




## **Eclipse IDE**

### **Manual de Usuario**

Versión: 0100

Fecha: 14/12/2020

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito de la Junta de Andalucía.

 <b>JUNTA DE ANDALUCÍA</b>	<b>Eclipse IDE</b> <b>Manual de Usuario</b>	Sergio Díaz Campos
---	--	--------------------

## HOJA DE CONTROL

<b>Organismo</b>	Sergio Díaz Campos		
<b>Proyecto</b>	Eclipse IDE		
<b>Entregable</b>	Manual de Usuario		
<b>Autor</b>	Sergio Díaz Campos		
<b>Versión/Edición</b>	0100	<b>Fecha Versión</b>	14/12/2020
<b>Aprobado por</b>	María Aurora Dacal Asin	<b>Fecha Aprobación</b>	17/12/2020
		<b>Nº Total de Páginas</b>	35

## REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
0100	Versión inicial	Sergio Díaz Campos	14/12/2020

## CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos
Sergio Díaz Campos

## Índice

1	DESCRIPCIÓN DEL SISTEMA.....	5
1.1	Objeto.....	5
1.2	Alcance .....	5
1.3	Funcionalidad .....	5
2	MAPA DEL SISTEMA.....	6
2.1	Modelo Lógico .....	6
2.2	Navegación .....	7
3	DESCRIPCIÓN DEL SISTEMA.....	9
3.1	Project Explorer .....	9
3.1.1	Mensajes de error .....	9
3.1.2	Ayudas contextuales .....	9
3.2	Área de desarrollo .....	10
3.2.1	Mensajes de error .....	10
3.2.2	Ayudas contextuales .....	10
3.3	Ayuda de desarrollo .....	11
3.3.1	Mensajes de error .....	11
3.3.2	Ayudas contextuales .....	11
3.3.3	Debug .....	12
3.4	Consola .....	14
3.4.1	Mensajes de error .....	14
3.4.2	Ayudas contextuales .....	14
3.5	Barra de menús .....	15
3.5.1	Mensajes de error .....	15
3.5.2	Ayudas contextuales .....	15
4	FAQ.....	26
4.1	¿Qué es la plataforma Eclipse? .....	26
4.2	¿Dónde consigo e instalo Eclipse? .....	26
4.3	¿Cómo instalo nuevos Plug-Ins?.....	27
4.4	¿Dónde puedo obtener ayuda? .....	27
4.5	¿Cómo puedo reportar un error?.....	28
4.6	¿Cómo y cuándo guardo el workspace?.....	28
4.7	¿Cómo puedo extender el JDT? .....	28
4.8	¿Qué es Junit? .....	29

5	ANEXOS .....	30
6	GLOSARIO .....	31
7	BIBLIOGRAFÍA Y REFERENCIAS .....	35

## 1 DESCRIPCIÓN DEL SISTEMA

### 1.1 Objeto

En este tutorial vamos a explicar sobre la herramienta de desarrollo para Java conocida como Eclipse IDE.

Puede compartir este PDF con cualquier persona que crea que podría beneficiarse de él, descargue la última versión [aquí](#).

Es PDF a modo de guía de usuario de eclipse no oficial y gratuito creado con fines educativos.

Las imágenes pueden ser propiedad de sus respectivos dueños a menos que se especifique lo contrario. Todas las marcas comerciales y marcas registradas son propiedad de sus respectivos dueños de empresas.

Utilice el contenido presentado en este libro bajo su propio riesgo; no se garantiza que sea correcto ni precisa, envíe sus comentarios y correcciones a [sergiodiazcampos@gmail.com](mailto:sergiodiazcampos@gmail.com)

### 1.2 Alcance

Durante el desarrollo de este documento, os explicaremos qué es Eclipse, los requisitos que hay que cumplir para poder instalarlo, la instalación del JDK de Java, la descarga e instalación del ejecutable de Eclipse, la configuración del JDK dentro del IDE (Entorno de Desarrollo Integrado), la puesta a punto del workspace, y unos cuantos para hacer el desarrollo más eficiente.

### 1.3 Funcionalidad

En la web oficial de Eclipse, se define como “un IDE para todo y para nada en particular”. Eclipse es un tipo de workbench (banco de trabajo) sobre el cual se pueden agragar herramientas para el desarrollo de cualquier lenguaje, mediante la instalación de los plugins adecuados.

Aparte de integrar diversos lenguajes sobre un mismo IDE, su arquitectura de plugins permite introducir otras aplicaciones que puedan resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces, ayuda en línea para librerías, etc.

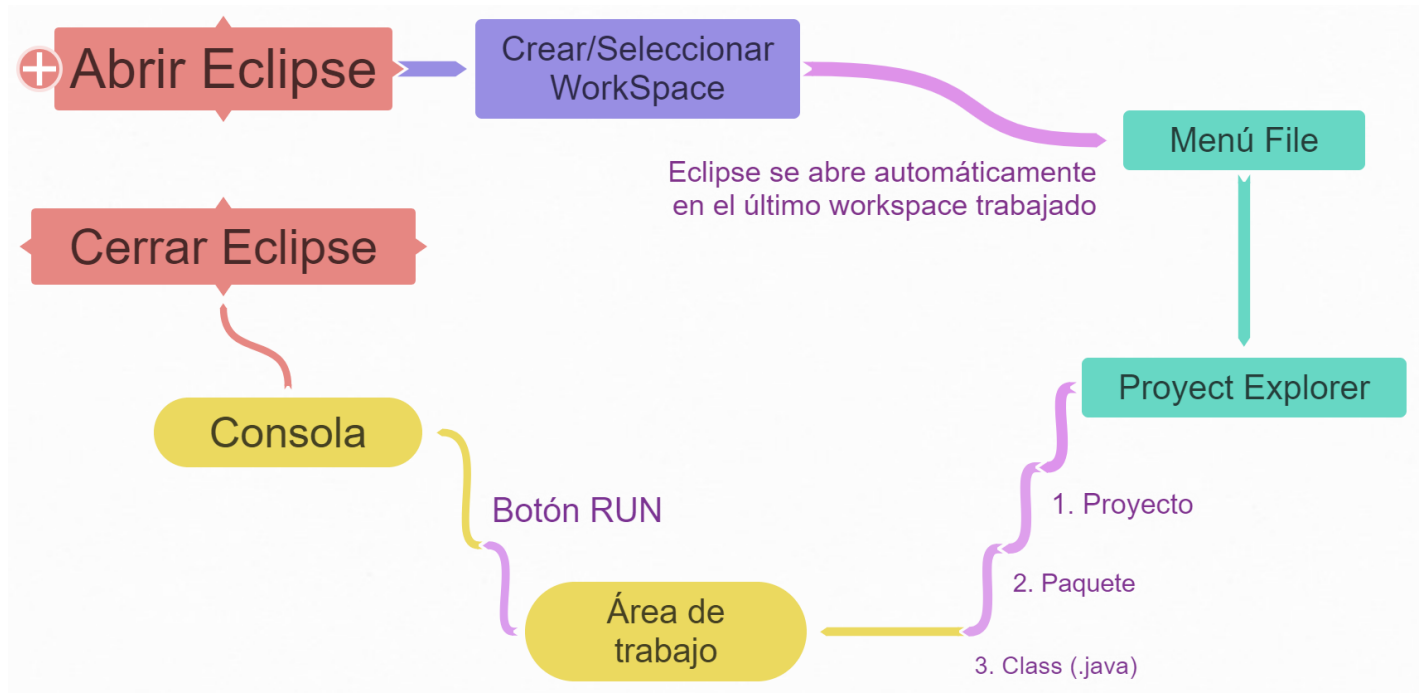
Eclipse es una de las herramientas que hay dentro del Proyecto Eclipse. El Proyecto Eclipse recoge tanto el desarrollo del IDE Eclipse como también algunos de los plugins más importantes (como el JDT para Java, o el CDT para C/C++).

A modo resumen, se trata de una plataforma de código abierto y uso libre que puede ser ampliada por medio de la instalación de plugins.

Todo esto la hace una de las herramientas más completas y usadas para el desarrollo Java por muchos profesionales y empresas en todo el mundo. Proporciona integración con otras herramientas que facilitan el desarrollo de software en dicho lenguaje, y está preparado para trabajar con espacios de trabajo, los workspaces.

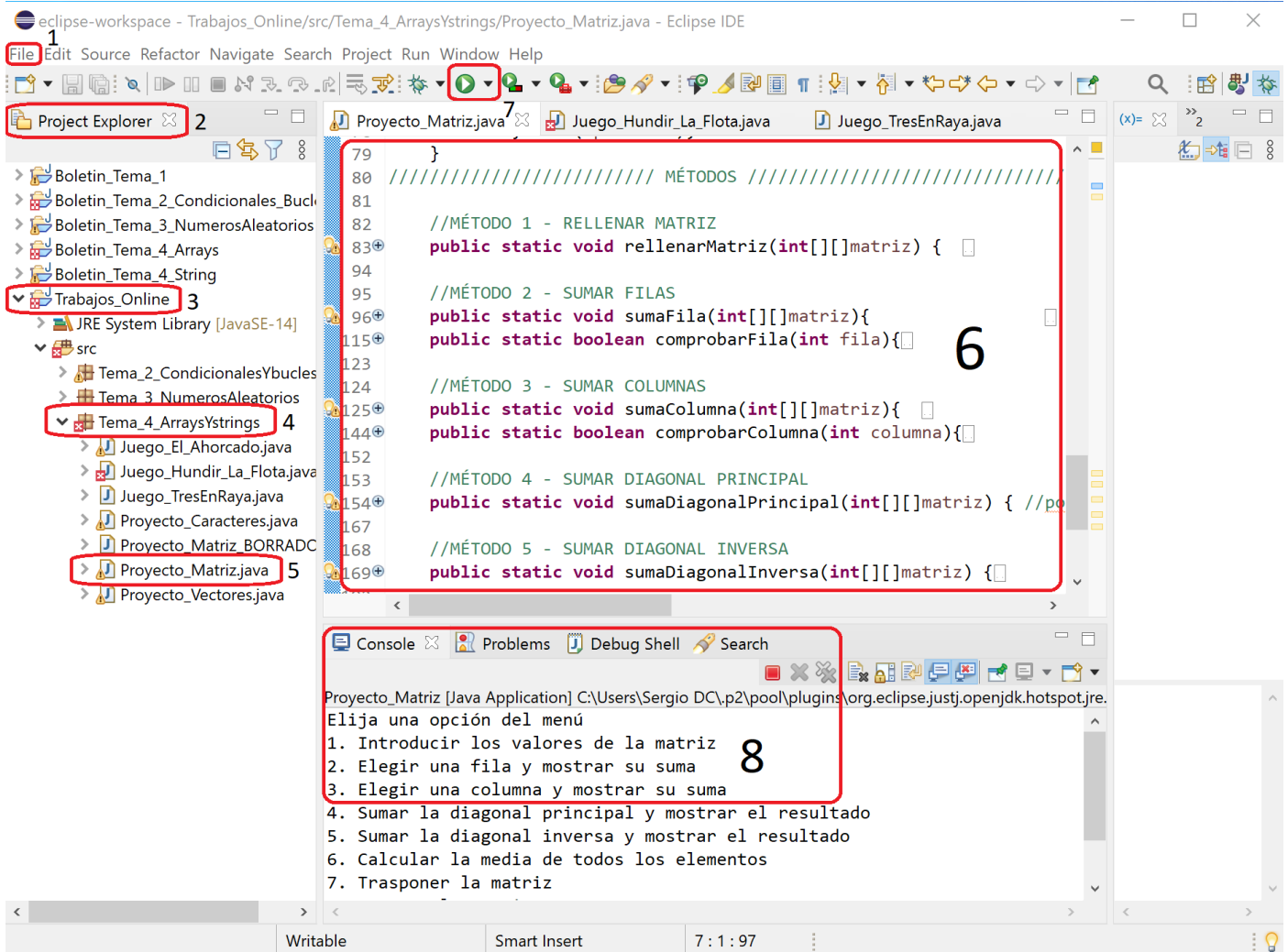
## 2 MAPA DEL SISTEMA

### 2.1 Modelo Lógico



En el siguiente sub-apartado se explica visualmente este esquema

## 2.2 Navegación



A modo de vista general, cuando se abre la aplicación de Eclipse, el usuario encontrará visualmente lo que se muestra en la imagen y, en primer lugar, comenzará a moverse y a dirigir su mirada hacia las secciones rodeadas con un círculo rojo, en relación al orden que muestran los números indicados.

