

ESPECIFICACIONES PARA LA INSTALACIÓN Y CONFIGURACION DE UNA APLICACIÓN UTILIZANDO GRADA (GENERADOR RÁPIDO Y AUTOMATIZADO PARA EL DESARROLLO DE APLICACIONES)

Versión: 1.0

Fecha: 24/11/2014

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

1. INTRODUCCIÓN

1.1 Objeto

El presente documento describe como desplegar una aplicación web utilizando GRADA (Generador Rápido y Automatizado para el Desarrollo de Aplicaciones).

1.2 Descripción general

La solución GRADA es un desarrollo que permite disponer de manera rápida de aplicaciones web pequeñas y medias sobre una funcionalidad estándar (mantenimiento de datos básico, gestión de permisos, conservación de datos históricos, búsquedas habituales, etc) acorde con el patrón CRUD. La aplicación es capaz de generar la lógica básica necesaria para una funcionalidad típica en entornos de gestión a partir de un modelo de datos que cumpla unos criterios predefinidos. La aplicación es parametrizable para ampliar esta funcionalidad básica con criterios de filtrado de información por roles, comportamiento asociado a campos, etc.

1.3 Definiciones y Acrónimos

1.3.1 Definiciones

Concepto	Definición

1.3.2 Acrónimos

Acrónimo	Concepto
GRADA	Generador Rápido y Automatizado para el Desarrollo de Aplicaciones
CRUD	Patrón de mantenimiento de datos (Crear, Leer, Actualizar y Borrar)

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

2. CREACIÓN DEL MODELO DE DATOS FÍSICO

2.1 Tablas del metamodelo

El metamodelo se compone de tablas principales y tablas temporales para su regeneración. Las tablas principales son:

1. **md_accion**: Tabla que almacena todas las pantallas del aplicativo
2. **md_alerta**: Tabla con los datos necesarios para la gestión de alertas en la aplicación
3. **md_búsqueda**: Tabla temporal para guardar las búsquedas globales de la aplicación
4. **md_clavelista**: Tabla temporal para guardar los registros consultados por un usuario
5. **md_columna**: Tabla que almacena todas las columnas del metamodelo
6. **md_columna_accion**: Tabla que almacena las acciones que se realizan antes de grabar un dato en la columna correspondiente
7. **md_correo**: Tabla que almacena la información necesaria para el envío de correos
8. **md_estado**: Tabla auxiliar para los estados de tramitación
9. **md_estilo_web**: Tabla para la configuración de determinados parámetros de presentación de la aplicación
10. **md_filtro**: Tabla para la parametrización de que campos se ven en que pantallas. Por defecto se ven todos salvo que estén en esta lista
11. **md_gesproy**: Tabla para configurar la conexión con Redmine
12. **md_log**: Tabla de logs de la aplicación
13. **md_navegacion**: Tabla que recoge la navegación por la aplicación de los distintos usuarios
14. **md_opcion**: Tabla que almacena las opciones de los distintos menús
15. **md_operacion**: Tabla auxiliar con las posibles opciones de actualización de los datos para las acciones sobre columnas
16. **md_permisos**: Tabla con los permisos sobre pantallas de los distintos roles
17. **md_referencia**: Tabla que guarda las claves primarias y claves ajenas del metamodelo
18. **md_referencia_columna**: Tabla que guarda las columnas que intervienen en las claves primarias y claves ajenas del metamodelo
19. **md_rol**: Tabla que almacena los roles de la aplicación
20. **md_tabla**: Tabla que almacena las tablas del metamodelo
21. **md_tipoaccion**: Tabla auxiliar que guarda los posibles tipos de acciones a realizar antes de actualizar los datos sobre una determinada columna

