

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

	<p>Tutorial desarrollado por: Roberto Canales Mora 2003-2004.</p> <p>Si te gusta lo que ves, puedes contratarme para impartir cursos presenciales en tu empresa o para ayudarte en proyectos (Madrid).</p> <p>Estamos creando las bases de la empresa en la que seguro te gustaría trabajar ... ayudanos a hacerla crecer ... presentándonos a tus jefes</p> <p>Contacta: rcanales@autentia.com.</p>	
---	---	---

Una vez escuche esta frase:

Cuando todo esta bajo control, es que no vamos suficientemente deprisa

La mayoría de las veces construimos estos tutoriales al mismo tiempo que recordamos como hacer cosas o probamos nuevas tecnologías, poniendo más celo en el contenido que en la forma (cosa que veo que perturba a muchos de los lectores por lo que os pido disculpas). Es cuestión de criterios aunque trataremos de mejorar.

Espero que seáis comprensivos y generosos ayudando a encontrar las erratas, faltas de ortografía, enlaces rotos u otros posibles errores en estas páginas. Escribidme siempre que encontréis algun error y pronto ya no habrá y os estaremos francamente agradecidos. Este es tu Web colabora libremente ...

[Roberto Canales](#)

Descargar este documento en formato PDF [junit.pdf](#)

Free Java Unit Tester

Automated Java unit
test/coverage Extends
JUnit/HttpUnit. IDE enabled

Code Coverage for Java

Method, Statement, Branch
Coverage. HTML & XML Reports.
IDE Plugins.

Diseño página web por 0€

Contratando un año de hosting
página web a medida igratis!

Busca-ServicioTecnico

Marcas electrodomesticos
Telefonos : Atencion, Reparacion.

Anuncios Goooooogle

JUnit: Pruebas unitarias en Java

Cuando realizados desarrollos profesionales, debemos tener en cuenta que no solo hay que ejecutar el proyecto... hay que ejecutarlo bien.

Un proyecto no esta terminado hasta que esta en producción, tiene su documentación y dispone de las garantías adecuadas de que va a funcionar en todos los escenarios posibles.

Además, normalmente, se desarrolla en un entorno, se empaqueta y posteriormente se entrega a un equipo de calidad para que lo pruebe y lo pase a otro (pre-producción y producción).

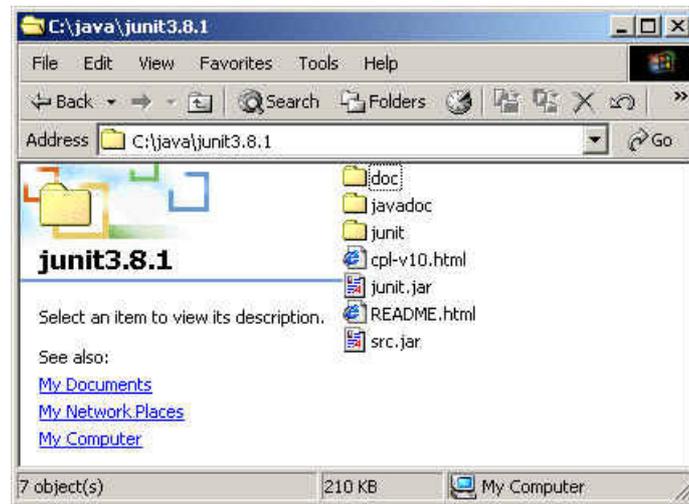
Todo esto hay que tenerlo en cuenta a la hora de dimensionar, diseñar y ejecutar un proyecto.

Del mismo modo, tambien debemos tener en cuenta que no lo podemos hacer de un modo distinto en cada proyecto. Hemos de utilizar mecanismos estandares y ampliables simple que se pueda.

Otro punto que cabe recordar (y que muchas veces se olvida) es que la tecnología es y debe ser un medio para que nuestros clientes consigan objetivos empresariales y el puritanismo tecnológico a nuestro cliente normalmente le dá igual tenemos que alcanzar un equilibrio entre calidad y resultados

Digo esto porque seguro que alguien descubre como hacer test unitarios y se le ocurre que a partir de ahora todo el sistema tenga test unitarios (economicamente no es muy viable tenemos que usar el sentido común)

Lo primero que vamos a hacer es decargarnoslo, y descomprimirlo.

