

Universidad del Valle de Guatemala

Facultad de Ingeniería



Diseño, desarrollo y publicación de un sistema integrado de control de miembros para equipos SIFE

Trabajo académico presentado por Donald Fernando Leiva Arévalo  
para optar al grado académico de Licenciado en Ingeniería  
en Ciencias de la Computación

Guatemala  
2008



Universidad del Valle de Guatemala

Facultad de Ingeniería



Diseño, desarrollo y publicación de un sistema integrado de control de miembros para equipos SIFE

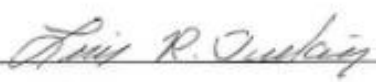
Trabajo académico presentado por Donald Fernando Leiva Arévalo  
para optar al grado académico de Licenciado en Ingeniería  
en Ciencias de la Computación

Guatemala  
2008

Vo. Bo.:

(f)   
Ing. Luis Roberto Furlán Collver

Tribunal Examinador:

(f)   
Ing. Luis Roberto Furlán Collver

(f)    
Ing. Sergio Izquierdo Bloemen

(f)    
Ing. Celso Fernando Cerezo Bregni

Fecha de aprobación: 8 de diciembre de 2008

## **PREFACIO**

SIFE es una red mundial de estudiantes, académicos y empresarios que promueven proyectos orientados a la libre empresa para mejorar la calidad de vida de sus comunidades. En la Universidad del Valle de Guatemala, un equipo SIFE empezó a funcionar en el 2004 aunque no duró más de un año. No fue hasta el 2006 cuando esta iniciativa se revivió. En este segundo intento se han tratado de corregir varios de los errores de organización cometidos anteriormente, cuidando de mantener control entre la asociación para asegurar que las operaciones del día al día estén en línea con los objetivos globales. Resultado de este control, el equipo ha logrado alcanzar estabilidad, obteniendo reconocimiento internacional.

La tecnología ha desempeñado un papel muy importante en todo esto, sobre todo por medio de sistemas computarizados que han ayudado a en las labores diarias de los integrantes de SIFE UVG. En este trabajo se presentan los fundamentos para la elaboración de un sistema computarizado de control de miembros y actividades para SIFE UVG. Entre los planes a corto plazo para este sistema está liberarlo para que sea usado por otros equipos SIFE alrededor del mundo.

# ÍNDICE

Prefacio .....	v
Índice.....	vi
Índice de figuras .....	viii
Resumen.....	ix
I.    Introducción.....	1
A.    Antecedentes .....	1
B.    Justificación .....	2
C.    Objetivos.....	2
D.    Marco teórico .....	2
II.   La entidad SIFE .....	4
A.    La organización SIFE .....	4
B.    Organización de SIFE UVG .....	5
III.  Programación orientada a objetos.....	7
A.    Modelo cliente-servidor.....	7
B.    Uso de AJAX para aplicaciones en internet .....	8
C.    Sistemas de administración de contenidos (CMS).....	10
D.    Programación de componentes en Joomla! .....	10
IV.  Diseño y desarrollo de la propuesta .....	12
A.    Análisis general.....	12
B.    Casos de uso.....	14

C.	Diseño de base de datos.....	16
D.	Diseño de componente .....	17
E.	Características del sistema .....	18
V.	Resultados y conclusiones .....	26
VI.	Recomendaciones.....	27
VII.	Bibliografía .....	29
VIII.	Apéndices .....	31
A.	Herramientas y tecnologías usadas .....	31
IX.	Anexos.....	33
A.	Modelo original de crónica (formulario en papel) .....	33
B.	Código fuente en el lado del servidor.....	35

## ÍNDICE DE FIGURAS

Figura 1. Pilares SIFE y su relación con los proyectos.....	4
Figura 2. Componentes para la sustentabilidad de un equipo SIFE. ....	5
Figura 3. Organización de equipo SIFE UVG.....	6
Figura 4. Modelo cliente-servidor.....	7
Figura 5. Interacción asincrónica de una aplicación AJAX. ....	8
Figura 6. Esquema de funcionamiento de un CMS. ....	10
Figura 7. Sitio de internet de SIFE UVG ( <a href="http://www.sifeuvg.org.gt">http://www.sifeuvg.org.gt</a> ).....	13
Figura 8. Caso de uso para director de proyecto. ....	14
Figura 9. Caso de uso para miembro de proyecto. ....	15
Figura 10. Caso de uso para recursos humanos.....	15
Figura 11. Caso de uso para junta directiva. ....	16
Figura 12. Diagrama de entidad-relación.....	17
Figura 13. Opción de <i>Nueva</i> en la versión beta.....	19
Figura 14. Opción de <i>Listar</i> en el sistema beta. ....	20
Figura 15. Opción de <i>Reportes</i> en el sistema beta.....	21
Figura 16. Opción de <i>Buscar</i> en el sistema beta. ....	22
Figura 17. Sistema con varias pestañas abiertas.....	23
Figura 18. Ejemplo de opciones de ordenamiento y filtrado de proyectos.....	24
Figura 19. Ingreso de datos para la opción de definición de proyectos.....	25

## **RESUMEN**

El enfoque de este trabajo es el diseño e implementación de un sistema integrado que pueda ser usado por equipos SIFE alrededor del mundo. El sistema está enfocado a: 1) administración de proyectos, 2) registro de actividades y 3) control de miembros. Este sistema fue desarrollado como una aplicación de internet para que 1) pueda ser usado en cualquiera computadora independientemente del sistema operativo que tenga y 2) evitar tener que instalar aplicaciones adicionales que requieran actualización. En este trabajo también se proporciona una introducción a lo que es SIFE, su organización, las tecnologías y herramientas usadas y características del sistema.

# I. INTRODUCCIÓN

## A. Antecedentes

El equipo SIFE de la Universidad del Valle de Guatemala se inició en el año 2004, el mismo año en el que SIFE Guatemala inició operaciones. Aunque contaba con estudiantes destacados y motivados, la falta de organización y control de actividades hizo que el grupo se disolviera. No fue hasta finales del año 2006 cuando hubo un segundo intento por formar un equipo SIFE en la Universidad del Valle. Este nuevo equipo también se enfrentó a problemas de organización y control de actividades. La solución propuesta fue llevar un sistema de registros llamado “crónicas”.

Este sistema consiste en apuntar hora, fecha, lugar, asistencia y detalles relevantes de cada reunión y actividad llevada a cabo por los miembros del equipo. Todo esto se hace por medio de papel y lápiz. El proceso ha sido recibido con bastante resistencia, siendo una de las quejas principales lo tedioso que resulta llenar los formularios a mano. Consecuencia de esto, muchas actividades no llegan a registrarse.

Otro problema es contabilizar datos importantes, como el total de horas trabajadas en un proyecto o por un miembro particular. Para poder calcular estos datos, es necesario hacer una revisión manual de todos los formularios. Aunque esto se hace con el apoyo de una hoja de cálculo, trasladar toda la información del papel a la computadora y estarlo actualizando constantemente con nueva información es una tarea que consume bastante tiempo.

La falta de un estándar en estos formularios también ha hecho que algunos datos se ingresen en formatos diferentes, como por ejemplo la fecha, haciendo que sea mucho más fácil equivocarse a la hora de trasladar esta información a una computadora. Un problema relacionado es la omisión de datos importantes. Por ser un sistema en papel, dichas omisiones no se detectan hasta después de que los formularios han sido entregados.

SIFE UVG también carece de control de miembros. El departamento de recursos humanos debe actualizar su base de datos cada vez que lo juzga necesario por lo que muchas veces los miembros del equipo, incluyendo la junta directiva, no están al tanto de los miembros de nuevo ingreso, cambios de los integrantes de un proyecto o personas que se han retirado del equipo. Tampoco se lleva control de los planes y objetivos propuestos y cumplidos en cada proyecto.

En resumen, no hay un sistema de control estandarizado ni centralizado, por lo que resulta difícil conocer el estado actual del equipo en un momento determinado a pesar de

que esta información es de importancia para la junta directiva, asesores, casa matriz SIFE y patrocinadores potenciales. Este problema también lo sufren equipos de SIFE Centroamérica, específicamente equipos de SIFE El Salvador y SIFE Nicaragua.

## **B. Justificación**

La justificación del proyecto se fundamenta en poder contar con una herramienta tecnológica accesible, fácil de usar y con una licencia que se adapte a las necesidades de SIFE UVG.

## **C. Objetivos**

### **1. General**

Diseñar, desarrollar y publicar un sistema de control de miembros, proyectos y actividades que sea de utilidad tanto para el equipo SIFE UVG como para equipos SIFE alrededor del mundo.

### **2. Específicos**

- Proveer un registro histórico de actividades y miembros de equipos SIFE que permita determinar si las operaciones del día a día están en línea con la visión y misión del equipo.
- Asistir en la evaluación de productividad de los miembros de los equipos SIFE.
- Asistir en la determinación del éxito de un proyecto SIFE.
- Publicar la herramienta en el sitio de Internet de SIFE UVG para que esté a disposición de los equipos SIFE.
- Crear una comunidad a partir de esta herramienta para fomentar la unidad y competencia cooperativa (*coopetencia*) entre equipos SIFE.

## **D. Marco teórico**

Este proyecto se fundamenta en los conocimientos adquiridos a lo largo de la licenciatura en Ingeniería en Ciencias de la Computación. Las áreas más relevantes necesarias para comprender en su totalidad este proyecto son:

- **Programación orientada a objetos:** Utilizado para la implementación del programa de software (código) así como el uso de herramientas basadas en diseño de patrones.
- **Base de datos:** Utilizado para almacenar toda la información por medio de un diseño y modelo basado en operaciones básicas (CRUD) y uso óptimo de la información (normalización).
- **Interacción humano-computador:** Utilizado para crear una interfaz gráfica fácil y cómoda de usar para el usuario final.
- **Análisis de sistemas:** Utilizado para analizar y tomar los requerimientos del sistema.
- **Ingeniería de software:** Utilizado para implementar el proyecto.

- **Administración de proyectos:** Utilizado para planificar y ejecutar el proyecto, así como diseñar partes del sistema basados en los fundamentos de la administración de proyectos.

## II. LA ENTIDAD SIFE

### A. La organización SIFE

SIFE es una red internacional de estudiantes, empresarios y académicos que promueven el desarrollo humano a través de la libre empresa. Para esto se organizan equipos de estudiantes que diseñan y desarrollan proyectos que ayuden a miembros de sus comunidades.

Estos proyectos se concentran en seis áreas, llamados pilares SIFE: economía de mercado, habilidades de éxito, emprendimiento, alfabetización financiera, ética en los negocios y sustentabilidad. Los objetivos de SIFE son que los miembros de las comunidades, como aspirantes a empresarios, familias de escasos recursos, personas marginadas y niños puedan aprender sobre estas seis áreas, continuar con los proyectos propuestos por estudiantes SIFE y mejorar su calidad de vida.



Figura 1. Pilares SIFE y su relación con los proyectos.

Cada año los equipos SIFE presentan los resultados de sus proyectos a nivel regional, nacional y mundial. Un panel de jueces, integrado por empresarios y ejecutivos, seleccionan a los equipos que consideran los mejores dentro de cada país para competir a nivel mundial. La selección se basa en los criterios mencionados anteriormente. También se debe entregar un reporte anual con un resumen de todos los resultados obtenidos durante año.

#### 1. Organización de equipos SIFE

Los equipos SIFE se organizan individualmente a nivel universitario. Cada campus universitario puede tener como máximo un equipo SIFE. Cada equipo SIFE puede organizarse en grupos más pequeños para llenar cupo en departamentos o proyectos dentro del mismo equipo. Aparte de requerir un representante, la organización interna queda a juicio del mismo equipo.

Para poder pertenecer a un equipo SIFE de una universidad, el único requisito es ser estudiante registrado de dicha casa de estudios. Puede ser un estudiante de licenciatura o posgrado y de cualquier carrera. Un profesor universitario puede pertenecer al equipo como asesor de este. También puede ser asesor de un proyecto específico.

La sustentabilidad del equipo es un factor importante dentro de los equipos SIFE. La casa matriz de SIFE recomienda considerar y evalúa los siguientes aspectos dentro de un equipo SIFE:



Figura 2. Componentes para la sustentabilidad de un equipo SIFE.

- **Reclutamiento y organización del equipo:** Se refiere a la estructura organizacional del equipo, forma de organizar reclutamiento e integración de personal.
- **Plan de sucesión:** Se refiere a los planes existentes para seleccionar a la siguiente junta directiva y las acciones y curso a seguir por esta misma.
- **Soporte administrativo:** Se refiere al apoyo que recibe el equipo por parte de su casa de estudios.
- **Junta de asesores:** Se refiere al grupo de asesores que supervisan al equipo y sus proyectos.
- **Alumni:** Se refiere al grupo de antiguos miembros del equipo y el apoyo que estos brindan.
- **Fondos:** Se refiere al capital disponible y fuentes de ingresos para la ejecución de proyectos y operaciones.
- **Medios:** Se refiere a la cobertura de medios de comunicación como periódicos, radio, televisión e internet.

