



Universidad Nacional Autónoma de Nicaragua

Unan-León

FACULTAD DE CIENCIAS Y TECNOLOGÍAS
INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Desarrollo de una aplicación Web publicitaria con implementación de Yii framework y PostgreSQL.

Tesis para optar al título de Ingenieros en Sistemas de Información

Integrantes:

- Horacio Javier Rodríguez Altamirano
- Cristian Adán Reyes Blanco
- Daniel Enrique Pérez Sánchez

Tutor: Msc. Aldo René Martínez Delgadillo.



DEDICATORIA

La concepción de este proyecto está dedicada a mis padres, pilares fundamentales en mi vida. Sin ellos, jamás hubiese podido conseguir lo que hasta ahora he logrado. Su tenacidad y lucha insaciable han hecho de ellos el gran ejemplo a seguir y destacar, no solo para mí, sino para mis hermanos y familia en general. También dedico este proyecto a mis hermanos, los que me alentaron desde el principio, y durante todo el tiempo en que se elaboró este proyecto, que sin ellos, no hubiese podido ser posible.

Horacio Javier Rodríguez Altamirano



DEDICATORIA

Dedico este proyecto de tesis a Dios y a mis padres. A Dios porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar, a mis padres, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento. Depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad. Es por ellos que soy lo que soy ahora. Los amo con mi vida.

Cristian Adán Reyes Blanco



DEDICATORIA

Dedico este trabajo a Dios, Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos. A mis padres y hermanos, por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, por los ejemplos de perseverancia, los que me ha infundado siempre y por el valor mostrado para salir adelante. A mis amistades, por permitirme contar con su confianza y por facilitarme en muchas ocasiones las herramientas que necesite para poder elaborar en conjunto con mis compañeros este proyecto.

Daniel Enrique Pérez Sánchez



AGRADECIMIENTO

Los resultados de este proyecto, están dedicados a todas aquellas personas que, de alguna forma, son parte de su culminación. Mis sinceros agradecimientos están dirigidos a mis compañeros de trabajo en este proyecto, quienes fueron de mucha ayuda durante el desarrollo de este trabajo. A nuestras familias por siempre brindarnos su apoyo, tanto sentimental, como económico. Pero, principalmente nuestros agradecimientos están dirigidos hacia la excelentísima autoridad de nuestro tutor y profesores, sin los cuales no hubiésemos podido salir adelante.

Horacio Javier Rodríguez Altamirano



AGRADECIMIENTO

Este proyecto es el resultado del esfuerzo conjunto de todos los que formamos el grupo de trabajo. Por esto agradezco a mi tutor Msc. Aldo Rene Martínez Delgadillo, mis compañeros de tópicos, Horacio Javier Altamirano y Daniel Enrique Pérez Sánchez y mi persona, quienes a lo largo de este tiempo han puesto a prueba sus capacidades y conocimientos en el desarrollo de este proyecto el cual ha finalizado llenando todas nuestras expectativas. A mis padres quienes a lo largo de toda mi vida han apoyado y motivado mi formación académica, a quienes creyeron en mí en todo momento y no dudaron de mis habilidades, gracias a su paciencia y enseñanza y finalmente un eterno agradecimiento a esta prestigiosa universidad la cual abrió sus puertas a jóvenes como nosotros, preparándonos para un futuro competitivo y formándonos como personas de bien.

Cristian Adán Reyes Blanco.



AGRADECIMIENTO

En primer lugar a Dios por haberme guiado por el camino de la felicidad hasta ahora; en segundo lugar a cada uno de los que son parte de mi familia a mi padre E. B. Pérez, mi madre M. Sánchez,

a mis hermanos; por siempre haberme dado su fuerza y apoyo incondicional que me han ayudado y llevado hasta donde estoy ahora. Por último a mis compañeros de tesis porque en esta armonía grupal lo hemos logrado y a mi tutor de tesis quién nos ayudó en todo momento, Msc. Aldo Rene Martínez Delgadillo.

Daniel Enrique Pérez Sánchez



1. RESUMEN	10
2. INTRODUCCIÓN	11
2.1 ANTECEDENTES	11
2.2 PLANTEAMIENTO DEL PROBLEMA	12
2.3 JUSTIFICACIÓN	13
2.3.1 Alcance Mediante el uso de esta aplicación se implementaran los siguientes aspectos:	13
2.4 OBJETIVOS	14
2.4.1 OBJETIVO GENERAL	14
2.4.2 OBJETIVOS ESPECÍFICOS	14
3. MARCO TEÓRICO.....	15
3.1 TIPOS DE APLICACIONES WEB.....	15
3.2 COMERCIO ELECTRÓNICO	16
3.2.1 Tipos de comercio electrónico.....	17
3.2.2 Soluciones para la implementación de un sistema web e-commerce.....	18
3.2.3 Elementos de una tienda online	19
3.3 HERRAMIENTAS DE DESARROLLO	20
3.3.1 HTML5.....	20
3.3.2 CSS3.....	20
3.3.3 PHP	20
3.3.4 Responsive web design (Diseño web adaptable)	20
3.3.5 Bootstrap	21
3.3.6 Patrón de diseño Modelo Vista Controlador.....	21
3.3.7 Yii Framework	23
3.3.8 API Google Maps	24
3.3.9 PostgreSQL.....	26
4. DISEÑO METODOLÓGICO	29
4.1 DESCRIPCIÓN GENERAL DEL PROYECTO	29
4.2 ETAPAS DEL TRABAJO	30
4.3 MATERIALES	31
4.3.1 Hardware.....	31
4.3.2 Software	32
5. RESULTADOS	33
5.1 DESPLIEGUE DEL SISTEMA	33
5.2 DISEÑO DEL SISTEMA	34
5.2 FUNCIONAMIENTO DEL SISTEMA	34
6. ASPECTOS FINALES.....	40
6.1 CONCLUSIONES	40
6.2 RECOMENDACIONES.....	40
7. BIBLIOGRAFÍA	41
8. ANEXOS	42



INDICE DE ILUSTRACIONES

Ilustración 1: Modalidades de comercio electrónico.....	17
Ilustración 2: Versiones soportadas por los diferentes navegadores web	21
Ilustración 3: Arquitectura MVC. Tomada POO y MVC en PHP por Eugenia Bahit.	22
Ilustración 4: Ejemplo de ubicación aplicando google maps.	25
Ilustración 5: Resultado de google maps.	25
Ilustración 6: Manera general los componentes más importantes en un sistema PostgreSQL	27
Ilustración 7: Ciclo de vida Clásico. (Ian Sommerville, 2011).	29
Ilustración 8: Despliegue	33
Ilustración 9: Despliegue 2.	34
Ilustración 10: Autenticación de usuario.	35
Ilustración 11: Crear un nuevo perfil.....	36
Ilustración 12: Perfil de usuario.....	37
Ilustración 13: Actualizar perfil de usuario.	38
Ilustración 14: Gestión de perfil de usuario.	39
Ilustración 15: Registrar Productos.	39
Ilustración 16: Interfaces Hardware	43
Ilustración 17: Modelo Relacional	46
Ilustración 18: Diagrama de casos de usos.	47
Ilustración 19: Diagrama de Clases. Clase User.	48
Ilustración 20: Diagrama de Secuencia de Registro de perfil.	49
Ilustración 21: Diagrama de Secuencia de Ingreso al Sistema.	49
Ilustración 22: Diagrama de Secuencia de Registrar Datos.	50
Ilustración 23: Diagrama de Secuencia de Actualizar Datos.	50
Ilustración 24: Diagrama de Secuencia de Eliminar Datos.	51
Ilustración 25: Diagrama de Secuencia de Buscar Datos.	51
Ilustración 26: Página de Bienvenida de Yii Framework.....	52
Ilustración 27: Estructura básica.....	53
Ilustración 28: Configuración del módulo Gii.....	55
Ilustración 29: Especificando rutas de alias.	56
Ilustración 30: Configuración conexión con el SGBD.	57
Ilustración 31: Login para realizar un CRUD con Gii.....	57
Ilustración 32: Opciones para generar código.	58
Ilustración 33: Generador de modelo.	59
Ilustración 34: Generador CRUD de Gii.	60
Ilustración 35: Mensaje satisfactorio de creación CRUD.	61
Ilustración 36: Vista del listado de todos los usuarios.	61
Ilustración 37: Vista ver un usuario.....	62
Ilustración 38: Vista de crear un nuevo usuario.....	62



Ilustración 39: Vista gestión de usuarios..... 62
Ilustración 40: Vista actualizar datos usuario..... 63
Ilustración 41: Modulo de usuarios. 64



1. Resumen

En el mercado se encuentran múltiples frameworks y herramientas que permiten desarrollar de manera eficiente una aplicación web, muchas de ellas se encuentran bien estructuradas y presentan ventajas entre ellas.

De ahí este proyecto que surge con la finalidad de crear una aplicación web de comercio electrónico en donde se logre apreciar la funcionalidad que nos brinda a los desarrolladores el uso de las nuevas herramientas existentes en el mercado al momento de crear una aplicación web de esta complejidad. Como grupo hemos optado por mostrar Yii Framework como una posible alternativa, desarrollando una pequeña aplicación web de comercio electrónico publicitario, ampliando nuestro nivel de competitividad como desarrolladores ante la constante actualización de las herramientas web existentes.



2. Introducción

2.1 Antecedentes

En la actualidad el avance de las tecnologías y la constante actualización de los lenguajes de programación han sido de gran influencia en el incremento de aplicaciones web a la hora de crear sitios destinados a la compra, venta y publicación de productos en línea (e-commerce).

Los sistemas web de comercio electrónico están diseñados para brindar múltiples servicios destinados a diferentes propósitos, algunos sistemas están diseñados solo para realizar publicaciones de productos, carritos de compra, compras con envíos, y otros son integrales lo que permite abarcar varios aspectos.

Como antecedentes de nuestro proyecto podemos destacar las siguientes aplicaciones desarrolladas:

El primer antecedente se elaboró un trabajo final de grado titulada “**Aplicación de gestión de playas**” optando por el título de grado de ingeniería en informática en la Universidad Universitat Jaume I, Comunidad Valenciana, España. Elaborado por el A. Daniel Higuera Goold, el documento contiene un total de 82 páginas. Este proyecto se centra en la gestión de las diferentes playas de la empresa Mediterránea Grupo, en el que se tiene que gestionar una gran cantidad de eventos, como pueden ser meteorológico o las partes de accidentes. En la creación de esta aplicación hicieron uso del framework Yii y ha sido diseñada para tener una interfaz adaptable a multitud de dispositivo, para ello se ha utilizado el front-end responsive framework Bootstrap. El trabajo fue dirigido por el tutor académico Sr. José Luis Llopis Borrás y supervisado por el Sr. Héctor Segarra Castillo, en el año 2014.

El desarrollo de una aplicación web e-commerce es un tema que se ha abordado con anterioridad en la Universidad:

El segundo antecedente se elaboró una tesis titulada “**Comercio Electrónico**”, elaborado por los Br. Luis Salvador Matamoros Cáceres, Br. Lester Israel Gutiérrez Vílchez y Br. William José Osejo Palma. El documento contiene un total de 70 páginas y se crea una página web para poder vender en internet. Esta página pretendía que la empresa que adquiriera esta Aplicación pudiera promover y vender sus productos. En la creación de estas aplicaciones hicieron uso de ASP (Pages Active Server) la cual es una tecnología diseñada para la creación de páginas dinámicas. El trabajo fue dirigido por el Msc. Martin Ibarra Padilla, en el año 2003.

El tercero antecedente que destaca, es la tesis “**Propuestas de prácticas de laboratorio para la asignatura de Comercio Electrónico**”, elaborado por los Br. Elving David González Olivas, Br. Gabriel Ramón Rodríguez Jirón y Br. Kenia del Carmen Miranda Ruiz. El documento consta de un total de 473 páginas en las que se proponen una serie de prácticas de laboratorios enfocadas a la carrera de Ingeniería en Telemática de la UNAN-León, perteneciente al componente curricular de Comercio Electrónico en la asignatura de ELECTIVA VI, perteneciente al ciclo 8 del pensum 2011. Con la finalidad de instalar e implementar diferentes plataformas web (OpenCart, Magento, Os-Commerce) para la simulación de un sitio de comercio electrónico. El trabajo fue dirigido por el MSC. Aldo Rene Martínez Delgadillo, en el año 2016.

En vista que gran mayoría de las aplicaciones generalmente se desarrollan bajo un lenguaje de programación nativo, estas se limitan a ser utilizadas a través de una sola plataforma (Android, iOS, Windows phone). Tomando en cuenta lo antes mencionado se optó por crear una aplicación bajo un entorno web por ser multiplataforma y de bajo coste.



2.2 Planteamiento del Problema

El incremento del e-commerce ha sido evidente en nuestro país ya que ha demostrado ser de gran utilidad en cuanto a la promoción de productos, pero a su vez esto plantea una problemática para aquellos usuarios que tienen la necesidad de realizar alguna publicación y por diferentes motivos se ven sujetos a ofrecer sus productos en las distintas redes sociales o realizar pagos en páginas destinada a brindar este servicio, lo que implica en la mayoría de los casos altos costos monetarios.

Es difícil encontrar una aplicación e-commerce publicitario que ayude a los usuarios a ofrecer o publicar productos de una forma eficiente, ágil e intuitiva, que permita realizar publicaciones siempre y cuando la persona esté registrada en la aplicación sin costo alguno.

Sumado a lo anterior es necesario utilizar una herramienta de desarrollo que facilite crear una aplicación web publicitaria. Como una propuesta para este proyecto decidimos emplear el Framework Yii debido a que existen una variedad de aplicaciones webs con diferentes fines, pero ninguna aplicación destinada a este propósito desarrollada con este framework, el cual cuenta con una comunidad en constante crecimiento y una serie de extensiones que facilitan el trabajo en cuanto a desarrollo, permitiendo su integración al proyecto de manera rápida y a bajos costos tanto en recursos hardware y software, razón por la que hemos decidido desarrollar nuestra aplicación empleando esta tecnología.

Por tales razones se plantea las siguientes interrogantes:

¿Es posible desarrollar una aplicación web publicitaria que permita gestionar datos de manera eficaz con la implementación de Yii Framework?

¿Se puede implementar Yii-Framework para la creación y desarrollo del backend de una aplicación web?

¿Qué herramientas podemos incorporar para el desarrollo de una aplicación web que nos facilite el diseño de una interfaz responsiva adaptable a diversos dispositivos como PCs, móviles y tablets?

¿Qué gestor de base de datos usar para el almacenamiento de toda la información que se manipulará a través de la aplicación web?

¿Es posible integrar Google Maps API para geo referenciar las publicaciones realizadas y facilitar los parámetros de búsqueda establecidos por los usuario?



2.3 Justificación

El desarrollo de una aplicación web e-commerce publicitario sería de gran utilidad para la gran cantidad de personas o usuarios que desean promocionar sus productos en internet y que se ven limitados a simples publicaciones en las redes sociales o bien elevar costos al contratar los servicios de sitios que quizá no brinden todas las funcionalidades que se desean.