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

22. **md_tipocalculo**: Tabla auxiliar que guarda los posibles tipos de cálculo a realizar antes de actualizar los datos sobre una determinada columna
23. **md_tipolog**: Tabla auxiliar con los distintos tipos de registros del log
24. **md_tipopantalla**: Tabla que recoge las posibles pantallas que se generan automáticamente
25. **md_tipotramite**: Tabla auxiliar para agrupar los flujos de tramitación
26. **md_tramitacion**: Tabla auxiliar que almacena los flujos de tramitación
27. **md_tramitacion_columna**: Tabla auxiliar para establecer que campos se actualizan a la vez que se cambia el estado de un trámite
28. **md_usuario**: Tabla que almacena los usuarios de la aplicación
29. **md_usuariorol**: Tabla que almacena los roles de los usuarios de la aplicación

Y las tablas temporales son:

1. **md_accion_espejo**: Tabla temporal para la regeneración del metamodelo
2. **md_alerta_copia**: Tabla temporal para guardar la información de alertas anterior a la regeneración del metamodelo
3. **md_alerta_espejo**: Tabla temporal para la regeneración del metamodelo
4. **md_columna_accion_copia**: Tabla temporal para guardar la información de acciones sobre columnas anterior a la regeneración del metamodelo
5. **md_columna_accion_espejo**: Tabla temporal para la regeneración del metamodelo
6. **md_columna_copia**: Tabla temporal para guardar la información de columnas anterior a la regeneración del metamodelo
7. **md_columna_espejo**: Tabla temporal para la regeneración del metamodelo
8. **md_filtro_copia**: Tabla temporal para guardar la configuración de los campos que no se ven en determinadas pantallas anterior a la regeneración del metamodelo
9. **md_filtro_espejo**: Tabla temporal para la regeneración del metamodelo
10. **md_opcion_copia**: Tabla temporal que guarda los datos de las opciones de menú anteriores a la regeneración del metamodelo
11. **md_opcion_espejo**: Tabla temporal para la regeneración del metamodelo
12. **md_permisos_copia**: Tabla temporal que guarda los permisos anteriores a la regeneración del metamodelo
13. **md_permisos_espejo**: Tabla temporal para la regeneración del metamodelo
14. **md_referencia_columna_espejo**: Tabla temporal para la regeneración del metamodelo
15. **md_referencia_copia**: Tabla temporal que guarda las claves primarias y ajenas anteriores a la regeneración del metamodelo
16. **md_referencia_espejo**: Tabla temporal para la regeneración del metamodelo
17. **md_rol_espejo**: Tabla temporal para la regeneración del metamodelo
18. **md_tabla_espejo**: Tabla temporal para la regeneración del metamodelo

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

19. **md_tramitacion_columna_copia**: Tabla temporal para guardar los campos que se actualizan a la vez que se cambia el estado de un trámite antes de la regeneración del metamodelo
20. **md_tramitacion_espejo**: Tabla temporal para la regeneración del metamodelo

2.2 Reglas para la definición del modelo de datos

1. En cada tabla debería existir una columna llamada "codigo" que debe ser su clave primaria, aunque no es obligatorio. Si existe, la aplicación genera la lógica automática para generar dicho código secuencialmente.
2. La relación entre tablas será mediante claves ajenas que al menos relacionen el campo de clave primaria anterior con el campo correspondiente en la clave relacionada. Se recomienda que comience por "cod_" y una pequeña descripción de la tabla referenciada (por ejemplo, cod_aud para la relación con el campo "codigo" de la tabla "auditoria").
3. En cada tabla debe existir una columna llamada "descripcion" que la aplicación utiliza para mostrarla en los campos desplegados que automáticamente asocia en las pantallas de alta, consulta, modificación, etc.
4. Las tablas deben tener en el campo COMMENT dos textos separados por un guion. El primero podrá ser uno de los siguientes:
 - PRINCIPAL: Indica que la tabla tiene un mantenimiento completo y no es una tabla auxiliar ni intermedia. Aparecerá en el menú inicial
 - AUXILIAR: Es una tabla auxiliar (provincia, localidad, etc)
 - INTERMEDIA: Es una tabla de detalle de la principal
 - NO GESTIONABLE: No habrá ninguna pantalla donde se traten sus datos, tendrá que ser directamente en base de datos

El segundo será la descripción de la tabla y se utilizará en todas las pantallas para darle sentido semántico, utilizándolo en las cabeceras de las pantallas.

