

Depuración remota de programas desplegados en el servidor Glassfish con NetBeans

Tutorial

Objetivo

El **objetivo** de este tutorial es:

- Habilitar la **depuración de aplicaciones** en GlassFish
- Compilación de un proyecto *genérico* Java con opción de **debug**
- Importar el proyecto dentro de **NetBeans para una depuración sencilla del código**. La depuración puede hacerse contra un servidor local o remoto, a través de **JPDA**

Queda **fuera del alcance** del mismo:

- La **compilación** del proyecto desde el propio NetBeans (esta tarea deberá seguirse realizando desde la línea de comandos). Es posible realizar dicha tarea también desde netbeans si se emplea otro tipo de importación de proyectos Java (Ej: *Java Web Project from Existing Project*), pero la forma será distinta para cada una de las prácticas del curso por lo que no vamos a cubrirlo.
- **NetBeans no va a ejecutar el código cargado**, sólo se utiliza como interfaz para activar puntos de ruptura y visualizar variables. El IDE por tanto puede mostrar errores al no encontrar funciones que se llaman dentro del código fuente cargado, pero se pueden ignorar porque no afectan a la correcta ejecución del código desplegado en el o los servidores de aplicaciones, donde sí se han especificado correctamente todas las bibliotecas en el *classpath*. En cualquier caso estos “avisos” son fácilmente subsanables añadiendo al *classpath* del IDE los .jar que se requieran (ej: javaee.jar) como veremos más adelante.

NOTA IMPORTANTE:

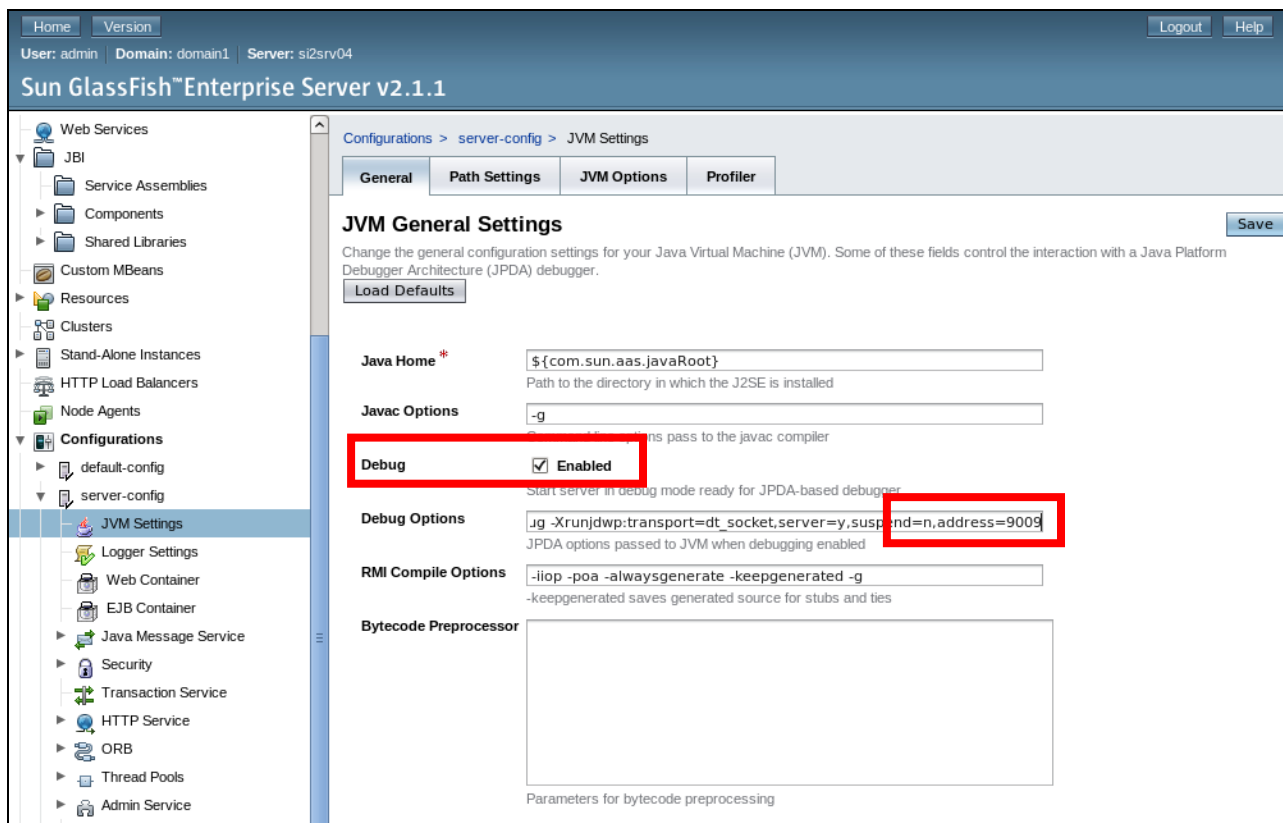
- **No olvide realizar siempre las pruebas de rendimiento con programas que se hayan compilado sin la opción de *debug*.**

