

Calendario General

Control de versiones del documento

Versión	Fecha	Autor	Motivo del cambio
1.0	14/05/2015	Antonio Morcillo Martínez	Creación del documento
1.1	29/05/2015	Antonio Morcillo Martínez	Cambios en la gestión de las plantillas

CALENDARIO GENERAL	1
<u>PORTLET CALENDARIO GENERAL (GA CALENDAR-PORTLET)</u>	4
CONFIGURACIÓN DEL <i>PORTLET</i>	4
ARQUITECTURA <i>FRONT-END</i> DEL <i>PORTLET</i>	4
MODELO DE CLASES	6
PLANTILLA PARA LA VISUALIZACIÓN DEL DETALLE DE UN EVENTO	7
PÁGINA DETALLE DE DÍA	7
<u>CONEXIÓN CON EL GESTOR DE CONTENIDOS DE <i>LIFERAY</i></u>	7
<u>PARÁMETROS DEL MÓDULO</u>	8
<u>DEPENDENCIAS</u>	8

Ilustraciones

ILUSTRACIÓN 1. DIAGRAMA DE COMPONENTES	5
ILUSTRACIÓN 2. MODELO DE CLASES	6

Portlet Calendario General (GA_CALENDAR-portlet)

Durante el desarrollo de los primeros portales sobre la nueva plataforma de portales temáticos sobre *Liferay v 6.2.2 GA2* se constató que el *portlet* calendario que viene de serie en *Liferay* presentaba una serie de incompatibilidades con *Internet Explorer 8* que hacían imposible su uso por parte de los usuarios.

A pesar de que *Internet Explorer 8* es un navegador relativamente antiguo y no soportado por su fabricante, el hecho de que la inmensa mayoría del parque de ordenadores de la CARM siga teniendo *Windows XP* hace que sea imprescindible disponer de un componente calendario compatible con este navegador

Después de evaluar distintos enfoques se ha optado por realizar el desarrollo de un *portlet* nuevo que permitiera mostrar eventos procedentes de distintos calendarios *Liferay* en las vistas habituales en este tipo de componentes (vista por mes, vista por semana, vista por día). El *portlet* dispondrá también de una vista detalle donde poder consultar todos los datos del evento. Además el *portlet* debe ser capaz de integrarse con el módulo de Inscripciones de la CARM para *Liferay* permitiendo la inscripción a un determinado evento cuando se den las condiciones.

El objeto del *portlet* es, por tanto, dotar a los usuarios de los distintos portales que necesiten de calendarios, de unas vistas compatibles con la mayoría de navegadores para que los eventos puedan ser consultados.

Los administradores seguirán usando el *portlet* de *Liferay* para la gestión de los distintos calendarios alojados en la plataforma.

Configuración del *portlet*

El *portlet* se configura desde el modo de edición del mismo accesible desde la opción "Preferencias" disponible para los administradores. El *portlet* dispone de dos campos para realizar la configuración de la instancia de *portlet* desplegada en una página:

- Ids calendario: Identificadores de los calendarios *Liferay* a mostrar
- Ruta plantilla: Permite especificar un catálogo de plantillas distinto a las ofrecidas por defecto para personalizar la apariencia del calendario.

Arquitectura *front-end* del *portlet*

Desde el punto de vista de *front-end* el *portlet* se apoya en el componente de calendarios *Bootstrap Calendar* (<http://bootstrap-calendar.azurewebsites.net/>) para renderizar los distintos modos de visualización de los calendarios.

El componente calendario consta de un fichero *JS*, una hoja de estilos *CSS* y un catálogo de plantillas *Underscore* para modelar la apariencia. Todos estos recursos estáticos se encuentran en el tema *Liferay CARM-Theme* y para modificarlos será necesario modificar el tema. Estos cambios en principio no tienen porqué afectar al comportamiento del calendario.

Para la vista detalle del evento se emplea un panel modal y una plantilla *Velocity*. La gestión de este modal y plantilla la realiza el *portlet* estando la plantilla alojada en el gestor de contenidos de *Liferay* y siendo por tanto editable por los administradores de la plataforma sin necesidad de recompilar el calendario.

