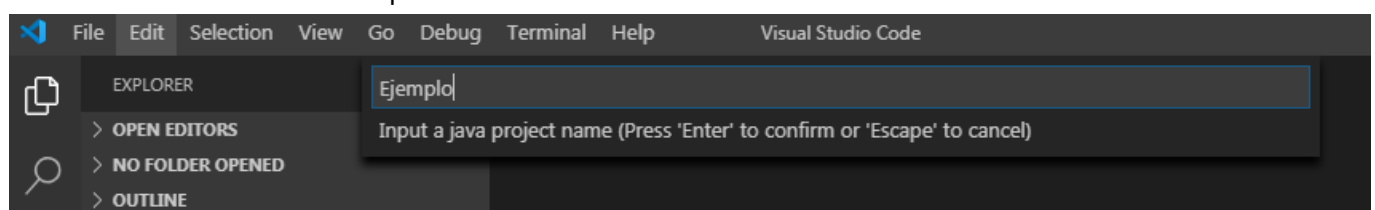
- Pechamos a pestana *settings.json* gardando os cambios.

## Primeiros pasos

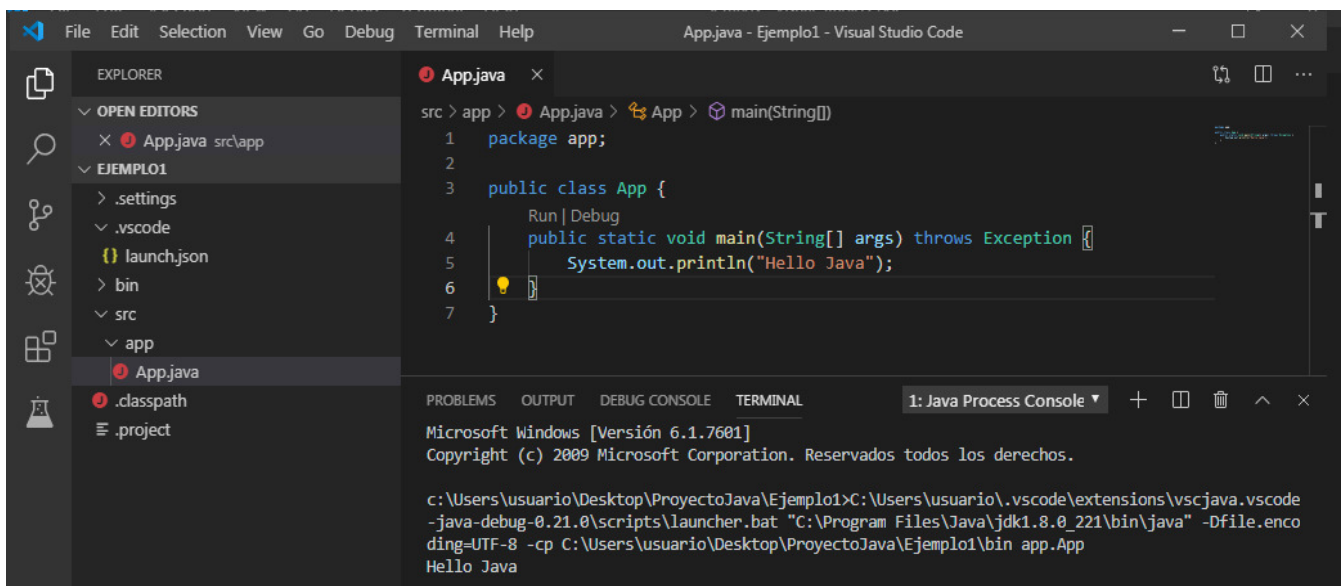
Para crear un proxecto Java a opción máis rápida é abrir a paleta de comandos (menú *View > Command Palette*) e tecleando *Create Java Project*.



Logo seleccionamos un cartafol creado previamente ou o creamos nese mesmo momento, e dámoslle un nome ao noso proxecto:



Automaticamente creárase a estrutura de cartafóles necesario, entre eles *src > app* e alí o arquivo *App.java*. Poderemos editalo e executalo con *Ctrl+F5*.



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

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



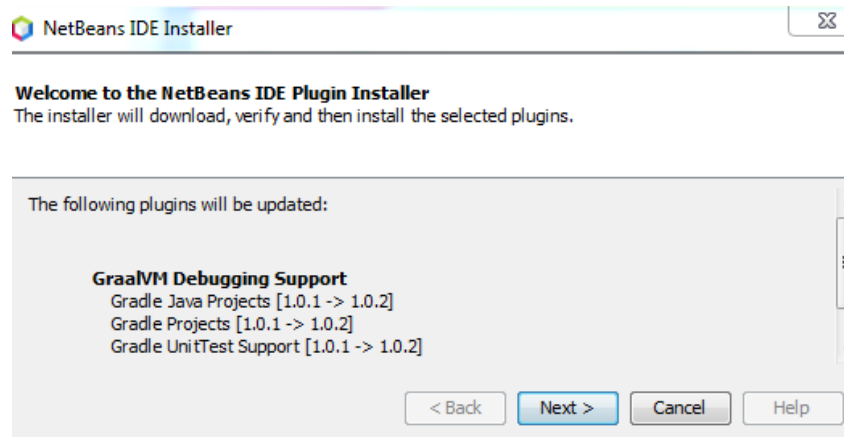
**Tarefa 2.5.** Instala Visual Studio Code e crea un proxecto novo Java. Fai unha captura de pantalla da estrutura de arquivos do proxecto, modifica o tamaño da letra (Settings), e executa algún comando desde a paleta.



## 2. Actualización, personalización, automatización e xestión de módulos do contorno de desenvolvemento

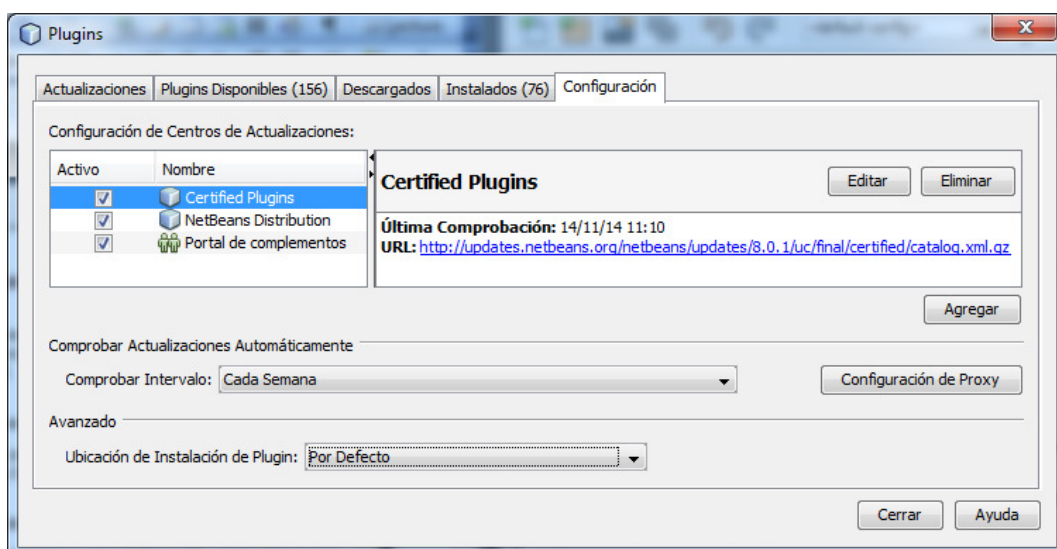
### 2.1 Actualizar NetBeans

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*.

