

Universidad Central “Marta Abreu” de las Villas
Facultad Matemática Física y Computación
Ciencia de la Computación



Trabajo de Diploma

Aplicación Web para el Control de Almacén de la Cervecería Manacas

Autor:

Adrián Martínez Jacomino

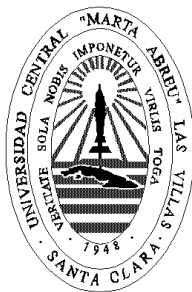
Tutores

Msc. Addel Arnaldo Goya Jorge

Lsc.Yaima López Galindo

Santa Clara

2017



Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Dpto

PENSAMIENTO

El Lenguaje ha de ser matemático, geométrico, escultórico.

La idea ha de encajar exactamente en la frase, tan exactamente que no pueda quitarse nada de la frase; sin quitar eso mismo de la idea.

José Martí

DEDICATORIA

AGRADECIMIENTOS

RESUMEN

La Empresa productora de Cerveza “Antonio Díaz Santana” de Manacas cuenta con un gran volumen de productos en su almacén de productos. Actualmente los operarios de dicho almacén disponen de un sistema manual para documentar sus acciones sobre los productos, lo cual dificulta la rapidez y eficacia del proceso. Por tanto, se deben establecer políticas que regulen la gestión de todos sus productos. Se desarrolla una aplicación Web siguiendo estándares de código abierto que permita operar de manera rápida y confiable el proceso de control de recursos y productos en el almacén de la cervecería.

ABSTRACT

The "Antonio Diaz Santana" Beer Production Company of Manacas has a large volume of products in its product warehouse. At the moment the workers of this warehouse have a manual system to document their actions on the products, which hinders the speed and efficiency of the process. Therefore, policies should be established to regulate the management of all its products. It develops a web application following open source standards that allows to operate quickly and reliably the process of control of resources and products in the warehouse of the brewery.

Tabla de Contenido

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	5
1.1 Descripción de la fábrica de Cervezas “Antonio Díaz Santana”	5
1.1.1 ¿Cómo se gestiona el almacén?	6
1.1.1.1 <i>Proceso de recepción</i>	8
1.1.1.2 <i>Tipos de almacenamiento</i>	8
1.1.1.3 <i>Control de caducidades que usan</i>	9
1.2 Entorno Web	10
1.2.1 Características	11
1.2.2 Lenguajes de Programación para Web	12
HTML	13
PHP	14
JavaScript	15
1.2.3. Servidores para Web.....	15
1.2.3.1 Apache	16
1.2.3.2 Gestores de Bases de Datos.....	16
1.3 Herramientas para el desarrollo de software	19
1.3.1 PhpStorm.....	20
1.3.2 Visual Paradigm.....	22
1.4 Frameworks de Trabajo	22
1.4.1 Zend Framework	23
1.4.2 CakePHP	23
1.4.3 Kumbia	23
1.4.4 Symfony 3.....	24
1.5. Buenas Razones para escoger Symfony 3	25
1.5.1 Doctrine.....	27
1.5.2 Symfony y el patrón de arquitectura MVC.....	28
CAPÍTULO 2. Diseño y Análisis de la Aplicación para el Control del Almacén de La Cervecería Manacas.	31
2.1. Diagrama de casos de uso del sistema	31
2.2. Modelo Entidad-Relación.....	38

2.2.1 Descripción de las Tablas	38
2.3. Diagramas de Actividades	41
2.4. Diagrama de Secuencia	42
2.5. Diagrama de Componentes.....	43
2.6. Diagrama de despliegue.....	44
2.7. Requisitos funcionales	45
2.8. Requisitos no funcionales	45
CAPÍTULO 3. PRUEBAS DE LA APLICACIÓN.	49
3.1 Evaluación de calidad web.	49
3.1.1 Importancia de hacer las pruebas de calidad.	49
3.1.1.1 Tipos de Pruebas de Software	50
3.1.2 Pruebas de calidad del Software realizadas al sistema.....	51
3.2 Pruebas de caja negra	53
3.2.1 Pruebas funcionales	¡Error! Marcador no definido.
CONCLUSIONES	56
RECOMENDACIONES	57
REFERENCIAS BIBLIOGRÁFICAS	58

Tabla de Figuras

<i>Figura 1.1: Mapa de la distribución del almacén.</i>	10
<i>Figura 1.2: Filosofía de trabajo de Symfony</i>	29
<i>Figura 2.1: Diagrama de casos de usos del sistema</i>	32
<i>Figura 2.2: Vista de los Productos existentes</i>	33
<i>Figura 2.3: Vista del Inicio del usuario administrador</i>	33
<i>Figura 2.4: Vista del Inicio del usuario administrador</i>	34
<i>Figura 2.5: Vista de los Usuarios existentes en el sitio</i>	35
<i>Figura 2.6: Vista de la página "Sobre Nosotros"</i>	36
<i>Figura 2.7: Vista de la galería</i>	36
<i>Figura 2.8: Vista de los Productos próximos a caducar</i>	37
<i>Figura 2.9: Vista del Inicio del usuario administrador</i>	37
<i>Figura 2.10: Diagrama Entidad-Relación</i>	40
<i>Figura 2.11: Diagrama de actividad correspondientes a Gestionar Productos y Gestionar Usuarios</i>	41
<i>Figura 2.12: Diagrama de Secuencia para Adicionar Usuario</i>	42
<i>Figura 2.13: Diagrama de Secuencia Mostrar Productos de una cuenta</i>	43
<i>Figura 2.14: Diagrama de Componentes</i>	44
<i>Figura 2.15: Diagrama de Despliegue</i>	45

Tabla de Tablas

<i>Tabla 1.1: Cuadro comparativo de marcos de trabajo en PHP</i>	25
<i>Tabla 2.1: Muestra la explicación del caso de uso "Gestionar Producto".</i>	32
<i>Tabla 2.2: Muestra la explicación del caso de uso "Gestionar Usuario".</i>	34
<i>Tabla 2.3: Muestra la explicación del caso de uso " Sobre Nosotros".</i>	35
<i>Tabla 2.4: Muestra la explicación del caso de uso " Listar Productos Próximos a caducar".</i>	37

INTRODUCCIÓN

INTRODUCCIÓN

Desde los tiempos más remotos la práctica ha demostrado la necesidad del uso de la automatización para el logro de un fin determinado. En el mundo moderno el desarrollo y empleo de las páginas web se ha convertido en el principal mecanismo de presentar información. Usando esta tecnología de forma que se sepa aprovechar correctamente es posible obtener ventajas con respecto a otras empresas o personas a la hora de mostrar determinadas estadísticas.

La organización puede ser vista desde dos ópticas distintas, la organización como una función general del ciclo administrativo de la dirección y la organización como sistema organizativo.

La dirección se relaciona también con todos los enfoques de organización que manejan o definen varios autores, pues al ser los sistemas organizativos sistemas creados por el hombre como lo son el departamento de recursos humanos, departamento de mantenimiento, una institución hospitalaria, diciéndole sistema más destacado a nivel jerárquico y organizativo (la organización de las fuerzas militares). Corresponde al hombre o mujer la creación consciente de los nexos, los flujos de información entre las diferentes subdivisiones estructurales, el establecimiento de la departamentalización, las diferentes formas de coordinación entre las partes, los niveles de subordinación, el establecimiento de las atribuciones, funciones, deberes, derechos, profesiogramas, niveles de competencia, áreas de responsabilidad, entre otros elementos propios de esta función general, destacándose el trabajo de los directivos en los diferentes niveles jerárquicos como otros especialistas en organización.

Un grupo de factores casuales respaldan que la organización vista como sistema organizativo se estudie hoy día con gran interés, profundidad y científicidad, y es por ello que la misma se puede considerar como un componente dominante de la sociedad contemporánea, puesto que está diseñada para el logro de objetivos y resultados dejando muy claramente que estas son precursoras al surgimiento de la administración. Y esto nos lleva a explicar la organización como sistema, si se quiere decir constituyen un conjunto de elementos dinámicamente relacionados que forman una actividad con el único fin de alcanzar un objetivo, pero siempre operando sobre datos, energía y materia, para proveer a su

organizaciones o medios entre otros de información, energía y materia (procesada) para el mejor beneficio por parte de la organización o de agentes externos de la misma si se pretende exportar los resultados.

El almacén de la Empresa Cervecera de Manacas, hoy presenta un grupo de deficiencias de carácter organizacional (estructural) que afectan con mucha incidencia la eficiencia, la eficacia y el alto desempeño de la organización. Una mejor administración de los recursos permite tener parámetros para evaluar cómo se está realizando el trabajo, por tanto, la actividad de dirección resulta más efectiva y completa. Muchas entidades todavía no comprenden la necesidad de una automatización en algunas de sus áreas, y sólo la ven como el simple hecho de plasmar miles de movimientos de recursos en un papel y no en un proceso computarizado y ordenado; donde lo estratégico no sólo viene dado por el tiempo, sino que incluye los cambios y transformaciones que deben lograrse con cualquier acción que se realice, tanto en términos cuantitativos, como cualitativos. Es decir que es capaz de cambiar por completo la situación de una institución en muchos aspectos, además de proporcionarle una base para el enfrentamiento de los diferentes cambios que pueden ocurrir en un entorno, que hoy más que nunca es cambiante y violento. La empresa cervecera “Antonio Díaz Santana” no está exenta a los cambios del entorno, por lo que se hace necesario realizar esta aplicación que plasme una mejora en los objetivos de la organización de su Almacén.

Por ello, en tal sentido y teniendo en cuenta la situación económica actual de Cuba, donde todos estamos inmersos en accionar y ofrecer verdaderos aportes en aras de actualizar el modelo económico de la nación, es que el presente proyecto asume como interrogante ¿Cómo mejorar la organización del almacén de la Empresa Cervecera de Manacas en función de guiar la estrategia y alcanzar el alto desempeño organizacional?, constituyendo esta interrogante un problema a resolver.

Problema de Investigación

Como realizar una mejor gestión y control de los recursos en el almacén de la cervecería “Antonio Díaz Santana” de manera que contribuya a un mejor desempeño de la empresa.

Objetivo General

Desarrollar un sistema siguiendo estándares de código abierto para el desarrollo de aplicaciones Web que facilite la gestión eficiente de los recursos para el almacén de la cervecería “Antonio Díaz Santana”.

Objetivos Específicos

1. Describir el caso de estudio para el proceso de control de almacén de la fábrica.
2. Diseñar e implementar una base de datos relacional que permita al departamento administrativo gestionar la información para el control del almacén.
3. Desarrollar una aplicación Web que permita gestionar los productos y materias primas existentes en el almacén.
4. Validar el sistema propuesto mediante la creación de casos de pruebas de caja blanca y caja negra.

Preguntas de investigación

- ¿Cómo realizar un análisis situacional de la organización?
- ¿En qué medida la revisión y búsqueda bibliográfica permite desarrollar la construcción de la aplicación que se propone?
- ¿Sobre qué modelo se define la estrategia de la empresa?
- ¿Cuáles serán los beneficios de realizar una aplicación para controlar el almacén en la empresa?

Justificación

Constituye una labor estresante e improductiva controlar de manera manual, todos los suministros, materias primas y productos ya sean terminados o ingresados en la empresa. Este problema se puede corregir mediante la implementación de una página Web que facilite, a la vez que organice y optimice el proceso de control dentro de dicha empresa.

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA DE
LA INVESTIGACIÓN

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

En el capítulo a continuación se analizarán las características que tiene la fábrica en cuanto a la forma y requisitos que deben tenerse para un buen funcionamiento de su almacén. También se abordan las principales dificultades presentadas por el actual sistema de control de almacenes que opera en la Fábrica “Antonio Díaz Santana”. Es objetivo también del capítulo, realizar un análisis del entorno Web y las tecnologías existentes que pueden apoyar el desarrollo de una aplicación Web, teniendo en cuenta que, no existen muchas herramientas gratuitas efectivas en internet que permitan realizar estudios de usabilidad a cualquier página Web que se necesite. Las herramientas que resultan efectivas necesitan conexión a internet permanente, por lo que en Cuba se hace difícil su utilización.