Activar la depuración remota en el servidor de aplicaciones

En la consola de administración del servidor de aplicaciones sobre el que quiere desplegar:

- Busque la configuración que corresponde a la instancia del servidor bajo análisis. Normalmente, será la instancia *server*, y su configuración, *server-config*.
- Seleccione la pantalla *JVM Settings*.
- En la pestaña *JVM General Settings*, active la opción *Debug - Enabled*
- En la línea *Debug Options*, compruebe el puerto en el que escucha el *debugger*. Por defecto es el 9009.

IMPORTANTE: Rearranque el servidor.



Compile el código de su proyecto con la opción de depuración.

Edite el archivo *build.xml* que genera el proyecto, y en todos los *targets* de compilación añada el atributo *debug="on"*.

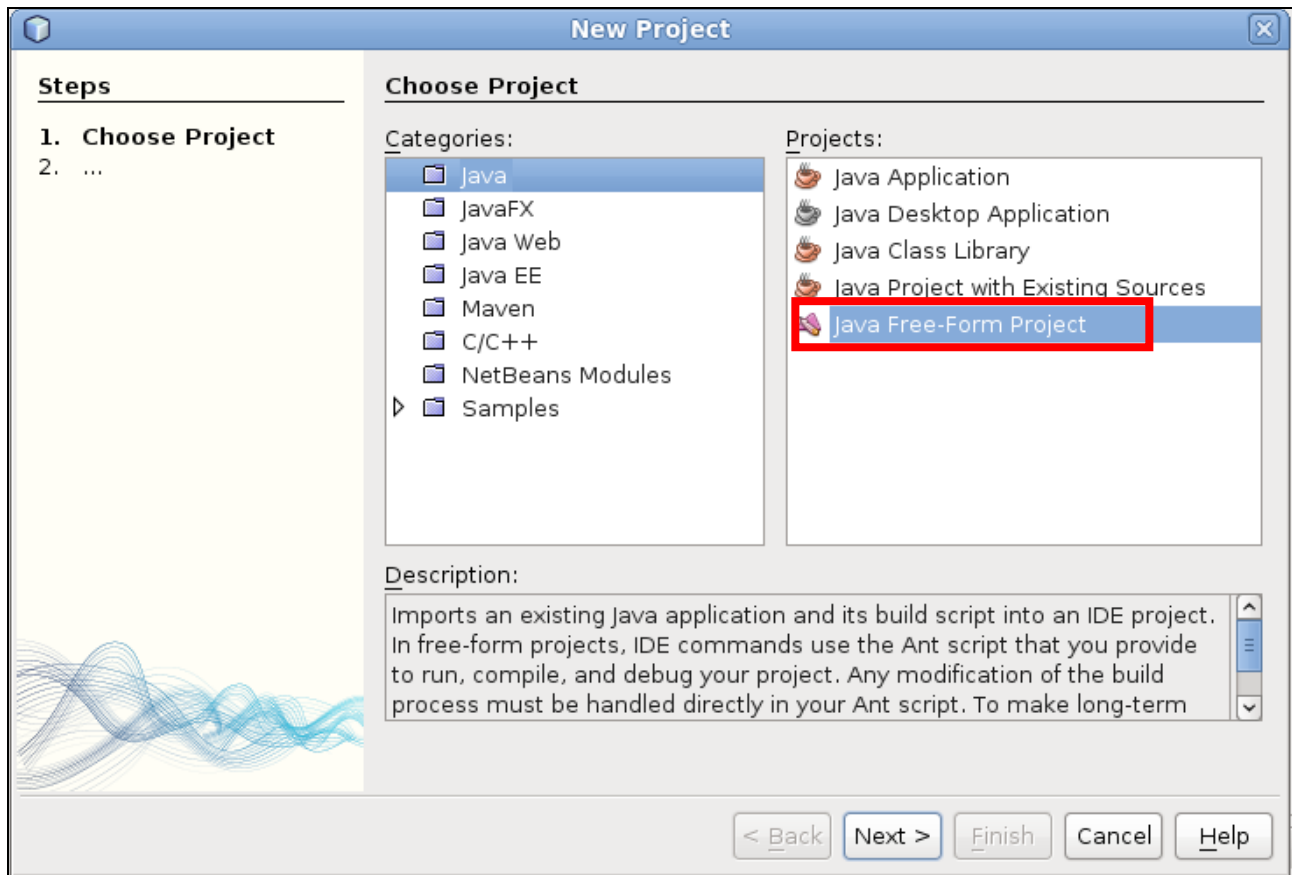
```

<target name="compilar-cliente" depends="montar-jerarquia" description="Compila el cliente">
  <javac srcdir="${src.client}"
        destdir="${build.client}/WEB-INF/classes"
        includes="${paquete}/**"
        debug="on">
    <classpath refid="compile.client.classpath"/>
  </javac>
</target>
  