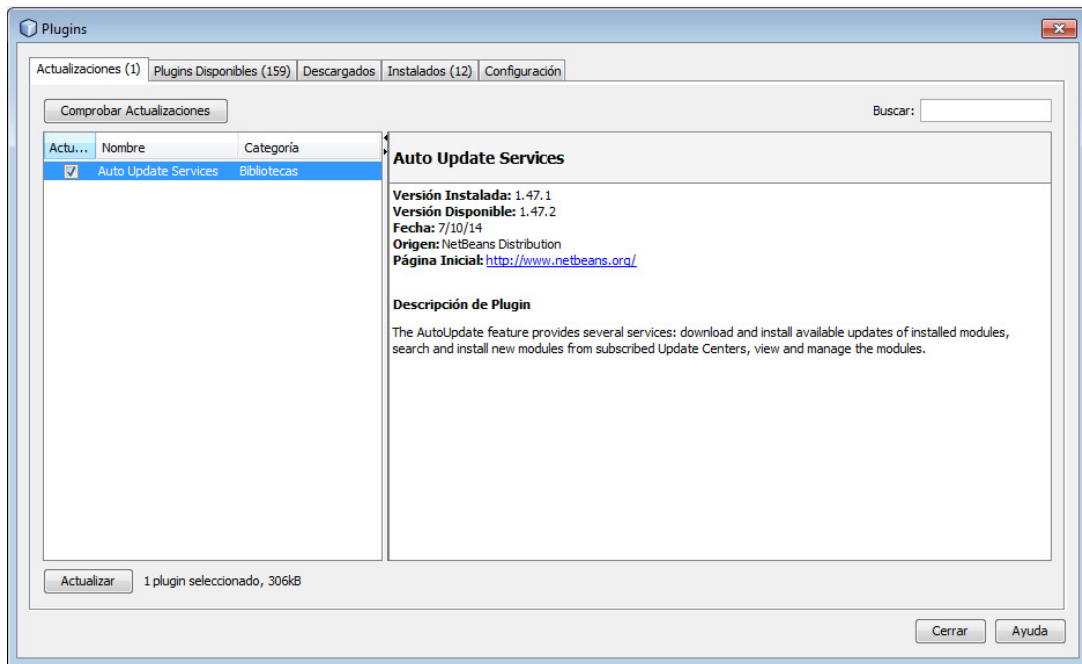


Un complemento ou *plugin* é un grupo de módulos dependentes. Estes módulos poden ser utilizados por varios complementos polo que cando se actualiza un complemento poden actualizarse algúns dos módulos que o compón e por tanto afectar a outros complementos.

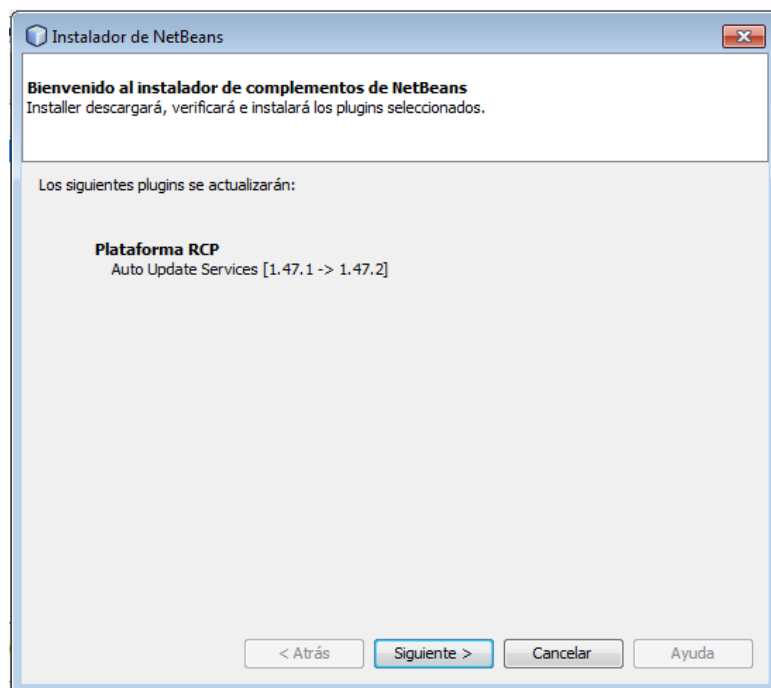
A pestana **Configuración (Settings)** da ventá *Plugins* mostra as características do centro de actualizacións. Nela aparece por defecto 3 url fontes nas que conseguir actualizacións e novos complementos e pódese engadir, activar, desactivar, editar ou eliminar urls. Na mesma pestana pódese indicar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



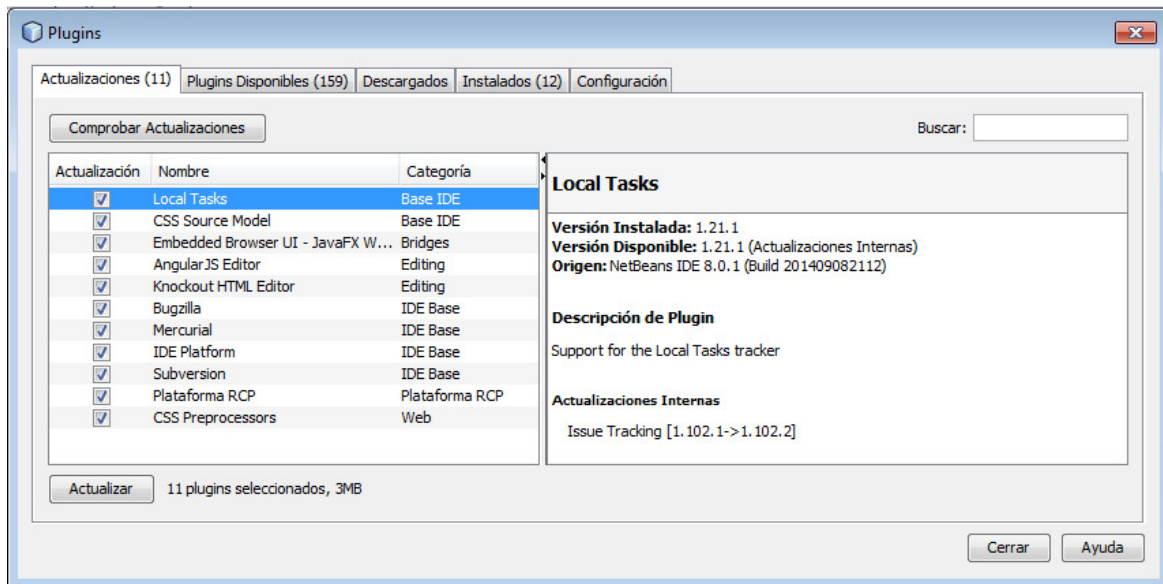
Para comprobar actualizaci3n en calquera momento d3bese utilizar a pestana *Actualizaciones* da vent3 *Plugins* e premer no bot3n *Comprobar Actualizaciones*. Pode ocorrer que segundo a versi3n instalada, non te3a instalado o complemento *Auto Update Service* para facilitar as actualizaci3ns polo que esa pode ser a primeira actualizaci3n que apareza:




De premer no bot3n Actualizar, o instalador informará, como fai sempre, do plugin que se vai a actualizar, e haber3 que confirmar a actualizaci3n. A actualizaci3n 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 encontralas, 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 2.6. Comprobar actualizacións.**

 **Tarefa 2.7. Configurar actualizacións.** Esta tarefa vai consistir en engadir a url <https://github.com/stapler/netbeans-stapler-plugin> ao centro de actualizacións como url activa de referencia para novos complementos ou actualizacións, e configurar as actualizacións para que non as busque de forma automática.

## 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 operación de personalizar o contorno pode facerse basicamente dende dúas opcións do menú principal: Ver e **Herramientas>Opciones**.