Ejemplo:

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

Básico Opciones Índices Llaves foráneas Código CREATE Código ALTER

Nombre: activo
 Comentario: PRINCIPAL - Activos

Columnas: + Agregar - Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterm...
1	codigo	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Sin valor
2	descripcion	VARCHAR	100	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Sin valor
3	des_extensa	VARCHAR	500	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Sin valor
4	cod_est	INT	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor
5	cod_sit	INT	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor

5. A su vez, las columnas deben tener en su campo COMMENT la descripción que se mostrará como literal en todas las pantallas.

Nombre: activo
 Comentario: PRINCIPAL - Activos

Columnas: + Agregar - Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado	Comentario
1	codigo	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...	Codigo Activo
2	descripcion	VARCHAR	100	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Sin valor predeter...	Nombre Activo
3	des_extensa	VARCHAR	500	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Sin valor predeter...	Descripcion Activo
4	cod_est	INT	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...	Estado
5	cod_sit	INT	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...	Situacion Activo
6	cod_tac	INT	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...	Tipo Activo
7	cod_org	INT	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...	Organo Responsable

6. Toda tabla que sea "principal" debe tener un campo "activo" para la baja lógica, es decir, la aplicación nunca borra físicamente el registro sino que utiliza este campo para pasar de "1" a "0". La aplicación no permite la baja física salvo para las tablas "intermedias" que no tengan un campo "activo".

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

7. Las tablas "principales" tendrán un campo que comenzará por "uid_" y con una pequeña descripción de la tabla (por ejemplo "uid_act" para el usuario de la tabla "activo") para registrar el usuario que ha creado el registro.
8. También las tablas "principales" tendrán un campo "roles" que incluirán los roles del usuario que ha creado el registro separados por ";". Con esta información las consultas filtrarán los datos para aquellos usuarios que tengan alguno de los roles del registro que buscan.
9. La aplicación permite la gestión de ficheros en las tablas. Para ello hay que crear dos campos: "nombrefichero" y "fichero". El primero debe ser VARCHAR y sirve para guardar el nombre del fichero y el segundo debe ser LONGBLOB y almacena el fichero físico. Si existen estos campos, la aplicación se comporta de manera que aparecen las opciones de carga del fichero, descarga si existe y limpieza (borrado) del fichero.
10. También permite una tramitación sencilla. Para ello hay que utilizar las tablas md_tipotramite, md_tramitacion y md_tramitacion_columna. La primera es para agrupar un conjunto de trámites (por ejemplo la lógica de tramitación puede variar por años, tipo de subvención, etc). La segunda recoge el tipo de trámite, el estado inicial y final de la tramitación. Y la tercera especifica qué columnas son modificables en función de en qué fase de la tramitación estemos. Además se puede indicar el orden en que aparecerán los campos de manera diferente al del resto de pantallas de la aplicación. La tabla que contenga los expedientes a tramitar deberá tener los campos "tramite" y "estado" y una clave ajena con las tablas md_tipotramite y md_estado. Además debe existir una tabla con el nombre igual a la tabla que tramita los expedientes cuyo nombre termine por "_t" y que su clave primaria este formada por el campo "codigo" y un nuevo campo llamado "orden". Debe tener además un campo llamado "fecha". Por ejemplo: auditoria.codigo = auditoria_t.codigo. También tendrán un campo "uid_mod" para guardar el usuario que realiza la tramitación.
11. Si se quiere una gestión de históricos hay que crear una tabla con el mismo nombre pero que finalice en "_h", y de manera similar a las tablas que finalizan "_t" tengan un campo "orden", "fecha", "uid_mod", se modifique la clave primaria de la misma forma y exista una relación con la tabla origen a partir de los campos "codigo" de ambas tablas. Por ejemplo: auditoria.codigo = auditoria_h.codigo.