1.1 Descripción de la fábrica de Cervezas “Antonio Díaz Santana”

La empresa está ubicada en Manacas, municipio Santo Domingo. Una fábrica dedicada a la producción de cervezas y maltinas. La firma Bacardí S.A dedicada a la producción de bebidas deseaba extender sus redes de beneficio económico pues la edificación de una fábrica cervecera en el centro del país le proporcionaba, menos costo de distribución, ya que a la provincias Las Villas y Camagüey el producto (cerveza y maltita) era enviado desde la Habana y Santiago de Cuba, además mayor propaganda en las provincias centrales para un aumento en las ventas, en este caso de la Cerveza Hatuey sobre la Polar y la y una mano de obra barata puesto que el salario fijado estaba por debajo del que devengaban los obreros de las demás cervecerías. La vieja demanda de los obreros se materializa, al equipo de sus salarios con el de los trabajadores de las demás Cervecerías Cubanas. Para poder alcanzar estos logros ha sido necesario un esfuerzo titánico de los trabajadores cerveceros, fundamentalmente de los que trabajan en el área de mantenimiento, puesto que la mayoría de los equipos son de origen norteamericano y el férreo bloqueo impuesto no permiten la entrada de materiales y piezas de repuesto para la reparación y mantenimiento de los mismos. Se señala también, que los trabajadores de la cervecería de Manacas han apoyado todas las medidas del gobierno revolucionario participando activamente en la fundación del PCC y las Milicias Nacionales Revolucionarias, en la limpia del Escambray, en la movilización de la

crisis de octubre, en numerosas movilizaciones agrícolas, manteniéndose una brigada en la zafra azucarera, y en cumplimiento de misiones internacionalistas. Reseña histórica de la Cervecería Manacas En 1998 se comenzó la inversión en una Línea de Cerveza Dispensada, la cual ya está en funcionamiento. Con la inversión de más de 2 000 000 USD los sistemas fundamentales de la fábrica quedan listos para enfrentar mayores niveles de producción con la máxima calidad, lo que permitirá a la Empresa introducirse definitivamente en el mercado turístico y poder reinvertir en la ampliación y modernización de la planta .La Fábrica distribuye cerveza embotellada a Villa Clara, Matanzas, Cienfuegos, Sancti Spíritus, y La Habana, en tanto la dispensada llega a la Atenas de Cuba, la Perla del Sur, y el territorio anfitrión, al igual que la de a granel, mientras los caldos o sustento de la cerveza tocan a la puerta de Pinar del Río y El Cotorro. El plan anual asciende a 4 millones 200 mil cajas de cerveza; de ellas, 975 mil son embotelladas. Se pronostica un incumplimiento de 250 mil cajas debido a las irregularidades tecnológicas. En el caso de la dispensada cumplirán con 730 mil cajas. También augura buenos saldos la de a granel, no así los caldos que ya avizora incumplimientos motivado por dificultades de transportación y otras circunstancias de las embotelladoras que reciben el producto. En el área de fermentación existe un equipo original que data de 1953. Aquí se añaden los caldos puros de levadura que se convierten lentamente en cerveza.

1.1.1 ¿Cómo se gestiona el almacén?

Según (FIAEP 2014) es un hecho que el desorden en almacén o en el área de trabajo de sus operarios provoca graves pérdidas a la empresa. Por ejemplo, desconocer que tenemos la existencia de algún producto en almacén conlleva a realizar compras demás. Se debe tener en cuenta que, dentro de los costos logísticos de una empresa, el almacenamiento y manipulación de los productos representa casi un 30% aproximadamente. Es fácil comprender cómo la figura del Jefe del Almacén, pasa de ser un simple custodio de los productos, a proyectarse a la de un manager, capaz de conseguir la máxima eficiencia y eficacia tanto de los recursos como de infraestructura y tecnológicos que tiene a disposición. Actualmente mediante sistemas manuales se lleva todo el control del almacén, cosa que se quisiera automatizar, para así facilitar la rapidez de la localización e identificación de los productos, como también permitir asignar al artículo un número de ubicación en el almacén,

que le facilita al operario la identificación, localización y control del producto recogido, ya sea el almacenamiento o movimientos dentro de un mismo recinto, de materiales, materias primas y productos semielaborados, hasta el punto de consumo por un cliente externo o interno.

La empresa realiza un análisis de inventario. Se refiere al análisis estadístico realizado para establecer si las existencias que fueron previamente determinadas son las que se deberían tener en la planta, es decir aplicar el dicho "nada sobra y nada falta", pensando siempre en la rentabilidad que pueden producir estas existencias. Principios de posición fija es una metodología de la empresa para el almacén de materiales. Esto trae un aprovechamiento adecuado del espacio físico, mayor rapidez de operación en los procesos de entrega, control eficiente y expedito del manejo de materiales, reducción de costos de almacenamiento en el sistema de posición fija. Cada producto ocupa una posición permanente dentro del Área asignada en el Almacén, por lo cual existe una posición biunívoca entre espacio disponible y producto almacenado, de tal manera que cuando no hay entrada, el espacio queda vacío pero reservado para el producto asignado. El almacén responde fundamentalmente al requerimiento de espacio adecuado, dimensionamiento claro ubicación y manipulación eficiente de materiales y mercancías, utilización máxima del volumen disponible (mts2-mts3), costos operacionales mínimos, maquinaria y equipo adecuado, talento Humano capacitado y estandarización de los procedimientos de trabajo. Para gestionar el almacén se lleva a cabo el siguiente proceso _

- 1) Planificación y Organización
- 2) Recepción de materiales
- 3) Almacenamiento
- 4) Movimientos internos
- 5) Despacho de materiales.

El almacenamiento de materiales depende de la dimensión y características de los materiales. Estos pueden requerir una simple estantería hasta sistemas complicados, que involucran

grandes inversiones y tecnologías complejas. La elección del sistema de almacenamiento de materiales depende de los siguientes factores:

1. Espacio disponible para el almacenamiento de los materiales.
2. Tipos de materiales que serán almacenados.
3. Número de artículos guardados.
4. Velocidad de atención necesaria.

Una gran cantidad de organizaciones tienen en sus almacenes una gran cantidad de artículos que no tienen una misma característica, muchos de estos artículos son relativamente de bajo costo, en tanto que otros son bastante costosos y representan gran parte de la inversión de la empresa. Algunos de los artículos del inventario, aunque no son especialmente costosos tienen una rotación baja y en consecuencia exigen una inversión considerable; otros artículos, aunque tienen un costo alto por unidad, rotan con suficiente rapidez para que la inversión necesaria sea relativamente baja.

1.1.1.1 Proceso de recepción

Es el registro de mercancías que ingresan a la empresa, ya sean procedentes de zonas externas (proveedores) como internas (empresa) es necesario: - Recepcionar y verificar mercancías. – Revisión técnica con el solicitante de la compra. - Generar los documentos pertinentes (guías internas). - Generar nuevos códigos cuando sea necesario. - Rechazar inconsistencias. - Informar del proceso. - Ajustarse al calendario de recepciones. La calidad del producto final depende en gran medida del proceso de recepción (FIAEP 2014).

1.1.1.2 Tipos de almacenamiento

De Temporada o Promocionales: Los productos sujetos a estacionalidades son ubicados en áreas de fácil recogida y abastecimiento para minimizar los costos de manipulación. Cuarentena de Alto Riesgo: Existencias de alto valor, tóxicas, sustancias químicas peligrosas, etc, requieren condiciones especiales de almacenamiento, incluyendo acceso restringido. Mucho control y supervisión. De Temperatura Controlada: Es necesario tener en cuenta la seguridad de los operarios; deben protegerse de los cambios bruscos de temperatura. La

manipulación puede ser también más lenta debido al tiempo limitado que se puede pasar en estos ambientes.

1.1.1.3 Control de caducidades que usan

Existen opiniones encontradas al respecto de comprar por volumen o comprar la materia prima conforme se vaya necesitando. La experiencia ha mostrado que, dadas las circunstancias de cercanía y conveniencia, es mejor no comprar material perecedero por volumen sino hacer que el proveedor entregue sus productos en pequeñas remesas o comprarlo y escogerlo nosotros mismos. Esto permitirá contar siempre con materia prima de óptima calidad y la preferencia del cliente.

Por otro lado, la compra por volumen abre la posibilidad de obtener mejores precios por nuestros productos. En sí, lo recomendable es comprar por volumen los productos con caducidad mayor a 5 días. Los más frescos como la carne, el pollo, cerdo, pescados, mariscos y las verduras de poca duración en refrigerador se deberían comprar diariamente, dependiendo de la necesidad de la empresa y de la motivación del empresario para llevar a cabo todas las acciones tendientes a ofrecer productos de la máxima calidad. Las caducidades deben revisarse desde el momento que se reciba la mercancía (FIAEP 2014).

Consultar con el proveedor el modo de leer su fecha de fabricación y caducidad, esto es en casos específicos como la cerveza, refresco, y de algunos otros productos si es el caso. Se llevará un control de caducidades mediante un formato el cual será revisado semanalmente por el encargado de almacén y verificado por la gerencia.

La mercancía con caducidad más próxima es la primera a la que se le dará salida y será dada de baja en el control de caducidades una vez se haya transferido a algún departamento.

Revisar el control de caducidades de las demás áreas, presionando en que se lleve semanalmente y haciéndoles selectivos para verificar que todo esté en orden.

El acomodo de la mercancía perecedera y con fecha de caducidad debe ser conforme a esta, los productos más a la mano serán los que tengan la caducidad más próxima, para el momento de hacer transferencias dar la mercancía con la menor fecha de caducidad.

Por último, se muestra un mapa de la forma que se distribuyen los productos en nuestro almacén.

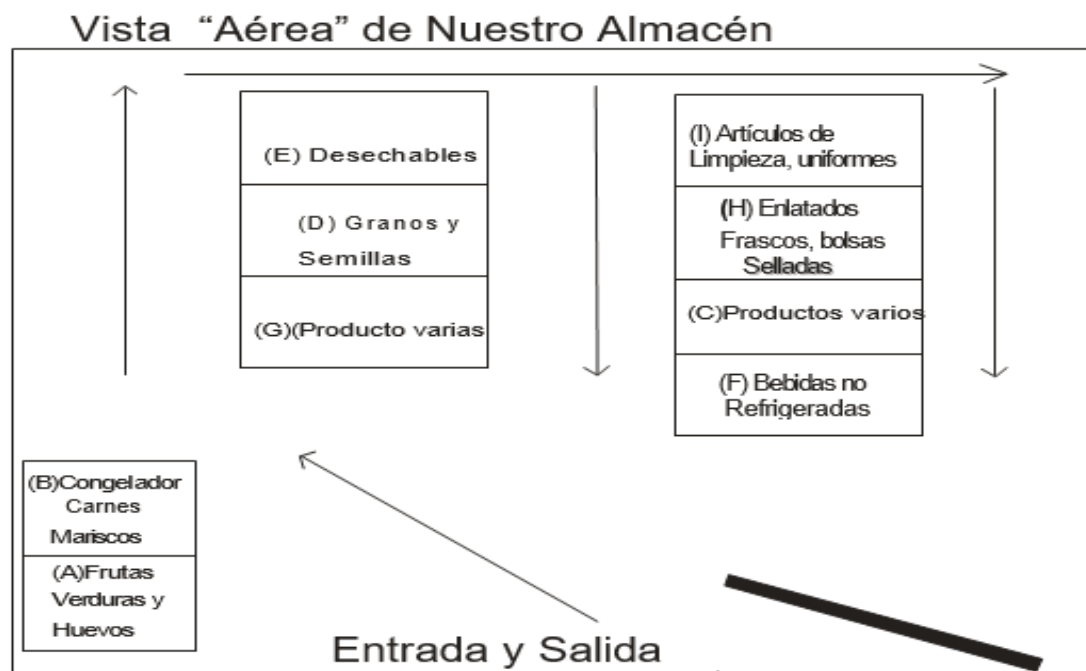


Figura 1.1: Mapa de la distribución del almacén.

1.2 Entorno Web

Una página Web es un sistema de documentos enlazados y accesibles a través de Internet, que pueden contener texto, imágenes, vídeos, etc. Un sitio Web es un grupo de páginas Web, generalmente comunes a un dominio de Internet. Estas son accedidas desde una URL (Uniform Resource Locator), raíz común llamada portada, que es una secuencia de caracteres, de acuerdo a un formato estándar, usado para nombrar recursos como documentos e imágenes, por su localización. Esa dirección es única para cada uno de los recursos de información disponibles en Internet.

Según (YUEN 2003) una aplicación es una capa de presentación elaborada para ser consumida desde un navegador Web, que permite ser usada en cualquier sistema operativo, convirtiéndose en una aplicación multiplataforma que puede ser visualizada por cualquier

número de computadoras a la vez y de forma independiente. Trae consigo, además, facilidad de uso, versatilidad y velocidad de interfaz.