```

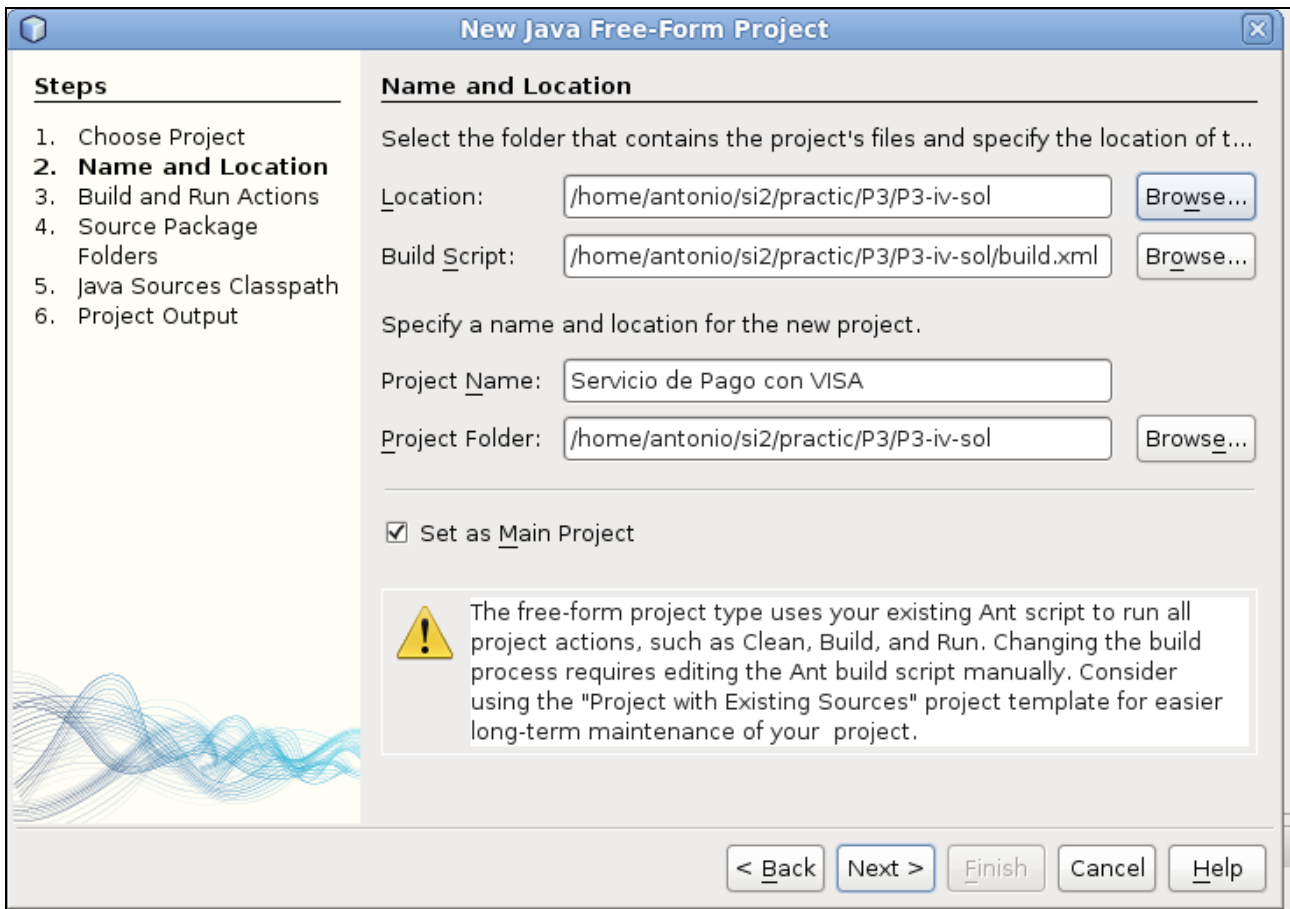
Compile y despliegue de la manera habitual.

