



AspectJ

Tutorial AspectJ

Diego Pastás
Miguel Peña
Sebastián Camacho

¿QUÉ ES ASPECTJ?

- Es una extensión orientada a aspectos para el lenguaje Java
- Fue creado en 1998 año en Xerox PARC
- Cualquier programa válido en Java, es un programa válido en AspectJ
- Conceptos base para el funcionamiento de un sistema son escritos en Java
- Conceptos transversales a todo el programa son escritos en AspectJ



¿CÓMO INSTALO ASPECTJ?

- Vamos a la página oficial de descargas de AspectJ:



TUTORIAL CONFIGURACIÓN INICIAL ASPECTJ

1. Al entrar a la página escogemos la versión que necesitamos (por ejemplo, la última versión estable). Descargamos el ejecutable .jar (aspectj-1.x.x.jar)

Downloads

See below for a list of available AspectJ downloads. You probably want one of: the most recent development build, the most recent milestone build, or the latest stable release. Installation instructions and more information on IDE support can be found at the bottom of this page.

Most Recent Build

- **Last Known Good developer build**

AspectJ Compiler, Browser, Ant tasks, and Documentation. Only download this version if you are prepared to work with a pre-release compiler. The most recent stable build below is the currently supported release version.

aspectj-DEVELOPMENT-201901140838.jar (size: 17826665 bytes)

Changes in this build

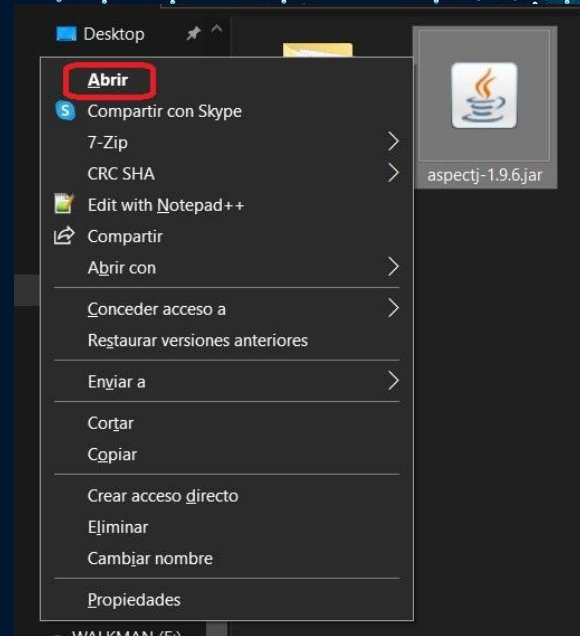
This download is updated after every successful build and test cycle.

Latest Stable Release

- **AspectJ 1.9.6, Released 22 Jul 2020**

AspectJ compiler, browser, documentation tool, Ant tasks, and documentation.

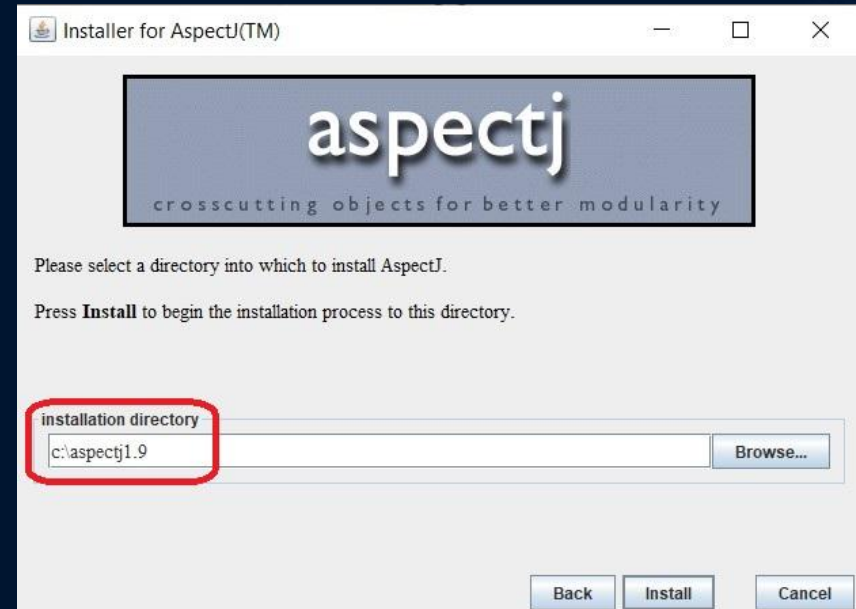
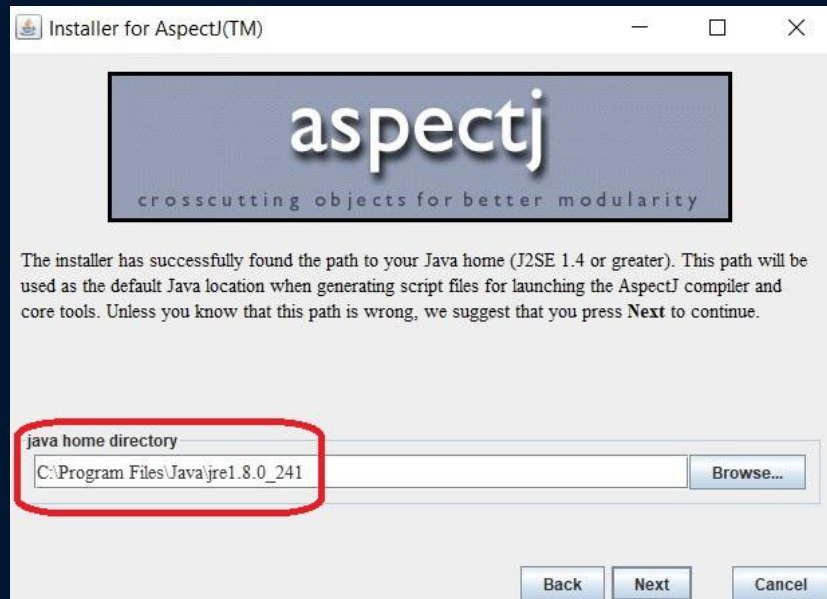
aspectj-1.9.6.jar (~17M)
aspectj-1.9.6-src.jar