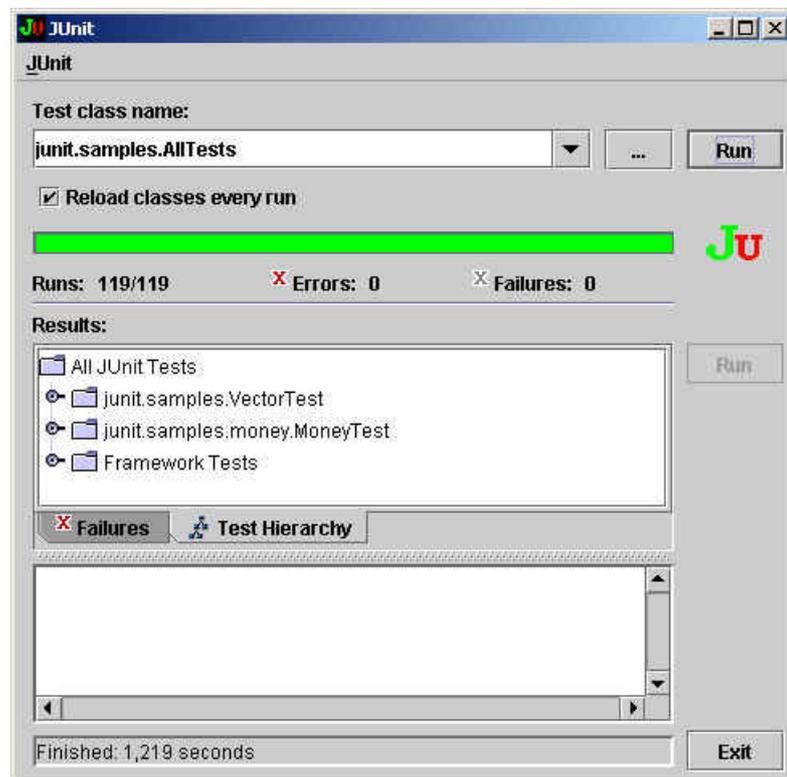


Luego tenemos que asegurarnos de poner en la variable de entorno classpath **junit.jar**

Ahora ejecutamos el comando de prueba

java junit.swingui.TestRunner junit.samples.AllTests

Y comprobamos el resultado:



Si no os a funcionado seguro que es un problema de classpath

Bueno, ya esta configurado el entorno ahora vamos a centrarnos en los principios de los Test Unitarios y como aplicarlo a nuestros programas.

Hacemos un programa mínimo para que funcionen nuestro propios tests y vemos los requerimientos básicos ...

```
import junit.framework.*;

public class pruebas extends TestCase
{
    pruebas(String tituloTest)
    {
        super(tituloTest);
    }

    protected void runTest()
    {
        int i = 0;
    }
}
```

```

    this.assertTrue("Valor Igual a 3",i==3);
}

public static Test suite()
{
    TestSuite suite= new TestSuite("Pruebas de clase pruebas");
    suite.addTest(new pruebas("Clase principal"));
    return suite;
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    // nuestro programa hace lo que deseemos
}
}

```

Para que una clase sea invocable desde JUnit, tenemos que hacer de derive de **junit.framework.TestCase**

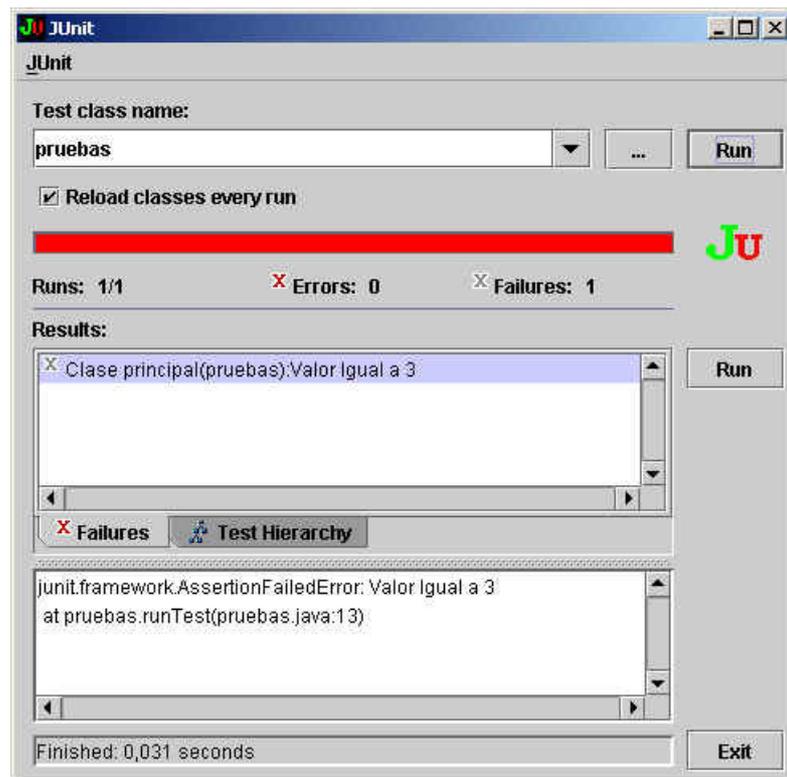
Nuestro test, debe redefinir el método **protected void runTest()**