La capacidad para actualizar y mantener aplicaciones Web sin distribuir ni instalar software en las computadoras de los clientes es la razón clave de su popularidad que provoca a su vez, que la mayoría de los clientes no necesiten entrenamiento adicional para manipular la interfaz de la aplicación. Además, alojando la aplicación en un servidor Web, se logra una disponibilidad total del producto, así como su código fuente.

1.2.1 Características

El éxito de un sitio Web depende de varias características:

- **Navegación:** Es la facilidad con la que un visitante accede y encuentra el contenido o servicio que sea de su interés.
- **Estética:** Comprende los aspectos de apariencia y composición de las páginas Web.
- **Búsqueda:** Es la capacidad de localizar o ubicar el contenido de interés para los usuarios de acuerdo con sus propios criterios.
- **Contenido:** Es la calidad y cantidad de información de utilidad para el visitante, por tanto, este renglón debe prevalecer. Un sitio debe presentar buenos contenidos que satisfagan las necesidades de los usuarios.
- **Eficiencia en el uso:** ¿Qué tan rápido realiza el usuario sus objetivos?
- **Fácil de recordar:** Una vez que el usuario ha utilizado el sistema, ¿Qué tanto recuerda del mismo?

La facilidad de uso de un entorno Web está completamente relacionada con el diseño del mismo.

El sitio debe estar construido de forma tal que los usuarios sepan qué hacer en cualquier momento y cómo hacerlo, facilitándoles el aprendizaje de sus características en el menor

tiempo posible. Inmediatamente al acceder al sitio, el visitante debe conocer los objetivos y alcance del mismo.

Suelen ayudar una distribución y diseño gráfico adecuados, así como un lenguaje claro e legible lo que influye en el usuario positivamente, promoviéndolo a permanecer en el entorno Web e incitándolo a volver a acceder al mismo. En (Keeker, 2006) se hace referencia a lo importante que resulta que los mejores contenidos o los que puedan despertar mayor interés, queden expuestos de manera visible para que el visitante conozca de su presencia rápidamente.

Diseñar la estructura de la página principal diferenciando visualmente áreas, y agrupar la información por tipo, funcionalidad y prioridad, es también de vital importancia, así como garantizar la claridad, fidelidad y precisión de la información. Teniendo en cuenta estos aspectos se puede lograr la construcción de una aplicación Web balanceada y atractiva, que facilite a los usuarios la navegación y les conduzca a la satisfacción de alcanzar sus objetivos correctamente.

1.2.2 Lenguajes de Programación para Web

Existen disímiles de lenguajes de programación para desarrollar una Web. Estos han ido surgiendo debido a las tendencias y necesidades de los desarrolladores. Desde los inicios de Internet, fueron apareciendo demandas de los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación dinámicos para la Web, que permitieran interactuar con los usuarios y utilizar sistemas de bases de datos.

Estos Lenguajes de programación se pueden dividir en dos grupos:

- Lenguajes del lado del servidor o lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Entre estos se encuentran: PERL, ASP y PHP.
- Lenguajes del lado del cliente (como HTML y Java, JavaScript y CSS, los cuales son incluidos en el código HTML) que representan aquellos lenguajes que pueden

ser directamente interpretados por el navegador y no necesitan un tratamiento previo.

HTML

Desde el surgimiento de Internet se han publicado sitios web gracias al lenguaje HTML (Hyper Text Markup Language) desarrollado por World Wide Web Consortium (W3C). Este es un lenguaje estático para el desarrollo de sitios web. Los archivos pueden tener las extensiones .htm, .html, son pequeños y de despliegue rápido. Es de fácil aprendizaje y lo admiten todos los exploradores como Internet Explorer, Firefox, Safari, Opera o Netscape.

Además, es muy sencillo, está basado en el uso de etiquetas, consistentes en un texto ASCII encerrado dentro de un par de paréntesis angulares (<...>). Donde el manejo de estas permite definir la estructura lógica del documento HTML, aplicar distintos estilos al texto, la inclusión de hiperenlaces, imágenes y ficheros multimedia. También al disponer de múltiples herramientas permite acceder a bases de datos de forma sencilla, es multiplataforma pues funciona tanto para Unix como para Windows (ECHEVERRIA 1995).

Adicionalmente se ha definido un lenguaje de hojas de estilo CSS (Cascading Style Sheets) creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Además, es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

De acuerdo a (EGUÍLUZ PÉREZ 2008) separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de

letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

PHP

Según se puede apreciar en (PHP 2007) se habla de PHP (Hypertext Preprocessor) como un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas.

Entre sus ventajas se encuentran que:

- Es un lenguaje multiplataforma.
- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos
- Contiene una biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable.

Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de arquitectura Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes (SÆTHER BAKKEN 2001)

JavaScript

Javascript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios Web. No requiere de compilación ya que el lenguaje funciona del lado del cliente. Los navegadores son los encargados de interpretar estos códigos. Es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página Web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript se pueden crear diferentes efectos e interactuar con los usuarios.

En (CHAFFER 2007) se enumeran las características de este lenguaje, entre ellas se pueden citar:

- Es un lenguaje basado en acciones que posee menos restricciones.
- Gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del ratón, aperturas, utilización de teclas, cargas de páginas, entre otros.
- Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, y Mozilla Firefox.

1.2.3. Servidores para Web

Un servidor Web es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales, síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del

cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI (Open System Interconnection).

1.2.3.1 Apache

Uno de los servidores de aplicación Web existentes es Apache. Debido a las características y ventajas que ofrece, ha sido el seleccionado para el desarrollo de la presente aplicación; además de ser un requisito del cliente y porque forma parte de la política de informatización de esta organización. A continuación, se hace una descripción de este.

Apache es un servidor Web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1. Se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation (APACHE, 2012).

Según (Ruiz, 2007), Apache es el servidor web más difundido y utilizado en Internet. Tal vez lo que lo hace más atractivo es su alta estabilidad, seguridad y facilidad de expansión. Debido a que es software libre, quienquiera que desee descargarlo no tendrá que pagar costosas licencias para su utilización, pudiendo disponer de su código fuente para modificar y adaptar el servidor a sus necesidades.

Entre sus ventajas más específicas se encuentran que:

- Es modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Código abierto.
- Multiplataforma.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

1.2.3.2 Gestores de Bases de Datos

Un gestor de base de datos o Sistema de Gestión de Base de Datos (SGBD) es un software que permite administrar la información de las bases de datos. 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; por tanto, cualquier operación que el usuario hace contra la base de datos está controlada por el gestor.

Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo.

Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server, etc. Entre sus características generales se encuentran:

- Aceptan definiciones de esquemas y vistas (definición de diferentes bases de datos).
- Manipulan los datos siguiendo las órdenes de los usuarios.
- Cuidan que se respete la seguridad e integridad de los datos.
- Permiten definir usuarios y las restricciones de acceso para cada uno de ellos.
- Controlan la concurrencia y las operaciones asociadas a la recuperación de los fallos.

Microsoft SQL Server 2000

De acuerdo a (MICROSOFT 2011) Microsoft SQL Server 2000 es parte de la familia de productos diseñados para mantener almacenar y procesar datos para sistemas nativos de Windows y sitios Web, puede ser usado por grandes compañía. Provee servicios fundamentales como el tradicional cliente/servidor, el cual alcanza un correcto desempeño y es altamente configurable en cuanto a la ingeniería de las bases de datos relacionales.

Algunas de las características de este SGBD:

1. Soporte de transacciones.
2. Soporta procedimientos almacenados.
3. Incluye un potente entorno gráfico que permite el uso de comandos de lenguajes de definición y manipulación de datos.

4. Permite administrar información de otros servidores de datos.

MySQL (MySQL 2008) es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que, de otra manera, se vulneraría la licencia GPL. Un lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas Web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

Para la implementación del BD se utiliza MySQL el cual, combinado con PHP, es el lenguaje estándar a la hora de crear sitios de comercio electrónico o páginas web dinámicas. Este producto puede funcionar en múltiples plataformas de hardware, con requerimientos relativamente pequeños, sin perder estabilidad. MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones en todo el mundo, lo que supera la base instalada de cualquier otra herramienta de bases de datos. El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible. Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia

Ventajas y desventajas de MySQL

Ventajas

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

1.3 Herramientas para el desarrollo de software

Las herramientas de desarrollo de software, consolidadas en la literatura en la forma de Ingeniería de Software asistida por computadora (Computer Aided Software Engineering), suponen la asistencia en el proceso de ciclo de vida del software, contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información y se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta se consigue agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.

- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes de software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos (Rivas et al., 2006).

1.3.1 PhpStorm

Sin duda PhpStorm es el mejor editor de texto para programar en PHP: Resalta los nombres de las funciones y clases, identifica variables, encuentra posibles errores, hace refactoring de PHP que de por sí ya es todo un logro y tiene cientos de combinaciones de teclas que permiten programar sin casi tocar el mouse y si eres programador sabes que eso es más que deseable. Además, soporta JavaScript y HTML como los mejores editores.

En el caso de PhpStorm, éste trae soporte para muchos frameworks PHP como Symfony y Laravel, a través de plugins.

Este IDE puede integrarse con Git, GitHub y Subversión, compilar preprocesadores como LESS, SaSS, Stylus y CoffeeScript en tiempo real, etc.

Si utilizas Grunt, el IDE detecta de forma automática que lo estás usando y te añade al proyecto una consola (aparte de la terminal) dedicada a mostrarte las tareas que tienes implementadas en el Gruntfile.js.

Algunas funcionalidades

La capacidad de no permitir indexar algunas carpetas en el proyecto.

Refactorización

Crear scopes

Debug en servidores remotos

Selección de múltiples ocurrencias

En los IDEs la combinación correcta es Alt + J, pero trae una combinación adicional que permite seleccionar todas las ocurrencias de una vez: Alt + Ctrl + Shift + J.

Otro uso muy útil que aparece, es que puedes regresar hacia atrás las ocurrencias que ya seleccionaste, con la combinación: Alt + Shift + J.

Inspector de bases de datos

PhpStorm tiene prácticamente un completo Editor de Bases de Datos, y para el caso de Postgres, es mucho más rápido e intuitivo que el PgAdmin.

Directamente desde el IDE te puedes conectar a un servidor de BD, ver todas las que están disponibles, mostrar los datos de sus tablas, modificar tuplas de forma visual, realizar consultas y con rapidez.

Este es el soporte que tiene PHPStorm en cuanto a bases de datos:

Postgres

MySQL

Oracle

SQLite

DB2

SQLServer

SysBase

H2

Derby

HSQLDB

Para poder comunicarse con el servidor, el IDE necesita el driver del tipo de sistema de bases de datos correspondiente.

Para el trabajo diario en el proyecto, donde se interactúa con muchos ficheros, que están versionados, en ocasiones desorganizados, donde el proyecto está vinculado a bases de datos o algún framework en específico, se debe utilizar PhpStorm.

1.3.2 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Esta herramienta posee diferentes características entre las que se encuentran: Soporte de UML versión 2.1, ingeniería inversa, generación de código, soporte ORM (generación de objetos Java desde la base de datos), transformación de diagramas de Entidad-Relación en tablas de base de datos, integración IDE con Visual Studio, NetBeans y otros.

1.4 Frameworks de Trabajo

Se han realizado importantes progresos orientados a la reusabilidad del software a través de la aplicación del paradigma de la programación orientada a objetos y mediante el uso de los componentes tecnológicos. Sin embargo, estas tecnologías solo proveen el re-uso en el nivel individual, frecuentemente a menor escala. Con la aparición de los patrones de diseño se ha demostrado la reutilización de soluciones para resolver problemas de escala mayor enfocados en la solución de problemas sencillos. Sin embargo, el más complejo problema de la reutilización de soluciones es en el nivel de los grandes componentes, para que estos se puedan adaptar para las solicitudes individuales. Recientemente, el interés por la reutilización de software ha pasado de la reutilización de los componentes de forma independiente a la de todo el diseño de un sistema o la estructura de una aplicación. Un sistema de software que es capaz de permitir la reusabilidad a este nivel, permitiendo la realización de aplicaciones completas es llamado Marco de Trabajo (framework).

Un framework debe estar basado en la idea de facilitar la producción de un conjunto de sistemas específicos pero similares, dentro de una estructura genérica. Además, los marcos de trabajo pueden contener sub marcos de trabajo los cuales representen subconjuntos de componentes de un sistema más grande. Está formado por componentes personalizables e intercambiables para el desarrollo de una aplicación.

En la actualidad existen una gran cantidad de frameworks para el desarrollo de soluciones web, entre ellos cabe mencionar por su gran relevancia a CakePHP, Zend, Kumbia, Symfony, entre otros, se realizará una pequeña síntesis de cada uno de ellos, y luego en cuanto a sus características completaremos un cuadro comparativo con sus diferencias y similitudes.