Satisfacer las necesidades de los usuarios y optimizar las diferentes gestiones que realizan, se ha convertido en las principales motivaciones a la hora del desarrollo, razón por la cual se decidió realizar la aplicación en un entorno web por sus ventajas en escalabilidad y administración, de igual manera como lenguaje de programación se utilizó PHP por su flexibilidad, potencia y alto rendimiento; y en conjunto con el framework yii que permite una independencia de los datos y la lógica de negocio con la interfaz del usuario.

La metodología a seguir para el desarrollo de la aplicación se basa en el modelo en cascada, que permite elaborar un proyecto bien documentado, desde el inicio de sus etapas tiene como resultado un proyecto de calidad para hacerlo funcionar de manera óptima, sacrificando un poco de tiempo debe pasar por el proceso de pruebas para que el sistema opere correctamente, a pesar que esta metodología es la más empleada en el campo de desarrollo de software por las ventajas que ofrece, mencionadas anteriormente.

2.3.1 Alcance

Mediante el uso de esta aplicación se implementaran los siguientes aspectos:

- Ofrecer al usuario la información respectiva a los productos que estos requieran en el momento en que se solicite, comprendido en la información presentada la ubicación geográfica de cada artículo.
- Tener el acceso de la información por catálogos que sean sencillos de usar y visualizar.
- Presentar la información de manera que sea fácil, accesible, llamativa y sencilla para todo los usuarios de manera que no se les complique la forma de interactuar con el catalogo web.
- Mostrar un menú de búsqueda y adosado a ello un grupo de opciones que permitan realizar una búsqueda más puntual a lo que el usuario desea encontrar.



2.4 Objetivos

2.4.1 Objetivo General

- Desarrollar una aplicación web publicitaria que permita a los usuarios una óptima gestión o manipulación de datos tomando en cuenta el diseño, usabilidad y accesibilidad a través de diversos dispositivos, implementando tecnologías como Yii, Bootstrap y PostgreSQL, que faciliten y agilicen su creación.

2.4.2 Objetivos Específicos

- Usar el framework **Yii** para la creación y desarrollo del backend del sitio Web.
- Incorporar la extensión **Yii-booster** que permita desarrollar una interfaz adaptable a los diversos dispositivos como PCs, móviles y tablets.
- Hacer uso del Sistema Gestor de Base de Datos **PostgreSQL** orientado a objetos, para almacenar información relacionada a usuarios y productos e información geográfica.
- Integrar **Google Maps API** para georeferenciar las publicaciones que se efectúen en la aplicación y facilitar la selección de productos según los parámetros de búsqueda que los usuarios establezcan.



3. Marco Teórico

En este capítulo abordaremos diferentes temas, los cuales fueron requeridos para la elaboración de este proyecto, empezando con una pequeña descripción del entorno en que se desarrollará nuestra aplicación web, las ventajas y tipos de soluciones web existentes.

De igual forma se explicará la implementación del proyecto basándonos en el comercio electrónico, presentando definiciones, funcionalidades y elementos que conforman la aplicación aplicado al comercio electrónico.

Como el objetivo del presente proyecto se basa en desarrollar una aplicación web e-commerce publicitario implementando Yii framework, en esta sección del escrito se centrará en abordar más a detalles los componentes y metodología aplicada para la correcta realización de dicho proyecto.

Contiguamente en este mismo capítulo abordaremos las herramientas y lenguajes de programación utilizadas para el desarrollo de la aplicación web, de forma que se justifique el porqué de su elección del proyecto elaborado.

3.1 Tipos de aplicaciones web

Hemos encontrado interesante una sección de información contenida en la tesis de Jorge Pastor Pérez, basada en un estudio de usuarios realizando una encuesta para elaborar la clasificación de tipos de aplicaciones web ya que existen diferentes tipos basadas en distintos criterios.

La clasificación elaborada está formada por 13 tipos de aplicaciones web, mostradas a continuación:

- Comercio electrónico: Se utilizan para la venta de productos de consumo, debiendo ser capaces de gestionar los procesos de compra y pago.
- Marketing y presentación de productos: El objetivo principal no es la venta directa de productos al consumidor, sino publicitar de forma atractiva la imagen de una marca o línea de productos.
- Sitios de noticias y blogs: Están diseñadas para informar, y se nutren de entradas o noticias que se actualizan con regularidad.
- Correo electrónico, gestión de eventos y alojamiento de ficheros: Se caracterizan por organizar el trabajo personal de los usuarios, como el correo, la agenda o los documentos.
- Mapas, rutas y lugares: Tienen como objetivo ubicar en mapas lugares concretos o por categorías, permitiendo realizar búsquedas, generar rutas y con opciones de aumento.
- Foros de discusión: Estas aplicaciones Web se emplean para dar soporte a discusiones u opiniones y están organizadas en categorías que contienen foros. Dentro de cada foro se encuentran los temas de discusión, donde los usuarios pueden abrir nuevos temas o contestar a los ya existentes.
- Juego en línea: Estas son aplicaciones interactivas diseñadas para el entretenimiento.



- Redes sociales: Tienen como objetivo interconectar personas entre sí, de modo que cada uno de los miembros va formando su propia red de contactos de forma independiente, con los que puede comunicarse y compartir noticias e intereses.
- Vídeos, fotos, música y software: Este tipo de aplicaciones se caracterizan por sus posibilidades multimedia y de compartición en la red.
- Wikis, enciclopedias y diccionarios: Tienen como objetivo permitir la consulta de definiciones, documentación o artículos enciclopédicos y la edición colectiva de entradas.
- Información y servicios de la comunidad: Son utilizados por la administración para poner a disposición de cualquier ciudadano servicios como los ofrecidos por ayuntamientos, Hacienda o Seguridad Social.
- Banca electrónica: Está diseñado para poder realizar gestiones bancarias a distancia, como la consulta de datos económicos o transferencias.
- Mashups, buscadores y comparadores: Estas aplicaciones usan y combinan datos, presentaciones o funcionalidad procedentes de una o más fuentes para permitir la realización de búsquedas y comparaciones o crear nuevos servicios.

3.2 Comercio electrónico

Otro tema de gran importancia en este proyecto, es el comercio electrónico, saber en qué consiste y las funcionalidades que desempeña fue un elemento clave para entender de mejor manera en aplicar la tecnología correspondiente adaptándola a nuestras necesidades.

El comercio electrónico se puede definir como la compra-venta de productos y servicios a través de sistemas electrónicos, principalmente Internet, basado en el tratamiento y transmisión electrónica de datos, incluidos texto, imágenes y vídeo.

El comercio electrónico ha adquirido rápidamente una gran importancia económica y política con el avance de la tecnología y la expansión mundial de Internet, puesto que no existe ningún otro canal de venta con mayor potencial que el medio online.

Una de las principales diferencias entre el comercio electrónico y el comercio "físico" es el uso de la tecnología, y el desconocimiento de las mismas representa la principal barrera de entrada al comercio en internet, de esta manera podemos destacar que uno de los principales factores a la hora de decidirnos será las funcionalidades que la tecnología nos ofrezca y sus posibilidades de adaptarse a las necesidades que se presenten.



3.2.1 Tipos de comercio electrónico

Existen varias modalidades de comercio electrónico y cada una recibe su nombre en base al tipo de vendedor y comprador.



Ilustración 1: Modalidades de comercio electrónico

- Business to Business (B2B): Comercio realizado entre proveedores y clientes empresariales, o sea, de empresa a empresa.
- Business-to-consumer o business-to-customer (B2C): Es el comercio entre la empresa productora, vendedora o prestadora de servicios y el consumidor final.
- Consumer to Consumer (C2C): Comercio electrónico entre usuarios particulares de Internet. La comercialización de bienes o servicios no involucra a productores y sí a un consumidor final con otro consumidor final, sin intermediarios.
- Government to Business (G2B): Negocios entre gobierno y empresas, por ejemplo: las compras del Estado a través de internet por medio de licitaciones, concurso de precios, etc.
- Government to consumers (G2C): Comercio entre gobiernos, provincial, federal o municipal y consumidores, por ejemplo, el pago de impuestos, multas y tarifas públicas.

Cualquiera que sea la modalidad del comercio electrónico, el proceso siempre será on-line, donde el punto más importante será la calidad del sitio web, comenzando por el diseño, la forma de mostrar la información, la confianza, la seguridad y la facilidad de navegación y la diversidad de productos y servicios ofrecidos.

([Http://www.informatica-hoy.com.ar/aprender-informatica/Tipos-de-Comercio-Electronico.php](http://www.informatica-hoy.com.ar/aprender-informatica/Tipos-de-Comercio-Electronico.php), 2007-2016).



3.2.2 Soluciones para la implementación de un sistema web e-commerce

SaaS y Cloud Solutions: Las soluciones SaaS (Software as a Service) o también llamadas Cloud Solutions (Soluciones en la Nube) son tiendas pre configuradas que no necesitan de una programación por parte de técnicos propios o ajenos a la empresa.

En este caso las soluciones SaaS permiten en muy poco tiempo configurar los productos a vender y personalizar de forma muy simple el aspecto de la tienda y comenzar la venta de productos de forma casi inmediata, pero con la limitante de no poder hacer una adaptación progresiva de las funcionalidades que no estén contempladas en la solución SaaS.

Ejemplos de soluciones SaaS:

www.shopify.com

www.magentogo.com

www.prestabox.com

www.theetailers.com

www.ozongo.com

www.1and1.es

Soluciones Open Source: La tecnología Open Source o de Código Abierto han dado un impulso a las organizaciones por la facilidad de implantar soluciones tecnológicas en prácticamente todos los ámbitos a un coste reducido.

La ventaja que ofrece la tecnología basada en Open Source es que el código de la tecnología es público y de uso gratuito, sin pago de licencias y la comunidad de desarrolladores alimentan y fomentan el mantenimiento y crecimiento de la tecnología.

Prácticamente cualquier persona con conocimientos medios o avanzados de las distintas tecnologías de programación Open Source, pueden crear una tienda en Internet con un coste reducido, amplias funcionalidades y unos tiempos de desarrollo bastante asequibles.

No obstante como desventaja siempre cabe la posibilidad de que ante un gran crecimiento de la tienda necesitemos programaciones más específicas y adaptadas a los distintos procesos más particulares, lo que en muchas ocasiones, conllevará el desarrollo de una programación propia.

Ejemplos de soluciones Open Source:

www.magento.com

www.prestashop.com

www.zencart.com

www.opencart.com

Desarrollo a medida: Los desarrollos a medida a diferencia de las soluciones SaaS y de las Open Source conllevan una programación desde la base.

Ventajas de la programación de un desarrollo a medida:

- Adaptación al 100% a los procesos de la empresa (procesos contables, gestión de proveedores, gestión de stocks y almacén...etc.).
- Sin prácticamente límites de programación, más que los propios que pueda alcanzar el lenguaje de programación elegido.



- Independencia frente a actualizaciones de funcionalidades de terceros, algo muy común en las soluciones propietarias o pre configuradas.

Desventajas:

- Mayor tiempo necesario para la programación de la tienda.
- Definición desde la base de las funcionalidades requeridas.
- Programación más extensa.
- Contratación interna o externa de más recursos y más cualificados: si la programación se realiza desde la base, será necesario un equipo de programadores, arquitectos de información, técnicos de sistemas y más recursos necesarios para programar una tienda profesional a la altura de las soluciones preconfiguradas.

3.2.3 Elementos de una tienda online

Cuando realizamos una Tienda Online, debemos tener en cuenta que los elementos que la conforman deben sustentarse en 3 pilares clave:

- Diseño.
- Usabilidad.
- Accesibilidad.

Diseño: El diseño de la página es un factor crucial. Debe ser intuitivo y accesible.

- La tienda online debe estar decorada con los propios contenidos de tal manera que los productos sean los que destaquen por encima del diseño.
- Combinar colores para crear una visión cálida y agradable.
- Los menús de navegación deben seguir un orden establecido, así conseguimos no despistar al usuario.
- Disponer de un diseño orientado a la optimización de buscadores: es decir que debe tener una buena indexación.

Usabilidad: Decimos que una página Web es “usable” cuando muestra todo el contenido de una forma clara y sencilla de entender por el usuario, favoreciendo el acceso a sus funcionalidades y consiguiendo la satisfacción del usuario en su experiencia de navegación.

- El catálogo de productos debe estar visible desde el primer momento.
- Fácil acceso a los productos mediante una clara navegación por categorías y subcategorías.
- El carrito de la compra siempre debe estar visible.
- El Proceso de la compra debe estar enfocado al producto de una forma clara y rápida.
- Disponer de un potente buscador que nos ofrezca la posibilidad de acceder a nuestro catálogo de producto por distintos criterios (precio, fecha, orden alfabético...).
- Facilitar el acceso a apartados de servicio de la tienda como Información de contacto, forma de comprar, condiciones generales, etc.



Accesibilidad: Una tienda online es accesible si sus productos, contenidos y servicios pueden ser accedidos por el mayor número posible de personas.

- El catálogo de productos debe ser accesible por categorías y escaparates comerciales.
- Es también deseable mostrar productos destacados con un diseño algo distinto al resto de productos en especial si queremos promocionar un producto determinado.
- Escoger el tamaño del texto óptimo y destacando aquella información más relevante: nombre producto, botón comprar, precio, etc.
- Uso correcto del etiquetado de las imágenes favoreciendo su indexación.

3.3 Herramientas de desarrollo

3.3.1 HTML5

HTML5 es la última versión de HTML, se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.

Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance.

(HTML5, actualizado el 01 de Septiembre del 2015)

3.3.2 CSS3

CSS3 es la última evolución del lenguaje de las Hojas de Estilo en Cascada (Cascading Style Sheets), y pretende ampliar la versión CSS2.1. Actualmente permite la integración de las esquinas redondeadas, sombras, gradientes, transformaciones 2D y 3D, transiciones o animaciones, múltiples columnas de layouts.

Las partes experimentales son particulares para cada navegador y deberían ser evitadas en entornos de producción, o usadas con extrema precaución, ya que tanto la sintaxis como la semántica pueden cambiar en el futuro. (CSS3, developer.mozilla.org, actualizado el 12 de Mayo del 2016).

3.3.3 PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo del código HTML que recibe el navegador.

3.3.4 Responsive web design (Diseño web adaptable)

El diseño web adaptable o adaptativo, conocido por las siglas RWD del inglés Responsive Web Design, es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visualizarla. Hoy día las páginas web se visualizan en multitud de tipos de dispositivos como tabletas, teléfonos inteligentes, libros electrónicos, portátiles, PC, etcétera. Además, aún dentro de cada tipo, cada dispositivo tiene sus características concretas: tamaño de pantalla, resolución, potencia de CPU, capacidad de memoria, entre otras. Esta tecnología pretende que con un solo diseño web, se tenga una visualización adecuada en cualquier dispositivo. (Diseño web adaptable, actualizado el 23 de Octubre del 2015)



3.3.5 Bootstrap

Bootstrap es un framework front-end desarrollado y liberado por twitter, nos permite agilizar el desarrollo de nuestro diseño web, además nos permite utilizar una completa colección de elementos (tipografías, formularios, botones, tablas, barra de navegación, carrusel de imágenes y otros más) así como plugins de javascript opcional.

