

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE APIZACO

DIVISION DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**“ANÁLISIS Y DESARROLLO DE MODELO PARA EL SISTEMA
COTIZADOR DE UNA EMPRESA DE MADERA INDUSTRIAL”**

TESIS

PARA OBTENER EL GRADO DE:

MAESTRA EN SISTEMAS COMPUTACIONALES

PRESENTA:

ING. MARÍA DE LOS ÁNGELES HERNÁNDEZ PALAFOX

DIRECTORES:

M. en C. JOSÉ JUAN HERNÁNDEZ MORA

M. en C. MARÍA GUADALUPE MEDINA BARRERA



"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

Apizaco, Tlax., 02 de Agosto de 2017

No. de Oficio: DEPI/268/17

ASUNTO: Se Autoriza Impresión de Tesis de Grado.

ING. MARÍA DE LOS ÁNGELES HERNÁNDEZ PALAFOX
CANDIDATA AL GRADO DE MAESTRA
EN SISTEMAS COMPUTACIONALES
No. de Control: M10370131
P R E S E N T E.

Por este medio me permito informar a usted, que por aprobación de la Comisión Revisora asignada para valorar el trabajo, mediante la Opción: I Tesis de Grado por Proyecto de Investigación, de la Maestría en Sistemas Computacionales, que presenta con el tema: "ANÁLISIS Y DESARROLLO DE MODELO PARA EL SISTEMA COTIZADOR DE UNA EMPRESA DE MADERA INDUSTRIAL" y conforme a lo establecido en el Procedimiento para la Obtención del Grado de Maestría en el Instituto Tecnológico, la División de Estudios de Posgrado e Investigación a mi cargo le emite la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar un ejemplar del mismo debidamente encuadernado y seis copias en CD en formato PDF, para presentar su Acto de Recepción Profesional a la brevedad.

Sin otro particular por el momento, le envío un cordial saludo.

ATENTAMENTE
PENSAR PARA SERVIR, SERVIR PARA TRIUNFAR®


DR. JOSÉ FEDERICO CASCO VÁSQUEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN.



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE ESTUDIO
DE POSGRADO E INVESTIGACIÓN

JFCV*mebr.

C.p. Expediente.



Apizaco, Tlax., 01 de Agosto de 2017

ASUNTO: Aprobación del trabajo de Tesis de Maestría.

DR. JOSÉ FEDERICO CASCO VÁSQUEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN.
P R E S E N T E.

Por este medio se le informa a usted, que los integrantes de la **Comisión Revisora** para el trabajo de tesis de maestría que presenta la **ING. MARÍA DE LOS ANGELES HERNÁNDEZ PALAFOX**, con número de control **M10370131** candidata al grado de **Maestra en Sistemas Computacionales** y egresada del **Instituto Tecnológico de Apizaco**, cuyo tema es "**ANÁLISIS Y DESARROLLO DE MODELO PARA EL SISTEMA COTIZADOR DE UNA EMPRESA DE MADERA INDUSTRIAL**", fue:

A P R O B A D O

Lo anterior, al valorar el trabajo profesional presentado por la candidata y constatar que las observaciones que con anterioridad se le marcaron así como correcciones sugeridas para su mejora ya han sido realizadas.

Por lo que se avala se continúe con los trámites pertinentes para su titulación.

Sin otro particular por el momento, le envié un cordial saludo.

LA COMISIÓN REVISORA

M.C. JOSÉ JUAN HERNÁNDEZ MORA

M.C. MARÍA GUADALUPE MEDINA BARRERA

M.C.C. MARÍA JANAÍ SÁNCHEZ HERNÁNDEZ

M.C. JUAN RAMOS RAMOS

C. p - Interesada



Carr. México-Tehuacan s/n. Apizaco, Tlax. C.P. 76500
Tel: (01) 241 417 20 10 (L) 115 2106 (M) posgrado@itapizaco.edu.mx



AGRADECIMIENTOS

Agradezco a CONACYT el apoyo económico, para mi preparación.

A mis profesores que han aportado con un granito de arena a mi formación con su conocimiento, experiencia, paciencia y motivación.

A Dios por darme la oportunidad de concluir con esta labor de manera satisfactoria.

A mi esposo Alejandro Mendoza Cuapio agradezco su amor, apoyo, comprensión y confianza pues mis logros son también tuyos.

A mis padres Margarita Palafox González y Paulino Hernández Moreno por haberme forjado como la persona que soy en la actualidad; muchos de mis logros se los debo a ustedes.

A mis hermanos que siempre han estado presentes cuando los necesito por su motivación continua para alcanzar mis sueños.

De igual manera agradezco a todas las personas que han formado parte de mi vida, a las que me encantaría agradecerles por su amistad, consejos, apoyo, ánimo y compañía.

RESUMEN

El presente trabajo de tesis describe el desarrollo del modelo de base de datos (BD) para un sistema cotizador de madera industrial, siendo el primero en esta línea mercantil.

Este sistema permite realizar cálculos de los diferentes productos aplicando tipos de madera, acabados, tratamientos, tamaño de grosor, servicio de ingeniería, mano de obra, instalaciones en planta, en sitio, tiempo de instalación en semanas y transporte, todo esto con el fin de no generar pérdidas a la empresa.

El sistema tiene una base de datos que permite el acceso del usuario a la información, permitiéndole realizar operaciones y administrar los datos para poder mantener a la empresa informada de cuántas ventas fueron realizadas, cuántas se encuentran en proceso o canceladas, todo esto a través de reportes.

El desarrollo se basa en la metodología ágil SCRUM y en el patrón de diseño Modelo Vista Controlador (MVC). El modelo es uno de los tres módulos que conforma el sistema cotizador, igualmente este sistema lo conforman dos módulos más: la interfaz gráfica y el controlador.

Para la implementación del modelo se utiliza MySQL que es un sistema de gestión de base de datos, en conjunto con *workbench* para el modelado y administración de datos, permitiendo un manejo rápido e intuitivo para su manipulación.

Para la integración al sistema se utiliza *hibernate*, herramienta de mapeo objeto relacional para la plataforma java facilitando la conexión con la base de datos y el control del sistema.

ABSTRACT

The present thesis describes the development of a database model (BD) for a industrial quoting system for wood being the first in this line of trade.

This system allows you to make calculations of the different products by applying types of wood, finishes, treatments, size, thickness, and engineering service, manpower, facilities, in-plant, on-site installation time in weeks, and transport. All this calculation in order to not generate losses to the company.

