

¿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)


```
view plain print ?

01. DELIMITER $$
02.
03. -- Creamos el Schema si no existe
04. CREATE SCHEMA IF NOT EXISTS db_test;
05.
06. $$
07.
08. DROP PROCEDURE IF EXISTS db_test.procedureTemp;
09.
10. $$
11.
12. CREATE PROCEDURE db_test.procedureTemp()
13. BEGIN
14.
15.     -----
16.     -- Declaración de variables
17.     -----
18.
19.     DECLARE isOk      BOOL DEFAULT TRUE;
20.     DECLARE cuenta    INT DEFAULT 0;
21.     DECLARE cursor1 CURSOR FOR SELECT COUNT(*) FROM db_test.companies WHERE companykey='Autentia';
22.
23.     -- AL crear un índice que ya existe se genera el siguiente error
24.     DECLARE CONTINUE HANDLER FOR 1061 SET isOk = False;
25.
26.     -- AL crear un Foreign Key que ya existe se genera el siguiente error
27.     DECLARE CONTINUE HANDLER FOR 1005 SET isOk = False;
28.
29.     -- AL eliminar un Foreign Key que no existe se genera el siguiente error
30.     DECLARE CONTINUE HANDLER FOR 1025 SET isOk = False;
31.
32.     -- Para más información sobre Los errores:
33.     -- http://dev.mysql.com/doc/refman/5.0/en/declare-handlers.html
34.     -- http://dev.mysql.com/doc/refman/5.0/en/error-messages-server.html
35.
36.
37.     -----
38.     -- Creamos una tabla si no existiese, para ello consultamos Las tablas de MySQL que mantienen metainformación sobre Los esquemas
39.     -- Representa Los datos de empresas
40.     -----
41.
42.     SELECT COUNT(*) INTO cuenta FROM `information_schema`.`tables` WHERE TABLE_SCHEMA='db_test' AND TABLE_NAME='companies' LIMIT 1;
43.     IF (cuenta = 0) THEN
44.         CREATE TABLE db_test.companies (
45.             id INT(3) UNSIGNED NOT NULL AUTO_INCREMENT,
46.             companykey VARCHAR(25) NOT NULL,
47.             name VARCHAR(100) NOT NULL,
48.             url VARCHAR(255),
49.             PRIMARY KEY (id),
50.             UNIQUE companykeyIDX (companykey) -- No puede haber dos empresas con el mismo identificador
51.         ) ENGINE=InnoDB CHARSET=utf8 collate=utf8_general_ci;
52.     END IF;
53.
54.
55.     -----
56.     -- Creamos una tabla si no existiese.
57.     -- Representa Los datos de empleados (contactos) dentro de cada empresa
58.     -----
59.
60.     SELECT COUNT(*) INTO cuenta FROM `information_schema`.`tables` WHERE TABLE_SCHEMA='db_test' AND TABLE_NAME='employees' LIMIT 1;
61.     IF (cuenta = 0) THEN
62.         CREATE TABLE db_test.employees (
63.             id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
64.             employeekey VARCHAR(25) NOT NULL,
65.             companyid INT(3) UNSIGNED NOT NULL,
66.             name VARCHAR(20),
67.             lastNames VARCHAR(30),
68.             PRIMARY KEY (id),
69.             UNIQUE employeeIdx (companyid, employeekey), -- No puede haber dos empleados en La misma empresa con el mismo identificador
70.             KEY companyIDX (companyid), -- Índice necesario para crear La restricción Foreign Key con La tabla companie:
71.             CONSTRAINT empToCompFK FOREIGN KEY (companyid) REFERENCES companies(id) ON DELETE CASCADE ON UPDATE CASCADE
72.         ) ENGINE=InnoDB CHARSET=utf8 collate=utf8_general_ci;
73.     END IF;
74.
75.
76.     -----
77.     -- Insertamos una empresa de prueba si no existiese (Lo vamos a hacer a través de un cursor.
78.     -----
79.
80.     OPEN cursor1;
81.     FETCH cursor1 INTO cuenta;
82.     IF (cuenta = 0) THEN
83.         INSERT INTO db_test.companies (id, companykey, name, url) VALUES (1, 'Autentia', 'Autentia Real Business Solutions', 'http://www.autentia.com');
84.     END IF;
85.     CLOSE cursor1;
86.
87.     -----
88.     -- Insertamos un empleado de prueba si no existiese. (Lo vamos a hacer a traves de SELECT INTO.
89.     -----
90.
91.     SELECT COUNT(*) INTO cuenta FROM db_test.employees WHERE ((companyid=1) AND (employeekey='carlosgarcia'));
92.     IF (cuenta = 0) THEN
93.         INSERT INTO db_test.employees (companyid, employeekey, name, lastNames) VALUES (1, 'carlosgarcia','Carlos', 'García Pérez');
94.     END IF;
95.
96.
97.     -----
98.     -- Creamos el campo email del empleado si no existe
99.     -----
100.
101.     SELECT COUNT(*) INTO cuenta FROM `information_schema`.`COLUMNS` WHERE ((TABLE_SCHEMA='db_test') AND (TABLE_NAME='employees') AND (COLUMN_NAME='email')) LIMIT 1;
102.     IF (cuenta = 0) THEN
103.         ALTER TABLE db_test.employees ADD COLUMN email VARCHAR(100) NULL;
104.     END IF;
105.
106.
107.     -----
108.     -- Se accede mucho por el email del empleado, vamos a crear un índice por el campo email del empleado
109.     -----
110.
111.     -- Notas: No existe en el esquema una tabla que muestre Los índices (Aunque si Las restricciones en La tabla KEY COLUMN USAGE
112.     -- Una forma de recuperarLos sería a través de un cursor v "SHOW INDEX FROM db_test.employees", Pero es más fácil hacerLo
113.     -- controlando el error 1061 que se genera segun La documentación.
114.     ALTER TABLE db_test.employees ADD INDEX `idxEmployeeEmail` (email);
115.
116.
117.     -----
118.     -- Creamos una Foreign Key entre La tabla employeess y companie:
119.     -----
120.
121.     SELECT COUNT(*) INTO cuenta FROM `information_schema`.`KEY_COLUMN_USAGE` WHERE (CONSTRAINT_SCHEMA='db_test') AND
122.         (CONSTRAINT_NAME='FK_Temporal') AND
123.         (TABLE_NAME='employees');
124.
125.     IF (cuenta = 0) THEN
126.         ALTER TABLE db_test.employees ADD CONSTRAINT `FK_Temporal` FOREIGN KEY `FK_Temporal` (companyid) REFERENCES companies(id) ON DELETE CASCADE ON UPDATE CASCADE;
127.     END IF;
128.
129.     -- Borramos una ForeignKey.Si el Foreign Key no existiese se produciría el error 1025 (Como tenemos controlado el error el Script con
130.     ALTER TABLE db_test.employees DROP FOREIGN KEY `FK_Temporal`;
131.
132. END
133.
134.
135. $$
136.
137. DELIMITER ;
138.
139. -- Invocamos el procedimiento almacenado
140. CALL db_test.procedureTemp();
141.
142.
143. -- Borramos el procedimiento almacenado
144. DROP PROCEDURE IF EXISTS db_test.procedureTemp;
145.
```