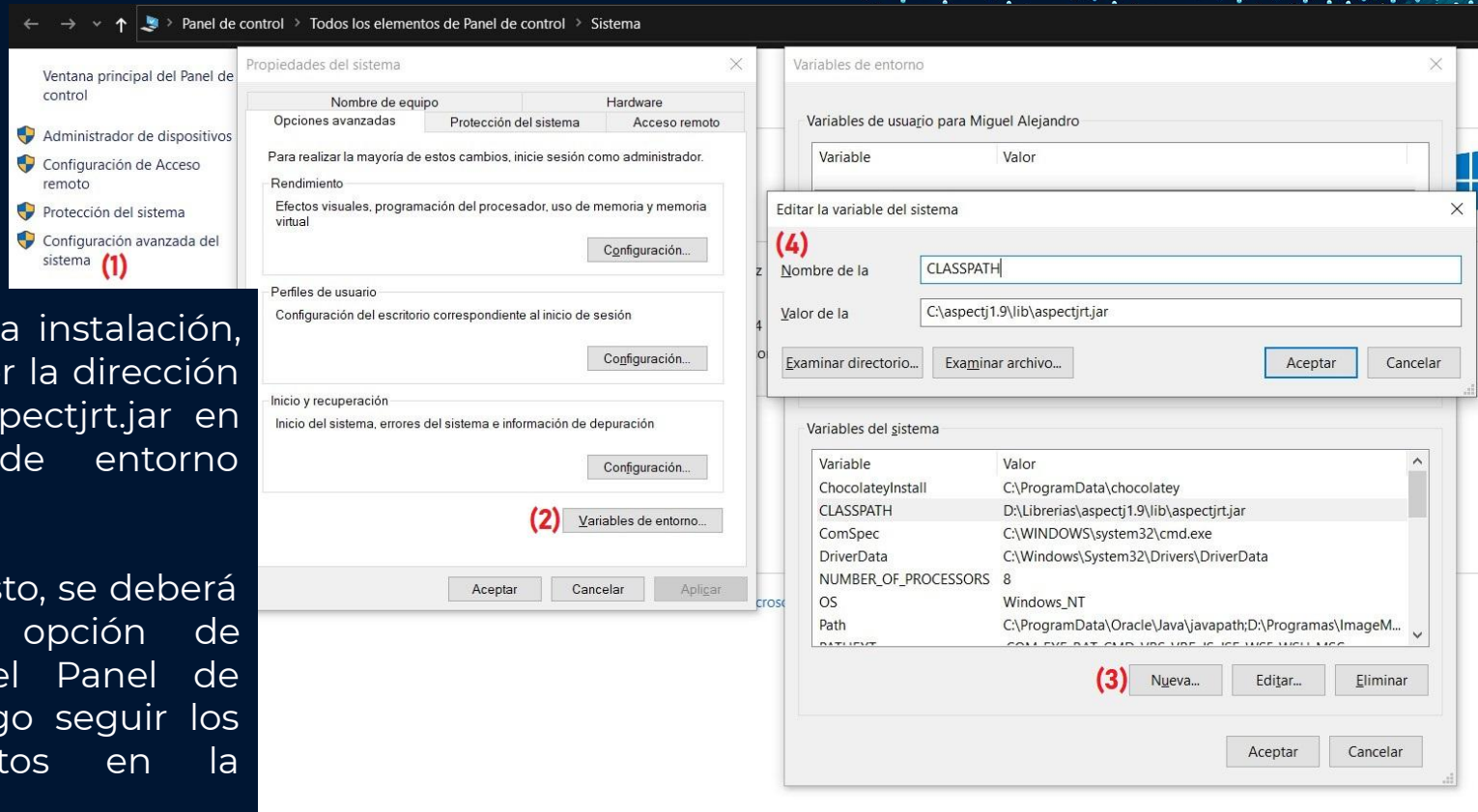
2. Vamos a la carpeta donde se descargó el jar, y lo ejecutamos

TUTORIAL CONFIGURACIÓN INICIAL ASPECTJ

3. Durante la ejecución del programa, se define la dirección donde está instalado java, y luego la dirección en la cual se instalarán las librerías de AspectJ