## B. Organización de SIFE UVG

La junta directiva del equipo SIFE UVG se compone de:

- **Presidente:** El presidente se encarga de la administración general de SIFE UVG. También es responsable de representar al equipo en actividades externas y de mantener la visión y misión de la asociación.

- **Vicepresidente:** Se encarga de administrar actividades específicas de los departamentos y proyectos, así como mantener la moral del equipo. Es el responsable principal de que exista armonía en el equipo.
- **Secretario:** Se encarga de organizar actividades y reuniones importantes, tomar nota de estas y dar comunicados importantes al resto del equipo.
- **Tesorero:** Se encarga de administrar los ingresos y egresos financieros del equipo.
- **Departamentos:**
  - **Director de Recursos Humanos:** Se encarga de mantener al día un listado de todos los miembros, asignar puestos dentro del equipo según capacidades, evaluación de personal y reclutamiento.
  - **Director de Tecnologías de la Información y Comunicación:** Se encarga de administrar toda la información del equipo, velar por el buen uso de esta y supervisar el sitio de internet, sistemas informáticos y material audiovisual para presentaciones.
- **Proyectos:**
  - **Directores de proyectos:** Cada proyecto tiene un director que se encarga de la planificación, ejecución y evaluación de este.

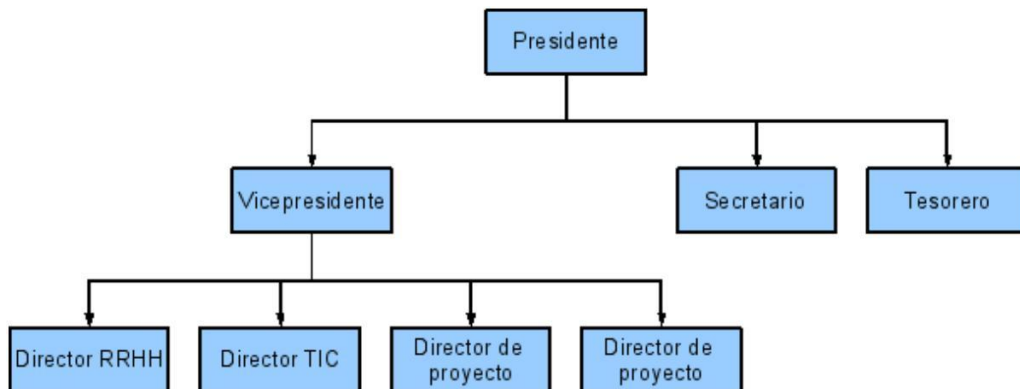


Figura 3. Organización de equipo SIFE UVG.

### III. PROGRAMACIÓN ORIENTADA A OBJETOS

#### A. Modelo cliente-servidor

El modelo cliente servidor consiste en una arquitectura donde existe un programa cliente y un programa servidor que se comunican entre sí.

Entre las características de un cliente están:

- Envía solicitudes y peticiones.
- Espera respuesta del servidor.
- Interactúa directamente con el usuario a través de una interfaz gráfica.

Un servidor tiene las siguientes características:

- Está constantemente a la espera de peticiones.
- Recibe peticiones de uno o varios clientes.
- Generalmente efectúa cálculos y almacena toda la información de la aplicación.

El programa cliente realiza peticiones al programa servidor, el cual proporciona una respuesta. La separación entre cliente y servidor es una separación de tipo lógica donde el programa cliente no necesariamente se limita a una sola máquina o programa.

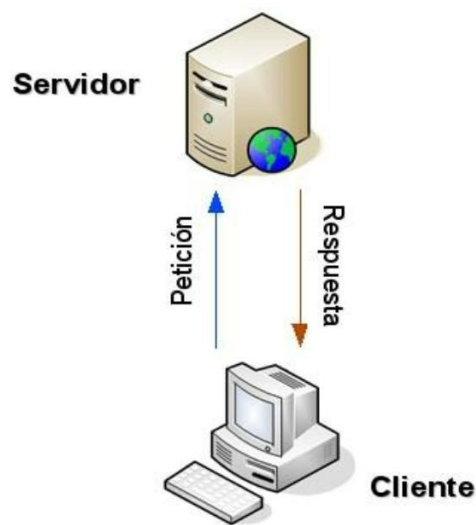


Figura 4. Modelo cliente-servidor.

## B. Uso de AJAX para aplicaciones en internet

AJAX (abreviación de “asynchronous Javascript and XML”) es un grupo de técnicas orientadas a la programación y desarrollo de aplicaciones de internet que poseen dos características esenciales: asincronía y actualización selectiva.

La asincronía se refiere a la capacidad de una aplicación AJAX de poder mantener una comunicación libre de restricciones temporales entre cliente y servidor. Esto se logra introduciendo una capa intermedia que sirve como mediador entre ambos. A esta capa usualmente se le denomina motor AJAX y se encarga tanto de procesar las acciones del usuario como de dar una respuesta apropiada a cada acción. El motor AJAX básicamente se encarga de abrir un nuevo canal de comunicación con el servidor cada vez que una acción lo requiera.

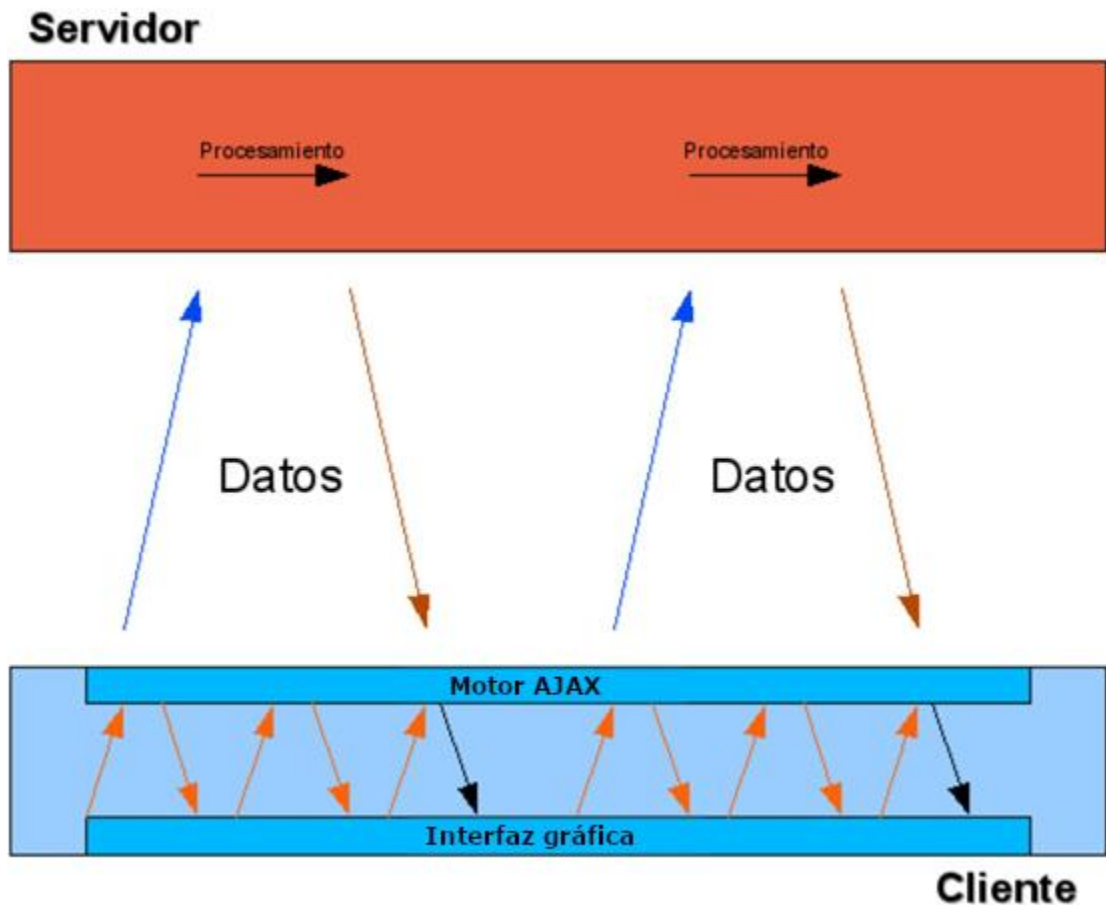


Figura 5. Interacción asincrónica de una aplicación AJAX.

Cuando el motor envía una petición, este no se dedica a esperar la respuesta del servidor sino más bien se desentiende del proceso. El navegador, por su parte, al recibir respuesta

del servidor la transfiere al motor para que este lo procese. Esto permite que el cliente pueda realizar varias peticiones seguidas sin tener que esperar respuesta de la primera petición.

La segunda característica de AJAX es la actualización selectiva, que se refiere a la capacidad del motor para agregar, modificar o eliminar elementos de una página de internet sin necesidad de cargarla nuevamente. Esto permite abstraer las características de varias páginas y juntarlas en una sola, actualizando solamente los elementos que sean necesarios.

Aparte de estas características, al ser un conjunto de técnicas, AJAX se puede dividir en componentes enfocados a las siguientes áreas: actualización, comunicación, integración, formato de datos y visualización.

La actualización se refiere a las técnicas utilizadas para modificar el contenido de una página de internet de manera dinámica. Está estrechamente relacionado a la característica de actualización selectiva. Actualmente la única tecnología que brinda dicha funcionalidad es el modelo de objetos de documentos mejor conocido como DOM. El DOM es una especificación de una interfaz por el consorcio W3C que se utiliza para representar archivos de HTML o XML como una estructura tipo árbol en la que cada elemento del documento constituye un nodo con sus respectivas propiedades. El DOM también proporciona métodos para realizar operaciones sobre los nodos. Los navegadores de internet construyen un árbol DOM antes de desplegar una página.

Para la comunicación originalmente se usaba el objeto XMLHttpRequest, parte del API del DOM, que permite transferir XML entre un navegador y un servidor web. Aunque este objeto sigue siendo el más usado, hay algunas alternativas como los IFrames y el objeto XMLHttpRequest.

La integración de una aplicación AJAX es muy importante para poder coordinar las tecnologías que los conforman y obtener los resultados deseados. El lenguaje utilizado normalmente para integración es JavaScript. JavaScript es un lenguaje interpretado orientado a objetos que se ejecuta del lado del cliente. Existen otras alternativas como Jscript y VBScript. Estos lenguajes proveen métodos y objetos para controlar distintos aspectos del navegador incluyendo su representación DOM interna.

El formato de los datos se refiere a la estructura que deben tener los datos para que tanto el cliente como el servidor se puedan comunicar. Se trata de un estándar de comunicación para transferencia de datos. Comúnmente este papel lo desempeña XML en las aplicaciones AJAX. Sin embargo, también se pueden utilizar otros formatos como CSV, JSON o incluso texto plano.

Por último, las tecnologías de visualización son empleadas para definir el formato visual de la interfaz gráfica. Las aplicaciones AJAX utilizan únicamente el HTML y el CSS. Aunque originalmente se utilizaba sólo el HTML para el formato visual, actualmente

se utiliza este junto con archivos externos de CSS que definen y centralizan la parte visual, permitiendo hacer cambios rápidos de estilo.

### C. Sistemas de administración de contenidos (CMS)

Un sistema de administración de contenidos es una aplicación utilizada para crear, actualizar, publicar y en general administrar contenido de una manera consistente. Los CMS se utilizan frecuentemente para sistematizar la gestión de contenido en una industria específica. En las aplicaciones de internet, los CMS generalmente se utilizan para la administración de contenidos estáticos y manejo de usuarios. Varios CMS como Joomla! tienen un API que permite extender su funcionalidad.

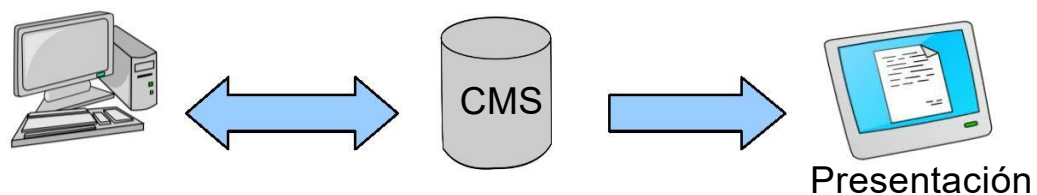


Figura 6. Esquema de funcionamiento de un CMS.

### D. Programación de componentes en Joomla!

El CMS Joomla! se puede descomponer en tres partes: *plugins*, módulos y componentes. Los *plugins* son pequeños fragmentos de código que realizan tareas antes de mostrar el contenido como por ejemplo filtrar palabras. Los módulos también son pequeños fragmentos de código, pero orientados a la presentación de datos. Por último están los componentes proveen la funcionalidad y presentación principal de una página. Joomla! permite ejecutar varios módulos y *plugins* por petición, pero sólo permite cargar un componente por petición.

Un componente en Joomla! usualmente se divide en dos partes: pública y administrador. Joomla! No obliga a hacer esta división, por lo que un componente puede tener funciones administrativas en la parte pública o simplemente tener un componente de uso exclusivo para la administración. Independientemente de esto, un componente de Joomla!, ya sea en la parte pública o administrativa, se compone de tres capas: clases, lógica y presentación.