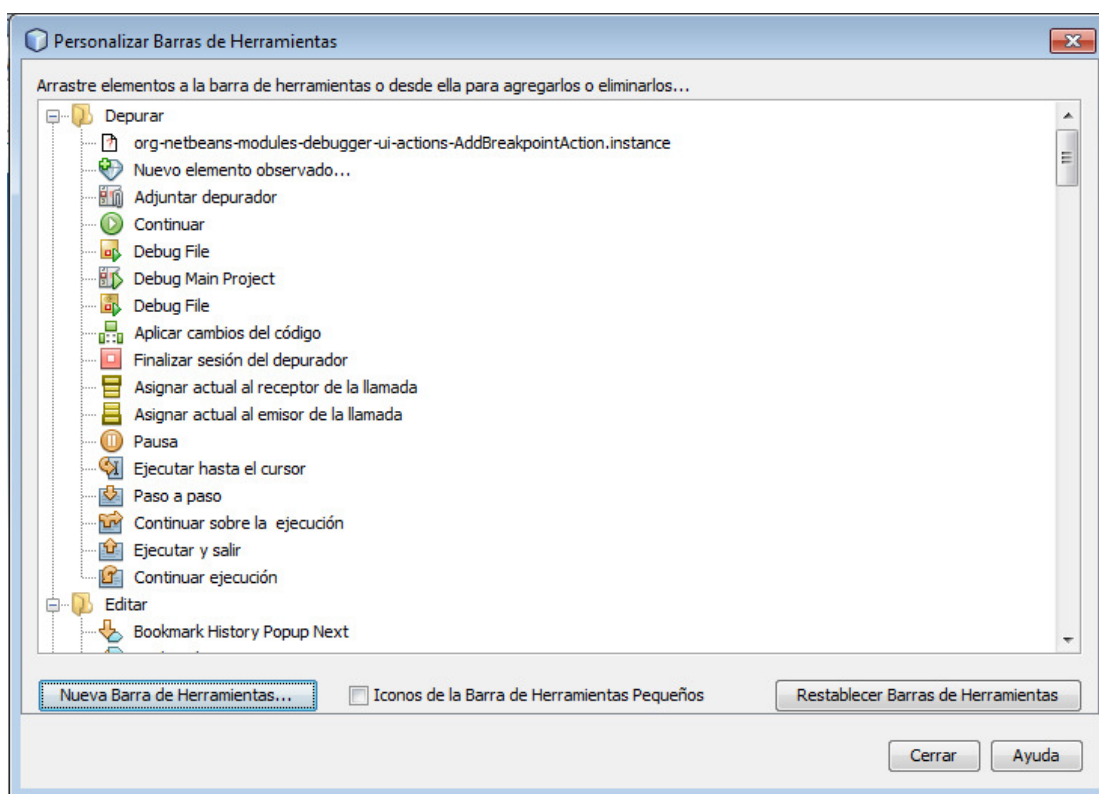
### Opción Ver

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 rato riba da zona punteada da esquerda da barra ata que apareza unha cruz e entón poderá ver o nome da barra.
- Deixar o rato riba da zona punteada da esquerda da barra ata que apareza unha cruz e arrastrar (mantendo pulsado o botón esquerdo do rato) ata chegar á localización onde se quere colocar a barra onde se soltará (deixando de premer o botón esquerdo 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:

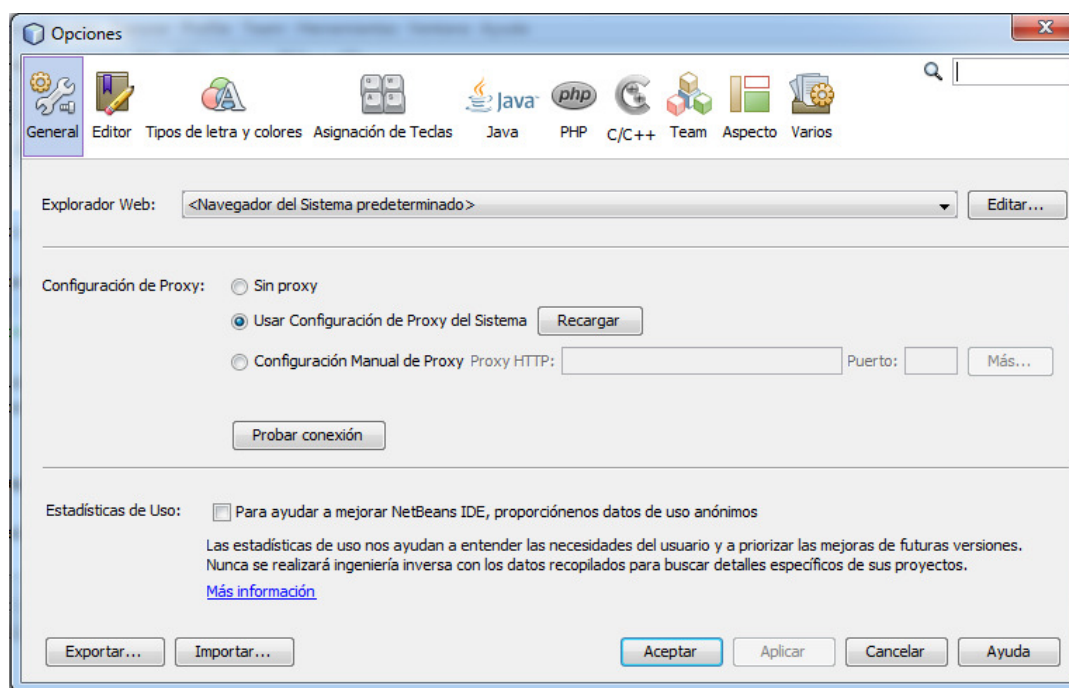
- Crear unha nova barra de ferramentas premendo no botón *Nueva Barra de Herramientas*. Neste caso aparece unha nova ventá na que se debe de dar un nome á nova barra; cando se acepta, aparece a nova barra na zona de barra de ferramentas. A continuación hai que arrastrar e soltar iconas dende a ventá *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 arrastrándooos 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 2.8. 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 *- Disminuír 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 algunhas. 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.

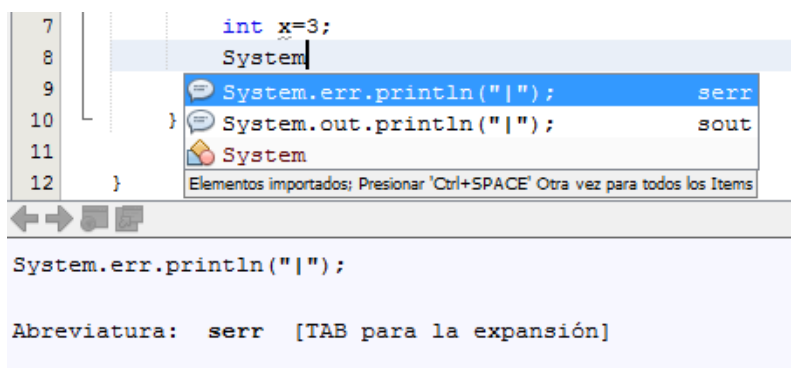
 **Tarefa 2.9. Exportar e importar personalización.** A tarefa vai consistir en realizar de forma secuencial as seguintes operacións: exportar a personalización actual a un arquivo denominado *personaliza1*, definir o navegador que vai a utilizar NetBeans de entre os instalados na máquina, poñer a cor 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* así se reflicte. De todas formas pódese pulsar unha vez Ctrl+barra espaciadora para abrir unha caixa con suxestións das que elixir unha coa que completar o código. Poden volver a pulsarse para ter a suxestión máis detallada.