TUTORIAL CONFIGURACIÓN INICIAL ASPECTJ

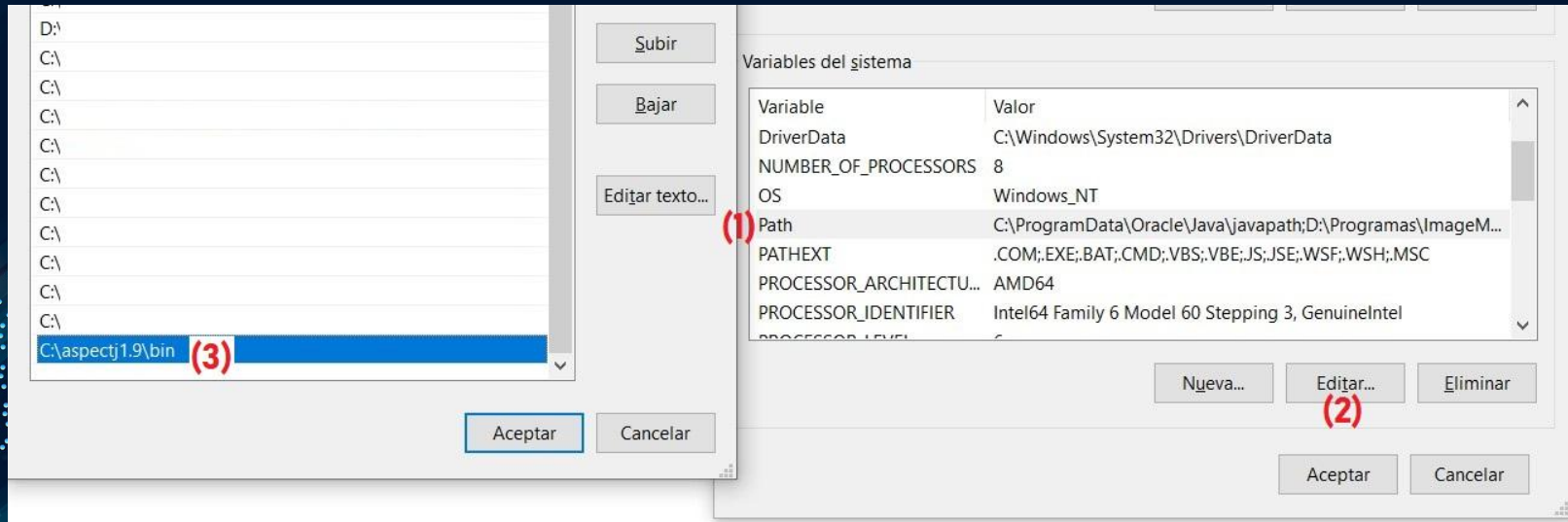


4. Al finalizar la instalación, se pedirá poner la dirección del archivo aspectjrt.jar en la variable de entorno CLASSPATH.

Para realizar esto, se deberá llegar a la opción de Sistema en el Panel de Control, y luego seguir los pasos descritos en la imagen.

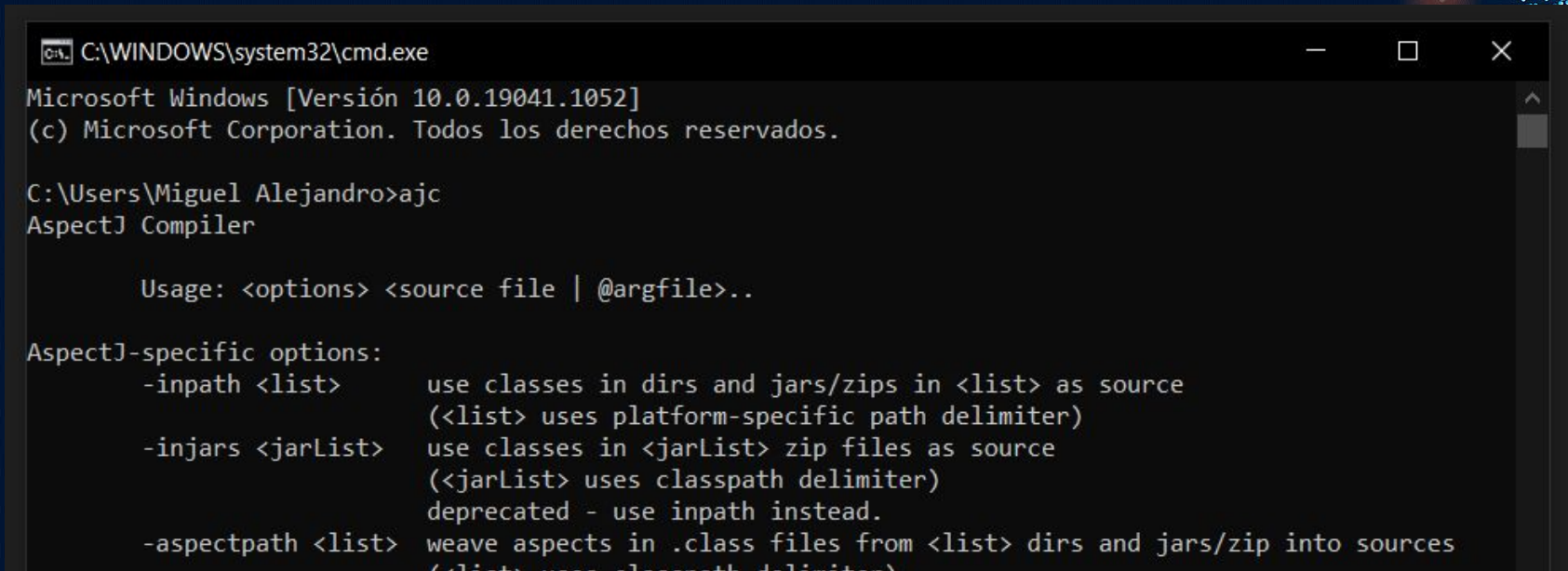
TUTORIAL CONFIGURACIÓN INICIAL ASPECTJ

5. Luego, es necesario agregar la carpeta bin del lugar donde se instaló AspectJ (en este caso en C:\aspectj1.9\bin) a la variable Path, para lo cual se edita como se muestra en la imagen



TUTORIAL CONFIGURACIÓN INICIAL ASPECTJ

6. Podemos comprobar que fue instalado correctamente al abrir Símbolo del Sistema (cmd.exe) y escribir "ajc" en la consola, el cual es el compilador para AspectJ. Deberá mostrar la configuración del compilador.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19041.1052]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Miguel Alejandro>ajc
AspectJ Compiler

Usage: <options> <source file | @argfile>..

