

ÍNDICE

1. HERRAMIENTAS DE DEPURACIÓN	2
2. TIPOS DE EJECUCIÓN	2
3. DEPURACIÓN CON ECLIPSE	2
3.1. PUNTOS DE RUPTURA.....	4
3.2. EXAMINADORES DE VARIABLES.	5
3.3. BOTONES DE DEPURACIÓN.	5

1. Herramientas de Depuración

Todo entorno de desarrollo, independientemente de la plataforma, así como del lenguaje de programación utilizado, suministra una serie de herramientas de depuración (conocidas como depurador), que nos permiten verificar el código generado, ayudándonos a realizar pruebas tanto estructurales como funcionales.

El depurador nos ayuda a analizar todo el programa mientras éste se ejecuta.

El depurador proporciona funciones como:

- Suspende la ejecución de un programa
- Examinar y establecer los valores de las variables
- Comprobar los valores devueltos por un determinado método, el resultado de una comparación lógica o relacional, etc.

2. Tipos de ejecución

Para depurar un programa, puede ser ejecutado de diferentes formas, de manera que en función del problema que queramos solucionar, nos resulte más sencillo un método u otro. Nos encontramos con los siguientes tipos de ejecución:

- **Paso a paso.** Algunas veces es necesario ejecutar un programa línea por línea, para buscar y corregir errores lógicos. El avance paso a paso a lo largo de una parte del programa puede ayudarnos a verificar que el código de un método se ejecute de forma correcta.
- **Paso a paso por procedimientos.** Nos permite introducir los parámetros que queremos a un método o función de nuestro programa, pero en vez de ejecutar instrucción por instrucción ese método, nos devuelve su resultado. Es útil, cuando hemos comprobado que un procedimiento funciona correctamente, y no nos interesa volver a depurarlo, sólo nos interesa el valor que devuelve.
- **Ejecución hasta una instrucción.** El depurador ejecuta el programa, y se detiene en la instrucción donde se encuentra establecido un punto de parada, a partir de ese punto, podemos hacer una depuración paso a paso o por procedimiento.
- **Ejecución de un programa hasta el final.** Ejecutamos las instrucciones de un programa hasta el final, sin detenernos en las instrucciones intermedias.

Los distintos modos de ejecución se van a ajustar a las necesidades de depuración que tengamos en cada momento.

3. Depuración con Eclipse

Para poder comprobar algunas de las funciones de depuración que nos ofrece Eclipse, vamos a crear un proyecto en Java con el siguiente código:

Clase Test

```
public class Test {  
  
    public static void main(String[] args) {  
  
        Contador contador = new Contador();  
        contador.contar();  
        System.out.println("Cuenta: " + contador.getResultado());  
  
    }  
  
}
```

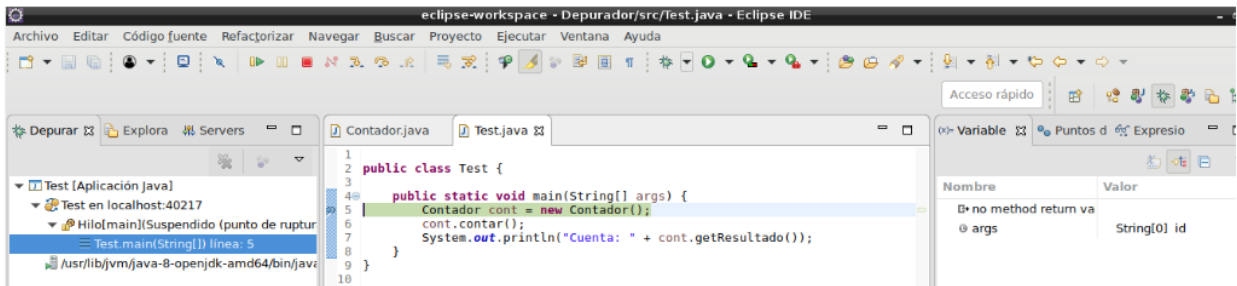
Clase Contador

```
public class Contador {  
    private int resultado = 0;  
  
    public int getResultado()  
    {  
        return resultado;  
    }  
  
    public void contar()  
    {  
        for(int i=0; i < 100 ; i++)  
        {  
            resultado = resultado + i + 1;  
        }  
    }  
}
```

Existen varias alternativas para lanzar la ejecución de un programa en modo debug. Una de ellas es pulsando con el botón derecho del ratón sobre la clase de inicio del proyecto (implementa el método main), y seleccionar en el menú contextual que aparece "Depurar como => Aplicación Java".