Dentro de este método ... haremos llamadas a los métodos de verificación con **assertTrue("Valor Igual a 3",i==3);**

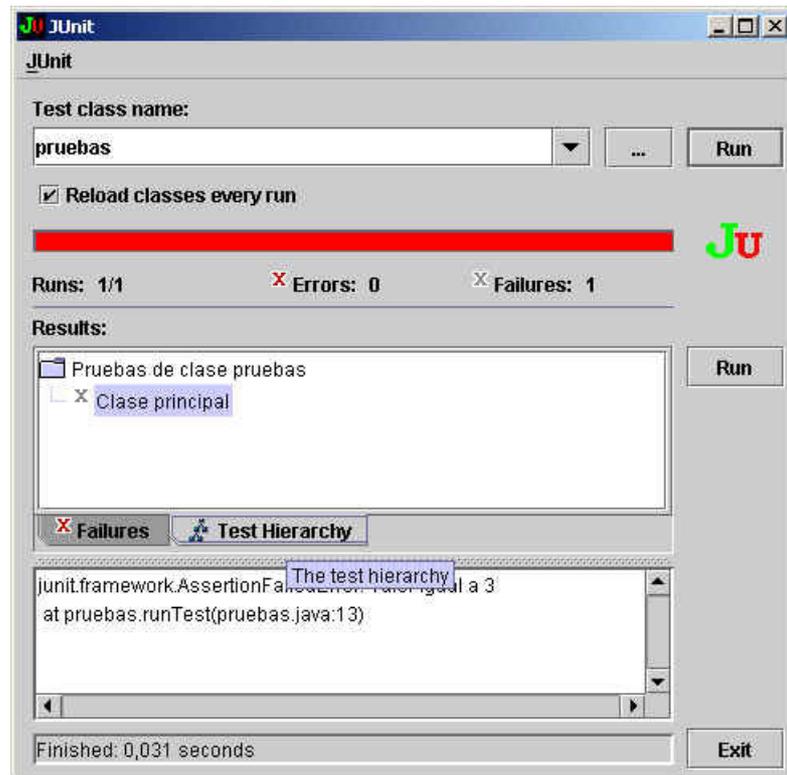
Para que podamos invocar desde una herramienta gráfica los test, debemos redefinir el método **public static Test suite()**

Y dentro de este método, añadiremos todos los test que deseemos **addTest(new pruebas("Clase principal"));**

Bueno ... compilamos y vemos que pasa ...



Vemos la otra lengüeta ...



Bueno ... seguro que el ejemplo no os a parecido demasiado claro ... asi que vamos a hacer unos pocos más ...

