

**E-NATURAL: DESARROLLO DE UNA APLICACIÓN  
ORIENTADA A LA GESTIÓN Y ADMINISTRACIÓN  
DE LOS SERVICIOS DE EMPRESAS RELACIONADAS  
CON ACTIVIDADES EN LA NATURALEZA**

**PARTE I**



**PROYECTO FIN DE GRADO**

**Escuela de Ingeniería Informática**

**Universidad de Las Palmas de Gran Canaria**

**JUNIO 2014**

**Autor:** Yeray Pérez Valiente

**Tutor1:** Agustín Sánchez Medina

**Tutor2:** Mónica Pellejero Silva

**Titulación:** Grado en Ingeniería en Informática





Proyecto fin de grado de la Escuela de Ingeniería Informática de la Universidad de Las Palmas de Gran Canaria presentado por el alumno:

**Yeray Pérez Valiente**

Titulo del proyecto:

**E-Natural: Desarrollo de una aplicación orientada a la gestión y administración de los servicios de empresas relacionadas con actividades en la naturaleza. Parte I**

Tutor1:

**Agustín Sanchez Medina**

Tutor2:

**Mónica Pellejero Silva**

## **DEDICATORIA**

A mis padres y a mi novia



## Índice general

1	Introducción .....	5
2	Estado actual y Objetivos .....	6
2.1	Estado actual del Arte .....	7
3	Competencias.....	11
4	Aportaciones Socio-Económicas .....	23
5	Planificación del trabajo.....	24
5.1	Metodología de desarrollo.....	24
5.2	Planificación y temporización .....	26
6	Desarrollo del trabajo.....	28
6.1	Requisitos del sistema.....	28
6.1.1	Modelo del dominio .....	28
6.1.2	Diagramas modelo de dominio .....	28
6.1.3	Diagramas entidad relación .....	30
6.1.4	Modelo relacional .....	34
6.2	Especificación de requisitos de usuario. ....	37
6.2.1	Descripción.....	37
6.2.2	Actores .....	38
6.2.3	Listados de actores y sus roles .....	39
6.2.4	Modelo de casos de uso.....	42
6.2.5	Lista de Casos de Uso .....	47
6.2.6	Prototipo de interfaz de usuario .....	48
6.3	Diseño arquitectónico .....	52
6.3.1	Características de esta arquitectura .....	52
6.3.2	Arquitectura de la aplicación .....	54
6.3.3	Detalles de la implementación.....	57
6.3.4	Diagrama de Despliegue.....	59
6.3.5	Diagramas de clases .....	61
6.3.6	Diagramas de secuencia.....	86
7	Conclusiones.....	91
7.1	Uso del Framework Codeigniter.....	91
7.2	Uso de Librería Grocery CRUD .....	91
7.3	Uso de componentes, tecnologías y estrategias.....	91
7.4	Resultado final del proyecto .....	92

8	Trabajo Futuro:.....	92
9	Anexos.....	94
9.1	Anexo I: Recursos utilizados.....	94
9.1.1	Recursos software.....	94
9.1.2	Recursos hardware.....	101
9.2	Anexo II: Especificación de casos de uso.....	102
9.3	Anexo III: Implementación.....	133
9.3.1	TinyMce.....	133
9.3.2	Codeigniter:.....	134
9.3.3	Paypal.....	177
9.3.4	Grocery Crud.....	190
9.3.5	Image Crud.....	204
9.3.6	Image Moo.....	206
9.3.7	JRating.....	210
9.4	Anexo IV: Manual de usuario.....	223
9.5	Anexo V: Índice de Imágenes.....	263
9.6	Anexo VI: Índice de figuras completo.....	266
9.7	Anexo VII: Índice de tablas completo.....	268
10	Bibliografía.....	269







# 1 Introducción

Un **sistema de información** es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad u objetivo. Todos estos elementos interactúan para procesar los datos (incluidos los procesos manuales y automáticos) y dan lugar a información más elaborada, que se distribuye de la manera más adecuada posible en una determinada organización, en función de sus objetivos.

Puede ser considerado como el uso de la tecnología de la información para respaldar o dar forma a la estrategia competitiva de la organización, a su plan para incrementar o mantener la ventaja competitiva o bien para reducir la ventaja de sus competidores.

Su función primordial es crear una diferencia con respecto a los competidores de la organización (o salvar dicha diferencia) que hagan más atractiva a ésta para los potenciales clientes. De este modo se pretende lograr ventajas que los competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. Apoyan el proceso de innovación de productos dentro de la empresa. Estos sistemas cambian significativamente el desempeño de un negocio al medirse por uno o más indicadores clave, entre ellos, la magnitud del impacto. Contribuyen al logro de una meta estratégica. Generan cambios fundamentales en la forma de dirigir una compañía, la forma en que compite o en la que interactúa con clientes y proveedores.

Un **sistema de gestión de reservas (CRS, Computerised Reservation System)** de alojamiento y actividades, permite a las empresas comercializar sus productos y servicios. Contiene información de horarios y tarifas además de una base de datos de reservas.

Los **sistema de gestión de contenido (CMS, Content Management System)**, son aplicaciones informáticas usadas para crear, editar, gestionar y publicar contenido digital. El gestor de contenidos genera páginas web dinámicas interactuando con el servidor web para generar la página web bajo petición del usuario, con el formato predefinido y el contenido extraído de la base de datos del servidor.

El **comercio electrónico (e-commerce, electronic commerce)**, consiste en la compra y venta de productos o de servicios a través de medios electrónicos, tales como Internet y otras redes informáticas.

La cantidad de comercio llevada a cabo electrónicamente ha crecido de manera extraordinaria debido a Internet. Una gran variedad de comercio se realiza de esta manera, estimulando la creación y utilización de innovaciones como la transferencia de fondos electrónica, la administración de cadenas de suministro, el marketing en Internet, el procesamiento de transacciones en línea (OLTP), el intercambio electrónico de datos (EDI), los sistemas de administración del inventario y los sistemas automatizados de recolección de datos.

La mayor parte del comercio electrónico consiste en la compra y venta de productos o servicios entre personas y empresas.

En este proyecto se va a desarrollar un sistema de información de apoyo a las empresas que dan servicios de actividades en la naturaleza, con especial atención a los que se dedican al alojamiento.

Para apoyar a este sistema se implementarán distintas herramientas dentro de la plataforma para poder realizar reservas y ventas de productos y servicios. Con el objeto de realizar las ventas las ofertas estarán centralizadas y además estas contarán con un escaparate propio para anunciar sus productos y servicios.

Para afrontar esta problemática se diseñará una herramienta que permite la creación de un hosting web donde anunciar, gestionar reservas, realizar compras y obtener información acerca de los productos de las empresas. Cabe aclarar que el portal diseñado tendrá una entrada única que permitirá acceder a las distintas empresas. Además, es de destacar que se implementarán herramientas que permiten interactuar a los clientes, mediante valoraciones y comentarios de los usuarios del sistema.

En lo que respecta al seguimiento de las actividades de los clientes, se dispondrá de un sistema de información basado en estadísticas propias, para analizar de manera precisa las preferencias de consumo y el tráfico en la web de cada empresa.

La información contenida en el sistema web, se basará en la información proporcionada por las propias empresas alojadas, esto permitirá distribuir y publicitar sus propios productos y servicios y darlos a conocer a los usuarios visitantes.

## 2 Estado actual y Objetivos

El sector turístico de las actividades en la naturaleza, ha experimentado en los últimos años una gran expansión. Así al haberse incrementado, la demanda cada vez más de actividades y servicios de este tipo, ha aparecido una mayor oferta de empresas especializadas en este sector de actividad.

Como consecuencia del mencionado crecimiento, este sector es una muy buena oportunidad de negocio. Además, las empresas dedicadas a esta actividad generalmente no disponen de los medios informáticos para establecerse de una forma profesional en la Web. Así, los principales problemas a los que se enfrentan estas empresas son el desconocimiento de la promoción en Internet y los motivos económicos. Por esta razón nace la idea de crear un servicio de gestión de empresas que prestan como servicio las actividades en la naturaleza y el alojamiento rural.

Los objetivos marcados de esta aplicación web son:

- Proporcionar un servicio de hosting web económico.
- Proporcionar una herramienta de gestión de productos.
- Proporcionar un escaparate para anunciar los productos de las empresas
- Centralizar la actividad del sector en un mismo siteweb.

## 2.1 Estado actual del Arte

En la actualidad encontramos portales con una oferta de productos y servicios entorno al turismo rural. No obstante, se limitan a la búsqueda de servicios de alojamiento rural, o de actividades relacionadas, comparativa de productos y redireccionamiento a otras webs particulares si las hubiese.

A continuación mostramos las siguientes web de ejemplo:

### Escapada rural:

Escapada rural es una web que se dedica a la búsqueda de alojamientos de casas rurales. En la parte de usuario ofrece un buscador de alojamientos en España. La posibilidad de crear un listado con tus alojamientos favoritos y además ponerles calificaciones.

En la parte de los propietarios dan la posibilidad de anunciar los alojamientos en la web mediante un servicio básico siempre gratuito. Además ofrece un servicio de pago para poder gestionar las reservas (Véase Imagen 1).



Imagen 1: Página web Escapada Rural

## Toprural:

Toprural ofrece a los usuarios un buscador de alojamientos rurales, publicar opiniones y suscripciones a un boletín mensual con ofertas.

Para los propietarios ofrecen la posibilidad de promocionar sus alojamientos y un seguimiento mediante estadísticas de la actividad relevante y la posibilidad de suscribirse a un boletín oficial solo para propietarios (Véase Imagen 2).

The screenshot shows the Toprural website interface. At the top left is the logo "Toprural" with the tagline "Nos apasiona el turismo rural". To the right are navigation links: "Guía de Opiniones", "Español", and "Acceso usuarios". A prominent orange button says "¡Anuncia tu casa!". Below the header is the main heading "Casas rurales, turismo rural". A search form includes a text input for "Destino o nombre de alojamiento:", radio buttons for "Alquiler completo" and "Alquiler por habitaciones", and a "Número de personas:" field. A "Buscar" button is at the bottom right of the form. Below the form is a map of Europe with the text "Otra forma de buscar casa rural: selecciona un país en el mapa". On the right side, under "Alojamientos Destacados", there are four featured listings with photos and titles: "Posada Santa Eulalia" (Cantabria, España), "Apartamentos Rurales San Martín" (Asturias, España), "La Oveja Verde" (Andalucía, España), and "Mas Garriga" (Cataluña, España). A "Publicidad" label is at the bottom of the featured listings section.

Imagen 2: Página web Top Rural

## Tripadvisor:

La web de viajes más grandes del mundo ya no sólo se dedica a mostrar las opiniones de los usuarios en los hoteles, ahora también tiene un directorio de actividades de los lugares. Los usuarios y negocios pueden crear actividades de los sitios y poner su opinión. Estas actividades son localizadas geográficamente y revisadas por los gestores de Tripadvisor para validar su autenticidad. Las actividades creadas son comentadas y valoradas por los usuarios. Estos además pueden añadir sus fotos referentes a la actividad (Véase Imagen 3).



Imagen 3: Página web Tripadvisor

### Canarying:

Es una empresa a nivel local que ofrece todo tipos de ofertas de servicios relacionados con deportes de aventura, astroturismo, casas rurales y turismo cultural de empresas ajenas. Ofrece la posibilidad de realizar reservas (Véase Imagen 4).



Imagen 4: Página web Canarying

## Yumping:

Jumping es una empresa que ofrece productos de ocio y deportes de aventura. Mediante un buscador de actividades y una localización se pueden encontrar los diferentes servicios ofrecidos. También tiene la posibilidad de realizar reservas y hacer comentarios (Véase Imagen 5).

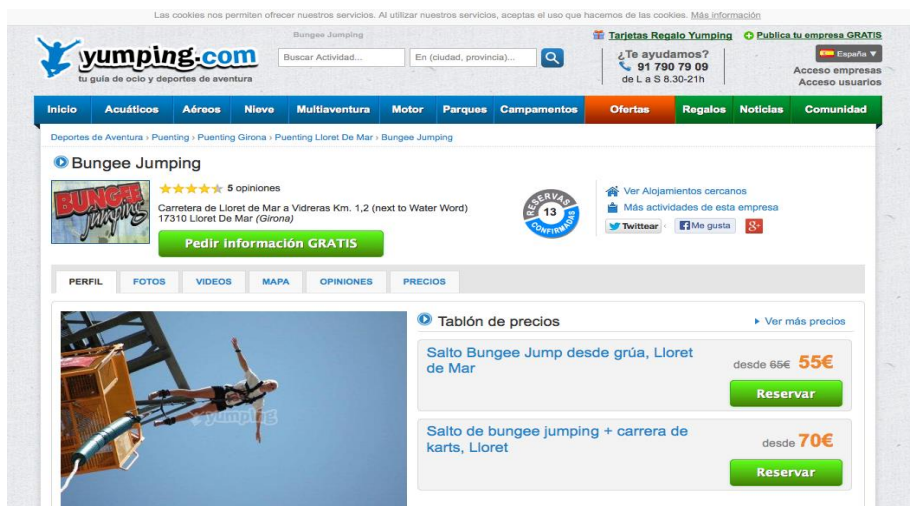


Imagen 5: Página web Yumping

## Daventuras:

Es un buscador de empresas de deportes de aventura y deportes extremos. Ofrecen turismo de aventura, campamentos y casas rurales. Solo realiza búsquedas y enlaza con las páginas webs de las empresas (Véase Imagen 6).



Imagen 6: Página web DAventuras



Enatural en cambio, además de los servicios anteriormente mencionados, proporciona un servicio de gestión de información completo para la creación, mantenimiento y hosting de sitios web para las empresas alojadas.

### 3 Competencias

En este apartado se mostrarán las distintas competencias que se pretenden cubrir con el presente proyecto:

CII01

***Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación vigente.***

Esta competencia queda cubierta con los capítulos Estado actual y objetivo del Trabajo de Fin de Grado y Requisitos Hardware y Software. En ellos se especifican, por un lado, los motivos de diseño, desarrollo, etc. explicando exhaustivamente las fases de definición de requisitos, análisis y diseño de los casos de uso necesarios para satisfacer las necesidades de la aplicación Web, y por otro lado se justifica la utilización del hardware más adecuado para la implementación de la misma, teniendo en cuenta parámetros de fiabilidad, calidad y seguridad.

En resumen, con esta elección de hardware y software, la implementación de este trabajo de fin de grado da como resultado una aplicación robusta y conforme a los principios éticos y a la legislación vigente.

CII02

***Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.***

De todos los elementos de un sistema de gestión de proyectos, la planificación es posiblemente la más importante, puesto que una mala o inexistente planificación conducirá a una mala realización en el proyecto, lo cual repercutirá enormemente tanto en la calidad como en el plazo de terminación del proyecto, ocasionando grandes perjuicios. Esta es la razón por la que esta competencia queda cubierta, puesto que la buena planificación de este trabajo de fin de grado ha dado lugar al propósito fundamental del mismo, que no es más que la finalización en los plazos establecidos así como conseguir de manera adecuada los objetivos propuestos inicialmente.

Si además tenemos en cuenta el capítulo Aportaciones, podemos apreciar el impacto del proyecto en el ámbito de aplicación en que nació. Hay que tener en cuenta que los objetivos que siempre se buscan al desarrollar una aplicación no deben limitarse a resolver el problema sino que hay que intentar elegir la solución más apropiada y conforme a la infraestructura con la que cuenta el cliente.

CII04

***Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.***

Pliego de especificaciones técnicas que han de regir en la contratación de una plataforma compuesta por todo el desarrollo de la plataforma web y todo lo relacionado con su implementación.

#### Objeto del contrato y condiciones

El contrato tiene como objeto, la implementación de la plataforma web dinámica VideoEduca para uso de la ULPGC.

#### Ámbito geográfico

El ámbito geográfico en que se prestará el servicio será mundial. Ya cualquier visitante externo puede acceder a la plataforma, pero sólo los alumnos matriculados en la UlpGC podrán disfrutar de la totalidad de sus servicios.

#### Requisitos de fiabilidad

El desarrollo de la aplicación debe acometerse de tal forma que su resultado garantice:

- Alta fiabilidad
- Comportamiento estable

#### Plazo de ejecución

El contratista se compromete a elaborar el software de conformidad con lo establecido en este documento y con la oferta que hubiese presentado, así como a entregarlo al contratante. Dicha entrega deberá hacerse en el plazo máximo de 6 meses, a contar desde la fecha de la firma del contrato.

El software se considerará debidamente entregado cuando se haya recibido la plataforma y a su vez esté disponible para su inmediato funcionamiento y se haya facilitado el manual de usuario y la documentación técnica correspondiente.

#### Documentación

Las empresas licitadoras deberán poner claramente de manifiesto las posibles actualizaciones de la aplicación.

#### Transferencia Tecnológica

Durante la ejecución de los trabajos objeto del contrato, el adjudicatario se compromete a facilitar en todo momento a las personas designadas por la Universidad de las Palmas de Gran Canaria a tales efectos la información que ésta solicite para disponer de un pleno conocimiento de las circunstancias en que se desarrollan los trabajos, así como de los eventuales problemas que pueden plantearse y de las tecnologías, métodos y herramientas utilizados para resolverlos.

#### Propiedad del resultado de los trabajos

Todos los documentos y resultados de los trabajos realizados serán propiedad de la Universidad de las Palmas de Gran Canaria que podrá ejercer el derecho de explotación en su centro de trabajo.

La empresa adjudicataria podrá hacer uso de los mismos, ya sea como referencia o base para trabajos futuros, siempre que cuente con la autorización expresa del contratante.

#### Entrega de la solución

Todo el proceso de suministro de software, documentación, manuales de instalación y configuración, manual de administración y de usuario, desarrollo de pruebas, plan de marcha atrás, se hará conforme al Proceso de Gestión de la Entrega definido por la Dirección de la Universidad de las Palmas de Gran Canaria.

#### Entorno Pre-explotación

De ser necesarias tareas correctivas una vez implementada la solución objeto de este contrato, con el fin de facilitar los trabajos, y realización de pruebas sin afectar al sistema en producción, será preciso que cualquier actividad a realizar sobre la plataforma se realice previamente en el entorno de pre-explotación similar al de producción.

Será preciso verificar y testear el correcto funcionamiento en este entorno, como paso previo al despliegue en producción.

CII18

#### **Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.**

Un análisis de las legislaciones que se han promulgado en diversos países arroja que las normas jurídicas que se han puesto en vigor están dirigidas a proteger la utilización abusiva de la información reunida y procesada mediante el uso de computadoras, e incluso en algunas de ellas se ha previsto formar órganos especializados que protejan los derechos de los ciudadanos amenazados por los ordenadores.

Desde hace aproximadamente diez años la mayoría de los países europeos han hecho todo lo posible para incluir dentro de la ley, la conducta punible penalmente, como el acceso ilegal a sistemas de cómputo o el mantenimiento ilegal de tales accesos, la difusión de virus o la interceptación de mensajes informáticos.

En la mayoría de las naciones occidentales existen normas similares a los países europeos. Todos estos enfoques están inspirados por la misma de preocupación de contar con comunicaciones electrónicas, transacciones e intercambios tan confiables y seguros como sea posible.

Dar un concepto sobre delitos informáticos no una labor fácil y esto en razón de que su misma denominación alude a una situación muy especial, ya que para hablar de “delitos” en el sentido de acciones tipificadas o contempladas en textos jurídico penales, se requiere que la expresión “delitos informáticos” este consignada en los códigos penales, lo cual en nuestro

país, al igual que en muchos otros, no ha sido objeto de tipificación aún; sin embargo, muchos especialistas en derecho informático emplean esta alusión a los efectos de una mejor conceptualización.

Normativa y regulación de la informática en el ámbito internacional.

En el contexto internacional, son pocos los países que cuentan con una legislación apropiada. Entre ellos, destacan, Estados Unidos, Alemania, Austria, Gran Bretaña, Holanda, Francia, España, Argentina y Chile.

Por esta razón a continuación se mencionan algunos aspectos relacionados con la ley en los diferentes países, así como con los delitos informáticos que persigue.

Alemania

Este país sancionó en 1986 la Ley contra la Criminalidad Económica, que contempla los siguientes delitos:

Espionaje de datos.

Estafa informática.

Alteración de datos.

Sabotaje informático.

Austria

La Ley de reforma del Código Penal, sancionada el 22 de Diciembre de 1987, sanciona a aquellos que con dolo causen un perjuicio patrimonial a un tercero influyendo en el resultado de una elaboración de datos automática a través de la confección del programa, por la introducción, cancelación o alteración de datos o por actuar sobre el curso del procesamiento de datos.

Además contempla sanciones para quienes comenten este hecho utilizando su profesión de especialistas en sistemas.

Estados Unidos

Este país adoptó en 1994 el Acta Federal de Abuso Computacional que modificó al Acta de Fraude y Abuso Computacional de 1986 con la finalidad de eliminar los argumentos hipertécnicos acerca de qué es y que no es un virus, un gusano, un caballo de Troya y en que difieren de los virus, la nueva acta proscribire la transmisión de un programa, información, códigos o comandos que causan daños a la computadora, a los sistemas informáticos, a las redes, información, datos o programas. La nueva ley es un adelanto porque está directamente en contra de los actos de transmisión de virus.

Asimismo, en materia de estafas electrónicas, defraudaciones y otros actos dolorosos relacionados con los dispositivos de acceso a sistemas informáticos, la legislación

estadounidense sanciona con pena de prisión y multa, a la persona que defraude a otro mediante la utilización de una computadora o red informática.

En el mes de Julio del año 2000, el Senado y la Cámara de Representantes de este país, tras un año largo de deliberaciones, establece el Acta de Firmas Electrónicas en el Comercio Global y Nacional. La ley sobre la firma digital responde a la necesidad de dar validez a documentos informáticos, mensajes electrónicos y contratos establecidos mediante Internet, entre empresas (para el B2B) y entre empresas y consumidores (para el B2C).

#### Chile

Chile fue el primer país latinoamericano en sancionar una Ley contra delitos informáticos, la cual entró en vigencia el 7 de junio de 1993. Esta ley se refiere a los siguientes delitos:

La destrucción o inutilización de los de los datos contenidos dentro de una computadora es castigada con penas de prisión. Asimismo, dentro de esas consideraciones se encuentran los virus.

Conducta maliciosa tendiente a la destrucción o inutilización de un sistema de tratamiento de información o de sus partes componentes o que dicha conducta.

#### Francia

En enero de 1988, este país dictó la Ley relativa al fraude informático, en la que se consideran aspectos como:

Intrusión fraudulenta que suprima o modifique datos.

Conducta intencional en la violación de derechos a terceros que haya impedido o alterado el funcionamiento de un sistema de procesamiento automatizado de datos.

Conducta intencional en la violación de derechos a terceros, en forma directa o indirecta, en la introducción de datos en un sistema de procesamiento automatizado o la supresión o modificación de los datos que éste contiene, o sus modos de procesamiento o de transmisión.

Supresión o modificación de datos contenidos en el sistema, o bien en la alteración del funcionamiento del sistema (sabotaje) impida, obstaculice o modifique su funcionamiento.

Conducta maliciosa que altere, dañe o destruya los datos contenidos en un sistema de tratamiento de información.

#### Gran Bretaña

Debido a un caso de hacking en 1991, comenzó a regir en este país la Computer Misuse Act (Ley de Abusos Informáticos). Mediante esta ley el intento,

Exitoso o no, de alterar datos informáticos es penado con hasta cinco años de prisión o multas. Esta ley tiene un apartado que especifica la modificación de datos sin autorización.

#### Holanda

El 10 de Marzo de 1993 entró en vigencia la Ley de Delitos Informáticos, en la cual se penaliza los siguientes delitos:

El hacking.

El preacking (utilización de servicios de telecomunicaciones evitando el pago total o parcial de dicho servicio).

La ingeniería social (arte de convencer a la gente de entregar información que en circunstancias normales no entregaría).

La distribución de virus.

Normativa y regulación de la informática en el ámbito europeo.

Hasta ahora, el principal esfuerzo europeo por regular el tema de los delitos 21 de noviembre de 2001. Este documento fue firmado por los representantes de cada país miembro del Consejo de Europa, aunque su eficacia depende de su posterior refrendo por los órganos nacionales de cada país firmante.

El “Convenio sobre la Ciberdelincuencia” permitió la definición de los delitos informáticos y algunos elementos relacionados con éstos, tales como “sistemas informáticos”, “datos informáticos”, o “proveedor de servicios”.

Estos delitos informáticos fueron clasificados en cuatro grupos:

1. Delitos contra la confidencialidad, la integridad y la disponibilidad de los datos y sistemas informáticos.

Acceso ilícito a sistemas informáticos.

Interceptación ilícita de datos informáticos.

Interferencia en el sistema mediante la introducción, transmisión, provocación de daños, borrado, alteración o supresión de éstos.

Abuso de dispositivos que faciliten la comisión de delitos

2. Delitos informáticos.

Falsificación informática que produzca la alteración, borrado o supresión de datos informático que ocasionen datos no auténticos.

Fraudes informáticos.

3. Delitos relacionados con el contenido.

Delitos relacionados con la pornografía infantil.

4. Delitos relacionados con infracciones de la propiedad intelectual y derechos afines.

Es conviene destacar que en el “Convenio sobre la Ciberdelincuencia” se encomienda a cada Parte que tome las medidas necesarias para tipificar como delito en su derecho interno cada uno de los apartados descritos en cada categoría.

En la Disposición 14221 del BOE núm. 226 de 2010, encontramos el Instrumento de Ratificación del Convenio sobre la Ciberdelincuencia, hecho en Budapest el 23 de noviembre de 2001.

Normativa y regulación de la informática en el ámbito nacional.

Leyes y Decretos Ley.

Ley Orgánica de Protección de datos de carácter personal: régimen sancionador aplicable (BOE no298 de 14/XII/99 que publicó la Ley Org. 15/1999 de 13 de Dic.)

Objeto: Proteger y garantizar las libertades públicas y derechos fundamentales de las personas, especialmente su HONOR e INTIMIDAD personal y familiar.

Aspectos de interés: Serán responsables: “Los responsables de los ficheros o de los tratamientos” y “los encargados de los tratamientos”.

Ley 7/1998 de 13 de Abril que regula las condiciones generales de contratación.

R.D. 1906/1999 de 17/XII que regula la contratación telefónica.

R.D.Ley 14/1999 de 17/XII sobre Firma Electrónica (BOE No224 de 18/XII)

Firma electrónica: Dispositivo electrónico que permite la identificación del signatario de las operaciones realizadas por Internet.

Identifica: El firmante (autenticación) y Evita el retracto (no repudio).

Código Penal

Ley Orgánica 10/1995 de 23/XI

Tipifica delitos y faltas por el uso de la informática, concretamente contra la Intimidad, Patrimonio, Socioeconómicos y Propiedad Intelectual.

Título X: “Delitos contra la intimidad, derecho a la propia imagen y la inviolabilidad del Domicilio”.

Apoderarse de papeles, e-mails, mensajes, otros...

Cracks: delitos

Obtener datos de terceros...

Recomendaciones de la APD

Información en la recogida de datos.

Cuando suministre datos personales a cualquier organización (proveedores de acceso, proveedores de contenido, vendedores a través de comercio electrónico, etc.) sea consciente de a quién se los facilita y con qué finalidad.

Finalidad para la que se recogen los datos

Desconfíe si los datos que le solicitan son excesivos para la finalidad con la que se recogen o innecesarios para el servicio que se le presta.

Tenga en cuenta que cuando introduce su dirección de correo electrónico en un directorio, lista de distribución o grupo de noticias, dicha dirección puede ser recogida por terceros para ser utilizada con una finalidad diferente, como por ejemplo, remitirle publicidad no deseada.

Cuando navegue por Internet, sea consciente de que los servidores Web que visita pueden registrar tanto las páginas a las que accede como la frecuencia y los temas o materias por las que busca, aunque no le informen de ello.

Seguridad en el intercambio de datos

Utilice, siempre que sea posible, las últimas versiones de los programas navegadores, ya que cada vez suelen incorporar mejores medidas de seguridad.

Considere la posibilidad de activar en dichos programas las opciones que alerten sobre los intercambios de datos no deseados y no rellene aquellos datos que no desee hacer públicos (por ejemplo, dirección de correo electrónico, nombre, apellidos, etc.).

No realice transacciones comerciales electrónicas a través de proveedores con sistemas inseguros o no fiables. Consulte el manual de su navegador para averiguar cómo informa de que se ha establecido una conexión con un servidor seguro.

Recuerde que existen sistemas de dinero electrónico que preservan el anonimato de sus compras en Internet.

Utilice los mecanismos de seguridad que tenga a su alcance para proteger sus datos de accesos no deseados. El medio más fiable para conseguirlo es el cifrado de los mismos.

Salvo que se utilicen mecanismos de integridad, autenticación y certificación (firma digital, notarios electrónicos, etc.) no confíe ciegamente en que la persona u organización que le remite un mensaje es quien dice ser y en que el contenido del mismo no se ha modificado, aunque esto sea así en la inmensa mayoría de las ocasiones.

Para terminar:

Siempre que se le soliciten datos personales que no esté obligado legalmente a suministrar, sopesese los beneficios que va a recibir de la organización que los recoge frente a los posibles riesgos de utilización irregular de los mismos.

Ante cualquier duda sobre la legalidad de la utilización de sus datos de carácter personal, póngase en contacto con la Agencia de Protección de Datos.



La protección jurídica de programas de ordenador. Piratería informática

El Real Decreto Legislativo 1/1996, por el que se aprueba el Texto Refundido sobre Propiedad Intelectual, la protección jurídica de los programas de ordenador, antes regulada por la Ley de Protección Jurídica de Programas de Ordenador y por la Ley de Propiedad Intelectual, crea un marco jurídico en contra de la piratería informática.

El Texto Refundido desarrolla una serie de medidas para combatir la piratería informática, como la posibilidad de que los fabricantes de programas de ordenador soliciten a la justicia española la realización de un registro sorpresa en empresas en las que existan sospechas fundadas o evidencias de delito.

España es uno de los países en los que se puede acudir a esta medida cautelar.

De esta manera se erradica la posibilidad de que los presuntos infractores puedan destruir las pruebas existentes, lo cual, indudablemente ocurrirá si se les notifica por adelantado la realización de un registro.

¿En qué casos se infringe la Ley?

Al copiar o distribuir un programa de ordenador o la documentación que le acompaña, incluidas aplicaciones, datos, códigos y manuales, sin permiso expreso o licencia del propietario de los derechos de explotación.

Al utilizar un programa sin la correspondiente licencia o autorización del fabricante, con independencia de que se utilice en un solo ordenador o en varios de forma simultánea.

Al utilizar programas de ordenador en un número de copias superior al autorizado por el fabricante en sus contratos o licencias de uso.

En empresas y demás organizaciones, al fomentar, consciente o inconscientemente, permitir, obligar o presionar a los empleados a realizar o distribuir copias no autorizadas del programa.

Al efectuar copias no autorizadas porque alguien lo requiere u obliga a ello. Al ceder o prestar el programa de forma que pueda ser copiado o al copiarlo mientras está en su posesión en calidad de cedido o prestado.

Al crear, importar, poseer o negociar con artículos destinados a burlar o neutralizar cualquier medio técnico aplicado para proteger el programa de ordenador.

Medidas Judiciales

Si finalmente existe evidencia de delito, las medidas judiciales que pueden adoptarse son:

Solicitar al Juez un registro sorpresa de las instalaciones del presunto infractor, tanto por la vía civil, como por la penal.

Solicitar al Juez la adopción urgente de medidas cautelares de protección.

Exigir indemnizaciones acordes con los daños materiales y morales causados.

El cierre del centro de actividad del infractor.

El secuestro de todos aquellos medios destinados a suprimir los dispositivos técnicos que protegen un programa desarrollado y comercializado por un fabricante de programas.

TFG01:

***Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.***

Esta competencia se alcanza gracias a la completa realización de este trabajo de fin de grado, abarcando todos los ámbitos dentro de las tecnologías específicas de la Ingeniería Informática.

## Legislación vigente

### Forma Jurídica

Para esta empresa la forma jurídica que se ha estimado más conveniente es la de Sociedad de Responsabilidad Limitada. El capital social de este tipo de sociedades mercantiles está dividido en participaciones sociales, indivisibles y acumulables, y se integran por las aportaciones de todos los socios, quienes no responderán personalmente de las deudas sociales. Este tipo de sociedades requiere un número socios mínimos de sólo 1 persona, la responsabilidad de los socios se limita al capital aportado y el capital social mínimo es de Mínimo 3.000 €. Este capital Deberá estar íntegramente suscrito y desembolsado en el momento de la constitución. Además, se debe tener en cuenta que Sólo podrán ser objeto de aportación social los bienes o derechos patrimoniales susceptibles de valoración económica, en ningún caso trabajo o servicios.

El marco legal a la que estaría sujeta es

Real Decreto Legislativo 1/2010, de 2 julio, por el que se aprueba el texto refundido de la Ley de Sociedades de Capital.

Orden JUS/3185/2010, de 9 de diciembre, por la que se aprueban los Estatutos-tipo de las sociedades de responsabilidad limitada.

Este tipo de sociedad tributa por el impuesto sobre sociedades

Los derechos de los socios son los siguientes:

Participar en el reparto de beneficios y en el patrimonio resultante de la liquidación de la sociedad.

Participar en las decisiones sociales y ser elegidos como administradores

Sus órganos sociales son los siguientes:

Junta General de socios

Órgano deliberante que expresa en sus acuerdos la voluntad social y cuya competencia se extiende fundamentalmente a los siguientes asuntos:

Censura de la gestión social, aprobación de cuentas anuales y aplicación del resultado.

Nombramiento y separación de los administradores, liquidadores, y en su caso de auditores de cuentas.

Modificación de los estatutos sociales.

Aumento o reducción del capital social.

Transformación, fusión y escisión de la sociedad.

Disolución de la sociedad.

## Los Administradores

Órgano ejecutivo y representativo a la vez, que lleva a cabo la gestión administrativa diaria de la empresa social y la representación de la entidad en sus relaciones con terceros.

La competencia para el nombramiento de los administradores corresponde exclusivamente a la Junta General.

Salvo disposición contraria en los estatutos se requerirá la condición de socio.

## 4 Aportaciones Socio-Económicas

Este trabajo final de grado pretende contribuir a que pequeños establecimientos relacionados con el sector del turismo rural, puedan incorporarse de forma gratuita a Internet y al comercio electrónico.

Partiendo del hecho que las empresas pequeñas no suelen tener recursos económicos como para poder disponer de páginas profesionales con un sistema de información de productos y servicios, posibilidad de gestionar reservas y realización de compras. Esta herramienta proporciona de forma fácil y económica todas estas características en un solo lugar. Además esto hace más competitivas a las empresas, ya que, sus productos y servicios se hacen visibles en todo el mundo y esto facilita la penetración en el sector.

También se facilita la generación de contenido de forma dinámica, los usuarios solo tendrán que introducir en el sistema la información de sus productos, sin esperas, sin retrasos, sin pagos adicionales.

En consecuencia existe un aumento de la productividad, ya que, la velocidad a la que se genera el sitio depende del gestor de la empresa y, de esta manera, no se depende de terceros que proporcionen los mismos servicios e incrementan los gastos.

En definitiva se proporciona un servicio para que las empresas que no pueden acceder de manera natural al sector puedan ser competitivas, puedan darse a conocer al usuario de forma fácil y sencilla, y en consecuencia aumenten sus ingresos y visibilidad en el sector.

## 5 Planificación del trabajo

### 5.1 Metodología de desarrollo

En este proyecto se ha optado por seguir un desarrollo de prototipos combinándolo con un desarrollo en cascada. Por tanto se han seguido las siguientes fases (Véase Imagen 7):

**1. Desarrollo de prototipos.**

**2. Análisis de requisitos de usuario.**

Definición de lo que el usuario espera del sistema. A través de entrevistas, encuestas estudios y prototipos.

**3. Análisis de requisitos software.**

En este apartado se trata de obtener una definición del software que se va a construir. Se construye un modelo lógico del software.

**4. Diseño arquitectónico.**

Se define la estructura del software, equivale a la estructura más su control. Se especifica cada módulo y se diseñan las interrelaciones entre ellos.

**5. Diseño detallado y producción.**

Se detalla cada componente de la arquitectura definido anteriormente. A continuación, se produce en los lenguajes de programación elegidos el sistema software para su posterior integración y testeo.

**6. Transferencia.**

Se implanta el sistema y se configura en un entorno operativo para dejarlo ejecutable para el usuario.

El desarrollo del prototipo ha seguido las siguientes fases:

1. Recolección de requisitos iniciales.
2. Diseño rápido.
3. Construcción del prototipo.
4. Evaluación del prototipo.

Esta primera metodología de desarrollo aplicada ha permitido identificar claramente los requisitos del usuario y evaluar los límites que podía alcanzar el proyecto. La segunda metodología usada, la clásica o en cascada, es el método más asentado por su historia y ha sido fuente de inspiración para otros modelos de proceso de software. Este modelo exige un enfoque sistemático y secuencial. En el que cada fase se ejecuta una vez y sienta las bases para la siguiente. Esto no quiere decir que en determinadas fases no se haga una realimentación a las anteriores, de hecho a medida que se va completando cada fase se van ajustando las anteriores.

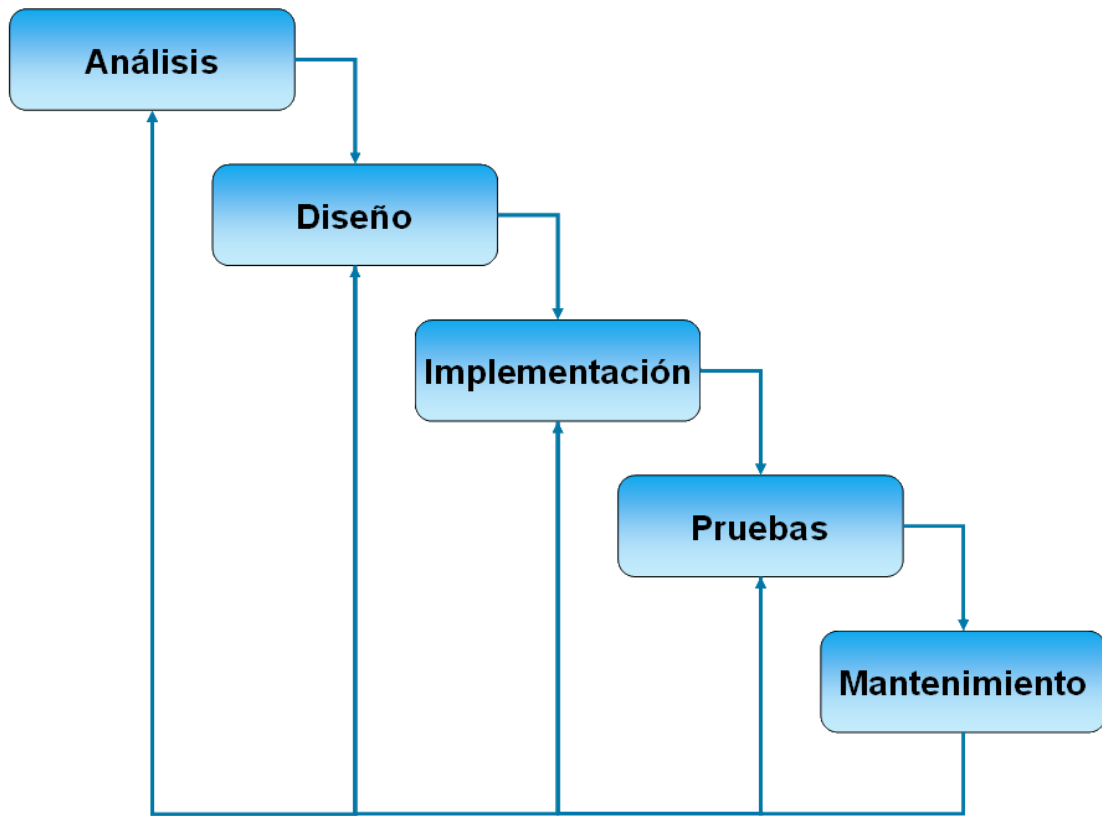


Imagen 7: Método de desarrollo en cascada

## 5.2 Planificación y temporización

A continuación mostramos las impresiones de la planificación y temporización del proyecto (Véase Tabla 1):

A la fase de Análisis le dedicamos unas treinta horas de proyecto, debido a la facilidad para encontrar manuales y ejemplos de las tecnologías web. También los requisitos fueron fáciles de analizar ya que eran términos sencillos y manejables. Las estrategias y metodologías a seguir fueron bastante claras desde el principio y la experiencia previa en este tipo de proyectos, fue la clave determinante para avanzar deprisa en este apartado.

En cuanto a la fase de Implementación, el diseño previo fue relativamente rápido utilizando herramientas “Mockup”, el diseño de la base de datos fue sencillo, el diseño de la plataforma “backend” fue relativamente sencillo pero el diseño del “Frontend” fue bastante más complicado y llevó mucho tiempo completar esta fase.

A la fase final le dedicamos bastante menos tiempo, ya que, la fase de pruebas la íbamos realizando también durante la fase de implementación.

FASE	HORAS	%
<b>1. Análisis</b>		
Estudio inicial de necesidades y requisitos	5	
Estudio de la herramienta en la que se va a realizar el desarrollo	10	
Anatomía y estructura de los servicios web.	5	
Anatomía y estructura de la aplicación web.	3	
Análisis y estrategia de contenidos	4	
Documentación de la información generada en la fase de Análisis	3	
<b>TOTAL FASE DE ANÁLISIS</b>	<b>30</b>	<b>10%</b>
<b>2. Desarrollo del proyecto</b>		
Diseño de aplicación y plataforma web	8	
Análisis y programación (aplicación web y servicios Web)	70	
Análisis y programación de base de datos	20	
Finalización de fase diseño	135	
Documentación de la información generada en la fase de Desarrollo	6	



<b>TOTAL FASE DE DESARROLLO</b>	<b>239</b>	<b>74,7%</b>
<b>3. Fase final</b>		
Pruebas finales	8	
Planificación del mantenimiento de la plataforma	7	
Planificación del seguimiento de resultados previsto	7	
Documentación de la información generada en la fase final	7	
<b>TOTAL FASE FINAL</b>	<b>29</b>	<b>9,6%</b>
<b>TOTAL HORAS</b>	<b>300</b>	

Tabla 1: Planificación y Temporización

## 6 Desarrollo del trabajo

### 6.1 Requisitos del sistema

#### 6.1.1 Modelo del dominio

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema.

Muchos de los objetos del dominio o pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio.

A continuación se presenta el modelo del dominio para el contexto en el que se engloba la aplicación. Debido a las dimensiones del modelo, este se ha dividido en varias partes para facilitar su comprensión y dar mayor claridad al mismo.

#### 6.1.2 Diagramas modelo de dominio

##### El usuario administrador:

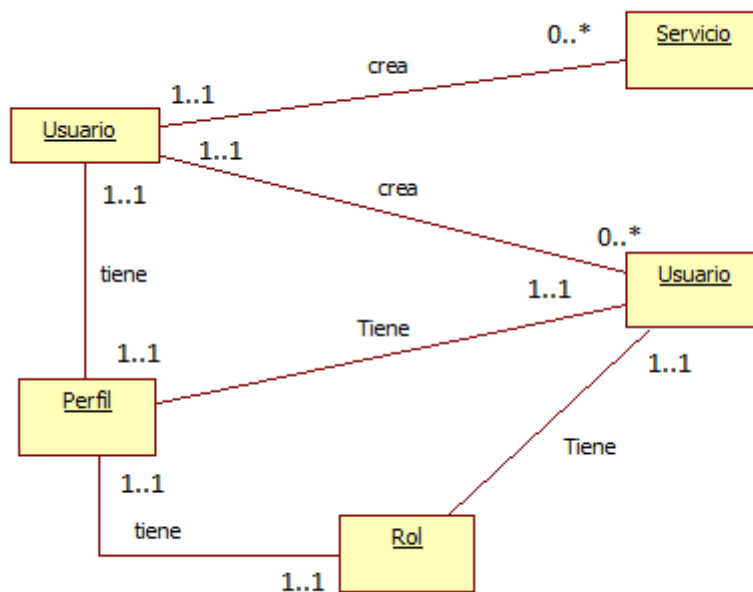


Figura 1: Modelo de dominio de usuario administrador

El administrador puede crear usuarios y servicios, los usuarios tienen un perfil y un rol único (Véase Figura 1).

**El usuario cliente:**

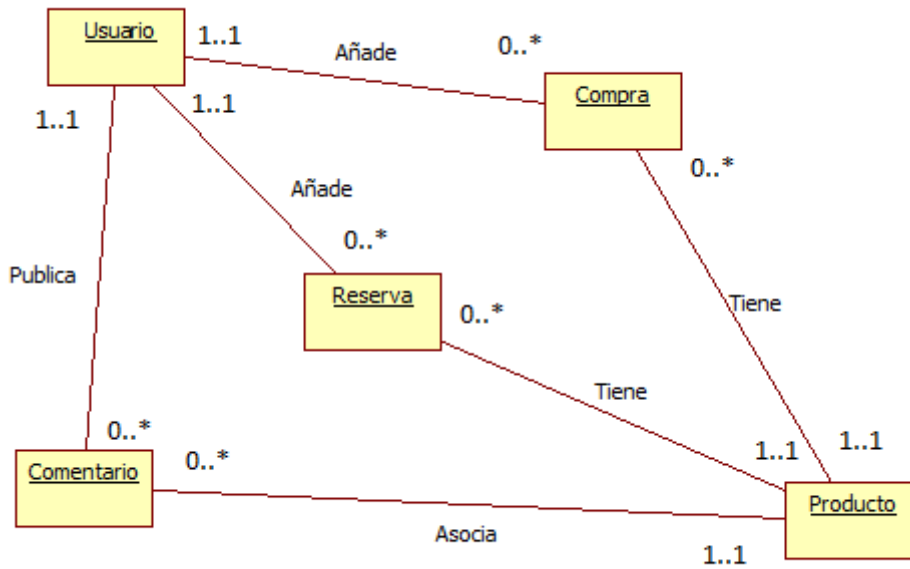


Figura 2: Modelo de dominio de usuario cliente

El cliente puede publicar comentarios y asociarlos a los productos, además puede añadir reservas y compras de productos (Véase Figura 2).

## El usuario gestor:

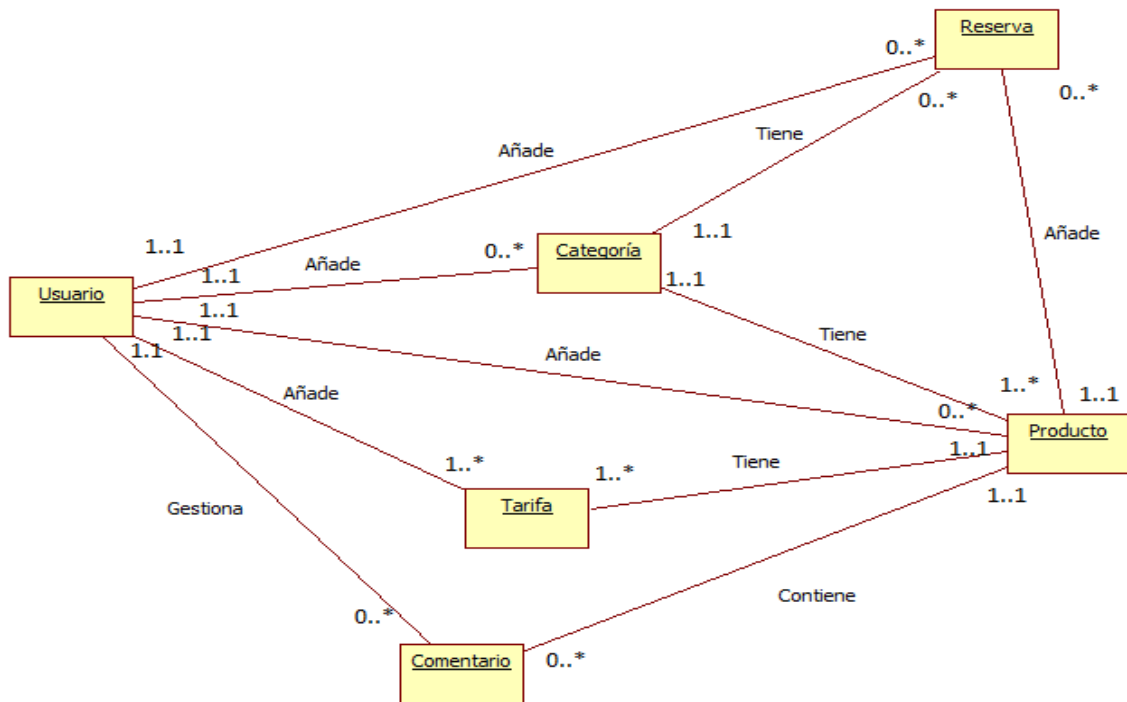


Figura 3: Modelo de dominio de usuario gestor

El usuario gestor puede crear categorías y productos, puede añadir tarifas y gestiona la visualización de los comentarios en los productos (Véase Figura 3).

### 6.1.3 Diagramas entidad relación

Un diagrama o modelo entidad-relación (es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

Para el TFG se han diseñado dos bases de datos:

La Base de Datos enatural\_db está compuesta por las tablas más importantes de la base de datos, antes de realizar el diseño estudiamos previamente sus entidades a continuación las enumeramos:

- **Usuario:** Es la entidad básica que representa a un participante del sistema, los tipos de usuarios pueden ser Usuario anónimo, Usuario Cliente, Usuario Gestor y Usuario Administrador. Sus características son: Id,Email,Password,Estado,Idweb.
- **Referencia:** Representa un código de activación de cuentas de usuario. Sus características son: Id, Código.
- **Rol:** Representa un rol de usuario, pueden ser Administrador, Gestor y Cliente. Sus características son: Id,Tipo.
- **Perfil:** Representa toda la información pública y privada relativa al usuario. Sus características son: Id,Edad,Iduser,Avatar,Nombre,Apellidos,Comentarios,Fecha de nacimiento.

- **Registro:** Representa una actividad realizada por alguno de los usuarios de la web. Sus características son: Id, Iditem, Accion, Fecha de emisión, Idweb.
- **Categoría:** Representa un tipo de producto. Sus características son: Id, Nombre, Descripción, Imagen.
- **Servicio:** Representa una actividad específica de la empresa. Sus características son: Id, Nombre, Descripción, Imagen, Idweb.
- **Tarifa:** Representa un precio temporal de un producto. Sus características son: Id, Fecha inicio, Fecha fin, Emisión, Tasas, Precio1, Precio2, impuestos.
- **Reserva:** Representa un registro que otorga de manera temporal el uso exclusivo de un producto a un usuario. Sus características son: Id, Fecha Inicio, Fecha Fin, Fecha de emisión, Información, Email, Dirección.
- **Producto:** Representa una actividad, servicio o alojamiento. Sus características son: Id, Galeria de imágenes, Archivos, Nombre, Imagen de portada, Descripción.
- **Comentario:** Representa una opinión de un producto dado. Sus características son: Id, Comentario, Activo.
- **Municipio/Provincia:** Representa una localización. Sus características son: Id municipio, Id Provincia, Codigo municipio, Digito control, Nombre.
- **Webtemplate:** Representa la plantilla o colección de plantillas de la web de clientes.

Seguidamente se muestra el diagrama entidad relación que previamente hemos diseñado para las relaciones entre entidades (Véase Figura 4):

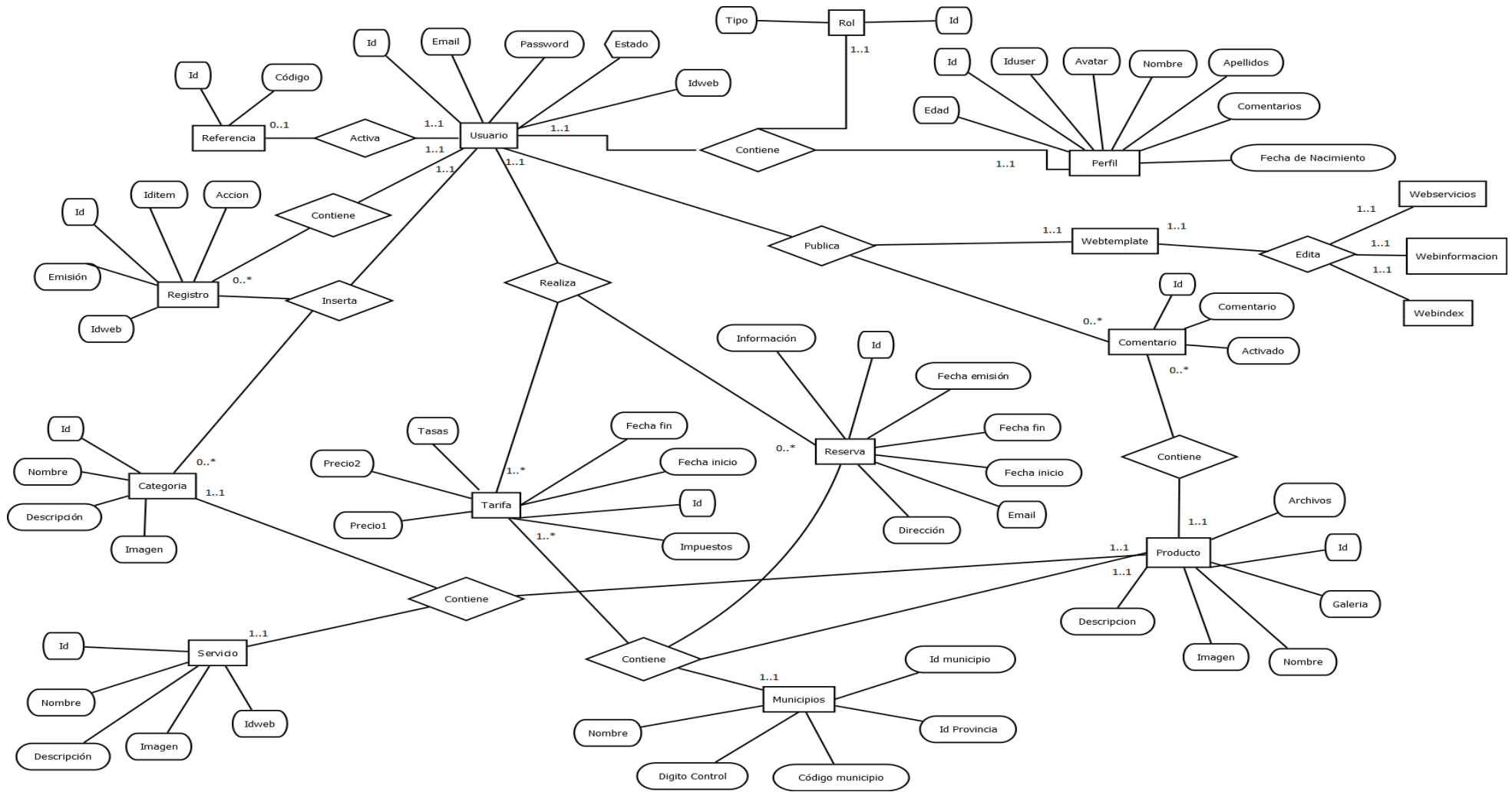


Figura 4: Diagrama entidad relación enatural\_db.

Por otro lado el esquema de la segunda base de datos statistics\_db fue mucho más simple ya que solo existían tres relaciones usuarios, servicios y productos . Una característica importante a tener en cuenta en esta base de datos es que las tablas se van a autogenerar dinámicamente, es decir, cada vez que un servicio, un producto o un usuario se creen en el sistema, se generaran tablas con la siguiente plantilla [servicio o usuario o producto]\_[id del objeto]\_[ id de la web]. De esta manera se guardaran en esta base de datos las estadísticas de uso de cada uno por separado, proporcionando una información exacta de su uso.

A continuación se muestra el diagrama entidad relación de la base de datos (Véase Figura 5):

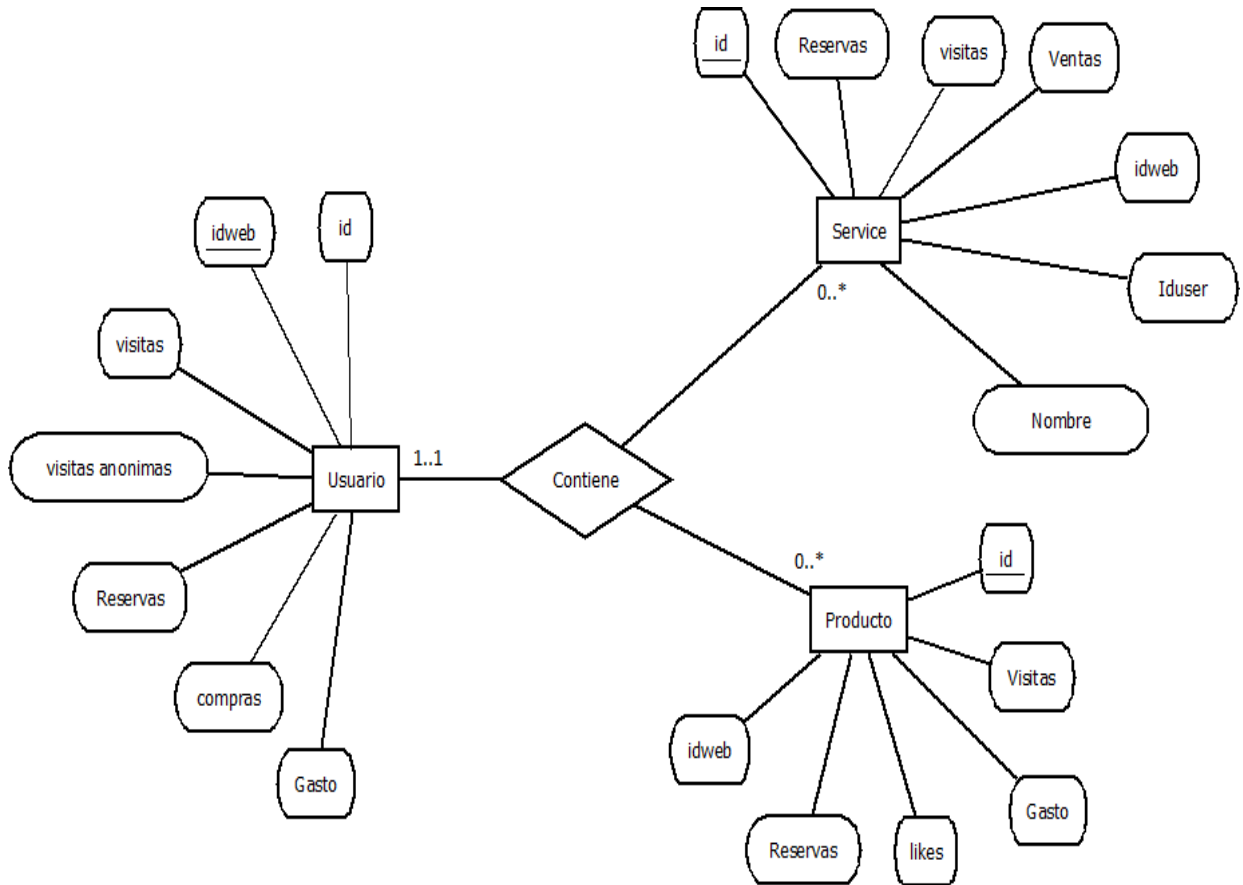


Figura 5: Diagrama entidad relación db\_statistics

#### 6.1.4 Modelo relacional

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que éstos se almacenen no tiene relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el Álgebra relacional y el Cálculo relacional. El Álgebra relacional permite describir la forma de realizar una consulta, en cambio, el Cálculo relacional sólo indica lo que se desea devolver.

