Anexos

1. Modelo de datos

Modelo de datos

-- -------------------------------------------------------------------

-- Proyecto Fin de Carrera

--

-- Definicion de tablas para MySQL

--

-- Autor: Javier Gonzalez

--

-- -------------------------------------------------------------------

DROP DATABASE SDP;

CREATE DATABASE IF NOT EXISTS SDP;

USE SDP;

SET FOREIGN\_KEY\_CHECKS = 0 ;

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- TABLAS DE CONFIGURACION Y GENERALES

-- PREFIJO: CFG

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- Tabla de configuracion de sistema

-- Cuando no hay acceso a la base de datos se corresponde con el

-- fichero de propiedades

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS CFG\_CONFIGURACION;

CREATE TABLE CFG\_CONFIGURACION (

clave VARCHAR (127) NOT NULL

,valor VARCHAR (255) NOT NULL

,grupo INTEGER NOT NULL

,tipo INTEGER NOT NULL

,mask VARCHAR (127)

,minimo VARCHAR (32)

,maximo VARCHAR (32)

,tooltip INTEGER NOT NULL

);

ALTER TABLE CFG\_CONFIGURACION ADD PRIMARY KEY ( clave );

-- -------------------------------------------------------------------

-- Tablas de codigos

-- Contiene las descripciones de los codigos

-- localizadas para cada idioma disponible

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS CFG\_CODIGOS ;

CREATE TABLE CFG\_CODIGOS (

grupo INTEGER NOT NULL -- Grupo de codigos

,codigo INTEGER NOT NULL -- Identificador

,lang VARCHAR(02) NOT NULL -- Codigo de idioma

,valor VARCHAR(64) NOT NULL -- Descripcion

);

ALTER TABLE CFG\_CODIGOS ADD PRIMARY KEY ( grupo , codigo , lang);

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- TABLAS OPERATIVAS BASE

-- PREFIJO: SDP

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- ----------------------------------------------------------

-- Areas y aplicaciones

-- ----------------------------------------------------------

DROP TABLE IF EXISTS SDP\_APLICACIONES CASCADE ;

CREATE TABLE SDP\_APLICACIONES (

aplicacion VARCHAR (32) NOT NULL

,id BIGINT UNSIGNED NOT NULL

,descripcion VARCHAR (512) NOT NULL

,padre BIGINT UNSIGNED NOT NULL

,volumen INTEGER NOT NULL

,uid VARCHAR (32) NOT NULL

,tms TIMESTAMP NOT NULL

);

ALTER TABLE SDP\_APLICACIONES ADD PRIMARY KEY ( aplicacion );

ALTER TABLE SDP\_APLICACIONES ADD UNIQUE INDEX ( id );

-- ----------------------------------------------------------

-- Lista de mascaras para asociar modulos a aplicaciones

-- ----------------------------------------------------------

DROP TABLE IF EXISTS SDP\_REL\_APP\_MOD CASCADE ;

CREATE TABLE SDP\_REL\_APP\_MOD (

idAppl BIGINT UNSIGNED NOT NULL -- Aplicacion

,mask VARCHAR (12) NOT NULL -- Mascara

,fijo INTEGER NOT NULL -- Numero de car. fijos (-1 todos)

,peso INTEGER NOT NULL -- Peso por wildcars (\* = 1000)

,uid VARCHAR (32) NOT NULL -- uid que ha generado la regla

);

ALTER TABLE SDP\_REL\_APP\_MOD ADD PRIMARY KEY ( idAppl , mask );

-- ----------------------------------------------------------

-- Definicion basica de un modulo

-- Pertenece a una aplicacion

-- ----------------------------------------------------------

DROP TABLE IF EXISTS SDP\_MODULOS CASCADE ;

CREATE TABLE SDP\_MODULOS (

idAppl BIGINT UNSIGNED NOT NULL -- Id de la aplicacion

,idModulo BIGINT UNSIGNED NOT NULL -- Id unico del modulo

,idVersion BIGINT UNSIGNED NOT NULL -- Id de la version actual

,tipo INTEGER NOT NULL -- Tipo de modulo

,nombre VARCHAR(32) NOT NULL -- Nombre del modulo

,estado INTEGER NOT NULL -- Activo (1) o Inactivo(0)

,uid VARCHAR(32) NOT NULL

,tms TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

ALTER TABLE SDP\_MODULOS ADD PRIMARY KEY ( idAppl , idModulo );

ALTER TABLE SDP\_MODULOS ADD UNIQUE INDEX ( idModulo );

-- ----------------------------------------------------------

-- ----------------------------------------------------------

-- TABLAS DE INFORMACION DE UN MODULO

-- PREFIJO: MOD

-- ----------------------------------------------------------

-- ----------------------------------------------------------

-- ----------------------------------------------------------

-- Un modulo tiene varias versiones

-- ----------------------------------------------------------

DROP TABLE IF EXISTS MOD\_VERSIONES CASCADE ;

CREATE TABLE MOD\_VERSIONES (

idVersion BIGINT UNSIGNED NOT NULL -- Secuencia de la version

,idModulo BIGINT UNSIGNED NOT NULL -- Id del modulo

,firma VARCHAR (64) NOT NULL -- firma digital

,nombre VARCHAR (32) NOT NULL -- Nombre del modulo

,tipo INTEGER NOT NULL -- Tipo de modulo

,fichero INTEGER NOT NULL -- Flag tiene ficheros

,descripcion VARCHAR (1024) NOT NULL -- Posible comentario

,uid VARCHAR (32) NOT NULL -- uid que crea o actualiza

,tms TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

ALTER TABLE MOD\_VERSIONES ADD PRIMARY KEY ( idVersion );

ALTER TABLE MOD\_VERSIONES ADD UNIQUE KEY ( idModulo , idVersion);

ALTER TABLE MOD\_VERSIONES ADD UNIQUE KEY ( firma );

ALTER TABLE MOD\_VERSIONES ADD FOREIGN KEY(idModulo)

REFERENCES SDP\_MODULOS (idModulo )

ON DELETE CASCADE;

-- ----------------------------------------------------------

-- Datos resumidos de la version

-- ----------------------------------------------------------

DROP TABLE IF EXISTS MOD\_RESUMEN CASCADE ;

CREATE TABLE MOD\_RESUMEN (

idVersion BIGINT UNSIGNED NOT NULL -- Identificador de la version

,bytes BIGINT UNSIGNED NOT NULL -- Bytes del modulo

,lineas INTEGER NOT NULL -- Lineas del modulo

,parrafos INTEGER NOT NULL -- Parrafos

,sentencias INTEGER NOT NULL -- Sentencias

,blancos INTEGER NOT NULL -- Lineas en blanco

,comentarios INTEGER NOT NULL -- Lineas de comentarios

,decoradores INTEGER NOT NULL -- Comentarios decoradores

,verbosData INTEGER NOT NULL -- Sentencias

,verbosIO INTEGER NOT NULL -- Sentencias

,verbosControl INTEGER NOT NULL -- Sentencias

,verbosFlujo INTEGER NOT NULL -- Sentencias

,verbosArit INTEGER NOT NULL -- Sentencias

,verbosLang INTEGER NOT NULL -- Sentencias

,ficheros INTEGER NOT NULL -- Persistencia

);

ALTER TABLE MOD\_RESUMEN ADD PRIMARY KEY ( idVersion );

ALTER TABLE MOD\_RESUMEN ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Divisisones y secciones

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS MOD\_SECCIONES ;

CREATE TABLE MOD\_SECCIONES (

idVersion BIGINT UNSIGNED NOT NULL

,divIdentification INTEGER NOT NULL

,divEnvironment INTEGER NOT NULL

,divData INTEGER NOT NULL

,divProcedure INTEGER NOT NULL

,sectConfiguration INTEGER NOT NULL

,sectInputOutput INTEGER NOT NULL

,sectFile INTEGER NOT NULL

,sectWorking INTEGER NOT NULL

,sectLocal INTEGER NOT NULL

,sectLinkage INTEGER NOT NULL

,sectScreen INTEGER NOT NULL

);

ALTER TABLE MOD\_SECCIONES ADD PRIMARY KEY ( idVersion );

ALTER TABLE MOD\_SECCIONES ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Ficheros y tablas

-- Maestro lo puede marcar:

-- El programa

-- El administrador

-- El sistema automaticamente

-- En este caso, se elige el que mas registros haya

-- procesado por estadistica

-- Si lo ha puesto el sistema, lo cambiara si ha

-- habido otro proceso que ha leido mas registros

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS MOD\_FICHEROS ;

CREATE TABLE MOD\_FICHEROS (

idVersion BIGINT UNSIGNED NOT NULL -- Bytes del modulo

,idFile INTEGER NOT NULL -- Id del registro

,logico VARCHAR (64) NOT NULL -- Nombre logico

,fisico VARCHAR (64) NOT NULL -- Nombre fisico

,tipo INTEGER NOT NULL -- Tipo de fichero segun TP

,acceso INTEGER NOT NULL -- Tipo de acceso

,maestro INTEGER NOT NULL -- Indicador de fichero maestro

,leido BIGINT UNSIGNED NOT NULL -- Registros leidos

);

ALTER TABLE MOD\_FICHEROS ADD PRIMARY KEY ( idVersion , idFile );