Se le dio más importancia al desarrollo adaptativo para móviles, por el cual adoptó una filosofía mobile-first (primero móvil), que es totalmente adaptable al tamaño de un móvil, ordenador o de un tablet sin crear otras hojas de estilos o escribir largas propiedades css para hacerlo adaptable.

Es compatible con la mayoría de los navegadores modernos web y móviles como se muestra en la siguiente imagen.

	Chrome	Firefox	Internet Explorer	Opera	Safari
Android	✔ Supported	✔ Supported	N/A	✘ Not Supported	N/A
iOS	✔ Supported	N/A		✘ Not Supported	✔ Supported
Mac OS X	✔ Supported	✔ Supported		✔ Supported	✔ Supported
Windows	✔ Supported	✔ Supported	✔ Supported	✔ Supported	✘ Not Supported

Ilustración 2: Versiones soportadas por los diferentes navegadores web

Extraoficialmente bootstrap se comporta bastante bien en linux con Chrome y Firefox a pesar que no se admiten oficialmente.

3.3.6 Patrón de diseño Modelo Vista Controlador

Al momento de crear un proyecto para un sistema web es necesario implementar un patrón de diseño que nos facilite la organización de código y posteriormente el mantenimiento y escalabilidad .La solución más utilizada es implementar el patrón de diseño Modelo Vista Controlador, el cual separa los datos y la lógica del negocio con el interfaz del usuario.

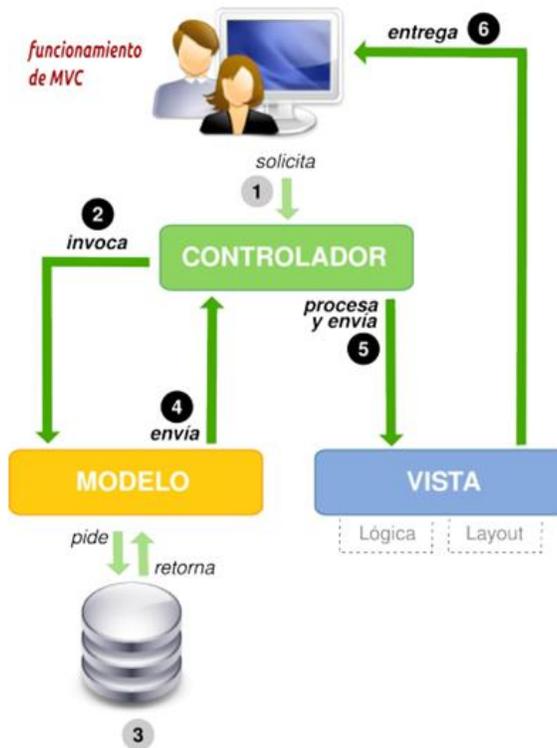


Ilustración 3: Arquitectura MVC. Tomada POO y MVC en PHP por Eugenia Bahit.

Para lograr esta independencia el patrón separa nuestro código en tres capas.

Modelo: Representa los datos con que el sistema opera, por tanto ejecuta todas las peticiones para interactuar con la información, tanto consultas como actualizaciones. Los datos se almacenan habitualmente en una base de datos, por lo tanto los modelos guardan todas las funciones que se comunican con las tablas y harán las correspondientes consultas (selects, inserts, updates, etc.).

Controlador: Responde a acciones del usuario e invoca peticiones al modelo cuando realiza alguna solicitud de información, como editar información o realizar un registro en la base de datos.

Esta capa sirve de intermediario entre las vistas y los modelos, respondiendo a las acciones que pueden requerirse en la implementación de los requerimientos del sistema. Aunque esta capa no manipula directamente los datos, ni tampoco muestra los datos directamente a la vista, simplemente es el puente para comunicar los modelos con las vistas para implementar las diversas necesidades del desarrollo.

Vista: Las vistas se encargaran de visualizar las interfaces de la información obtenida, o sea permite renderizar los estados de nuestra aplicación en páginas HTML. En las vistas solamente tenemos los códigos HTML que permite mostrar la salida de los datos.

En las vistas generalmente se trabaja con los datos, sin embargo no se realizan ninguna operación sobre ellos directamente. Sin embargo las vistas requieren de dicho modelo para visualizar la información requerida.



3.3.7 Yii Framework

Yii es un framework de desarrollo de alto rendimiento, muy flexible y versátil, escrito en **PHP5**, es orientado a objetos, de software libre y de alto rendimiento basado en componentes para el desarrollo de aplicaciones Web de gran escala. Yii permite desarrollar aplicaciones basadas en el modelo MVC (modelo-vista-controlador) que posteriormente mantendrá el crecimiento sin comprometer su funcionamiento o calidad para futuros cambios de nuestro sistema web.

Hoy en día hay muchos frameworks destinados al desarrollo de aplicaciones web, donde cada uno presenta puntos fuertes y débiles con respecto al otro, y las ventajas pueden ser más claras en un framework que en los demás. Al final no hay una decisión clara o definitiva sobre cual framework usar, ya que todo depende de las necesidades o funcionalidades que necesite el desarrollador, optando así por el framework que mejor se adecue a estas, además no es posible decir que una herramienta es mejor que otra sin haber hecho uso de ellas.

Como elección para nuestro trabajo hemos escogido Yii Framework. La elección la hemos hecho atendiendo a las características que tiene, su rápido aprendizaje y generación automática de código CRUD con vistas, controladores, funciones de búsqueda, modelos para generar el esqueleto de la aplicación.

Ventajas de Yii Framework

- Generador de código Gii. Yii nos ayuda a generar una gran cantidad de código para nuestro modelo, controlador y vistas. El único trabajo que nos queda por hacer es el de acomodar el código o diseño a nuestras necesidades específicas.
- Objetos a Acceso a Datos (DAO, Database Access Objects). Es un API genérica para acceder a los datos almacenados en diferentes sistemas de administración de bases de datos (DBMS), como resultado con yii podemos cambiar el sistema de administración de bases de datos a otro sin la necesidad de cambiar el código que usa DAO para acceder a los datos.
- Medidas de seguridad y prevención de ataques comunes. Yii provee medidas de seguridad para ataques como inyección SQL, Cross-Site scripting (XSS), cross-site request forgery (CSRF) y de manipulación de cookies, Incluye también las validación de entradas y filtrado.
- Manejo de formularios. Los formularios generados por Gii utilizan validación, esto significa que cuando ingresan datos no correspondientes en el campo, yii nos muestra mensajes de error en los campos rellenos.
- Integración de JQuery. Yii permite integrar en una sola línea de código PHP las funcionalidades de un selector de fecha o campos de entrada de auto-completado.



3.3.8 API Google Maps

Uno de los contenidos que han sido más demandados por los usuarios en internet es el de realizar búsquedas de localización por productos o servicios, pero pocos son los sistemas web e-commerce que han integrado el uso de mapas, por esta razón nos pareció de gran utilidad el implementar el API de Google Maps en nuestro sistema web. A continuación abordaremos un poco de esta API.

El API de Google Maps estándar es un servicio gratuito que permiten integrar mapas en los sitios web o aplicaciones móviles y se suele utilizar para aplicaciones no comerciales ya sea de pruebas o de desarrollo.

Las API de Google Maps están categorizadas por plataforma: Web, Android e iOS. En este documento nos centraremos en el la utilizar el API de Google Maps para la plataforma Web.

Ventajas que ofrece el API de Google Maps

- No es necesario disponer de un servidor web propio, ni disponer de un gestor de bases de datos.
- La plataforma se encarga de mantener nuestros mapas disponibles las 24 horas del día los 365 días del año.
- No hace falta planificar de antemano las posibles necesidades de capacidad. Solo es necesario ampliar según sea necesario y pagar por la capacidad que utilices en cada momento.
- Las interfaces de usuario son sencillas y elegantes.
- Posibilidad de contar con soporte.
- Cada vez estos servicios nos ofrecen más funcionalidades.
- No es necesario instalar ningún tipo de software.
- Posibilidad de ejecutar consultas espaciales sobre tus datos.

Para su funcionamiento aplicaremos las características de HTML5 y Javascript. Los mapas tan solo son imágenes obtenidas a través de peticiones AJAX (llamadas asíncronas al servidor) las cuales se sitúan en un capa con etiqueta div. Ya que tengamos el mapa cargado en el navegador cada vez que se emplea algunas de las herramientas del API se envía información a través de AJAX actualizando el contenido.

Un sencillo ejemplo de la ubicación de León con Google Maps:



```
<script>
window.onload = function(){

    var miLatLng = new google.maps.LatLng(12.437179, -86.876930);

    var mapOpciones = {
        zoom: 13,
        center: miLatLng
    };

    var mapa = new google.maps.Map(document.getElementById("mapa"), mapOpciones);

    var marker = new google.maps.Marker({
        position: miLatLng,
        map: mapa,
        title: "Localización de León"
    });
}
</script>
```

Ilustración 4: Ejemplo de ubicación aplicando google maps.

El código anterior nos da como resultado la ubicación de la ciudad de León como se muestra en la siguiente imagen.



Ilustración 5: Resultado de google maps.



3.3.9 Cloud9 IDE

Es un entorno de desarrollo integrado en línea. Es compatible con cientos de lenguajes de programación, incluyendo PHP, Ruby, Perl, Python, JavaScript con Node.js, and Go. Permite a los desarrolladores empezar con la codificación casi de manera inmediata con espacios de trabajo preconfigurados, permite colaborar en grupos de trabajos con características de codificación de colaboración, y las características de desarrollo web como vista previa en vivo y pruebas de compatibilidad del navegador.

Entre las ventajas de este VPS (Virtual Private Server) con respecto a lo que puedes conseguir con el desarrollo local:

- Sin configuración: Con crear tu usuario puedes disponer de tu servidor listo para usar con los lenguajes más habituales.
- Probar en un entorno más real: Puedes probar tu código en un entorno más real que a través de "localhost". En un dominio remoto, de Internet y un sistema operativo habitual en sitios en producción.
- Posibilidad de compartir con otras personas: sin que tengan que hacer nada en especial.
- Trabajar desde cualquier ordenador: con el único requisito que esté conectado a Internet y sin tener que instalar ningún software en ese ordenador.

3.3.10 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es muy conocido y usado en entornos de software libre y por las funcionalidades avanzadas que soporta, lo que lo sitúa al mismo nivel que muchos SGBD comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Componentes más importantes en un sistema PostgreSQL.

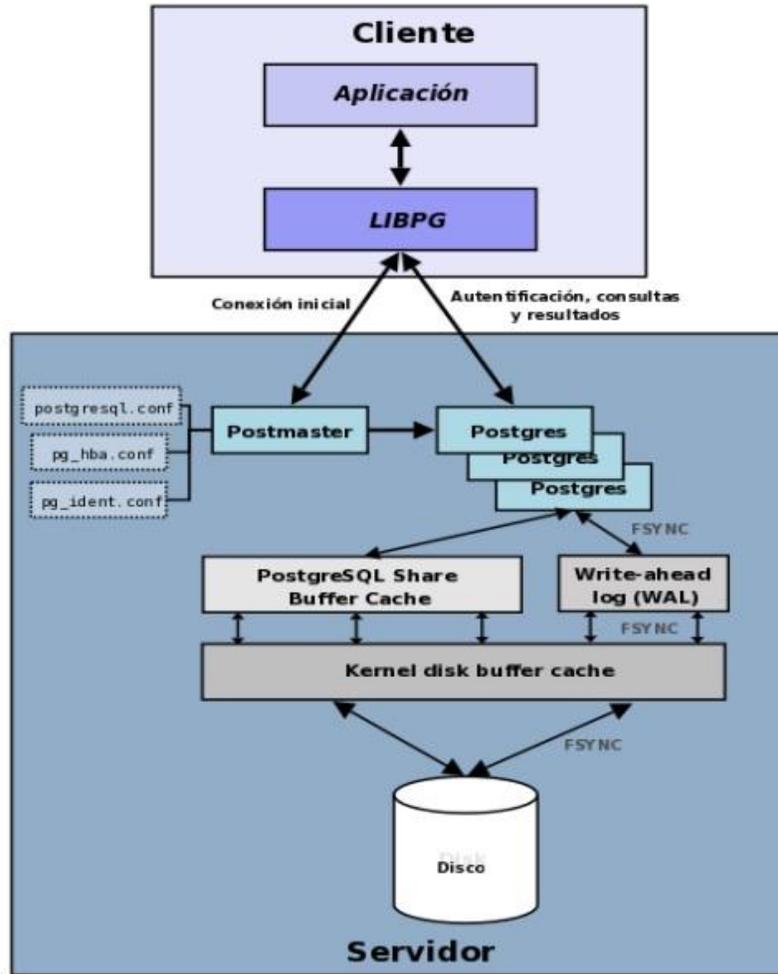


Ilustración 6: Manera general los componentes más importantes en un sistema PostgreSQL



Algunas de sus características son:

- PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.
- Bases de datos 100% ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), tiene soporte completo para claves foráneas, combinaciones, vistas, triggers y procedimientos almacenados.
- Integridad referencial.
- Cuenta con un rico conjunto de tipos de datos, incluyendo SERIAL, INTEGER, numéricos, booleanos, CHAR, VARCHAR, DATE, INTERVALO y TIMESTAMP, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.
- También es compatible con el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o vídeos.
- Su administración se basa en usuarios y privilegios con múltiples métodos de autenticación.
- Juegos de caracteres internacionales, los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Soporta Tablespaces para definir dónde almacenar la base de datos, el esquema, los índices, etc.
- Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida (Herencia).
- Acceso encriptado vía SSL.



4. Diseño metodológico

4.1 Descripción general del proyecto

Con la creación de este proyecto se pretende brindar acceso a los usuarios desde los distintos dispositivos móviles y de escritorio a los productos que están alojados en el servidor, por medio de un navegador web.

Durante la elaboración de este proyecto fue aplicado el modelo de ciclo de vida clásico denominado también como modelo en cascada, dado a que se aprecia que es un modelo sencillo y disciplinado, con un funcionamiento fácil de aprender y utilizar, además de estar dirigido por los resultados de cada etapa durante el desarrollo del proyecto.

