

# ¿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)

 Powered by 	Hosting Patrocinado por <b>enREDados.com</b> 
---	--

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

## Lanzamiento TNTConcept

[Autentia](#) da un paso más en su evolución: Lanzamiento de software propio. Ponemos a vuestra disposición el software que hemos construido para nuestra gestión interna, llamado TNTConcept (auTeNTia).

Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Hibernate, Maven, Subversion, etc.) y disponible en licencia GPL, seguro que a muchos profesionales independientes y PYMES os ayudará a organizar mejor vuestra operativa.

**Las cosas grandes empiezan siendo algo pequeño** ..... Saber más en: <http://tntconcept.sourceforge.net/>

<p><b>Tutorial desarrollado por:</b>  <b>Jose Carlos López (<a href="#">Autentia</a>) es consultor tecnológico de desarrollo de proyectos informáticos.</b>  <b>Contacta en <a href="mailto:jclopez@autentia.com">jclopez@autentia.com</a></b></p>	<p><a href="http://www.adictosaltrabajo.com">www.adictosaltrabajo.com</a> es el Web de difusión de conocimiento de <a href="http://www.autentia.com">www.autentia.com</a></p>  <p>real business solutions</p> <p style="text-align: center;"><a href="#">Catálogo de cursos</a></p>
--	--

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

[Firma en nuestro libro de Visitas](#) <-----> [Asociarme al grupo AdictosAlTrabajo en eConozco](#)

### Master Java Certificado

Temario Actualizado-UML-JSF-AJAX Trabajo Garantizado-Bolsa de Empleo  
[www.exes.es](http://www.exes.es)

### Learn about AJAX and Java

Free Webinar about AJAX & JSP, JSF, Struts. Architecture & live coding.  
[www.backbase.com](http://www.backbase.com)

### Web Stress Tool

Great Mid-Range Choice. Easy, Advanced Features. Free Trial.  
[www.neotys.com](http://www.neotys.com)

### Portal + BPM + ECM

Gestión unificada de personas, procesos y contenidos  
[www.polymita.com](http://www.polymita.com)

Anuncios 

## Accediendo a rutinas C y C++ desde Oracle

Creación: 09-04-2007

### 1. Introducción

En este tutorial os quiero mostrar como llamar a rutinas desarrolladas en c o cpp desde un procedimiento almacenado de ORACLE.

### 2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: HP COMPAQ Presario V6000 (Centrino Duo 1.66GHz, 2048 MB RAM, 100 GB HD)
- Sistema Operativo: Windows XP Home Edition
- Oracle 9.2
- MinGW 5 para windows.
- Eclipse como editor de ficheros

### 3. Creando el .c y el .cpp

Para comenzar vamos a crear dos pequeños ficheros, uno escrito en c, y otro escrito en cpp. Crearemos una función que escribirá en un fichero una pequeña frase. A continuación os muestro el código de nuestros archivos:

```
helloWord.c
```

```

#include <stdlib.h>

#include <stdio.h>

#include <windows.h>

#define EXPORT __declspec(dllexport)

EXPORT void HelloWorld(char* cadena, int* numero);

EXPORT void HelloWorld(char* cadena, int* numero)
{
FILE * f = fopen("c:\\HelloWordC.log", "aw");

fprintf(f, "Hola desde la libreria HelloWordC.dll:\n");

fprintf(f, "Datos antes de modificarse \nCadena: %s \nNumero: %d\n", cadena, *numero);

strcpy(cadena, "Modificada por helloWordC.dll");

*numero = 10;

fprintf(f, "Despues\nCadena: %s \nNumero: %d\n", cadena, *numero);

fflush(f);

fclose(f);
}

```

Al llamar a la función HelloWorld, se creará un fichero, si no existe ya, con el nombre HelloWordC.log en c:\ y se le añadirá la frase "Hola desde la librería HelloWordC.dll". Se imprimirán los parámetros antes y después de ser modificados, para ver que realmente los datos son obtenidos y modificados correctamente.

helloWord.cpp

```

#include <stdlib.h>

#include <stdio.h>

#include <windows.h>

#define EXPORT extern "C" __declspec(dllexport)

EXPORT void HelloWorld(char* cadena, int* numero);

EXPORT void HelloWorld(char* cadena, int* numero)
{
FILE * f = fopen("c:\\HelloWordCpp.log", "aw");

fprintf(f, "Hola desde la libreria HelloWordCpp.dll:\n");

fprintf(f, "Datos antes de modificarse \nCadena: %s \nNumero: %d\n", cadena, *numero);

strcpy(cadena, "Modificada por helloWordCpp.dll");

*numero = 20;

fprintf(f, "Despues\nCadena: %s \nNumero: %d\n", cadena, *numero);

fflush(f);

fclose(f);
}

```

La única diferencia entre las dos implementaciones está en la definición de EXPORT. En cpp se antepone *extern "C"* a *\_declspec(dllexport)*.