The system has a database that allows the user access to the information, allowing you to perform operations and manage the data in order to keep the company informed of how many sales were made, how many are in process, or canceled, all of this through reports.

The development is based on agile methodology, SCRUM and in the design pattern Model-View-Controller (MVC). The model is one of the three modules that make up the system rater, also this system is composed by two modules: the graphical interface and the controller.

The model implementation uses MySQL which is a database management systema, in conjunction with workbench for modeling and management of data, allowing operation fast and intuitive handling.

The integration to the system is used hibernate, object-relational mapping tool for the Java platform facilitating the connection with the database and the control of the system.

Contenido

Índice de figuras	viii
Índice de tablas.....	x
Acrónimos y abreviaturas	xi
INTRODUCCIÓN	1
Capítulo 1 Introducción	2
1.1. Descripción del problema.....	2
1.2. Justificación.....	3
1.3. Objetivos	4
1.3.1. Objetivo general.....	4
1.3.2. Objetivos específicos.....	4
1.4. Alcances.....	4
1.5. Limitaciones	4
1.6. Pregunta de investigación	5
1.7. Organización del contenido.....	5
1.8. Estado del arte	6
Capítulo 2 Marco teórico	14
2.1. Ingeniería del software	14
2.2. Modelo Vista Controlador.....	16
2.3. Metodologías Agiles	18
2.4. Metodología SCRUM	19
2.4.1. Proceso de SCRUM	20
2.5. Modelo de datos.....	22
2.6. Base de datos	23

2.6.1. Tipo de base de datos	23
2.6.2. Modelo de base de datos	25
2.7. Sistemas manejadores de base de datos	29
2.7.1. Características.....	30
2.7.2. ACID.....	31
2.8. Herramientas de mapeo.....	32
2.8.1. Hibernate	32
2.8.2. Java Persistence API.....	33
2.9. Arquitectura cliente – servidor.....	34
2.9.1. Ventajas.....	35
2.10. Modelo V.....	36
2.11. Evaluación del software	38
2.11.1. Pruebas de software.....	38
2.11.1.1. Pruebas de integridad	39
2.12. Estándar de evaluación de la calidad de software	41
2.13. Industria maderera y mercadeo	42
2.14. Venta de madera industrial	42
Capítulo 3 Metodología	44
3.1. Metodología de implementación	44
3.2. Modelo vista controlador	45
3.3. SCRUM para el modelo de base de datos	46
3.3.1. Planeación.....	48
3.4. Metodología para la elaboración del modelo de base de datos	49
3.5. Análisis de requerimientos	50
3.5.1. Diseño	51

3.5.2. Implementación del modelo	52
3.5.3. Pruebas	52
Capítulo 4 Implementación	54
4.1. Análisis de requerimientos	54
4.2. Definición de actores.....	57
4.3. Análisis de requisitos.....	60
4.4. Planeación de base de datos	61
4.5. Diseño de la base de datos.....	62
4.5.1. Diseño conceptual	62
4.5.2. Diseño Lógico	64
4.5.3. Diseño físico	66
4.6. Implementación de modelo de base de datos.....	67
4.6.1. Mapeo de la base de datos	67
4.6.2. Módulo de creación de usuario.....	67
4.6.3. Modelo de administración de clientes.....	69
4.6.4. Módulo de catálogos.....	70
4.6.5. Módulo de cotización	70
4.6.6. Módulo de generación de cotización en PDF	71
Capítulo 5 Pruebas y Resultados	73
5.1. Plan de pruebas	73
5.1.1. Objetivo	73
5.1.2. Alcance.....	73
5.2. Prueba de funcionamiento	73
5.2.1. Prueba de validación de formularios.....	74
5.2.2. Prueba de búsqueda	76

5.3. Prueba de usabilidad.....	77
5.3.1. Prueba de contenido.....	77
5.4. Prueba de integración	79
5.5. Pruebas de seguridad	92
5.6. Pruebas de rendimiento	95
5.6.1. Prueba de carga	95
5.7. Prueba de integridad	96
5.7.1. Integridad de dominio	96
5.7.2. Integridad de entidad.....	98
5.7.3. Integridad referencial	99
Capítulo 6 Conclusiones	104
6.1. Conclusiones.....	104
6.2. Trabajos futuros	106
6.3. Experiencia adquirida.....	106
Bibliografía	107
ANEXOS	110
Anexo A Diseño de DBDL LENGUAJE DE DISEÑO DE BASE DE DATOS .	111
Anexo B Minutas de Sprint	115
Anexo C Artículos publicados	124
Anexo E Carta de satisfacción.....	141

Índice de figuras

Figura 2.1 Arquitectura del patrón MVC (Crookshanks, 2005).....	17
Figura 2.2 Fases de scrum (A, S, V, & J., 2007).	21
Figura 2.3 Base de datos relacional (Cobo, 2014)	26
Figura 2.4 Base de datos Jerárquica (Cobo, 2014).....	27
Figura 2.5 Base de datos multidimensional (wordpress, 2016)	28
Figura 2.6 Modelo en red de base de datos (wordpress, 2016)	29
Figura 2.7 Arquitectura cliente servidor (Sommerville, 2011).	35
Figura 2.8 Modelo V (Martinez & Barrio, 2016)	36
Figura 3.1 Patrón de diseño con MVC (Fuente propia)	45
Figura 3.2 Metodología para el desarrollo del sistema cotizador basado en SCRUM (Suarez, 2013).....	47
Figura 3.3 Modelo para la creación de la base de datos (Fuente propia)	50
Figura 4.1 Diagrama de casos de uso del sistema general (Fuente propia)	59
Figura 4.2 Diagrama de caso de usos de administración de operaciones (Fuente propia)	60
Figura 4.3 Datos de caratula de cotización (Fuente del cliente).....	61
Figura 4.4 Tablas UML de cotizador, detallecotiza y acabados (Fuente propia) ...	62
Figura 4.5 Diagrama entidad- relación del sistema cotizador (Fuente propia)	63
Figura 4.6 Diagrama relacional de la base de datos (Fuente propia).....	65
Figura 4.7 DBDL de acabados	66
Figura 4.8 Mapeo de la base de datos (Fuente propia).....	67
Figura 4.9 Correlación entre la interfaz y la base de datos (Interfaz elaborada por personal de diseño).....	68
Figura 4.10 Iteración entre la base de datos e interfaz	69
Figura 4.11 Iteración entre la BD e interfaz de usuario	70
Figura 4.12 Correlación de la BD con interfaz del cotizador	71
Figura 4.13 Relación entre la generación de PDF y la BD	72
Figura 5.1 Prueba de formulario (Fuente propia)	75
Figura 5.2 Prueba de búsqueda	76