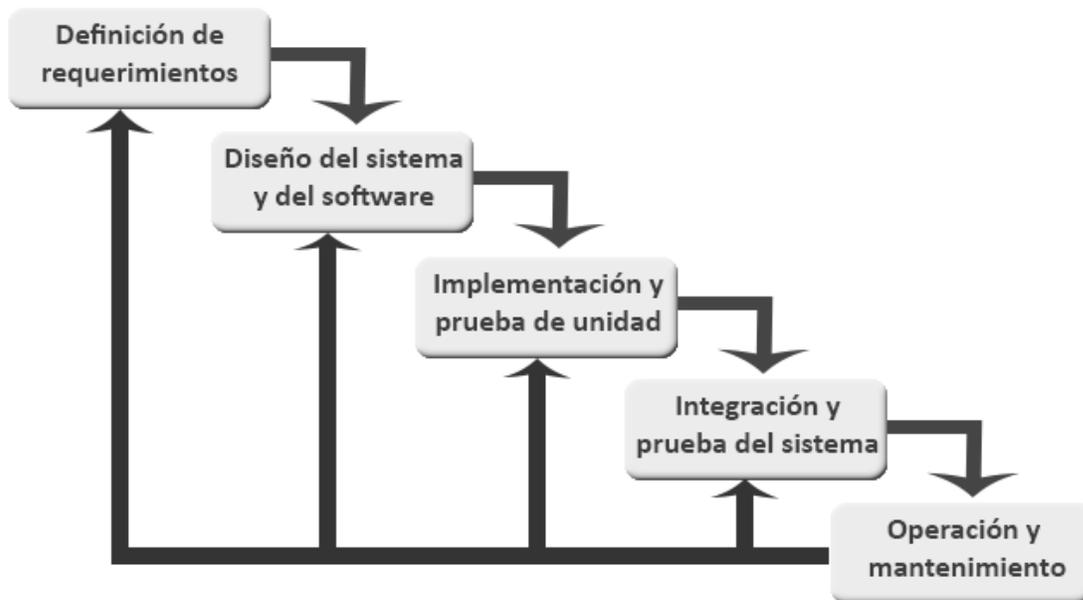


Ilustración 7: Ciclo de vida Clásico. (Ian Sommerville, 2011).



4.2 Etapas del trabajo

4.2.1 Etapa 1: Definición de requerimientos

Descripción

En esta etapa se comprende todo lo referente a lo que es la identificación de requisitos y el alcance que tendrá el sistema a desarrollar.

En esta etapa comprendimos las siguientes tareas:

- Identificar los requisitos que deben satisfacer el sistema.
- Delimitar los alcances expectantes del sistema en base a los requisitos previamente establecidos.
- Determinar el uso de las herramientas y tecnologías que se aplicaran en el desarrollo del nuevo entorno web.

4.2.2 Etapa 2: Diseño del sistema y del software

Descripción

En esta etapa del diseño se implementara los requerimientos del sistema hardware y software, se establecerá la arquitectura del sistema a seguir a través de la elaboración de los diagramas para la construcción del sistema web, en esta etapa también se creara el aspecto gráfico y de interacción de sistema.

En esta etapa comprendimos las siguientes tareas:

- a) Estructurar la información de los requisitos previamente establecidos a través diagramas de casos de usos y diagrama de clases. En esta etapa se elaborarán diagramas que permiten representar la interacción entre los usuarios y el sistema.
- b) Desarrollar métodos de validación que permitan mantener la integridad de los parámetros tomando en cuenta los requisitos de los usuarios.
- c) Estructurar las especificaciones del diseño de la aplicación. En esta etapa se plantea las posibles propuestas de las interfaces con las que podrán interactuar cada usuario de modo que le permita acceder a los datos que referencia nuestro sistema web.

4.2.3 Etapa 3: Implementación y prueba de unidad

Descripción

En esta etapa fue uno de los procesos más largo en la implementación del desarrollo, en el que aplicamos todo el diseño definido anteriormente a través del lenguaje de programación de PHP en conjunto del framework Yii. Para esta etapa se pensó en las siguientes acciones:

- a) En la primera fase se realizó la implementación del diseño que tendrá nuestro sistema web.
- b) En la segunda fase nos encargamos de aplicar el paradigma de programación orientado a objeto y unificar los módulos que se han trabajado de forma paralela e independiente.



c) En la tercera fase terminamos de realizar los últimos ajustes de nuestro sistema web.

4.2.4 Etapa 4: Integración y prueba del sistema

Descripción

En esta etapa se realizaran pruebas locales para verificar que el sistema se desempeñe correctamente, para determinar posibles errores que no fueron detectados durante la implementación del desarrollo, los cuales se procedió a corregir.

4.2.5 Etapa 5: Operación y mantenimiento

Descripción

En esta etapa el sistema se instalara y se pondrá en práctica, el mantenimiento incorpora corregir los errores que no se localizaron en las etapas anteriores del ciclo de vida, mejorar, incorporar y poner en funcionamiento las unidades del sistema e incrementar los servicios del sistema conforme se agregan nuevos requerimientos.

4.3 Materiales

4.3.1 Hardware

DESCRIPCIÓN	CARACTERÍSTICAS
Computadora portátil	Sistema operativo Windows 8
	Procesador Intel Celeron 1.80 GHz
	Memoria RAM 8 GB
	Disco duro 700 GB
Computadora escritorio	Sistema operativo Lubuntu
	Procesador Intel Pentium 4 2.6 GHz
	Memoria RAM 3 GB
	Disco duro 80 GB
Mini Laptop	Sistema operativo Windows 7 ultimate
	Procesador Intel Atom 1.60 GHz
	Memoria RAM 2 GB
	Disco duro 320 GB



Cloud Server	Sistema operativo Ubuntu server 14.04
	Procesador Intel Xeon 2.30 GHz
	Memoria RAM 512 MB
	Disco duro 2 GB

4.3.2 Software

DESCRIPCIÓN	CARACTERÍSTICAS
Sistema Operativo	Ubuntu server 14.04
Servidor Web	Apache
IDE de programación	Sublime Text 3
Lenguaje de programación	PHP
Framework de programación	Yii
Gestor de base de datos	PostgreSQL
Navegador Web	Google Chrome, Mozilla Firefox, Opera



5. Resultados

En el presente apartado se muestra los resultados obtenidos de la elaboración de un sistema web destinado a la publicidad colectiva de productos.

5.1 Despliegue del sistema

El sistema desarrollado se encuentra instalado en un servidor web alojado en la nube, el cual puede ser accedido desde cualquier ubicación que cuente con acceso a internet y a través de un navegador web de su preferencia (Google Chrome, mozilla firefox, etc.), con el objetivo de que los usuarios puedan hacer un fácil uso de la aplicación de manera segura y sin el riesgo de alterar la información almacenada.



Ilustración 8: Despliegue

El servidor web y la base de datos se encuentran instalados en el mismo servidor, se realizó de esta manera debido a la falta de equipos y recursos, aunque lo recomendable es instalarlo por separado.

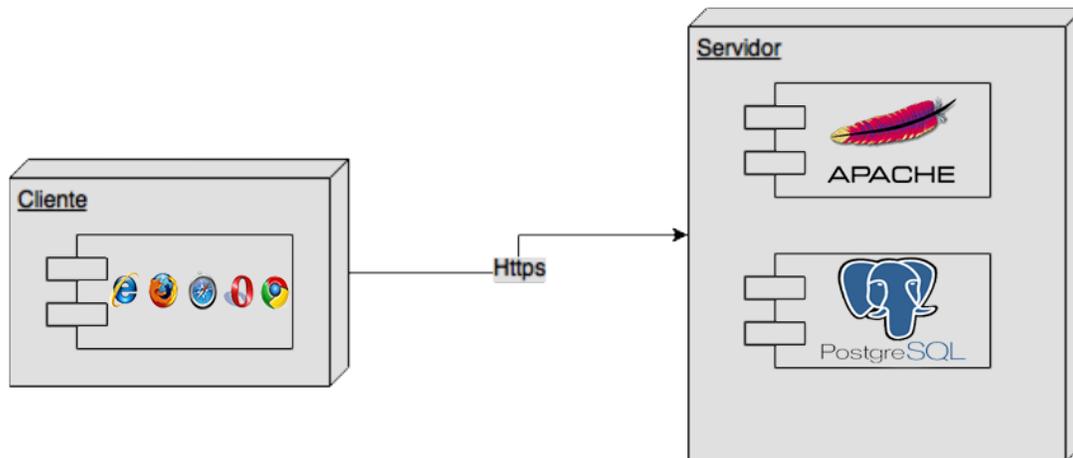


Ilustración 9: Despliegue 2.

5.2 Diseño del sistema

En el desarrollo de este sistema se emplearon los estándares de diseño web de la W3C, para un correcto acceso desde múltiples dispositivos y plataformas, garantizando que puedan visualizar nuestra aplicación, que tengan una correcta usabilidad y navegabilidad.

Apegándonos a la secuencia básica del estándar, el sistema está dividido en 4 partes.

Cabecera, la cual se muestra el logotipo de la página, la barra de navegación que se mantiene visible independientemente de donde estemos navegando, dando acceso al catálogo para visualizar múltiples categorías, además muestra el usuario que ha iniciado sesión y la opción de salida del sistema.

Sección uno, muestra como fondo un pequeño video de presentación, ya que consigue captar la atención de los usuarios, adoptando así una de la tendencia más popular en cuanto a diseño actualmente, y finalmente podemos destacar la integración de cajas de texto de manera que podamos interactuar y realizar búsquedas.

Sección dos, esta sección se presenta algunos de los productos más recientes en existencia dentro del sistema, visualizando algunas de sus características más relevantes.

Pie de página, esta sección muestra la información referente a los derechos de autor, enlaces re direccionados a las redes sociales.

5.2 Funcionamiento del sistema

El sistema podrá ser accedido por diferentes usuarios simultáneamente, por tal motivo para ingresar es necesario autenticarse con un nombre de usuario y contraseña. Al momento del desarrollo del sistema, se creó un usuario administrador, para que posteriormente él cree y asigne los permisos a como requieran conveniente.



Autenticación de usuario

Ingresa tu nombre de usuario

Ingresa tu contraseña

Iniciar sesión

No cerrar sesión

[Registrarse](#) | [¿Perdistes tu contraseña?](#)

Ilustración 10: Autenticación de usuario.

El acceso a la aplicación está basado en roles, de los cuales se destacan dos: usuario registrado y usuario administrador. Dependiendo con que usuario se acceda al sistema se presentará una u otra información.

- Usuario registrado: Tendrá acceso a la información referente a productos, catálogos y otros detalles relacionados a su perfil de usuario.
- Usuario administrador: Posee acceso total a las funcionalidades y podrá gestionar la información de los usuarios, productos, categorías y demás datos existentes en el sistema.



Crear un nuevo perfil.

The image shows a registration form titled "Regístrate" (Register) on a website. The form is displayed within a white modal window against a blue header. The header contains navigation links: "Inicio", "Catálogo", "Únete", and "Regístrate". The registration form includes the following fields and options:

- Nombre ***: Text input field.
- Apellido ***: Text input field.
- Telefono ***: Text input field with "0" as a placeholder.
- Línea Telefónica ***: Radio button options for "movistar" and "claro".
- Nombre de usuario**: Text input field.
- Correo**: Text input field.
- Contraseña**: Text input field.
- Confirmar contraseña**: Text input field.
- Registrar**: A green button at the bottom of the form.

At the bottom of the page, there is a dark footer with links for "Iniciar búsqueda", "Quiénes somos", "Publica tu tienda", and social media icons for Facebook and Twitter.

Ilustración 11: Crear un nuevo perfil.

La aplicación contará con un formulario de registro para que los visitantes ingresen sus datos y puedan crear un perfil, con lo cual podrán acceder a las funcionalidades siguientes:

- Actualización de los datos referentes a su propio perfil de usuario.
- Gestionar productos.
- Consultar productos por categorías.
- Realizar búsquedas.



Perfil de usuario.

 **Perfil de usuario** [Cambiar contraseña](#)

Información básica

Nombre: administrator

Apellido: admin

username: admin

E-mail: webmaster@example.com

Información de acceso

Registration date: 2016-02-01 20:57:39.323465

Last visit: 2016-04-27 18:12:31

[Editar mi cuenta](#)

Ilustración 12: Perfil de usuario.

Muestra una vista generalizada de la información del perfil del usuario de la sesión activa, además permitirá editar la información del usuario, la modificación de su contraseña en caso de ser necesario y la opción de darse de baja si así lo desea.



Actualizar perfil de usuario.

Actualizar perfil de usuario Cancelar

Información básica

Nombre *
administrator

Apellido *
admin

Nombre De Usuario
admin

Correo
webmaster@example.com

Guardar

Ilustración 13: Actualizar perfil de usuario.

La actualización de los datos del perfil del usuario se llevara a cabo a través de un formulario intuitivo de fácil manipulación, que permitirá a los usuarios modificar o editar la información de su perfil.



Gestión de perfil de usuario

username	E-mail	Superuser	Status	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
admin	webmaster@example.com	Yes	Active	
farah	farah@test.com	No	Active	
William	william@test.com	No	Active	
wilson	batistuta@gmail.com	No	Active	
Luis	Cesaritomelendez@hotmail.com	No	Active	
a2lisc2	a2lisc2@gmail.com	No	Active	
cccc	lasdkalsgnaskf@gmail.com	No	Active	
demo	demo@example.com	No	Active	
cristian	ltachi@test.com	Yes	Active	
daniel	progdaniel@test.com	No	Active	

Ilustración 14: Gestión de perfil de usuario.

La vista de administración de usuarios está dividida en tres secciones, la primera permitirá al administrador filtrar búsquedas de manera que pueda acceder rápidamente a los datos de un determinado perfil, la segunda muestra un listado de todos los usuarios registrados y una tercera sección que contiene las opciones de administración como lo son la consulta, edición y eliminación de los datos de cada perfil.

Registrar Productos

Agregar producto Gestión de productos

Categoría

Tipo De Moneda

Estado Fisico

Barrio

Nombre

Precio

Descripcion

Imagen
 No se ha seleccionado ningún archivo.

Ilustración 15: Registrar Productos.



Los usuarios podrán registrar los datos de sus productos a través de un formulario, en el cual le serán solicitados los detalles más relevantes como son la categoría, moneda, estado físico, barrio y municipio donde se encuentra el producto, nombre, precio y una pequeña descripción, para posteriormente ser almacenados en la base de datos de la aplicación.

6. Aspectos finales

6.1 Conclusiones

Durante el desarrollo de nuestro proyecto tesis utilizamos potentes herramientas como lo son Yii Framework, PostgreSQL, Bootstrap y Google Maps API, las que nos permitieron finalizar satisfactoriamente la aplicación. Dichas herramientas son de gran consideración y utilidad a la hora de crear aplicaciones o sitios de diversos tipos.