1. Típica pestaña de “Archivo” (file) donde se encuentran las opciones comunes a la mayoría de programas para abrir y guardar archivos, además de múltiples opciones más de gestión para los programas que llevemos creados.
2. Hay que tener en cuenta el WorkSpace sobre el cual estamos trabajando y el cual contiene los archivos que estamos trabajando actualmente y se mostrarán en la sección de la izquierda a modo de esquema desglosado.
3. El 1º nivel navegable de nuestros archivos son los Proyectos (icono de carpeta). Este tipo de carpeta dentro del programa, contendrá los archivos de los mismos programas que hayamos o estemos desarrollando. Éste a su vez, despliega Paquetes y Clases, los cuales los veremos a continuación.
4. El 2º nivel navegable corresponde a los Paquetes (Packs), los cuales funcionan a modo de segmentar en distintos grupos nuestros desarrollos según el criterio y comodidad orientativa del usuario.

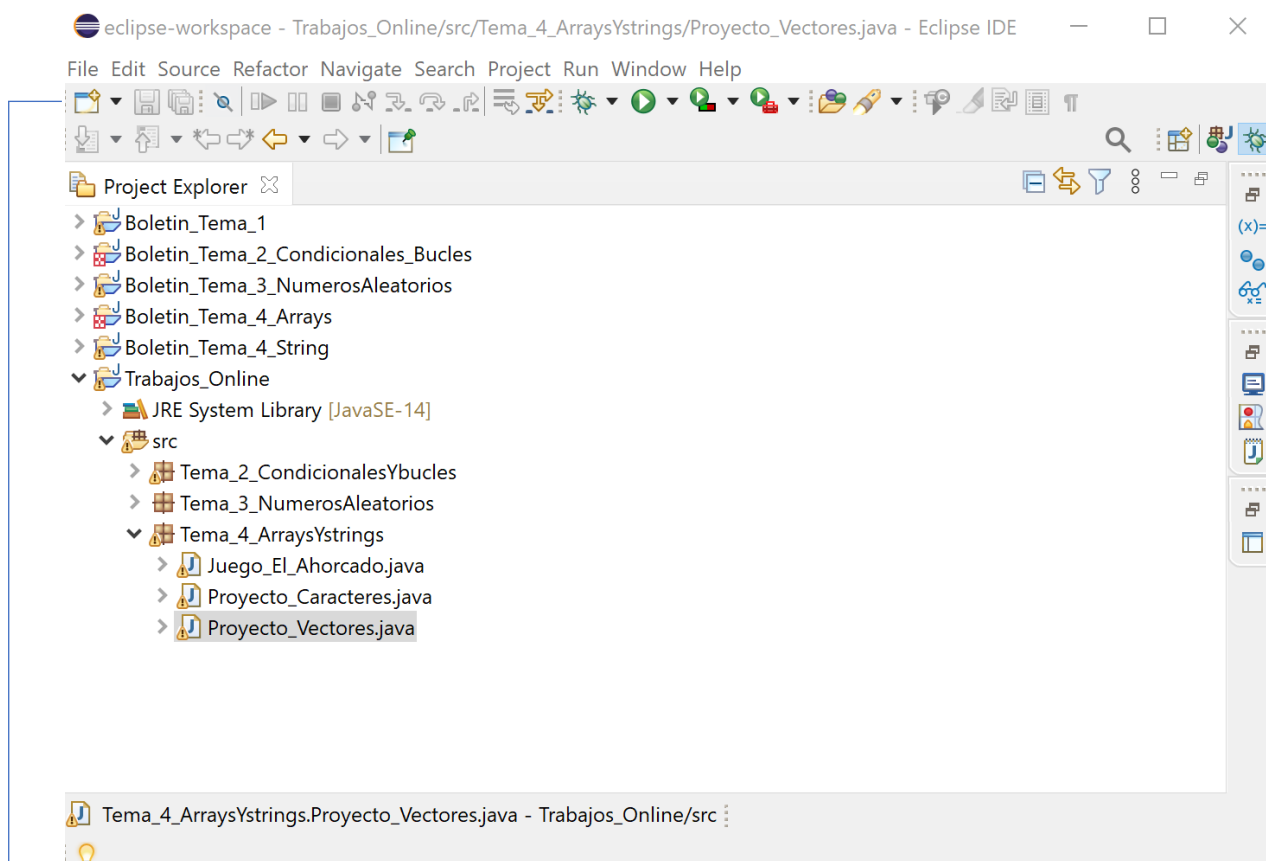
5. El 3º y último nivel pertenece a las Clases (class) que son los archivos que contienen directamente el código fuente que el usuario está desarrollando
6. En este espacio el usuario desarrolla su código fuente para la creación de su programa
7. Con este botón podrá compilarlo y ejecutarlo
8. Esta sección es la Consola, en la cual se muestra la ejecución del programa desarrollado (interpretación y ejecución del código fuente)



### 3 DESCRIPCIÓN DEL SISTEMA

En este apartado, vamos a presentar una guía rápida a modo resumen de las diferentes secciones e interfaces que ofrece Eclipse al abrirse, siguiendo una ruta que correspondería con la visión del usuario y manejo general del programa.

#### 3.1 Project Explorer



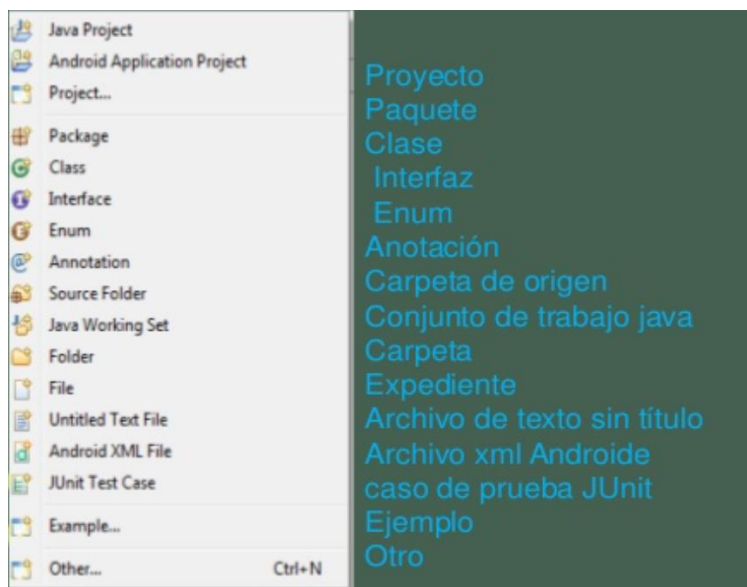
Muchas veces los desarrolladores Java tenemos un conjunto de proyectos eclipse con los que estamos trabajando que puede ser bastante amplio (10-20). Al final es imposible que todos entren de una forma cómoda en la lista principal

##### 3.1.1 Mensajes de error

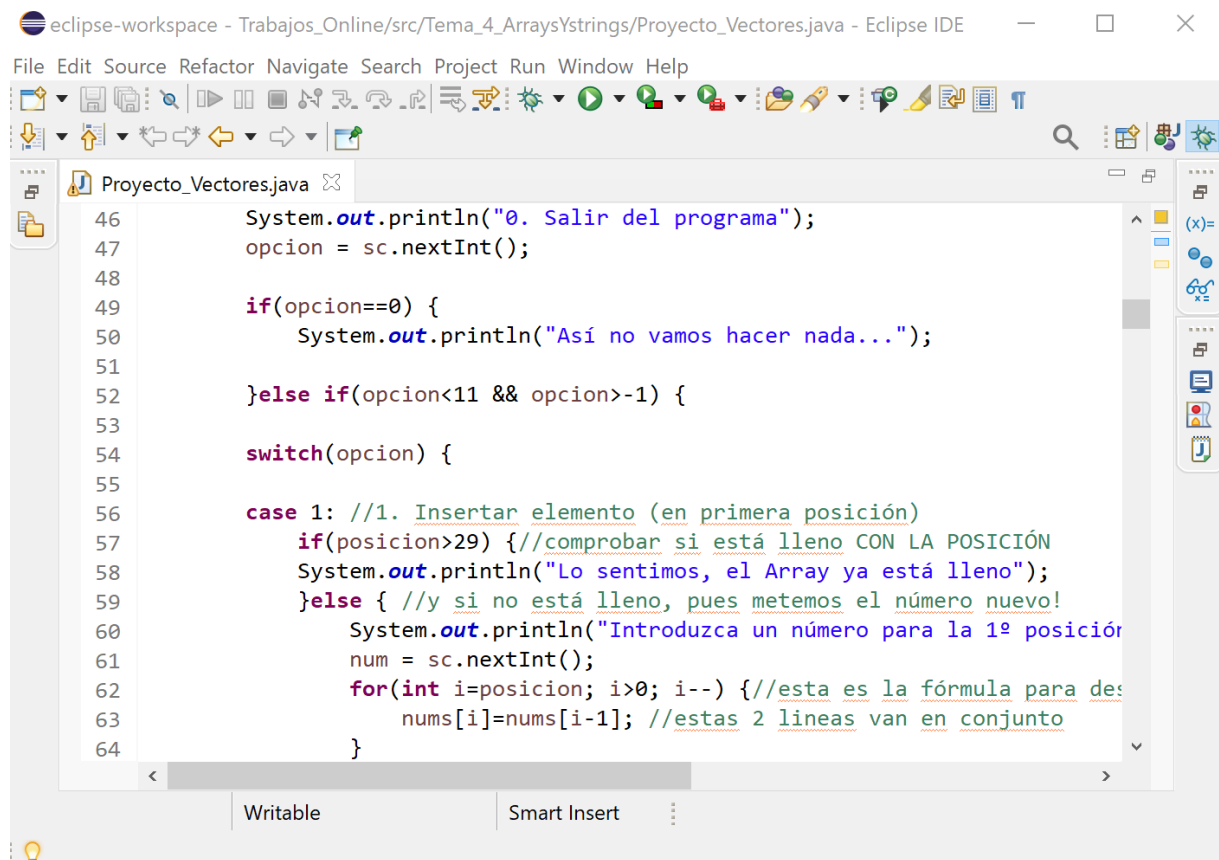
No procede.

##### 3.1.2 Ayudas contextuales

No procede



## 3.2 Área de desarrollo



Los programas reales, pueden llegar a tener cientos o miles de clases. Trabajar con la consola puede ser una pesadilla. La realidad, es que la mayoría de los programadores, usan un “Entorno de desarrollo integrado” IDE (integrated development environment).

Eclipse, es el IDE más usado para el desarrollo en Java. Es de código abierto y gratuito. Eclipse, permite extender sus funciones, mediante el desarrollo de plugins. Existen muchas versiones de Eclipse para programar en diferentes lenguajes e incluso muchas versiones para Java.

Existen otros entornos de desarrollo para Java, entre los que destacan Netbeans e IntelliJ IDEA. Netbeans, es el IDE de Oracle, que es el principal distribuidor de JDK. Es gratuito y muy usado en el ámbito académico, pero no tanto en el profesional. IntelliJ IDEA, por el contrario, es recomendado por muchos profesionales, pero no es gratuito y no es tan usado.

