

# GESTIÓN DE LA INFORMACIÓN DE UNA CORREDURÍA DE SEGUROS

Mireia Díaz Mansilla

Ingeniería Técnica en Informática de Gestión

Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú

Universitat Politècnica de Catalunya

## Resumen

Este proyecto final de carrera es el resultado de las distintas necesidades que surgen en el día a día, para llevar a cabo las gestiones de una pequeña correduría de seguros que se dedica al negocio de la mediación desde hace más de 30 años. La principal tarea es poder tener toda la información unificada para poder utilizarla de una manera ágil y eficiente, es ahí donde el uso de un sistema informatizado nos tiene que posibilitar gestionar toda la base de datos de nuestro cliente. Para llevar a cabo dicha labor se han analizado un conjunto de lenguajes informáticos, con la finalidad de poder escoger el más idóneo para las distintas tareas que se plantean, teniendo en cuenta la creación de un conjunto de módulos que se encargan de procesar los datos de los tomadores, clientes, productos, compañías, pólizas y recibos.

## 1. Introducción

El mercado asegurador se encuentra actualmente muy fragmentado en distintas compañías que apuestan fuertemente por todos los sistemas de la información. Las organizaciones dependen cada vez más de las tecnologías de la información para alcanzar sus objetivos corporativos, es un recurso imprescindible para cualquier empresa que quiera ofrecer a sus clientes una mayor calidad. La pequeña empresa debe poder utilizar toda la información disponible de una manera centralizada. El uso de un sistema informatizado tiene que posibilitar gestionar toda la base de datos de nuestro usuario para poderle sacar el máximo rendimiento, además los clientes son cada vez más exigentes y el mercado se encuentra en un proceso de saturación, donde el cliente dispone de más información sobre los distintos productos, es por esto que hemos creado un sistema que nos permita gestionar de manera estratégica y dinámica nuestra relación con ellos.

El motivo principal de este PFC nace a nivel personal, dada la situación actual y como alternativa a una nueva salida laboral, cuando se decide ampliar el negocio familiar, el nuevo empleado se ve limitado cuando tiene que obtener toda la información posible de sus clientes. La forma de trabajar del propietario queda obsoleta ya que no existe una base de datos unificada y el empleado no puede obtener todos los datos del cliente en el momento deseado.

Es en este punto donde interviene el concepto de CRM, la necesidad de orientarse hacia el cliente para poder establecer una relación y poder construir estrategias a nivel de marketing, almacenaje de datos, tratamiento analítico e informes para poder alcanzar los objetivos, este concepto es mucho más amplio y sólo pretendemos tenerlo como guía para introducir algunos conceptos que nos ayuden a mejorar la gestión comercial del negocio.

## 2. Objetivos

Debida a la gran abundancia de información disponible en la cartera del cliente, una de las necesidades que surgen es la de proveer al cliente de medios para manejar de manera más efectiva el gran volumen, dinamismo y complejidad de la información, el usuario se ve enfrentado con la necesidad de navegar entre un gran volumen de datos para buscar y localizar la información que necesita y que le es de importancia, esto representa frecuentemente una desorientación entre los diferentes caminos de búsqueda que debe aplicar para conseguir el objetivo. Cuando reciben llamadas o emails por parte de los clientes, generalmente no se cuenta con los mecanismos y las herramientas tecnológicas apropiadas para localizar los datos y actualizarlos, por lo que la tarea se hace más difícil de controlar, con la consecuencia de pérdida de información, de tiempo y un incremento de papel.

El objetivo es disponer de una herramienta estratégica para potenciar la eficacia de la actividad, el sistema debe de ser capaz de procesar todos los datos tanto a nivel de altas, modificaciones, eliminaciones y consultas de cualquiera de los módulos implementados, esto se consigue mediante el correcto diseño de la base de datos, por lo tanto el nuevo sistema de gestión será el primer y más importante punto de contacto para el usuario, ya que mantendrá todos los datos relevantes del negocio.

A modo de síntesis el sistema de gestión deberá cumplir con las siguientes características:

- Deberá convertirse en el punto principal de contacto, al tener toda la información relativa del cliente el usuario obtiene los datos de inmediato, sin tener que estar buscando en otras aplicaciones para saber quién es y qué relación tiene con otros tomadores.
- Otra de las principales es el registro y seguimiento de las pólizas contratadas, de esta manera se podrá emprender de una forma más estratégica enfocada al marketing.