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

2.3 Carga de las tablas del metamodelo inicial

Una vez creado el modelo de datos, se deben lanzar los ficheros Tablas Metadatos.sql y Datos Metadatos.sql que se incluyen en el código fuente, los cuales, crean las tablas básicas del metamodelo y cargan los datos mínimos necesarios para acceder a la aplicación y poder lanzar el proceso inicial que regenera el metamodelo con la información de tablas, columnas, etc del modelo de datos que hayamos definido.

2.4 Instalación

1. Creación de la instancia de base de datos resultante de la creación del modelo de datos propio del aplicativo con las tablas por defecto previas.
2. Creación del contenedor en el servidor TOMCAT que en su fichero context.xml debe tener parametrizados dos datasource, uno para acceder al esquema information_schema y otro al esquema de la aplicación, que deben coincidir con los datasources especificados en el profile correspondiente en el pom.xml asociado al proyecto a la hora de la generación del artefacto.
3. Una vez generado el artefacto (fichero .war) para el perfil correspondiente se despliega en el TOMCAT de la forma habitual.
4. Para acceder al aplicativo previamente hay que sustituir en la tabla md_usuario el valor "usuario" o "admin" por el uid de un usuario existente en el LDAP de la organización. La validación será introduciendo dicho usuario y la contraseña del LDAP en la pantalla de login correspondiente. Una vez validado en la aplicación debemos seleccionar el menú de administración para lanzar la opción "Recarga de Metadatos":

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014



Menu Principal	Menu de Tablas Auxiliares	Menu de Tablas de Metadatos	Menu de Administracion	Menu de Vistas
** Configuracion Correo **				
Alta de Configuracion Correo		Consulta de Configuracion Correo		Consulta total de Configuracion Correo
** Estilo Web Metadatos **				
Alta de Estilo Web Metadatos		Consulta de Estilo Web Metadatos		Consulta total de Estilo Web Metadatos
** Log Metadatos **				
Consulta de Log Metadatos		Consulta total de Log Metadatos		
** Datos de Navegación Metadatos **				
Consulta de Datos de Navegación Metadatos		Consulta total de Datos de Navegación Metadatos		
** Usuarios Metadatos **				
Alta de Usuarios Metadatos		Consulta de Usuarios Metadatos		Consulta total de Usuarios Metadatos

Recarga de Metadatos		Recarga de Acciones		Recarga de Vistas
Volcado de Tablas Espejo		Consulta LDAP		Envio de correo interno

Al lanzar esta opción, se cargan en las tablas del metamodelo la información que exista en el esquema information_schema de MySQL correspondiente al esquema de la aplicación y a partir de ese momento toda la lógica básica se carga en el aplicativo.

2.5 Reglas para completar la configuración de la aplicación

1. En **md_estilo_web** se configuran distintos parámetros de la aplicación:

- "¿Se puede consultar el LDAP?"
- "Nº Filas del listado"
- "Maximo nº registros consultables"
- "Nº Registros para filtrado"
- "Maximo nº registros detalle"
- "Nº filas del menu"
- "Nº megas tamaño fichero maximo"
- "Longitud si el valor es nulo"
- "Maxima longitud"
- "Longitud visible"
- "Longitud visible corta"
- "¿Calcular Totales?"
- "Maximo nº registros buscados"
- "Campos a realizar busqueda"
- "Banner inicial"

