

¿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)

» Estás en: [Inicio](#) [Tutoriales](#) [Pruebas automáticas con FTP](#)**Francisco Javier Martínez Páez**

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Telecomunicaciones

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)
 Tu foto impresa en una
 original pieza con carcasa
 personalizable

 por sólo
9,95€


Fecha de publicación del tutorial: 2015-02-05

Tutorial visitado 6 veces [Descargar en PDF](#)

Pruebas automáticas con FTP

0. Índice de contenidos.

- 1. Basado en hechos reales
- 2. Ejemplo
- 3. Conclusiones

1. Basado en hechos reales:

Esta frase, queridos amigos, nos ha perseguido durante muchos años a los profesionales del desarrollo de software:

No toques ese código no se vaya a romper, además, eso lo hizo Federico que ya no está.

Cambiado Federico (que se ha ido al cielo de los programadores, como decía un antiguo jefe mío cuando un compañero cambiaba de proyecto o se marchaba a otra empresa) por el nombre que corresponda y seguro que esa frase estará oculta por ahí en algún rincón de vuestra memoria.

Y es que no hay nada que asuste más a un desarrollador de Software que romper algo que ya funcionaba, por lo que en lugar de refactorizar algo cuando es necesario, pues solemos optar por alternativas menos adecuadas basadas en patrones muy conocidos como: *CPCI* (*Copy Paste and change it*) combinándolo con *LTIAGFUT* (*Look, there is a goat floating up there*).

Entonces, uno como el que os escribe, que ya va para viejuno y ha provocado por tanto muchos desaguisados, ha aprendido con sangre, que las pruebas automáticas no deberían ser una opción. Echando una manita en un proyecto, la primera historia que cayó en mi tejado fue ampliar la funcionalidad de una clase que hacía las veces de *adaptador (wrapper)* sobre una librería FTP. Bueno, pues eso, que uno empieza a mirar y descubre que el código pide una refactorización para poder ampliar la funcionalidad de manera adecuada y para corregir alguna que otra cosita. Pero no pasa nada porque tengo tests... o no ... pues eso que no... que no tengo, no había tests de esta parte... y ¿Por qué no hay ?, me pregunté... supongo que ya más o menos os imagináis la respuesta: básicamente porque es un wrapper sobre FTP y como vamos a montar los tests si necesito un servidor FTP, etc... y entonces, vuelvo a oír esa frase tan vieja como la siesta surgiendo desde mi propio inconsciente: Paco, no toques ese código no se vaya a romper. Pues esto es como dejar de fumar: hay que superar ese momentito en el que consideras que la solución a todos tus problemas es echarte un cigarrito. Supéralo y mótate los tests adecuados. Si miras un poco por internet resulta que hay gran cantidad de frameworks que facilitan el trabajo. 5 minutos me llevó encontrar esto: *MockFtpServer*. Esta librería nos permite simular un servidor FTP con lo que poder realizar nuestros tests automáticos.

2. Ejemplo:

La documentación del site es muy buena y la librería es muy intuitiva por lo que no me voy a extender demasiado: Para incluir la dependencia si usais maven:

```

1 <dependency>
2   <groupId>org.mockftpserver</groupId>
3   <artifactId>MockFtpServer</artifactId>
4   <version>2.4</version>
5   <scope>test</scope>
6 </dependency>

```

Para levantar un servidor FTP para vuestras pruebas si usais JUnit:

```

1 @BeforeClass
2 public static void startFTPServer() {
3     FakeFtpServer fakeFtpServer = new FakeFtpServer();
4     fakeFtpServer.setServerControlPort(FTP_PORT);
5     fakeFtpServer.addUserAccount(new UserAccount("usuario", "password", "c:\\data"));
6
7     FileSystem fileSystem = new WindowsFakeFileSystem();
8     fileSystem.add(new DirectoryEntry("c:\\data\\pruebas"));
9

```

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» 2015: ¡Volvemos a la oficina!

» [Curso JBoss de Red Hat](#)

» Si eres el responsable o líder técnico, considérate desafortunado. No puedes culpar a nadie por ser gris

» Portales, gestores de contenidos documentales y desarrollos a medida

» [Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer](#)

[Histórico de noticias](#)

Últimos Tutoriales

» Oracle - Importación de Datos con Data Pump Import [impdp]

» Oracle - Exportación de Datos con Data Pump Export [expdp]

» Flexibilidad laboral: mejoras para empleados y empresarios

» Instalación de un entorno Hadoop con Ambari en AWS

» Política de reintentos con Spring Retry

```

10 fakeFtpServer.setFileSystem(fileSystem);
11 fakeFtpServer.start();
12 }

```

3. Conclusiones:

Cumplir los principios F.I.R.S.T (Fast.Isolated.Repeatable.Self-validating.Timely) a veces es complicado; los cuatro primeros son duros de conseguir cuando tenemos que interactuar con herramientas externas (servidores de correo, servidores FTP, bases de datos, sistemas de autorización...), es decir, cuando nos salimos del límite del sistema. Sin embargo existen gran cantidad de herramientas que nos facilitan esa labor haciendo de Fake y que proporcionan un mecanismo para evitar esa vieja excusa de "esto no se puede probar".

Últimos Tutoriales del Autor

- » [Introducción a PhoneGap](#)
- » [JBossESB y JBDS](#)
- » [Apache Hadoop - HDFS](#)
- » [Apache Hadoop-MapReduce](#)
- » [Apache Camel, primeros pasos](#)

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

Share |

[+1](#) [0](#)

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA

Impulsores

Comunidad

¿Ayuda?

sin clicks

0 personas han traído clicks a esta página

+ + + + + + + +

powered by [karmacracv](#)

Copyright 2003-2015 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