1.4.1 Zend Framework

Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos.

La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. Facilita la programación a través del patrón MVC y automatiza tareas más específicas, como el acceso a base de datos, el filtrado de datos ingresados a la aplicación o la búsqueda en un sitio web ordenando los resultados por relevancia (CHASE 2006).

1.4.2 CakePHP

CakePHP es un framework de desarrollo rápido de aplicaciones de código abierto en PHP.

Inspirado en Rails, para la construcción de sitios web que utilizan una base de datos como fuente de recursos, posee una infraestructura que tiene como finalidad permitir el desarrollo de aplicaciones web de manera ágil y estructurada, sin perder flexibilidad (ANDERSON 2006).

1.4.3 Kumbia

Kumbia es un framework para aplicaciones web libre escrito en PHP5. Kumbia fomenta la velocidad y eficiencia en la creación y mantenimiento de aplicaciones web, reemplazando tareas de codificación repetitivas. Kumbia intenta proporcionar facilidades para construir aplicaciones robustas para entornos comerciales. Esto significa que el marco de trabajo es muy flexible y configurable (KUMBIA 2012).

1.4.4 Symfony 3

Symfony 3 es un marco de trabajo PHP que facilita el desarrollo de las aplicaciones web. Se encarga de todos los aspectos comunes de las aplicaciones web, dejando que el programador se dedique a aportar valor, desarrollando las características únicas de cada proyecto. Este marco de trabajo ayuda a aumentar la productividad y mejorar la calidad de las aplicaciones web concebidas mediante su uso (POTENCIER 2008).

Características principales:

- Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- Funciona con todas las bases de datos más comunes (MySQL, PostgreSQL, SQLite, Oracle, Microsoft SQL Server).
- Compatible solamente con PHP 5, para asegurar el mayor rendimiento y acceso a las características más avanzadas de este lenguaje.
- Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- Flexible y extensible mediante un completo mecanismo de plugins.
- La vista usa plantillas (templates y layout) que pueden ser construidas por diseñadores Web que no posean conocimientos de este marco de trabajo.
- Publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo.
- Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en la base de datos, ofreciendo una mejor experiencia para el usuario.
- El manejo de caché reduce el uso de banda ancha y la carga del servidor.
- La facilidad de soportar autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.
- Traducido a más de 40 idiomas y fácilmente se puede llevar a cualquier otro idioma (EGUÍLUZ PÉREZ 2011).

Cuadro comparativo de marcos de trabajo en PHP.

Tabla 1.1: Cuadro comparativo de marcos de trabajo en PHP

Características	Symfony 3	Zend	CakePHP
Incorporación del patrón Modelo Vista Controlador.	X	X	X
Uso de plantillas en PHP	X	X	X
Posibilidad de uso de plantillas en Smarty	X		X
Soporte para PHP4.			X
Integración con otras herramientas a través de plugins.	X		X
Generación de código PHP	X		X
Mapeo de Objetos a bases de datos relacionales	X		X
Independiente del manejador de base de datos	X	X	X
Implementación de ayudantes de plantillas	X		X
Almacenamiento en caché de las vistas	X		X
Interfaz de línea de comandos para la creación y mantenimiento de aplicaciones	X		X
Soporte para PHP5	X	X	X

1.5. Buenas Razones para escoger Symfony 3

Symfony 3 cuenta con un motor de plantillas llamado Twig, teniendo en sí un sistema muy recomendado por los desarrolladores para el desarrollo de las vistas. Twig, entre otras cosas por su sintaxis sencilla, escape automático de html y la inclusión de funciones personalizadas y filtro, también tiene la posibilidad de definir variables en él, sin necesidad de pasarlas desde el controlador. Twig tiene muchos beneficios, como su velocidad, ya que compila el código de las plantillas a código PHP optimizado. También ganaremos en seguridad, debido a que Twig evalúa el código no confiable. Además, se va a mejorar la flexibilidad a la hora de crear las plantillas debido a la posibilidad que ofrece de implementar filtros y etiquetas personalizadas, así como nos permite la herencia de plantillas, por lo que no es más que si dos páginas comparten la misma estructura y muchas etiquetas html gracias a Twig se puede crear una plantilla base que incluya todos los elementos comunes y después hacer que cada

plantilla individual herede de la nueva plantilla base. Y, por último, hay que reseñar que las plantillas generadas son muy fáciles de leer y entender, tanto por los desarrolladores, como por los diseñadores web, por eso y demás la comunidad que hay detrás de Twig, lleva varios años siendo el principal motor de plantillas que se usa en Symfony 3. Existen disímiles razones que hacen que Symfony 3 sea un marco de trabajo importante para el desarrollo de aplicaciones Web, estas son:

- Gran reputación pues es un entorno estable y reconocido a nivel internacional. Symfony 3 también posee una comunidad activa, compuesta en su mayoría de desarrolladores, integradores, usuarios y otros colaboradores que participan en el enriquecimiento continuo de esta herramienta.
- Detrás de Symfony 3 existe una empresa fundada por su creador: Sensio. Creado hace más de 12 años, Sensio es una agencia web concebida para sus propias necesidades, el marco de trabajo Symfony 3 hoy en día sigue siendo la herramienta de uso diario por sus propios equipos para desarrollar proyectos de clientes. Con miras a un desarrollo sostenible Symfony 3 se distribuye bajo licencia MIT Open Source, que no impone restricciones y permite el desarrollo de código abierto, así como las aplicaciones propietarias.

Cuenta con una gran cantidad de referencias, entre ellas contamos con intranets corporativas, redes sociales, sitios de la comunidad, sitios para el público en general, etc. (EGUÍLUZ PÉREZ 2011).

Trabajar con este framework significa lidiar con los fundamentos de la Web, con buenas prácticas de desarrollo y con la utilización de algunas de las mejores librerías PHP de la actualidad. Una ventaja muy importante de Symfony 3 es la filosofía de construir tu aplicación, no tus herramientas, ya que es el framework el que se las ofrece al desarrollador. Por ejemplo: en cualquier aplicación o sitio Web, el objetivo es interpretar cada petición entrante y crear la respuesta adecuada. El problema está en que, cuando la aplicación crece, es más difícil de organizar y de mantener el código y, además, en la mayoría de los casos, los desarrolladores se van a encontrar con las mismas tareas complejas: persistencia de información a la base de datos, procesar y reutilizar plantillas, manejar formularios, validar los datos del usuario, administrar la seguridad del sitio, etc. Todas estas tareas tediosas y

complejas las ofrece el framework como herramientas desacopladas o independientes que los desarrolladores pueden utilizar cuando quieran. Entre las más destacadas están:

- **HttpFoundation:** contiene las clases Request y Response, así como otras clases para manejar sesiones y cargar archivos.
- **Routing:** potente y rápido sistema de enrutado que permite al desarrollador asignar una URI específica a cierta información acerca de cómo se debe manejar dicha petición. Mediante este sistema de enrutado se dispone de URLs totalmente flexibles.
- **Form:** una completa y flexible plataforma para crear formularios y procesar los datos presentados en ellos.
- **Validator:** un sistema para crear reglas sobre datos y así comprobar si los datos que presenta el usuario son válidos o no siguiendo esas reglas.
- **Templating:** juego de herramientas para utilizar plantillas, que soporta desde la herencia de plantillas (es decir, una plantilla está decorada con un diseño) y hasta otras tareas comunes de las plantillas.
- **Security:** una poderosa librería para manejar todo tipo de seguridad dentro de una aplicación.

1.5.1 Doctrine

Doctrine es un potente y completo sistema ORM (Object Relational Mapper) para PHP 5.2+ con un DBAL (Data Base Abstraction Layer) incorporado, que viene por defecto con Symfony. Tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. Por otro lado, como la biblioteca es bastante grande ésta tiene un método para ser compilada al pasar a producción. Además de todas las ventajas que conlleva un ORM, uno de sus puntos fuertes es su lenguaje DQL (Doctrine QueryLanguage). Inspirado en el HQL de Hibernate, este lenguaje permite escribir

opcionalmente consultas a la base de datos en un lenguaje OO (Orientado a Objetos) brindando a los desarrolladores una poderosa alternativa al SQL manteniendo una máxima flexibilidad sin necesidad de duplicar código (SENSIOLABS 2009).

Cuando se trabaja con ORM, necesitas crear el conjunto de clases que representa el modelo de la aplicación, luego estas clases serán vinculadas al esquema de la base de datos de forma automática con un motor ORM. Aunque son cosas diferentes, cuando se diseña un modelo relacional y un modelo de clases, suelen ser muy parecidos. Doctrine aprovecha esta similitud y permite generar de forma automática el modelo de clases basándose en el modelo relacional de tablas. Es decir, si tenemos una tabla llamada usuarios, se autogenerará una clase llamada Usuarios cuyas propiedades son las columnas de dichas tablas (SENSIOLABS 2009).

1.5.2 Symfony y el patrón de arquitectura MVC

Symfony 3 está basado en un patrón clásico del diseño web conocido como arquitectura MVC (Modelo-Vista-Controlador), que está formado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el controlador), los datos (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

Por tanto, una misma aplicación que se debe ejecutar tanto en un navegador estándar como en un navegador de dispositivo móvil, solamente necesita crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original.

El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.



Figura 1.2: Filosofía de trabajo de Symfony

CAPÍTULO 2.

**DISEÑO Y ANÁLISIS DE LA
APLICACIÓN PARA EL CONTROL DE
ALMACÉN DE LA CERVECERÍA
MANACAS.**

CAPÍTULO 2. Diseño y Análisis de la Aplicación para el Control del Almacén de La Cervecería Manacas.

Este capítulo tiene como objetivo fundamental, abordar los aspectos tanto metodológicos como informáticos que fueron escogidos para la construcción de la aplicación. Se presentarán diagramas que describen las funcionalidades de cada actor del sistema, así como el diagrama conceptual de la base de datos, el diagrama de clases que permite dar respuesta a cada caso de uso de la aplicación y el diagrama de Casos de Usos del sistema junto con su descripción. En el capítulo 1 se realizó un estudio de las herramientas candidatas por lo que, para su desarrollo se emplea el lenguaje PHP, el cual aportará el dinamismo a la página además de la interacción con la base de datos. Se integra con HTML para definir el contenido de las páginas, JavaScript para mostrar algunos de los resultados y para algunas cuestiones de dinamismo y CSS para definir los estilos visuales. Se utilizó MySQL debido a que es un Sistema Gestor de Bases de Datos (SGBD) rápido, relacional, y flexible, posibilitando realizar múltiples y rápidas consultas; siendo esencial para soluciones que contengan almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Para culminar el capítulo se presentará una visualización de la aplicación, mediante un breve diseño visual de la misma.

2.1. Diagrama de casos de uso del sistema

Los diagramas de casos de uso son importantes para visualizar, especificar, y documentar el comportamiento de un elemento. Aunque son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso. En los conceptos se debe detallar más de un caso de uso para poder identificar qué es lo que hace un caso de uso, mientras los diagramas de Casos de Usos son esenciales para una descripción coherente del comportamiento deseado, limitando del sistema cuanto puede interactuar cada actor.

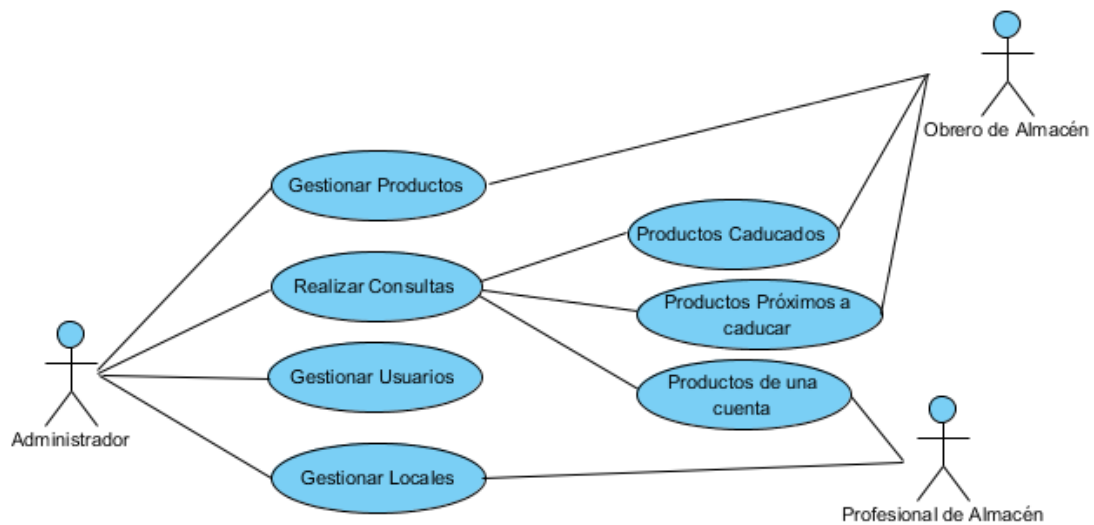


Figura 2.1: Diagrama de casos de usos del sistema

Descripciones de los casos de uso

Las tablas a continuación describen de forma detallada en qué consisten los casos de uso del sistema.

