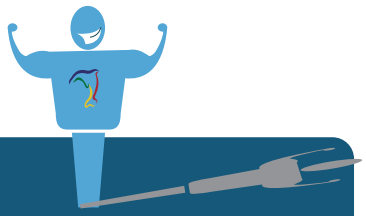
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

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

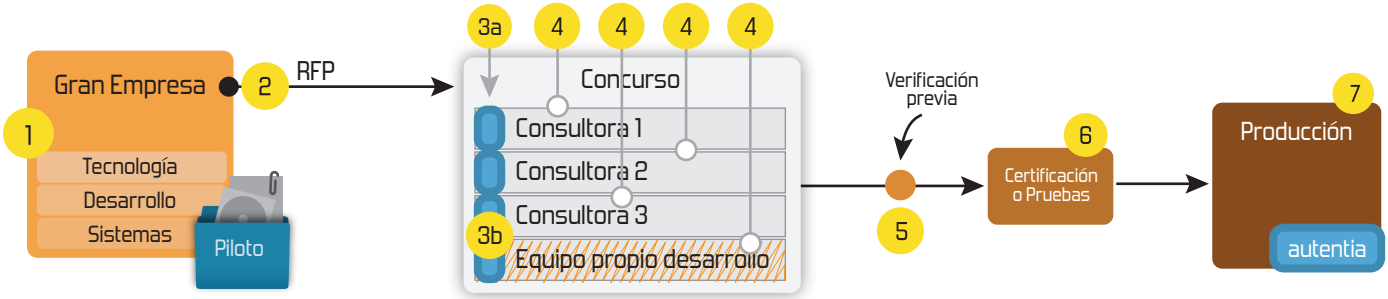
1. Desarrollo de componentes y proyectos a medida



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
- 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
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Hosting patrocinado por enredados autentia

Ultimas Noticias

» Creamos el grupo amigos de AdictosAlTrabajo en Facebook
» ¡¡¡¡ESPAAAAAAAAAAAAAAAÑA!!!!
» VIII Charla Autentia - Alfresco Community Edition - Vídeos y Material
» IX Charla Autentia - Android
» Autentia Cycling Day
» La conferencia Agile-Spain2010 ya está aquí ¿Te la vas a perder?
» VIII Charla Autentia - Alfresco Community

+Noticias Destacadas

» Creamos el grupo amigos de AdictosAlTrabajo en Facebook
» VIII Charla Autentia - Alfresco Community Edition - Vídeos y Material
» IX Charla Autentia - Android
» La conferencia Agile-Spain2010 ya está aquí ¿Te la vas a perder?

+Comentarios Cómic

+Enlaces

Catálogo de servicios Autentia

Tríptico (6,3 MB)

Cómic (3,1 MB)

Tutorial desarrollado por

Enrique Viñé Lerma

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática por la Universidad Politécnica de Madrid.

Puedes encontrarme en Autentia

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (3,1 MB)

AdictosAlTrabajo.com es el Web de difusión de conocimiento de Autentia.

autentia real business solutions

Catálogo de cursos

Descargar este documento en formato PDF: introduccionJodaTime.pdf

Fecha de creación del tutorial: 2010-06-30

Acceso de usuarios registrados:

E-mail:
Contraseña:
Entrar

Deseo registrarme

He olvidado mis datos de acceso

Registra tu empresa:

Descubre las ventajas de registrar tu empresa en AdictosAlTrabajo...