- El utilizar Frameworks que permitan la integración de módulos pre-programados agilizan el desarrollo de diversas funcionalidades de la aplicación a desarrollar.
- El gestor de base de datos PostgreSQL categorizado como un software de licencia libre, permitió la manipulación de los datos de la aplicación con utilidades similares e incluso de igual nivel que muchos SGBD comerciales.
- El desarrollo de un diseño responsivo a través de la extensión Yii-Booster permitió a los usuarios un mejor acceso a la aplicación, lo que evito limitar su funcionalidad solo a equipos de escritorio, abarcando así la gran cantidad de usuarios conectados a través de dispositivos móviles.
- La referencia de los productos a través de la tecnología de geo localización con la integración de un mapa permitió a los usuarios realizar búsquedas, ayudando así a los usuarios a visualizar los productos cerca de un área a fin.

6.2 Recomendaciones

Al concluir esta tesis, se plantean las siguientes recomendaciones:

- Incentivar el uso de Yii-Framework para el desarrollo de futuras aplicaciones web dentro de la Facultad de Ciencias y Tecnologías.
- A futuras mejoras del sistema, integrar la funcionalidad del carrito de compras a través de la extensión Shopping-Cart de Yii-Framework, lo cual facilitara las transacciones de compra y venta de productos desde la misma aplicación.
- Adquisición de un dominio de alojamiento de la web en un proveedor de hosting.
- Implementar SSL para que la comunicación entre cliente y servidor sea segura.



7. Bibliografía

Asociación Española de la Economía Digital (adigital), 2012, Libro Blanco del comercio electrónico.

CSS3 (actualizado el 12 de Agosto del 2015). Obtenido el 16 de Noviembre del 2015, de Developer Mozilla: <https://developer.mozilla.org/es/docs/Web/CSS/CSS3>.

Diseño web adaptable (actualizado el 23 de Octubre del 2015). Obtenido el 16 de Noviembre del 2015, de Wikipedia: https://es.wikipedia.org/wiki/Diseño_web_adaptable.

Fc. Javier González Rivera (2014). Maqueta Dcc Postgresql.

<http://www.informatica-hoy.com.ar/aprender-informatica/Tipos-de-Comercio-Electronico.php>.

HTML5 (actualizado el 01 de Septiembre del 2015). Obtenido el 16 de Noviembre del 2015, de Developer Mozilla: <https://developer.mozilla.org/es/docs/HTML/HTML5>.

Pérez, J. P. (2013). Estudio y clasificación de tipos de aplicaciones Web y determinación de atributos de usabilidad más relevantes, tesis de grado Ingeniería en informática. Valencia: Universidad Politécnica de Valencia.

PostgreSQL (The PostgreSQL Global Development Group 2015): <http://www.postgresql.org/about/>.

User module, Yii Framework extension <http://www.yiiframework.com/extension/yii-user/>.

Sommerville, I. (2011). Ingeniería del Software, Novena Edición. Madrid: Pearson Educación.



8. Anexos

A continuación se plasmarán de tal forma que no den lugar a ambigüedades, los requerimientos que fueron necesarios para el desarrollo de la aplicación.

Alcance

El sistema será desarrollado para trabajar en un entorno web lo que permitirá su utilización en distintos dispositivos y sistemas operativos que cuenten con un navegador web, brindando así un mejor acceso a la información mediante las funcionalidades que se detallarán de manera general a continuación:

- Registrarse
- Identificarse
- Editar/consultar perfil de administrador
- Añadir/editar/eliminar/consultar perfil de usuario
- Añadir/editar/eliminar/consultar categoría
- Añadir/editar/eliminar/consultar municipio
- Añadir/editar/eliminar/consultar barrio
- Añadir/editar/eliminar/consultar producto
- Filtrar búsqueda por nombre producto/categoría/barrio/municipio/rango de precio.

Características de los usuarios

Los usuarios del sistema deberán poseer un nivel de educación superior, así como habilidades y experiencia básica en el uso de tecnologías informáticas, además serán los que publicarán toda la información referente a sus productos.

Requerimientos

Esta sección contiene todos los requerimientos del software a un nivel de detalle suficiente para permitir a los programadores diseñar el sistema para satisfacer esos requerimientos.

Requisitos comunes de interfaces

Interfaces de usuario

Las interfaces serán desarrolladas de manera que puedan ser utilizadas intuitivamente por los usuarios. Se debe utilizar el lenguaje del usuario, con expresiones y palabras que le resulten familiares. No puede contener diálogos ambiguos, que lleven a falsas interpretaciones o que genere dudas al usuario. Los mensajes de error deben expresarse en un lenguaje común y sencillo, indicando con precisión el problema y sugiriendo las posibles alternativas o soluciones.

Interfaces de Comunicación

La aplicación será accesible a través de Internet y seguirá la arquitectura cliente-servidor, el protocolo de comunicación entre la aplicación cliente (navegador web) y el servidor web será HTTP.



Interfaces Hardware

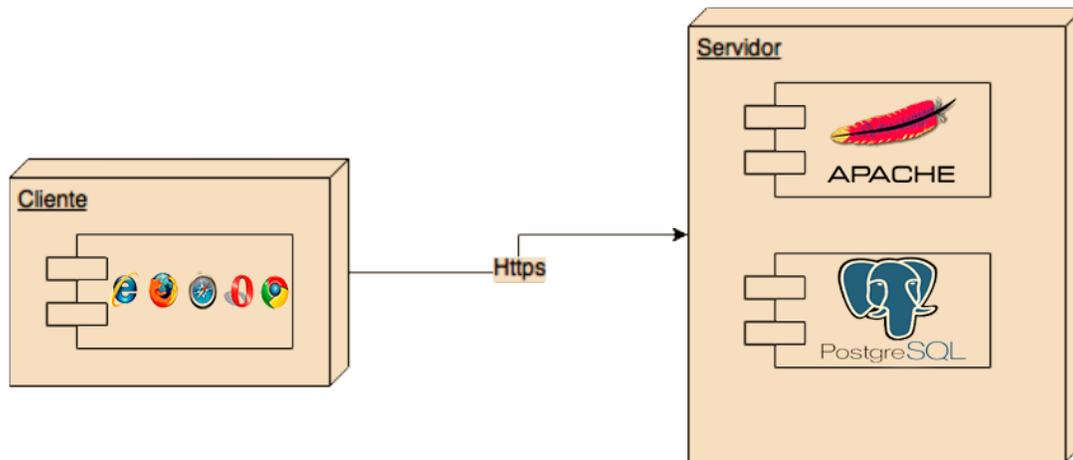


Ilustración 16: Interfaces Hardware

Interfaces Software

Es recomendable utilizar las últimas versiones de los navegadores web más populares o utilizados por los usuarios, entre los cuales podemos destacar Google Chrome, Mozilla Firefox, Opera, Internet Explorer y Safari para la correcta visualización de todas las interfaces del sistema.

Para el equipo servidor se requiere los siguientes software:

- Ubuntu Server 14.04
- Servidor HTTP Apache 2.4
- PostgreSQL 9.4.7

Requerimientos funcionales

Numero de Requisito	Descripción
Requisitos Generales	
RF01	Para tener acceso a la información que muestra la aplicación el usuario deberá autenticarse escribiendo su nombre de usuario y su contraseña.
RF02	En todo momento se mostrará una opción para cerrar sesión.
RF03	Solo se mostrarán las opciones para las que tenga permiso el usuario que ha iniciado sesión.
RF04	Será necesario pedir confirmación de la acción a realizar en el caso de borrar cualquier información
RF05	Los campos están validados según el tipo de dato que reciben.
RF06	Al iniciar sesión cada usuario almacena la información referente a la última sesión que inicio.
RF07	Todas las secciones de la aplicación seguirán el mismo patrón de diseño, el cual consta de un menú para acceder a las distintas secciones, un



	cuerpo de página donde se muestra la información necesaria de cada sección y un pie de página.
RF08	El menú de navegación deberá ser visible en todo momento.
RF09	Todas las ventanas modales se mostrarán en el centro de la ventana del navegador.
RF10	En todo momento se mostrará el nombre del usuario con el que se ha iniciado sesión.
RF11	Al momento de borrar o actualizar un registro, el usuario será redireccionado a la vista de los registros que se están modificando.
Administrador	
RF12	La aplicación contara con un Administrador por defecto que podrá acceder a través de su respectivo nombre de acceso y contraseña, además de sus credenciales.
RF13	El Administrador podrá editar los datos de su respectivo perfil.
RF14	El administrador tendrá acceso a los datos de los perfiles de los usuarios registrados de ser estrictamente necesario.
RF15	El Administrador será el encargado de gestionar las categorías, de tal modo que tras haberse autenticado será capaz de crear, editar o eliminar una determinada categoría.
RF16	El Administrador será el encargado de gestionar los municipios, de tal modo que tras haberse autenticado será capaz de crear, editar o eliminar un determinado municipio.
RF17	El Administrador será el encargado de gestionar los barrios, de tal modo que tras haberse autenticado será capaz de crear, editar o eliminar un determinado barrio.
RF18	El administrador tendrá acceso a los datos de los productos publicados por los usuarios registrados, de ser este contenido inapropiado será eliminado por el Administrador de la aplicación.
Usuario No Registrado	
RF19	El usuario no registrado deberá registrar sus datos a través de un formulario para posteriormente ingresar al sistema y hacer usos de sus funcionalidades.
RF20	Tendrá acceso a visualizar los datos de los productos publicados por los usuarios registrados.
RF21	Podrá navegar entre las categorías existentes así como los campos de búsqueda por municipio, barrios, nombre del producto y rango de precio.
Usuario Registrado	



RF22	Los usuarios deberán ser capaz de acceder al sistema mediante el uso de su nombre de usuario y contraseña.
RF23	Los usuarios podrán acceder a su perfil para realizar la modificación de su información personal.
RF24	Los usuarios podrán publicar los datos de sus productos especificando los detalles solicitados en el formulario de ingreso.
RF25	Tendrá acceso a visualizar los datos de los productos publicados por otros usuarios registrados.
RF26	Tendrán acceso al panel de administración de sus productos, donde serán capaces de actualizar o eliminar la información deseada.
RF27	Podrá navegar entre las categorías existentes así como los campos de búsqueda por municipio, barrios, nombre del producto y rango de precio.

Requerimientos No funcionales

Numero de Requisito	Descripción
RNF1	Diseño adaptable. La interfaz gráfica del sistema debe ser adaptable a dispositivos móviles y a diferentes resoluciones de pantalla
RNF2	Estabilidad del sistema. El sistema debe ser estable para evitar la repetición de las tareas
RNF3	El sistema debe estar diseñado para ser fácilmente usable por todo tipo de usuarios.