Tabla 2.1: Muestra la explicación del caso de uso "Gestionar Producto".

Caso de uso:	Gestionar Producto
Propósito	Permite a los actores operar con los productos en la BD
Actores	Administrador y Obrero de Almacén
Precondiciones	Acceder al sistema
Fuente: elaboración propia.	

PRODUCTOS EXISTENTES

Mostrar 10 entradas

Buscar:

Fecha	Nombre	Cantidad de Productos	Tipo de Local	Identidad del Local	Acciones
09/05/17 12:00 am	q	2300	Sección	13	<div> <div></div> <div></div> </div>
15/05/17 12:00 am	Cloro	30	Estante	6	<div> <div></div> <div></div> </div>
15/05/17 12:00 am	sal	50	Casilla	8	<div> <div></div> <div></div> </div>

Eliminar Producto

Editar Producto

Figura 2.2: Vista de los Productos existentes

Control de Almacén

MENÚ

Sobre Nosotros

Perfil de Usuario

Locales

Usuarios

Productos

Insertar Producto

Listar Productos

Listar Productos Caducados

Listar Productos Próximos a Caducar

Listar Productos de una Cuenta

Cervecería de Manacas

Bienvenido a la Cervecería de Manacas

SUS DATOS

VISUALICE SUS DATOS Y MODIFIQUELOS SI DESEA

Nombre Completo: Miladys Jacomino

Carné de Identidad: 64082807458

Nombre de Usuario: mili

Tipo de Usuario: Administrador

MODIFICAR DATOS

6

Cantidad de Locales Vacíos

50.00%

3

Cantidad de Locales Llenos

25.00%

Figura 2.3: Vista del Inicio del usuario administrador

33

El caso de uso Gestionar Producto se inicia cuando los actores logran entrar al sistema mediante su correcta autenticación y acceden al menú correspondiente "Productos", donde se muestra la información y todas las acciones que se pueden hacer en el sistema sobre los Productos, ya sea una vez listados todos los productos existentes en ese momento dentro el almacén, se puede editar cualquier producto; donde se muestra la información almacenada de ese producto para su posterior modificación; o eliminar algún seleccionado, además de poder insertar los datos correspondientes a un nuevo producto.

Tabla 2.2: Muestra la explicación del caso de uso "Gestionar Usuario".

Caso de uso:	Gestionar Usuario
Propósito	Permite al actor insertar los usuarios en la BD
Actor	Administrador
Precondiciones	El sistema se encuentra disponible el Administrador se encuentra autenticado.
Fuente: elaboración propia.	

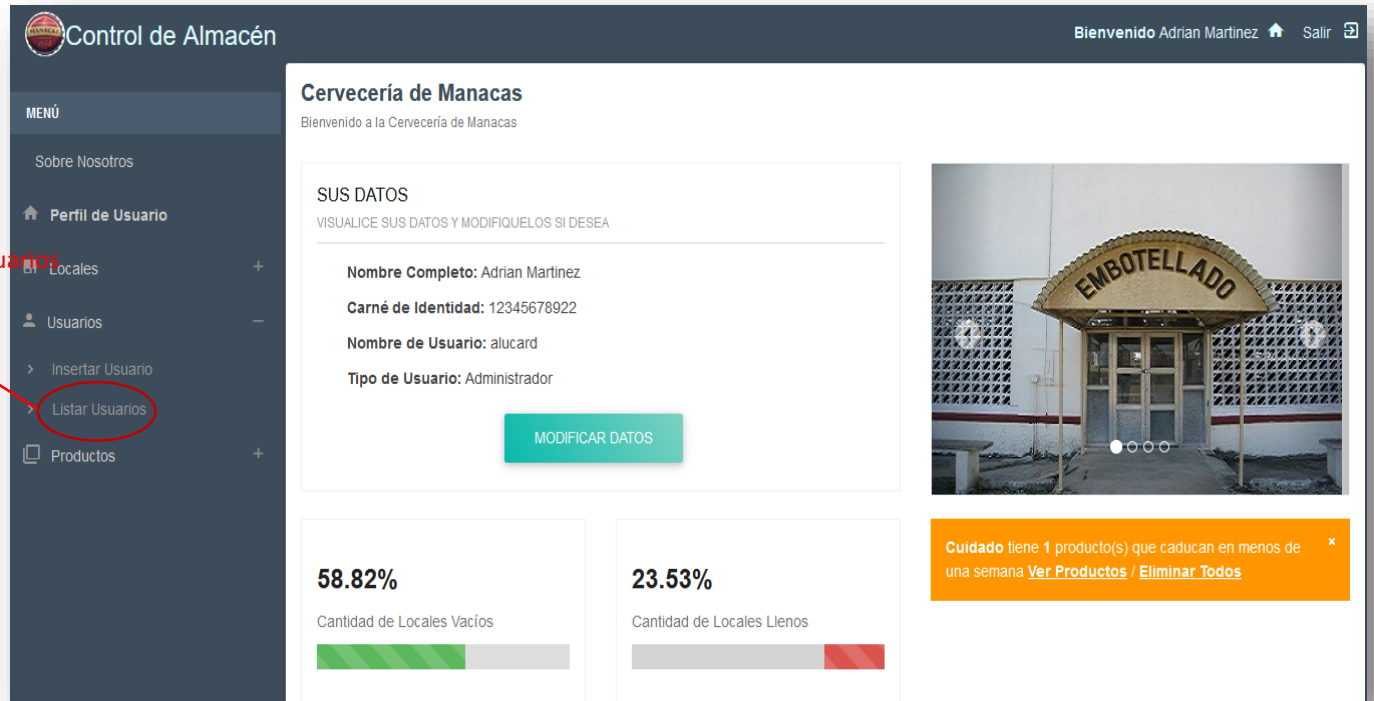


Figura 2.4: Vista del Inicio del usuario administrador

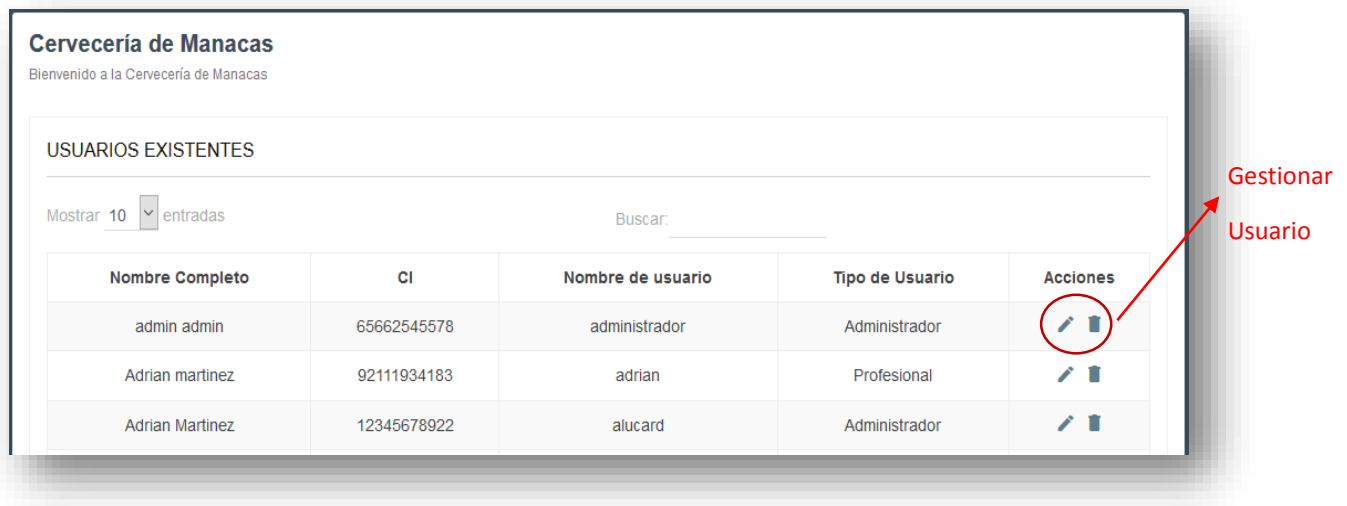


Figura 2.5: Vista de los Usuarios existentes en el sitio

El caso de usos anterior se inicia cuando el actor logra entrar al sistema mediante su correcta autenticación y accede al menú correspondiente de Listar Usuarios, donde tienen la opción de “Eliminar usuarios”, o “Editar usuario”, el actor podrá modificar cualquier dato que corresponda al usuario que seleccione, además de también poder registrar algún usuario nuevo que valla a formar parte del sistema.

Tabla 2.3: Muestra la explicación del caso de uso " Sobre Nosotros".

Caso de uso:	Sobre Nosotros
Propósito	Documentar al usuario que interactúe, sobre datos de la fábrica
Actores	Todos los Usuarios del sistema
Precondiciones	El sistema se encuentra disponible y algún actor esta logeado
Fuente: elaboración propia.	

El caso “Sobre Nosotros”, nos muestra datos de la empresa, además de poder acceder a una galería de imágenes donde encontramos fotos tanto exteriores como interiores de la Fábrica.



Figura 2.6: Vista de la página "Sobre Nosotros"



Figura 2.7: Vista de la galería

Tabla 2.4: Muestra la explicación del caso de uso "Listar Productos Próximos a caducar".

Caso de uso:	Listar Productos Próximos a caducar
Propósito	Permite a los actores insertar, y eliminar las cuentas en la BD
Actores	Administrador y obrero de Almacén
Precondiciones	El sistema se encuentra disponible y algún actor esta logeado
Fuente: elaboración propia.	



Figura 2.8: Vista de los Productos próximos a caducar

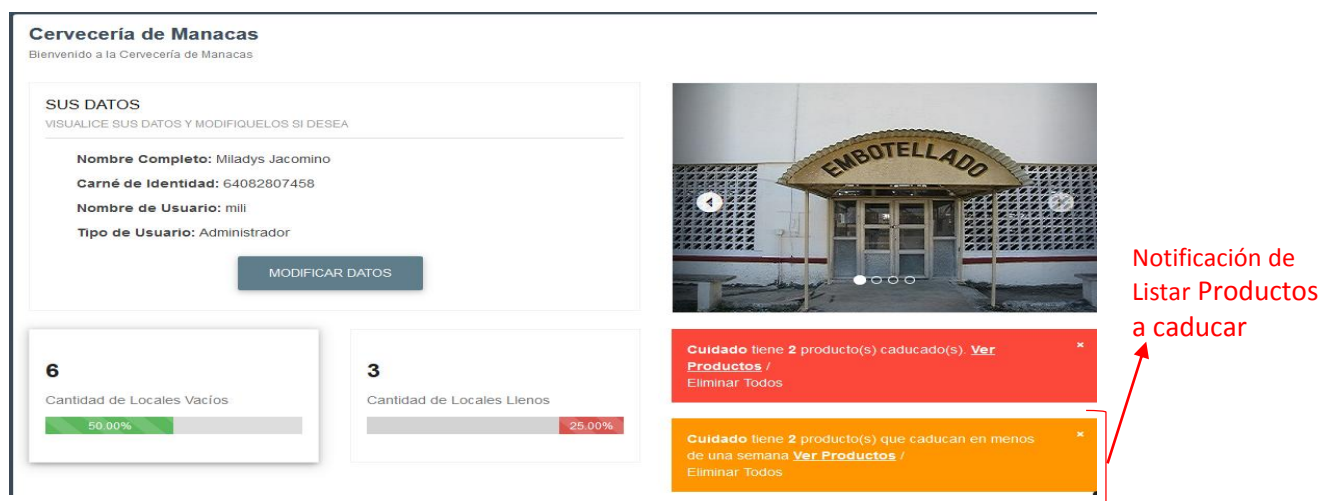


Figura 2.9: Vista del Inicio del usuario administrador

El caso de uso de Listar Productos Próximos a caducar, incluye a los actores Administrador y Obrero de Almacén, los cuales tienen el privilegio de una vez logeado en el sistema, visualizar una notificación donde puedan saber cuáles son aquellos productos que según una decisión caducarán en una menos de una semana, teniendo la posibilidad de listarlos y saber cuáles son, o eliminarlos todos, ya que no se permiten tener productos caducados dentro del almacén.