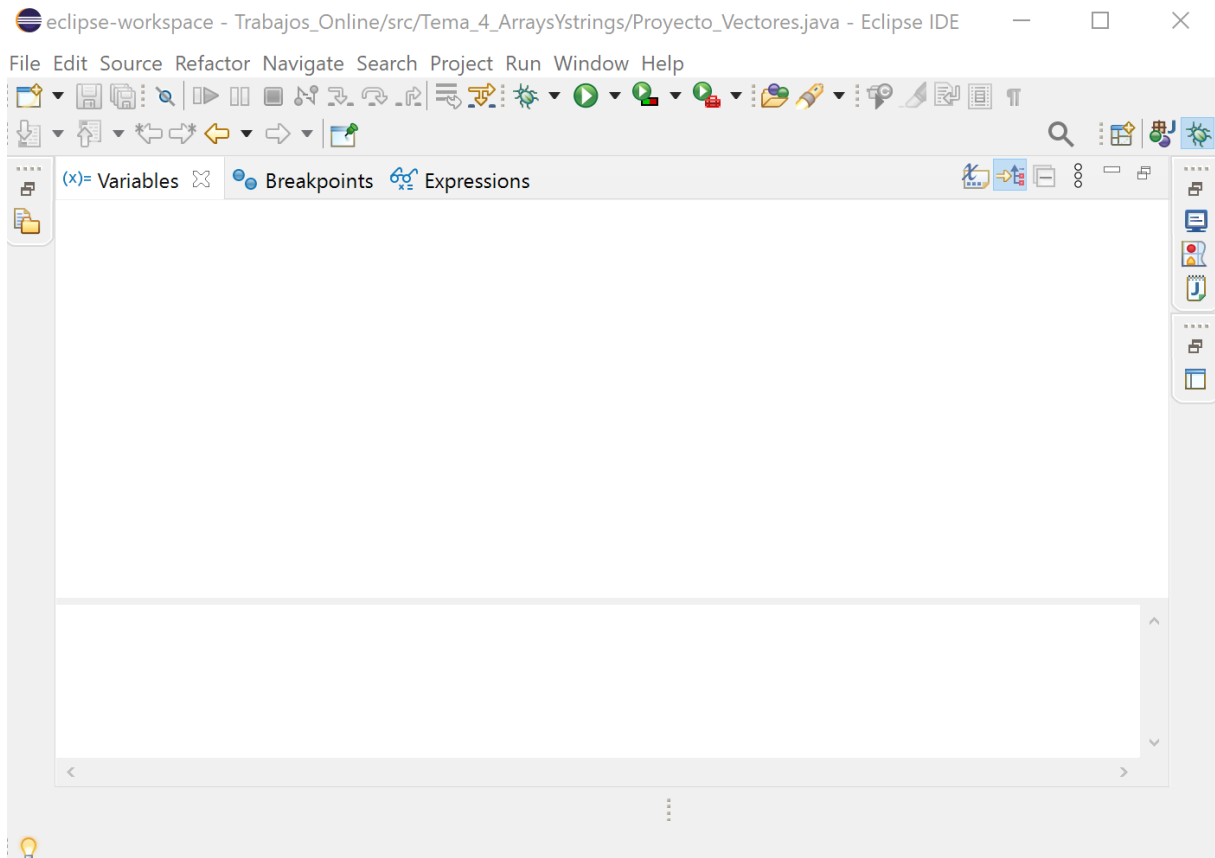
### 3.2.1 Mensajes de error

No procede.

### 3.2.2 Ayudas contextuales

No procede

### 3.3 Ayuda de desarrollo



A continuación, explicaré la herramienta DEBUG con la que se puede entender mejor este apartado.

#### 3.3.1 Mensajes de error

No procede.

#### 3.3.2 Ayudas contextuales

No procede

### 3.3.3 Debug

En este apartado de ayuda al desarrollo, he considerado muy conveniente desarrollar una explicación sobre la herramienta DEBUG, ya que probablemente sea la herramienta de apoyo más eficaz y fácil de usar de Eclipse.

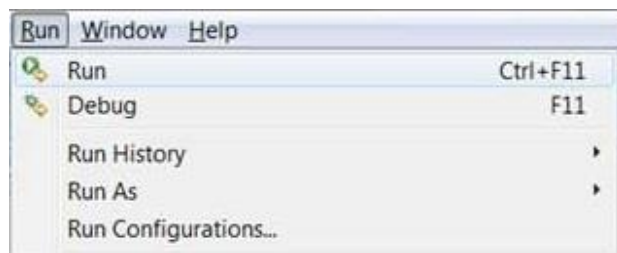
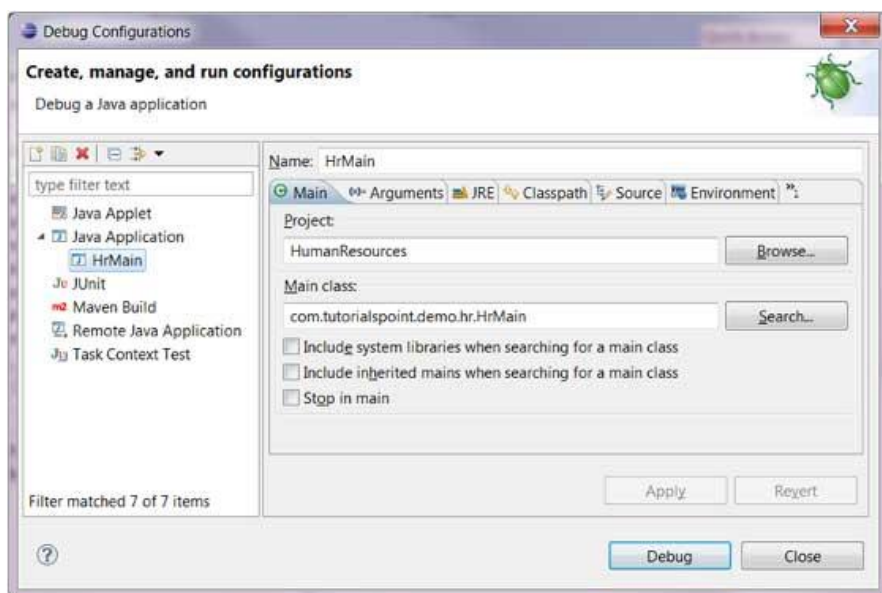
En el Explorador de paquetes, para depurar programas Java, siga estos pasos:

- Haga clic en la función principal contiene la clase de Java
- Seleccione Depurar Como> Java Application

Esta operación se puede realizar también a través de las teclas de acceso directo, tecla de acceso combinación Alt + Shift + D, J.

La operación anterior crea una nueva [configuración de depuración \(configuración de depuración\)](#), y utilizar esa configuración para iniciar las aplicaciones Java.

Si se ha creado la configuración de depuración, se puede elegir la depuración conveniente y seleccionar la clase correspondiente por el menú Ejecutar y haga clic en el botón para iniciar la aplicación Java.

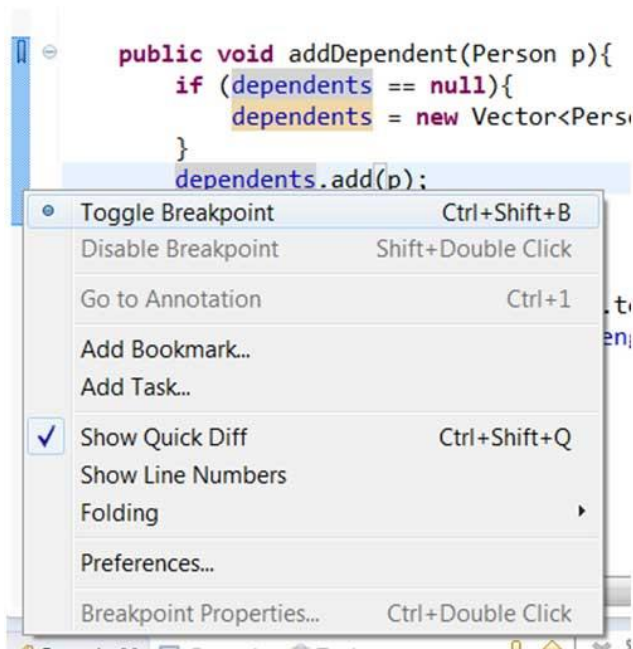


Antes de volver a cargar utilizando el modo de depuración en acceso directo (F11).

Cuando se utiliza el modo de depuración en el programa Java que pide al usuario cambiar a la perspectiva de depuración.

La perspectiva de depuración proporciona una descripción de otros puntos de vista parte de resolver problemas.

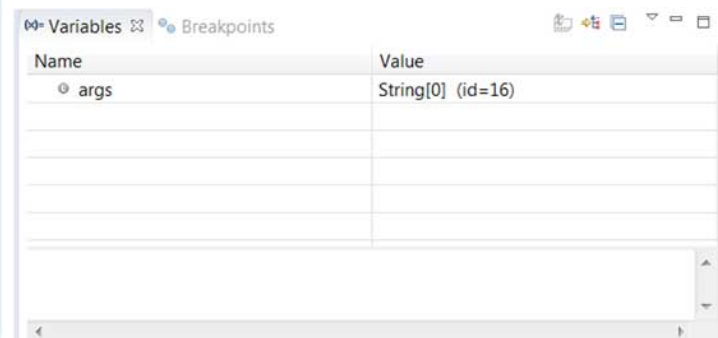
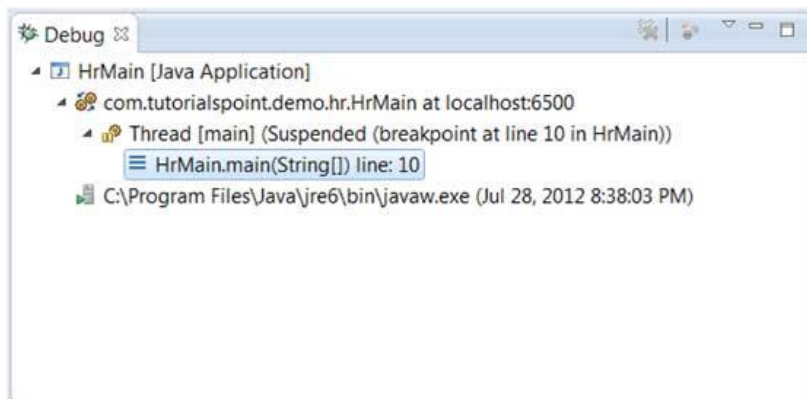
El mismo editor de Java puede establecer puntos de interrupción. Haga clic en la barra de marcadores en el editor y seleccione Alternar punto de corte (toggle point) para definir el punto de interrupción de depuración.



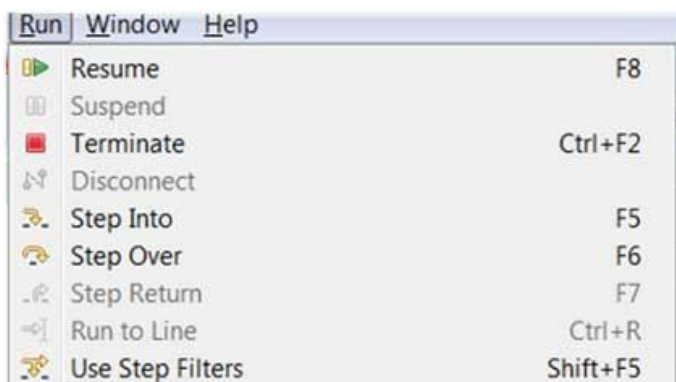
Los puntos de interrupción se pueden ver en la barra de marcadores. También se puede ver en la vista Puntos de interrupción (tira vertical a la izquierda del editor)

Cuando el programa se ejecuta, la etiqueta de código del punto de interrupción JVM se bloquea, entonces se puede ver el uso de memoria y la ejecución del programa de control.

Cuando el programa se cuelga, depuración (debug) se puede examinar la pila de llamadas a la derecha, es decir, examinar tanto las variables como los toggle points.