Diseño del sistema

Modelo Relacional

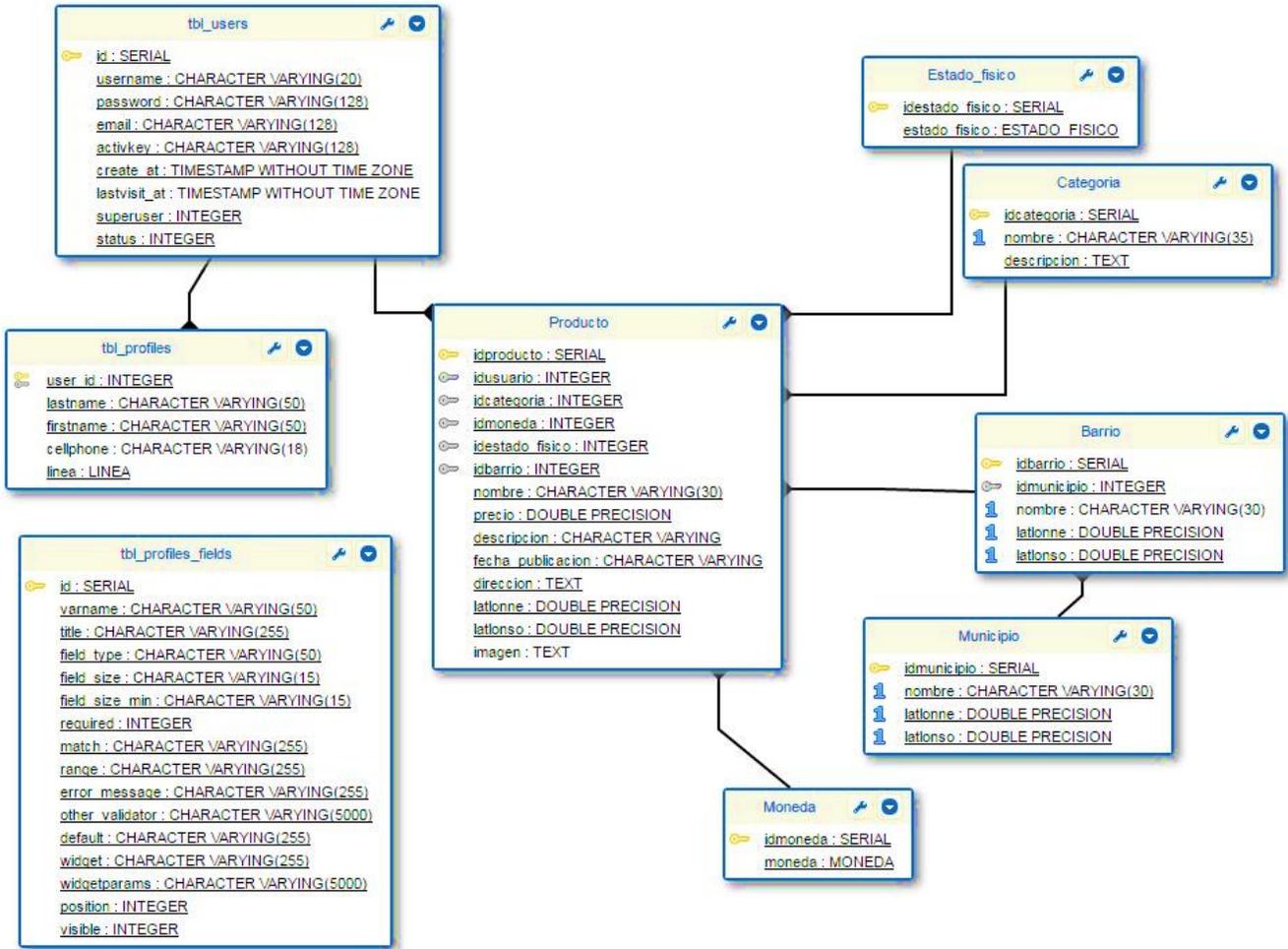


Ilustración 17: Modelo Relacional



Diagrama de casos de usos

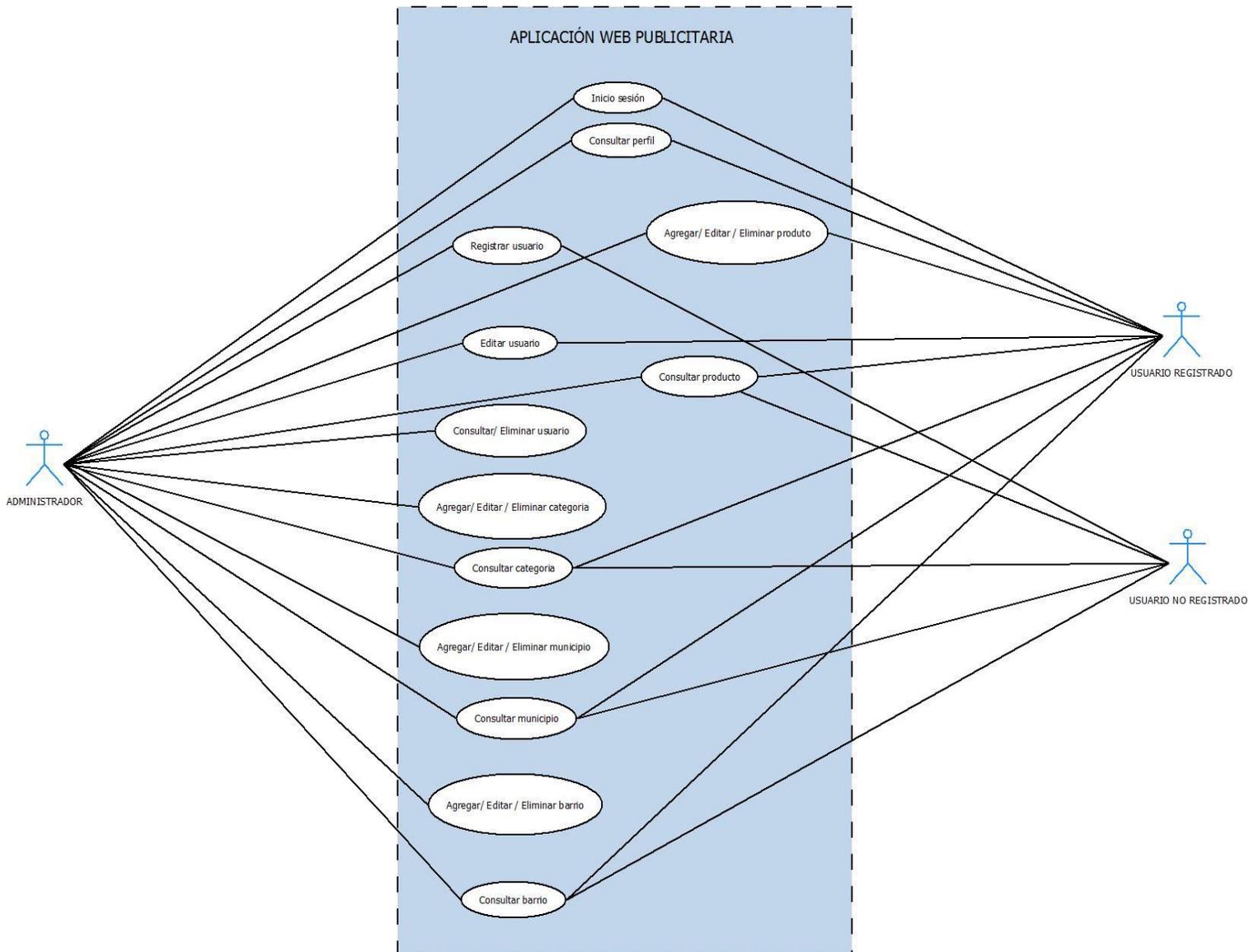


Ilustración 18: Diagrama de casos de usos.



Diagrama de Clases.

El diagrama de clases de la aplicación, al utilizar el Yii framework, corresponde directamente con el modelo conceptual de la base de datos. Habitualmente existen algunas diferencias entre ambos diagramas, al no realizarse una conversión directa de la base de datos a las clases del lenguaje seleccionado. No obstante, en este caso el framework mapea todas las tablas de la base de datos y utiliza un generador para crear una clase correspondiente a cada una de ellas. Por ello, se ha optado por mostrar el diagrama de clases de una entidad en particular (Figura 21), siendo esta extrapolable al resto de entidades de la base de datos.

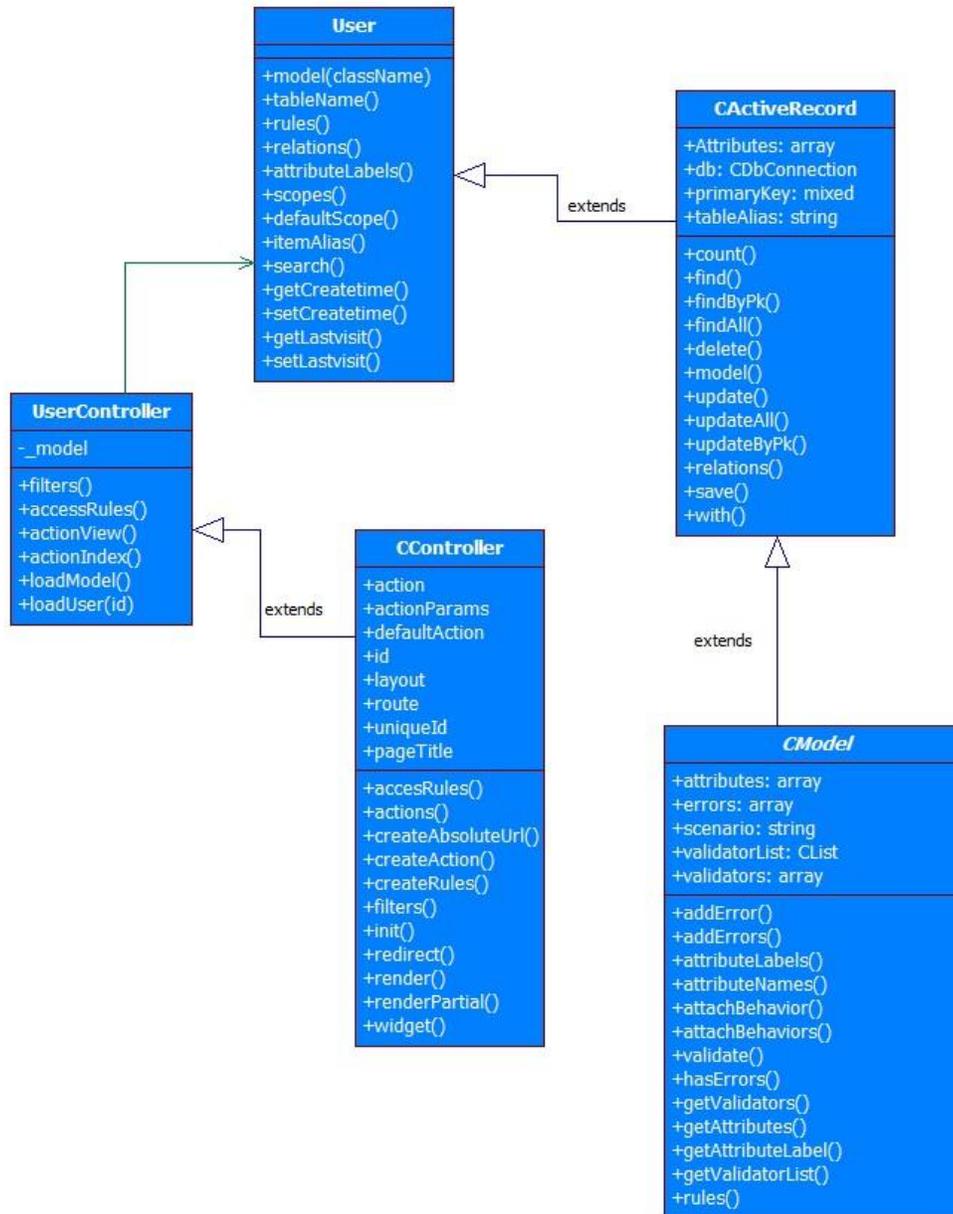


Ilustración 19: Diagrama de Clases. Clase User.



Diagrama de Secuencia.

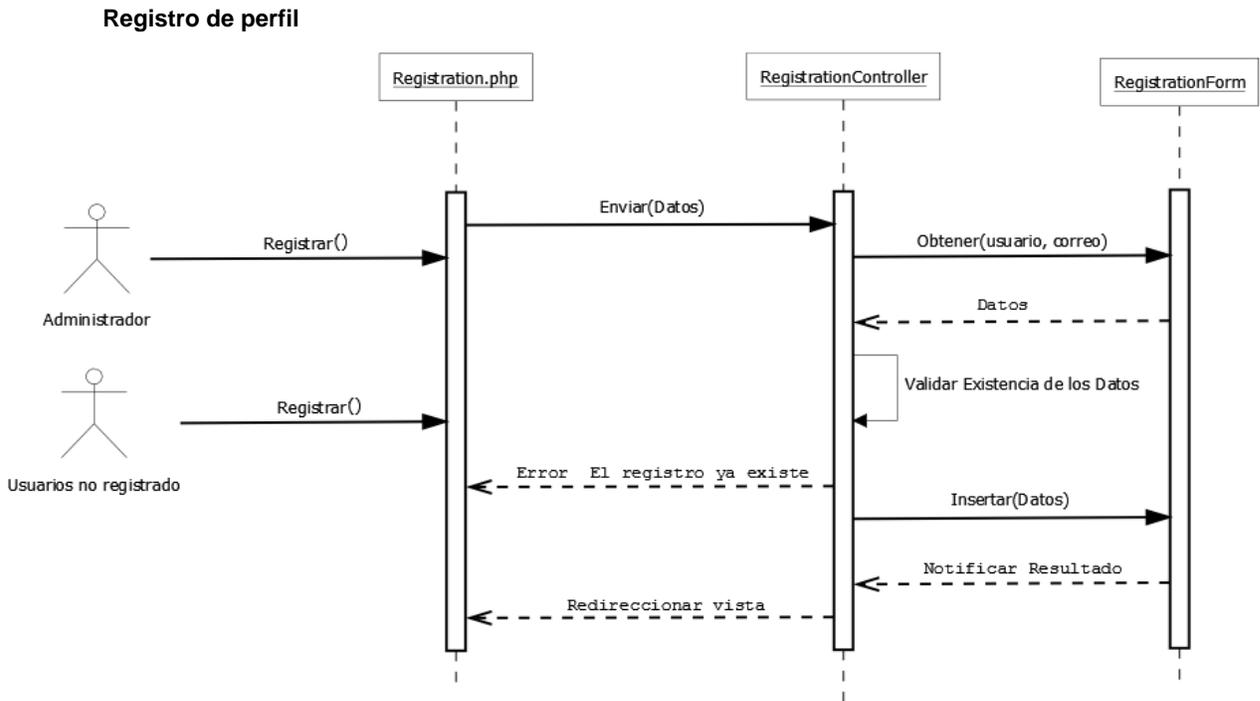


Ilustración 20: Diagrama de Secuencia de Registro de perfil.

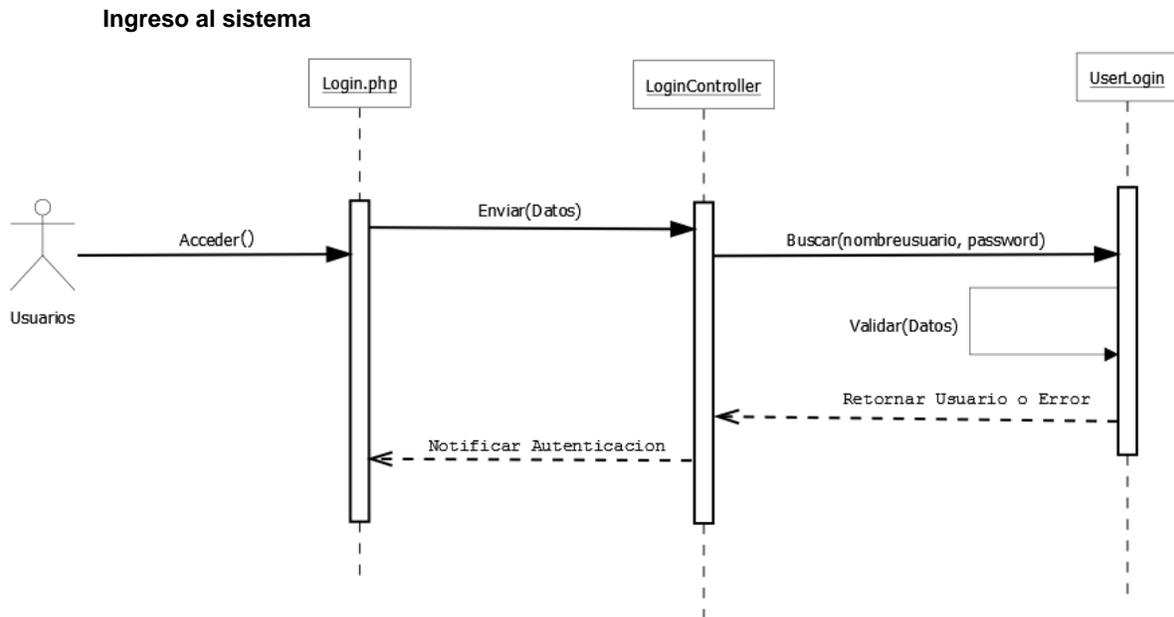


Ilustración 21: Diagrama de Secuencia de Ingreso al Sistema.



Registrar Datos.

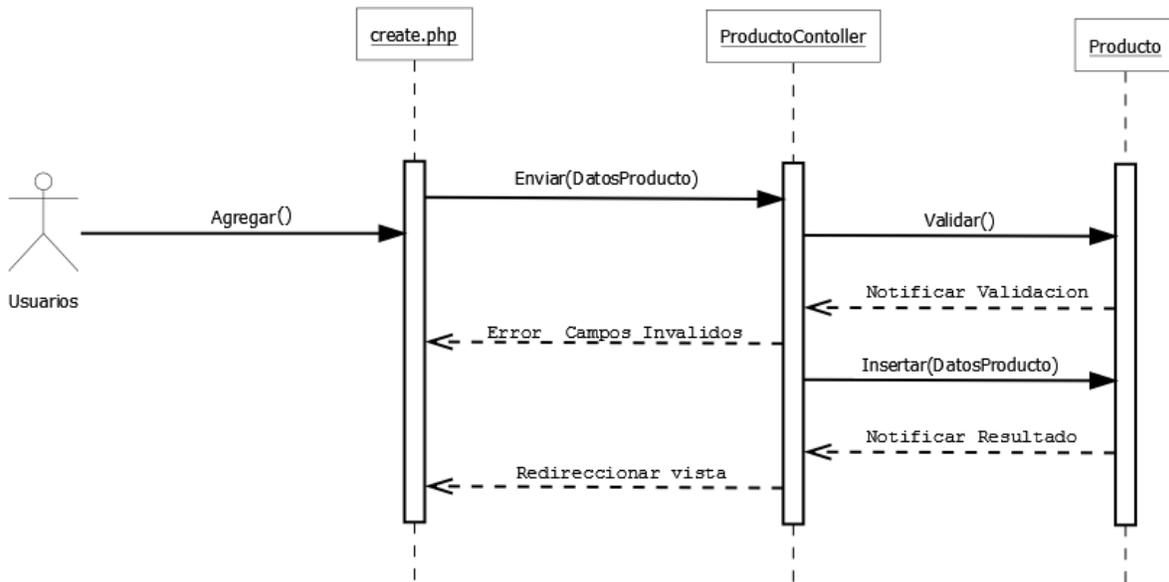


Ilustración 22: Diagrama de Secuencia de Registrar Datos.