A continuación se muestra el modelo relacional de la base de datos enatural\_db (Véase Figura 6):



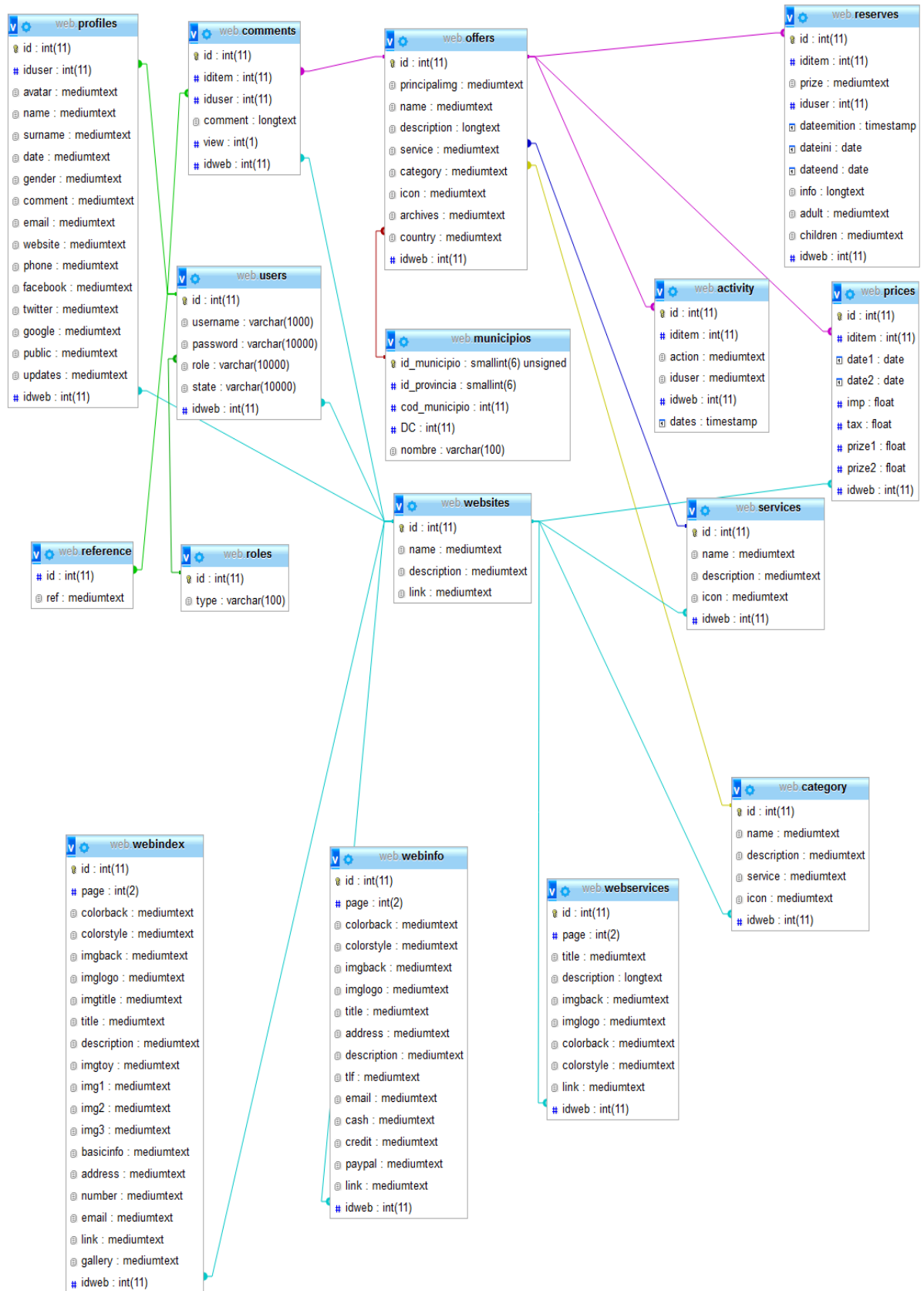


Figura 6: Modelo relacional enatural\_db.

A continuación, en la figura 7 se muestra el modelo relacional de la base de datos db\_statistics:

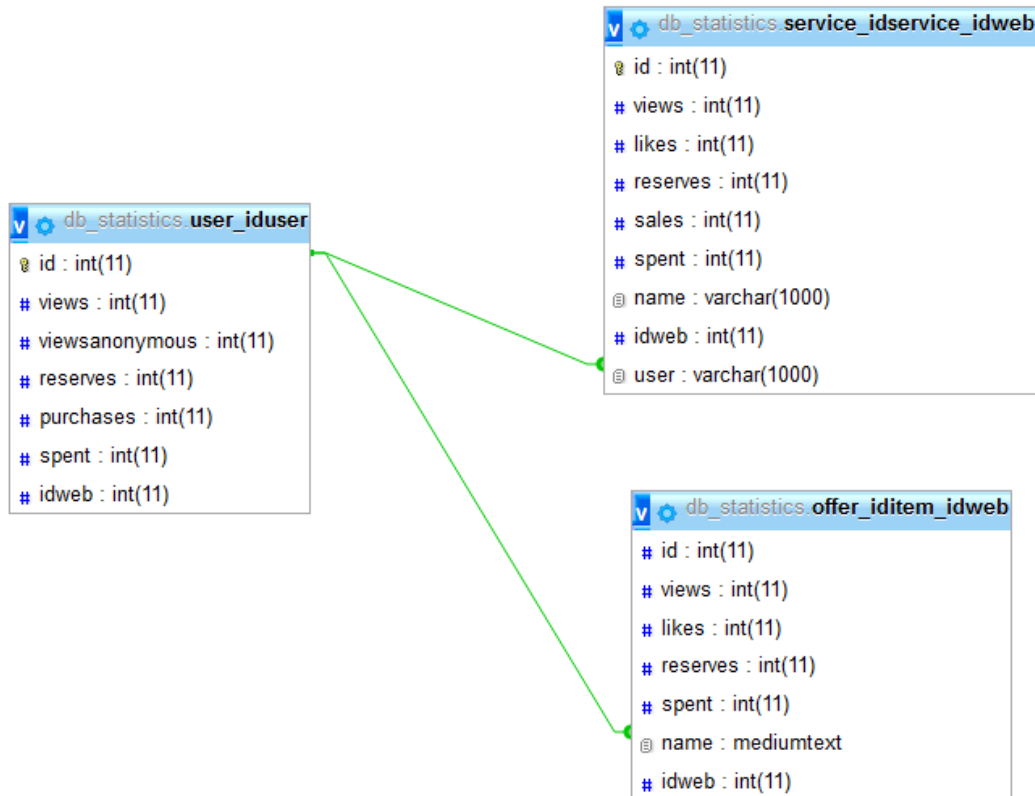


Figura 7: Modelo relacional db\_statistics

## 6.2 Especificación de requisitos de usuario.

### 6.2.1 Descripción.

El presente proyecto tiene como propósito elaborar un portal web que ofrece a los clientes la posibilidad de encontrar servicios relacionados con empresas del sector de actividades en la naturaleza. A su vez, ofrece a las empresas poder promocionar sus productos y servicios mediante un alojamiento web, gestionar su contenido y realizar reservas de productos.

Seguidamente, se presenta el glosario de conceptos que se utilizará a lo largo de este trabajo:

**Usuarios anónimos:** el usuario puede encontrar a las empresas mediante un buscador definiendo el servicio, la categoría y la ubicación. También puede visualizar los contenidos web.

**Usuario registrado:** el usuario puede realizar las mismas acciones que el usuario anónimo pero además puede realizar compras, realizar reservas y añadir comentarios a los productos. Además puede editar su perfil, visualizar reservas y cancelar reservas.

**Usuario Gestor:** el usuario puede:

- Gestionar categorías: buscar, añadir, editar y eliminar.
- Gestionar productos: buscar, añadir, editar y eliminar.
- Gestionar tarifas de productos: buscar, añadir, editar y eliminar.
- Gestionar reservas: buscar, añadir, editar y eliminar.
- Gestionar descuentos: buscar, añadir, editar y eliminar.
- Gestionar comentarios: buscar, mostrar y no mostrar.
- Gestor web.
- Visualizar estadísticas y visualizar registro de eventos.
- Editar perfil.

**Usuario Administrador:** este usuario puede:

- Gestionar usuarios: buscar, añadir y eliminar.
- Gestionar servicios: buscar, añadir, editar y eliminar.
- Gestionar categorías: buscar, añadir, editar y eliminar.
- Gestionar productos: buscar, añadir, editar y eliminar.
- Gestionar tarifas de productos: buscar, añadir, editar y eliminar.
- Gestionar reservas: buscar, añadir, editar y eliminar.
- Gestionar descuentos: buscar, añadir, editar y eliminar.
- Gestionar comentarios: buscar, mostrar y no mostrar.
- Gestor web.
- Visualizar estadísticas y visualizar registro de eventos.
- Editar perfil.

**Categoría:** se entiende como una forma de etiquetar el servicio.

**Producto/Servicio:** es el bien que ofrece la empresa. Tiene una fecha de comienzo y final, un precio y una reserva, y demás datos relacionados.

**Tarifa:** es el precio asociado al producto o servicio en un rango de periodos de fechas determinado..

**Reserva:** es la guarda que se realiza de un producto o servicio durante un tiempo determinado, muestra una fecha de entrada y de salida y una tarifa asociada.

**Comentario:** son textos asociados a las opiniones de los usuarios registrados relacionadas con los productos.

## 6.2.2 Actores

En el sistema habrá varios actores de software(Véase Figura 8). Sus relaciones se muestran a continuación:

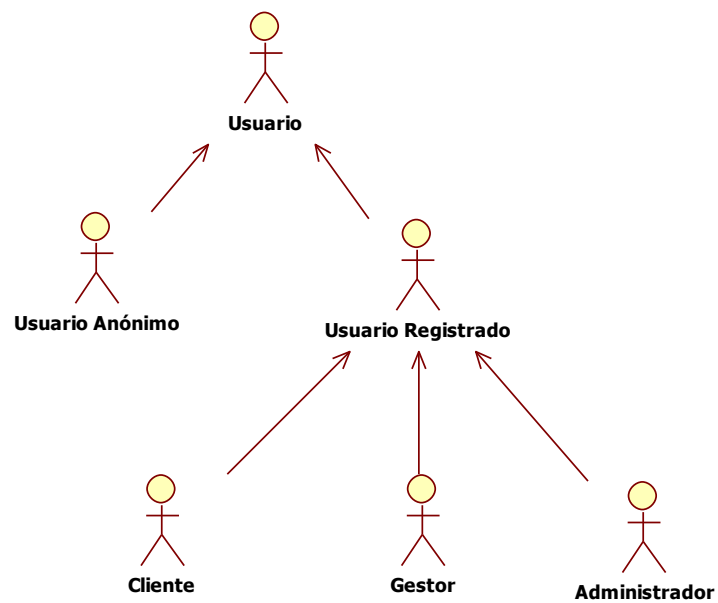


Figura 8: Actores

### 6.2.3 Listados de actores y sus roles

A continuación se muestran el listado de actores, sus roles y definiciones (Véase Tabla 2)

Actor (roles)	Tipo	Definición
<b>Usuario Administrador</b>	Principal	El usuario que controla y gestiona la base de datos y los contenidos de las empresas.
<b>Usuario Gestor</b>	Principal	El usuario que puede gestionar el contenido web de la empresa y visualizar las actividades.
<b>Usuario Cliente</b>	Principal	Puede realizar las mismas acciones que el usuario anónimo pero además puede realizar compras, realizar reservas y añadir comentarios a los productos. Además puede editar su perfil, visualizar reservas y cancelar reservas.
<b>Usuario Anónimo</b>	Principal	El usuario puede encontrar a las empresas mediante un buscador definiendo el servicio, la categoría y la ubicación. También puede visualizar los contenidos web.

Tabla 2: Listado de actores y roles

#### Listado de actores y sus objetivos

Seguidamente se muestran el listado de actores y sus objetivos (Véase Tabla 3).

Actor (roles)	Objetivos	Resumen de acciones
<b>Usuario</b>	Visualizar web	Muestra en detalle el producto seleccionado.
	Registrarse	Usuario se registra y se envía un email confirmando validación
<b>Usuario Registrado</b>	Cerrar sesión	Cierra la sesión actual para pasar a ser un usuario anónimo.
	Gestionar perfil	Guarda las modificaciones realizadas en el perfil de usuario.
<b>Usuario Cliente</b>	Visualizar producto	Muestra los datos del producto seleccionado.
	Reservar producto	Realizar reserva del producto. Visualizar reservas

		y cancelar reservas.
	Comentar producto	Introducir comentario de un producto seleccionado
<b>Usuario Gestor</b>	Gestionar categoría	Guarda los datos relacionados con la categoría.
		Guarda los cambios de la categoría seleccionada y visualizada.
		Elimina la categoría seleccionada de la lista de categorías.
		Filtrar categoría de la lista de categorías.
	Gestionar producto	Guarda los datos relacionados con el producto.
		Guarda los cambios del producto seleccionado y lo visualiza.
		Elimina el producto seleccionado de la lista de productos.
		Filtrar producto de la lista de productos.
	Gestionar tarifa	Guarda los datos relacionados con la tarifa.
		Guarda los cambios de la tarifa seleccionada y la visualiza.
		Elimina la tarifa seleccionada de la lista de tarifas.
		Filtrar tarifa de la lista de tarifas.
	Gestionar reserva	Guarda los datos relacionados con la reserva.
		Guarda los cambios de la reserva seleccionada y la visualiza.
		Elimina la reserva seleccionada de la lista de reservas.
		Filtrar reserva de la lista de reservas.
	Gestionar comentarios	Filtrar un comentario de la lista de comentarios.
		Visualiza los comentarios seleccionados.
		No visualiza los comentarios seleccionados.
	Edición web	Guardar datos relacionados con la web y generar plantilla

		web.
	Visualizar registro	Visualizar toda la actividad relacionada con la web
<b>Usuario administrador</b>	Gestionar usuario	Guarda los datos relacionados con usuarios.
		Guarda los cambios del usuario seleccionado y lo visualiza.
		Elimina el usuario seleccionado de la lista de usuarios.
		Filtrar usuario de la lista de usuarios.
	Gestionar servicio	Guarda los datos relacionados con servicios.
		Guarda los cambios del servicio seleccionado y lo visualiza.
		Elimina el servicio seleccionado de la lista de servicios.
		Filtrar servicio de la lista de servicios.

Tabla 3: Listado de actores y sus objetivos

\*El usuario administrador tiene además todas las tareas del usuario gestor.

## 6.2.4 Modelo de casos de uso

### Casos de usos de las tareas comunes para todos los usuarios:

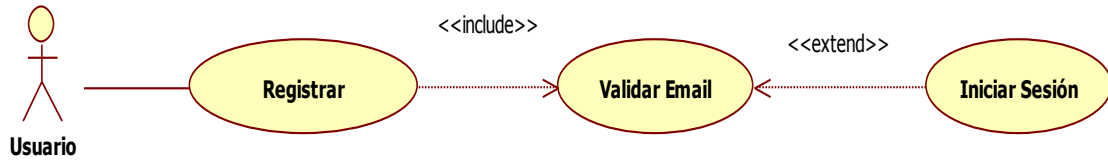


Figura 9: Caso de uso Registrar

Los usuarios anónimos pueden en cualquier momento auto-registrarse en el sistema para acceder a las operaciones de reservar, comprar y comentar los productos (Véase Figura 9).

### Casos de usos por roles:

#### Clientes:

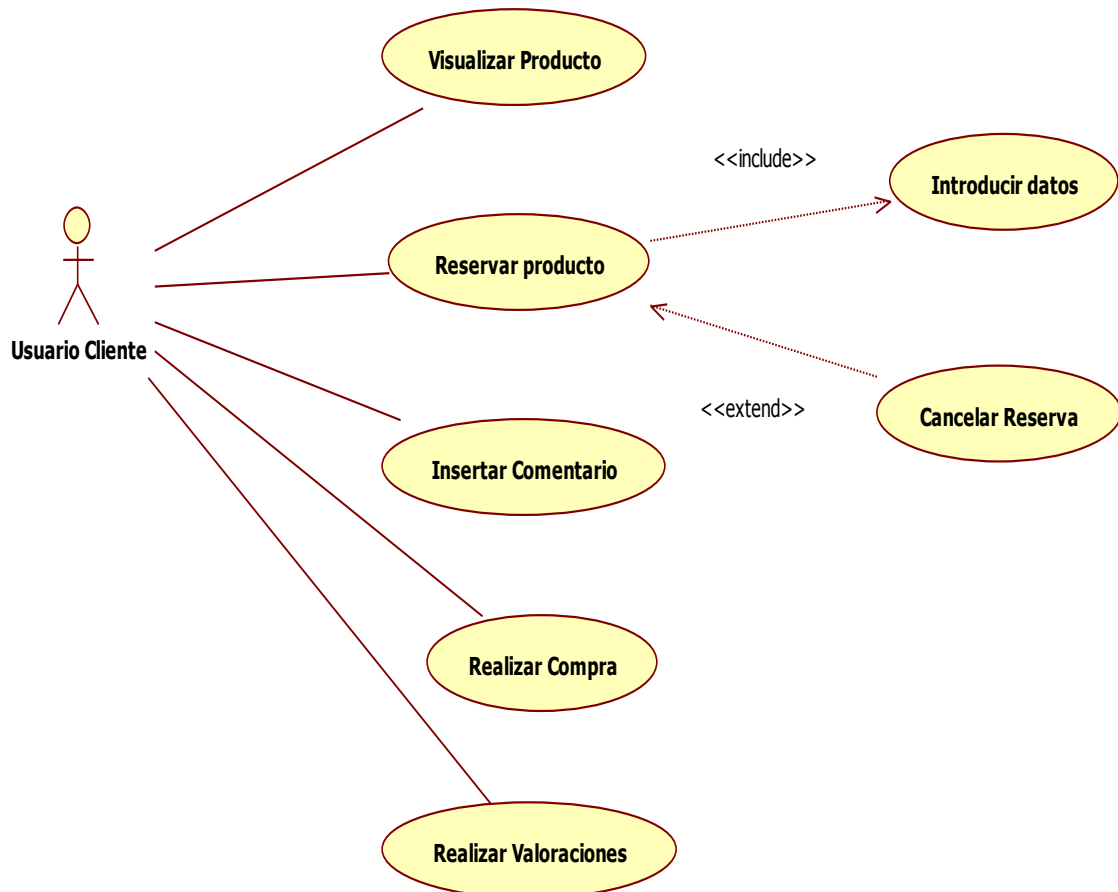


Figura 10: Casos de uso rol cliente



El cliente tendrá todas las operaciones de usuario registrado, podrá visualizar los productos, reservar productos, cancelar reservas y realizar compras de productos a través de Paypal. Además el usuario cliente podrá realizar valoraciones de productos y de empresas (Véase Figura 10).

**Gestores:**

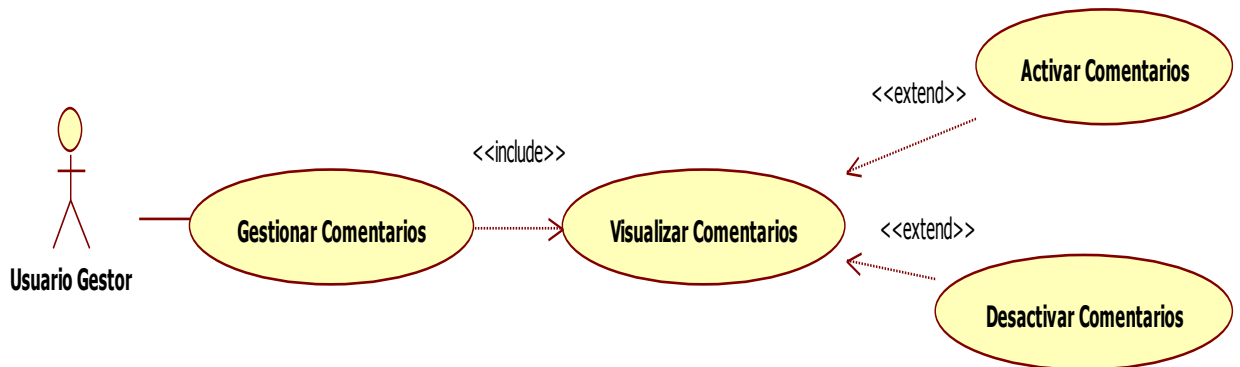


Figura 11: Caso de uso Gestionar Comentarios

Los gestores podrán mostrar o no mostrar los comentarios que se realicen en sus productos. No se podrán borrar de la base de datos (Véase Figura 11).

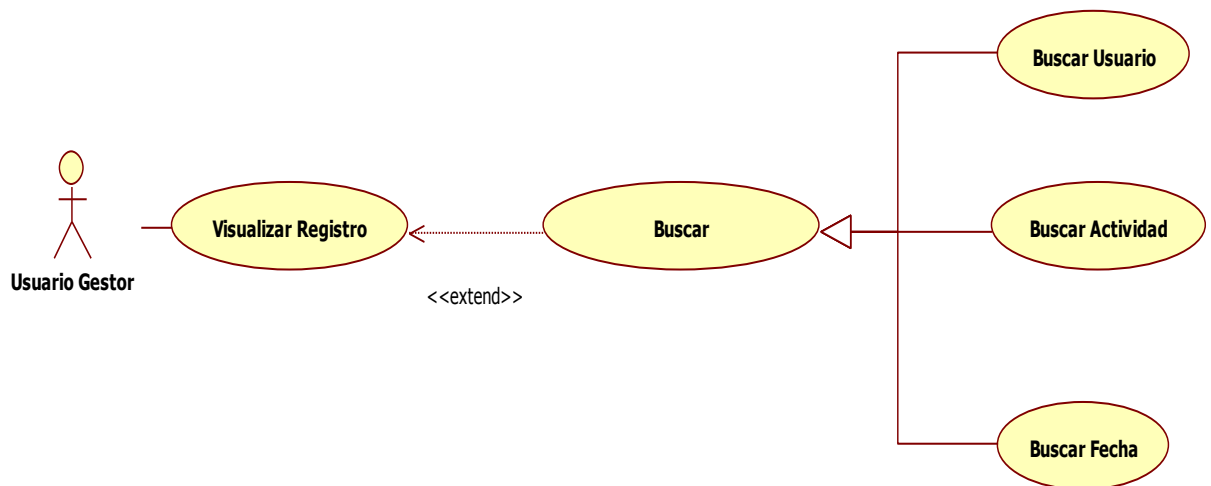
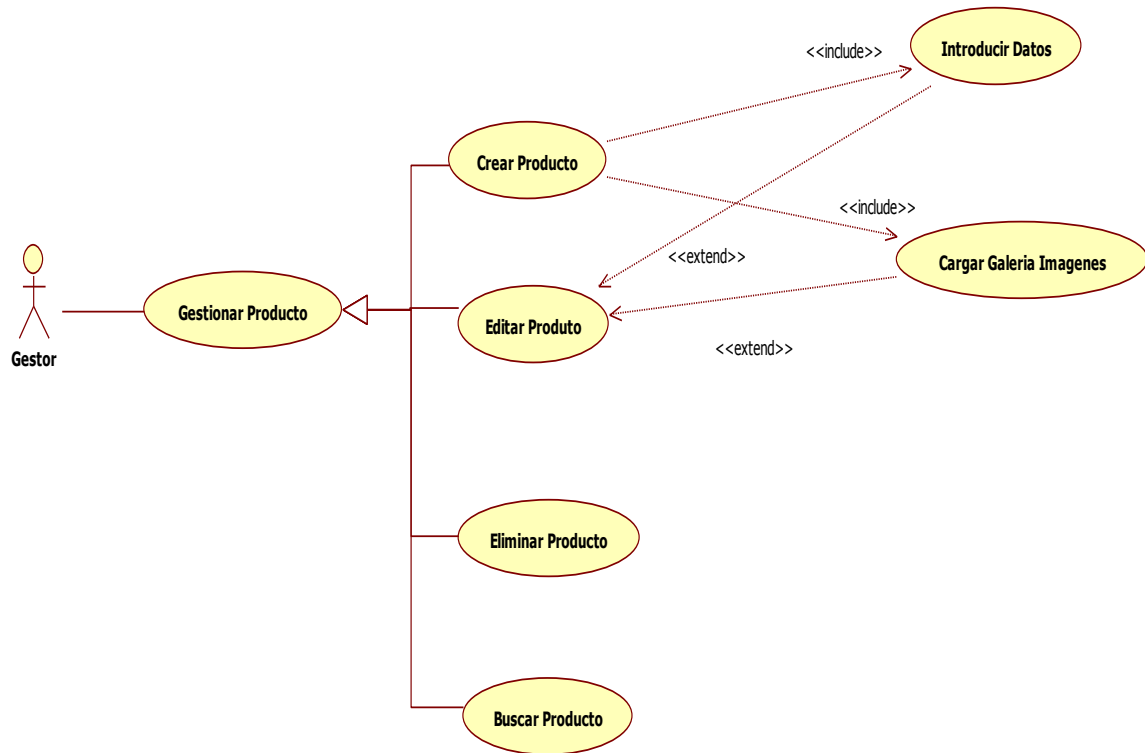


Figura 12: Caso de uso Visualizar Registro

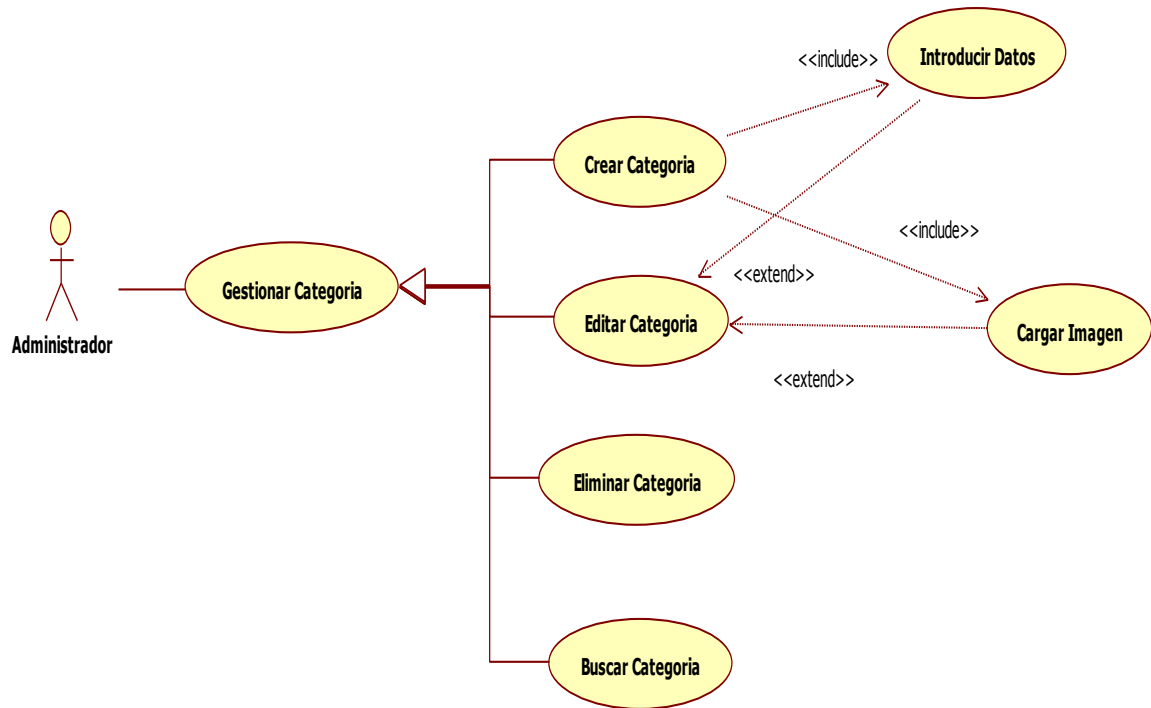
Los gestores podrán visualizar las actividades mediante un visor de registro que se realicen en su sitio web (Véase Figura 12).



**Figura 13: Caso de uso Gestionar Producto**

El usuario gestor podrá crear o editar productos introduciendo los datos del producto, una imagen que lo represente y además la posibilidad de tener una galería de imágenes o archivos relacionados con el producto. También se podrá tener las opciones de eliminar y buscar productos relacionados con los servicios y categorías de la empresa (Véase Figura 13).

**Administrador:**



**Figura 14: Caso de uso Gestionar Categoría**

El usuario gestor podrá crear o editar categorías introduciendo los datos necesarios, subiendo una imagen y eligiendo un servicio para la categoría. También podrán eliminar y buscar categorías de servicio relacionados con las actividades de las empresas (Véase Figura 14).

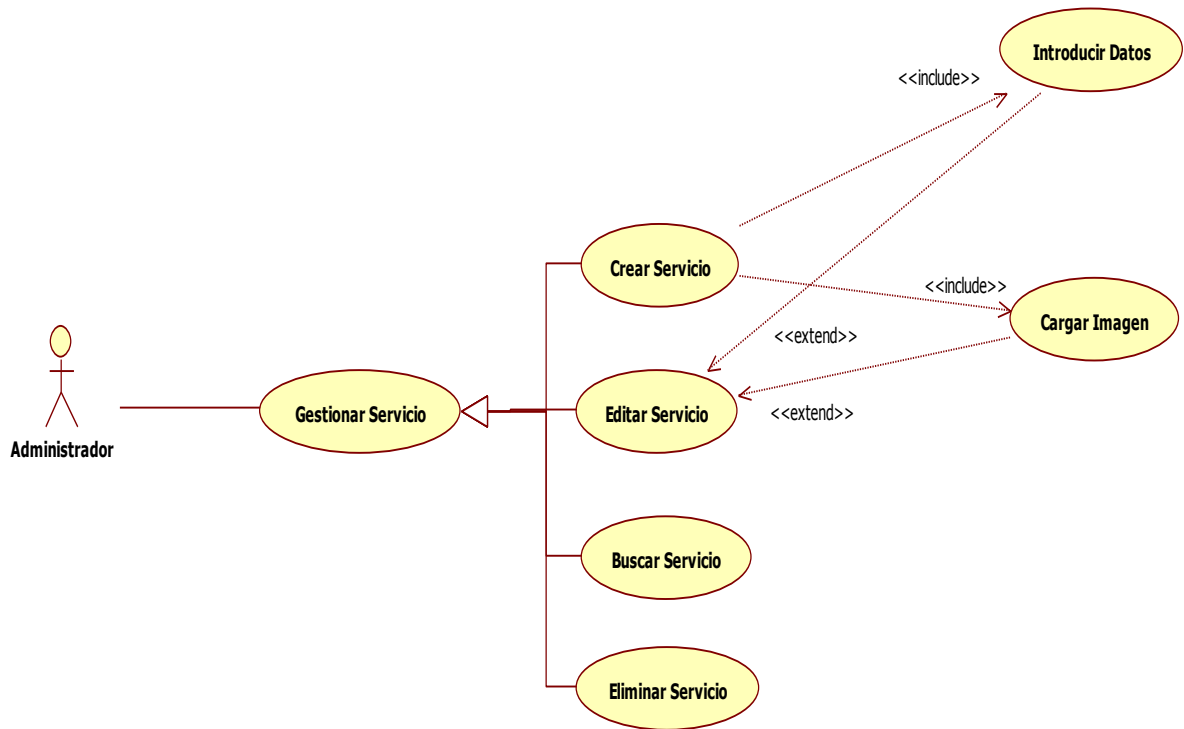


Figura 15: Caso de uso Gestionar Servicio

El administrador podrá crear o editar los servicios comunes para todas las empresas de la web. También tendrá la posibilidad de buscar y eliminar servicios (Véase Figura 15).

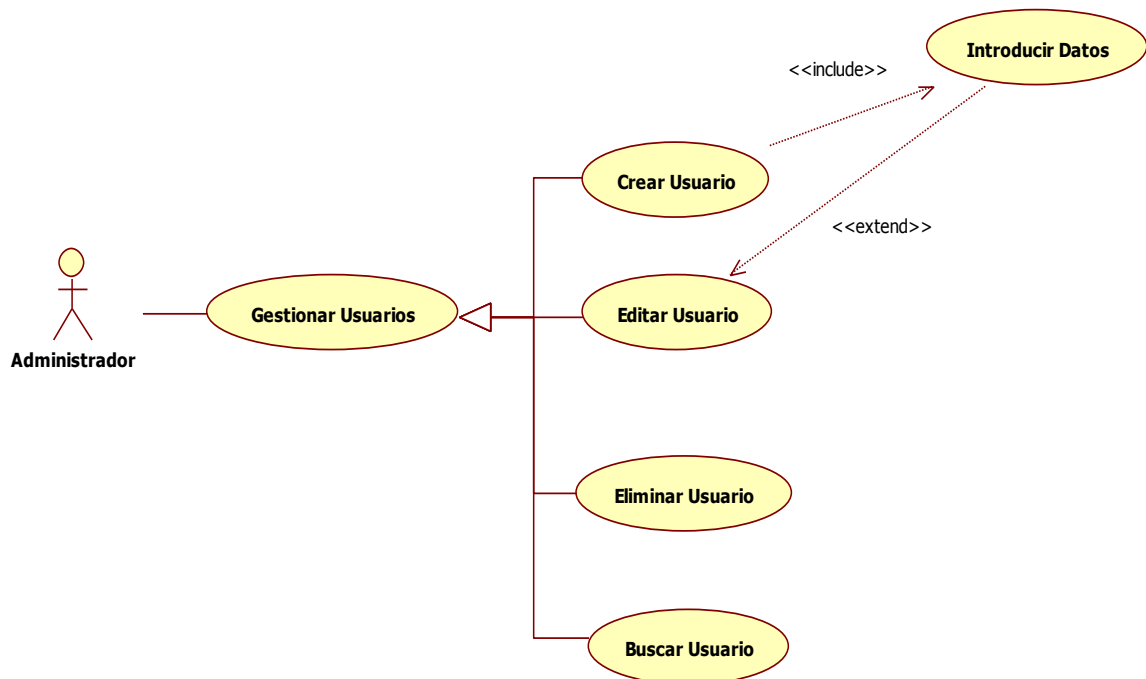


Figura 16: Caso de uso Gestionar Usuarios

El administrador podrá crear o editar a todos los usuarios con roles de gestor y clientes dentro del sistema. También podrá eliminarlos y buscar usuarios (Véase Figura 16).

### 6.2.5 Lista de Casos de Uso

A continuación mostramos una tabla con todo el listado de casos de uso (Véase Tabla 4):

Actor(Roles)	Caso de Uso	Nº Caso de uso
Usuario	Registrarse	01
	Visualizar Web	02
	Visualizar Producto	03
Usuario Registrado	Iniciar Sesión	04
	Cerrar Sesión	05
Usuario Cliente	Realizar Compra	06
	Realizar Reserva	07
	Insertar Comentario	08
Usuario Gestor	Visualizar Comentarios	09
	Visualizar Registro	10
	Crear Tarifa	11
	Editar Tarifa	12
	Eliminar Tarifa	13
	Buscar Tarifa	14
	Crear Reserva	15
	Eliminar Reserva	16
	Buscar Reserva	17
	Crear Producto	18
	Editar Producto	19
	Eliminar Producto	20
	Buscar Producto	21
Usuario Administrador	Crear Categoría	22
	Editar Categoría	23
	Eliminar Categoría	24
	Buscar Categoría	25
	Crear Usuario	26
	Editar Usuario	27
	Eliminar Usuario	28
	Buscar Usuario	29
	Crear Servicio	30
	Editar Servicio	31
	Eliminar Servicio	32
	Buscar Servicio	33

Tabla 4: Lista de Casos de uso

La especificación de los casos de uso se encuentran en el anexo.

## 6.2.6 Prototipo de interfaz de usuario

Los Mockups o prototipos de interfaz de usuario nos muestra como quedaría una aplicación antes de su diseño final, de este modo se representa la aplicación de forma visual sin la implementación del diseño.

### Aplicación Web:

Interfaz principal de usuario gestor:



Figura 17: Prototipo de interfaz de usuario gestor

En la interfaz se diferencia un menú a la izquierda con el que se podrá navegar por la aplicación, en la parte central, se visualizan gráficos de estadísticas para el control de la web y en la parte superior derecha se visualiza la foto de perfil del gestor y un menú para editar perfil y cerrar sesión (Véase Figura 17).

Interfaz de producto:

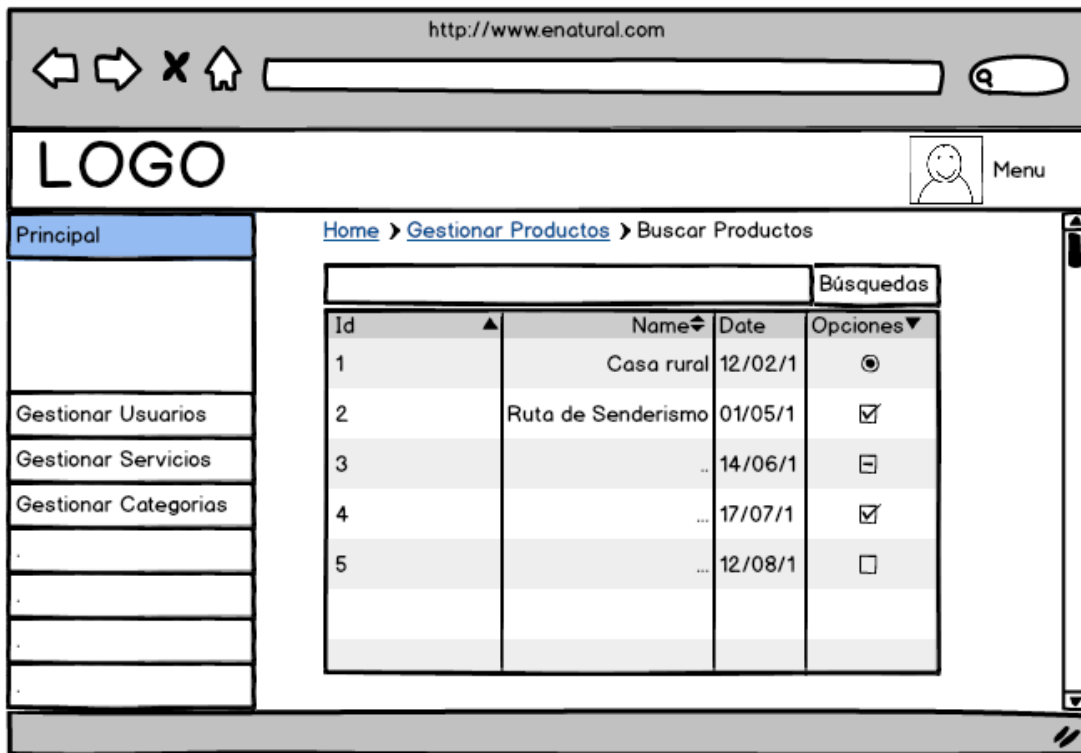


Figura 18: Prototipo de interfaz de producto

En la interfaz se dispone de una tabla con la información asociada a la lista de productos, un buscador de productos y también se podrán realizar diferentes operaciones como editar y eliminar productos (Véase Figura 18).

Interfaz editor web:

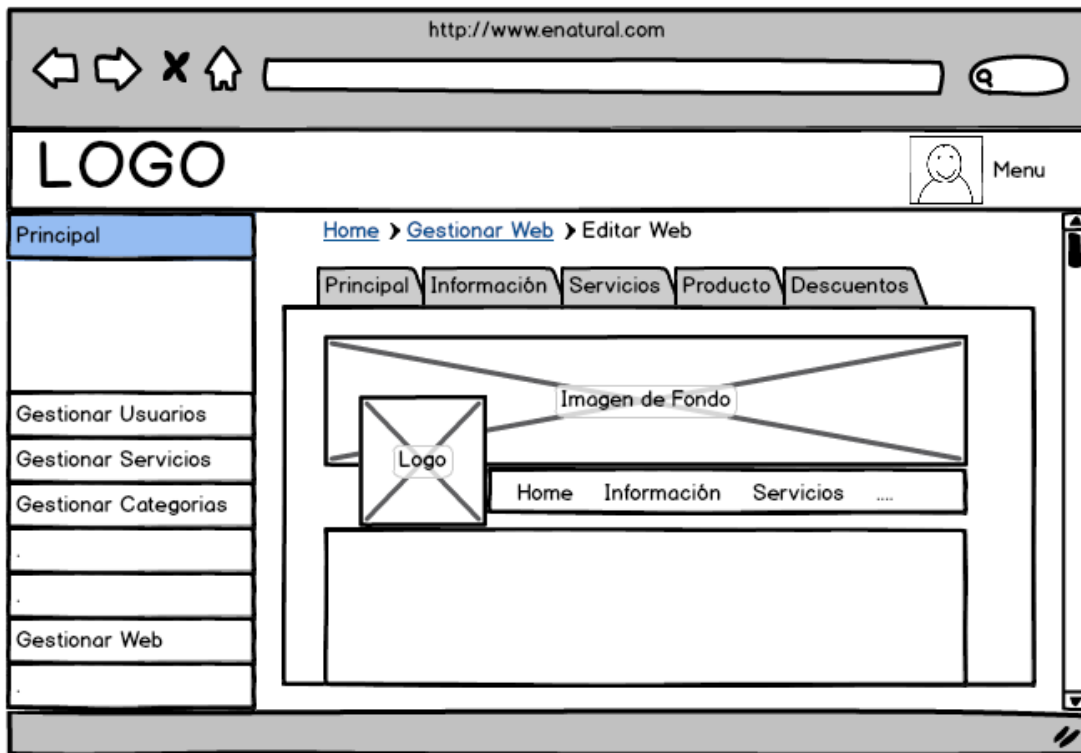


Figura 19: Prototipo de interfaz de Editor Web

En la interfaz del editor web se dispondrá de una plantilla web que el usuario podrá ir configurando y se podrá visualizar a través del buscador del sitio (Véase Figura 19).



Interfaz Página principal:

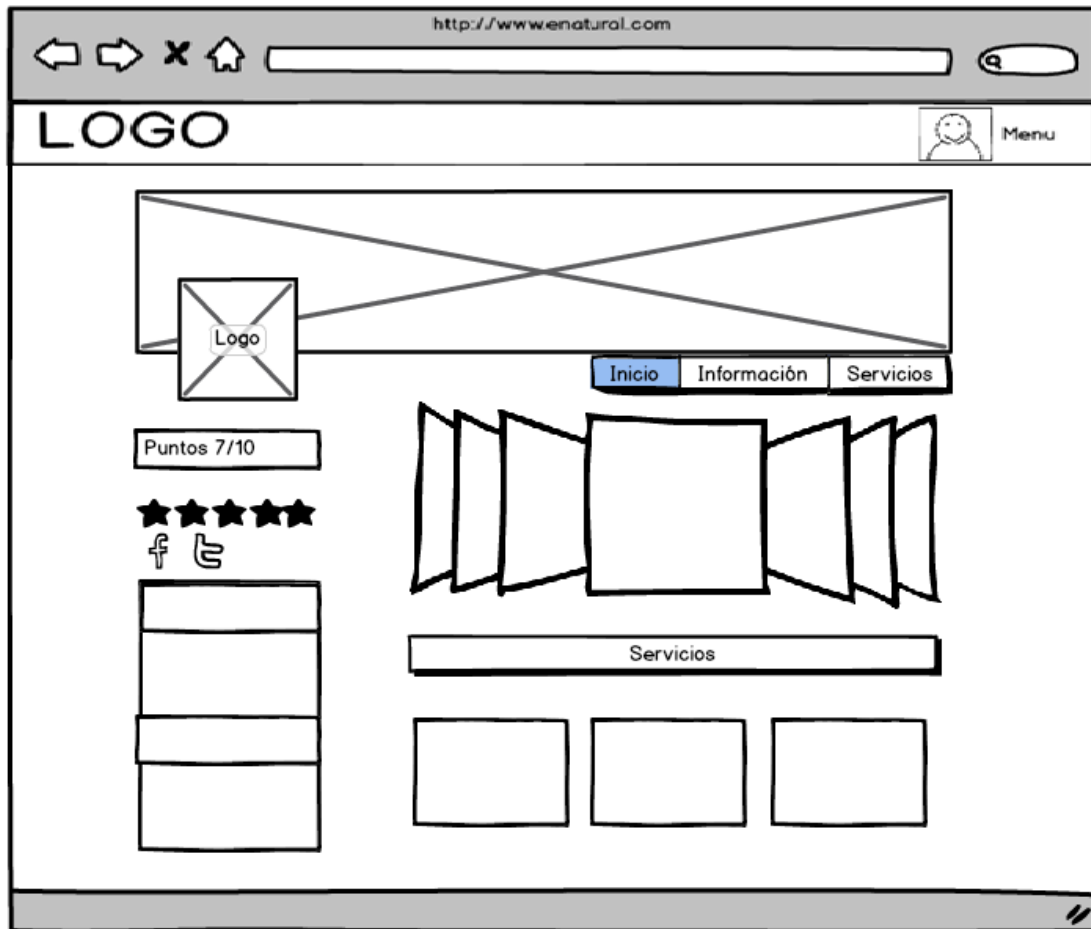


Figura 20: Prototipo de página web principal

Muestra como quedaría la página principal de la empresa una vez creada (Véase Figura 20).

## 6.3 Diseño arquitectónico

La arquitectura es de tipo cliente-servidor. Este tipo de arquitecturas están basadas en que uno de los lados se encuentra el servidor y por el otro lado se encuentra el cliente. Los clientes envían solicitudes al servidor y este último procesa las solicitudes (Véase Imagen 8).

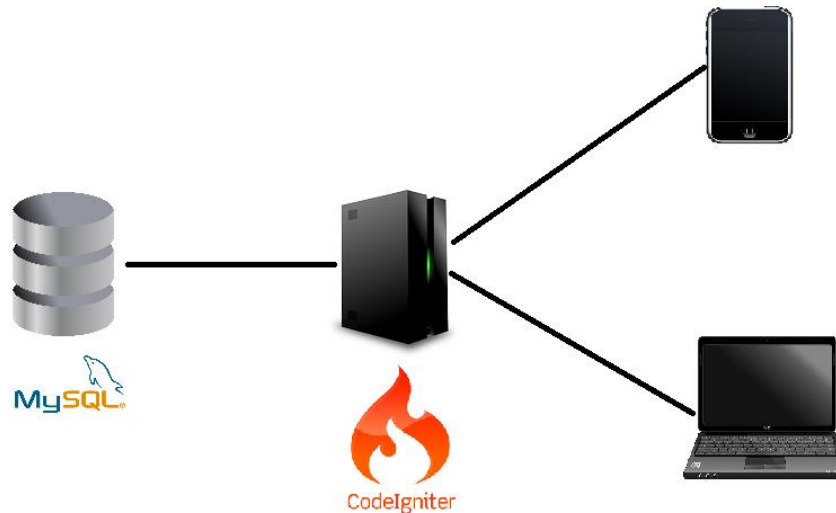


Imagen 8: Diseño arquitectónico

### 6.3.1 Características de esta arquitectura

Seguidamente se enumeran las principales características de la arquitectura:

- El servidor puede dar respuesta a varios clientes a la vez
- El cliente web no depende del sistema operativo que se esté ejecutando.
- Cualquier cambio producido en el servidor o en la base de datos no tendrá repercusión para el cliente.
- Existen varias vías por las que el cliente podrá comunicarse con el servidor, el cliente web accederá a los servicios mediante un portal web. Otra vía de acceso podrán ser dispositivos móviles.

**Funciones de los clientes:**

- Mostrar los datos al usuario, como servicios y productos.
- Enviar datos al servidor.
- Enviar archivos al servidor.

**Funciones del servidor:**

- Procesar las peticiones que realizan los clientes.
- Validar formularios.
- Realizar operaciones crear, editar, eliminar y actualizar a la base de datos.
- Devolver respuestas a los clientes.

### 6.3.2 Arquitectura de la aplicación

Este proyecto utiliza el patrón modelo-vista-controlador impuesto por el framework utilizado (Véase Imagen 9).

Que en este caso se trata del framework Codeigniter.

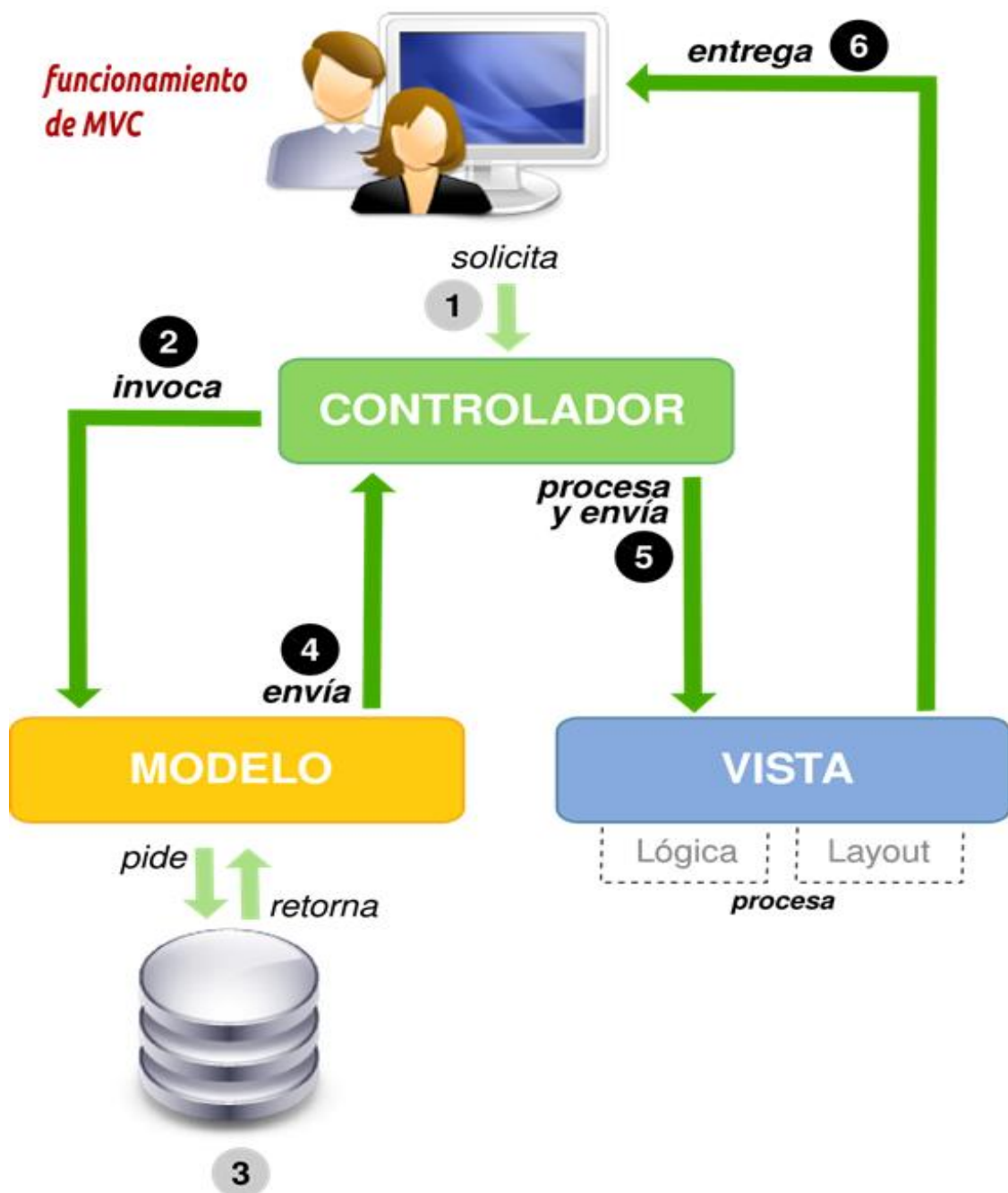


Imagen 9: Patrón modelo-Vista-Controlador

**Vista (capa de presentación):** es la capa superior y representa la interfaz del usuario de la aplicación. Esta interfaz contiene código html, javascript, php, jquery. El formato de presentación viene dado por las hojas de estilo CSS. La función de esta capa es la de interactuar con el usuario permitiendo que se comunique de forma amigable con el sistema.

**Controlador (capa de negocio):** esta capa se encarga de gestionar y administrar las peticiones del usuario. Hace de intermediario entre la capa de presentación y la capa de datos. Es la lógica de la aplicación.

**Modelo (capa de datos):** es la capa más inferior donde se almacenan todos los datos del sistema. Es una capa intermedia que contiene métodos que interactúan con la base de datos.

#### **Framework Codeigniter:**

El **Modelo, Vista, Controlador** es típicamente utilizado para la creación de aplicaciones web y no sólo CodeIgniter lo implementa, sino también otra serie de frameworks de desarrollo web, en PHP u otros lenguajes. Es interesante porque separa en varios grupos las complejidades de las distintas partes que componen una página web, como la vista y la lógica, así como el acceso a la base de datos.

El Modelo - Vista - Controlador (en inglés Model - View - Controller) es un patrón de desarrollo o un estilo de arquitectura de software que separa el código fuente de las aplicaciones en tres grupos:

#### **Modelo:**

Todo el código que tiene que ver con el acceso a base de datos. En el modelo mantendremos encapsulada la complejidad de nuestra base de datos y simplemente crearemos funciones para recibir, insertar, actualizar o borrar información de nuestras tablas. Al mantenerse todas las llamadas a la base de datos en un mismo código, desde otras partes del programa podremos invocar las funciones que necesitemos del modelo y éste se encargará de procesarlas. En el modelo nos podrán preocupar cosas como el tipo de base de datos con la que trabajamos, o las tablas y sus relaciones, pero desde las otras partes del programa simplemente llamaremos a las funciones del modelo sin importarnos qué tiene que hacer éste para conseguir realizar las acciones invocadas.

#### **Vista:**

La vista codifica y mantiene la presentación final de nuestra aplicación de cara al usuario. Es decir, en la vista colocaremos todo el código HTML, CSS, Javascript, etc. que se tiene que generar para producir la página tal cual queremos que la vea el usuario. En la práctica la vista no sólo sirve para producir páginas web, sino también cualquier otra salida que queramos enviar al usuario, en formatos o lenguajes distintos, como pueden ser feeds RSS, archivos JSON, XML, etc.

#### **Controlador:**

El controlador podríamos decir que es la parte más importante, porque hace de enlace entre el modelo, la vista y cualquier otro recurso que se tenga que procesar en el servidor para generar la página web. En resumen, en el controlador guardamos la lógica de nuestras páginas y realizamos todas las acciones que sean necesarias para generarlas, ayudados del modelo o la vista (Véase Imagen 10).

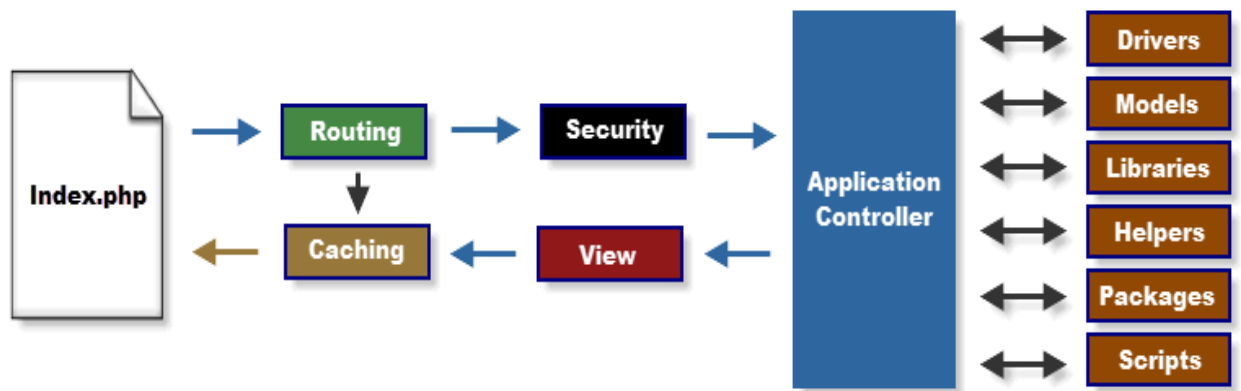


Imagen 10: Funcionamiento de los Controladores

### 6.3.3 Detalles de la implementación.

#### Estructura de la Aplicación:

La estructura de la aplicación se realiza bajo las recomendaciones impuestas por el Framework Codeigniter, mediante el patrón modelo-vista-controlador, la aplicación se subdivide en módulos (Véase Imagen 11).



Imagen 11: Estructura de directorios de la aplicación

La carpeta **assets** se añaden todos los ficheros de librerías, imágenes y css de la aplicación web.

La carpeta **Sites** irán todos los ficheros de imágenes que suban los gestores al servidor, las empresas se subdividirán en carpetas indicando el identificador de la empresa en la base de datos.

La carpeta **System** almacena todos los archivos que hacen que CI funcione.

La carpeta de aplicación es casi idéntica al contenido de la carpeta de sistema para que el usuario pueda tener archivos que sean particulares de la aplicación, por ejemplo si un usuario quiere cargar un atajo en una aplicación, lo ubicaría en la carpeta **system/application/helpers** en lugar de la de **system/helpers** (Véase Imagen 12).

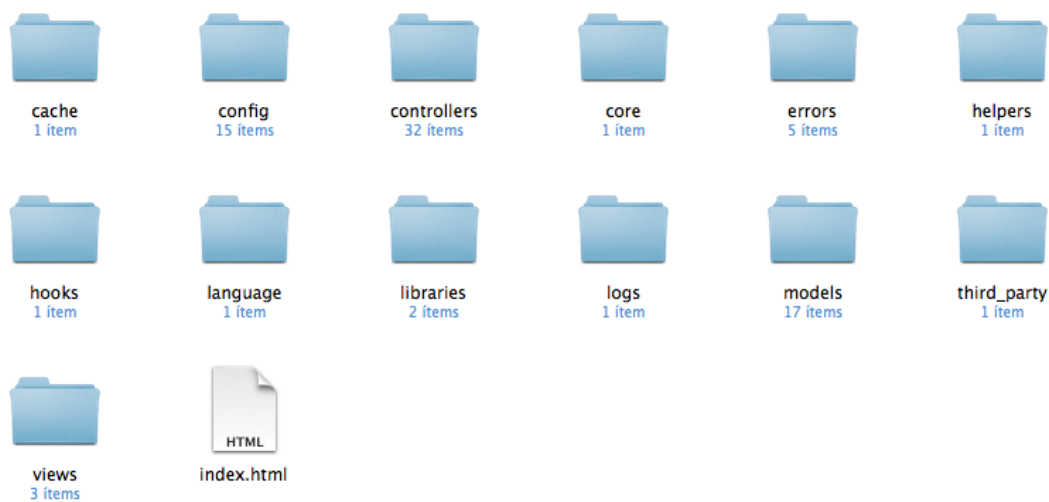


Imagen 12: Estructura de directorios de Codeigniter

La carpeta **Config** almacena todos los archivos de configuración relevantes para la aplicación. Eso incluye qué librerías debe auto-cargar la aplicación y detalles de la base de datos.

La carpeta **Controllers** almacena todos los controladores de la aplicación.

La carpeta **Errors** almacena todas las páginas de patrones de errores de la aplicación. Cuando un error aparece, se genera una página de error a partir de uno de estos patrones.

La carpeta **Hooks** contiene extensiones que modifican el funcionamiento de los archivos del núcleo de CI.

La carpeta **Language** contiene líneas de texto que pueden ser cargadas a través del módulo de lenguaje para crear sitios multilinguaje.

La carpeta **Libraries** contiene todas las librerías o módulos específicos para la aplicación.

La carpeta **Models** almacena los modelos de la aplicación.

La carpeta **Views** almacena las vistas de la aplicación.