ALTER TABLE MOD\_FICHEROS ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- ----------------------------------------------------------

-- Dependencias

-- ----------------------------------------------------------

DROP TABLE IF EXISTS MOD\_DEPENDENCIAS ;

CREATE TABLE MOD\_DEPENDENCIAS (

idVersion BIGINT UNSIGNED NOT NULL -- Id de la version del modulo

,modulo VARCHAR(64) NOT NULL -- Nombre del modulo o copy

,tipo INTEGER NOT NULL -- Tipo de dependencia

,subTipo INTEGER NOT NULL -- Tipo de dependencia

,estado INTEGER NOT NULL -- Verificado o no (0/1)

);

ALTER TABLE MOD\_DEPENDENCIAS ADD PRIMARY KEY ( idVersion , modulo );

ALTER TABLE MOD\_DEPENDENCIAS ADD KEY ( idVersion );

ALTER TABLE MOD\_DEPENDENCIAS ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- ----------------------------------------------------------

-- Grafo

-- ----------------------------------------------------------

DROP TABLE IF EXISTS MOD\_GRAFO ;

CREATE TABLE MOD\_GRAFO (

idVersion BIGINT UNSIGNED NOT NULL -- Id de la version del modulo

,idGrafo BIGINT UNSIGNED NOT NULL -- Id del camino

,orden BIGINT UNSIGNED NOT NULL -- Secuencia

,tipo INTEGER NOT NULL -- Tipo de nodo

,nombre VARCHAR(64) NOT NULL -- Nombre del nodo

,peso INTEGER NOT NULL -- Complejidad del nodo

);

ALTER TABLE MOD\_GRAFO ADD PRIMARY KEY ( idVersion , idGrafo, orden );

ALTER TABLE MOD\_DEPENDENCIAS ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Modulo fuente original comprimido

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS MOD\_FUENTE CASCADE ;

CREATE TABLE MOD\_FUENTE (

idVersion BIGINT UNSIGNED NOT NULL -- Identificador de la version del modulo

,source BLOB NOT NULL -- Fuente en formato ZIP

);

ALTER TABLE MOD\_FUENTE ADD PRIMARY KEY ( idVersion );

ALTER TABLE MOD\_FUENTE ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Modulo fuente original comprimido de programas

-- que han fallado en el analisis

-- Se analizaran para ver la construcion que ha fallado

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS MOD\_FUENTE\_ERR CASCADE ;

CREATE TABLE MOD\_FUENTE\_ERR (

tms TIMESTAMP NOT NULL -- Timestamp de ejecución

,nombre VARCHAR (32) NOT NULL -- Nombre del modulo

,uid VARCHAR (32) NOT NULL -- uid que la ha compilado

,source BLOB NOT NULL -- Fuente en formato ZIP

);

ALTER TABLE MOD\_FUENTE\_ERR ADD PRIMARY KEY ( tms );

-- ----------------------------------------------------------

-- Bloques de codigo

-- ----------------------------------------------------------

DROP TABLE IF EXISTS MOD\_BLOQUES ;

CREATE TABLE MOD\_BLOQUES (

idVersion BIGINT UNSIGNED NOT NULL

,orden INTEGER NOT NULL -- Numero de bloque

,lineBeg INTEGER NOT NULL -- Inicio de linea

,lineEnd INTEGER NOT NULL -- Fin de linea

,sentencias INTEGER NOT NULL -- Sentencias del bloque

,usado INTEGER NOT NULL -- Se ha ejecutado (1 = si)

);

ALTER TABLE MOD\_BLOQUES ADD PRIMARY KEY ( idVersion , orden );

ALTER TABLE MOD\_BLOQUES ADD KEY ( idVersion );

ALTER TABLE MOD\_BLOQUES ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- ----------------------------------------------------------

-- Lista de parrafos

-- ----------------------------------------------------------

DROP TABLE IF EXISTS MOD\_PARRAFOS ;

CREATE TABLE MOD\_PARRAFOS (

idVersion BIGINT UNSIGNED NOT NULL

,linea INTEGER NOT NULL -- Linea del parrafo

,nombre VARCHAR(64) NOT NULL -- Nombre del parrafo

,sentencias INTEGER NOT NULL -- Sentencias del parrafo

,referencias INTEGER NOT NULL -- Veces que se referencia

,indice INTEGER NOT NULL -- Indice en la taba Working

,cc INTEGER NOT NULL -- Complejidad ciclomatica McCabe

,isExit INTEGER NOT NULL -- Es un parrafo de exit?

);

ALTER TABLE MOD\_PARRAFOS ADD PRIMARY KEY ( idVersion , linea );

ALTER TABLE MOD\_PARRAFOS ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- ----------------------------------------------------------

-- Relacion entre parrafos

-- ----------------------------------------------------------

DROP TABLE IF EXISTS REL\_MOD\_PARRAFOS ;

CREATE TABLE REL\_MOD\_PARRAFOS (

idVersion BIGINT UNSIGNED NOT NULL

-- Id Unico.

-- Un parrafo A puede llamar a otro B varias veces

,id BIGINT UNSIGNED NOT NULL

,origen BIGINT UNSIGNED NOT NULL -- Linea del parrafo origen

,destino BIGINT UNSIGNED NOT NULL -- Linea del parrafo destino

);

ALTER TABLE REL\_MOD\_PARRAFOS ADD PRIMARY KEY ( idVersion , id );

ALTER TABLE REL\_MOD\_PARRAFOS ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- ----------------------------------------------------------

-- Sentencias no permitidas

-- ----------------------------------------------------------

DROP TABLE IF EXISTS MOD\_BADSTMTS ;

CREATE TABLE MOD\_BADSTMTS (

idVersion BIGINT UNSIGNED NOT NULL

,orden INTEGER NOT NULL -- Id Unico.

,sentencia VARCHAR(32) NOT NULL -- Sentencia no permitida

,begLine INTEGER NOT NULL -- Inicio de la sentencia

,endLine INTEGER NOT NULL -- Fin de la sentencia

,columna INTEGER NOT NULL -- Columna

);

ALTER TABLE MOD\_BADSTMTS ADD PRIMARY KEY ( idVersion , orden );

ALTER TABLE MOD\_BADSTMTS ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- TABLAS CARGADAS COMO RESULTADO DE UNA EJECUCION

-- PREFIJO: TRP

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS TRP\_SESIONES ;

CREATE TABLE TRP\_SESIONES (

idSesion VARCHAR(64) NOT NULL -- Id de la sesion

,idModulo VARCHAR(64) NOT NULL -- Id del modulo principal

,elapsed BIGINT UNSIGNED NOT NULL -- Elapsed time

,cpu BIGINT UNSIGNED NOT NULL -- CPU time

,suspend BIGINT UNSIGNED NOT NULL -- Suspend time

,leido BIGINT UNSIGNED NOT NULL -- Registros leidos

,escrito BIGINT UNSIGNED NOT NULL -- Registros escritos

,finished INTEGER NOT NULL -- Ha finalizado el proceso?

,uid VARCHAR(32) NOT NULL -- uid que lo ha ejecutado

,tms TIMESTAMP NOT NULL -- Timestamp de ejecución

);

ALTER TABLE TRP\_SESIONES ADD PRIMARY KEY ( idSesion, idModulo );

ALTER TABLE TRP\_SESIONES ADD FOREIGN KEY ( idModulo )

REFERENCES MOD\_VERSIONES ( firma )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Modulos invocados en la sesion incluyendo el modulo principal

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS TRP\_MODULOS ;

CREATE TABLE TRP\_MODULOS (

idSesion VARCHAR (64) NOT NULL -- Id de la sesion

,idModulo VARCHAR (64) NOT NULL -- Id de la version

,orden BIGINT UNSIGNED NOT NULL -- Contador de mensajes

,modulo VARCHAR (64) NOT NULL -- Nombre del modulo

,veces BIGINT UNSIGNED NOT NULL -- Numero de ocurrencias

,totElapsed BIGINT UNSIGNED NOT NULL -- Tiempo total

,totCpu BIGINT UNSIGNED NOT NULL -- Tiempo total en modo uid

,totSuspend BIGINT UNSIGNED NOT NULL -- Tiempo total en modo uid

,intElapsed BIGINT UNSIGNED NOT NULL -- Tiempo bloque

,intCpu BIGINT UNSIGNED NOT NULL -- Tiempo bloque en modo uid

,intSuspend BIGINT UNSIGNED NOT NULL -- Tiempo bloque en modo wait/suspend

);

ALTER TABLE TRP\_MODULOS ADD PRIMARY KEY ( idSesion , idModulo );

ALTER TABLE TRP\_MODULOS ADD FOREIGN KEY ( idSesion )

REFERENCES TRP\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- ----------------------------------------------------------

-- Registros de cobertura

-- Una sesion puede afectar a varios modulos

-- ----------------------------------------------------------

DROP TABLE IF EXISTS TRP\_COBERTURA ;

CREATE TABLE TRP\_COBERTURA (

idSesion VARCHAR(64) NOT NULL -- Firma de la sesion

,idModulo VARCHAR(64) NOT NULL -- Firma del modulo

,orden BIGINT UNSIGNED NOT NULL -- Numero de mensaje

,flags VARCHAR(4000) NOT NULL -- Tabla working con flags de uso

);