Actualizar Datos.

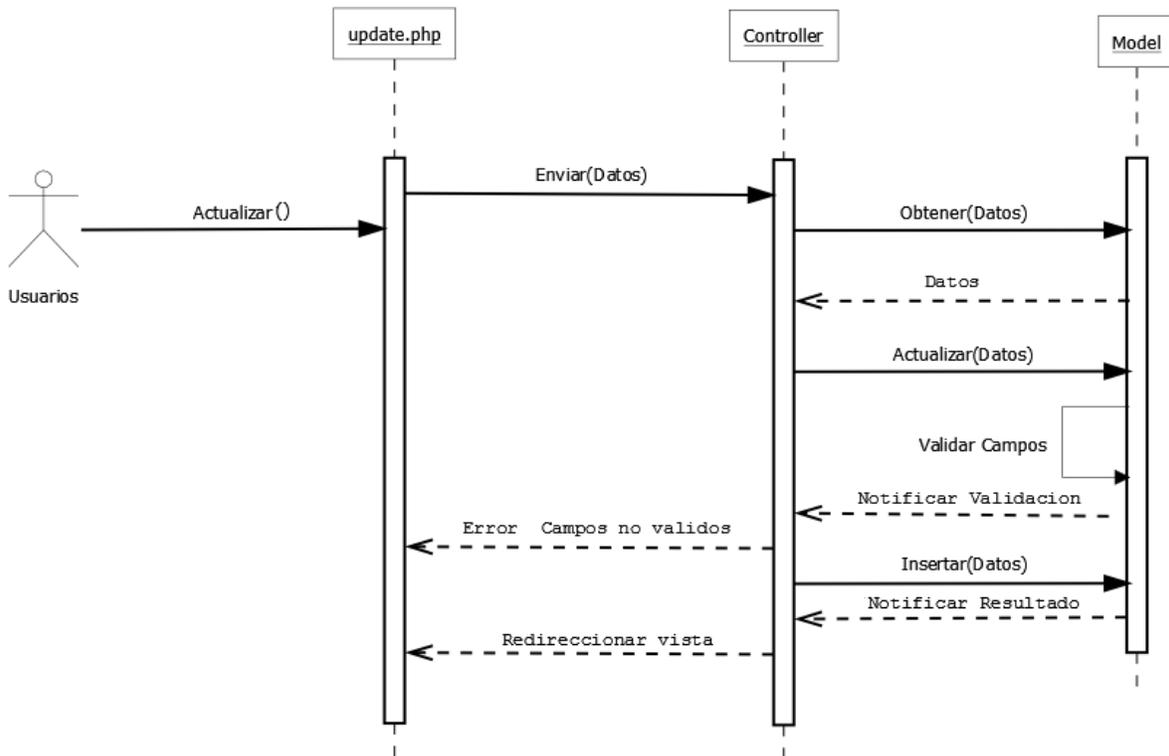


Ilustración 23: Diagrama de Secuencia de Actualizar Datos.



Eliminar Datos.

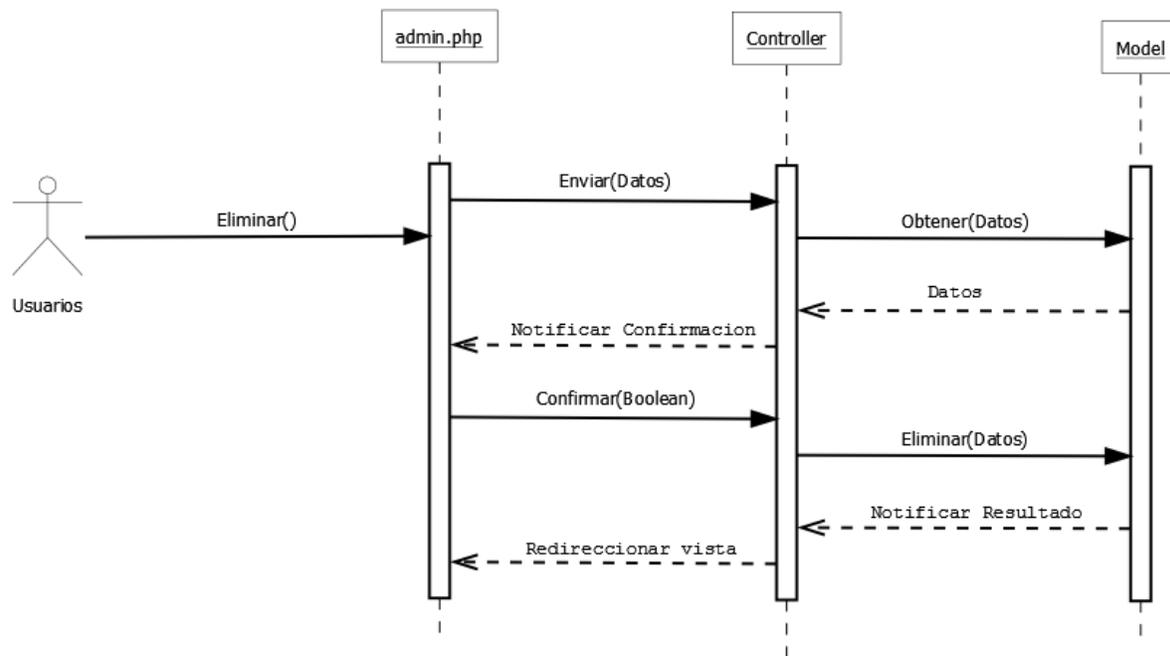


Ilustración 24: Diagrama de Secuencia de Eliminar Datos.

Buscar Datos.

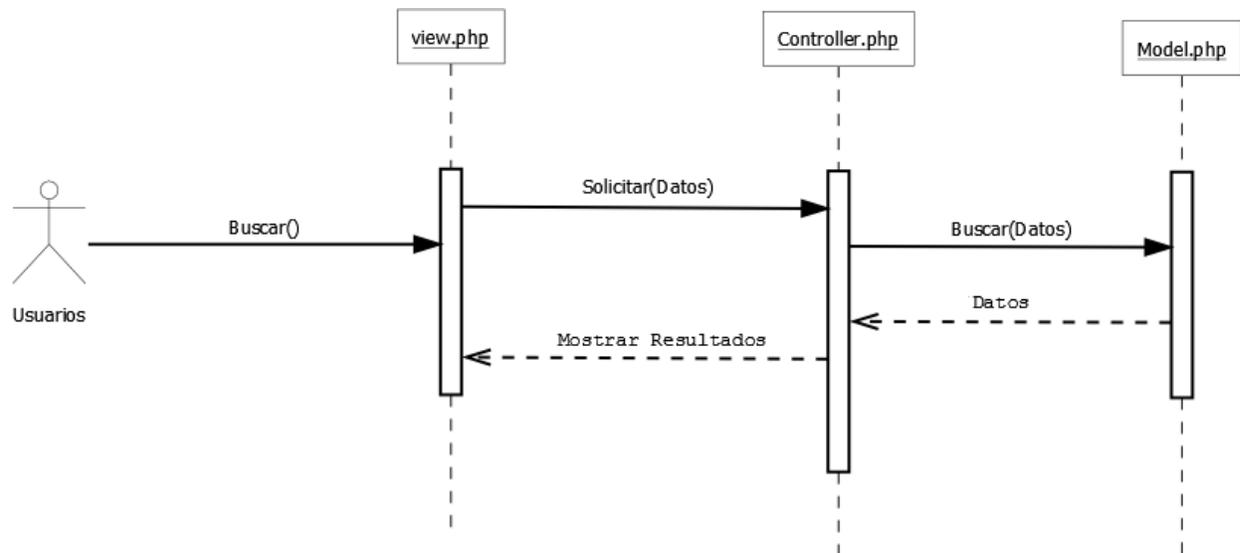


Ilustración 25: Diagrama de Secuencia de Buscar Datos.



INSTALACIÓN DE YII FRAMEWORK

En nuestro apartado de herramienta de desarrollo habíamos explicado que es el framework yii y las diversas ventajas que brinda, en este apartado explicaremos la instalación de este framework, la estructura básica de una aplicación elaborada con este framework y su generador de código llamado Gii.

¿Como puedo descargarlo?

Para instalar y descargar Yii debemos de haber instalado anteriormente nuestro servidor web Apache, una vez que tengamos acceso a nuestro servidor seguiremos una serie de pasos sencillo que se describen a continuación:

- Ir a la carpeta donde se descargara el framework Yii:

```
cd /var/www/
```

- Una vez ubicado en este directorio descargamos la versión de Yii más estable:

```
wget https://github.com/yiisoft/yii/releases/download/1.1.17/yii-1.1.17.467ff50.zip
```

- A continuación se procede a descomprimir el archivo descargado en el directorio especificado y se cambia los permisos para que solo el usuario propietario tenga permisos de escritura, los demas usuarios solo tendrán permisos de lectura y ejecución:

```
unzip yii-1.1.17.467ff50.zip -d /var/www/  
chmod -R 755 /var/www/ yii-1.1.17.467ff50
```

- Por ultimo se procedera a crear un proyecto:

```
php yiic.php webapp /var/www/miPrimerApp
```

Preguntará si esta de acuerdo con la ruta donde se creara la aplicación le damos que si para que proceda con la creación.

Una vez que terminemos de realizar los pasos anteriores, tendremos una página de bienvenida con las siguientes pestañas para navegar como about, contact, login. Viendose de la siguiente forma:

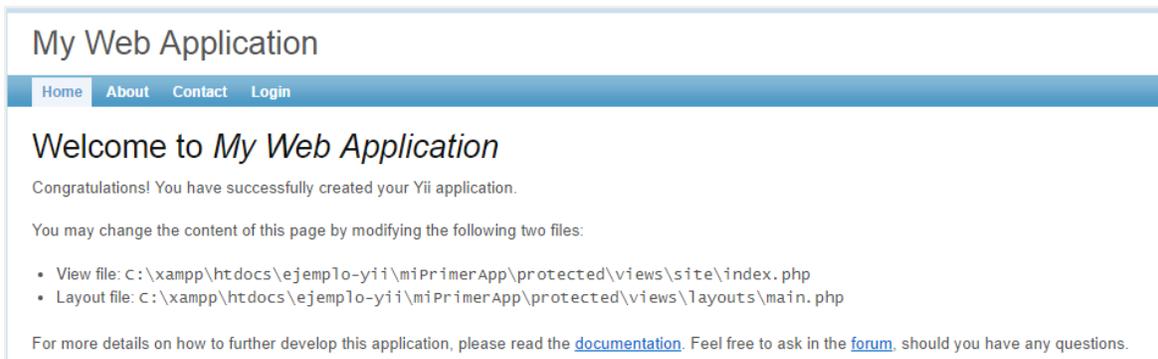


Ilustración 26: Página de Bienvenida de Yii Framework



Estructura básica creada por el framework Yii

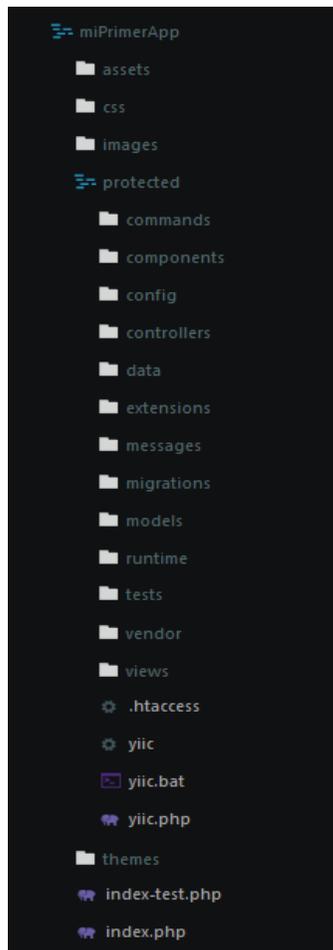


Ilustración 27: Estructura básica.

Esta imagen muestra la estructura de los directorios de nuestra aplicación que son utilizados para cumplir varios propósitos. Cada uno de estos pueden ser personalizados en caso de necesitarse:

Protected: Este es el directorio base de la aplicación el cual contiene todos los archivos de scripts PHP y de datos sensibles a la seguridad. Yii crea un alias predeterminado llamado `application` asociado con esta ruta. Este directorio y todo lo que se encuentra dentro de él debe ser protegido de poder ser accedido por los usuarios Web. Puede ser personalizado vía `CWebApplication::basePath`.

Protected/runtime: Este directorio contiene archivos privados y temporarios generados durante el tiempo de ejecución de la aplicación. El proceso de servidor Web debe tener acceso de escritura en el mismo. Puede ser personalizado vía `CApplication::runtimePath`.

Protected/extensions: En el se encuentran todas las extensiones de terceros. Puede ser personalizado vía `CApplication::extensionPath`.

Protected/modules: Este posee todos los módulos de la aplicación cada uno representado por un subdirectorio.



Protected/Components: Se ubican los componentes utilizados en la aplicación, normalmente se ubica el UserIdentity y el Controller aunque en algunos casos se agregan otros que sean necesarios dependiendo de la aplicación por ejemplo algún componente necesario para servicios web.

Protected/Config: Se encuentran los archivos de configuración de la aplicación, la conexión de la base de datos, el lenguaje de la aplicación, el nombre, etc.

Protected/controllers: Directorio que contiene todos los archivos de clase controlador. Puede ser personalizado vía **CWebApplication::ControllerPath**.

Protected/views: Aquí encontraremos todos los archivos de vista de controladores, archivos de vista de esquema (layout) y los distintos sitios de la aplicación. Puede ser personalizado vía **CWebApplication::ViewPath**.

Protected/views/layouts: Este directorio contiene todos los archivos de vista de diseño (layout). Puede ser personalizado vía **CWebApplication::LayoutPath**.

Assets: Archivos asset privados que pueden ser publicados para convertirse en accesible para los usuarios Web. Este directorio debe tener permisos de escritura habilitados para el proceso de servidor Web. Puede ser modificado vía **CAssetManager::BasePath**.

Themes: Contiene varios temas que pueden ser aplicados a la aplicación. Puede ser personalizado vía **CThemeManager::BasePath**.

Css: Se encuentran todas las hojas de estilo que se utilizarán para dar un mejor diseño a nuestra aplicación.

Images: Se encuentran todas las imágenes que se utilizarán, los iconos, logotipos, etc.

Configurando generador de código

Una de las herramientas que trae el framework Yii es su generador de código Gii que está implementado como un módulo dentro de nuestra aplicación Yii existente, puede simplificar en gran medida la fabricación de los modelos, controladores y vistas a utilizar dentro de nuestro proyecto web.

