# Actualización de la Arquitectura

Luego de realizar una investigación más intensiva sobre la tecnología GWT, se entendió que se debía modificar la arquitectura inicialmente propuesta para lograr una mejor performance y eliminar framework no requeridos.

El siguiente diagrama muestra la arquitectura actualizada para el sistema Tempore.

**<GWT>**

**<Client>**

Widgets – CSS – Image - JavaScript

**<Server>**

Servlet

**<Services>**

Java Class

**<DAOs>**

ORM - Hibernate

**<Base de Datos>**

Tablas

**<DTO>**

POJO Class

**<Entity>**

POJO Class Mapping

La principal modificación a la propuesta inicial fue la eliminación del framework Spring como implementador del patrón MVC.

Esta eliminación tiene como objetivo lograr una aceleración en la respuesta de la aplicación al no requerir una capa de framework tan robusta como la presta Spring. En reemplazo a esta eliminación se decide utilizar el mismo comoponte inicialmente propuesto, GWT (Google Web Toolkit),

## Responsabilidad de cada Layer

1. GWT
   1. Client

Capa responsable de la vista. Apartir de componentes como widgets, css e imágenes, es la encargada de generar las pantallas que serán visualizadas.

La comunicación con el servidor será por medio de XML-RPC

Comunicación es AJAX

EntryPoint para generación de javascript.

Browser con java script

Actualizacion de paneles.

* 1. Server

Servlets para comunicar con el back end.

Encapsulamiento bajo XML-RPC,

Facil de implementar

1. Services

Servicio encargado de la validación, lógica y transaccionabilidad de las reglas de negocio.

1. DAOs

Data acces object. Capa encargada de canalizar la persistencia y acceso único a la base de datos.

1. Base de Datos

Persistencia de la información de la aplicación.

1. Entity

Entidad utilizada por los DAOs para realizar los mapeos entre las clases Java y las tablas de la base de datos.

1. DTO

Clases Pojos que tiene la responsabilidad de comunicar el front-end con el back-end