Las clases modelan objetos del componente, guardando una correspondencia con las tablas de la base de datos. Por lo tanto, las propiedades de una clase en Joomla! se definen haciendo una correspondencia uno a uno con sus respectivos campos a una tabla en la base de datos. Par esto deben tener los mismos nombres. Además, la clase debe extenderse de una clase base de Joomla! llamada *mosDBTable*. Está clase provee métodos genéricos para el ingreso, actualización y eliminación de datos. En el constructor deben indicarse el nombre de la tabla con la que se hace la correspondencia y la llave primaria de la misma. Los métodos de la clase permiten agregar funcionalidad como validaciones. Por motivos de compatibilidad con versiones anteriores, las clases que utiliza Joomla! son las de PHP4.

La capa lógica de Joomla! indica el curso a seguir a partir de los eventos que este genera. Los eventos se basan en las peticiones del cliente. Es en esta capa donde generalmente se accede a la base de datos usando la clase *mosDatabase* además de realizar cálculos e implementar algoritmos con los datos. La lógica también se encarga de conectar las clases con la capa de presentación.

La capa de presentación muestra los resultados de las operaciones y las regresa al cliente para ser desplegadas. Generalmente la presentación genera HTML a partir de los datos proporcionados por la capa de lógica, aunque se puede usar para otros propósitos. La capa de presentación generalmente se implementa como una clase con métodos estáticos por lo que al ser llamada desde la capa de lógica no es necesario instanciarla.

## **IV. DISEÑO Y DESARROLLO DE LA PROPUESTA**

### **A. Análisis general**

El equipo SIFE UVG cuenta con un sistema de registro de actividades. Este registro se lleva totalmente a mano, llenando formularios (crónicas) donde se registra:

- Lugar
- Fecha y hora
- Nombre del proyecto
- Tipo de actividad
- Temas a tratar (agenda)
- Planes para la siguiente reunión

Adicionalmente, el encargado de apuntar estos datos también debe tomar asistencia y el tiempo que tomó realizar la actividad. Estos últimos datos son importantes para contar el total de horas trabajadas por cada miembro en cada proyecto. Esta información se incluye en el reporte anual.

Los encargados de ingresar esta información generalmente son los directores de proyecto. Sin embargo, en ausencia del director, esta labor queda a cargo de otro integrante del equipo. Un director de proyecto sólo debe poder consultar crónicas de su proyecto. Miembros de la junta directiva como el presidente y vicepresidente deben ser capaces de consultar información de todos los proyectos.

El departamento de recursos humanos mantiene una hoja de cálculo con un listado de los miembros del equipo y sus datos de contacto. Este listado se actualiza cada vez que el director lo juzgue necesario por lo que no siempre está al día. Los datos que incluye esta lista son:

- Nombre completo
- Carrera
- Proyecto/Departamento al que pertenece
- Puesto
- Teléfono
- Correo electrónico

Al departamento de recursos humanos también le interesa saber quiénes son los miembros más productivos y activos. Sin embargo, para obtener estos datos se deben revisar todas las crónicas. Por lo tanto, recursos humanos tiene acceso a todas las crónicas, aunque en principio sólo debería poder consultar las de su departamento.

Un sistema computarizado permite resolver varios de estos problemas.

Primero, con el ingreso de datos, se pueden hacer los cálculos correspondientes sin necesidad de tener que consultar cada crónica individualmente de forma manual, ahorrando tiempo que se puede invertir en otras cosas. También permite un control más estricto de los roles de cada integrante, limitando las operaciones que cada uno puede realizar sobre las crónicas. Esto es sobre todo importante para el caso de recursos humanos, ya que le permitiría acceder a las métricas de desempeño sin tener que consultar otra información que no le corresponde.

Un sistema computarizado también permite centralizar la información con la ayuda de un manejador de base de datos, asegurando que todos los datos (y por ende información) esté al día y sea consistente. Pueden surgir problemas si se considera que para realizar un sistema computarizado hace falta tener acceso a una computadora, un sistema operativo específico e instalar los programas necesarios.

Ya que los usuarios son estudiantes universitarios, como mínimo deben poder tener acceso a un laboratorio de cómputo universitario por lo que el acceso a una computadora no es un problema real. Los problemas de sistema operativo e instalación se pueden solucionar haciendo el programa como una aplicación de internet.

Por otro lado, SIFE UVG cuenta con un sitio de internet. Aparte de información de la organización para los visitantes, cada miembro del equipo tiene un usuario. Este se usa principalmente para subir y descargar archivos, enviar mensajes y actualizar su perfil. Este sitio utiliza una versión modificada de la serie 1.0.x de Joomla!. Para evitar redundancia, un sistema computarizado debe poder integrarse a la funcionalidad que ofrece el sitio.



Figura 7. Sitio de internet de SIFE UVG (Fuente: <http://www.sifeuvg.org.gt>)

## B. Casos de uso

El sistema contempla los roles de director de proyecto, miembro de proyecto, recursos humanos y junta directiva. Los casos de uso correspondientes son:

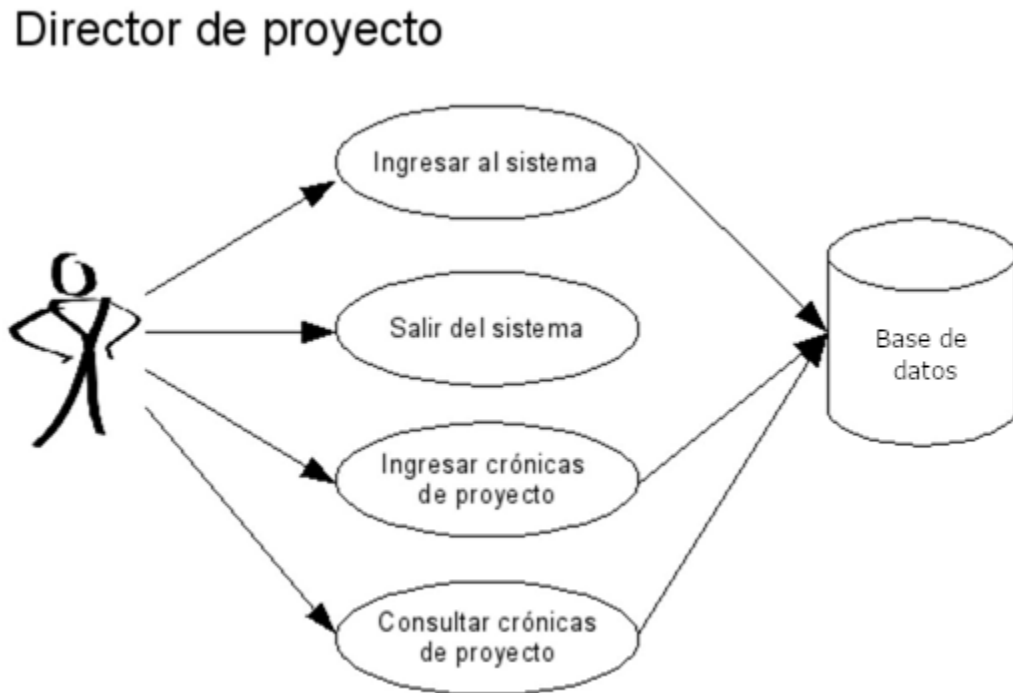


Figura 8. Caso de uso para director de proyecto.

## Miembro de proyecto



Figura 9. Caso de uso para miembro de proyecto.

## Recursos Humanos

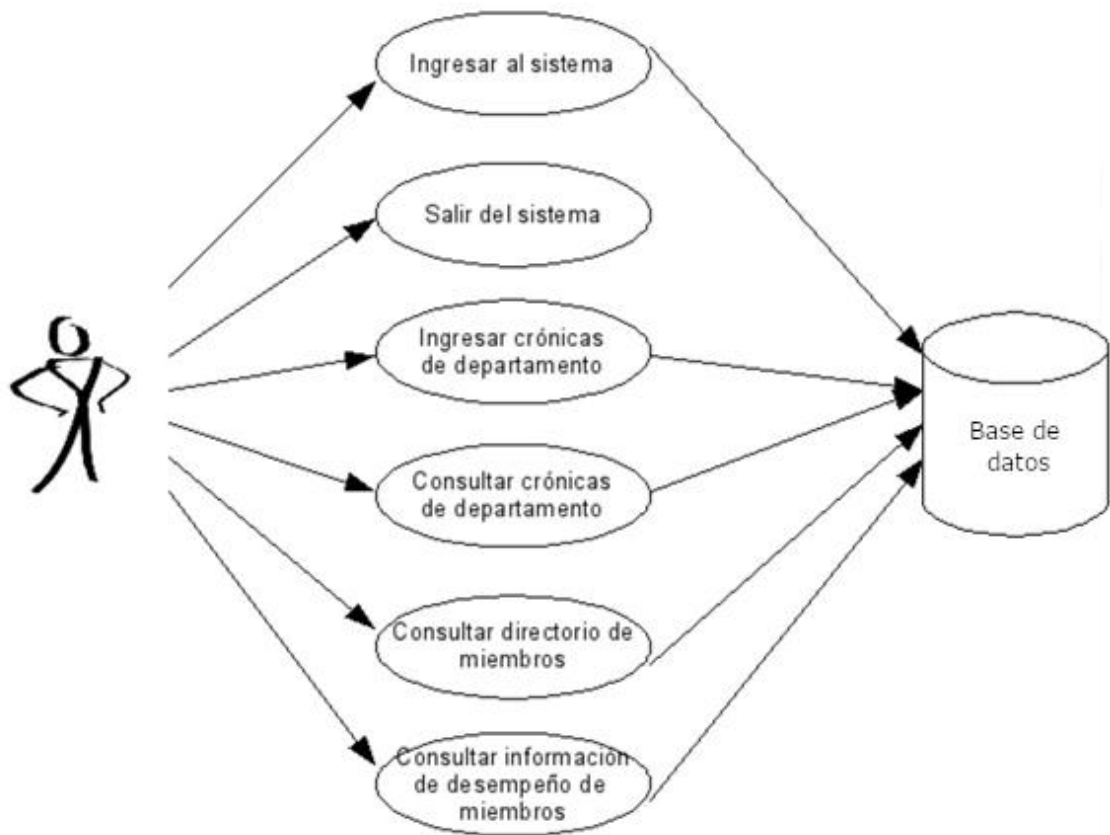


Figura 10. Caso de uso para recursos humanos.

## Junta Directiva



Figura 11. Caso de uso para junta directiva.

### C. Diseño de base de datos

Toda la información necesaria para hacer cálculos y generar reportes del sistema puede ser almacenada en una base de datos relacional. Esta se puede dividir en varias áreas principales:

- **Datos relacionados con usuarios y permisos:** Las tablas de esta área contienen información de los usuarios del sistema, grupos de permisos y relación con las tablas de Joomla!.
- **Datos relacionados con las universidades:** Contienen información general de las universidades.
- **Datos relacionados con los proyectos:** Contienen información general de los proyectos, miembros que pertenecen al mismo, objetivos propuestos y cumplidos y responsables de ejecutar cada objetivo.
- **Datos relacionados con el registro de actividades:** Esta área contiene información de todas las actividades realizadas, costos e ingresos de cada una y miembros que participaron en dicha actividad.

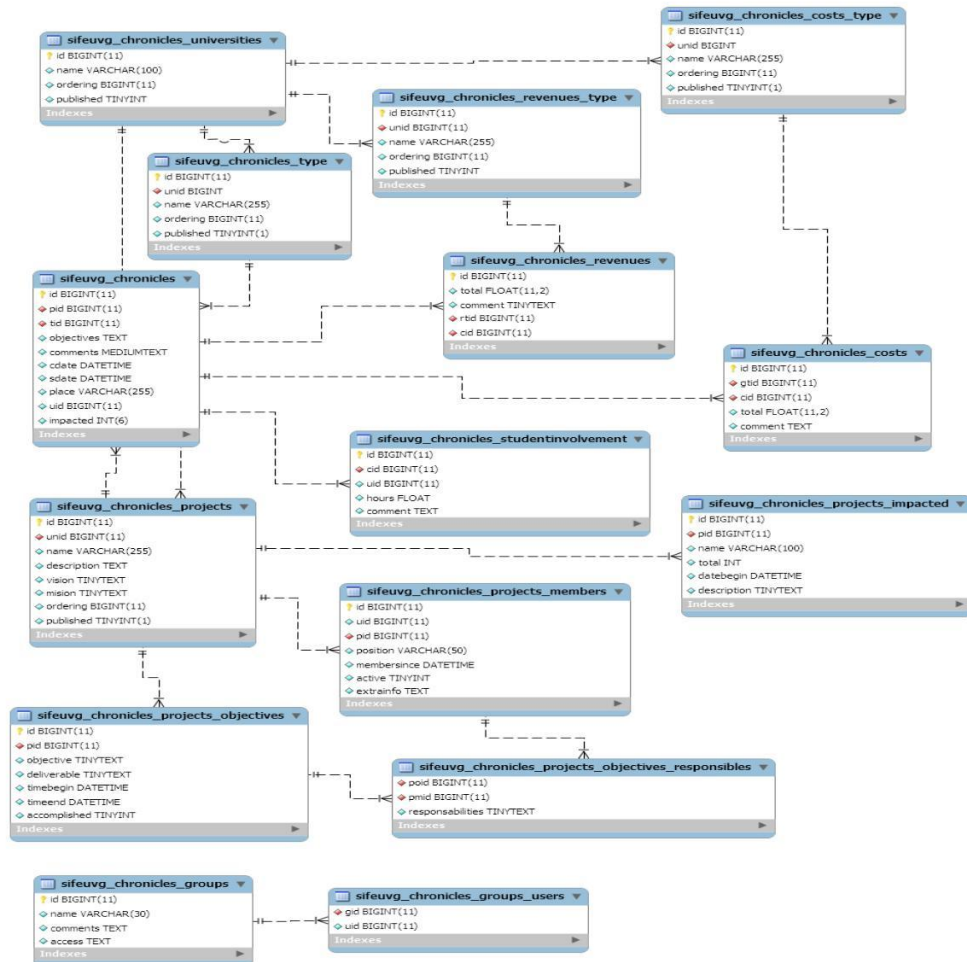


Figura 12. Diagrama de entidad-relación.