Para utilizar esta herramienta debemos configurar nuestro archivo que se encuentra ubicado en **/protected/config/main.php**

Debemos modificarlo de la siguiente manera:



```
'modules'=>array(  
    // uncomment the following to enable the Gii tool  
  
    'gii'=>array(  
        'class'=>'system.gii.GiiModule',  
        'password'=>'123',  
        // If removed, Gii defaults to localhost only. Edit carefully to taste.  
        'ipFilters'=>array('127.0.0.1',':::1'),  
    ),  
),
```

Ilustración 28: Configuración del módulo Gii.

Esta sección suele estar comentada por lo que debemos de descomentarla y especificar nuestra clave que en nuestro ejemplo ingresamos un clave fácil de recordar pero lo recomendado es que sea una clave difícil de decifrar por cuestiones de seguridad.

Por razones de seguridad este módulo solo es accesible desde nuestra servidor local (127.0.0.1), si deseamos que sea accesible desde otras computadoras hay que añadir las respectivas IP's. Se recomienda que cuando nuestro proyecto pase a producción, comentemos nuevamente esta sección.

Agregando YiiBooster

Es una extensión de Yii que nos permite configurar nuestras vistas con el famoso framework de Boostrops mencionado en el apartado de herramienta de desarrollo. Mostramos como instalar esta extensión a nuestra aplicación:

- Descargamos YiiBooster de la siguiente dirección
<https://sourceforge.net/projects/yii booster/files/latest/download?source=files>
- Una vez descargada descomprimos a la carpeta protected/extensions y configuramos el archivo main para implementar las funcionalidades de diseño de Boostrops



```
// preloading 'log' component
'preload'=>array('log','booster'),

// autoloading model and component classes
'import'=>array(
    'application.models.*',
    'application.components.*',
),

'modules'=>array(
    // uncomment the following to enable the Gii tool

    'gii'=>array(
        'class'=>'system.gii.GiiModule',
        'password'=>'123',
        // If removed, Gii defaults to localhost only. Edit carefully to taste.
        'ipFilters'=>array('127.0.0.1','::1'),
        'generatorPaths'=>array('ext.bootstrap.gii'),
    ),
),

// application components
'components'=>array(
    'user'=>array(
        // enable cookie-based authentication
        'allowAutoLogin'=>true,
    ),
    'booster'=>array(
        'class'=>'ext.bootstrap.components.Booster',
    ),
),
```

Ilustración 29: Especificando rutas de alias.

Agregamos a la configuración `components` como clave-valor llamado **booster**, este valor lo cargamos en **preload** y en nuestro generador de código **Gii** agregamos la ruta **extensiones/bootstrap/gii**, para que implemente la plantilla de diseño de bootstrap cuando creamos un **CRUD**.

Creando nuestro primer CRUD con Gii

Para utilizar la herramienta de **Gii** primero tendremos que realizar las conexiones con la base de datos a trabajar, en nuestro caso realizamos una base de datos llamado **pruebaGii** con una sola tabla llamada **tbl_user**.

Configuramos el archivo **/protected/config/database.php** e ingresamos las configuraciones para que nuestra aplicación se conecte con nuestra base de datos, viéndose de la siguiente forma:



```
<?php
// This is the database connection configuration.
return array(
    // 'connectionString' => 'sqlite:'.dirname(__FILE__).'/../data/testdrive.db',
    // uncomment the following lines to use a MySQL database

    'connectionString' => 'pgsql:host=localhost;dbname=pruebaGii',
    'emulatePrepare' => true,
    'username' => 'postgres',
    'password' => 'underunan',
    'charset' => 'utf8',
);
```

Ilustración 30: Configuración conexión con el SGBD.

Para crear un **CRUD** (Create Read Update Delete) con **Gii** debemos de ingresar a la siguiente url **localhost/miPrimerApp/index.php?r=gii** e ingresamos la contraseña que configuramos en el archivo **main.php** anteriormente, como resultado nos muestra la siguiente imagen:

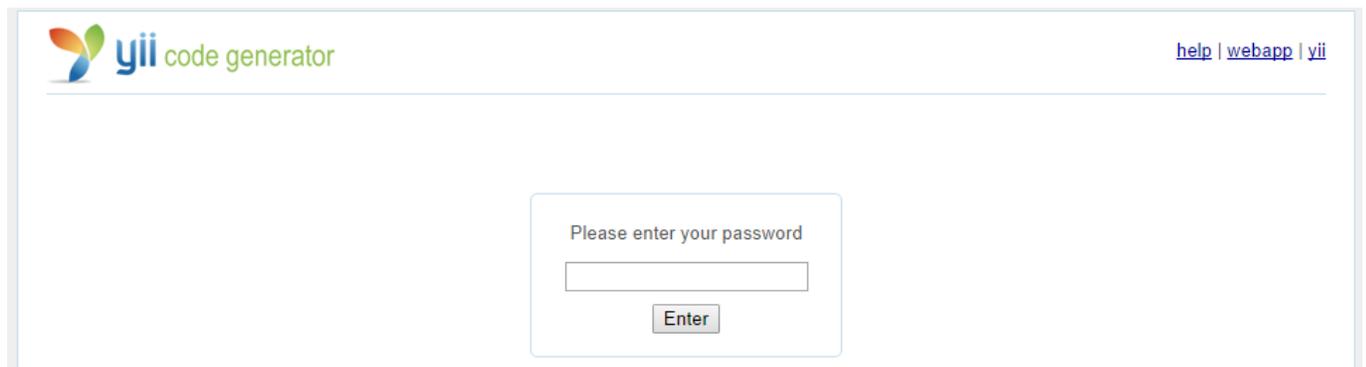


Ilustración 31: Login para realizar un CRUD con Gii.

A continuación nos mostrara una imagen de bienvenida y los distintos generadores de codigos para construir rapidamente nuestra aplicación:



Welcome to Yii Code Generator!

You may use the following generators to quickly build up your Yii application:

- [Controller Generator](#)
- [Crud Generator](#)
- [Form Generator](#)
- [Model Generator](#)
- [Module Generator](#)
- [Bootstrap Generator](#)

Ilustración 32: Opciones para generar código.

- **Controller Generator:** Ayuda a generar rápidamente una nueva clase de controlador, una o varias acciones del controlador y sus correspondientes vistas.
- **Crud Generator:** Genera un controlador y las vista que implementan las operaciones CRUD (Create Read Update Delete), para el modelo de datos especificado.
- **Form Generator:** Genera un archivo de script para la vista que muestra un formulario para recoger los datos para la clase del modelo especificado.
- **Model Generator:** Genera una clase del modelo para la tabla de la base de datos especificada.
- **Module Generator:** Ayuda a generar el código del esqueleto de un módulo.
- **Bootstrap Generator:** Genera un controlador y vistas para implementar las operaciones CRUD de un modelo en específico.



Ya que conocemos cada una de las opciones que nos genera **Gii** empecemos con crear nuestro modelo de datos, primero vamos a ver la opción **Model Generator** desde la cual se realiza el mapeo de las tablas que están en nuestra base de datos a las clases (.php) que representan nuestro modelo, como es el caso **tbl_user**, **Gii** nos muestra la clase que se va a generar y que en su contenido se escribiría el código necesario para establecer este mapeo.

The screenshot shows the Gii Model Generator interface. On the left is a sidebar with a 'Generators' menu where 'Model Generator' is selected. The main area is titled 'Model Generator' and contains the following configuration options:

- Database Connection ***: db
- Table Prefix**: [empty]
- Table Name ***: tbl_user
- Model Class ***: TblUser
- Base Class ***: CActiveRecord
- Model Path ***: application.models
- Build Relations**:
- Use Column Comments as Attribute Labels**:
- Code Template ***: default (C:\xampp\htdocs\yii\framework\gii\generators\model\ten)

At the bottom, there are 'Preview' and 'Generate' buttons. Below them is a table showing the generated code file:

Code File	Generate
models\TblUser.php	unchanged

Ilustración 33: Generador de modelo.



Una vez que hemos creado nuestro modelo vamos a generar una parte muy interesante de nuestra aplicación que es la administración de usuarios registrados. Para ello vamos a la opción Bootstrap Generator, como mencionamos anteriormente nos permite generar las opciones de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar).

The screenshot shows the Bootstrap Generator interface. On the left is a sidebar with 'Generators' and a list of options including 'Controller Generator'. The main area is titled 'Bootstrap Generator' and contains a description: 'This generator generates a controller and views that implement CRUD operations for the specified data model.' Below this, there are several fields: 'Model Class *' with the value 'TblUser', 'Controller ID *' with 'tblUser', and 'Base Controller Class *' with 'Controller'. A 'Code Template *' field shows a default path. At the bottom, there are 'Preview' and 'Generate' buttons. Below the buttons is a table listing the generated files:

Code File	Generate
	<input checked="" type="checkbox"/>
controllers\TblUserController.php	new <input checked="" type="checkbox"/>
views\tblUser_form.php	new <input checked="" type="checkbox"/>
views\tblUser_search.php	new <input checked="" type="checkbox"/>
views\tblUser_view.php	new <input checked="" type="checkbox"/>
views\tblUser\admin.php	new <input checked="" type="checkbox"/>
views\tblUser\create.php	new <input checked="" type="checkbox"/>
views\tblUser\index.php	new <input checked="" type="checkbox"/>
views\tblUser\update.php	new <input checked="" type="checkbox"/>
views\tblUser\view.php	new <input checked="" type="checkbox"/>

Ilustración 34: Generador CRUD de Gii.

Como podemos ver en la imagen anterior nos ha creado la clase controlador **TblUserController.php** de esta clase se deriva los siguientes métodos:

```
public function actionIndex() {...}
public function actionView($id) {...}
public function actionCreate() {...}
public function actionUpdate($id) {...}
public function actionDelete($id) {...}
```



Estos métodos son invocados desde la vistas para realizar las operaciones **CRUD** contra la tabla **tbl_user** y por supuesto se pueden modificar según nuestras necesidades. Pulsemos “**Generate**” para que **Gii** autogenera nuestro código, como muestra la siguiente imagen:

The controller has been generated successfully. You may [try it now](#).

```
Generating code using template "C:\xampp\htdocs\yii\framework\gii\generat
generated controllers\TblUserController.php
generated views\tblUser\_form.php
generated views\tblUser\_search.php
generated views\tblUser\_view.php
generated views\tblUser\admin.php
generated views\tblUser\create.php
generated views\tblUser\index.php
generated views\tblUser\update.php
generated views\tblUser\view.php
done!
```

Ilustración 35: Mensaje satisfactorio de creación CRUD.

Como resultado obtenemos la creación de las vistas de diseño de nuestra administración de la tabla **tblUser**.

Home » Tbl Users

Tbl Users

Operations

- Create TblUser
- Manage TblUser

Displaying 1-2 of 2 results.

ID: 1 Username: horacio Password: 123 Email: horacio@hotmail.es
ID: 2 Username: javier Password: 12345 Email: javier@hotmail.com

Ilustración 36: Vista del listado de todos los usuarios.



Home » Tbl Users » 1

View TblUser #1

ID	1
Username	horacio
Password	123
Email	horacio@hotmail.es

- Operations
- List TblUser
- Create TblUser
- Update TblUser
- Delete TblUser
- Manage TblUser

Ilustración 37: Vista ver un usuario.

Home » Tbl Users » Create

Create TblUser

Fields with * are required.

Username *

Password *

Email *

- Operations
- List TblUser
- Manage TblUser

Ilustración 38: Vista de crear un nuevo usuario.

Home » Tbl Users » Manage

Manage Tbl Users

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-2 of 2 results.

ID	Username	Password	Email	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	horacio	123	horacio@hotmail.es	
2	javier	12345	javier@hotmail.com	

- Operations
- List TblUser
- Create TblUser

Ilustración 39: Vista gestión de usuarios.



Home » Tbl Users » 0 » Update

Update TblUser 1

Fields with * are required.

Username *

Password *

Email *

Operations

- List TblUser
- Create TblUser
- View TblUser
- Manage TblUser

Ilustración 40: Vista actualizar datos usuario.

Módulo Yii-user

Un módulo en Yii Framework es una unidad auto-contenida de software que tiene sus propios modelos, vistas, controladores y demás componentes de soporte. Estos pueden ser accedidos de la misma forma como se acceden a los controladores convencionales, además resultan útiles para dividir el desarrollo y mantenimiento de aplicaciones muy grandes en módulos independientes, facilitan la reutilización de código a alto nivel y se utilizan para implementar funcionalidades como autenticación, manejo de usuarios, manejo de comentarios, etc.

Yii-user

Es una extensión de Yii Framework destinada al registro y administración de perfiles de usuarios. Este puede ser descargado a través del siguiente enlace: www.yiiframework.com/extension/yii-user/.

Este módulo nos brinda las siguientes utilidades:

- Logarse por medio del nombre de usuario o email
- Registro
- Activación de perfil (Verificación de email, administrador o no)
- Recuperación de contraseñas (Envío de clave de recuperación al email del usuario)
- Página de perfil de usuario
- Uso de perfiles para cada usuario
- Administración de usuarios
- Widget en el campo de perfil para ver, editar y guardar los datos
- Widget para la carga de archivos y campos Datetime (jQueryUI datepicker)
- Manejo básico de auditorías a través de campos que permiten ver y controlar la interacción de los usuarios con la aplicación
- API

Requerimientos

Para utilizar el módulo Yii-user es necesario Yii Framework 1.1 o una versión superior.



Instalación

El proceso de instalación es relativamente sencillo, simplemente se debe descomprimir el archivo descargado **yii-user-0.3-61-gfc69518.zip** en el directorio protected de nuestra aplicación.

Luego se debe editar la configuración del archivo **main.php**

```
11     'modules'=>array(  
12         #...  
13         'user'=>array(  
14             # Método de encriptación (php hash function)  
15             'hash' => 'md5',  
16             # Activar el envío de email  
17             'sendActivationMail' => false,  
18             # Permitir el acceso a usuarios inactivos  
19             'loginNotActiv' => false,  
20             # Activar usuario despues del registro  
21             'activeAfterRegister' => true,  
22             # Inicio de sesión automático despues del registro  
23             'autoLogin' => true,  
24             # Ruta del registro  
25             'registrationUrl' => array('/user/registration'),  
26             # Ruta para la recuperacion de contraseña  
27             'recoveryUrl' => array('/user/recovery'),  
28             # Inicio de sesión desde la ruta  
29             'loginUrl' => array('/user/login'),  
30             # Pagina a redireccionar despues del inicio de sesión  
31             'returnUrl' => array('/user/profile'),  
32             # Pagina a redireccionar despues de cerrar la sesión  
33             'returnLogoutUrl' => array('/user/login'),  
34         ),  
35         #...  
36     ),  
37     #...  
38     'user'=>array(  
39         // Activar autenticación basada en cookie  
40         'class' => 'WebUser',  
41         'allowAutoLogin'=>true,  
42         'loginUrl' => array('/user/login'),  
43     ),  
44     #...  
45 ),
```

Ilustración 41: Modulo de usuarios.