



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Sistema electrónico de presentación de solicitudes de marcas ABLEIP, en plataforma web.**

**TESIS**

Que para obtener el título de  
**Ingeniero en Computación**

**P R E S E N T A**

Raymi Valdés Fernández

**DIRECTOR DE TESIS**

Ing. Cruz Sergio Aguilar Díaz



Ciudad Universitaria, Cd. Mx., 2017



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# ÍNDICE

INTRODUCCIÓN .....	4
CAPÍTULO 1 ANTECEDENTES .....	5
1.1 PANORAMA GENERAL .....	5
1.2 OBJETIVO GENERAL .....	6
1.3 FINALIDAD DEL SISTEMA.....	7
CAPÍTULO 2 IDENTIFICACIÓN DE LA PROBLEMÁTICA .....	8
2.1 ANÁLISIS DEL PROCESO DE REGISTRO EN LA ACTUALIDAD.....	8
2.2 DEFINICIÓN DE LOS ALCANCES DEL SISTEMA .....	9
2.3 DEFINICIÓN DE LOS REQUERIMIENTOS.....	11
CAPÍTULO 3 FUNDAMENTOS TEÓRICOS.....	13
3.1 INGENIERÍA DE SOFTWARE .....	13
3.2 METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS.....	16
3.3 LENGUAJES DE PROGRAMACIÓN.....	24
3.4 BASES DE DATOS.....	40
3.5 VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE .....	51
3.6 PRUEBAS A REALIZAR .....	53
CAPÍTULO 4 ANÁLISIS Y DISEÑO DEL SISTEMA .....	56
4.1 JUSTIFICACIÓN DE LAS HERRAMIENTAS .....	56
4.2 ANÁLISIS Y DISEÑO DEL SISTEMA .....	60
4.3 ORGANIZACIÓN Y DEPURACIÓN DE LA INFORMACIÓN.....	66
4.4 DISEÑO DE MÓDULOS .....	69
CAPÍTULO 5 IMPLEMENTACIÓN DEL SISTEMA .....	72
5.1 CREACIÓN DE LA BASE DE DATOS.....	72
5.2 INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR DE DESARROLLO....	73
5.3 INSTALACIÓN DE WAMP SERVER 2.0 .....	74
5.4 MÓDULO DE REGISTRO Y VALIDACIÓN DE USUARIO .....	75
5.5 MÓDULO DE ADMINISTRACIÓN DE USUARIOS.....	77
5.6 MÓDULO DE MARCAS .....	80
5.7 MÓDULO DE ADMINISTRACIÓN DE ARCHIVOS.....	85
CAPÍTULO 6 PLAN DE PRUEBAS.....	87
6.1 EJECUCIÓN DE PRUEBAS .....	87
RESULTADOS.....	90
CONCLUSIONES.....	92
APÉNDICES .....	94
APÉNDICE A.....	94
APÉNDICE B.....	96

APÉNDICE C.....	99
GLOSARIO .....	110
REFERENCIAS ELECTRÓNICAS.....	113
REFERENCIAS BIBLIOGRÁFICAS.....	114

# INTRODUCCIÓN

El trabajo que se explica a lo largo de este documento, no es más que el resultado de la creación de un sistema cuya función principal será brindar servicios en el campo de la Propiedad Industrial, básicamente, servicios de marcas en todas sus vertientes y posibilidades.

El sistema se desarrollará en un ambiente Web y puesto a disposición de los posibles clientes a través de Internet, de manera que ellos puedan acceder e interactuar con el mismo teniendo así la posibilidad de disfrutar de nuestros servicios. Es indispensable que el sitio Web como tal, cumpla con normas de seguridad de manera que los datos de los clientes estén completamente protegidos y que nuestras bases de datos estén dotadas también de ciertas medidas de seguridad que velen porque nadie que no sea personal autorizado pueda acceder a su contenido, así como la permanente disponibilidad de los servicios que brinda este sistema para lograr una experiencia de usuario esperada.

El capítulo uno muestra un panorama general del sistema, dando a conocer el objetivo de la realización del mismo, así como la finalidad del sistema.

El capítulo dos muestra los antecedentes y los problemas existentes y de cómo se realizan los procesos actualmente.

El capítulo tres plantea todos los fundamentos teóricos necesarios para llevar a cabo este sistema.

El capítulo cuatro muestra el análisis y diseño del sistema propuesto para la realización del sistema.

El capítulo cinco ya muestra la implementación del sistema desde la instalación hasta la explicación de cada módulo.

El capítulo seis muestra las pruebas realizadas al sistema.

# **CAPÍTULO 1 ANTECEDENTES**

## **1.1 PANORAMA GENERAL**

El desarrollo y el uso creciente de Internet han impactado a las sociedades y empresas en todo el mundo. Sólo los países desarrollados han adoptado esta tecnología desde el principio, pero ahora es frecuente en todas partes. El comercio electrónico permite a las empresas pasar a través de fronteras sin necesidad de estar físicamente en un lugar. Esto ha dado lugar a la eficiencia del negocio y abrió nuevas vías de comercio.

### **La herramienta del comercio**

El Internet reduce el tiempo de comercialización y disminuye los costos de comunicación, por lo que es una herramienta de comercio valorado por las empresas internacionales. La entrega de información de Internet en un formato digital a bajo costo ayuda a las empresas operar con mayor eficiencia, y pueden pasar los ahorros de costes para el cliente.

### **Eficiencias de la cadena de suministro**

Las empresas dedicadas al comercio internacional encuentran que Internet les ayuda a mejorar la gestión de la cadena de suministro para sus productos y servicios. Esto reduce los gastos generales asociados con las funciones de mercadeo, de distribución y de transporte. El Internet es la mejora de la eficiencia de la distribución comercial y la logística, que permite el desplazamiento de la información sin la necesidad de transporte, seguro ni otros gastos. Estas eficiencias de la cadena de suministro hacen que sea más fácil vender productos, como prendas de vestir y de salud, a través del comercio electrónico.

## Negocios Rurales y Pequeñas Empresas

Internet permite a las pequeñas empresas acceder a los clientes en todo el mundo sin gastos de transporte. Internet provee un impacto igualador, una pequeña empresa puede utilizar varios efectos especiales en su sitio web, al igual que un negocio más grande. Empresas rurales y regionales también se benefician del comercio electrónico y la reducción de los costos de transporte. Las empresas pueden agruparse para presentar información sobre sus especialidades regionales y de atender las consultas de los clientes. Por ejemplo, una Cámara de Comercio local podría proporcionar información sobre las pequeñas empresas de la región.

### 1.2 OBJETIVO GENERAL

El objetivo de este proyecto es proporcionar un sistema electrónico en plataforma Web que facilite el proceso de registro de marcas, que permita la administración de los documentos que se generan a lo largo de todo el proceso del registro de marca, y así establecer una comunicación directa con el cliente, además de proporcionar un servicio amigable y de fácil funcionamiento para el usuario.

El sistema contará con varios módulos los cuales serán mencionados más adelante, y permitirán entre otras cosas realizar cotizaciones de servicios que la empresa **ABLEIP** lleva a cabo, efectuar llenado de formatos de presentaciones de solicitudes de registro de marca, administrar los documentos que se van generando durante todo el proceso del servicio solicitado por el cliente, crear facturas y notas de débitos para la realización de los pagos pertinentes de cada servicio, enviar y recibir correos electrónicos desde la plataforma, entre otras funcionalidades.

## OBJETIVOS ESPECÍFICOS

- Cálculo de cotización de la Solicitud del Registro de Marcas.
- Solicitud del Registro de Marcas.
- Rastreo del Estado de Trámites.
- Reportes de altas/bajas/cambios en el registro de casos.
- Administración de archivos.
- Sistema de notificaciones y envío de correos electrónicos.

### 1.3 FINALIDAD DEL SISTEMA

La finalidad de los sistemas de información, como las de cualquier otro sistema dentro de una organización, son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas.

En este caso en particular el sistema electrónico **ABLEIP** tiene como finalidad establecer un canal de comunicación entre el cliente (usuario) y el agente (sistema) mucho más estrecho, y mediante este canal de comunicación directa se puede realizar un intercambio de información de una manera rápida, eficiente y segura, lo cual proporciona un beneficio parejo para las dos partes que intervienen en este proceso.

Este sistema tiene como finalidad establecer un nuevo modelo de negocio en esta rama de los servicios, ya que los servicios legales y de usos intelectuales todavía son propensos al uso de sistemas más arcaicos e ineficientes en comparación con las nuevas tecnologías ya existentes.



# CAPÍTULO 2 IDENTIFICACIÓN DE LA PROBLEMÁTICA

## 2.1 ANÁLISIS DEL PROCESO DE REGISTRO EN LA ACTUALIDAD

El registro de una Marca en México en la actualidad se hace mediante dos métodos, el primero es en línea usando el portal del Instituto Mexicano de la Propiedad (IMPI) en el cual se puede realizar una solicitud de registro de marca siempre y cuando el agente o solicitante de la marca cuente con las credenciales necesarias para poder acceder al portal, dichas credenciales las proporciona el IMPI a los agentes que estén registrados y que proporcionen sus servicios en México de lo contrario no se podrá acceder.

Como segundo método tenemos la más tradicional, la cual es llenar un formato de papel “físico” y llevarlo a las oficinas del instituto y presentarlo en ventanillas, dicho proceso es más ineficiente ya que consume tiempo y recursos, los cuales se pueden ahorrar usando el primer método.

Para cualquiera de los dos métodos se requiere realizar una búsqueda previa de la marca que consiste en la investigación a nivel nacional dentro de la base de datos del IMPI, o solicitada ante dicha autoridad, para verificar la viabilidad o registrabilidad del signo distintivo de interés.

Existen varios tipos de búsquedas, por ejemplo:

### **Búsqueda de identidad**

Consiste, como su nombre lo indica, en hacer una investigación a nivel nacional dentro de la base de datos del IMPI, para verificar si existen o no palabras iguales o idénticas a la propuesta (en cuanto a su escritura).

### **Búsqueda figurativa**

Consiste en la investigación a nivel nacional de aquellos diseños, logos, figuras o letras estilizadas semejantes o iguales a la propuesta (por su forma, colores, diseños y/o

figuras), ya registradas o solicitadas a registro con anticipación, y que puedan obstaculizar el registro de interés.

### **Búsqueda fonética**

Consiste en la investigación a nivel nacional, dentro de la base de datos del IMPI, para verificar si existen o no palabras semejantes o iguales a la propuesta (en cuanto a su fonética o escritura), ya registradas o solicitadas a registro con anticipación, y que puedan obstaculizar el registro de interés.

### **Búsqueda mixta**

Consiste en la investigación a nivel nacional de aquellas marcas propuestas que combinan elementos nominativos y figurativos, a efecto de verificar su registrabilidad en conjunto o en forma global, frente a las demás existentes.

## **2.2 DEFINICIÓN DE LOS ALCANCES DEL SISTEMA**

Las partes de cada fase se refieren a diferentes servicios, pero como veremos en el desarrollo de este proyecto los procedimientos y datos a utilizar de las bases de datos son muy similares. Además, se incluirá como parte de estas fases un sistema de administración de cobro y facturas, así como la posibilidad que el sistema debe brindarle a nuestra empresa de administrar completamente tanto el sistema como la base de datos desde nuestras oficinas.

Este sistema requiere de la autenticación del cliente para poder hacer uso de nuestros servicios, por tanto, a las fases anteriores se le suma el proceso de registro del usuario de manera que quede registrado en nuestra base de datos. Esto será muy ventajoso para el cliente mismo, de modo que una vez introducidos sus datos y esté registrado, no tendrá que agregar sus datos cada vez que quiera hacer uso de nuestro sistema.

El sistema deberá contar también con diferentes posibilidades entre las que están, posibilidad de que los clientes descarguen a su computadora archivos que nuestra empresa va a poner a su disposición a través del sistema de administración del sitio

que debemos tener. Tendrá que estar dotado de la posibilidad de enviar en determinados momentos, reportes y archivos con formatos predefinidos que posteriormente se le enviarán a la persona encargada del desarrollo de este sistema, estos archivos deberán llegar a nuestra empresa siempre como documentos editables (.doc o .pdf editable). Y por último debe constar de un medio donde cada cliente pueda ver el estado de sus casos y de esta manera poder darle seguimiento y deberá estar desarrollado en toda su totalidad en los idiomas inglés y español.

Para que el usuario pueda obtener su nombre de usuario y contraseña se podrá utilizar un generador de usuarios y contraseñas o el programador podrá generar un algoritmo de manera que cumpla con las especificaciones de que no se pueden repetir.

Lo que hemos explicado hasta el momento son, grosso modo, las ideas más generales del proyecto, a continuación, pasaremos a explicar más detalladamente como debe funcionar nuestro sistema según ha sido planificado y estudiado por los expertos en la materia.

Es muy importante destacar antes de pasar a la explicación del sistema como tal que el mismo tendrá un periodo de prueba de 6 meses por fase. Esto quiere decir que después de que el sistema esté terminado y montado para que pueda ser accedido a través de Internet, nuestra empresa necesitará comenzar con un proceso de promoción y publicidad de nuestros servicios, lo cual tomará un tiempo considerable.

En el momento en que los usuarios empiecen a hacer uso del sistema podrán aparecer errores de funcionamiento que el equipo de programadores deberá subsanar como parte del trabajo y que además vendrá incluido en el pago.

## 2.3 DEFINICIÓN DE LOS REQUERIMIENTOS

La tarea de análisis de los requerimientos es un proceso de descubrimiento y refinamiento, el cliente y el desarrollador tienen un papel activo en la ingeniería de requerimientos de software. El cliente intenta plantear un sistema que en muchas ocasiones es confuso para él, sin embargo, es necesario que describa los datos, que especifique las funciones y el comportamiento del sistema que desea. El objetivo es que el desarrollador actúe como un negociador, un interrogador, un consultor, o como persona que consulta y propone para resolver las necesidades del cliente.

Para implementar este sistema necesitamos contar con un servidor Web que permita realizar los registros de las marcas y agilizar los procesos. Dicho servidor deberá contar con las siguientes características:

Hardware:

Componente	Mínimo	Recomendado
Procesador	2,5 GHz	Procesadores duales con una velocidad de 3 GHz o superior cada uno
RAM	1 GB	2 GB
Disco	Sistema de archivos: partición formateada con 40 GB de espacio disponible en disco como mínimo	Sistema de archivos: partición formateada con 120 GB de espacio disponible además del espacio necesario para los sitios web, conjunto redundante de discos Independientes RAID 0 o RAID 1
Red	Conexión de 10/100 Mb por segundo (Kbps) entre los equipos cliente y el servidor	Conexión 1 Gb entre los equipos cliente y el servidor

## Software:

Componente	Mínimo	Recomendado
Sistema Operativo	Cualquier sistema que soporte Apache en su versión 2.1 o superior	GNU/Linux
Servicio	Apache 2.1	Apache 2.4.xx
Base de datos	MySQL	MySQL 5.4.xx
Lenguaje de desarrollo	PHP Javascript HTML jQuery + AJAX	PHP 5.xx Javascript HTML 5 jQuery + AJAX

# **CAPÍTULO 3 FUNDAMENTOS TEÓRICOS**

## **3.1 INGENIERÍA DE SOFTWARE**

Esta disciplina trasciende la actividad de programación, que es el pilar fundamental a la hora de crear una aplicación. El ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto.

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los sistemas.

La ingeniería de software, por lo tanto, incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema.



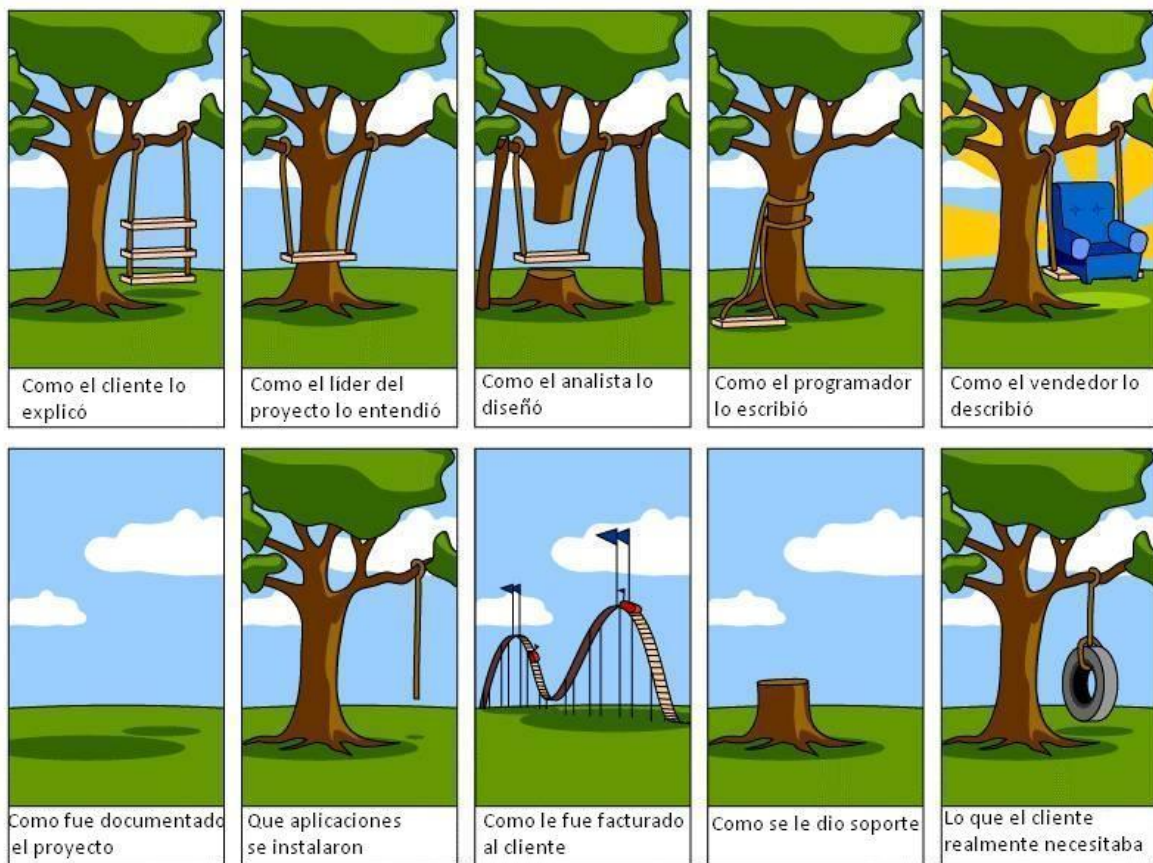
**Fig. 3.1 Etapas de la ingeniería de software.**

Cabe destacar que el proceso de desarrollo de software implica lo que se conoce como ciclo de vida del software, que está formado por cuatro etapas: concepción, elaboración, construcción y transición.

La concepción fija el alcance del proyecto y desarrolla el modelo de negocio; la elaboración define el plan del proyecto, detalla las características y fundamenta la arquitectura, la construcción es el desarrollo del producto, y la transición es la transferencia del producto terminado a los usuarios.

Una vez que se completa este ciclo, entra en juego el mantenimiento del software. Se trata de una fase de esta ingeniería donde se solucionan los errores descubiertos (muchas veces advertidos por los propios usuarios) y se incorporan actualizaciones para hacer frente a los nuevos requisitos.

El proceso de mantenimiento incorpora además nuevos desarrollos, para permitir que el software pueda cumplir con una mayor cantidad de tareas.



**Fig. 3.2 El ejemplo del columpio.**

Un campo directamente relacionado con la ingeniería de software es la arquitectura de sistemas, que consiste en determinar y esquematizar la estructura general del proyecto, diagramando su esqueleto con un grado relativamente alto de especificidad y señalando los distintos componentes que serán necesarios para llevar a cabo el desarrollo, tales como aplicaciones complementarias y bases de datos. La arquitectura de sistemas es un punto fundamental del proceso y es muchas veces la clave del éxito de un producto informático.

Los avances tecnológicos y su repercusión en la vida social han afectado inevitablemente el proceso de desarrollo de software por diversos motivos, como ser el



acceso indiscriminado de los usuarios a cierta información que hasta hace un par de décadas desconocía por completo y que no pueden comprender, dado que no poseen el grado de conocimiento técnico necesario. Un consumidor bien informado es un consumidor al que no se puede timar, ya que sabe lo que necesita y tiene la capacidad de analizar las diferentes ofertas del mercado, comparando las propuestas y prestaciones de los productos; sin embargo, un consumidor mal informado es como un niño caprichoso que llora, grita y patalea sin parar.

La primera de todas las etapas del trabajo que realizan los ingenieros de software consiste en estudiar minuciosamente las características que se creen necesarias para el programa a desarrollar, y es éste el punto en el cual deben encontrar un equilibrio (cada vez más difícil de alcanzar) entre las demandas excesivas de los malos consumidores y las posibilidades de la compañía. El tiempo es dinero, y las empresas del mundo informático lo saben muy bien.

Cada función de un programa, cada rasgo que lo vuelva más cómodo, más inteligente, más accesible, se traduce en una cantidad determinada de tiempo, que a su vez acarrea los sueldos de todas las personas involucradas en su desarrollo. Pero además del costo de producción necesario para realizar cada una de las piezas de un programa, la ingeniería de software debe decidir cuáles de ellas tienen sentido, son coherentes con el resto y son necesarias para comunicar claramente la esencia y los objetivos de la aplicación.

## **3.2 METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS**

Son métodos que indican cómo hacer más eficiente el desarrollo de sistemas de información. Para ello suelen estructurar en fases la vida de dichos sistemas con el fin de facilitar su planificación, desarrollo y mantenimiento.

Las metodologías de desarrollo de sistemas deben definir: objetivos, fases, tareas, productos y responsables, necesarios para la correcta realización del proceso y su seguimiento.

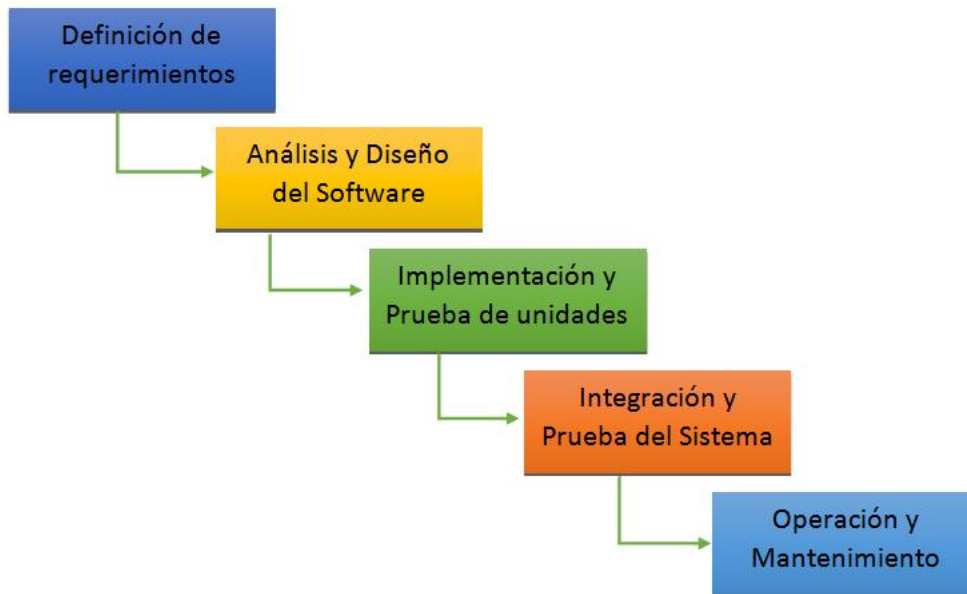
Los principales objetivos de una metodología de desarrollo son:

- Asegurar la uniformidad y calidad tanto del desarrollo como del sistema en sí.
- Satisfacer las necesidades de los usuarios del sistema.
- Conseguir un mayor nivel de rendimiento y eficiencia del personal asignado al desarrollo.
- Ajustarse a los plazos y costes previstos en la planificación.
- Generar de forma adecuada la documentación asociada a los sistemas.
- Facilitar el mantenimiento posterior de los sistemas.