Importe en NetBeans el código de su proyecto.

- Genere un nuevo proyecto tipo **Java Free-Form Project**.



- Pulse *Next>*. En la siguiente pantalla, seleccione en Location el directorio raíz donde se encuentra el código (será aquel que contiene el archivo build.xml). NetBeans abre este archivo y rellena el resto de los campos del formulario.



Steps

1. Choose Project
2. **Name and Location**
3. Build and Run Actions
4. Source Package Folders
5. Java Sources Classpath
6. Project Output

Name and Location

Select the folder that contains the project's files and specify the location of t...

Location:


Build Script:

Specify a name and location for the new project.

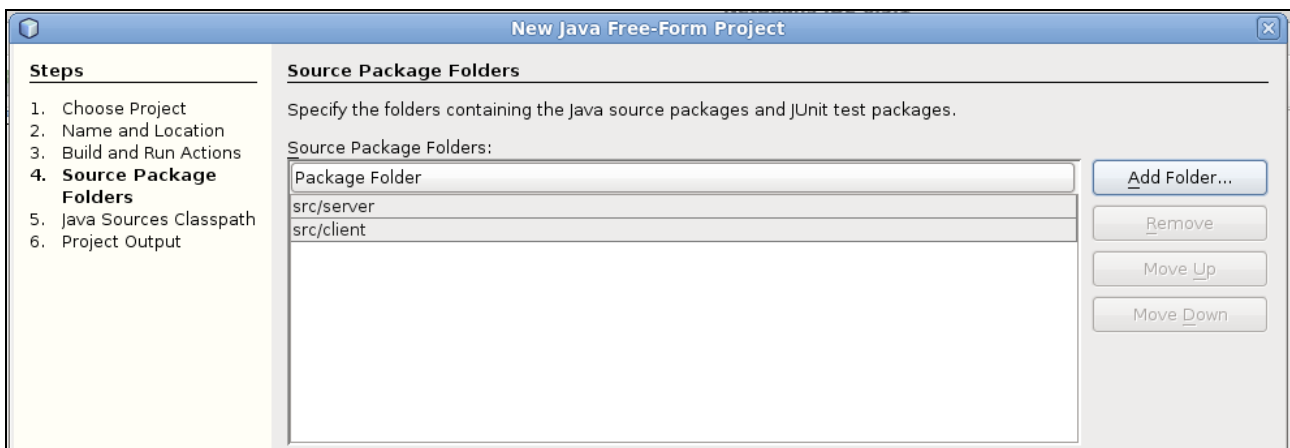
Project Name:

Project Folder:

☒ Set as Main Project

 The free-form project type uses your existing Ant script to run all project actions, such as Clean, Build, and Run. Changing the build process requires editing the Ant build script manually. Consider using the "Project with Existing Sources" project template for easier long-term maintenance of your project.

- Pulse *Next>*. En la siguiente pantalla, **opcionalmente** (no es necesario para depurar), seleccione los **objetivos que construyen y limpian el código del proyecto** (estos pasos los podrá ejecutar desde NetBeans, pero, para realizar el despliegue, se necesita añadir la librería de ant de Glassfish, y esa tarea cae fuera del alcance de este tutorial).
- Pulse *Next>*. En la siguiente pantalla, añada los directorios que contienen el código fuente de su programa. Por ejemplo, en este programa se generan dos desplegables, *cliente* y *servidor*, que tienen sus fuentes en los directorios *src/client* y *src/server*, respectivamente.