### 6.3.4 Diagrama de Despliegue

Cuando el sistema es finalizado ha de ser desplegado en los componentes físicos (hardware) que deben cumplir unos requisitos software. Dichos componentes necesarios para este despliegue son (Véase Imagen 13):

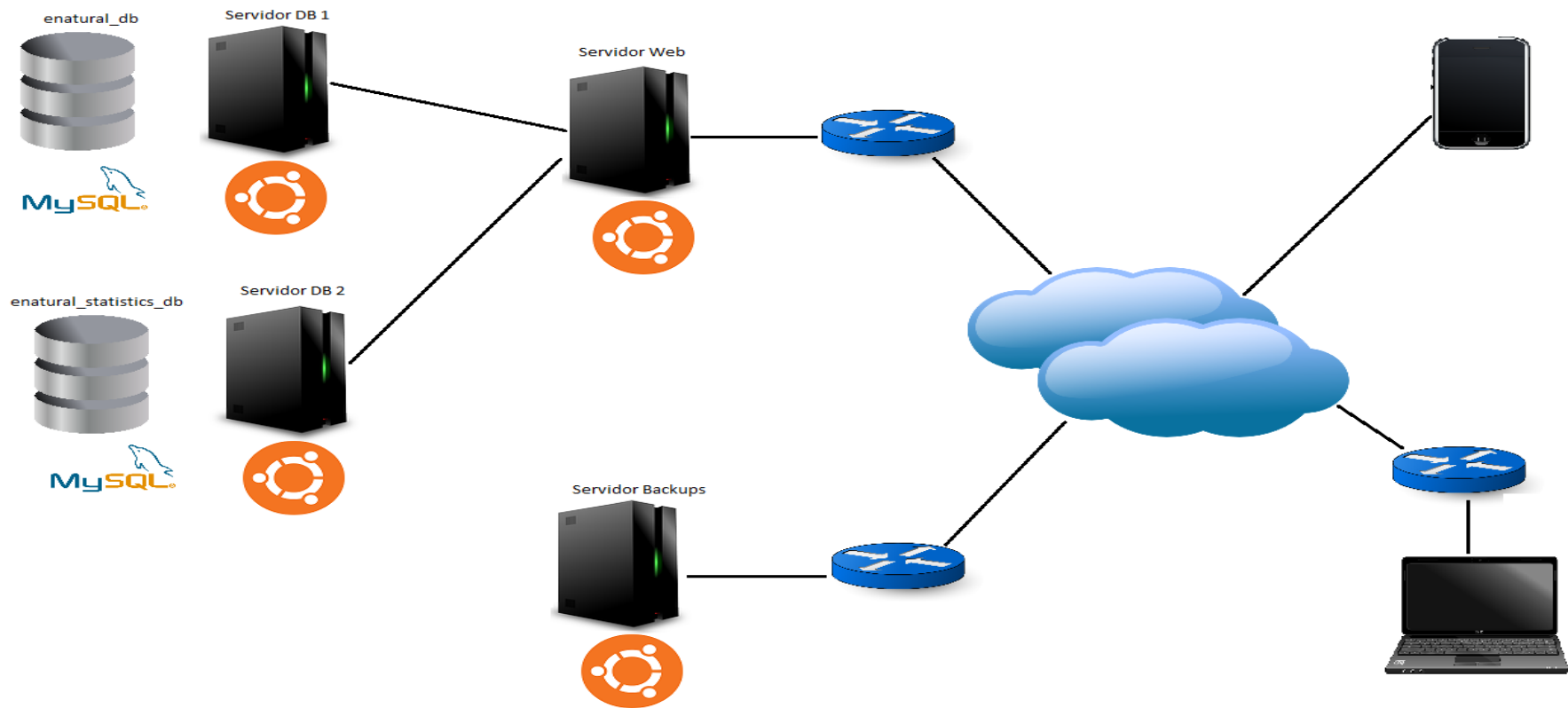


Imagen 13: Diagrama de despliegue

## **Servidor principal**

Requisitos hardware:

- Procesador: 2x Intel® Xeon® E7-4820, 8C, 2.00GHz, 18M Cache, 5.86GT/s, 105W TDP, Turbo, HT, DDR3-980MHz
- Memoria: 16GB Memory for 2 CPUs, 1066MHz (4x4GB 2R LV RDIMMs), 2 Memory Risers, 1333MHz DIMMs
- Almacenamiento: 4x 1TB, SATA, 2.5-in, 7.2K RPM Hard Drive
- Tarjeta controladora RAID: PERC H700 Integrated RAID Controller, 512MB Cache
- Conectividad RAID: RAID5 for PERC H200/H700, 4 HDDs
- Tarjetas de red: Intel Gigabit ET Dual Port Server Adapter, Cu, PCIe-4

## **Servidor de Base de datos 1:**

Requisitos hardware:

- Procesador: 2x Intel® Xeon® E7-4820, 8C, 2.00GHz, 18M Cache, 5.86GT/s, 105W TDP, Turbo, HT, DDR3-980MHz
- Memoria: 16GB Memory for 2 CPUs, 1066MHz (4x4GB 2R LV RDIMMs), 2 Memory Risers, 1333MHz DIMMs
- Almacenamiento: 4x 2TB, SATA, 2.5-in, 7.2K RPM Hard Drive
- Tarjeta controladora RAID: PERC H700 Integrated RAID Controller, 512MB Cache
- Conectividad RAID: RAID5 for PERC H200/H700, 4 HDDs
- Tarjetas de red: Intel Gigabit ET Dual Port Server Adapter, Cu, PCIe-4

## **Servidor de Base de datos 2:**

Requisitos hardware:

- Procesador: 2x Intel® Xeon® E7-4820, 8C, 2.00GHz, 18M Cache, 5.86GT/s, 105W TDP, Turbo, HT, DDR3-980MHz
- Memoria: 16GB Memory for 2 CPUs, 1066MHz (4x4GB 2R LV RDIMMs), 2 Memory Risers, 1333MHz DIMMs
- Almacenamiento: 4x 2TB, SATA, 2.5-in, 7.2K RPM Hard Drive
- Tarjeta controladora RAID: PERC H700 Integrated RAID Controller, 512MB Cache
- Conectividad RAID: RAID5 for PERC H200/H700, 4 HDDs
- Tarjetas de red: Intel Gigabit ET Dual Port Server Adapter, Cu, PCIe-4

## **Servidor de Copia:**

Requisitos hardware:

- Procesador: Intel® Xeon® E3-1220, 4C/4T, 3.10GHz, 8M Cache, 80W TDP, Turbo
- Memoria: 4GB Memory (1x4GB), 1600Mhz, Dual Ranked, Low Volt UDIMM
- Almacenamiento: 2x 1TB, SATA, 2.5-in, 7.2K RPM Hard Drive
- Tarjeta controladora RAID: PERC H200 Integrated RAID Controller

- Conectividad RAID: RAID1 for PERC H200/H700, 2 HDDs Tarjetas de red: Intel® PRO/1000PT GbE Single Port Server Adapter, Cu, PCIe-1

**Requisitos software:**

- SO: Ubuntu Server
- Copias de seguridad: Bacula

**Clientes móviles:**

- Cualquier dispositivo movil.

**Clientes Navegadores (versiones más recientes):**

- Mozilla Firefox
- Safari
- Google Chrome

### 6.3.5 Diagramas de clases

Seguidamente se mostrará los diagramas de clase correspondientes a los módulos de la aplicación:



Categorías:

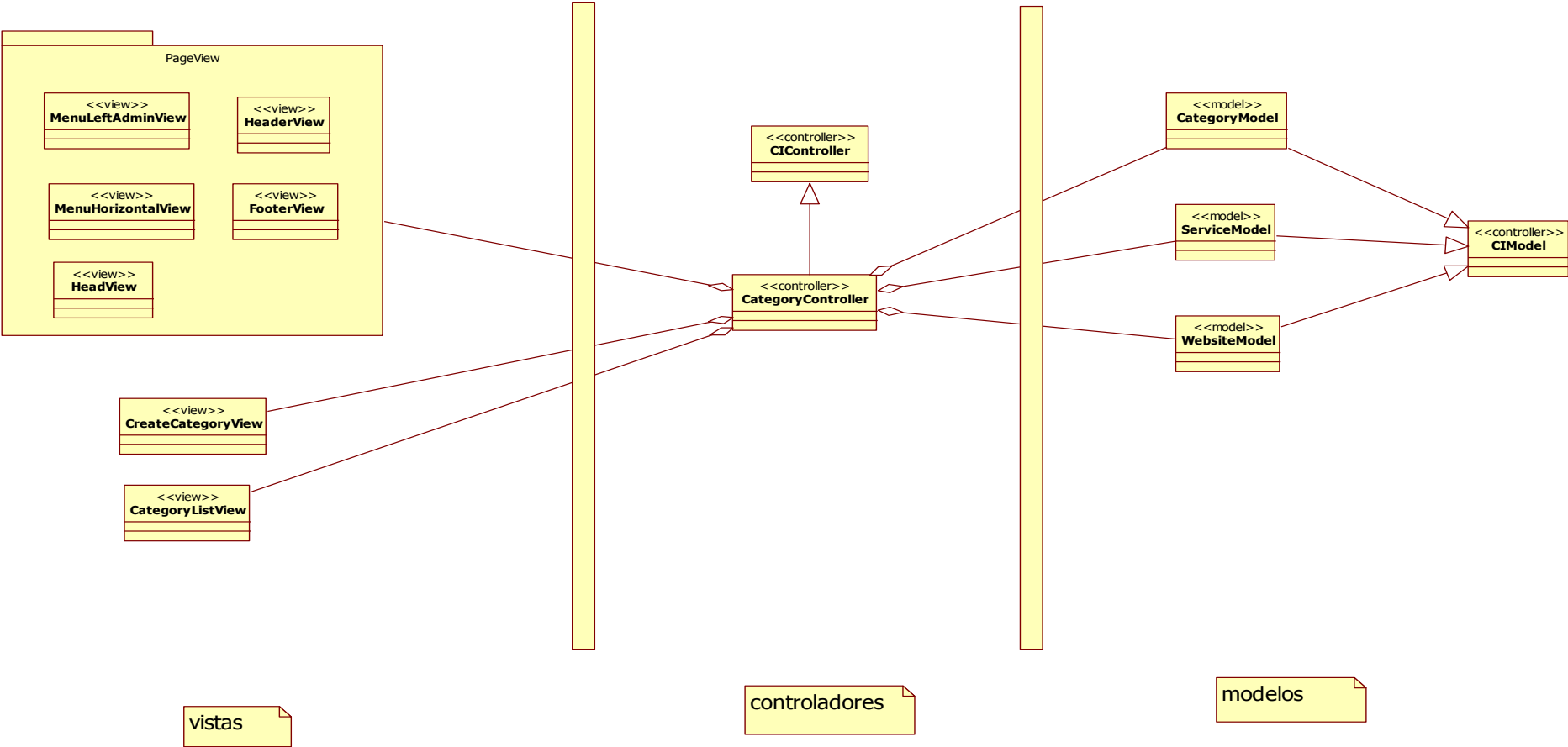


Figura 21: Diagrama de clases Categorías

Home:

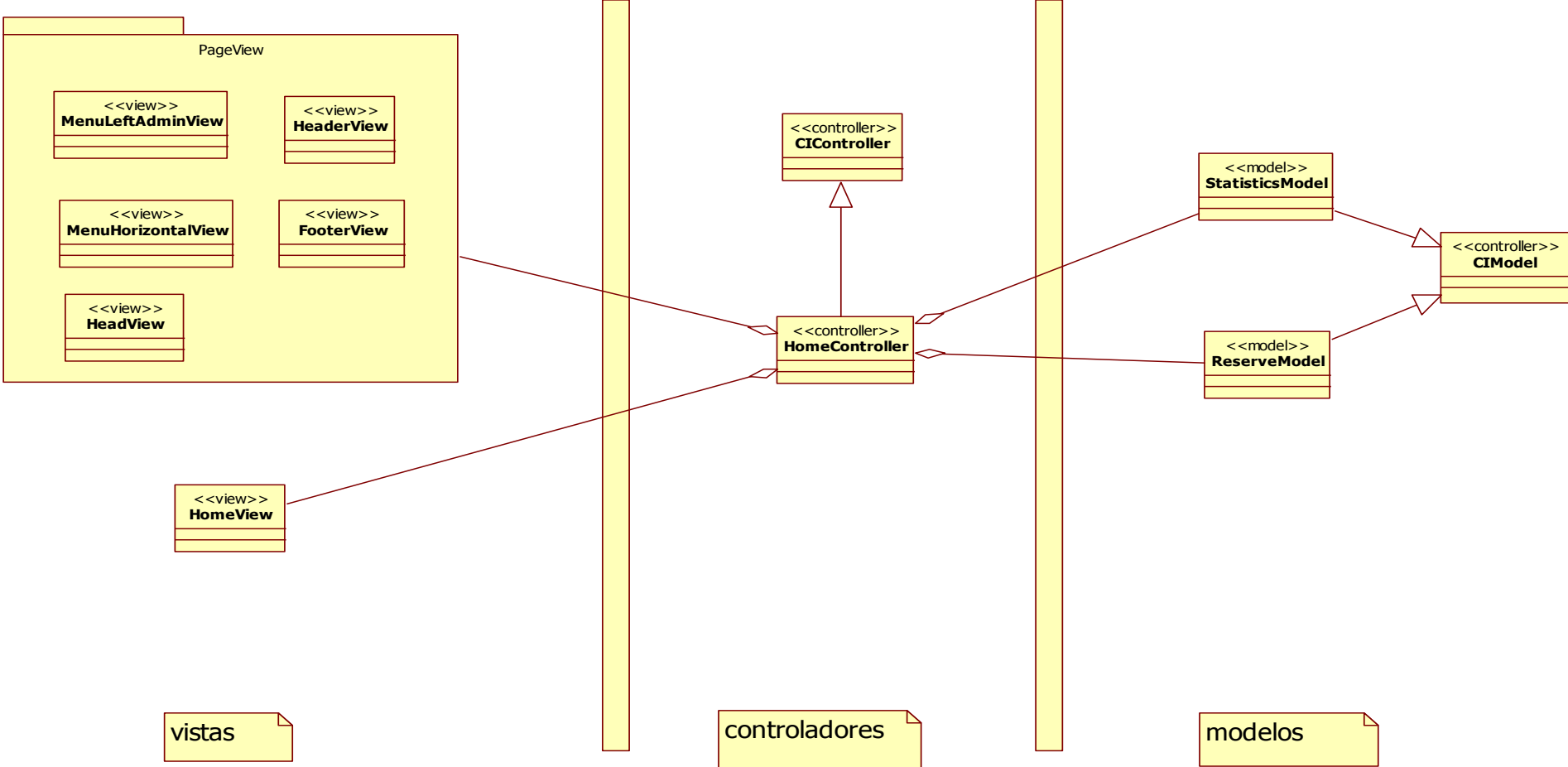


Figura 22: Diagrama de clases Home

Registro de eventos:

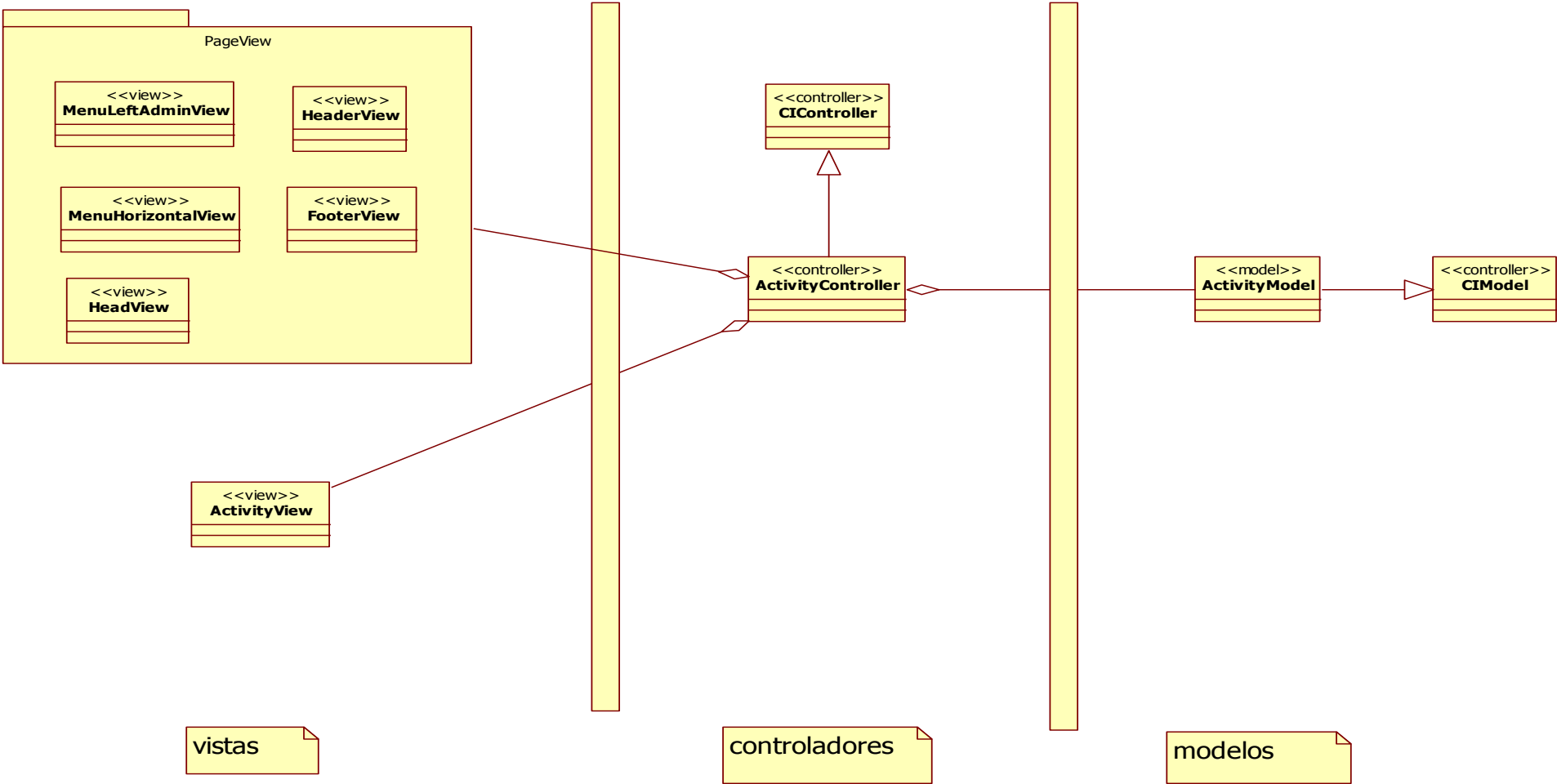


Figura 23: Diagrama de clases Registro de eventos





Comentarios:

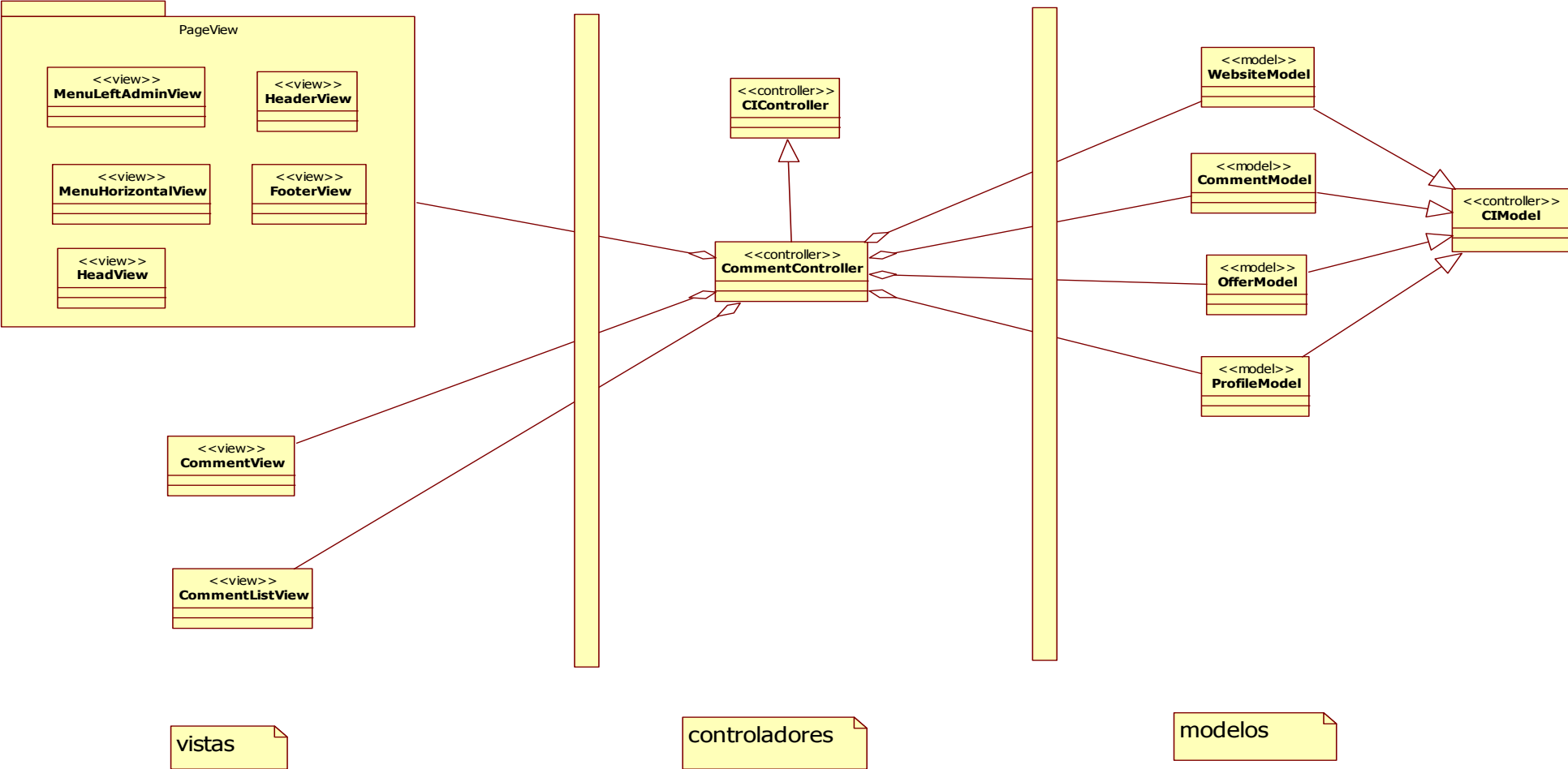


Figura 24: Diagrama de clases comentarios

Productos:

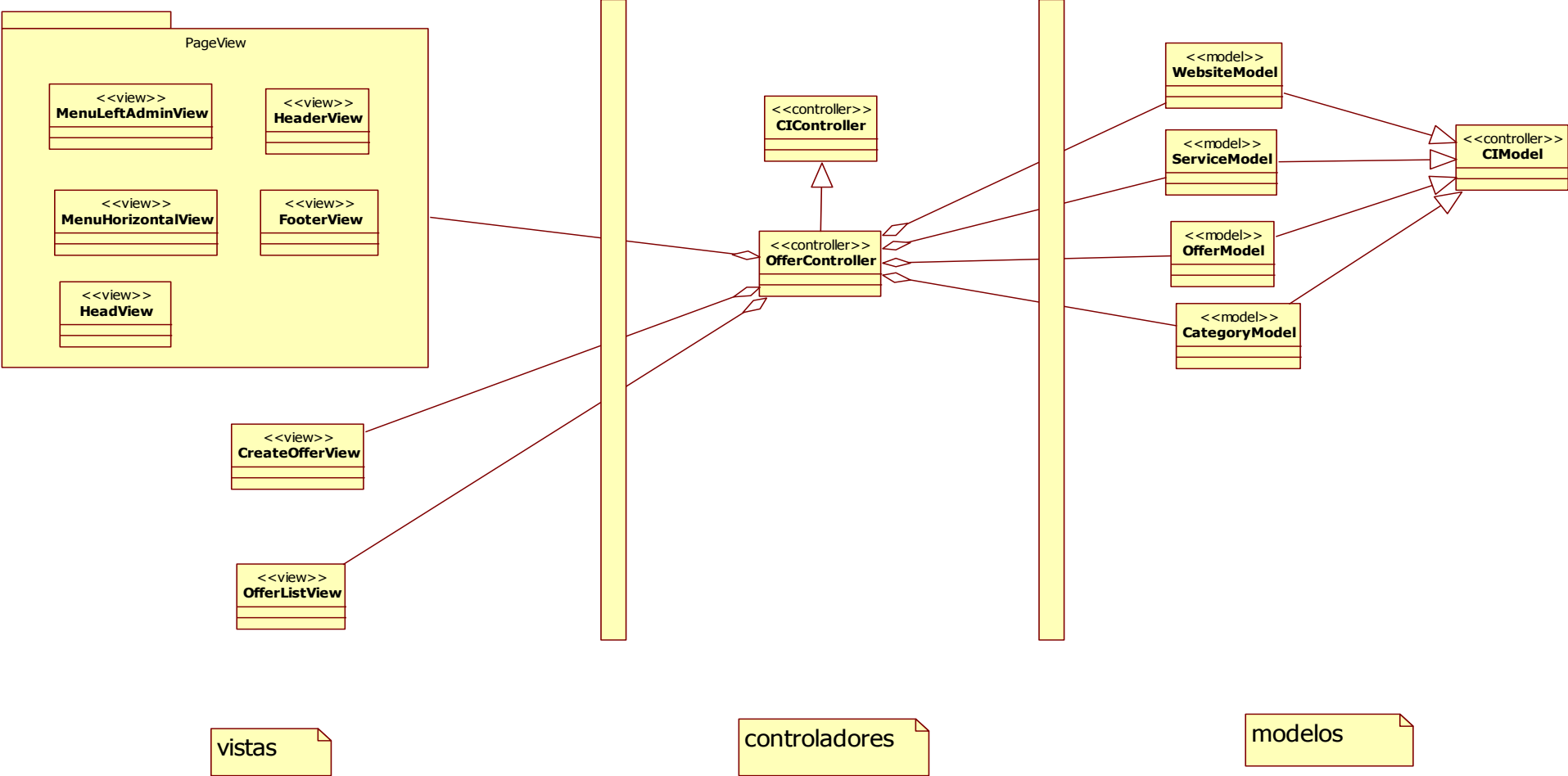


Figura 25: Diagrama de clases productos



Sites:

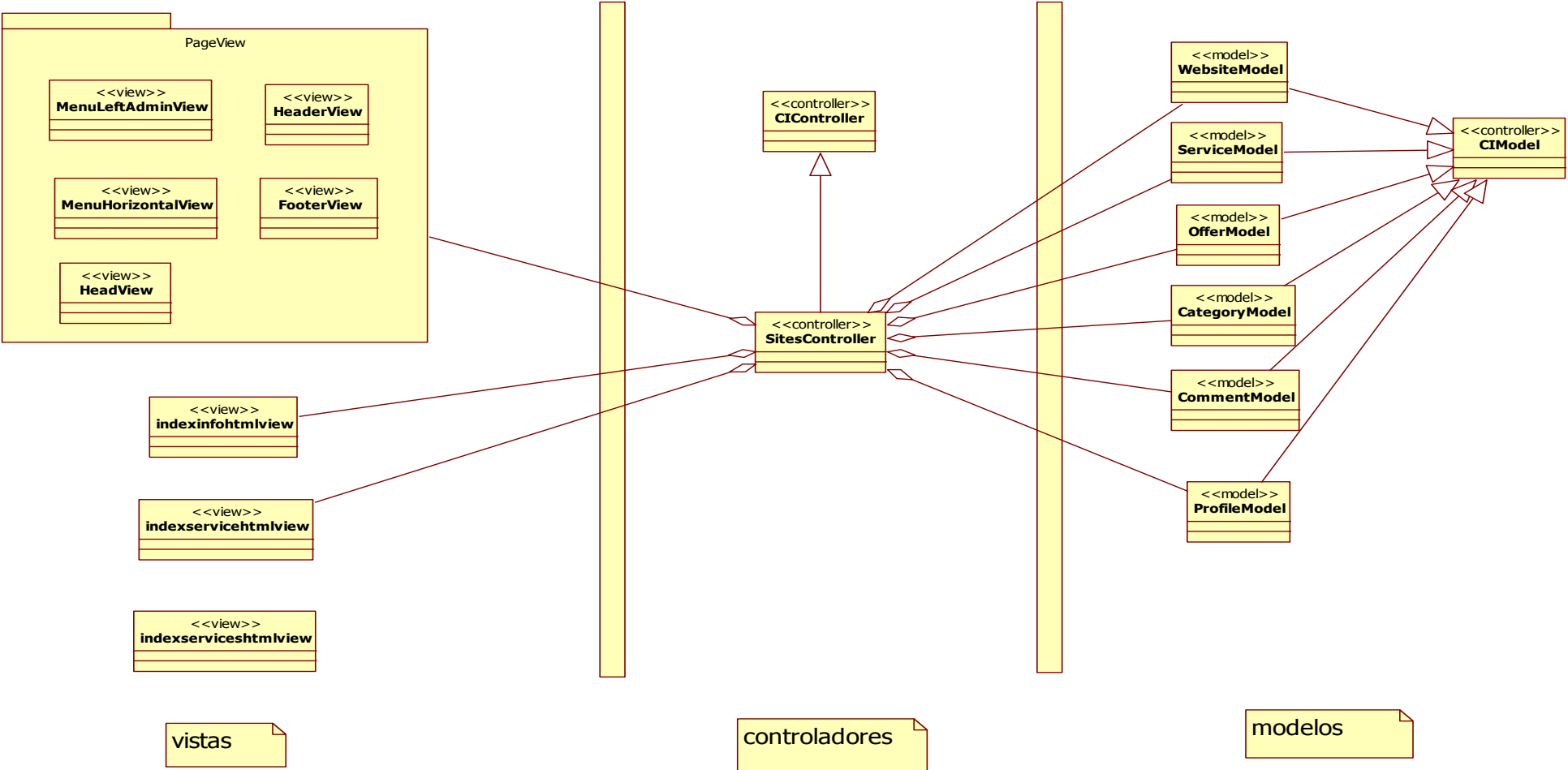


Figura 27: Diagrama de clases sites



## WebEditorInfo:

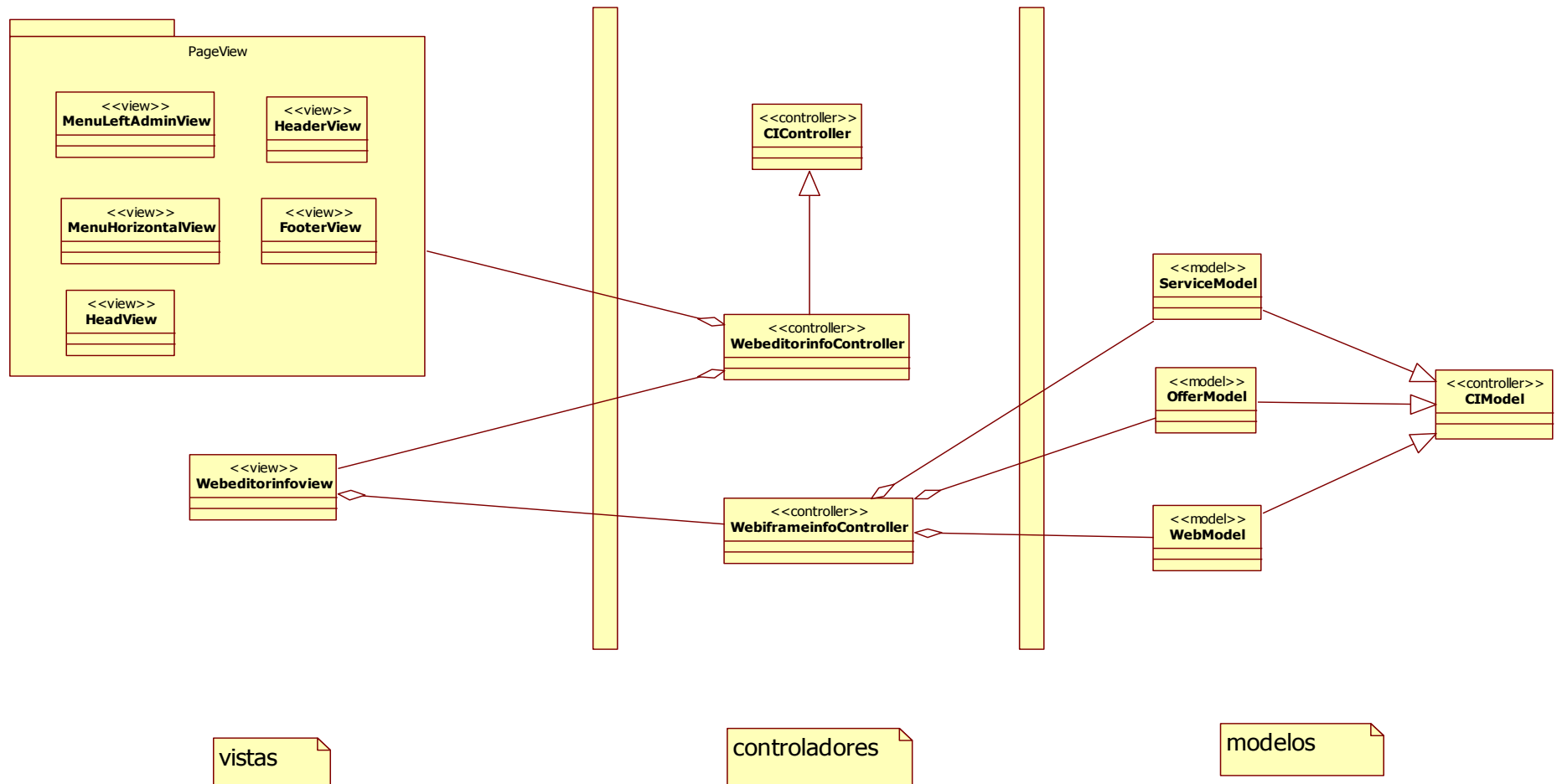


Figura 29: Diagrama de clases webeditorinfo

WebEditorServices:

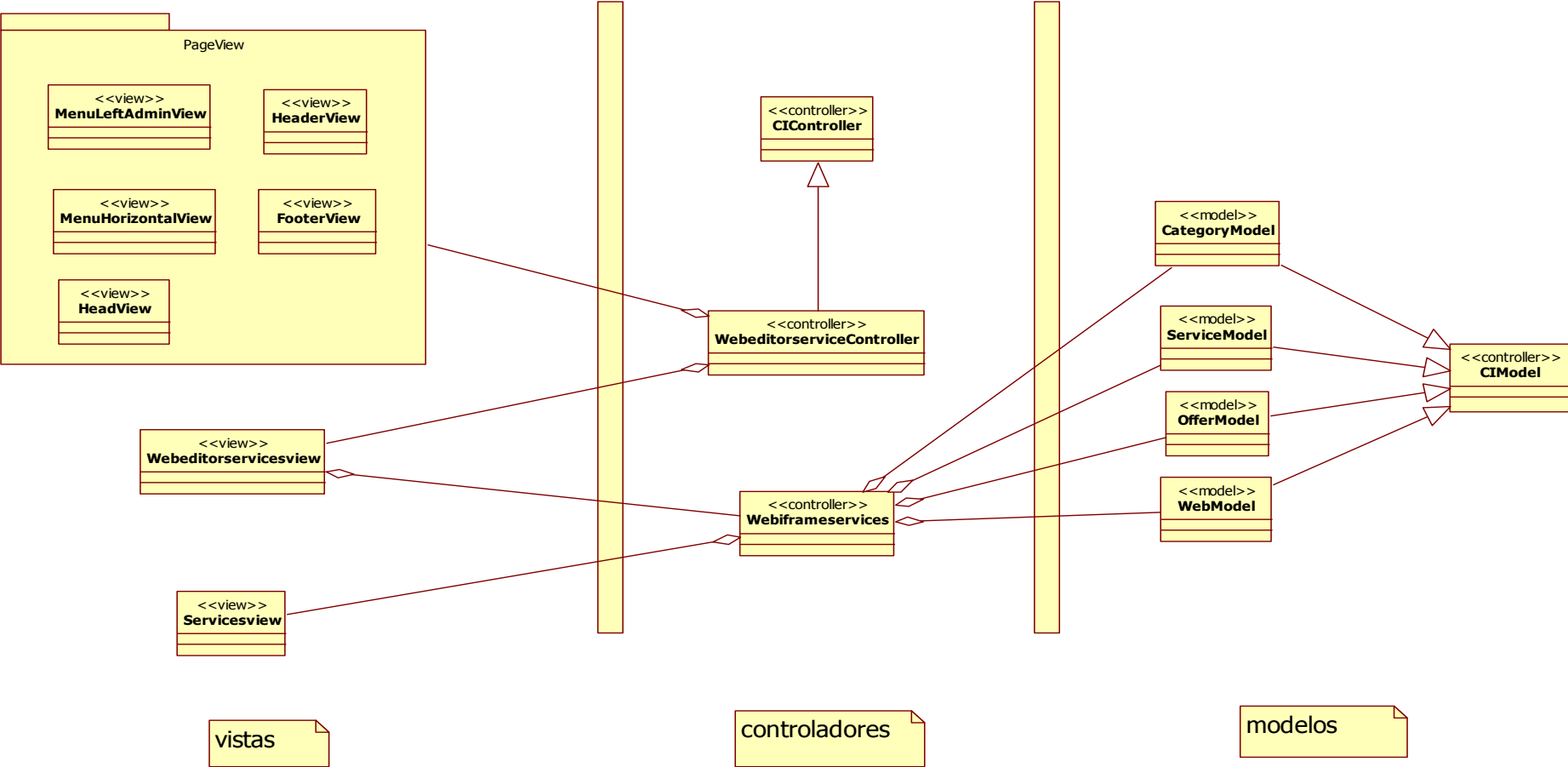


Figura 30: Diagrama de clases webeditorservices

Controlador web:

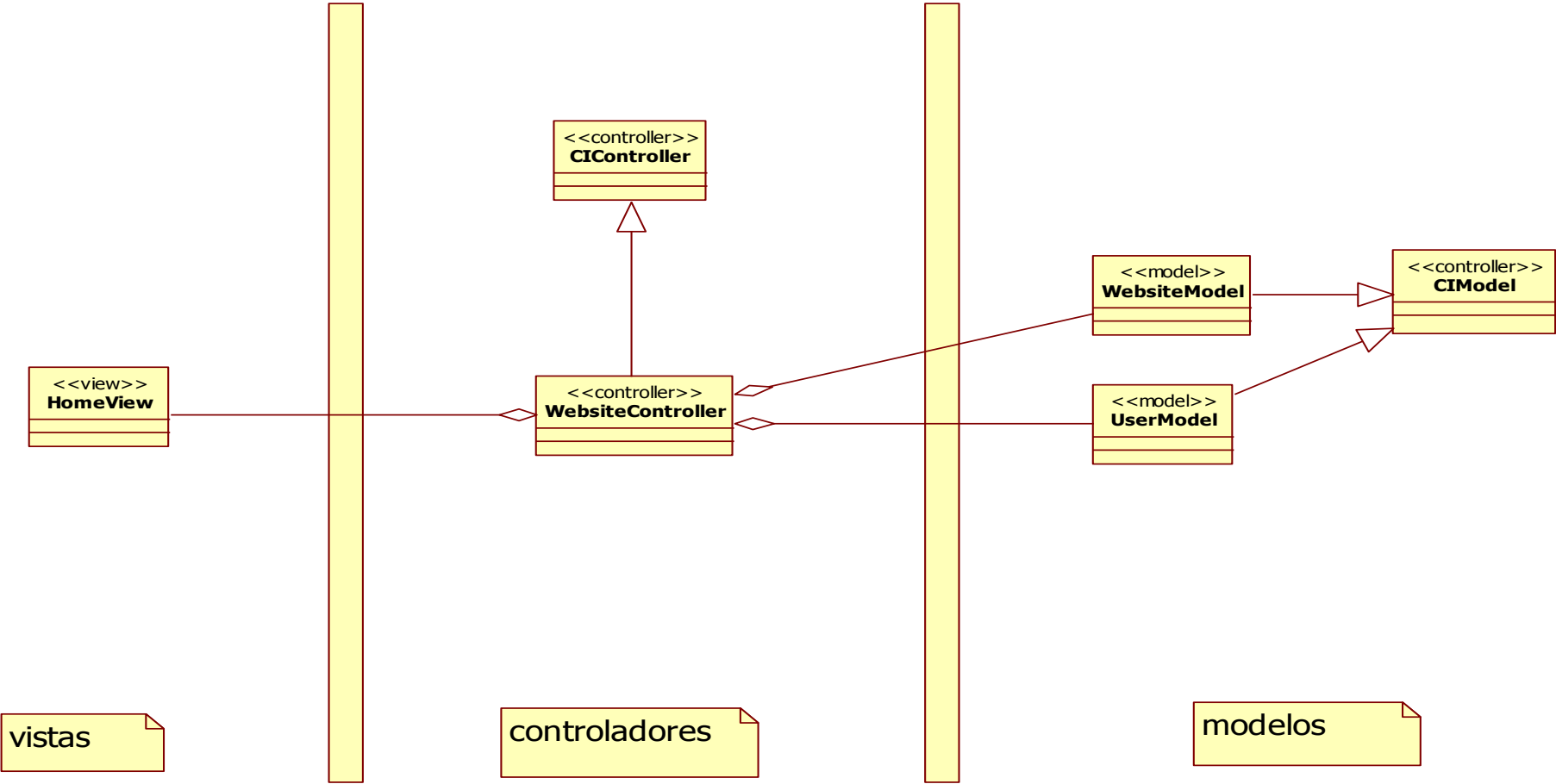


Figura 31: Diagrama de clases website



# Tarifa

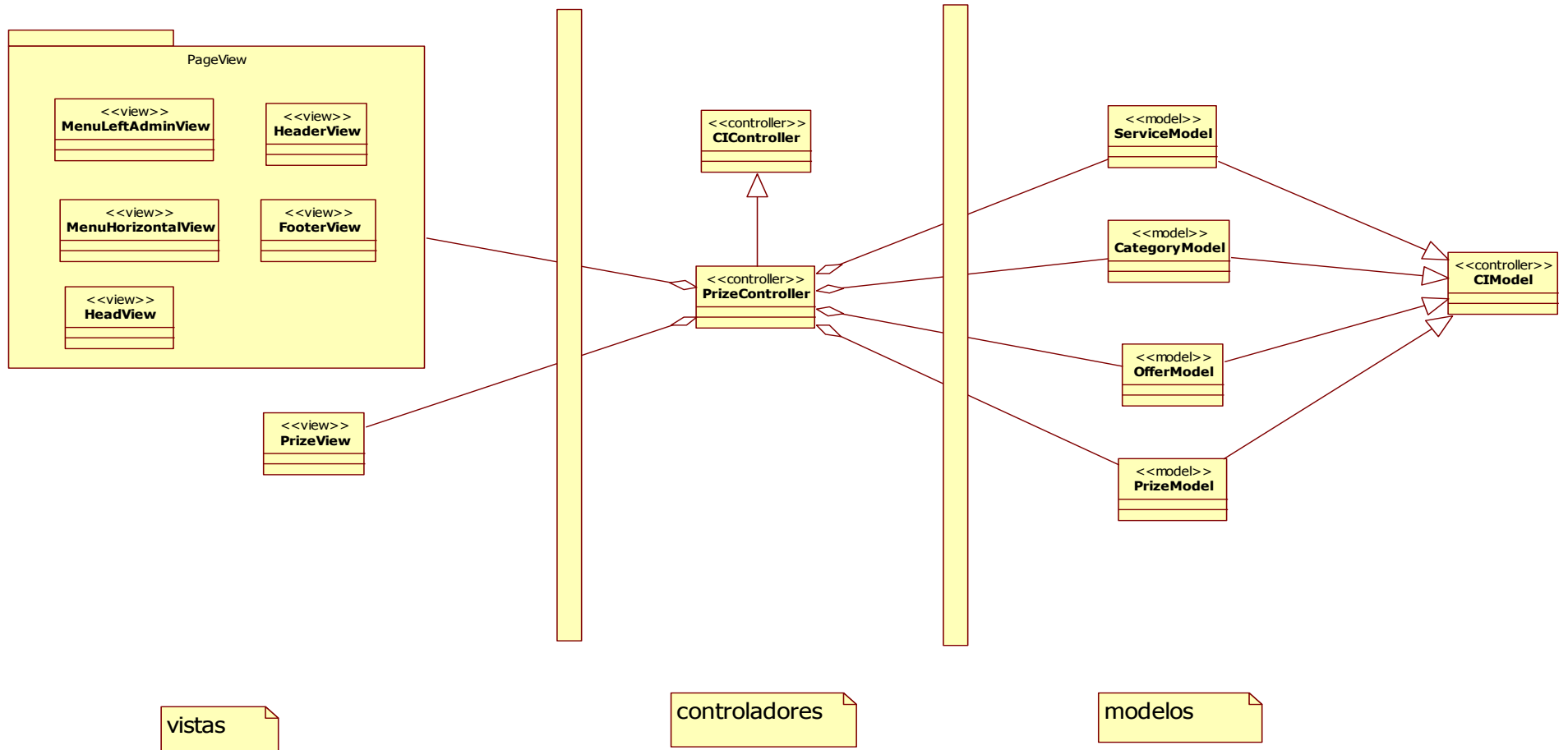


Figura 32: Diagrama de clases tarifa

Registro login:

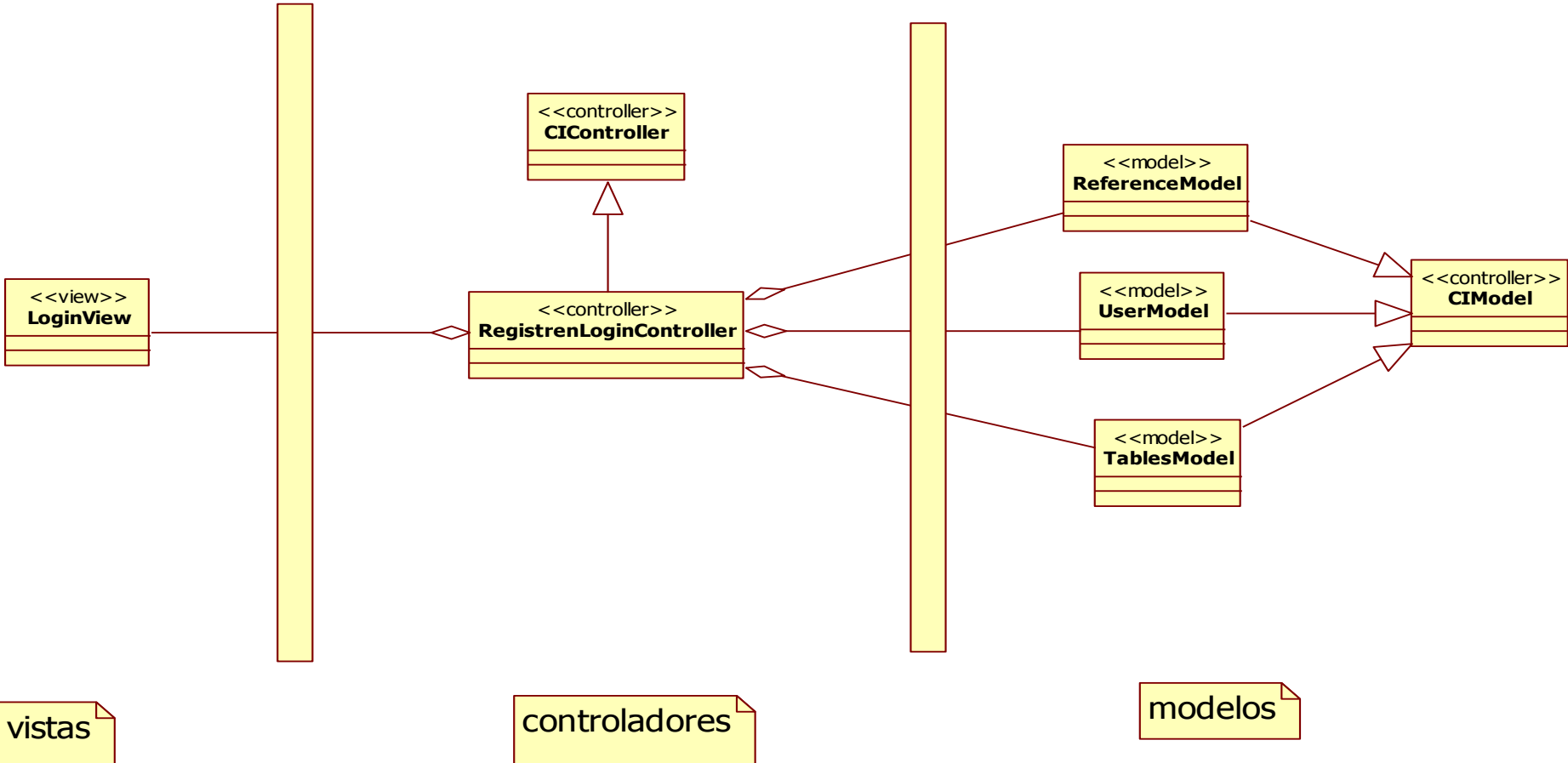


Figura 33: Diagrama de clases login

Reservas:

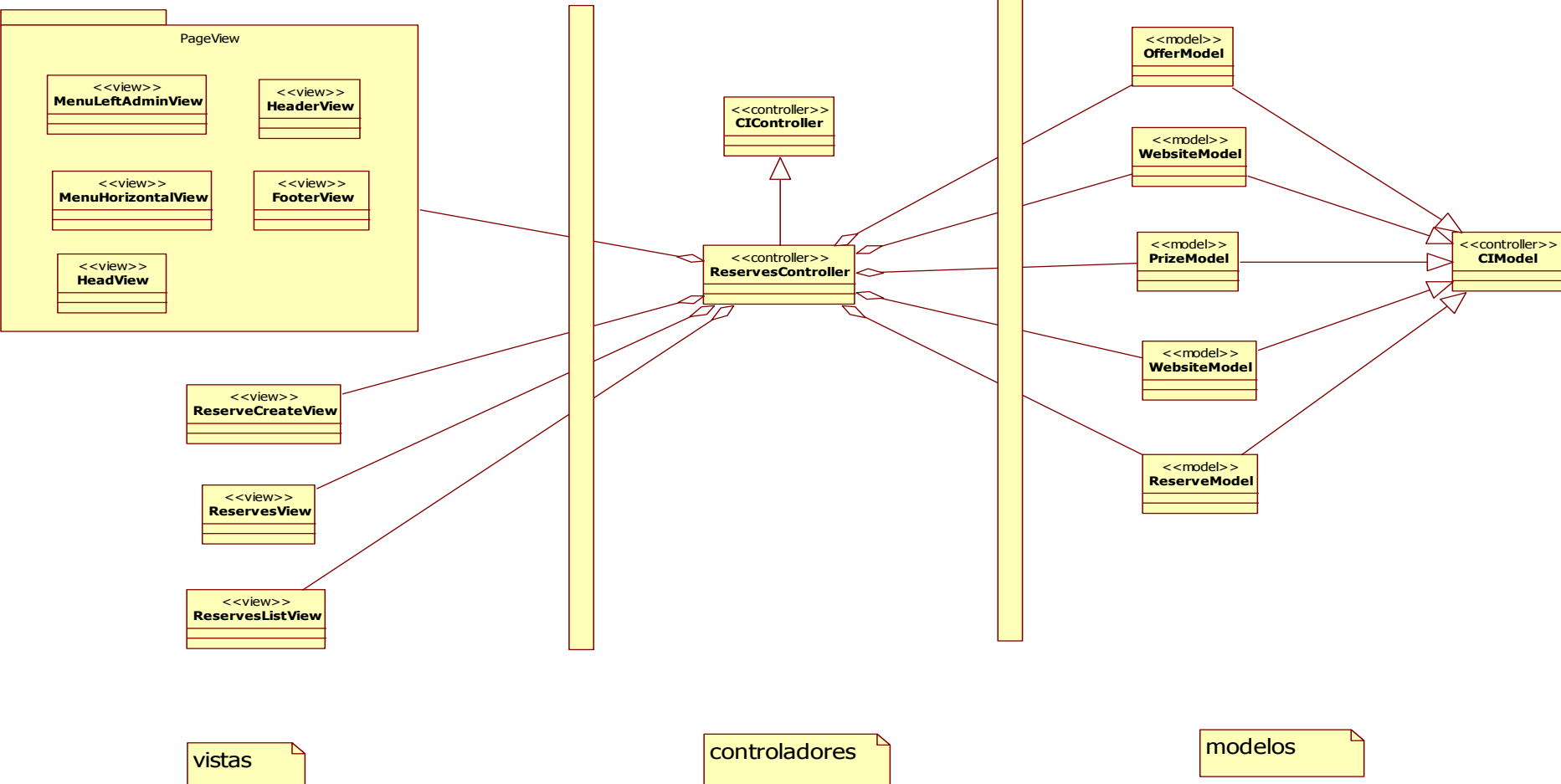


Figura 34: Diagrama de clases reserva

Save index:

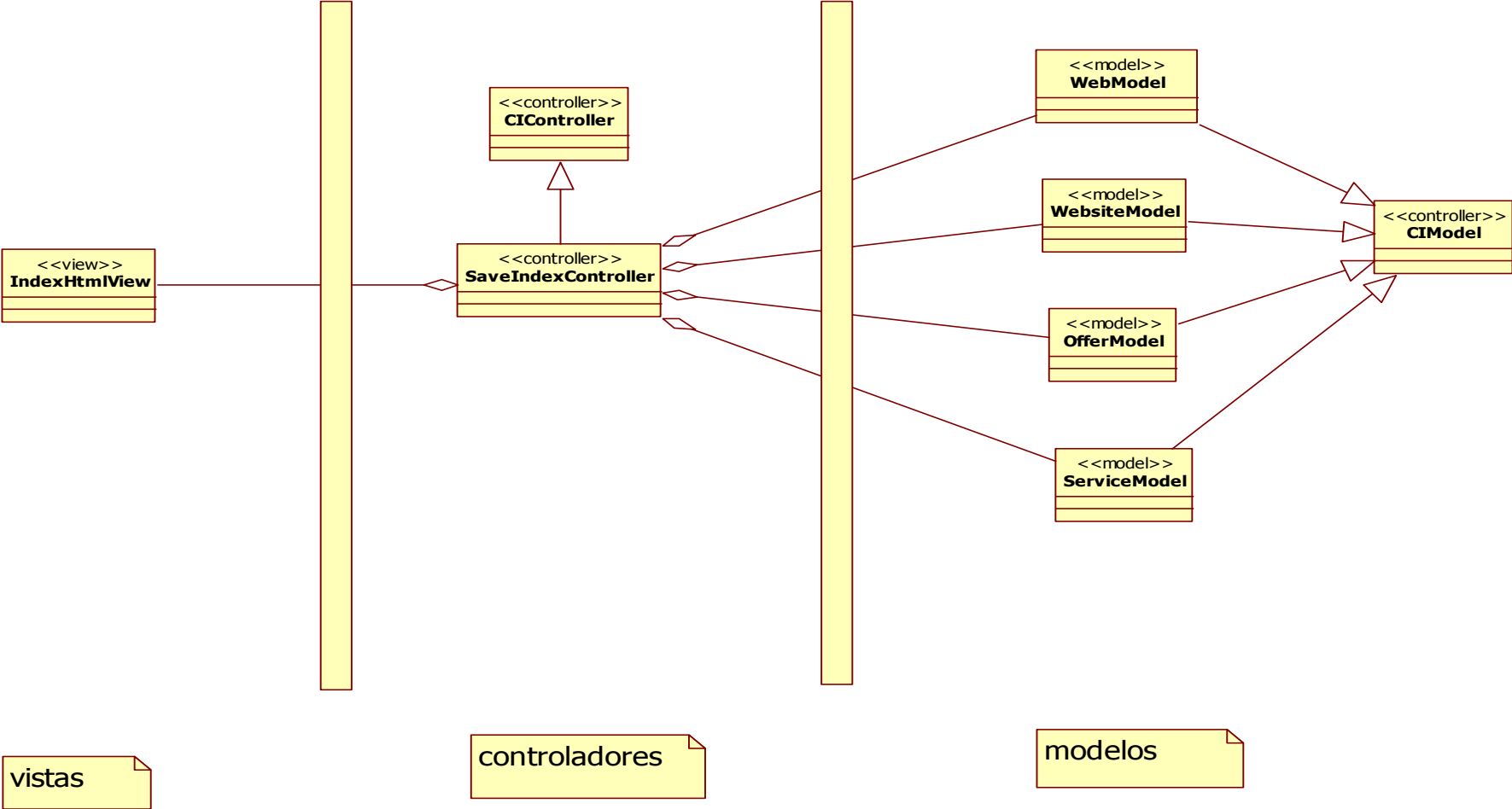


Figura 35: Diagrama de clases saveindex

Usuario:

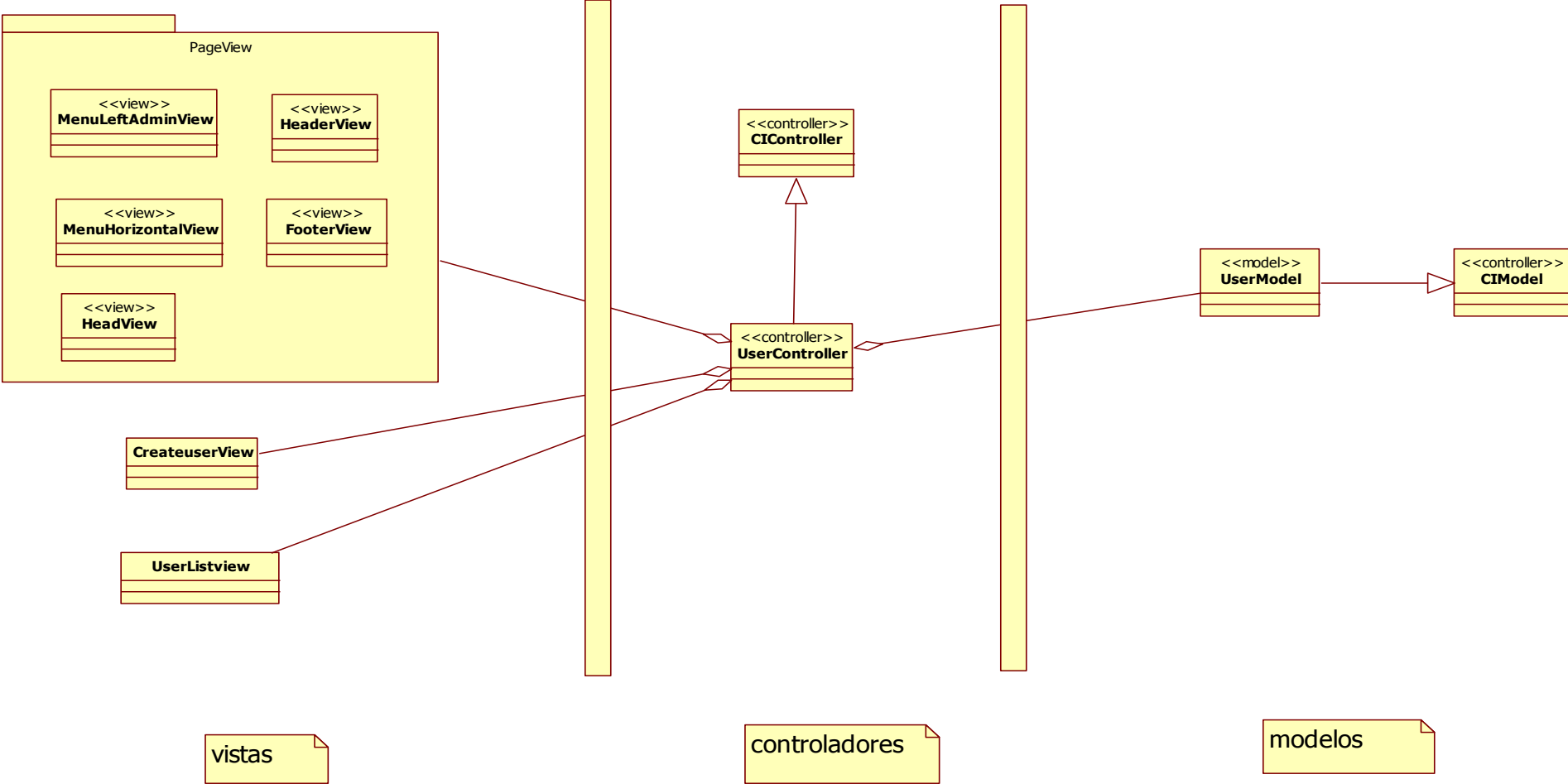


Figura 36: Diagrama de clases usuario



Guardar Servicio:

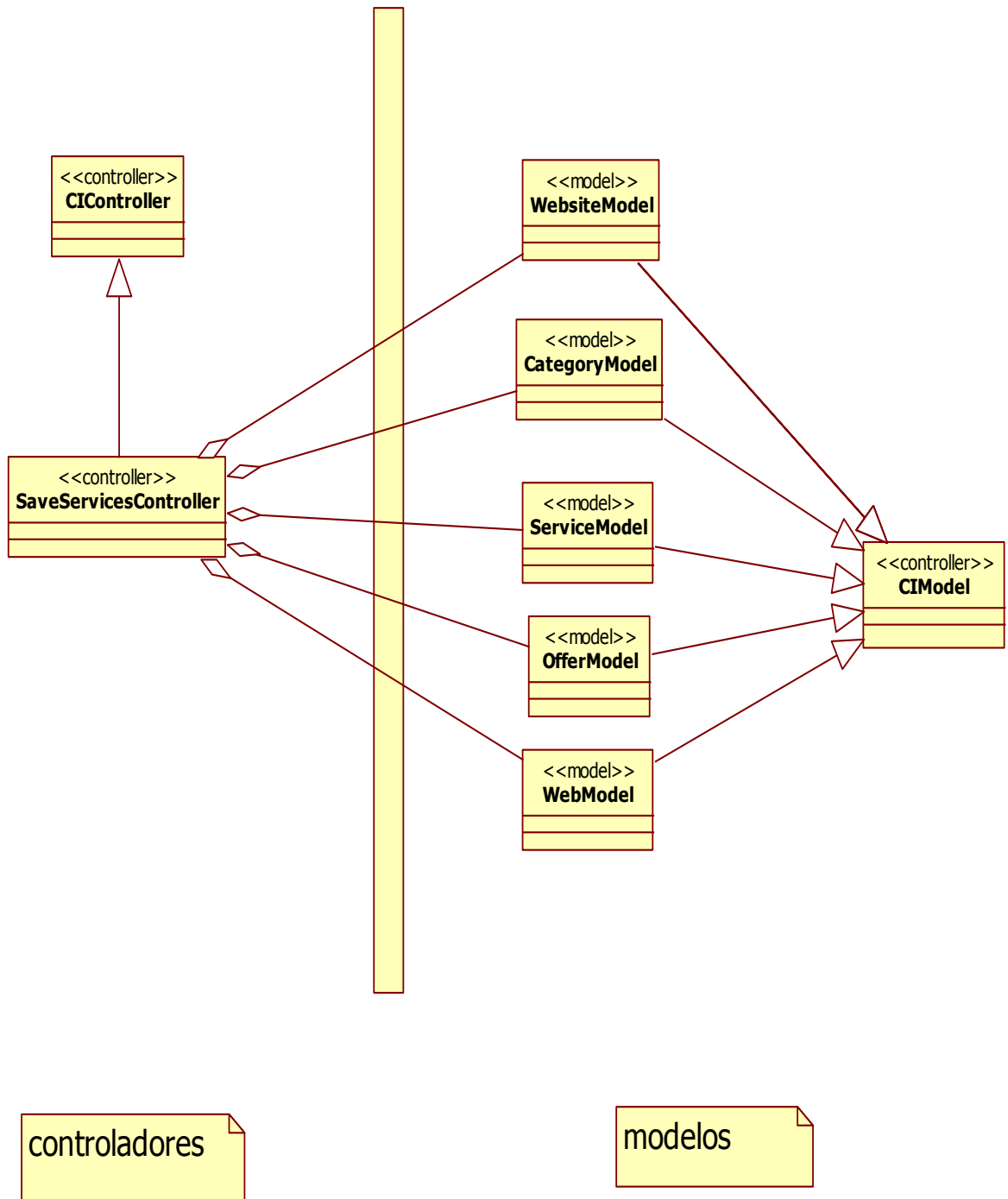


Figura 38: Diagrama de clases guardar servicio

Referencias:

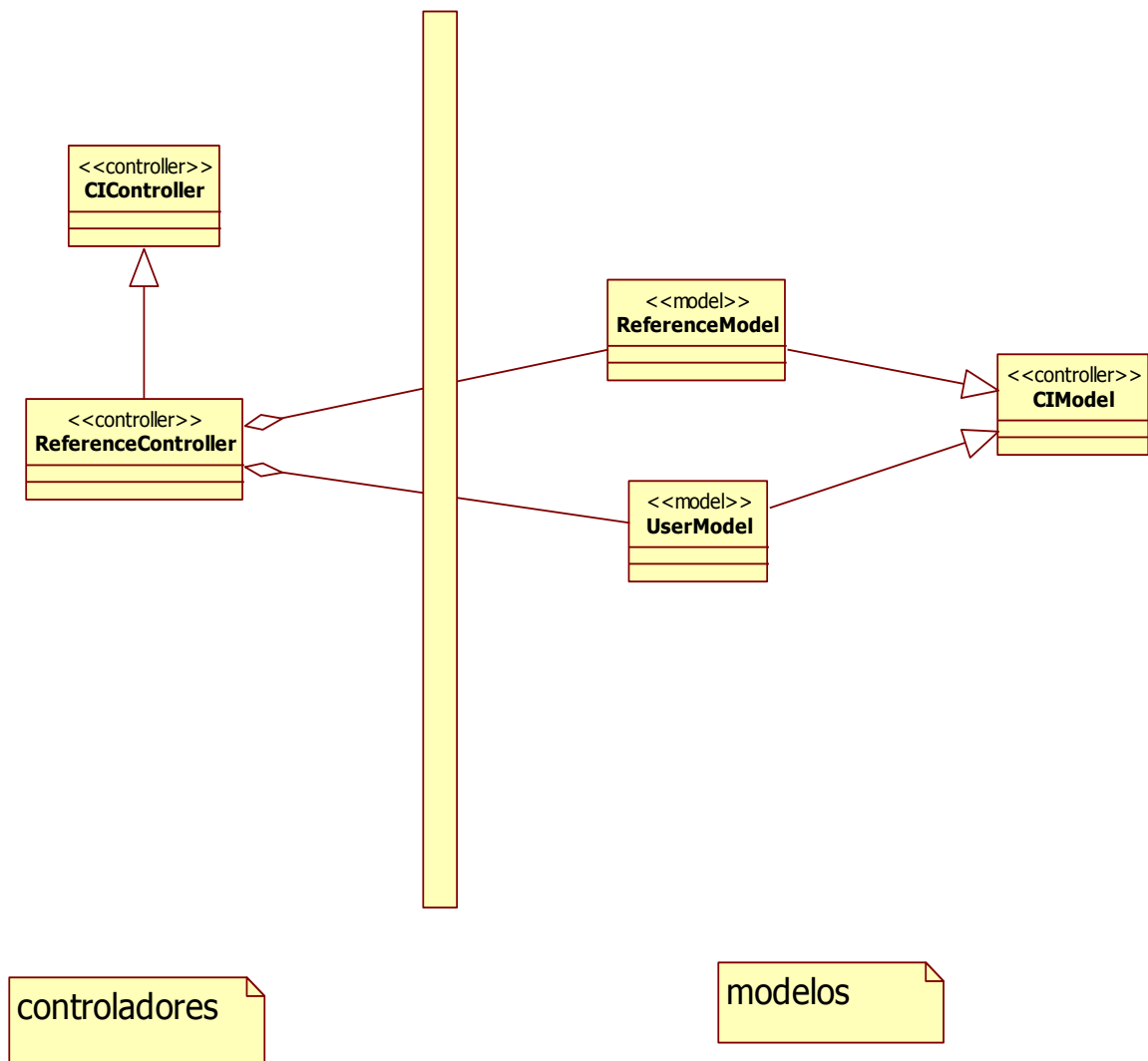


Figura 39: Diagrama de clases referencias



Save info:

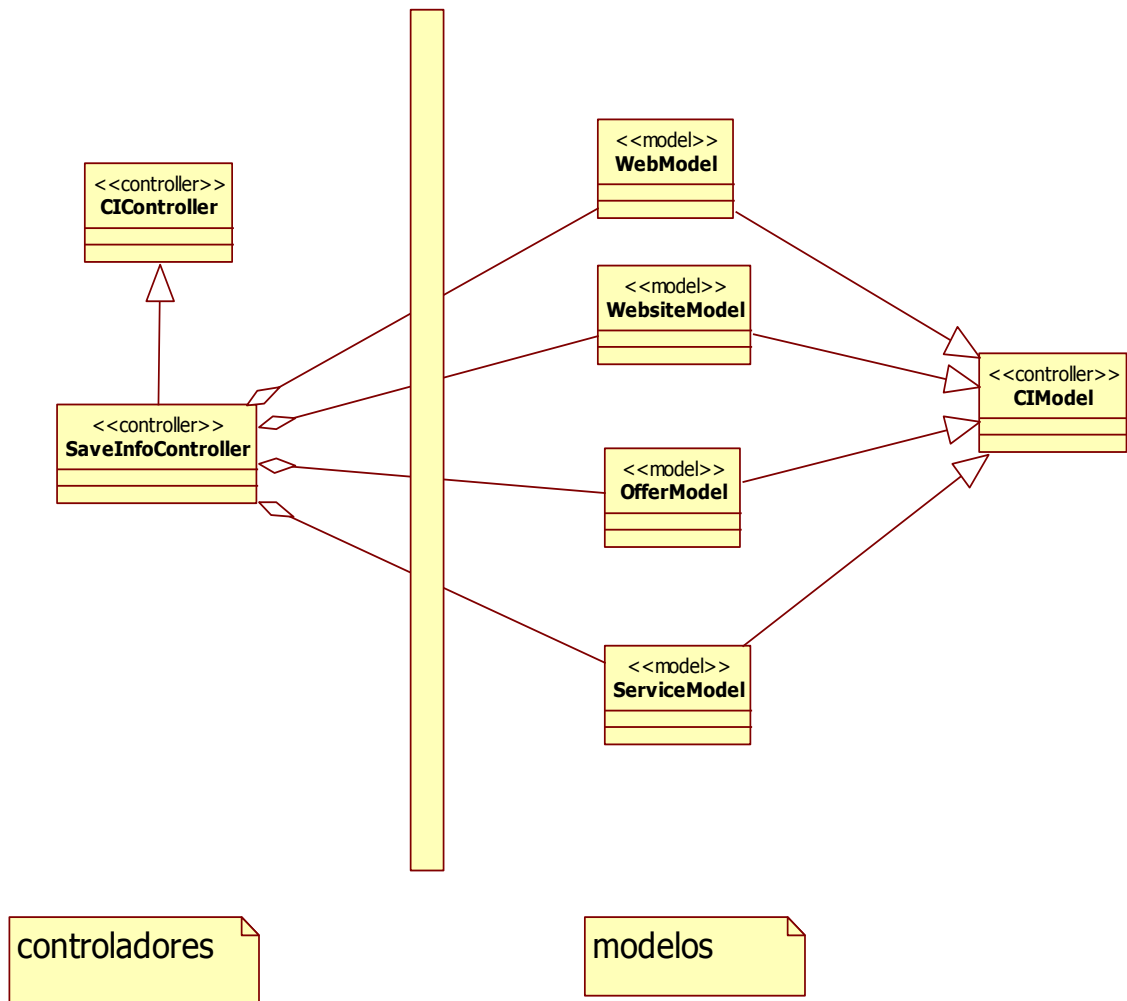


Figura 40: Diagrama de clases saveinfo

## Login

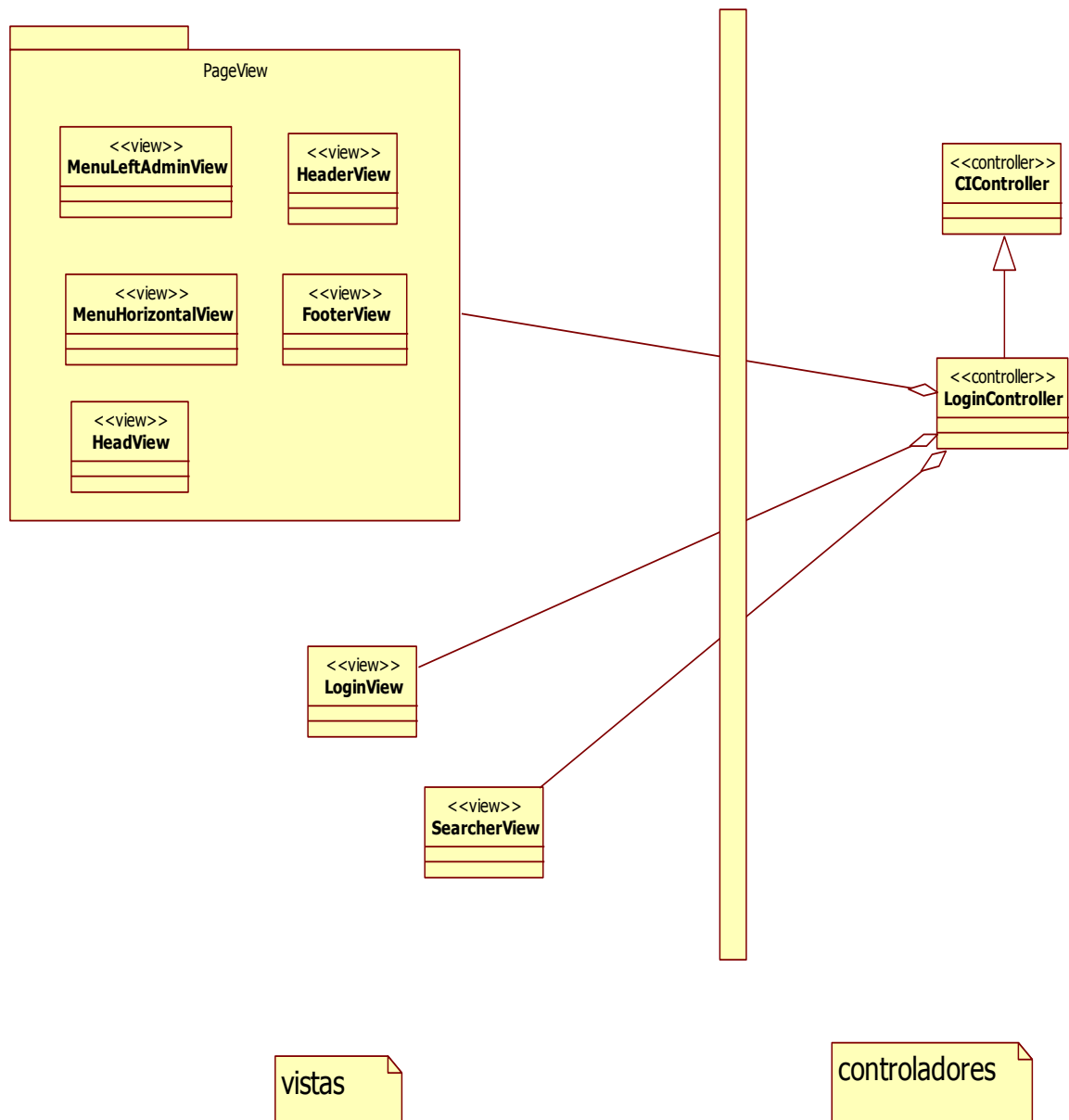


Figura 41: Diagrama de clases login

## Login2

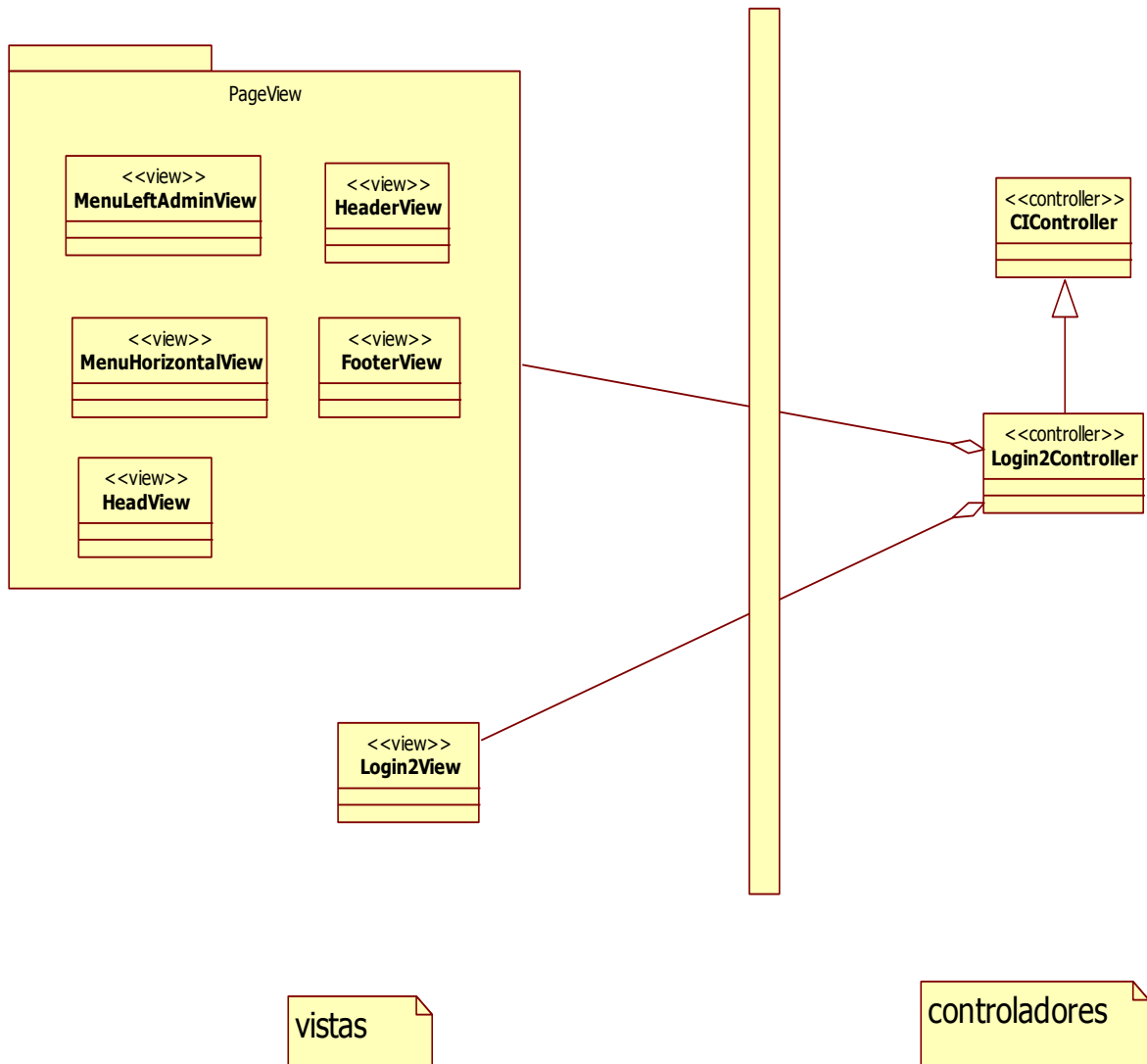


Figura 42: Diagrama de clases login2

### 6.3.6 Diagramas de secuencia

Los diagramas de secuencia están elaborados en lenguaje UML y representan de forma esquemática las acciones que debe ir realizando un rol en el sistema a lo largo de un periodo de tiempo para completar una acción. La elaboración de estos diagramas se apoya en el desarrollo previo de los diagramas de clases y los casos de uso que son elementos típicos en la ingeniería del software.

#### Crear usuario:

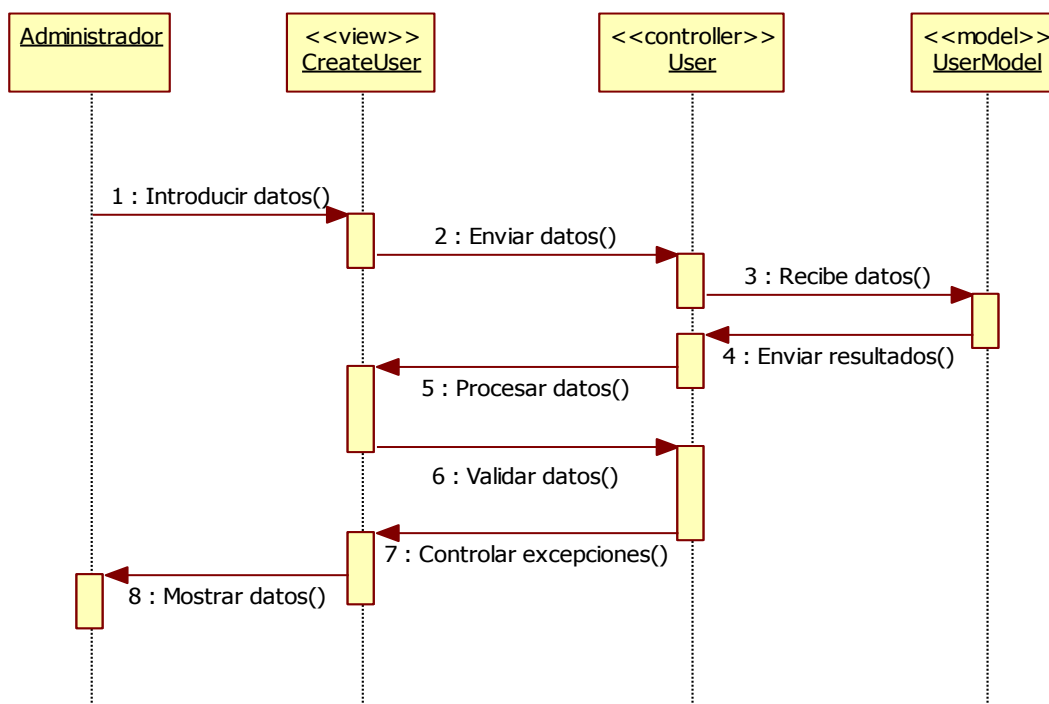


Figura 43: Diagrama de secuencia crear usuario

El actor principal de este diagrama es un usuario del sistema, el usuario ya está registrado. Este usuario rellena un formulario para la creación de un nuevo usuario en el sistema, los datos son recogidos por el controlador y redirigidos si están correctos al modelo del sistema, que guardará la información en la base de datos. Al crear el usuario de manera manual, podrá elegir el rol y no tendrá que ser activado por email (Véase Figura 43).

### Crear Servicio:

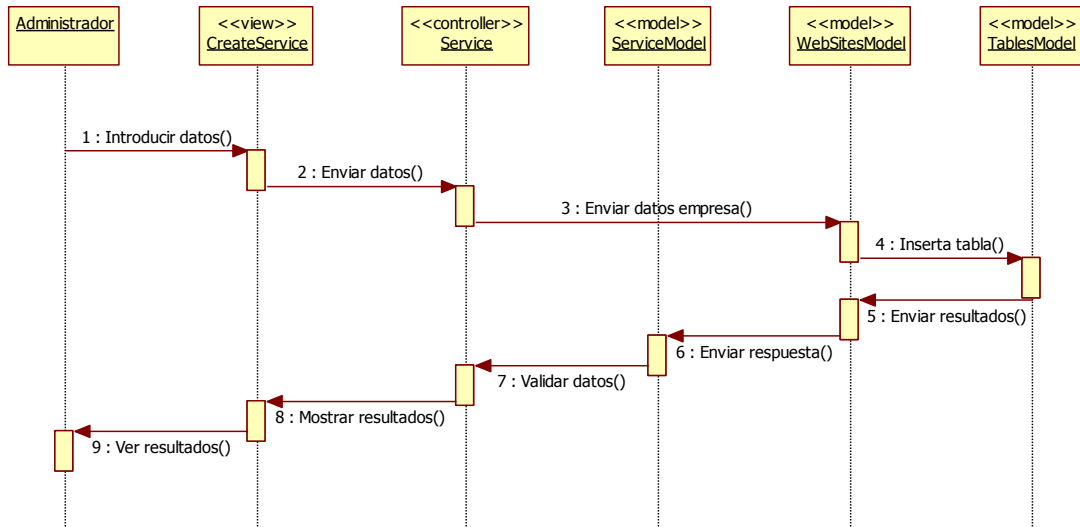


Figura 44: Diagrama de secuencia crear servicio

El actor principal de este diagrama es un usuario del sistema, el usuario ya está registrado. El usuario rellena el formulario de creación de un nuevo servicio. A continuación el controlador recoge los datos y los envía al modelo de servicio para guardar la información en la base de datos. Mediante el controlador del website se creará paralelamente una tabla dinámica en la base de datos de estadísticas, para guardar la información relevante de uso de ese servicio (Véase Figura 44).

### Crear Reserva:

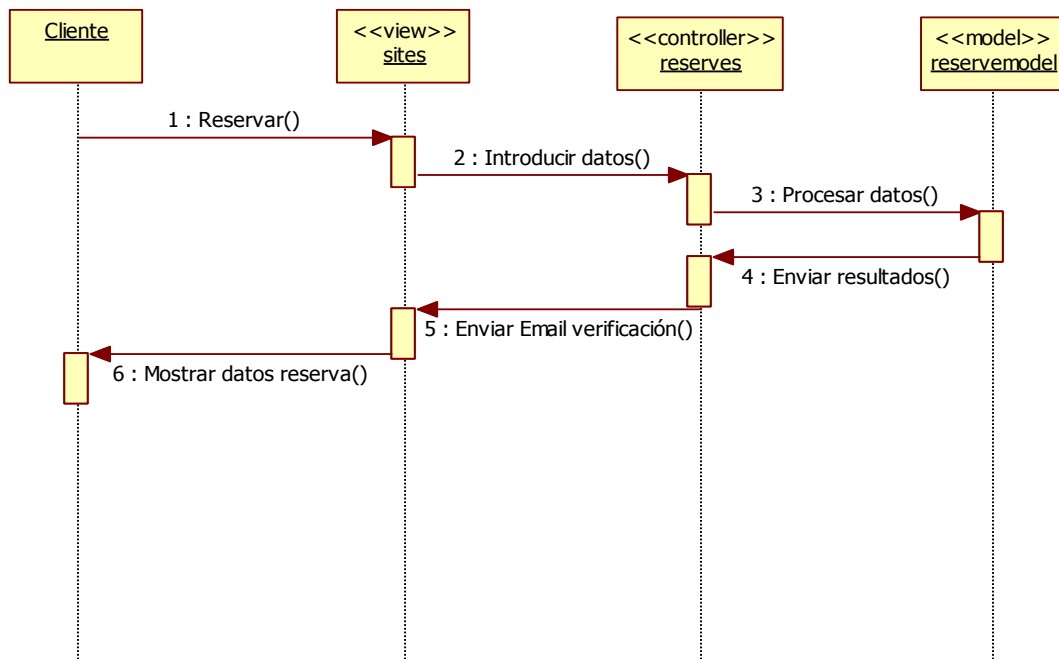


Figura 45: Diagrama de secuencia crear reserva

El actor principal de este diagrama es un usuario del sistema, el usuario ya está registrado. El usuario rellena el formulario de creación de reservas, los datos son recogidos por el controlador y redirigidos si están correctos al modelo del sistema, que guardará la información en la base de datos. Al finalizar este proceso y si todo ha ido correctamente, se enviará un correo de información de la reserva al usuario (Véase Figura 45).

**Insertar Comentario:**

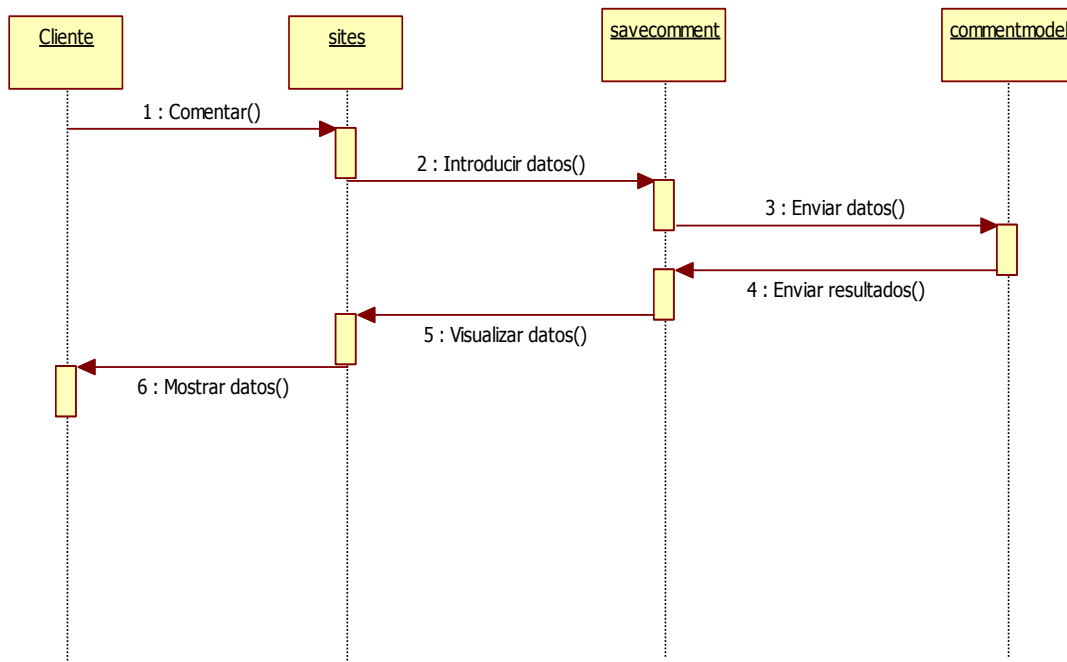


Figura 46: Diagrama de secuencia insertar comentario

El actor principal de este diagrama es un usuario del sistema, el usuario ya está registrado. El usuario rellena un formulario, los datos son recogidos por el controlador y redirigidos si están correctos al modelo del sistema, que guardará la información en la base de datos. Al finalizar este proceso se muestra en la vista el comentario insertado (Véase Figura 46).

### Comprar producto:

El actor principal de este diagrama es un usuario del sistema, el usuario ya está registrado. El usuario activa la opción comprar producto los datos son recogidos por el controlador y redirigidos a la web de sandbox de Paypal, el usuario inserta los datos de su cuenta y si están correctos se realiza la compra en Paypal. El controlador de Paypal redirige los datos al controlador y se inserta los datos en la base de datos (Véase Figura 47).

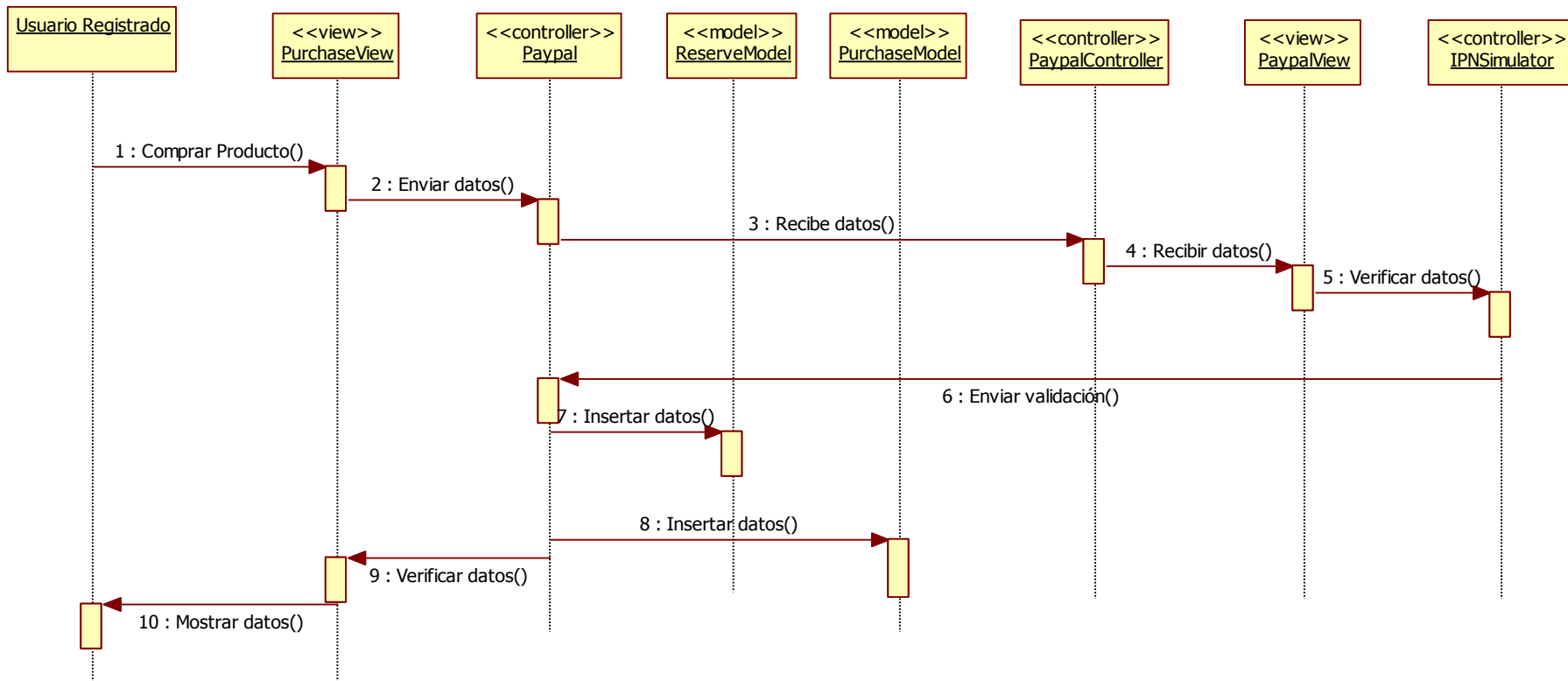


Figura 47: Diagrama de secuencia comprar producto





## 7 Conclusiones

### 7.1 Uso del Framework Codeigniter

En el inicio de un proyecto de estas características debemos decidir que estrategia de programación se va a usar para conseguir el objetivo fijado. En este caso la estrategia estaba clara al tratarse de un proyecto web, se optó por desarrollar un proyecto con patrones modelo-vista-controlador.

Aunque existen multitud de Frameworks que usan este patrón, se decidió elegir Codeigniter debido a la facilidad de uso y a sus *helpers* que proporcionaba la plataforma. Este framework tiene una curva de aprendizaje bastante alta pero, una vez que se domina el redimiento que se obtiene es considerable. Así, una vez que se va conociendo cómo es el flujo de ejecución y el manejo de todos sus componentes, es posible desarrollar aplicaciones en muy poco tiempo haciendo uso de toda la potencia de su arquitectura y librerías. Gracias a todos los mencionados componentes ha sido posible tener elementos tan importantes como un sistema de verificación de usuarios, validación de formularios, interacción coherente y mapeada con la base de datos, paginación, sencillo manejo con respuestas JSON, control de peticiones y respuestas AJAX, etc.

Sin duda, y no menos importante, el trabajar con el patrón modelo-vista-controlador permite abstraerse e implementar componentes reutilizables. Otro punto importante a tener en cuenta es que Codeigniter trabaja cien por cien con objetos, lo cual me ha permitido continuar el rodaje y aprendizaje continuo con esta filosofía, que es utilizada de forma profesional. Por todo ello el uso de un framework y en concreto Codeigniter ha sido muy fructífero y desde luego muy recomendable para desarrollar cualquier aplicación web.

### 7.2 Uso de Librería Grocery CRUD

En un proyecto en el que el uso de operaciones C.R.U.D. (Create, Read, Update, Delete) es fundamental, era inevitable tener que usar algún tipo de herramienta para afianzar los posibles errores asociados a la entrada de información del sistema y la compatibilidad con los distintos navegadores. Esta librería, que es cien por cien compatible con el Framework Codeigniter, fue la solución. Así, no solo permite un uso adecuado de la información recibida sino que, además, esta automatizada y estandarizada, permitiendo de manera fácil e intuitiva la creación de sistema de gestión de contenidos o CMS (Create Management System).

### 7.3 Uso de componentes, tecnologías y estrategias.

En este apartado de las conclusiones me gustaría remarcar la importancia de la reutilización e integración de código. Gracias a librerías y componentes como JQuery, JavaScript, JRating, Bootstrap, CSS3, Spinner y otras ha sido posible mostrar un buen aspecto y funcionalidad. Resulta imprescindible y recomendable en este tipo de proyectos utilizar los citados elementos que colaboran de forma activa al desarrollo de las aplicaciones.

Bitbucket, el control de versiones privado, para el desarrollo de la aplicación web ha supuesto un componente esencial para el avance del proyecto. Con este proyecto he conseguido comprender mejor y coger soltura en la utilización de estos elementos que ahora me resultan imprescindibles para desarrollar cualquier aplicación web.

## 7.4 Resultado final del proyecto

La conclusión final de este proyecto es que ha finalizado cumpliendo los objetivos y metas que se propusieron teniendo como resultado:

Un portal web con un sistema de información que permite introducir todos los datos necesarios para la descripción de ofertas en turismo rural con especial atención en el alojamiento.

Un sistema de gestión de contenidos que autogenera páginas web profesionales para las empresas del sistema, con posibilidad de edición de páginas.

Un sistema de gestión de reservas y venta de productos.

Un sistema de clasificación de productos y páginas web del sistema, mediante votación de usuarios registrados.

Esto permite que la salida de este proyecto concluya con un portal web dónde se localizan empresas de turismo rural, que se publicitan y venden sus productos.

## 8 Trabajo Futuro:

Este proyecto a pesar de su complejidad puede ofrecer todavía multitud de mejoras. Su expansión se puede realizar por una gran cantidad de frentes, algunos de ellos se comentan a continuación:

### 1.-Ampliación de las plantillas del editor web:

Actualmente el editor dispone de una sola plantilla configurable muy básica, se podrían generar multitud de plantillas configurables y añadirlas en el editor para que el usuario pudiera seleccionar la más idónea para su actividad comercial. Las plantillas podrían tener distintos diseños, imágenes, combinaciones de colores e iconos diferentes.

### 2.-Interfaz multi-idioma:

En los tiempos que corren tener una página en múltiples idiomas es algo básico. Al ser una actividad turística los clientes potenciales pueden ser de otros países. Por ello, sería recomendable disponer de al menos uno o dos idiomas más. El Framework Codeigniter dispone de una clase "Language Class", ésta permite abstraer la lógica de la aplicación del idioma de la interfaz mediante ficheros donde se contiene la información del idioma con las etiquetas, pudiendo haber tantos idiomas como ficheros disponibles.

### **3.-Compras con tarjeta de crédito:**

En la actualidad no solo se paga con Paypal, los clientes deberían disponer de otras formas de pago. Implementar una pasarela de pago con seguridad cifrada será uno de los siguientes pasos a realizar.

Paypal o Paymil ofertan servicios de pagos dónde implementan toda la pasarela de pago.

### **4.- Descuentos:**

Las empresas deberían poder ofertar descuentos de sus productos, el próximo paso debería ser la creación de un gestor de descuentos. Este gestor permitiría la creación de descuentos temporales asociados a productos de la empresa.

### **5.- Promociones:**

Las empresas deberían disponer de un gestor de promociones y la posibilidad de hacer regalos sobre los productos ofertados, para fidelizar a los clientes.

### **6.-Mejorar las búsquedas:**

Será necesario crear un sistema de calificaciones para ordenar de manera equitativa y justa según la valoración de los usuarios.

### **7.- Interfaz Web y Móvil:**

Cambiar periódicamente el diseño de la web y el móvil, para dar una imagen de fluidez y dinamismo y no estancarse en el pasado.

## 9 Anexos

### 9.1 Anexo I: Recursos utilizados

#### 9.1.1 Recursos software

##### SO - Ubuntu Linux.

Ubuntu es un sistema operativo basado en Debian y que se distribuye como software libre y gratuito, el cual incluye su propio entorno de escritorio denominado Unity. Está orientado al usuario novel y promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia de usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto.

Cada seis meses se publica una nueva versión de Ubuntu. Esta recibe soporte por parte de Canonical durante nueve meses por medio de actualizaciones de seguridad, parches para bugs críticos y actualizaciones menores de programas. Las versiones LTS (Long Term Support), que se liberan cada dos años, reciben soporte durante cinco años en los sistemas de escritorio y de servidor (Véase Imagen 14).



Imagen 14: S.O. Ubuntu Linux

Se ha utilizado mayoritariamente para hacer pruebas en el servidor.

## SO – Mac OSX.

Es el nombre del sistema operativo creado por Apple para su línea de computadoras Macintosh. Es conocido por haber sido uno de los primeros sistemas dirigidos al gran público en contar con una interfaz gráfica compuesta por la interacción del mouse con ventanas, icono y menús. Desde la existencia del sistema operativo en los primeros años dentro de la línea Macintosh ha centrado sus esfuerzos en favor de que la máquina resultara más agradable al usuario, diferenciándolo de otros sistemas contemporáneos, como MS-DOS, que eran un desafío técnico. El equipo de desarrollo del Mac OS original incluía a Bill Atkinson, Jef Raskin y Andy Hertzfeld.

Esta fue la base del Mac OS *clásico*, desarrollado íntegramente por Apple, cuya primera versión vio la luz en 1985. Su desarrollo se extendería hasta la versión 9 del sistema, lanzada en 1999. A partir de la versión 10 (Mac OS X), el sistema cambió su arquitectura totalmente y pasó a basarse en Unix, sin embargo su interfaz gráfica mantiene muchos elementos de las versiones anteriores.

Hay una gran variedad de versiones sobre cómo fue desarrollado el Mac OS original y dónde se originaron las ideas subyacentes. Pese a esto documentos históricos prueban la existencia entre el proyecto Macintosh y el proyecto Alto de Xerox PARC. Las contribuciones iniciales del Sketchpad de Ivan Sutherland y el On-Line System de Doug Engelbart también fueron significativas (Véase Imagen 15).



Imagen 15: S.O. MaCosX

## PhpMyAdmin

PhpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web utilizando Internet. Actualmente puede crear y eliminar Bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 62 idiomas. Se encuentra



Imagen 16: PhpMyAdmin

disponible bajo la licencia GPL. Este proyecto se encuentra vigente desde el año 1998, siendo el mejor evaluado en la comunidad de descargas de SourceForge.net.

Esta herramienta ha sido utilizada para gestionar la base datos. Se ha usado principalmente en el servidor de pruebas para la inserción y consulta de datos en nuestra aplicación web (Véase Imagen 16).

## Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. Apache presenta entre otras características altamente configurables, bases de datos



Imagen 17: Apache

de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en

su configuración. Sus características principales y por el que es muy extensamente utilizado es por ser un servidor muy modular, de código abierto, multi-plataforma y extensible.

Ha sido el servidor utilizado para desarrollar la aplicación en preproducción (Véase Imagen 17).

## GitHub

Es una plataforma de desarrollo colaborativo de software que sirve para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome). El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.



Imagen 18: Github

Esta herramienta se ha utilizado para llevar el control de las versiones del proyecto y facilitar el desarrollo colaborativo de la aplicación (Véase Imagen 18).

## Bitbucket

Bitbucket es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de revisiones Mercurial y Git. Bitbucket ofrece planes comerciales y gratuitos. Se ofrece cuentas gratuitas con un número



Imagen 19: Bitbucket

ilimitado de repositorios privados (que puede tener hasta cinco usuarios en el caso de cuentas

gratuitas). Este sistema en línea permite gestionar y compartir multitud de repositorios públicos y privados en la nube. Desde septiembre de 2010, los repositorios privados no se muestran en las páginas de perfil si un usuario sólo tiene depósitos privados, el sitio web dará el mensaje “Este usuario no tiene repositorios”. El servicio está escrito en Python. Es similar a GitHub, que utiliza Git.

Esta herramienta se ha utilizado junto con Git para llevar el control de las versiones del proyecto (Véase Imagen 19).

## Microsoft Visio Premium 2010

Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Visio comenzó a formar parte de los productos de Microsoft cuando fue adquirida la compañía Visio en el año 2000. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más, que permiten iniciar al usuario en los



Imagen 20: Microsoft Visio

lenguajes de programación.

Esta herramienta se ha utilizado para elaborar diagramas de casos de uso y diagramas de bases de datos (Véase Imagen 20).

## Notepad++

Notepad++ es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. De soporte nativo a Microsoft Windows. Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores. Se distribuye bajo los términos de la Licencia Pública General de GNU.



Imagen 21: Notepad++

Se ha utilizado esta herramienta para la edición de diferentes tipos de código, javascript, jquery, html, php, etc (Véase Imagen 21).

## Sublime text 2

Sublime Text es un editor de texto y editor de código fuente creado en Python desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado *Vintage mode*. Se distribuye de forma gratuita, sin embargo no es software libre o de código abierto, se puede obtener

una licencia para su uso ilimitado, pero el no disponer de esta no genera ninguna limitación más allá de una alerta cada cierto tiempo. Se ha utilizado esta herramienta para edición de código php. (Véase Imagen 22)



Imagen 22: Sublime Text 2

## HTML5

Representa toda la estructura en la que está fundamentada la web hoy en día. Sus siglas significan HyperText Markup Language y hace referencia al lenguaje de marcado para la elaboración de páginas web. Este lenguaje se escribe en forma de etiquetas y se encarga de definir y estructurar cada uno de los elementos que aparecen en una página web. Fue publicado por primera vez en Internet por Tim Berners-Lee en 1991. Actualmente la W3C se encarga de desarrollar estándares abiertos para asegurar el correcto crecimiento de la web. La última versión de HTML, HTML 5, ofrece nuevas etiquetas para incrustar videos y gráficos de manera directa en la propia página entre otras muchas nuevas características. Se ha utilizado para el desarrollo web y móvil de la aplicación (Véase Imagen 23).



Imagen 23: Html5

## PHP

Es un lenguaje de programación de uso general que trabaja del lado del servidor y fue diseñado para el desarrollo web de contenido dinámico. Es uno de los primeros lenguajes que permitió incorporar código directamente en el HTML en vez de un fichero externo que procesara datos. Forma parte de la familia de lenguajes interpretados y el servidor web es quien se encarga de procesarlo y generar la página web



Imagen 24: Php

resultante. Es posible utilizarlo tanto en sistemas

UNIX como en Microsoft Windows apoyándose en servidores como Apache, CGI e ISAPI. Es un lenguaje que derivó en su nacimiento (1995) de los lenguajes más comunes de programación estructurada como C y Perl. Actualmente es uno de los lenguajes más utilizados en el desarrollo en la web y constituye una alternativa a otros lenguajes como Microsoft ASP y las páginas JSP de Java. PHP tiene incorporados unos puentes que permiten realizar conexiones con diferentes tipos de bases de datos como MySQL, PostgreSQL, Oracle, ODBC, Microsoft SQL Server, Firebird y SQLite (Véase Imagen 24).



## AJAX

AJAX no es en sí un lenguaje de programación sino una técnica de desarrollo web para crear aplicaciones interactivas. Como su propio nombre indica, Asynchronous JavaScript And XML, utiliza el lenguaje de programación JavaScript y la estructuración de datos XML en el lado del cliente para comunicarse de forma asíncrona con el servidor. Esta tecnología permite modificar elementos de



Imagen 25: Ajax

la página web sin necesidad de recargarla o ir a otra página. De esta forma es posible incrustar datos recuperados de la B.B.D.D. en un elemento concreto de la página. Su principal ventaja es que evita la necesidad de cargar todo el contenido cuando sólo es necesario refrescar o cargar algunos elementos de la página. Su funcionamiento está basado en peticiones realizadas mediante XMLHttpRequest y procesado y modificado de elementos en la parte del cliente con JavaScript y DOM (Véase Imagen 25).

## CSS

Las hojas de estilo en cascada son un lenguaje que permiten describir estilos de un documento (página web) a través de marcas. Esta información puede estar incrustada en el propio HTML o por el contrario separada en ficheros .css. Su sintaxis está basado en uno o más selectores y un bloque de estilos donde se definen los valores de las propiedades del documento. Los selectores pueden ser únicos cuando se trata del id de un elemento o múltiple cuando se habla de una clase de elementos (Véase Imagen 26).



Imagen 26: Css3

## SQL

El SQL es un lenguaje de consulta estructurado de acceso a bases de datos relaciones que permite especificar diversos tipos de operaciones entre ellas. Una de sus características es el manejo del álgebra y el cálculo relaciones que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información en bases de datos, así como hacer cambios en ella. En SQL una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros (Véase Imagen 27).



Imagen 27: MySql

## JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor. JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y

convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM) (Véase Imagen 28).



Imagen 28: JavaScript

## JQuery

Es una biblioteca de código JavaScript que permite de manera sencilla interactuar con los documentos HTML, manipular su DOM e integrar AJAX. Es un software libre y de código abierto bajo la licencia del MIT y GNU v2. Se utiliza como una potente herramienta que permite a los desarrolladores crear efectos atractivos y dinámicos en las páginas web de una manera rápida y limpia. Cuenta con las siguientes



Imagen 29: JQuery

características (Véase Imagen 29):

- Selección de elementos DOM, Eventos.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.

## XAMPP

Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor webApache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

El programa está liberado bajo la licencia GNU y actúa como

un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas.

Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado. Puede encontrarse tanto en versión completa, así como en una versión más ligera que es portátil (Véase Imagen 30).



Imagen 30: XAMPP

### 9.1.2 Recursos hardware

**Para la realización de la aplicación web se han necesitado los siguientes equipos hardware:**

**Equipo informático:** Dónde se realizó la mayor parte del desarrollo.

Procesador: Intel® Core™ i5 1,8Ghz

Memoria RAM: 4,00GB

Disco duro: 250GB

Tarjeta gráfica: Intel HD graphics 4000 1024 MB

Pantalla panorámica de 13.3''

Teclado y ratón

**Servidor de pruebas:** Dónde se realizaron pruebas de mantenimiento y rendimiento.

Procesador: Intel core2 DUO CPU E8400 3GHZ

Memoria RAM: 4,00GB

Disco duro: 250GB

Tarjeta gráfica: Ati Radeon HD 2400 XT

Pantalla: 15''

## 9.2 Anexo II: Especificación de casos de uso

### Usuario:

A continuación mostramos los casos de uso con más detalle para el usuario no registrado (Véase Tabla 5, 6 y 7).

Nombre:	Registrarse
Descripción:	Con este caso de uso una persona puede convertirse en usuario de la aplicación. Simplemente tiene que rellenar un formulario estándar de registro dando una serie de datos para hacerlo. La aplicación le enviará un correo de bienvenida y le dará acceso al sistema
Precondición:	Usuario autenticado.
Parámetros:	Nombre Contraseña E-Mail
Flujo de Ejecución:	<ol style="list-style-type: none"> <li>1. El usuario accede al formulario de registrar usuario</li> <li>2. El usuario proporciona sus datos personales: <ul style="list-style-type: none"> <li>• El usuario introduce su correo electrónico</li> <li>• El usuario introduce una contraseña</li> </ul> </li> <li>3. El usuario hace clic en el botón de “Registrar” del formulario</li> <li>4. El sistema realiza una serie de verificaciones: <ul style="list-style-type: none"> <li>• El sistema comprueba que el nombre utilizado no está en uso</li> <li>• El sistema comprueba que la dirección de correo es válida</li> </ul> </li> <li>5. Si todas las comprobaciones tienen éxito, el usuario se añade a la base de datos del sistema</li> <li>6. El sistema envía un correo a la dirección indicada por el usuario.</li> <li>7. El usuario recién creado se autentifica en el sistema</li> <li>8. El sistema muestra la pantalla inicial al usuario</li> </ol>
Caminos Alternativos:	<p>(4)El usuario no existe.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5)Corrupción de datos en la base de datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
Postcondición:	Usuario registrado en el sistema.

Tabla 5: Descripción caso de uso Registrarse

<b>Nombre:</b>	<b>Visualizar Web</b>
<b>Descripción:</b>	Una vez el usuario ha realizado la búsqueda se muestra una lista de enlaces web, el usuario puede seleccionar un enlace y acceder.
<b>Precondición:</b>	Realizada búsqueda.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- Al usuario se le muestra una lista de enlaces</li> <li>2.-El usuario selecciona el enlace.</li> <li>3.- El sistema muestra la página web.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(3)La página web no existe.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Ninguna.

Tabla 6: Descripción caso de uso Visualizar Web

<b>Nombre:</b>	<b>Visualizar Producto</b>
<b>Descripción:</b>	Una vez el usuario este visualizando la web, puede seleccionar los productos que ofrece la empresa. Al seleccionar el producto se muestra toda la información asociada, foto, galería de fotos y descripción.
<b>Precondición:</b>	Ninguno.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- Al usuario se le muestra una lista de productos</li> <li>2.-El usuario selecciona el producto</li> <li>3.- El sistema muestra la una página con la información del producto.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(3)La página web no existe.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(3)El producto no existe en la base de datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Ninguna.

Tabla 7: Descripción caso de uso Visualizar Producto

### Usuario Registrado:

A continuación mostramos los casos de uso con más detalle para el usuario registrado (Véase Tablas 8 y 9).

<b>Nombre:</b>	<b>Iniciar Sesión</b>
<b>Descripción:</b>	Un usuario registrado de la aplicación para entrar en la misma debe autenticarse en el sistema para obtener sus datos y acceder a la pantalla inicial. Si no está autenticado y desea acceder a un recurso propio de la aplicación, esta deberá cancelar la petición del usuario y llevarlo hasta la pantalla de identificación.
<b>Precondición:</b>	Usuario no autenticado.
<b>Parámetros:</b>	Email Contraseña
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"><li>1. El usuario accede al formulario de registrar usuario</li><li>2. El usuario proporciona sus datos personales:<ul style="list-style-type: none"><li>• El usuario introduce su nombre</li><li>• El usuario introduce la contraseña</li></ul></li><li>3. El usuario hace clic en el botón de identificarse del formulario</li><li>4. El sistema realiza una serie de verificaciones:<ul style="list-style-type: none"><li>• El sistema comprueba que el nombre existe en la base de datos</li><li>• El sistema comprueba que la contraseña es correcta</li></ul></li><li>5. Si todas las comprobaciones tienen éxito el usuario queda autenticado en el sistema</li><li>6. El sistema muestra la pantalla inicial al usuario</li></ol>
<b>Caminos Alternativos:</b>	(2)Usuario no existente. <ul style="list-style-type: none"><li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida</li></ul> (2)Contraseña incorrecta. <ul style="list-style-type: none"><li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida</li></ul>
<b>Postcondición:</b>	Sesión iniciada.

Tabla 8: Descripción caso de uso Iniciar Sesión

<b>Nombre:</b>	<b>Cerrar Sesión</b>
<b>Descripción:</b>	El usuario una vez autenticado puede cerrar la sesión en cualquier momento.
<b>Precondición:</b>	Usuario Autenticado.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"><li>1.- El usuario se encuentra dentro de la aplicación web.</li><li>2.-El usuario selecciona haciendo click en el menú desplegable situado a la derecha de la barra principal de la página junto a la imagen de avatar de usuario "Cerrar Sesión".</li><li>3.- El sistema envía al usuario a la pantalla principal.</li></ol>
<b>Caminos Alternativos:</b>	Ninguno.
<b>Postcondición:</b>	Sesión Cerrada.

Tabla 9: Descripción caso de uso Cerrar Sesión

### Usuario Cliente:

A continuación mostramos los casos de uso con más detalle para el usuario cliente (Véase Tablas 10, 11 y 12).

<b>Nombre:</b>	<b>Realizar compra</b>
<b>Descripción:</b>	El usuario una vez autenticado puede seleccionar un producto y comprarlo mediante pago con Paypal.
<b>Precondición:</b>	Usuario autenticado y visualizando producto.
<b>Parámetros:</b>	Datos bancarios.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"><li>1.- El usuario visualiza el producto.</li><li>2.-El usuario selecciona haciendo click en botón “comprar ahora”.</li><li>3.-Se accede a la web de Paypal.</li><li>4.- Introducir datos de Paypal y seguir los pasos del sistema.</li><li>5.- La compra se ha realizado.</li><li>6.- La información de la compra se almacena en la base de datos.</li></ol>
<b>Caminos Alternativos:</b>	<p>(4)Errores de Paypal.</p> <ul style="list-style-type: none"><li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li></ul> <p>(4)Corrupción de datos en la base de datos.</p> <ul style="list-style-type: none"><li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li></ul> <p>(6)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"><li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li></ul>
<b>Postcondición:</b>	Compra realizada

Tabla 10: Descripción caso de uso Realizar Compra

<b>Nombre:</b>	<b>Realizar reserva</b>
<b>Descripción:</b>	El usuario una vez autenticado puede seleccionar un producto y hacer una reserva en el sistema.
<b>Precondición:</b>	Usuario autenticado y visualizando producto.
<b>Parámetros:</b>	Nombre Apellidos Dirección País Provincia/Municipio Email Código postal Teléfono Fecha inicio Fecha fin Nº Adultos Nº Niños Comentarios
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario visualiza el producto.</li> <li>2.-El usuario selecciona haciendo click en botón “Reservar”.</li> <li>3.-El usuario accede al formulario de reserva.</li> <li>4.-El usuario introduce los datos necesarios para completar la solicitud.</li> <li>5.-Se calculan y muestran los datos de la reserva.</li> <li>6.-El usuario hace click en el botón “Reservar ahora”.</li> <li>7.- Se envía un email con la información de la reserva.</li> <li>8.- La reserva se ha realizado.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(4)Errores al introducir datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(4)No se dispone de tarifas.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(6)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Reserva realizada

Tabla 11: Descripción caso de uso Realizar Reserva



<b>Nombre:</b>	<b>Insertar Comentario</b>
<b>Descripción:</b>	Una vez el usuario este visualizando el producto puede realizar comentarios sobre sus valoraciones.
<b>Precondición:</b>	Usuario autenticado y Visualizando el producto.
<b>Parámetros:</b>	Comentario
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- Al usuario se le muestra el producto.</li> <li>2.-El usuario introduce el comentario.</li> <li>3.- El sistema inserta el comentario en la base de datos.</li> <li>4.- El sistema muestra el comentario insertado.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(3)Error en la base de datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(4)No existen comentarios</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Comentario guardado.

Tabla 12: Descripción caso de uso Insertar Comentario

### **Usuario Gestor:**

A continuación mostramos los casos de uso con más detalle para el usuario gestor (Véase Tablas desde la 13 hasta la 25).

<b>Nombre:</b>	<b>Visualizar Comentarios</b>
<b>Descripción:</b>	El usuario gestor puede buscar comentarios de productos para decidir si se muestran o no se muestran en el producto seleccionado.
<b>Precondición:</b>	Usuario autenticado .
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"><li>1.- Al usuario se le muestra una lista de productos.</li><li>2.-El usuario selecciona un producto.</li><li>3.- El sistema muestra una tabla con los comentarios del producto seleccionado.</li><li>4.- El usuario selecciona un comentario.</li><li>5.- El usuario visualiza el comentario.</li></ol>
<b>Caminos Alternativos:</b>	<p>(3)Error en la base de datos</p> <ul style="list-style-type: none"><li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li></ul> <p>(3)No existen comentarios</p> <ul style="list-style-type: none"><li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li></ul>
<b>Postcondición:</b>	Ninguno.

Tabla 13: Descripción caso de uso Visualizar Comentarios

<b>Nombre:</b>	<b>Visualizar Registro</b>
<b>Descripción:</b>	El usuario gestor una vez autenticado puede visualizar todas las actividades que se realizan los gestores en la aplicación web.
<b>Precondición:</b>	Usuario gestor autenticado.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<p>1.- El usuario accede a la aplicación web.</p> <p>2.-El usuario selecciona haciendo click en el menú principal "Registro de Eventos".</p> <p>3.- El sistema muestra mediante una tabla todos los datos relativos a los eventos.</p>
<b>Caminos Alternativos:</b>	<p>(3) Los Datos no existen en la base de datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(3) No se puede conectar a la base de datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Ninguno.

Tabla 14: Descripción caso de uso Visualizar Registro

<b>Nombre:</b>	<b>Crear Tarifa</b>
<b>Descripción:</b>	El usuario gestor una vez autenticado puede crear una o distintas tarifas para un producto.
<b>Precondición:</b>	Usuario gestor autenticado.
<b>Parámetros:</b>	Fecha inicio Fecha fin Precio Adulto Precio Niño Impuestos Tasas
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Tarifas".</li> <li>3.- El sistema muestra mediante una tabla todos los productos.</li> <li>4.- El usuario selecciona un producto.</li> <li>5.- El usuario hace click en el icono nueva tarifa.</li> <li>6.- El usuario introduce los datos.</li> <li>7.- El sistema guarda la nueva tarifa.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(3)No existen productos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(6) Error de base de datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Tarifa Guardada.

Tabla 15: Descripción caso de uso Crear Tarifa

<b>Nombre:</b>	<b>Editar Tarifa</b>
<b>Descripción:</b>	El usuario gestor una vez autenticado puede editar una tarifa de producto.
<b>Precondición:</b>	Usuario gestor autenticado.
<b>Parámetros:</b>	Fecha inicio Fecha fin Precio Adulto Precio Niño Impuestos Tasas
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal “Gestionar Tarifas”.</li> <li>3.- El sistema muestra mediante una tabla todos los productos.</li> <li>4.- El usuario selecciona un producto.</li> <li>5.- El usuario hace click en el icono editar tarifa.</li> <li>6.- El usuario introduce los datos.</li> <li>7.- El sistema modifica la tarifa seleccionada.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(3)No existen productos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(6) Error de base de datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Tarifa modificada.

Tabla 16: Descripción caso de uso Editar Tarifa

<b>Nombre:</b>	<b>Eliminar Tarifa</b>
<b>Descripción:</b>	El usuario gestor una vez autenticado puede eliminar una tarifa de un producto.
<b>Precondición:</b>	Usuario gestor autenticado.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Tarifas".</li> <li>3.- El sistema muestra mediante una tabla todos los productos.</li> <li>4.- El usuario selecciona un producto.</li> <li>5.- El usuario hace click en el icono eliminar tarifa.</li> <li>6.- El sistema muestra un mensaje de seguridad.</li> <li>7.- El usuario selecciona aceptar</li> <li>6.- El sistema eliminar la tarifa seleccionada.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(3)No existen productos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7) El usuario selecciona la opción cancelar eliminar.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(8) Error la tarifa no pudo eliminarse.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Tarifa eliminada.

Tabla 17: Descripción caso de uso Eliminar Tarifa

<b>Nombre:</b>	<b>Buscar Tarifa</b>
<b>Descripción:</b>	El usuario gestor una vez autenticado puede buscar una tarifa de un producto.
<b>Precondición:</b>	Usuario gestor autenticado.
<b>Parámetros:</b>	Producto Fecha inicio Fecha fin Precio Adulto Precio Niño Impuestos Tasas
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Tarifas".</li> <li>3.- El sistema muestra mediante una tabla todos los productos.</li> <li>4.- El usuario busca en la tabla la tarifa.</li> <li>5.- El sistema muestra el/los productos asociados a esta tarifa.</li> <li>6.- El usuario selecciona el producto.</li> <li>7.- El usuario selecciona la tarifa.</li> </ol>
<b>Camínos Alternativos:</b>	<p>(3)No existen productos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5) No existen tarifas con estos parámetros.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Ninguno.