Supongamos que una de nuestras clases se encarga de leer los parámetros de un fichero ... y queremos asegurarnos que cuando pasemos entre entornos ese fichero esté

```
import junit.framework.*;
import java.io.*;

public class leerPropiedades extends TestCase
{
    String nombreFichero="parametros.ini";

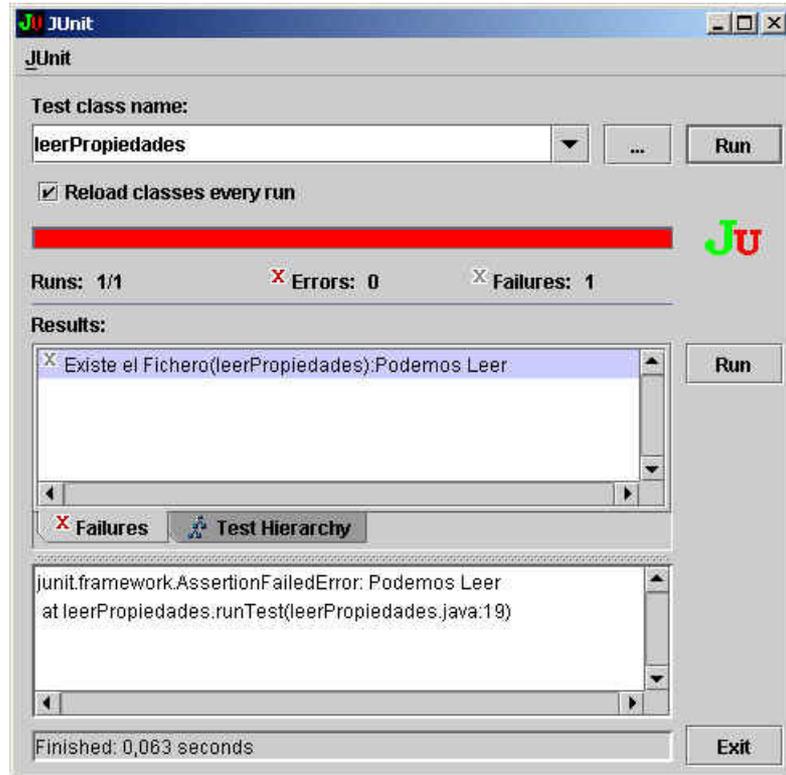
    public leerPropiedades(String nombreTest)
    {
        super(nombreTest);
    }

    protected void runTest()
    {
        try
        {
            File fichero = new File(nombreFichero);
            this.assertTrue("Podemos Leer",fichero.canRead());
        }
        catch(Exception e)
        {
            this.fail("Fallo al identificar Fichero " + nombreFichero);
        }
    }

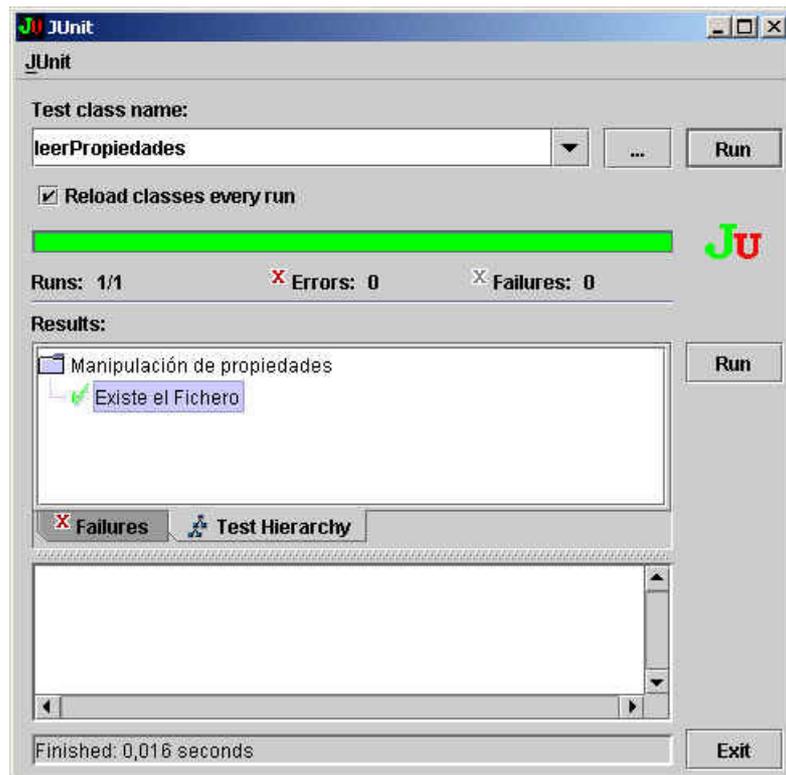
    public static Test suite()
    {
        TestSuite suite= new TestSuite("Manipulación de propiedades");
        suite.addTest(new leerPropiedades("Existe el Fichero"));
        return suite;
    }

    public static void main(String[] args) {
    }
}
```

El resultado si el fichero no existe o no tenemos privilegios sería



Si el fichero existe ... la salida sería



Lo vamos a dejar de momento aquí .. en próximas entregas expondremos técnicas más avanzadas

[Sobre el Autor ..](#)

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con



Somos expertos en:
J2EE, C++ , OOP, UML, Vignette, Creatividad ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	<input type="text"/>
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto

[Rendimiento de aplicaciones Web](#)

[Introducción a ANT](#)

[Desarrollo Gráfico Scripts Ant](#)

[Introducción a log4j](#)

Descripción

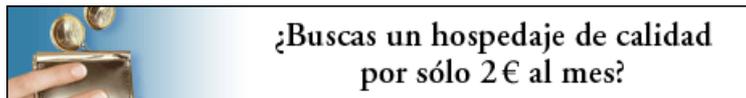
En este tutorial veremos una introducción al funcionamiento de la Suite e-Test de Empirix.

En el mundo Java, la compilación, verificación e instalación de aplicaciones se ha normalizado con este potente paquete llamado ANT.

Os mostramos como crear y ejecutar scripts Ant (para automatizar tareas en el mundo Java) con las herramientas gratuitas Antelope y NetBeans

En un desarrollo Java es vital normalizar los logs para posteriormente poder depurar el funcionamiento de nuestra aplicación. Os mostramos como usar Log4J.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Opimizado 800X600