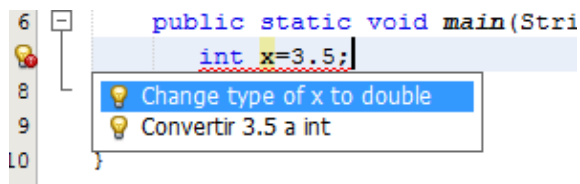




## Tarefa 2.10. Activar completado de código.

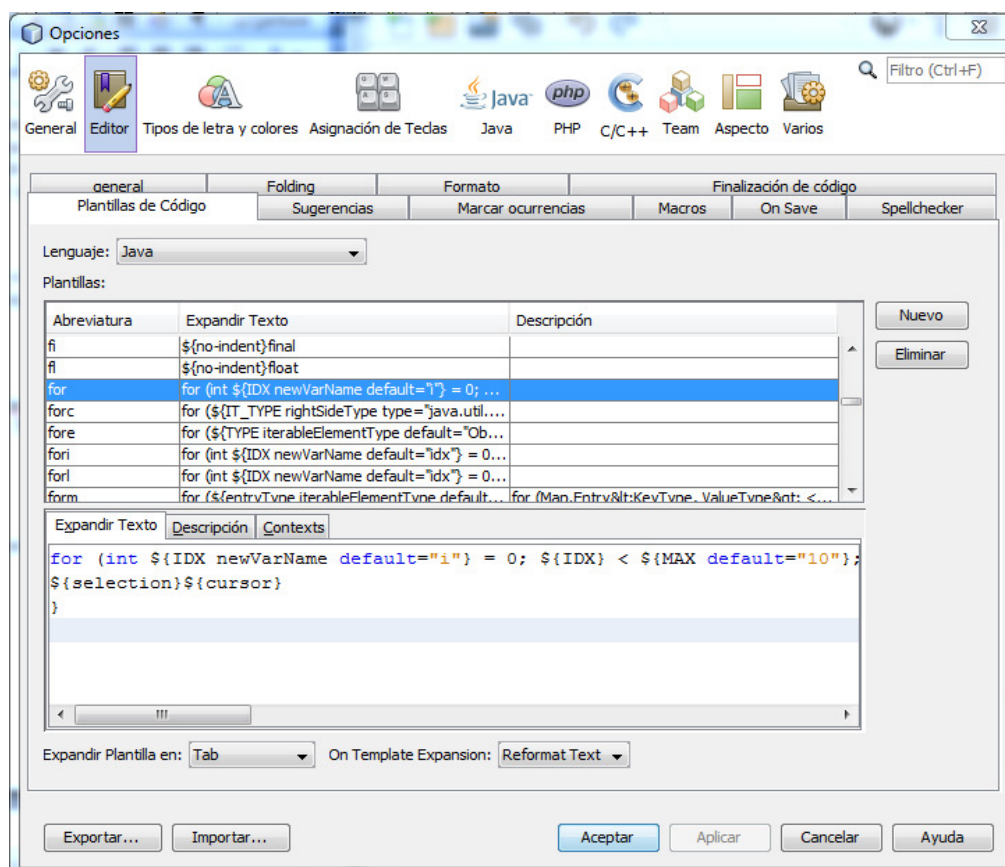
### Suxestións

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*.




### Modelos 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 linguaxe 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 2.11. Utilizar modelo de código existente.



 **Tarefa 2.12. Crear modelo de código.** A tarefa consiste en crear 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>
```


## 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 procedemento para a creación da macro realízase durante a edición de código e consiste en:

- Iniciar a gravación da macro coa icona **Iniciar grabación 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 2.13. 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) {} deixando unha liña en branco entre as chaves e o cursor ao principio dela. Para crear e probar esta macro pode utilizarse o código fonte Java seguinte:

```
import java.util.Scanner;