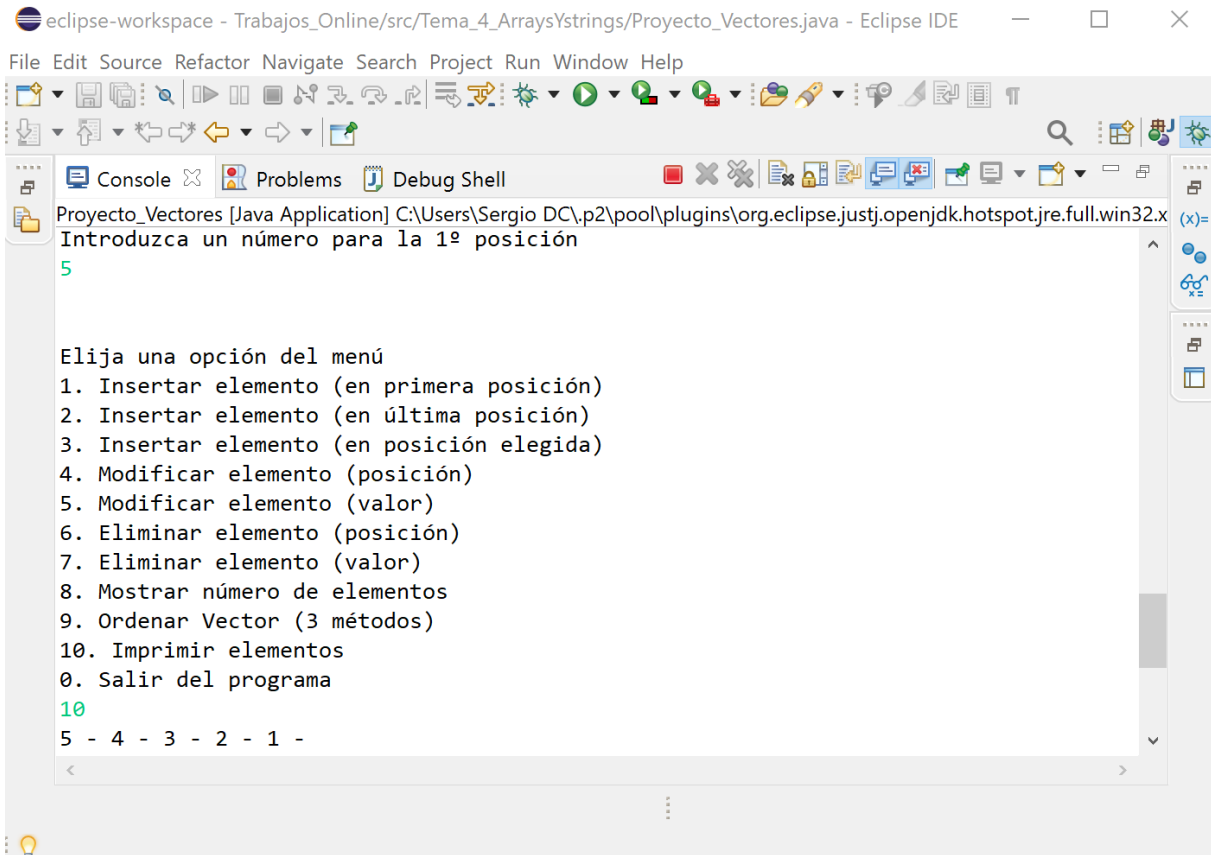


Adicionalmente, en el menú de RUN se muestran las opciones de continuar (Reanudar), saltar (Paso a paso) de la línea de código en función (Paso a paso) y similares.



La imagen de arriba muestra las teclas de acceso directo Resume, el paso hacia y Step Over y otras asociaciones.

### 3.4 Consola



The screenshot shows the Eclipse IDE interface with the console window open. The console output is as follows:

```

Proyecto_Vectores [Java Application] C:\Users\Sergio DC\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x
Introduzca un número para la 1ª posición
5

Elija una opción del menú
1. Insertar elemento (en primera posición)
2. Insertar elemento (en última posición)
3. Insertar elemento (en posición elegida)
4. Modificar elemento (posición)
5. Modificar elemento (valor)
6. Eliminar elemento (posición)
7. Eliminar elemento (valor)
8. Mostrar número de elementos
9. Ordenar Vector (3 métodos)
10. Imprimir elementos
0. Salir del programa
10
5 - 4 - 3 - 2 - 1 -

```

Frecuentemente, los programas tienen necesidad de comunicarse con un usuario. Los usuarios son las personas que interactúan con los programas.

Esta comunicación tiene dos sentidos; entrada y salida (input & output):

- **Entrada:** El usuario da información al programa.
- **Salida:** el programa da información al usuario.

De esta comunicación, se encarga la consola, y sirve para ver el resultado de la ejecución de un programa.

#### 3.4.1 Mensajes de error

No procede.

#### 3.4.2 Ayudas contextuales

No procede

### 3.5 Barra de menús

eclipse-workspace - Trabajos\_Online/src/Tema\_4\_ArraysYstrings/Juego\_Hundir\_La\_Flota.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Tradicionalmente en todos los programas solemos encontrar arriba la típica “Barra de Menús”, y en el caso de Eclipse, es tal cual se muestra en la foto.

Aparte de estos 10 menús desplegables, y como en la mayoría de programas, bajo esta barra se incluye otra “sub-barra” que contiene botones directos con las funciones más frecuentemente usadas a la hora de desarrollar un programa en este caso, funciones que ya se encuentran dentro de los diferentes menús desplegables, pero como decimos, se sitúan también aquí para ahorrarle tiempo al usuario en pulsar más botones de la cuenta y perder repetidamente su tiempo en menús desplegables. Estas funciones de acceso directo son totalmente personalizables, es decir, se pueden eliminar y agregar otras diferentes de los menús desplegables.

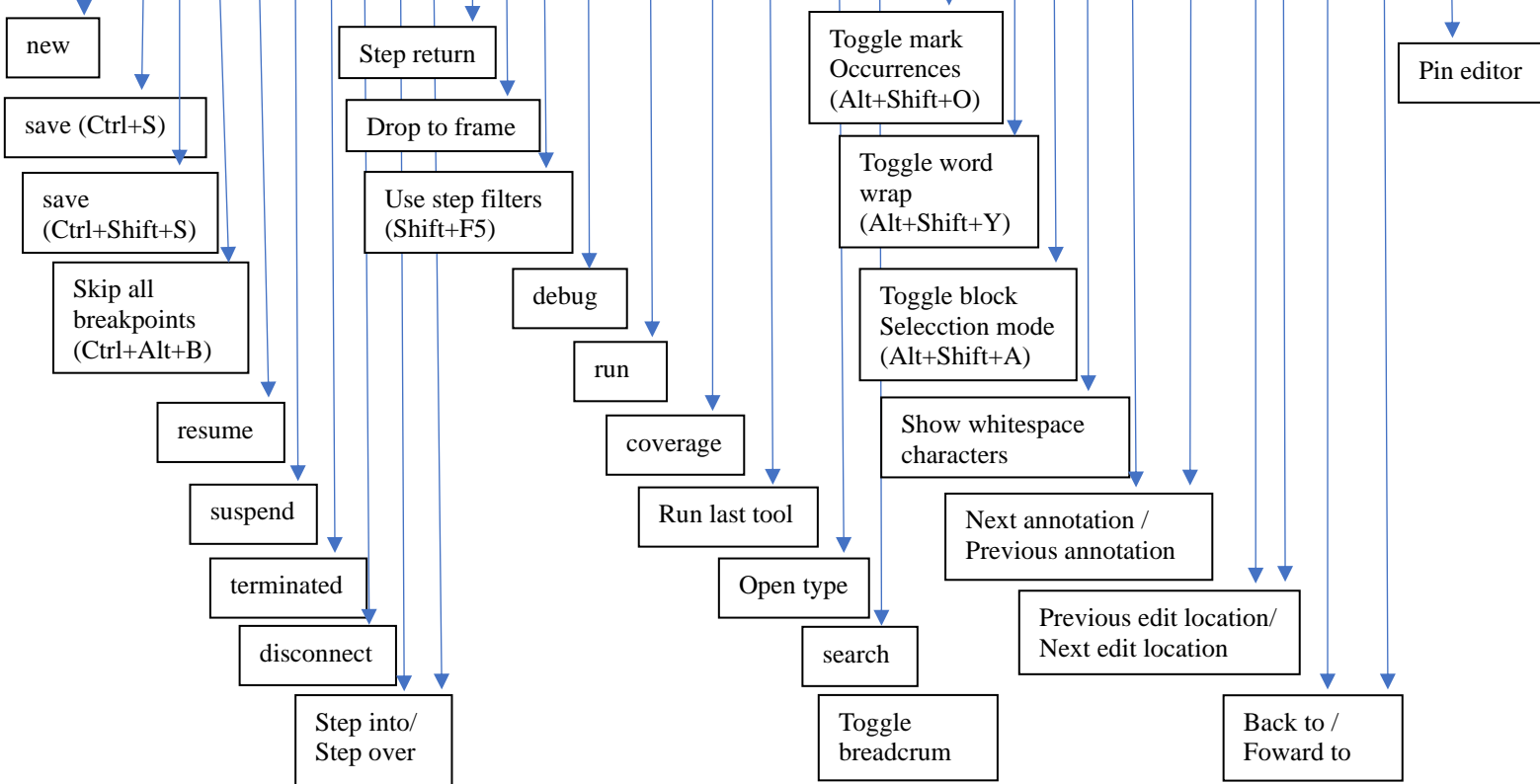
#### 3.5.1 Mensajes de error

No procede.

#### 3.5.2 Ayudas contextuales

eclipse-workspace - Trabajos\_Online/src/Tema\_4\_ArraysYstrings/Juego\_Hundir\_La\_Flota.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help





### 3.5.2.1 File

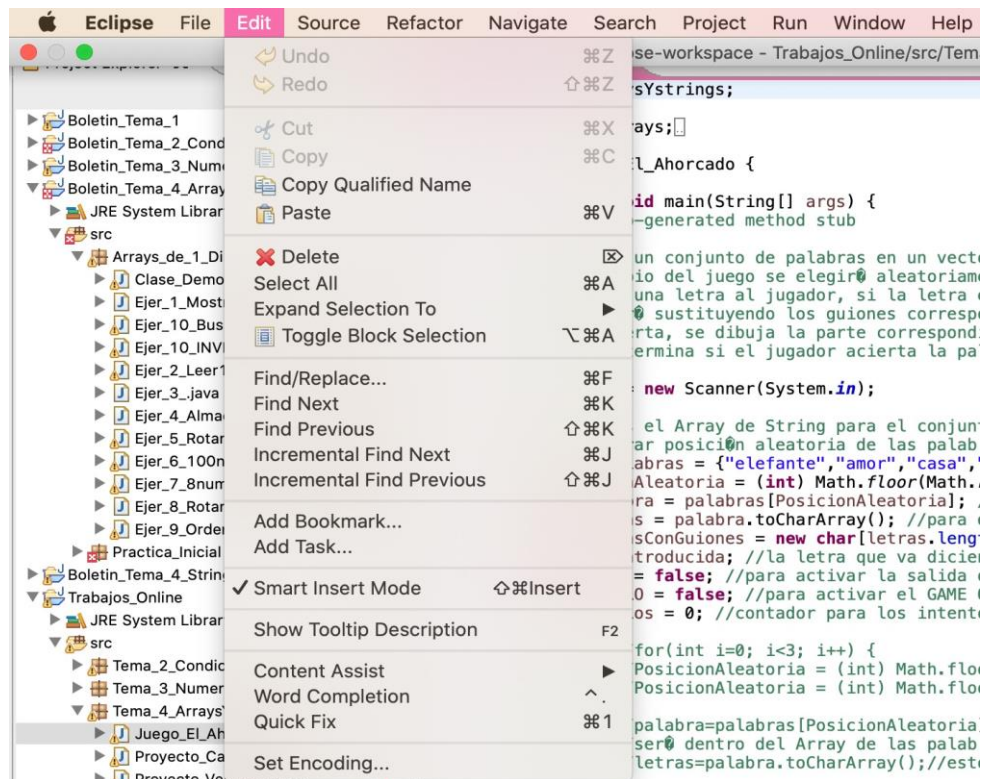


Más conocido en castellano como "Archivo", es el clásico menú que tanto conocemos ya que engloba las diferentes posibilidades de tratamiento del archivo tales como

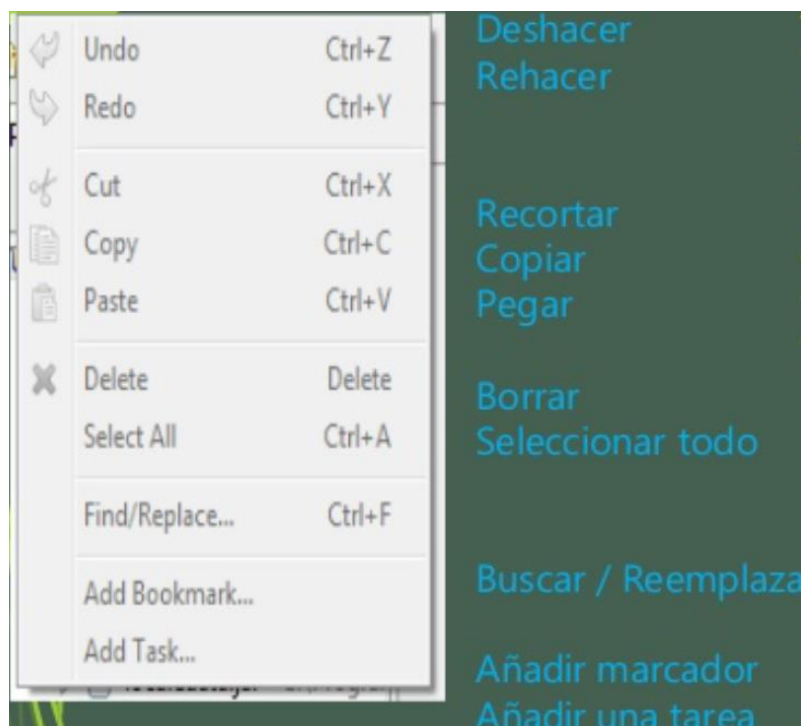
En el entorno de desarrollo Eclipse, todo archivo se almacena dentro de un proyecto. Esto quiere decir que todo documento, carpeta, archivo de código fuente (.java) y código compilado (.class) tiene que estar contenido dentro de un proyecto.