Registrar mi empresa

Listado de empresas ya registradas

Introducción a Joda Time

Índice de contenidos

1. Introducción
2. Configurar nuestra aplicación para utilizar Joda Time
3. Conceptos clave
4. Trabajar con fechas
    1. Construcción de fechas
    2. Conversión entre fechas de Joda Time y fechas del JDK
    3. Campos de fechas y propiedades
    4. Formateo de fechas
5. Trabajar con intervalos y períodos
    1. Intervalos

Google

Web
www.adictosaltrabajo
Buscar

Últimos tutoriales

2010-06-30
Introducción a Primefaces

2010-06-30

2. Períodos
3. Formateo de períodos

6. Conclusiones
7. Enlaces útiles

1. Introducción

Joda Time es un API Java que permite trabajar con fechas de una forma más sencilla, potente y eficiente que el API estándar de fechas de Java. Joda Time incluye algunos conceptos como intervalos, duraciones y períodos, que están bastante mal soportados en el API estándar.

Joda Time es bastante "viejo", lleva desarrollándose al menos desde el 2002 y la última actualización es del año pasado, pero muchos desarrolladores lo desconocen y a veces resulta muy útil para realizar ciertas operaciones complejas con fechas (complejas si no utilizamos este API, claro).

Las características principales de Joda Time son:

- Facilidad de uso, con métodos de acceso directos a los campos de una fecha.
- Facilidad de extensión. Extender la clase Calendar del JDK puede resultar muy complicado si necesitásemos utilizar un sistema de calendario personalizado. Joda-Time soporta múltiples calendarios (8 actualmente) por medio de un sistema extensible basado en la clase Chronology. A pesar de todo, la mayoría de los mortales probablemente nunca necesitemos extender ninguno de los dos sitemas en nuestro trabajo cotidiano.
- Funcionalidades avanzadas para el cálculo y formateo de fechas, que en muchos casos son difíciles de imitar con el API estándar de Java.
- Es muy fácil la conversión de fechas entre Joda Time y el JDK.
- Una documentación bastante buena del uso general del API y un javadoc detallado que nos permiten utilizar funcionalidades avanzadas.
- Integración con otros sistemas como Hibernate o los tags de JSP.
- Es Open Source, bajo la licencia Apache License Version 2.0.

2. Configurar nuestra aplicación para utilizar Joda Time

Para utilizar el API de Joda Time en nuestra aplicación sólamente tendremos que descargarnos el .jar correspondiente (que ocupa unos 2 MB) y añadirlo a nuestro classpath. O, mejor aún, si estamos utilizando Maven para gestionar nuestro proyecto, tendremos que añadir la dependencia siguiente a nuestro "pom.xml":

```
<dependency>
    <groupId>joda-time</groupId>
    <artifactId>joda-time</artifactId>
    <version>1.6</version>
</dependency>
```

3. Conceptos clave

Concepto	Descripción	Interfaz / Clase Abstracta	Implementaciones principales
Instante	Representa un instante concreto en la línea temporal (una fecha), con precisión de milisegundos. Un instante se representa internamente como el número de milisegundos transcurridos desde "1970-01-01T00:00Z"	ReadableInstant	DateTime MutableDateTime Instant
Intervalo	Representa un intervalo de tiempo entre dos instantes que utilizan la misma cronología y zona horaria. Los intervalos son abiertos por la derecha, es decir, incluyen su instante inicial, pero no el final.	ReadableInterval	Interval MutableInterval
Duración	Representa un período de tiempo en milisegundos. No dependen de una cronología ni zona horaria.	ReadableDuration	Duration
Período	Representa un período de tiempo en términos de años, meses, semanas, días, horas, minutos, segundos y milisegundos. Se diferencia de una duración en que no es exacto en términos de milisegundos, por ejemplo, sumar un período de "1 mes" al día "16 de Febrero" devolverá "16 de Marzo", que no es lo mismo que sumar 30 días o $2.592x10^6$ milisegundos. Hay períodos de un solo campo (Years, Days...) o de varios cualesquiera (Period).	ReadablePeriod	Period MutablePeriod Years Months Days Hours Minutes

Introducción a Joda Time

2010-06-29
mapeo Vistas Hibernate

2010-06-29
Eclipse Helios, la nueva versión 3.6 de Eclipse

2010-06-28
Mac Automator y como renombrar múltiples ficheros

2010-06-21
Prototipado rápido de pantallas (mocks) con Google Drawings

2010-06-17
IAQ (Interesting Asked Questions), SPI ¿qué es, hay que usarlo, o no, cuándo?