## D. Diseño de componente

Para el diseño del componente se siguió el modelo de Joomla! con algunas variantes. Para facilitar la actualización y hacer más sencillas las tareas de los usuarios, se integró la funcionalidad de administración en el lado público, para evitar que los usuarios tengan que acceder por separado al lado público y administrativo.

El componente hace uso del esquema de usuarios de Joomla! para autenticación. Sin embargo, extiende esta funcionalidad usando un esquema de permisos para asignar roles. Los permisos se manejan por universidad, proyecto, opciones y operaciones. Esto permite crear más roles aparte de los definidos por los casos de uso, permitiendo que la aplicación se adapte a cambios futuros. Los permisos se representan con cadenas de caracteres (*strings*), haciendo referencia a las llaves primarias de la tabla de universidades y proyectos y a las opciones del programa. Las opciones del programa se dividen en tres grandes rubros: proyectos, crónicas y administración. Las operaciones que se pueden realizar corresponden

a ingresar, obtener, actualizar y eliminar (*create, retrieve, update y delete*, CRUD por sus siglas en inglés).

En la capa lógica, se efectúan las operaciones sobre la base de datos, así como todos los cálculos requeridos. En esta capa también están todas las validaciones, para evitar inconsistencia en los datos almacenados, así como ataques que comprometan la seguridad del sistema. Siguiendo el esquema de roles, toda combinación de opción de programa y operación tiene asignado una función encargada de procesar dicha combinación. Por ejemplo, la operación de agregar en la opción de crónicas tiene asignada una sola función que se encarga de validar la entrada y almacenar los datos en la base de datos.

La capa de presentación se utilizó para generar un motor AJAX en lugar de HTML directamente. Posteriormente el motor se encarga de la presentación una vez haya cargado en el lado del cliente. Esto también implica que el motor se encarga de generar la interfaz gráfica. El motor AJAX se genera dependiendo del rol que tenga asignado el usuario.

Además de estas capas, también se utilizaron dos archivos de constantes.

El primero guarda información como URLs, números para asignar secciones y otro tipo de constantes utilizados internamente, sobre todo por la capa lógica. Esto permite actualizar aspectos del programa sin tener que revisar todo el código fuente.

El segundo archivo consta de constantes de texto. La programación en la capa lógica y capa de presentación hacen referencia a las constantes de este archivo en lugar de utilizar los textos como literales. Esto también permite que el componente pueda utilizar varios de estos archivos, permitiendo tener el programa en múltiples idiomas.

## **E. Características del sistema**

El desarrollo del sistema se dividió en dos partes. Primero se realizó una versión preliminar (denominada versión beta) para probar la lógica del componente, así como la aceptación de los usuarios finales. Una vez probada la consistencia lógica del programa, se extendió la capa de presentación para aprovechar las tecnologías AJAX de mejor manera.

### **1. Características de la versión beta**

El sistema beta se enfocó en el manejo de crónicas, dividiendo sus operaciones en cuatro opciones: *Nueva, Listar, Reportes y Buscar*. Cada usuario tiene asignado un rol donde se define el acceso a cada una de estas opciones.

La opción de *Nueva* permite ingresar una nueva crónica. El usuario debe escoger el proyecto o departamento al cual pertenece la crónica. La lista de proyectos/departamentos está limitada al rol del usuario, por lo que sólo le aparecerán los elementos que tenga asignados. También permite agregar los participantes en la actividad. El listado de integrantes se genera a partir de los usuarios registrados en el sistema. Los campos obligatorios se indican con un asterisco. El sistema está validado, indicándole al usuario los campos que ha dejado en blanco o que no tienen un valor válido al tratar de enviar el

formulario. Para llevar control, también se registran la fecha y hora de ingreso, así como el usuario que ingresó la crónica. No se pueden realizar ingresos anónimos.

**CRÓNICAS**

[Nueva](#) | [Listar](#) | [Reportes](#) | [Buscar](#)

**Datos Generales**

\*Proyecto/Departamento: TIC

\*Fecha: 2008-10-01

\*Hora: 15 : 00

\*Lugar: UVG, I-106

\*Motivo: Actividad

\*Impactados: 10

\*Objetivos cumplidos: Presentar sistema SIFE

**Agregar Objetivo**

Comentarios adicionales:

**\*Student Involvement**

Miembro	Horas	Comentarios adicionales
Leiva, Donald (admin)	1	

**Agregar Miembro**

**Gastos**

Tipo	Total (\$)	Comentarios adicionales
Otro		

**Agregar Gasto**

**Enviar**

Los campos marcados con \* son obligatorios.

Figura 13. Opción de *Nueva* en la versión beta.

La opción de *Listar* muestra un listado de todas las crónicas ingresadas en el sistema. Este listado está limitado al rol del usuario, por lo que un usuario sólo podrá ver las crónicas de los proyectos/departamentos que tenga asignados. El sistema provee navegación, así como la opción de exportar los datos de una crónica particular a una hoja de cálculo o un documento de texto. Si el usuario tiene permisos de actualización, también puede editar una crónica particular.

**CRÓNICAS**

Nueva  Listar  Reportes  Buscar

Proyecto/Depto.	Tipo	Lugar	Fecha			
Otro	Reunión	UVG	09-07-2007			
OplaCorn	Actividad	ICTA	11-07-2007			
Otro	Reunión	UVG	12-07-2007			
Recursos Humanos	Reunión	UVG	16-07-2007			
Recursos Humanos	Reunión	UVG	19-07-2007			
Recursos Humanos	Reunión	UVG	20-07-2007			
OplaCorn	Reunión	Campus UVG	02-08-2007			
OplaCorn	Reunión	Campus UVG	02-08-2007			
OplaCorn	Capacitación	Mc donalds, Utatlan	02-08-2007			
La Limonada	Reunión	UVG	03-08-2007			
Junta Directiva	Reunión	Cafeteria UVG	09-08-2007			
OplaCorn	Reunión	oficina SIFE-UVG	09-08-2007			
La Limonada	Reunión	Campus UVG	10-08-2007			
Shopping4hope	Reunión	UVG	13-08-2007			
Shopping4hope	Reunión	UVG	16-08-2007			
Shopping4hope	Reunión	UVG	16-08-2007			
Shopping4hope	Reunión	UVG	17-08-2007			
Shopping4hope	Reunión	UVG	17-08-2007			
Junta Directiva	Reunión	I-201	17-08-2007			
TIC	Actividad	Casa de Carlos Villagrán	18-08-2007			

<< Inicio < Anterior 1 2 3 4 5 6 7 8 9 10 Siguiente > Final >>

20 ↓ Página 1 de 17

Figura 14. Opción de *Listar* en el sistema beta.

La opción de *Reportes* permite ver un resumen de varios datos almacenados en el sistema. Entre estos se encuentra el total de horas trabajadas y el costo total por cada estudiante y por proyecto/departamento. Cada reporte tiene opciones específicas para filtrar los resultados. Los reportes se pueden generar como HTML (opción por defecto), como una hoja de cálculo o como documento de texto.

**CRÓNICAS**

Nueva | Listar | **Reportes** | Buscar

**Tipo de Reporte**

- Gastos por rubro
- Gastos por proyecto
- Student Involvement por miembro
- Student Involvement por proyecto

**Tipo de Archivo**

- Documento HTML (WEB)
- Documento de Excel
- Documento de Word

**Proyectos**

- Shopping4hope
- OplaCorn
- Incubadora de Negocios
- Programa de Cap. Empresarial
- Recursos Humanos
- Promoción
- TIC
- Junta Directiva
- Otro
- La Limonada
- Red Tacana
- I. M. A. G. I. N. A.
- Capacitación de Talento

**Opciones**

**Rango de Fechas**  
 Entre  ... y  ...

**Horas**  
 Entre  y

**Ordenar por**

**Ver Reporte**

Figura 15. Opción de *Reportes* en el sistema beta.

La opción de *Buscar* permite hacer una búsqueda de crónicas dentro de campos específicos. El resultado de la búsqueda es un listado de crónicas que cumplen con los criterios de búsqueda.



Figura 16. Opción de *Buscar* en el sistema beta.

## 2. Características de la versión final

Para la versión final se cambió casi exclusivamente la capa de presentación, por lo que el cambio mayor fue en la interfaz gráfica. Esta se generó tomando en cuenta que un usuario puede realizar varias tareas a la vez, como por ejemplo consultar datos para ingresar uno nuevo. Debido a esto, la interfaz se hizo empleando un modelo de pestañas (*tabs*), permitiendo abrir varias al mismo tiempo. Para fácil acceso, el menú siempre es visible al lado derecho del sistema.

El sistema también proporciona ayuda contextual para ayudar al usuario a entender que significan los diferentes elementos del sistema.

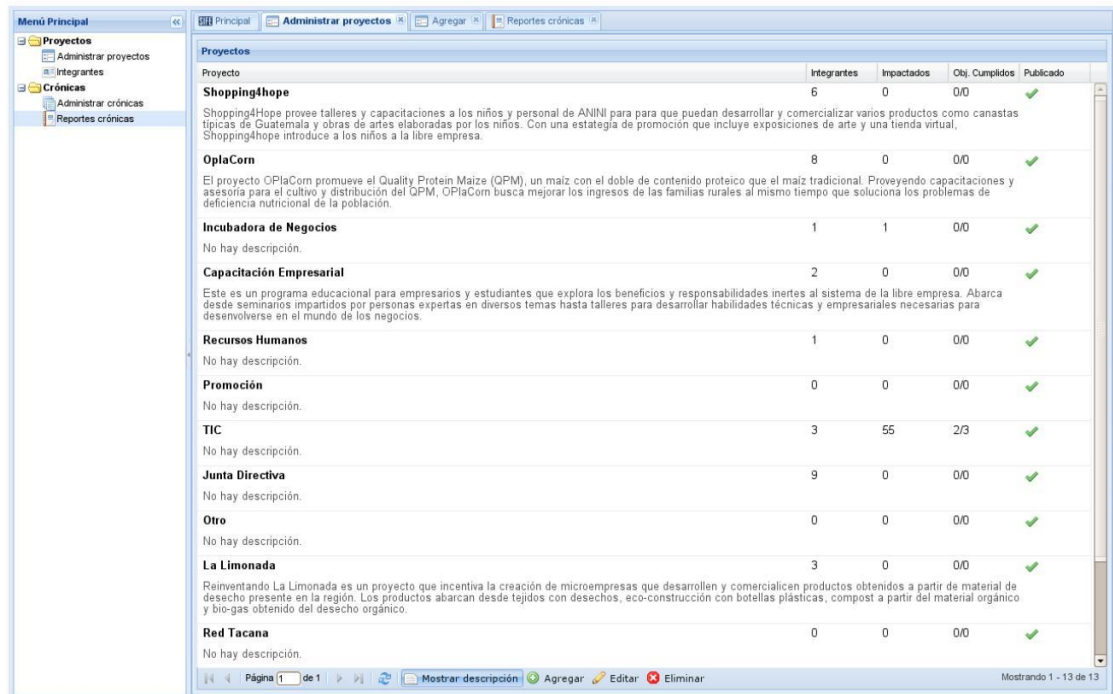


Figura 17. Sistema con varias pestañas abiertas.

En esta versión se presentan opciones para administración de proyectos y control de miembros además del registro de actividades (crónicas). Una diferencia importante entre la versión beta y la final es que las operaciones de ingreso, actualización, eliminación y consulta general (listado) ya no aparecen como opciones separadas, sino que están agrupadas dentro de una misma opción.

La presentación de datos se hace por medio de rejillas (*grids*) con navegación. Los datos que se muestran se pueden ordenar por cualquier de los campos visibles. También se pueden filtrar usando siempre los campos visibles. Al fondo de la rejilla aparecen varias opciones, las cuales se generan dependiendo de los permisos que tiene asignado el rol actual.