Se recomienda una serie de pautas a la hora de trabajar con la herramienta, una primera fase sería registrar las compañías, los productos, los clientes y los tomadores en el sistema, almacenando toda la información disponible, la siguiente fase será asignar al tomador la póliza contratada, y una tercera fase será ir actualizando los recibos que dicha póliza genere, de esta forma se conseguirá recuperar la información de forma rápida, fácil y eficaz.

Un tomador será la persona que tenga alguna póliza contratada con la empresa, la idea es agrupar a los tomadores que forman un vínculo entre ellos con un número de cliente, de esta manera poder ofrecerles los servicios que más se ajusten, la compañía hace referencia a las distintas entidades aseguradoras con las que trabaja el mediador, los productos son los servicios que la empresa ofrece a los tomadores, se pueden identificar por el nombre técnico o por el nombre comercial, en un mismo nombre técnico pueden existir diferentes clases de productos comerciales, el ramo comercial representa el nombre del producto final y es el utilizado cuando se da de alta una póliza, el ramo técnico se utiliza para poder globalizar un conjunto de ramos comerciales y poder utilizar un término generalizado sobre un ramo concreto. La póliza hace referencia a los productos y compañía que el tomador tiene contratado con la empresa y el recibo servirá como constancia del pago realizado de la póliza contratada.

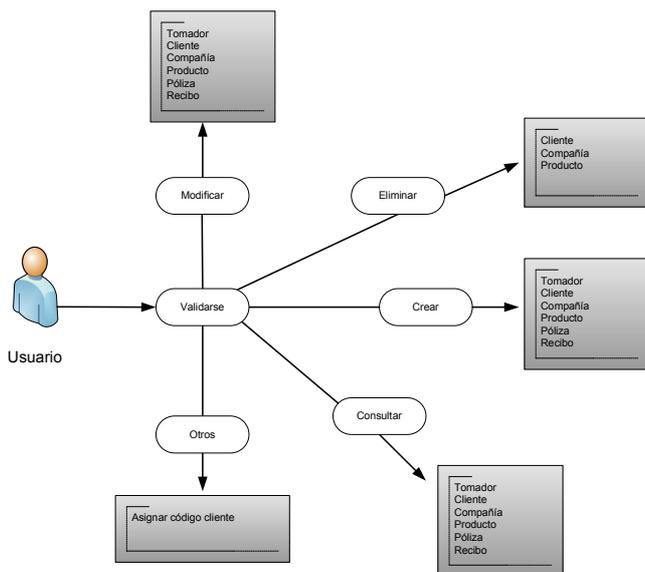


Fig. 1. Requisitos funcionales del sistema

### 3. Evaluación tecnológica

Para llevar a cabo dicha labor se han analizado un conjunto de lenguajes informáticos, con sus ventajas e inconvenientes, con la finalidad de poder escoger el más idóneo para las distintas tareas que se plantean, teniendo en cuenta la creación de un conjunto de módulos a desarrollar que se encargan de procesar los datos de los tomadores, clientes, productos, compañías, pólizas y recibos.

Los principales candidatos han sido: Java y Php, y se ha decidió programar con PHP, por las siguientes razones:

Rápido en ejecución, lenguaje multiplataforma, no requiere instalación en el cliente, conexión con la mayoría de los gestores de base de datos y fácil de aprender

Para facilitar el trabajo, se ha optado por elegir el framework Codeigniter, dicho framework nos permite desarrollar nuestra aplicación siguiendo el patrón MVC con que facilitará la organización y esquematización de nuestro proyecto, es simple, sencillo y eficaz, viene con un rango lleno de librerías que permiten realizar las tareas de desarrollo web más comúnmente necesarias, como acceder a una base de datos, mandar un email, validar datos de un formulario, etc.

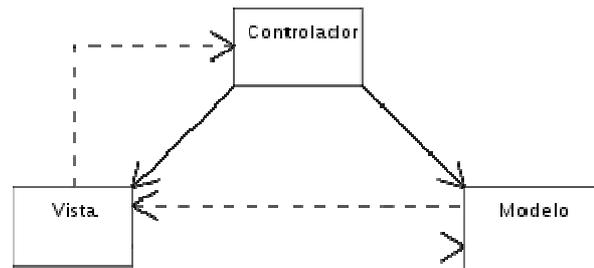


Fig. 2. Ejemplo de pie de figura

El patrón MVC (Model-View-Controller) divide una aplicación en tres partes diferenciadas:

**Model:** El model es responsable de los datos de la aplicación, y de las transacciones asociadas a ellos. Es decir, la lógica de negocio.

**View:** El view es responsable de visualizar los datos. Es decir, la construcción de páginas.

**Controller:** Es responsable de desacoplar las interacciones entre el model y el mundo exterior. Es decir, controla las interacciones y no necesita saber nada de cómo funciona el componente view.