2010-06-14
Instalar PostgreSQL en Mac OS X Snow Leopard

2010-06-14
Integrando tus redes sociales con HootSuite

2009-02-26
Edición de vídeo para publicar en YouTube (II)

2010-06-14
Primeros pasos en comunidad

2010-06-11
Cómo analizar con Sonar un proyecto sin maven

2010-06-11
DBUnit y aplicaciones JDBC

2010-06-11
Edición de vídeo para publicar en YouTube (I)

2010-06-10
Generación de Informes con JasperReports en PHP

2010-06-07
Introducción a BPMN

2010-06-07
Instalación Intalio Server

Parcial	Representa una fecha de forma incompleta y sin ninguna zona horaria. Por ejemplo, un parcial podría representar el día "7 de Julio" (sin especificar el año), las "12:00" o "Enero de 1994".	ReadablePartial	LocalDate LocalTime LocalDateTime Partial
Cronología	Representa un sistema de calendario/medición del tiempo, como la clase Calendar del JDK. Para la mayoría de las aplicaciones, no necesitaremos utilizar estas clases, y nos valdrá con utilizar la implementación por defecto: ISOChronology.	Chronology	ISOChronology GregorianChronology JulianChronology
Zona horaria	Una zona horaria se aplica con el patrón Decorator sobre una cronología.	DateTimeZone (clase abstracta)	No se usan directamente, sino a través de métodos factoría de DateTimeZone

4. Trabajar con fechas

Para todos los ejemplos que siguen a partir de ahora, supondremos que nuestro Locale es "es-ES".

4.1. Construcción de fechas

```
// Fecha y hora actuales
DateTime date = new DateTime();

// Especificar una fecha en formato ISO
date = new DateTime("2010-06-25"); // horas, minutos, segundos y milisegundos a 0
date = new DateTime("2010-06-25T13:30:00"); // milisegundos a 0

// Especificar indicando año, mes, día, horas, minutos, segundos y milisegundos
date = new DateTime(2010, 6, 25, 13, 30, 0, 0);

// Especificar zona horaria
date = new DateTime(DateTimeZone.forID("Europe/London"));

// Especificar cronología
date = new DateTime(BuddhistChronology.getInstance());
```

4.2. Conversión entre fechas de Joda Time y fechas del JDK

```
// de Joda a JDK
DateTime dt = new DateTime();
java.util.Date     date     = dt.toDate();
java.util.Calendar calendar = dt.toCalendar(Locale.US);

// de JDK a Joda
dt = new DateTime(date);
dt = new DateTime(calendar);
```

4.3. Campos de fechas y propiedades

Joda Time separa la representación de un instante de tiempo (DateTime) del cálculo de los campos de calendario. Es decir, nosotros tendremos una fecha (representada por un número de milisegundos transcurridos desde "1970-01-01T00:00Z") y, cuando queramos obtener, por ejemplo, el día de la semana, se realizará el cálculo en función de la cronología y zona horaria para obtener el valor correspondiente.

```
DateTime dt = new DateTime();

// Obtener el día de la semana (lunes, martes...)
int dayOfWeek = dt.getDayOfWeek();

// Obtener el número de minutos transcurridos en el día
int minuteOfDay = dt.getMinuteOfDay();

// Obtener la semana del año
int weekOfYear = dt.getWeekOfWeekyear();
```

La clase DateTimeConstants contiene una serie de constantes enteras con los valores de los días de la semana, los meses, etc.