Source Package Folders

Specify the folders containing the Java source packages and JUnit test packages.

Source Package Folders:

Package Folder
src/server
src/client

- Pulse *Next>*. En la siguiente pantalla, **opcionalmente**, añada el *classpath* de los paquetes a generar. Puede conocerlos consultando el archivo *build.xml*, en los elementos *<path>* que contiene al comienzo del mismo, tras la declaración de propiedades. Por ejemplo, se puede añadir la librería de implementación de J2EE:

```
/usr/local/glassfish/lib/javaee.jar
```

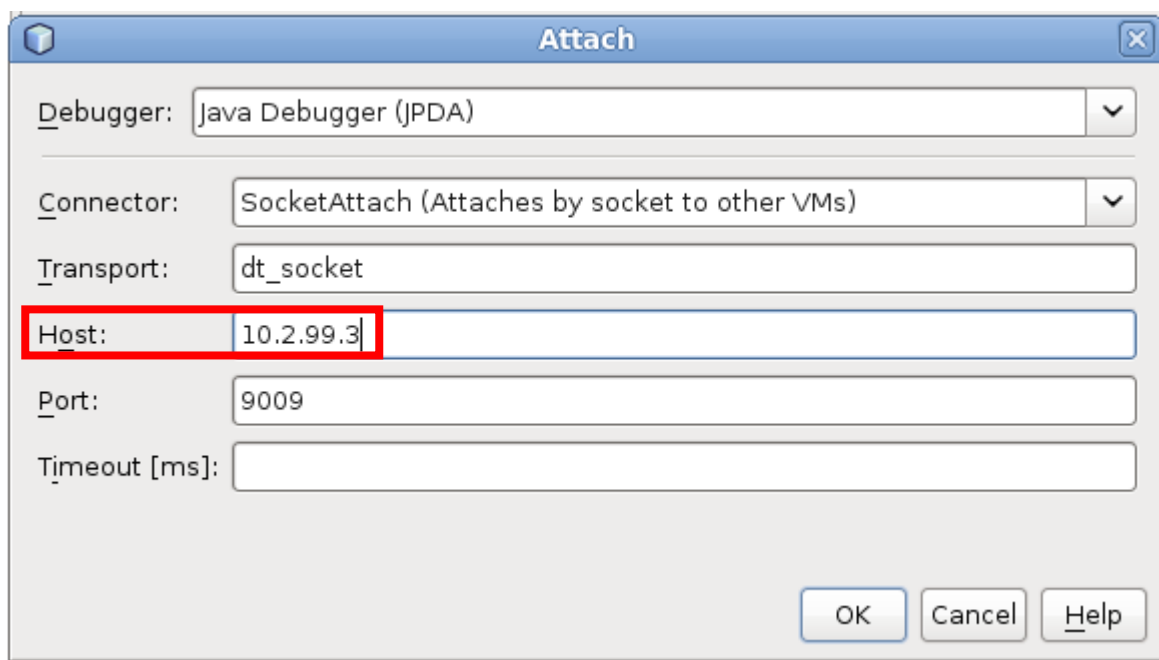
También, en clientes que necesitan acceder a objetos remotos, se añadiría la librería de los *stubs* de dichos objetos.

Este último paso es totalmente opcional, ya que **NetBeans no va a ejecutar el código cargado**.

- Pulse *Finish*. Ha creado el proyecto.

Conectar NetBeans con el servidor o servidores de aplicaciones en los que se encuentra la aplicación a depurar