Tabla 18: Descripción caso de uso Buscar Tarifa

<b>Nombre:</b>	<b>Crear Reserva</b>
<b>Descripción:</b>	El usuario una vez autenticado puede seleccionar un producto y hacer una reserva en el sistema de forma manual para un cliente.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Nombre Apellidos Dirección País Provincia/Municipio Email Código postal Teléfono Fecha inicio Fecha fin Nº Adultos Nº Niños Comentarios
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal “Gestionar Reservas”.</li> <li>3.- El sistema muestra mediante una tabla todos los productos.</li> <li>4.- El usuario busca el producto para realizar la reserva.</li> <li>5.- El usuario introduce los datos en el formulario de reserva.</li> <li>6.- El sistema guarda los datos de la reserva.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(3)No existen productos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5)Error al introducir los datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(6)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Reserva realizada.

Tabla 19: Descripción caso de uso Crear Reserva



<b>Nombre:</b>	<b>Eliminar Reserva</b>
<b>Descripción:</b>	El usuario una vez autenticado eliminar reservas de productos de forma manual.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal “Gestionar Reservas”.</li> <li>3.- El sistema muestra mediante una tabla todos los productos.</li> <li>4.- El usuario busca el producto para eliminar la reserva.</li> <li>5.- El usuario hace click en el icono eliminar.</li> <li>6.- El sistema muestra un mensaje de seguridad.</li> <li>7.- El usuario hace click en aceptar.</li> <li>8.- El sistema elimina la reserva.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(3)No existen productos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7)El usuario cancela la operación de eliminar.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(8)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Reserva eliminada.

Tabla 20: Descripción caso de uso Eliminar Reserva

<b>Nombre:</b>	<b>Buscar Reserva</b>
<b>Descripción:</b>	El usuario una vez autenticado puede buscar una reserva determinada en el sistema.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Nombre Apellidos Dirección País Provincia/Municipio Email Código postal Teléfono Fecha inicio Fecha fin Nº Adultos Nº Niños Comentarios Tarifa
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Reservas".</li> <li>3.- El sistema muestra mediante una tabla todos las reservas.</li> <li>4.- El usuario busca la reserva.</li> <li>5.- El sistema muestra la lista de reservas.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(5)No existen productos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Ninguna.

Tabla 21: Descripción caso de uso Buscar Reserva

<b>Nombre:</b>	<b>Crear Producto</b>
<b>Descripción:</b>	El usuario una vez autenticado puede crear nuevos productos asociados a un servicio y una categoría.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Nombre Descripción Servicio Categoría Imagen de Producto Galería Archivos
<b>Flujo de Ejecución:</b>	<p>1.- El usuario accede a la aplicación web.</p> <p>2.-El usuario selecciona haciendo click en el menú principal “Gestionar Productos”.</p> <p>3.-El usuario selecciona haciendo click en el submenú principal “Crear Producto”.</p> <p>3.- El sistema muestra un formulario de creación del producto.</p> <p>4.- El usuario introduce los datos.</p> <p>5.- El sistema guarda la información del producto.</p>
<b>Caminos Alternativos:</b>	<p>(4)Errores de datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Producto guardado.

Tabla 22: Descripción caso de uso Crear Producto

<b>Nombre:</b>	<b>Editar Producto</b>
<b>Descripción:</b>	El usuario una vez autenticado puede editar la información mostrada en los productos.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Nombre Descripción Servicio Categoría Imagen de Producto Galería Archivos
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Productos".</li> <li>3.-El usuario selecciona haciendo click en el submenú principal "Buscar Producto".</li> <li>3.- El sistema muestra una tabla con todos los productos.</li> <li>4.- El usuario busca el producto.</li> <li>5.- El usuario selecciona el producto.</li> <li>6.- El sistema muestra un formulario de edición del producto.</li> <li>7.- El usuario introduce los datos o modifica datos.</li> <li>8.- El sistema guarda la información del producto.</li> </ol>
<b>Camino Alternativos:</b>	<p>(3)No existen productos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7)Errores de datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(8)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Producto modificado.

Tabla 23: Descripción caso de uso Editar Producto

<b>Nombre:</b>	<b>Eliminar Producto</b>
<b>Descripción:</b>	El usuario una vez autenticado puede eliminar productos.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal “Gestionar Productos”.</li> <li>3.-El usuario selecciona haciendo click en el submenú principal “Buscar Producto”.</li> <li>3.- El sistema muestra una tabla con todos los productos.</li> <li>4.- El usuario busca el producto.</li> <li>5.- El usuario selecciona el producto.</li> <li>6.- El usuario hace click en el icono eliminar producto.</li> <li>7.- El sistema muestra un aviso de seguridad.</li> <li>8.- El usuario hace click en aceptar.</li> <li>9- El sistema elimina el producto.</li> </ol>
<b>Camino Alternativos:</b>	<p>(3)No existen productos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7)Errores de datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(8)El usuario cancela la operación eliminar.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(9)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Producto eliminado.

Tabla 24: Descripción caso de uso Eliminar Producto

<b>Nombre:</b>	<b>Buscar Producto</b>
<b>Descripción:</b>	El usuario una vez autenticado puede buscar un producto determinado dentro del sistema.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Nombre Descripción Servicio Categoría Imagen de Producto Galería Archivos
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Productos".</li> <li>3.-El usuario selecciona haciendo click en el submenú principal "Buscar Producto".</li> <li>3.- El sistema muestra una tabla con todos los productos.</li> <li>4.- El usuario busca el producto.</li> <li>5.- El sistema muestra una tabla con los productos.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(5)No existen productos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Ninguno.

Tabla 25: Descripción caso de uso Busca Producto

### Usuario administrador:

A continuación mostramos los casos de uso con más detalle para el usuario gestor (Véase Tablas desde la 26 hasta la 37).

<b>Nombre:</b>	<b>Crear Categoría</b>
<b>Descripción:</b>	El usuario una vez autenticado puede crear nuevas categorías de productos asociados a un servicio en el sistema.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Nombre Servicio Imagen de Icono
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"><li>1.- El usuario accede a la aplicación web.</li><li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Categorías".</li><li>3.-El usuario selecciona haciendo click en el submenú principal "Crear Categoría".</li><li>4.- El sistema muestra un formulario de creación de la categoría.</li><li>5.- El usuario introduce los datos de la categoría.</li><li>6.- El sistema guarda la categoría.</li></ol>
<b>Camino Alternativos:</b>	(5)Errores de datos <ul style="list-style-type: none"><li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li></ul> (6)Error en conexión de base de datos no se han podido guardar los datos. <ul style="list-style-type: none"><li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li></ul>
<b>Postcondición:</b>	Categoría guardada.

Tabla 26: Descripción caso de uso Crear Categoría

<b>Nombre:</b>	<b>Editar Categoría</b>
<b>Descripción:</b>	El usuario una vez autenticado puede editar categorías de productos asociados a un servicio en el sistema.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Nombre Servicio Imagen de Icono
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Categorías".</li> <li>3.-El usuario selecciona haciendo click en el submenú principal "Buscar Categoría".</li> <li>4.- El sistema muestra una tabla con las categorías.</li> <li>5.- El sistema muestra un formulario de edición de la categoría.</li> <li>6.- El usuario introduce o modifica los datos de la categoría.</li> <li>7.- El sistema guarda la categoría.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(4)No existen categorías.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(6)Errores de datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Categoría modificada.

Tabla 27: Descripción caso de uso Editar Categoría



<b>Nombre:</b>	<b>Eliminar Categoría</b>
<b>Descripción:</b>	El usuario una vez autenticado puede eliminar categorías de productos asociados a un servicio en el sistema.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Categorías".</li> <li>3.-El usuario selecciona haciendo click en el submenú principal "Buscar Categoría".</li> <li>4.- El sistema muestra una tabla con las categorías.</li> <li>5.- El usuario hace click en el icono eliminar.</li> <li>6.- El sistema muestra un mensaje de seguridad.</li> <li>7.- El usuario hace click en aceptar.</li> <li>8.- El sistema elimina la categoría.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(4)No existen categorías.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7)El usuario cancela la operación eliminar.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(8)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Categoría eliminada.

Tabla 28: Descripción caso de uso Eliminar Categoría

<b>Nombre:</b>	<b>Buscar Categoría</b>
<b>Descripción:</b>	El usuario una vez autenticado puede buscar categorías de productos asociados a un servicio dentro d el sistema.
<b>Precondición:</b>	Usuario autenticado.
<b>Parámetros:</b>	Nombre Servicio Imagen de Icono
<b>Flujo de Ejecución:</b>	<p>1.- El usuario accede a la aplicación web.</p> <p>2.-El usuario selecciona haciendo click en el menú principal “Gestionar Categorías”.</p> <p>3.-El usuario selecciona haciendo click en el submenú principal “Buscar Categoría”.</p> <p>4.- El usuario introduce los parámetros de búsqueda.</p> <p>5.- El sistema muestra una tabla con las categorías.</p>
<b>Caminos Alternativos:</b>	<p>(5)No existen categorías.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Ninguno.

Tabla 29: Descripción caso de uso Buscar Categoría

<b>Nombre:</b>	<b>Crear Usuario</b>
<b>Descripción:</b>	El administrador una vez autenticado puede crear nuevos usuarios y aplicarle los roles correspondientes.
<b>Precondición:</b>	Usuario administrador autenticado.
<b>Parámetros:</b>	Datos personales (email, password). Rol determinado según tipo de usuario.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Usuarios".</li> <li>3.-El usuario selecciona haciendo click en el submenú principal "Crear usuario".</li> <li>4.- El sistema muestra un formulario de creación del usuario.</li> <li>5.- El usuario introduce los datos correspondientes.</li> <li>6.- El sistema guarda la categoría.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(5)Errores de datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(6)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Usuario guardado.

Tabla 30: Descripción caso de uso Crear Usuario

<b>Nombre:</b>	<b>Editar Usuario</b>
<b>Descripción:</b>	El usuario una vez autenticado puede editar usuarios registrados en el sistema
<b>Precondición:</b>	Usuario administrador autenticado.
<b>Parámetros:</b>	Datos personales (password).
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Usuarios".</li> <li>3.-El usuario selecciona haciendo click en el submenú principal "Buscar Usuarios".</li> <li>4.- El sistema muestra una tabla con los usuarios.</li> <li>5.-El usuario selecciona haciendo click en el icono de opciones "Editar" del perfil de usuario a modificar.</li> <li>6.- El sistema muestra un formulario de edición del usuario.</li> <li>7.- El usuario introduce o modifica los datos del usuario.</li> <li>7.- El sistema guarda la modificación del usuario.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(4)No existen usuarios.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(6)Errores de datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Usuario modificado.

Tabla 31: Descripción caso de uso Editar Usuario

<b>Nombre:</b>	<b>Eliminar Usuario</b>
<b>Descripción:</b>	El usuario una vez autenticado puede eliminar un usuario en el sistema.
<b>Precondición:</b>	Usuario administrador autenticado.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Usuarios".</li> <li>3.-El usuario selecciona haciendo click en el submenú principal "Buscar Usuarios".</li> <li>4.- El sistema muestra una tabla con los usuarios.</li> <li>5.- El usuario hace click en el icono eliminar.</li> <li>6.- El sistema muestra un mensaje de seguridad.</li> <li>7.- El usuario hace click en aceptar.</li> <li>8.- El sistema elimina el usuario.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(4)No existe el usuario.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7)El usuario cancela la operación eliminar.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(8)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Usuario eliminado.

Tabla 32: Descripción caso de uso Eliminar Usuario

<b>Nombre:</b>	<b>Buscar Usuario</b>
<b>Descripción:</b>	El usuario una vez autenticado puede buscar usuarios dentro del sistema.
<b>Precondición:</b>	Usuario administrador autenticado.
<b>Parámetros:</b>	Usuario Rol Estado
<b>Flujo de Ejecución:</b>	<p>1.- El usuario accede a la aplicación web.</p> <p>2.-El usuario selecciona haciendo click en el menú principal “Gestionar Usuarios”.</p> <p>3.-El usuario selecciona haciendo click en el submenú principal “Buscar Usuarios”.</p> <p>4.- El usuario introduce los parámetros de búsqueda.</p> <p>5.- El sistema muestra una tabla con los usuarios.</p>
<b>Caminos Alternativos:</b>	<p>(5)No existen usuarios.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Ninguno.

Tabla 33: Descripción caso de uso Buscar Usuario

<b>Nombre:</b>	<b>Crear Servicio</b>
<b>Descripción:</b>	El usuario una vez autenticado puede crear nuevos servicios en el sistema.
<b>Precondición:</b>	Usuario administrador autenticado.
<b>Parámetros:</b>	Nombre Descripción del servicio Imagen de Icono
<b>Flujo de Ejecución:</b>	<p>1.- El usuario accede a la aplicación web.</p> <p>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Servicios".</p> <p>3.-El usuario selecciona haciendo click en el submenú principal "Crear Servicio".</p> <p>4.- El sistema muestra un formulario de creación del servicio</p> <p>5.- El usuario introduce los datos del servicio.</p> <p>6.- El sistema guarda el servicio.</p>
<b>Caminos Alternativos:</b>	<p>(5)Errores de datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(6)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Servicio guardado.

Tabla 34: Descripción caso de uso Crear Servicio

<b>Nombre:</b>	<b>Editar Servicio</b>
<b>Descripción:</b>	El usuario una vez autenticado puede editar servicios en el sistema.
<b>Precondición:</b>	Usuario administrador autenticado.
<b>Parámetros:</b>	Nombre Servicio Imagen de Icono
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Servicios".</li> <li>3.-El usuario selecciona haciendo click en el submenú principal "Buscar Servicio".</li> <li>4.- El sistema muestra una tabla con los servicios.</li> <li>5.- El sistema muestra un formulario de edición del servicio.</li> <li>6.- El usuario introduce o modifica los datos del servicio.</li> <li>7.- El sistema guarda el servicio.</li> </ol>
<b>Caminos Alternativos:</b>	<p>(4)No existen servicios.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(6)Errores de datos</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Servicio modificado.

Tabla 35: Descripción caso de uso Editar Servicio



<b>Nombre:</b>	<b>Eliminar Servicio</b>
<b>Descripción:</b>	El usuario una vez autenticado puede eliminar servicios en el sistema.
<b>Precondición:</b>	Usuario administrador autenticado.
<b>Parámetros:</b>	Ninguno.
<b>Flujo de Ejecución:</b>	<ol style="list-style-type: none"> <li>1.- El usuario accede a la aplicación web.</li> <li>2.-El usuario selecciona haciendo click en el menú principal “Gestionar Servicios”.</li> <li>3.-El usuario selecciona haciendo click en el submenú principal “Buscar Servicio”.</li> <li>4.- El sistema muestra una tabla con los servicios.</li> <li>5.- El usuario hace click en el icono eliminar.</li> <li>6.- El sistema muestra un mensaje de seguridad.</li> <li>7.- El usuario hace click en aceptar.</li> <li>8.- El sistema elimina el servicio.</li> </ol>
<b>Camino Alternativos:</b>	<p>(4)No existen servicios.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(7)El usuario cancela la operación eliminar.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(8)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Servicio eliminado.

Tabla 36: Descripción caso de uso Eliminar Servicio

<b>Nombre:</b>	<b>Buscar Servicio</b>
<b>Descripción:</b>	El usuario una vez autenticado puede buscar servicios dentro del sistema.
<b>Precondición:</b>	Usuario administrador autenticado.
<b>Parámetros:</b>	Nombre Servicio Imagen de Icono
<b>Flujo de Ejecución:</b>	<p>1.- El usuario accede a la aplicación web.</p> <p>2.-El usuario selecciona haciendo click en el menú principal "Gestionar Servicios".</p> <p>3.-El usuario selecciona haciendo click en el submenú principal "Buscar Servicio".</p> <p>4.- El usuario introduce los parámetros de búsqueda.</p> <p>5.- El sistema muestra una tabla con los servicios.</p>
<b>Caminos Alternativos:</b>	<p>(5)No existen servicios.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul> <p>(5)Error en conexión de base de datos no se han podido guardar los datos.</p> <ul style="list-style-type: none"> <li>• Se emite un mensaje informando al usuario y se vuelve al punto de partida.</li> </ul>
<b>Postcondición:</b>	Ninguno.

Tabla 37: Descripción caso de uso Buscar Servicio

## 9.3 Anexo III: Implementación

En este apartado mostraremos varias tecnologías utilizadas en la implementación de este proyecto.

### 9.3.1 TinyMce

Para la implementación de editores de texto dentro del ámbito de la aplicación web hemos utilizado las librerías TinyMce:

TinyMce es un completo editor de texto basado en html y javascript, es un completo editor WYSIWYG y es de código libre (Véase Imagen 31).



Imagen 31: TinyMce

Mediante unas pocas líneas de código en las vistas podremos obtener un potente editor de texto.

Ejemplo de uso:

Código fuente:

```
<!-- Place inside the <head> of your HTML -->
<script type="text/javascript" src="<your installation
path>/tinymce/tinymce.min.js"></script>
<script type="text/javascript">
  tinymce.init({
    selector: "textarea"
  });
</script>

<!-- Place this in the body of the page content -->
<form method="post">
  <textarea></textarea>
</form>
```

Y el editor se ve de esta manera en nuestras vistas (Véase Imagen 32).

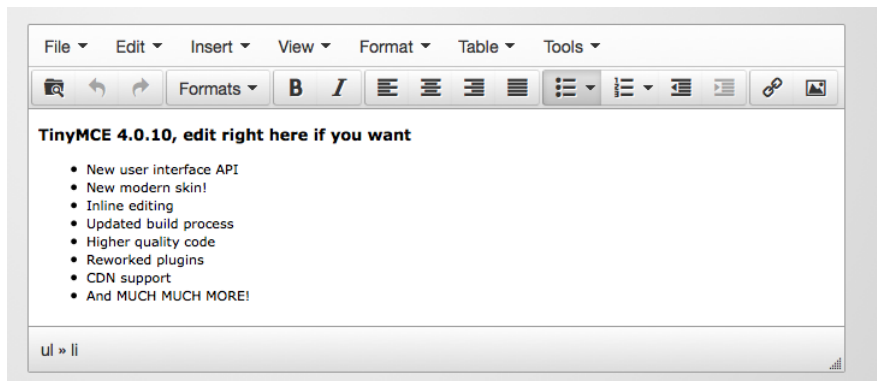


Imagen 32: Editor TinyMce

### 9.3.2 Codeigniter:

En este proyecto se han usado los siguientes helpers de codeigniter:

#### 9.3.2.1 Helper Directory:

El archivo del Helper Directory contiene funciones que lo ayudan a trabajar con directorios.

##### Cargar este Helper

Este helper se carga usando el siguiente código:

```
$this->load->helper('directory');
```

Están disponibles las siguientes funciones:

##### directory\_map('directorio fuente')

Esta función lee la ruta del directorio especificado en el primer parámetro y arma un array con él y todo su contenido. Ejemplo:

```
$map = directory_map('./midirectorio/');
```

**Nota:** Las rutas son casi siempre relativas al archivo **index.php** principal.

Las subcarpetas contenidas dentro del directorio también se mapearán. Si desea controlar la profundidad de la recursión, puede hacer eso usando el segundo parámetro (un entero). Una profundidad de **1** mapeará solamente el directorio de nivel superior:

```
$map = directory_map('./midirectorio/', 1);
```

Por defecto, los archivos ocultos no se incluirán en el array devuelto. Para anular este comportamiento, puede establecer un tercer parámetro a **TRUE** (booleano):

```
$map = directory_map('./midirectorio/', FALSE, TRUE);
```

Cada nombre de carpeta será un índice del array, mientras que sus archivos contenidos se indexarán numéricamente.

### 9.3.2.2 *Helper Email*

El Helper Email provee algunas funciones que lo ayudan a trabajar con emails. Para una solución de email más robusta, vea la Clase Email de CodeIgniter.

#### Cargar este Helper

Este helper se carga usando el siguiente código:

```
$this->load->helper('email');
```

Están disponibles las siguientes funciones:

#### **valid\_email('email')**

Verifica si un email es un email correctamente formateado. Advierta que en realidad no prueba que el email se vaya a recibir, solamente que la dirección está correctamente formada.

Devuelve **TRUE/FALSE**

```
$this->load->helper('email');
```

```
if (valid_email('email@somesite.com'))
```

```
{ echo 'el email es válido';
```

```
} else {
```

```
    echo 'el email no es válido';
```

```
}
```

#### **send\_email('destinatario', 'asunto', 'mensaje');**

Envía un email usando la función **mail()** nativa de PHP. Para una solución de email más robusta, vea la Clase Email de CodeIgniter.

### 9.3.2.3 *Helper Form*

El archivo del Helper Form contiene funciones que lo ayudan a trabajar con formularios.

#### Cargar este Helper

Este helper se carga usando el siguiente código:

```
$this->load->helper('form');
```

Están disponibles las siguientes funciones:

## form\_open()

Crea una etiqueta de apertura de formulario con una URL base **construida a partir de sus preferencias de configuración**. Opcionalmente se le permitirá agregar atributos de formulario y campos de entrada ocultos, y siempre se agregará el atributo **accept-charset** basado en el valor del conjunto de caracteres en su archivo de configuración.

El beneficio principal de usar esta etiqueta en lugar de codificar su propio HTML, es que le permite a su sitio ser más portable si cambian las URLs.

Aquí hay un ejemplo simple:

```
echo form_open('email/send');
```

El ejemplo anterior crearía un formulario que apunta a su URL base, más el segmento de URI "email/send":

## Agregar Atributos

Los atributos se pueden agregar al pasar un array asociativo al segundo parámetro:

El ejemplo anterior crearía un formulario similar a esto:

```
<form method="post" accept-charset="utf-8"
  action="http://example.com/index.php/email/send" class="email" id="mi_form" />
```

## Agregar Campos de Entrada Ocultos

Los campos ocultos se pueden agregar al pasar un array asociativo al tercer parámetro:

```
<form method="post" accept-charset="utf-8"
  action="http://example.com/index.php/email/send" />
```

```
$attributes = array('class' => 'email', 'id' => 'mi_form');
```

```
echo form_open('email/send', $attributes);
```

```
<form method="post" accept-charset="utf-8"
```

```
  action="http://example.com/index.php/email/send" class="email" id="mi_form" />
```

```
$hidden = array('username' => 'Jose', 'member_id' => '234');
```

```
echo form_open('email/send', "", $hidden);
```

## form\_open\_multipart()

Esta función es absolutamente idéntica a **form\_open()** anterior, excepto que agrega un atributo multipart que es necesario si quisiera usar el formulario para subir archivos.

## form\_hidden()

Le permite generar campos de entrada ocultos. Puede presentar una cadena "name"/"value" para crear un campo:

O puede presentar un array asociativo para crear varios campos:

```
<form method="post" accept-charset="utf-8"
action="http://example.com/index.php/email/send">
<input type="hidden" name="username" value="Jose" />
<input type="hidden" name="member_id" value="234" />
form_hidden('usuario', 'natalia_natalia');
```

// Produciría:

```
<input type="hidden" name="usuario" value="natalia_natalia" />
$data = array(
'name' => 'Natalia Natalia',
'email' => 'nn@example.com',
'url' => 'http://example.com'
);
echo form_hidden($data);
```

// Produciría:

```
<input type="hidden" name="name" value="Natalia Natalia" />
<input type="hidden" name="email" value="nn@example.com" />
<input type="hidden" name="url" value="http://example.com" />
```

### **form\_input()**

Le permite generar un campo de entrada de texto estándar. Mínimamente puede pasarle el nombre del campo y el valor en el primer y segundo parámetros:

```
echo form_input('usuario', 'natalia_natalia');
```

O puede pasarle un array asociativo conteniendo cualquier dato que desee que su formulario contenga:

```
$data = array(
    'name'
    'id'
```

```

        'value'
        'maxlength'
        'size'
        'style'
    => 'usuario',
        => 'usuario',
    => 'natalia_natalia',
    => '100',
        => '50',
        => 'width:50%',
    );
echo form_input($data);
// Produciría:
<input type="text" name="usuario" id="usuario" value="natalia_natalia"
    maxlength="100" size="50" style="width:50%" />

```

Si quisiera que su formulario contuviera algún dato adicional, como Javascript, podría pasárselo en una cadena como tercer parámetro:

#### **form\_password()**

Esta función es idéntica en todos los aspectos a la función **form\_input()** anterior salvo que permite generar un tipo "password".

#### **form\_upload()**

Esta función es idéntica en todos los aspectos a la función **form\_input()** anterior salvo que permite generar un tipo "file", permitiéndole usarla para subir archivos.

#### **form\_textarea()**

Esta función es idéntica en todos los aspectos a la función **form\_input()** anterior salvo que permite generar un tipo "textarea". **Nota:** En lugar de los atributos "maxlength" y "size" del ejemplo anterior, deberá establecer "rows" y "cols".

#### **form\_dropdown()**

Le permite crear un campo de lista desplegable estándar. El primer parámetro contendrá el nombre del campo, el segundo parámetro contendrá un array asociativo de opciones y el



tercer parámetro contendrá el valor que desee que se seleccione. También puede pasarle un array de varios elementos mediante el tercer parámetro y CodeIgniter creará un selector múltiple. Ejemplo:

```
$js = 'onClick="alguna_funcion()";  
  
echo form_input('usuario', 'natalia_natalia', $js);  
  
$opciones = array(  
    's' => 'Camisa S',  
    'm' => 'Camisa M',  
    'l' => 'Camisa L',  
    'xl' => 'Camisa XL',  
);  
  
$camisas_en_venta = array('s', 'l');  
  
echo form_dropdown('camisas', $opciones, 'l');  
  
// Produciría:  
  
select name="camisas">  
    <option value="s">Camisa S</option>  
    <option value="m">Camisa M</option>  
    <option value="l" selected="selected">Camisa L</option>  
    <option value="xl">Camisa XL</option>  
</select>  
  
echo form_dropdown('camisas', $opciones, $camisas_en_venta);  
  
// Produciría:  
  
<select name="camisas" multiple="multiple">  
    <option value="s" selected="selected">Camisa S</option>  
    <option value="m">Camisa M</option>  
    <option value="l" selected="selected">Camisa L</option>  
    <option value="xl">Camisa XL</option>  
</select>
```

Si quisiera que la apertura `<select>` contuviera datos adicionales, como un atributo `id` o JavaScript, puede pasarle una cadena como cuarto parámetro:

Si el array pasado como `$opciones` es multidimensional, `form_dropdown()` producirá un `<optgroup>` con la clave del array como rótulo.

### `form_multiselect()`

Le permite crear un campo de selección múltiple estándar. El primer parámetro contendrá el nombre del campo, el segundo parámetro contendrá un array asociativo de opciones, y el tercer parámetro contendrá el valor o valores que desea que estén seleccionados. El uso de parámetros es idéntico a usar el anterior `form_dropdown()`, salvo por supuesto que el nombre del campo tendrá que usar la sintaxis del array `POST`, por ejemplo `foo[]`.

### `form_fieldset()`

Le permite generar los campos "fieldset"/"legend".

```
$js = 'id="camisas" onChange="alguna_funcion();"';
echo form_dropdown('camisas', $opciones, 'large', $js);
echo form_fieldset('Información de la dirección');
echo "<p>contenido del conjunto de campos aquí</p>\n";
echo form_fieldset_close();

// Produce

<fieldset>
<legend>Información de la dirección</legend>
    <p>contenido del conjunto de campos aquí</p>
</fieldset>
```

Similar a otras funciones, puede enviar un array asociativo en el segundo parámetro si prefiere establecer atributos adicionales.

```
$attributes = array('id' => 'address_info', 'class' => 'address_info');
echo form_fieldset('Información de la dirección', $attributes);
echo "<p>contenido del conjunto de campos aquí</p>\n";
echo form_fieldset_close();

// Produce

<fieldset id="address_info" class="address_info">
```

```
<legend>Información de la dirección</legend>
```

```
<p>contenido del conjunto de campos aquí</p>
```

```
</fieldset>
```

### **form\_fieldset\_close()**

Produce una etiqueta `</fieldset>` de cierre. La única ventaja de usar esta función es que le permite pasar datos a ella, los que se agregarán debajo de la etiqueta. Por ejemplo:

### **form\_checkbox()**

Le permite generar un campo de casilla de verificación. Ejemplo sencillo:

El tercer parámetro contiene un booleano **TRUE/FALSE** para determinar si la caja se debería marcar o no. Similar a las otras funciones de formulario en este helper, también puede pasarle a la función un array de atributos:

```
$string = "</div></div>";
```

```
echo fieldset_close($string);
```

```
// Produciría:
```

```
</fieldset>
```

```
</div></div>
```

```
echo form_checkbox('newsletter', 'accept', TRUE);
```

```
// Produciría:
```

```
<input type="checkbox" name="newsletter" value="accept" checked="checked" />
```

```
$data = array(
```

```
    'name'    => 'newsletter',
```

```
    'id'      => 'newsletter',
```

```
    'value'   => 'accept',
```

```
    'checked' => TRUE,
```

```
    'style'   => 'margin:10px',
```

```
);
```

```
echo form_checkbox($data);
```

```
// Produciría:
```

```
input type="checkbox" name="newsletter" id="newsletter" value="accept"
```

```
checked="checked" style="margin:10px" />
```

Como con otras funciones, si quisiera que la etiqueta contenga datos adicionales, como JavaScript, puede pasarle una cadena como cuarto parámetro:

```
$js = 'onClick="alguna_funcion()";  
echo form_checkbox('newsletter', 'accept', TRUE, $js);
```

Esta función es idéntica en todos los aspectos a la función `form_checkbox()` anterior, salvo que establece un tipo "radio".

### `form_submit()`

Le permite generar un botón enviar estándar. Ejemplo sencillo:

Similar a otras funciones, puede pasar un array asociativo en el primer parámetro si prefiere establecer sus propios atributos. El tercer parámetro le permite agregar datos adicionales al formulario, como JavaScript.

### `form_label()`

Le permite generar un `<label>`. Ejemplo sencillo:

Similar a otras funciones, puede pasar un array asociativo en el tercer parámetro si prefiere establecer sus propios atributos.

```
echo form_submit('mi_submit', 'Enviar mensaje');  
  
// Produciría:  
  
<input type="submit" name="mi_submit" value="Enviar mensaje" />
```

```
echo form_label('Cual es su nombre?', 'username');
```

```
// Produciría:
```

```
<label for="username">Cual es su nombre?</label>
```

```
$attributes = array(  
    'class' => 'mi_clase',  
    'style' => 'color: #000;',  
);  
  
echo form_label('Cual es su nombre?', 'username', $attributes);  
  
// Produciría:
```

```
<label for="username" class="mi_clase" style="color: #000;">Cual es su nombre?
</label>
```

### **form\_reset()**

Le permite generar un botón estándar de reset. El uso es idéntico a [form\\_submit\(\)](#). [form\\_button\(\)](#)

Le permite generar un elemento de botón estándar. Mínimamente puede pasar el nombre del botón y contenido en el primer y segundo parámetros:

```
echo form_button('nombre','contenido');

// Produciría

<button name="nombre" type="button">contenido</button>
```

O puede pasarle un array asociativo conteniendo cualquier dato que desee que su formulario contenga:

```
$data = array(
    'name' => 'button',
    'id' => 'button',
    'value' => 'true',
    'type' => 'reset',
    'content' => 'Reset'
);

echo form_button($data);

// Produciría:

<button name="button" id="button" value="true" type="reset">Reset</button>
```

Si desea que su formulario contenga algunos datos adicionales, como JavaScript, puede pasarlos como una cadena en el tercer parámetro:

### **form\_close()**

Produce una etiqueta `</form>` de cierre. La única ventaja de usar esta función es que le permite pasar datos a ella, los que se agregarán debajo de la etiqueta. Por ejemplo:

```
$js = 'onClick="alguna_funcion()";
```

```

echo form_button('mi_boton', 'Clic aquí', $js);

$string = "</div></div>";

echo form_close($string);

// Produciría:

</form>

</div></div>

```

### form\_prep()

Le permite usar con seguridad HTML y caracteres tales como comillas dentro de elementos de formulario sin romper el formulario. Considere este ejemplo:

Como la cadena anterior contiene un conjunto de comillas, ese causará que el formulario se rompa. La función `form_prep` convierte HTML para que se pueda usar en forma segura:

```
$string = 'Aquí hay una cadena que contiene texto entre "comillas".';
```

```
<input type="text" name="mi_formulario" value="$string" />
```

```
<input type="text" name="mi_formulario" value="<?php echo form_prep($string); ?>" />
```

**Nota:** Si usa alguna de las funciones del Helper Form listadas en esta página los valores del formulario se prepararán automáticamente, por lo que no hay necesidad de llamar a esta función. Úsela solamente si está creando sus propios elementos de formulario.

### set\_value()

Le permite establecer el valor de una entrada de formulario o de un textarea. Tiene que suministrar el nombre del campo mediante el primer parámetro de la función. El segundo parámetro (opcional) le permite establecer un valor por defecto para el formulario. Ejemplo:

El formulario anterior mostrará "0" cuando se cargue por primera vez. [set\\_select\(\)](#).

Si usa un menú `<select>`, esta función le permite mostrar el elemento de menú que se seleccionó. El primer parámetro tiene que contener el nombre del menú select, el segundo parámetro tiene que contener el valor de cada elemento y el tercer parámetro (opcional) le permite establecer un elemento por defecto (usar el booleano `TRUE/FALSE`). Ejemplo:

```
<input type="text" name="cantidad"
```

```
    value="<?php echo set_value('cantidad', '0'); ?>" size="50" />
```

```
<select name="mi_select">
```

```
    <option value="uno" <?php echo set_select('mi_select', 'uno', TRUE); ?> >
```

```
    Uno</option>
```

```

<option value="dos" <?php echo set_select('mi_select', 'dos'); ?> >
Dos</option>
<option value="tres" <?php echo set_select('mi_select', 'tres'); ?> >
Tres</option>
</select>

```

### set\_checkbox()

Le permite mostrar una casilla de verificación en el estado en que se envió. El primer parámetro tiene que contener el nombre de la casilla de verificación, el segundo parámetro tiene que contener su valor y el tercer parámetro (opcional) le permite establecer un elemento por defecto (usar el booleano **TRUE/FALSE**). Ejemplo:

### set\_radio()

Le permite mostrar botones de radio en el estado en que se enviaron. Esta función es idéntica a la función **set\_checkbox()** anterior.

```

<input type="checkbox" name="mi_casilla" value="1"
    <?php echo set_checkbox('mi_casilla', '1'); ?> />
<input type="checkbox" name="mi_casilla" value="2"
    <?php echo set_checkbox('mi_casilla', '2'); ?> />
<input type="radio" name="mi_radio" value="1"
    <?php echo set_radio('mi_radio', '1', TRUE); ?> />
<input type="radio" name="mi_radio" value="2"
    <?php echo set_radio('mi_radio', '2'); ?> />

```

#### 9.3.2.4 Helper Database

CodeIgniter viene con una clase de base de datos abstracta muy rápida y completa que soporta tanto las estructuras tradicionales como los patrones **Active Record**. Las funciones de base de datos ofrecen una sintaxis clara y sencilla.

#### Inicio Rápido: Código de Ejemplo

La siguiente página contiene código de ejemplo mostrando como se usa la clase **database**. Para obtener detalles completos, por favor lea las páginas individuales que describen cada función.

### Inicializar la Clase Database

El siguiente código carga e inicializa la clase database basado en los valores de su configuración:

```
$this->load->database();
```

Una vez cargada la clase, está lista para usarse como se describe a continuación.

### Consulta Estándar con Resultados Múltiples (Versión Objetos)

```
$query = $this->db->query('SELECT nombre, titulo, email FROM mi_tabla');
```

```
foreach ($query->result() as $row)
```

```
{
```

```
    echo $row->titulo;
```

```
    echo $row->nombre;
```

```
    echo $row->email;
```

```
}
```

```
echo 'Resultados totales: ' . $query->num_rows();
```

La función **result()** anterior devuelve un array de objetos. Ejemplo: **\$row->titulo**.

### Consulta Estándar con Resultados Múltiples (Versión Array)

```
$query = $this->db->query('SELECT nombre, titulo, email FROM mi_tabla');
```

```
foreach ($query->result_array() as $row)
```

```
{
```

```
    echo $row['titulo'];
```

```
    echo $row['nombre'];
```

```
    echo $row['email'];
```

```
}
```



La función `result_array()` anterior devuelve un array de índices de array estándar. Ejemplo: `$row['titulo']`.

### Comprobar Resultados

Si ejecuta consultas que pueden no producir resultados, le aconsejamos probar primero de usar la función `num_rows()`:

```
$query = $this->db->query("SU CONSULTA");
```

```
if ($query->num_rows() > 0)
```

```
{
```

```
    foreach ($query->result() as $row)
```

```
    {
```

```
        echo $row->titulo;
```

```
        echo $row->nombre;
```

```
        echo $row->cuerpo;
```

```
    }  
}
```

### Consulta Estándar con Resultado Simple

La función `row()` anterior devuelve un objeto. Ejemplo: `$row->nombre`.

### Consulta Estándar con Resultado Simple (Versión Array)

La función `row_array()` anterior devuelve un array. Ejemplo: `$row['nombre']`.

### Inserción Estándar

#### Consulta Active Record

El Patrón **Active Record** le da una forma simplificada de devolver los datos:

```
$query = $this->db->query('SELECT nombre FROM mi_tabla LIMIT 1');
```

```
$row = $query->row();
```

```
echo $row->nombre;
```

```
$query = $this->db->query('SELECT nombre FROM mi_tabla LIMIT 1');
```

```
$row = $query->row_array();
```

```
echo $row['nombre'];
```

```

$sql = "INSERT INTO mi_tabla (titulo, nombre)
        VALUES (".$this->db->escape($titulo).", ".$this->db->escape($nombre).")";
$this->db->query($sql);
echo $this->db->affected_rows();
$query = $this->db->get('nombre_de_tabla');
foreach ($query->result() as $row)
{
    echo $row->titulo;
}

```

La función **get()** anterior devuelve todos los resultados desde la tabla suministrada. La clase **Active Record** contiene un complemento completo de funciones para trabajar con datos.

#### Inserción Active Record

```

$data = array(
    'titulo' => $titulo,
    'nombre' => $nombre,
    'fecha' => $fecha
);
$this->db->insert('mi_tabla', $data);
// Produce: INSERT INTO mi_tabla (titulo, nombre, fecha) VALUES ('{$titulo}',
// '{$nombre}', '{$fecha}')

```

#### Configuración de la Base de Datos

**CodeIgniter** tiene un archivo de configuración que le permite almacenar los valores de conexión de la base de datos (usuario, contraseña, nombre de la base de datos, etc.). El archivo de configuración está ubicado en **application/config/database.php**. También puede establecer valores de conexión de base de datos para entornos específicos al ubicar **database.php** en la carpeta de configuración del entorno respectivo.

Los parámetros de configuración se almacenan en un array multidimensional con este prototipo:

```

$db['default']['hostname'] = "localhost";

```

```
$db['default']['username'] = "root";  
$db['default']['password'] = "";  
$db['default']['database'] = "nombre_de_base_de_datos";  
$db['default']['dbdriver'] = "mysql";  
$db['default']['dbprefix'] = "";  
$db['default']['pconnect'] = TRUE;  
$db['default']['db_debug'] = FALSE;  
$db['default']['cache_on'] = FALSE;  
$db['default']['cachedir'] = "";  
$db['default']['char_set'] = "utf8";  
$db['default']['dbcollat'] = "utf8_general_ci";  
$db['default']['swap_pre'] = "";  
$db['default']['autoinit'] = TRUE;  
$db['default']['stricton'] = FALSE;
```

La razón por la que usamos un array multidimensional en lugar de uno más simple, es para permitirle almacenar opcionalmente varios conjuntos de valores de conexión. Si, por ejemplo, ejecuta varios entornos (desarrollo, producción, prueba, etc.) bajo una instalación simple, puede configurar un grupo de conexión para cada uno y luego cambiar entre grupos según se necesite. Por ejemplo, para configurar un entorno "prueba" podría hacer esto:

```
$db['prueba']['hostname'] = "localhost";  
$db['prueba']['username'] = "root";  
$db['prueba']['password'] = "";  
$db['prueba']['database'] = "nombre_de_base_de_datos";  
$db['prueba']['dbdriver'] = "mysql";  
$db['prueba']['dbprefix'] = "";  
$db['prueba']['pconnect'] = TRUE;  
$db['prueba']['db_debug'] = FALSE;  
$db['prueba']['cache_on'] = FALSE;  
$db['prueba']['cachedir'] = "";
```

```
$db['prueba']['char_set'] = "utf8";  
$db['prueba']['dbcollat'] = "utf8_general_ci";  
$db['prueba']['swap_pre'] = "";  
$db['prueba']['autoinit'] = TRUE;  
$db['prueba']['stricton'] = FALSE;
```

Entonces, para decirle globalmente al sistema que use ese grupo, podría establecer esta variable localizada en el archivo de configuración:

```
$active_group = "prueba";
```

**Nota:** El nombre "prueba" es arbitrario. Puede ser cualquiera que desee. Por defecto usamos la palabra "default" para la conexión primaria, pero también se la puede renombrar a algo más relevante en el proyecto.

### Active Record

La Clase **Active Record** se habilita o deshabilita globalmente estableciendo la variable **\$active\_record** en el archivo de configuración de la base de datos a **TRUE/FALSE** (booleano). Si no está usando la Clase **Active Record**, establecer el valor a **FALSE** usará menos recursos cuando la Clase **Database** esté inicializada.

```
$active_record = TRUE;
```

### Explicación de valores:

**hostname** - El nombre del host de su servidor de base de datos. Frecuentemente es "localhost".

**username** - El usuario utilizado para conectar con la base de datos.

**password** - La contraseña utilizada para conectar con la base de datos.

**database** - El nombre de la base de datos con la que se quiere conectar.

**dbdriver** - El tipo de base de datos. Por ejemplo: mysql, postgres, odbc, etc. Tiene que especificarse en minúsculas.

**dbprefix** - Un prefijo opcional para tablas que se agregará al nombre de las tablas al ejecutar consultas del Active Record. Esto permite que varias instalaciones de CodeIgniter compartan una sola base de datos.

**pconnect** - **TRUE/FALSE** (booleano) - Si desea usar una conexión persistente.

**db\_debug** - **TRUE/FALSE** (booleano) - Si se tienen que mostrar los errores de la base de datos.

**cache\_on** - **TRUE/FALSE** (booleano) - Si está habilitado el cacheo de consultas, ver también el Driver Caché.

**cachedir** - La ruta absoluta en el servidor del directorio de cacheo de consultas de la base de datos.

**char\_set** - El conjunto de caracteres usado en la comunicación con la base de datos.

**dbcollat** - La codificación de caracteres usada en la comunicación con la base de datos.

**swap\_pre** - Un prefijo de tabla por defecto que se tiene que intercambiar con **dbprefix**. Esto es útil para aplicaciones distribuidas donde puede correr consultas escritas manualmente, y se necesita que el prefijo siga siendo personalizable por el usuario.

**autoinit** - Si conectar o no automáticamente a la base de datos cuando se carga la biblioteca. Si está establecido a **FALSE**, la conexión tomará lugar antes de ejecutar la primera consulta.

**stricton** - **TRUE/FALSE** (booleano) - Si forzar conexiones en "Modo Estricto", bueno para asegurar el SQL estricto mientras se desarrolla una aplicación.

**port** - El número de puerto de base de datos. Para usar este valor tiene que agregar una línea al array de configuración de la base de datos.

```
$db['default']['port'] = 5432;
```

**Nota:** Algunas clases de CodeIgniter, tales como Sessions, necesitan que Active Records esté habilitado para acceder a ciertas funcionalidades.

**Nota:** Para bases de datos MySQL y MySQLi, este parámetro se usa solamente como una copia de respaldo si el servidor está corriendo PHP < 5.2.3 o MySQL < 5.0.7 (y las consultas de creación de tablas se hicieron con DB Forge). Hay una incompatibilidad en PHP con [mysql\\_real\\_escape\\_string\(\)](#) que puede hacer su sitio vulnerable a inyecciones de SQL si está usando un conjunto de caracteres multi-byte y se están corriendo versiones menores a esas. Los sitios que usan conjuntos de caracteres y ordenamiento Latin-1 o UTF-8 no están afectados.

**Nota:** Dependiendo de que plataforma esté usando (MySQL, Postgres, etc.) no se necesitarán todos los valores. Por ejemplo, al usar SQLite no necesitará suministrar un usuario o contraseña, y la base de datos será la ruta a su archivo de base de datos. La información anterior asume que está usando MySQL.

## Conectar a una Base de Datos

Hay dos formas de conectar a una base de datos:

### Conectar Automáticamente

La función "auto conectar" cargará e instanciará la clase **database** con cada carga de página. Para habilitar la "auto conexión", agregar la palabra database al array de la biblioteca, como se indica en el siguiente archivo:

[application/config/autoload.php](#)

### Conectar Manualmente

Si solamente algunas páginas requieren de conectividad de base de datos, puede conectar manualmente a la base de datos agregando esta línea de código en cualquier función donde se la necesite, o en el constructor de la clase para hacer que la base de datos sea global a esa clase.

```
$this->load->database();
```

Si la función anterior **no** contiene información alguna en el primer parámetro, conectará con el grupo especificado en su archivo de configuración de base de datos. Para la mayoría de la gente, este es el método preferido.

### Parámetros Disponibles

1. Los valores de conexión de la base de datos, pasado tanto como un array o como una cadena DSN. 2. **TRUE/FALSE** (booleano). Si devolver el ID de conexión (ver más abajo Conectar a Varias Bases de Datos). 3. **TRUE/FALSE** (booleano). Si habilitar la Clase **Active Record**. Establecido a **TRUE** por defecto.

### Conectar Manualmente a una Base de Datos

El primer parámetro de esta función se puede usar **opcionalmente** para especificar un grupo de base de datos en particular en el archivo de configuración, o incluso puede presentar valores de conexión para una base de datos que no se especifica en su archivo de configuración. Ejemplos:

Para elegir un grupo específico del archivo de configuración puede hacer esto:

```
$this->load->database('nombre_de_grupo');
```

Donde **nombre\_de\_grupo** es el nombre del grupo de conexión en su archivo de configuración. Para conectar manualmente a la base de datos deseada, puede pasar un array de valores:

```
$config['hostname'] = "localhost";
```

```
$config['username'] = "mi_usuario";
```

```
$config['password'] = "mi_contraseña";
```

```
$config['database'] = "mi_DB";
```

```

$config['dbdriver'] = "mysql";
$config['dbprefix'] = "";
$config['pconnect'] = FALSE;
$config['db_debug'] = TRUE;
$config['cache_on'] = FALSE;
$config['cachedir'] = "";
$config['char_set'] = "utf8";
$config['dbcollat'] = "utf8_general_ci";
$this->load->database($config);

```

Para mayor información sobre cada uno de estos valores, por favor vea la página de configuración.

O puede presentar los valores de base de datos como un Nombre de Fuente de Datos (DSN). Los DSNs tienen que tener este prototipo:

Para anular los valores de configuración por defecto al conectar con una cadena DSN, agregar las variables de configuración como un query string.

### Conectar a Varias Bases de Datos

Si necesita conectar a más de una base de datos simultáneamente, puede hacer lo siguiente:

Estableciendo el segundo parámetro a **TRUE** (booleano) la función devolverá el objeto database. Al conectar de este modo, usará su nombre de objeto para ejecutar comandos en lugar de la sintaxis usada en

esta guía. En otras palabras, en lugar de ejecutar comandos con:

En su lugar usará:

```
$dsn = 'dbdriver://usuario:contraseña@nombre_host/base_de_datos';
```

```
$this->load->database($dsn);
```

```
$dsn = 'dbdriver://usuario:contraseña@nombre_host/base_de_datos?
```

```
char_set=utf8&dbcollat=utf8_general_ci&cache_on=true&cachedir=/ruta/al/cache';
```

```
$this->load->database($dsn);
```

```
$DB1 = $this->load->database('grupo_uno', TRUE);
```

```
$DB2 = $this->load->database('grupo_dos', TRUE);
```

**Nota:** Cambiar las palabras "grupo\_uno" y "grupo\_dos" a los nombres de grupo específicos para los que está conectando (o pase los valores de conexión como se indicó antes).

```
$this->db->query();
```

```
$this->db->result();
```

etc...

```
$DB1->query();
```

```
$DB1->result();
```

etc...

### Reconectar / Mantener la Conexión Viva

Si se excede el tiempo de espera de inactividad del servidor de base de datos mientras está realizando alguna tarea pesada de PHP (por ejemplo, procesando una imagen), debería considerar hacer ping al servidor usando el método **reconnect()** antes de enviar otras consultas, el cual puede mantener la conexión viva o restablecerla.

```
$this->db->reconnect();
```

### Cerrar Manualmente la Conexión

Mientras que CodeIgniter se encarga inteligentemente de cerrar las conexiones de bases de datos, la conexión se puede cerrar explícitamente.

```
$this->db->close();
```

### Consultas

```
$this->db->query()
```

Para realizar una consulta, usar la siguiente función:

```
$this->db->query('PONER AQUI LA CONSULTA');
```

La función **query()** devuelve un objeto de resultado de base de datos cuando se ejecutan consultas tipo "leer", las cuales puede usar para mostrar sus resultados. Cuando se ejecutan consultas tipo "escribir" la función simplemente devuelve **TRUE** o **FALSE** dependiendo del éxito o fracaso. Al devolver datos, normalmente asignará la consulta a una variable, así:

```
$query = $this->db->query('PONER AQUI LA CONSULTA');
```

```
$this->db->simple_query()
```



Esta es una versión simplificada de la función `$this->db->query()`. **SOLAMENTE** devuelve **TRUE/FALSE** en caso de éxito o fracaso. **NO** devuelve un conjunto de resultados de base de datos, ni establece el temporizador de consultas, ni compila datos enlazados, o almacena consultas para depuración. Simplemente le permite realizar una consulta. La mayoría de los usuarios rara vez usan esta función.

### Agregar Manualmente Prefijos de Base de Datos

Si configuró un prefijo de base de datos y quisiera agregarlo manualmente, puede usar lo siguiente.

### Proteger Identificadores

En muchas bases de datos es recomendable proteger las tablas y nombres de campos - por ejemplo con backticks en MySQL. Las consultas del **Active Record** están protegidas automáticamente, sin embargo, si necesita proteger manualmente un identificador, puede usar:

```
$this->db->protect_identifiers('nombre_de_tabla');
```

Esta función también agregará un prefijo de tabla a su tabla asumiendo que tiene un prefijo establecido en su archivo de configuración de la base de datos. Para habilitar el prefijado, establecer a **TRUE** (booleano) el segundo parámetro:

```
$this->db->protect_identifiers('nombre_de_tabla', TRUE);
```

### Escapar Consultas

Es una muy buena práctica de seguridad escapar los datos antes de enviarlos a la base de datos. CodeIgniter tiene tres métodos para ayudarle a hacer esto:

1. **`$this->db->escape()`**: Esta función determina el tipo de datos por lo que solamente puede escapar datos de cadena. También automáticamente agrega comillas simples alrededor de los datos, por lo que Ud no tiene que hacerlo:

```
$this->db->dbprefix('nombre_de_tabla');
```

```
// imprime: prefijo_nombre_de_tabla
```

```
$sql = "INSERT INTO tabla (titulo) VALUES(".$this->db->escape($titulo).")";
```

2. **`$this->db->escape_str()`**: Esta función escapa los datos pasados a ella, independientemente del tipo. La mayoría de las veces usará la función anterior en lugar de esta. Usar esta función así:

3. **`$this->db->escape_like_str()`**: Se debería usar este método cuando las cadenas se usan en condiciones LIKE por lo que los comodines de LIKE ('%', '\_') en cadenas también se escapan adecuadamente.

### Enlazado de Consultas

El enlazado le permiten simplificar la sintaxis de sus consultas, al permitir que el sistema junte las consultas por Ud. Considere el siguiente ejemplo:

Los signos de pregunta en la consulta se reemplazan automáticamente con los valores en el array en el segundo parámetro de la función **query()**.

```
$sql = "INSERT INTO tabla (titulo) VALUES(""
```

```
    $this->db->escape_str($titulo).")";
```

```
$search = '20% raise';
```

```
$sql = "SELECT id FROM tabla WHERE column LIKE '%".
```

```
$this->db->escape_like_str($search)."%";
```

```
$sql = "SELECT * FROM alguna_tabla WHERE id = ? AND estado = ? AND autor = ?";
```

```
$this->db->query($sql, array(3, 'vivo', 'Ricardo'));
```

El beneficio secundario de usar enlazado es que los valores se escapan automáticamente, produciendo consultas más seguras. No tiene que recordar escapar manualmente los datos; el motor lo hace automáticamente por Ud.

## Generar Resultados de Consultas

Hay varias formas de generar resultados de consultas:

### result()

Esta función devuelve un **array de objetos**, o un **array vacío** en caso de falla. Normalmente, Ud usará esta función en un bucle **foreach**, como este:

```
$query = $this->db->query("SU CONSULTA");
```

```
foreach ($query->result() as $row)
```

```
{
```

```
    echo $row->title;
```

```
    echo $row->name;
```

```
    echo $row->body;
```

```
}
```

La función anterior es un alias de **result\_object()**. Si ejecuta consultas que pueden **no** producir un resultado, lo animamos a probar primero el resultado:

```
$query = $this->db->query("SU CONSULTA");
```

```

if ($query->num_rows() > 0)
{
    foreach ($query->result() as $row)
    {
        echo $row->title;

        echo $row->name;

        echo $row->body;
    }
}

```

También puede pasar una cadena a **result()**, la que representa una clase a instanciar por cada objeto de resultado (**nota:** se tiene que cargar esta clase).

### result\_array()

Esta función devuelve un resultado de consulta como un array puro, o un array vacío cuando no se produce resultado. Normalmente, usará esta función dentro de un bucle **foreach**, así:

```

$query = $this->db->query("SELECT * FROM users;");

foreach ($query->result('User') as $user)
{
    echo $row->name; // llamar atributos

    echo $row->reverse_name(); // o métodos definidos en la clase 'User'
}

$query = $this->db->query("SU CONSULTA");

foreach ($query->result_array() as $row)
{
    echo $row['title'];

    echo $row['name'];

    echo $row['body'];
}

```

### row()

Esta función devuelve una fila simple de resultado. Si su consulta tiene más de una fila, devolverá solamente la primera. El resultado se devuelve como un **objeto**. Aquí hay un ejemplo de uso:

```
$query = $this->db->query("SU CONSULTA");  
  
if ($query->num_rows() > 0)  
{  
    $row = $query->row();  
  
    echo $row->title;  
  
    echo $row->name;  
  
    echo $row->body;  
  
}
```

Si quiere que se devuelva una fila específica, tiene que enviar el número de fila como un dígito en el primer parámetro:

```
$row = $query->row(5);
```

También puede agregar un segundo parámetro de cadena, que es el nombre de la clase con la que instanciar la fila:

### **row\_array()**

Idéntica a la función **row()** anterior, salvo que devuelve un array. Ejemplo:

```
$query = $this->db->query("SELECT * FROM users LIMIT 1;");  
  
$query->row(0, 'User')  
  
echo $row->name; // llamar atributos  
  
echo $row->reverse_name(); // o métodos definidos en la clase 'User'  
  
$query = $this->db->query("SU CONSULTA");  
  
if ($query->num_rows() > 0)  
{  
    $row = $query->row_array();  
  
    echo $row['title'];  
  
    echo $row['name'];  
  
}
```

```
    echo $row['body'];  
}
```

Si quiere que se devuelva una fila específica, puede enviar el número de fila como un dígito en el primer parámetro:

```
$row = $query->row_array(5);
```

Además, puede ir hacia adelante/atrás/primer/última en sus resultados, usando estas variaciones:

Por defecto, devuelven un objeto, a menos que ponga la palabra "array" en el parámetro:

### **Funciones Helper de Resultados `$query->num_rows()`**

Cantidad de filas devueltas por la consulta. **Nota:** En este ejemplo, `$query` es la variable a la que se le asigna el objeto de resultado de la consulta:

### **`$query->num_fields()`**

Cantidad de CAMPOS (columnas) devueltos por la consulta. Asegurarse de llamar la función usando su objeto de resultado de consulta:

### **`$query->free_result()`**

Libera la memoria asociada con el resultado y borra el ID del recurso de resultado. Normalmente PHP libera su memoria automáticamente al final de la ejecución del script. Sin embargo, si está ejecutando un montón de consultas en un script en particular, podría querer liberar el resultado después de que cada resultado de consulta se haya generado de forma de reducir el consumo de memoria. Ejemplo:

```
$row = $query->first_row()  
  
$row = $query->last_row()  
  
$row = $query->next_row()  
  
$row = $query->previous_row()  
  
$row = $query->first_row('array')  
  
$row = $query->last_row('array')  
  
$row = $query->next_row('array')  
  
$row = $query->previous_row('array')  
  
$query = $this->db->query('SELECT * FROM mi_tabla');  
  
echo $query->num_rows();
```

```

$query = $this->db->query('SELECT * FROM mi_tabla');

echo $query->num_fields();

$query = $this->db->query('SELECT title FROM mi_tabla');

foreach ($query->result() as $row)
{
    echo $row->title;
}

$query->free_result(); // El objeto de resultado $query no estará más disponible

$query2 = $this->db->query('SELECT name FROM alguna_tabla');

$row = $query2->row();

echo $row->name;

$query2->free_result(); // El objeto de resultado $query2 no estará más disponible

```

### La Clase Active Record

CodeIgniter usa una versión modificada del Patrón de Base de Datos **Active Record**. Este patrón permite que la información sea obtenida, insertada y actualizada en la base de datos con mínimo código. En algunos casos solamente son necesarias una o dos líneas de código para ejecutar una acción en la base de datos. CodeIgniter no necesita que cada tabla de base de datos tenga su propio archivo de clase. En lugar de eso, provee una interfaz más simplificada.

Más allá de la simplicidad, el mayor beneficio de usar las funcionalidades del **Active Record** es que le permite crear aplicaciones independientes de las bases de datos, ya que cada adaptador de base de datos genera la sintaxis de la consulta. Esto también permite consultas más seguras, ya que el sistema escapa automáticamente los valores.

### Seleccionar Datos

Las siguientes funciones le permiten construir sentencias SELECT de SQL.

#### **\$this->db->get()**

Ejecuta la consulta de selección y devuelve el resultado. Puede utilizarse por sí mismo para recuperar todos los registros de una tabla:

El segundo y tercer parámetros le permiten establecer una cláusula "limit" y "offset":

Advertirá que la función anterior se asigna a una variable llamada `$query`, la cual se usa para mostrar el resultado:

Por favor consulte la página funciones de resultado para ver la discusión completa independientemente de la generación del resultado.

**Nota:** Si tiene intención de escribir sus propias consultas, puede deshabilitar esta clase en su archivo de configuración de la base de datos, permitiéndole al núcleo de la Base de Datos y al adaptador usar pocos recursos.

```
$query = $this->db->get('mi_tabla');  
  
// Produce: SELECT * FROM mi_tabla  
  
$query = $this->db->get('mi_tabla', 10, 20);  
  
// Produce: SELECT * FROM mi_tabla LIMIT 20, 10 (en MySQL. Otras bases de datos  
// pueden tener sintaxis ligeramente diferentes)  
  
$query = $this->db->get('mi_tabla');  
foreach ($query->result() as $row)  
{  
    echo $row->titulo;  
}
```

### `$this->db->get_where()`

Idéntica a la función anterior, excepto que le permite agregar una cláusula "where" en el segundo parámetro, en lugar de usar la función `db->where()`:

```
$query = $this->db->get_where('mi_tabla', array('id' => $id), $limit, $offset);
```

Por favor, leer acerca de la función `where()` abajo para más información.

**Nota:** `get_where()` era conocida antes como `getwhere()`, la cual fue eliminada.

### `$this->db->select()`

Le permite escribir la porción SELECT de una consulta:

```
$this->db->select('titulo, contenido, $fecha');  
  
$query = $this->db->get('mi_tabla');  
  
// Produce: SELECT titulo, contenido, $fecha FROM mi_tabla
```

**Nota:** Si está seleccionando todo (\*) de una tabla no necesita usar esta función. Al omitirse, CodeIgniter asume que desea seleccionar todo (SELECT \*)

`$this->db->select()` acepta un segundo parámetro opcional. Si lo establecer como **FALSE**, CodeIgniter no intentará proteger sus nombres de campo o tabla con backticks. Esto es útil si necesita una sentencia compuesta de selección.