Además de obtener directamente los valores para los diferentes campos de una fecha, la clase DateTime dispone de métodos para recuperar estos campos como una propiedad (de tipo DateTime.Property). Estas propiedades permiten realizar operaciones adicionales sobre los campos de la fecha. En la documentación oficial de Joda Time se encuentra la lista completa de los campos disponibles para una fecha.

```
DateTime dt = new DateTime();

// Obtener la propiedad correspondiente al día de la semana
Property dayOfWeek = dt.dayOfWeek();

// Obtener el día de la semana como una cadena localizada según la zona horaria ("lunes", "martes"...)
String sDayOfWeek = dayOfWeek.getAsText();

// Obtener la fecha correspondiente al lunes de la semana actual (las semanas comienzan en lunes)
DateTime lunes = dayOfWeek.setCopy(DateTimeConstants.MONDAY);
```


2010-05-31
JSF 2 Ajax y Bean Validation

2010-05-27
Instalación Intalio Designer

2010-05-26
Generar facturas electrónicas de manera gratuita

2010-05-25
Atajos de teclado más utilizados para Eclipse

2010-05-24
Botones para añadir eventos directamente a calendario (Google Calendar y Outlook)

2010-05-24
Procesar un fichero EDI con Talend

2010-05-17
Añadir botones para social bookmarking y sharing service.

2010-05-17
Intercomunicación entre portlets.

2010-05-17
Actualización de bugzilla a la versión 3.6.

2010-05-11
eCobertura y EclEMMA: Plugins para la medición de cobertura del código.

2010-05-11
VirtualBox. Configuración de la conexión de red.

2010-05-11
Múltiples datasources en JasperReports

2010-05-10
Utilización de arquetipos y plugins de LifeRay para Maven.

2010-05-07
Instalar un componente en Talend Open Studio (TOS)

2010-05-06
Talend Open Studio

Casi todas las clases de Joda Time son "inmutables". Por ejemplo, cuando modifiquemos los campos de una fecha, realmente estaremos obteniendo una nueva fecha con el valor para ese campo modificado. En el ejemplo anterior, el método setCopy obtiene una "copia" de la fecha con el valor modificado, pero no modifica para nada la fecha original.

No siempre será necesario utilizar la clase DateTime.Property para realizar operaciones sobre una fecha. Las operaciones más frecuentes disponen de métodos directos en la clase DateTime:

```
// Sumar dos meses a una fecha
DateTime dosMesesDespues = dt.plusMonths(2);

// Obtener la fecha correspondiente al lunes de la semana actual
DateTime lunes = dt.withDayOfWeek(DateTimeConstants.MONDAY);
```

4.4. Formateo de fechas

La clase DateTimeFormatter permite convertir cadenas en fechas y viceversa, utilizando una representación específica de las mismas.

```
// Crear un formatter con una representación específica
DateTimeFormatter fmt = DateTimeFormat.forPattern("dd-MMMM-yyyy");

// Obtener un formatter localizado
DateTimeFormatter americanFmt = fmt.withLocale(Locale.US);

// Obtener una fecha a partir de su representación
DateTime dt = fmt.parseDateTime("25-junio-2010");

// Escribir una fecha con el formato especificado
System.out.println(fmt.print(dt)); // escribe "25-junio-2010"
System.out.println(americanFmt.print(dt)); // escribe "25-June-2010"

// Métodos de acceso directo
System.out.println(dt.toString("dd-MMMM-yyyy")); // escribe "25-junio-2010"
System.out.println(dt.toString("dd-MMMM-yyyy", Locale.US)); // escribe "25-June-2010"
```

También es posible crear formatos utilizando la clase DateTimeFormatterBuilder, aunque para la mayoría de los casos, no será necesario. Esta clase funciona de forma similar a PeriodFormatterBuilder, que se utiliza para crear "formateadores" para los períodos, de la cual se muestra un ejemplo al final del siguiente apartado.