Ejemplo en SQLServer.

```
view plain print ?
01.  -----
02.  -- Creamos una base de datos si no existiese.
03.  -----
04.
05.  IF NOT EXISTS (SELECT * from sys.databases where name = 'db_test')
06.  BEGIN
07.      CREATE DATABASE db_test;
08.  END
09.
10.  -- Establecemos la base de datos prodeterminada
11.  USE db_test;
12.
13.  -----
14.  -- Creamos una tabla si no existiese.
15.  -- Representa los datos de empresas
16.  -----
17.
18.
19.  IF NOT EXISTS (SELECT * FROM sys.sysobjects WHERE name='companies' AND xtype='U')
20.  BEGIN
21.      CREATE TABLE companies (
22.          id          INTEGER          NOT NULL IDENTITY,
23.          companykey  VARCHAR(25)     NOT NULL,
24.          name        VARCHAR(100)    NOT NULL,
25.          url         VARCHAR(255),
26.          PRIMARY KEY (id)
27.      );
28.  END
29.
30.  -----
31.  -- Creamos un índice si no existe.
32.  -- No puede haber dos empresas con el mismo identificador
33.  -----
34.
35.  IF NOT EXISTS (SELECT * FROM sys.sindexes WHERE name='companyKeyIDX' AND id=(SELECT id FROM sys.sysobjects WHERE name='companies'))
36.  BEGIN
37.      CREATE UNIQUE INDEX "companyKeyIDX" ON companies (companykey);
38.  END
39.
40.  -----
41.  -- Creamos una tabla si no existiese.
42.  -- Representa los datos de empleados (contactos) dentro de cada empresa
43.  -----
44.
45.
46.  IF NOT EXISTS (SELECT * FROM sys.sysobjects WHERE name='employees' AND xtype='U')
47.  BEGIN
48.      CREATE TABLE employees (
49.          id          INTEGER          NOT NULL IDENTITY,
50.          employeekey  VARCHAR(25)     NOT NULL,
51.          companyid    INTEGER          NOT NULL,
52.          name         VARCHAR(20),
53.          lastNames    VARCHAR(30),
54.          PRIMARY KEY (id),
55.          CONSTRAINT employeeIdx UNIQUE (companyid, employeekey) -- No puede haber dos empleados en la misma empresa con el mismo identificador
56.      )
57.  END
58.
59.  -----
60.  -- Insertamos una empresa de prueba si no existiese
61.  -----
62.
63.
64.  IF NOT EXISTS (SELECT id FROM companies WHERE companykey='Autentia')
65.  BEGIN
66.      SET IDENTITY_INSERT companies ON; -- Necesario para poder insertar en un campo IDENTITY (Autonumérico);
67.      INSERT INTO companies (id, companykey, name, url) VALUES (1, 'Autentia', 'Autentia Real Business Solutions', 'http://www.autentia.com');
68.      SET IDENTITY_INSERT companies OFF;
69.  END
70.
71.  -----
72.  -- Insertamos un empleado de prueba si no existiese. (Lo vamos a hacer a través de SELECT INTO
73.  -----
74.
75.
76.  IF NOT EXISTS (SELECT id FROM employees WHERE ((companyid=1) AND (employeekey='carlosgarcia'))))
77.  BEGIN
78.      INSERT INTO employees (companyid, employeekey, name, lastNames) VALUES (1, 'carlosgarcia', 'Carlos', 'García Pérez');
79.  END
80.
81.  -----
82.  -- Creamos el campo email del empleado si no existe
83.  -----
84.
85.
86.  IF NOT EXISTS (SELECT id FROM sys.syscolumns where name='email' and id=(SELECT id FROM sys.sysobjects WHERE name='employees' AND xtype='U'))
87.  BEGIN
88.      ALTER TABLE employees ADD email VARCHAR(100) NULL;
89.  END
90.
91.  -----
92.  -- Se accede mucho por el email del empleado, vamos a crear un índice por el campo email del empleado
93.  -----
94.
95.
96.  IF NOT EXISTS (SELECT id from sys.sindexes WHERE ((name='idxEmployeeEmail') AND (id=(SELECT id FROM sys.sysobjects WHERE name='employees' AND xtype='U'))))
97.  BEGIN
98.      CREATE INDEX "idxEmployeeEmail" ON employees (email);
99.  END
100.
101.  -----
102.  -- Creamos una Foreign Key entre la tabla employees y companie:
103.  -----
104.
105.  IF NOT EXISTS (SELECT * from sys.objects WHERE (name='empToCompFK') AND (type='F') AND (parent_object_id=(SELECT id FROM sys.sysobjects WHERE name='employees' AND :
106.  BEGIN
107.      ALTER TABLE employees ADD CONSTRAINT empToCompFK FOREIGN KEY (companyid) REFERENCES companies(id) ON DELETE CASCADE ON UPDATE CASCADE;
108.  END
```