2.2. Modelo Entidad-Relación

La aplicación brinda la posibilidad de realizar estudios simultáneos a múltiples páginas. Para eso se hace necesario almacenar datos a medida de la interacción de los usuarios a la Web. Dichos datos aumentarán mucho, en dependencia de la cantidad de productos que se vayan registrando en el software. Por ello se necesita una base de datos para almacenar dichos datos y trabajar con ellos fácilmente. Explicado anteriormente al inicio de este capítulo las razones por las que se escogió, se usa MySQL.

2.2.1 Descripción de las Tablas

La Base de Datos consta de ocho tablas, de las cuales una es abstracta, a continuación, la descripción de las tablas que conforman nuestra Base de Datos.

La tabla *User* almacena los datos de los usuarios existentes en la aplicación. La misma contiene los siguientes atributos: *id_user*, *nombre*, *apellido*, *password*, *usuario*, *role*. El primero hace referencia a la tabla *Producto*, es un identificado autogenerado de cada usuario del sistema. El segundo y el tercero son el nombre y apellidos respectivamente del usuario en cuestión en ese momento. El cuarto indica la contraseña del usuario que está codificada, y *Usuario* es el nombre del usuario dentro del sistema (No su nombre propio). Por último, *role* es lo que diferencia en cuanto a privilegios a cada usuario, ya sea (Administrador, Obrero de Almacén, o Profesional).

La tabla *Producto* se encarga de almacenar los productos ingresados por los usuarios que tengan ese permiso. La misma contiene como atributos: *id_producto* es un identificador autogenerado para identificar a cada producto, *id_user* es una referencia a la tabla *User*, *id_casilla* es una referencia a la tabla *Casilla*, *id_departamento* es una referencia a la tabla

Departamento, id_estante es una referencia a la tabla Estante, fecha_entrada son los valores en día mes y año del momento que un producto es ingresado en el Almacén, tiempVida_dia, tiempVida_mes, tiempVida_anho, no es más que el tiempo en criterio de día, mes y año respectivamente que puede estar el producto en el almacén y pueda ser utilizable, que no paca a ser un producto ocioso, localizacion es el lugar donde se encontrará dicho producto una vez puesto en el almacén(Casilla, Departamento, Estante), nomb_producto es el nombre real del producto, cuenta representa en la categoría a la que va a pertenecer dicho producto lo cual se especifica con un número fijo (183-Materia Prima y Materiales, 185-Partes y Piezas de Repuestos,187-Útiles y Herramientas,192-Vestuarios y Lencerías, 196-Producciones para Insumo o Autoconsumo, 208-Inventarios Ociosos), unit_medida representa en la unidad que se va a archivar dicho producto (paquete, caja, Kg, Lata, Lingote).

La tabla Producto-Ubicacion y Ubicacion-Producto son tablas que no aparecen en nuestro modelo, sin embargo, en la implementación de la base de datos, el propio Symfony 3 las crea intermedias para almacenar en ellas la ubicación de un producto mediante sus identificadores, esto es posible pues existe una relación mucho a mucho entre las tablas Producto y Ubicacion.

La tabla Ubicacion representa, un lugar en específico donde se pondrá un producto, contiene solo su identificador como atributo.

La tabla Estante representa, un lugar en específico para colocar un producto, contiene 3 atributos: id_Ubicacion, que referencia a la tabla ubicación de la cual hereda, max_numProductos, es un entero que especifica la cantidad máxima de productos que puede contener ese Estante y el atributo ocupadoque nos va diciendo cuanto de ese Estante está siendo utilizado, una vez llenado se procederá a colocar productos en el próximo Estante que se encuentre vacío y de no existir ninguna, se colocará el producto en el que este menos lleno.

La tabla Seccion representa, un lugar en específico para colocar un producto, contiene 3 atributos: id_Ubicacion, que referencia a la tabla ubicación de la cual hereda, max_numProductos, es un entero que especifica la cantidad máxima de productos que puede contener esa Seccion y el atributo ocupado que nos va diciendo cuanto de esa Seccion está siendo utilizado, una vez llenado se procederá a colocar productos en la próxima Seccion que se encuentre vacía y de no existir ninguna, se colocara el producto en la que este menos lleno.

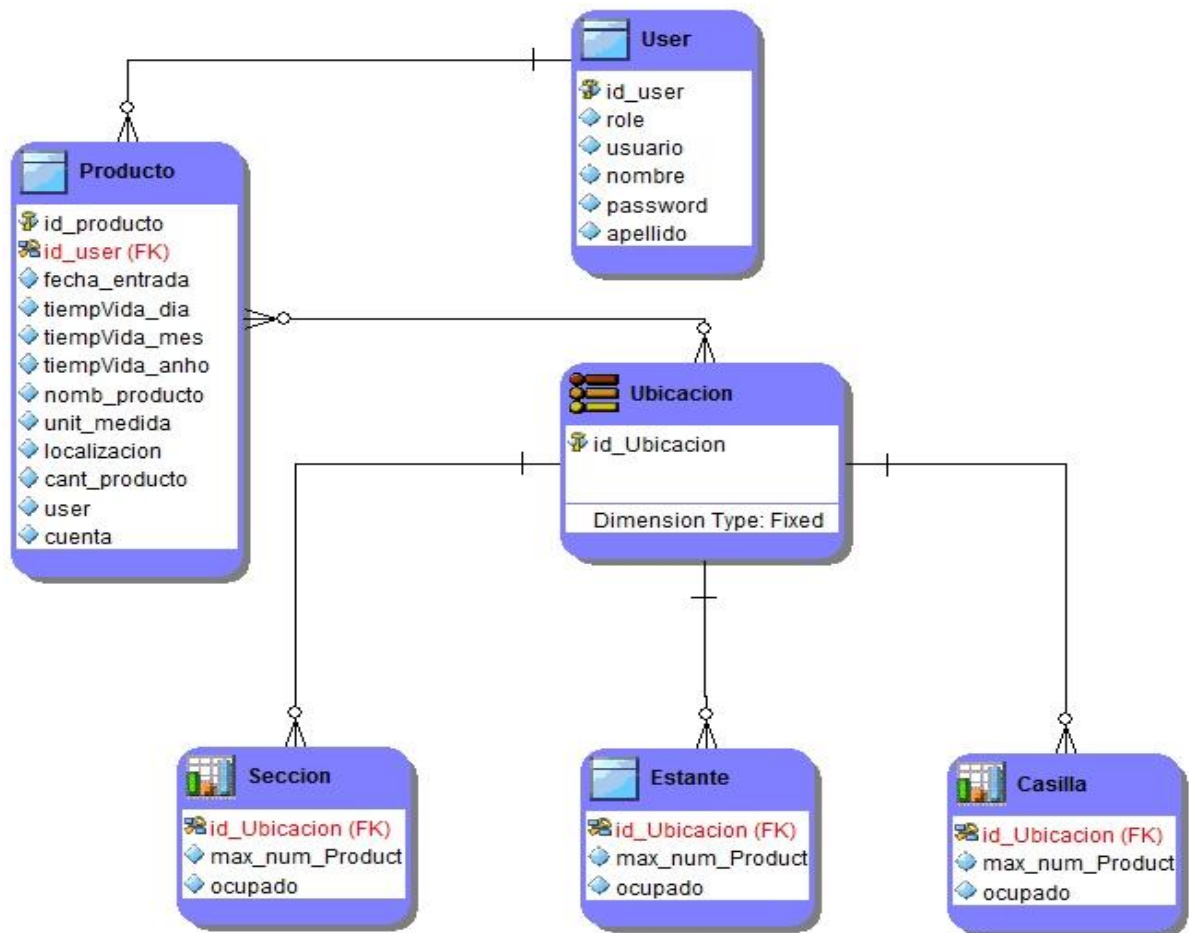


Figura 2.10: Diagrama Entidad-Relación

La tabla Casilla representa, un lugar en específico para colocar un producto, contiene 3 atributos: id_Ubicacion, que referencia a la tabla ubicación de la cual hereda, max_numProductos, es un entero que especifica la cantidad máxima de productos que puede contener esa Casilla y el atributo ocupado que nos va diciendo cuanto de esa Casilla está siendo utilizada, una vez llenada se procederá a colocar productos en la próxima Casilla que se encuentre vacía y de no existir ninguna, se colocara el producto en la que este menos lleno.

2.3. Diagramas de Actividades

El propósito de los diagramas de actividad es modelar un proceso de flujo de trabajo, en este caso se confeccionaron los diagramas de actividad correspondientes a los casos de uso que realiza un Administrador, Gestionar usuarios y Gestionar productos.

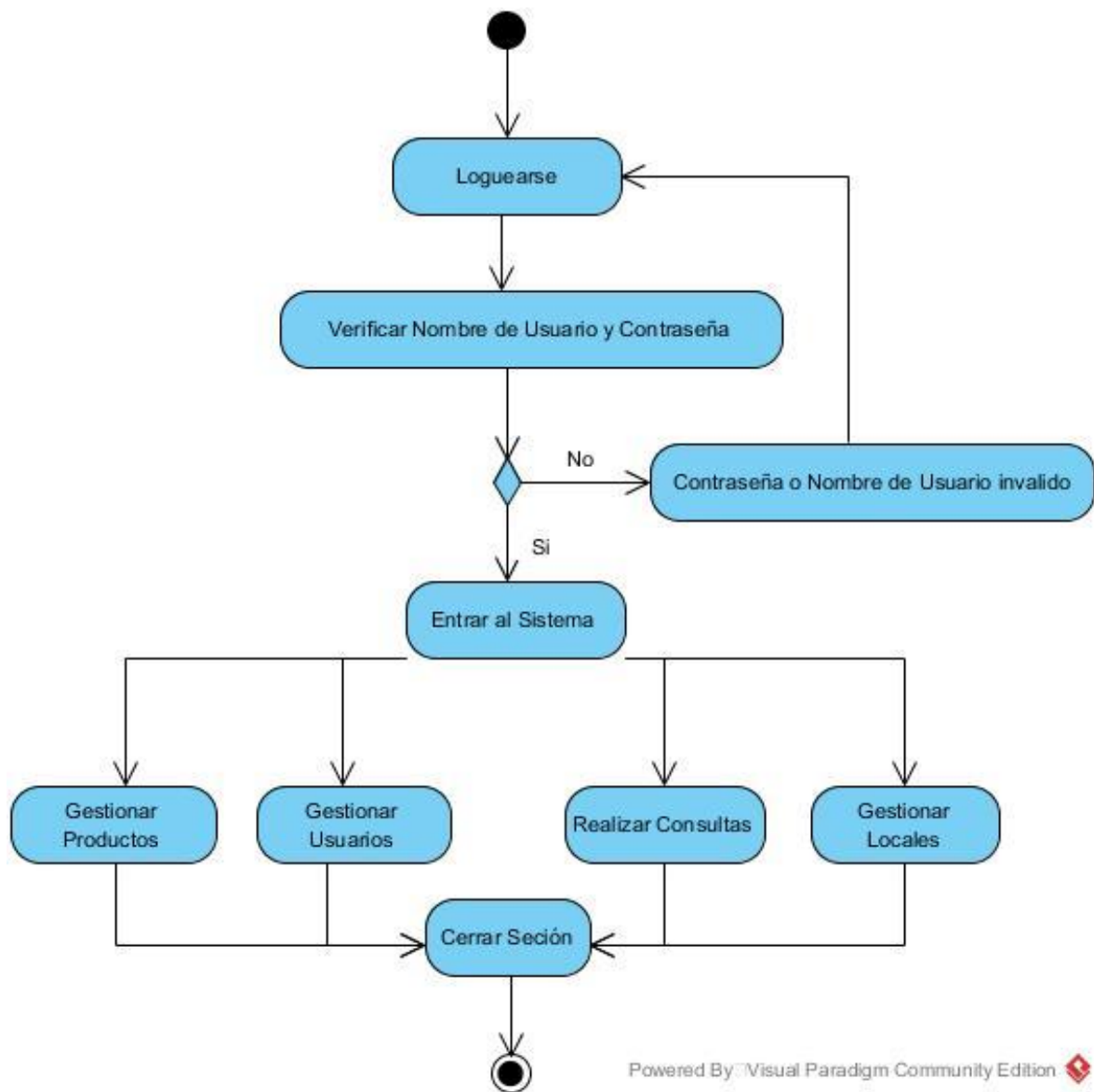


Figura 2.11: Diagrama de actividad correspondientes a Gestionar Productos y Gestionar Usuarios