ALTER TABLE TRP\_COBERTURA ADD PRIMARY KEY ( idSesion, idModulo );

ALTER TABLE TRP\_COBERTURA ADD FOREIGN KEY ( idSesion )

REFERENCES TRP\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Registros de persistencia

-- Una sesion puede ejecutar varias veces el mismo modulo

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS TRP\_PERSISTENCIA ;

CREATE TABLE TRP\_PERSISTENCIA (

idSesion VARCHAR(64) NOT NULL -- Firma de la sesion

,idModulo VARCHAR(64) NOT NULL -- Firma del modulo

,orden BIGINT UNSIGNED NOT NULL -- Numero del mensaje

,flags VARCHAR(4000) NOT NULL -- Flags de bloques

);

ALTER TABLE TRP\_PERSISTENCIA ADD PRIMARY KEY ( idSesion, idModulo);

ALTER TABLE TRP\_PERSISTENCIA ADD FOREIGN KEY ( idSesion )

REFERENCES TRP\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- ----------------------------------------------------------

-- Registros de llamadas a parrafos

-- Datos obtenidos de la working

-- ----------------------------------------------------------

DROP TABLE IF EXISTS TRP\_PARR\_WORKING ;

CREATE TABLE TRP\_PARR\_WORKING (

idSesion VARCHAR(64) NOT NULL -- Firma de la sesion

,idModulo VARCHAR(64) NOT NULL -- Firma del modulo

,orden BIGINT UNSIGNED NOT NULL -- Numero de mensaje

,datos VARCHAR(4095) NOT NULL -- Tabla working de contadores

);

ALTER TABLE TRP\_PARR\_WORKING ADD PRIMARY KEY ( idSesion, idModulo );

ALTER TABLE TRP\_PARR\_WORKING ADD FOREIGN KEY ( idSesion )

REFERENCES TRP\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Registros de llamadas a parrafos

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS TRP\_PERFORM ;

CREATE TABLE TRP\_PERFORM (

idSesion VARCHAR(64) NOT NULL -- Firma de la sesion

,idModulo VARCHAR(64) NOT NULL -- Firma del modulo

,orden BIGINT UNSIGNED NOT NULL -- Numero del mensaje

,etiqueta VARCHAR(64) NOT NULL -- Etiqueta: modulo, parrafo, nombre

,totElapsed BIGINT UNSIGNED NOT NULL -- Tiempo total

,totCpu BIGINT UNSIGNED NOT NULL -- Tiempo total en modo uid

,totSuspend BIGINT UNSIGNED NOT NULL -- Tiempo total en suspend

,intElapsed BIGINT UNSIGNED NOT NULL -- Tiempo bloque

,intCpu BIGINT UNSIGNED NOT NULL -- Tiempo bloque de CPU

,intSuspend BIGINT UNSIGNED NOT NULL -- Tiempo bloque de suspend

);

ALTER TABLE TRP\_PERFORM ADD PRIMARY KEY ( idSesion , idModulo, etiqueta , orden );

ALTER TABLE TRP\_PERFORM ADD FOREIGN KEY ( idSesion )

REFERENCES TRP\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Registros de uso de parrafos

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS TRP\_PARRAFOS ;

CREATE TABLE TRP\_PARRAFOS (

idSesion VARCHAR(64) NOT NULL -- Firma de la sesion

,idModulo VARCHAR(64) NOT NULL -- Firma del modulo

,orden BIGINT UNSIGNED NOT NULL -- Numero del mensaje

,etiqueta VARCHAR(64) NOT NULL -- Nombre del parrafo

,veces BIGINT UNSIGNED NOT NULL -- Numero de instancias

,totElapsed BIGINT UNSIGNED NOT NULL -- Tiempo total

,totCpu BIGINT UNSIGNED NOT NULL -- Tiempo total CPU

,totSuspend BIGINT UNSIGNED NOT NULL -- Tiempo total Suspend

,intElapsed BIGINT UNSIGNED NOT NULL -- Tiempo bloque

,intCpu BIGINT UNSIGNED NOT NULL -- Tiempo bloque CPU

,intSuspend BIGINT UNSIGNED NOT NULL -- Tiempo bloque Suspend

);

ALTER TABLE TRP\_PARRAFOS ADD PRIMARY KEY ( idSesion , idModulo, etiqueta , orden );

ALTER TABLE TRP\_PARRAFOS ADD FOREIGN KEY ( idSesion )

REFERENCES TRP\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Registros de llamadas a modulos

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS TRP\_CALL ;

CREATE TABLE TRP\_CALL (

idSesion VARCHAR(64) NOT NULL -- Firma de la sesion

,orden BIGINT UNSIGNED NOT NULL -- Numero del mensaje

,idCalling VARCHAR(64) NOT NULL -- Firma del modulo llamante

,idCalled VARCHAR(64) NOT NULL -- Firma del modulo llamado

,modulo VARCHAR(64) NOT NULL -- Nombre de la rutina llamada

,veces BIGINT UNSIGNED NOT NULL -- Numero de instancias

,totElapsed BIGINT UNSIGNED NOT NULL -- Tiempo total

,totCpu BIGINT UNSIGNED NOT NULL -- Tiempo CPU

,totSuspend BIGINT UNSIGNED NOT NULL -- Tiempo Suspend

,intElapsed BIGINT UNSIGNED NOT NULL -- Tiempo bloque

,intCpu BIGINT UNSIGNED NOT NULL -- Tiempo bloque CPU

,intSuspend BIGINT UNSIGNED NOT NULL -- Tiempo bloque Suspend

);

ALTER TABLE TRP\_CALL ADD PRIMARY KEY ( idSesion, orden, idCalling, idCalled );

ALTER TABLE TRP\_CALL ADD FOREIGN KEY ( idSesion )

REFERENCES TRP\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- TABLAS SUMARIZADAS DE LAS EJECUCIONES

-- PREFIJO: SES

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- Ejecuciones

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS SES\_SESIONES ;

CREATE TABLE SES\_SESIONES (

idSesion VARCHAR(64) NOT NULL -- Id de la sesion

,idVersion BIGINT UNSIGNED NOT NULL -- Id del modulo principal

,elapsed BIGINT UNSIGNED NOT NULL -- Elapsed time

,cpu BIGINT UNSIGNED NOT NULL -- CPU time

,suspend BIGINT UNSIGNED NOT NULL -- Suspend time

,leido BIGINT UNSIGNED NOT NULL -- Registros procesados

,escrito BIGINT UNSIGNED NOT NULL -- Registros generados

,finished INTEGER NOT NULL -- Ha finalizado el proceso?

,uid VARCHAR(32) NOT NULL -- uid que lo ha ejecutado

,tms TIMESTAMP NOT NULL -- Timestamp de ejecución

);

ALTER TABLE SES\_SESIONES ADD PRIMARY KEY ( idSesion, idVersion );

ALTER TABLE SES\_SESIONES ADD KEY ( idVersion );

ALTER TABLE SES\_SESIONES ADD KEY ( tms );

ALTER TABLE SES\_SESIONES ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Persistencia

-- En una misma sesion pueden existir varias ejecuciones

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS SES\_PERSISTENCIA ;

CREATE TABLE SES\_PERSISTENCIA (

idSesion VARCHAR(64) NOT NULL -- Id de la sesion

,idVersion BIGINT UNSIGNED NOT NULL -- Id de la version modulo

,orden BIGINT UNSIGNED NOT NULL -- Cuenta de ejecuciones

,idFile INTEGER NOT NULL -- Id del fichero

,maestro INTEGER NOT NULL -- Indicador de maestro

,acOpen BIGINT UNSIGNED NOT NULL -- Apertura

,acClose BIGINT UNSIGNED NOT NULL -- Cierre

,acInsert BIGINT UNSIGNED NOT NULL -- Insert / Write

,acRead BIGINT UNSIGNED NOT NULL -- Read

,acUpdate BIGINT UNSIGNED NOT NULL -- Update / Rewrite

,acDelete BIGINT UNSIGNED NOT NULL -- Delete

,acSelect BIGINT UNSIGNED NOT NULL -- Select

,acTotal BIGINT UNSIGNED NOT NULL -- Total de registros

);

ALTER TABLE SES\_PERSISTENCIA ADD PRIMARY KEY ( idSesion , idVersion , orden , idFile);

ALTER TABLE SES\_PERSISTENCIA ADD FOREIGN KEY ( idSesion )

REFERENCES SES\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Acumulado de entradas/salidas realizadas

-- Total de registros leidos escritos etc en la sesion

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS SES\_IO ;

CREATE TABLE SES\_IO (

idSesion VARCHAR(64) NOT NULL -- Id de la sesion

,idVersion BIGINT UNSIGNED NOT NULL -- Id de la version modulo

,acOpen BIGINT UNSIGNED NOT NULL -- Apertura

,acClose BIGINT UNSIGNED NOT NULL -- Cierre

,acInsert BIGINT UNSIGNED NOT NULL -- Insert / Write

,acRead BIGINT UNSIGNED NOT NULL -- Read

,acUpdate BIGINT UNSIGNED NOT NULL -- Update / Rewrite

,acDelete BIGINT UNSIGNED NOT NULL -- Delete

,acSelect BIGINT UNSIGNED NOT NULL -- Select

,acTotal BIGINT UNSIGNED NOT NULL -- Total de registros

,tms TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP

);