AspectJ-specific options:
-inpath <list>      use classes in dirs and jars/zips in <list> as source
                    (<list> uses platform-specific path delimiter)
-injars <jarList>  use classes in <jarList> zip files as source
                    (<jarList> uses classpath delimiter)
                    deprecated - use inpath instead.
-aspectpath <list> weave aspects in .class files from <list> dirs and jars/zip into sources
                    (<list> uses classpath delimiter)
```


¿EN CUÁLES IDE'S PUEDO INTEGRAR ASPECTJ?

ECLIPSE IDE



INTELLIJ IDEA



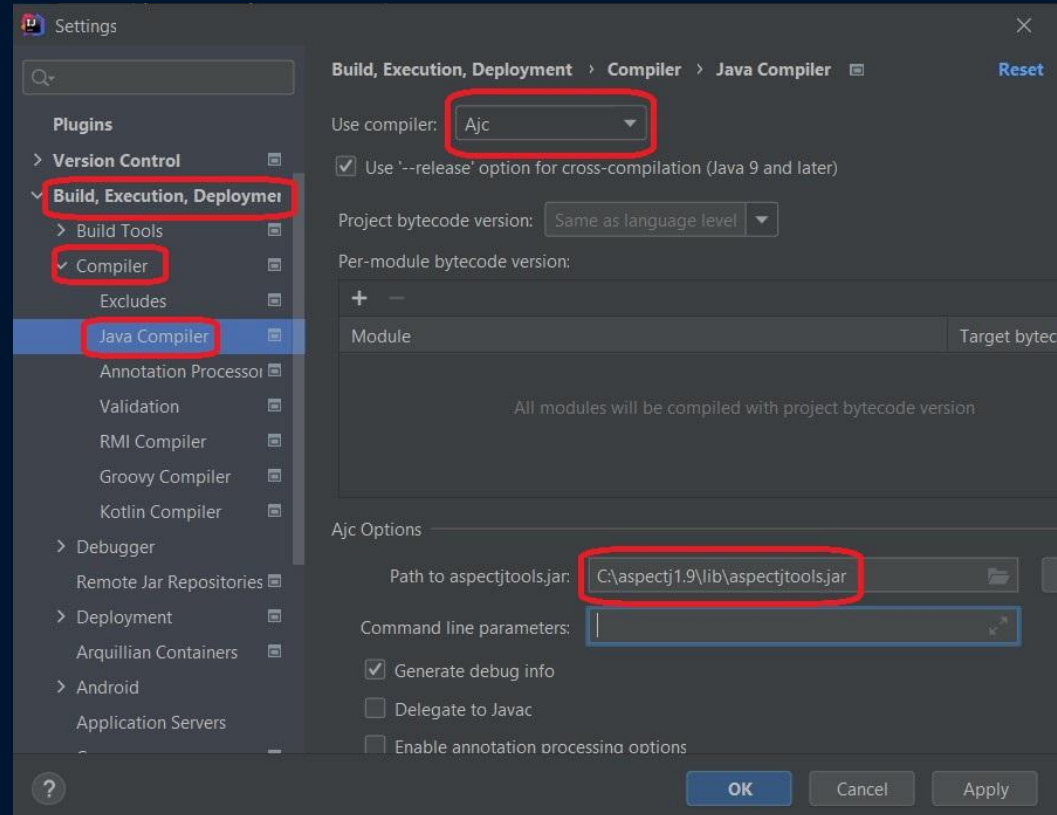
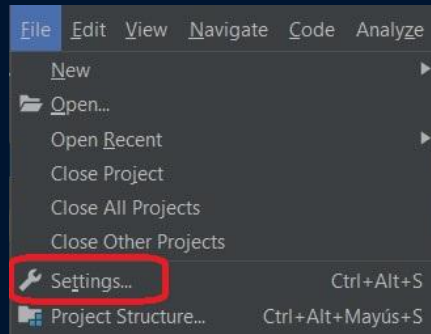
NETBEANS IDE



TUTORIAL CONFIGURACIÓN PROYECTOS INTELLIJ IDEA

1. Configuración del compilador:

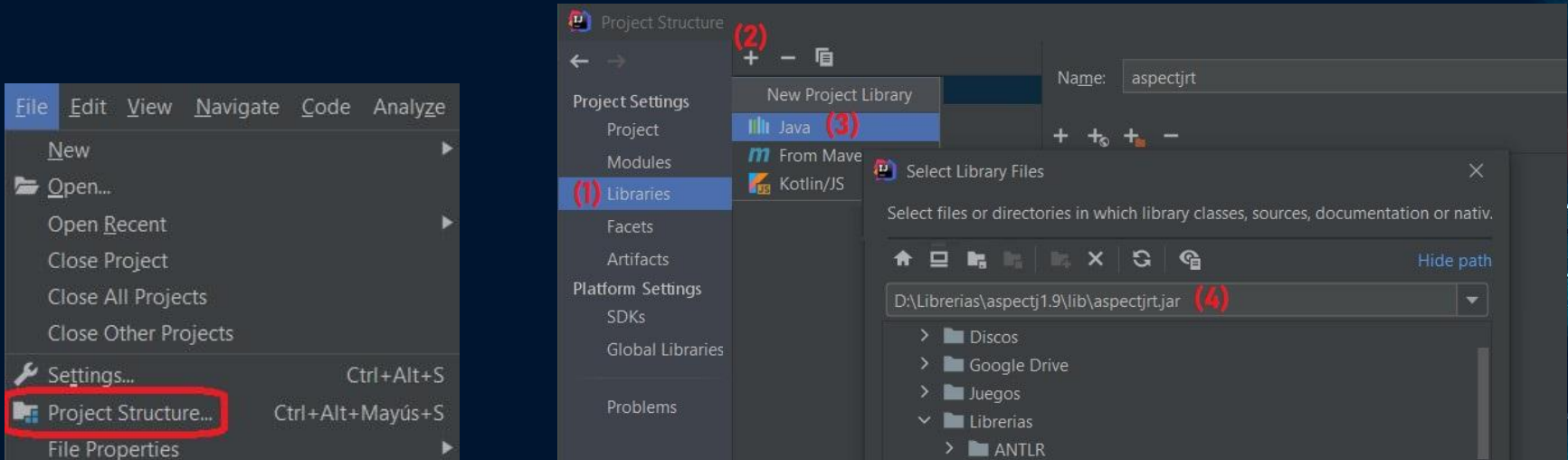
- Se va a la configuración en File -> Settings
- Luego se entra a Build, Execution, Deployment -> Compiler -> Java Compiler
- Se cambia el compilador por Ajc (compilador de AspectJ), y se indica la ruta donde está el archivo aspectjtools.jar, el cual está en la carpeta lib del directorio de instalación de AspectJ



TUTORIAL CONFIGURACIÓN PROYECTOS INTELLIJ IDEA

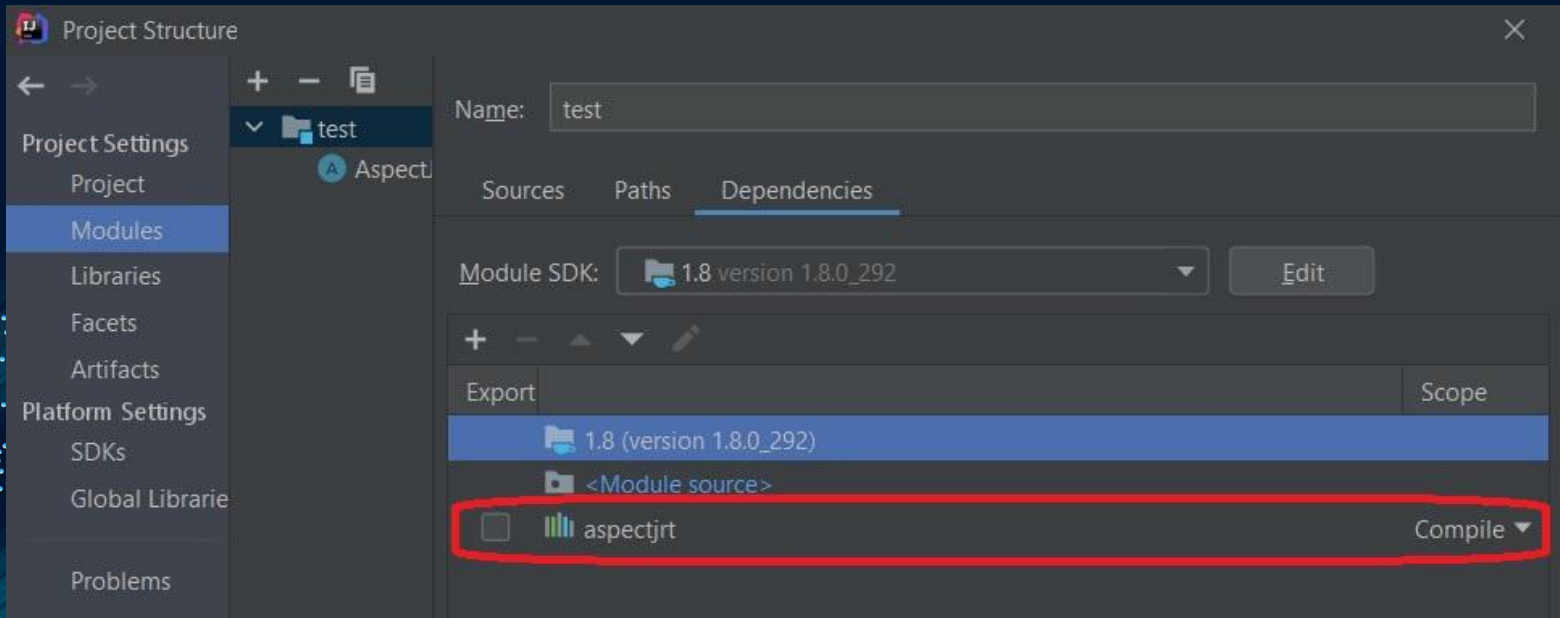
2. Configuración del Proyecto:

- Se va a la configuración del proyecto en File -> Project Structure