## 4. Creando las librerías

Vamos a crear un fichero makefile para facilitarnos la compilación de los ficheros y la creación de la librería. El contenido es el siguiente:

```
JDK_HOME=$(JAVA_HOME)

WIN32_CPP=g++

WIN32_C=gcc

CPPFILE=helloWord.cpp

CFILE=helloWord.c

CPPLIBNAMEWIN32=helloWordCpp.dll

CLIBNAMEWIN32=helloWordC.dll

CFLAGSWIN32=-DLOG -I$(JDK_HOME)\include -I$(JDK_HOME)\include\win32 -shared -Wl,--kill-at -Wall

all:

@echo "Uso: make {win32 | help}"

win32: $(FILE) $(JNIFILE)

$(WIN32_CPP) $(CFLAGSWIN32) -o $(CPPLIBNAMEWIN32) $(CPPFILE)

$(WIN32_C) $(CFLAGSWIN32) -o $(CLIBNAMEWIN32) $(CFILE)

help:

@echo "Uso: make win32"
```

Se crearán dos librerías. Una llamada helloWordCpp.dll generada a partir del fuente helloWord.cpp y otra llamada helloWordC.dll, generada a partir del fuente helloWord.c

Estas dos librerías son las que invocaremos desde Oracle.

## 5. Configurando Oracle para poder acceder a librerías externas

Debemos editar el listener de oracle (archivo listener.ora), situado en %oracle\_dir%\ora92\network\admin\listener.ora, y añadir a cada entrada de SID\_DESC la opción (ENVS = "EXTPROC\_DLLS=ANY"). A continuación os muestro un ejemplo:

[...]

```
SID_LIST_LISTENER =
    (SID_LIST =
        (SID_DESC =
            (SID_NAME = PLSExtProc)
            (ORACLE_HOME = C:\oracle\ora92)
            (PROGRAM = extproc)
            (ENVS = "EXTPROC_DLLS=ANY")
        )
    )
    (SID_DESC =
```

```

        (GLOBAL_DBNAME = WOODY)

        (ORACLE_HOME = C:\oracle\ora92)

        (SID_NAME = WOODY)

        (ENVS = "EXTPROC_DLLS=ANY")

    )

)

```

[...]

Una vez editado el fichero, debemos reiniciar el servicio del listener de Oracle.

## 6. Llamada a las librerías

Lo primero que debemos hacer es crear un objeto LIBRARY asociado a la librería a la que queremos acceder. Lanzamos en la consola de Oracle:

```

CREATE OR REPLACE LIBRARY helloWordC AS 'C:\helloWordC.dll'; /

CREATE OR REPLACE LIBRARY helloWordCpp AS 'C:\helloWordCpp.dll'; /

```

Como vemos, las librerías se encuentran en [C:\](#). Oracle no comprueba que la librería es correcta (el path, el código, etc.) hasta el momento de la ejecución, por lo tanto no sabríamos si lo hemos hecho bien hasta que no lanzáramos la prueba.

Ahora vamos a asociar dos procedimientos almacenados con las dos funciones HelloWord de las librerías. Lanzamos en la consola de Oracle:

```

CREATE OR REPLACE PROCEDURE helloC(cadena IN OUT VARCHAR, numero IN OUT PLS_INTEGER) AS LANGUAGE C
LIBRARY helloWordC name "HelloWord"; /

CREATE OR REPLACE PROCEDURE helloCpp(cadena IN OUT VARCHAR, numero IN OUT PLS_INTEGER) AS LANGUAGE C
LIBRARY helloWordCpp name "HelloWord"; /

```

Hemos asociado un procedimiento helloC al procedimiento HelloWord() de la librería de C, y un procedimiento helloCpp asociado al HelloWord() de la librería de Cpp. Para mapear los parámetros podemos consultar el capítulo 2 del manual de oracle ([http://www.unix.org.ua/oreilly/oracle/prog2/ch21\\_04.htm](http://www.unix.org.ua/oreilly/oracle/prog2/ch21_04.htm)). Nos indica que VARCHAR mapea a un tipo char\* de C, y PLS\_INTEGER mapea a un tipo de dato int\* de C.