ALTER TABLE SES\_IO ADD PRIMARY KEY ( idSesion );

ALTER TABLE SES\_IO ADD KEY ( idVersion , tms );

ALTER TABLE SES\_IO ADD FOREIGN KEY ( idSesion )

REFERENCES SES\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Tiempos consumidos en cada parrafo de cada modulo por sesion

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS SES\_PARRAFOS ;

CREATE TABLE SES\_PARRAFOS (

idSesion VARCHAR(64) NOT NULL -- Id de la sesion

,idVersion BIGINT UNSIGNED NOT NULL -- Id de la version modulo

,orden BIGINT UNSIGNED NOT NULL -- Varios modulos por sesion

,nombre VARCHAR(64) NOT NULL -- Etiqueta: modulo, parrafo, nombre

,veces BIGINT UNSIGNED NOT NULL -- Numero de veces que se ha ejecutado

,totElapsed BIGINT UNSIGNED NOT NULL -- Tiempo total

,totCpu BIGINT UNSIGNED NOT NULL -- Tiempo total CPU

,totSuspend BIGINT UNSIGNED NOT NULL -- Tiempo total Suspend

,intElapsed BIGINT UNSIGNED NOT NULL -- Tiempo bloque

,intCpu BIGINT UNSIGNED NOT NULL -- Tiempo bloque CPU

,intSuspend BIGINT UNSIGNED NOT NULL -- Tiempo bloque Suspend

,avgTotElapsed BIGINT UNSIGNED NOT NULL -- Media tiempo total

,avgTotCpu BIGINT UNSIGNED NOT NULL -- Media tiempo CPU

,avgTotSuspend BIGINT UNSIGNED NOT NULL -- Media tiempo Suspend

,avgIntElapsed BIGINT UNSIGNED NOT NULL -- Media tiempo bloque

,avgIntCpu BIGINT UNSIGNED NOT NULL -- Media tiempo bloque CPU

,avgIntSuspend BIGINT UNSIGNED NOT NULL -- Media tiempo bloque Suspend

,minTotElapsed BIGINT UNSIGNED NOT NULL -- Minimo

,minTotCpu BIGINT UNSIGNED NOT NULL -- Minimo CPU

,minTotSuspend BIGINT UNSIGNED NOT NULL -- Minimo Suspend

,minIntElapsed BIGINT UNSIGNED NOT NULL -- Minimo bloque

,minIntCpu BIGINT UNSIGNED NOT NULL -- Minimo bloqur CPU

,minIntSuspend BIGINT UNSIGNED NOT NULL -- Minimo bloque Suspend

,maxTotElapsed BIGINT UNSIGNED NOT NULL -- Maximo

,maxTotCpu BIGINT UNSIGNED NOT NULL -- Maximo CPU

,maxTotSuspend BIGINT UNSIGNED NOT NULL -- Maximo Suspend

,maxIntElapsed BIGINT UNSIGNED NOT NULL -- Maximo bloque

,maxIntCpu BIGINT UNSIGNED NOT NULL -- Maximo bloque CPU

,maxIntSuspend BIGINT UNSIGNED NOT NULL -- Maximo bloque Suspend

);

ALTER TABLE SES\_PARRAFOS ADD PRIMARY KEY ( idSesion, idVersion, orden, nombre );

ALTER TABLE SES\_PARRAFOS ADD FOREIGN KEY ( idSesion )

REFERENCES SES\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Arbol de llamadas de la sesion

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS SES\_ARBOL ;

CREATE TABLE SES\_ARBOL (

idSesion VARCHAR(64) NOT NULL -- Id de la sesion

,idCalling BIGINT UNSIGNED NOT NULL -- Id de la version del modulo llamante

-- Id de la version del modulo llamado (0 si no se monitoriza)

,idCalled BIGINT UNSIGNED NOT NULL

,nombre VARCHAR(32) NOT NULL -- Nombre del modulo llamado

,veces BIGINT UNSIGNED NOT NULL -- Numero de veces que se ha ejecutado

,totElapsed BIGINT UNSIGNED NOT NULL -- Tiempo total

,totCpu BIGINT UNSIGNED NOT NULL -- Tiempo total CPU

,totSuspend BIGINT UNSIGNED NOT NULL -- Tiempo total Suspend

,avgElapsed BIGINT UNSIGNED NOT NULL -- Media tiempo total

,avgCpu BIGINT UNSIGNED NOT NULL -- Media tiempo CPU

,avgSuspend BIGINT UNSIGNED NOT NULL -- Media tiempo Suspend

,tms TIMESTAMP NOT NULL

);

ALTER TABLE SES\_ARBOL ADD PRIMARY KEY ( idSesion, idCalling, idCalled);

ALTER TABLE SES\_ARBOL ADD FOREIGN KEY ( idCalling )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Modulos y tiempos consumidos en la sesion

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS SES\_MODULOS ;

CREATE TABLE SES\_MODULOS (

idSesion VARCHAR(64) NOT NULL -- Id de la sesion

,idVersion BIGINT UNSIGNED NOT NULL -- Id de la version modulo

,orden BIGINT UNSIGNED NOT NULL -- Varios modulos por sesion

,nombre VARCHAR(64) NOT NULL -- Nombre del modulo

,veces BIGINT UNSIGNED NOT NULL -- Numero de veces que se ha ejecutado

,totElapsed BIGINT UNSIGNED NOT NULL -- Tiempo total

,totCpu BIGINT UNSIGNED NOT NULL -- Tiempo total CPU

,totSuspend BIGINT UNSIGNED NOT NULL -- Tiempo total Suspend

,intElapsed BIGINT UNSIGNED NOT NULL -- Tiempo bloque

,intCpu BIGINT UNSIGNED NOT NULL -- Tiempo bloque CPU

,intSuspend BIGINT UNSIGNED NOT NULL -- Tiempo bloque Suspend

,avgTotElapsed BIGINT UNSIGNED NOT NULL -- Media tiempo total

,avgTotCpu BIGINT UNSIGNED NOT NULL -- Media tiempo CPU

,avgTotSuspend BIGINT UNSIGNED NOT NULL -- Media tiempo Suspend

,avgIntElapsed BIGINT UNSIGNED NOT NULL -- Media tiempo bloque

,avgIntCpu BIGINT UNSIGNED NOT NULL -- Media tiempo bloque CPU

,avgIntSuspend BIGINT UNSIGNED NOT NULL -- Media tiempo bloque Suspend

,minTotElapsed BIGINT UNSIGNED NOT NULL -- Minimo tiempo total

,minTotCpu BIGINT UNSIGNED NOT NULL -- Minimo tiempo CPU

,minTotSuspend BIGINT UNSIGNED NOT NULL -- Minimo tiempo Suspend

,minIntElapsed BIGINT UNSIGNED NOT NULL -- Minimo tiempo bloque

,minIntCpu BIGINT UNSIGNED NOT NULL -- Minimo tiempo bloque CPU

,minIntSuspend BIGINT UNSIGNED NOT NULL -- Minimo tiempo bloque Suspend

,maxTotElapsed BIGINT UNSIGNED NOT NULL -- Maximo tiempo total

,maxTotCpu BIGINT UNSIGNED NOT NULL -- Maximo tiempo CPU

,maxTotSuspend BIGINT UNSIGNED NOT NULL -- Maximo tiempo Suspend

,maxIntElapsed BIGINT UNSIGNED NOT NULL -- Maximo tiempo bloque

,maxIntCpu BIGINT UNSIGNED NOT NULL -- Maximo tiempo bloque CPU

,maxIntSuspend BIGINT UNSIGNED NOT NULL -- Maximo tiempo bloque Suspend

,tms TIMESTAMP NOT NULL -- Timestamp de la sesion

);

ALTER TABLE SES\_MODULOS ADD PRIMARY KEY ( idSesion, idVersion, orden );

ALTER TABLE SES\_MODULOS ADD FOREIGN KEY ( idSesion )

REFERENCES SES\_SESIONES ( idSesion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- TABLAS SUMARIZADAS DEL USO DE MODULOS

-- PREFIJO: SUM

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS SUM\_MODULOS ;

CREATE TABLE SUM\_MODULOS (

idVersion BIGINT UNSIGNED NOT NULL -- Id de la version modulo

,idModulo BIGINT UNSIGNED NOT NULL -- Id del modulo

,nombre VARCHAR(64) NOT NULL -- Nombre del modulo

,veces BIGINT UNSIGNED NOT NULL -- Numero de veces que se ha ejecutado

,avgElapsed BIGINT UNSIGNED NOT NULL -- Media de tiempo

,avgCpu BIGINT UNSIGNED NOT NULL -- Media de CPU

,avgSuspend BIGINT UNSIGNED NOT NULL -- Media de suspend

,minElapsed BIGINT UNSIGNED NOT NULL -- Minimo de tiempo

,minCpu BIGINT UNSIGNED NOT NULL -- Minimo de CPU

,minSuspend BIGINT UNSIGNED NOT NULL -- Minimo de suspend

,maxElapsed BIGINT UNSIGNED NOT NULL -- Minimo de tiempo

,maxCpu BIGINT UNSIGNED NOT NULL -- Maximo de CPU

,maxSuspend BIGINT UNSIGNED NOT NULL -- Maximo de Suspend

,avgLeido BIGINT UNSIGNED NOT NULL -- Media de registros leidos

,avgEscrito BIGINT UNSIGNED NOT NULL -- Media de registros escritos

,minLeido BIGINT UNSIGNED NOT NULL -- Minimo de registros leidos

,minEscrito BIGINT UNSIGNED NOT NULL -- Minimo de registros escritos

,maxLeido BIGINT UNSIGNED NOT NULL -- Maximo de registros leidos

,maxEscrito BIGINT UNSIGNED NOT NULL -- Maximo de registros escritos

);

