

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
DISEÑO DE SOFTWARE
Paradigma Orientado a Aspectos

Objetivos Específicos

1. Implementar requerimientos utilizando el paradigma orientado a aspectos.
2. Controlar cuando se ejecutan los aspectos transversales usando las instrucciones correspondientes.

Resultado de Aprendizaje

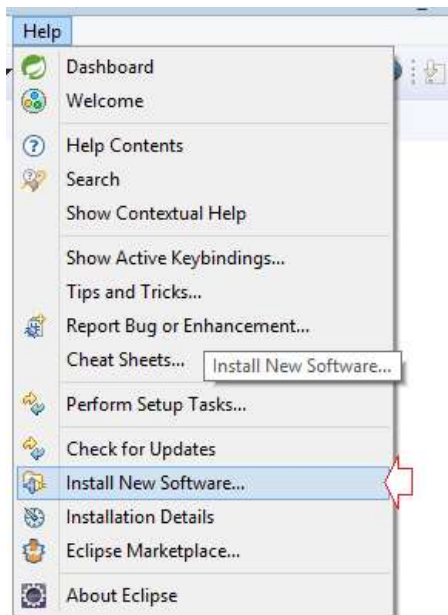
1. Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas para la disciplina del programa.

Sección A: Instalar AspectJ con ajdt en Eclipse

Descargar Eclipse 2022-09

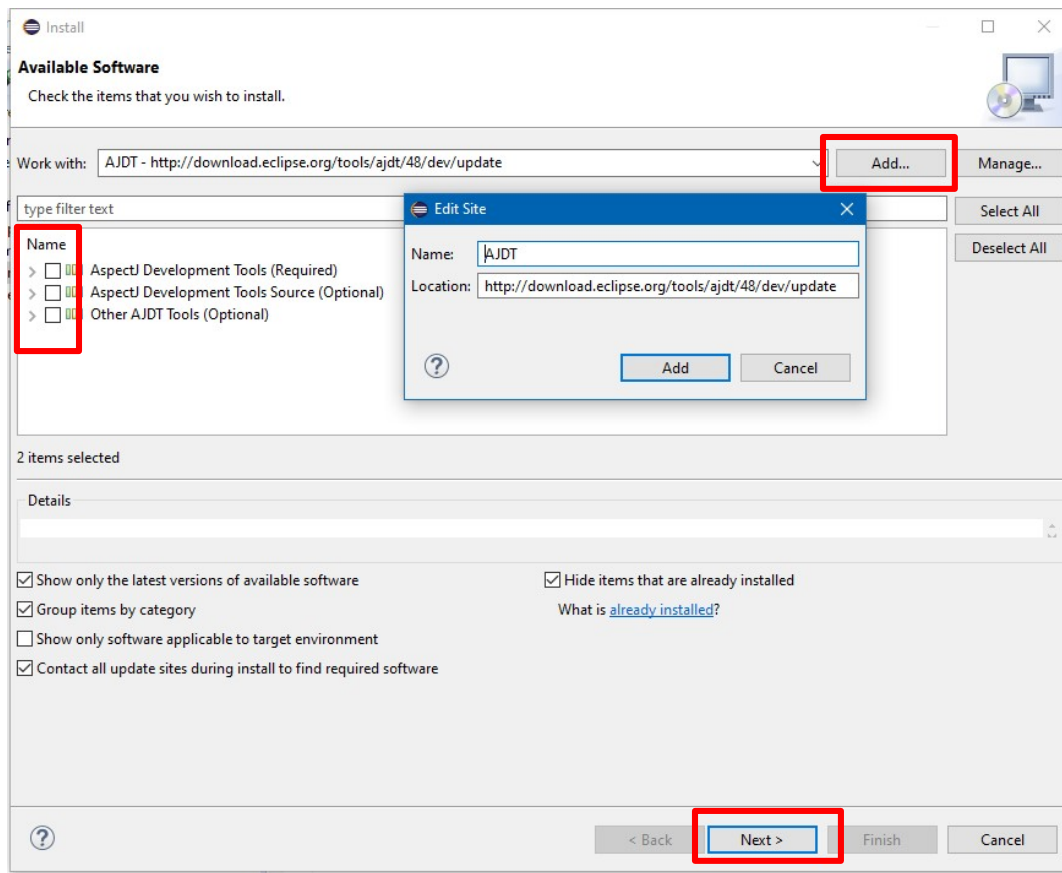
<https://www.eclipse.org/downloads/>

Ejecutarlo y seleccionar: Menú Help -> Install New Software



Agregar el repositorio para descargar AspectJ ingresando los datos como se indica en [3]:

- **Name:** AJDT
- **Location :** <https://download.eclipse.org/tools/ajdt/423/dev/update>