Figura 5.3 Prueba de contenido (Fuente propia)	78
Figura 5.4 Integración del modelo MVC (Control e Interfaz elaborado por personal encargado)	92
Figura 5.5 Respaldo de base de datos.....	92
Figura 5.6 Carpeta con respaldos de la base de datos	93
Figura 5.7 Prueba de seguridad	94
Figura 5.8 Prueba de rendimiento de la BD.	95
Figura 5.9 Sintaxis de creación de tabla con referencia	99
Figura 5.10 Inserción de datos desde consola.....	100
Figura 5.11 Inserción de datos desde interfaz.....	101
Figura 5.12 Inserción de datos	101

Índice de tablas

Tabla 1.1 Análisis comparativo del estado del arte(Fuente propia).....	12
Tabla 2.1 Comparativa Hibernate vs Java Persistence API	34
Tabla 3.1 Requerimientos para la pila de producto	48
Tabla 4.1 Requerimientos funcionales para el sistema cotizador.....	55
Tabla 4.2 Requerimientos no funcionales para el sistema cotizador.....	56
Tabla 4.3 Descripción de las fases de la metodología del modelo de BD	57
Tabla 4.4 Definición de actores	58
Tabla 5.1 Pruebas de integración.....	80
Tabla 5.2 Diccionario de datos de cliente.....	96
Tabla 5.3 Prueba de dominio	97
Tabla 5.4 Resultados de prueba de dominio	97
Tabla 5.5 Prueba de entidad	98
Tabla 5.6 Resultado de pruebas en el sprint 5	102
Tabla 5.7 Resultado de pruebas en el sprint 8	102
Tabla 5.8 Porcentajes de evaluación	102

Acrónimos y abreviaturas

MVC	Modelo vista controlador.
SGBD	Sistema gestor de base de datos.
DBDL	Database Design Language- Lenguaje de diseño de base de datos.
BD	Base de datos
ISO	International organization for standardization- Organización internacional de normalización.
SCRUM	Framework de desarrollo ágil
XML	Extensible markup language- Lenguaje de marcas extensible.
DES	Data Encryption Standard- Estándar de encriptación de datos.
AES	Advanced Encryption Standard- Estándar de encriptación avanzada.
RC2	Clave simétrica de cifrado por bloques.

INTRODUCCIÓN

En la empresa de madera industrial se realiza el proceso de venta de vigas para llevar a cabo proyectos de diferentes magnitudes. Generalmente las personas de forma inconsciente están interactuando con una base de datos realizando operaciones durante su jornada laboral. Los sistemas de información automatizados simplifican el acceso a una base de datos, en donde se tiene la posibilidad de mostrarse de forma que el usuario ordenada y

Con el sistema se puede tener un fácil acceso a los datos almacenados.

Hoy en día existe una gran cantidad de información que se debe mantener segura, integrad y accesible, así como tener una administración para que esté disponible en todo momento. El valor que tiene nos lleva a utilizar bases de datos relacionales que gracias a ellas se pueden resolver gran cantidad de problemas como lentitud en las búsquedas de datos, así como la perdida de información.

La empresa de madera industrial tiene hoy en dia la problemática a la hora de realizar cotizaciones ya que para realizar una propuesta al cliente se debe pasar por un proceso demasiado lento. Debido a este problema se propone la implementación de un sistema web a través de un modelo de base de datos preciso, le da integridad consistencia.

En el modelo almacenará información referente a costos de fabricación, material, instalación, productos, usuarios, totales de pies tabló, cotizaciones generadas y reportes de las mismas.

Para la implementación de la base de datos se trabaja en conjunto con dos módulos: la interfaz gráfica y el módulo de control que es donde se encuentra toda la lógica de negocios y con la unión de ambos se tiene el sistema de cotización completo. Esta forma de trabajo consistió es la división del patrón de diseño Modelo Vista Controlador (MVC) y en conjunto con la metodología ágil SCRUM permitió trabajar con pequeñas iteraciones en un tiempo determinado.

Capítulo 1

Introducción

1.1. Descripción del problema

Actualmente la empresa de madera industrial maneja un archivo de Microsoft Excel para hacer cotizaciones, por lo cual los operadores del sistema presentan algunos inconvenientes al momento de generar una propuesta de venta a un cliente.

Las principales causas están relacionadas con el manejo del archivo puesto que no cuenta con las bases necesarias para su entendimiento, considerando que una hoja de cálculo está representada por filas y columnas lo que dificulta su uso, ya que a un vendedor poco familiarizado con el documento se le complicaría la realización y/o la manipulación de una cotización.

Otra de las problemáticas consiste en no llevar un registro adecuado sobre las ventas y propuestas realizadas, puesto que se llega a extraviar información al sobrescribir la hoja de cálculo.

Considerando que el archivo es reutilizado, se ocasiona que los precios de los productos y materiales no son modificados continuamente por lo que las cotizaciones llegan a presentar errores en las operaciones que se realizan, arrojando resultados elevados o muy bajos, creando pérdidas; en cuanto menos propuestas sean aceptadas menos producción tiene la empresa.

Al analizar estos problemas se propone la solución con la implementación de un sistema que permita el almacenamiento de la información, así como el manejo de datos para realizar cotizaciones con resultados precisos.

1.2. Justificación

Existiendo un gran giro dentro del desarrollo tecnológico, se ha detectado la necesidad de desarrollar una aplicación capaz de realizar cotizaciones de proyectos específicos, que fluyen dentro del negocio de venta de madera industrial.

Esta aplicación será capaz de sincronizarse con la base de datos donde se guardará información necesaria para la funcionalidad del sistema.

Es necesario recalcar que el modelo de BD realizará operaciones desde cualquier sitio, teniendo la seguridad de que la información generada sobre una cotización será guardada correctamente. Al mismo tiempo se brindarán los datos necesarios para realizar las cotizaciones, generar reportes, agregar, modificar, consultar y eliminar materiales, productos y servicios que la empresa ofrece.

La aplicación tendrá el acceso restringido para usuarios externos al uso de la misma y los usuarios que tendrán acceso contendrán, lo hará mediante un rol para poder realizar operaciones correspondientes a su puesto.

Con la ayuda del sistema de cotización el personal encargado de realizar las propuestas, tendrá la facilidad de realizar de forma rápida y precisa cada solicitud, dado que contará con una interfaz agradable, donde los datos serán manipulados de forma intuitiva.