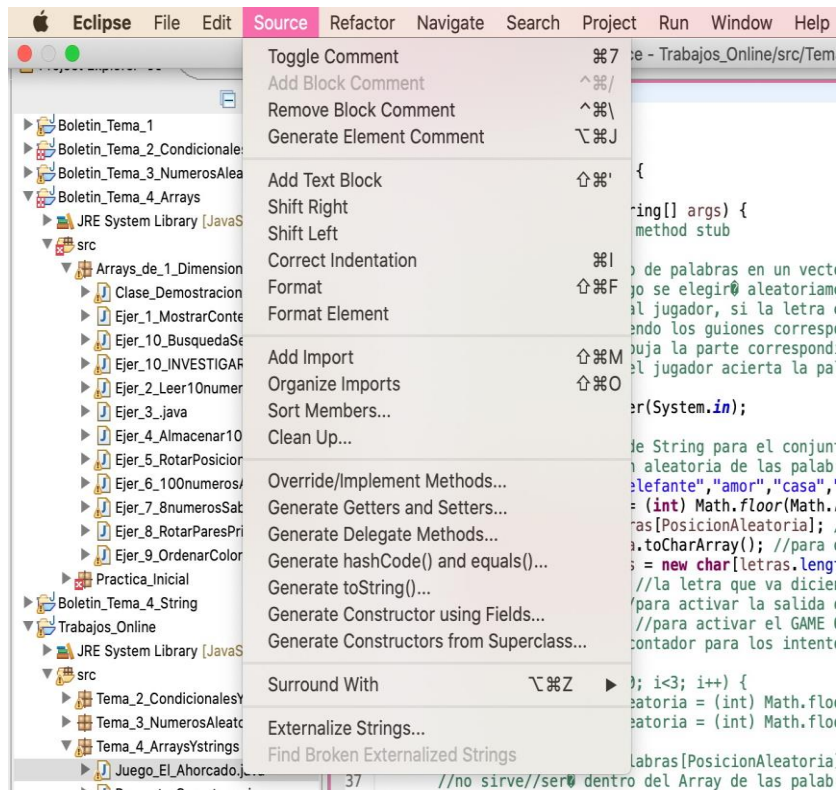
### 3.5.2.2 Edit



El menú EDIT sirve para crear o editar programas Java, un “Java Project” debería ser creado. Nótese que dentro de un proyecto de Java también puede almacenarse toda la información relacionada con el proyecto (la cual no tiene por qué reducirse estrictamente a código fuente, sino que también puede contener documentación y otros archivos relacionados).

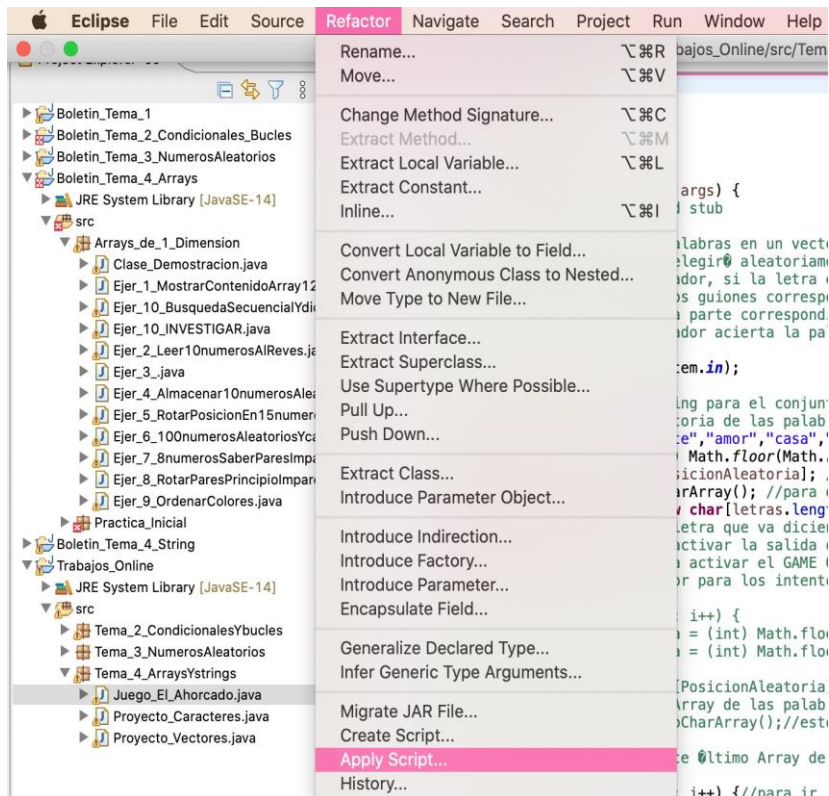


### 3.5.2.3 Source



En Source se almacenan los archivos de código fuente de Java (.java), de manera que Eclipse sepa dónde encontrarlos y pueda realizar la compilación automática de los mismos cada vez que un archivo sea guardado.

### 3.5.2.4 Refactor

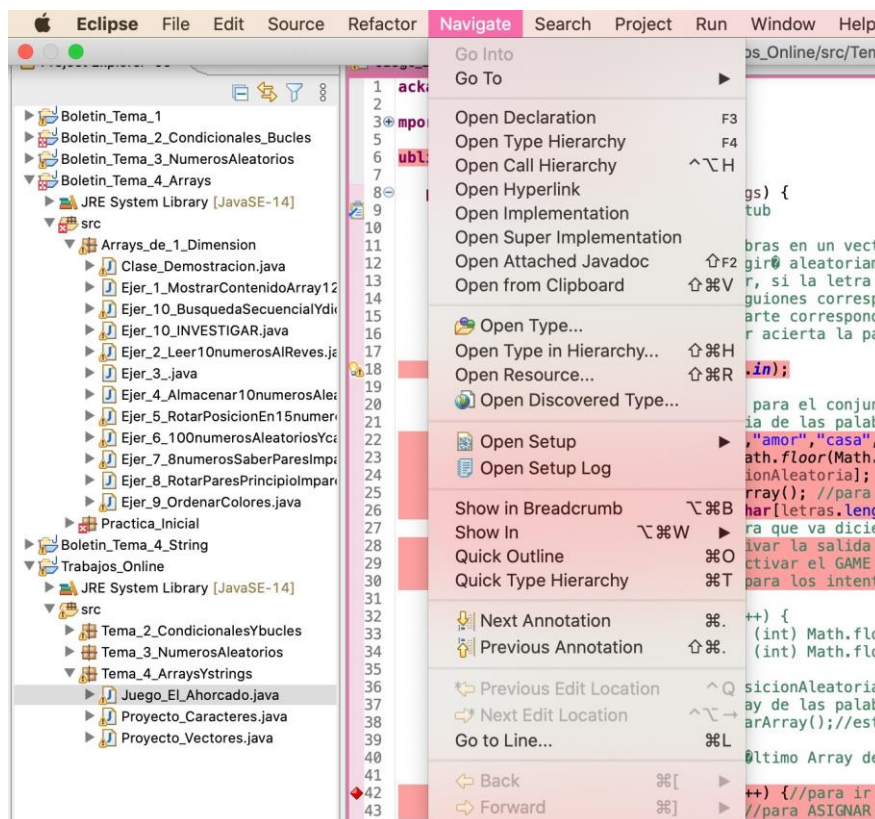


Muchas veces tenemos código ya hecho que está funcionando. También muchas veces necesitamos tocar ese código para que haga más cosas, para hacer más eficiente un algoritmo, más vistosa la salida del programa, porque tenemos otro proyecto que se parece, etc, etc.

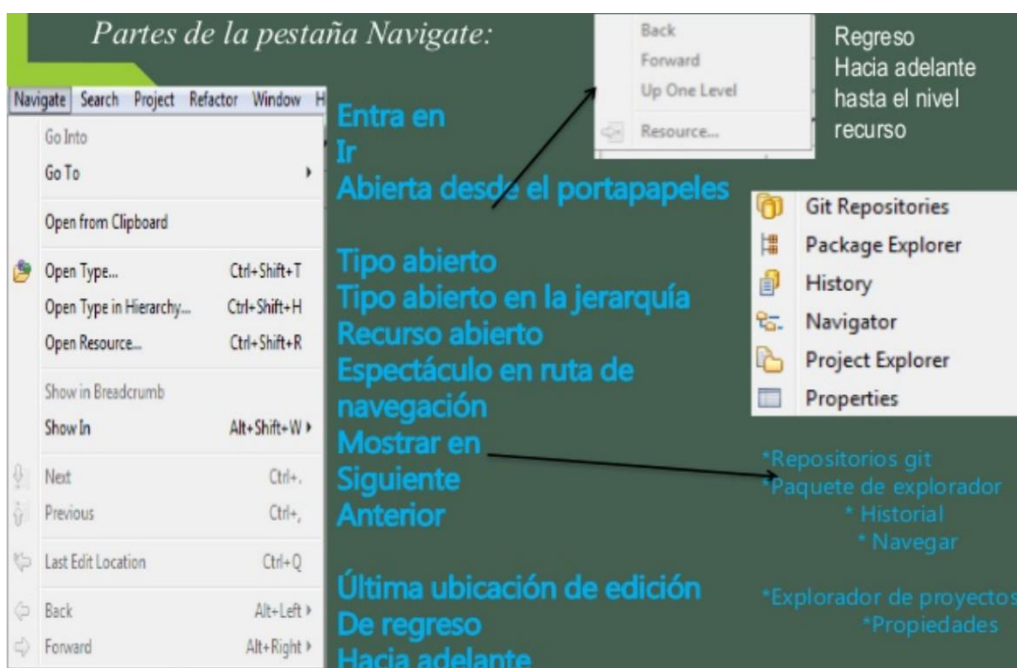
Rename...	Alt+Shift+R	Rebautizar
Move...	Alt+Shift+V	Movimiento
Android		
Change Method Signature...	Alt+Shift+C	Método de cambio de la firma
Extract Method...	Alt+Shift+M	Método extracto
Extract Local Variable...	Alt+Shift+L	Extraer variable local
Extract Constant...		Extraer constante
Inline...	Alt+Shift+I	En línea
Convert Local Variable to Field...		Convertir variable local de campo
Convert Anonymous Class to Nested...		Convertir clase anónima a anidado
Move Type to New File...		Tipo de cambio a nuevo archivo
Extract Superclass...		Superclase extracto
Extract Interface...		Interfaz extracto
Use Supertype Where Possible...		Súper tipo utilizar cuando sea posible
Push Down...		Empuje hacia abajo
Pull Up...		Levantar
Extract Class...		Clase extracto
Introduce Parameter Object...		Objeto de parámetro introduce
Introduce Indirection...		Indirección introduce
Introduce Factory...		Fábrica introduce
Introduce Parameter...		Parámetro introduce
Encapsulate Field...		Campo encapsular