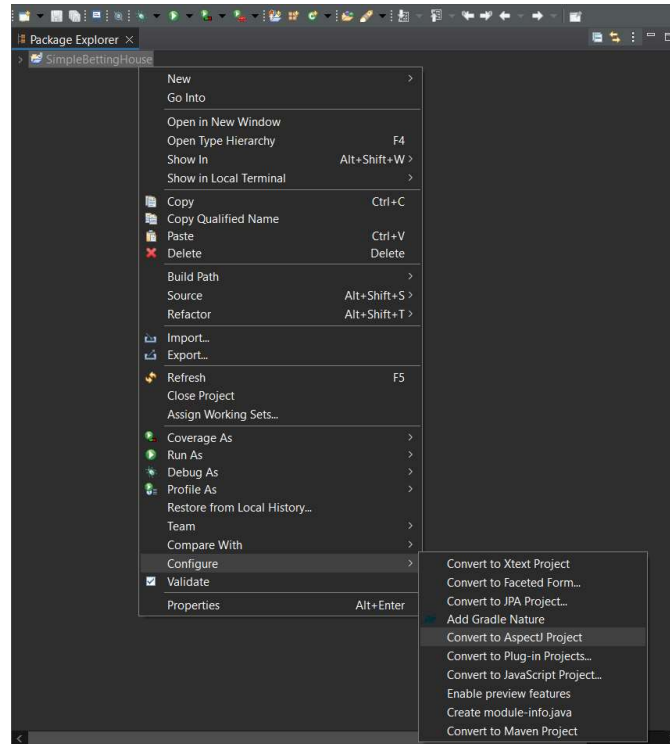


Seleccione los 3 grupos de herramientas, acepte la licencia y reinicie el IDE.

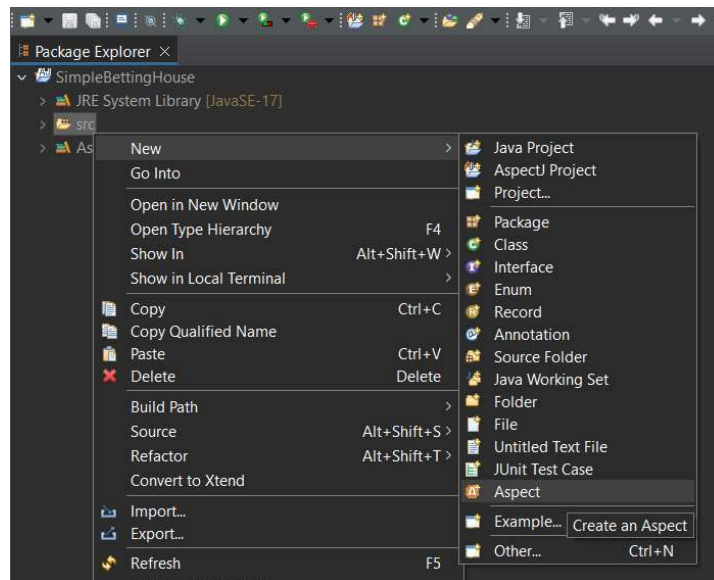
Sección B: Preparación del proyecto con AspectJ

1. Crear un repositorio para este proyecto en GitHub.
2. Clonar en su computadora el repositorio remoto.
3. Descomprimir el proyecto SimpleBettingHouse dentro del repositorio local.
4. Abrir el proyecto utilizando eclipse y dar clic derecho sobre el proyecto.

5. Seleccionar Configure → Convert to AspectJ Project.

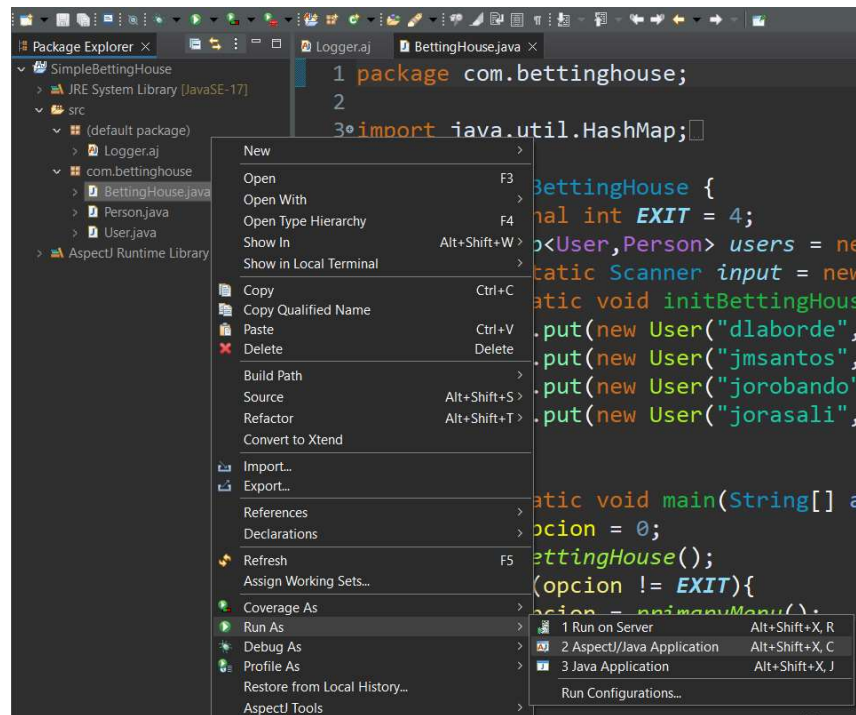


6. Crear el aspecto indicado en la “Sección C” en un nuevo archivo llamado “Logger.aj” dentro de la carpeta “src”. Puede utilizar como guía los comentarios del archivo “Aspectos_guia.java”.