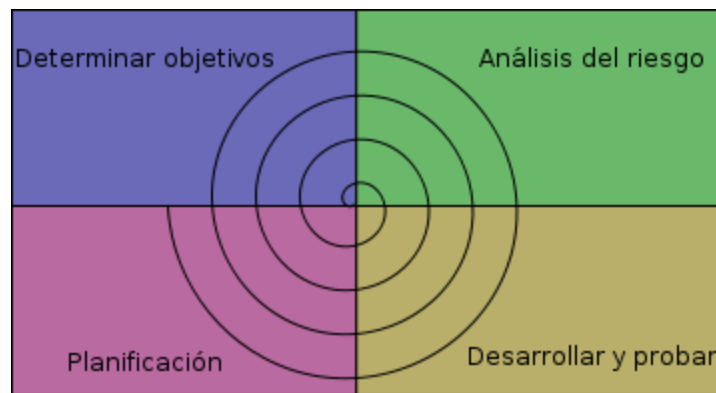
Dentro de las metodologías existentes se encuentran dentro de las más importantes:

**TRADICIONALES:**

- **Cascada**
- **Espiral**
- **Prototipo**



**Fig. 3.3 Representación de la metodología de cascada.**



**Fig. 3.4 Representación de la metodología de espiral.**

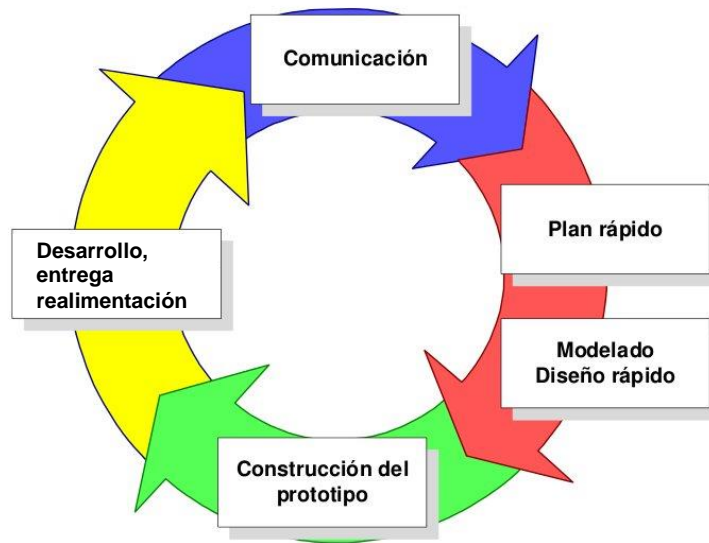


Fig. 3.5 Representación de la metodología de prototipo.

**ÁGILES:**

- Scrum
- XP
- Lean UX

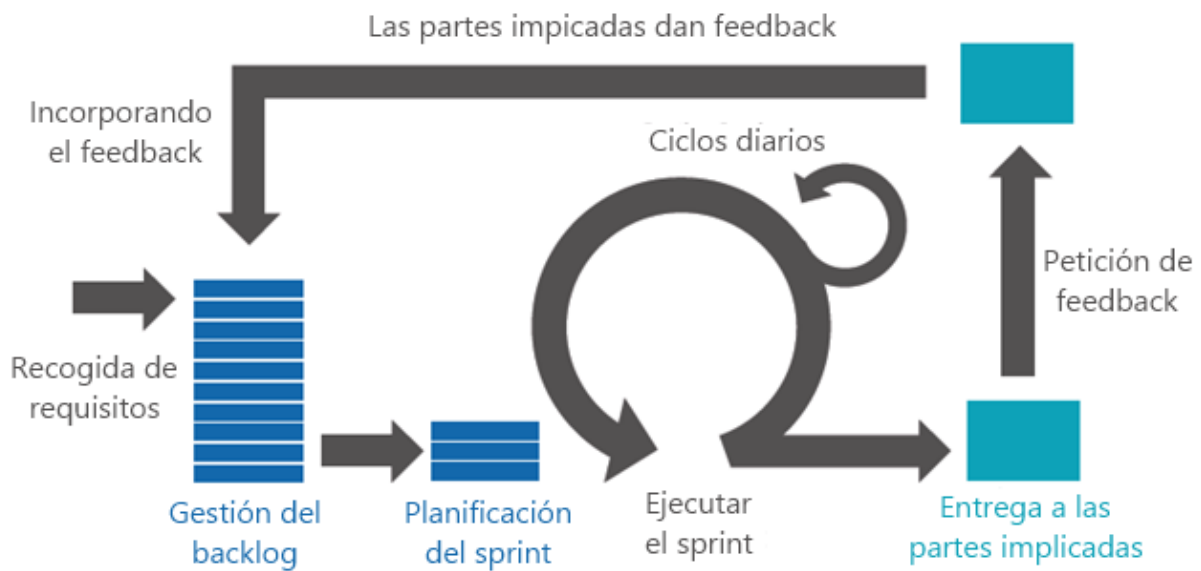


Fig. 3.6 Representación de la metodología de Scrum.

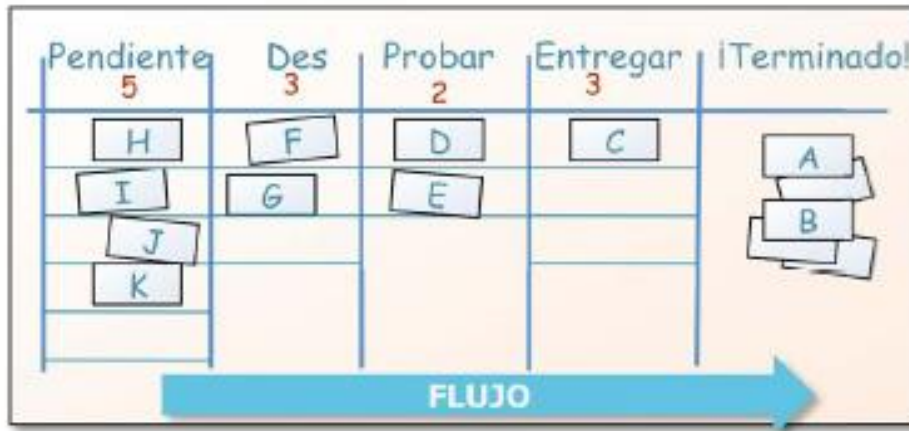


Fig. 3.7 Representación de la tabla de Kanban.

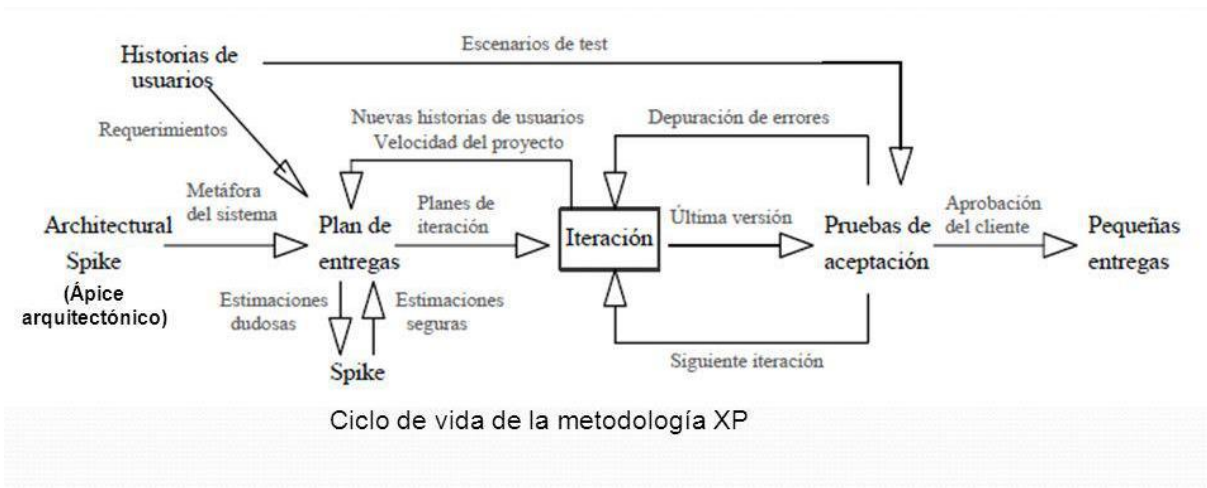


Fig. 3.8 Representación de la metodología de XP.

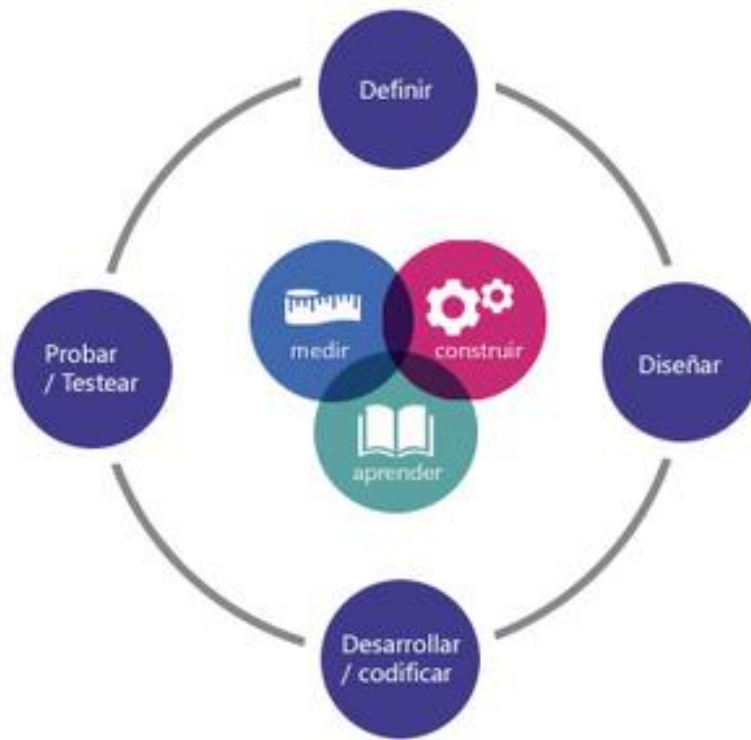


Fig. 3.9 Representación de la metodología de Lean UX.

En la siguiente figura se muestra una pequeña representación comparativa de los dos tipos de metodologías mostradas anteriormente, y como nos damos cuenta la metodología tradicional es una metodología más lineal y casi invariable, lo contrario de las metodologías ágiles las cuales tienen un proceso iterativo con el cliente y van desarrollando el proyecto de la mano realizando cambios a lo largo del trayecto del desarrollo.



**Fig. 3.10 Comparativo de Metodologías.**

El desarrollo de este proyecto, se llevará a cabo utilizando la metodología ágil de **Scrum** la cual nos permitirá realizar un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesitan obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costos se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

### **El proceso**

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de realimentación y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas.

Las actividades que se llevan a cabo en Scrum son las siguientes:

## Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

1. **Selección de requisitos** (4 horas máximo). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente los solicita.
2. **Planificación de la iteración** (4 horas máximo). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto-assignan las tareas.

## Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización (15 min. máximo). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido.

En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el Facilitador (*Scrum Master*) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad. Elimina los obstáculos que el equipo no puede resolver por sí mismo. Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad. Durante la iteración, el cliente junto con el equipo refina la lista de requisitos (para prepararlos



para las siguientes iteraciones) y, si es necesario, cambian o replanifican los objetivos del proyecto para maximizar la utilidad de lo que se desarrolla y el retorno de inversión.

## **Inspección y adaptación**

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

1. **Demostración** (4 horas máximo). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
2. **Retrospectiva** (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

## **3.3 LENGUAJES DE PROGRAMACIÓN**

Para hablar de lenguajes de programación antes me quiero referir a los diferentes paradigmas de programación que existen los cuales nos proporcionan la propuesta tecnológica adoptada por una comunidad de programadores y desarrolladores, cuyo núcleo central es incuestionable en cuanto a que únicamente trata de resolver uno o varios problemas claramente delimitados; la resolución de estos problemas debe suponer consecuentemente un avance significativo en al menos un parámetro que afecte a la ingeniería de software.



Fig. 3.11 Lenguajes de Programación.

Aquí presento una tabla comparativa a modo de resumen, en donde plasmo las principales características de algunos de los paradigmas de programación, así como también una serie de ventajas y desventajas de cada paradigma para entender mejor, la finalidad de cada uno.

<b>Paradigma</b>	<b>¿De qué trata?</b>	<b>Ventajas</b>	<b>Desventajas</b>
Imperativo	Es un paradigma de programación que describe la programación en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea.	<ul style="list-style-type: none"> <li>-Manipulación controlada</li> <li>-Ajusta los problemas</li> <li>-Control de datos de manera secuencial</li> </ul>	<ul style="list-style-type: none"> <li>-Programación lineal</li> <li>-Codificación</li> <li>-Los programas no son seguros</li> <li>-Lenguajes de alto nivel y muy complejo</li> </ul>
Funcional	Este tipo de paradigma demuestra un tipo de programación en forma de funciones matemáticas. Los lenguajes puramente funcionales evitan efectos secundarios que pueden ser comunes en otros tipos de paradigmas.	<ul style="list-style-type: none"> <li>-Brevedad</li> <li>-Programas más fáciles de entender</li> <li>-Menos errores mayor fiabilidad</li> <li>-Reusabilidad</li> </ul>	<ul style="list-style-type: none"> <li>-Modelo alejado del modelo de Von Neumann</li> <li>-Se usan arquitecturas paralelas</li> </ul>

Paradigma	¿De qué trata?	Ventajas	Desventajas
Lógico	Se definen reglas lógicas que luego a través de un motor de inferencias lógicas, resolver problemas planteados por el sistema. Algunos programas que utilizan el paradigma lógico.	<ul style="list-style-type: none"> <li>-Simplicidad</li> <li>-Cercanía a las especificaciones del problema realizada con lenguajes formales</li> <li>-Sencillez, potencia y elegancia</li> <li>-Metodología rigurosa de especificación</li> <li>-Sencillez en la implementación de estructuras complejas</li> </ul>	<ul style="list-style-type: none"> <li>- Poco eficientes</li> <li>-Poco utilizado en aplicaciones reales</li> </ul>

Paradigma	¿De qué trata?	Ventajas	Desventajas
Orientado a Objetos	Este tipo de programación posee como idea principal encapsular estado y operaciones en objetos. Estos objetos pueden comunicarse entre sí, poseen clases y herencia. Se puede destacar como ventaja la facilidad de reutilización de código.	<ul style="list-style-type: none"> <li>-Reusabilidad</li> <li>-Mantenibilidad</li> <li>-Adaptación</li> <li>-Fiabilidad</li> <li>-Fomenta la reutilización y extensión del código.</li> <li>-Permite crear sistemas más complejos.</li> <li>-Relacionar el sistema al mundo real.</li> <li>-Facilita la creación de programas visuales.</li> <li>-Construcción de prototipos</li> <li>-Agiliza el desarrollo de software</li> <li>-Facilita el trabajo en equipo</li> <li>-Facilita el mantenimiento del software</li> </ul>	<ul style="list-style-type: none"> <li>-Cambio en la forma de pensar de la programación tradicional a la orientada a objetos.</li> <li>-La ejecución de programas orientados a objetos es más lenta.</li> <li>-La necesidad de utilizar bibliotecas de clases obliga a su aprendizaje y entrenamiento.</li> <li>-Exige conocer bien la teoría de objetos.</li> <li>-Requiere mayor capacidad de los programadores.</li> </ul>

Paradigma	¿De qué trata?	Ventajas	Desventajas
Estructurado	<p>Básicamente el código se divide en bloques, estructuras, que pueden o no comunicarse entre sí. Este software se controla con secuencia, selección e interacción. Otorga una mejor comprensión del código y permite su reutilización de una forma más fácil.</p>	<ul style="list-style-type: none"> <li>- Tiende a ser mucho más fácil de comprender.</li> <li>- Reducción de los costos de mantenimiento.</li> <li>- Los programas quedan mejor documentados internamente.</li> <li>- Aumento de la productividad del programador</li> <li>- Programas más sencillos y más rápidos</li> <li>- Se facilita la utilización de las otras técnicas para el mejoramiento de la productividad en programación</li> </ul>	<ul style="list-style-type: none"> <li>- Obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar problemático su manejo; esto se resuelve empleando la programación modular, definiendo módulos interdependientes programados y compilados por separado.</li> <li>- Un método un poco más sofisticado es la programación por capas, en la que los módulos tienen una estructura jerárquica en la que se pueden definir funciones dentro de funciones o de procedimientos.</li> </ul>

Paradigma	¿De qué trata?	Ventajas	Desventajas
Computación en la nube	Se entiende a aquella tecnología que permite ofrecer servicios de computación a través de Internet, de manera que todo aquello que puede ofrecer un sistema informático es ofrecido como servicio, al cual pueden a su vez acceder los usuarios sin que tengan conocimientos en la gestión de los diversos recursos que utilizan.	<ul style="list-style-type: none"> <li>- Costos.</li> <li>- Competitividad.</li> <li>- Disponibilidad.</li> <li>- Abstracción de la parte técnica.</li> <li>- Acceso desde cualquier punto geográfico.</li> <li>- Escalabilidad.</li> <li>- Concentración de esfuerzos en los procesos de negocio.</li> </ul>	<ul style="list-style-type: none"> <li>- Privacidad.</li> <li>- Falta de control sobre recursos.</li> <li>- Dependencia.</li> <li>- Integración.</li> </ul>

Paradigma	¿De qué trata?	Ventajas	Desventajas
Declarativo	No se basa en el cómo se hace algo (cómo se logra un objetivo paso a paso), sino que describe (declara) cómo es algo. En otras palabras, se enfoca en describir las propiedades de la solución buscada, dejando indeterminado el algoritmo (conjunto de instrucciones) usado para encontrar esa solución.	<ul style="list-style-type: none"> <li>- Razonados matemáticamente, lo que permite el uso de mecanismos matemáticos para optimizar el rendimiento de los programas.</li> <li>- Son fiables, elegantes y expresivos</li> </ul>	<ul style="list-style-type: none"> <li>- El principal obstáculo del paradigma declarativo es el descubrimiento del algoritmo básico para resolver problemas. Por esta razón, los lenguajes declarativos tienden a ser de propósito específico, diseñados para usarse en aplicaciones particulares.</li> <li>- Otra desventaja de la programación declarativa está relacionada con la eficiencia.</li> </ul>

**Tabla. 3.1** Tabla representativa de los paradigmas de programación.



## Lenguaje de programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo.

Por otro lado, el término "lenguaje natural" define un medio de comunicación compartido por un grupo de personas (por ejemplo: inglés o francés).

Los lenguajes que los equipos usan para comunicarse entre ellos no tienen nada que ver con los lenguajes de programación; se los conoce como protocolos de comunicación. Se trata de dos conceptos totalmente diferentes. Un lenguaje de programación es muy estricto:

**“A CADA instrucción le corresponde UNA acción de procesador.”**

El lenguaje utilizado por el procesador se denomina lenguaje máquina. Se trata de datos tal como llegan al procesador, que consisten en una serie de 0 y 1 (datos binarios).

El lenguaje máquina, por lo tanto, no es comprensible para los seres humanos, razón por la cual se han desarrollado lenguajes intermediarios comprensibles para el hombre. El código escrito en este tipo de lenguaje se transforma en código máquina para que el procesador pueda procesarlo.

A continuación, se muestra en la Tabla 3.2 una muestra de varios lenguajes de programación orientados a **Plataforma Web**, que actualmente son los más u en la programación actualmente. Cabe destacar que en esta tabla no se toman en cuenta otros lenguajes de programación porque no entran dentro del objetivo de la tesis ni dentro de los parámetros de la investigación.

Lenguaje	Ventajas	Desventajas	Características
<b>HTML (maquetado)</b>	<p>Sencillo que permite describir hipertexto.</p> <p>Texto presentado de forma estructurada y agradable.</p> <p>No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.</p> <p>Archivos pequeños.</p> <p>Despliegue rápido.</p> <p>Lenguaje de fácil aprendizaje.</p> <p>Lo admiten todos los exploradores.</p>	<p>Lenguaje estático.</p> <p>La interpretación de cada navegador puede ser diferente.</p> <p>Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.</p> <p>El diseño es más lento.</p> <p>Las etiquetas son muy limitadas.</p>	<p>Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html).</p>

<p><b>JavaScript</b></p>	<p>Lenguaje de scripting seguro y fiable.</p> <p>Los scripts tienen capacidades limitadas, por razones de seguridad.</p> <p>El código Javascript se ejecuta en el cliente.</p>	<p>Código visible por cualquier usuario.</p> <p>El código debe descargarse completamente.</p> <p>Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo <u>CSS</u>).</p>	<p>Interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.</p>
--------------------------	--	--	--

<p><b>PHP</b></p>	<p>Muy fácil de aprender.          Se caracteriza por ser un lenguaje muy rápido.          Soporta en cierta medida la orientación a objeto. Clases y herencia.          Es un lenguaje multiplataforma: Linux, Windows, entre otros.          Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.          Capacidad de expandir su potencial utilizando módulos.          Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.          Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.          Incluye gran cantidad de funciones.          No requiere definición de tipos de variables ni manejo detallado del bajo nivel.</p>	<p>Se necesita instalar un servidor web.          Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto, puede ser más ineficiente a medida que las solicitudes aumenten de número.          La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.          La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.          Dificulta la modularización.          Dificulta la organización por capas de la aplicación.</p>	<p>PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).</p>
-------------------	--	---	---

<b>ASP.NET</b>	<p>Completamente orientado a objetos.</p> <p>Controles de usuario y personalizados.</p> <p>División entre la capa de aplicación o diseño y el código.</p> <p>Facilita el mantenimiento de grandes aplicaciones.</p> <p>Incremento de velocidad de respuesta del servidor.</p> <p>Mayor velocidad.</p> <p>Mayor seguridad.</p>	<p>Mayor consumo de recursos.</p>	<p>Este es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .NET.</p> <p>El ASP.NET fue desarrollado para resolver las limitantes que brindaba tu antecesor ASP. Creado para desarrollar web sencillas o grandes aplicaciones.</p>
----------------	---	-----------------------------------	--

<b>JSP</b>	<p>Ejecución rápida del servlets.</p> <p>Crear páginas del lado del servidor.</p> <p>Multiplataforma. Código bien estructurado. Integridad con los módulos de Java.</p> <p>La parte dinámica está escrita en Java.</p> <p>Permite la utilización se servlets.</p>	Complejidad de aprendizaje.	<p>Código separado de la lógica del programa.</p> <p>Las páginas son compiladas en la primera petición.</p> <p>Permite separar la parte dinámica de la estática en las páginas web.</p> <p>Los archivos se encuentran con la extensión (jsp).</p> <p>El código JSP puede ser incrustado en código HTML.</p>
------------	---	-----------------------------	---

<b>Python</b>	<p>Libre y fuente abierta.</p> <p>Lenguaje de propósito general.</p> <p>Gran cantidad de funciones y librerías.</p> <p>Sencillo y rápido de programar.</p> <p>Multiplataforma.</p> <p>Licencia de código abierto (Opensource).</p> <p>Orientado a Objetos. Portable.</p>	<p>Lentitud por ser un lenguaje interpretado.</p>	<p>Programación orientada a objetos.</p> <p>Programación estructurada.</p> <p>Programación funcional.</p> <p>Programación orientada a aspectos.</p>
---------------	--	---	---

<b>Ruby</b>	<p>Permite desarrollar soluciones a bajo Costo.</p> <p>Software libre.</p> <p>Multiplataforma.</p>		<p>Existe diferencia entre mayúsculas y minúsculas.</p> <p>Múltiples expresiones por líneas, separadas por punto y coma “;”.</p> <p>Dispone de manejo de excepciones.</p> <p>Ruby puede cargar librerías de extensiones dinámicamente si el (Sistema Operativo) lo permite.</p> <p>Portátil.</p>
-------------	--	--	--

**Tabla. 3.2** Tabla representativa de algunos de los lenguajes de programación.



## **3.4 BASES DE DATOS**

### **Definición**

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

### **Características**

- Entre las principales características de los sistemas de base de datos podemos mencionar:
- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

## Tipos

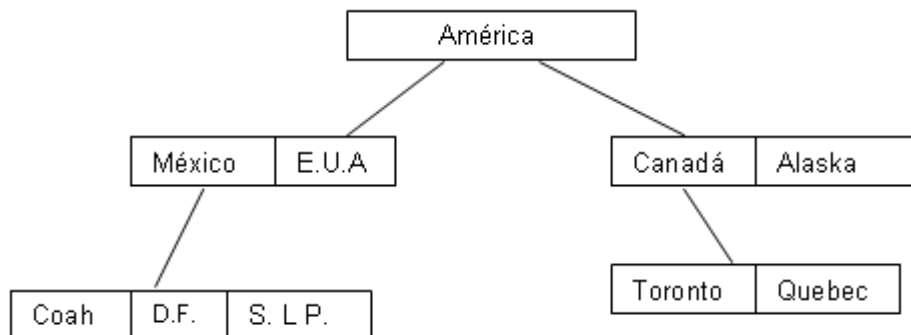
- Bases de datos jerárquicas.
- Bases de datos en red.
- Bases de datos relacionales.

### Las bases de datos jerárquicas

En una base de datos jerárquica se organizan los datos utilizando estructuras arborescentes (en árbol).

Un ÁRBOL es una estructura jerárquica en la que los elementos se suelen denominar NODOS y existen dependencias entre los nodos.

La dependencia es de 1:M del tipo padre/hijo. Un hijo no puede tener más de un padre, pero un padre varios hijos.