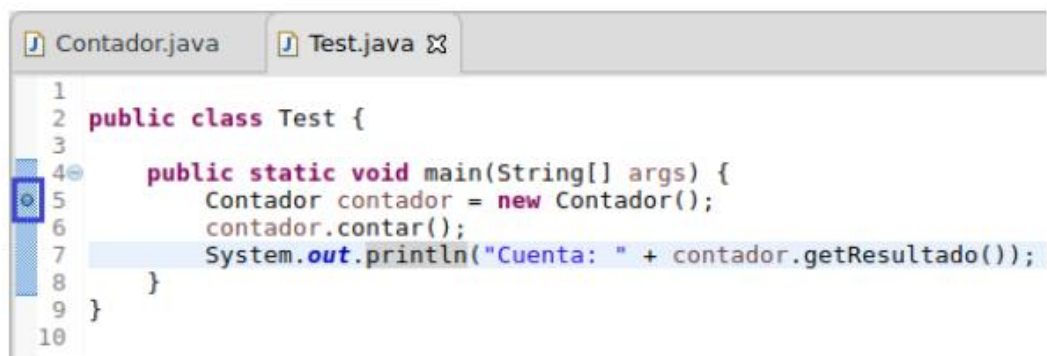
En **modo depuración**, **Eclipse** da la opción de trabajar con la perspectiva depurar, que ofrece una serie de vistas muy interesante para este tipo de ejecución, tales como: la vista de visualización y cambio de variables, la vista de puntos de parada establecidos o la pila de llamadas entre otras.



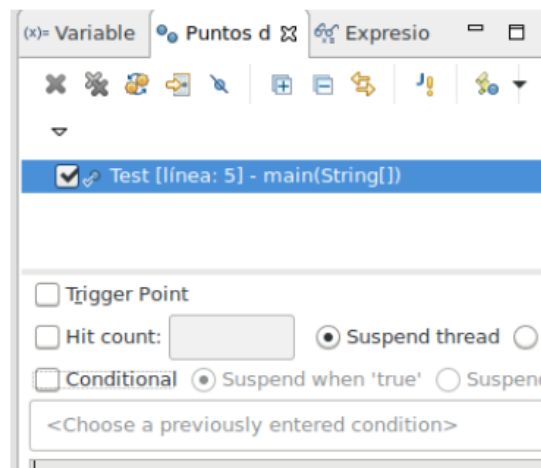
3.1. Puntos de ruptura.

Al solicitar una ejecución en modo debug, si no hemos establecido puntos de parada en el código, éste se ejecutará hasta el final del mismo modo que lo haría en una ejecución normal.

Para establecer un punto de parada, basta con hacer doble clic en el margen izquierdo de la línea de código donde se va a establecer.



El punto de parada es dado de alta y ya aparece en la vista de puntos de parada.



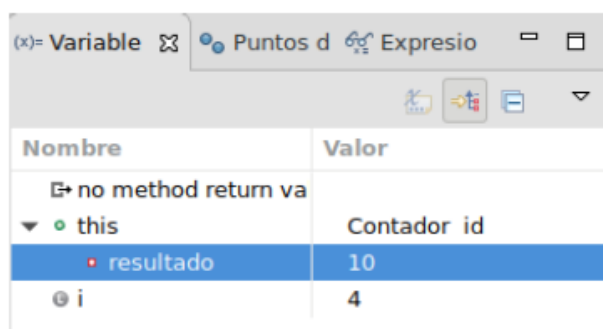
Eclipse permite definir puntos de parada condicionales para personalizar cuándo o por qué se para el programa. Si, por ejemplo, seleccionamos en un punto de parada la opción **Hit count** y determinamos un valor, el programa parará cuando haya pasado por este punto el número de veces indicado.

Es posible crear condiciones más elaboradas dependientes de otras variables disponibles en el contexto de la ejecución. El programa sólo parará cuando la condición definida se cumpla y la ejecución pase por esa línea de código.

En el momento que tenemos detenido el programa, se pueden realizar diferentes labores: por un lado, se pueden examinar las variables, y comprobar que los valores que tienen asignados son correctos, o se pueden iniciar una depuración paso a paso e ir comprobando el camino que toma el programa a partir del punto de ruptura. Una vez realiza la comprobación, podemos abortar el programa, o continuar la ejecución normal del mismo.

3.2. Examinadores de variables.

Durante el proceso de implementación y prueba de software, una de las maneras más comunes de comprobar que la aplicación funciona de manera adecuada, es chequear que las variables vayan tomando los valores adecuados en cada momento. **Eclipse** nos proporciona la **vista variables** donde podemos ir comprobando el valor que van tomando las variables activas en la zona de código donde está el programa parado.



Además, pulsando sobre el valor de cualquiera de estas variables es posible modificarlas. Esto nos permite evaluar nuevos escenarios de prueba con datos diferentes.

3.3. Botones de depuración.

Cuando estamos en el proceso de depuración de un programa con la perspectiva **Depurar**, Eclipse nos ofrece una serie de botones en su barra de herramientas que pasamos a describir a continuación.

	Ctrl + Alt + B	Desactiva temporalmente todos los puntos de parada del código.
	F8	Continúa la ejecución del programa. Se detendrá en el siguiente punto de parada.
		Pausa/detiene la ejecución del programa en el punto de código donde se encuentre al ser pulsado.
	Ctrl + F2	Finaliza la ejecución del programa.
		Función no considerada en este manual. Consultar manual de Eclipse.
	F5	Si el programa se encuentra detenido en la llamada a un método, al pulsar este botón la ejecución pasa a la primera línea del mismo.
	F6	Si el programa se encuentra detenido en la llamada a un método, al pulsar este botón la ejecución del método se hace por completo, sin depurar su implementación.
	F7	Avanza la ejecución del programa hasta que nos salimos del método actual y vamos hasta el sitio donde fue llamado.