Para la elaboración del sistema se utilizará la metodología ágil SCRUM la cual nos permite trabajar con pequeñas iteraciones en tiempos determinados, de igual manera esta metodología nos permite trabajar con el cliente para obtener mejores resultados. A su vez se utiliza el patrón de diseño MVC para el desarrollo e implementación del sistema dividido en tres módulos, por lo que facilitará su desarrollo e integración.

1.3. Objetivos

1.3.1. Objetivo general

“Desarrollar un modelo de base de datos para un sistema que realizará cotizaciones de madera industrial.”

1.3.2. Objetivos específicos

- Analizar los requerimientos del cliente.
- Analizar una metodología de aplicación para el desarrollo de una base de datos.
- Diseñar el modelo de la base de datos.
- Realizar la conexión a base de datos.
- Generar las consultas correspondientes, que permita el almacenamiento y administración de las cotizaciones generadas.
- Hacer pruebas de comunicación con la base de datos.

1.4. Alcances

- Diseño de un modelo funcional de acceso a datos.
- Automatización el proceso de generación cotizaciones.
- Almacenamiento de información de manera íntegra.
- Generación de vista de distintos tipos de reportes fundamentales para la empresa.
- Obtener información oportuna y actualizada en el momento en que el cliente lo solicite, agilizando el proceso de venta del producto.

1.5. Limitaciones

- Este trabajo se dirige principalmente a la especificación de la base de datos y no al diseño de la interfaz gráfica de usuario y la lógica de negocios, pero son parte de la integración del sistema.

- El sistema está basado al uso de internet

1.6. Pregunta de investigación

¿Cómo desarrollar un modelo de base de datos capaz de administrar información para realizar una cotización precisa? ¿Con fundamentos de requerimientos del cliente, la base de datos es funcional a partir del modelo MVC?

1.7. Organización del contenido

La estructura que contiene este trabajo se describe a continuación:

En el Capítulo 1 se describe el planteamiento del problema, la justificación, los objetivos el general y los específicos, los alcances y limitaciones, la organización del documento y el estado del arte, describiendo los trabajos relacionados con el sistema de cotización.

En el Capítulo 2 se describe el marco teórico donde se presentan los conceptos que se deben conocer para el entendimiento de la tesis, tales como: un sistema de software, las herramientas necesarias para el desarrollo de la aplicación, patrón de diseño MVC (Modelo, Vista, Controlador), metodologías de desarrollo ágil, base de datos e información referente al proceso de venta de madera industrial.

En el Capítulo 3 se presenta la metodología para la elaboración del modelo de base de datos que será parte de la integración del sistema de cotización, igualmente se menciona cada una de las fases que se llevaran a cabo durante el desarrollo del modelo.

En el Capítulo 4 se realiza el desarrollo e implementación del modelo de base de datos para el sistema cotizador. Se realiza los diferentes modelados de la BD y la integración con los dos módulos que son parte del sistema.

En el capítulo 5 se muestran diversas pruebas de software realizadas a la base de datos y al sistema cotizador, mostrando los resultados al integrar los tres módulos.

En el capítulo 6 se observan las conclusiones de este trabajo, las recomendaciones y las experiencias que se tuvieron durante la elaboración e integración del sistema.

1.8. Estado del arte

En este apartado se describen los trabajos relacionados con las bases de datos en varios sistemas de información, así como la utilidad que tienen en el almacenamiento de datos, tomando como referencia para el desarrollo de un modelo de base de datos en este proyecto.

Seguridad en desarrollo de aplicaciones web a través de anotaciones de la base de datos. (El-Hajj & Brahim, 2016)

Wassim El-Hajj describe un marco donde impone la seguridad a través de la construcción de aplicaciones web, donde el desarrollador aplicará el mínimo esfuerzo posible, solo necesita anotar los atributos de la base de datos con una clase de seguridad para comprobar la confiabilidad y la integridad de las aplicaciones web.

El marco se compone de cuatro categorías:

Ultra secreto: Son datos que nunca se deben propagar a la salida, pueden ser usados internamente pero nunca pueden ser enviados al navegador del cliente como resultados, por ejemplo, tarjetas de crédito, precio real en subastas, palabras claves de seguridad etc.

Solo la divulgación: son datos clasificados como única revelación, estos datos no están permitidos para visualización en grupo, sino solo una instancia de los datos. Una consulta a la base de datos que incluye a los datos etiquetados como único resultado, mostrando una matriz que contiene los datos clasificados como única revelación.

Necesita enmascarar: Los datos marcados como necesario enmascarar son los datos que se pueden propagar a la salida con ninguna restricción en el número de tupas, sino con la restricción de que los datos deben ser enmascarados o cifrados antes de ser enviados a la salida.

Público: en esta clase de seguridad todos los tipos de datos se muestran al público sin ningún tipo de limitación.

Medición de la eficiencia de la herramienta gráfica de usuario MyStarSQL, para el sistema gestor de base de datos MSQL server 5.x con respecto a otras herramientas gráficas. (Alvarado, 2011)

Iván Orlando describe el alto porcentaje de uso de MySQL en las empresas. Explica cómo es que MySQL es un sistema gestor de bases de datos, que se distribuye bajo licencia GNU y se caracteriza por ser un sistema de alto desempeño, con un alto grado de confiabilidad, seguridad, escalabilidad, y ayudando a la reducción de costos. Aborda el punto débil que tiene MySQL al carecer de una interfaz gráfica para la manipulación de datos, más sin embargo eso no demerita su eficiencia con la que trabaja, menciona varios softwares que interactúan con MySQL para poder tener una visualización gráfica de la base de datos; software como WorkBench, Navicat, PHPMyAdmin, Heidi SQL, de tal forma que recalca la facilidad con la que se puede crear, manipular y respaldar una base de datos con MySQL.

Conceptos y evaluación del enfoque entidad – relación extendido al enfoque para el diseño de base de datos en un entorno multi-paradigma en una herramienta de modelado de sistemas de información. (Dimitrieski, 2015)

Vladimir Dimitrieski describe en su artículo el enfoque basado en la relación de entidades extendido y la clase de modelos de datos, apoyado en formularios, clase de modelo de datos, así como en una herramienta de modelado de sistemas de información multi-paradigma. En su trabajo muestra la herramienta MIST que admite el enfoque para el diseño de una BD y proporciona una especificación formal de la especificación de esquemas de la base de datos y su transformación en el modelo relacional, también permite la generación de código de lenguaje consulta estructurado de procedimientos para la aplicación de restricciones de la base de datos.