 - a. En la pestaña librerías, se importa la referencia a librería aspectjrt.jar, la cual se encuentra en la carpeta lib de la carpeta de instalación



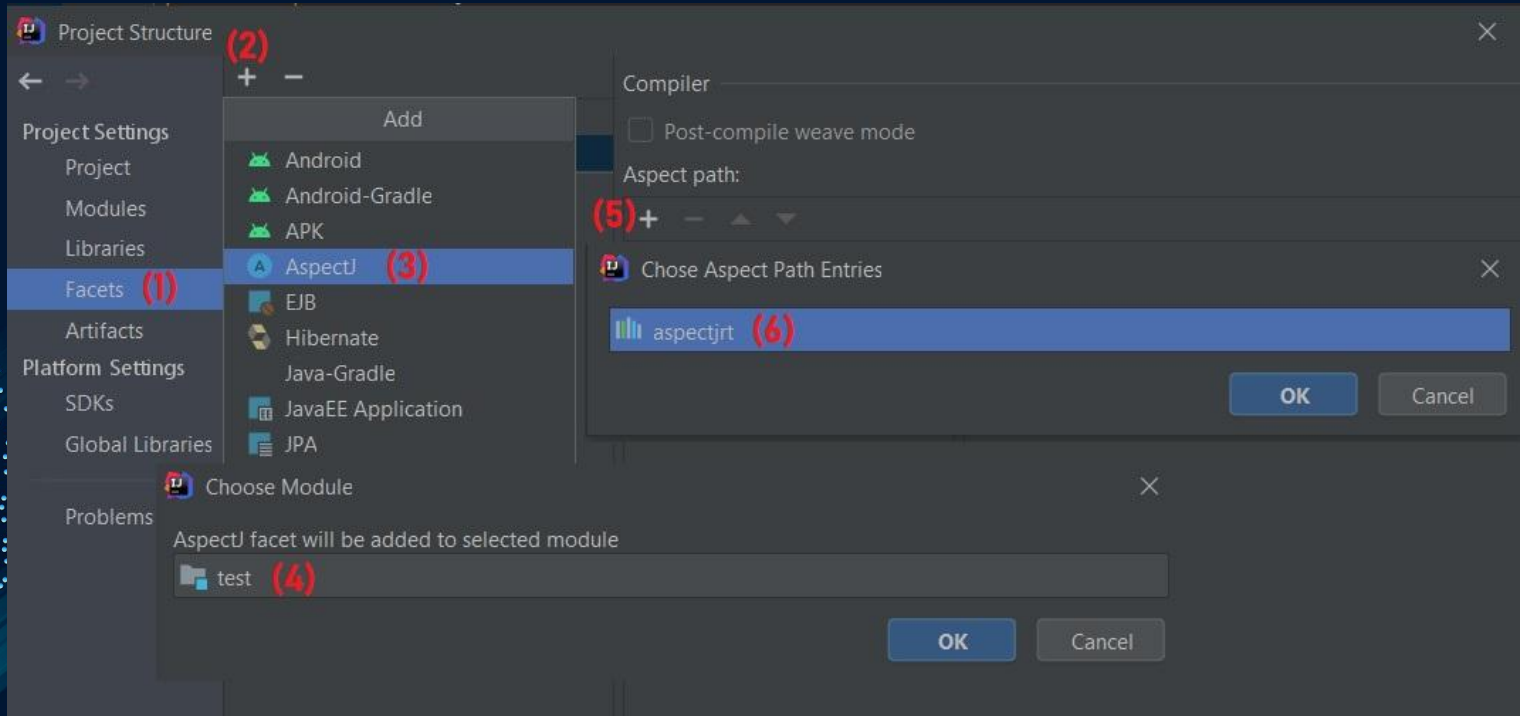
TUTORIAL CONFIGURACIÓN PROYECTOS INTELLIJ IDEA

- b. En la pestaña modules, asegurarse que se encuentre la dependencia a la librería recién agregada. En este caso, el nombre del proyecto es "test"



TUTORIAL CONFIGURACIÓN PROYECTOS INTELLIJ IDEA

c. En la pestaña facets, se relaciona el módulo con el plug-in de AspectJ integrado en IntelliJ. Se crea el elemento siguiendo los pasos indicados en la siguiente imagen.



¿CÓMO EMPIEZO A PROGRAMAR EN ASPECTJ?

- Mientras que la lógica del programa queda intacta y con extensión .java, la extensión de los programas en AspectJ es .aj
- Se busca realizar ciertos "cortes transversales" (crosscutting) del código base, para agregar una funcionalidad transversal, como un log o persistencia de datos
- Es necesario familiarizarse con los conceptos clave, como:
 - Aspect - Aspectos
 - JoinPoint - Puntos de Unión
 - Comodines / Operadores
 - Pointcut - Puntos de Corte
 - Advice - Consejos
 - Introduction - Introducción
 - Compile-time declaration



ASPECT

- Unidad central de AspectJ, de la misma forma que una clase para Java
- Es la definición del código que expresa las reglas de crosscutting
- Los Pointcut, Advices, Introduction y declaraciones se combinan en los aspectos