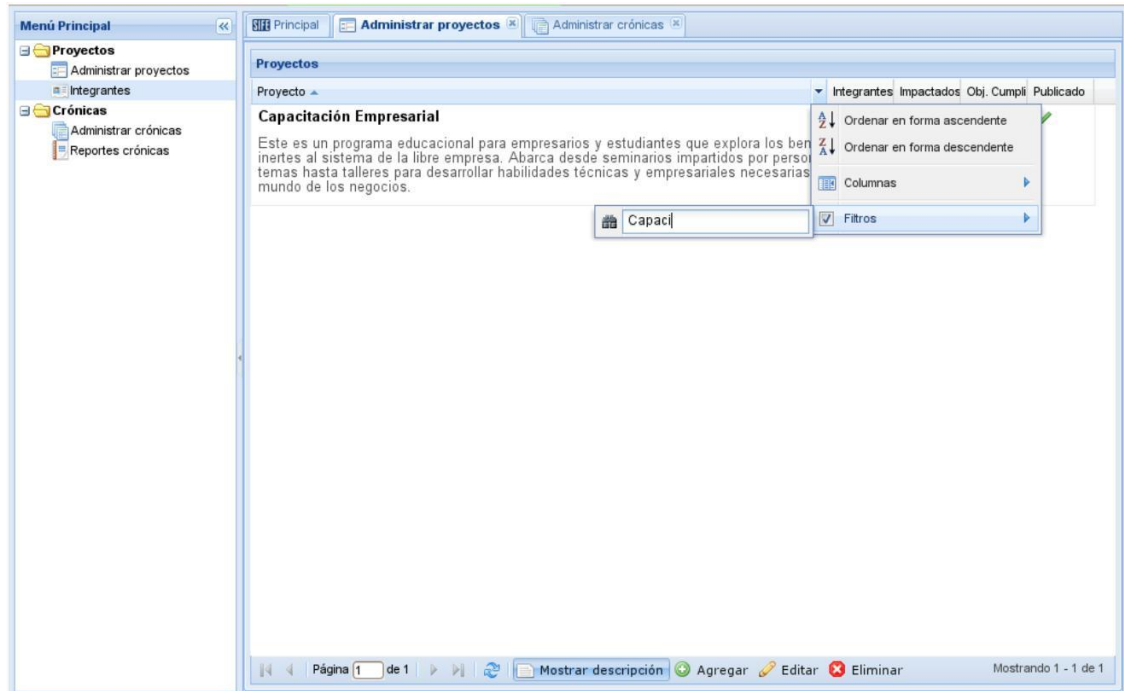


Figura 18. Ejemplo de opciones de ordenamiento y filtrado de proyectos.

En la mayoría de los casos, estas opciones incluyen ingresar, actualizar y eliminar datos de la opción correspondiente. Al seleccionar agregar o editar, se abre una nueva pestaña para no perder la consulta actual. Cada formulario de ingreso/actualización está validado. Al enviar los datos, si hay un problema con algún campo, la aplicación indica dicho problema, resaltando el campo afectado.

The screenshot shows a web application interface with a main menu on the left and a form titled "Agregar proyecto" on the right. The main menu includes "Proyectos" (with sub-items "Administrar proyectos" and "Integrantes") and "Crónicas" (with sub-items "Administrar crónicas" and "Reportes crónicas"). The "Agregar proyecto" form is titled "Agregar proyecto" and contains the following fields:

- Nombre:** Nuevo proyecto
- Universidad:** Universidad del Valle de Guatemala
- Descripción:** (empty text area)
- Visión:** (empty text area)
- Misión:** (empty text area)
- Publicado:**

At the bottom of the form, there are two buttons: "Agregar" and "Reiniciar".

Figura 19. Ingreso de datos para la opción de definición de proyectos.

## V. RESULTADOS Y CONCLUSIONES

- La introducción de técnicas AJAX ha generado un cambio en la forma de desarrollar aplicaciones de internet. Ha permitido aumentar la interactividad y acercarse a lo ofrecido por las aplicaciones de escritorio.
- La integración con el CMS Joomla! fue exitosa. La aplicación corre usando el mismo esquema de usuarios, generando el contenido necesario según cada rol.
- La versión beta tuvo una buena aceptación por parte de los usuarios finales, descartándose por completo el proceso de crónicas manuales. A pesar de la falta de documentación, los usuarios no encontraron mayor problema para usar la herramienta. Desafortunadamente, a la fecha de la elaboración de este trabajo aún no se han podido medir las respuestas de los usuarios con la versión final.
- La aplicación ha permitido reducir los tiempos de recolección de datos para la realización de informes. En 2007, para este proceso se necesitó la colaboración de casi todo el equipo durante un mes. En 2008, con la versión beta de la aplicación sólo fue necesario la intervención de una persona durante tres días.

## VI. RECOMENDACIONES

- Durante el desarrollo de esta aplicación, la nueva versión de Joomla! (serie 1.5) aún se encontraba bajo desarrollo. Actualmente hay una versión estable, por lo que vale la pena considerar migrar la aplicación a esta nueva versión. Para esto también hay que migrar el resto del sitio de SIFE UVG sobre el que corre la aplicación.
- La mayor parte de la aplicación carece de algún medio para exportar los datos. Por la forma en que se despliegan, la mejor opción es generar archivos separados por comas (CSV). Este tipo de archivos se pueden abrir fácilmente en una hoja de cálculo.
- Los datos también se pueden representar gráficamente. Por ejemplo, se podrían generar gráficas tipo pie para ver la distribución de miembros respecto a ciertos parámetros como género y carrera o ver tendencias de ingresos y egresos por medio de una gráfica de líneas. Para esto se puede usar la biblioteca gráfica GD2.
- Con los datos de los proyectos, miembros y objetivos se pueden generar diagramas de Gantt para facilitar la administración de estos.
- Uno de los problemas de equipos SIFE es organizar eventos y reuniones a los que puedan asistir la mayoría de los miembros. El programa se podría integrar con una agenda que incluya horario de clases, por ejemplo, y calcular los períodos en los que todos o al menos la mayoría puede estar disponible. Para esto se pueden emplear algoritmos genéticos. Puede ser necesario dar más tiempo de ejecución para asegurar obtener una respuesta.
- Con los datos de costo e ingresos se pueden realizar modelos basados en la ingeniería financiera para evaluar la rentabilidad de proyectos nuevos y existentes, así como hacer proyecciones y ayudar en la toma de decisiones financieras.
- Se pueden agregar *screencasts* integrados al sistema a modo de manual de usuario para facilitar el uso de la herramienta. Estos screencasts se pueden grabar con aplicaciones como Wink.
- La aplicación está diseñada para ser usada por varios equipos en varias universidades. Sin embargo, aumentar su exposición también aumenta el uso de ancho de banda lo que puede obligar a aumentar los costos de alojamiento para mantener este servicio. Para evitar esto se puede restringir la cantidad de usuarios por universidad o bien cobrar una cuota para cubrir estos gastos.
- La aplicación fue hecha con los textos por separado, para poderla hacer multilingüaje. Por el momento sólo está disponible en español, pero se puede traducir a otros idiomas como inglés y mandarín. Esto es importante al

momento de abrir la herramienta para otros equipos SIFE en países que no se hable español.

- El sistema guarda información de los usuarios que efectúan operaciones.
- Con esta información se puede desarrollar un módulo nuevo dentro del sistema para sacar estadísticas de uso.

## VII. BIBLIOGRAFÍA

Garret, Jesse. (2005). *Ajax: A New Approach to Web Applications*. Recuperado el 13 de septiembre de 2008 de <http://adaptivepath.com/publications/essays/archives/000385.php>

Hopmann, Alex. (2007). The Story of XMLHTTP.  
<http://www.alexhopmann.com/xmlhttp.htm>. Recuperado el 13 de septiembre de 2008 de <http://www.alexhopmann.com/xmlhttp.htm>

Kendall, Kenneth; J. Kendall. 1997. *Análisis y Diseño de Sistemas*. 3ra. edición. México: Editorial Pearson educación. 913 págs.

Koontz, Harold; H. Weihrich. 2002. *Administración: Una perspectiva global*. 12a. edición. Editorial McGraw Hill. 832 págs.

Mozilla Developer. (2006). *JavaScript Overview*. Recuperado el 25 de agosto de 2008 de [http://developer.mozilla.org/en/docs/Core\\_JavaScript\\_1.5\\_Guide:JavaScript\\_Overview#What\\_is\\_JavaScript.3F](http://developer.mozilla.org/en/docs/Core_JavaScript_1.5_Guide:JavaScript_Overview#What_is_JavaScript.3F)

Powell, Thomas. (2006). *AJAX is the future of app development*. Recuperado el 02 de septiembre de 2008 de <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9001812>

Pressman, Roger. 2006. *Ingeniería de software: un enfoque práctico*. 6ta. edición. México: Editorial McGraw Hill. 958 págs.

SIFE. 2007. *SIFE Global Handbook 2007-2008*. 1ra edición: SIFE. 64 págs.

Silberchatz, Abraham; H. Korth y S. Sudarshan. 2002. *Fundamentos de bases de datos*. 4ta. edición. Madrid: Editorial McGraw Hill. 787 págs.

Verzuh, Erick. 2005. *The Fast Forward MBA Project Management*. 2da. edición: Wiley. 402 págs.

World Wide Web Consortium. (2008). *Document Object Model*. Recuperado el 15 de septiembre 2008 de <http://www.w3.org/DOM/>

World Wide Web Consortium. (2008). *The XMLHttpRequest Object*. Recuperado el 15 de septiembre 2008 de <http://www.w3.org/TR/XMLHttpRequest>

White, Alexei. (2005). *Measuring the benefits of AJAX*. Recuperado el 02 de septiembre de 2008 de <http://www.developer.com/java/other/article.php/3554271>

## VIII. APÉNDICES

### A. Herramientas y tecnologías usadas

Para la elaboración de la aplicación y este trabajo se utilizaron las siguientes herramientas:

- **HTML:** Utilizado como plantilla para cargar el motor AJAX.
- **CSS:** Utilizado para estilos.
- **JSON:** Utilizado como medio de comunicación entre el servidor y el motor AJAX.
- **Javascript:** Utilizado para la programación del lado del cliente.
- Lenguaje integrador de AJAX.
- **PHP:** Utilizado en la programación del lado del servidor. Lenguaje en el que está escrito Joomla!. Aunque se usó la versión 5, por compatibilidad sólo se usaron funciones de la versión 4.
- **MySQL:** Manejador de base de datos. Se utilizó la versión 5. La serie Joomla! 1.0.x usa este manejador de base de datos.
- **Apache:** Servidor web. PHP se ejecuta sobre Apache.
- **Joomla!:** CMS utilizado por el sitio de SIFE UVG. La aplicación se desarrolló como componente de Joomla!, aunque por la forma en que se diseñó, se puede usar en otros sistemas con pequeñas modificaciones.
- **ExtJs:** Biblioteca de funciones para aplicaciones AJAX. El motor AJAX hace uso de ella.
- **Wine:** Emulador de entorno Windows para Linux.
- **MySQL Workbench:** Diagramador de entidad-relación. Usado para diseñar la base de datos. Genera scripts. Para Windows.
- **MySQLYog:** Front-end para MySQL. Para Windows, utilizado en Linux con Wine.
- **QuantaPlus:** Editor de texto para programación para Linux.
- **Firefox:** Navegador de internet. Utilizado para probar la aplicación.
- **Google Chrome:** Navegador de internet. Utilizado para probar la aplicación.
- **Internet Explorer:** Navegador de internet. Utilizado para probar la aplicación.
- **Safari:** Navegador de internet. Utilizado para probar la aplicación.
- **FireBug:** Extensión de Firefox para hacer debugging de páginas de internet. Utilizado para asistir en el desarrollo.
- **GIMP Image editor:** Utilizado para manipular imágenes.
- **openOffice.org Writer:** Procesador de palabras. Utilizado para escribir documentación incluyendo este trabajo.

- **openOffice.org Draw:** Diagramador. Utilizado para realizar la mayoría de diagramas.

## IX. ANEXOS

### A. Modelo original de crónica (formulario en papel)

Se adjunta una hoja de captura de datos que se usó de base para el desarrollo del sistema, elaborada por miembros del equipo SIFE. La hoja se muestra sin alteraciones.