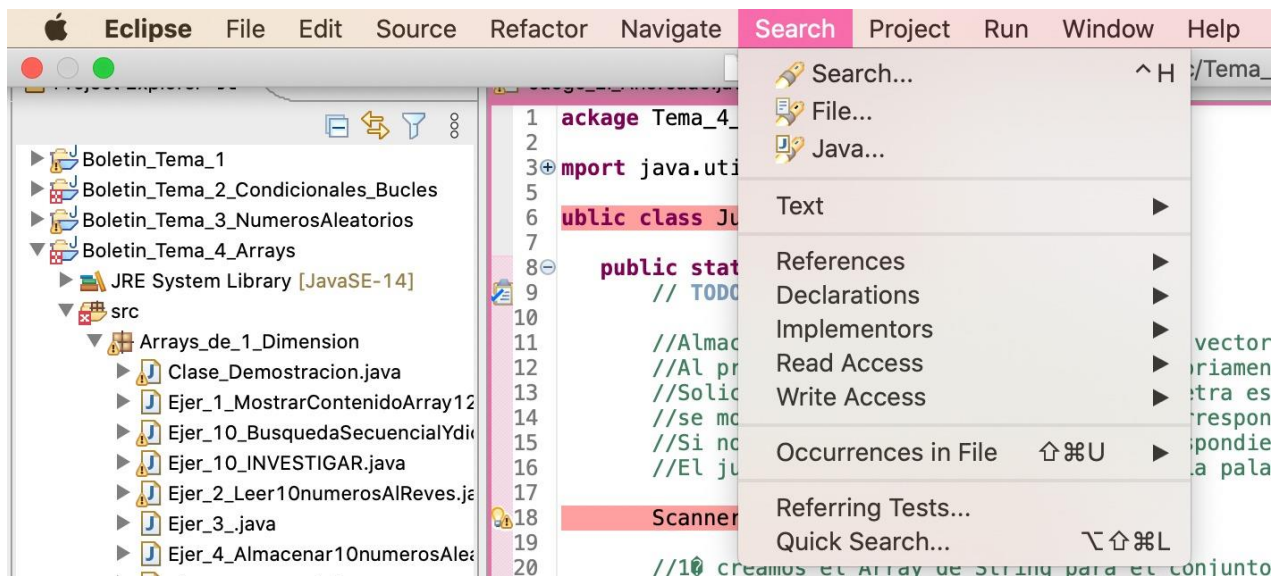
### 3.5.2.5 Navigate



La vista Navigator de la perspectiva Resource y la vista Package Explorer de la perspectiva Java contienen un directorio con el nombre del proyecto. Como se comento anteriormente, se puede pasar de una perspectiva a otra pulsando sobre los botones de la barra vertical izquierda.



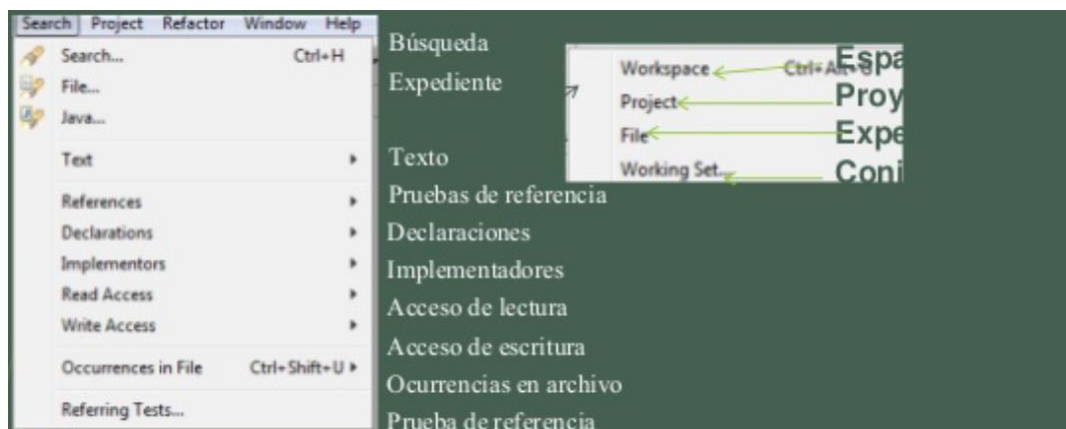
### 3.5.2.6 Search



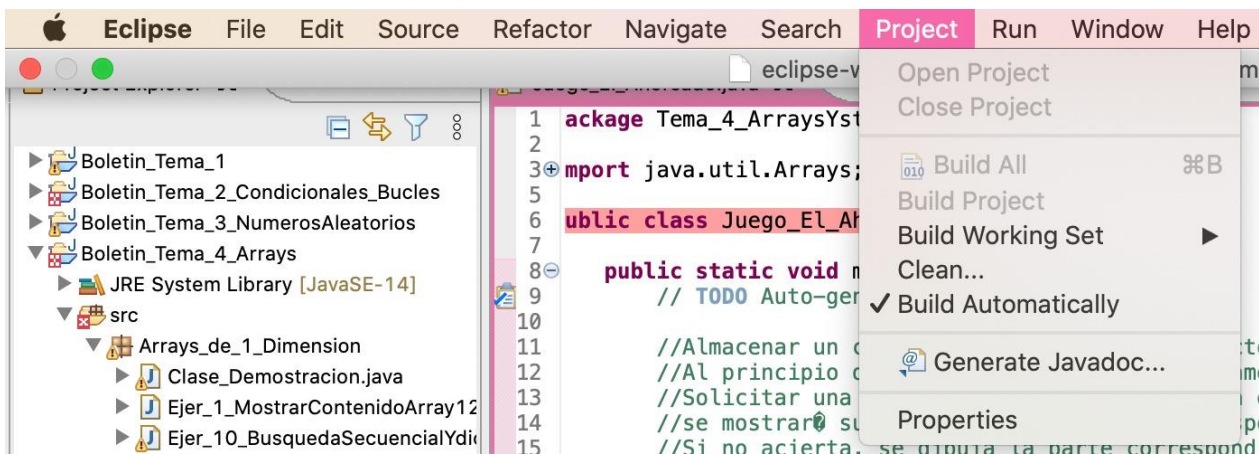
Para realizar una búsqueda dentro de Eclipse, el menú "Search" de la barra superior de menús debería ser seleccionado. También se pueden lanzar búsquedas pulsando el icono de linterna.

Hay varios tipos de búsqueda dentro de Eclipse.

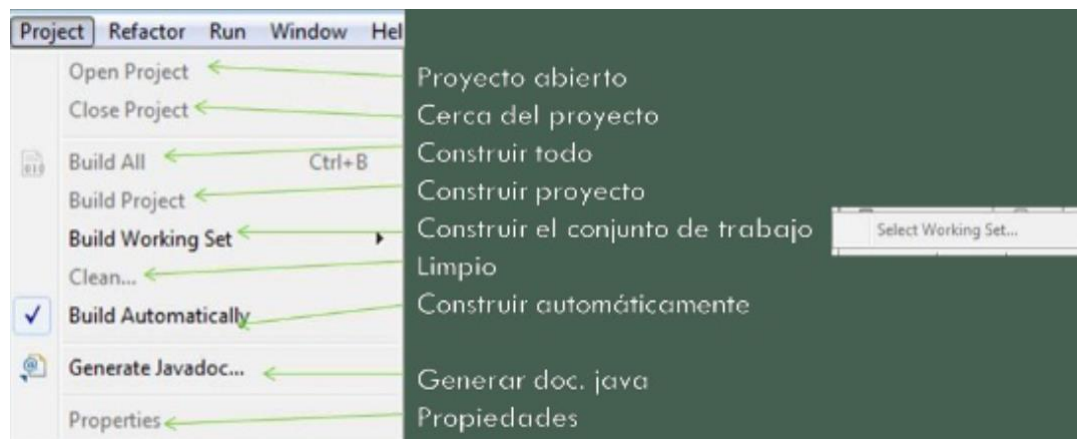
La búsqueda de archivos "File Search" es una búsqueda textual que puede ser ejecutada sobre archivos de todo tipo, siendo equivalente a una búsqueda tradicional.



### 3.5.2.7 Project



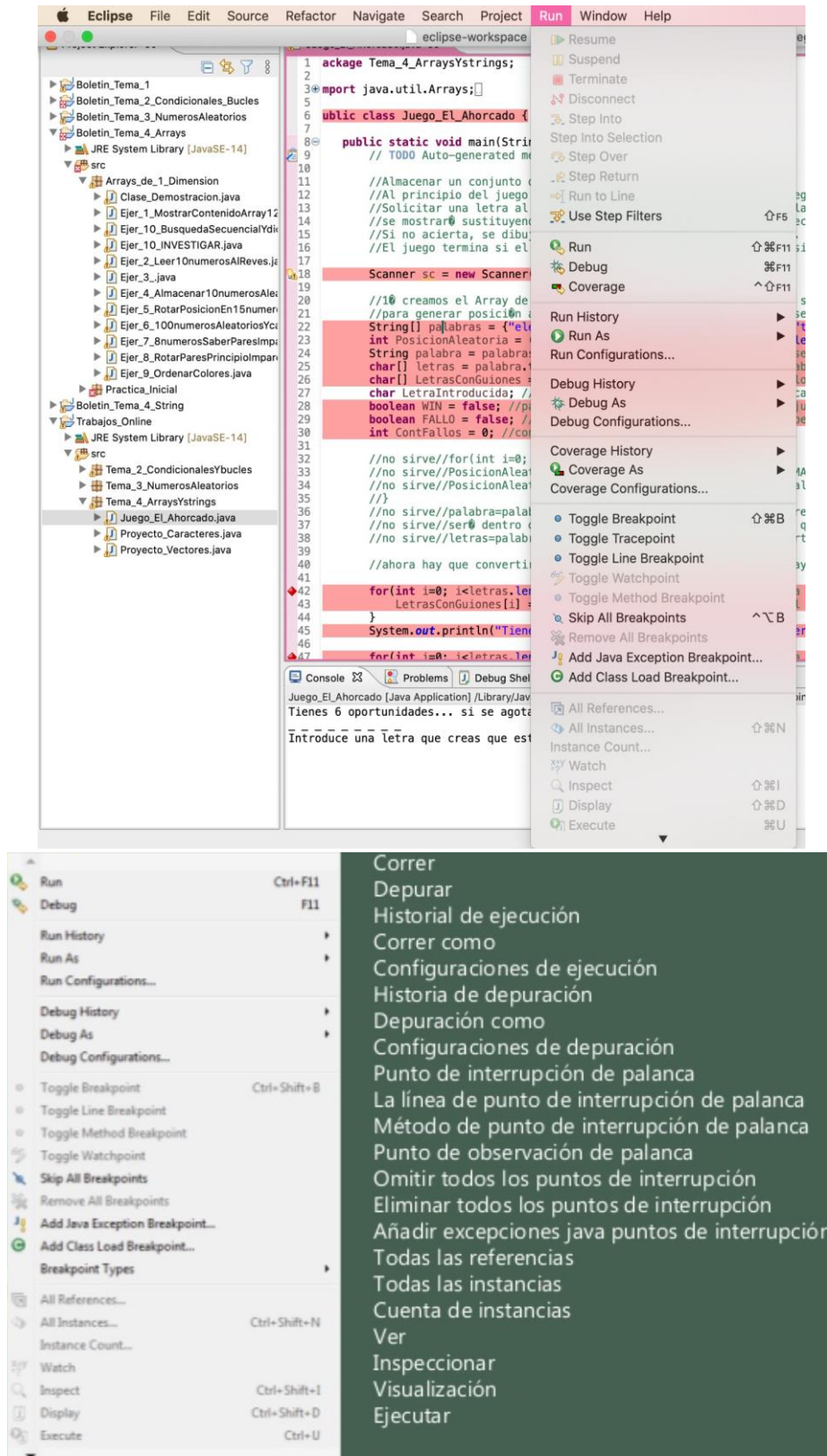
Manejo de archivos, manejo de texto, editores de texto. Ó lo deberían crearse para almacenar documentos y otros archivos, pero no código Java que se desee compilar. Por ello, siempre que se creen archivos ".java" sería recomendable crear un "Java Project".



