

Arquitectura del sistema

La solución consta de dos sistemas¹ que denominaremos Sistema A (el sistema que utilizarían el grueso de los usuarios) y Sistema B (el sistema que utilizarían los administradores). El Sistema A, se implementará con balanceo de carga sobre tres réplicas del mismo utilizando *Nginx* (Figura 1). Para el desarrollo de cada uno de los sistemas (Sistema A y Sistema B) se ha optado por una Arquitectura de tres capas principales (Figura 2). A continuación se detalla cada una de las capas que componen ambos sistemas.

Capa de presentación

La primer capa es la capa de presentación de los datos. En esta capa se implementa un framework MVC por medio *Angular.js*. Esto permite paralelizar el desarrollo de la interfaz junto con el del resto de la aplicación, sin utilizar template engines ni *helpers* de HTML. Por el contrario *Angular.js* trabaja sobre HTML estático con datos de prueba obtenidos de *Node.js*. En esta capa se desarrolla la interfaz gráfica de ambos sistemas.

Capa de negocio

En esta segunda capa se desarrolla el resto de ambos sistemas. Toda esta capa se implementa sobre el framework Zend de PHP.

Se compone a su vez de las siguientes capas:

Nivel de seguridad y validación. Esta subcapa se encarga de generar y validar los *tokens* de seguridad de acceso a los servicios Web de la subcapa siguiente.

REST API: Esta subcapa provee los servicios Web del tipo REST que se utilizarán para consultas y modificaciones de la base de datos.

Servicios: Esta subcapa implementa los servicios que se proveen en la subcapa superior libre de todo control de seguridad y enfocada en la optimización de los mismos. También es en esta capa que se consultan otros servicios, como ser los de AFIP.

DAO: Es la subcapa de acceso a los datos. Implementada en el framework *Doctrine* se encarga del mapeo de objetos a la base de datos.

Esta separación en subcapas, tiene infinidad de ventajas, como ser:

- Separar la lógica de los datos, de la forma en que se presentan.
- No hay lógica de negocio en los clientes del sistemas, todo está en estas subcapas.
- Dividir apropiadamente las responsabilidades de cada capa, permitiendo un mayor y más eficiente testeo.
- Mayor posibilidad de escalabilidad. Se pueden potenciar con cambios de mejores componentes aquellas subcapas que así lo requieran, debido al bajo nivel de acoplamiento entre las mismas.

Capa de datos

En esta capa se almacenan los datos. Consiste fundamentalmente del servicio de gestión de base de datos, que en este caso se utilizará SQL Server.

¹ Documento "Alcance funcional.doc", pág.4-5

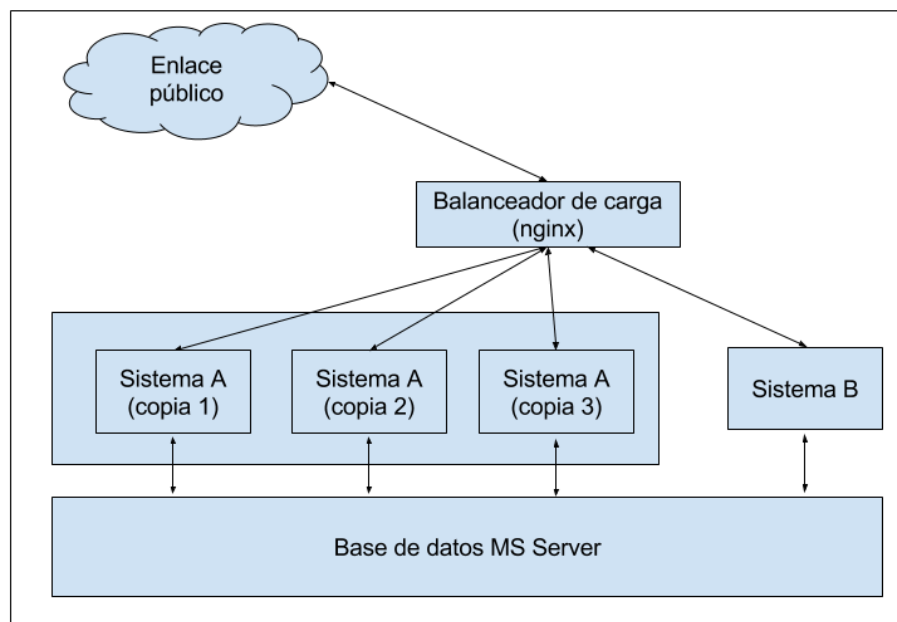


Figura 1. Arquitectura general de producción

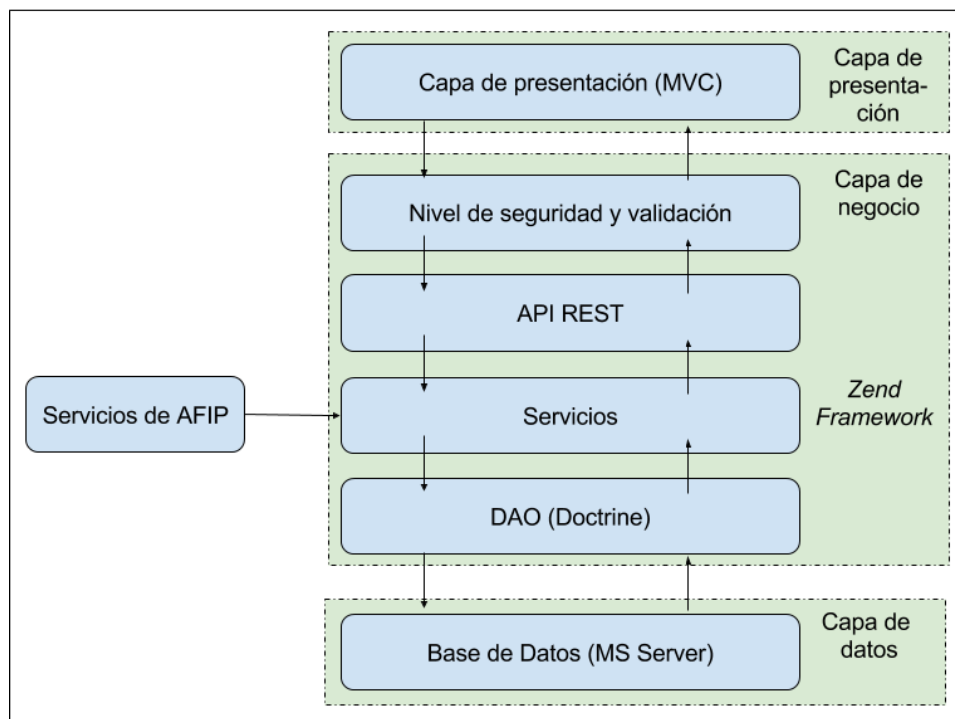


Figura 2. Arquitectura detallada por aplicación