5. Trabajar con intervalos y períodos

5.1. Intervalos

```
// Crear un intervalo entre el 1 de Enero y la fecha actual
DateTime inicio = new DateTime(2010, 1, 1, 0, 0, 0, 0);
DateTime fin = new DateTime();
Interval interval = new Interval(inicio, fin);

// Recuperar el inicio y fin del intervalo
DateTime i = interval.getStart();
DateTime f = interval.getEnd();

// Comprobar si una fecha determinada está dentro del intervalo
boolean ok = interval.contains(new DateTime("2010-04-13"));

// Conversión a duración o período
Duration duration = interval.toDuration();
Period period = interval.toPeriod();
```

5.2. Períodos

```
// ----- Períodos de un único campo -----
// Años desde el 16/03/1976
Years years34 = Years.yearsBetween(new DateTime("1976-03-16"), new DateTime());

// Obtener el número de meses desde un intervalo
Interval interval = new Interval(new DateTime("2010-01-01"), new DateTime("2010-12-01"));
Months months11 = Months.monthsIn(interval);

// Constante predefinida
Days days1 = Days.ONE;

// Especificar un número de días
Days days15 = Days.days(15);

// ----- Períodos de varios campos -----
// 1 año, 6 meses, 2 semanas, 3 días y 12 horas
Period period = new Period(1, 6, 2, 3, 12, 0, 0, 0);

// Años, meses, semanas, días, horas, minutos, segundos y milisegundos desde el 01/01/2000
Period period2 = new Period(new DateTime("2000-01-01"), new DateTime());

// Años, meses y días desde el 01/01/2000
Period period3 = new Period(new DateTime("2000-01-01"), new DateTime(), PeriodType.yearMonthDay());
```

(TOS) 4.0

2010-05-05
Gestión de los Requisitos (y II) : los casos de uso

2010-05-05
Gestión de los requisitos

2010-05-04
Declaración de IVA trimestral en la AEAT por Internet

2010-05-04
Certificados en Firefox (FNMT y AEAT)

2010-04-26
JCaptcha - Generación de Captchas en Java

2010-04-23
Instalar Puente PHP-Java en Tomcat

2010-04-22
AppWidget Android: Ejemplo usando BroadcastReceiver y Localización

2010-04-20
Facelets en JSF 2: sistema de plantillas y componentes por composición.

2010-04-19
DbVisualizer free version.

2010-04-09
Session TimeOut en RichFaces, con el soporte de Jboss Seam.

2010-04-08
Jetspeed-2 de Apache Software Foundation

2010-04-07
Primeros pasos con Balsamiq Mockups

2010-03-18
Revisando los ejemplos de Cocos2d para IPhone.

2010-03-16
Organización de eventos con StageHQ

2010-03-15

Como puede verse en los últimos ejemplos, cuando creamos un período sin especificar su tipo, se considera el tipo estándar, que incluye los valores para todos los campos. Si queremos un período que no incluya algunos campos (como las semanas, horas, etc.) debemos especificarlo de manera explícita.

5.3. Formateo de períodos

Para terminar vamos a ver un ejemplo de cómo utilizar un PeriodFormatter para convertir una cadena del tipo "HH:mm:ss" a un período que contendrá las horas, minutos y segundos correspondientes, y viceversa. Para crear el Formatter utilizaremos los métodos de la factoría PeriodFormatterBuilder, que permite construir formatos bastante complejos.

```
// Crear el PeriodFormatter
PeriodFormatter durationFormatter = new PeriodFormatterBuilder()
        .minimumPrintedDigits(2) // Número de dígitos que se mostrarán en la salida para los campos siguientes
        .printZeroAlways() // Indica que deben mostrarse los campos siguientes aunque su valor sea 0
        .appendHours()
        .appendSeparator(":")
        .appendMinutes()
        .appendSeparator(":")
        .appendSeconds()
        .toFormatter();

// Obtener un período a partir de un String
Period period = durationFormatter.parsePeriod("07:12:38");

// Obtener el String que representa el período
String sPeriod = durationFormatter.print(period);
```