Además, muestra que el código java, puede almacenar y procesar los datos de la base de datos a partir de un modelo de clases.

MIST proporciona las especificaciones de un esquema de DB, de aplicaciones empresariales y sus elementos de interfaz de usuario. Sin dejar a un lado que los

diseñadores utilizan la relación – entidad para sus esquemas de bases de datos. Así que la herramienta admite transformaciones de modelos habilitados por los componentes EER2Rel, EER2Class y R2FT, es un diseñador de base de datos el que puede seleccionar el modelo de datos que es más apropiado para el dominio del problema, o de sus conocimientos y preferencias.

Un enfoque orientado al diseño de base de datos para información geográfica (SIG) (Carrion, 2010)

Daniela Carrión hace mención en su artículo el uso de un modelo conceptual y lógico para representar grandes cantidades de datos que permita administrar la información geográfica.

A borda principalmente la gestión de los datos para la manipulación y difusión para los posibles usuarios. Dentro del diseño de la base de datos centra su atención en la congruencia e integración de los datos, así como en la interoperabilidad para compartir datos para diferentes aplicaciones.

El uso de un modelo relacional le permitió una notable flexibilidad, así como también proporcionar una base para el logro de la independencia de los datos en sus máquinas, mostrando calidad deseable en el comportamiento de la información. Teniendo las siguientes características dentro de la aplicación de los dos modelos.

Mínima redundancia de datos, acceso controlado a los datos, fácil organización desde el punto de vista de los usuarios, precisión y consistencia de los datos.

La combinación de usuario y perspectiva de la base de datos de palabras clave para resolver consultas sobre bases de datos relacionales. (Bergamaschi, 2015)

Sonia Bergamaschi describe un enfoque para resolver las consultas en las bases de datos relacionales, siendo las palabras clave un estándar para la búsqueda en las aplicaciones web. Trabaja en tres pasos fundamentales.

Hacer coincidir las palabras clave con las estructuras de la base de datos a través de mapeo.

Descubrir maneras para que la combinación sea coincidente y captar el significado de la palabra clave.

Seleccionar los mejores resultados y combinarlos para que los usuarios obtengan lo que tenían en mente.

Se presenta un sistema para la búsqueda llamado QUEST es completamente personalizado y capaz de proporcionar resultados precisos, independientemente del tamaño de la base de datos, ofreciendo consultas a bases de datos estructuradas.

NoSQL, la nueva tendencia en el manejo de datos (Herrera & Valenzuela, 2016)

Harol Andrey Herrera, menciona que NoSQL es la nueva tendencia en el manejo de los datos, que ofrece una mayor libertad en la definición de esquemas, ya que la responsabilidad recae sobre la aplicación y no sobre el motor de la base de datos, Además fortalece dos requisitos importantes en una aplicación como son la escalabilidad y la disponibilidad. De acuerdo a su forma de almacenar la información, se clasifica en: llave valor, familias de columnas, almacenes de documentos y de grafos.

NoSQL cuenta con las ventajas del escalamiento horizontal y de bajo costo en la infraestructura que las soporta lo que permite ofrecer a los desarrolladores de soluciones ágiles, así como más oportunidades de crecimiento y evolución incremental.

Base de datos utilizando el cifrado de seguridad. (Singh, 2015)

Prabhsimran Singh discute diversos esquemas de cifrado de la base de datos, el cifrado de base de datos es una estrategia clave para proteger el contenido de los datos, la idea principal detrás de esto es que en caso de que el intruso accede a los datos sea imposible de ver su contenido, el cifrado se realiza por dos maneras:

Codificación: un proceso en el que el texto se convierte en texto cifrado con ayuda de teclas, utilizando la misma clave podemos descifrar el texto cifrado a texto normal. El cifrado se realiza mediante algoritmos como DES, AES, RC2 128 256 AES etc.

El Hashing: es un proceso unidireccional en cuyo texto se convierte en un valor hash (forma cifrada). Este enfoque se utiliza generalmente para cifrado de contraseñas para inicio de sesión.

Las pruebas utilizando la base de datos web y diagrama E-R modelo de transición de estado. (Begum, abril 8, 2016)

Begum Sabeena nos presenta las pruebas para base de datos de la web donde son necesarias para la funcionalidad y desempeño. Las pruebas se realizan a través de la generación de casos de prueba basados en el sistema utilizando en la aplicación. El procedimiento básico para las pruebas es:

- Generar el modelo de base de datos y la aplicación web.
- Crear el conjunto de hechos *prolog* y las reglas.
- Generar las consultas mediante la combinación de la base de datos y la aplicación web.
- Ejecutar los casos de prueba para producir el informe de prueba.

Esta estrategia comprueba si la aplicación web realiza las funciones correctamente con las distintas operaciones de la base de datos. El modelo de transición de estado es tomado como modelo para la aplicación web que genera las diversas rutas de transición que atraviesa el usuario al utilizar la aplicación. Las operaciones que se realizan son de inserción, eliminación y las modificaciones que realiza el usuario entrada y salida de datos. El probador de base de datos es responsable de elaborar el informe de prueba para estas operaciones y proporciona el resultado que la aplicación está funcionando de acuerdo a las operaciones.

Modelo de diseño de bases de datos en condiciones de recursos limitados

(Klochkv & Klochkova, 2016)

Yury Klochkov nos menciona que hoy en día las bases de datos son componentes esenciales de los sistemas, lo que el mantenimiento requiere considerables recursos, menciona un modelo para realizar un mantenimiento de la base de datos en condiciones de recursos limitados.

Las bases de datos se basan en el establecimiento de relaciones en forma de correlaciones y dependencias entre trayectorias de desarrollo, la relación entre la trayectoria de desarrollo de la base de datos de información es representada por su dependencia analítica y los parámetros internos y externos de una organización, en cada punto de su trayectoria de desarrollo se considera como una combinación de proceso estático y dinámico. Un estado estático es caracterizado por valores estables como actividades que no requieren aumento o cambio a la información, en cambio a los dinámicos muestran cierto cambio con regularidad y es cambiante durante la trayectoria de desarrollo.

Sabiendo que los recursos son limitados y habiendo determinado el carácter de dependencia del nivel de calidad de la base de datos sobre la cantidad de recursos dedicados, se resolver el problema de la distribución óptima de recursos entre las bases de datos. Desarrollo de soporte de información dentro de una organización que definimos como una combinación de procesos estáticos y dinámicos.