ALTER TABLE SUM\_MODULOS ADD PRIMARY KEY ( idVersion );

ALTER TABLE SUM\_MODULOS ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Tiempos consumidos en cada parrafo de cada modulo por sesion

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS SUM\_PARRAFOS ;

CREATE TABLE SUM\_PARRAFOS (

idVersion BIGINT UNSIGNED NOT NULL -- Id de la version modulo

,nombre VARCHAR(64) NOT NULL -- Etiqueta: modulo, parrafo, nombre

,sesiones BIGINT UNSIGNED NOT NULL -- Numero de sesiones

,veces BIGINT UNSIGNED NOT NULL -- Numero de veces que se ha ejecutado

,linea INTEGER NOT NULL -- Linea dentro del modulo fuente

,totElapsed BIGINT UNSIGNED NOT NULL -- Tiempo total

,totCpu BIGINT UNSIGNED NOT NULL -- Tiempo total CPU

,totSuspend BIGINT UNSIGNED NOT NULL -- Tiempo total Suspend

,intElapsed BIGINT UNSIGNED NOT NULL -- Tiempo bloque

,intCpu BIGINT UNSIGNED NOT NULL -- Tiempo bloque CPU

,intSuspend BIGINT UNSIGNED NOT NULL -- Tiempo bloque Suspend

,avgTotElapsed BIGINT UNSIGNED NOT NULL -- Media tiempo total

,avgTotCpu BIGINT UNSIGNED NOT NULL -- Media tiempo CPU

,avgTotSuspend BIGINT UNSIGNED NOT NULL -- Media tiempo Suspend

,avgIntElapsed BIGINT UNSIGNED NOT NULL -- Media tiempo bloque

,avgIntCpu BIGINT UNSIGNED NOT NULL -- Media tiempo bloque CPU

,avgIntSuspend BIGINT UNSIGNED NOT NULL -- Media tiempo bloque Suspend

,minTotElapsed BIGINT UNSIGNED NOT NULL -- Minimo tiempo total

,minTotCpu BIGINT UNSIGNED NOT NULL -- Minimo tiempo CPU

,minTotSuspend BIGINT UNSIGNED NOT NULL -- Minimo tiempo Suspend

,minIntElapsed BIGINT UNSIGNED NOT NULL -- Minimo tiempo bloque

,minIntCpu BIGINT UNSIGNED NOT NULL -- Minimo tiempo bloque CPU

,minIntSuspend BIGINT UNSIGNED NOT NULL -- Minimo tiempo bloque Suspend

,maxTotElapsed BIGINT UNSIGNED NOT NULL -- Maximo tiempo total

,maxTotCpu BIGINT UNSIGNED NOT NULL -- Maximo tiempo total CPU

,maxTotSuspend BIGINT UNSIGNED NOT NULL -- Maximo tiempo total Suspend

,maxIntElapsed BIGINT UNSIGNED NOT NULL -- Maximo tiempo bloque

,maxIntCpu BIGINT UNSIGNED NOT NULL -- Maximo tiempo bloque CPU

,maxIntSuspend BIGINT UNSIGNED NOT NULL -- Maximo tiempo bloque Suspend

,tms TIMESTAMP NOT NULL

);

ALTER TABLE SUM\_PARRAFOS ADD PRIMARY KEY ( idVersion, nombre );

ALTER TABLE SUM\_PARRAFOS ADD FOREIGN KEY ( idVersion )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- Arbol de llamadas

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS SUM\_ARBOL ;

CREATE TABLE SUM\_ARBOL (

idCalling BIGINT UNSIGNED NOT NULL -- Id de la version del modulo llamante

-- Id de la version del modulo llamado (0 si no se monitoriza)

,idCalled BIGINT UNSIGNED NOT NULL

,nombre VARCHAR(32) NOT NULL -- Nombre del modulo llamado

,sesiones BIGINT UNSIGNED NOT NULL -- Numero de sesiones monitorizadas

,veces BIGINT UNSIGNED NOT NULL -- Numero de veces que se ha ejecutado

,totElapsed BIGINT UNSIGNED NOT NULL -- Tiempo total

,totCpu BIGINT UNSIGNED NOT NULL -- Tiempo total CPU

,totSuspend BIGINT UNSIGNED NOT NULL -- Tiempo total Suspend

,avgElapsed BIGINT UNSIGNED NOT NULL -- Media tiempo total

,avgCpu BIGINT UNSIGNED NOT NULL -- Media tiempo CPU

,avgSuspend BIGINT UNSIGNED NOT NULL -- Media tiempo Suspend

,tms TIMESTAMP NOT NULL

);

ALTER TABLE SUM\_ARBOL ADD PRIMARY KEY ( idCalling, idCalled, nombre );

ALTER TABLE SUM\_ARBOL ADD FOREIGN KEY ( idCalling )

REFERENCES MOD\_VERSIONES ( idVersion )

ON DELETE CASCADE ;

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

-- TABLAS DE GESTION DE MENSAJES Y LOG

-- PREFIJO: LOG

-- -------------------------------------------------------------------

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS LOG\_LOGGING ;

CREATE TABLE LOG\_LOGGING (

secuencia BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY -- Clave unica

,idAppl BIGINT UNSIGNED NOT NULL -- Identificador de la aplicacion

,idModulo BIGINT UNSIGNED NOT NULL -- Identificador del modulo

,idVersion BIGINT UNSIGNED NOT NULL -- Identificador de la version

,idTipo INTEGER NOT NULL -- Identificador del tipo de mensaje

,idMsg INTEGER NOT NULL -- Identificador del mensaje

,uid VARCHAR(32) NOT NULL -- uid

,objeto VARCHAR(64) NOT NULL -- Nombre del objeto

,tms TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP

-- Campos para rellenar el mensaje

,dato0 VARCHAR(64) -- Campo 0

,dato1 VARCHAR(64) -- Campo 1

,dato2 VARCHAR(64) -- Campo 2

,dato3 VARCHAR(64) -- Campo 3

,dato4 VARCHAR(64) -- Campo 4

,dato5 VARCHAR(64) -- Campo 5

,dato6 VARCHAR(64) -- Campo 6

,dato7 VARCHAR(64) -- Campo 7

,dato8 VARCHAR(64) -- Campo 8

,dato9 VARCHAR(64) -- Campo 9

);

ALTER TABLE LOG\_LOGGING ADD INDEX ( tms DESC );

ALTER TABLE LOG\_LOGGING ADD INDEX ( idTipo );

ALTER TABLE LOG\_LOGGING ADD INDEX ( tms DESC, idMsg );

ALTER TABLE LOG\_LOGGING ADD INDEX ( idAppl , idModulo );

ALTER TABLE LOG\_LOGGING ADD INDEX ( uid );

-- -------------------------------------------------------------------

-- Tabla de mensajes

-- Esta localizada para cada idioma disponible

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS LOG\_MSG ;

CREATE TABLE LOG\_MSG (

idMsg INTEGER NOT NULL -- Identificador del mensaje

,lang CHAR(2) NOT NULL -- Codigo de idioma

,msg VARCHAR(255) NOT NULL -- Formato del mensaje

);

ALTER TABLE LOG\_MSG ADD PRIMARY KEY ( idMsg , lang );

-- -------------------------------------------------------------------

-- Tabla de tipos de mensajes

-- Contiene el teto descriptivo del tipo de mensaje

-- -------------------------------------------------------------------

DROP TABLE IF EXISTS LOG\_TIPO ;

CREATE TABLE LOG\_TIPO (

idTipo INTEGER NOT NULL -- Identificador del mensaje

,lang CHAR(2) NOT NULL -- Codigo de idioma

,msg VARCHAR(255) NOT NULL -- Formato del mensaje

);

ALTER TABLE LOG\_TIPO ADD PRIMARY KEY ( idTipo , lang );

SET FOREIGN\_KEY\_CHECKS = 1 ;

COMMIT;

Datos iniciales

-- -------------------------------------------------------------------

-- Proyecto Fin de Carrera

--

-- Datos de carga iniciales para el sistema

--

-- Autor: Javier Gonzalez

--

-- -------------------------------------------------------------------

USE SDP;

-- --------------------------------------------------------

-- Aplicaciones

-- --------------------------------------------------------

DELETE FROM SDP\_APLICACIONES;

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 1 , 'General' , 0 , 50 , 'U000001' , CURRENT\_TIMESTAMP , 'Aplicacion para modulos no identificados');