public class parimpar {
    public static void main(String[] args) {
        short n;
        Scanner teclado = new Scanner(System.in);
        System.out.printf("Teclee o número enteiro entre %d e %d:",
Short.MIN_VALUE, Short.MAX_VALUE);
        n = teclado.nextShort();
        if(n%2==0) {
            System.out.printf("%d é par\n",n);
        }
        else{
            System.out.printf("%d é impar\n",n);
        }System.out.println();
    }
}
```

## 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* mostráranse 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 desexa 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 clic riba dun nome de complemento aparece na parte dereita da ventá información detallada sobre el.



**Tarefa 2.14. Instalar complemento dispoñible no IDE: Rainbow braces.** Este complemento, unha vez instalado, dálle cor aos corchetes, parénteses e chaves, de forma que é máis sinxelo identificar o elementos de cerre cos correspondentes de apertura.

Se o complemento a instalar non está dispoñible nas ULR 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 2.15.** Desde a páxina de plugins de Netbeans, fai una lista cos máis descargados, explicando a súa funcionalidade.

 **Tarefa 2.16. Instala un complemento non dispoñible no IDE:** Darcula LAF for NetBeans (tema escuro), Open containing folder (sobre arquivos abertos no editor, premendo co botón dereito sobre a pestana co seu nome, permite abrir a súa carpeta contenedora no sistema operativo) EasyUML (diagramas de clases UML)

### 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.

 **Tarefa 2.17.** Desactiva complemento dos instalados previamente.

 **Tarefa 2.18. Activar categoría.** A tarefa consiste en activar a categoría PHP. Certas categorías como PHP non están activadas cando se fai unha instalación nova de NetBeans. aínda que isto non significa que non se poidan crear ou utilizar proxectos nesas linguaxes xa que nese caso NetBeans activaríaas automaticamente.

### 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

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 2.19. Instalar extensión en Visual Studio.** A tarefa consiste en instalar as seguintes extensións e comprobar o seu funcionamento: Beautify, Prettier, Guides, Live Server, Bracket Pair Colorizer, Intellicode.

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

### 3.1 Introducción

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


A condición indispensable para que se poida editar, compilar, enlazar e executar código C++ en NetBeans é que o equipo que se estea utilizando **teña instalado un compilador C/C++** e NetBeans estea configurado para poder utilizalo.

O seguinte paso, xa en Netbeans, hai que ir a Plugins, revisar que están activos Centros de actualizacións de Netbeans e **instalar o plugin C/C++**. Logo en Ferramentas > **Opcións > C/C++**. **Add, e engadimos a ruta do compilador de C.**

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 máis cómoda e permite realizar máis operacións que as básicas.

#### Compilador C/C++ e configuración de NetBeans

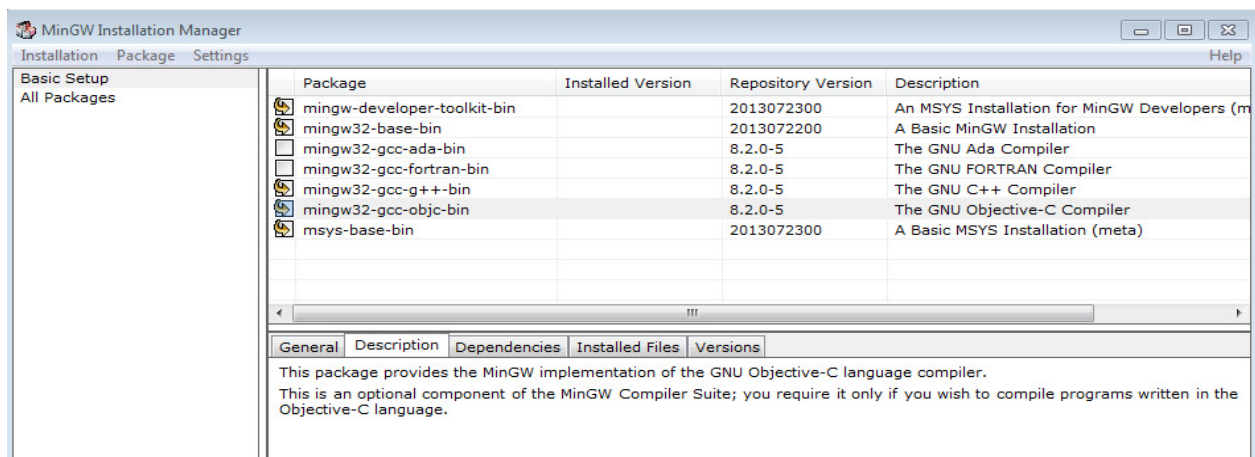
En <https://netbeans.org/community/releases/80/cpp-setup-instructions.html#mingw> pódese consultar información detallada sobre o proceso e ademais hai enlaces para descargar o compilador.

 **Tarefa 2.20. Instalar compilador C/C++ e configurar NetBeans para que poida crear proxecto en C++.** Descargar MinGW e instalalo en c:\. Marcar os paquetes developer-toolkit-bin, base-bin, gcc-g++-bin, e gcc-objc-bin.

E prememos no menú superior Installation > Apply changes. Logo hai que Engadir á variable de entorno PATH a ubicación do cartafol "bin" do compilador.

En Netbeans > Plugins, revisar que están activos Centros de actualizacións de Netbeans e instalar o plugin C/C++. Logo en Ferramentas > Opcións > C/C++. Add, e engadimos a ruta do compilador de C.






## Crear proxecto

Para crear un proxecto C++ que dará lugar a unha aplicación sinxela hai que:

- Elixir *Archivo->Proyecto nuevo*.
- Seleccionar a categoría: *C/C++* e o tipo de proxecto *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 desexa 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 da carpeta 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 2.21 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 máis 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 construción 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 erros no proceso de compilación ou enlace, non se xerará o executable e a información sobre eles aparecen na pestana de construción e execución.

Se non hai erros 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 dobre clic sobre o executable. Se a aplicación é de consola, abrírase a ventá de comandos ou consola do sistema e executaranse as instrucións 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.