**Fig. 3.12 Representación de la base de datos jerárquica.**

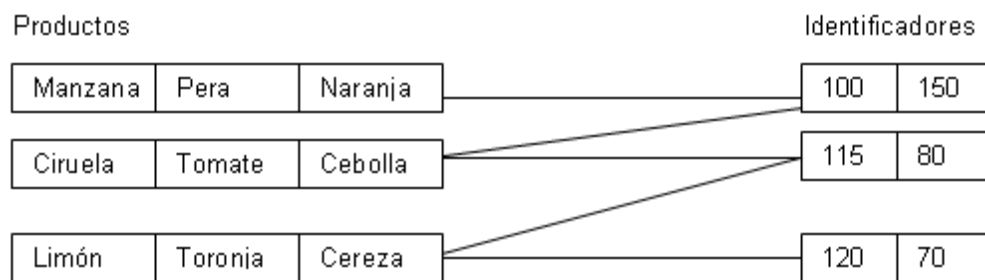
### Las bases de datos en red

También en desuso, en una base de datos en red se utiliza la estructura de grafo/red, como en el caso anterior los distintos objetos están relacionados entre sí

mediante relaciones del tipo 1:M pero en este caso un objeto puede estar relacionado como hijo con varios elementos que serán sus padres. Las relaciones que se crean se denominan SET y el equivalente al padre se denomina PROPIETARIO (OWNER) y el equivalente al hijo se denomina MIEMBRO (MEMBER).

Un ejemplo de sistema en red es el CODASYL. También existen modelos para realizar el diseño de datos orientado a bases de datos en red.

En red podríamos representar lo mismo que la estructura anterior y además lo siguiente:



**Fig. 3.13 Representación de la base de datos en red.**

Los sistemas jerárquicos y en red constituyen la primera generación de los SGBD. Pero estos sistemas presentan algunos inconvenientes:

- Es necesario escribir complejos programas de aplicación para responder a cualquier tipo de consulta de datos, por simple que ésta sea.
- La independencia de datos es mínima.
- No incluyen controles de integridad.
- Por lo que pronto fueron sustituidos por los sistemas relacionales.

## **Las bases de datos relacionales**

Esta es la estructura que se ha impuesto para aplicaciones de gestión, consiste en organizar los datos en forma de tablas, las relaciones entre los objetos se consiguen incluyendo en la tabla del hijo, la clave del objeto padre. Como son las que utilizaremos durante todo el módulo hemos reservado un apartado especial para ellas.

## **Las bases de datos orientadas a objetos**

Es un modelo más reciente, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). La información que contienen se organiza en atributos y el comportamiento en operaciones.

## **Las bases de datos multidimensionales**

En una base de datos multidimensional los datos se almacenan en tablas de múltiples dimensiones en vez de tablas bidimensionales como las del modelo relacional. Se utilizan para grandes volúmenes de información.

## **Manejadores**

Los Sistemas de Gestión de Base de Datos (en inglés *DataBase Management System*) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

## Tipos de dato

En MySQL existen bastantes tipos diferentes disponibles, de modo que será mejor que los agrupemos por categorías: de caracteres, enteros, de coma flotante, tiempos, bloques, enumerados y conjuntos.

Tipo	Descripción
<code>varchar (n)</code>	Puede contener cualquier carácter. Longitud variable con un tamaño máximo de $n$ bytes, siendo el límite para $n$ de 65536
<code>char (n)</code>	Puede contener cualquier carácter, pero con longitud fija de $n$ bytes. Los espacios no ocupados se rellenan con blancos. Valor máximo de $n$ es 255
<code>date</code>	Una fecha
<code>datetime</code>	Una fecha y una hora
<code>time</code>	Una hora
<code>year (n)</code>	Un año con $n$ dígitos (2 o 4)
<code>bit (n)</code>	Un número de $n$ bits. Sólo en versiones de MySQL superiores a la 5
<code>bool</code>	Lógico: 0 falso, 1 verdadero
<code>tinyint (n)</code>	Un entero con un máximo de $n$ cifras. Valores entre -128 y 127
<code>smallint (n)</code>	Un entero con un máximo de $n$ cifras. Valores entre -32768 y 32767
<code>mediumint (n)</code>	Un entero con un máximo de $n$ cifras. Valores entre -8388608 y 8388607
<code>int (n)</code>	Un entero con un máximo de $n$ cifras. Valores entre -2147483648 y 2147483647
<code>bigint (n)</code>	Un entero con un máximo de $n$ cifras. Para valores no incluidos en los intervalos anteriores
<code>float (n, d)</code>	Un número real con $n$ cifras de las cuales $d$ son decimales. Valores entre -3.402823466E+38 y 3.402823466E+38
<code>double (n, d)</code>	Un número real con $n$ cifras de las que $d$ son decimales. Para valores con mayor precisión que los anteriores, o no incluidos en el intervalo
<code>enum ('a', 'b', 'c'...)</code>	Puede incluir cualquiera de los elementos enumerados en la lista

Tabla 3.3 Tipos de datos SQL.

## Fundamentos de la normalización<sup>1</sup>

La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes.

Los datos redundantes desperdician el espacio de disco y crean problemas de mantenimiento. Si hay que cambiar datos que existen en más de un lugar, se deben cambiar de la misma forma exactamente en todas sus ubicaciones. Un cambio en la dirección de un cliente es mucho más fácil de implementar si los datos sólo se almacenan en la tabla Clientes y no en algún otro lugar de la base de datos.

¿Qué es una "dependencia incoherente"? Aunque es intuitivo para un usuario mirar en la tabla Clientes para buscar la dirección de un cliente en particular, puede no tener sentido mirar allí el salario del empleado que llama a ese cliente. El salario del empleado está relacionado con el empleado, o depende de él, y por lo tanto se debería pasar a la tabla Empleados. Las dependencias incoherentes pueden dificultar el acceso porque la ruta para encontrar los datos puede no estar o estar interrumpida.

Hay algunas reglas en la normalización de una base de datos. Cada regla se denomina una "forma normal". Si se cumple la primera regla, se dice que la base de datos está en la "primera forma normal". Si se cumplen las tres primeras reglas, la base de datos se considera que está en la "tercera forma normal". Aunque son posibles otros niveles de normalización, la tercera forma normal se considera el máximo nivel necesario para la mayor parte de las aplicaciones.

Al igual que con otras muchas reglas y especificaciones formales, en los escenarios reales no siempre se cumplen los estándares de forma perfecta. En

general, la normalización requiere tablas adicionales y algunos clientes consideran éste un trabajo considerable. Si decide infringir una de las tres primeras reglas de la normalización, asegúrese de que su aplicación se anticipa a los problemas que puedan aparecer, como la existencia de datos redundantes y de dependencias incoherentes.

<sup>1</sup>Referencia electrónica: <https://support.microsoft.com/es-mx/help/283878/description-of-the-database-normalization-basics>

Forma normal	Características
Primera forma normal (1NF)	<ul style="list-style-type: none"> <li>• Elimina los grupos repetidos de las tablas individuales.</li> <li>• Crea una tabla independiente para cada conjunto de datos relacionados.</li> <li>• Identifica cada conjunto de datos relacionados con una clave principal.</li> <li>• No usar varios campos en una sola tabla para almacenar datos similares. Por ejemplo, para realizar el seguimiento de un elemento del inventario que proviene de dos orígenes posibles, un registro del inventario puede contener campos para el Código de usuario 1 y para el Código de usuario 2.</li> </ul>
Segunda forma normal (2NF)	<ul style="list-style-type: none"> <li>• Crea tablas independientes para conjuntos de valores que se apliquen a varios registros.</li> <li>• Relaciona estas tablas con una clave externa.</li> <li>• Los registros no deben depender de nada que no sea una clave principal de una tabla, una clave compuesta si es necesario. Por ejemplo, considere la dirección de un cliente en un sistema de contabilidad. La dirección se necesita en la tabla Clientes, pero también en las tablas Pedidos, Envíos, Facturas, Cuentas por cobrar y Colecciones. En lugar de almacenar la dirección de un cliente como una entrada independiente en cada una de estas tablas, almacénela en un lugar, ya sea en la tabla Clientes o en una tabla Direcciones independiente.</li> </ul>



Forma normal	Características
Tercera forma normal (3NF)	<ul style="list-style-type: none"> <li>• Elimina los campos que no dependan de la clave.</li> <li>• Los valores de un registro que no sean parte de la clave de ese registro no pertenecen a la tabla. En general, siempre que el contenido de un grupo de campos pueda aplicarse a más de un único registro de la tabla, considere colocar estos campos en una tabla independiente. Por ejemplo, en una tabla Contratación de empleados, puede incluirse el nombre de la universidad y la dirección de un candidato. Pero necesita una lista completa de universidades para enviar mensajes de correo electrónico en grupo. Si la información de las universidades se almacena en la tabla Candidatos, no hay forma de enumerar las universidades que no tengan candidatos en ese momento. Cree una tabla Universidades independiente y vincúlela a la tabla Candidatos con el código de universidad como clave.</li> </ul>
Forma normal de Boyce-Codd (BCNF)	<ul style="list-style-type: none"> <li>• Es una forma normal utilizada en la normalización de bases de datos. Es una versión ligeramente más fuerte de la Tercera forma normal (3FN).</li> <li>• Requiere que no existan dependencias funcionales no triviales de los atributos que no sean un conjunto de la clave candidata.</li> <li>• Si no existen claves candidatas compuestas (con varios atributos), está en FNBC.</li> <li>• Si existen varias claves candidatas compuestas y éstas tienen un elemento</li> </ul>

Forma normal	Características
	común, puede no estar en FNBC. Sólo si, para cada dependencia funcional en la relación, el determinante es una clave candidata, estará en FNBC.
Cuarta forma normal (4NF)	<ul style="list-style-type: none"> <li>• La 4NF se asegura de que las dependencias multivaluadas independientes estén correctas y eficientemente representadas en un diseño de base de datos.</li> <li>• Una tabla está en 4NF si y solo si está en Tercera forma normal o en BCNF (Cualquiera de ambas) y no posee dependencias multivaluadas no triviales. La definición de la 4NF confía en la noción de una dependencia multivaluada.</li> <li>• Una tabla con una dependencia multivaluada es una donde la existencia de dos o más relaciones independientes muchos a muchos causa redundancia; y es esta redundancia la que es suprimida por la cuarta forma normal.</li> </ul>
Quinta forma normal (5FN)	<ul style="list-style-type: none"> <li>• También conocida como forma normal de proyección-uniión (PJ/NF).</li> <li>• Solamente en contadas ocasiones una tabla 4NF no se corresponde con una 5NF.</li> <li>• Estas son situaciones en las cuales una restricción compleja del mundo real, que limita las combinaciones válidas de los valores de atributos en la tabla 4NF, no está implícita en la estructura de esa tabla.</li> <li>• Si esa tabla no se normaliza a 5NF, la tarea de mantener la consistencia</li> </ul>

Forma normal	Características
	<p>lógica de los datos dentro de la tabla debe ser llevada en parte por la aplicación responsable de inserciones, borrados, y actualizaciones a ella; y hay un riesgo elevado de que los datos dentro de la tabla se vuelvan inconsistentes.</p> <ul style="list-style-type: none"> <li>• Por el contrario, el diseño 5NF excluye la posibilidad de tales inconsistencias.</li> </ul>
Forma normal de dominio/clave (DKNF)	<ul style="list-style-type: none"> <li>• Una restricción del dominio especifica los valores permitidos para un atributo dado, mientras que una restricción clave especifica los atributos que identifican únicamente una fila en una tabla dada.</li> <li>• Está es el santo grial de la Base de datos y es alcanzado cuando cada restricción en la relación es una consecuencia lógica de la definición de claves y dominios, y, haciendo cumplir las restricciones y condiciones de la clave y del dominio, causa que sean satisfechas todas las restricciones.</li> <li>• Así, esto evita todas las anomalías no-temporales.</li> </ul>

**Tabla 3.4 Formas normales.**

### **3.5 VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE**

Los certificados de seguridad son una medida de confianza adicional para las personas que visitan y hacen transacciones en su página web, le permite cifrar los datos entre la computadora del cliente y el servidor que representa a la página. El significado más preciso de un certificado de seguridad es que con él logramos que los datos personales sean encriptados y así imposibilitar que sean interceptados por otro usuario. Ahora es muy común ver en nuestros exploradores el protocolo de seguridad https; mediante éste, básicamente nos dice que la información que se envía a través de internet, entre el navegador del cliente y el servidor donde está alojada la página, se encripta de forma que es casi imposible que otra persona reciba, vea o modifique los datos confidenciales del cliente. Las ventajas de este sistema las podemos ver fácilmente, ya que sí es seguro, podemos medir la confianza de nuestro sitio en cuanto al volumen de ventas en línea. Para los clientes es fundamental realizar compras de manera segura y así identificar que la información llegará al servidor correcto.

Los certificados de seguridad brindan confianza en línea, al obtener un certificado, su cliente podrá conocer la información sobre su empresa. Al ofrecer seguridad, aumentará el número de clientes y usuarios, realizando más compras en su sitio web y así tener una experiencia en internet más rentable.

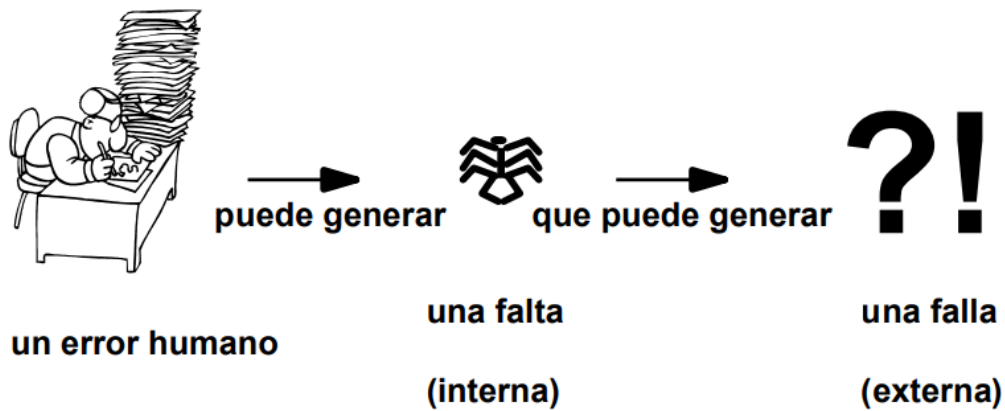


Fig. 3.14 Errores, faltas y fallas

### Objetivos

- Descubrir defectos (para corregirlos)
  - Provocar fallas (una forma de detectar defectos)
  - Revisar los productos (otra forma de detectar defectos)
- Evaluar la calidad de los productos
  - El probar o revisar el software da una idea de la calidad del mismo

### Definición

- **Sommerville**
  - Verificación
    - Busca comprobar que el sistema cumple con los requerimientos especificados (funcionales y no funcionales)
    - ¿El software está de acuerdo con su especificación?
  - Validación
    - Busca comprobar que el software hace lo que el usuario espera.
    - ¿El software cumple las expectativas del cliente?

- **Boehm**
  - Verificación
    - ¿Estamos construyendo el producto correctamente?
  - Validación
    - ¿Estamos construyendo el producto correcto?
- **Ghezzi**
  - Verificación
    - Todas las actividades que son llevadas a cabo para averiguar si el software cumple con sus objetivos
- **IEEE**
  - Verificación
    - El proceso de evaluar un sistema o componente para determinar si los productos de una determinada fase de desarrollo satisfacen las condiciones impuestas al inicio de la fase.
  - Validación
    - El proceso de evaluación de un sistema o componente durante o al final del proceso de desarrollo para determinar si cumple con los requisitos especificados

## 3.6 PRUEBAS A REALIZAR

### Prueba de estrés<sup>2</sup>

Uno de los análisis que se suelen es la prueba de stress. Esta evaluación pone a prueba la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremas. Entre estas condiciones se incluyen el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas.

El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan bugs (defectos) potencialmente peligrosos.

Cuando hablamos de aplicaciones web, una posible condición extrema puede ser el acceso de un enorme número de usuarios en poco tiempo. Efectos similares pueden obtenerse con un ataque de denegación de servicio (DDoS) a través de un software malicioso. Los efectos de la saturación pueden ser la pérdida o adulteración de datos, el uso excesivo de recursos incluso una vez finalizada la situación de stress, un mal funcionamiento de componentes de la aplicación o la aparición de errores inesperados.

Un buen plan de pruebas de stress debe contemplar el desarrollo de no uno, sino varios casos de stress. Cada caso diferirá en el volumen del estímulo a aplicar sobre la aplicación (cantidad de usuarios, cantidad de peticiones, etc.), el tiempo que durará cada estímulo y la duración total del experimento, entre otras variables. Además, deberá contar con una serie de resultados esperados. Todos los casos deben ponerse en práctica, registrándose al término de cada uno estadísticas sobre el uso de CPU, memoria, conexión y otros recursos. Al finalizar, se comparan los resultados obtenidos con los esperados y se obtienen conclusiones sobre el rendimiento de la aplicación. Si se encontraron problemas, es necesario revisar el diseño o el código de la aplicación para descubrir el origen del conflicto.

### **Prueba de carga<sup>3</sup>**

Hay diversos software en el mercado que están orientados a este tipo de simulaciones, todos los cuales ofrecen características similares. Entre los datos más relevantes que es posible obtener se cuenta:

- Tiempo de acceso de los usuarios a los datos
- Volumen de datos y ancho de banda utilizado
- Archivos solicitados y tiempos usados en transferencia de datos
- Tiempo de espera de los usuarios tras hacer un clic
- Tiempo de respuesta a clicks de usuarios
- Niveles de error existentes tras clicks de usuarios

Como se puede apreciar del listado anterior, los reportes que se obtienen a través de esta vía se refieren a tiempos de acceso que tienen los usuarios que acceden al Sitio Web y la degradación que ocurre en los servicios cuando aumenta el volumen de visitantes concurrentes.

Un ejemplo de las pruebas que se pueden realizar en este tema se puede ver en este gráfico que muestra los tiempos que demora en atender los requerimientos por las direcciones solicitadas tras un click de usuarios.

Cada una de las líneas representa un valor importante de tener en cuenta:

#### **Click time:**

demora del sitio en entregar los datos tras el primer click.

#### **Time to First Byte:**

tiempo que se demora tras el click, en enviar el primer byte de datos.

#### **Time to Connect:**

tiempo de demora tras enviar el click, en establecer la conexión entre servidor y cliente.

**Time for DNS:**

tiempo de demora para resolver la dirección solicitada en el click.

Con los resultados obtenidos con pruebas de este tipo se debe hacer una revisión acuciosa de los sistemas, con el fin de hacer las optimizaciones que aparezcan como necesarias. Asimismo, se debe tener en cuenta que será normal la existencia de situaciones excepcionales que harán que los servicios no funcionen adecuadamente.

**Prueba de validación de campos**

Esta prueba se realizará de forma independiente a las pruebas anteriores mencionadas, la cual consiste en verificar que los campos de los formularios de todo el sistema tengan un mínimo de validación por campos que puede incluir, solo números, solo letras, emails, contraseñas. Así como también se debería verificar la validación de los campos obligatorios y los opcionales dándole seguridad al usuario de que no tendrá que regresar en el proceso varias veces a no ser que introduzca la información de forma incorrecta.

<sup>2</sup>Referencia electrónica: <http://www.4rsoluciones.com/blog/pruebas-de-stress-sobre-aplicaciones-web-2/>

<sup>3</sup>Referencia electrónica: <http://www.guiadigital.gob.cl/articulo/pruebas-de-carga>



# CAPÍTULO 4 ANÁLISIS Y DISEÑO DEL SISTEMA

## 4.1 JUSTIFICACIÓN DE LAS HERRAMIENTAS

### ¿Por qué usar la metodología SCRUM?

Porque es un enfoque de gestión ágil que facilita la administración de proyectos, programas y portafolios de cualquier tamaño y complejidad, facilitando el flujo de información, la comunicación entre el equipo de trabajo y la entrega de valor con oportunidad a los interesados de la organización.

Como ya se había mencionado algunas de las principales ventajas de utilizar SCRUM son:

- **Desarrollo ágil**
- **Fácil escalabilidad**
- **Producto de alta calidad**

Una de las ventajas que más importa en este caso es que proporciona un producto de alta calidad gracias al hecho de que está enfocado completamente en el problema del usuario y a que prioriza los módulos que agregan mayor valor para la organización. No obstante, el equipo de desarrollo cuenta con constante realimentación sobre las funcionalidades esperadas y los posibles cambios estratégicos. Además, los avances del equipo de desarrollo están siempre visibles para el resto de los involucrados en el proyecto. Casos de éxito que ejemplifican el uso de esta metodología se encuentran Spotify, Google, Adobe, etc.

### ¿Por qué usar el lenguaje de programación PHP y javascript con biblioteca jQuery?

PHP es un lenguaje totalmente libre y abierto, del que tengo haciendo uso y tengo dominio desde hace 6 años. Es fácil desplegar un proyecto en PHP porque

dispones de paquetes totalmente autoinstalables que integran PHP: Apache y MySQL, tanto para UNIX (LAMP), como para Windows (WAMP). Ambos fáciles de instalar y configurar, como si de una aplicación cualquiera de escritorio se tratara. Obtienes así en tu computadora un servidor local con acceso a base de datos, listo para que puedas desarrollar fácilmente cualquier proyecto que te propongas, sin perder el tiempo en problemas de compatibilidades o instalando *add ons* que puedas necesitar. Además, PHP ofrece un fácil acceso a bases de datos y soporta muchas de ellas, no únicamente MySQL.

Además, al hacer uso de Javascript podemos realizar las consultas mediante el uso de “*Ajax request*” y mediante este método podemos realizar las consultas de la base de datos y realizar el envío de formularios sin necesidad de que se ejecute todo directamente desde el archivo PHP (lado del servidor) y realiza desde el navegador (lado del cliente), lo cual facilita que el proceso sea más continuo y menos difícil.

### **¿Por qué usar MYSQL como manejador de base de datos?**

MySQL no es el único gestor del mercado, pero es el que se asocia por defecto a otros proyectos open source como *Moodle, Chamilo, Joomla, Wordpress* o *Drupal*.

Una vez dicho esto, comencemos analizando rápidamente qué son los gestores de bases de datos *open source*:

Para todos aquellos que no están acostumbrados a usar esta terminología los Sistemas Gestores de Base de Datos (SGBD) son las aplicaciones que nos permiten tener un espacio (Base de Datos) donde poder guardar la información de nuestras aplicaciones de una forma estructurada para posteriormente acceder a ella desde la misma u otra aplicación.

Centrándonos en el universo Web, existen numerosos gestores de BD que son capaces de soportar cualquier desarrollo web, pero los más usados son: MySQL y PostgreSQL.

En mi caso, llevo trabajando más de 5 años con webs y otras aplicaciones soportadas por MySQL y mi experiencia tanto educativa como profesional ha sido realmente grata, no solo por la documentación oficial sino por la estabilidad, popularidad y seguridad que nos aporta este gestor que además es Open Source (de código abierto).

Más de 3000 ISVs y OEMs, incluyendo 8 de los 10 mayores, y 17 de los 20 principales usuarios de software de todo el mundo confían en MySQL como base de datos de sus productos. En este documento se revisan las diez razones técnicas de por qué MySQL es capaz de satisfacer las necesidades de muchos usuarios líderes, así como miles de nuevas empresas, para hacer funcionar sus aplicaciones, que van desde telecomunicaciones y la seguridad, a Contabilidad de pequeñas empresas y Educación. El documento analiza muchas de las nuevas características clave de MySQL, e incluye numerosos ejemplos de clientes ISV y OEM.

### **¿Por qué usar un framework como Bootstrap?<sup>1</sup>**

Bootstrap es un framework muy útil en el desarrollo de sitios web. Una de sus principales características, que de seguro te animarán a usarlo, es la rapidez con la que creas un sitio web: en menos de un día te puedes asegurar con más del 50% del desarrollo del mismo.

Es open source, de fácil acceso para disponer de su código y empezar a trabajar en tu sitio web; con Bootstrap puedes usar y personalizar de manera sencilla todos sus elementos (como las Barras de Navegación, Formularios, Tablas, Botones, Glyphicons, etc).

Las interfaces que llegas a crear con Bootstrap son de gran usabilidad, y lo que es mejor: dispones de un sistema grid de 12 columnas que trae por defecto para distribuir muy bien todos los elementos que desees colocar en tu sitio web, además, te interesará saber que usa “responsive web design”, todos los elementos que trae Bootstrap 3 se adaptan perfectamente a tus diseños sobre cualquier dispositivo móvil, lo que es muy importante, ya que la mayoría de los sitios que usan responsive web design generan conversión.