```
[access specification] aspect <AspectName>  
  [extends class-or-aspect-name]  
  [implements interface-list]  
  [<association-specifier> (Pointcut)] {  
  ... aspect body  
}
```



ASPECT : PROPIEDADES

Los aspectos pueden:

- Pueden incluir métodos y variables
- Pueden tener especificaciones de acceso
- Pueden declararse a sí mismas abstractas
- Pueden extender clases y aspectos abstractos, así como implementar interfaces
- Pueden estar embebidas dentro de clases e interfaces y aspectos anidados
- Pueden ser marcados como privilegiados (privileged)

Los aspectos **NO** pueden:

- **NO** pueden ser directamente instanciados
- **NO** pueden heredar de un aspecto en concreto

JOINPOINTS

- Es cualquier punto identificable en la ejecución de un programa
- Pueden ser la llamada de un método, la instanciación de una clase, o también la obtención (get) o asignación (set) de una variable perteneciente a un método
- **NO** son equivalentes a los pointcuts, ya que un pointcut es una selección de una parte de los joinpoints



POINTCUTS

Un pointcut es un elemento de programa que selecciona puntos de unión y expone datos del contexto de ejecución de esos puntos de unión.

Los pointcuts se utilizan principalmente por consejo(advance). Se pueden componer con operadores booleanos para construir otros cortes de puntos.

DECLARACIÓN DE POINTCUTS

POINTCUTS CON NOMBRE

