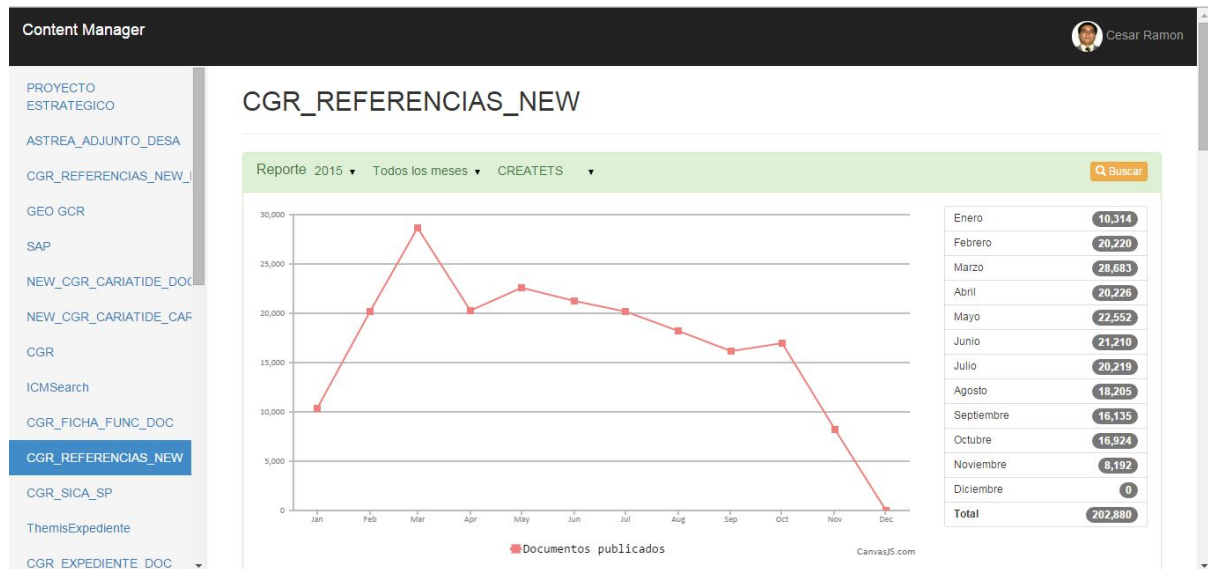


JOTA Content Manager



¿Que es JOTA Content Manager?

Jota Content Manager es un sistema desarrollado por Jhonatan Calderón que permite administrar y navegar la información de un IBM Content Manager además de proveer servicios REST para el desarrollo de aplicaciones externas.

¿Qué información se puede administrar y navegar con JCM?

JCM entrega un menú con la lista de Item Types a los que el usuario de conexión puede acceder.

Al seleccionar un Item Type se desplegará un reporte gráfico con los reportes insertados en un año específico o mes en particular a partir de la fecha de creación o algún campo tipo Date que el Item Type contenga.

Además del gráfico se obtiene el esquema referente al Item Type el cual permite seleccionar los datos que se desean obtener o por qué datos se desea filtrar.

Los datos se exportarán en una tabla con paginador indicando una columna para ordenar y la posibilidad de hacer un preview del archivo adjunto. Como funcionalidad extra permite la exportación de la información a un archivo CSV.