En la construcción del PeriodFormatter, se han utilizado las funciones "minimunPrintedDigits" y "printZeroAlways" al principio. Estos valores se utilizarán para cada campoque se añada después, aunque podrían modificarse para un campo concreto. Por ejemplo, si quisiéramos que para los minutos sólo se imprimiese un dígito y no apareciesen los segundos si su valor fuese cero, podríamos modificar el código anterior como sigue:

```
PeriodFormatter durationFormatter = new PeriodFormatterBuilder()
        .minimumPrintedDigits(2) // Número de dígitos que se mostrarán en la salida para los campos siguientes
        .printZeroAlways() // Indica que deben mostrarse los campos siguientes aunque su valor sea 0
        .appendHours()
        .appendSeparator(":")
        .minimumPrintedDigits(1) // Para los minutos queremos mostrar sólamente 1 dígito
        .appendMinutes()
        .appendSeparator(":") // Si los segundos son 0, tampoco se mostrará este separador
        .minimumPrintedDigits(2) // Pero para los segundos queremos mostrar de nuevo 2
        .printZeroNever() // Si los segundos son 0, no se mostrarán
        .appendSeconds()
        .toFormatter();
```

6. Conclusiones

A la vista de lo expuesto en el tutorial, se pueden extraer las siguientes conclusiones sobre la manipulación de fechas en Java:

- Si nuestra aplicación va a utilizar sólamente cálculos comunes con fechas, no hay demasiada diferencia entre utilizar la JDK o Joda Time, por lo cual es más que recomendable utilizar el API estándar.
- En el caso de que tengamos que realizar cálculos o conversiones más complejas, Joda Time nos ofrece un API muy potente capaz de cubrir probablemente todas nuestras necesidades.
- La mayoría de los frameworks utilizados en Java (JSF, Spring...) utilizan las clases del JDK para trabajar con fechas. Si optamos por utilizar Joda Time en nuestras aplicaciones, no deberíamos tener muchos problemas, ya que la conversión entre ambos tipos resulta muy sencilla.
- Joda Time puede hacer que nuestro código sea más sencillo, legible y fácil de entender.
- Si necesitamos trabajar con conceptos como duraciones o períodos, entonces es casi obligatorio el uso de Joda Time, ya que éstos conceptos no están ni mucho menos tan bien soportados en el JDK.

Y eso es todo. Espero que este tutorial os resulte útil la próxima vez que tengáis que manipular fechas o duraciones en vuestras aplicaciones.

Un saludo.

7. Enlaces útiles

- Página oficial de Joda Time
- Estándar de fechas ISO 8601
- tz database

Share |

Retrasar la carga de Javascript con jQuery.getScript().

2010-03-15
Optimización de páginas web con Page Speed.

2010-03-09
JSF 2 ya está aquí !!! The JSF Return, ahora más sencillo que nunca !!!

2010-03-08
Instalación de tus programas en tu IPhone.

Últimas ofertas de empleo

2010-04-28
Comercial - Compras - CORDOBA.

2010-04-25
Otras Simpatías - MADRID.

2010-04-25
Atención a cliente - Call Center - MADRID.

2010-04-21
Comercial - Ventas - MADRID.

2009-06-25
Atención a cliente - Call Center - BARCELONA.

Anuncios Google
Java
Free Java Tutorial
Java Chat API
SQL String

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ¡vota!

Muy malo Malo Regular Bueno Muy bueno

Votar

(Sólo para usuarios registrados)

» Registrate y accede a esta y otras ventajas «

Anímate y coméntanos lo que pienses sobre este tutorial

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» Registrate y accede a esta y otras ventajas «

Autor **Mensaje de usuario registrado**

- Puedes inscribirte en nuestro servicio de notificaciones haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.
- Añadir a favoritos Technorati. ADD THIS BLOG TO MY Technorati FAVORITES

SOME RIGHTS RESERVED Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido (Ver todos los tutoriales). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

soluciones reales para su negocio

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.