```
pointcut namePointcut(): pointcutType(Signature);  
  
adviceType() : namePointcut(){  
    //..  
}
```

POINTCUTS ANÓNIMOS

```
adviceType() : pointcutType(Signature){  
    //..  
}
```

COMODINES Y OPERADORES

Tres comodines (wildcards) están presentes en AspectJ:

- * - Indica cualquier número de caracteres excepto el punto
- .. - Indica cualquier número de caracteres incluyendo los puntos
- + - Indica cualquier subclase o subinterfaz de un tipo dado

Dos tipos de operadores están presentes en AspectJ:

Operadores Unarios:

- ! - Negación de un PointCut

Operadores Binarios: Se usan para combinar PointCuts

- && - Se activa sólo si ambos pointcuts se activan
- || - Se activa si cualquiera de los pointcuts se activa



TIPOS POINTCUTS

- **call(signature)**

Verifica que cada método para llamar al joinpoint cuya *signature* coincida.

- **execution(signature)**

Verifica cada método para ejecutar al joinpoint cuya *signature* coincida.



TIPOS POINTCUTS

- **get(signature)**

Elige el contenido de referencia para leer sobre al que se aplique para el joinpoint .

- **set(signature)**

Elige el contenido de referencia para escribir sobre al que se aplique para el joinpoint .



TIPOS POINTCUTS

- **handler**(typePattern)

Elige el controlador de excepciones para el joinpoint y coincide con el *typePattern*.

- **initialization**(constructorPattern)

Selecciona la inicialización de objeto para el joinpoint y coincide con *ConstructorPattern*.



TIPOS POINTCUTS

- **preinitialization(ConstructorPattern)**

Elige cada joinpoint de preinicialización de objeto cuya campo coincida con ConstructorPattern.

- **staticinitialization(TypePattern)**

Selecciona el joinpoint de la ejecución de inicializador estático y coincida con el *TypePattern*



TIPOS POINTCUTS BASADOS EN EL ESTADO

- **this(signature)**

Devuelve el objeto asociado con un joinpoint en particular o limita el alcance de un joinpoint mediante el uso de un tipo de clase.

- **target(signature)**

Devuelve el objeto de destino del joinpoint o limita el alcance de un joinpoint .

- **args(signature)**

Comprueba si el objeto es el indicado para el punto de unión y también es utilizado para acceder al target.



TIPOS POINTCUTS DE CONTROL BASADOS EN FLUJO

- **cflow(pointcut)**

Selecciona todos los joinpoints encontrados durante la ejecución del programa, comenzando en el punto de unión elegido por otro pointcut explícitamente nombrado en la declaración.

- **cflowbelow(pointcut)**

Solo selecciona los puntos de unión que están por debajo del punto de unión inicial elegido por el pointcut especificado en la declaración.



TIPOS POINTCUTS

- **within(typePattern)**

Captura los puntos de unión dentro de una clase, clases internas o aspectos

- **within-code(typePattern)**

Captura los puntos de unión dentro de una clase, clases internas o aspectos

- **if(BooleanExpr)**

Captura los puntos de unión en base a la expresión booleana.



CONTEXT

- Solo serían la mitad de útiles si no fuera por la función de exposición de contexto
- Los pointcuts permiten pasar parámetros a sus advice si los parámetros están expuestos correctamente.
- this, target y args → esta es la única forma en que se pueden exponer los parámetros.



ADVICE

Es el código a ser ejecutado en un Join Point que ha sido seleccionado por un Pointcut.

```
before() : execution(void Account.credit(float)) {  
    System.out.println("About to perform credit operation");  
}
```

TIPOS ADVICE

- **before()**: se ejecute antes de cada joinpoint.
- **after()** : se ejecuta después del joinpoint sin importar el resultado del joinpoint.
- **after() returning**: se ejecuta después solo si el joinpoint retorna normalmente.
- **after() throwing**: se ejecuta después si el joinpoint terminó lanzando una excepción.
- **around()**: Se ejecuta en lugar del joinpoint.



INTRODUCTION

Es una instrucción transversal estática que introduce cambios en las clases, interfaces y aspectos.

```
public aspect MinimumBalanceAspect {  
    private float Account._minimumBalance;  
  
    ... methods and advice using _minimumBalance  
}
```

COMPILE-TIME DECLARATION

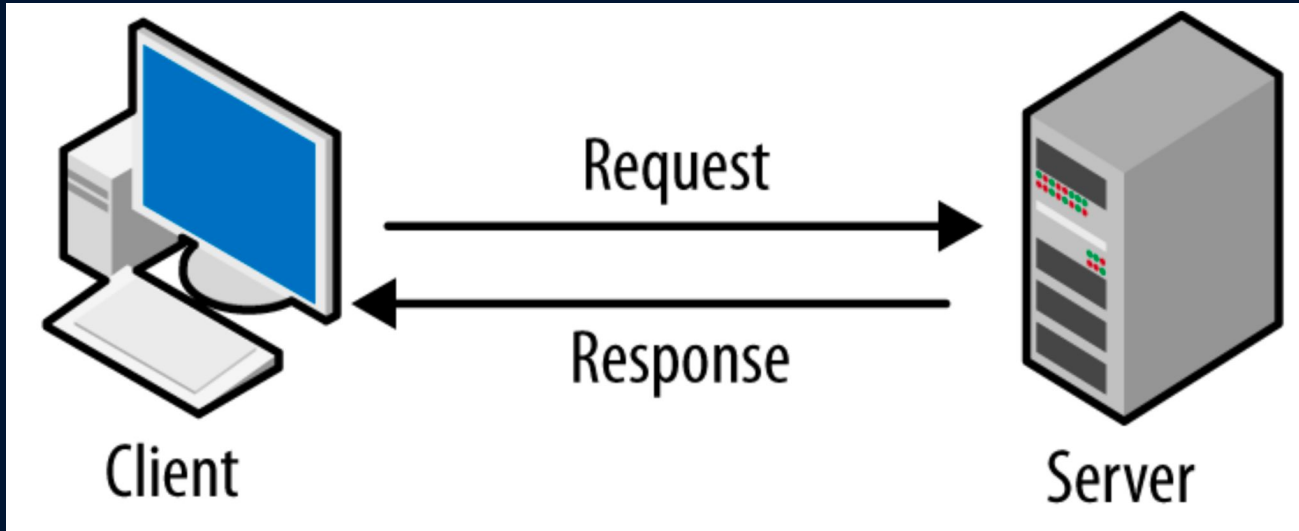
La declaración en tiempo de compilación es una instrucción transversal estática que le permite agregar advertencias y errores en tiempo de compilación al detectar ciertos patrones de uso.