2.4. Diagrama de Secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista del escenario del negocio, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos. Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos. A continuación, se muestran los diagramas de secuencia para los casos de uso Adicionar usuario y Mostrar productos de una cuenta, que mostrará todos los productos que existen dentro del almacén de acuerdo a una cuenta seleccionada. Véase figura

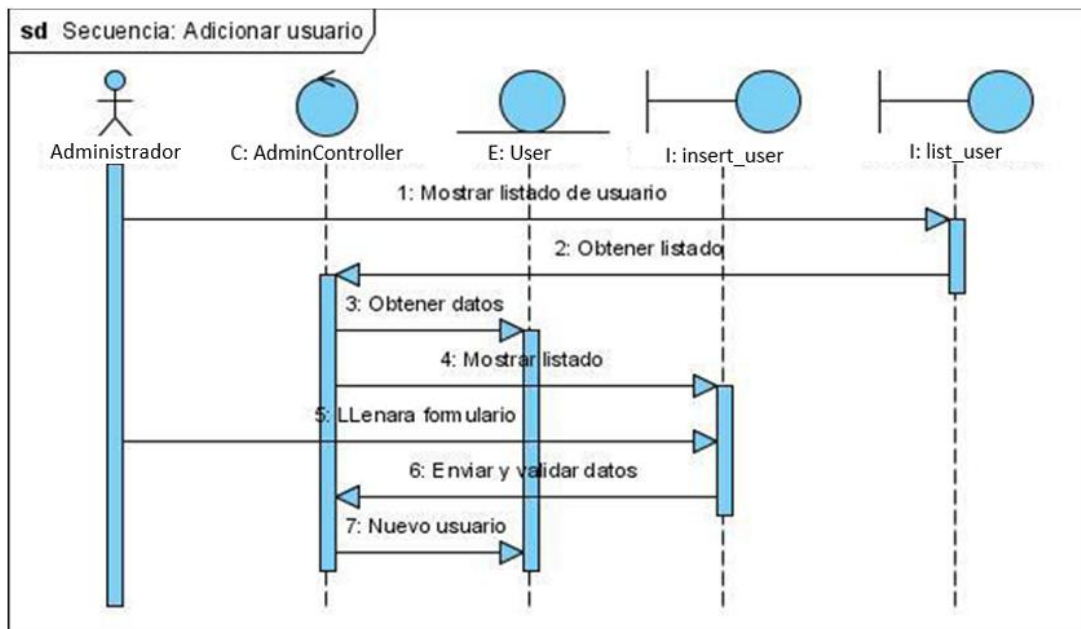


Figura 2.12: Diagrama de Secuencia para Adicionar Usuario

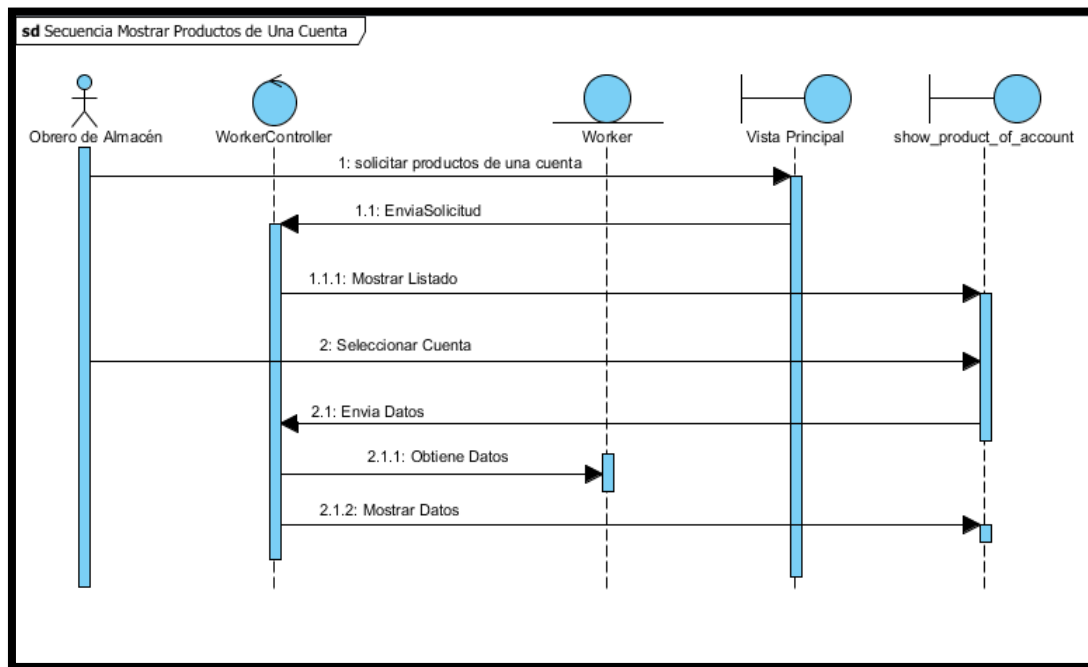


Figura 2.13: Diagrama de Secuencia Mostrar Productos de una cuenta

2.5. Diagrama de Componentes

El diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema, así como la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. Cada diagrama describe un apartado del sistema. En él se situarán librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema. Véase **¡Error! No se encuentra el origen de la referencia.**

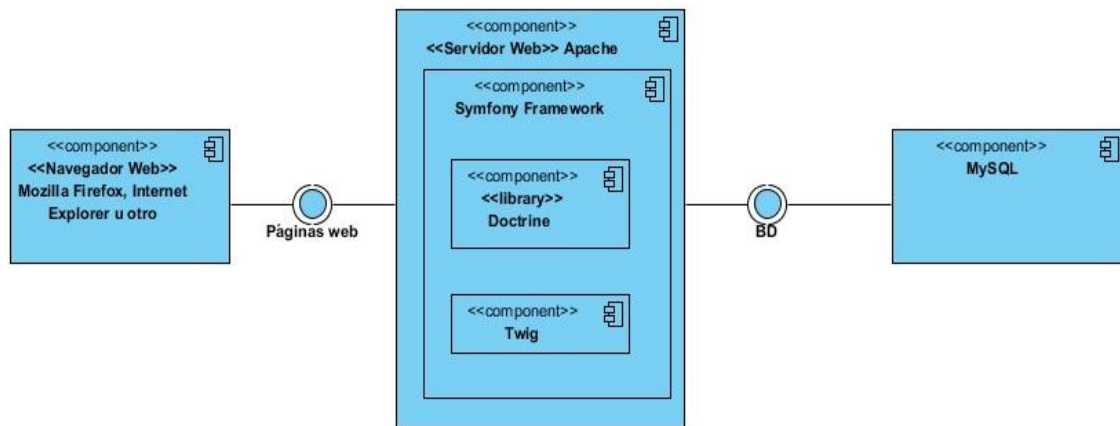


Figura 2.14: Diagrama de Componentes

2.6. Diagrama de despliegue

El modelo de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos software. Con frecuencia conocemos cómo será la arquitectura física del sistema antes de comenzar su desarrollo. Por tanto, podemos modelar los nodos y las conexiones del modelo de despliegue tan pronto como comience el flujo de trabajo de los requisitos. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. En este caso para el desarrollo de la aplicación se cuenta con una computadora que será el servidor de Base de datos, teniendo como Sistema Gestor de Base de Datos MySQL, de igual manera la misma computadora será el servidor web con la aplicación instalada, también vemos en el diagrama las computadoras clientes a través de la cual se conectan los usuarios. En la siguiente figura se muestra el diagrama de despliegue correspondiente al sistema, donde existe una relación uno a mucho (1-m) entre cliente y servidor. Mediante el protocolo HTTP se establece una conexión de uno o varios clientes con una máquina que actúa de servidor sobre la cual corren, el servidor Web y el gestor de base de datos. Como requisitos tenemos que debemos tener una actualizada instalación del cualquier navegador, para mostrar satisfactoriamente el sitio. Véase **¡Error! No se encuentra el origen de la referencia.**

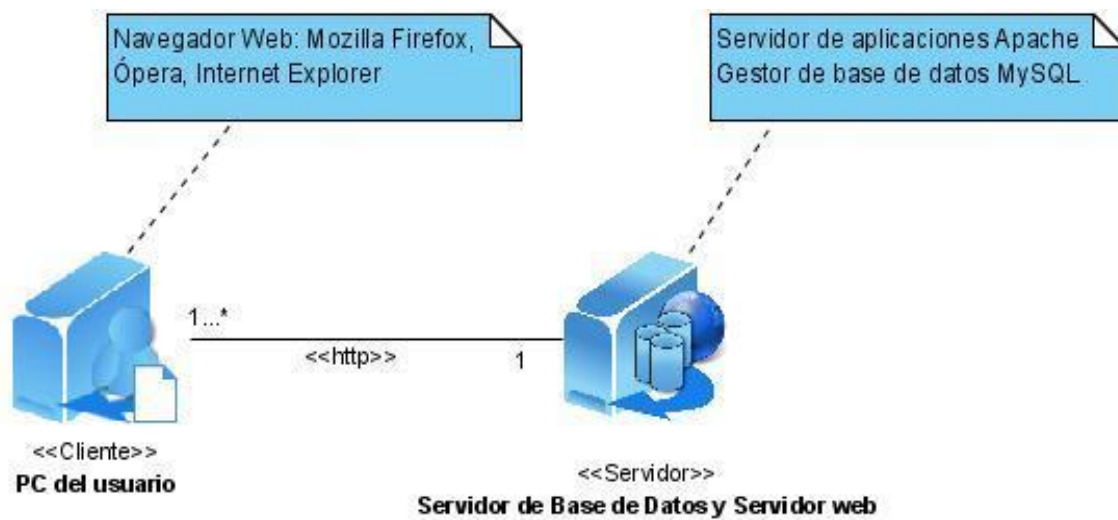


Figura 2.15: Diagrama de Despliegue

2.7. Requisitos funcionales

Los requisitos funcionales son importantes dado a que captan de qué forma el usuario va a usar el sistema, no son más que una representación de todas las funcionalidades necesarias, es decir que el software que se desea crear cumpla con las características deseadas por el cliente. A continuación, las funciones que deberá ser capaz de llevar a cabo incluyendo las acciones que podrán ser ejecutadas por el usuario. Para cumplir los objetivos de esta aplicación la misma tendrá que:

- a. Permitir al usuario introducir usuario y contraseña del sistema.
- b. Validar datos introducidos por el usuario.
- c. Permitir al usuario utilizar las funcionalidades que brinda todo el sistema.

2.8. Requisitos no funcionales

Los requerimientos no funcionales no son más que las propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Estos describen las restricciones del sistema o del proceso de desarrollo; no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento (JACOBSON 2004).

A continuación, se muestran los requerimientos no funcionales

- a. Interfaz del sistema: Diseño sencillo, permitiendo la utilización del sistema sin mucho entrenamiento.
- b. Usabilidad: El sistema ha sido diseñado de tal forma que permite a los usuarios tener control en todo momento del sitio web, evitando para eso la sobrecarga de información.
- c. Rendimiento: La aplicación debe ser eficiente, precisa y rápida.
- d. Soporte: El producto debe recibir mantenimiento ante cualquier fallo que ocurra. El sistema es de fácil instalación. Este debe ser integrado, de manera que se mantenga la colaboración entre todos los subsistemas que lo componen. La base de datos que utilizará el sistema como medio de almacenamiento de la información estará soportada sobre un gestor de bases de datos MySQL, permitiéndole interactuar con otros sistemas estableciendo vías de compatibilidad.
- e. Portabilidad: Es un sistema multiplataforma porque el mismo es utilizable en varios sistemas operativos.
- f. Seguridad: Este es quizás el tipo de requerimiento más difícil, que provocará los mayores riesgos si no se maneja correctamente. Debe garantizar la conectividad e integridad de los datos almacenados a través de la red. Esto está garantizado por el sistema operativo. El sistema impondrá un inscripto de contraseña (bcrypt) que viene integrado con el propio framework sobre el que se trabajó.
- g. Disponibilidad: Significa que los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.
- h. Confiabilidad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

- i. Ayuda y documentación en línea: Dispondrá de una ayuda bien detallada sobre las principales opciones del sistema, facilitando así el trabajo con la aplicación.
- j. Software: Del lado del servidor se debe contar con un sistema operativo compatible, para la instalación de la aplicación y debe ser instalado el Apache como servidor web, el PHP versión 5.3 o superior, como lenguaje de programación del lado del servidor y el MySQL como gestor de base. Del lado del cliente se necesita un navegador que interprete las funciones básicas de JavaScript, recomendado Mozilla Firefox 9.0 o superior.
- k. Hardware: Del lado del servidor se necesita una máquina que debe contar las siguientes características de hardware: procesador Pentium IV 2 GHz de frecuencia, 1Gb de memoria RAM, 40 Gb de capacidad de disco duro y tarjeta de red de 100 Mbps. Del lado del cliente las computadoras de los usuarios requerirán como mínimo un procesador Pentium III, 128 Mb de memoria RAM. Estas máquinas deben estar conectadas en red con el servidor a través de una tarjeta de red de 100 Mbps.
- l. Restricciones en el diseño y la implementación: Se utilizarán herramientas de desarrollo que garanticen la calidad de todo el ciclo de desarrollo del producto.