Tabla 1.1 Análisis comparativo del estado del arte (Fuente propia)

N°	Tipo	Año de publicación	Área de aplicación	Característica	Aplicación a BD	Referencia
1	Artículo	2016	Desarrollo de BD	Marco para la seguridad en aplicaciones web para comprobar la confiabilidad e integridad de estas aplicaciones.	Seguridad de datos.	(El-Hajj & Brahim, 2016)
2	Artículo	2011	Desarrollo de BD	La medición de eficiencia en las herramientas manejadoras de base de datos	Manejadores de BD en interfaz gráfica.	(Alvarado, 2011)
3	Artículo	2015	Desarrollo de BD	Un enfoque basado en el diagrama E-R extendido en un entorno multiparadigma.	Diseño de la BD.	(Dimitrieski, 2015)
4	Artículo	2010	Desarrollo de software	Uso de modelos conceptual y lógico para presentar información de forma geográfica.	Almacenamiento y diseño de BD.	(Carrion, 2010)
5	Artículo	2015	Desarrollo de software	Uso de palabras claves para consultas de BD relacionales.	Búsqueda de datos en la BD.	(Bergamaschi, 2015)
6	Artículo	2016	Desarrollo de BD.	Una nueva tendencia en el manejo de la base de datos.	Manejador de base de datos.	(Herrera & Valenzuela, 2016)
7	Artículo	2015	Desarrollo de BD	Esquemas de cifrado de la base de datos.	Seguridad para el acceso a los datos.	(Singh, 2015)

Tabla 1.1 Continuación del Análisis comparativo del estado del arte

N°	Tipo	Año de publicación	Área de aplicación	Característica	Aplicación a BD	Referencia
8	Artículo	2016	Pruebas de software	Pruebas utilizando el diagrama E-R y el modelo de transición de estado.	Pruebas del software y BD.	(Begum, abril 8, 2016)
9	Artículo	2016	Desarrollo de BD.	Un modelo para brindar mantenimiento a la base de datos con recursos limitados.	Mantenimiento a la BD.	(Klochkv & Klochkova, 2016)

En conclusión, con los artículos analizados nos da una visión y partida para la elaboración del modelo de la base de datos, mostrando un panorama del uso de esquemas, los diferentes manejadores y administradores de base de datos, las restricciones que debe contener en cuestiones de seguridad y un poco de cómo realizar pruebas a partir del diagrama entidad relación.

Capítulo 2

Marco teórico

2.1. Ingeniería del software

La ingeniería de software es una disciplina de ingeniería que se interesa por todos los aspectos de la producción de software, desde las primeras etapas de la especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación. En esta definición se presentan dos frases clave:

Disciplina de ingeniería: Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde es adecuado. Sin embargo, selectiva y siempre tratan de encontrar soluciones a problemas, incluso cuando no hay teorías ni métodos aplicables. Los ingenieros también reconocen que deben trabajar ante restricciones organizacionales y financieras, de modo que buscan soluciones dentro de tales limitaciones.

Todos los aspectos de la producción del software: La ingeniería de software no sólo se interesa por los procesos técnicos del desarrollo de software, sino también incluye actividades como la administración del proyecto de software y el desarrollo de herramientas, así como métodos y teorías para apoyar la producción de software.

La ingeniería busca obtener resultados de la calidad requerida dentro de la fecha y del presupuesto. A menudo esto requiere contraer compromisos: Los ingenieros no deben ser perfeccionistas. Sin embargo, las personas que diseñan programas para sí mismas podrían pasar tanto tiempo como deseen en el desarrollo del programa.

En general, los ingenieros de software adoptan en su trabajo un enfoque sistemático y organizado, pues usualmente ésta es la forma más efectiva de producir software de alta calidad. No obstante, la ingeniería busca seleccionar el método más adecuado para un conjunto de circunstancias y, de esta manera, un acercamiento al desarrollo más creativo y menos formal sería efectivo en ciertas situaciones. El desarrollo menos formal es particularmente adecuado para la creación de sistemas basados en la Web, que requieren una mezcla de habilidades de software y diseño gráfico. (Sommerville, 2011)

Los ingenieros de software se concentran en el desarrollo del producto de software, existiendo dos tipos de productos:

1. Productos genéricos: son sistemas aislados producidos por una organización de desarrollo y que se venden al mercado abierto a cualquier cliente que le sea posible comprarlo.
2. Productos personalizados: son sistemas requeridos por un cliente en particular. Un contratista de software especialmente para ese cliente.

Para que un producto de software sea considerado de calidad se requiere cumplir con los siguientes criterios:

- Entender el problema de cliente antes de desarrollar una aplicación.
- Diseño adecuado.
- Cumplir con las normas de calidad.
- Tener la facilidad para recibir mantenimiento.

La ingeniería de software es muy importante para el desarrollo de un producto de calidad, pero consigo lleva un conjunto de metodologías y arquitecturas para que tenga un proceso de elaboración correcto.

La rama de ingeniería de software se preocupa en crear procesos que aseguren la calidad en los programas, la solución más utilizada es la organización del código del sistema.

Esto se puede lograr con el patrón de diseño que por nombre lleva Modelo, vista, controlador (MVC), que organiza en base a su función separando el código en tres capas.

En general, los ingenieros de software adoptan en su trabajo un enfoque sistemático y organizado, pues usualmente ésta es la forma más efectiva de producir software de alta calidad. No obstante, la ingeniería busca seleccionar el método más adecuado para un conjunto de circunstancias y, de esta manera, un acercamiento al desarrollo más creativo y menos formal sería efectivo en ciertas situaciones. El desarrollo menos formal es particularmente adecuado para la creación de sistemas basados en la Web, que requieren una mezcla de habilidades de software y diseño gráfico. (Sommerville, 2011)

2.2. Modelo Vista Controlador

MVC fue pensado en 1978 como una solución obvia al problema general, dando a los usuarios control sobre su información como se ve desde múltiples perspectivas.

El objetivo de nivel superior era apoyar el modelo mental del usuario, un espacio de información relevante para que pueda inspeccionar y editar esta información.

MVC ha creado una sorprendente cantidad de interés. Algunos incluso utilizan textos variantes con el fin de hacer frente a la computadora de control del usuario. MVC fue concebido como una solución general al problema de los usuarios, para el control de un conjunto de datos grandes y complejos. La parte más difícil fue dar con buenos nombres para los diferentes componentes arquitectónicos modelo-vista-editor fue el primer conjunto. (Reenskaug, 2007)

Este es uno de los patrones de diseño más antiguos originalmente desarrollado en el lenguaje Smalltalk, este un patrón muy común hoy en día, sobre todo con la proliferación de aplicaciones web y móviles. (Crookshanks, 2005)

Modelo: Representa los datos y la lógica de negocio del sistema, Los modelos representan el conocimiento. Un modelo podría ser un objeto único o podría ser algún tipo de estructura de los objetos.