Vamos a crear otro procedimiento llamado test para poder probar nuestras librerías:

```

CREATE OR REPLACE PROCEDURE test

AS

    cadena VARCHAR(50);

    numero PLS_INTEGER;

begin

    cadena := 'Hola mundo';

    numero := 0;

    dbms_output.put_line('Valores iniciales');

    dbms_output.put_line('Cadena:' || cadena);

    dbms_output.put_line('Numero:' || to_char(numero));

    helloC(cadena, numero);

    dbms_output.put_line('Valores modificados con helloWordC.dll');

    dbms_output.put_line('Cadena:' || cadena);

    dbms_output.put_line('Numero:' || to_char(numero));

    helloCpp(cadena, numero);

    dbms_output.put_line('Valores modificados con helloWordCpp.dll');

    dbms_output.put_line('Cadena:' || cadena);

```

```
dbms_output.put_line('Numero:' || to_char(numero));
```

```
end; /
```

Se imprimen los valores iniciales de las variables y se realiza las llamadas a las funciones, imprimiéndose los parámetros para comprobar que se modifican correctamente.

Ahora lanzamos nuestro test. Primero hay que activar el serveroutput de oracle para que se impriman las variables:

```
set serveroutput on;
```

Lanzamos el test:

```
exec test;
```

Comprobamos que los cambios se realizan correctamente, viendo el resultado por pantalla en la consola de Oracle. Comprobamos también que se generan los dos archivos de texto en [C:\](#) con el contenido descrito anteriormente.

## 7. El código fuente

[Aquí](#) adjunto el fuente y las dll de windows.

## 8. Sobre el autor

José Carlos López Díaz, Ingeniero en Informática

<mailto:jclopez@autentia.com>

Autentia Real Business Solutions S.L - "Soporte a Desarrollo"



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](#).  
[Puedes opinar sobre este tutorial aquí](#)



## Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿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?**

[info@autentia.com](mailto:info@autentia.com)

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

**Autentia = Soporte a Desarrollo & Formación**

Gestión de contenidos

[Autentia S.L.](#) Somos expertos en:  
**J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..**  
 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.

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

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

### Nombre Corto

[Cliente y Servidor DCOM con MS Visual C++](#)

[Introducción a Dev-C++](#)

[Instalación de Oracle BPEL](#)

[Instalación de Oracle8i](#)

[Primeros programas OpenGL con Visual C++](#)

[Instalación de Oracle Application Server 10g \(IAS\)](#)

[Implementación de clases java con métodos nativos escritos en C++ \(JNI\)](#)

[Testeando Oracle Forms 10G con Load Runner](#)

### Descripción

Os mostramos como construir un servidor y cliente de Automatización OLE con las facilidades que proporciona el entorno de desarrollo Microsoft Visual C++

Os mostramos el entorno de desarrollo Dev-C++, una buena opción a la hora de desarrollar en C/C++ utilizando un entorno gratuito, siempre mejor que hacerlo a mano.

EN este tutorial aprenderemos a instalar la herramienta de Oracle BPEL

Cesar Crespo nos enseña las particularidades de la instalación de Oracle8i

Con este tutorial, como insertar código OpenGL en un programa Visual C++, creando un sistema planetario básico.

En este tutorial os mostramos como instalar el Oracle Application Server, a fin de desplegar alguna aplicación web sobre el mismo

El objetivo de este tutorial es mostrar como se puede implementar una clase Java con métodos nativos escritos en C++ y contenidos en una DLL de Windows, utilizando JNI

En este tutorial os enseñamos a utilizar la herramienta de pruebas Load Runner para testear aplicaciones Oracle Forms 10G

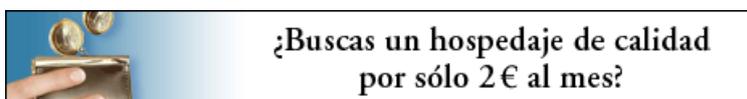
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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](mailto:rcanales@adictosaltrabajo.com) para su resolución.

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



[www.AdictosAlTrabajo.com](http://www.AdictosAlTrabajo.com) Optimizado 800X600