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 2 , 'Test' , 0 , 125 , 'U000002' , CURRENT\_TIMESTAMP , 'Aplicacion de pruebas' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 3 , 'Cuentas Personales' , 0 , 250 , 'U000003' , CURRENT\_TIMESTAMP , 'Cuentas personales' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 4 , 'Prestamos y creditos' , 0 , 1250 , 'U000004' , CURRENT\_TIMESTAMP , 'Prestamos y creditos' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 5 , 'Fondos' , 0 , 250 , 'U000005' , CURRENT\_TIMESTAMP , 'Fondos de inversion' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 6 , 'Recursos Humanos' , 0 , 135 , 'U000006' , CURRENT\_TIMESTAMP , 'Sistemas de recuros humanos' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 7 , 'Contabilidad' , 0 , 2450 , 'U000007' , CURRENT\_TIMESTAMP , 'Sistemas contables' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 8 , 'Tesoreria' , 0 , 1750 , 'U000008' , CURRENT\_TIMESTAMP , 'Tesoreria' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 101 , 'IVP' , 2 , 50 , 'U000002' , CURRENT\_TIMESTAMP , 'Installation Verification Process' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 102 , 'Sort' , 2 , 75 , 'U000002' , CURRENT\_TIMESTAMP , 'Ordenaciones' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 103 , 'Arbol' , 2 , 100, 'U000002' , CURRENT\_TIMESTAMP , 'Montaje de arboles' );

INSERT INTO SDP\_APLICACIONES ( id , aplicacion , padre , volumen , uid , tms , descripcion)

VALUES ( 104 , 'Errores' , 2 , 100, 'U000002' , CURRENT\_TIMESTAMP , 'Programas erroneos' );

DELETE FROM SDP\_REL\_APP\_MOD;

INSERT INTO SDP\_REL\_APP\_MOD ( idAppl , mask , fijo , peso , uid ) VALUES ( 1 , '\*' , 0 , 10000 , 'system');

INSERT INTO SDP\_REL\_APP\_MOD ( idAppl , mask , fijo , peso , uid ) VALUES ( 101 , 'IVP\*' , 4 , 1000 , 'system');

INSERT INTO SDP\_REL\_APP\_MOD ( idAppl , mask , fijo , peso , uid ) VALUES ( 102 , 'SRT\*' , 4 , 1000 , 'system');

INSERT INTO SDP\_REL\_APP\_MOD ( idAppl , mask , fijo , peso , uid ) VALUES ( 103 , 'ARB\*' , 4 , 1000 , 'system');

INSERT INTO SDP\_REL\_APP\_MOD ( idAppl , mask , fijo , peso , uid ) VALUES ( 104 , 'BAD\*' , 4 , 1000 , 'system');

DELETE FROM CFG\_CODIGOS;

-- Tipos de modulos (Grupo 1)

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 0 , 'XX' , 'Unknow' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 10 , 'XX' , 'Area / Application');

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 11 , 'XX' , 'Area' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 12 , 'XX' , 'Application' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 20 , 'XX' , 'Program' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 21 , 'XX' , 'Main program' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 22 , 'XX' , 'Module' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 30 , 'XX' , 'Copy' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 31 , 'XX' , 'Working copy' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 32 , 'XX' , 'Code working' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 0 , 'ES' , 'No identificado' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 10 , 'ES' , 'Area / Aplicacion' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 11 , 'ES' , 'Area' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 12 , 'ES' , 'Aplicacion' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 20 , 'ES' , 'Programa' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 21 , 'ES' , 'Programa principal');

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 22 , 'ES' , 'Rutina' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 30 , 'ES' , 'Copy' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 31 , 'ES' , 'Copy Working' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 1 , 32 , 'ES' , 'Copy Codigo' );

-- Tipos de dependencias (Grupo 2)

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 10 , 'XX' , 'General copy' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 11 , 'XX' , 'File description' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 13 , 'XX' , 'Working copybook' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 14 , 'XX' , 'Linkage copybook' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 15 , 'XX' , 'Code copybook' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 20 , 'XX' , 'Module' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 21 , 'XX' , 'Static Module' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 22 , 'XX' , 'Dynamic Module' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 23 , 'XX' , 'Variable' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 10 , 'ES' , 'Copy General' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 11 , 'ES' , 'Copybook ficheros' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 13 , 'ES' , 'Copybook Working' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 14 , 'ES' , 'Copybook Linkage' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 15 , 'ES' , 'Copybook codigo' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 20 , 'ES' , 'Modulo' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 21 , 'ES' , 'Modulo estatico' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 22 , 'ES' , 'Modulo dinamico' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 2 , 23 , 'ES' , 'Variable' );

-- Agrupacion de los parametros de configuracion (Grupo 3)

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 0 , 'XX' , 'General' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 1 , 'XX' , 'Messaging' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 2 , 'XX' , 'Parser' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 3 , 'XX' , 'Web' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 4 , 'XX' , 'Databaes' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 5 , 'XX' , 'System' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 0 , 'ES' , 'General' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 1 , 'ES' , 'Mensajeria' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 2 , 'ES' , 'Parser' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 3 , 'ES' , 'Web' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 4 , 'ES' , 'Base de datos' );

INSERT INTO CFG\_CODIGOS (grupo , codigo , lang , valor) VALUES ( 3 , 5 , 'ES' , 'Sistema' );

-- --------------------------------------------------------

-- Configuracion

-- Grupo 0 - Datos generales

-- Grupo 1 - Mensajeria

-- Grupo 2 - Parser

-- Grupo 3 - Web

-- Grupo 5 - Base de datos / Sistema

-- --------------------------------------------------------

DELETE FROM CFG\_CONFIGURACION;

INSERT INTO CFG\_CONFIGURACION (tooltip, grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 0 , 1, 'N/A' , 'N/A' , 'N/A', 'inst.name' , 'Serendipity');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 2 , 4, 'N/A' , 'N/A' , 'N/A', 'parser.dir.input' , 'p:/cases/in');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 2 , 4 , 'N/A' , 'N/A' , 'N/A', 'parser.dir.output' , 'p:/cases/out');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 2 , 2 , '[0-9]+' , '0' , '72', 'parser.margin.left' , '7');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 2 , 2 , '[0-9]+' , '12' , '255', 'parser.margin.right' , '72');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 1 , 'N/A' , '1' , '255', 'jms.type' , 'WMQ');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 1 , 'N/A' , '1' , '255', 'jms.queue.manager' , 'QMGR\_SDP');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 1 , 'N/A' , '1' , '255', 'jms.queue.parser' , 'SDP.PARSER');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 1 , 'N/A' , '1' , '255', 'jms.queue.TRAPPER' , 'SDP.TRAPPER');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 1 , 'N/A' , '1' , '255', 'jms.queue.collector' , 'SDP.COLLECTOR');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 1 , 'N/A' , '1' , '255', 'jms.queue.output' , 'SDP.CONSOLIDATOR');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 1 , 'N/A' , '1' , '255', 'jms.hostname' , 'localhost');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 2 , 'N/A' , '1' , '5', 'jms.port' , '1414');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 2 , '[0-9]+' , '1' , '60', 'jms.wait' , '1');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 2 , '[0-9]+' , '1' , '100', 'jms.group' , '10');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 1 , 2 , '[0-9]+' , '0' , '1000000', 'jms.commit.interval' , '1');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave, valor) VALUES

( 50000, 3 , 2 , '[0-4]' , '0' , '4', 'date.interval' , '0');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave , valor) VALUES

( 50000, 5 , 2, '[0-9]+' , '1' , '25' , 'code.max.versions' , '1');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave , valor) VALUES

( 50000, 5 , 2, '[0-9]+' , '1' , '100' , 'code.min.coverage' , '75');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave , valor) VALUES

( 50000, 5 , 2, '[0-9]+' , '0' , '1000' , 'code.max.complexity' , '10');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave , valor) VALUES

( 50000, 5 , 2, '[0-9]+' , 'N/A' , 'N/A' , 'sesion.desv.elapsed' , '25');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave , valor) VALUES

( 50000, 5 , 2, '[0-9]+' , 'N/A' , 'N/A' , 'sesion.desv.input' , '10');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave , valor) VALUES

( 50000, 5 , 2, '[0-9]+' , 'N/A' , 'N/A' , 'sesion.desv.output' , '10');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave , valor) VALUES

( 50000, 3 , 2, '[0-9]+' , '100' , '1000' , 'web.log.max.records' , '250');

INSERT INTO CFG\_CONFIGURACION (tooltip , grupo , tipo, mask, minimo, maximo, clave , valor) VALUES

( 50000, 3 , 1, 'N/A' , 'N/A' , 'N/A' , 'web.log.tms.interval' , 'H');

COMMIT;

-- -------------------------------------------------------------------

-- Proyecto Fin de Carrera

--

-- Mensajes de logging y alertas

--

-- Autor: Javier Gonzalez

--

-- -------------------------------------------------------------------

USE SDP;

DELETE FROM LOG\_MSG;