<sup>1</sup> Referencia electrónica: <https://www.drauta.com/que-es-bootstrap-principios-basicos>

## 4.2 ANÁLISIS Y DISEÑO DEL SISTEMA

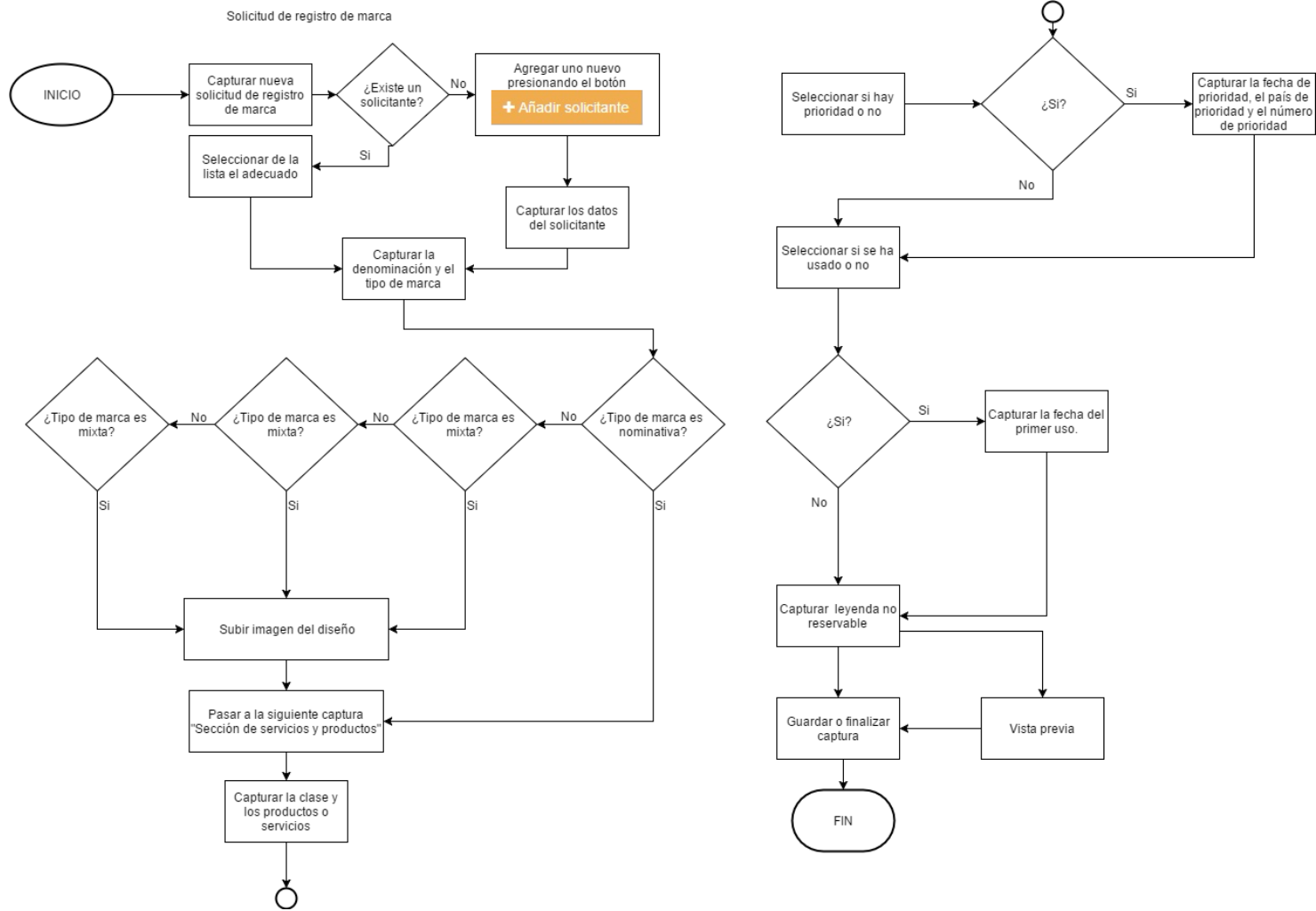
### Análisis de los Datos

#### Datos de Solicitante

- Nombre
- Dirección
- País
- Nacionalidad

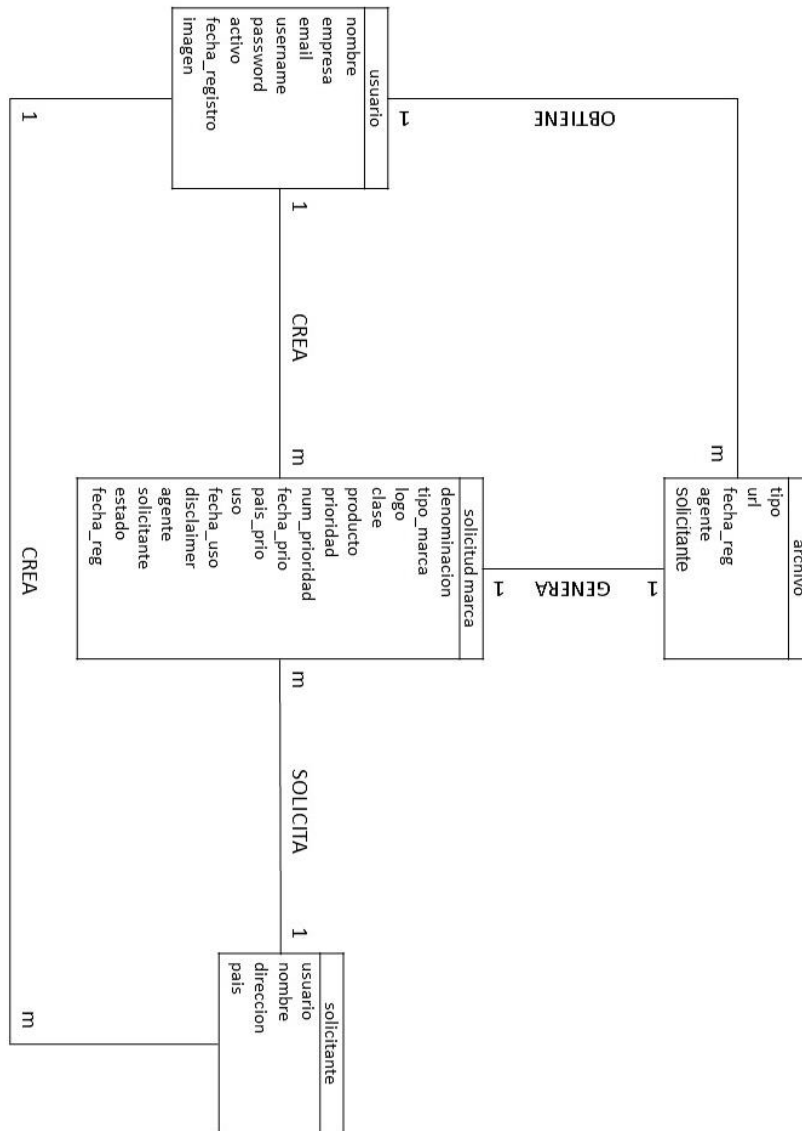
#### Registro de solicitud de marca

- Denominación
- Tipo
- Logo (dependiendo del tipo)
- Productos / Servicios
- Prioridad
- Uso



**Fig. 4.1 Diagrama de flujo de captura de solicitud de registro de marca.**

## Diagrama conceptual



**Fig. 4.2 Diagrama conceptual de la base de datos del prototipo de “ABLEIP”**

Como vemos este diagrama conceptual (Fig. 4.2) nos muestra la organización en un primer inicio de las tablas que se van a usar para representar nuestra base de datos, este diagrama nos establece la base de nuestra base de datos y de una manera muy básica organiza y representa como es que se va a distribuir la información en la base de datos, queda aclarar que las otras relaciones con las diferentes tablas restantes no se ilustran debido a peticiones de la compañía propietaria del sistema **ABLEIP**.

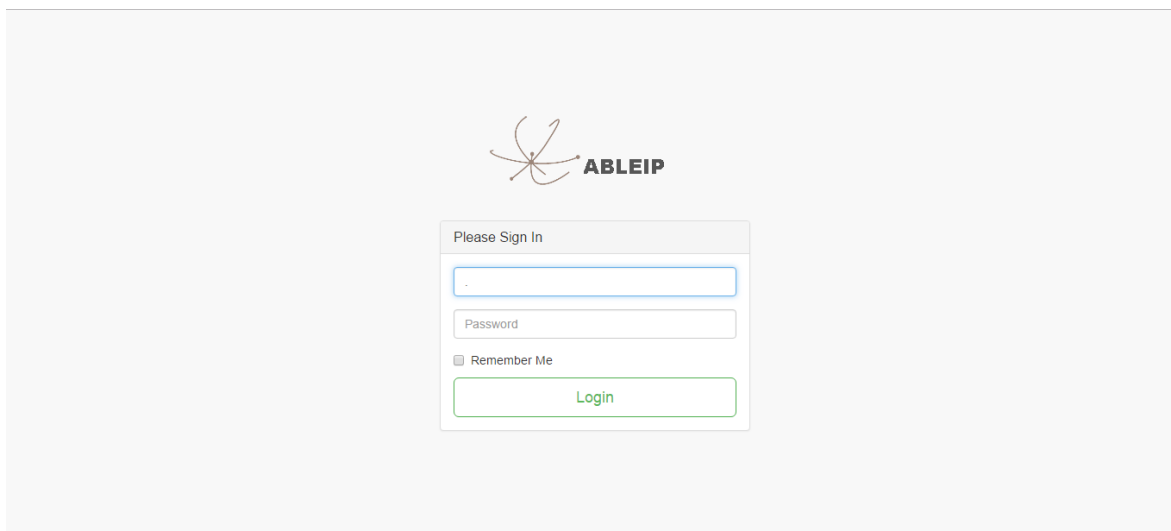
## Diccionario de Datos

Nombre	Tamaño	Tipo	Descripción
id	20	entero	Llave primaria que identifica el registro
nombre	250	carácter	Nombre del usuario.
empresa	250	carácter	Empresa del usuario.
email	100	carácter	Correo electrónico del cliente.
username	150	carácter	Nombre de usuario del usuario registrado.
password	150	carácter	Contraseña encriptada del usuario registrado.
activo	1	numérico (entero)	Identificador que representa la bandera de si el usuario registrado verificó su cuenta de correo electrónico.
dt_registro	50	carácter	Fecha de registro del usuario.
Img	250	carácter	Dirección de url donde se encuentra almacenada la imagen de perfil del usuario en el almacenamiento de ftp del servidor.

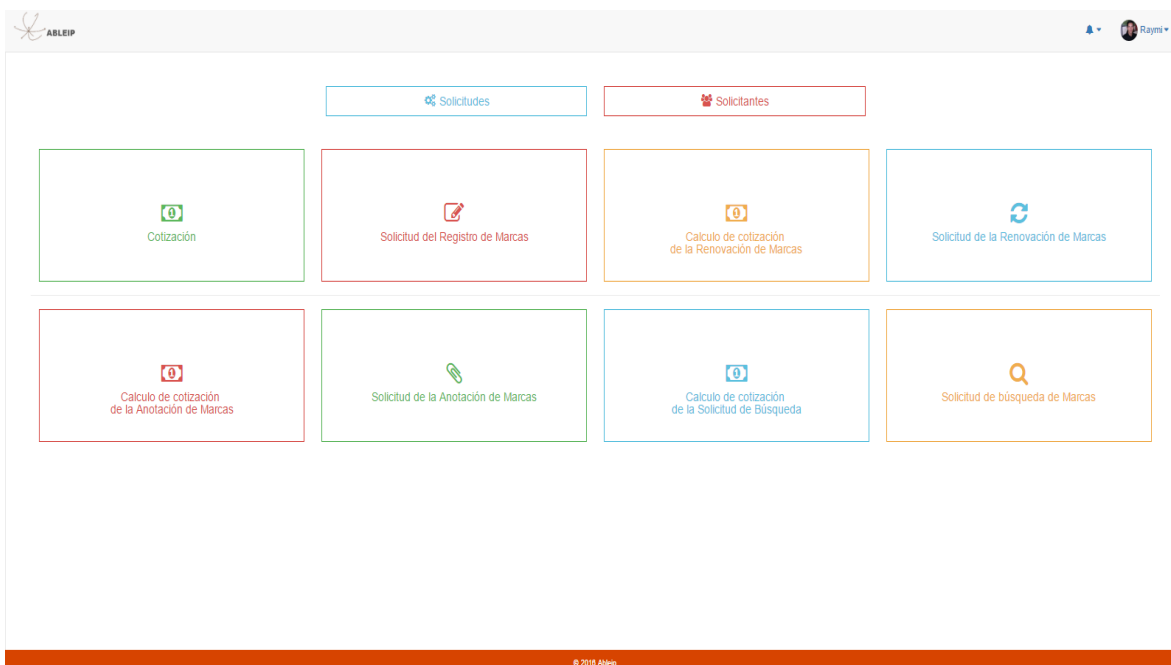
**Tabla 4.1 Tabla usuario en la base de datos**



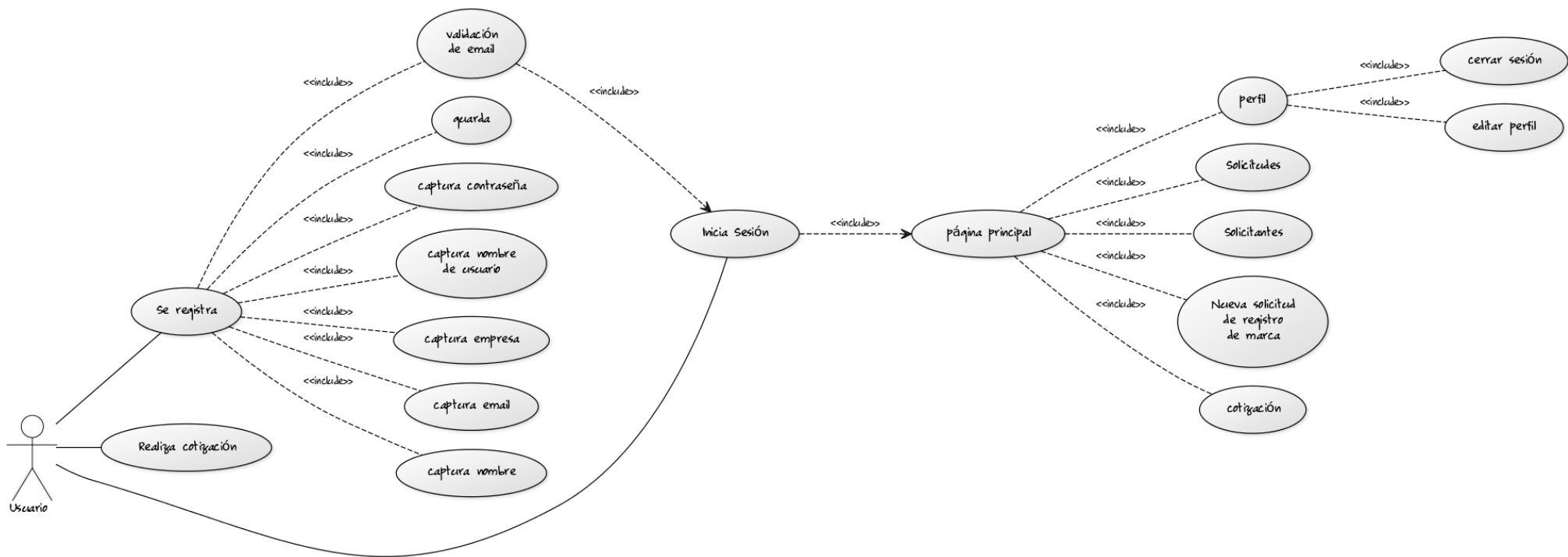
## Diseño de las Pantallas



**Fig 4.3 Diseños propuestos para el desarrollo del sistema pantalla de inicio de sesión.**



**Fig 4.4 Diseños propuestos para el desarrollo del sistema pantalla principal.**



**Fig 4.5 Caso de uso de las posibles formas de la primera interacción con el sistema.**

### 4.3 ORGANIZACIÓN Y DEPURACIÓN DE LA INFORMACIÓN.

Para que pueda quedar claro en este apartado solo se muestran algunas tablas de la base de datos del sistema, en este caso la empresa propietaria de este sistema no desea que se revele la distribución de la información que la compañía manipula, debido a que muchos datos están protegidos bajo un acuerdo de confidencialidad con el cliente.

#### MER

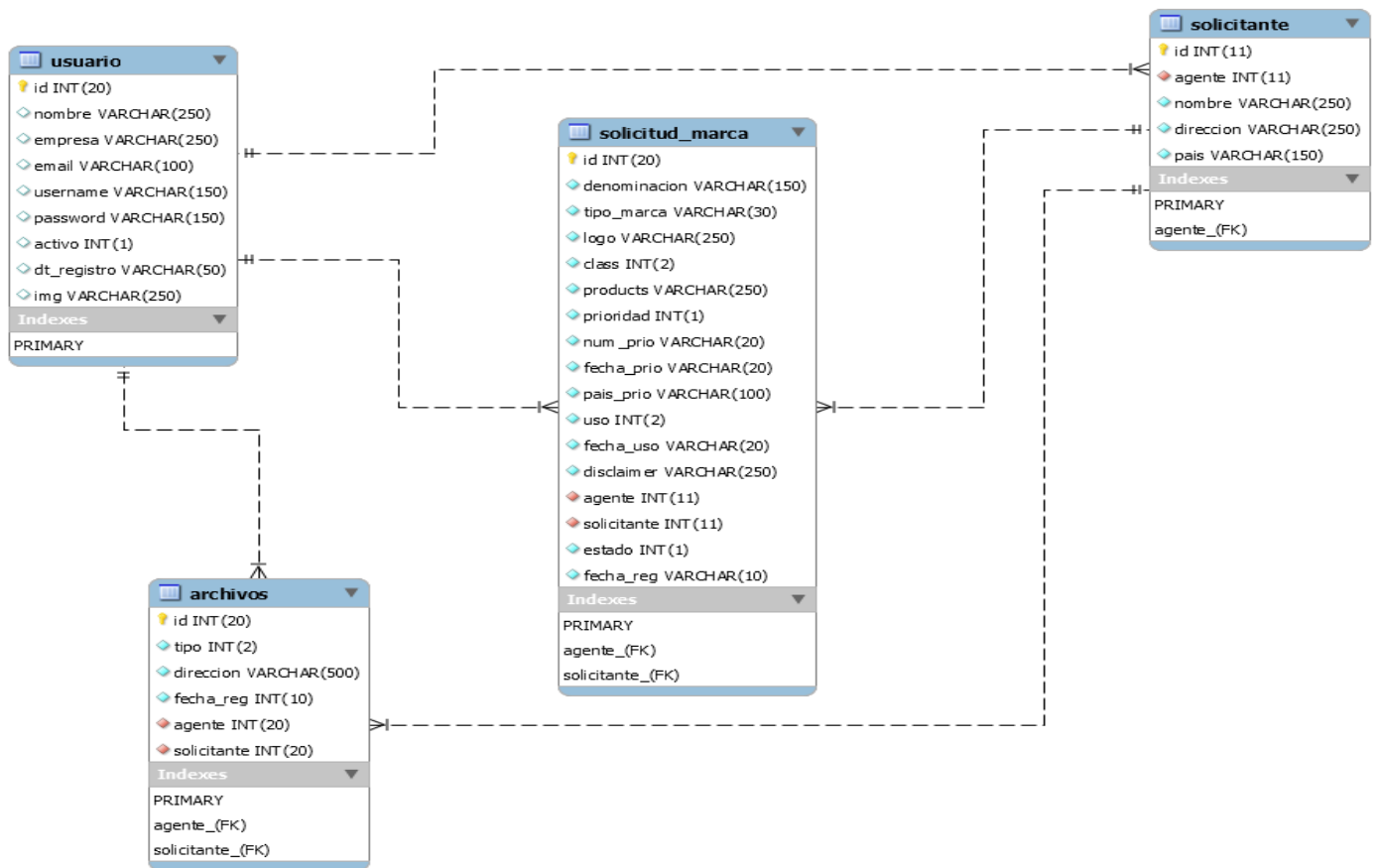


Fig. 4.6 Modelado de datos, modelo relacional de la base de datos de tres tablas del prototipo de “ABLEIP”.

Como se ha explicado anteriormente el sistema estará desarrollado sobre tres plataformas o herramientas que nos permiten que el sistema cumpla con los requerimientos antes planteados, por lo que en este punto describiremos como se implementó el uso de las bases de datos para el desarrollo del sistema. Primeramente, es de vital importancia realizar un modelado de las tablas mediante un modelado de datos en este caso se utilizó el modelo relacional de la herramienta proporcionada por el programa informático de la empresa Oracle (MySQL workbench), estos tipos de modelados ya sea el modelo relacional o el modelo entidad-relación nos permiten hacer una mejor organización de las ideas y del flujo de datos que se realizará cuando se ponga en funcionamiento el sistema.

La Fig. 4.6 ilustra el modelo relacional de la base de datos del prototipo inicial de este sistema y el que se utilizará para describir el funcionamiento principal de dicho sistema, cabe destacar que esto no limita el aumento en el número de tablas de la base de datos, así como su desempeño en el sistema.

En este caso la Fig. 4.6 ilustra el modelo entidad-relación (MER) de las tablas que tenemos a modo de ejemplo de nuestra base de datos, dicho modelo representa la relación entre las cuatro tablas base del sistema y gracias a este modelo se puede realizar un análisis más profundo de cómo van a quedar organizados los datos que serán usados y que tipo de relación se establecerá entre los datos y las tablas, además este modelo nos ayuda a la normalización de los datos en nuestra base de datos.

En este caso se ha realizado una primera forma normal de la siguiente forma:

- Se eliminaron los grupos repetidos en las tablas individuales
- Existe una clave primaria relacionando todas las tablas
- Se separaron los datos independientes de las tablas (por ejemplo: se creó la tabla **solicitante** separada de la tabla **usuario**, y la tabla **archivo** separada de la tabla **solicitud de marca**)

- No se usan varios datos que tienen el mismo valor de información en una misma tabla evitando la redundancia.

Hay varias reglas en la normalización de una base de datos, en este caso al aplicarle estas reglas a la base de datos se considera que la nuestra está en la primera forma normal y por lo tanto se considera que está normalizada en un primer nivel.

En caso de ser necesario se podría normalizar a un nivel más pero debido a que es un prototipo esta base de datos está sujeta a cambios a medida que se realice la etapa de desarrollo, otra característica de este proyecto está en que como se está utilizando la metodología SCRUM los cambios van a ser prácticamente continuos lo cual hace poco viable la normalización de la base de datos en caso de que sea necesaria cada vez que se realice un cambio en alguna de las tablas, ya que este proceso requiere un análisis y exploración de las tablas.

## 4.4 DISEÑO DE MÓDULOS

Para el desarrollo del sistema necesitamos diseñar los módulos del sistema que permitirán navegar por las diferentes funcionalidades del mismo.

En las siguientes imágenes se mostrarán los diseños propuestos en sus inicios y durante el proceso de desarrollo.

La descripción de cada módulo y su funcionamiento se explica desde el subtítulo 5.4 hasta el 5.7.



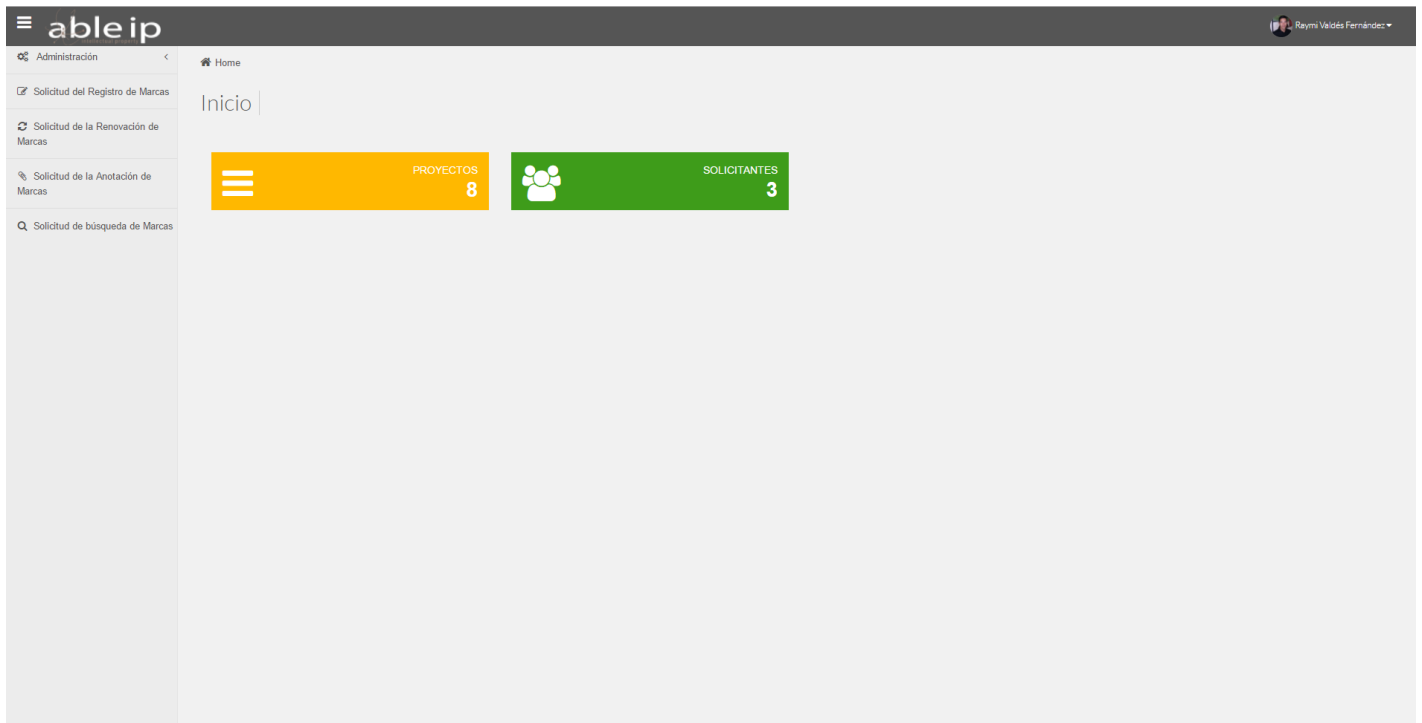
Fig. 4.7 a) Pantalla de Inicio de sesión (primer prototipo).