Fecha:

Crónica #

Hora:  
Lugar:

Nombre del Proyecto:

Tema a tratar:

Reunión

Actividad

Otro

Agenda (objetivos a tratar en reunión):

- 
- 
- 
- 

1. Para que se esta haciendo la actividad?

*\*/Aquí discutirán el objetivo principal de la reunión./\**

2. Que se hizo durante esta sesión?

- Objetivos alcanzados:

- 
- 
- 

- Temas nuevos a tratar en el transcurso de la siguiente sesión:

- 
- 
-



*\*| Aquí los project manager's podrán discutir sobre los objetivos alcanzados en la reunión, así como los temas nuevos a realizar en el transcurso de esta reunión a la siguiente |\**

---

3. Que se va hacer a raíz de eso?

*\*| Aquí discutirán sobre las decisiones que se tomarán para alcanzar los objetivos deseados |\**

*\*| La forma en que se presenten esto es a su criterio, yo solo les doy un esbozo de cómo podría hacerse la crónica, cualquier duda preguntenme. Ahora lo que si deseo es q sean puntuales ya que es más fácil recopilar información de esa manera.*

*Adjunto a las crónicas siempre deben de tener el listado de los integrantes que asistieron a la reunión, así como las horas que tomó por cada integrante, para el student involvement, y la hoja debe estar firmada por uds (la hoja en sucio es para su archivo, a mi me mandan todo electrónicamente a [sifeuvg@gmail.com](mailto:sifeuvg@gmail.com))*

*\*|*

## B. Código fuente en el lado del servidor

### 1. Ejemplo de clase

```
<?php
// no direct access
defined( 'VALID_MOS' ) or die( 'Restricted access' );

/**
 * Category database table class
 * @package Joomla
 * @subpackage Cronicas
 */

class mosChroniclesProjects extends mosDBTable {
    /** @var int Primary key */
    var $id          = null;
    /** @var int */
    var $uid         = null;
    /** @var string */
    var $name        = null;
    /** @var string */
    var $description = null;
    /** @var string */
    var $vision      = null;
    /** @var string */
    var $mission     = null;
    /** @var int */
    var $ordering    = null;
    /** @var int */
    var $published   = null;

    /**
     * @param database A database connector object
     */
    function mosChroniclesProjects( &$db ) {
        $this->mosDBTable( '#__chronicles_projects', 'id', $db );
    }
}
```

```

}

/** overloaded check function */
function check() {

    /** check for valid name */
    if (trim( $this->name ) == "" ) {
        $this->error =
        _SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_NAME_BLANK;
        return false;
    }

    /** check for existing name */
    $query = "SELECT id"
        . "\n FROM #__chronicles_projects "
        . "\n WHERE name = " . $this->_db->Quote( $this->name ) . " AND unid=" .
    $this->unid;
    $this->_db->setQuery( $query );

    $xid = intval( $this->_db->loadResult() );
    if ( $xid && $xid != intval( $this->id ) ) {
        $this->error =
        _SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_NAME_EXISTS;
        return false;
    }

    if ( $this->unid == 0 ) {
        $this->error =
        _SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_UNIVERSITY_BLANK;
        return false;
    }

    if ( $this->published === null )
        $this->published = 0;

    return true;
}
}

```

## 2. Ejemplo de lógica

```

// REMEMBER: Grant access only if original author AND has permission for that project
// no direct access
defined( '_VALID_MOS' ) or die( 'Restricted access' );
session_start();
require_once( $mainframe->getPath( 'front_html' ) );
require_once( $mainframe->getPath( 'class' ) );

```

```

require_once( $mosConfig_absolute_path . "/components/$option/constants/" .
$mosConfig_lang . '.php');
require_once( $mosConfig_absolute_path . "/components/$option/constants.php");
$mainframe->setPageTitle( _SIFEAPP_TITLE );

if (!empty($my))
{
    $query = "SELECT access \n" .
    "FROM #__chronicles_groups_users cgu \n" .
    "LEFT JOIN #__chronicles_groups cg ON cg.id=cgu.gid \n" .
    "WHERE cgu.uid=" . $my->id;

    $database->setQuery($query);
    $accessObj = $database->loadObjectList();

    if (empty($accessObj))
    {
        die ( _SIFEAPP_RESTRICTED_ACCESS);
    } else
    {
        $accessStr = "";

        foreach ($accessObj as $aObj)
        {
            $accessStr .= $aObj->access . '|';
        }
        $myAccess = new UserAccessHandler($accessStr);
    }
} else
{
    die ( _SIFEAPP_RESTRICTED_ACCESS);
}

switch ($task) {
    // Projects tasks
    // Projects Main Manager
    case 'projlist':
        projList($option);
        break;
    case 'projload':
        projLoad($option);
        break;
    case 'projsave':
        projSave($option);
        break;
}

```

```

    case 'projdel':
        projDel($option);
    break;
    default:
        mainApplication();
    break;
}

function projList($option)
{
    global $database, $myAccess, $mosConfig_list_limit;

    $univIDs = $myAccess->getUniversities(_SIFEAPP_OPTION_PROJECT,
    _SIFEAPP_PROJECT_EDIT, 'r');
    $proyIDs = $myAccess->getProjects(_SIFEAPP_OPTION_PROJECT,
    _SIFEAPP_PROJECT_EDIT, 'r');

    $start= intval(mosGetParam($_POST, 'start', 0));
    $limit = intval(mosGetParam($_POST, 'limit', $mosConfig_list_limit));
    $sort = mosGetParam($_POST, 'sort', 'cp.unid, cp.id');
    $dir = mosGetParam($_POST, 'dir', "");
    $filter = mosGetParam($_REQUEST, 'filter', "");
    $fwhere = getFilterWhere($filter);

    // Base Query
    $baseQuery = "SELECT cp.id as cpid, cu.id as cuid, cu.name as unname, cp.name
as pname, " .
        "IFNULL(cp.description, "" . _SIFEAPP_NO_DESCRIPTION . "") as
descr, \n" .
        "(SELECT IFNULL(COUNT(*), 0) " .
        "FROM #__chronicles_projects_members cpm " .
        "WHERE cpm.pid=cpid AND cpm.active=1) members,\n" .
        "(SELECT IFNULL(SUM(total), 0) " .
        "FROM #__chronicles_projects_impacted cpi " .
        "WHERE cpi.pid=cpid) impacted,\n" .
        "(SELECT IFNULL(COUNT(*), 0) " .
        "FROM #__chronicles_projects_objectives cpo " .
        "WHERE cpo.pid=cpid) totalobj,\n" .
        "(SELECT IFNULL(COUNT(*), 0) " .
        "FROM #__chronicles_projects_objectives cpo " .
        "WHERE cpo.pid=cpid AND cpo.accomplished=1) accomobj, \n" .
        "cp.published published \n" .
        "FROM #__chronicles_projects cp \n" .
        "LEFT JOIN #__chronicles_universities cu ON (cp.unid=cu.id) \n" .
        "WHERE " . (in_array(0, $univIDs) ? "" : ("cu.id IN (" . implode(',',
$univIDs) . ") AND ")) .

```

```

        (in_array(0, $proyIDs) ? "" : ("cp.id IN (". implode(',', $proyIDs) . ") AND
    ")).
        "cu.published=1 \n" .
        "ORDER BY $sort $dir \n";

    // Row count
    $query = "SELECT COUNT(*) as tot \n" .
        "FROM ($baseQuery) as tpt \n" .
        "WHERE $fwhere";

    $database->setQuery($query);
    $rowsTotal = $database->loadResult();

    if ($rowsTotal > 0)
    {
        // Results query
        $query = "SELECT * \n" .
            "FROM ($baseQuery) as tpt \n" .
            "WHERE $fwhere \n" .
            "LIMIT " . $start . ', ' . $limit;

        $database->setQuery($query);
        $rows = $database->loadObjectList();

        echo '({"total":' . $rowsTotal . ', "results":' . json_encode($rows) . '})';
    }
    else {
        echo '({"total":0, "results":""})';
    }
    exit();
}

function projLoad($option)
{
    global $database, $myAccess;

    if (!$myAccess->hasAccessOption(_SIFEAPP_OPTION_PROJECT,
        _SIFEAPP_PROJECT_EDIT, 'r'))
    {
        echo "{success: false, errors: { reason: '" .
            _SIFEAPP_RESTRICTED_ACCESS . "' }}";
        exit();
    }

    $sid = intval( mosGetParam( $_REQUEST, 'id', 0 ) );

```

```

if ($Sid <= 0)
{
    echo "{success: false, errors: { reason: " . _SIFEAPP_NO_ID . " }}";
    exit();
}

$row = new mosChroniclesProjects($database);

if (!$row->load( (int)$Sid ))
{
    echo "{success: false, errors: { reason: " . $row->getError() . " }}";
    exit();
}

$arrData = getDataArray($row);

echo '{"success":true, "data":' . json_encode($arrData) . '}';

exit();
}

function projSave($option)
{
    global $database, $myAccess;

    if (!$myAccess->hasAccessOption(_SIFEAPP_OPTION_PROJECT,
    _SIFEAPP_PROJECT_EDIT, 'cu'))
    {
        echo "{success: false, errors: { reason: " .
    _SIFEAPP_RESTRICTED_ACCESS . " }}";
        exit();
    }
    $rowProj = new mosChroniclesProjects($database);

    if (!$rowProj->bind( $_POST )) {
        echo "{success: false, errors: { reason: " . $rowProj->getError() . " }}";
        exit();
    }

    if (!$rowProj->check())
    {
        echo "{success: false, errors: { reason: " . $rowProj->getError() . " }}";
        exit();
    }

    if (!$rowProj->store()) {

```

```

        echo "{success: false, errors: { reason: " . $rowProj->getError() . " }}";
        exit();
    }

    echo '({success: true})';
    exit();
}

function projDel($option)
{
    global $database, $myAccess;
    if (!$myAccess->hasAccessOption(_SIFEAPP_OPTION_PROJECT,
    _SIFEAPP_PROJECT_EDIT, 'd'))
    {
        echo "{success: false, errors: { reason: " .
    _SIFEAPP_RESTRICTED_ACCESS . " }}";
        exit();
    }

    $sid = !empty($_REQUEST['id']) ? $_REQUEST['id'] : 0;;

    $query = "DELETE FROM #__chronicles_projects \n" .
    "WHERE id IN (" . $sid . ")";

    $database->setQuery($query);

    if (!$database->query())
        echo "{success: false, errors: { reason: " . $database->getErrorMsg() . "
}}";
    else
        echo '{"success":true}';
    exit();
}

function loadLibrary($option)
{
    global $itemid, $myAccess;
    // Define buttons from main menu
    $arrButtons = "";

    // Chronicles Menu Buttons
    $arrButtons['chronicles'] = "";

    $arrButtons['chronicles'][] = array(
        'title' => _SIFEAPP_OPTION_CHRONICLE_LIST,
        'id' => 'tabchr1',

```

```

        'tooltiptext' => _SIFEAPP_OPTION_CHRONICLE_LIST_TOOLTIP,
        'icon' => _SIFEAPP_CHRONICLES_IMG_OPT_CHR_LIST,
        'urlload' => "index.php?option=$option&task=chrlist&Itemid=$Itemid",
        'func' => 'chronicles_list',
        'accopt' => _SIFEAPP_OPTION_CHRONICLE,
        'accoptnum' => _SIFEAPP_CHRONICLES_LIST,
        'accrud' => "
    );

    $arrButtons['chronicles'][] = array(
        'title' => _SIFEAPP_OPTION_CHRONICLE_REPORT,
        'id' => 'tabchr2',
        'tooltiptext' => _SIFEAPP_OPTION_CHRONICLE_REPORT_TOOLTIP,
        'icon' => _SIFEAPP_CHRONICLES_IMG_OPT_CHR_REPORT,
        'urlload' =>
"index.php?option=$option&task=chrreport&Itemid=$Itemid",
        'func' => 'chronicles_reports',
        'accopt' => _SIFEAPP_OPTION_CHRONICLE,
        'accoptnum' => _SIFEAPP_CHRONICLES_REPORT,
        'accrud' => "",
        'params' => array('project' =>
getProjObjList(_SIFEAPP_OPTION_CHRONICLE,
_SIFEAPP_CHRONICLES_REPORT))
    );

    // Projects Menu Buttons
    $arrButtons['projects'] = "";

    $arrButtons['projects'][] = array(
        'title' => _SIFEAPP_OPTION_PROJECT_EDIT,
        'id' => 'tabproj1',
        'tooltiptext' => _SIFEAPP_OPTION_PROJECT_EDIT_TOOLTIP,
        'icon' => _SIFEAPP_CHRONICLES_IMG_OPT_PROJECT_EDIT,
        'urllist' => "index.php?option=$option&task=projlist&Itemid=$Itemid",
        'func' => 'project_edit',
        'accopt' => _SIFEAPP_OPTION_PROJECT,
        'accoptnum' => _SIFEAPP_PROJECT_EDIT,
        'accrud' => "
    );

    $arrButtons['projects'][] = array(
        'title' => SIFEAPP_OPTION_PROJECT_MEMBERS,
        'id' => 'tabproj2',
        'tooltiptext' => SIFEAPP_OPTION_PROJECT_MEMBERS_TOOLTIP,
        'icon' =>
SIFEAPP_CHRONICLES_IMG_OPT_MEMBERS_DIRECTORY,