#### `$this->db->select_max()`

Escribe una porción "SELECT MAX(campo)" en su consulta. Opcionalmente puede incluir un segundo parámetro para renombrar el campo de resultado.

#### `$this->db->select_min()`

Escribe una porción "SELECT MIN(campo)" en su consulta. Como con `select_max()`, opcionalmente puede incluir un segundo parámetro para renombrar el campo de resultado.

```
$this->db->select('(SELECT SUM(pagos.cantidad) FROM pagos
    WHERE pagos.factura_id=4) AS cantidad_pagada', FALSE);
```

```
$query = $this->db->get('mi_tabla');
```

```
$this->db->select_max('edad');
```

```
$query = $this->db->get('miembros');
```

```
// Produce: SELECT MAX(edad) AS edad FROM miembros
```

```
$this->db->select_max('edad', 'edad_miembro');
```

```
$query = $this->db->get('miembros');
```

```
// Produce: SELECT MAX(edad) AS edad_miembro FROM miembros
```

```
$this->db->select_min('edad');
```

```
$query = $this->db->get('miembros');
```

```
// Produce: SELECT MIN(edad) AS edad FROM miembros
```

#### `$this->db->select_avg()`

Escribe una porción "SELECT AVG(campo)" en su consulta. Como con `select_max()`, opcionalmente puede incluir un segundo parámetro para renombrar el campo de resultado.



### `$this->db->select_sum()`

Escribe una porción "SELECT SUM(campo)" en su consulta. Como con `select_max()`, opcionalmente puede incluir un segundo parámetro para renombrar el campo de resultado.

### `$this->db->from()`

Le permite escribir la porción FROM de su consulta:

```
$this->db->select_avg('edad');
```

```
$query = $this->db->get('miembros');
```

```
// Produce: SELECT AVG(edad) AS edad FROM miembros
```

```
$this->db->select_sum('edad');
```

```
$query = $this->db->get('miembros');
```

```
// Produce: SELECT SUM(edad) AS edad FROM miembros
```

```
$this->db->select('titulo, contenido, $fecha');
```

```
$this->db->from('mi_tabla');
```

```
$query = $this->db->get();
```

```
// Produce: SELECT titulo, contenido, $fecha FROM mi_tabla
```

**Nota:** Como se mostró antes, la porción FROM de su consulta se puede especificar en la función `$this->db->get()`, por lo que puede usar el método que prefiera.

### `$this->db->join()`

Le permite escribir la porción JOIN de su consulta:

```
$this->db->select('*');
```

```
$this->db->from('blogs');
```

```
$this->db->join('comentarios', 'comentarios.id = blogs.id');
```

```
$query = $this->db->get();
```

```
// Produce:
```

```
// SELECT * FROM blogs
```

```
// JOIN comentarios ON comentarios.id = blogs.id
```

Se pueden hacer varias llamadas de función si necesita varios joins en una consulta.

Si necesita un tipo específico de JOIN puede especificarlo mediante el tercer parámetro de la función. Las opciones

son: left, right, outer, inner, left outer, y right outer.

### `$this->db->where()`

Esta función le permite establecer cláusulas **WHERE** usando uno de los estos cuatro métodos:

**Método simple de clave/valor:** Advierta que se agrega el signo igual por Ud. Si usa varias llamadas de función, se encadenarán todas juntas con **AND** entre ellas:

**Método personalizado de clave/valor:** Puede incluir un operador en el primer parámetro para controlar la comparación:

**Método del array asociativo:** También puede incluir sus propios operadores usando este método:

```
$this->db->join('comentarios', 'comentarios.id = blogs.id', 'left');  
  
// Produce: LEFT JOIN comentarios ON comentarios.id = blogs.id  
  
$this->db->where('nombre', $nombre);  
  
// Produce: WHERE nombre = 'Jose'  
  
$this->db->where('nombre', $nombre);  
  
$this->db->where('titulo', $titulo);  
  
$this->db->where('estado', $status);  
  
// WHERE nombre = 'Jose' AND titulo = 'jefe' AND estado = 'activo'  
  
$this->db->where('nombre !=', $nombre);  
  
$this->db->where('id <', $id);  
  
// Produce: WHERE nombre != 'Jose' AND id < 45  
  
$array = array('nombre' => $nombre, 'titulo' => $titulo, 'estado' =>  
$status);  
  
$this->db->where($array);  
  
// Produce: WHERE nombre = 'Jose' AND titulo = 'jefe' AND estado = 'activo'  
  
$array = array('nombre !=' => $nombre, 'id <' => $id, '$fecha >' => $fecha);  
  
$this->db->where($array);
```

### Cadena personalizada:

Puede escribir sus propias cláusulas manualmente:

`$this->db->where()` acepta un tercer parámetro opcional. Si se lo establece a **FALSE**, CodeIgniter no intentará proteger sus nombres de campos o tabla con backticks.

```
$this->db->where('MATCH (campo) AGAINST ("valor")', NULL, FALSE);
```

### `$this->db->or_where()`

Esta función es idéntica a la anterior, excepto que las instancias múltiples se unen con OR:

**Nota:** `or_where()` antes era conocida como `orwhere()`, la que fue eliminada.

### `$this->db->where_in()`

Genera una consulta SQL WHERE campo IN ('item', 'item') unida mediante AND si corresponde.

### `$this->db->or_where_in()`

Genera una consulta SQL WHERE campo IN ('item', 'item') unida mediante OR si corresponde.

### `$this->db->where_not_in()`

Genera una consulta SQL WHERE campo NOT IN ('item', 'item') unida mediante AND si corresponde.

```
$where = "nombre='Jose' AND estado='jefe' OR estado='activo'";
```

```
$this->db->where($where);
```

```
$this->db->where('nombre !=', $nombre);
```

```
$this->db->or_where('id >', $id);
```

```
// Produce: WHERE nombre != 'Jose' OR id > 50
```

```
$nombres = array('Federico', 'Tomas', 'Juan');
```

```
$this->db->where_in('usuario', $nombres);
```

```
// Produce: WHERE usuario IN ('Federico', 'Tomas', 'Juan')
```

```
$nombres = array('Federico', 'Tomas', 'Juan');
```

```
$this->db->or_where_in('usuario', $nombres);
```

```
// Produce: OR usuario IN ('Federico', 'Tomas', 'Juan')
```

```
$nombres = array('Federico', 'Tomas', 'Juan');  
  
$this->db->where_not_in('usuario', $nombres);  
  
// Produce: WHERE usuario NOT IN ('Federico', 'Tomas', 'Juan')
```

### `$this->db->or_where_not_in()`

Genera una consulta SQL WHERE campo NOT IN ('item', 'item') unida mediante OR si corresponde.

### `$this->db->like()`

Esta función le permite generar cláusulas **LIKE**, útiles para hacer búsquedas. **Nota:** Todos los valores pasados a esta función se escapan automáticamente.

#### 1. Método simple de clave/valor:

Si usa varias llamadas a la función, se encadenarán juntas con **AND** entre ellas:

Si quiere controlar donde se ubica el comodín (%), puede usar un tercer parámetro opcional. Las opciones son 'before', 'after' y 'both' (que es el valor por defecto).

```
$nombres = array('Federico', 'Tomas', 'Juan');  
  
$this->db->or_where_not_in('usuario', $nombres);  
  
// Produce: OR usuario NOT IN ('Federico', 'Tomas', 'Juan')  
  
$this->db->like('titulo', 'match');  
  
// Produce: WHERE titulo LIKE '%match%'  
  
$this->db->like('titulo', 'match');  
  
$this->db->like('cuerpo', 'match');  
  
// WHERE titulo LIKE '%match%' AND cuerpo LIKE '%match%'  
  
$this->db->like('titulo', 'match', 'before');  
  
// Produce: WHERE titulo LIKE '%match'  
  
$this->db->like('titulo', 'match', 'after');  
  
// Produce: WHERE titulo LIKE 'match%'  
  
$this->db->like('titulo', 'match', 'both');  
  
// Produce: WHERE titulo LIKE '%match%'
```

## 2. Método del array asociativo:

```
$array = array('titulo' => $match,  
              'pagina1' => $match,  
              'pagina2' => $match);  
  
$this->db->like($array);  
  
// WHERE titulo LIKE '%match%' AND pagina1 LIKE '%match%' AND pagina2 LIKE  
// '%match%'
```

### `$this->db->or_like()`

Esta función es idéntica a la anterior, excepto que las instancias múltiples se unen mediante OR:

**Nota:** `or_like()` antes era conocida como `orlike()`, la cual fue eliminada.

### `$this->db->not_like()`

Esta función es idéntica a `like()`, excepto que genera sentencias NOT LIKE:

### `$this->db->or_not_like()`

Esta función es idéntica a `not_like()`, excepto que las instancias múltiples se unen mediante OR:

### `$this->db->group_by()`

Le permite escribir la porción GROUP BY de su consulta:

También puede pasarle un array de múltiples valores:

**Nota:** `group_by()` antes era conocida como `groupby()`, la cual fue eliminada.

```
$this->db->like('titulo', 'match');  
  
$this->db->or_like('cuerpo', $match);  
  
// WHERE titulo LIKE '%match%' OR cuerpo LIKE '%match%'
```

```
$this->db->not_like('titulo', 'match');
```

```
// WHERE titulo NOT LIKE '%match%
```

```
$this->db->like('titulo', 'match');
```

```
$this->db->or_not_like('cuerpo', 'match');
```

```
// WHERE titulo LIKE '%match%' OR cuerpo NOT LIKE '%match%'
```

```
$this->db->group_by("titulo");
```

```
// Produce: GROUP BY titulo
```

```
$this->db->group_by(array("titulo", "$fecha"));
```

```
// Produce: GROUP BY titulo, $fecha
```

### **\$this->db->distinct()**

Agrega la palabra clave "DISTINCT" a la consulta.

### **\$this->db->having()**

Le permite escribir la porción HAVING de su consulta. Hay dos sintaxis posibles, uno o dos argumentos:

También puede pasarle un array de múltiples valores:

Si está usando una base de datos para la que CodeIgniter escapa las consultas, puede evitar de escapar el contenido pasando un tercer parámetro opcional, y estableciéndolo a **FALSE**.

### **\$this->db->or\_having()**

Idéntica to **having()**, salvo que separa varias cláusulas mediante "OR".

### `$this->db->order_by()`

Le permite establecer una cláusula ORDER BY. El primer parámetro contiene el nombre de la columna por la que querría ordenar. El segundo parámetro le permite establecer la dirección del resultado. Las opciones son **asc**, **desc**, o **random**.

```
$this->db->distinct();

$this->db->get('tabla');

// Produce: SELECT DISTINCT * FROM tabla

$this->db->having('usuario_id = 45');

// Produce: HAVING usuario_id = 45

$this->db->having('usuario_id', 45);

// Produce: HAVING usuario_id = 45

$this->db->having(array('titulo =' => 'Mi Titulo', 'id <' => $id));

// Produce: HAVING titulo = 'Mi Titulo', id < 45

$this->db->having('usuario_id', 45);

// Produce: HAVING `usuario_id` = 45 en algunas bases de datos como MySQL

$this->db->having('user_id', 45, FALSE);

// Produce: HAVING user_id = 45

$this->db->order_by("titulo", "desc");

// Produce: ORDER BY titulo DESC
```

También puede pasar su propia cadena en el primer parámetro:

O se pueden hacer varias llamadas a la función si necesita varios campos.

### `$this->db->limit()`

Le permite limitar la cantidad de filas que desea que la consulta devuelva:

El segundo parámetro le permite establecer un desplazamiento del resultado.

### `$this->db->count_all_results()`

Le permite determinar la cantidad de filas en una consulta del **Active Record**. Las consultas aceptarán restrictores tales como **where()**, **or\_where()**, **like()**, **or\_like()**, etc. Ejemplo:

```

$this->db->order_by('titulo desc, nombre asc');
// Produce: ORDER BY titulo DESC, nombre ASC
$this->db->order_by("titulo", "desc");
$this->db->order_by("nombre", "asc");
// Produce: ORDER BY titulo DESC, nombre ASC

$this->db->limit(10);
// Produce: LIMIT 10
$this->db->limit(10, 20);
// Produce: LIMIT 20, 10 (en MySQL. Otras bases de datos pueden tener una sintaxis
// ligeramente diferente)

echo $this->db->count_all_results('mi_tabla'); // Produce un entero, como 25
$this->db->like('titulo', 'match'); $this->db->from('mi_tabla');
echo $this->db->count_all_results();
// Produce un entero, como 17

```

### **\$this->db->count\_all()**

Le permite determinar la cantidad de filas en una tabla en particular. Presente el nombre de la tabla como primer parámetro. Ejemplo:

### **Insertar Datos**

#### **\$this->db->insert()**

Genera una cadena insert de SQL basada en los datos que suministra, y ejecuta la consulta. Tanto puede pasar un **array** como un **objeto** a la función. Aquí hay un ejemplo usando un array:

```

echo $this->db->count_all('mi_tabla'); // Produce un entero, como 25

$data = array(

```



```

'titulo' => 'Mi titulo' ,
'nombre' => 'Mi nombre' ,
'$fecha' => 'Mi $fecha'
);
$this->db->insert('mi_tabla', $data);
// Produce: INSERT INTO mi_tabla (titulo, nombre, $fecha) VALUES ('Mi titulo',
// 'Mi nombre', 'Mi $fecha')

```

El primer parámetro contendrá el nombre de la tabla y el segundo un array asociativo de valores. Aquí hay un ejemplo usando un objeto:

```

/*
class MiClase {
}*/
var $titulo = 'Mi Titulo';
var $contenido = 'Mi Contenido';
var $$fecha = 'Mi $fecha';
$object = new MiClase;
$this->db->insert('mi_tabla', $object);
// Produce: INSERT INTO mi_tabla (titulo, contenido, $fecha) VALUES ('Mi Titulo',
// 'Mi Contenido', 'Mi $fecha')

```

El primer parámetro contendrá el nombre de la tabla y el segundo es un objeto.

**Nota:** Todos los valores se escapan automáticamente para producir consultas más seguras.

### `$this->db->insert_batch()`

Genera una cadena insert de SQL basada en los datos provistos, y ejecuta la consulta. A la función se le puede pasar tanto un **array** como un **objeto**. Aquí hay un ejemplo usando un array:

```

$data = array(
    array(
        'titulo' => 'Mi titulo' ,

```

```

    'nombre' => 'Mi nombre' ,
    'fecha' => 'Mi fecha'
), array(
    'titulo' => 'Otro titulo' ,
    'nombre' => 'Otro nombre' ,
    'fecha' => 'Otra fecha'
));

$this->db->insert_batch('mi_tabla', $data);

// Produce: INSERT INTO mi_tabla (titulo, nombre, fecha) VALUES ('Mi titulo',
// 'Mi nombre', 'Mi fecha'), ('Otro titulo', 'Otro nombre', 'Otra fecha')

```

El primer parámetro contendrá el nombre de la tabla y el segundo es un array asociativo de valores.

Esta función le permite establecer valores para **inserciones** o **actualizaciones**. **Se puede usar en lugar de pasar datos directamente a un array para las funciones de inserción o**

#### **actualización:**

Si usa varias llamadas de función, se ensamblarán adecuadamente en función de si usted está haciendo una inserción o una actualización:

```

$this->db->set('nombre', $nombre);

$this->db->insert('mi_tabla');

// Produce: INSERT INTO mi_tabla (name) VALUES ('{$nombre}')

$this->db->set('nombre', $nombre);

$this->db->set('titulo', $titulo);

$this->db->set('estado', $status);

$this->db->insert('mi_tabla');

```

## set()

también aceptará un tercer parámetro opcional (**\$escape**), que evitará que los datos se escapen si se lo

establece a **FALSE**. Para ilustrar la diferencia, aquí se usa **set()** con y sin el parámetro de escape.

También puede pasar un array asociativo a esta función:

O un objeto:

```
$this->db->set('campo', 'campo+1', FALSE);
```

```
$this->db->insert('mi_tabla');
```

```
// Devuelve: INSERT INTO mi_tabla (campo) VALUES (campo+1)
```

```
$this->db->set('campo', 'campo+1');
```

```
$this->db->insert('mi_tabla');
```

```
// Devuelve: INSERT INTO mi_tabla (campo) VALUES ('campo+1')
```

```
$array = array('nombre' => $nombre, 'titulo' => $titulo, 'estado' => $status);
```

```
$this->db->set($array);
```

```
$this->db->insert('mi_tabla');
```

```
/*
```

```
    class Mi_clase {
```

```
    }*/
```

```
var $titulo = 'Mi titulo';
```

```
var $contenido = 'Mi contenido';
```

```
var $fecha = 'Mi fecha';
```

```
$object = new Mi_clase;
```

```
$this->db->set($object);
```

```
$this->db->insert('mi_tabla');
```

## Actualizar Datos

### `$this->db->update()`

Genera una cadena update de SQL y ejecuta la consulta basada en los datos provistos. A la función puede pasarle un **array** o un **objeto**. Aquí hay un ejemplo usando un array:

```
$data = array(
    'titulo' => $titulo,
    'nombre' => $nombre,
    'fecha' => $fecha
);
$this->db->where('id', $id);
$this->db->update('mi_tabla', $data);
// Produce:
// UPDATE mi_tabla
// SET title = '{$titulo}', name = '{$nombre}', date = '{$fecha}'
// WHERE id = $id
```

O puede proporcionar un objeto:

```
/*
class Mi_clase {
    var $titulo = 'Mi titulo';
    var $nombre = 'Mi nombre';
    var $fecha = 'Mi fecha';
}*/
$object = new Mi_clase;
$this->db->where('id', $id);
$this->db->update('mi_tabla', $object);
// Produce:
```

```
// UPDATE mi_tabla

// SET title = '{$titulo}', name = '{$nombre}', date = '{$fecha}'

// WHERE id = $id
```

**Nota:** Todos los valores pasados a esta función se escapan, produciendo consultas más seguras. Advertirá el uso de la función `$this->db->where()`, permitiéndole establecer la cláusula WHERE. Opcionalmente

puede pasar esta información directamente a la función `update()` como una cadena: `$this->db->update('mi_tabla', $data, "id = 4");`

O como un array:

```
$this->db->update('mi_tabla', $data, array('id' => $id));
```

También puede usar la función `$this->db->set()` descrita antes cuando se realicen actualizaciones.

## Borrar Datos

### `$this->db->delete()`

Genera una cadena delete de SQL y ejecuta la consulta.

El primer parámetro es el nombre de la tabla y el segundo la cláusula where. También puede usar las funciones `where()` u `or_where()` en lugar de pasarle los datos al segundo parámetro de la función:

```
$this->db->delete('mi_tabla', array('id' => $id));
```

```
// Produce:
```

```
// DELETE FROM mi_tabla
```

```
// WHERE id = $id
```

```
$this->db->where('id', $id);
```

```
$this->db->delete('mi_tabla');
```

```
// Produce:
```

```
// DELETE FROM mi_tabla
```

```
// WHERE id = $id
```

Si quiere borrar más de una tabla, se puede pasar un array de nombres de tablas a `delete()`.

Si quiere borrar todos los datos de una tabla, puede usar las funciones `truncate()` o `empty_table()`.

#### `$this->db->empty_table()`

Genera una cadena delete de SQL y ejecuta la consulta.

#### `$this->db->truncate()`

Genera una cadena truncate de SQL y ejecuta la consulta.

**Nota:** Si el comando TRUNCATE no está disponible, `truncate()` se ejecutará como "DELETE FROM table".

#### **Método de Encadenamiento**

El Método de Encadenamiento le permite simplificar la sintaxis conectando varias funciones. Considere este ejemplo:

### 9.3.3 Paypal

Para este proyecto se ha utilizado el método de pago mediante Paypal.

Para controlar el uso del pago se ha utilizado la librería Paypal.php que controla la configuración de intercambio de información con la página de Sandbox de paypal, a continuación mostramos la librería utilizada:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

/**
 * CodeIgniter
 *
 * An open source application development framework for PHP 5.1.6 or newer
 *
 * @package   CodeIgniter
 * @author    Romaldy Minaya
 * @copyright Copyright (c) 2011, PROTOS.
 * @license   GLP
 * @since     Version 1.0
 * @version  1.0
 */

//-----

/**
 * Paypal Class
 *
 * @package   CodeIgniter
 * @subpackage Libraries
 * @category  Payment process
 * @author    Romaldy Minaya
```

\*

//-----

## Documentation

This class let you make the payment proces based on paypal API, effortless and easy.

\*1)Use the same documentation about the vars from paypal page. <http://bit.ly/j4wRR>

\*2)Customize the payment proces as you desire.

\*3)Build with love.

## Implementation

\*1)Copy this code in your controller's function

```
$config['business']                = 'demo@demo.com';  
$config['cpp_header_image']       = ""; //Image header url [750 pixels wide by 90 pixels high]  
$config['return']                 = 'sucess.php';  
$config['cancel_return']          = 'shopping.php';  
$config['notify_url']             = 'process_payment.php'; //IPN Post  
$config['production']             = TRUE; //Its false by default and will use sandbox  
$config['discount_rate_cart']     = 20; //This means 20% discount  
$config["invoice"]               = '843843'; //The invoice id  
  
$this->load->library('paypal',$config);
```



```
#$this->paypal->add(<name>,<price>,<quantity>[Default 1],<code>[Optional]);
```

```
$this->paypal->add('T-shirt',2.99,6); //First item
```

```
$this->paypal->add('Pants',40); //Second item
```

```
$this->paypal->add('Blowse',10,10,'B-199-26'); //Third item with code
```

```
$this->paypal->pay(); //Process the payment
```

The notify url is where paypal will POST the information of the payment so you can save that POST directly into your DB and analyze as you want.

With `$config["invoice"]` is how you identify a bill and you can compare,save or update that value later on your DB.

For test purposes i do recommend to save the entire POST into your DB and analyze if its working according to your needs before putting it in production mode. EX.

```
$received_post = print_r($this->input->post(),TRUE);
```

```
//Save that variable and analyze.
```

Note: html reference page <http://bit.ly/j4wRR>

```
*/
```

```
class Paypal {
```

```
    var $config          = Array();
```

```
    var $production_url = 'https://www.paypal.com/cgi-bin/webscr?';
```

```
    var $sandbox_url    = 'https://www.sandbox.paypal.com/cgi-bin/webscr?';
```

```

var $item          = 1;

/**
 * Constructor
 *
 * @param string
 * @return void
 */
public function __construct($props = array())
{
    $this->__initialize($props);
    log_message('debug', "Paypal Class Initialized");
}

// -----

/**
 * initialize Paypal preferences
 *
 * @access public
 * @param array
 * @return bool
 */
function __initialize($props = array())
{
    #Account information
    $config["business"] = ""; //Account email or id
    $config["cmd"]      = '_cart'; //Do not modify

```

```

$config["production"]    = FALSE;

#Custom variable here we send the billing code-->

$config["custom"] = "";

$config["invoice"] = ""; //Code to identify the bill

#API Configuration-->

$config["upload"]      = '1'; //Do not modify

$config["currency_code"] = 'EUR'; //http://bit.ly/anciiH

$config["disp_tot"] = 'Y';

#Page Layout -->

$config["cpp_header_image"]      = ""; //Image header url [750 pixels wide by 90
pixels high]

$config["cpp_cart_border_color"] = '000'; //The HTML hex code for your principal
identifying color

$config["no_note"]      = 1; //[0,1] 0 show, 1 hide

#Payment Page Information -->

$config["return"]      = ""; //The URL to which PayPal redirects
buyers' browser after they complete their payments.

$config["cancel_return"] = ""; //Specify a URL on your website that displays a
"Payment Canceled" page.

$config["notify_url"]   = base_url() . 'paypal/pay/notify_payment'; //The URL
to which PayPal posts information about the payment (IPN)

$config["rm"] = '2'; //Leave this to get payment information

$config["lc"] = 'ES'; //Languaje [EN,ES]

#Shipping and Misc Information -->

```

```

$config["shipping"]      = "";
$config["shipping2"]    = "";
$config["handling"]     = "";
$config["tax"]           = "";
$config["discount_amount_cart"] = ""; //Discount amount [9.99]
$config["discount_rate_cart"]   = ""; //Discount percentage [15]

```

```

#Customer Information -->

```

```

$config["first_name"]    = "";
$config["last_name"]     = "";
$config["address1"]     = "";
$config["address2"]     = "";
$config["city"]         = "";
$config["state"]        = "";
$config["zip"]          = "";
$config["email"]        = "";
$config["night_phone_a"] = "";
$config["night_phone_b"] = "";
$config["night_phone_c"] = "";

```

```

/*
 * Convert array elements into class variables
 */
if (count($props) > 0)
{
    foreach ($props as $key => $val)
    {

```

```

        $config[$key] = $val;
    }
}
$this->config = $config;
}

// -----

/**
 * Perform payment process
 *
 * @access public
 * @param array
 * @return void
 */
function pay(){

    #Convert the array to url encode variables
    $vars = http_build_query($this->config);

    if($this->config['production'] == TRUE){
        header('LOCATION:'.$this->production_url.$vars);
    }else{
        header('LOCATION:'.$this->sandbox_url.$vars);
    }
}
}

```

```

// -----

/**
 * Add a product to the list
 *
 * @access public
 * @param array
 * @return void
 */

function add($item_name = "", $item_amount = NULL, $item_qty = NULL, $item_number =
NULL){
    $this->config['item_name_'. $this->item] = $item_name;
    $this->config['amount_'. $this->item] = $item_amount;
    $this->config['quantity_'. $this->item] = $item_qty;
    $this->config['item_number_'. $this->item] = $item_number;
    $this->item++;
}
}

// END Paypal Class

/* End of file Paypal.php */

/* Location: ./application/libraries/ Paypal.php */

```

Para generar el pedido del producto, se ha creado el controlador Paypal.php que se muestra a continuación:

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

session_start();

class Pay extends CI_Controller

{

function __construct()

{

parent::__construct();

$this->load->helper('string');

}

public function payment($id)

{

$this->load->model('reserve_model', 'rmm');

$this->load->model('offer_model', 'omm');

if ($this->session->userdata('logged_in')) {

    $session_data = $this->session->userdata('logged_in');

    $reserve = $this->rmm->getReserveId($id);

    $config['business'] = 'yerapv-facilitator@gmail.com';

    $config['cpp_header_image'] = base_url() . 'assets/img/logo.png'; //Image header url
[750 pixels wide by 90 pixels high]

    $config['return'] = base_url() . 'paypal/pay/notify_payment';

    $config['cancel_return'] = '';

    $config['notify_url'] = base_url() . 'paypal/pay/notify_payment'; //IPN Post

    $config['production'] = FALSE; //Its false by default and will use sandbox

    $config['discount_rate_cart'] = 0; //This means 20% discount

    // $config["invoice"] = random_string('numeric', 8); //The invoice id

    foreach ($reserve as $info) {
```

```

        $config["invoice"] = $info->ticket;
    }
    $this->load->library('paypal', $config);
    foreach ($reserve as $info) {
        $data['iditem'] = $info->iditem;
        $data['idweb'] = $info->idweb;
        $product = $this->omm->getOfferItemIds($data);
        foreach ($product as $infopro) {
            $this->paypal->add($infopro->name, floatval($info->idprize), 1); //First item
        }
    }
    // $this->paypal->add('Pants', 40); //Second item
    // $this->paypal->add('Blowse', 10, 10, 'B-199-26'); //Third item with code
    $this->paypal->pay(); //Process the payment
}
}
public function notify_payment()
{
    $i = 0;
    $ipn = $this->input->post();
    foreach ($this->input->post() as $key => $value) {
        if ($i == 1) {
            $invoice = $value;
        }
        $i++;
    }
    if ($invoice == NULL || $invoice == "") {

```



```

        $this->cancelview();
    } else {
        $report = print_r($ipn, TRUE);
        $this->load->model('reserve_model', 'rmm');
        $this->load->model('purchases_model', 'pmm');
        $this->load->helper('file');
        $reserve = $this->rmm->getReserveTicket($invoice);
        if ($reserve) {
            foreach ($reserve as $info) {
                $data['iditem'] = $info->iditem;
                $data['iduser'] = $info->iduser;
                $data['money'] = $info->idprize;
                $data['ticket'] = $info->ticket;
                $data['idweb'] = $info->idweb;
                $data['report'] = $report;
            }
            $this->pmm->insertPurchase($data);
            $this->rmm->setPurchaseReserveld($invoice);
        }
        $this->view();
    }
}

function view(){
    if ($this->session->userdata('logged_in')) {
        $this->load->model('profile_model', 'uum');
        $session_data = $this->session->userdata('logged_in');
        $menu['menu'] = array("Principal", "");
    }
}

```

```

$menu['url'] = array("/web/index.php/home", "");

$data['title'] = "Gestión de Perfil";

$data['css'] = "";

$data['idweb'] = $session_data['idweb'];

$data['profile_list'] = $this->uum->getProfileList($data);

$data['avatar'] = $session_data['avatar'];

$data['role'] = $session_data['role'];

$this->load->view('admin/head_view', $data);

$this->load->view('admin/header_view', $data);

$this->load->view('admin/menuleftadmin_view');

$this->load->view('admin/menuhorizontal_view', $menu);

$this->load->view('admin/payment_view', $data);

$this->load->view('admin/footer_view', $data);
}
}
function cancelview()
{
if ($this->session->userdata('logged_in')) {
    $this->load->model('profile_model', 'uum');

    $session_data = $this->session->userdata('logged_in');

    $menu['menu'] = array("Principal", "Gestión de Perfil");

    $menu['url'] = array("/web/index.php/home", "/web/index.php/profile");

    $data['title'] = "Gestión de Perfil";

    $data['css'] = "";

    $data['idweb'] = $session_data['idweb'];

    $data['profile_list'] = $this->uum->getProfileList($data);

    $data['avatar'] = $session_data['avatar'];

```

```

$data['role'] = $session_data['role'];

$this->load->view('admin/head_view', $data);

$this->load->view('admin/header_view', $data);

$this->load->view('admin/menuleftadmin_view');

$this->load->view('admin/menuhorizontal_view', $menu);

$this->load->view('admin/cancelpayment_view', $data);

$this->load->view('admin/footer_view', $data);

}

}

}

?>

```

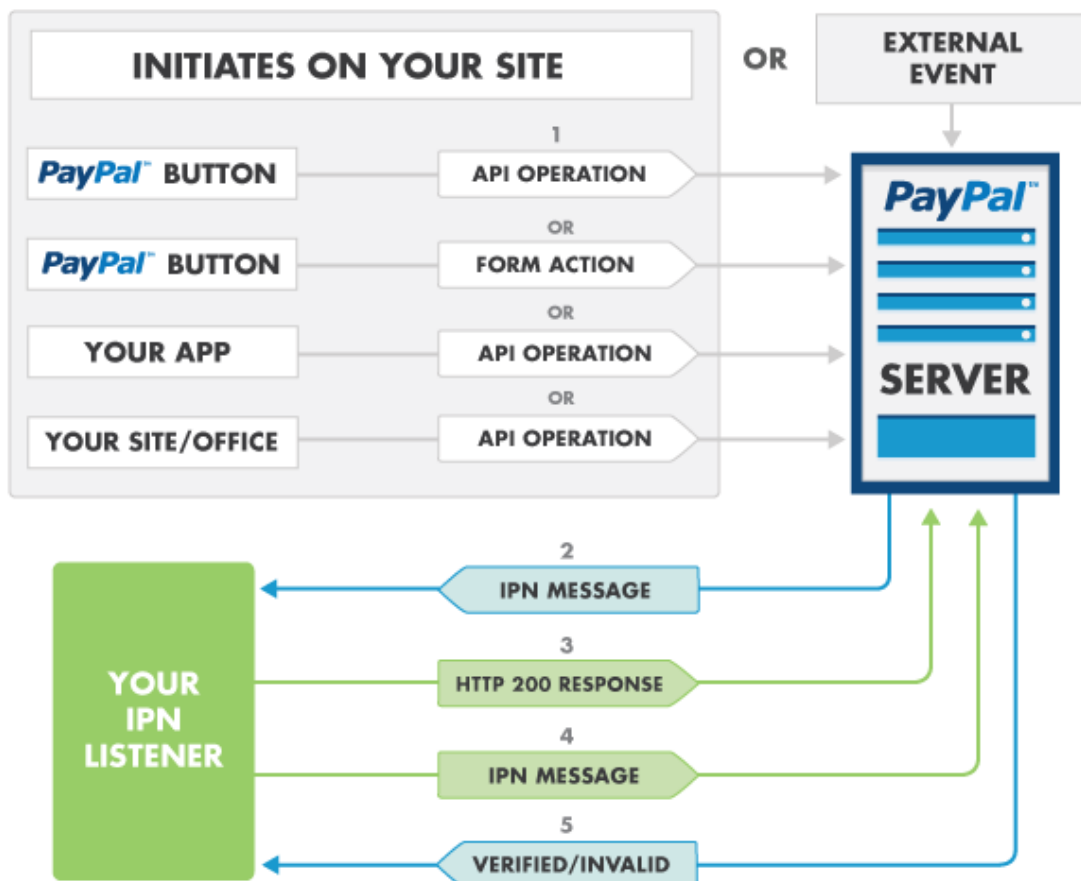


Imagen 33: Funcionamiento API de Paypal

Para realizar el pago, la aplicación web envía el identificador de producto al controlador paypal.php, seguidamente el controlador se conecta a la base de datos y obtiene el código de reserva del producto, genera un array de datos del producto y envía los datos a la web de sandbox de Paypal. El usuario realiza la compra del producto y el controlador IPN de Paypal envía un email de confirmación y un array de confirmación. El controlador Paypal.php recoge los datos con la función notify\_payment() (Véase Imagen 33).

### 9.3.4 Grocery Crud



Imagen 34: Grocery Crud

Grocery Crud es un librería (Véase Imagen 34) en Php para generar CRUD estables en pocos minutos. Lo único que se necesita es configurar el acceso a la base de datos, el archivo de configuración para la conexión se encuentra en application/config/database.php.

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/**
 * CodeIgniter
 *
 * An open source application development framework for PHP 5.1.6 or newer
 * ... mpla mpla mpla
 */

$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = '';
$db['default']['password'] = '';
$db['default']['database'] = '';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
```

```

$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = "";
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = "";
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
$db['default']['failover'] = array();

/* End of file database.php */
/* Location: ./application/config/database.php */

```

Asegurate que has introducido los parametros de tu base de datos correctamente por ejemplo:

```

$db['default']['hostname'] = 'HOST';
$db['default']['username'] = 'USUARIO';
$db['default']['password'] = 'CONTRASEÑA';
$db['default']['database'] = 'NOMBRE_DE_LA_BASE_DE_DATOS';

```

Una vez configurado el acceso a la base de datos, se deben crear las tablas para guardar los datos de la aplicación. Instalamos la librería Grocery\_Crud.php en application/libraries/Grocery\_CRUD.php. Seguidamente mostramos un ejemplo de uso sencillo de la tabla “customers”(Véase Imagen 35).

```

function full_example()
{
    $crud = new grocery_CRUD();

    $crud->set_table('customers')
        ->set_subject('Customer')
        ->columns('customerName','contactLastName','phone','city','country','creditLimit')
        ->display_as('customerName','Name')
        ->display_as('contactLastName','Last Name');

    $crud->fields('customerName','contactLastName','phone','city','country','creditLimit');
    $crud->required_fields('customerName','contactLastName');

    $output = $crud->render();

    $this->_example_output($output);
}

```





















Name	Last Name	Phone	City	Country	from Employee	CreditLimit	Tools
Atelier graphique	Schmitt	40.32.2555	Nantes	France	Hernandez	21000	 
Signal Gift Stores	King	7025551838	Las Vegas	USA	Thompson	71800	 
Australian Collectors, Co.	Ferguson	03 9520 4555	Melbourne	Australia	Fixter	117300	 
La Rochelle Gifts	Labrune	40.67.8555	Nantes	France	Hernandez	118200	 
Baane Mini Imports	Bergulfsen	07-98 9555	Stavern	Norway	Jones	81700	 
Mini Gifts Distributors Ltd.	Nelson	4155551450	San Rafael	USA	Jennings	210500	 
Havel & Zbyszek Co	Piestrzeniewicz	(26) 642-7555	Warszawa	Poland			 
Blauer See Auto, Co.	Keitel	+49 69 66 90 2555	Frankfurt	Germany	Jones	59700	 
Mini Wheels Co.	Murphy	650555787	San Francisco	USA	Jennings	64600	 
Land of Toys Inc.	Lee	2125557818	NYC	USA	Vanauf	114900	 

Imagen 35: Tabla Grocery Crud

Para el proyecto hemos usado diferentes funciones de la API de Grocery Crud.

### Set Validation Rules:

Reglas de las tablas.

```
function example_set_rules() {
    $crud = new grocery_CRUD();

    $crud->set_table('products');
    $crud->columns('productName', 'buyPrice');

    $crud->set_rules('buyPrice', 'buy Price', 'numeric');
    $crud->set_rules('quantityInStock', 'Quantity In
Stock', 'integer');

    $output = $crud->render();

    $this->_example_output($output);
}
```

### Set a Relation:

Relaciones entre tablas de bases de datos.

```
function employees_management()
{
    $crud = new grocery_CRUD();
    $crud->set_theme('datatables');
    $crud->set_table('employees');
    $crud->display_as('officeCode', 'Office City');
    $crud->set_subject('Employee');

    $crud->set_relation('officeCode', 'offices', 'city');

    $output = $crud->render();

    $this->_example_output($output);
}
```

```
}
```

### Set upload file:

Para cargar archivos en la web.

```
function employees_management()
{
    $crud = new grocery_CRUD();

    $crud->set_theme('datatables');
    $crud->set_table('employees');
    $crud->set_relation('officeCode','offices','city');
    $crud->display_as('officeCode','Office City');
    $crud->set_subject('Employee');

    $crud->required_fields('lastName');

    $crud->set_field_upload('file_url','assets/uploads/files');

    $output = $crud->render();

    $this->_example_output($output);
}
```

A continuación se muestra el funcionamiento de upload de forma visual en Grocery Crud (Véase Imagen 36):

### Paso 1

Office City	File url	JobTitle
Paris	pdftest.pdf	Sale Manager (EMEA)
Paris		Sales Rep
London		Sales Rep
San Francisco		Sales Manager (NA)
Paris		Sales Rep
San Francisco		VP Marketing

### Paso 2

gponaur@classicmodelcars.com

Paris **quick ajax delete**

pdftest.pdf delete

Sale Manager (EMEA)

Cancel

### Paso 3

**ajax uploading**

Select Office City

File url : Upload a file

JobTitle :

Update changes Cancel **same button interface for all the browsers**

### Paso 4

Email : **Uploading with progress status and cancelation**

Office City : Paris

File url : Amazon\_Kindle.jpg 90% from 0.1MB Cancel

JobTitle : Sales Rep

Update Changes Cancel

### Paso 5

x2311 **Drop the files from your file manager, without pressing the button browse.**

lbott@classicmodelcars.com

London

Drop files here to upload

Sales Rep

Cancel

Imagen 36: Procedimiento cargar imagen en Grocery Crud



## Set a Relation-n-n:

Relaciones entre multiples tablas (Véase Imagen 37).

```
function film_management ()
{
    $crud = new grocery_CRUD();

    $crud->set_table('film');
    $crud->set_relation_n_n('actors', 'film_actor', 'actor',
    'film_id', 'actor_id', 'fullname','priority');
    $crud->set_relation_n_n('category', 'film_category', 'category',
    'film_id', 'category_id', 'name');

    $crud->unset_columns('description','special_features','last_update');
    $crud->fields('title', 'description', 'actors', 'category',
    'release_year', 'rental_duration', 'rental_rate', 'length',
    'replacement_cost', 'rating', 'special_features');

    $output = $crud->render();

    $this->_example_output($output);
}
```

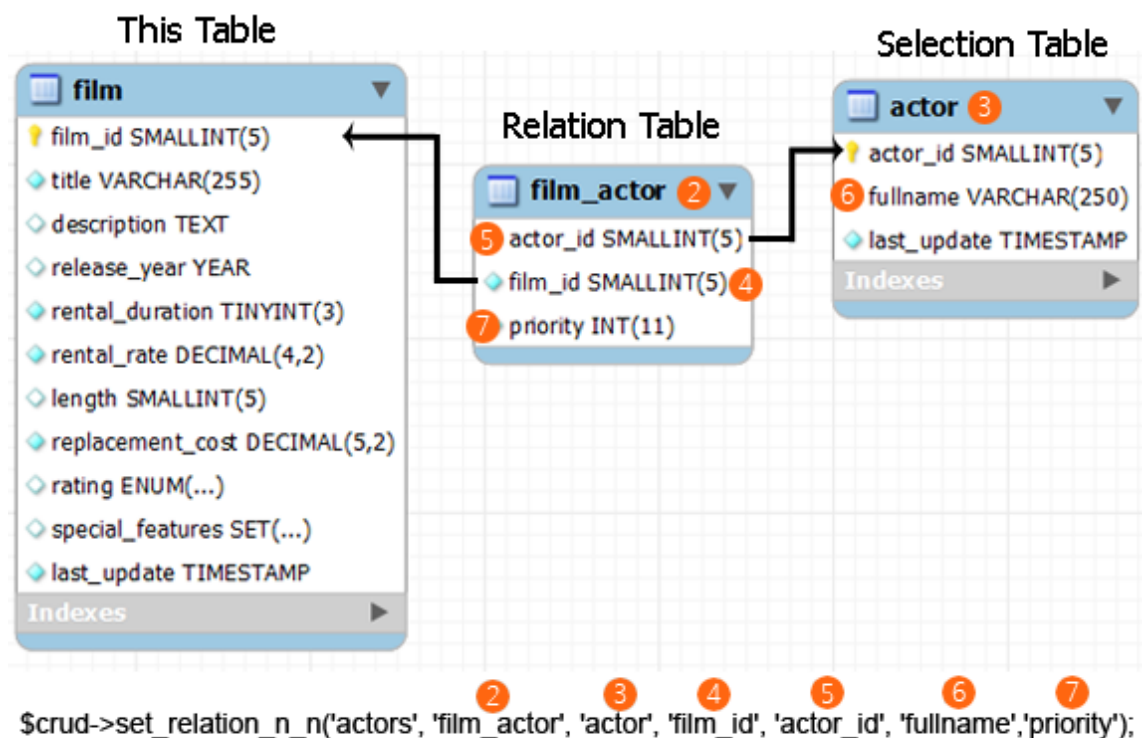


Imagen 37: Relación de multiples tablas en Grocery Crud

## Add a custom action control:

Añadir funciones extras a las tablas.

```
function offices_management_with_actions()
{
    $crud = new grocery_CRUD();

    $crud->set_theme('datatables');
    $crud->set_table('offices');
    $crud->set_subject('Office');
    $crud->required_fields('city');
    $crud->columns('city','country','phone');

    $crud->add_action('More', '', 'demo/action_more','ui-icon-plus');
    $crud->add_action('Photos', '', '', 'ui-icon-image', array($this, 'just_a_test'));
    $crud->add_action('Smileys',
'http://www.grocerycrud.com/assets/uploads/general/smiley.png',
'demo/action_smiley');

    $output = $crud->render();

    $this->_example_output($output);
}

function just_a_test($primary_key , $row)
{
    return site_url('demo/action/action_photos').'?country='.$row->country;
}

function action_photos()
{
    //Just a quick filtering of $_GET
    $country = preg_replace("/([\^a-zA-Z0-9\.\-\_\_]+?)\{1}/i", ' ',
$_GET['country']);

    echo "<div style='font-size:16px;font-family:Arial'>";
    echo "Just a test function for photos button and country:
<b>".$country."</b><br/>";
    echo "<a
href='".site_url('demo/offices_management_with_actions')." '>Go back to
example</a>";
    echo "</div>";
    die();
}

function action_more($id)
{
    echo "<div style='font-size:16px;font-family:Arial'>";
    echo "Just a test function for more button and id:
<b>".(int)$id."</b><br/>";
    echo "<a
href='".site_url('demo/offices_management_with_actions')." '>Go back to
example</a>";
    echo "</div>";
    die();
}
```

```

function action_smiley($id)
{
    echo "<div style='font-size:16px;font-family:Arial'>";
    echo "Just a test function for action button smiley and id:
<b>".(int)$id."</b><br/>";
    echo "<a
href='".site_url('demo/offices_management_with_actions')."'>Go back to
example</a>";
    echo "</div>";
    die();
}

```

### Callback\_add\_filed:

Llamadas a funciones antes de añadir datos.

```

function example_callback_add_field(){
    $crud = new grocery_CRUD();

    $crud->set_table('offices');
    $crud->set_subject('Office');
    $crud->required_fields('city');
    $crud-
>columns('city','country','phone','addressLine1','postalCode');

    $crud-
>callback_add_field('phone',array($this,'add_field_callback_1'));
    $crud-
>callback_add_field('state',array($this,'add_field_callback_2'));

    $output = $crud->render();

    $this->_example_output($output);
}

function add_field_callback_1()
{
    return '+30 <input type="text" maxlength="50" value=""
name="phone" style="width:462px">';
}

function add_field_callback_2()
{
    return '<input type="text" maxlength="50" value="" name="state"
style="width:400px"> ( for U.S. only )';
}

```

### Callback\_edit\_fields:

Llamadas a funciones antes de editar datos.

```

function example_callback_edit_field(){
    $crud = new grocery_CRUD();

    $crud->set_table('offices');
    $crud->set_subject('Office');
    $crud->required_fields('city');
}

```

```

    $crud-
>columns('city','country','phone','addressLine1','postalCode');

    $crud-
>callback_edit_field('phone',array($this,'edit_field_callback_1'));

    $output = $crud->render();

    $this->_example_output($output);
}

function edit_field_callback_1($value, $primary_key)
{
    return '+30 <input type="text" maxlength="50" value="'.$value.'"
name="phone" style="width:462px">';
}

```

### Callback\_column:

Edición de información en las columnas de las tablas.

```

function example_callback_column(){
    $crud = new grocery_CRUD();

    $crud->set_table('products');
    $crud->set_subject('Product');
    $crud->unset_columns('productDescription','productScale');
    $crud->callback_column('buyPrice',array($this,'valueToEuro'));
    $crud-
>callback_column('quantityInStock',array($this,'product_scale_callback
'));

    $output = $crud->render();

    $this->_example_output($output);
}

function valueToEuro($value, $row)
{
    return $value.' &euro;';
}

function product_scale_callback($value, $row)
{
    return $value." - scale: <b>".$row->productScale."</b>";
}

```

### Callback\_before\_insert:

Procesar datos antes de insertar en la tabla.

```

function example_callback_before_insert(){

    $crud = new grocery_CRUD();

    $crud->set_table('offices');
    $crud->set_subject('Office');
    $crud->required_fields('city');
}

```

```

    $crud-
>columns('city','country','phone','addressLine1','postalCode');
    $crud->callback_before_insert(array($this,'checking_post_code'));

    $output = $crud->render();

    $this->_example_output($output);
}

function checking_post_code($post_array)
{
    if(empty($post_array['postalCode']))
    {
        $post_array['postalCode'] = 'Not U.S.';
    }
    return $post_array;
}

```

### Add\_Fields:

Añadir etiquetas nuevas en las tablas.

```

function customers_example() {
    $crud = new grocery_CRUD();

    $crud->set_table('customers')
        ->set_subject('Customer')
        ->columns('customerName','contactLastName','creditLimit');

    $crud-
>add_fields('customerName','contactLastName','city','creditLimit');
    $crud->edit_fields('customerName','contactLastName','city');

    $crud->required_fields('customerName','contactLastName');

    $output = $crud->render();

    $this->_example_output($output);
}

```

### Callback\_Add\_fields:

Añadir datos en formularios antes de añadir información.

```

function example_callback_add_field(){
    $crud = new grocery_CRUD();

    $crud->set_table('offices');
    $crud->set_subject('Office');
    $crud->required_fields('city');
    $crud-
>columns('city','country','phone','addressLine1','postalCode');

    $crud-
>callback_add_field('phone',array($this,'add_field_callback_1'));
    $crud-
>callback_add_field('state',array($this,'add_field_callback_2'));

    $output = $crud->render();
}

```

```

        $this->_example_output($output);
    }

    function add_field_callback_1()
    {
        return '+30 <input type="text" maxlength="50" value=""
name="phone" style="width:462px">';
    }

    function add_field_callback_2()
    {
        return '<input type="text" maxlength="50" value="" name="state"
style="width:400px"> ( for U.S. only )';
    }

```

### Callback\_After\_Delete:

Añadir function de procesamiento que se ejecuta después de eliminar datos.

```

public function user() {
    $crud = new grocery_CRUD();

    $crud->set_table('cms_user');
    $crud->set_subject('User List');
    $crud->required_fields('username');

    $crud->columns('username','email','real_name','active');
    $crud->change_field_type('active', 'true_false');

    $crud->callback_after_delete(array($this,'user_after_delete'));

    $output = $crud->render();

    $this->_example_output($output);
}

public function user_after_delete($primary_key)
{
    return $this->db->insert('user_logs',array('user_id' =>
$primary_key,'action'=>'delete', 'updated' => date('Y-m-d H:i:s')));
}

```

### Callback\_After\_Insert:

Añadir function de procesamiento de datos después de insertar.

```

public function users() {
    $crud = new grocery_CRUD();

    $crud->set_table('users');
    $crud->set_subject('User');
    $crud->required_fields('username');

    $crud->columns('username', 'email', 'real_name', 'active');
    $crud->fields('username', 'email', 'password', 'real_name',
'active');
}

```

```

        $crud->callback_after_insert(array($this,
'log_user_after_insert'));
        $crud->callback_after_update(array($this,
'log_user_after_update'));

        $output = $crud->render();

        $this->_example_output($output);
    }

function log_user_after_insert($post_array,$primary_key)
{
    $user_logs_insert = array(
        "user_id" => $primary_key,
        "created" => date('Y-m-d H:i:s'),
        "last_update" => date('Y-m-d H:i:s')
    );

    $this->db->insert('user_logs',$user_logs_insert);

    return true;
}

```

### Callback\_After\_update:

Procesar datos antes de actualizar.

```

public function users() {
    $crud = new grocery_CRUD();

    $crud->set_table('users');
    $crud->set_subject('User');
    $crud->required_fields('username');

    $crud->columns('username', 'email', 'real_name', 'active');
    $crud->fields('username', 'email', 'password', 'real_name',
'active');

    $crud->callback_after_insert(array($this,
'log_user_after_insert'));
    $crud->callback_after_update(array($this,
'log_user_after_update'));

    $output = $crud->render();

    $this->_example_output($output);
}

function log_user_after_update($post_array,$primary_key)
{
    $user_logs_update = array(
        "user_id" => $primary_key,
        "last_update" => date('Y-m-d H:i:s')
    );

    $this->db->update('user_logs',$user_logs_update,array('user_id' =>
$primary_key));

    return true;
}

```

Uso de inclusiones de bases de datos:

### Where

```
public function webpages ()
{
    $crud = new grocery_CRUD();

    $crud->where('status','active');

    $crud->set_table('webpages');
    $crud->order_by('priority');
    $crud->set_subject('Webpage');
    $crud->columns('menu_title','url','status','priority');
    $crud->change_field_type('status','hidden','active');

    $output = $crud->render();
    $this->_view_output($output);
}
```

### Order By

```
function customers_example() {
    $crud = new grocery_CRUD();

    $crud->set_table('customers')
        ->set_subject('Customer')
        ->columns('customerName','contactLastName','creditLimit');

    $crud->fields('customerName','contactLastName','city','creditLimit');

    $crud->order_by('creditLimit','desc');

    $output = $crud->render();

    $this->_example_output($output);
}
```

### Or Like

```
function example_with_or_like() {

    $crud = new grocery_CRUD();

    $crud->like('productName','Moto');
    $crud->or_like('productName','Car');
    $crud->or_like('productName','Bicycle');

    $crud->set_table('products');
    $crud->columns('productName','buyPrice');

    $output = $crud->render();

    $this->_example_output($output);
}
```



## Or where

```
function example_with_or_where() {  
  
    $crud = new grocery_CRUD();  
  
    $crud->where('productName', 'Motorcycle');  
    $crud->or_where('productName', 'Car');  
    $crud->or_where('productName', 'Bicycle');  
  
    $crud->set_table('products');  
    $crud->columns('productName', 'buyPrice');  
  
    $output = $crud->render();  
  
    $this->_example_output($output);  
}
```

## Like

```
function example_with_like() {  
  
    $crud = new grocery_CRUD();  
  
    $crud->like('productName', 'Motor');  
  
    $crud->set_table('products');  
    $crud->columns('productName', 'buyPrice');  
  
    $output = $crud->render();  
  
    $this->_example_output($output);  
}
```

## Set\_Theme

Cambiar temas visuales en las tablas.

```
function offices_management()  
{  
    $crud = new grocery_CRUD();  
  
    $crud->set_theme('datatables');  
  
    $crud->set_table('offices');  
    $crud->set_subject('Office');  
    $crud->required_fields('city');  
    $crud->  
>columns('city', 'country', 'phone', 'addressLine1', 'postalCode');  
  
    $output = $crud->render();  
  
    $this->_example_output($output);  
}
```

### 9.3.5 Image Crud

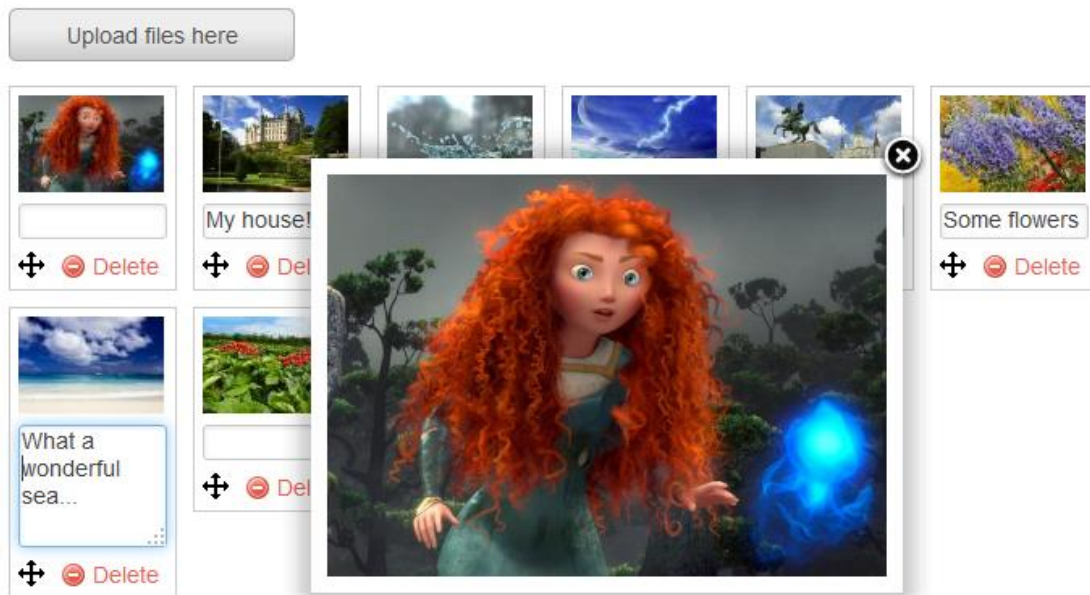


Imagen 38: Image Crud

Image Crud es un cargador automático de galerías de fotos, que sigue la misma filosofía que Grocery Crud (Véase Imagen 38).

Para cargar la galería de imágenes primero se debe crear la tabla en la base de datos.

```
CREATE TABLE IF NOT EXISTS `images` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `url` varchar(250) DEFAULT NULL,  
  `offer_id` int(11) DEFAULT NULL,  
  Category_id int(11) DEFAULT NULL,  
  `priority` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
);
```

```

function example1()
{
    $image_crud = new image_CRUD();

    $image_crud->set_table('example_1');

    //If your table have by default the "id" field name as a primary
    key this line is not required
    $image_crud->set_primary_key_field('id');

    $image_crud->set_url_field('url');
    $image_crud->set_image_path('assets/uploads');

    $output = $image_crud->render();

    $this->_example_output($output);
}

```

### Galeria de fotos por orden de prioridad

```

function example2()
{
    $image_crud = new image_CRUD();

    $image_crud->set_table('example_2');
    $image_crud->set_primary_key_field('id');
    $image_crud->set_url_field('url');
    $image_crud->set_ordering_field('priority')
    ->set_image_path('assets/uploads');

    $output = $image_crud->render();

    $this->_example_output($output);
}

```

### Galeria de fotos por categorías

```

function example3()
{
    $image_crud = new image_CRUD();

    $image_crud->set_table('example_3');

    $image_crud->set_primary_key_field('id');
    $image_crud->set_url_field('url');

    $image_crud->set_relation_field('category_id')
    ->set_ordering_field('priority')
    ->set_image_path('assets/uploads');

    $output = $image_crud->render();

    $this->_example_output($output);
}

```

## Galería de fotos con edición de títulos

```
function example4()
{
    $image_crud = new image_CRUD();

    $image_crud->set_table('example_2');

    $image_crud->set_primary_key_field('image_id');
    $image_crud->set_url_field('url');
    $image_crud->set_title_field('title');
    $image_crud->set_ordering_field('priority')
        ->set_image_path('assets/uploads');

    $output = $image_crud->render();

    $this->_example_output($output);
}
```

### 9.3.6 Image Moo

Image Moo es una librería de manipulación de imágenes. Para poder manipular las imágenes deberemos instalar la librería en `application/libraries/image_moo.php`.

#### Cropping (Recortar)

##### Simple crop

```
$this->image_moo
    ->load("DSC01707.JPG")
    ->resize(100,100)
    ->save_dynamic(); (Véase Imagen 39).
```



Imagen 39: Recorte simple

##### Crop keep proportions but outputting a fixed size

```
$this->image_moo
    ->load('DSC01707.JPG')
    ->set_background_colour("#49F")
    ->resize(100,100,TRUE)
    ->save_dynamic(); (Véase Imagen 40)
```



Imagen 40: Recorte proporcional

### Resize with crop

```
$this->image_moo  
->load('DSC01707.JPG')  
->resize_crop(100,100)  
->save_dynamic(); (Véase Imagen 41).
```



Imagen 41: Recorte Redimensionado

### Borders (Bordes)

#### Simple photo style border

```
$this->image_moo  
->load('DSC01707.JPG')  
->resize_crop(100,100)  
->border(5, "#ffffff")  
->border(1, "#000000")  
->save_dynamic(); (Véase Imagen 42).
```



Imagen 42: Bordes simples

#### Rounded corners

```
$this->image_moo  
->load('DSC01707.JPG')  
->resize_crop(100,100)  
->round(5)  
->save_dynamic(); (Véase Imagen 43).
```



Imagen 43: Borders redondeados

### 3d effect

```
$this->image_moo  
->load('DSC01707.JPG')  
->resize_crop(100,100)  
->border_3d(5)  
->save_dynamic(); (Véase Imagen 44).
```



Imagen 44: Bordos efectos 3D

### Watermarking (Marcas de agua)

#### Using an image

```
$this->image_moo  
->load('DSC01707.JPG')  
->load_watermark("matmoo.gif")  
->resize(400,400)  
->watermark(5)  
->save_dynamic(); (Véase Imagen 45).
```



Imagen 45: Marcas de agua con imagen

#### Using your own text

```
$this->image_moo  
->load('DSC01707.JPG')  
->make_watermark_text("MatMoo.com", "DANUBE__.TTF", 25, "#000")  
->resize(400,400)  
->watermark(2)
```

->save\_dynamic(); (Véase Imagen 46).



Imagen 46: Marcas de agua con texto

### 9.3.7 JRating

Jrating es un plugin JQuery que en el proyecto ha sido usado para valorar las webs y productos de las empresas.

El funcionamiento de este plugin es bastante sencillo y fácil de configurar.

Incluimos el CSS y los JS.

```
<!-- include CSS & JS files -->
<!-- CSS file -->
<link rel="stylesheet" type="text/css" href="jRating.jquery.css"
media="screen" />
<!-- jQuery files -->
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jRating.jquery.js"></script>
```

Añadir el html.

```
<!-- basic exemple -->
<div class="exemple">

  <!-- in this exemple, 12 is the average and 1 is the id of the line
to update in DB -->
  <div class="basic" data-average="12" data-id="1"></div>

  <!-- in this other exemple, 8 is the average and 2 is the id of
the line to update in DB -->
  <div class="basic" data-average="8" data-id="2"></div>

</div>
```

Añadir el JRating Plugin (Véase Imagen 47).