Fig. 4.7 b) Pantalla de Inicio de sesión (segundo prototipo).



Fig. 4.8 a) Pantalla principal (primer prototipo).



**Fig. 4.8 b) Pantalla principal (segundo prototipo).**



# CAPÍTULO 5 IMPLEMENTACIÓN DEL SISTEMA

## 5.1 CREACIÓN DE LA BASE DE DATOS

En este apartado vamos a establecer en el manejador de bases de datos la serie de tablas de la base de datos ya concebida anteriormente mediante la creación de las tablas de forma conceptual, en este momento se establecen las tablas con los tipos de datos y nombre de las columnas para que tengan una estructura funcional para el manejador de bases de datos, y este a su vez pueda realizar las consultas mediante las peticiones que le haga el sistema.

usuario	archivo	solicitante
Campo: id Tipo: int	Campo: id Tipo: int	Campo: id Tipo: int
Campo: nombre Tipo: varchar Tamaño: 250	Campo: agente Tipo: int Tamaño: 20	Campo: agente Tipo: int Tamaño: 20
Campo: empresa Tipo: varchar Tamaño: 250	Campo: solicitante Tipo: int Tamaño: 20	Campo: nombre Tipo: varchar Tamaño: 250
Campo: email Tipo: varchar Tamaño: 100	Campo: tipo Tipo: int Tamaño: 500	Campo: direccion Tipo: varchar Tamaño: 250
Campo: username Tipo: varchar Tamaño: 150	Campo: marca Tipo: int Tamaño: 11	Campo: pais Tipo: varchar Tamaño: 150
Campo: password Tipo: varchar Tamaño: 150	Campo: class Tipo: varchar Tamaño: 150	
Campo: activo Tipo: int Tamaño: 1	Campo: ubicacion Tipo: varchar Tamaño: 200	
Campo: dt_registro Tipo: varchar Tamaño: 50		
Campo: img Tipo: varchar Tamaño: 250		

**Tabla 5.1 Del Diccionario de Datos y el Diagrama entidad relación**

Fragmento de código SQL por favor ver **Apéndice A**.

## 5.2 INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR DE DESARROLLO

En este caso para realizar la instalación del servidor web, se utilizará Apache como la plataforma del servidor web.

Características del Apache HTTP Server

- Gratuito
- Ampliamente Configurable
- Multiplataforma
- Seguro
- Excelente documentación
- Entre otros.

**APACHE**  
HTTP SERVER



©Copyright 1999 Apache Software Foundation



Fig. 5.1 Funcionamiento del servidor Apache.

### 5.3 INSTALACIÓN DE WAMP SERVER 2.0

Para desarrollar la instalación de WampServer debemos de bajar la versión con la cual vamos a trabajar, en todo caso si quieres trabajar con la última versión, esta la puedes encontrar en la página principal de WampSever.

Una vez descargado el programa procederemos a la instalación de la misma, la cual se mostrará a continuación paso a paso.



**Fig. 5.2 Instalación de WAMPSEVER.**

Este proceso quedará plasmado en el apéndice C con el objetivo de no hacer la descripción del trabajo de tesis muy explicativa en aspectos que no son tan importantes para el objetivo final de la misma ya que el proceso de instalación y configuración del ambiente de desarrollo se hace un poco extenso.

## 5.4 MÓDULO DE REGISTRO Y VALIDACIÓN DE USUARIO



Fig. 5.3 Pantalla de registro de usuarios.

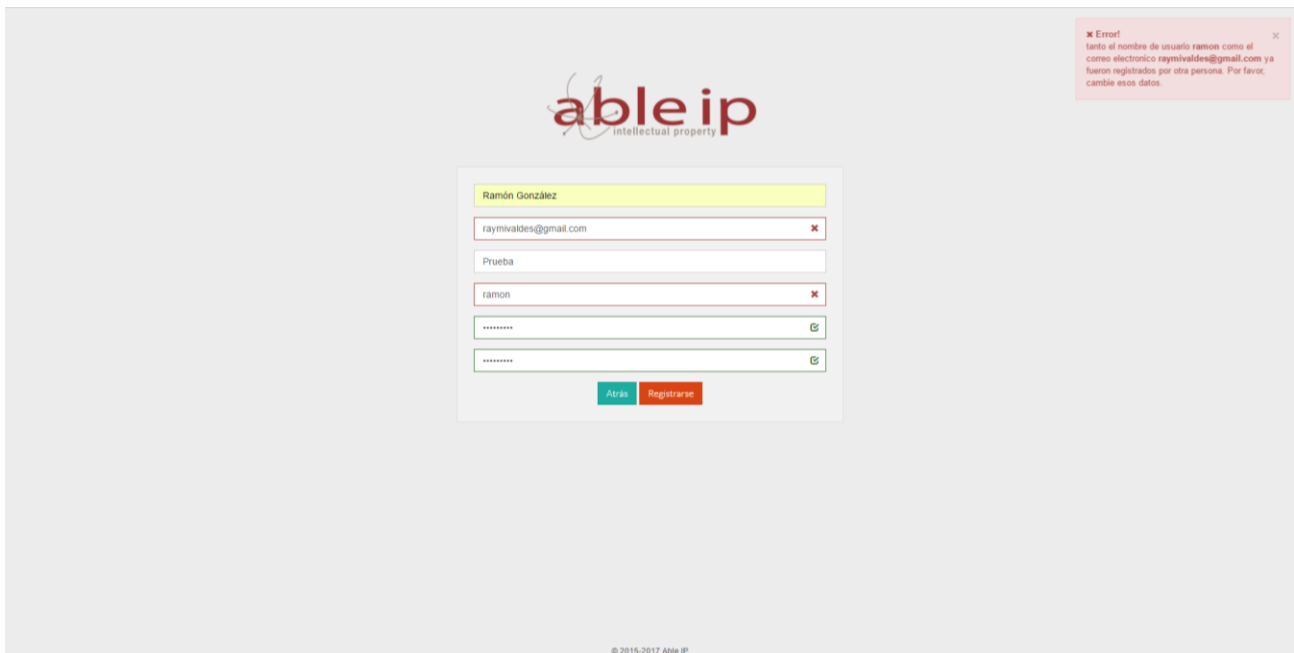


Fig. 5.4 Pantalla de inicio de sesión.

Estos dos módulos son los encargados de contabilizar, validar y registrar los usuarios mediante la asignación de identificadores únicos, a los cuales puedan registrarse en el sistema cumpliendo con los requisitos necesarios.



**Fig. 5.5** Proceso de registro, ejecución de la validación.



**Fig. 5.6** Proceso de registro, resultado de la validación con error.



**Fig. 5.7 Proceso de registro, resultado de la validación satisfactorio.**

Este proceso de registro se completa con la activación de cuenta mediante el envío de un número hexadecimal de 16 caracteres generado aleatoriamente el cual contiene un activador y un número de id generado con encriptación MD5 al igual que las contraseñas del sistema usan esta encriptación.

## **5.5 MÓDULO DE ADMINISTRACIÓN DE USUARIOS**

Para la administración de usuarios en el sistema en general, se hace uso de identificadores, los cuales se mantienen activos hasta que la sesión se destruye, se cierre la sesión o se cierre el navegador.

¿Cómo funciona?

Básicamente es el encargado de mantener la sesión abierta para un usuario específico y en el cliente (refiriéndome al cliente de la arquitectura cliente-servidor) específico el cual en este caso es el navegador de internet (por ejemplo, Mozilla Firefox, Google Chrome, Safari, etc.), ya que el lenguaje PHP permite utilizar las ya conocidas **cookies**. De esta forma el sistema siempre se mantiene activo y controlando si el usuario ha iniciado sesión y qué usuario.

¿Qué pasa si no he iniciado sesión y quiero acceder a uno de los servicios de la aplicación?

En caso de querer realizar alguna acción, éste no te dejará ni siquiera acceder ya que verificará si existe alguna **cookie** con validez dentro del sistema en caso de que la

respuesta sea negativa, redireccionará al usuario a la página inicial que es meramente informativa y en caso de ser positivo se podrá continuar haciendo uso del sistema.



**Fig. 5.8** Página principal, usuario sin autenticarse.



**Fig. 5.9** Alerta de intruso de usuario sin autenticarse.



**Fig. 5.10 Cuenta regresiva para el redireccionamiento.**



**Fig. 5.11 Redireccionamiento hacia la página de inicio.**



## 5.6 MÓDULO DE MARCAS

### Componente de solicitantes

El módulo de marca es el más extenso y al cual se debe este sistema ya que ocupa el 80 por ciento del sistema.

Primeramente, hablaremos de los componentes que conforman este módulo, para esto hay que empezar con el componente de los solicitantes el cual permite almacenar todos los solicitantes que los usuarios van agregando al sistema, los cuales se diferencian mediante los diferentes usuarios lo cual permite que cada solicitante que el usuario agregue se queden registrados para uso futuro, en la actualidad la única acción posible es borrar, pero en un futuro se podrán editar los datos del solicitante.

Otro de los aspectos importantes a considerar dentro de este componente es que se puede crear, agregar o guardar cualquier solicitante de dos métodos, el primer método se encuentra en el apartado de solicitante mediante la opción de “Agregar nuevo solicitante”.

Nombre	Dirección	País	Acción
Raymi Valdés Fernández	Ocotepéc 188 casa 7, San Jerónimo Lídice	México	[Borrar]
asda	quequeque	Angalla	[Borrar]
ALDECOA & ELIAS S DE RL DE CV	Ave Conteras 516 despachos 103-104, San Jerónimo Lídice	México	[Borrar]

**Fig. 5.12 Método 1 para agregar solicitantes.**

El segundo método consiste en realizar la captura de los datos desde la creación de la solicitud de registro de marca, para esto hay que estar en el apartado de registro de solicitud de marca y dentro del formulario ya varias opciones para agregar un solicitante a la solicitud de marca. La primera forma consiste en seleccionar de una lista un solicitante previamente capturado, y la segunda es creándolo en ese instante, cabe destacar que el solicitante que se cree durante el proceso de captura de los datos para una solicitud de registro de marca ya queda registrado para siempre en el sistema a no ser que el usuario lo elimine.

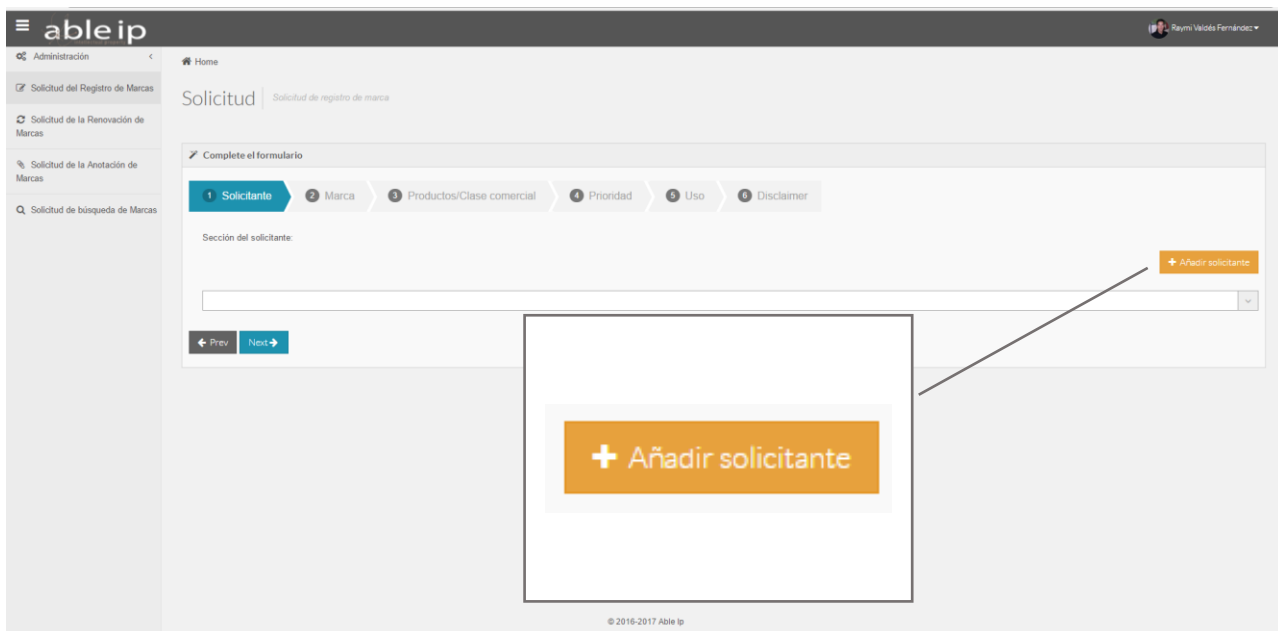


Fig. 5.13 Método 1 para agregar solicitantes.

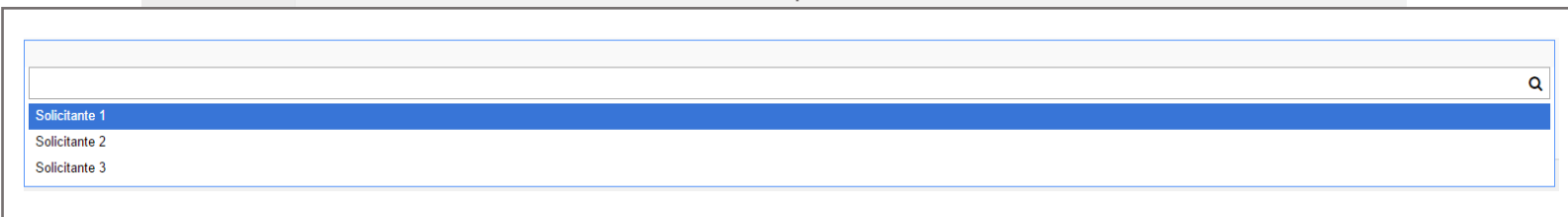
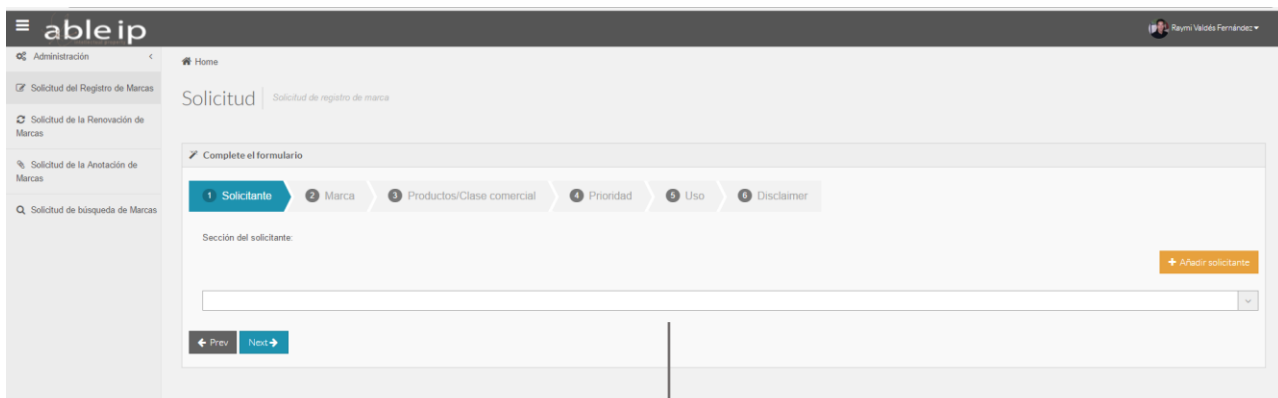
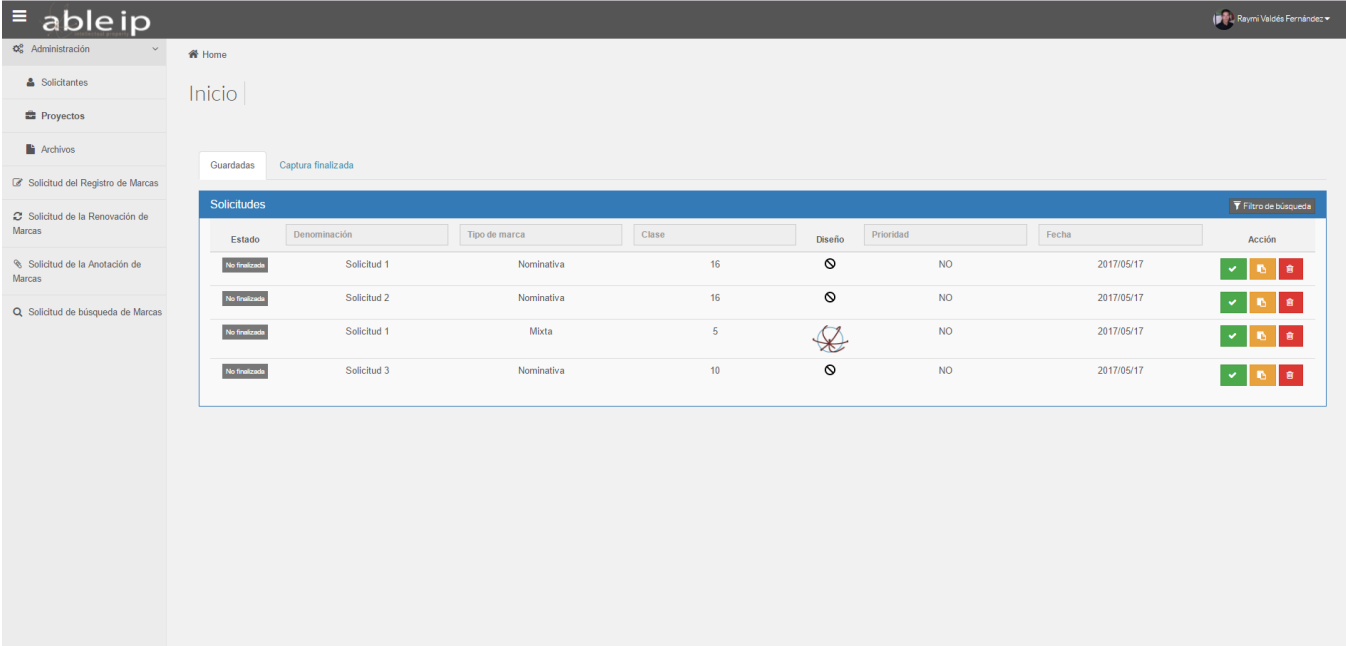


Fig. 5.14 Método para seleccionar solicitantes durante la captura de datos de la solicitud de registro de marca.

## Componente de proyectos

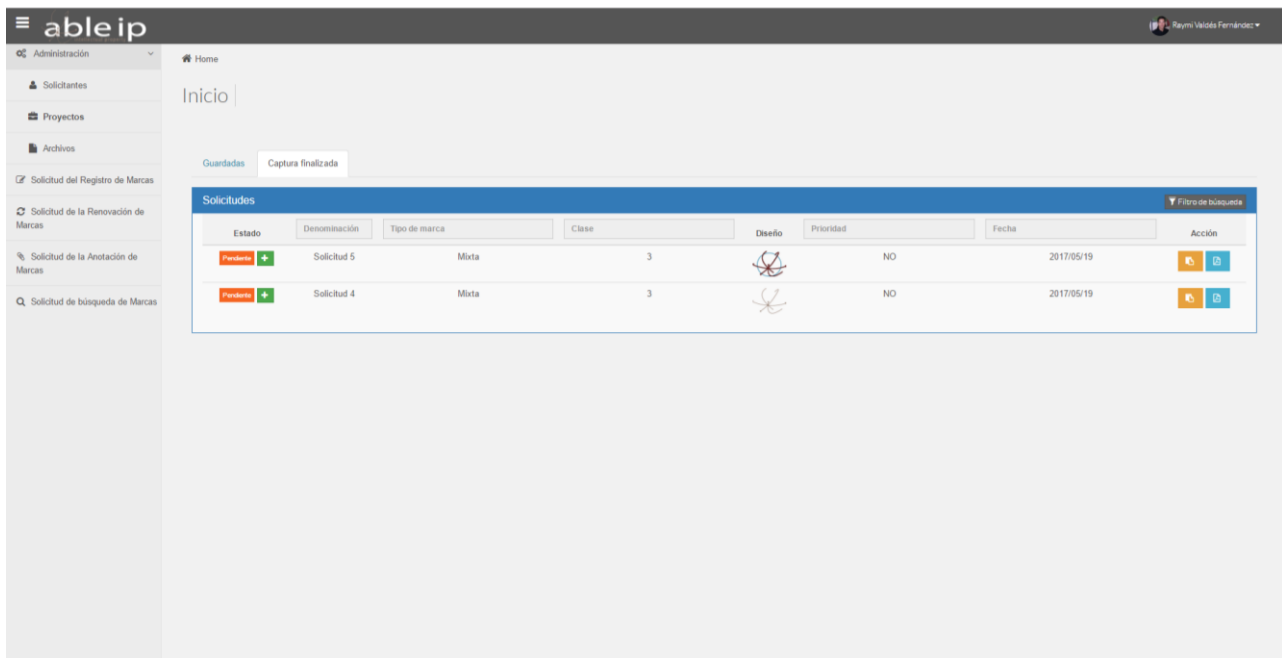
Otro de los componentes que conforma el módulo de marcas es el componente de proyectos el cual le permite al usuario de una forma sencilla listar todas las solicitudes que han capturado en el sistema dependiendo del estado en que se encuentren, se le irá mostrando a medida que el estado de la captura vaya pasando por las diferentes etapas hasta concluir de forma satisfactoria o en caso contrario no terminar el proceso debido a problemas que van surgiendo durante el proceso del registro de la marca.



The screenshot shows the 'able ip' web application interface. The left sidebar contains navigation options: Administración, Solicitantes, Proyectos, Archivos, and a search bar for 'Solicitud de búsqueda de Marcas'. The main content area is titled 'Inicio' and shows a filter for 'Guardadas' (Saved) applications. Below this is a table of 'Solicitudes' (Applications) with the following data:

Estado	Denominación	Tipo de marca	Clase	Diseño	Prioridad	Fecha	Acción
No finalizada	Solicitud 1	Nominativa	16	⊗	NO	2017/05/17	✓ ↻ 🗑️
No finalizada	Solicitud 2	Nominativa	16	⊗	NO	2017/05/17	✓ ↻ 🗑️
No finalizada	Solicitud 1	Mixta	5	⊗	NO	2017/05/17	✓ ↻ 🗑️
No finalizada	Solicitud 3	Nominativa	10	⊗	NO	2017/05/17	✓ ↻ 🗑️

**Fig. 5.15** Componente de listado de proyectos “Guardadas” en un estado de “captura no finalizada”.



**Fig. 5.16 Componente de listado de proyectos “captura finalizada” en un estado de “pendiente”.**

Como se puede observar el administrador de proyectos está dividido en dos secciones las cuales se describirán a continuación.

Secciones:

- **Guardadas.** En esta sección se encuentran los proyectos (solicitudes de varios tipos) que no se le ha concluido el proceso hasta su finalización, esto puede deberse a varias causas, información incompleta, cambio de información, desconocimiento de la forma de uso del sistema, en esta sección los proyectos estarán almacenados hasta que se les complete el proceso de finalizar captura la cual consta de revisar de nuevo toda la información con el fin de asegurar que esté bien o por el contrario si el usuario desea eliminar el proyecto también puede.
- **Captura finalizada.** Esta sección es la que contiene todas las solicitudes que ha realizado el usuario, en esta sección se pueden ir monitorizado los proyectos y ver en qué estado están en el proceso de presentación de la solicitud en la oficina de propiedad industrial correspondiente.