7. La sintaxis se puede encontrar en el enlace [\[https://www.eclipse.org/aspectj/doc/released/progguide/index.html\]](https://www.eclipse.org/aspectj/doc/released/progguide/index.html)

8. Para ejecutar se debe dar clic derecho sobre el archivo “BettingHouse.java” y seleccionar Run as → AspectJ/JavaApplication.



9. Después de terminar el aspecto elimine el archivo Aspecto_guia.java
10. Agregar al README.md una captura que muestre cómo funciona el aspecto de ejemplo “Usuario creado”, luego agregue una captura de su aspecto (No olvide subir su versión al repositorio remoto)

Sección C: Manejo de un aspecto

Requerimiento:

Después de cada vez que un usuario se registre, inicie sesión y cierre sesión, se debe mostrar en pantalla y guardar información en un archivo llamado "Register.txt" cuando realice el registro (Sign Up), y en el archivo "Log.txt" cuando realice el inicio de sesión (Log In) y cierre de sesión (Log Out), tenga en cuenta que se debe ejecutar solo un “advice” que englobe las acciones de inicio de sesión y cierre de sesión. La información que se debe registrar incluye el tipo de acción realizada, el usuario involucrado y la hora en que se realizó la acción. Los tipos de acciones que deben considerarse son:

- a. Registrar usuario
- b. Iniciar sesión
- c. Cerrar sesión

Ejemplo de los archivos:

Register.txt:

```
Usuario registrado: [nickname = rparrales, password = esperanza123] Fecha: [Sun May 21 13:57:23 ECT 2023]  
Usuario registrado: [nickname = igomez, password = kingsleague66] Fecha: [Sun May 21 13:59:51 ECT 2023]
```

Log.txt:

```
Sesión iniciada por usuario: [rparrales] Fecha: [Sun May 21 14:00:46 ECT 2023]  
Sesión cerrada por usuario: [rparrales] Fecha: [Sun May 21 14:00:55 ECT 2023]
```

Recomendaciones

1. Uso de modelos de generación de lenguaje como BARD o ChatGPT para proporcionar explicaciones acerca de AspectJ.
Recuerda que al utilizar el mismo prompt varias veces, es probable que obtengas respuestas diferentes de un modelo de lenguaje. La validez de estas respuestas puede variar y dependerá de tu criterio para determinar cuál usar.

Sugerencias de prompts claves para el taller:

ChatGPT:

1. ¿A qué se refieren las expresiones this, target y args en AspectJ? Dame un ejemplo de su uso en código
2. ¿Cómo puedo obtener el nombre del método dentro de un aspecto en AspectJ?

Bard:

1. How can I obtain the method name within an aspect in AspectJ?
2. What do the expressions this, target, and args refer to in AspectJ? Give me an example of their usage in code.

Entregables

1. Un documento en formato **pdf** que incluya: (i) la identificación de los integrantes del equipo en la primera página; (ii) un índice de contenido en la segunda página; (ii) capturas relevantes de pantalla concernientes al seguimiento de las instrucciones; (iv) enlace al repositorio Git que evidencie participación de **todos** los integrantes; (v) cualquier otra evidencia que considere pertinente.

Referencias

- [1] Git - Using Git in Eclipse IDE (EGit plugin), <https://www.logicbig.com/tutorials/misc/git/eclipse.html>
- [2] access class variable in aspect class, <https://stackoverflow.com/questions/7819410/access-class-variable-in-aspect-class>
- [3] Alexander Kriegisch, Setting up an AspectJ development environment, <https://github.com/eclipse/org.aspectj/blob/master/docs/developer/IDE.md>

Rúbrica de Calificación

Descripción	Valor
Sección A	
Instalación correcta de AspectJ	10
Sección B	
Repositorio Git con participación de todos los integrantes	15
Ejecución correcta del aspecto ejemplo	15
Sección C	
Implementación de aspecto para registrar un usuario (Sign up)	20
Implementación de aspecto para manejo de sesión (Log in y Log out)	30
Persistencia en archivo de cada tipo de acción y hora en que se realizó	10
Total	100
No subir a Aula Virtual los entregables requeridos de acuerdo con lo especificado	-50

Late Submission Policy

Delay (§)	Penalty (Ω)
1 hour or less	loss of 10%
1 to 6 hours	loss of 20%
6 to 24 hours	loss of 30%
Over 24 hours:	loss of 100%

(§) every clock hour counts including weekends or holidays

(Ω) automatic and non-negotiable penalty