```

```

        'urlload' =>
"index.php?option=$option&task=projmembers&Itemid=$itemid",
        'func' => 'project_members',
        'accopt' => SIFEAPP_OPTION_PROJECT,
        'accoptnum' => SIFEAPP_PROJECT_MEMBERS,
        'accrud' => '1'
    );

    $arrButtons['projects'][] = array(
        'title' => SIFEAPP_OPTION_PROJECT_REPORT,
        'id' => 'tabproj3',
        'tooltiptext' => SIFEAPP_OPTION_PROJECT_REPORT_TOOLTIP,
        'icon' =>
SIFEAPP_CHRONICLES_IMG_OPT_PROJECT_REPORT,
        'urlload' =>
"#index.php?option=$option&task=projreports&Itemid=$itemid",
        'func' => 'project_reports',
        'accopt' => SIFEAPP_OPTION_PROJECT,
        'accoptnum' => SIFEAPP_PROJECT_REPORT,
        'accrud' => '1'
    );

// Administration Menu Buttons
    $arrButtons['admin'] = "";

    $arrButtons['admin'] = array(
        'title' => SIFEAPP_OPTION_ADMIN_UNIVERSITY,
        'id' => 'tabadmin1',
        'tooltiptext' => SIFEAPP_OPTION_ADMIN_UNIVERSITY_TOOLTIP,
        'icon' =>
SIFEAPP_CHRONICLES_IMG_OPT_ADMIN_UNIVERSITY,
        'urlload' =>
"index.php?option=$option&task=adminuniv&itemid=$itemid",
        'func' => 'admin_univ',
        'accopt' => SIFEAPP_OPTION_ADMIN,
        'accoptnum' => SIFEAPP_ADMIN_UNIVERSITY,
        'accrud' => '1'
    );

    $arrButtons['admin'] = array(
        'title' => 'SIFEAPP_OPTION_ADMIN_GROUPS',
        'id' => 'tabadmin2',
        'tooltiptext' => SIFEAPP_OPTION_ADMIN_GROUPS_TOOLTIP,
        'icon' => SIFEAPP_CHRONICLES_IMG_OPT_ADMIN_GROUPS,
        'urlload' =>
"index.php?option=$option&task=admingr&Itemid=$itemid",

```

```

        'func'      => 'admin_groups',
        'accept'   => SIFEAPP_OPTION_ADMIN,
        'acceptnum' => SIFEAPP_ADMIN_GROUPS,
        'accrud'   => '1'
    );

    $arrButtons['admin'] = array(
        'title'     => SIFEAPP_OPTION_ADMIN_USERS,
        'id'        => 'tabadmin3',
        'tooltiptext' => SIFEAPP_OPTION_ADMIN_USERS_TOOLTIP,
        'icon'      => SIFEAPP_CHRONICLES_IMG_OPT_ADMIN_USERS,
        'urlload'   =>
        "index.php?option=$option&task=adminuser&Itemid=$Itemid",
        'func'      => 'admin_users',
        'accept'   => SIFEAPP_OPTION_ADMIN,
        'acceptnum' => SIFEAPP_ADMIN_USERS,
        'accrud'   => '1'
    );

    // Blocks no-access buttons
    $buttons['chronicles'] = array();
    foreach ($arrButtons['chronicles'] as SeleButton) {
        if ($myAccess->hasAccessOption(SeleButton['accept'],
        SeleButton['acceptnum'], SeleButton['accrud'])) {
            $buttons['chronicles'][] = SeleButton;
        }
    }

    $buttons['projects'] = array();
    foreach ($arrButtons['projects'] as SeleButton) {
        if ($myAccess->hasAccessOption(SeleButton['accept'],
        SeleButton['acceptnum'], SeleButton['accrud'])) {
            $buttons['projects'][] = SeleButton;
        }
    }

    $buttons['admin'] = array();
    foreach ($arrButtons['admin'] as SeleButton) {
        if ($myAccess->hasAccessOption(SeleButton['accept'],
        SeleButton['acceptnum'], SeleButton['accrud'])) {
            $buttons['admin'][] = SeleButton;
        }
    }

    HTML_SIFEUVGApp::library($option, &$buttons);
    exit;

```

```

}

function mainApplication() {
    HTML_SIFEUVGApp::main();
}

```

### 3. Ejemplo de presentación

```

<?php

// no direct access
defined('_JEXEC') or die('Restricted access');

/**
 * @package Joomla
 * @subpackage Cronicas
 */

class HTML_SIFEUVGApp {

    function library($option, $buttons) {
        global $mosConfig, $Itemid, $myAccess;
        header("Content-type: application/x-javascript; charset=UTF-8");
        ?>
        Ext.BLANK_IMAGE_URL = '<?php echo $mosConfig->live_site;
        ?>/includes/js/extjs/resources/images/default/s.gif';

        // create namespace
        Ext.namespace('SIFEUVGApp');

        // create application
        SIFEUVGApp.app = function() {
            // private variables.

            // private functions

            // public space
            return {
                // public properties, e.g. strings to translate

                // public methods
                init: function() {
                    Ext.QuickTips.init();
                    Ext.menu.RangeMenu.prototype.icons = {
                        gt: '<?php echo $mosConfig->live_site;
                        ?>/includes/js/extjs/plugins/img/greater_than.png',

```

```

        lt: '<?php echo $mosConfig->live_site;
?>/includes/js/extjs/plugins/img/less_then.png',
        eq: '<?php echo $mosConfig->live_site;
?>/includes/js/extjs/plugins/img/equals.png'
    };
    Ext.grid.filter.StringFilter.prototype.icon = '<?php echo $mosConfig-
>live_site; ?>/includes/js/extjs/plugins/img/find.png';

    <?php if (is_array($buttons['chronicles'])) { ?>
        // Chronicles Nodes
        var chrNode = new Ext.tree.TreeNode({
            text: '<?php echo
SIFEAPP_CHRONICLES; ?>',
            expanded: true,
            cls: 'mainmenu-node'
        });

        <?php foreach ($buttons['chronicles'] as $i => $button)
            { ?>
            chrNode.appendChild({
                text: '<?php echo htmlspecialchars($button['title'], ENT_QUOTES);
?>',
                leaf: true,
                icon: '<?php echo $button['icon']; ?>',
                qtip: '<?php echo htmlspecialchars($button['tooltiptext'],
ENT_QUOTES); ?>',
                listeners: {
                    dblclick: function(node, el) {
                        <?php if (!empty($button['func']))
                            {
                                var $func = array("self", $button['func']);
                                call_user_func($func, array($button, $i));
                            } ?>
                    }
                }
            });
        <?php } ?>
    <?php } ?>

    <?php if (is_array($buttons['projects'])) { ?>
        // Projects Buttons
        var prjNode = new Ext.tree.TreeNode({
            text: '<?php echo SIFEAPP_PROJECTS; ?>',
            expanded: true,
            cls: 'mainmenu-node'
        });

```

```

<?php foreach ($buttons['projects'] as $i => $button)
    {
    ?>
    prjNode.appendChild({
        text: '<?php echo htmlspecialchars($button['title'], ENT_QUOTES);
?>',
        leaf: true,
        icon: '<?php echo $button['icon']; ?>',
        qtip: '<?php echo htmlspecialchars($button['tooltiptext'],
ENT_QUOTES); ?>',
        listeners: {
            dblclick: function(node, e) {
                <?php if (!empty($button['func']))
                    {
                    $func = array("self", $button['func']);
                    call_user_func($func, array($button, $i));
                    }
                ?>
            }
        });
    <?php } ?>
<?php } ?>

<?php if (is_array($buttons['admin'])) { ?>
    // Administrator Buttons
    var adminNode = new Ext.tree.TreeNode({
        text: '<?php echo SIFEAPP_ADMIN; ?>',
        expanded: true,
        cls: 'mainmenu-node'
    });

    <?php foreach ($buttons['admin'] as $i => $button) { ?>
        adminNode.appendChild({
            text: '<?php echo htmlspecialchars($button['title'], ENT_QUOTES);
?>',
            leaf: true,
            icon: '<?php echo $button['icon']; ?>',
            qtip: '<?php echo htmlspecialchars($button['tooltiptext'],
ENT_QUOTES); ?>',
            listeners: {
                dblclick: function() {
                    <?php if (!empty($button['func']))
                        {
                        $func = array("self", $button['func']);

```

```

        call_user_func($func, array($button, $i));
    }
}
});
<?php } ?>
<?php } ?>
?>

```

```

var panelOptions = new Ext.tree.TreePanel({
    split: true,
    title: '<?php echo SIFEAPP_MAIN_MENU_TITLE; ?>',
    region: 'west',
    collapsible: true,
    collapseMode: "mini",
    width: 200,
    minWidth: 150,
    border: true,
    margins: '5 0 5 5',
    baseCls: 'x-panel',
    root: new Ext.tree.TreeNode(),
    rootVisible: false,
    lines: false
});

```

```

<?php if (is_array($buttons['projects'])) { ?>
    panelOptions.getRootNode().appendChild(prjNode);
<?php } ?>

```

```

<?php if (is_array($buttons['chronicles'])) { ?>
    panelOptions.getRootNode().appendChild(chrNode);
<?php } ?>

```

```

<?php if (is_array($buttons['admin'])) { ?>
    panelOptions.getRootNode().appendChild(adminNode);
<?php } ?>

```

```

// Main (Tabbed) Panel
var panelMain = new Ext.TabPanel({
    id: 'panelMain',
    region: 'center',
    deferredRender: false,
    autoScroll: true,
    margins: '5 5 5 0',
    activeTab: 0,
    enableTabScroll: true,

```

```

        items: [{
            id: 'tab1',
            autoLoad: '<?php echo SIFEAPP_CHRONICLES_MAIN_TAB_URL;
?>',

            title: '<?php echo SIFEAPP_MAIN_TAB_TITLE; ?>',
            closable: false,
            autoScroll: true,
            iconCls: 'tab-mainicon',
            icon: '<?php echo SIFEAPP_CHRONICLES_IMG_CMN_ADD; ?>'
        }]
    });

    // Configure viewport
    var viewport = new Ext.Viewport({
        layout: 'border',
        items: [panelOptions, panelMain]
    });
}
};
})(); // end of app
// end of file

<?php
}

function project_edit(&$button, $i) {
    global $mosConfig, $mosConfig_list_limit, $myAccess;
    if ($myAccess->hasAccessOption(_SIFEAPP_OPTION_PROJECT,
SIFEAPP_PROJECT_EDIT, "")) {
        ?>
        if (!panelMain.findById('<?php echo $button['id']; ?>')) {
            var store = new Ext.data.JsonStore({
                root: 'results',
                idProperty: 'cpid',
                totalProperty: 'total',
                remoteSort: true,
                fields: [
                    {name: 'cuid'},
                    {name: 'cpid'},
                    {name: 'name'},
                    {name: 'unname'},
                    {name: 'prname'},
                    {name: 'descr'},
                    {name: 'members'},
                    {name: 'impacted'},
                    {name: 'totalobj'},

```

```

        {name: 'accomobj'},
        {name: 'published'}
    ],
    proxy: new Ext.data.HttpProxy({
        url: '<?php echo $mosConfig-
>live_site;?>/<?php echo sefRelToAbs($button['urllist']); ?>',
        method: 'POST'
    })
});

// Project Load Main Form function
function fn_prjfrm(id) {
    if (!panelMain.findById('panoptprojectadd' + id))
    {
        var panelForm = new Ext.FormPanel({
            frame: true,
            labelWidth: 75,
            url: '<?php echo $mosConfig-
>live_site;?>/<?php echo
sefRelToAbs(SIFEAPP_CHRONICLES_PROJECTS_ADD_SEND_URL); ?>',
            frame: true,
            layout: 'Fit',
            width: 450,
            id: 'chrprjeditaddfrm' + id,
            bodyStyle: 'padding: 5px 5px 0;',
            title: (id == 0) ? '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_TITLE; ?>' : '<?php echo
SIFEAPP_PROJECT_EDIT_UPDATE_TITLE; ?>',
            items: [{
                xtype: 'fieldset',
                title: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_TITLE; ?>',
                autoHeight: true,
                defaults: {anchor: '100%'},
                defaultType: 'textfield',
                items: [
                    {xtype: 'hidden',
                        name: 'id'},
                    {xtype: 'textfield',
                        name: 'name', fieldLabel: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_NAME; ?>', allowBlank: false,
                        blankText: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_NAME_BLANK; ?>'},
                    {
                        xtype:
'combo',

```

```

Ext.data.JsonStore({
    store: new
    'results',
    root:
    ['id', 'name'],
    fields:
        autoLoad: true,
        url:
        '<?php echo $mosConfig->live_site; ?>/<?php echo
        sefRelToAbs(SIFEAPP_CHRONICLES_PROJECTS_UNIVERSITY_URL); ?>'
        }),
        fieldLabel:
        '<?php echo SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_UNIVERSITY; ?>',
        displayField:
        'name',
        valueField:
        'id',
        name: 'unid',
        hiddenName:
        'unid',
        blankText:
        '<?php echo
        SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_UNIVERSITY_BLANK; ?>',
        typeAhead:
        true,
        mode: 'local',
        forceSelection: false,
        triggerAction:
        'all',
        emptyText:
        '<?php echo
        SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_UNIVERSITY_EMPTY; ?>',
        selectOnFocus: true,
        allowBlank:
        false
    },
    {xtype: 'textarea',
name: 'description', fieldLabel: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_DESCRIPTION; ?>'},
    {xtype: 'textarea',
name: 'vision', fieldLabel: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_VISION; ?>'},