**Cabe aclarar que este prototipo está simplificado para México con el objetivo de que la explicación de la teoría de propiedad intelectual sea de mayor facilidad de entendimiento, pero en próximas actualizaciones se extenderá su alcance a nivel internacional**

## **Componente de solicitud de registro de marcas**

Ya para finalizar el módulo de marcas queda por explicar el último y más importante de los componentes el cual consiste en un formulario de solicitud de registro de marca el cual permite introducir todos los datos según la elección que el usuario realice, en este componente se hace referencia a los otros dos componentes de forma implícita mediante la validación de los solicitantes y las solicitudes previas realizadas para validar si existe alguna anterioridad y en ese caso no se hace ningún cambio en la anterior y la nueva se guarda en la base de datos sin tener conflictos con la anterior, en el caso de finalizar la captura se hace mediante el uso del identificador de la fila de la tabla antes guardada y lo único necesario a hacer es cambiar el estatus de la solicitud de un estado guardado a un estado finalizado.

Este componente hace referencia a más de 4 tablas de la base de datos y las relaciona a la vez para que se puedan mostrar los resultados de forma satisfactoria y permite que las consultas sean exitosas.

A continuación, veremos cómo se muestra este componente en el apéndice B

## 5.7 MÓDULO DE ADMINISTRACIÓN DE ARCHIVOS

Este módulo es el encargado de la generación de archivos y de generar una estructura de archivos que genera una carpeta principal o “raíz” capaz de almacenar todos los archivos de las marcas distribuidos en esta carpeta “raíz” mediante la asignación de nombres dependiendo de la denominación de la marca y la clase a la cual pertenezca; un ejemplo del nombre de la carpeta sería “Solicitud 1 en clase 15”, dentro de cada carpeta se encontrarán todos los archivos ya sea generados por el sistema **ABLEIP**, archivos que el usuario podrá subir (no implementado en este prototipo) y archivos que el administrador del sistema también podrá subir en caso de ser necesario (no implementado en este prototipo).

Además, una vez generados los archivos y listados en este módulo el usuario puede disponer de ellos y descargarlos si lo desea.

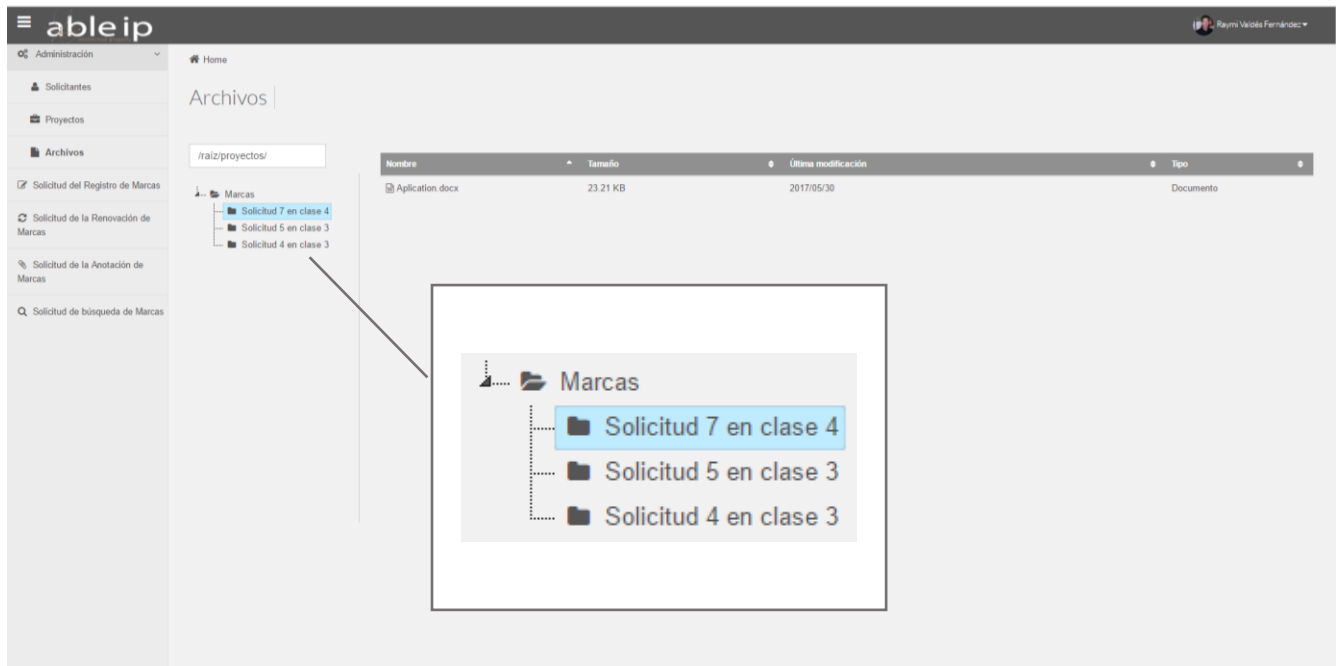
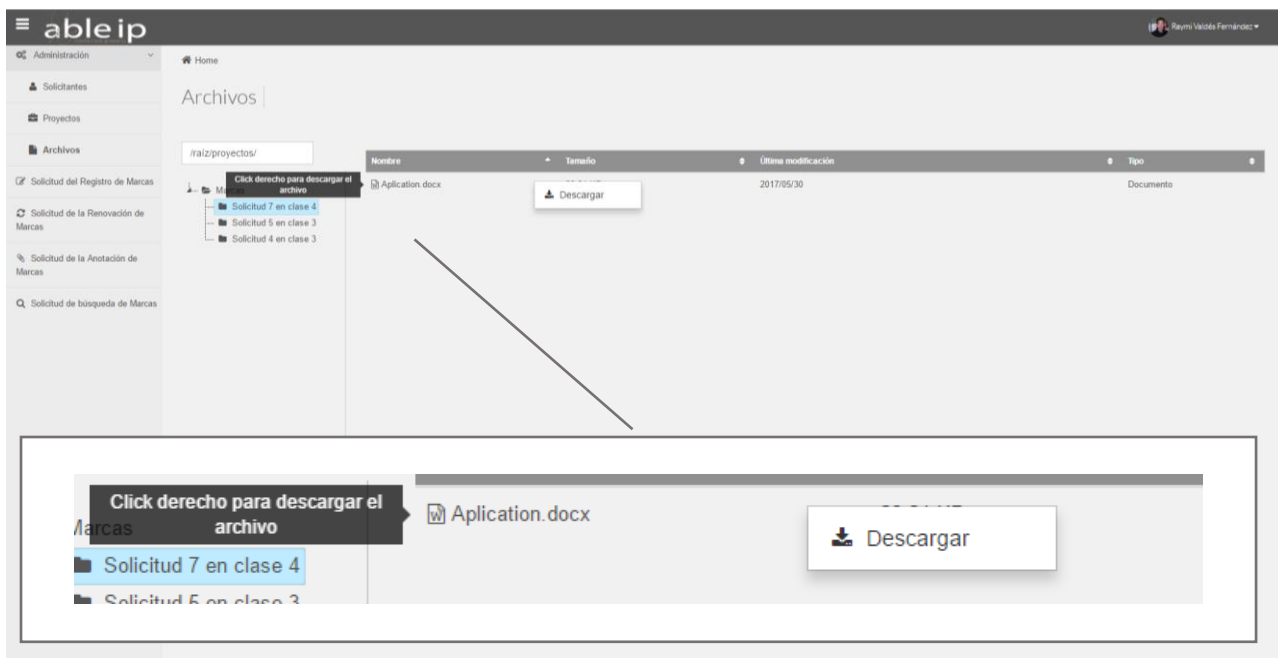


Fig. 5.23 Administrador de archivos



**Fig. 5.24 Archivos listados en la carpeta y la opción de descarga.**

# CAPÍTULO 6 PLAN DE PRUEBAS

## 6.1 EJECUCIÓN DE PRUEBAS

En este punto ejecutaremos algunas pruebas de estrés sobre todo para verificar que la concurrencia de usuario no va a afectar el rendimiento del sistema, lo cual es importante, ya que mientras un usuario está realizando una consulta en la base de datos realizando una captura de datos en una solicitud de registro de marca.

Por otro lado, la carga de trabajo para las pruebas solo fue de 10 usuarios simultáneos con número de click o consultas de un número aleatorio en aumento en un tiempo de 5 mins, estos parámetros se establecieron así sabiendo y partiendo desde la base de que el sistema no está enfocado a que reciba una gran carga de trabajo. A pesar de esto los resultados son sobresalientes y no arrojan algún error de fallo durante el aumento de la carga de trabajo.

### Datos de la prueba de estrés

Test Type: RAMP (run test for 5 minutes)

User Simulation: ramp test with up to 10 simultaneous users - 20 seconds between clicks (Random)

URL#	Name	Click Delay [s]	URL	POST data (or @filename@)	Username	Password
1		0	http://raymaso.com/ableip/			

Tabla 6.1 Datos.

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s	Cookies
1	35	34	0	254	569,194	526.86	
2	31	30	0	285	502,230	469.62	
3	27	26	0	295	435,266	453.56	
4	25	24	0	290	401,784	461.70	
5	19	18	0	258	301,338	519.01	
6	20	19	0	252	318,079	531.78	
7	19	18	0	250	301,338	535.69	
8	14	13	0	259	217,633	516.27	
9	12	11	0	250	184,151	535.14	
10	13	12	0	258	200,892	519.01	

Tabla 6.2 Resultados por usuarios

### Resultados por URL



URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1		202	0	0.00	54,232	268

Tabla 6.3 Resultados por URL.

Pruebas de velocidad y respuesta

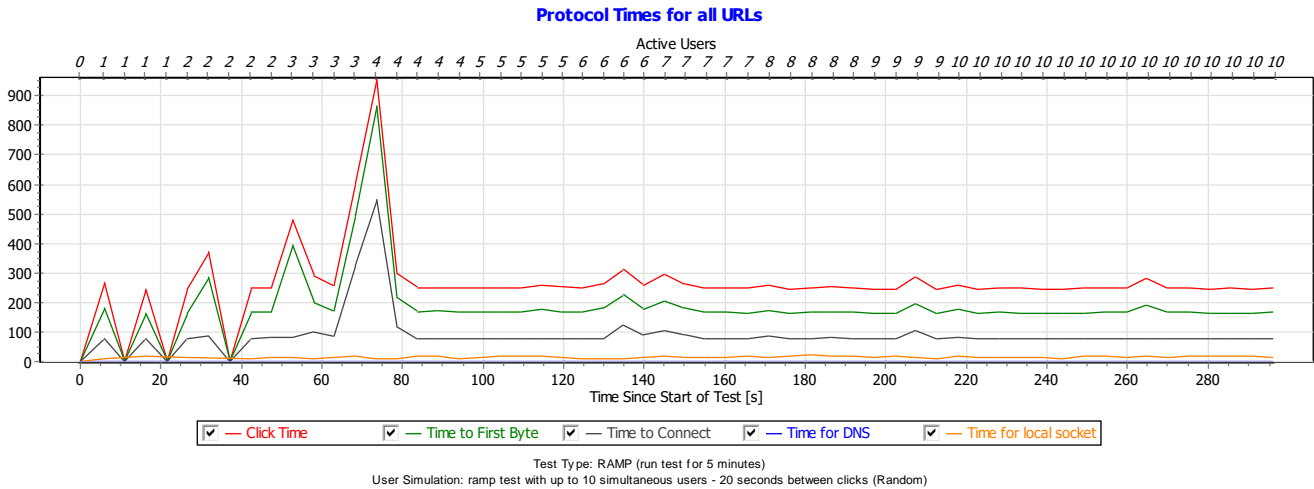


Figura 6.4 Gráfica de tiempos de conexión y respuesta.

Pruebas de consumo de datos y utilización del ancho de banda

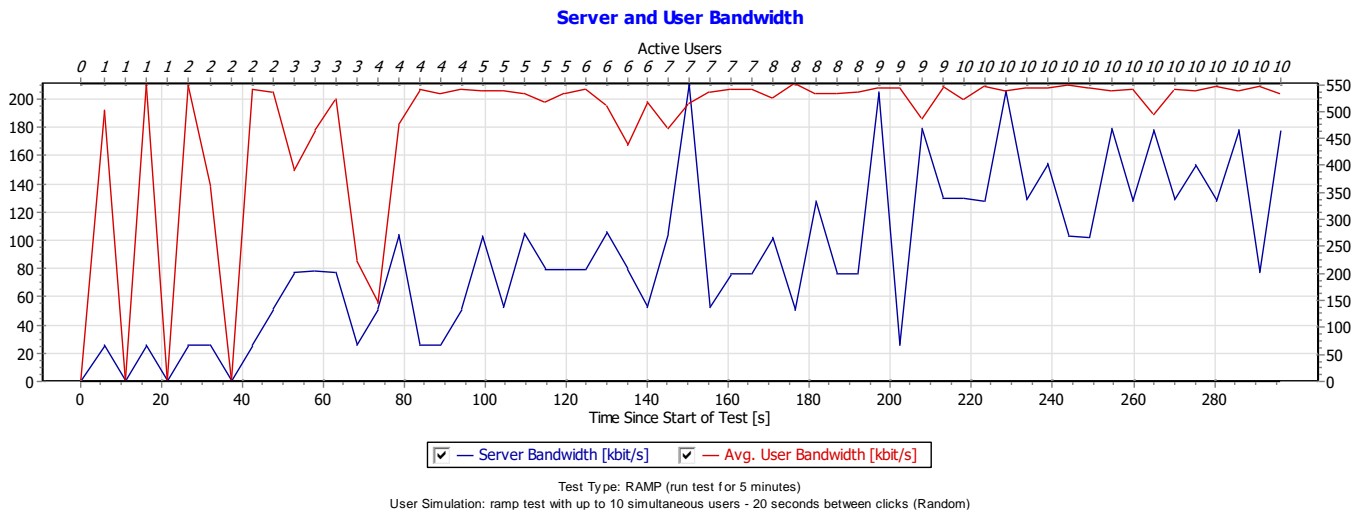
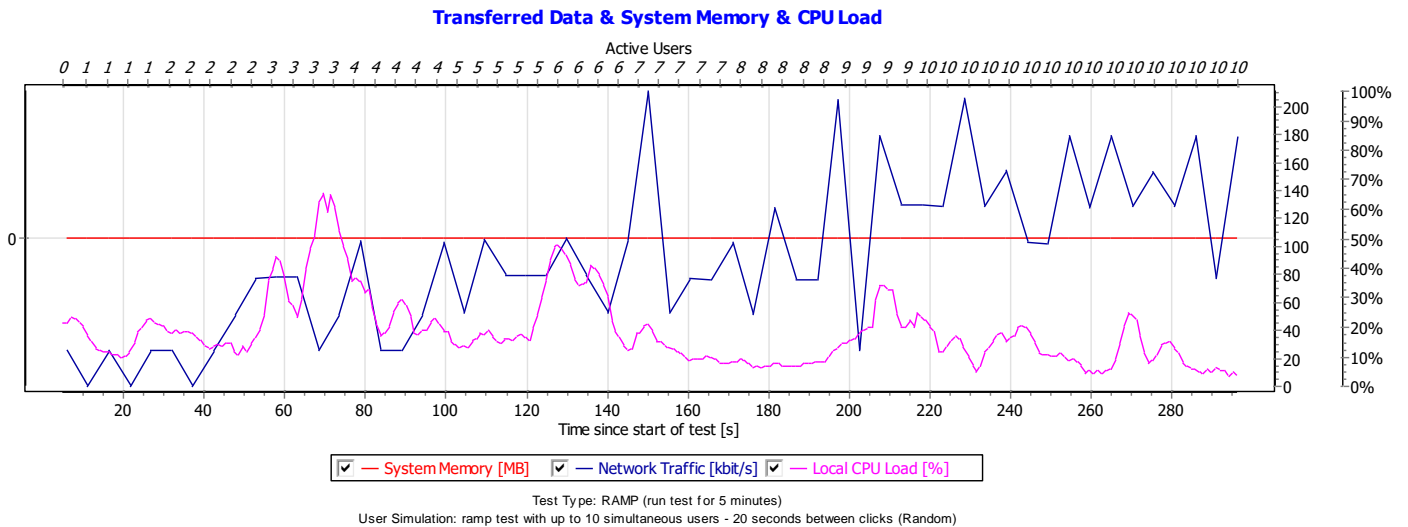
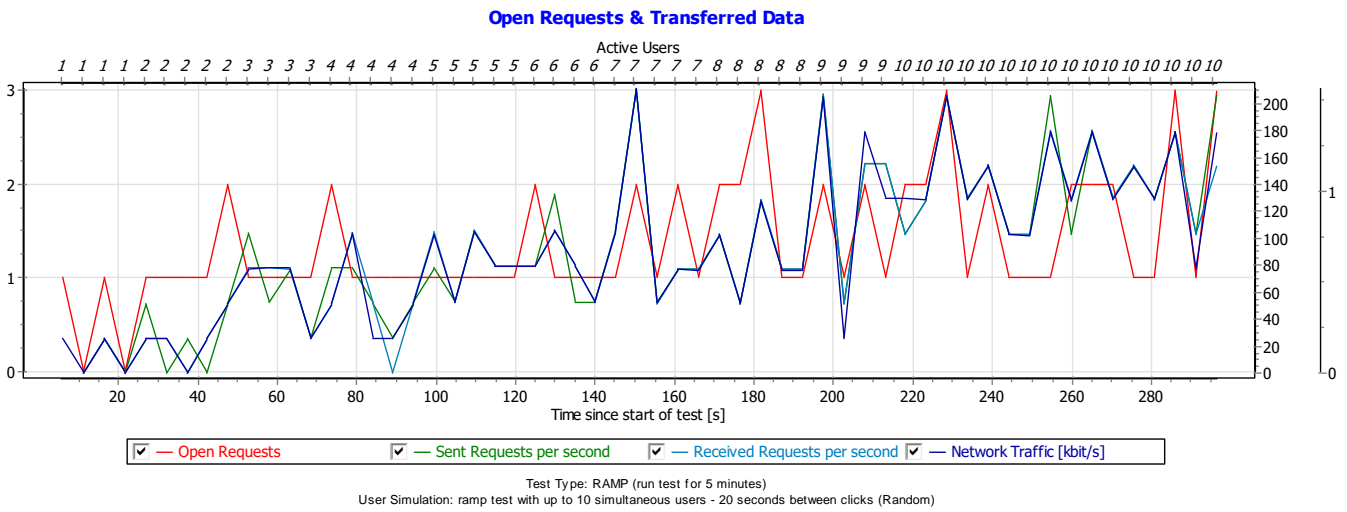


Figura 6.5 Ancho de banda del servidor y el cliente.

# Prueba de consumo de recursos



**Figura 6.6 Datos transferidos, carga del CPU y de la memoria.**



**Figura 6.7 Datos transferidos y solicitudes (POST, GET).**

## RESULTADOS

Para la presente tesis se investigó el estado actual de los sistemas en la propiedad intelectual en el sector privado en México y como resultado de la investigación y el análisis se propuso una solución más eficaz y automatizada para que los clientes (usuarios) puedan realizar los procesos de solicitud mediante un sistema en línea y con respuestas automáticas de forma que le permita a este sector expandir sus horizontes mediante los sistemas electrónicos ya que este sector tiende a ser conservador en el aspecto de la tecnología.

Para esto se crea **ABLEIP** el cual es un sistema electrónico en línea el cual permite a los clientes de determinada empresa realizar las solicitudes de marcas de manera sencilla y rápida permitiéndoles economizar tiempo y costos, además al ser en línea permite que en cualquier parte del mundo se puede dar seguimiento a los proyectos y trámites que están almacenados en el sistema, a su vez el sistema permite realizar cotizaciones sin tener que ser parte del sistema como usuario, ya que este servicio se hace de manera informativa y para atraer a los posibles clientes.

Durante el proceso de creación de la presente tesis, se fue diseñando y estructurando cada módulo para lograr los objetivos planteados, los cuales se llegaron a cumplir satisfactoriamente.

- **Cálculo de cotización de la solicitud del registro de marcas.** Está realizado de forma sencilla y amigable, de modo que sin tener que formar parte del sistema como usuario autenticado se puedan realizar cotizaciones, con resultados de envío por correo electrónico.
- **Solicitud del registro de marcas.** Este fue el reto del sistema, el núcleo, 80 por ciento del sistema es el módulo de marca el cual parte de un formulario realizado de una manera sencilla y amigable para el usuario con el fin de lograr un proceso de creación de la solicitud de manera rápida y sin trabas ni problemas.
- **Rastreo del estado de trámites.** En este apartado se logró hacer una distribución de la información de manera identificada y controlada por elementos identificadores de los usuarios, así como con los identificadores de las mismas solicitudes de marcas, para tener una organización por nivel de usuario, solicitud de marca, dándole al usuario la posibilidad de visualizar el proceso del “proyecto” durante el recibo de notificaciones de estado.
- **Reportes de altas/bajas/cambios en el registro de casos.** Uno de los puntos importantes fue realizar un sistema de administración con niveles de acceso en dependencia del tipo de usuario, el cual a pesar de no ser un módulo demasiado

grande va creciendo y tomando más personalidad a medida que se aumenten los alcances de este sistema.

- **Administración de archivos.** Al ser un sistema de administración y que brinda un servicio a cierto sector del mercado es necesario crear y generar archivos en varias extensiones (.pdf, .docx, .jpg) los cuales contienen la información consultada de bases de datos e información recabada de formularios que el mismo usuario llena, para esto es necesario organizarla y esto se logró mediante la participación de otros módulos, generando vistas amigables y de fácil acceso a los usuarios.
- **Sistema de notificaciones y envío de correos electrónicos.** Este sistema de notificaciones es un sistema sencillo pero gran utilidad que hace uso de API'S que permiten el envío de correos electrónicos de manera fácil y notificaciones en cualquier momento dentro del sistema, lo cual hace un sistema más atractivo y con interacción con el usuario.

La presente tesis acota el sistema hasta los objetivos antes mencionados, pero el sistema **ABLEIP** no se limita a esos módulos solamente, ya que este sistema todavía sigue en desarrollo y en un futuro contará con varias mejoras, como, por ejemplo:

- Cálculo de cotización de la Renovación de Marcas
- Solicitud de la Renovación de Marcas
- Cálculo de cotización de la Anotación de Marcas
- Solicitud de la Anotación de Marcas
- Cálculo de cotización de la Solicitud de Búsqueda
- Solicitud de Búsqueda
- Pagos en línea con carrito de compra

## CONCLUSIONES

Para concluir con este proyecto de tesis, es válido hacer un resumen de lo realizado y explicado.

El sistema realizado **ABLEIP** logra de manera fácil y eficaz a la par de amigable, transmitirle al usuario la sensación de sencillez, un ejemplo claro se encuentra en el proceso guiado de la realización de la solicitud de registro de marca, para realizar este sistema se investigó sobre el entorno en este segmento del mercado y se tomó la decisión de realizar **ABLEIP** con los objetivos ya antes mencionados y con visión de expansión mundial.

Cada objetivo listado en la presente tesis, ya sea el objetivo general del sistema como los objetivos específicos se cumplieron en su totalidad, permitiéndole al sistema permanecer funcional y a su vez que se le puedan realizar mejoras y cambios, al cierre de esta tesis se encuentra en una versión alfa por lo que le queda mucho camino por recorrer, dentro de los cambios o mejoras podrían ser:

- Generación de pagos
- Facturación y notas de débito
- Carga de archivos por parte del usuario
- Mejor adaptación al ambiente móvil

En general el sistema parte de la idea de automatizar el proceso de registro de marcas para los usuarios externos del país en que se presenta y facilitar y organizar la información necesaria para realizar este proceso legal de la manera más eficiente posible evitando un máximo de factor humano involucrado dando así un protagonismo importante a los sistemas electrónicos en este proceso.

En México existe la posibilidad de realizar este proceso de solicitudes de marcas en línea proporcionado por el gobierno mexicano mediante el IMPI (Instituto Mexicano de la Propiedad Industrial) permitiéndole a que las personas en México puedan realizar este proceso, las diferencias que quiero hacer son dos importantes, la primera es que los clientes, agente o usuarios fuera del país que no cuenten con una clave SAT no podrán hacer uso de este servicio por lo que automáticamente queda descartado para agentes de otros países que no cuenten con una empresa o persona física en el país que pueda realizar este trámite en línea y la segunda diferencia con **ABLEIP** es que no guarda los solicitantes por usuarios, esto de forma rápida parece no muy grave pero hay que hacer notar que en México el año pasado se presentaron 57,684 marcas de las cuales el 60% se realizaron mediante este sistema de "Marcas en línea", por lo tanto es de gran ayuda realizar un guardado de los solicitantes para no tener que hacer el registro cada vez que se necesite registrar una marca ya que grandes empresas pueden presentar hasta más de 100 marcas en un lapso no tan grande de entre de 1 a 5 días, por lo que esto mejoraría el proceso y lo haría más eficiente.