Vista: la vista proporciona las entradas al sistema y muestra los resultados al cliente. Esto es por lo general algún tipo de interfaz gráfica de usuario entendible y cumpliendo las especificaciones del cliente.

Controlador: es el que acepta la entrada de la vista y solicita los datos del modelo; determina las acciones tanto para la vista y el modelo basado en la lógica programada.

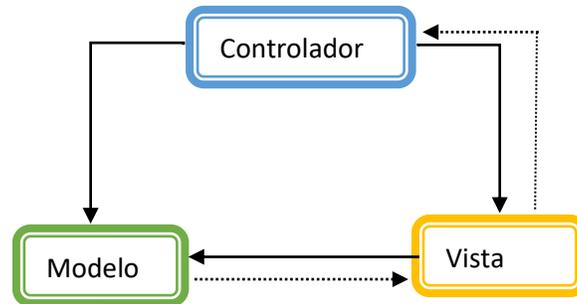


Figura 2.1 Arquitectura del patrón MVC (Crookshanks, 2005)

En la figura número. 2.1 se muestra la relación entre el modelo, vista y el controlador, donde las líneas solidas indican una asociación directa y las punteadas una indirecta. (Crookshanks, 2005)

La ventaja principal de este modelo es que la vista se separa del modelo y por lo tanto el diseño puede cambiarse sin afectar el modelo. Además, el modelo y controlador puede ser probado y verificado sin vistas a través de las pruebas unitarias y controladores simulados.

Esto es extremadamente beneficioso en el escenario actual de las aplicaciones distribuidas donde una aplicación de teléfono móvil, una aplicación web y un cliente de escritorio pueden conectarse con el mismo modelo de datos. El controlador, que se sienta delante del modelo, se puede analizar cada solicitar y optimizar la respuesta basada en el cliente. (Crookshanks, 2005)

2.3. Metodologías Ágiles

Ágil, como concepto se dio a conocer a finales de la década de 1990 a través de los esfuerzos para hacer frente a dificultades percibidas con los procesos de desarrollo de soluciones existentes que están arraigados, y debían su rigidez, impulsada por el plan de prácticas.

Por lo tanto, proyectos ágiles exhiben características de comunicación abierta entre actores heterogéneos, comportamiento emergente dentro de la auto-organización de los equipos y una cultura de apertura y aprendizaje. (Switzerland, 2015)

Uno de los grandes pasos dados en la industria del software fue aquel en que se plasmó el denominado modelo de cascada ya que sirvió como base para la formulación del análisis estructurado, el cual fue uno de los precursores en el camino hacia la aplicación de prácticas estandarizadas dentro de la ingeniería de software. Este modelo surge como respuesta al modelo para codificar y probar que era el que predominaba en la década de los sesenta. En esta época ya existían modelos iterativos e incrementales, pero no eran disciplinados ni estaban formalizados.

A principios de la década de los 90, surgió un enfoque que fue bastante revolucionario para su momento ya que iba en contra de toda creencia de que mediante procesos altamente definidos se iba a lograr obtener software en tiempo, costo y con la calidad requerida. El enfoque fue planteado por primera vez por James Martin (En su libro *Rapid application development*) en 1991 y se dio a conocer en la comunidad de Ingeniería de Software con el nombre de RAD o *Rapid Application Development*. RAD consistía en un entorno de desarrollo altamente productivo, en el que participaban grupos pequeños de programadores utilizando herramientas que generaban código en forma automática tomando como entradas sintaxis de alto nivel. En general, se considera que este fue uno de los primeros hitos en pos de la agilidad en los procesos de desarrollo. (A, S, V, & J., 2007)

Una metodología de desarrollo de aplicaciones consiste en una serie de fases y procesos que intervienen en aspectos administrativos y técnicos de forma iterativa,

que busca como fin la generación de un producto o una mejora del mismo, para el caso: una base de datos. Para efectos de desarrollo de bases de datos, existen las metodologías ágiles de desarrollo, que proponen de forma iterativa y rápida la generación de algún segmento o módulo de la base de datos; sin descuidar aspectos como planificación y pruebas. (Suarez, 2013)

2.4. Metodología SCRUM

SCRUM, se basa en la teoría del control empírico de los procesos, emplea un enfoque iterativo e incremental para optimizar la previsibilidad y controlar los riesgos. SCRUM surge en 1986, de un artículo de la *Harvard Business Review* titulado “*The New Product Development Game*” de Hirotaka Takeuchi e Ikujiro Nonaka, que introducía las mejores prácticas más utilizadas en 10 compañías japonesas altamente innovadoras. A partir de ahí y tomando referencias al juego de rugby, Ken Schwaber y Jeff Sutherland formalizan el proceso conocido como SCRUM en el año 1996. (SCRUMGUIDES, 2016)

SCRUM puede describirse mejor como una metodología de desarrollo de producto con leves aspiraciones de gestión de proyectos (por ejemplo, seguimiento y generación de informes de desarrollo) como su enfoque radica en la gestión de requisitos de software y desarrollo; SCRUM es un modelo de desarrollo ágil caracterizado por:

Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.

Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.

Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada. (Manager, 2015)

Una de las principales características de SCRUM es que en cada iteración todas las etapas de la creación de un producto se solapan. En cada Sprint se realiza la planificación, análisis, creación y comprobación de lo que se va a entregar al final

del mismo. Divide el trabajo en pequeños entregables, priorizados y estimulados al esfuerzo.

El marco técnico de SCRUM está formado por:

- Roles, representan una responsabilidad en el proceso y no a la posición dentro de la organización:
 - El dueño del producto: es el responsable desde el punto de vista del negocio
 - El dueño del producto.
 - El SCRUM Master.
- Artefactos:
 - Pila del producto.
 - Pila del sprint.
 - incremento.
- Eventos
 - Sprint.
 - Reunión de planificación del sprint.
 - SCRUM diario.
 - Revisión del sprint.
 - Retrospectiva del sprint.

SCRUM es un marco de trabajo de procesos que ha sido utilizado para gestionar el desarrollo de productos complejos desde principios de los años 90. SCRUM no es un proceso o una técnica para construir productos, sino un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas, hace patente la eficacia relativa de las prácticas de gestión de producto y de desarrollo, de modo que se puedan mejorar.