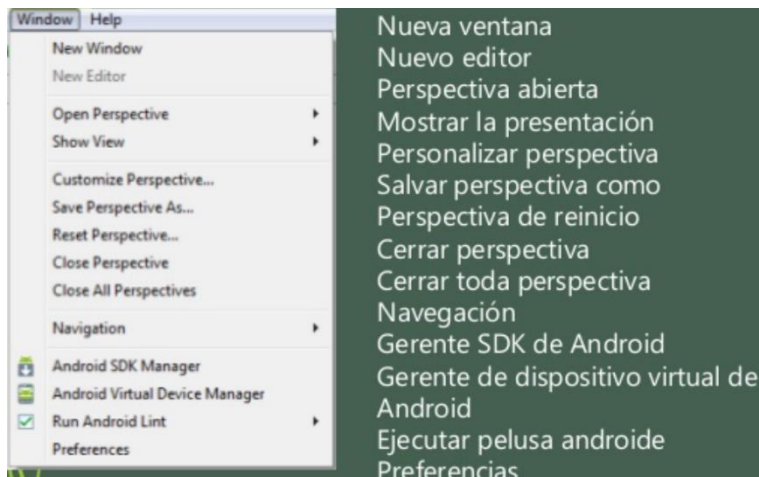
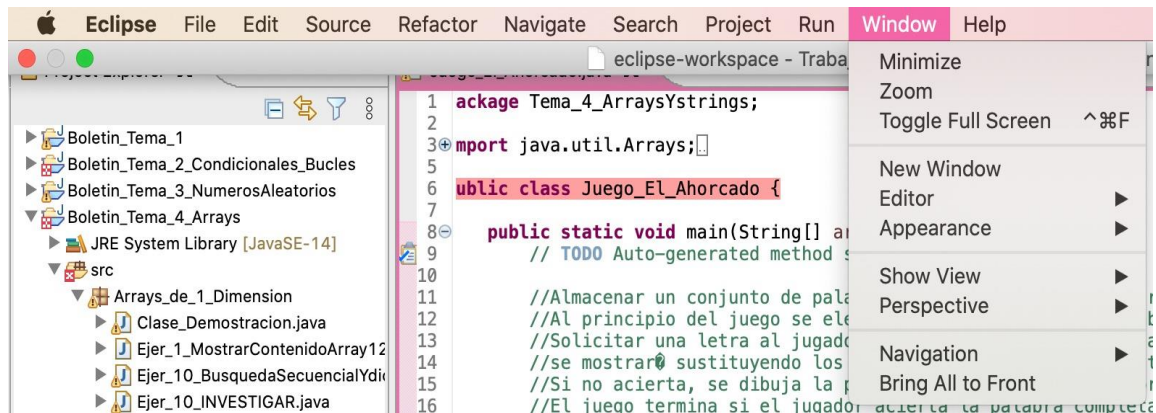


### 3.5.2.8 Run

Es la barra de opciones que puedes utilizar para correr el programa que estás creando en eclipse.



### 3.5.2.9 Window



Es la barra de opciones que contiene eclipse para poder crear otros archivos tales como personalizar perspectiva.

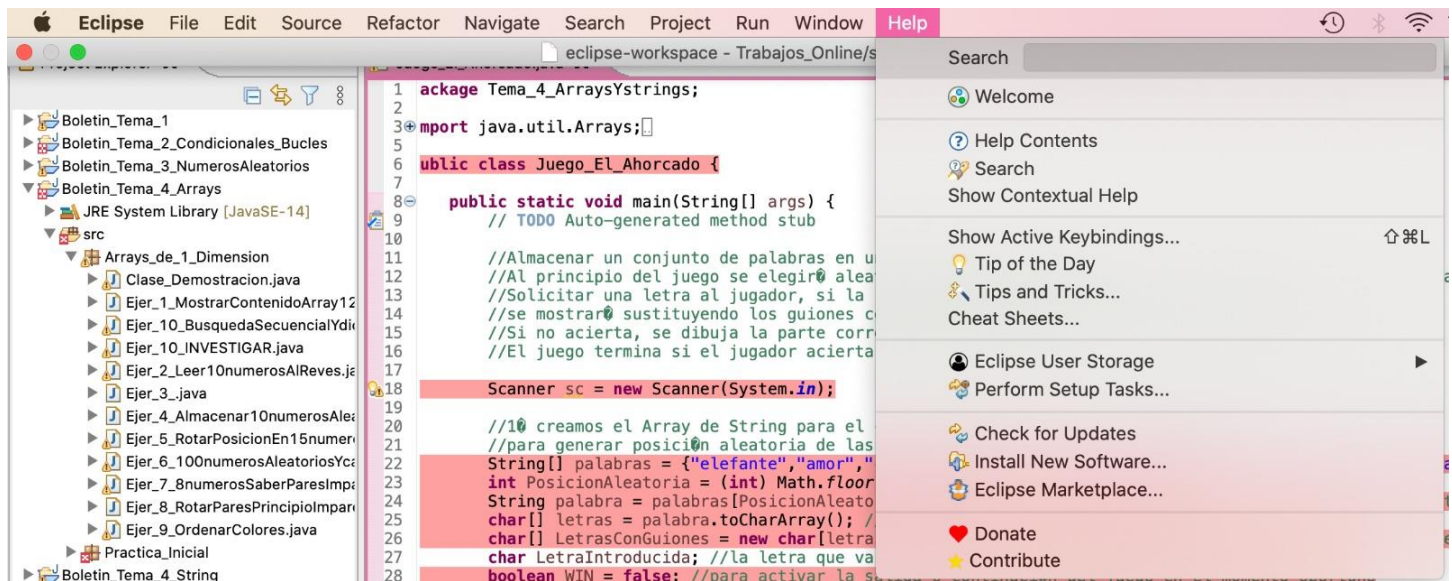
Una perspectiva de Eclipse es una agrupación de vistas y editores de manera que den apoyo a una actividad completa del proceso de desarrollo software. Sin embargo, es posible crear perspectivas propias añadiendo nuevas vistas y cambiando su distribución en la pantalla. Las perspectivas pueden seleccionarse haciendo clic en los iconos de perspectiva del lateral izquierdo o eligiendo "Window > Open Perspective" del menú.

Perspectivas:

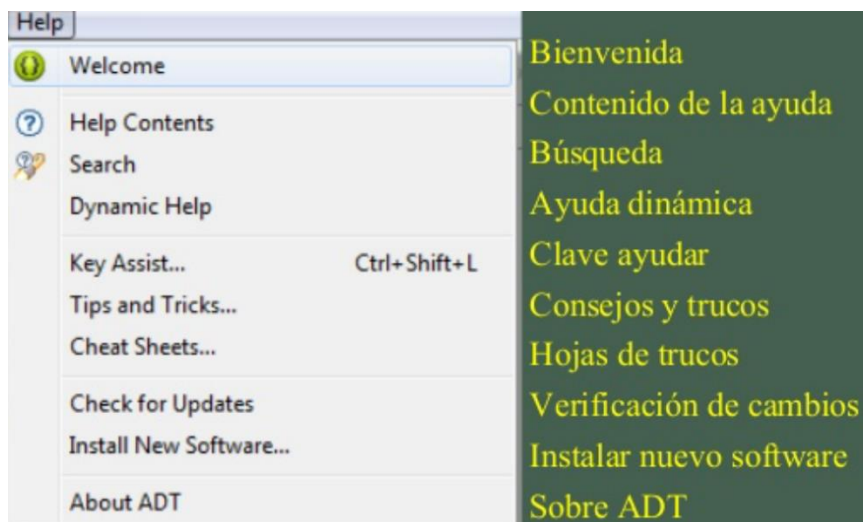
- Resource: esta perspectiva está estrechamente relacionada con el sistema de archivos puesto que representa la localización física de los recursos almacenados dentro de los proyectos Java: esta perspectiva se centra en tareas de programación, mostrando paquetes, clases, métodos y atributos en sus vistas asociadas.
- Plug-in development: la perspectiva de desarrollo de plug-in permite a los desarrolladores añadir nuevos módulos de Eclipse.
- Install/Update: permite gestión de la configuración. Muestra los componentes instalados así como sus versiones y conflictos.
- Debug: relacionada con la tarea de depuración. Se centra en los procesos ejecutados, puntos de ruptura, variables, salida, etc. Java Browsing: esta perspectiva permite ojear rápidamente código, proyectos, paquetes y jerarquías.



### 3.5.2.10 Help



Es la barra de opciones que nos asesora en caso de tengamos algún problema y no sabemos cómo solucionarlo podemos acudir a esta parte de eclipse.



## 4 FAQ

### 4.1 ¿Qué es la plataforma Eclipse?

Quienes descargan la plataforma genérica Eclipse & generalmente por error; se sienten algo confundidos por lo que ven. La plataforma fue concebida como la base genérica de un IDE. Es decir, la plataforma es un IDE sin ningún lenguaje de programación en particular en mente. Puede crear proyectos genéricos, editar archivos en un editor de texto genérico y compartir los proyectos y archivos con un servidor de sistema de versiones simultáneas (CVS). La plataforma es esencialmente una versión glorificada de un navegador de sistema de archivos.

Como usuario final, lo que no ve cuando descarga y ejecuta la plataforma es que la arquitectura está diseñada desde cero para ser extensible. De hecho, todo lo que ve es un complemento, y todo lo que ve se puede modificar, reemplazar o aumentar utilizando varios ganchos. Para trazar una analogía informática, es como el Protocolo de Internet (IP): excesivamente genérico, no muy interesante en sí mismo, pero una base sólida sobre la que se pueden construir aplicaciones muy interesantes.

En Eclipse 3.0 se ha producido un subconjunto cuidadosamente diseñado de la plataforma Eclipse: la plataforma de cliente enriquecido (RCP). A pesar del nombre, esta no es la versión de la plataforma vendida por grandes ganancias a clientes ricos. Esta es la parte de la plataforma que es interesante para aplicaciones que no son de entorno de desarrollo. Pensamos que esta parte era tan interesante que dedicamos casi la mitad del libro solo a RCP.

### 4.2 ¿Dónde consigo e instalo Eclipse?

A continuación, se muestran los tipos de versiones que se pueden descargar desde el sitio de descarga de Eclipse (<http://eclipse.org/downloads>), para aumentar el nivel de cercanía a la secuencia CVS HEAD:

La última versión, utilizada por la mayoría de productos del mercado. Estos lanzamientos tienen poco riesgo de sorpresas, sin embargo, esas compilaciones pueden tener un retraso de hasta 6 meses en lo que se está trabajando ahora. Los números de versión de ejemplo para estas versiones son 2.1.2 y 3.0.

Compilaciones estables (las llamadas compilaciones de hitos), que suelen tener una diferencia de 6 semanas y ofrecen una gran colección de características estables. Estas compilaciones se prueban moderadamente, lo que las hace lo suficientemente confiables para las necesidades de la mayoría de los desarrolladores. Un ejemplo es Eclipse 3.0M4, que se refiere a la compilación del hito 4 de Eclipse 3.0.

La integración se construye, se realiza cada semana y, a veces, más en caso de falla o cuando se acerca la fecha de lanzamiento. Estas compilaciones integran contribuciones de varios subequipos de Eclipse para probar las colaboraciones de varios complementos.

Cada versión o compilación tiene un conjunto correspondiente de notas de compilación y resultados de prueba. Asegúrese de consultar estas notas antes de seleccionar una compilación determinada. Los conjuntos de pruebas automatizados se ejecutan contra las compilaciones nocturnas y de integración. Hacer que una construcción supere las pruebas puede aumentar el nivel de confianza que la gente tiene en ella. En todos los casos, las compilaciones se envían como un archivo comprimido y la instalación es una simple cuestión de descomprimirlas en cualquier lugar de su máquina local.

Alternativamente, es posible que ya haya instalado Eclipse sin saberlo. Muchos productos comerciales se basan en Eclipse, por lo que al instalar estos productos, a menudo se instala una versión determinada de Eclipse. Por lo general, estos productos no tienen reparos en basarse en Eclipse, por lo que puede descubrir fácilmente la ubicación de la instalación de eclipse investigando el directorio de instalación

del producto. Por lo general, el directorio contendrá un subdirectorío llamado eclipse que contiene la instancia de Eclipse incrustada. Mientras ejecuta su producto, puede activar la opción de menú Ayuda> Acerca de ... y luego hacer clic en el icono de Eclipse para ver qué versión de la plataforma se está utilizando.

### **4.3 ¿Cómo instalo nuevos Plug-Ins?**

Su mejor enfoque es utilizar Eclipse Update Manager.

Los desarrolladores de complementos más experimentados, y a veces los editores de complementos perezosos, han aprendido a encontrar el directorio eclipse / plugins e instalar sus complementos allí manualmente. Obviamente, este es un enfoque más peligroso, ya que no se realiza ninguna certificación sobre la idoneidad del complemento; puede depender de otros complementos no disponibles en su instalación. En el caso de conflictos de compatibilidad, no descubrirá hasta que use el complemento que podría romperse.