En resumen, creo que **ABLEIP** es un sistema con grandes bases y que puede llegar muy lejos, tan lejos como nuestra imaginación y los modelos de negocios nos dejen crecer, y

aplicando una de los beneficios de los sistemas electrónicos es que se pueden ir modificando sobre la marcha y así se puede ir brindando un mejor servicio a medida que la realimentación con el usuario crezca.

Para finalizar quiero agradecerle a la Facultad de Ingeniería por estos años de preparación y decir que la Facultad de Ingeniería me ha aportado grandes conocimientos en innovación, desde metodologías para definir un problema hasta crear sistemas propios. Es una gran experiencia estudiar la carrera de ingeniería en esta facultad y es un honor. Se comparte con personas que piensan distinto, lo que es un gran aporte a mi formación profesional. Los conocimientos adquiridos a lo largo de todo el proceso de formación me sirvieron para tener herramientas en las cuales apoyarme para desarrollar esta idea y otras a futuro, creo que la Facultad está cumpliendo con su objetivo primario de crear profesionales que pueden estar a la altura de cualquier situación ya sea profesional o en su vida cotidiana, generando así individuos que hacen grandes aportes a la sociedad y aportando cada uno un granito de arena para ir mejorando como personas día a día.

# APÉNDICES

## APÉNDICE A

### SQL Creación de tablas

```
-- Base de datos: `ableip`

CREATE DATABASE ableip;

-----

--
-- Estructura de tabla para la tabla `solicitante`
--

DROP TABLE IF EXISTS `solicitante`;
CREATE TABLE IF NOT EXISTS `solicitante` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `agente` int(11) NOT NULL,
  `nombre` varchar(250) COLLATE utf8_unicode_ci NOT NULL,
  `direccion` varchar(250) COLLATE utf8_unicode_ci NOT NULL,
  `pais` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`))

-- Volcado de datos para la tabla `solicitante`
--

INSERT INTO `solicitante` (`id`, `agente`, `nombre`, `direccion`,
`pais`) VALUES
(7, 1, 'Raymi Valdés Fernández', 'Ocotepc 188 casa 7, San Jeronimo
Lidices', 'México')

-----

--
-- Estructura de tabla para la tabla `solicitud_marca`
--

DROP TABLE IF EXISTS `solicitud_marca`;
CREATE TABLE IF NOT EXISTS `solicitud_marca` (
  `id` int(20) NOT NULL AUTO_INCREMENT,
  `denominacion` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `tipo_marca` varchar(30) COLLATE utf8_unicode_ci NOT NULL,
  `logo` varchar(250) COLLATE utf8_unicode_ci NOT NULL,
```

```

`class` int(2) NOT NULL,
`products` varchar(250) COLLATE utf8_unicode_ci NOT NULL,
`prioridad` int(1) NOT NULL,
`num_prio` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
`fecha_prio` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
`pais_prio` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
`uso` int(2) NOT NULL,

`fecha_uso` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
`disclaimer` varchar(250) COLLATE utf8_unicode_ci NOT NULL,
`agente` int(11) NOT NULL,
`solicitante` int(11) NOT NULL,
PRIMARY KEY (`id`))

```

```

-----
--
-- Estructura de tabla para la tabla `usuario`
--

```

```

DROP TABLE IF EXISTS `usuario`;
CREATE TABLE IF NOT EXISTS `usuario` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(250) COLLATE utf8_unicode_ci NOT NULL,
  `empresa` varchar(250) COLLATE utf8_unicode_ci NOT NULL,
  `email` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
  `username` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `password` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `activo` int(1) NOT NULL,
  `dt_registro` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `img` varchar(250) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`))

```



# APÉNDICE B

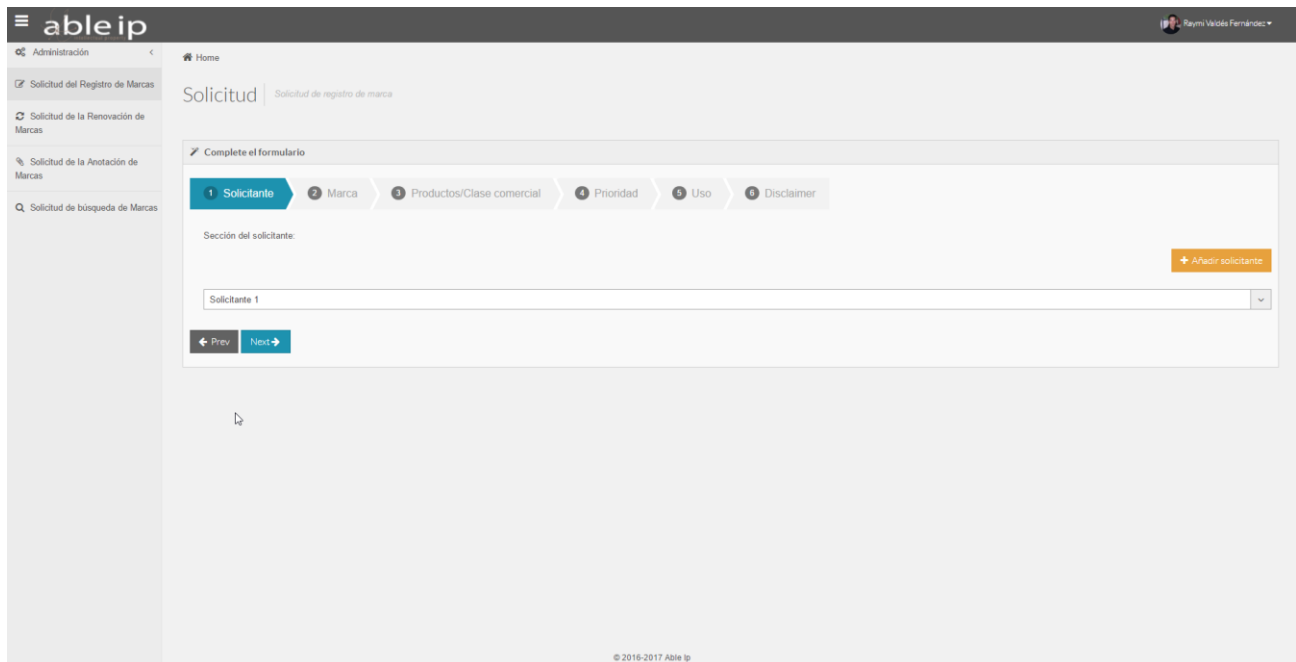


Fig. 5.17 Inicio de proceso de solicitud de marca

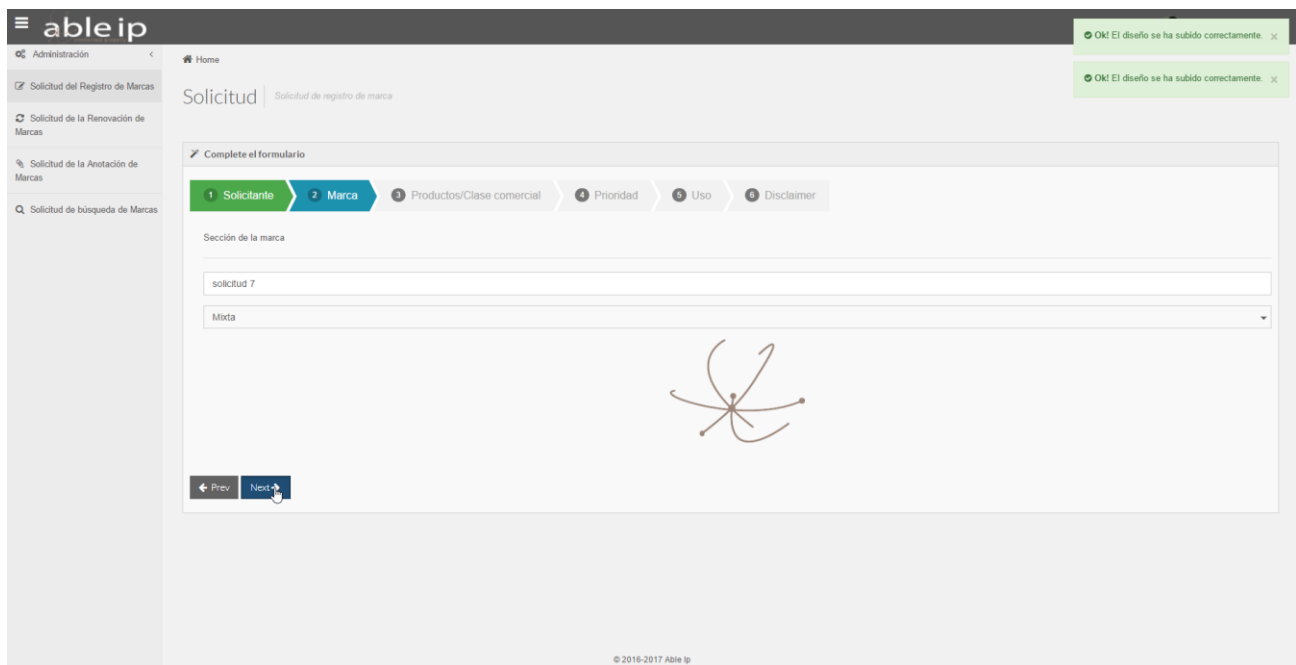
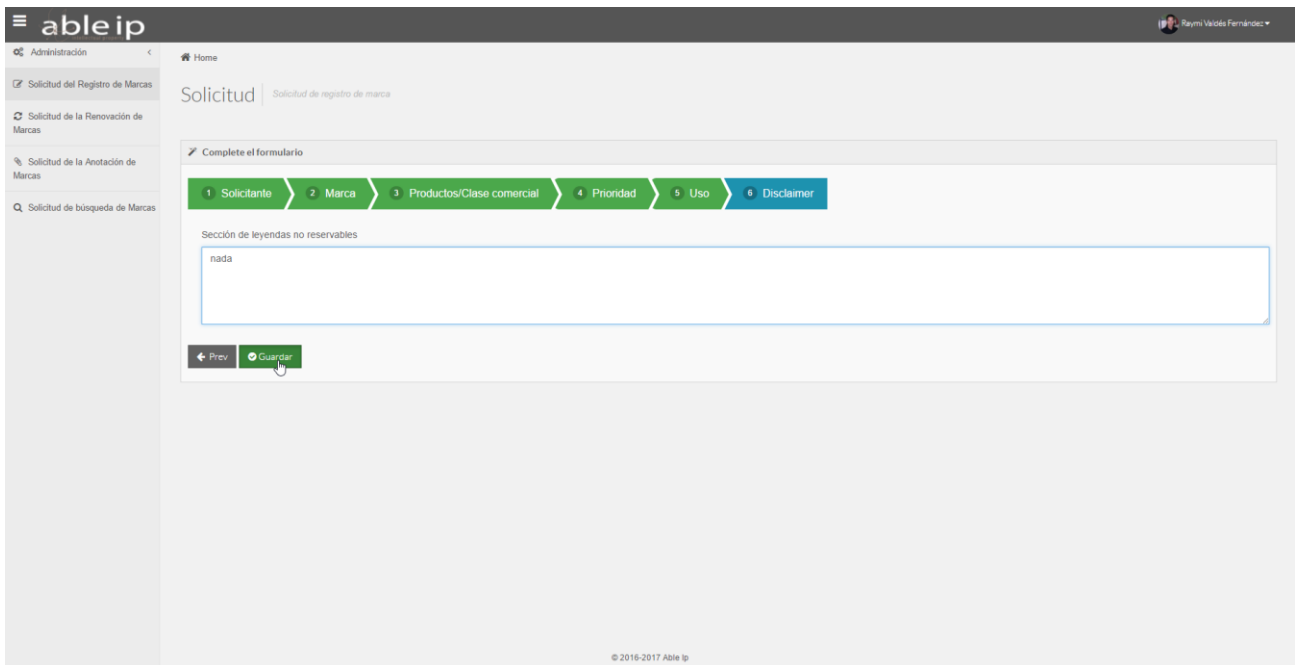
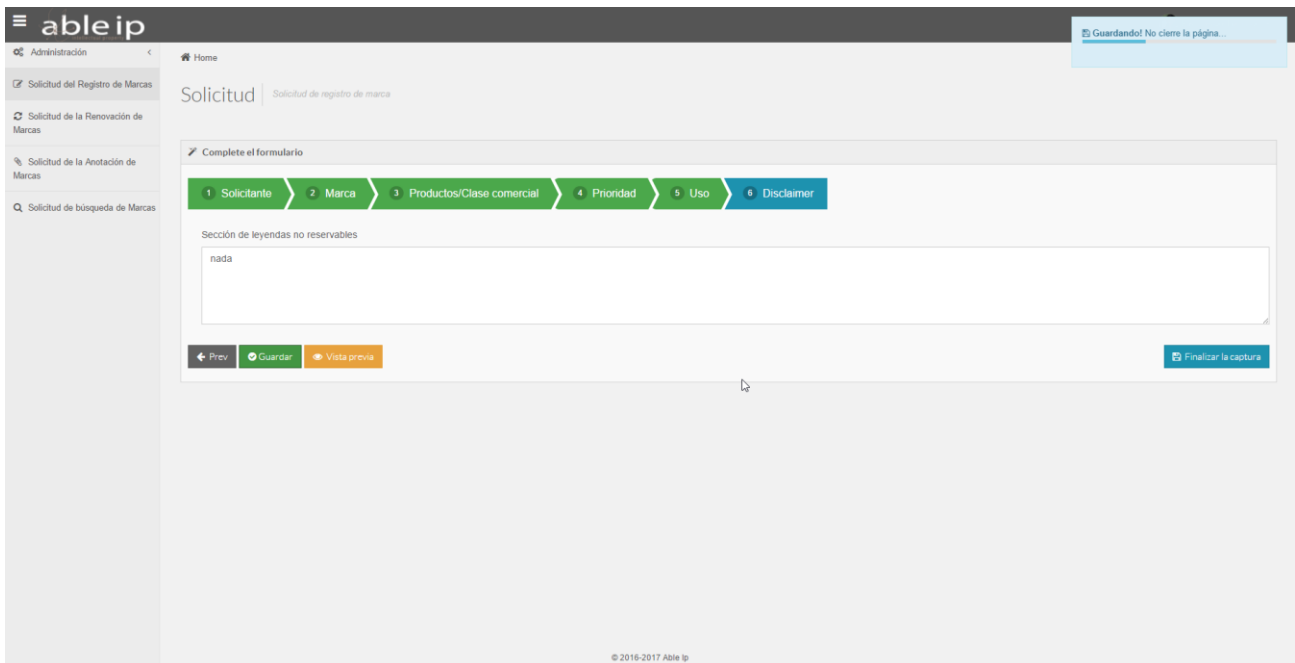


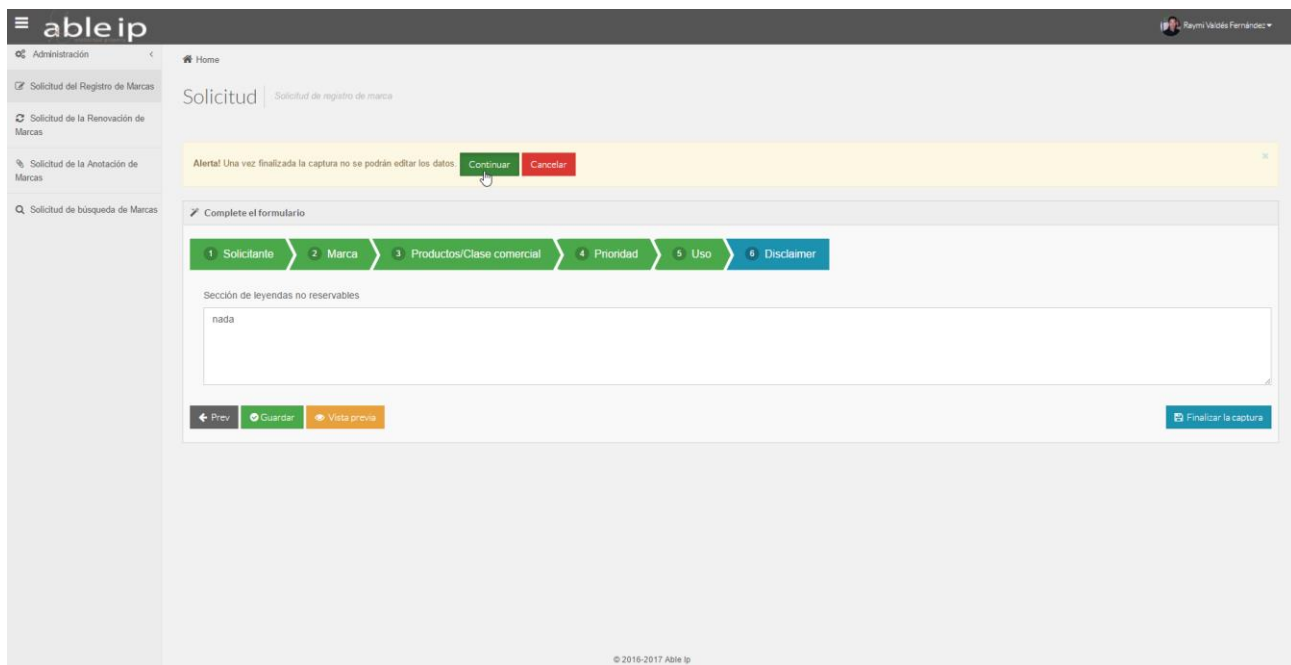
Fig. 5.18 Subida de Imagen



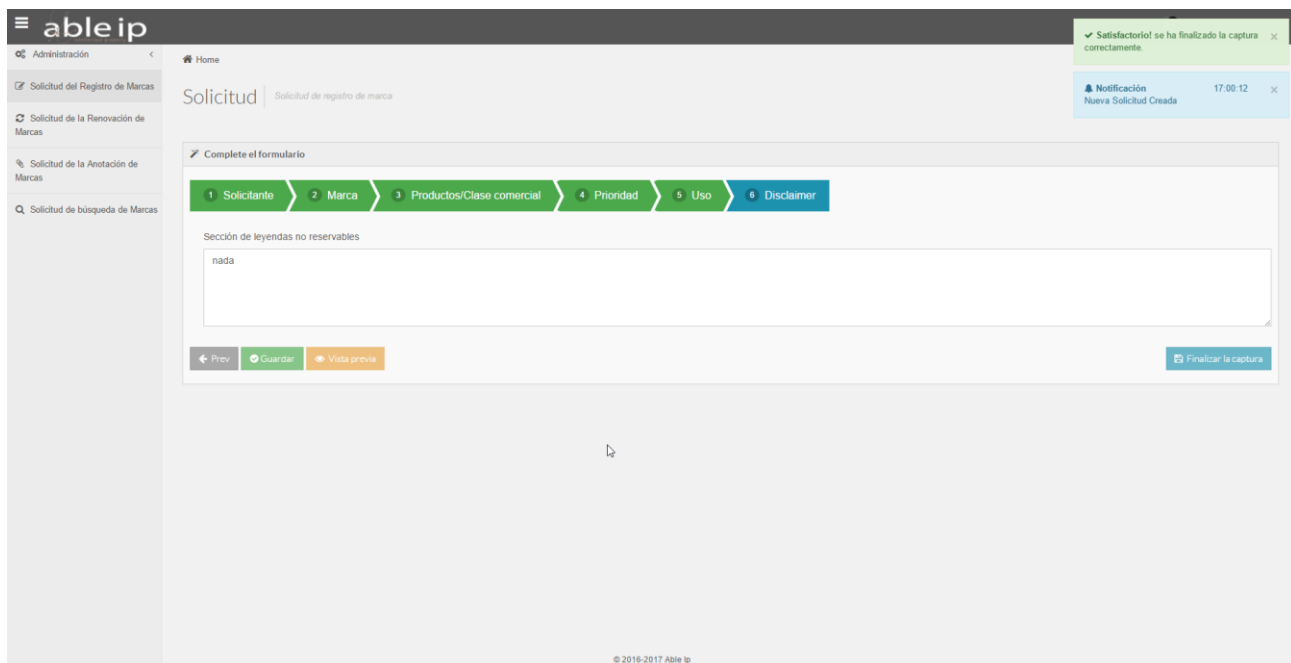
**Fig. 5.19 Guardado de la solicitud**



**Fig. 5.20 Visualización de las opciones desbloqueadas**



**Fig. 5.21 Confirmación de finalización de captura**



**Fig. 5.22 Representación de las notificaciones**

De la Fig. 5.17 a la Fig. 5.22 se muestra el proceso sencillo de como presentar una solicitud de registro de marca en el sistema **ABLEIP**, teniendo como resultado esperado éxito, durante todo el proceso se lanzan notificaciones y alertas para que el usuario esté al tanto de que el proceso va transcurriendo de forma satisfactoria o no.

## APÉNDICE C

### Instalando Apache

Utilizaremos Apache 2.4.17 HTTP Server para Windows como nuestro servidor Web.

¿Cómo instalar el Servidor Apache?

En la práctica Apache se instala de varias formas:

- El programa exclusivo, al que después se le agregan manualmente distintos módulos. Es lo que se explica en este artículo.
- El paquete XAMPP que incluye Apache, la base de datos MySQL, PHP y Perl.
- Otras distribuciones como LAMP o MAMP.

La instalación de la aplicación es sencilla.

Descarga de Apache.org. la última versión para Windows, puedes utilizar el siguiente vínculo: <https://httpd.apache.org/download.cgi#apache22>

1. Crea dos carpetas en la unidad C, la primera de nombre Apache y la segunda servidor\_web.
2. Descomprime el archivo descargado y ejecútalo, sigue los pasos de la instalación y de los datos que te piden solo escoge el destino de la instalación, que será la carpeta que creaste en C:\Apache, los otros datos déjalos de la forma predeterminada para configurarlos más tarde.
3. El programa al instalarse crea un icono en el área de notificación que te permitirá: iniciar, detener y reiniciar Apache; tienes que tener en cuenta que cualquier cambio que hagas en el archivo de configuración no tendrá efecto hasta que reinicies el servidor.

¿Cómo configurar el Servidor Apache?

Toda la configuración para el funcionamiento de Apache se guarda en un archivo de texto nombrado: **httpd.conf** que se encuentra en la ruta *C:\Apache\conf*, lo podemos editar en cualquier editor de texto como el Bloc de notas pero un programa recomendado es *Notepad++*, software libre que es inmejorable.

Puedes descargar Notepad++ desde aquí.

Tienes dos opciones a continuación:

1. Primera opción, la más sencilla, descarga en el siguiente link una copia del archivo **httpd.conf**, descomprímelo, cópialo o muévelo a la carpeta C:\Apache\conf y sustituye el archivo original, ya tendrás listo para funcionar el servidor.
2. La otra opción, más avanzada pero no difícil, abre el archivo httpd.conf y edita manualmente las líneas que se indican:

*Todas las líneas que comienzan con el símbolo # son comentarios, explican en cada sección las distintas opciones pero se encuentran en inglés.*

La línea 52 **Listen** indica el puerto y dirección IP por el que el servidor va a recibir las peticiones, puedes usarla de las siguientes maneras:

1. El servidor va recibir peticiones solo de la misma PC: **Listen localhost:80**
2. Recibirá peticiones de otras máquinas en una red local: **Listen 80**

En la línea 149 **DocumentRoot** es necesario especificar la ruta de la carpeta local que contendrá las páginas y archivos a servir, en tu caso será la carpeta que creaste en C:/servidor\_web, quedaría de la siguiente forma:

**DocumentRoot "C:/servidor\_web"**

La línea 177 **<Directory>** establece los permisos necesarios al directorio anterior, quedaría:

**<Directory "C:/servidor\_web">**

Esta es la configuración con los parámetros esenciales para comenzar a utilizar Apache. Guarda los cambios realizados y reinicia el servidor dando clic en el icono del área de notificación.

También es posible configurar Apache usando un archivo de texto llamado .htaccess, en el que se agregan todos los parámetros necesarios. Es usado cuando hospedamos nuestros archivos en un servidor web en la red, en el que no tenemos acceso al archivo principal de configuración httpd.conf.

## **Como conectarse a internet a través de Apache desde tu PC**

Para que tu conexión a internet pase a través del servidor será necesario configurarlo como un proxy forward para eso en el archivo de configuración httpd.conf descomenta, (quitar el signo #) las siguientes líneas:

- LoadModule proxy\_module modules/mod\_proxy.so
- LoadModule proxy\_http\_module modules/mod\_proxy\_http.so
- Después agrega en el final del archivo la siguiente línea: ProxyRequests On

Cierra y guarda los cambios.

- Accede a las *Opciones de internet* mediante el Panel de Control.
- En la pestaña *Conexiones* pulsa el botón *Configuración de LAN* y marca la casilla *Usar un servidor Proxy*
- Escribe en *Dirección: 127.0.0.1* y en *Puerto: 80* o el que vayas a usar.
- Presiona *Aceptar* en todas las ventanas.
- Reinicia el servidor.