Se ha utilizado el lenguaje HTML (HyperText Markup Language) para dar cierto estilo y hacer más atractiva visualmente la aplicación web, se ha hecho uso de CSS (Cascading Style Sheets), tanto XHTML como CSS se han usado principalmente dentro de Codeigniter en la zona de las Vistas, el uso de CSS nos ha facilitado el diseño de la vista, ya que indicando la referencia a la hoja de estilo todas las pantallas se ajustaban al mismo diseño.

Cuando se han introducido fechas en el sistema se ha utilizado funciones de Javascript.

Nuestra lógica de negocio está compuesta por :

**usuario:** módulo encargado de gestionar la entrada del sistema y de comprobar las credenciales de los usuarios que acceden al sistema.

**tomador:** módulo encargado de gestionar todos los temas relacionados con los tomadores (listado, recuperar datos, añadir nuevos, modificar, y eliminar).

**cliente:** módulo encargado de gestionar todos los temas relacionados con los clientes (listado, recuperar datos, añadir nuevos, modificar, y eliminar).

compañía: módulo encargado de gestionar todos los temas relacionados con las compañías (listado, recuperar datos, añadir nuevos, modificar, y eliminar).

producto: módulo encargado de gestionar todos los temas relacionados con los productos (listado, recuperar datos, añadir nuevos, modificar, y eliminar).

ramo: módulo encargado de gestionar todos los temas relacionados con el ramo comercial y el ramo técnico. (listado, recuperar datos).

póliza: módulo encargado de gestionar todos los temas relacionados con las pólizas (listado, recuperar datos, añadir nuevos, modificar, y eliminar).

recibo: módulo encargado de gestionar todos los temas relacionados con los recibos (listado, recuperar datos, añadir nuevos y modificar).

Para la elección del sistema de gestión de bases de datos, se han descartado los que son de pago como Oracle y Microsoft SQL, ya que la empresa no desea hacer ningún tipo de inversión, y después de analizar un conjunto de SGBD gratuitos, por todas las características de unos y otros los principales candidatos han sido MySQL y PostgreSQL.

Se decide escoger MySQL, tiene acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones, potente para las consultas de bases de datos, las consultas hechas con SQL son fácilmente portables a otros sistemas y plataformas, velocidad al realizar las operaciones, seguridad en cuanto a los permisos y privilegios, facilidad configuración e instalación, bajo costo en requerimientos para la elaboración de base de datos y junto con Apache y PHP forman un buen equipo para servir páginas web con contenido dinámico.

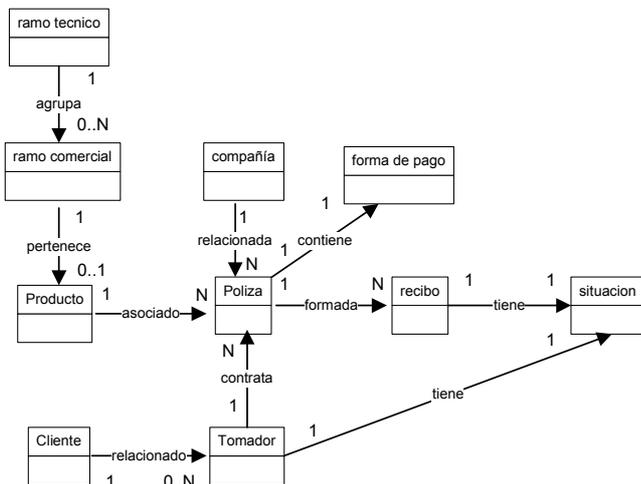


Fig. 2. Modelo conceptual del sistema

Para el servidor web se decide incorporar Apache porque tiene un alto rendimiento con PHP, estabilidad en la red, y mucha documentación. Para la instalación del servidor web se ha utilizado un paquete XAMPP, es una forma fácil de instalar la distribución Apache que contiene MySQL, PHP y Perl.