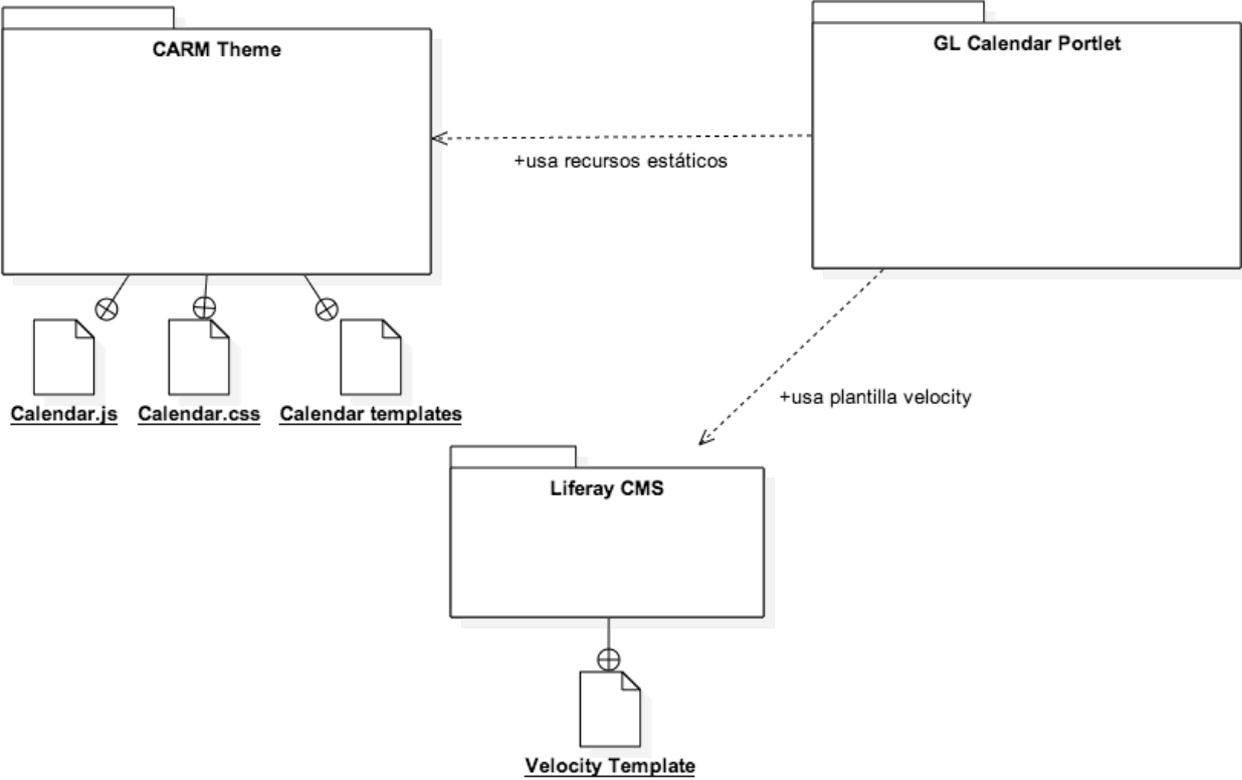


Ilustración 1. Diagrama de componentes

Modelo de clases

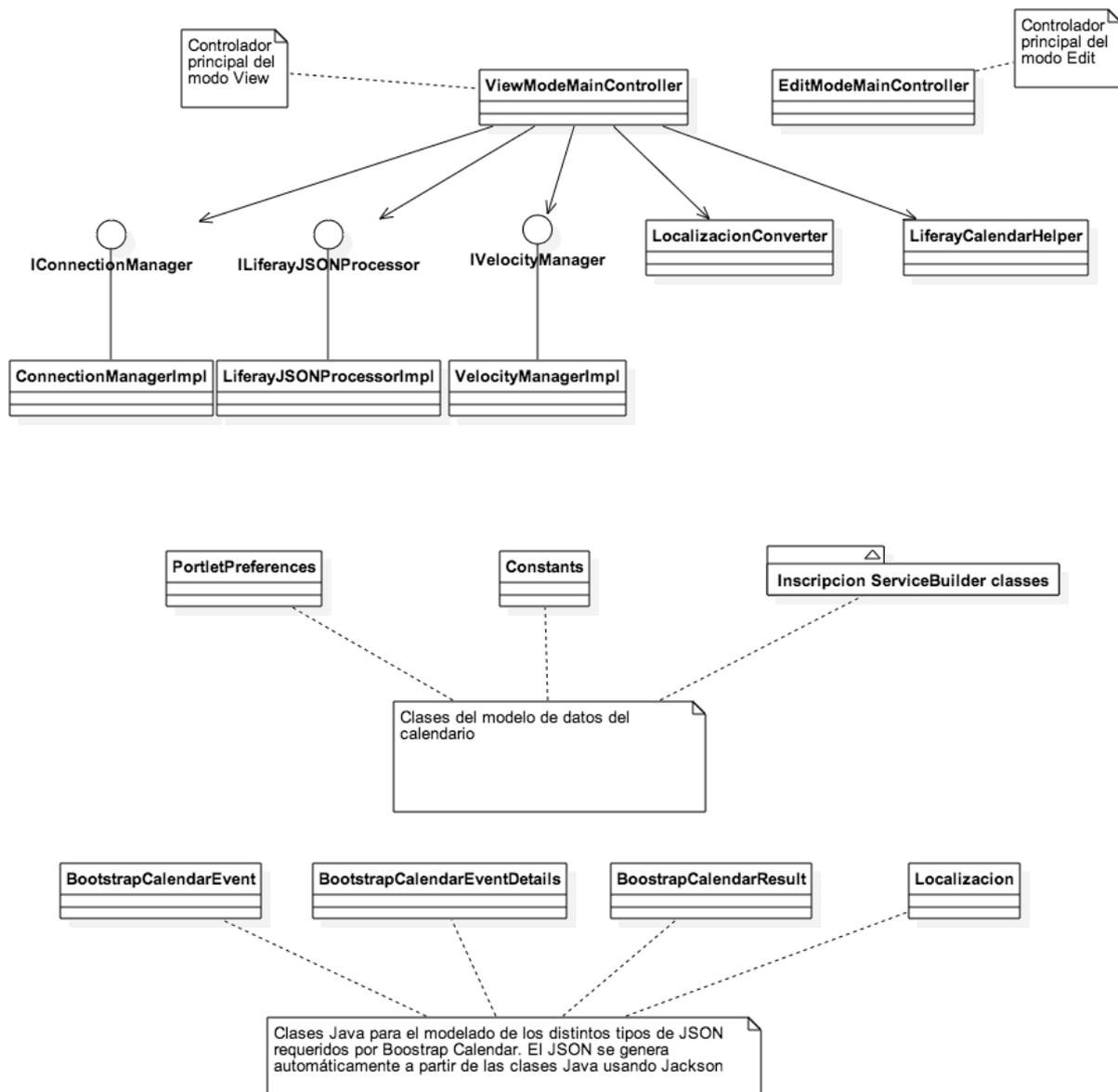


Ilustración 2. Modelo de clases

A continuación se describen las clases e interfaces más relevantes:

- **ViewModeMainController:** Punto de entrada para el calendario y sus distintas acciones. Responsable tanto de mostrar las vistas del calendario como de servir el JSON necesario por el calendario como fuente de datos de sus eventos.
- **EditModeMainController:** Punto de entrada para el modo de edición. Responsable de mostrar el formulario de edición, validarlo y guardarlo como *PortletPreferences* de la instancia de *portlet*.
- **IConnectionManager:** Interfaz que define los distintos métodos de conexión HTTP necesarios para acceder a los datos.

- **ILiferayJSONProcessor**: Define los métodos necesarios para generar los distintos formatos JSON necesarios por el calendario. Entre ellos la lista de eventos, los detalles de un evento, la respuesta de error, etc.
- **IVelocityManager**: Interfaz que define los métodos para obtener el HTML de respuesta de un detalle de calendario a partir de una plantilla velocity.
- **LocalizacionConverter**: Convierte los valores de localización a un objeto del modelo JSON listo para ser enviado al calendario
- **LiferayCalendarHelper**: Se encarga de la gestión de la inscripción a eventos del usuario autenticado
- **ConnectionManagerImpl**: Implementación de la interfaz *IConnectionManager* usando URL Connection Java estándar

Plantilla para la visualización del detalle de un evento

Los detalles de los eventos se muestran en un panel modal de *Bootstrap*. El contenido de este panel se maqueta mediante una plantilla *Velocity* que está alojada en el gestor de contenidos de *Liferay*.