**Tarefa 2.22** 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 cumpra 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, procésaaas, e envía os resultados ao motor de PHP, ou un código de error, se é que non puido 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.

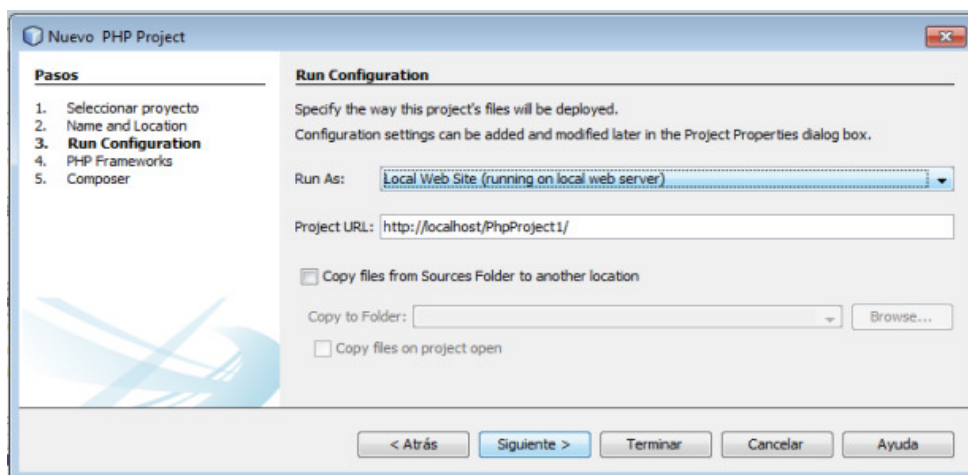


#### Tarefa 2.23. Instalar un servidor XAMPP en local.

- Descargamos XAMPP en 32bit ou 64bits, na súa versión portable e descargámola sobre c:\ (ten que ser un cartafol raíz, c:\xampp, d:\xampp, etc)
- Para arrancar o servidor executamos: *xampp-control.exe* e iniciamos Apache e MySql con botón *Start*. Na propia máquina podemos abrir *http://localhost* para comprobar que funciona.
- Os arquivos 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, clicamos *Stop* no panel de control e *Quit* (*xampp-control.exe*)
- O cartafol onde depositaremos os documentos php,html, etc. é *htdocs*.
- Ademáis do servidor, dispoñemos da ferramenta web de xestión da BD: *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 indicárase 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 desexa 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* (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 2.24. 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 e segue as instrucións para, desde phpmyadmin, crear a base de datos e táboas necesarias en MySQL.
- Comproba 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 execútaa.



**Tarefa 2.25. Editar proxecto PHP en NetBeans. (Ver anexo ao final do tema).**

- Realiza os cambios precisos no código PHP da aplicación entregada para que o contrasinal almacenada no ficheiro estea encriptada.
- Realiza os cambios precisos no código PHP para que o usuario poda cambiar o contrasinal.
- Crea código JavaScript para que, cando o usuario introduza o novo contrasinal dúas veces, ambas sexan iguais e teñan una lonxitude mínima de 4 caracteres.
- Mellora a presentación da aplicación incorporando melloras HTML, CSS e BootStrap.

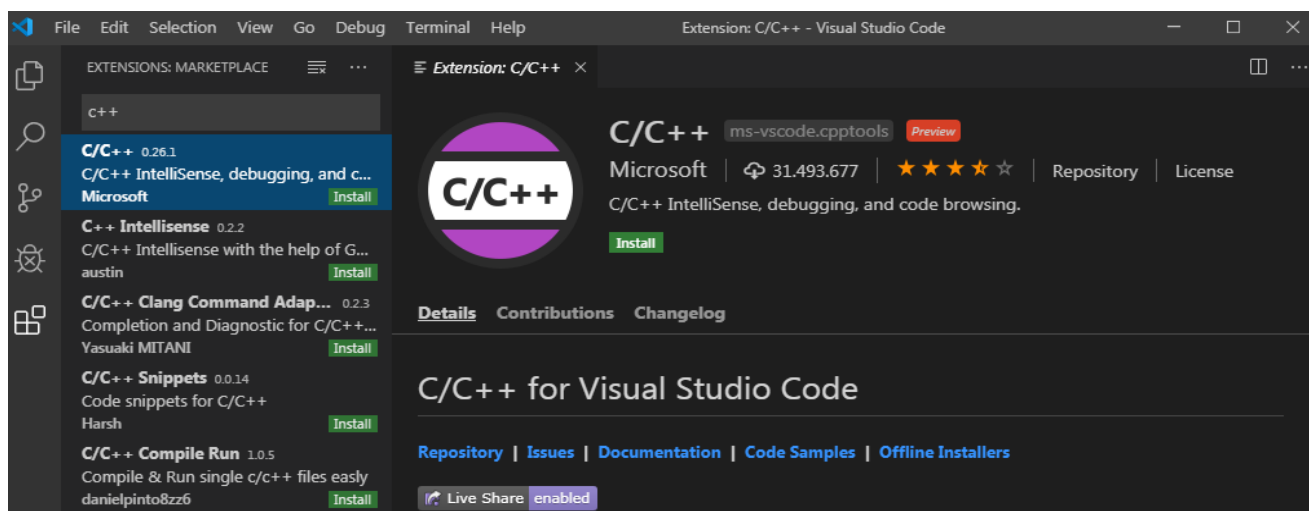
## 3.4 Proxecto Visual C++ en VisualStudio Code

Para a edición e execución de proxectos en C e C++ desde Visual Studio Code e preciso instalar o compilador de C/C++, instalar extensións no IDE e crear uns arquivos de configuración como detallaremos a continuación.

### Instalación

Para desenvolver proxectos en C e C++ precisamos realizar as seguintes tarefas:

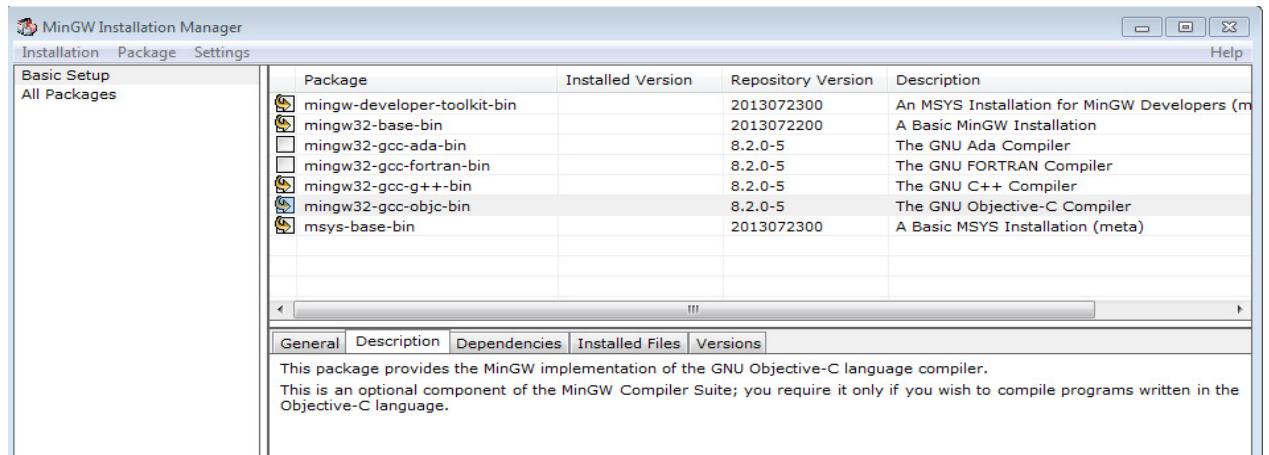
- En Visual Studio Code, instalar a extensión: *C++ extension for VS Code*. Premendo á esquerda na icona de *Extensions*, buscamos *C/C++* e instalamos.



- Instalar un compilador de C. Por compatibilidade cunha extensión que veremos no apartado seguinte (Easy C++ projects) instalaremos as seguintes versións do compilador:

- GCC Windows 10: Install GCC and Make on Windows through WSL: [Windows Subsystem for Linux setup](#)
- GCC Windows 8.1 or lower: Install GCC and Make using [Cygwin](#) or [MinGW](#)

No noso caso, na máquina virtual instalamos MinGW sobre C:\. Marcamos os paquetes a instalar:



E prememos no menú superior *Installation > Apply changes*.

c) Engadir á variable de entorno PATH a ubicación do cartafol *"bin"* do compilador (será *C:\MinGW\bin*).

d) Para comprobar que a instalación foi correcta, desde unha consola: `gcc --version`

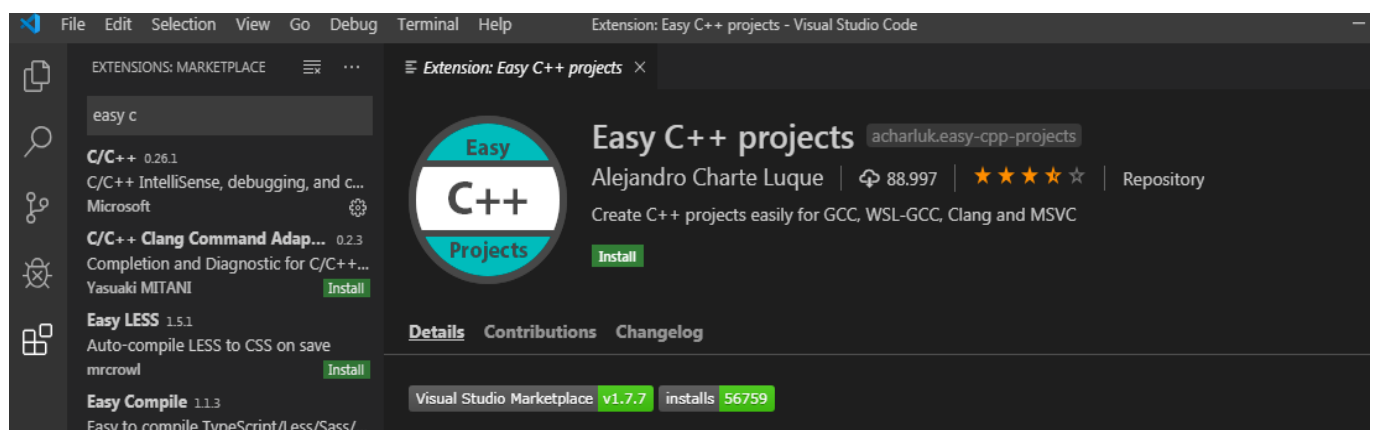
## Configuración

Se facemos a configuración pola nosa conta debemos crear/editar varios arquivos de configuración: ruta do compilador, a configuración da tarefa de compilación (*Build task*), a configuración para a depuración de programas (*debug settings*), etc.

Estes pasos podemos velos en detalle nesta ligazón: <https://code.visualstudio.com/docs/cpp/config-mingw>

Para evitar todo este proceso podemos instalar en Visual Studio unha extensión que fai todo este traballo por nós: Easy C++ Projects. Esta é a ligazón de instalación:

<https://marketplace.visualstudio.com/items?itemName=ACharLuk.easy-cpp-projects>

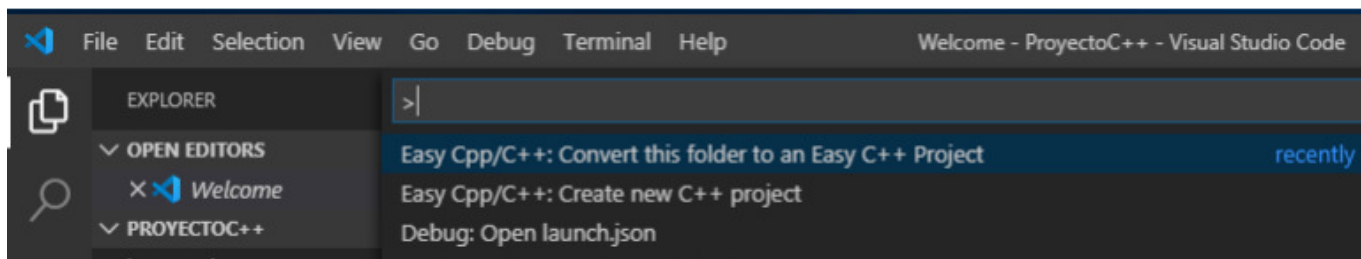


## Crear e executar proxectos

Para crear un proxecto en C++ primeiro debemos crear un cartafol para el mesmo, y en Visual Studio, *File > Open Folder*, abrimos ese cartafol.

A continuación, desde menú *View >> Command Palette*: tecleamos *easy cpp* y presionamos *"Easy Cpp/C++: Convert this folder to an Easy C++ Project"*.

Con este paso crease un cartafol chamado *.vscode* para a configuración no noso proxecto.



Agora deberíamos, desde a paleta de comandos, executar os comandos precisos para a edición dos ficheiros que levan a configuración para a compilación e execución do proxecto. No noso caso, para aforrar tempo, ya os temos listos, so temos que movelos á carpeta *.vscode*. Estes serían os comandos e os arquivos:

- C/C++: Edit Configurations (UI) : xerando o arquivo **c\_cpp\_properties.json**, no que indicamos á ruta do compilador (C:\MinGW\bin\gcc.exe), ModointelliSense gcc-x64.