```

```

                                                                 {xtype: 'textarea',
name: 'mision', fieldLabel: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_MISION; ?>',
                                                                 {xtype: 'checkbox',
name: 'published', inputValue: '1', fieldLabel: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_GENERAL_INFO_PUBLISHED; ?>'}
                                                                 ]
                                                                 }],
                                                                 buttons: [{
                                                                 text: (id == 0) ? '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_SEND; ?>' : '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_SEND_UPDATE; ?>',
                                                                 handler: function() {
                                                                 if
(panelForm.getForm().isValid()) {
                                                                 panelForm.getForm().submit({
                                                                 method: 'POST',
                                                                 waitTitle: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_SEND_MSG_WAIT_TITLE; ?>',
                                                                 waitMsg: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_SEND_MSG_WAIT_TEXT; ?>',
                                                                 success: function(form, action) {
                                                                 Ext.Msg.show({
                                                                 title: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_SEND_MSG_SUCCESS_TITLE; ?>',
                                                                 msg: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_SEND_MSG_SUCCESS_TEXT; ?>',
                                                                 icon: Ext.MessageBox.INFO,
                                                                 buttons: Ext.MessageBox.OK,
                                                                 fn: function(btn, text) {
                                                                 if (btn == 'ok') {
                                                                 panelForm.getForm().reset();

```

```

        grid.getStore().reload();
    }
}
});

},
failure:
function(form, action) {
    if (action.failureType == 'server') {
        obj = Ext.util.JSON.decode(action.response.responseText);
        Ext.Msg.show({
            title: '<?php echo SIFEAPP_MSG_ERROR_TITLE; ?>',
            msg: obj.errors.reason,
            icon: Ext.MessageBox.ERROR,
            buttons: Ext.MessageBox.OK
        });
    } else {
        Ext.Msg.show({
            title: '<?php echo SIFEAPP_MSG_ERROR_TITLE; ?>',
            msg: '<?php echo SIFEAPP_MSG_ERROR_CONN_TEXT; ?>' +
action.response.responseText,
            icon: Ext.MessageBox.ERROR,
            buttons: Ext.MessageBox.OK
        });
    }
});
}
});
}

```

```

                                                                    } else {

                Ext.Msg.show({
                                                                    title:
'<?php echo SIFEAPP_MSG_ERROR_TITLE; ?>',
                                                                    msg:
'<?php echo SIFEAPP_PROJECT_EDIT_ADD_SEND_MSG_INVALID_TEXT; ?>',
                                                                    icon:
Ext.MessageBox.ERROR,

                buttons: Ext.MessageBox.OK
                                                                    });
                }
            }, {
                text: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_RESET; ?>',
                handler: function() {
                    Ext.Msg.confirm(
                        '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_RESET_MSG_TITLE; ?>',
                        '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_RESET_MSG_TEXT; ?>',
                        function(btn,
                                                                    if (btn
                                                                    == 'yes') {
                            panelForm.getForm().reset();
                                                                    }
                                                                    }
                                                                    );
                }
            }
        });

        // Loads data
        if (id != 0) {
            panelForm.getForm().load({
                url: '<?php echo $mosConfig-
>live_site; ?>/<?php echo
sefRelToAbs(SIFEAPP_CHRONICLES_PROJECTS_ADD_LOAD_URL); ?>',
                params: {id: id},
                method: 'POST',
                waitMsg: 'Loading',
                timeout: 10000,

```

```

failure: function(frm, action)
{
    if (action.failureType
    == 'server') {
        Ext.util.JSON.decode(action.response.responseText);
        Ext.Msg.show({
            title:
            '<?php echo SIFEAPP_MSG_ERROR_TITLE; ?>',
            msg:
            obj.errors.reason,
            icon:
            Ext.MessageBox.ERROR,
            buttons: Ext.MessageBox.OK
        });
    } else {
        Ext.Msg.show({
            title:
            '<?php echo SIFEAPP_MSG_ERROR_TITLE; ?>',
            msg:
            '<?php echo SIFEAPP_MSG_ERROR_CONN_TEXT; ?>' +
            action.response.responseText,
            icon:
            Ext.MessageBox.ERROR,
            buttons: Ext.MessageBox.OK
        });
    }
}

var panel = new Ext.Panel({
    title: (id == 0) ? '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_PROJECT; ?>' : '<?php echo
SIFEAPP_PROJECT_EDIT_UPDATE_PROJECT; ?>',
    id: 'panoptprojectadd' + id,
    layout: 'fit',
    height: '100%',
    iconCls: 'tab-optprojedit',
    closable: true,
    autoScroll: true,
    items: [panelForm],

```

```

        frame: true
    });

    return panelMain.add(panel).show();
}
else
{
    panelMain.activate('panoptprojectadd' + id);
}
}

// Members Project Grid function
function fn_prjmembers(pid) {
    <?php self::project_edit_members(); ?>
}

// Impacted Project Grid function
function fn_prjimpacted(pid) {
    <?php self::project_edit_impacted(); ?>
}

// Objectives Project Grid function
function fn_prjobjectives(pid) {
    <?php self::project_edit_objectives(); ?>
}

var filters = new Ext.grid.GridFilters({
    filters: [
        {type: 'string', dataIndex: 'unname'},
        {type: 'string', dataIndex: 'prname'},
        {type: 'string', dataIndex: 'description'},
        {type: 'numeric', dataIndex: 'members'},
        {type: 'numeric', dataIndex: 'impacted'},
        {type: 'numeric', dataIndex: 'accomobj'},
        {type: 'boolean', dataIndex: 'published'}
    ]
});

var pagingBar = new
Ext.PagingToolbar({
    pageSize: <?php echo $mosConfig_list_limit; ?>,
    store: store,
    displayInfo: true,
    displayMsg: '<?php echo
SIFEAPP_GRID_NAVIGATION_PAGING; ?>',

```

```

        emptyMsg: '<?php echo
SIFEAPP_GRID_NAVIGATION_DISPLAY; ?>',
        plugins: [filters],
        items: [
            {
                pressed: true,
                enableToggle: true,
                text: '<?php echo
SIFEAPP_PROJECT_EDIT_HEADER_DESCRIPTION_BUTTON; ?>',
                icon: '<?php echo
SIFEAPP_CHRONICLES_IMG_PRJ_SHOW_DESC; ?>',
                cls: 'x-btn-text-icon',
                toggleHandler: function(btn,
pressed) {
                    var view = grid.getView();
                    view.showDescription =
pressed;
                    view.refresh();
                }
            },
            {
                text: '<?php echo
SIFEAPP_PROJECT_EDIT_ADD_PROJECT; ?>',
                icon: '<?php echo
SIFEAPP_CHRONICLES_IMG_CMN_ADD; ?>',
                cls: 'x-btn-text-icon',
                handler: function(btn, el) {
                    fn_prjfrm(0);
                }
            },
            {
                text: '<?php echo
SIFEAPP_PROJECT_EDIT_UPDATE_PROJECT; ?>',
                icon: '<?php echo
SIFEAPP_CHRONICLES_IMG_CMN_EDIT; ?>',
                cls: 'x-btn-text-icon',
                handler: function(btn, el) {
                    var selMod =
grid.getSelectionModel();
                    if (selMod.hasSelection()) {
                        var rec =
selMod.getSelected();
                        fn_prjfrm(rec.data.cpid);
                    }
                }
            }
        ]
    }
}

```

```

        } else {
            Ext.Msg.show({
                title: '<?php
echo SIFEAPP_PROJECT_EDIT_UPDATE_GRID_SELECT_TITLE; ?>',
                msg: '<?php
echo SIFEAPP_PROJECT_EDIT_UPDATE_GRID_SELECT_TEXT; ?>',
                icon:
Ext.MessageBox.ERROR,
                buttons:
Ext.MessageBox.OK
            });
        }
    },
    '-',
    {
        text: '<?php echo
SIFEAPP_PROJECT_EDIT_DELETE_PROJECT; ?>',
        icon: '<?php echo
SIFEAPP_CHRONICLES_IMG_CMN_DELETE; ?>',
        cls: 'x-btn-text-icon',
        handler: function(btn, el) {
            var selMod =
grid.getSelectionModel();
            if (selMod.hasSelection()) {
                Ext.Msg.confirm(
                    '<?php echo
SIFEAPP_PROJECT_EDIT_DELETE_CONFIRM_MSG_TITLE; ?>',
                    '<?php echo
SIFEAPP_PROJECT_EDIT_DELETE_CONFIRM_MSG_TEXT; ?>',
                    function(btn,
                        text) {
                            if (btn
                                == 'yes') {
                                    var recs = selMod.getSelections();
                                    var arrcpid = new Array(recs.length);
                                    for (var i = 0; i < recs.length; i++) {
                                        arrcpid[i] = recs[i].data.cpid;
                                    }
                                    Ext.Ajax.request({

```

```

url: '<?php echo SIFEAPP_CHRONICLES_PROJECTS_ADD_DELETE_URL;
?>',

method: 'POST',

params: {id: arrcpid},

success: function(conn, action, options) {

    obj = Ext.util.JSON.decode(conn.responseText);

    if (obj.success) {

        Ext.Msg.show({

            title: '<?php echo
SIFEAPP_PROJECT_EDIT_DELETE_MSG_SUCCESS_TITLE; ?>',

            msg: '<?php echo
SIFEAPP_PROJECT_EDIT_DELETE_MSG_SUCCESS_TEXT; ?>',

            icon: Ext.MessageBox.INFO,

            buttons: Ext.MessageBox.OK

        });

        grid.getStore().reload();

    } else {

        Ext.Msg.show({

            title: '<?php echo SIFEAPP_MSG_ERROR_TITLE; ?>',

            msg: obj.errors.reason,

            icon: Ext.MessageBox.ERROR,

            buttons: Ext.MessageBox.OK

        });

    }
}

```



```

        return String.format('<span
class="gridProjUniversity">{0}</span>', value);
    }

    function rendProject(value, p, record)
    {
        return String.format('<span
class="gridProjProject">{0}</span>', value);
    }

    function rendMembers(value, p, record)
    {
        return String.format('<span
class="gridProjMembers">{0}</span>', value);
    }

    function rendImpacted(value, p, record)
    {
        return String.format('<span
class="gridProjImpacted">{0}</span>', value);
    }

    function rendAccomObj(value, p, record)
    {
        return String.format('<span
class="gridProjAccomObj">{0}/{1}</span>', value, record.data.totalobj);
    }

    function rendPublished(value, p, record)
    {
        var ftext = "";
        if (value == 0)
            ftext = '';
        else
            ftext = '';
        return String.format(ftext);
    }

    var grid = new Ext.grid.GridPanel({
        title: '<?php echo
_SIFEAPP_PROJECT_EDIT_GRIDTITLE; ?>',
        bbar: pagingBar,
        store: store,
        columnWidth: 1,

```

```

        height: '100%',
        layout: 'fit',
        enableColLock: false,
        loadMask: true,
        columns: [
            {header: "cuid", hidden: true,
dataIndex:'cuid', sortable: false, hideable: false},
            {header: "cpid", hidden: true,
dataIndex: 'cpid', sortable: false, hideable: false},
            {header: "<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_UNIVERSITY; ?>", width: 300, hidden: true,
dataIndex: 'unname', sortable: false, renderer: rendUniversity, tooltip: '<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_UNIVERSITY_TOOLTIP; ?>'},
            {header: "<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_PROJECT; ?>", width: 300, dataIndex:
'pname', sortable: true, renderer: rendProject, tooltip: '<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_PROJECT_TOOLTIP; ?>'},
            {header: "<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_MEMBERS; ?>", width: 35, dataIndex:
'members', sortable: true, renderer: rendMembers, tooltip: '<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_MEMBERS_TOOLTIP; ?>'},
            {header: "<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_IMPACTED; ?>", width: 35, dataIndex:
'impacted', sortable: true, renderer: rendImpacted, tooltip: '<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_IMPACTED_TOOLTIP; ?>'},
            {header: "<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_OBJECTIVES_ACCOMPLISHED; ?>",
width: 35, dataIndex: 'accomobj', sortable: true, renderer: rendAccomObj, tooltip: '<?php
echo
_SIFEAPP_PROJECT_EDIT_HEADER_OBJECTIVES_ACCOMPLISHED_TOOLTIP;
?>'},
            {header: "<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_PUBLISHED; ?>", width: 35, dataIndex:
'published', sortable: true, renderer:
rendPublished, tooltip: '<?php echo
_SIFEAPP_PROJECT_EDIT_HEADER_PUBLISHED_TOOLTIP; ?>'},
        ],
        plugins: filters,
        listeners: {
            celldblclick: function(obj, ridx, cidx,
e)
            {
                var record =
grid.getStore().getAt(ridx);
                var fieldName =
grid.getColumnModel().getDataIndex(cidx);

```

```

        if (fieldName == 'members')
        {
            fn_prjmembers(record.get('cpid'));
        }
        else if (fieldName ==
'impacted')
        {
            fn_prjimpacted(record.get('cpid'));
        }
        else if (fieldName ==
'accomobj')
        {
            fn_prjobjectives(record.get('cpid'));
        }
        else
        {
            fn_prjfrm(record.get('cpid'));
        }
    },
    //autoExpandColumn: 'descr',
    viewConfig: {
        forceFit: true,
        enableRowBody: true,
        showDescription: true,
        getRowClass: function(record,
rowIndex, p, store)
        {
            if (this.showDescription)
            {
                p.body = '<p>' +
record.data.descr + '</p>';
                return 'x-grid3-row-
expanded';
            }
            return 'x-grid3-row-
collapsed';
        }
    }
});

store.load();

```