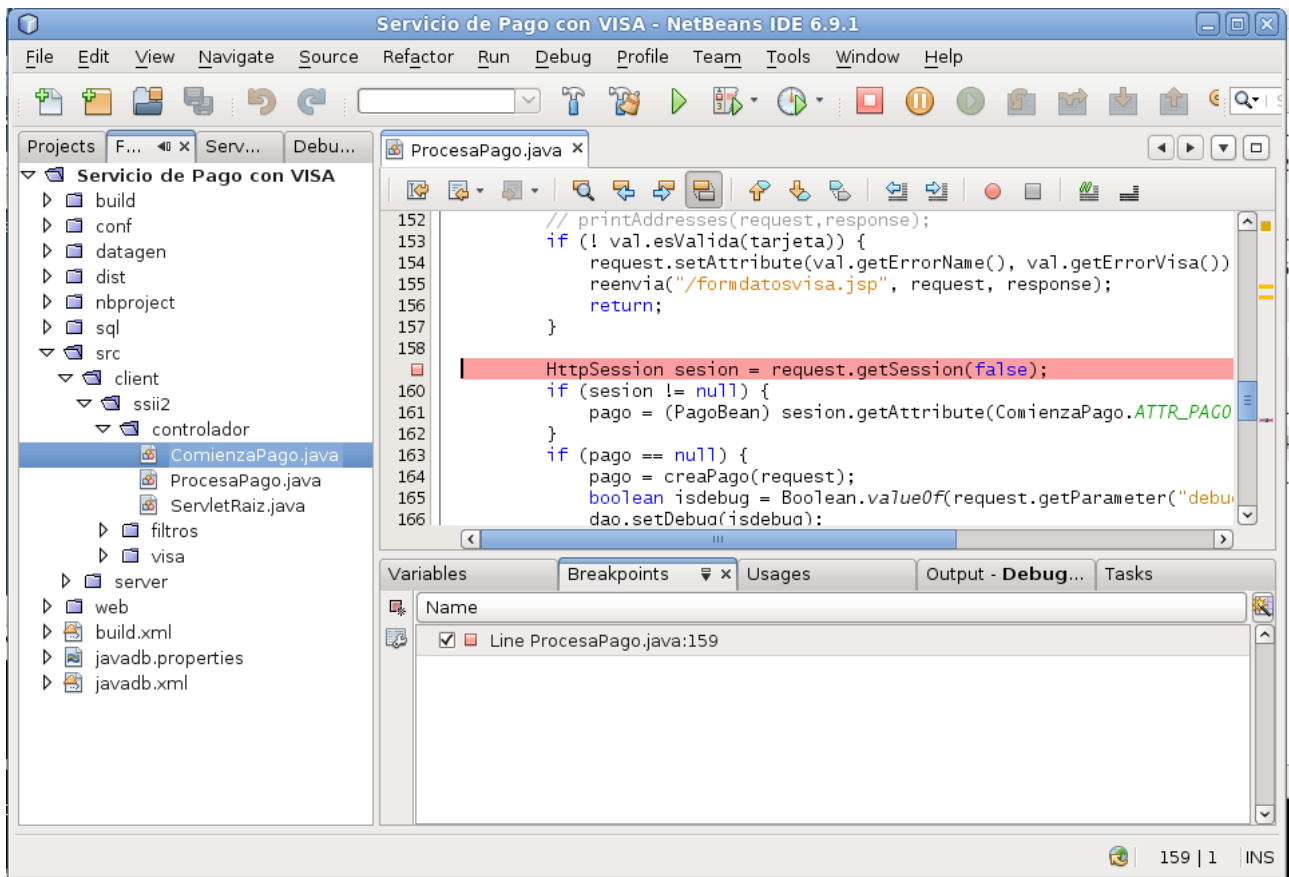
En la barra de acción de NetBeans, seleccione *Debug -> Attach Debugger*. En el campo *Host*, introduzca la IP del ordenador donde reside el servidor de aplicaciones, y, si es necesario, modifique el valor del puerto con el que obtuvo en el paso 1.



Pulse OK. Si todo es correcto, en la zona de navegación del lateral izquierdo aparecerán todos los hilos de proceso activos del servidor en ese momento. El sistema está listo para depurar.