```
.vscode > {} c_cpp_properties.json > ...
1  {
2      "configurations": [
3          {
4              "name": "Win32",
5              "includePath": [
6                  "${workspaceFolder}/**"
7              ],
8              "defines": [
9                  "_DEBUG",
10                 "UNICODE",
11                 "_UNICODE"
12             ],
13             "compilerPath": "C:\\MinGW\\bin\\gcc.exe",
14             "cStandard": "c11",
15             "cppStandard": "c++17",
16             "intelliSenseMode": "gcc-x64"
17         }
18     ],
19     "version": 4
20 }
```

- Configure Default Build Task → Create tasks.json file from template → Others: Example to run an arbitrary external command, xerando o arquivo **tasks.json**

Na liña 10 poñeríamos o nome do programa co código fonte "programa.cpp" e xusto antes o nome que terá o executable, pero sin a extensión ".exe", no noso caso "programa".

```
.vscode > {} tasks.json > ...
1  {
2      // See https://go.microsoft.com/fwlink/?LinkId=733558
3      // for the documentation about the tasks.json format
4      "version": "2.0.0",
5      "tasks": [
6          {
7              "label": "echo",
8              "type": "shell",
9              "command": "g++",
10             "args": ["-g", "-o", "programa", "programa.cpp"],
11             "group": {
12                 "kind": "build",
13                 "isDefault": true
14             }
15         }
16     ]
17 }
```

- Debug: Open launch.json → Select environment: C++ (GDB/LLDB), xerando o arquivo **launch.json**

```
.vscode > {} launch.json > ...
1  {
2      // Use IntelliSense to learn about possible attributes.
3      // Hover to view descriptions of existing attributes.
4      // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
5      "version": "0.2.0",
6      "configurations": [
7          {
8              "name": "(gdb) Iniciar",
9              "type": "cppdbg",
10             "request": "launch",
11             "program": "${workspaceFolder}/programa.exe.",
12             "args": [],
13             "stopAtEntry": false,
14             "cwd": "${workspaceFolder}",
15             "environment": [],
16             "externalConsole": false,
17             "MIMode": "gdb",
18             "miDebuggerPath": "C:\\MinGW\\bin\\gcc.exe",
19             "setupCommands": [
20                 {
21                     "description": "Habilitar la impresión con sangría para gdb",
22                     "text": "-enable-pretty-printing",
23                     "ignoreFailures": true
24                 }
25             ]
26         }
27     ]
28 }
```

Unha vez rematada a configuración (no noso caso simplemente copiar os tres arquivos a .vscode) creamos no cartafol do proxecto (non no .vscode) un programa en C++ como o do exemplo, e executamos **Ctrl** + **Mayusc** + **B** para construír a solución (Build Solution).

Con este último paso obteremos o executable que podemos probar, tanto desde o contorno Visual Studio como desde o sistema operativo.

## 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 máis 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” o descomentalas premendo “Ctrl+U”
- **Buscar e abrir arquivos:** “Ctrl + T”
- **Quick Fix.** “Ctrl + .” Para resolver erros no código, por exemplo facer as importacións das clases en Java.
- **Múltiples cursores:** control click, “esc” para volver ao cursor simple.
- **Eliminar tabuladores e espazos del final:** Un dos trucos máis fáciles, sinxelos, rápidos e eficaces que podes aplicar para eliminar espazos, tabulacións ou liñas en branco que polo xeral son innecesarios, é premendo “Ctrl+E” o “Ctrl+o”. Isto evitará que teñas que premer unha e outra vez a tecla “Supr” para borrarlos.

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 ditos espazos por puntos.

- **Preparar un comentario enriba da liña actual:** “Ctrl+Enter”.
- **Copiar a liña actual: enteira:** colocar o cursor enriba da liña que queres copiar e premer “Ctrl+C” sen ter que seleccionar nada.
- **Seleccionar varias liñas de forma vertical:** “Alt” + arrastrar.
- **Xerar código** (como getters, setter, etc...): botón dereito sobre o código > Source Action
- **Buscar en varios arquivos** arquivos: “Ctrl+F” ou “Ctrl+Shift + F” incluíndo 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.
- **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)





**Tarefa 2.26. Crear proxecto Visual C++ en Visual Studio.** 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 C++
- Editar o arquivo programa.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. O código pode ser como o que se mostra a continuación.
- Mostrar as utilidades de edición que ofrece o IDE descritas previamente.
- Compilar e executar o proxecto en Visual Studio.
- Para gravar e editar o vídeo podes empregar unha ferramenta gratuíta como Active Presenter.