Bueno, espero que os haya servido de utilidad este pequeño tutorial, yo personalmente lo usaré a modo de consulta, pues hay veces que uno se tira mucho tiempo en otros entornos y luego vienen bien estos tutoriales para refrescar la memoria :-).

Un saludo.
Carlos García. Creador de [MobileTest](#).

- Puedes opinar sobre este tutorial [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 FAVORITES](#)



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



Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subcripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
XAMPP	Nuestro compañero Alfonso nos muestra este interesante paquete que integra un servidor Apache, PHP, Perl y MySQL, sencillo de instalar y multiplataforma	2008-02-04	1900	pdf
Administracion Web de MySQL	Os mostramos como instalar en vuestro PC phpMySQL, un potente gestor Web de MySQL utilizado en la mayoría de los Hostings.	2003-12-27	16445	pdf
Apache, MySQL y PHP	Os mostramos como configurar Apache, MySQL y PHP en vuestra máquina	2003-12-27	39357	pdf
MySql en Windows	MySQL es una de las principales bases de datos "gratuitas" que podemos encontrar en Internet. En este tutorial aprenderéis a instalarlo en Windows	2003-06-23	56039	pdf
Desarrollos Web en PHP con AppServ 2.5.6 y Eclipse PDT	En este tutorial veremos cómo configurar un entorno para desarrollar aplicaciones web con PHP, basado en PHP 5.1.4, MySQL 5.0, Apache HTTP Server 2.2.2, phpMyAdmin 2.8.2 (incluido en AppServ 2.5.6) y Eclipse PDT Europa.	2008-06-04	927	pdf
Modelado de MySQL con herramientas gratuitas	Os mostramos otra alternativa de modelado gráfico de MySql.	2004-01-25	24170	pdf
Administración Web de MySQL	En este tutorial se mostrará como administrar MySQL de forma rápida y muy sencilla a través de páginas webs implementadas con tecnología PHP, para ello se utilizará la herramienta PHPmyAdmin	2007-04-03	4218	pdf
JDBC y MySQL	En el tutorial anterior vimos como instalar MySQL en Windows, ahora vamos a ver como acceder desde una aplicación Java.	2003-06-24	65933	pdf
Modelado Gráfico de MySQL	Os mostramos como instalar y utilizar DeZing e Importer de Datanamic para crear el modelo lógico y físico de bustra base de datos MySQL	2004-01-07	20553	pdf
CMP Entity Beans y MySql	Os mostramos como crear un Entity Bean con persistencia controlada por el servidor, configurado para usar MySql	2003-10-03	21355	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 canales@adictosaltrabajo.com para su resolución.