El modo de operar del *portlet* para localizar la plantilla *Velocity* es el siguiente:

- Los sitios disponen de un campo personalizado '**plantillaDetalleEventos**' cuyo valor por defecto es **GL_PLT_CALENDARIO_DETALLE**
- El sitio Global del que heredan todos los sitios dispone de una plantilla *Velocity* llamada **GL_PLT_CALENDARIO_DETALLE**
- El *portlet* lee ese campo personalizado e intenta localizar la plantilla, primero en el sitio actual y, si no la encuentra, en el sitio Global.

El administrador de un sitio podría modificar este comportamiento creando plantillas para el detalle del calendario y cambiando el valor de su campo personalizado para que apunte a las nuevas plantillas.

Página Detalle de día

El *portlet* además necesita que exista una página pública en el sitio llamada '**ver-eventos**'. El calendario enviará a esta página cuando el usuario haga clic en un día.

El administrador puede configurar esta página con la disposición y *portlets* que considere necesario aunque lo habitual sería tener una página con una única columna y solamente el *portlet* calendario para que cogiese todo el espacio disponible.

Es obligatorio que en la página '**ver-eventos**' el administrador coloque un *portlet* *Calendario General* y lo configure de la misma forma que aquellos calendarios que ha dispuesto en las páginas "*normales*" del portal.

Conexión con el gestor de contenidos de *Liferay*

Al ser el calendario de *Liferay* un módulo no integrado en el core, el resto de plugins no pueden hacer uso de sus servicios de forma sencilla. No obstante el *API Rest* sí que está disponible por tanto el *portlet* accederá a los datos de eventos y calendarios haciendo conexiones HTTP al *API Rest* expuesto por el calendario.

El API requiere un usuario *liferay* puesto que es un servicio autenticado. Como este *portlet* debe ofrecerse a usuarios no autenticados, el *portlet* usará por dentro un usuario propio para autenticar sus conexiones al API del calendario. Tanto el usuario como la contraseña del mismo se encuentran parametrizados para facilitar el traspaso entre entornos. A continuación se listan todas las propiedades del Calendario.

Parámetros del módulo

- **rest-facade.auth.user**: Nombre del usuario *liferay* con el que hacer la conexión. Debe ser el mismo campo con el que esté *Liferay* configurado para autenticar. (*screenname*, *email*, etc.)
- **rest-facade.auth.password**: Password del usuario *liferay* con el que hacer la conexión
- **rest-facade.calendar.tableName**: Nombre del campo donde debe buscar los campos personalizados
- **rest-facade.calendar.columnIdentificadorLocalizacionName**: Nombre del campo personalizado
- **rest-facade.calendar.columnMaxInscritosName**: Nombre del campo personalizado
- **rest-facade.calendar.columnFechaInicioInscripcionName**: Nombre del campo personalizado
- **rest-facade.calendar.columnFechaFinInscripcionName**: Nombre del campo personalizado
- **rest-facade.calendar.columnPermiteInscripcionesName**: Nombre del campo personalizado
- **rest-facade.calendar.detailTemplateId**: Identificador de la plantilla *Velocity* con la que dibujar el detalle
- **rest-facade.url.getLocalizacion**: URL donde obtener los datos de localización del evento
- **rest-facade.url.calendarInfo**: URL donde obtener los datos de un calendario
- **rest-facade.url.getEvents**: Url donde obtener los eventos de un calendario
- **rest-facade.url.getBooking**: Url donde obtener los detalles de un evento

Dependencias

A continuación se detallan las dependencias incluidas en el *pom.xml* (más las heredadas del *pom* padre)

- **guava v 18.0**: Utilidades de Google
- **spring-mvc y spring-portlet-mvc v 4.1.0 Release**: Empleado para crear controladores REST mediante anotaciones y asociarlos a rutas de forma sencilla
- **spring-context, spring-jdbc, spring-orm y spring-tx v 4.1.0 Release**: Acceso a capa de datos por el Service Builder
- **spring-core v 4.1.0 Release**: Empleado para inyección de dependencias entre las distintas clases e interfaces del módulo
- **jackson-databind y jackson-mapper**: Empleado para generar salida JSON de forma automática a partir de clases Java
- **velocity, commons-lang y commons-codec**: Empleados para el acceso a plantillas *velocity* y codificación de la salida.

- **jstl v 1.2:** Utilidades estándar para *JSP*