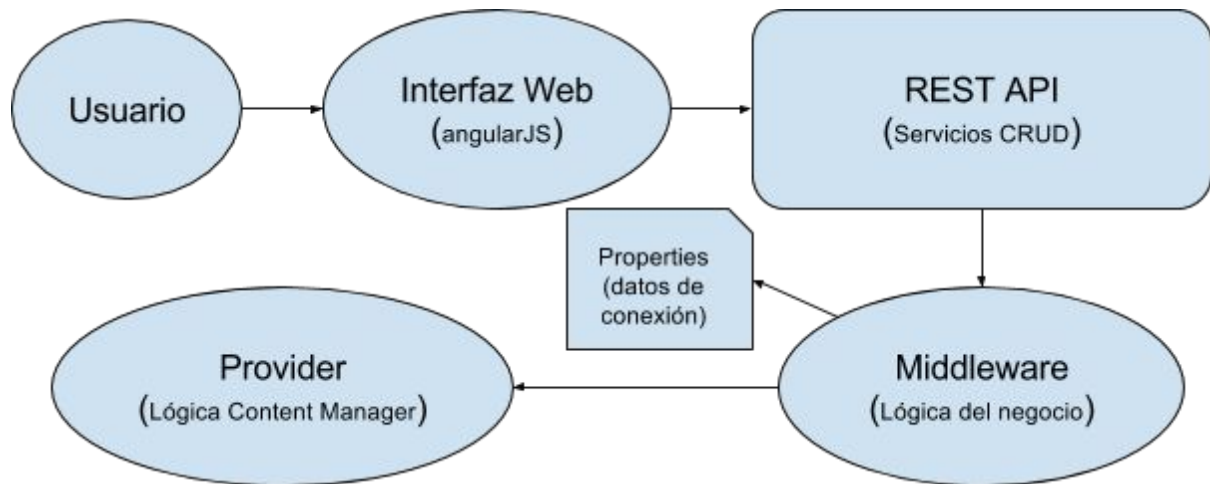
Además de la visualización de datos se entrega un formulario “tipo” que permite la inserción o modificación de un registro, validando la inserción de datos obligatorios de acuerdo al esquema obtenido. Por otro lado existe la posibilidad de adjuntar un archivo a un registro en particular indicando su Component Id o también eliminar un registro de la misma forma.

Por último se entrega una sección para realizar búsquedas masivas indicando una lista ya sea de ids o campos específicos exportando en un txt si existen o no en el IBM Content Manager. Esto es un valor agregado para la reportería y procesos de migración.

The screenshot displays the JOTA Content Manager interface with three main sections:

- Esquema (Schema):** A list of fields for an Item Type, each with a dropdown menu for selection. The fields are: Identificador Sica SP, Categoría Sica SP, Estado Sica SP, COMPONENTID, ITEMID, CREATETS, CREATEUSERID, and VERSIONID.
- Datos (Data):** A table view showing a list of records. The table has columns for #, COMPONENTID, and CREATETS. The records are numbered 1 to 10. Below the table is a pagination control showing 'Mostrando del 1 al 10 de 440.277 registros' and a 'Preview' button.
- Nuevo registro (New record):** A form for creating or editing a record. It includes fields for Identificador Sica SP, Categoría Sica SP, and Estado Sica SP, each with a dropdown menu. There is also a 'Subir archivo' (Upload file) section with a 'ComponentID' field and an 'Adjunto' (Attachment) section with a 'Seleccionar archivo' (Select file) button and a 'Ningún archivo seleccionado' (No file selected) message. A 'Guardar' (Save) button is at the bottom.

Estructura de JCM:



Datos técnicos:

La aplicación al igual que todas las aplicaciones se componen de un lado servidor y un lado cliente. El lado servidor y más importante se compone principalmente de 3 capas.

La primera capa es la “Provider” la cual contiene toda la lógica del content manager con métodos tales como connect, disconnect, getEntities, getSchema, getTotal, getData, insert, update, delete, createQuery, esta capa es la única que interactúa con la API de IBM Content Manager y por ende hace uso de sus métodos y clases.

La segunda capa es la “Middleware” la cual contiene la lógica del negocio, en este caso la lógica de JCM es servir como una aplicación genérica. Contiene los métodos que consumen la primera capa y entregan una respuesta en formato JSON parseado a un String.

La tercera capa corresponde a la “API Rest” que ofrece JCM, esta sirve de puente para cualquier aplicación externa que consuma Servicios REST y mediante la obtención de datos JSON consume el middleware. Esta capa es la que se expone vía web y es la que será consumida por un cliente, en este caso AngularJS para proveer una interfaz dinámica y genérica para cualquier IBM Content Manager.

El sistema es una base muy potente para la integración y nuevos desarrollos, 100% escalable ya que está desarrollado bajo las nuevas tendencias de arquitectura de software, mejorando la visualización, la velocidad y su transparencia en la codificación.

Herramientas usadas: Java, Java-Jersey, RestFull, Json, AngularJS, Html5, CSS3, JQuery, WebLogic, Eclipse.