```
<script type="text/javascript">
$(document).ready(function(){
  // simple jRating call
  $(".basic").jRating();

  // more complex jRating call
  $(".basic").jRating({
    step:true,
    length : 20, // nb of stars
    onSuccess : function(){
      alert('Success : your rate has been saved :');
    }
  });

  // you can rate 3 times ! After, jRating will be disabled
  $(".basic").jRating({
    canRateAgain : true,
    nbRates : 3
  });
});
```



```

// get the clicked rate !
$(".basic").jRating({
  onClick : function(element,rate) {
    alert(rate);
  }
});
});
</script>

```



Imagen 47: JRating

Para guardar los datos en la base de datos creamos la librerías `assets/jrating/php/jrating.php` y `assets/jrating/php/jrating.jquery.js`.

```
<?php
```

```
$aResponse['error'] = false;
```

```
$aResponse['message'] = "";
```

```
$aResponse['server'] = "";
```

```
if (isset($_POST['action'])) {
```

```
    if (htmlentities($_POST['action'], ENT_QUOTES, 'UTF-8') == 'rating') {
```

```
        $id = intval($_POST['idBox']);
```

```
        $rate = floatval($_POST['rate']);
```

```
        $iduser = intval($_POST['idUser']);
```

```
        $idweb = intval($_POST['idWeb']);
```

```
        $conexion = mysql_connect("localhost", "root", "");
```

```
        mysql_select_db("web", $conexion);
```

```
        $sqlsel = "SELECT * FROM socialrating WHERE id='" . $id . "' AND idweb='" . $idweb . "' AND iduser='" . $iduser . "'";
```

```
        $resultado = mysql_query($sqlsel) or die (mysql_error());
```

```
        if (mysql_num_rows($resultado) > 0) {
```

```
            //$sql = "INSERT INTO socialrating (id,rate,iduser,idweb) VALUES ($id,$rate,$iduser,$idweb)";
```

```

    $sql = "UPDATE socialrating SET rate=" . $rate . " WHERE id=" . $id . " AND idweb=" .
$idweb . " AND iduser=" . $iduser . """;

    mysql_query($sql);

} else {

    $sql = "INSERT INTO socialrating (id,rate,iduser,idweb) VALUES
($id,$rate,$iduser,$idweb)";

    mysql_query($sql);

}

$mysql_close($conexion);

// if request successful

$success = true;

// else $success = false;

// json datas send to the js file

if ($success) {

    $aResponse['message'] = 'Your rate has been successfully recorded. Thanks for your rate
:~';

    $aResponse['server'] = '<strong>Success answer :</strong> Success : Your rate has been
recorded. Thanks for your rate :)<br />';

    $aResponse['server'] .= '<strong>Rate received :</strong> ' . $rate . '<br />';

    $aResponse['server'] .= '<strong>ID to update :</strong> ' . $id;

    echo json_encode($aResponse);

} else {

    $aResponse['error'] = true;

    $aResponse['message'] = 'An error ocured during the request. Please retry';

    $aResponse['server'] = '<strong>ERROR :</strong> Your error if the request crash !';

    echo json_encode($aResponse);

}

} else {

    $aResponse['error'] = true;

```

```

    $aResponse['message'] = "'action" post data not equal to \'rating\'';

    $aResponse['server'] = '<strong>ERROR :</strong> "action" post data not equal to
\'rating\'';

    echo json_encode($aResponse);

}

} else {

    $aResponse['error'] = true;

    $aResponse['message'] = '$_POST[\'action\'] not found';

    $aResponse['server'] = '<strong>ERROR :</strong> $_POST[\'action\'] not found';

    echo json_encode($aResponse);

}

```

### JRating.jquery.js

```

(function($) {

    $.fn.jRating = function(op) {

        var defaults = {

            /** String vars */

            bigStarsPath :
'http://localhost/web/assets/jrating/jquery/icons/stars.png', // path of the icon stars.png

            smallStarsPath :
'http://localhost/web/assets/jrating/jquery/icons/small.png', // path of the icon small.png

            phpPath : 'http://localhost/web/assets/jrating/php/jRating.php', //
path of the php file jRating.php

            type : 'big', // can be set to 'small' or 'big'

            /** Boolean vars */

            step:false, // if true, mouseover binded star by star,

            isDisabled:false, // if true, user could not rate

```

showRateInfo: true, // show rates informations when cursor moves  
onto the plugin

canRateAgain : false, // if true, the user could rates {nbRates} times  
with jRating.. Default, 1 time

sendRequest: true, // send values to server

*/\*\* Integer vars \*/*

length:5, // number of star to display

decimalLength : 0, // number of decimals.

rateMax : 10, // maximal rate - integer from 0 to 9999 (or more)

mouseover

rateInfosX : -45, // relative position in X axis of the info box when

mouseover

rateInfosY : 5, // relative position in Y axis of the info box when

nbRates : 1,

*/\*\* Functions \*/*

onSuccess : null, // Fires when the server response is ok

onError : null, // Fires when the server response is not ok

onClick: null // Fires when clicking on a star

};

if(this.length>0)

return this.each(function() {

*/\*vars\*/*

var opts = \$.extend(defaults, op),

newWidth = 0,

starWidth = 0,

starHeight = 0,

bgPath = "",

hasRated = false,

globalWidth = 0,

```

nbOfRates = opts.nbRates;

if($(this).hasClass('jDisabled') || opts.isDisabled)
    var jDisabled = true;
else
    var jDisabled = false;

getStarWidth();
$(this).height(starHeight);

var average = parseFloat($(this).attr('data-average')), // get the
average of all rates

idBox = parseInt($(this).attr('data-id')), // get the id of the box

idUser = $(this).attr('iduser'),
idWeb = $(this).attr('idweb'),

widthRatingContainer = starWidth*opts.length, // Width of the
Container

widthColor = average/opts.rateMax*widthRatingContainer, // Width of
the color Container

quotient =
$('<div>',
{
    'class' : 'jRatingColor',
    css:{
        width:widthColor
    }
}).appendTo($(this)),

```

```

average =
$(<div>,
{
    'class' : 'jRatingAverage',
    css:{
        width:0,
        top:- starHeight
    }
}).appendTo($(this)),

```

```

jstar =
$(<div>,
{
    'class' : 'jStar',
    css:{
        width:widthRatingContainer,
        height:starHeight,
        top:- (starHeight*2),
        background: 'url('+bgPath+') repeat-x'
    }
}).appendTo($(this));

```

```

$(this).css({width:
widthRatingContainer,overflow:'hidden',zIndex:1,position:'relative'});

```

```

if(!jDisabled)
$(this).unbind().bind({

```

```

mouseenter : function(e){
    var realOffsetLeft = findRealLeft(this);
    var relativeX = e.pageX - realOffsetLeft;
    if (opts.showRateInfo)
    var tooltip =
    $('<p>',{
        'class' : 'jRatingInfos',
        html : getNote(relativeX)+' <span
class="maxRate">/ '+opts.rateMax+'</span>',
        css : {
            top: (e.pageY + opts.rateInfosY),
            left: (e.pageX + opts.rateInfosX)
        }
    }).appendTo('body').show();
},
mouseover : function(e){
    $(this).css('cursor','pointer');
},
mouseout : function(){
    $(this).css('cursor','default');
    if(hasRated) average.width(globalWidth);
    else average.width(0);
},
mousemove : function(e){
    var realOffsetLeft = findRealLeft(this);
    var relativeX = e.pageX - realOffsetLeft;
    if(opts.step) newWidth =
    Math.floor(relativeX/starWidth)*starWidth + starWidth;

```

```

else newWidth = relativeX;

average.width(newWidth);

if (opts.showRateInfo)
    $("p.jRatingInfos")
        .css({
            left: (e.pageX + opts.rateInfosX)
        })
        .html(getNote(newWidth) + ' <span class="maxRate">/
'+opts.rateMax+'</span>');
    },
    mouseleave : function(){
        $("p.jRatingInfos").remove();
    },
    click : function(e){
        var element = this;

        /*set vars*/
        hasRated = true;
        globalWidth = newWidth;
        nbOfRates--;

        if(!opts.canRateAgain || parseInt(nbOfRates) <= 0)
            $(this).unbind().css('cursor','default').addClass('jDisabled');

        if (opts.showRateInfo)
            $("p.jRatingInfos").fadeOut('fast',function(){$(this).remove();});

        e.preventDefault();

        var rate = getNote(newWidth);

        average.width(newWidth);

```



```

CODE **/
                                /** ONLY FOR THE DEMO, YOU CAN REMOVE THIS
                                */
                                $(' .datasent p').html('<strong>idBox :
</strong>'+idBox+'<br /><strong>rate : </strong>'+rate+'<br /><strong>idUser :
</strong>'+idUser+'<br /><strong>idWeb : </strong>'+idWeb+'<br /><strong>action
:</strong> rating');

                                $(' .serverResponse
p').html('<strong>Loading...</strong>');

                                /** END ONLY FOR THE DEMO **/

                                if(opts.onClick) opts.onClick( element, rate );

                                if(opts.sendRequest) {

                                        $.post(opts.phpPath,{

                                                idBox : idBox,

                                                rate : rate,

                                                idUser: idUser,

                                                idWeb : idWeb,

                                                action : 'rating'

                                        },

                                        function(data) {

                                                if(!data.error)

                                                {

                                                        /** ONLY FOR THE
DEMO, YOU CAN REMOVE THIS CODE **/

                                                        $(' .serverResponse p').html(data.server);

                                                        /** END ONLY FOR THE
DEMO **/

```

```

display an alert box,
jNotify Plugin :) http://www.myjqueryplugins.com/jNotify
*/
opts.onSuccess( element, rate );

DEMO, YOU CAN REMOVE THIS CODE **/

$('.serverResponse p').html(data.server);
DEMO **/

display an alert box,
jNotify Plugin :) http://www.myjqueryplugins.com/jNotify
*/
opts.onError( element, rate );

/** Here you can
or use the
exemple :
if(opts.onSuccess)
}
else
{

/** ONLY FOR THE

/** END ONLY FOR THE

/** Here you can
or use the
exemple :
if(opts.onError)
}
},
'json'

```

```

        );
    }

}

});

function getNote(relativeX) {
    var noteBrut =
parseFloat((relativeX*100/widthRatingContainer)*parseInt(opts.rateMax)/100);
    var dec=Math.pow(10,parseInt(opts.decimalLength));
    var note = Math.round(noteBrut*dec)/dec;
    return note;
};

function getStarWidth(){
    switch(opts.type) {
        case 'small' :
            starWidth = 12; // width of the picture
small.png
            starHeight = 10; // height of the picture
small.png
            bgPath = opts.smallStarsPath;
            break;
        default :
            starWidth = 23; // width of the picture
stars.png
            starHeight = 20; // height of the picture
stars.png
            bgPath = opts.bigStarsPath;

```

```
        }  
    };  
  
    function findRealLeft(obj) {  
        if( !obj ) return 0;  
        return obj.offsetLeft + findRealLeft( obj.offsetParent );  
    };  
});  
  