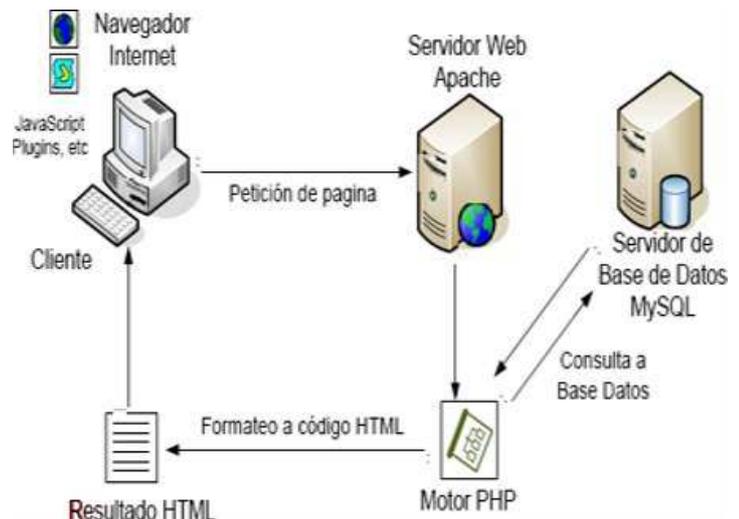


Fig. 3. Esquema servidor web

#### 4. Diseño Web

El diseño como página web será la división de la pantalla en distintas secciones que nos permitiría navegar por la aplicación.

A la hora de decidir cómo íbamos a mostrar y organizar la información se ha tenido en cuenta que el usuario no está familiarizado con sistemas informáticos por lo que la aplicación, debe ser sencilla, intuitiva y controlable.



Fig. 4. Página principal

Una de las fases en el desarrollo de la aplicación ha sido la evaluación de la usabilidad. No sólo se ha evaluado su funcionamiento, sino que también ha sido necesario comprobar si la aplicación satisface las necesidades del usuario.

- Se ha pensado en qué modo el usuario desearía encontrar la información.
- Se ha tenido en cuenta las limitaciones de los recursos del usuario a nivel de ancho de banda, capacidad de procesamiento y resolución de pantalla.
- Se ha diseñado un menú sencillo.



Fig.5. Listado productos

En la figura 5 se puede apreciar las distintas secciones en las que queda dividida la aplicación. En ella se pueden distinguir cuatro partes diferenciadas. En la parte superior e inferior se aprecian la cabecera y el pie respectivamente y la parte central dividida en dos secciones, el menú y el contenido. En la cabecera se muestra la fecha actual de esta manera se consigue dar una sensación de actualidad.

En el centro tendremos el listado de los diferentes módulos desarrollados, y a su derecha las acciones que podemos realizar, ver los datos detallados, modificar o eliminar la póliza, con un solo clic el usuario puede acceder rápidamente a la solución que necesita. Dentro del menú se dan opciones diferentes, como volver a listar, hacer una búsqueda o añadir un nuevo ítem.

Fig.6. Búsqueda póliza

La Fig.6 nos muestra el formulario que se aplica al realizar la búsqueda de la póliza, se ha considerado definir varios campos de búsqueda para poder facilitar al usuario la localización de la póliza.

Fig.6. Añadir póliza

La Fig.6 nos muestra el formulario que se aplica al realizar la nueva alta de la póliza, para facilitar la introducción de los nuevos datos, se ha decidido cargar los datos en combos, para que el usuario pueda escoger entre los datos del sistema.

## 5. Conclusiones

Los objetivos principales de este proyecto han sido desarrollados con éxito, ya que se ha diseñado e implementado el sistema para gestionar todas las relaciones existentes.

Gracias al patrón MVC por la división de las capas y la relación entre ellas ha sido más fácil de asumir los conceptos de la web, la curva de aprendizaje que supone trabajar con nuevas tecnologías ha sido muy elevada, pero una vez adquirido los conocimientos la implementación ha sido mucho más sencilla.

Otros de los aspectos que se ha podido valorar es la importancia de realizar todos los pasos en el proyecto, ya que se ha podido constatar que cualquier olvido o error en una etapa anterior, ha supuesto la necesidad de modificar cosas y volver atrás en todo momento. Si bien es cierto que a priori no siempre es posible tener clara la visión final del sistema, es importante que ya desde el principio ésta sea lo más certera posible, ya que de este modo se evitan correcciones y pérdida de tiempo, lo cual viene a demostrar una vez más, que es mejor perder un poco de tiempo en aspectos fundamentales como son el análisis de requisitos o el modelado de objetos, pues en conjunto siempre se acaba ganando.

Por último concluir que la aplicación en su estado actual no se puede considerar como acabada porque puede tener numerosas ampliaciones, siempre hay algo más que se puede añadir para aumentar su funcionalidad, por ejemplo módulo de clientes potenciales, módulo de estadísticas o módulo de siniestros.

## Referencias

- [1] Manual del Mediador de Seguros (Cecas Módulo 3).
- [2] <http://es.wikipedia.org/wiki/PHP>
- [3] <http://ellislab.com/codeigniter/user-guide/installation/downloads.html>
- [4] [http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)
- [5] <http://www.apachefriends.org/es/xampp.html>