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

2. En **md_accion** se puede configurar si una pantalla para una determinada tabla es visible o no.
3. En **md_alerta** se puede configurar si aparece una alerta de que un campo de una tabla ha superado un determinado valor (días, etc)
4. En **md_columna** se puede configurar si un campo de una tabla es visible, el orden en que se muestra, si es obligatorio o no (aunque siempre dependerá si en la tabla física lo es), si se calcula el total, el máximo o el mínimo sobre esta columna, si es insertable, es decir, si en las pantallas de alta se graba, si es modificable, es decir, en las pantallas de modificación se puede actualizar, si es un campo que activa un filtro, y si hay un filtro para que valor se muestra (ejemplo: en la tabla activo y el campo cod_tac –tipo de activo- el valor filtro tendría el valor "cod_tac=1" que es para los tipos de activos "aplicación")
5. En **md_columna_accion** se puede configurar si cuando se añade, modifica o siempre una columna, se graba un valor constante, o se calcula un máximo, o se graba el valor de una columna de una tabla relacionada, etc.
6. En **md_correo** se puede tener distintos valores de configuración de correo
7. En **md_filtro** se introducirán las columnas que no queremos que se muestren en la pantalla que elijamos (ejemplo: para la pantalla de consulta de la tabla "activos" podemos introducir la columna "observaciones" de tal forma que dejaría de mostrarse)
8. En **md_opcion** introduciremos el menú (Menu principal, Menu de Tablas Auxiliares, Menu de Tablas Metadatos, Menú Administración y Menú de Vistas) y las distintas pantallas que se muestran
9. En **md_permisos** se introducirán los roles y las acciones permitidas (consulta de tablas, grabar datos, etc)
10. En **md_referencia** existe el campo "filtro" en el que se especificará el criterio para que se muestre la tabla relacionada por la clave ajena (ejemplo: para la clave ajena "FK_caracteristica_activo" que relaciona la tabla "Activos" con la tabla "Características del activo" el valor de filtro es "cod_tac in (1,6)" que significa que

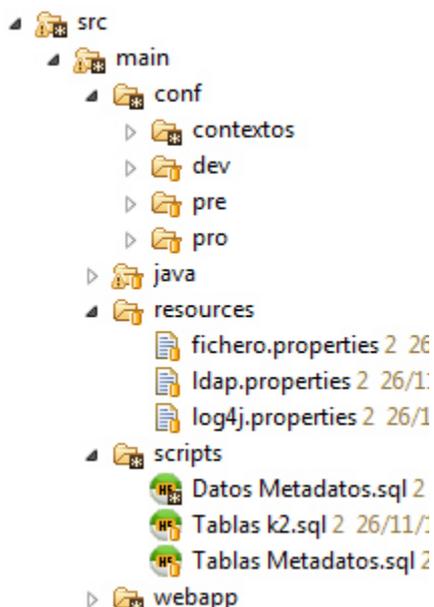
	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

para el campo "Tipo de activo" -cod_tac- los valores 1 -Aplicación- y 6 -Base de Datos- se muestra la tabla en la pantalla donde se muestran los activos y todas sus tablas relacionadas, pero no para el resto de valores de "cod_tac").

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

ANEXO. Generación fichero .war

La estructura del proyecto es la siguiente:



Todo el código cuelga de la carpeta "src/main" que a su vez tiene las siguientes carpetas:

- **conf**

Esta carpeta contiene todos los datos de configuración que requieren ficheros específicos por entorno, etc. A su vez contiene las carpetas:

- contextos: En el caso de que en Tomcat se parametrize que el fichero context.xml se localice en la carpeta META-INF de la aplicación aquí se recogería por cada aplicación una carpeta con el nombre de la aplicación.
- dev: Los distintos ficheros web.xml del entorno de desarrollo
- pre: Idem del entorno de pre-producción
- pro: Idem del entorno de producción

- **java**

Esta carpeta contiene todo el código de servidor de la aplicación.

- **resources**

	ESPECIFICACIONES PARA INSTALACIÓN DE APLICACIONES UTILIZANDO "GRADA"	CONFIDENCIAL	
		Código:	
		Versión:	1.0
		Fecha:	24/11/2014

Ficheros de configuración comunes y parametrizables para todas las aplicaciones

- **scripts**

Esta carpeta contiene los scripts de base de datos del metamodelo y de las distintas aplicaciones que se vayan desplegando

- **webapp**

Esta carpeta contiene todo el código de presentación de la aplicación

Como la aplicación está mavenizada, la generación del fichero .war se realiza adaptando las propiedades del perfil correspondiente definidos en el fichero pom.xml para el aplicativo concreto y lanzando el comando:

mvn clean package -P <Perfil aplicación>

o bien si utilizamos el IDE Eclipse lanzando el goal "clean package -P <Perfil aplicación>".