Depurar con NetBeans

Establezca Breakpoints dentro de su código (*servlets*, *EJBs*, *Beans...*)



Ejecute la aplicación desde un navegador normal. Por ejemplo:

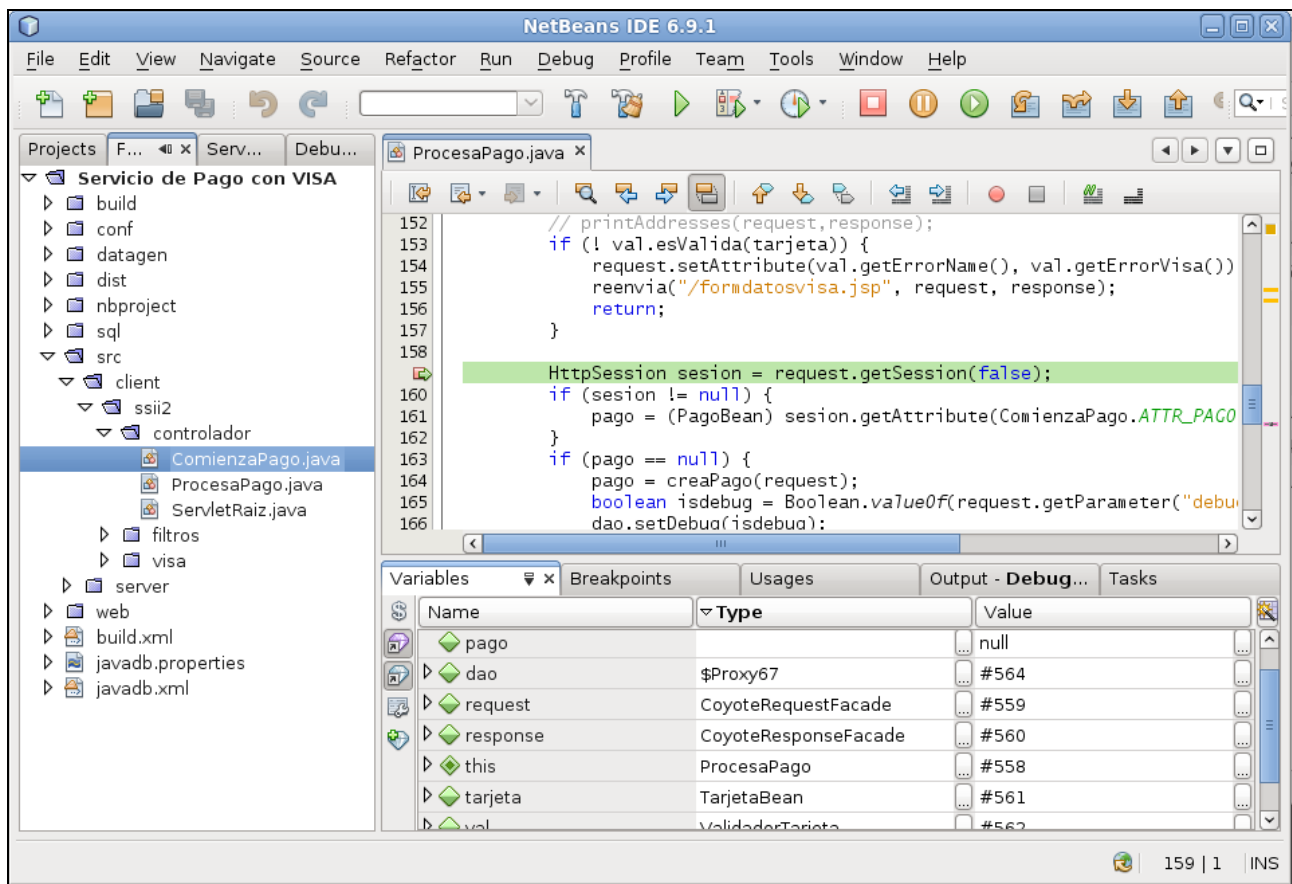
Pago con tarjeta

Numero de visa:
 Titular:
 Fecha Emisión:
 Fecha Caducidad:
 CVV2:

Id Transacción: 1
 Id Comercion: 100
 Importe: 10.0

Prácticas de Sistemas Informáticos II

Al pulsar "Pagar", el servlet ProcesaPago se detendrá en el breakpoint establecido en NetBeans.



Puede examinar variables (*watch*), poner más *breakpoints* pulsando en la barra de la izquierda entre otras tareas. Utilice los botones de la barra superior para *Continuar* (F5), *Siguiente Paso* (F8), *Entrar en llamada* (F7) etc

El navegador queda a la espera de recibir una respuesta. Es posible que, si se tarda mucho en generar el *Response*, el navegador haya dado timeout y ya no vea la respuesta en pantalla, pero esto no afecta a la lógica de ejecución del lado del servidor.