    }  
})(jQuery);
```

## 9.4 Anexo IV: Manual de usuario

A continuación se muestra una completa guía de usuario para demostrar el funcionamiento de la aplicación:

### Usuario Cliente:

Los usuarios anónimos pueden hacer uso del buscador, visualizar productos y visualizar webs. Para poder realizar comentarios, valorar productos y realizar reservas deberán estar registrados. Para registrarse en el sistema existen dos maneras. La primera es que un usuario gestor registre manualmente al usuario. Y la otra es que el usuario se autoregiste validando su cuenta de email y contraseña.

### Registrar Usuario:

El usuario hace click en enlace “Registrarse” situado en la parte superior derecha de la pantalla principal de la aplicación (Véase Imagen 48).



Imagen 48: Enlace de Registrar Usuario

A continuación aparece una pantalla de registro de usuario con dos campos, el campo usuario que deberá ser una cuenta de email válida, y el campo contraseña (Véase Imagen 49).

Una pantalla de registro de usuario con un fondo verde. En la parte superior izquierda hay un logo con una hoja verde y el texto 'enatural'. En el centro hay un formulario con un título 'Registrarse'. El formulario contiene dos campos de entrada: 'Email' y 'Contraseña'. Debajo de los campos hay dos botones: 'Volver' y 'Registrar'.

Imagen 49: Registrar Usuario

Si se ha realizado el registro correctamente se enviará un email a la cuenta del usuario para su posterior validación y se mostrará un mensaje de aviso (Véase Imagen 50).

**Correo de validación enviado.** Revise su correo y haga click en el enlace para activar su cuenta.

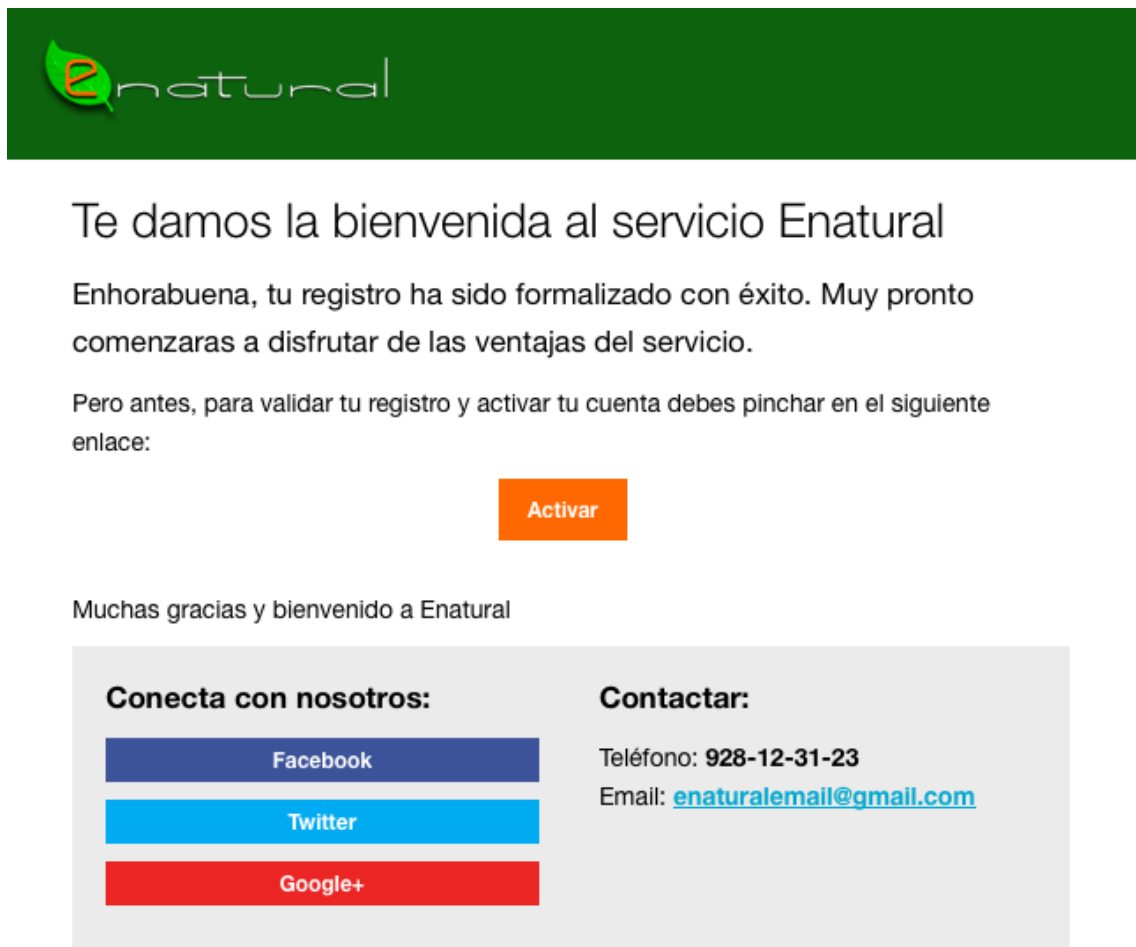
Imagen 50: Mensaje de correo de validación

En este momento el usuario esta registrado pero no esta activado, si intentamos entrar sin validar el sistema nos muestra el siguiente aviso (Véase Imagen 51).

**¡Error!** El usuario esta registrado pero todavía no ha activado la cuenta.

Imagen 51: Mensaje de aviso usuario no ha activado la cuenta.

Para terminar el registro en el sistema el usuario deberá de consultar su correo y abrir el mensaje de correo y hacer clicn en el enlace "Activar" (Véase Imagen 52) .



**Te damos la bienvenida al servicio Enatural**

Enhorabuena, tu registro ha sido formalizado con éxito. Muy pronto comenzaras a disfrutar de las ventajas del servicio.

Pero antes, para validar tu registro y activar tu cuenta debes pinchar en el siguiente enlace:

**Activar**

Muchas gracias y bienvenido a Enatural

**Conecta con nosotros:**

- Facebook
- Twitter
- Google+

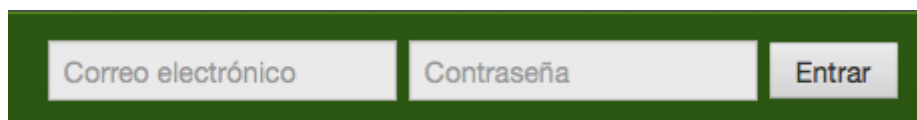
**Contactar:**

Teléfono: **928-12-31-23**  
Email: [enaturalemail@gmail.com](mailto:enaturalemail@gmail.com)

Imagen 52: Mensaje de correo de activación de cuenta de usuario.

### Iniciar Sesión:

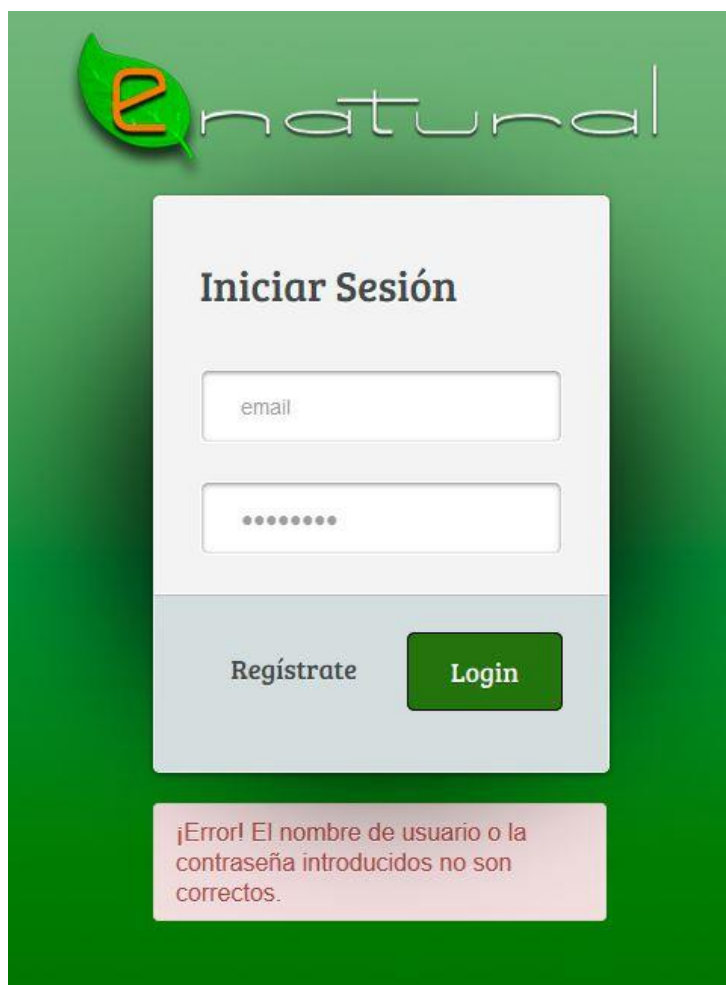
El usuario accede con un usuario y contraseña. El usuario estará conformado por una dirección email válida y una contraseña. Se podrá acceder de dos maneras distintas, la más habitual es dirigirse a la parte superior derecha de la pantalla como se muestra a continuación (Véase Imagen 53) :



Formulario de inicio de sesión con tres elementos: un campo de texto etiquetado "Correo electrónico", un campo de texto etiquetado "Contraseña", y un botón etiquetado "Entrar".

Imagen 53: Iniciar sesión I.

Si el usuario se equivoca al introducir su correo o contraseña, el sistema dará un aviso y redirigirá al usuario a otra pantalla de Login (Véase Imagen 54):



Pantalla de inicio de sesión con el logo "e natural" en la parte superior. El formulario contiene un campo "email" y un campo de contraseña. Debajo de los campos hay un botón "Regístrate" y un botón "Login". Un mensaje de error en un recuadro rojo indica: "¡Error! El nombre de usuario o la contraseña introducidos no son correctos."

Imagen 54: Iniciar sesión II.

Valorar Web:

El usuario anónimo podrá visualizar las valoraciones hechas por los usuarios de la web. Pero para poder valorar deberá de iniciar sesión (Véase Imagen 55).



Imagen 55: Valoraciones de la web sin iniciar sesión.

Una vez iniciada la sesión se puede realizar las valoraciones haciendo click en las estrellas, se podrán hacer valoraciones de la puntuación total de la web, puntuación de los servicios y de la calidad/Precio de la web. Además se podrán realizar recomendaciones de Google+, Twitter y Me gusta de Facebook (Véase Imagen 56).



**Puntos /7,90**

1 valoraciones de clientes

Comentarios

Servicio

Calidad/Precio

**g+1** Recomendar esto en Google

0

**f** Me gusta

**Twitter** 0 **Compartir**

Si lo deseas puedes valorar la web

Puntuación de la Web:

Voto Guardado.

Servicios:

Calidad/Pr **8.7 / 10**

Imagen 56: Valoraciones de la web iniciada la sesión.

Una vez realizada la valoración se podrá cambiar en cualquier otro momento haciendo click en el botón “Editar Valoraciones” (Véase Imagen 57):

**Puntos /8,43**

2 valoraciones de clientes

Comentarios

Servicio

Calidad/Precio

**g+1** **Has hecho +1 en esto**

0

**f** Me gusta

**Twitter** 0 **Compartir**

Puntuación de la Web:

Servicios:

Calidad/Precio:

Ya has valorado

**Editar Valoraciones**

Imagen 57: Editar valoraciones.

### Valorar Producto:

Se realizan las mismas operaciones que para valorar una web (Véase Imagen 58):



Imagen 58: Valorar producto.

### Reservar:

Para poder realizar una reserva el usuario registrado debe hacer click en el botón “Reservar” situado en la parte izquierda central de la página del producto (Véase Imagen 59):



Imagen 59: Reservar producto.

Seguidamente el usuario deberá realizar un formulario de reserva para ver la disponibilidad, información y precio de la reserva (Véase Imagen 60):

Inicio Información Servicios

**Reserva**

Información de la Reserva:

Precio Total: 721.2 Día/s o Noche/s: 4 Adultos: 2

Niños: 0 % Añadido: 0

Inicio	Fin	Precio Adulto	Precio Niño	Impuestos	Tasas
07-02-2014	30-09-2014	90.15€	0€	0%	0€

Tratamiento:  
 Sr.

Nombre:  
 Yeray

Apellidos

Imagen 60: Formulario de reserva de producto.

Una vez completado el formulario de reserva, si lo desea, el usuario puede hacer click sobre el botón “Reservar Ahora” para realizar la petición de la reserva en el sistema (Véase Imagen 61):

Código Postal

Teléfono

Email

Si lo desea puede poner otro email válido para la confirmación de su reserva.

Fechas disponibles para la Reserva:

Entrada:  Salida:

Adultos:  Niños(r)

Si lo desea puede añadir comentarios a:

**March 2014**

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Imagen 61: Realizar reserva de producto.

Una vez la petición se ha realizado correctamente, el sistema mandará automáticamente un email de confirmación de reserva(Véase Imagen 62).



## ¡Enhorabuena!

Ha realizado correctamente la reserva del producto.

### Información de la reserva

Nombre de la reserva: Casa del Palomar

Nombre de usuario: Yeray

Apellidos: Pérez Valiente

Importe total: 87.7

Información de la reserva:

### Condiciones de la reserva

La reserva deberá abonarse 48 horas antes de la finalización de la reserva. De no ser así, la reserva quedará cancelada. Además el usuario podrá cancelar la reserva cuando quiera en el apartado de la web "Mis reservas"

[Pagar](#)

<b>Connecta con nosotros:</b>	<b>Contactar:</b>
<a href="#">Facebook</a>	Teléfono: <b>928-12-31-23</b>
<a href="#">Twitter</a>	Email: <a href="mailto:enaturalemail@gmail.com">enaturalemail@gmail.com</a>
<a href="#">Google+</a>	

Imagen 62: Email validación de reserva.

La reserva podrá abonarse de dos formas distintas. La primera forma con el propio email de confirmación haciendo click en el enlace "Pagar" y la segunda a través del menú desplegable situado en la parte superior derecha de pantalla en el menú "Mis reservas"(Véase Imagen 63):

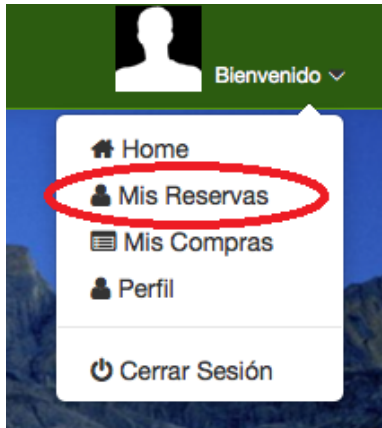


Imagen 63: Mis reservas.

Una vez situado en “Mis Reservas” (Véase Imagen 64) hacer click en el botón “Comprar” para realizar la compra a través de Paypal (Véase Imagen 65):

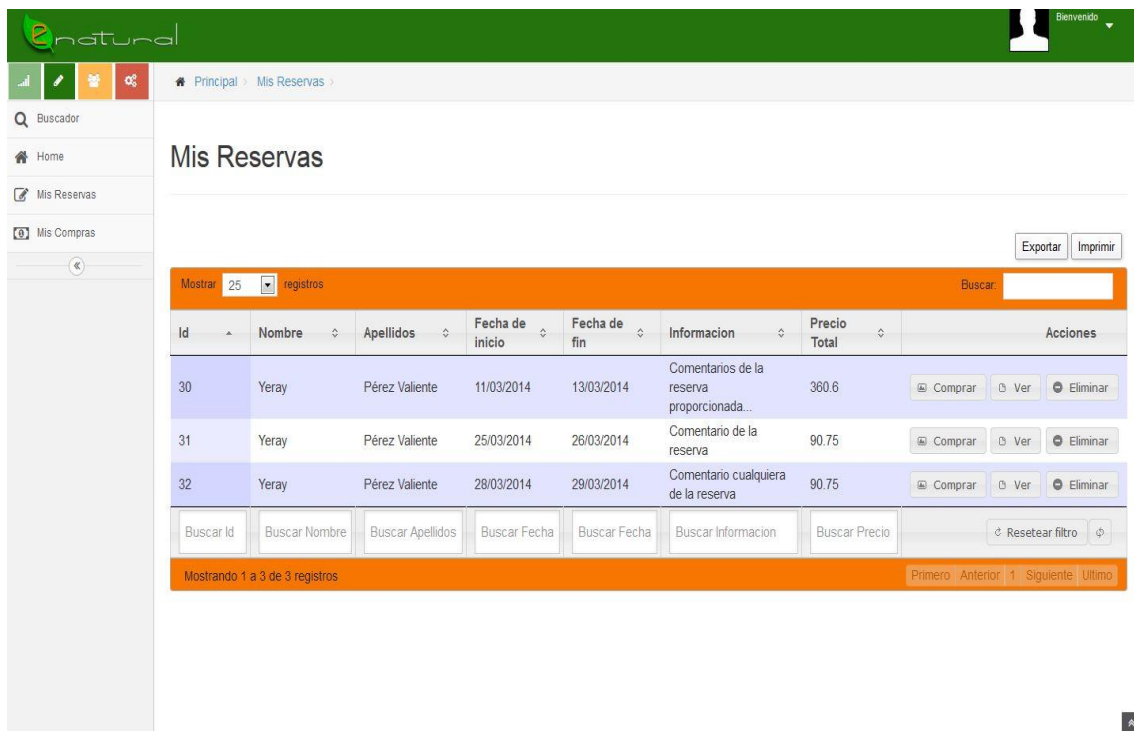
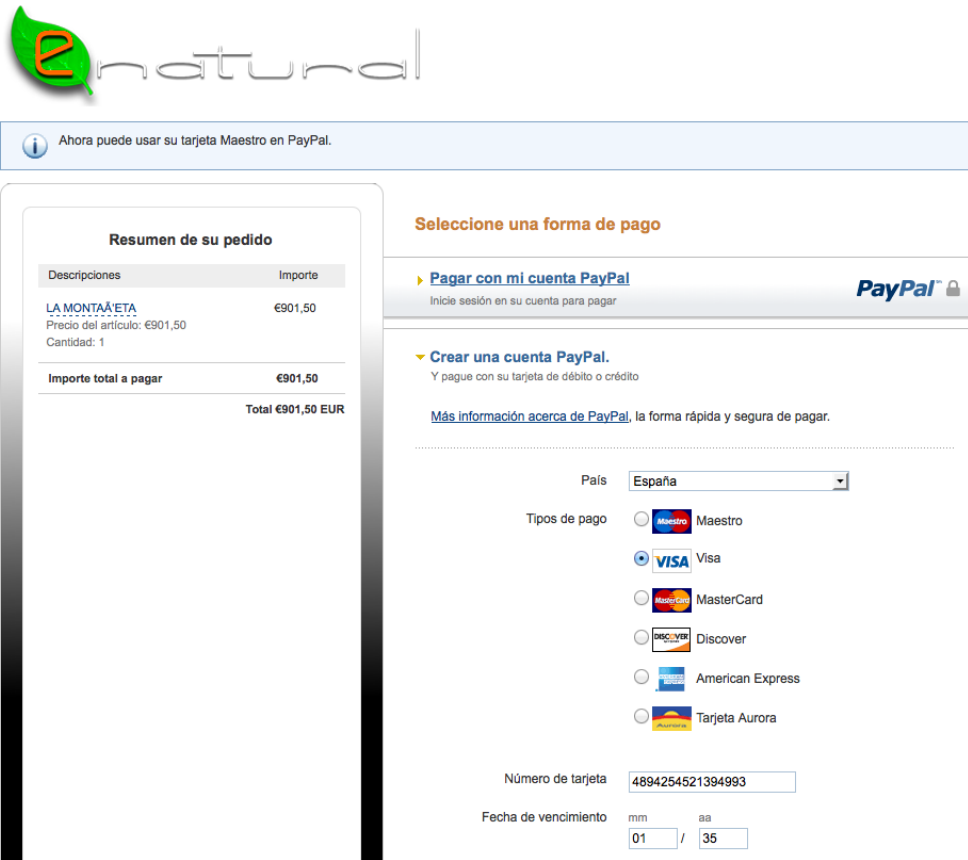


Imagen 64: Reservas realizadas por el usuario.



Imagen 65: Comprar

A continuación, el usuario deberá seguir los pasos del sistema Paypal externo a la aplicación web (Véase Imagen 66, 67 y 68):



Ahora puede usar su tarjeta Maestro en PayPal.

### Resumen de su pedido

Descripciones	Importe
LA MONTAÁETA Precio del artículo: €901,50 Cantidad: 1	€901,50
<b>Importe total a pagar</b>	<b>€901,50</b>
<b>Total €901,50 EUR</b>	

### Seleccione una forma de pago

[Pagar con mi cuenta PayPal](#) **PayPal**   
Inicie sesión en su cuenta para pagar

[Crear una cuenta PayPal.](#)  
Y pague con su tarjeta de débito o crédito.  
[Más información acerca de PayPal](#), la forma rápida y segura de pagar.

País: España

Tipos de pago:

Maestro

Visa

MasterCard

Discover

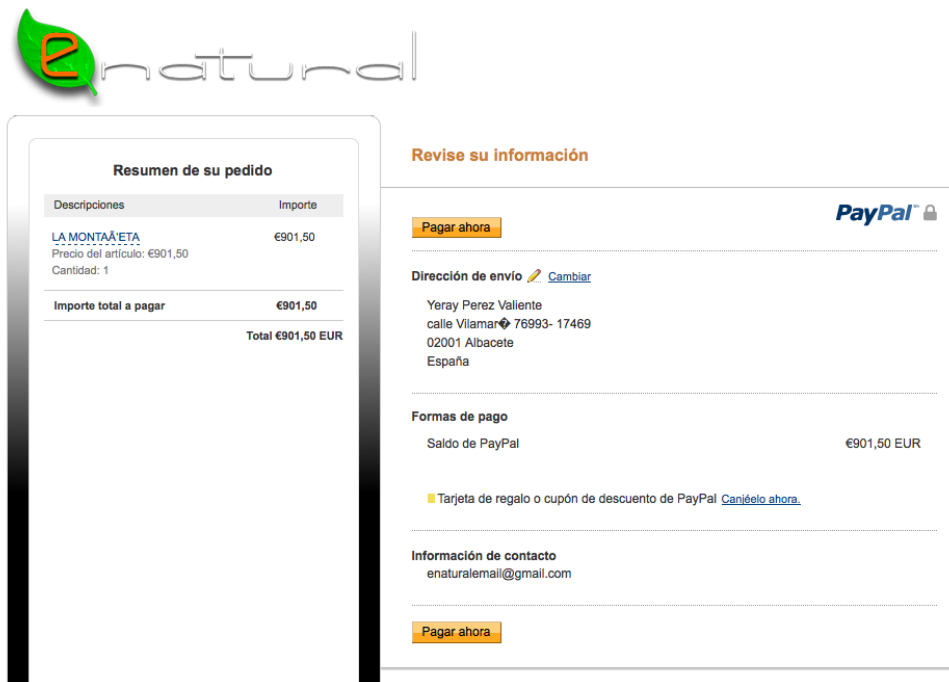
American Express

Tarjeta Aurora

Número de tarjeta: 4894254521394993

Fecha de vencimiento: mm / aa  
01 / 35

Imagen 66: Paypal I.



### Resumen de su pedido

Descripciones	Importe
LA MONTAÁETA Precio del artículo: €901,50 Cantidad: 1	€901,50
<b>Importe total a pagar</b>	<b>€901,50</b>
<b>Total €901,50 EUR</b>	

### Revise su información

**PayPal**

[Pagar ahora](#)

**Dirección de envío** [Cambiar](#)

Yeray Perez Valiente  
calle Vilamar 76993- 17469  
02001 Albacete  
España

**Formas de pago**

Saldo de PayPal €901,50 EUR

Tarjeta de regalo o cupón de descuento de PayPal [Canjéelo ahora.](#)

**Información de contacto**  
enaturalemil@gmail.com

[Pagar ahora](#)

Imagen 67: Paypal II.

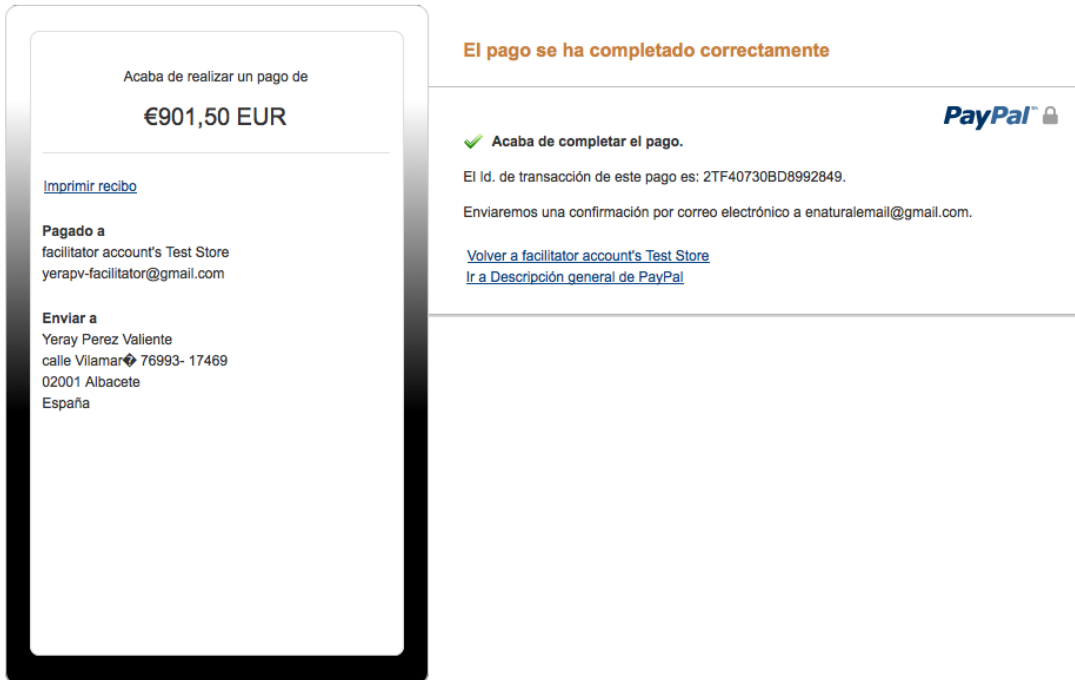


Imagen 68: Paypal III.

Una vez realizada la comprobación del pago por parte del gestor de la web, se quita la reserva del apartado “Mis reservas” y pasa al apartado “Mis compras” del menú de usuario situado en la parte superior derecha de la pantalla (Véase Imagen 69):

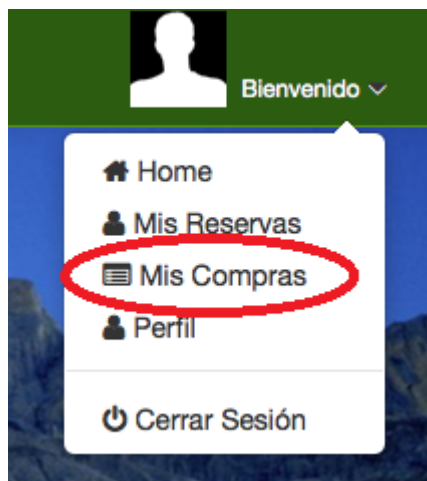
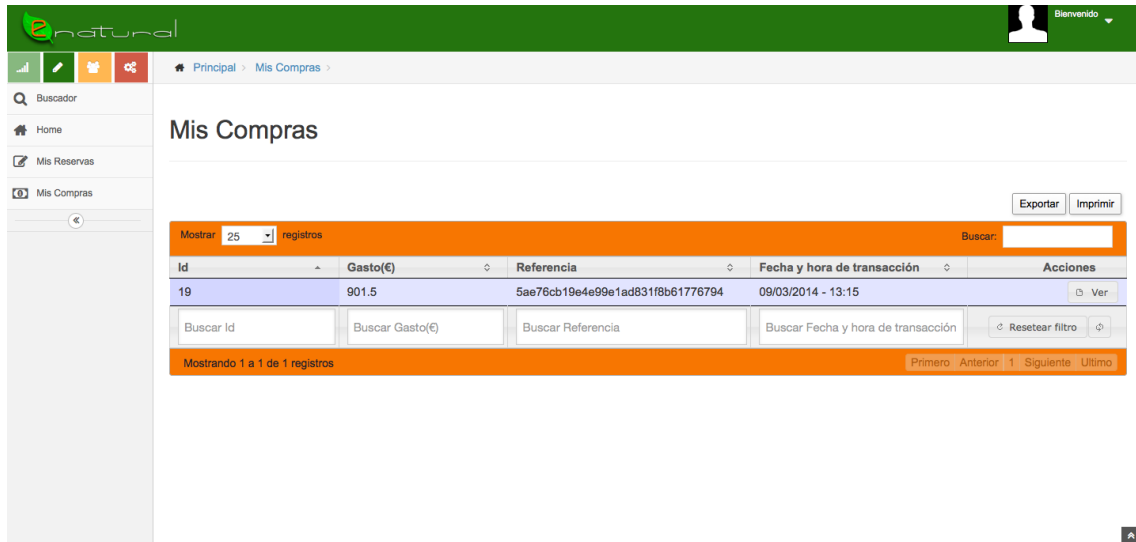


Imagen 69: Mis compras.



Una vez situado en “Mis Compras” (Véase Imagen 70) podrá ver las compras realizadas en todas las webs del sistema y la información asociada a cada compra.



enatural

Bienvenido

Principal > Mis Compras >

Buscador

Home

Mis Reservas

Mis Compras

### Mis Compras

Exportar Imprimir

Mostrar 25 registros

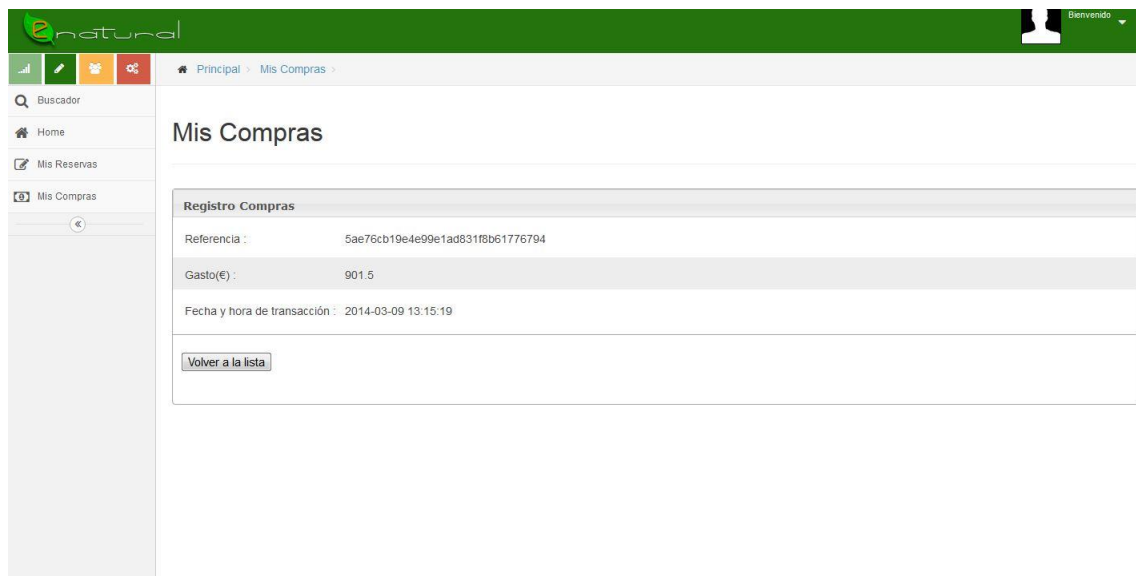
Id	Gasto(€)	Referencia	Fecha y hora de transacción	Acciones
19	901.5	5ae76cb19e4e99e1ad831f8b61776794	09/03/2014 - 13:15	Ver

Buscar:

Buscar Id    Buscar Gasto(€)    Buscar Referencia    Buscar Fecha y hora de transacción    Resetear filtro

Mostrando 1 a 1 de 1 registros    Primero Anterior 1 Siguiente Ultimo

Imagen 70: Mis compras realizadas.



enatural

Bienvenido

Principal > Mis Compras >

Buscador

Home

Mis Reservas

Mis Compras

### Mis Compras

#### Registro Compras

Referencia : 5ae76cb19e4e99e1ad831f8b61776794

Gasto(€) : 901.5

Fecha y hora de transacción : 2014-03-09 13:15:19

Volver a la lista

Imagen 71: Información de registro de compra.

Home:

El usuario en toda la aplicación podrá acceder a la parte privada de la aplicación, haciendo click en el botón “Home” del menú desplegable situado en la parte superior derecha de la aplicación ( Véase Imagen 72 y 73):

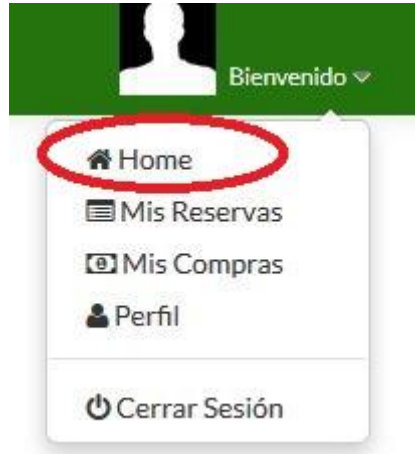


Imagen 72: Home

A screenshot of the user's main dashboard. The header is green with the 'enatural' logo and a user profile icon labeled 'Bienvenido'. The main content area is titled 'Historial de Visitas' and contains a table with columns: Id, Iduser, Link, Date, and Acciones. The table lists 11 records of visits to various web pages on 27/03/2014. A search bar and a 'Mostrar 25 registros' dropdown are visible above the table.

Id	Iduser	Link	Date	Acciones
19	144	http://www.yeraypv.com/web/webs/product/1/69	27/03/2014 - 12:50	<a href="#">Ir a Web</a>
20	144	http://www.yeraypv.com/web/webs/page/1/1	27/03/2014 - 12:50	<a href="#">Ir a Web</a>
21	144	http://www.yeraypv.com/web/webs/product/1/13	27/03/2014 - 12:50	<a href="#">Ir a Web</a>
22	144	http://www.yeraypv.com/web/webs/product/1/13	27/03/2014 - 12:50	<a href="#">Ir a Web</a>
23	144	http://www.yeraypv.com/web/webs/product/3/8	27/03/2014 - 12:52	<a href="#">Ir a Web</a>
24	144	http://www.yeraypv.com/web/webs/product/1/69	27/03/2014 - 12:56	<a href="#">Ir a Web</a>
25	144	http://www.yeraypv.com/web/webs/product/1/69	27/03/2014 - 12:56	<a href="#">Ir a Web</a>
26	144	http://www.yeraypv.com/web/webs/product/1/2	27/03/2014 - 12:59	<a href="#">Ir a Web</a>
27	144	http://www.yeraypv.com/web/webs/product/1/2	27/03/2014 - 13:01	<a href="#">Ir a Web</a>
28	144	http://www.yeraypv.com/web/webs/product/1/2	27/03/2014 - 13:10	<a href="#">Ir a Web</a>
29	144	http://www.yeraypv.com/web/webs/product/1/2	27/03/2014 - 13:19	<a href="#">Ir a Web</a>
30	144	http://www.yeraypv.com/web/webs/product/1/2	27/03/2014 - 13:27	<a href="#">Ir a Web</a>

Imagen 73: Principal de usuario.

### Cerrar Sesión:

El usuario podrá cerrar su sesión en cualquier momento que lo desee, deberá hacer click en el enlace “Cerrar Sesión” situado en el menú desplegable de la parte superior derecha de la aplicación (Véase Imagen 74):

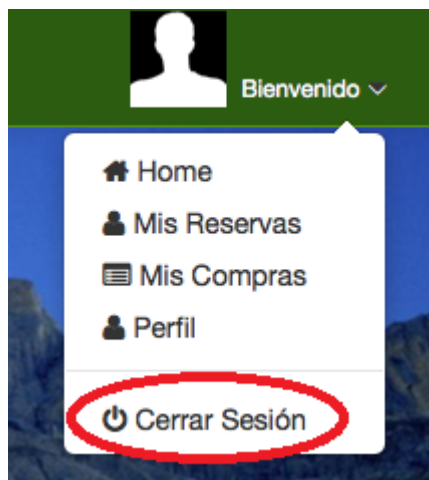


Imagen 74: Cerrar sesión de usuario.

### **Usuario Gestor:**

Los gestores pueden acceder a la aplicación en cualquier momento, al acceder serán redirigidos a la pantalla principal de control.

#### Cerrar sesión

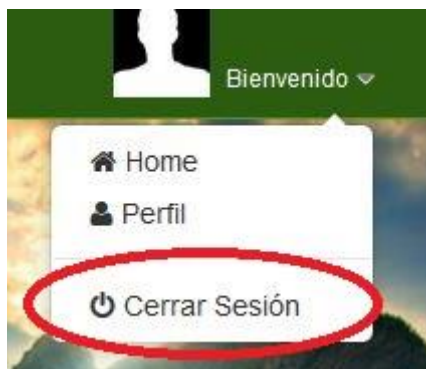


Imagen 75: Cerrar sesión gestor.

El usuario podrá cerrar sesión desde cualquier parte de la aplicación dirigiéndose a la parte superior derecha, desplegando el menú de usuario y haciendo clic en el botón "cerrar sesión" (Véase Imagen 75).

Cuando se realiza esta acción el sistema redirige al usuario a la pantalla principal de la aplicación. Para volver a entrar el usuario tendrá que volver a iniciar sesión.

#### Pantalla Principal:

Una vez el usuario se ha autenticado accederá al área de control del gestor (Véase Imagen 76):

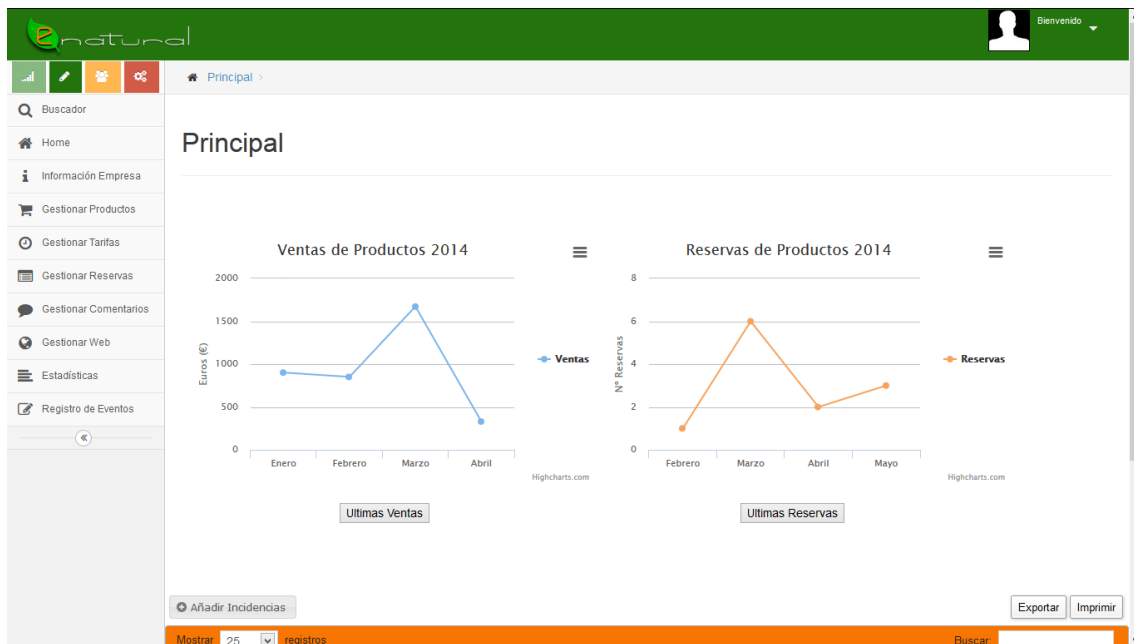


Imagen 76: Pantalla principal del usuario gestor.

### Menú:

En la parte izquierda el usuario dispone de un menú para navegar por las distintas opciones que ofrece la aplicación, este menú varía según el perfil del usuario (Véase Imagen 77):



Imagen 77: Menú

### Buscador:

Dirige al usuario a la pantalla del buscador (Véase Imagen 78).



Imagen 78: Buscador

**Principal:**

Dirige al usuario a la pantalla inicial del panel de control (Véase Imagen 79).

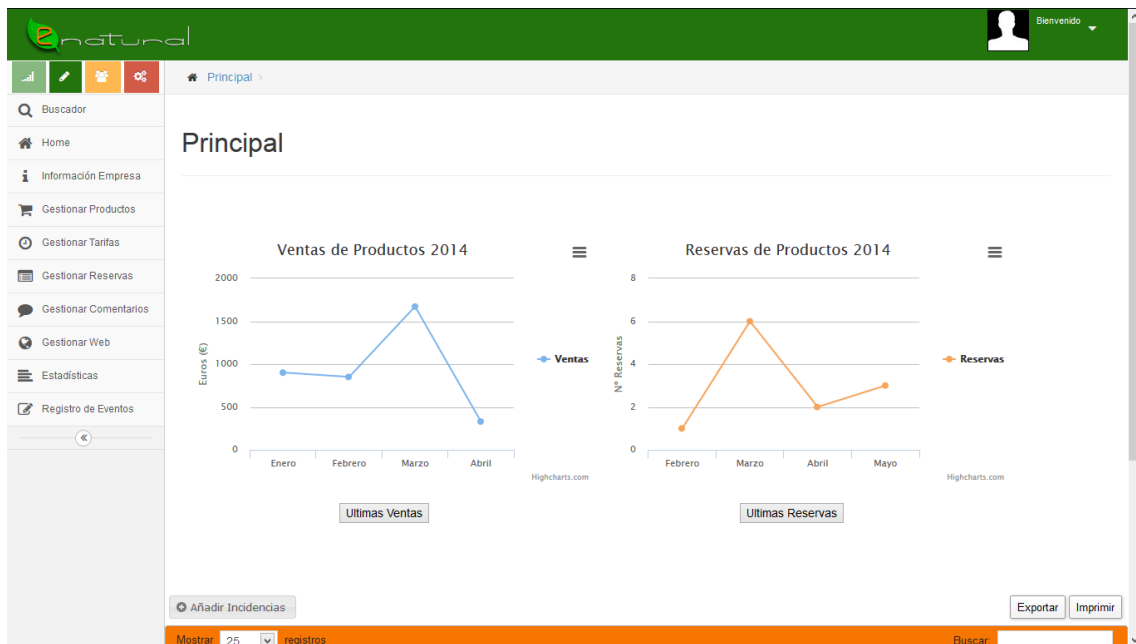


Imagen 79: Pantalla de control.

**Información Empresa:**

Esta opción dispone de varias operaciones que puede realizar el usuario relativo a la información de la empresa (Véase Imagen 80 y 81).

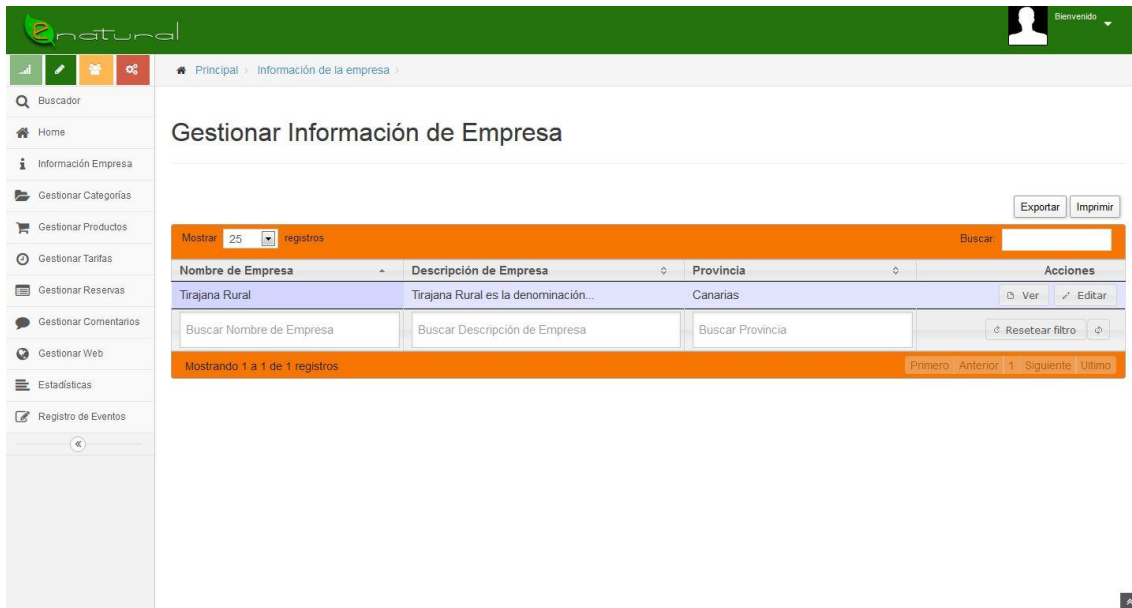


Imagen 80: Gestionar información de empresa.

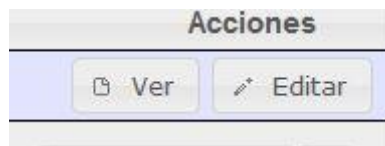


Imagen 81: Acciones de Gestionar información de empresa.

Puede ver la información relativa a la empresa (Véase Imagen 82). Se muestra el nombre, descripción y provincia.

### Gestionar Información de Empresa

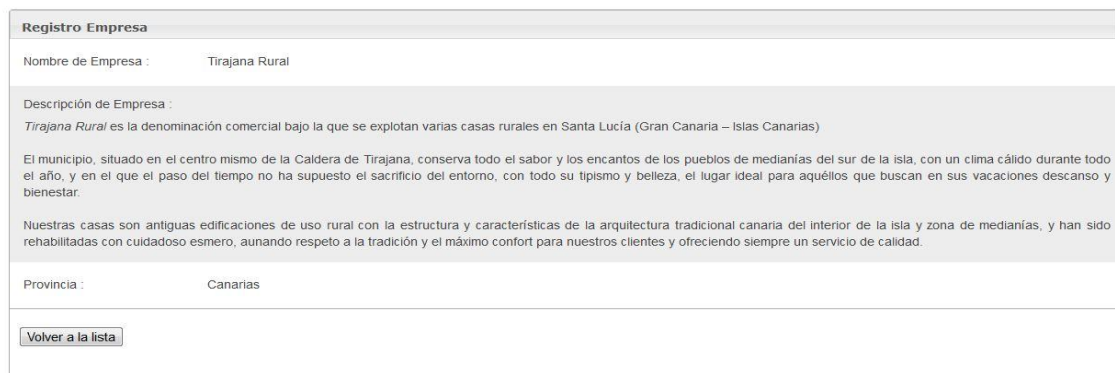


Imagen 82: Registro de empresa.

Puede editar la información de la empresa (Véase Imagen 83), dicha información aparecerá en el buscador de la aplicación y será útil para realizar las búsquedas de productos.

## Gestionar Información de Empresa

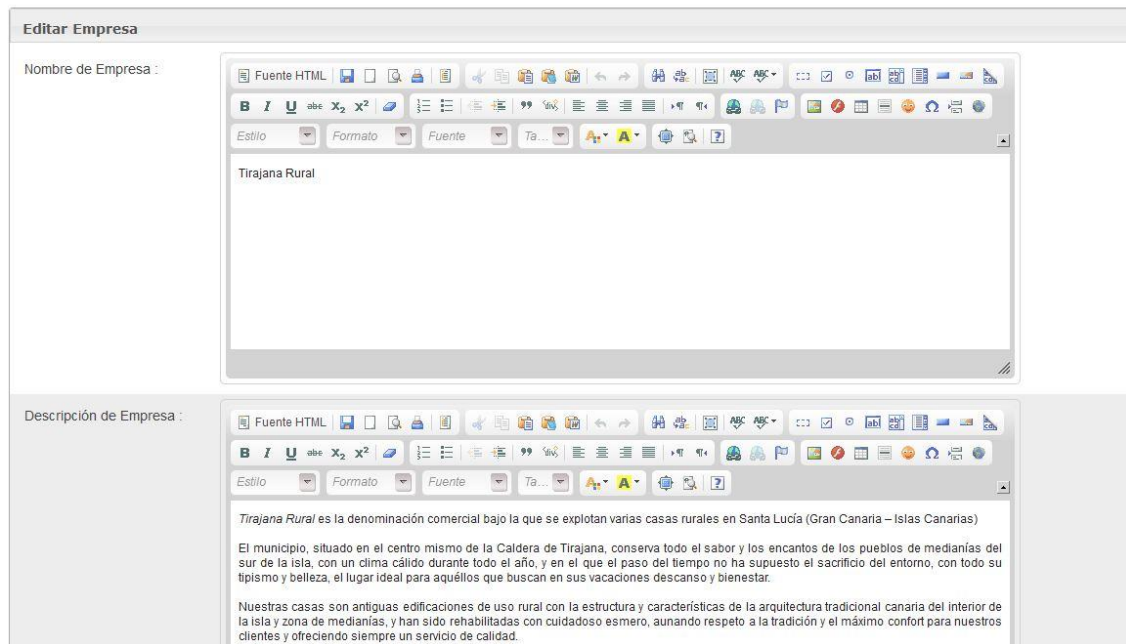


Imagen 83: Editar información de empresa.

## Gestionar Productos:

El usuario tiene la posibilidad de gestionar los productos de la empresa, las operaciones disponibles son la creación de productos, creación de galerías de imágenes del producto, edición, eliminación y búsqueda de productos (Ver Imagen 84 y 85).

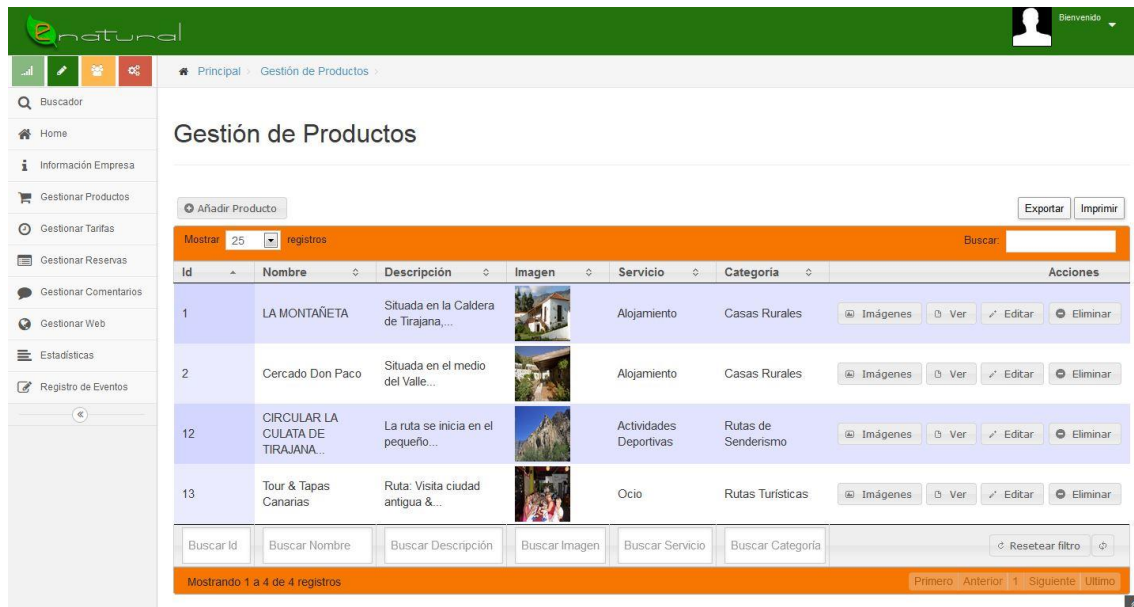


Imagen 84: Gestionar productos.





Imagen 85: Acciones de Gestionar productos.

### Crear producto:

Para crear un nuevo producto el usuario debe hacer click en "Añadir producto"(Véase Imagen 86)



Imagen 86: Añadir producto.

El usuario tiene la posibilidad de crear todos los productos de la empresa que estén disponibles para cualquier usuario, registrado o no registrado. Para la creación de un producto serán necesarios un nombre de producto, una descripción, asociarlo a un servicio y una categoría de producto (Véase Imagen 87).

## Gestión de Productos

A web form titled "Añadir Producto" with a light gray background. It contains several input fields and a rich text editor. At the top left, the title "Añadir Producto" is displayed. Below it, there is a label "Imagen:" followed by a button "Subir un archivo". The next field is "Nombre\*" with an empty text input box. Below that is "Descripción:" followed by a rich text editor with a complex toolbar. At the bottom, there are three dropdown menus: "Servicio\*" with "Seleccionar Servicio", "Categoría\*" with "Seleccionar Categoría", and "Localidad/Municipio:" with "Seleccionar Localidad/Municipio".

Imagen 87: Formulario de gestión de productos.

El producto tendrá una imagen principal y se puede subir también una galería de imágenes (Véase Imagen 88).

## Galería de Imágenes de Producto

Gestión de Productos

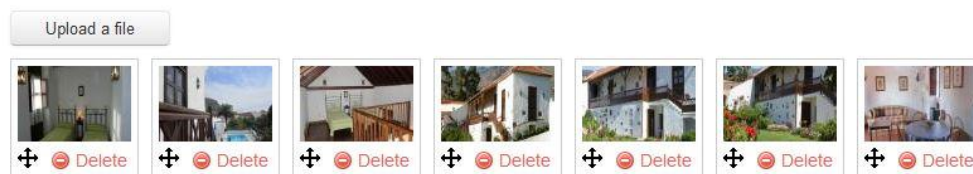


Imagen 88: Galería de imágenes de producto.

### Buscar Producto:

El usuario puede buscar un producto determinado de varias formas, buscando mediante un parámetro concreto (Véase Imagen 89) o realizar una búsqueda general (Véase Imagen 90).



Imagen 89: Búsqueda por parámetros de productos.



Imagen 90: Búsqueda general de productos.

Al usuario se le muestra una lista de productos. El usuario además tiene la posibilidad de editar y eliminar el producto.

### Editar Producto:

El usuario podrá añadir o modificar información del producto (Véase Imagen 91).

- Gestionar Productos
- Gestionar Tarifas
- Gestionar Reservas
- Gestionar Comentarios
- Gestionar Web
- Estadísticas
- Registro de Eventos

### Editar Producto

Nombre\*:


Descripción:

Situada en la Caldera de Tirajana, la mayor caldera volcánica de la isla, y atravesada por el barranco de Tirajana. La zona tiene un enorme interés paisajístico con impresionantes vistas y rincones de ensueño.

La casa situada a la entrada del pueblo, surgió de la rehabilitación de una construcción centenaria que a lo largo de su vida tuvo múltiples usos y destinos, desde secadero de tabaco cultivado en la zona hasta Juzgado de Paz, y como edificación de apoyo a los cultivos que sus propietarios tenían en la zona.

La rehabilitación se acometió respetando en su integridad la estructura original de la casa en tres grandes unidades estructurales, conservando elementos de la arquitectura tradicional canaria: sólidas paredes de piedra roja revestidas con encalados rústicos, amplias puertas y ventanales que dan una extraordinaria luminosidad a los interiores haciéndolas extremadamente acogedoras, y tejado a cuatro aguas en estructura de madera de tea y cubierta de teja canaria. De cada una de esas tres unidades ha surgido una de las Casas Rurales: Casa Rural la Montañeta I, Casa Rural la Montañeta II y Casa Rural la Montañeta III a las que se accede a través de un porche que parte de la parcela de dos mil metros cuadrados ajardinada que comparten.

Imagen:


eliminar

Servicio\*:

Categoría\*:

Imagen 91: Editar producto.

**Ver Producto:**

El usuario puede ver la información asociada al producto (Véaser Imagen 92).

**Gestión de Productos**

#### Registro Producto

Nombre: LA MONTAÑETA

Descripción:

Situada en la Caldera de Tirajana, la mayor caldera volcánica de la isla, y atravesada por el barranco de Tirajana. La zona tiene un enorme interés paisajístico con impresionantes vistas y rincones de ensueño.

La casa situada a la entrada del pueblo, surgió de la rehabilitación de una construcción centenaria que a lo largo de su vida tuvo múltiples usos y destinos, desde secadero de tabaco cultivado en la zona hasta Juzgado de Paz, y como edificación de apoyo a los cultivos que sus propietarios tenían en la zona.

La rehabilitación se acometió respetando en su integridad la estructura original de la casa en tres grandes unidades estructurales, conservando elementos de la arquitectura tradicional canaria: sólidas paredes de piedra roja revestidas con encalados rústicos, amplias puertas y ventanales que dan una extraordinaria luminosidad a los interiores haciéndolas extremadamente acogedoras, y tejado a cuatro aguas en estructura de madera de tea y cubierta de teja canaria. De cada una de esas tres unidades ha surgido una de las Casas Rurales: Casa Rural la Montañeta I, Casa Rural la Montañeta II y Casa Rural la Montañeta III a las que se accede a través de un porche que parte de la parcela de dos mil metros cuadrados ajardinada que comparten.

La Montañeta I y la Montañeta II cuentan con un dormitorio con dos camas sencillas, situado en un cómodo y espacioso attillo de madera a modo de tronja, que la diferencia de la Montañeta III en la que todas las dependencias se encuentran al mismo nivel; además todas ellas poseen una amplia superficie que abarca sala de estar-comedor, cocina y baño completo con plato de ducha y todos los servicios y comodidades de una casa moderna.

Imagen: [597ed-6519ff27fee1cca489b9b762f8bcfe9eb0813334.jpg](#)

Servicio: Alojamiento

Categoría: Casas Rurales

[Volver a la lista](#)

Imagen 92: Ver producto.

### Eliminar Producto:

El usuario puede eliminar el producto. El sistema le muestra un mensaje de alerta de seguridad y el usuario podrá decidir si eliminar el producto o cancelar la operación (Véase Imagen 93).

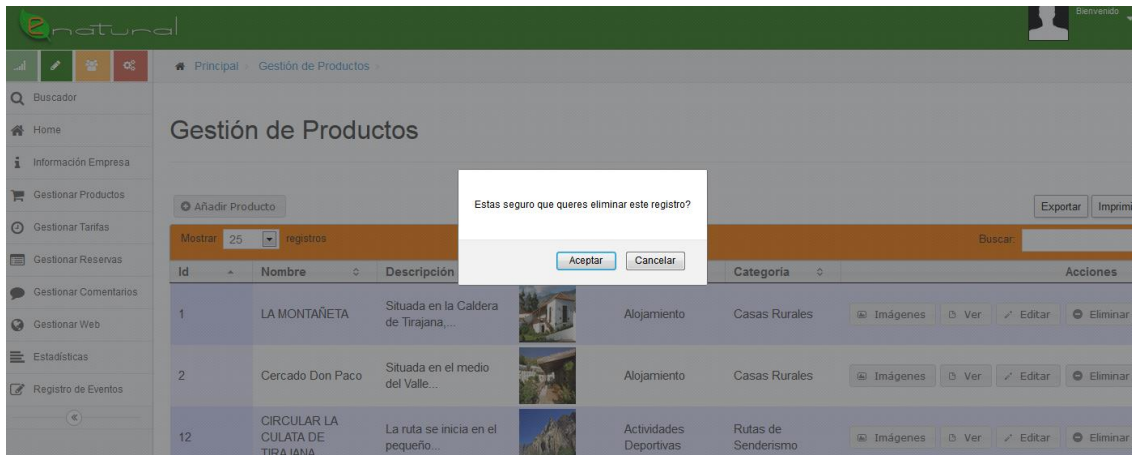


Imagen 93: Eliminar producto.

### Gestionar Tarifas:

El usuario puede asignar tarifa/as a un producto determinado (Véase Imagen 94).

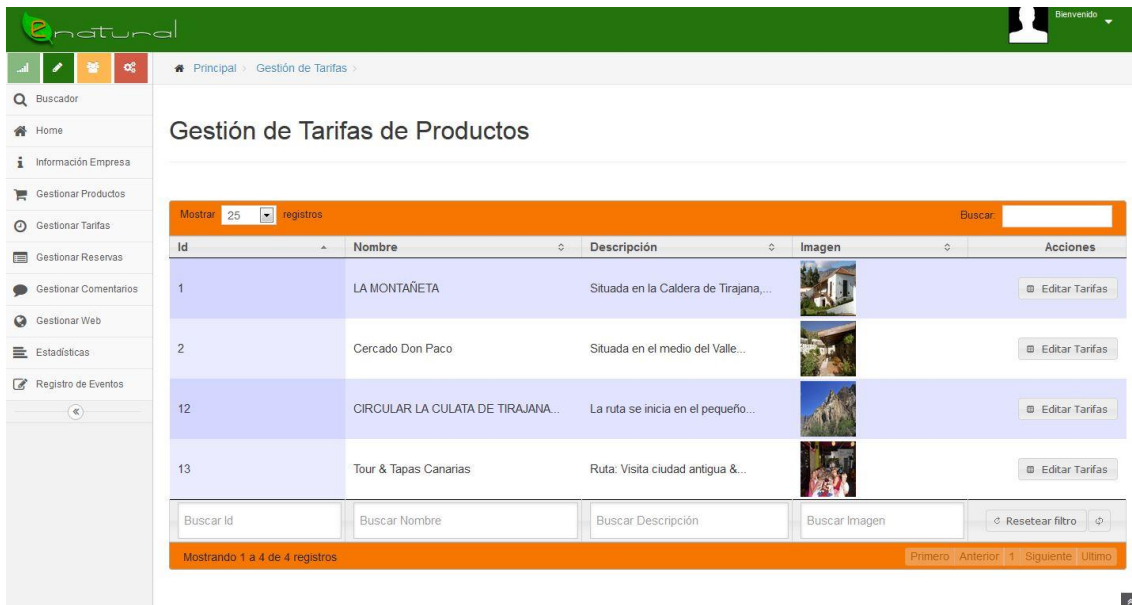


Imagen 94: Gestión de tarifas de productos.

Se muestra un listado con todos los productos disponibles, el usuario selecciona un producto y puede editar las tarifas disponibles o crear unas nuevas (Véase Imagen 95):

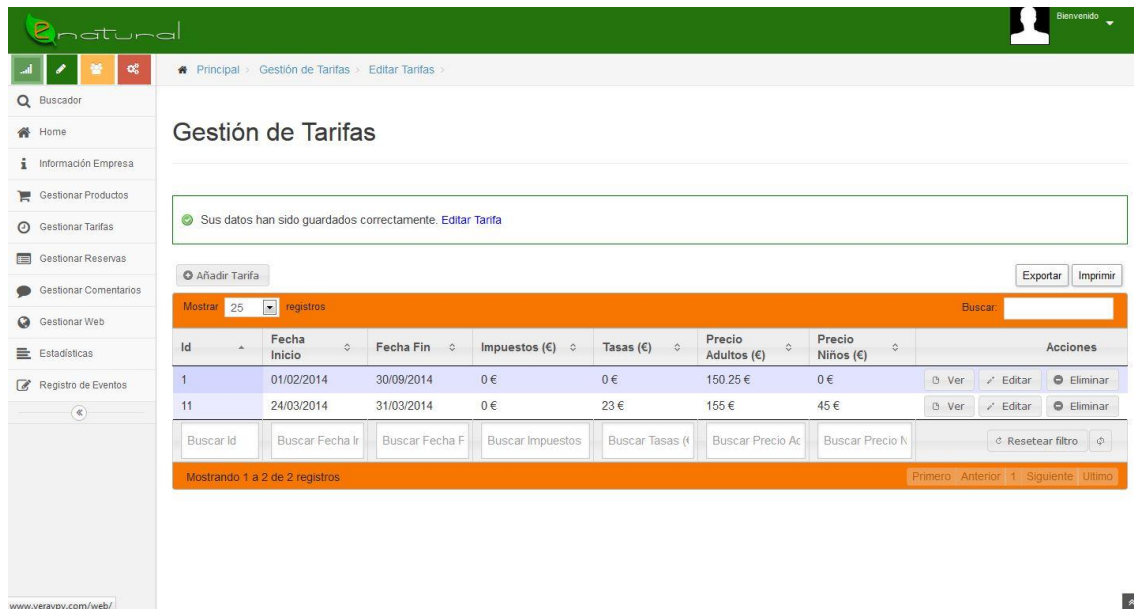
## Acciones



Imagen 95: Editar tarifas de productos.

### Editar Tarifas de producto:

Se muestra un listado con todas las tarifas disponibles para ese producto (Véase Imagen 96). Y una serie de acciones disponibles para la tarifa del producto ( Véase Imagen 97).



www.veravpv.com/web/

Imagen 96: Gestion de tarifas de producto.

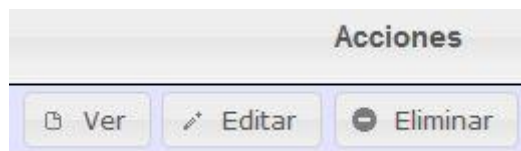


Imagen 97: Acciones de gestión de tarifas de producto.

### Añadir Tarifa:

El usuario podrá añadir tarifas haciendo click en añadir tarifas y rellenando el formulario (Véase Imagen 98 y 99).



Imagen 98: Añadir tarifa I.

## Gestión de Tarifas

Añadir Tarifa	
Name :	<input type="text"/>
Fecha Inicio* :	<input type="text"/> <a href="#">Resetear</a> (dd/mm/yyyy)
Fecha Fin* :	<input type="text"/> <a href="#">Resetear</a> (dd/mm/yyyy)
Impuestos (€) :	<input type="text"/>
Tasas (€) :	<input type="text"/>
Precio Adultos (€)* :	<input type="text"/>
Precio Niños (€) :	<input type="text"/>
<input type="button" value="Guardar"/> <input type="button" value="Guardar y volver a la lista"/> <input type="button" value="Cancelar"/>	

Imagen 99: Añadir tarifa II.

### Editar Tarifa:

El usuario podrá editar la tarifa disponible del producto seleccionado modificando la información del formulario (Véase Imagen 100).

## Gestión de Tarifas

Editar Tarifa	
Fecha Inicio* :	<input type="text" value="01/02/2014"/> <a href="#">Resetear</a> (dd/mm/yyyy)
Fecha Fin* :	<input type="text" value="30/09/2014"/> <a href="#">Resetear</a> (dd/mm/yyyy)
Impuestos (€) :	<input type="text" value="0.0"/>
Tasas (€) :	<input type="text" value="0.0"/>
Precio Adultos (€)* :	<input type="text" value="150.25"/>
Precio Niños (€) :	<input type="text" value="15.4"/>
<input type="button" value="Actualizar cambios"/> <input type="button" value="Actualizar y volver a la lista"/> <input type="button" value="Cancelar"/>	

Imagen 100: Editar Tarifa.

### Eliminar Tarifa:

El usuario podrá eliminar cualquier tarifa disponible del producto seleccionado (Véase Imagen 101)

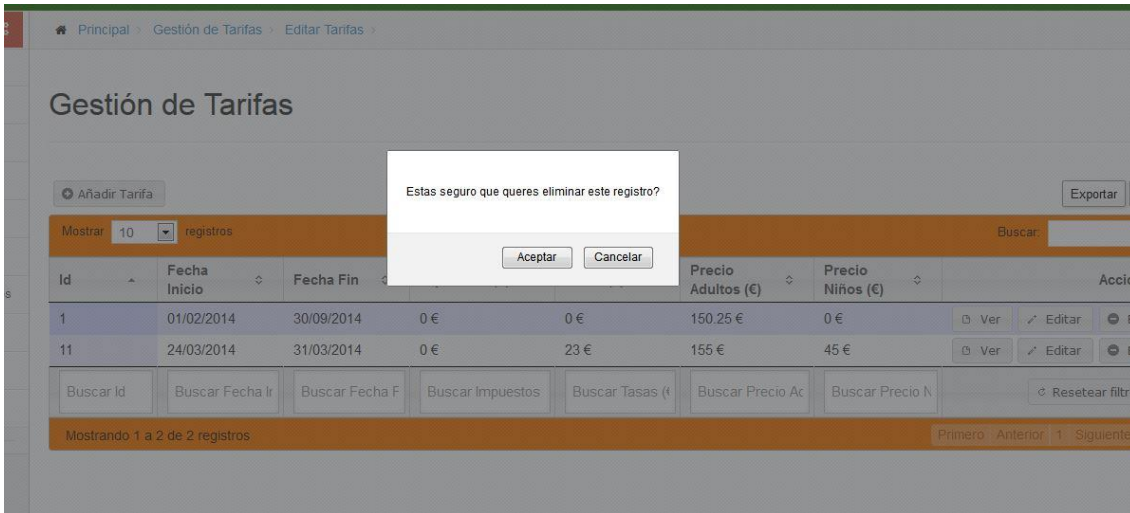


Imagen 101: Eliminar tarifa.

### Ver Tarifa:

También se podrá ver la información asociada a cualquier tarifa de producto (Véase Imagen 102).

### Gestión de Tarifas

Registro Tarifa	
Fecha Inicio :	2014-02-01
Fecha Fin :	2014-09-30
Impuestos (€) :	
Tasas (€) :	
Precio Adultos (€) :	150.25
Precio Niños (€) :	
<input type="button" value="Volver a la lista"/>	

Imagen 102: Ver tarifa.

## Gestionar Reservas:

El usuario gestor podrá en cualquier momento generar reservas de modo manual sobre productos de la empresas (Véase Imagen 103).

Mostrar 25 registros

Id	Nombre	Descripción	Imagen	Acciones
1	LA MONTAÑETA	Situada en la Caldera de Tirajana,...		Reservas
2	Cercado Don Paco	Situada en el medio del Valle...		Reservas
12	CIRCULAR LA CULATA DE TIRAJANA...	La ruta se inicia en el pequeño...		Reservas
13	Tour & Tapas Canarias	Ruta: Visita ciudad antigua &...		Reservas

Mostrando 1 a 4 de 4 registros

Imagen 103: Gestión de reservas.

Para poder realizar operaciones sobre las reservas de los productos el usuario deberá hacer click en el botón “Reservas” (Véase Imagen 104).



Imagen 104: Reservas

Seguidamente el usuario puede seleccionar la reserva o crear una nueva (Véase Imagen 105).

Añadir Reserva

Id. Producto	Id. Usuario	Nombre	Apellidos	Fecha de inicio	Fecha de fin	Información de la reserva	Precio total	Acciones
1	97	sadsad	asdsa	02/03/2014	07/03/2014	cakljsd	751.25	Ver Editar Eliminar
1	1	yeray	p	09/03/2014	11/03/2014		601	Ver Editar Eliminar
1	136	Yeray	Pérez Valiente	12/03/2014	15/03/2014	Información de la reserva proporcionada... Si lo desea puede añadir comentarios...	901.5	Ver Editar Eliminar
2	107	sadas	asdas	02/03/2014	06/03/2014		380.6	Ver Editar Eliminar
2	136	Yeray	Pérez Valiente	11/03/2014	13/03/2014	Comentarios de la reserva proporcionada...	380.6	Ver Editar Eliminar

Mostrando 1 a 5 de 5 registros

Imagen 105: Editar reservas.



### Crear Reserva:

Para poder añadir una reserva el usuario deberá hacer clic en el botón “Añadir Reserva” (Véase Imagen 106).



Imagen 106: Añadir reserva.

Al usuario se le muestra un formulario de creación de reservas de un producto. Se deberán todos los campos del formulario para crear la reserva (Véase Imagen 107).

Añadir Reserva	
Id. Usuario :	<input type="text"/>
Nombre :	<input type="text"/>
Apellidos :	<input type="text"/>
Dirección :	<input type="text"/>
Código Postal :	<input type="text"/>
Tlf. :	<input type="text"/>
País :	<input type="text"/>
Provincia :	<input type="text" value="Seleccionar Provincia"/>
Fecha de emisión :	<input type="text"/> <a href="#">Resetear</a> (dd/mm/yyyy) hh:mm:ss
Fecha de inicio :	<input type="text"/> <a href="#">Resetear</a> (dd/mm/yyyy)
Fecha de fin :	<input type="text"/> <a href="#">Resetear</a> (dd/mm/yyyy)
Precio Adulto :	<input type="text"/>
Precio Niño :	<input type="text"/>

Imagen 107: Formulario de añadir reserva.

Al usuario se le muestra un listado de todas las reservas disponibles de la empresa. El usuario puede seleccionar una reserva y realizar una serie de operaciones (Véase Imagen 108):

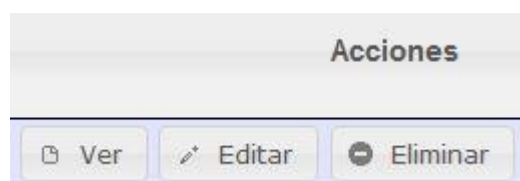


Imagen 108: Acciones de editar reservas.

### Editar Reserva:

El usuario podrá añadir o modificar información de la reserva mediante un formulario (Véase Imagen 109).

Editar Reserva	
Id. Usuario :	<input type="text" value="97"/>
Nombre :	<input type="text" value="sadsad"/>
Apellidos :	<input type="text" value="asdsa"/>
Direccion :	<input type="text" value="asdsad"/>
Código Postal :	<input type="text" value="35017"/>
Tif. :	<input type="text" value="0"/>
País :	<input type="text" value="Murcia, Región de"/>
Provincia :	<input type="text" value="Abánades"/>
Fecha de emisión :	<input type="text" value="25/02/2014 09:01:47"/> <a href="#">Resetear (dd/mm/yyyy) hh:mm:ss</a>
Fecha de inicio :	<input type="text" value="02/03/2014"/> <a href="#">Resetear (dd/mm/yyyy)</a>
Fecha de fin :	<input type="text" value="07/03/2014"/> <a href="#">Resetear (dd/mm/yyyy)</a>

Imagen 109: Formulario de editar reserva.

### Eliminar Reserva:

El usuario puede eliminar la reserva seleccionada. El sistema le muestra un mensaje de alerta de seguridad y el usuario podrá decidir si eliminar la reserva o cancelar la operación (Véase Imagen 110).

### Editar reservas

[+ Añadir Reserva](#)

Mostrar  registros Buscar:

Id. Producto	Id. Usuario	Nombre	Apellidos	de inicio	de fin	Información de la reserva	Precio total	
1	97	sadsad	asdsa	02/03/2014	07/03/2014	cakjsd	751.25	<input type="button" value="Ver"/> <input type="button" value="Edi"/>
1	1	yeray	p	09/03/2014	11/03/2014		601	<input type="button" value="Ver"/> <input type="button" value="Edi"/>
1	136	Yeray	Pérez Valiente	12/03/2014	15/03/2014	Información de la reserva proporcionada...	901.5	<input type="button" value="Ver"/> <input type="button" value="Edi"/>
2	107	sadas	asdas	02/03/2014	06/03/2014	Si lo desea puede añadir comentarios...	360.6	<input type="button" value="Ver"/> <input type="button" value="Edi"/>
2	136	Yeray	Pérez Valiente	11/03/2014	13/03/2014	Comentarios de la reserva proporcionada...	360.6	<input type="button" value="Ver"/> <input type="button" value="Edi"/>

Estas seguro que quieres eliminar este registro?

Imagen 110: Eliminar reserva.

### Visualizar Reserva:

El usuario podrá visualizar un listado de tarifas asociadas al producto (Véase Imagen 111).

Registro Reserva	
Id. Usuario :	136
Nombre :	Yeray
Apellidos :	Pérez Valiente
Dirección :	Calle 123
Código Postal :	35017
Tif :	928123123
País :	Canarias
Provincia :	
Fecha de emisión :	2014-03-09 13:13:31
Fecha de inicio :	2014-03-12

March 2014

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Imagen 111: Visualizar reserva.

### Gestionar Comentarios:

El usuario tiene la opción de ocultar o mostrar comentarios en los productos de la página web de la empresa. Al usuario se le muestra una lista de los productos disponibles (Véase Imagen 112).

Id	Nombre	Descripción	Imagen	Servicio	Categoría	Acciones
1	LA MONTAÑETA	Situada en la Caldera de Tirajana...		Alojamiento	Casas Rurales	Editar Comentarios
2	Cercado Don Paco	Situada en el medio del Valle...		Alojamiento	Casas Rurales	Editar Comentarios
12	CIRCULAR LA CULATA DE TIRAJANA...	La ruta se inicia en el pequeño...		Actividades Deportivas	Rutas de Senderismo	Editar Comentarios
13	Tour & Tapas Canarias	Ruta: Visita ciudad antigua &...		Ocio	Rutas Turísticas	Editar Comentarios

Imagen 112: Gestionar comentarios.

El usuario tiene la posibilidad modificar la visualización de comentarios (Véase Imagen 113).



Imagen 113: Editar comentarios.

Al usuario se le muestra un listado de los comentarios del producto (Véase Imagen 114).

## Editar comentarios



Id	Comentario	Estado	Fecha	Acciones
10	Comentarios...ddd	Visualizar	23/02/2014 - 12:50	✎ Editar
11	Comentario2	Visualizar	23/02/2014 - 12:51	✎ Editar
12	llkklj	Visualizar	23/02/2014 - 13:29	✎ Editar

Mostrar 25 registros      Buscar:

Exportar    Imprimir

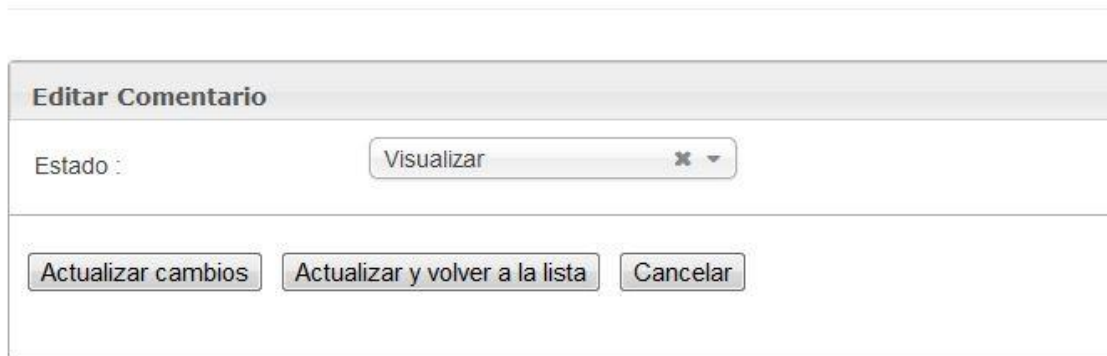
Buscar Id    Buscar Comentario    Buscar Estado    Buscar Fecha   

Mostrando 1 a 3 de 3 registros      Primero    Anterior    1    Siguiente    Ultimo

Imagen 114: Editar comentarios de producto.

El usuario puede modificar el estado de visualización del comentario seleccionado (Véase Imagen 115).

## Editar comentarios



### Editar Comentario

Estado :

Imagen 115: Estado del comentario.

## Gestionar Web:

El usuario podrá configurar el aspecto visual de las páginas web de la empresa (Veáse Imagen 166).

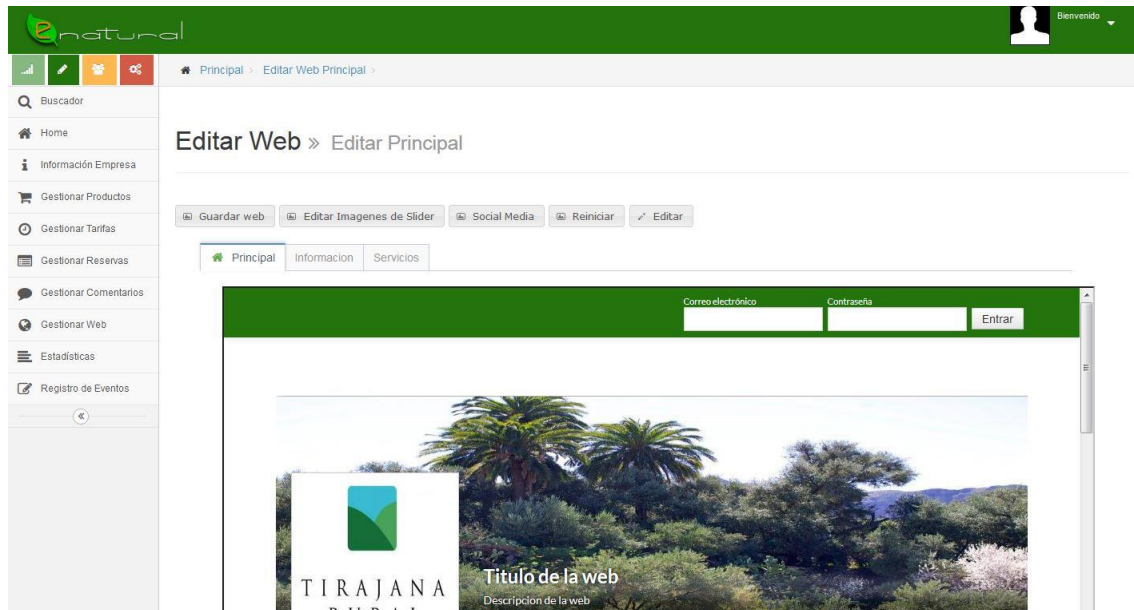


Imagen 116: Editor web.

El usuario gestor podrá realizar distintas acciones en el editor:

Acciones de la página web principal (Véase Imagen 117).



Imagen 117: Menu de acciones de página principal.

### Guardar web:

El usuario podrá guardar los cambios realizados. Hasta que el usuario no haga clic en guardar la página temporal no aplicará los cambios a la página web original.

### Reiniciar:

El usuario puede reiniciar a la página web original en cualquier momento y deshacer los cambios.

### Editar Imágenes de Slider:

El usuario puede editar las imágenes del slider principal( Véase Imagen 118).

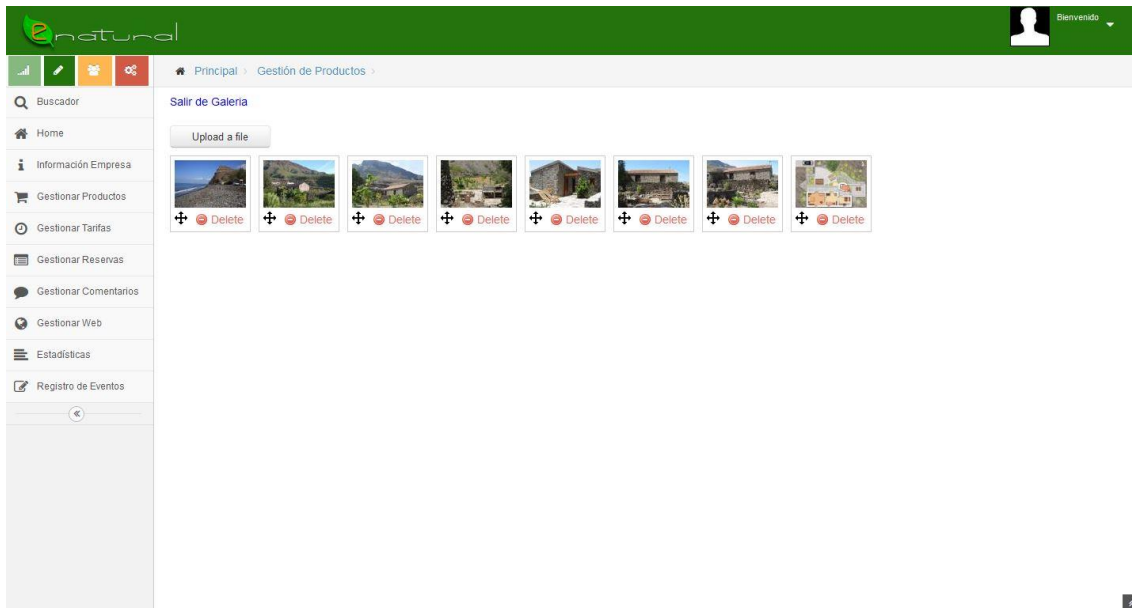


Imagen 118: Galería de imágenes del Slider principal.

### Social Media:

El usuario puede crear, editar, visualizar y eliminar datos de enlaces a las redes sociales más importantes (Véase Imagen 119).

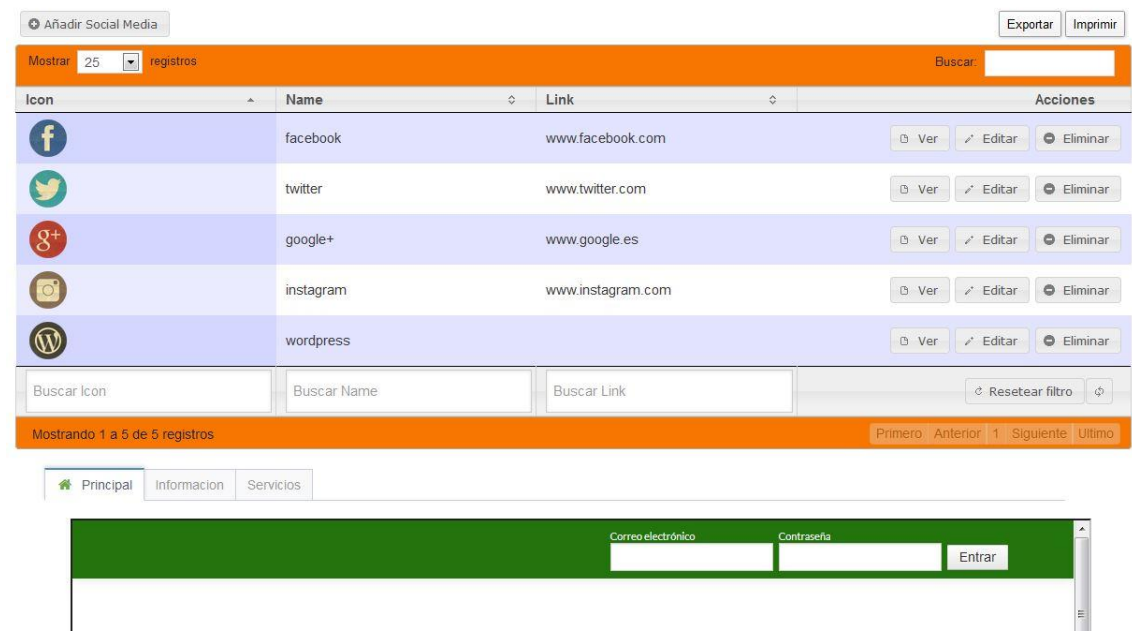


Imagen 119: Social media.

### Editar página principal:

El usuario puede editar los datos de la página principal mediante un formulario (Véase Imagen 120).

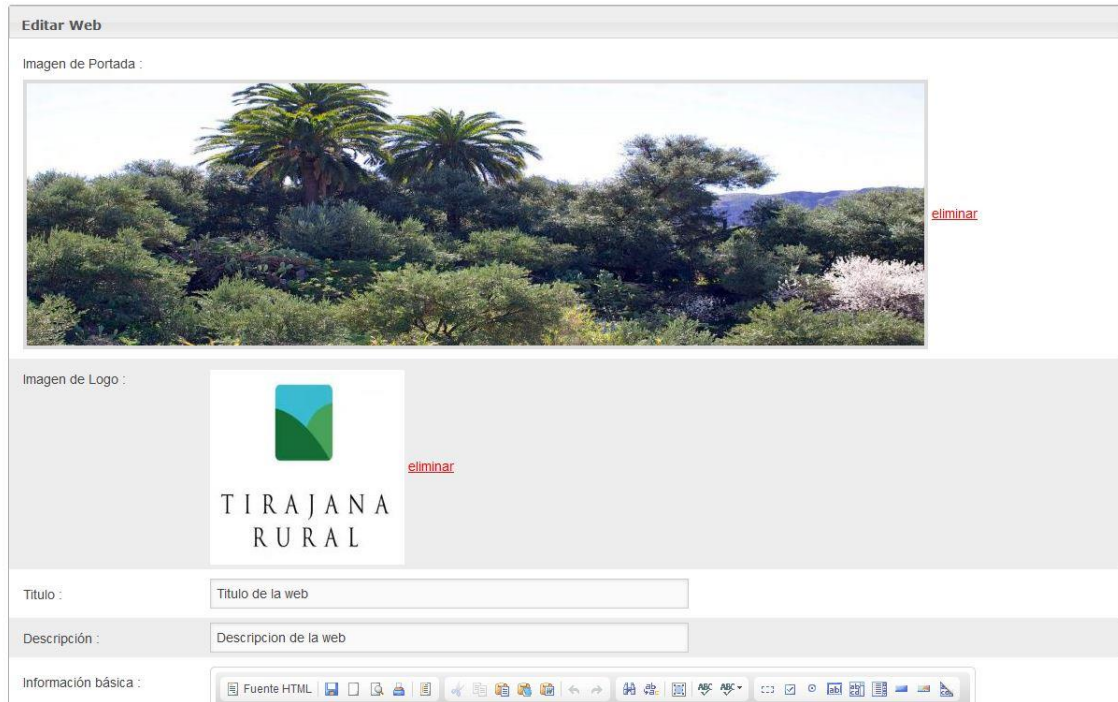


Imagen 120: Formulario de edición de página principal.

Página de información:

El usuario podrá editar la página de información mediante un formulario (Véase Imagen 121).

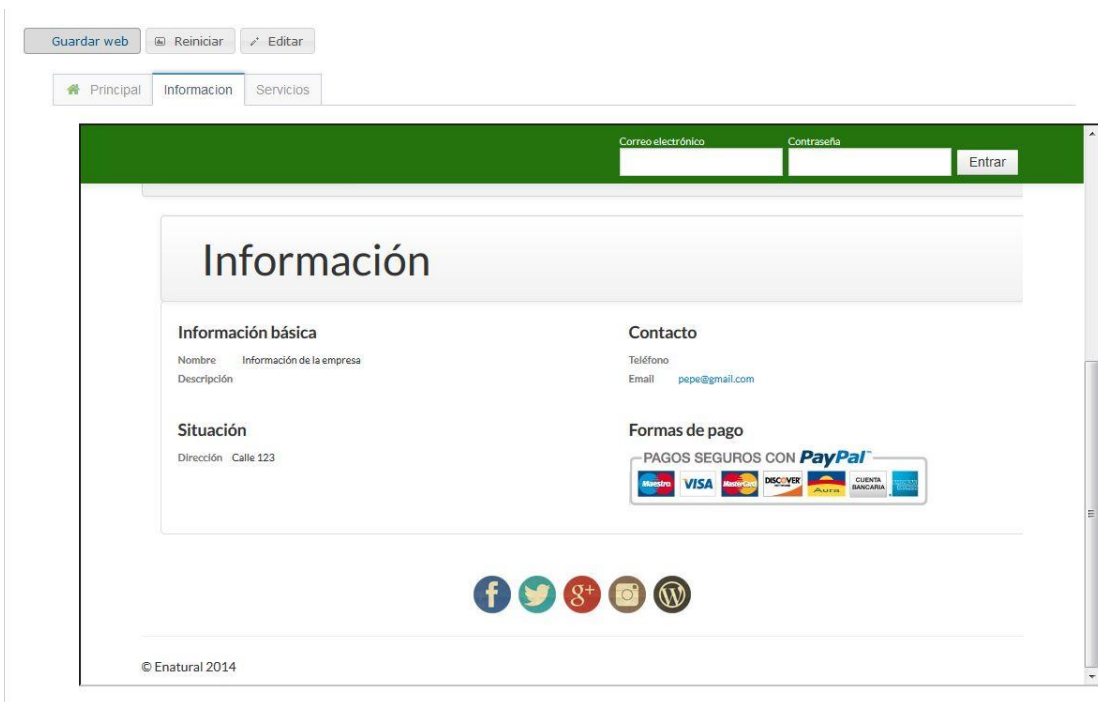


Imagen 121: Edición web de página de información.

### Página de Servicios:

El usuario podrá editar la página de servicios mediante un formulario (Véase Imagen 122).

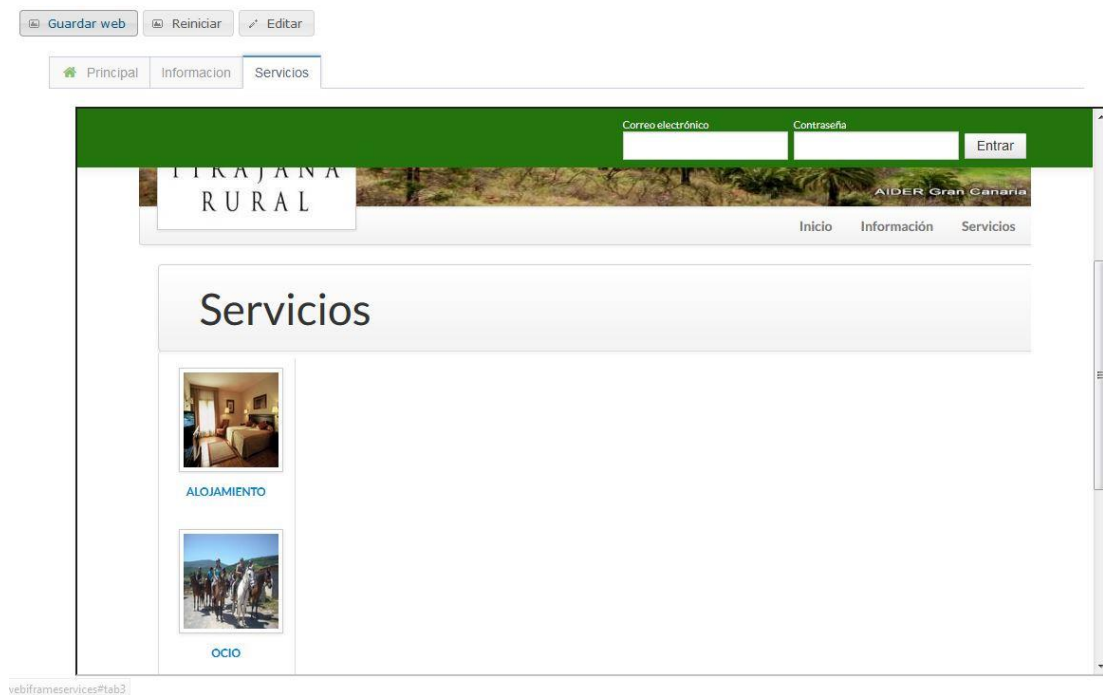


Imagen 122: Edición web de página de servicios.

### Registro de Eventos:

Al usuario se le muestra un listado de todas las actividades realizadas por la empresa en la aplicación por todos los usuarios del sistema. Si lo desea el usuario puede filtrar la información (Véase Imagen 123).

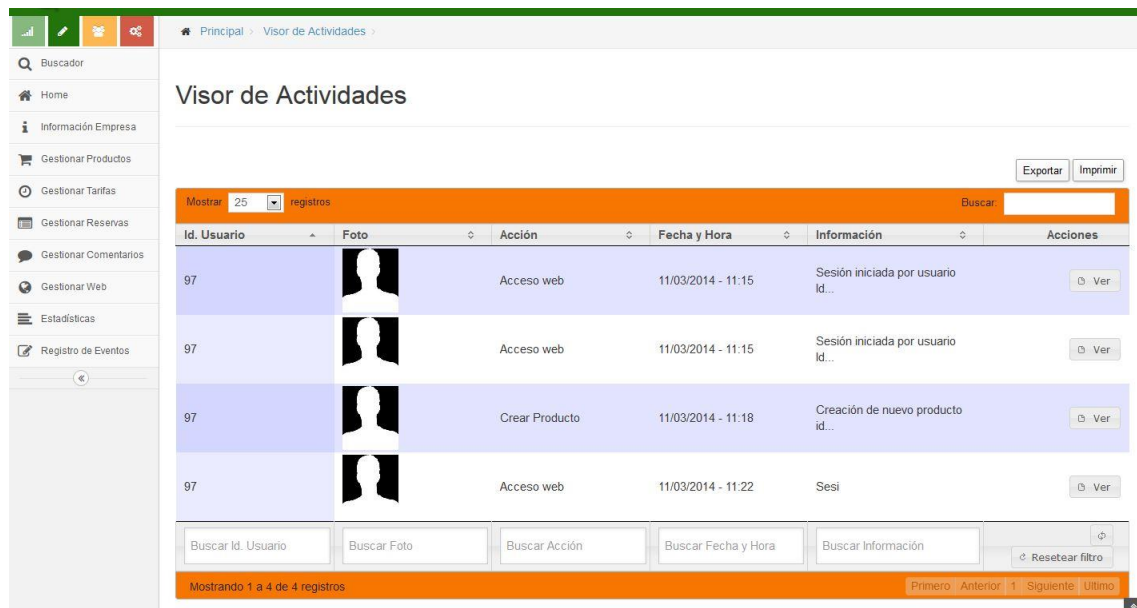


Imagen 123: Visor de actividades.



El usuario puede la ver información ampliada de los registros (Véase Imagen 124).

Registro Actividad	
Foto :	<a href="http://www.yeraypv.com/web/assets/img/anonimo.jpg">http://www.yeraypv.com/web/assets/img/anonimo.jpg</a>
Acción :	Acceso web
Id. Usuario :	97
Fecha y Hora :	2014-03-11 11:15:50
Información :	Sesión iniciada por usuario Id = 97

[Volver a la lista](#)

**Imagen 124: Ver registro de actividad.**







## 9.5 Anexo V: Índice de Imágenes

Imagen 1: <i>Página web Escapada Rural</i> .....	7
Imagen 2: <i>Página web Top Rural</i> .....	8
Imagen 3: <i>Página web Tripadvisor</i> .....	9
Imagen 4: <i>Página web Canarying</i> .....	9
Imagen 5: <i>Página web Yumping</i> .....	10
Imagen 6: <i>Página web DAventuras</i> .....	10
Imagen 7: Método de desarrollo en cascada .....	25
Imagen 8: Diseño arquitectónico .....	52
Imagen 9: Patrón modelo-Vista-Controlador.....	54
Imagen 10: Funcionamiento de los Controladores .....	56
Imagen 11: Estructura de directorios de la aplicación.....	57
Imagen 12: Estructura de directorios de Codeigniter .....	57
Imagen 13: Diagrama de despliegue .....	59
Imagen 14: S.O. Ubuntu Linux.....	94
Imagen 15: S.O. MaCosX .....	95
Imagen 16: PhpMyAdmin.....	96
Imagen 17: Apache.....	96
Imagen 18: Github.....	96
Imagen 19: Bitbucket .....	97
Imagen 20: Microsoft Visio .....	97
Imagen 21: Notepad++.....	97
Imagen 22: Sublime Text 2 .....	98
Imagen 23: Html5.....	98
Imagen 24: Php .....	98
Imagen 25: Ajax.....	99
Imagen 26: Css3 .....	99
Imagen 27: MySql.....	99
Imagen 28: JavaScript.....	100
Imagen 29: JQuery.....	100
Imagen 30: XAMPP.....	101
Imagen 31: TinyMce.....	133
Imagen 32: Editor TinyMce .....	134
Imagen 33: Funcionamiento API de Paypal.....	189
Imagen 34: Grocery Crud .....	190
Imagen 35: Tabla Grocery Crud.....	192
Imagen 36: Procedimiento cargar imagen en Grocery Crud.....	194
Imagen 37: Relación de multiples tablas en Grocery Crud .....	195
Imagen 38: Image Crud .....	204
Imagen 39: Recorte simple.....	206
Imagen 40: Recorte proporcional .....	207
Imagen 41: Recorte Redimensionado .....	207
Imagen 42: Bordos simples .....	207

Imagen 43: Borders redondeados.....	208
Imagen 44: Bordes efectos 3D .....	208
Imagen 45: Marcas de agua con imagen.....	208
Imagen 46: Marcas de agua con texto .....	209
Imagen 47: JRating .....	211
Imagen 48: Enlace de Registrar Usuario .....	223
Imagen 49: Registrar Usuario.....	223
Imagen 50: Mensaje de correo de validación .....	224
Imagen 51: Mensaje de aviso usuario no ha activado la cuenta. ....	224
Imagen 52: Mensaje de correo de activación de cuenta de usuario. ....	224
Imagen 53: Iniciar sesión I.....	225
Imagen 54: Iniciar sesión II.....	225
Imagen 55: Valoraciones de la web sin iniciar sesión. ....	226
Imagen 56: Valoraciones de la web iniciada la sesión. ....	227
Imagen 57: Editar valoraciones.....	227
Imagen 58: Valorar producto. ....	228
Imagen 59: Reservar producto.....	229
Imagen 60: Formulario de reserva de producto. ....	229
Imagen 61: Realizar reserva de producto. ....	230
Imagen 62: Email validación de reserva.....	231
Imagen 63: Mis reservas. ....	232
Imagen 64: Reservas realizadas por el usuario.....	232
Imagen 65: Comprar.....	232
Imagen 66: Paypal I.....	233
Imagen 67: Paypal II.....	233
Imagen 68: Paypal III.....	234
Imagen 69: Mis compras.....	234
Imagen 70: Mis compras realizadas.....	235
Imagen 71: Información de registro de compra.....	235
Imagen 72: Home .....	236
Imagen 73: Principal de usuario.....	236
Imagen 74: Cerrar sesión de usuario. ....	237
Imagen 75: Cerrar sesión gestor. ....	238
Imagen 76: Pantalla principal del usuario gestor.....	239
Imagen 77: Menú .....	239
Imagen 78: Buscador.....	240
Imagen 79: Pantalla de control.....	240
Imagen 80: Gestionar información de empresa. ....	241
Imagen 81: Acciones de Gestionar información de empresa. ....	241
Imagen 82: Registro de empresa. ....	241
Imagen 83: Editar información de empresa.....	242
Imagen 84: Gestionar productos.....	242
Imagen 85: Acciones de Gestionar productos. ....	243
Imagen 86: Añadir producto. ....	243
Imagen 87: Formulario de gestión de productos.....	243

Imagen 88: Galería de imágenes de producto.....	244
Imagen 89: Búsqueda por parámetros de productos.....	244
Imagen 90: Búsqueda general de productos.....	244
Imagen 91: Editar producto.....	245
Imagen 92: Ver producto.....	245
Imagen 93: Eliminar producto.....	246
Imagen 94: Gestión de tarifas de productos.....	246
Imagen 95: Editar tarifas de productos.....	247
Imagen 96: Gestion de tarifas de producto.....	247
Imagen 97: Acciones de gestión de tarifas de producto.....	247
Imagen 98: Añadir tarifa I.....	247
Imagen 99: Añadir tarifa II.....	248
Imagen 100: Editar Tarifa.....	248
Imagen 101: Eliminar tarifa.....	249
Imagen 102: Ver tarifa.....	249
Imagen 103: Gestión de reservas.....	250
Imagen 104: Reservas.....	250
Imagen 105: Editar reservas.....	250
Imagen 106: Añadir reserva.....	251
Imagen 107: Formulario de añadir reserva.....	251
Imagen 108: Acciones de editar reservas.....	251
Imagen 109: Formulario de editar reserva.....	252
Imagen 110: Eliminar reserva.....	252
Imagen 111: Visualizar reserva.....	253
Imagen 112: Gestionar comentarios.....	253
Imagen 113: Editar comentarios.....	254
Imagen 114: Editar comentarios de producto.....	254
Imagen 115: Estado del comentario.....	254
Imagen 116: Editor web.....	255
Imagen 117: Menu de acciones de página principal.....	255
Imagen 118: Galería de imágenes del Slider principal.....	256
Imagen 119: Social media.....	256
Imagen 120: Formulario de edición de página principal.....	257
Imagen 121: Edición web de página de información.....	257
Imagen 122: Edición web de página de servicios.....	258
Imagen 123: Visor de actividades.....	258
Imagen 124: Ver registro de actividad.....	259

## 9.6 Anexo VI: Índice de figuras completo

Figura 1: Modelo de dominio de usuario administrador .....	28
Figura 2: Modelo de dominio de usuario cliente .....	29
Figura 3: Modelo de dominio de usuario gestor .....	30
Figura 4: Diagrama entidad relación enatural_db. ....	32
Figura 5: Diagrama entidad relación db_statistics .....	33
Figura 6: Modelo relacional enatural_db.....	35
Figura 7: Modelo relacional db_statistics .....	36
Figura 8: Actores .....	38
Figura 9: Caso de uso Registrar .....	42
Figura 10: Casos de uso rol cliente.....	42
Figura 11: Caso de uso Gestionar Comentarios .....	43
Figura 12: Caso de uso Visualizar Registro.....	43
Figura 13: Caso de uso Gestionar Producto .....	44
Figura 14: Caso de uso Gestionar Categoria .....	45
Figura 15: Caso de uso Gestionar Servicio .....	46
Figura 16: Caso de uso Gestionar Usuarios.....	46
Figura 17: Prototipo de interfaz de usuario gestor .....	48
Figura 18: Prototipo de interfaz de producto .....	49
Figura 19: Prototipo de interfaz de Editor Web.....	50
Figura 20: Prototipo de página web principal.....	51
Figura 21: Diagrama de clases Categorías.....	63
Figura 22: Diagrama de clases Home .....	64
Figura 23: Diagrama de clases Registro de eventos.....	65
Figura 24: Diagrama de clases comentarios.....	67
Figura 25: Diagrama de clases productos .....	68
Figura 26: Diagrama de clases servicios.....	69
Figura 27: Diagrama de clases sites .....	70
Figura 28: Diagrama de clases editor web .....	71
Figura 29: Diagrama de clases webeditorinfo.....	72
Figura 30: Diagrama de clases webeditorservices .....	73
Figura 31: Diagrama de clases website .....	74
Figura 32: Diagrama de clases tarifa .....	75
Figura 33: Diagrama de clases login .....	76
Figura 34: Diagrama de clases reserva.....	77
Figura 35: Diagrama de clases saveindex.....	78
Figura 36: Diagrama de clases usuario.....	79
Figura 37: Diagrama de clases Paypal .....	80
Figura 38: Diagrama de clases guardar servicio.....	81
Figura 39: Diagrama de clases referencias.....	82
Figura 40: Diagrama de clases saveinfo .....	83
Figura 41: Diagrama de clases login.....	84
Figura 42: Diagrama de clases login2 .....	85



Figura 43: Diagrama de secuencia crear usuario .....	86
Figura 44: Diagrama de secuencia crear servicio .....	87
Figura 45: Diagrama de secuencia crear reserva .....	87
Figura 46: Diagrama de secuencia insertar comentario .....	88
Figura 47: Diagrama de secuencia comprar producto .....	89

## 9.7 Anexo VII: Índice de tablas completo

Tabla 1: Planificación y Temporización .....	27
Tabla 2: Listado de actores y roles .....	39
Tabla 3: Listado de actores y sus objetivos .....	41
Tabla 4: Lista de Casos de uso .....	47
Tabla 5: Descripción caso de uso Registrarse .....	102
Tabla 6: Descripción caso de uso Visualizar Web .....	103
Tabla 7: Descripción caso de uso Visualizar Producto .....	103
Tabla 8: Descripción caso de uso Iniciar Sesión .....	104
Tabla 9: Descripción caso de uso Cerrar Sesión .....	104
Tabla 10: Descripción caso de uso Realizar Compra .....	105
Tabla 11: Descripción caso de uso Realizar Reserva .....	106
Tabla 12: Descripción caso de uso Insertar Comentario .....	107
Tabla 13: Descripción caso de uso Visualizar Comentarios .....	108
Tabla 14: Descripción caso de uso Visualizar Registro .....	109
Tabla 15: Descripción caso de uso Crear Tarifa .....	110
Tabla 16: Descripción caso de uso Editar Tarifa .....	111
Tabla 17: Descripción caso de uso Eliminar Tarifa .....	112
Tabla 18: Descripción caso de uso Buscar Tarifa .....	113
Tabla 19: Descripción caso de uso Crear Reserva .....	114
Tabla 20: Descripción caso de uso Eliminar Reserva .....	115
Tabla 21: Descripción caso de uso Buscar Reserva .....	116
Tabla 22: Descripción caso de uso Crear Producto .....	117
Tabla 23: Descripción caso de uso Editar Producto .....	118
Tabla 24: Descripción caso de uso Eliminar Producto .....	119
Tabla 25: Descripción caso de uso Buscar Producto .....	120
Tabla 26: Descripción caso de uso Crear Categoría .....	121
Tabla 27: Descripción caso de uso Editar Categoría .....	122
Tabla 28: Descripción caso de uso Eliminar Categoría .....	123
Tabla 29: Descripción caso de uso Buscar Categoría .....	124
Tabla 30: Descripción caso de uso Crear Usuario .....	125
Tabla 31: Descripción caso de uso Editar Usuario .....	126
Tabla 32: Descripción caso de uso Eliminar Usuario .....	127
Tabla 33: Descripción caso de uso Buscar Usuario .....	128
Tabla 34: Descripción caso de uso Crear Servicio .....	129
Tabla 35: Descripción caso de uso Editar Servicio .....	130
Tabla 36: Descripción caso de uso Eliminar Servicio .....	131
Tabla 37: Descripción caso de uso Buscar Servicio .....	132

## 10 Bibliografía

Luke Welling, Laura Thomson. "Desarrollo web con PHP y MySQL". Anaya Multimedia.

Matt Doyle. "PHP Práctico". Anaya Multimedia.

Julie C. Meloni. "HTML5, CSS3 y JavaScript". Anaya Multimedia.

David Flanagan, "jQuery". O'Really.

Douglas Crockford. "JavaScript The Good Parts". O'Really.

Zachary Kessin. "HTML5 Applications". O'Really.

PHP:

<http://www.php.net>

Grocery Crud:

<http://www.grocerycrud.com/>

Codeigniter:

<http://ellislab.com/codeigniter/user-guide/>

Image Moo:

[http://www.matmoo.com/digital-dribble/codeigniter/image\\_moo/](http://www.matmoo.com/digital-dribble/codeigniter/image_moo/)

Image Crud:

<http://www.grocerycrud.com/image-crud>

Ajax:

<http://www.w3schools.com/Ajax/>

jQuery:

<http://api.jquery.com/>

jQuery UI:

<http://jqueryui.com/>

Bootstrap:

<http://getbootstrap.com/2.3.2/index.html>

<http://fortawesome.github.io/Font-Awesome/>

Javascript:

<http://www.w3schools.com/js/DEFAULT.asp>

<http://www.w3.org/standards/webdesign/script.html>

Paypal:

<https://developer.paypal.com/webapps/developer/applications/accounts>

JRating:

<http://demos.myjqueryplugins.com/jrating/>

Email Codeigniter:

<http://www.renato.16mb.com/tutorial-codeigniter-class-e-mail>

HTML5:

<http://www.w3.org/standards/webdesign/htmlcss.html>

[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

CSS:

[http://www.w3schools.com/css/css3\\_intro.asp](http://www.w3schools.com/css/css3_intro.asp)