Puede comparar la instalación de complementos con la instalación de aplicaciones en Windows. Por supuesto, puede instalar sus bibliotecas de vínculos dinámicos (DLL) en el directorio System32 o jugar con la variable de entorno PATH. Pero, ¿cómo va a eliminar la aplicación más adelante cuando ya no la necesite? En Windows, se han diseñado programas de instalación especializados y la desinstalación es fácil a través del menú Inicio. Eclipse Update Manager puede verse como el equivalente de Eclipse de InstallShield y el Registro de Windows combinados.

Para el desarrollo diario y la creación de prototipos de complementos pequeños, es posible que aún tenga la tentación de utilizar el proceso de instalación manual. Podría, pero le recomendamos encarecidamente que no lo haga. Crear una función y un sitio de actualización correspondiente es un juego de niños con los asistentes de PDE y mejorará en gran medida la calidad de su trabajo. Con el tiempo, querrá compartir sus frutos con otros, y tener un sitio de actualización listo desde el principio hará que sea mucho más fácil presumir de sus conocimientos sobre Eclipse con sus amigos y colegas.

### **4.4 ¿Dónde puedo obtener ayuda?**

Eclipse ofrece una amplia ayuda a través de la opción de menú Ayuda> Contenido de la ayuda .... Esto iniciará un servidor Tomcat para que se ejecute desde dentro de su Eclipse y se abrirá un navegador para mostrar el contenido. El navegador de ayuda tiene un útil motor de búsqueda para buscar en toda la documentación de Eclipse. A veces, este texto de ayuda se denomina de forma confusa ayuda en línea, quizás porque la ayuda se replica en línea en el sitio web de Eclipse. Sin embargo, su ayuda local también funcionará cuando esté desconectado de la red.

Alternativamente, se puede invocar una ayuda más enfocada y sensible al contexto en cualquier momento presionando F1. El widget seleccionado actualmente tiene la oportunidad de cumplir con la solicitud de ayuda. Su árbol de contención se camina hacia arriba hasta que se pueda amueblar una ayuda localizada adecuada. Aparecerá una ventana emergente cerca del mouse y se proporcionan enlaces a la documentación. Si la ayuda no parece útil, recuerde cómo funciona la ruta de búsqueda de la ayuda F1 y no dude en ingresar un informe de error para asegurarse de que la ayuda se proporcione en el nivel adecuado.

Seleccione Ayuda> Hojas de referencia ... para obtener instrucciones paso a paso sobre tareas comunes. Los pasos tienen vínculos convenientes a ayuda relacionada con el tema.

Si no encuentra la respuesta que necesita en el sistema de ayuda, hay muchas más fuentes de información sobre Eclipse. En eclipse.org encontrará una gran cantidad de recursos, incluidos artículos de expertos de Eclipse, grupos de noticias, listas de correo y listas de libros y tutoriales. Quizás lo mejor de todo es que el código fuente de Eclipse es el lugar más confiable para buscar información.

## 4.5 ¿Cómo puedo reportar un error?

Los proyectos de código abierto exitosos prosperan gracias al apoyo de su comunidad. La forma más fácil de contribuir a la comunidad es informar errores. Si tiene problemas con Eclipse o quiere sugerir una forma de mejorarlo, diríjase a <https://bugs.eclipse.org>. Deberá crear una cuenta antes de ingresar un informe, pero cualquiera puede explorar la base de datos de errores existente sin registrarse. La página <https://bugs.eclipse.org> tiene mucha información útil sobre cómo ingresar un buen informe de errores. Siga los consejos con atención, porque la calidad y claridad de un informe de error marca una gran diferencia en la rapidez con la que los desarrolladores pueden rastrear y resolver el problema.

Si su error implica un bloqueo, una congelación o incluso una pausa larga sin capacidad de respuesta, consulte Cómo informar de un punto muerto para obtener más detalles sobre qué tipo de información proporcionar.

## 4.6 ¿Cómo y cuándo guardo el workspace?

Si está utilizando el entorno de trabajo Eclipse IDE, no lo hace. Cuando se apaga, el banco de trabajo guardará automáticamente el espacio de trabajo. El espacio de trabajo también realizará sus propios guardados periódicos del espacio de trabajo, llamados instantáneas, de vez en cuando. Tenga en cuenta que la información más esencial en el espacio de trabajo; como los archivos y carpetas recién creados dentro de Eclipse; siempre se almacenan en el disco inmediatamente. Guardar el espacio de trabajo simplemente implica almacenar metadatos, como marcadores, y su imagen en memoria de los proyectos. El espacio de trabajo está diseñado de modo que si un usuario saca el cable de alimentación de la computadora de la pared en cualquier momento, el árbol de recursos seguirá estando en buen estado para que el espacio de trabajo pueda reiniciarse en un estado coherente con una pérdida mínima de información.

No obstante, es posible que su complemento solicite explícitamente un guardado o una instantánea del espacio de trabajo. Si está escribiendo una aplicación RCP, es responsable de invocar mínimamente guardar antes del cierre.

## 4.7 ¿Cómo puedo extender el JDT?

El primer paso es leer Ayuda> Contenido de la ayuda ...> Guía para desarrolladores de complementos de JDT. Esta guía proporciona una descripción excelente de cómo las herramientas de desarrollo de Java mantienen un modelo de un programa de Java y qué se puede hacer con él.

El complemento `org.eclipse.jdt.core` proporciona una API para consultar y manipular programas Java. Sus principales capacidades incluyen

- Analizar y compilar el código fuente de Java
- Manipulación de archivos fuente Java, utilizando varios modelos de objetos
- Evaluar (ejecutar) fragmentos de código
- Buscar, formatear e invocar Content Assist en archivos de origen

El complemento `org.eclipse.jdt.ui` es responsable de todos los elementos de la interfaz de usuario de las herramientas de desarrollo de Java, incluidas todas las vistas de exploración de Java y el editor de Java. El complemento también proporciona API en varias áreas para permitir que otros complementos personalicen o amplíen las herramientas de desarrollo de Java:

- Clases de acción para agregar acciones JDT a vistas en otros complementos
- Exportar código a JAR

- Participación en refactorizaciones
- Editores de texto Java y desplazamientos de texto
- Componentes del asistente para crear proyectos y archivos Java

Además de estos dos complementos principales de JDT, otros complementos también proporcionan API relacionadas con el desarrollo de Java. El complemento `org.eclipse.jdt.debug` proporciona soporte para iniciar y depurar programas Java. El complemento `org.eclipse.jdt.launching` proporciona API para instalar y configurar máquinas virtuales Java y para buscar el código fuente correspondiente a una biblioteca Java. Finalmente, el complemento `org.eclipse.jdt.unit` tiene soporte para ejecutar pruebas JUnit y para monitorear programáticamente la ejecución de pruebas JUnit.

#### **4.8 ¿Qué es Junit?**

JUnit es un marco de prueba simple que se envía con Eclipse. JUnit se puede usar para escribir pruebas para programas Java desarrollados con JDT y la plataforma también lo usa mucho para probar complementos.

Al inspeccionar los complementos de Eclipse en CVS, notará bastantes proyectos como `org.eclipse.core.tests.resource`. Todos estos proyectos contienen pruebas PDE JUnit para realizar pruebas de regresión en complementos.

## 5 ANEXOS

No procede.

## 6 GLOSARIO

Término	Descripción
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
Application Programming Interface (API)	A set of routines, including functions, methods, and procedures, that exposes application functionality to support integration and extend applications.
ATL	ATLAS Transformation Language
ATLAS	<a href="#">Atlantic Data Systems</a>
BIRT	Business Intelligence and Reporting Tool
CDT	C/C++ Development Toolkit
CDO	Connected Data Objects
Client API & web UI	COSMOS components used to access or visualize data from multiple MDRs or Data Managers
Context	TBD
COSMOS	Community-driven Systems Management in Open Source
CU	Configuration Unit
CVS	Concurrent Versions System
Data Adapter	Is a product-specific component that is implemented by each product to share its data
Data Broker	This is the component where all the web services that share data register themselves
Data Collection Runtime	TBD
Data Manager	Is a component that hosts one or more Data Adapters, and handles communication with the COSMOS environment
Distro	Is a set of (i.e. open source) software components packaged into a working whole and distributed to the user community
Domain model	The root of an SML-IF document. Contains a set of definition and instance documents <a href="#">SML Spec</a>
Definition Documents	The subset of documents in a model that describes the schemas and rules that govern the structure and content of the model's documents. The SML-IF specification defines two kinds of definition documents - XML Schema documents that conform to SML's profile of XML Schema, and rule documents that conform to SML's profile of Schematron. <a href="#">SML Spec</a>
DMS	Data Mediator Service

ECF	Eclipse Communication Framework
EIP	Enterprise Integration Pattern
EL	Expression Language (e.g. JSF EL)
EMF	Eclipse Modeling Framework
EMFT	Eclipse Modeling Framework Technology
EMP	Eclipse Modeling Project
EMS	Enterprise Messaging System (?)
ESB	Enterprise Service Bus (?)
FAQ	Frequently Asked Questions
GEF	Graphical Editing Framework
GMF	Graphical Modeling Framework
GUI	Graphical User Interface
ID	Identifier
IDE	Integrated Development Environment
Instance Documents	The subset of documents in a model that describe the structure and content of the modeled entities. <a href="#">SML Spec</a>
IU	Installable Unit
IWG	Industry Working Group
J2SE	Java 2 Standard Edition
JAX-WS	Java API for XML Web Services
JDK	Java Development Kit
JDT	Java Development Tooling
JET	Java Emitter Templates
JRE	Java Runtime Environment
JSF	JavaServer Faces
JSF EL	JSF Expression Language



Management Annotation	TBD
Management Domain	A COSMOS component used to bootstrap the COSMOS system
Management Interfaces	TBD
M2M	Model-to-Model or Machine-to-Machine
M2M-IWG	Machine-to-Machine Industry Working Group
Metadata Registry	A repository that contains access and configuration information about the structure of data that enables a program to process information.
MDE	Model-Driven Engineering
MQTT	MQ Telemetry Transport (a protocol used by Eclipse Paho)
MWT	Multiplatform Widget Toolkit
NAB	Native Application Builder
OCL	Object Constraint Language
OS	Operating System or sometimes Open Source
OSGi	(originally Open Services Gateway initiative)
PDE	Plug-in Development Environment
PTP	Parallel Tools Platform
Query API	The query API provides a web service interface to the data store(s). Its binding will be constructed in a manner analogous to the Data Collection adapters where extensions can be created to implement any desired query mechanism without requiring dependence on the type or location of the underlying data store.
QVT	Query/View/Transformation
RAP	Rich AJAX Platform
RCP	Rich Client Platform
RDT	Remote Development Tools
Resource domain	A set of definition and/or template documents that can be used to build define a domain
Resource Model editor	An editor that can create domain model instances, based on a set of predefined templates
RSE	Remote System Explorer
SCA	Service Component Architecture

SDK	Software Development Kit
SDO	Service Data Objects
SML	Service Modeling Language
SML-IF	SML-Interchange Format. This is a way to aggregate and share SML definitions and instances, identifying the aggregation and defining document aliases and rule bindings. See also the <a href="#">SML Spec</a>
SML-IF instance validation	The process of validating the content of an SML-IF resource, including SML instance documents contained by the SML-IF resource, applying any Schematron rules defined as a definition document and bound to SML instances. See also the <a href="#">SML Spec</a>
SML Repository APIs	A set of APIs used to retrieve SML documents from a repository implementation. COSMOS provides a file-system based repository.
SML validation	The action of validating SML documents. The SML specification extends both the XML Schema 1.0 and Schematron specifications.
SOA	Service-Oriented Architecture
SSE	Server-Side Eclipse <i>but also</i> Structured Source Editing
STP	SOA Tools Platform
SWT	Standard Widget Toolkit
TBD	To Be Determined
Template document	An SML instance defining a common pattern that can be re-used and adapted in different domain models
Template editor	An editor that can create definition and template documents
UI	User Interface
UML	Unified Modeling Language
VM	Virtual Machine
WS	Web Services <i>but also</i> Windowing System
WSDL	Web Services Description Language
WTP	Web Tools Platform
XML	eXtensible Markup Language
XSDL	XML Structure Definition Language

## 7 BIBLIOGRAFÍA Y REFERENCIAS

No procede.