```
declare error : <pointcut> : <message>;
```

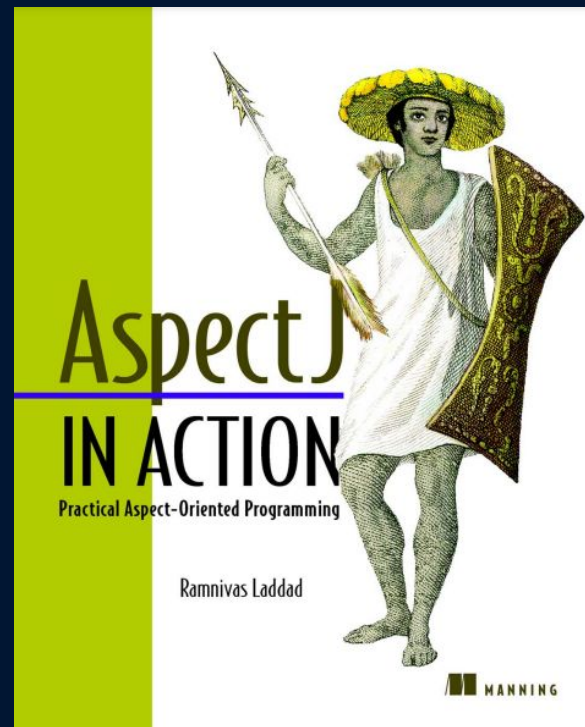
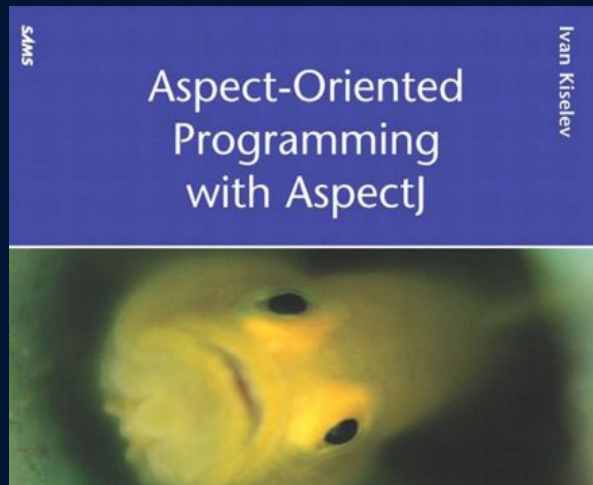
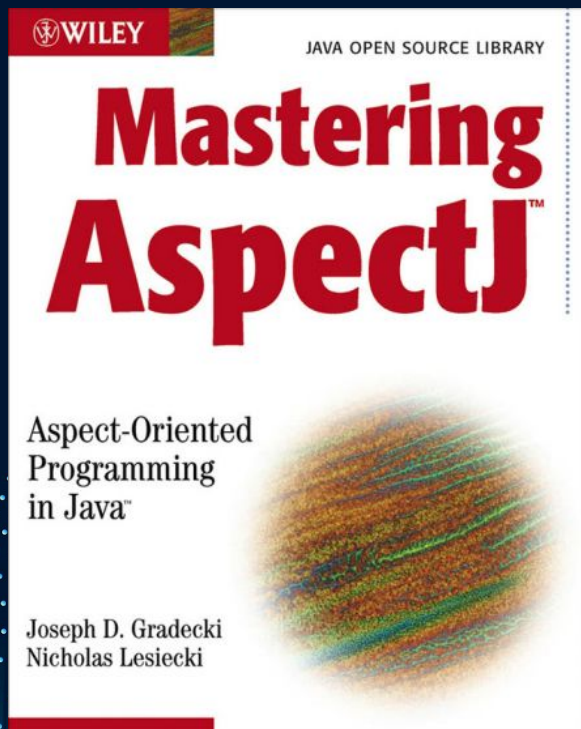
```
declare warning : <pointcut> : <message>;
```



EJEMPLO CONCRETO: PROTOCOLO TCP



LIBROS DE APRENDIZAJE



REFERENCIAS

- <https://folderit.net/es/blog/aspect-oriented-programming-aspectj-es/>
- <https://www.youtube.com/watch?v=8o0E2SkNqNU>
- Integrate AspectJ with NetBeans Platform Development - <https://www.javacodegeeks.com/2013/10/integrate-aspectj-with-netbeans-platform-development.html>
- AspectJ install with Eclipse + example Tuto(AOP) - <https://www.youtube.com/watch?v=EJMUgOrffFY>
- AspectJ - IntelliJ IDEA - <https://www.jetbrains.com/help/idea/aspectj.html>
- AspectJ Downloads - The Eclipse Foundation - <https://www.eclipse.org/aspectj/downloads.php>
- <https://www.eclipse.org/aspectj/doc/next/progguide/semantics-pointcuts.html>
- Programación Orientada a Aspectos - Conceptos claves - http://ferestrepoca.github.io/paradigmas-de-programacion/poa/poa_teor%C3%ADa/Pages/evolucion.html



**¡GRACIAS
POR SU
ATENCIÓN!**