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 1, 'XX', 'Creado nuevo modulo: %s');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 2, 'XX', 'Modulo %s recompilado');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 3, 'XX', 'Creada nueva version del modulo %s');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 1001, 'XX', 'El tiempo de ejecucion se ha desviado un %s%% de su media');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 1002, 'XX', 'Se ha consumido un %s%% del tiempo de CPU');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 1003, 'XX', 'Se ha consumido un %s%% del tiempo en modo Suspend');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 1004, 'XX', 'El modulo se ha desviado un %s%% de su media');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 1011, 'XX', 'Se ha procesado un %s%% de registros menos que la media actual');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 1012, 'XX', 'Se ha procesado un %s%% de registros mas que la media actual');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 1013, 'XX', 'Se han generado un %s%% de registros menos que la media actual');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 1014, 'XX', 'Se han generado un %s%% de registros mas que la media actual');

INSERT INTO LOG\_MSG (idMsg, lang, msg)

VALUES ( 2001, 'XX', 'Usuario %s dado de alta como usuario');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES ( 2002, 'XX', 'Usuario %s dado de alta como administrador');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES ( 2003, 'XX', 'Usuario %s modificado');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES ( 2101, 'XX', '%s modificado de %s a %s');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES (10001, 'XX', 'Codigo de usuario invalido');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES (10002, 'XX', 'Se require un nombre de usuario');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES (10003, 'XX', 'Se requiere al menos un apellido');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES (10004, 'XX', 'Se requiere una direcion de correo electronico');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES (10005, 'XX', 'Direccion de correo electronico incorrecta');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES (10006, 'XX', 'Rol incorrecto');

INSERT INTO LOG\_MSG (idMsg, lang, msg) VALUES (10007, 'XX', 'Ya existe ese usuario');

COMMIT;

1. el sistema de mensajería

DEFINE PROCESS (PRC.PARSER) APPLTYPE(UNIX) APPLICID(SDPParser)

DEFINE PROCESS (PRC.TRAPPER) APPLTYPE(UNIX) APPLICID(SDPTrapper)

DEFINE PROCESS (PRC.COLLECTOR) APPLTYPE(UNIX) APPLICID(SDPCollector)

DEFINE QLOCAL (SDP.INIT) PUT(ENABLED) GET(ENABLED) +

NOTRIGGER NOSHARE

DEFINE QLOCAL (SDP.PARSER) +

PUT(ENABLED) GET(ENABLED) INITQ(SDP.INIT) +

MAXDEPTH(999999999) MAXMSGL(4194304) +

TRIGGER TRIGTYPE(FIRST) PROCESS(PRC.PARSER) +

DEFPSIST(YESY) DEFSOPT(SHARED) USAGE(NORMAL)

DEFINE QLOCAL (SDP.TRAPPER) +

PUT(ENABLED) GET(ENABLED) INITQ(SDP.INIT) +

MAXDEPTH(999999999) MAXMSGL(4194304) +

TRIGGER TRIGTYPE(FIRST) PROCESS(PRC.TRAPPER) +

DEFPSIST(YES) DEFSOPT(SHARED) USAGE(NORMAL)

DEFINE QLOCAL (SDP.COLLECTOR) +

PUT(ENABLED) GET(ENABLED) INITQ(SDP.INIT) +

MAXDEPTH(999999999) MAXMSGL(4194304) +

TRIGGER TRIGTYPE(FIRST) PROCESS(PRC.COLLECTOR) +

DEFPSIST(YES) DEFSOPT(SHARED) USAGE(NORMAL)

DEFINE QLOCAL (SDP.INIT) PUT(ENABLED) GET(ENABLED) +

NOTRIGGER NOSHARE

1. Script de compilación

#!/bin/sh

############################################################

# Utilidad para compilar un programa COBOL y analizarlo

#

# Autor : Javier Gonzalez Grandez

# Fecha : Ago - 2015

# Version: 1.0

############################################################

############################################################

# Variables globales

############################################################

TMPDIR=/tmp

OUTEXE=/PFC/cases/exe

OUTDLL=/PFC/cases/shared

VERBOSE=0

JAVA\_CMD="java -Djava.library.path=/opt/mqm/lib64"

############################################################

# Funciones

############################################################

showHelp() {

echo "Utilidad para compilar y preparar programas COBOL "

echo " "

echo "Uso: `basename $0` [opciones] lista\_de\_programas\_fuente "

echo "Opciones: "

echo " -o | --out Directorio de salida "

echo " -h | --help Muestra esta ayuda "

echo " -v | --verbose Muestra informacion de progreso "

exit 0

}

compila() {

SRC=`basename $2`

BASE=${SRC%%.\*}

if [ $VERBOSE -gt 0 -a $4 -eq 0 ] ; then

echo -n "Compilando $BASE "

fi

if [ $1 -eq 0 ] ; then

OPT=-x

EXT=exe

else

OPT=-m

EXT=so

fi

cobc $OPT -I. -o ${3}/${BASE}.${EXT} $2

return $?

}

############################################################

# Code

############################################################

OPTS=$(getopt -n "$0" -o o:vh --long "out:,verbose,help" -- "$@")

if [ $? -ne 0 ]; then

echo An unexpected error was occured

exit 9

fi

eval set -- "$OPTS"

while true;

do

case "$1" in

-h | --help)

showHelp

shift

;;

-v | --verbose)

VERBOSE=1

shift

;;

-o | --out)

if [ -n "$2" ] ; then

OUTDIR=$2

else

echo "ERROR: Falta el directorio de salida"

exit 1

fi

shift 2

;;

--) shift

break

;;

esac

done

if [ -z "$SDP\_ENV" ] ; then

echo ERROR No se ha establecido el entorno de ejecucion

exit 92

fi

if [ $# -eq 0 ] ; then

echo No se han indicado programas para compilar

fi

# Compilar los programas

for src in $\* ; do

SRC=`basename $src`

#verificar si tiene extension

FULL=${SRC##\*.}

if [ "$FULL" = "$SRC" ] ; then

SOURCE=${src}.${COB\_EXT}

else

SOURCE=$src

fi

# Identificar si es programa o rutina

DLL=0

egrep -qi "DIVISION[ \t]+USING" $SOURCE

if [ $? -eq 0 ] ; then

DLL=1

fi

compila $DLL $SOURCE $TMPDIR 0

rc=$?

if [ $rc -ne 0 ] ; then

if [ $VERBOSE -gt 0 ] ; then

echo " - KO"

echo "ERROR: El programa fuente no es correcto"

fi

exit $rc

fi

NEWSRC=`basename $SOURCE`

${JAVA\_CMD} -jar ${SDP\_JAR\_PARSER} -n ${TMPDIR}/${NEWSRC} $SOURCE

rc=$?

if [ $rc -ne 0 ] ; then

if [ $VERBOSE -gt 0 ] ; then

echo " - KO"

echo "ERROR: Se ha producido un error al analizar el programa"

fi

exit $rc

fi

# Establece el directorio de salida

if [ -z "$OUTDIR" ] ; then

if [ $DLL -eq 1 ] ; then

OUTDIR=$OUTDLL

else

OUTDIR=$OUTEXE

fi

fi

compila $DLL ${TMPDIR}/${SRC} $OUTDIR 1

rc=$?

# Guardar el programa erroneo para su analisis

if [ $rc -ne 0 ] ; then

${JAVA\_CMD} -jar ${SDP\_JAR\_PARSER} -e $src

exit $rc

fi

done

1. Script de configuración del entorno

#!/bin/sh

############################################################

# Utilidad para definir las variables de entorno y

# configurar el sistea

#

# Autor : Javier Gonzalez Grandez

# Fecha : Ago - 2015

# Version: 1.0

############################################################

############################################################

# Variables globales

############################################################

TMPDIR=/tmp

OUTEXE=/PFC/cases/exe

OUTDLL=/PFC/cases/shared

VERBOSE=0

############################################################

# Funciones

############################################################

configure() {

if [ -n "$SDP\_ENV" ] ; then

if [ $FORCE -eq 0 ] ; then

if [ $SILENT -eq 0 ] ; then

echo El entorno ya esta establecido

fi

return

fi

fi

if [ $SILENT -eq 0 ] ; then

echo Estableciendo el entorno de trabajo ...

fi

# Marcar el entorno a establecido

export SDP\_ENV=1

# Directorios de instalacion

export SDP\_ROOT=/PFC

export SDP\_TMP=/tmp

export SDP\_MQ=/opt/mqm

# Ubicacion de los datos y casos de pruebas

export SDP\_CASES=${SDP\_ROOT}/cases

export SDP\_DATA=${SDP\_CASES}/dat

export SDP\_COBOL=${SDP\_ROOT}/cases/cobol

export SDP\_CBL=${SDP\_COBOL}/cbl

export SDP\_CPY=${SDP\_COBOL}/cpy

# Ubicacion de los ejecutables y librerias

export SDP\_EXE=${SDP\_CASES}/exe # Ejecutables COBOL

export SDP\_LIB=${SDP\_ROOT}/lib # Librerias y shared objects

# Configuracion de los JAR

export SDP\_JAR=${SDP\_ROOT}/lib # JARs

export SDP\_JAR\_PARSER=${SDP\_JAR}/SDPParser-3.0.jar

export SDP\_JAR\_TRAPPER=${SDP\_JAR}/SDPTrapper-3.0.jar

export SDP\_JAR\_COLLECTOR=${SDP\_JAR}/SDPCollector-3.0.jar

# Configuracion del entorno GNU COBOL

export COB\_EXT=cbl

export COB\_COPY\_DIR=$SDP\_CPY

export COB\_LIBRARY\_PATH=.:$SDP\_LIB:$SDP\_EXE:$COB\_LIBRARY\_PATH

# Adecuacion del PATH

export PATH=.:$SDP\_ROOT/bin:$SDP\_EXE:$PATH

}

showHelp() {

echo "====================================================="

echo " P.F.C. - Javier Gonzalez "

echo "Establece el entorno para pruebas en el shell actual "

echo " "

echo "Uso: . setenv.sh [opciones] "

echo

echo "Es IMPORTANTE indicar '.' para cargar el shell "

echo

echo "Opciones: "

echo " -f | --force Vuelve a establecer el entorno "

echo " -q | --quiet No muesta mensajes "

echo

echo "====================================================="

echo

}

############################################################

# Code

############################################################

FORCE=0

SILENT=0

HELP=0

OPTS=$(getopt -n "$0" -o fqh --long "force,quiet,help" -- "$@")

if [ $? -ne 0 ]; then

echo Error de parametros

else

eval set -- "$OPTS"

while true;

do

case "$1" in

-h | --help)