```
#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";
```

## 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.

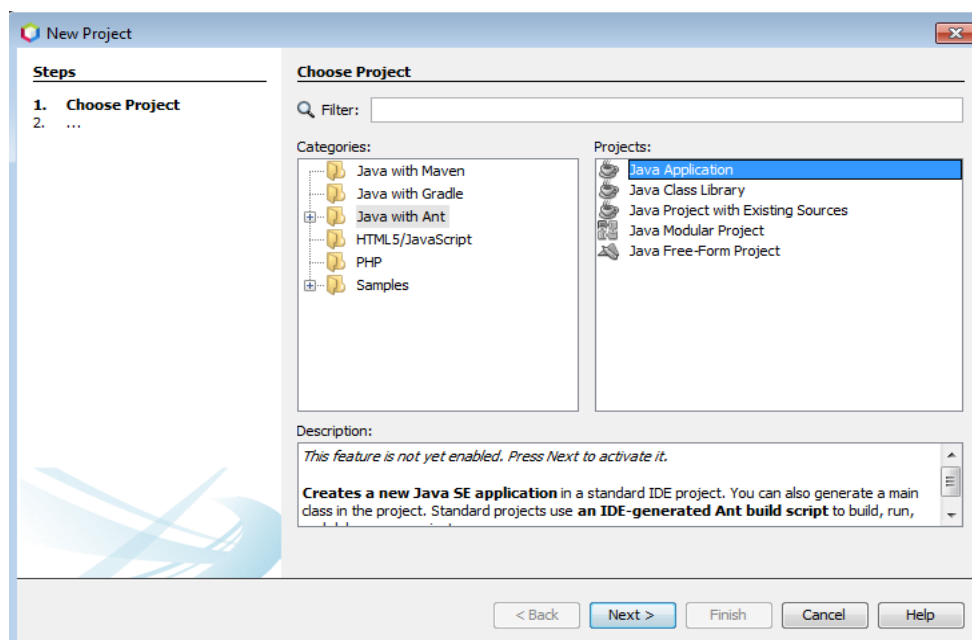
NetBeans facilita e automatiza as operacións a realizar sobre o código Java pero este non é o único procedemento para realizalas xa que a edición podería facerse cun editor de texto calquera, a compilación podería facerse dende a liña de comandos co compilador *javac* e as ferramentas de construción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

### Crear proxecto

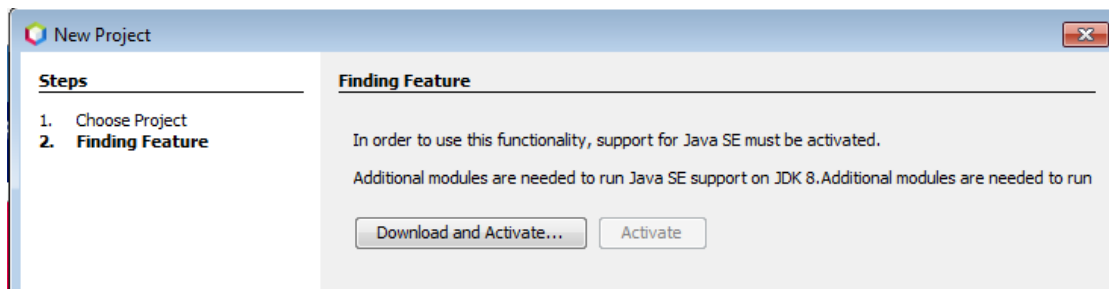
A creación dun proxecto Java que dará lugar a unha aplicación sinxela de consola é similar á creación dun proxecto noutra linguaxe. Unha diferenza propia de Java é por exemplo a utilización de paquetes: por defecto crearase un paquete Java co mesmo nome do proxecto e a clase principal chamarase como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

Para crear o proxecto: Menú File > New Project.

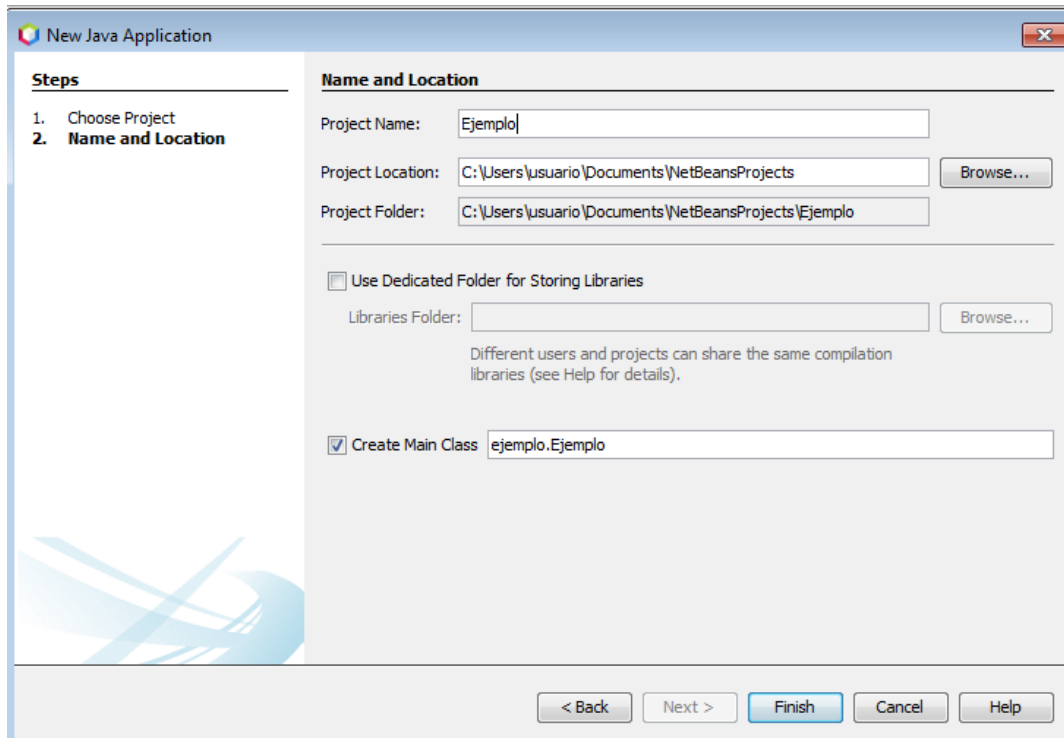
Aquí temos varias opcións, en función de como desexamos que se xestionen as dependencias (proxecto Maven, Gradle ou Ant). Por agora, non é importante para nós, podemos elixir *Java with Ant > Java Application*.



Se é a primeira vez, pedirá descargar e activar os módulos necesarios para soportar esa linguaxe.



Asignámoslle 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 “contenedor” 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 “contenedores” que queiramos e, por suposto, cada “contenedor” pode ter as clases que queiramos.


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

## Editar código fonte

NetBeans dispón de axudas en liña para escribir código como por exemplo: completado de código, consellos en liña e suxestións. Recoméndase utilizalas 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 2.27.** 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 enteiro pequeno e visualice se o número é primo ou non.

- Un algoritmo sinxelo para saber se un número é primo consiste en ir dividindo o número entre os números comprendidos entre 2 e o anterior ao número. Se algunha das divisións dá de resto 0 é que o número non é primo. Se as divisións nunca dan 0 é que o número é primo.
- Sobre o código anterior, proba os diferentes atallos de teclado que ten Netbeans.

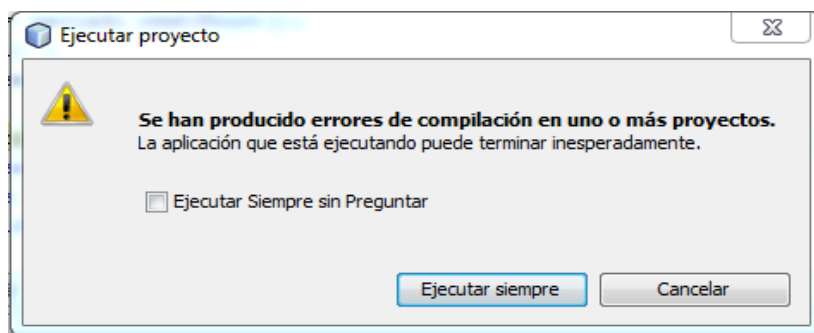
Zoom in y out	Alt + Roda rato.
Indentación do código	Alt + Shift + F - Boton dereito do rato > Format
Generar codigo	Alt + Insertar
Añadir líneas de comentarios y viceversa	Ctrl + Shift + C
Arreglar todas las importaciones de clase	Ctrl - Shift - I - Botón dereito do rato > Fix Imports
Corregir la importación de la clase seleccionada	Alt + Shift + I
Desplazar líneas a la izquierda / derecha	Alt + Shift + ← / Alt + Shift + →
Cambiar líneas hacia arriba / abajo	Alt + Shift + ↑ / Alt + Shift + ↓
Alternar selección rectangular	Ctrl + Shift + R
Copiar líneas	Ctrl + Shift + ↑
Copiar líneas hacia abajo	Ctrl + Shift + ↓
Eliminar linea actual	Ctrl + E
Inspeccionar miembros	Ctrl + F12
Inspeccionar jerarquía	Alt + F12
Código Completo	Ctrl + Espacio
Rebautizar	Ctrl + R

## Compilar e executar

A forma máis 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 arquivos (clases) con un 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 emerxente. De elixir na ventá anterior que se quere continuar coa execución, aparece na ventá *Salida* a información sobre os erros encontrados.



Se non hai erros 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 podería 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 2.28. Realiza as seguintes operación con proxecto anterior:

- Compila o proxecto anterior Java en NetBeans e executao no contorno Netbeans.
- Executao desde fóra de NetBeans (desde a liña de comandos, sobre o arquivo.class)
- Xenera un .jar, modificar o pom.xml e executao tamén desde liña de comandos.

## 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 unha clase soamente, dun proxecto noso, podemos simplemente copiala e pegala aínda 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 é noso debemos **facér 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**. Pestaña “Open projects” e incluimos 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 2.29. Crea un novo proxecto e incopora o .jar do proxecto anterior e usa no main do novo proxecto algunha das clases que teña o proxecto anterior.

## 5. Anexo:Práctica PHP en Netbeans

---

*Para as tarefas 2.24 e 2.25*

### 5.1 Descripción de la aplicación

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/desencryptado.

### 5.2 Instalación

Para instalar la aplicación descomprimir el .zip entregado sobre htdocs o, mejor, en una subcarpeta de htdocs (por ejemplo: appPHP)

A continuación, es necesario crear en MySQL la base de datos "practicas", un usuario llamado 'pepe' (contraseña 'pepa'), 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. Pestaña *Base de datos*. *Nombre de la base de datos*: practicas. Botón *Crear*.
3. Pestaña *Privilegios*. Agregar cuenta de usuario: pepe/localhost/pepa.
4. Menú lateral > BD practicas. Pestaña: *privilegios*. Pepe > *Editar privilegios*. Seleccionar todo y pulsar *Continuar*
5. Menú lateral > BD practicas. Pestaña: *Importar*. Seleccionar el archivo .sql para crear la tabla e insertar los datos, y pulsar *Continuar*.

## 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, y trabajar sobre la nueva carpeta para poder, en caso de problemas, volver a la situación inicial.
2. Añadir al archivo 2\_menu.php una nueva opción de menú para cambiar la contraseña:

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

3. Crear el documento con el formulario 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>
```

4. Crear el script en PHP 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>
```

5. 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");
```

6. Comprobar que acepta la nueva contraseña y verificar que la contraseña está encriptada en el archivo *password.txt*.

## 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>` la función JavaScript de validación.

```
<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>
```

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 desees 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.

## 5.6 Modificaciones en BootStrap

Se trata de modificar las páginas (o al menos alguna de ellas) para presentarlas con el formato BootStrap. Para ello haremos los siguientes cambios en las páginas.

1.- Introducir la plantilla BootStrap en los documentos que quieras modificar.

```
<!doctype html>
<?php include '1_validar_sesion.php';?>
<html lang="es">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
```

```

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<title>Menú principal</title>
</head>
<body>
    CONTENIDO AQUÍ
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/njGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
</body></html>

```

2.- Mostrar los mensajes con clases Bootstrap. Por ejemplo:

```
<a class="alert alert-success" . . .
```

3.- Mostrar los botones con clases Bootstrap, por ejemplo:

```
<a type="button" class="btn btn-success" href="/index.php">Volver al menú</a>
```

4.- Modificar el menú principal para que se muestre como una barra de navegación:

```

<nav class="navbar navbar-expand-sm navbar-dark bg-dark">
    <a class="navbar-brand" href="#">Menú</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbar1">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbar1">
        <div class="navbar-nav">
            <a class="nav-item nav-link active" href="/3_crearexamen.php">Examen PDF</a>
            <a class="nav-item nav-link dropdown-toggle"
data-toggle="dropdown" href="#"> Preguntas<span class="caret"></span></a>
            <div class="dropdown-menu">
                <a class="dropdown-item" href="/3_verpreguntas.php">Ver</a>
                <a class="dropdown-item" href="/3_anadirpregunta.php"> Añadir</a>
                <a class="dropdown-item" href="/3_borrarpregunta.php">Borrar</a>
            </div>
            <a class="nav-item nav-link" href="/3_desconectar.php">Desconectar</a>
            <a class="nav-item nav-link" href="/3_cambiarpass.php">Cambiar Pass</a>
        </div>
    </div>
</nav>

```

5.- Comprueba cómo se contrae la barra de navegación si hacemos la ventana del navegador más estrecha, simulando un dispositivo pequeño.