Lo anterior se aplica si usas el navegador Internet Explorer y Google Chrome, si usas Firefox las opciones anteriores tienes que ingresarlas en:

***Opciones >Configuración >Configurar como Firefox se conecta a Internet.***

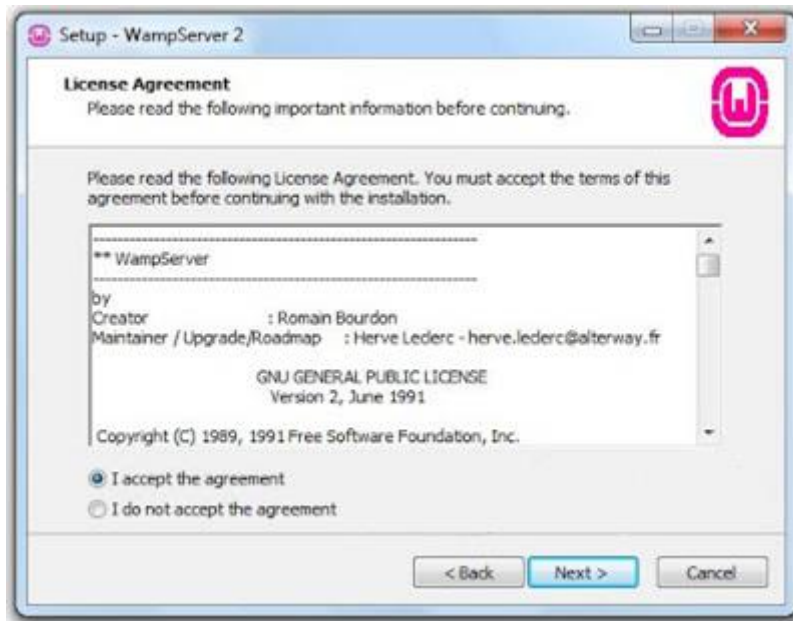
A partir de ahora toda tu conexión pasa a través de Apache, sea direcciones locales o externas.

1. Ejecutaremos el archivo WampServer2.1.1e-x64.exe que nos permitirá realizar la instalación del programa.

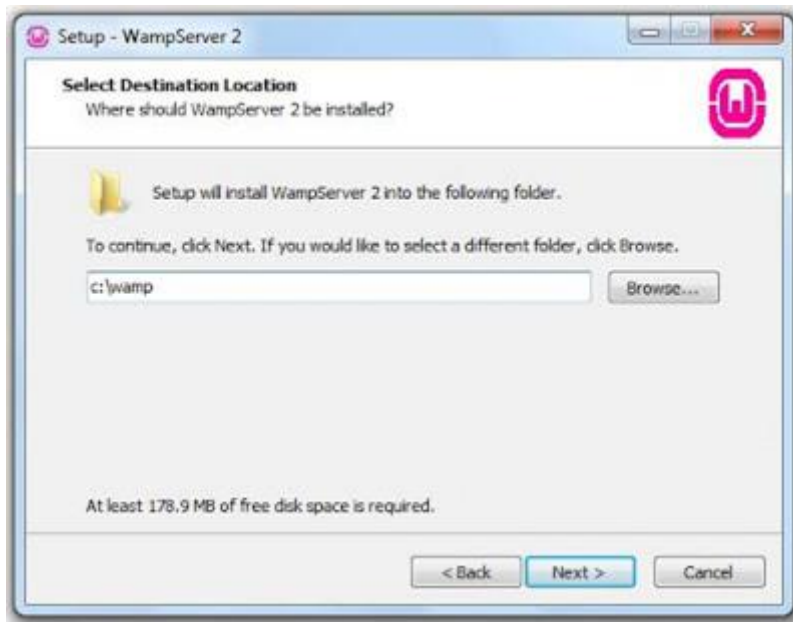


Como se puede observar en esta ventana nos muestra las versiones de los programas que se van a instalar, como es caso del Apache 2.2, MySQL 5.5, PHP 5.3, 5.7 que son los que maneja esta versión de WampSever.

2. Prosiguiendo con la instalación al dar siguiente se nos presentara la pantalla de común para aceptar las licencias respectivas del producto.

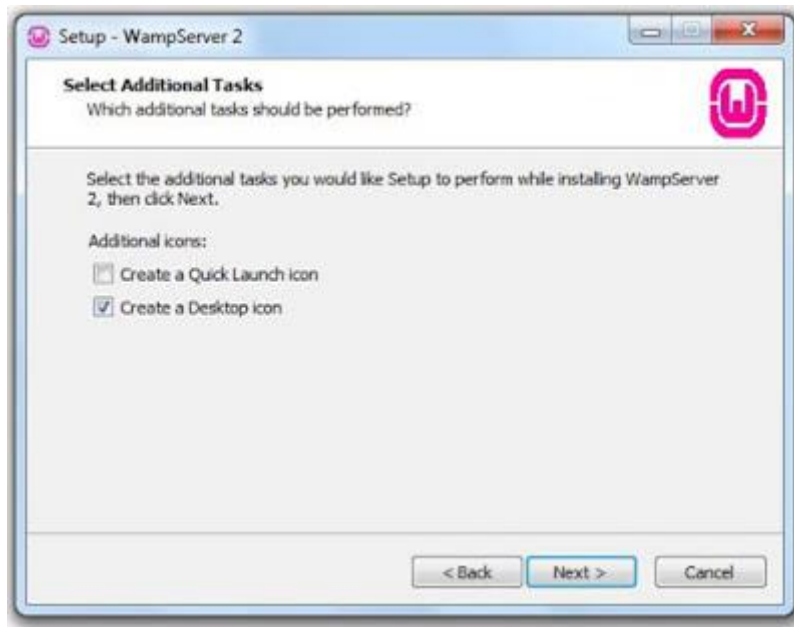


3. A continuación, se mostrará la carpeta donde se instalará el programa y el cual podremos acceder a cada uno de los recursos instalados por el mismo.

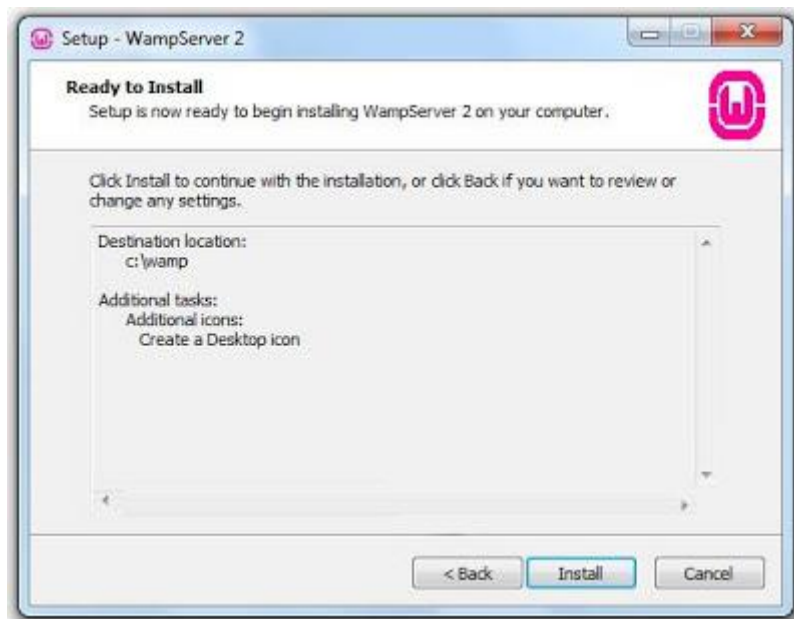


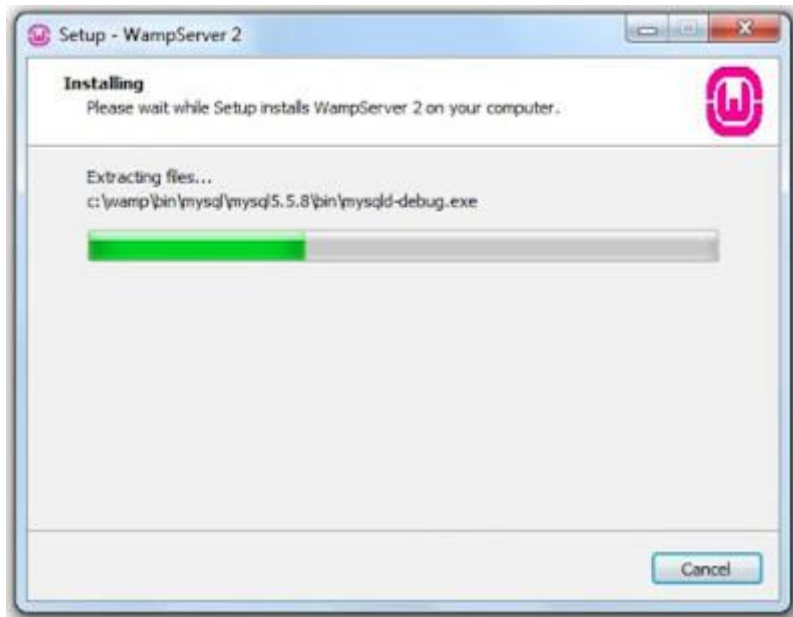
4. Luego no da la opción de crear un acceso directo en el escritorio, para ejecutar el WampServer cuando queramos utilizarlo.



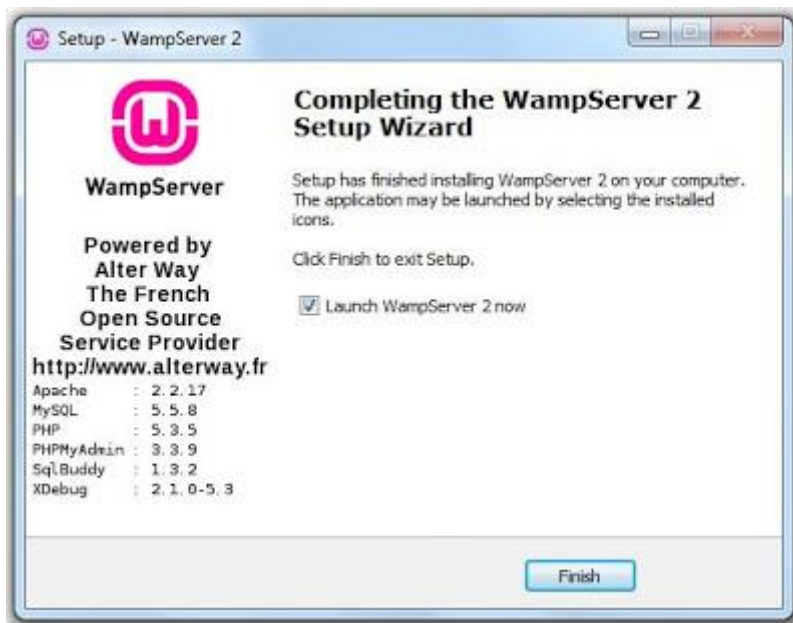


5. Por último, se realizará la instalación del WampServer el cual extraerá los archivos necesarios y los instalará, el proceso de instalación demorará de acuerdo con la capacidad de la computadora, pero de modo general se desarrolla de forma rápida.





En la parte final de la instalación veremos la siguiente ventana que nos informa que la instalación se ha completado, además de ofrecernos la opción de que se ejecute WampServer al finalizar la instalación.



## Ejecución y Configuración de WampServer

Una vez instalado el entorno de trabajo, vamos a ejecutarlo, podríamos haber utilizado la opción que nos da al finalizar la instalación o cuando queramos utilizarlo en otro momento debemos de ejecutarlo ya sea por el acceso directo creado en el escritorio o simplemente buscarlo dentro de los programas.

Como estamos desarrollando la instalación y es la primera vez que la vamos ejecutar debemos de fijarnos en pequeño detalle, una vez ejecutado el programa aparecerá en la parte inferior en la barra de tareas de nuestro sistema operativo (Windows) el icono de WampServer el cual debemos de tener en cuenta que este de color verde; haciendo un paréntesis puedes fijarte que cuando se ejecutó el programa este icono pasa por tres procesos: primero de color "rojo" que significa que se están cargando los servicios, luego pasa a un color "ámbar" que significa que el proceso de carga está siendo procesado y por último el color "verde" que nos asegura que la carga de los servicios fue exitosa.

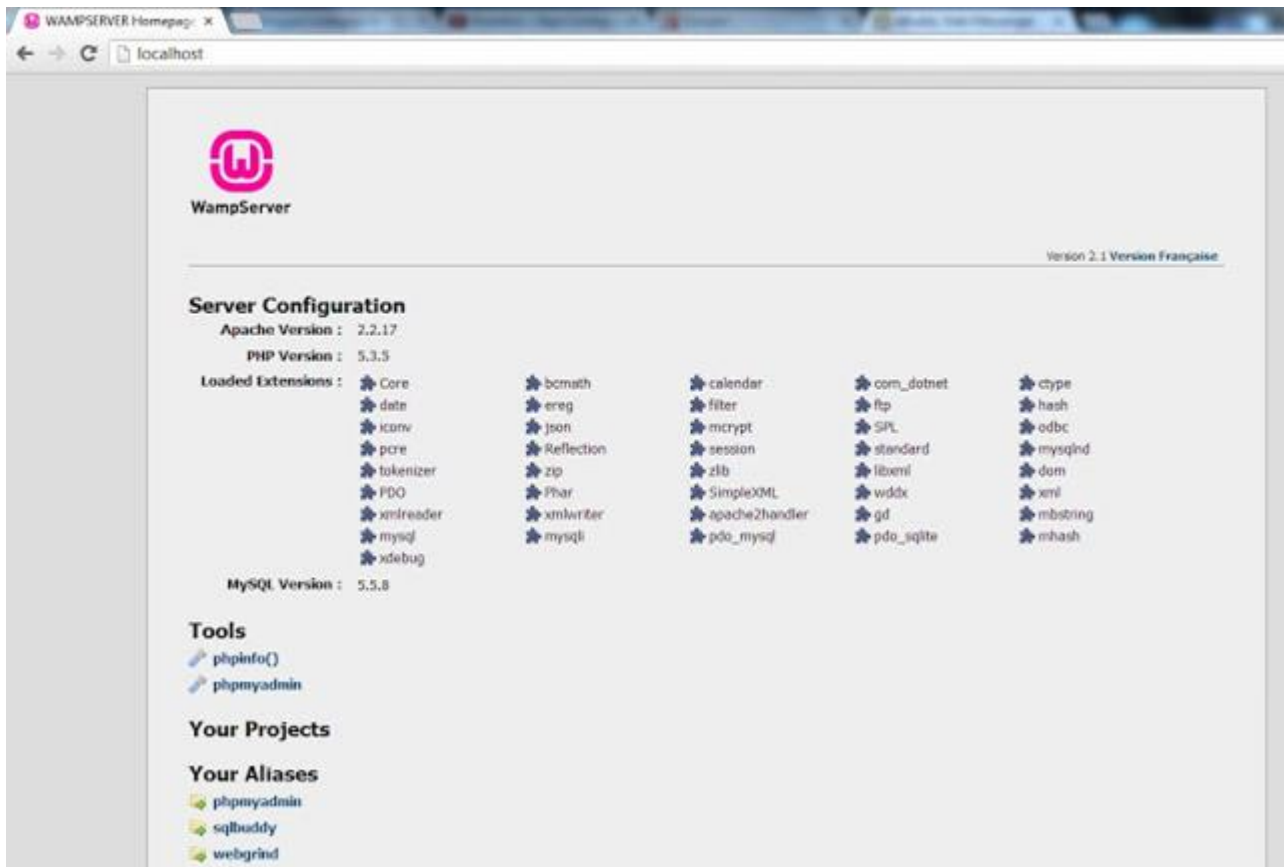


Supongamos que el icono no apareciera como se muestra en la imagen es probable que lo pueda encontrar dentro del panel de icono de notificación, en el cual si desea que este aparezca en la barra de tareas debe de personalizar la opción del icono cambiando en el comportamiento por "Mostrar iconos y notificaciones".

Ahora bien si el icono aparece de color "verde" esto indica que la ejecución y carga de los servicios fue exitosa, en caso de que se mantuviera ya sea con el color "ámbar" o "rojo", esto indica que se ha sucedido algún problema en la carga de los servicios.

Un problema común es el famoso conflicto entre el servidor Wamp que estamos instalando y predeterminado por el sistema operativo el IIS, para conocimiento el servidor web que por defecto viene instalado es el IIS quien ocupa el puerto 80 (Localhost), cuando instalamos el Wamp este también por defecto se aloja en el mismo puerto, ocasionando un conflicto que no nos permite utilizar el WampServer que acabamos de instalar.

En caso el IIS de su máquina no este ejecutado usted no tendrá problemas y observaremos la siguiente pantalla cuando escribamos "*Localhost*" en el navegador.



Esto significa que ya podemos trabajar con el servidor sin ningún inconveniente.

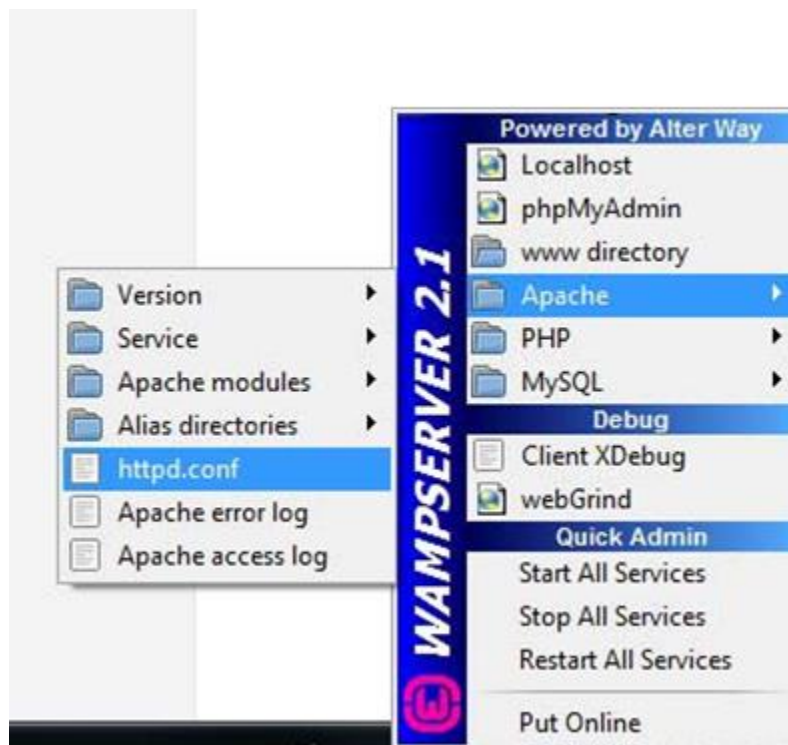
## Configuración de cambio de puerto

En caso contrario el icono no este de color "verde" y además al escribir "Localhost" en el navegador no podemos observar la página que se muestra en la imagen debemos de realizar una configuración para poder trabajar con WampServer.

De modo general cambiaremos el puerto donde trabaja el servidor WampServer que es por defecto es 80 por otro que para el caso será 50.

Pasos para la configuración de cambio de puerto:

1. Para cambiar el número del puerto debemos de cambiar la configuración del Apache, para realizarlo hacemos click sobre el icono que aparece en la barra de tareas y nos ubicamos en la carpeta Apache, dentro de ella abrimos el archivo de nombre "httpd.conf" tal como se muestra en la siguiente imagen.



- Una vez que hemos abierto el archivo que por defecto se abre en un bloc de notas, haciendo uso del menú Edición -> Buscar, hacemos la búsqueda respectiva ingresando el número 80 y cambiando en las siguientes líneas por el número 50, para que así se cambie el número de puerto de 80 por 50.

Primera Línea: Listen 50.

```
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#listen 12.34.56.78:80
Listen 50
```

Segunda Línea: Nombre del servidor "Servername localhost:50"

```
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
serverName localhost:50
```

- Después de hacer cambios y guardarlos debemos de reiniciar los servicios haciendo nuevamente click en el icono que aparece en la parte inferior en la barra de tareas y hacer click en "Restart All Services" con ello se reiniciarán los servicios y podemos observar que el icono cambia de color verificando que todo se realizó con éxito.



4. Por ultimo para verificar que la configuración se realizó correctamente abrimos el navegador, y escribimos "localhost:50" que como hemos visto en el cambio de puerto es la nueva ubicación del servidor WampServer y obtendremos la pantalla que se ha mostrado al inicio y podamos ya trabajar con el entorno de desarrollo.

## GLOSARIO

### A

**ABLEIP.** Sistema electrónico para clientes que requieren realizar solicitudes de registro de marca.

**AJAX.** Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

**ADDS-ON.** También conocidos como extensiones, plugins, snap-ins, etc, son programas que sólo funcionan anexados a otro y que sirven para incrementar o complementar sus funcionalidades. Las barras de Google y Yahoo! son add-ons para navegadores de internet y tienen opciones y funciones que incrementan las funciones del mismo.

**APACHE.** Es un servidor web **HTTP** de código abierto, para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

### C

**COOKIES.** Pequeños archivos usados por los sitios web para almacenar información en tu equipo (como información para iniciar sesión automáticamente y las preferencias de un sitio).

### D

**DOM.** Document Object Model (Modelo de Objetos del Documento), es esencialmente una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

### F

**FRAMEWORK.** Entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras

herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

## H

**HIPERTEXTO.** Es una herramienta con estructura no secuencial que permite crear, agregar, enlazar y compartir información de diversas fuentes por medio de enlaces asociativos. La forma más habitual de hipertexto en informática es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos.

**HTTP.** Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto), es el protocolo de comunicación que permite las transferencias de información en la **World Wide Web**. HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las **cookies**, que es información que un servidor puede almacenar en el sistema cliente.

## I

**IMPI.** Instituto Mexicano de la Propiedad Industrial, es el organismo estatal mexicano encargado de la protección de los derechos de propiedad industrial, a través del otorgamiento de títulos y/o efectuando los registros establecidos en la Legislación Nacional para tal fin.

## J

**JQUERY.** Es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol **DOM**, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

## O

**OPENSOURCE. Código abierto** es el software desarrollado y distribuido libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre. Para muchos el término “libre” hace referencia al hecho de adquirir un software de manera gratuita, pero más que eso, la libertad se refiere al poder modificar la fuente del programa sin restricciones de licencia, ya que muchas empresas de software encierran su código, ocultándolo, y restringiéndose los derechos a sí misma.

## P

**PLUGINS.** En informática, un complemento o plug-in es una aplicación (o programa informático) que se relaciona con otra para agregarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la interfaz de programación de aplicaciones. Complemento y plug-in se diferencian en que los plug-in son desarrollados por empresas reconocidas y tienen certificado de seguridad y los complementos pueden ser desarrollados por cualquiera.



## **S**

**SCRIPT.** Archivo de órdenes, archivo de procesamiento por lotes o, cada vez más aceptado en círculos profesionales y académicos, script es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

## **W**

**WWW.** Red informática mundial (WORLD WIDE WEB), es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener textos, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces.

## REFERENCIAS ELECTRÓNICAS

**Tema:** Filing of I.P Rights-Mexico Trademarks

**Liga:** <http://aldecoaelias.com/>

**Fecha de consulta:** 29-01-2016

**Tema:** Colección Guía de Usuarios (2017) IMPI

**Liga:** <http://www.gob.mx/impi/documentos/coleccion-guia-de-usuarios?idiom=es>

**Fecha de consulta:** 03-02-2016

**Tema:** PHP (1994-2017)

**Liga:** <http://php.net/manual/es/intro-what-is.php>

**Fecha de consulta:** 06-07-2017

**Tema:** MySQL (1995-2017)

**Liga:** <https://dev.mysql.com/doc/>

**Fecha de consulta:** 06-06-2017

**Tema:** Metodologías del desarrollo de software (2016)

**Liga:** <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>

**Fecha de consulta:** 15-03-2016

**Tema:** Qué es SCRUM (2013)

**Liga:** <https://proyectosagiles.org/que-es-scrum/>

**Fecha de consulta:** 09-05-2016

**Tema:** Why Use Scrum (2011)

**Liga:** <http://www.scrumalliance.org>

**Fecha de consulta:** 11-05-2016

## REFERENCIAS BIBLIOGRÁFICAS

**Título:** Métodos Ágiles y Scrum  
**Autores:** Alonso Álvarez García, Rafael de las Heras del Dedo, Carmen Lasa Gómez  
**Editor:** Anaya Multimedia-Anaya Interactiva, 2012  
**N.º de páginas:** 352 páginas  
**Fecha de consulta** 25-11-2016

**Título:** Tratado sobre Derecho de Marcas Tomo 1  
**Autores:** Carlos Fernández Novoa  
**Editor:** Marcial Pons  
**N.º de páginas:** 304 páginas  
**Fecha de consulta** 09-10-2016

**Título:** Tratado sobre Derecho de Marcas Tomo 2  
**Autores:** Carlos Fernández Novoa  
**Editor:** Marcial Pons  
**N.º de páginas:** 623 páginas  
**Fecha de consulta** 09-10-2016