HELP=1

shift

;;

-f | --force)

FORCE=1

shift

;;

-q | --quiet)

SILENT=1

shift

;;

--) shift

break

;;

esac

done

if [ $HELP -eq 1 ] ; then

showHelp

else

configure

fi

fi

Generación de la librería SDPTRAPB

#!/bin/sh

############################################################

# Pseudo make para construir la libreria dinamica

#

# HAY QUE INDICAR:

# 1.- SI LA LIBRERIA SERA 32 O 64 BITS

# 2.- SI SE USARA MQCLIENTE O MQSERVER

#

# Autor : Javier Gonzalez Grandez

# Fecha : Ago - 2015

# Version: 1.0

############################################################

SRC=./src

INC=./include

OUT=./dll/SDPTRAPB.so

MQ=/opt/mqm

if [ -z "$SDP\_ENV" ] ; then

echo "El entorno de trabajo no esta establecido"

echo "Ejecute sdpenv.sh"

exit 1

fi

############################################################

# DEFINIR EL TIPO DE LIBRERIA A UTILIZAR

# Comentar/descomentar lo que proceda

############################################################

LIBTYPE=-m64 # Libreria de 64 bits

#LIBTYPE=-m32 # Libreria de 32 bits

LIBMQM=$MQ/lib64 # Libreria de 64 bits

#LIBMQM=$MQ/lib # Libreria de 32 bits

MQCLI=mqm # Utiliza el servidor

#MQCLImqic # Utiliza el cliente

############################################################

# Defines para el proceso de compilacion

############################################################

DEFS=

# Nivel de debug => DEBUG\_LEVEL

DBG\_NONE=0

DBG\_CALL=1

DBG\_PUSH=2

DBG\_POP=4

DBG\_SESION=8

DBG\_MSG=16

DBG\_SEND=32

DBG\_TIMER=64

DBG\_MOD=128

DBG\_ALL=255

#DEFS="$DEFS -DDEBUG\_LEVEL=$DBG\_NONE"

# Tipo de proceso => PROCESS\_MODE

MODE\_GLOBAL=1

MODE\_MODULE=2

MODE\_DETAILED=4

#DEFS="$DEFS -DPROCESS\_MODE=$MODE\_GLOBAL"

# Agrupacion de mensajes ITEMS\_BY\_MSG

#DEFS="$DEFS -DITEMS\_BY\_MSG=numero"

# Tipo de firma digital => HASH\_SHA | HASH\_MD5

#DEFS="$DEFS -DHASH\_SHA"

#DEFS="$DEFS -DHASH\_MD5"

gcc $LIBTYPE -shared -fPIC $DEFS -I$INC -I$MQ/inc $SRC/\*.c -L$LIBMQM -l$MQCLI -o$OUT

cp $OUT /PFC/lib

Script de inicio de MQ

#!/bin/sh

### BEGIN INIT INFO

# Provides: MQ Series Service

# Required-Start: $syslog

# Required-Stop: $syslog

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: MQ Series services

# Description:

#

### END INIT INFO

############################################################

# Utilidad para arrancar el servidor MQ

#

# Autor : Javier Gonzalez Grandez

# Fecha : Ago - 2015

# Version: 1.0

############################################################

startMQ() {

sudo -u mqm /opt/mqm/bin/strmqm SDP.QMGR

}

stopMQ() {

sudo -u mqm /opt/mqm/bin -i SDP.QMGR

}

if [ $# -eq 0 ] ; then

startMQ

else

if [ "$1" == "start" ] ; then

startMQ

fi

if [ "$1" == "stop" ] ; then

stopMQ

fi

fi

Código 7: Script de gestión de MQSeries

1. Programas de prueba

| Prefijo | Grupo | Acción |
| --- | --- | --- |
|  |  |  |
| BAD\* | 01\* | Contiene sentencias no permitidas (GO TO) |
|  |  |  |
| vsftpd |  | Servidor ftp |
|  |  |  |
| gcc |  | Compilador GNU C |
|  |  |  |
| rpm |  | Gestor de paquetes RPM |
|  |  |  |
| Make |  | Utilidad para gestión de compilaciones |
|  |  |  |
| automake |  | Herramienta para la generación de Makefiles conformes con los estándares GNU |
|  |  |  |
| autoconf |  | Herramienta de generación automática de scripts de configuración |
|  |  |  |
| default-jre |  | Open Java Runtime Environment |
|  |  |  |
| default-jdk |  | Open Java Development Toolkit |
|  |  |  |
| open-cobol |  | Compilador Open Cobol |
|  |  |  |
| apache2 |  | Servidor de páginas Apache 2 |
|  |  |  |
| tomcat8 |  | Servidor de aplicaciones Apache Tomcat 8 |
|  |  |  |
| tomcat8-admin |  | Aplicación de administración Web de Apache Tomcat 8 |
|  |  |  |
| tomcat8-user |  | Herramienta para crear instancias de usuarios en Apache Tomcat |
|  |  |  |
| mysql-server |  | Servidor de base de datos MySQL |
|  |  |  |

1. Software de terceras partes

| Prefijo |  | Acción |
| --- | --- | --- |
|  |  |  |
| BAD\* | 01\* | Contiene sentencias no permitidas (GO TO) |
|  |  |  |
| vsftpd |  | Servidor ftp |
|  |  |  |
| gcc |  | Compilador GNU C |
|  |  |  |
| rpm |  | Gestor de paquetes RPM |
|  |  |  |
| Make |  | Utilidad para gestión de compilaciones |
|  |  |  |
| automake |  | Herramienta para la generación de Makefiles conformes con los estándares GNU |
|  |  |  |
| autoconf |  | Herramienta de generación automática de scripts de configuración |
|  |  |  |
| default-jre |  | Open Java Runtime Environment |
|  |  |  |
| default-jdk |  | Open Java Development Toolkit |
|  |  |  |
| open-cobol |  | Compilador Open Cobol |
|  |  |  |
| apache2 |  | Servidor de páginas Apache 2 |
|  |  |  |
| tomcat8 |  | Servidor de aplicaciones Apache Tomcat 8 |
|  |  |  |
| tomcat8-admin |  | Aplicación de administración Web de Apache Tomcat 8 |
|  |  |  |
| tomcat8-user |  | Herramienta para crear instancias de usuarios en Apache Tomcat |
|  |  |  |
| mysql-server |  | Servidor de base de datos MySQL |
|  |  |  |

## Configuración MQ

DEFINE PROCESS (PRC.PARSER) APPLTYPE(UNIX) APPLICID(SDPParser)

DEFINE PROCESS (PRC.TRAPPER) APPLTYPE(UNIX) APPLICID(SDPTrapper)

DEFINE PROCESS (PRC.COLLECTOR) APPLTYPE(UNIX) APPLICID(SDPCollector)

DEFINE QLOCAL (SDP.INIT) PUT(ENABLED) GET(ENABLED) +

NOTRIGGER NOSHARE

DEFINE QLOCAL (SDP.PARSER) +

PUT(ENABLED) GET(ENABLED) INITQ(SDP.INIT) +

MAXDEPTH(999999999) MAXMSGL(4194304) +

TRIGGER TRIGTYPE(FIRST) PROCESS(PRC.PARSER) +

DEFPSIST(YESY) DEFSOPT(SHARED) USAGE(NORMAL)

DEFINE QLOCAL (SDP.TRAPPER) +

PUT(ENABLED) GET(ENABLED) INITQ(SDP.INIT) +

MAXDEPTH(999999999) MAXMSGL(4194304) +

TRIGGER TRIGTYPE(FIRST) PROCESS(PRC.TRAPPER) +

DEFPSIST(YES) DEFSOPT(SHARED) USAGE(NORMAL)

DEFINE QLOCAL (SDP.COLLECTOR) +

PUT(ENABLED) GET(ENABLED) INITQ(SDP.INIT) +

MAXDEPTH(999999999) MAXMSGL(4194304) +

TRIGGER TRIGTYPE(FIRST) PROCESS(PRC.COLLECTOR) +

DEFPSIST(YES) DEFSOPT(SHARED) USAGE(NORMAL)

DEFINE QLOCAL (SDP.INIT) PUT(ENABLED) GET(ENABLED) +

NOTRIGGER NOSHARE