CAPÍTULO 3.

PRUEBAS DE LA APLICACIÓN

CAPÍTULO 3. PRUEBAS DE LA APLICACIÓN.

En el presente capítulo se describen los métodos de prueba que se realizaron a la aplicación para el control de almacén de la cervecería de Manacas. Se muestra además la descripción de las pruebas, los posibles errores y los resultados obtenidos. En el desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallas. La estabilidad, eficiencia y seguridad son conceptos que se relacionan a la calidad de un producto bien desarrollado. Las pruebas son de gran importancia en la garantía del software, una selección cuidadosa de los datos de prueba puede ofrecer mucha confianza en cuanto al desempeño que posee el programa. Para probar el correcto funcionamiento, tanto interno como externo, de una aplicación existen diferentes métodos de prueba. Los Métodos de Prueba de software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. Uno de estos métodos de prueba es el de caja negra que se enfoca directamente en la interfaz, sin importar el código, son pruebas en las que, a través de la entrada de datos, se trata de encontrar fallas. Otro de los métodos de prueba que existe es el de caja blanca que se centra en los detalles de la implementación del software, por lo que su diseño está fuertemente ligado al código fuente de la aplicación.

3.1 Evaluación de calidad web.

Pruebas de Calidad de Software, son los procesos que permiten verificar y revelar la calidad de un producto. Las pruebas de calidad de software se integran en las diferentes fases del ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas se trata de descubrir que errores tiene, además pueden predecir cuando un software.

3.1.1 Importancia de hacer las pruebas de calidad.

Toda prueba de software desempeña un papel fundamental en el desarrollo de cualquier tipo de aplicación. Estas pruebas son un elemento importante para la garantía de calidad del software y presenta una revisión final de las especificaciones, del diseño y de la codificación.

1. El sistema es entregado al cliente con todas las funcionalidades trabajando de manera adecuada.
2. Es una manera de comprobar que no existen errores en el sistema.

3. Se comprueba que no va tener incompatibilidades con distintos entornos de hardware o de sistemas operativos.
4. Mostrar hasta qué punto las funciones del software operan de acuerdo con las especificaciones y requisitos del cliente.

3.1.1.1 Tipos de Pruebas de Software

Las pruebas en conjunto tienen como objetivo general verificar y validar un software, independientemente de las características y el entorno donde se desarrollen, además de los recursos y los factores vinculados al proceso de desarrollo (ECURED 2009).

1-Funcionalidad

- Pruebas fijando su atención en la validación de las funciones, métodos, servicios, caso de uso.
- Asegurar que los datos o el sistema sean solamente accedido por los actores deseados.

2-Usabilidad

Prueba enfocada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea, asistente documentación de usuarios y materiales de entrenamiento.

3-Fiabilidad

- Capacidad de un sistema para mantener su nivel de rendimiento.
- Valoración a la adherencia a su diseño y formación. Este tipo de prueba es hecho a las aplicaciones Web asegurando que todos los enlaces están conectados, el contenido deseado es mostrado y no hay contenido huérfano.
- Evaluar cómo el sistema responde bajo condiciones anormales. (extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible).

4-Rendimiento

- **Benchmark:** Es un tipo de prueba que compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo de referencia conocido.
- **Contención:** Enfocada a la validación de las habilidades del elemento a probar para manejar aceptablemente la demanda de múltiples actores sobre un mismo recurso (registro de recursos, memoria).
- **Carga:** Usada para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales.

5-Soportabilidad

- **Configuración:** Enfocada a asegurar que funciona en diferentes configuraciones de hardware y software. Esta prueba es implementada también como prueba de rendimiento del sistema.
- **Instalación:** Enfocada a asegurar la instalación en diferentes configuraciones de hardware y software bajo diferentes condiciones (insuficiente espacio en disco, etc.)

3.1.2 Pruebas de calidad del Software realizadas al sistema.

El sitio web fue sometido a dos pruebas de calidad de software.

Usabilidad: el sistema web fue creado para el Almacén de una institución que posee características especiales tanto arquitectónicamente como en los servicios que ofrece, por tanto, se necesitaba construir una interfaz que estuviera acorde con estas particularidades. Tanto el diseño de interfaz del sistema, así como las funcionalidades fueron sometidas criterios de 10 usuarios que serán en un futuro próximo los que utilizaran el sistema, entre ellos solo 4 se desenvuelven en las ramas de la informática.

Después de una serie de preguntas a los encuestados, todas estas personas coinciden que a primera vista el sistema refleja el lugar al que se referencia, además no tuvieron

complicaciones en navegar dentro del sistema, aunque algunas de las personas sugirieron otras funcionalidades, que pueden ser incorporadas.

A parte de las personas encuestadas un factor importante es el juicio del cliente quien quedo complacido con el producto presentado.

Eficiencia: Para hacer las pruebas de eficiencia nos basamos en los distintos entornos de red desde los que se puede acceder al sitio. Estos entornos fueron creados de forma virtual con la herramienta Dummynet. Es originalmente una herramienta diseñada por probar los protocolos de la gestión de redes, y desde entonces usó para una variedad de aplicaciones incluso la dirección del ancho de banda. Este útil se puede usar con varios sistemas operativos como son Linux, Mac OS X y las versiones de Windows.

El Dummynet trabaja interceptando los paquetes a través de uno o más objetos llamados cañerías que simulan los efectos de las limitaciones del ancho de banda que la propagación tarda y las pérdidas de paquetes (GSP 2002).

Pasos seguidos para llevar a cabo la prueba con el programa Dummynet:

1. Instalar el software.
2. Se configura una conexión, es decir crear los entornos o puentes de red diferentes, introduciendo códigos por la consola CMD.
3. Tener la aplicación web ejecutándose en otra computadora.
4. Luego con el navegador Mozilla Firefox (usado para hacer las pruebas) se hacen los distintos llamados a la web y este proporciona las estadísticas del tiempo que demora el acceso a la misma.
5. Se borra el historial del navegador para que no quede ninguna huella del sitio y se pueda aplicar la prueba nuevamente desde el paso número 2 cambiando la velocidad de los puentes.

Siguiendo el estándar utilizado por el buscador Google se creó un estándar para medir la eficiencia teniendo en cuenta los tipos de conexiones más comunes de nuestro país. Por lo

que se creó los siguientes intervalos para la clasificación de la eficiencia según el tiempo de espera y los entornos de conexiones posibles.

El tiempo de respuesta del servidor al cliente depende del entorno de red que esté utilizando entonces:

1. Si el tiempo que demora el sitio en mostrarse completo esta entre 0 a 5000: la eficiencia es muy buena.
2. Si el tiempo que demora el sitio en mostrarse completo esta entre 5000 a 50000: la eficiencia es buena.
3. Si el tiempo que demora el sitio en mostrarse completo esta entre 50000 a 105000: la eficiencia es regular.
4. Si el tiempo que demora el sitio en mostrarse completo es mayor que 105000: la eficiencia es mala.

3.2 Pruebas de caja negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que comprueben completamente todos los requisitos funcionales de un sistema (PRESSMAN 2002).

Estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una “Caja Negra” cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas. No obstante, como el estudio de todas las posibles entradas y salidas de un programa sería impracticable se selecciona un conjunto de ellas sobre las que se realizan las pruebas. Para seleccionar el conjunto de entradas y salidas sobre las que trabajar, hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en nuestro sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos. Entonces, dado que la prueba exhaustiva es imposible, el objetivo final es pues, encontrar una serie de datos de entrada cuya probabilidad

de pertenecer al conjunto de entradas que causan dicho comportamiento erróneo sea lo más alto posible.

Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas
- La entrada se acepta de forma correcta
- Se produce una salida correcta
- La integridad de la información externa se mantiene

Las pruebas de caja negra pretenden encontrar estos tipos de errores:

- Funciones incorrectas o ausentes
- Errores en la interfaz
- Errores en estructuras de datos o en accesos a bases de datos externas
- Errores de rendimiento
- Errores de inicialización y de terminación

Una de las pruebas de caja negra que se realizaron al sistema para comprobar que los datos introducidos eran los correctos, fue poblar nuestra Base de Datos con información real que existía en el Almacén y luego se hizo un listado de los productos a mostrarle al usuario Administrador. Se accedió a la funcionalidad de listar los productos caducos al Obrero de Almacén, y se comprobó que el listado de los productos mostrados eran los correctos, se introdujeron datos de los usuarios que no estaban correctos, así como en su carnet de identidad se agregaron letras en ese campo y luego se agregaron números que fueran de una cantidad de cifra distinta a 11 y en ambos caso obtuvimos alertas donde se pedía que el carnet debe tener 11 cifras y que en ese campo solo se puede introducir números, se fue comprobando que el sitio notificara cada introducción errónea, para lograr una posterior y correcta información a la Base de Datos.

CONCLUSIONES

CONCLUSIONES

Con la implementación del sistema se logró dar cumplimiento a todos los requisitos funcionales identificados por lo que con esta solución el control dentro del Almacén en la Fábrica Cervecera de Manacas puede realizarse de una forma más automatizada y profesional.

- 1- Se realizó el diseño de una base de datos relacional que permite gestionar los datos necesarios para lograr una organización automática de ya mencionado almacén.
- 2- Se validó el funcionamiento de la aplicación web mediante diferentes métodos de prueba de software como el de caja blanca y el de caja negra. Se realizaron pruebas al sistema para comprobar el correcto funcionamiento de algunos de los casos de uso de la aplicación.

RECOMENDACIONES

Una vez desarrollado el sistema y cumplidos los objetivos propuestos en el trabajo se recomienda:

- Implementar versiones posteriores en caso que se le adicione alguna funcionalidad nueva al sistema.
- Dar seguimiento al trabajo para atender posibles recomendaciones y necesidades de los usuarios del sistema, una vez implantado, logrando hacer las mejoras pertinentes.

REFERENCIAS BIBLIOGRÁFICAS

- ANDERSON, J. M., L. E. (2006). "CakePHP: A rapid development Framework. CakePHP Programmer's Reference Guide. Cake Software Foundation Inc."
- APACHE, F. (2012). "Apache software foundation."
- CHAFFER, J. S., K (2007). "Learning JavaScript."
- CHASE, N. (2006). "Understanding the Zend Framework."
- ECURED (2009). "Calidad de software, tipos de pruebas."
- ECHEVERRIA, A. M. (1995). "Manual Práctico de Html. Madrid, España."
- EGUÍLUZ PÉREZ, J. (2008). "Introducción a CSS.", from www.librosweb.es
- EGUÍLUZ PÉREZ, J. (2011). "Symfony español." from In: WWW.LIBROSWEB.ES.
- FIAEP (2014). "CONTROL Y MANEJO DE INVENTARIOS Y ALMACÉN CUBA."
- GSP. (2002). "Manual Reference Pages." from www.info.iet.unipi.it.
- JACOBSON, I., BOOCH, G. & RUMBAUG, J (2004). "El proceso unificado de desarrollo de software, La Habana, Editorial Félix Varela."
- KUMBIA (2012). "Kumbia PHP "
- MICROSOFT. (2011). "Microsoft SQL Server." from www.microsoft.com/spain/sql/default.mspx.
- MySQL (2008). "Kumbia PHP."
- PHP (2007). "Hypertext Preprocessor."
- POTENCIER, F. Z., F (2008). "Symfony la guía definitiva."
- PRESSMAN, R. S. (2002). "Ingeniería del software."
- RUIZ, M. H. (2007). "Una introducción a APACHE."
- SÆTHER BAKKEN, S., AULBACH, A., SCHMID, E., WINSTEAD, J., TORBEN WILSON, L., LERDORF, R., ZMIEVSKI, A. & AHTO, J (2001). "Manual de PHP. In: MARTÍNEZ, R. (ed.).".
- SENSIOLABS (2009). "Doctrine ORM for PHP."
- YUEN, P. K. L. (2003). "Practical Web Technologies."