2.4.1. Proceso de SCRUM

El proceso de SCRUM se puede identificar por tres fases que son:

- 1) Planificación del sprint
- 2) Seguimiento del Sprint

3) Revisión del sprint.

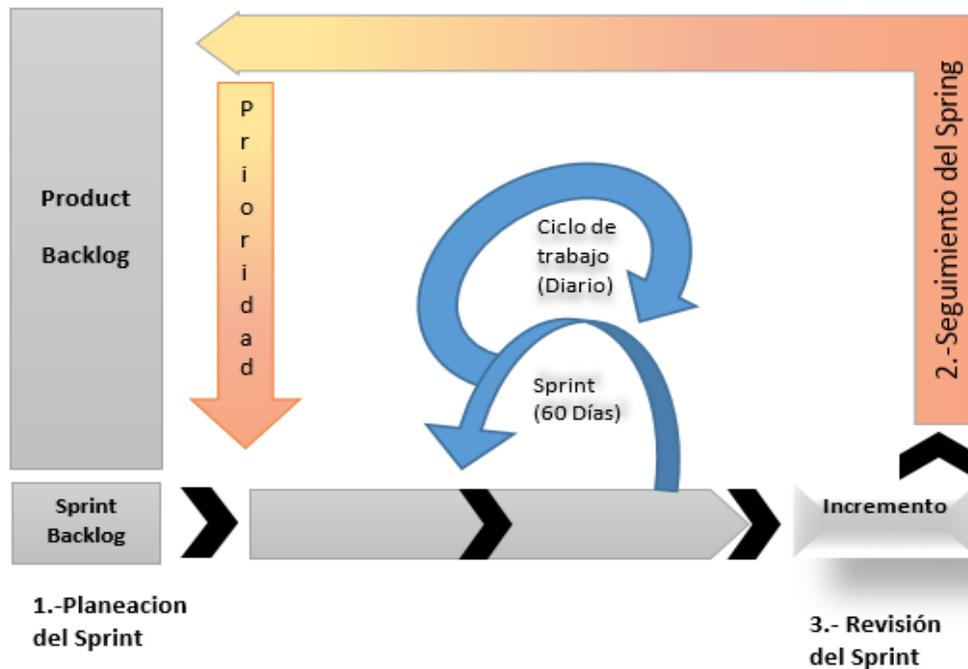


Figura 2.2 Fases de scrum (A, S, V, & J., 2007).

Planeación del Sprint: en esta fase se describe la pila del producto, donde se enlistan los requerimientos del sistema, donde en cada iteración es revisada por los miembros del equipo.

Seguimiento del Sprint: En esta fase se evalúa el avance de las tareas y el trabajo que está previsto para cierto periodo a través de reuniones. En estas reuniones estarán presentes el SCRUM Master y el equipo.

Revisión del Sprint: Los miembros del equipo y los clientes se reúnen para mostrar el trabajo de desarrollo de software que se ha completado. Se hace una demostración de todos los requerimientos finalizados dentro del Sprint. En este punto no es necesario que todos los miembros del equipo hablen. Pueden estar presentes pero la presentación está a cargo del SCRUM Master y el *Product Owner*. (A, S, V, & J., 2007).

2.5. Modelo de datos

El modelo de datos define la lógica de negocio (la base de datos es la perteneciente en esta capa).

Es una técnica para la organización y la documentación de los datos en el sistema, donde el resultado de un modelo de datos permite crear de forma rápida y sencilla una base de datos que cumpla con las necesidades de almacenamiento de datos.

Un modelo de datos se compone en el diseño conceptual, diseño lógico y diseño físico definiéndose como:

El **diseño conceptual** se parte de las especificaciones del usuario y se consiguen una representación del mundo real, en donde se describen las entidades y sus propiedades, además de las relaciones entre ellos.

El **diseño lógico** consiste en transformar el modelo conceptual obtenido en otro esquema que se puede procesar el sistema gestor de bases de datos. Por ejemplo, el modelo entidad- relación y el modelo relacional.

El **diseño físico** se parte del diseño lógico y da como resultado el esquema físico. Consiste en la implementación del modelo de relacional, dando lugar a la estructura de datos de almacenamiento en uno o varios soportes físicos. (Abraham Silberschatz, 2006).

2.6. Base de datos

El término bases de datos fue escuchado por primera vez en un simposio celebrado en California en 1963. Una base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos. (Valdés, 2016)

El uso de las bases de datos se desarrolló a partir de las necesidades de almacenar grandes cantidades de información o datos. Sobre todo, desde la aparición de las primeras computadoras, el concepto de bases de datos ha estado siempre ligado a la informática.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro. Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. (Valdés, 2016).

2.6.1. Tipo de base de datos

Las bases de datos pueden clasificarse de varias maneras, la utilidad de la misma. Según la variabilidad de los datos almacenados:

Bases De Datos Estáticas OLAP (*On Line Analytical Processing*). Estas son bases de datos de solo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones, solo se realizan consultas sobre los datos ya existentes para el análisis y toma de decisiones.

Este tipo de bases de datos son implementadas en *Business Intelligence* para mejorar el desempeño de las consultas con grandes volúmenes de información.

Bases De Datos Dinámicas OLTP (*On Line Transaction Processing*). Estas son bases de datos donde la información almacenada se modifica con el tiempo o en tiempo real, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta.

Un ejemplo es el sistema de un supermercado donde se van registrando cada uno de los artículos que el cliente está comprando y a su vez el sistema va actualizando el Inventario. (Morales, 2015).

2.6.2. Modelo de base de datos

En la actualidad el modelo relacional es el más usado, aunque existen otros modelos.

Modelo relacional: Este modelo es desarrollado por Codd¹, la representación lógica de las entidades y sus relaciones se representan en tablas bidimensionales.

Se llamará registro o tupla a cada fila de la tabla y campo o atributo a cada columna de la tabla. Uno de los requisitos de las tablas será que no puede haber tuplas repetidas. Una clave será un atributo o conjunto de atributos que identifique de forma única a una tupla. Las tablas deben de cumplir los siguientes requisitos.

Para manejar estas tablas se utilizan operaciones clásicas de la teoría de conjuntos (unión, intersección, diferencia y producto cartesiano) así como operaciones específicas del modelo relacional (Selección, proyección, reunión y división) (Cobo, 2014)

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

¹ **Edgar Frank Codd** científico informático inglés, conocido por crear el modelo relacional de bases de datos

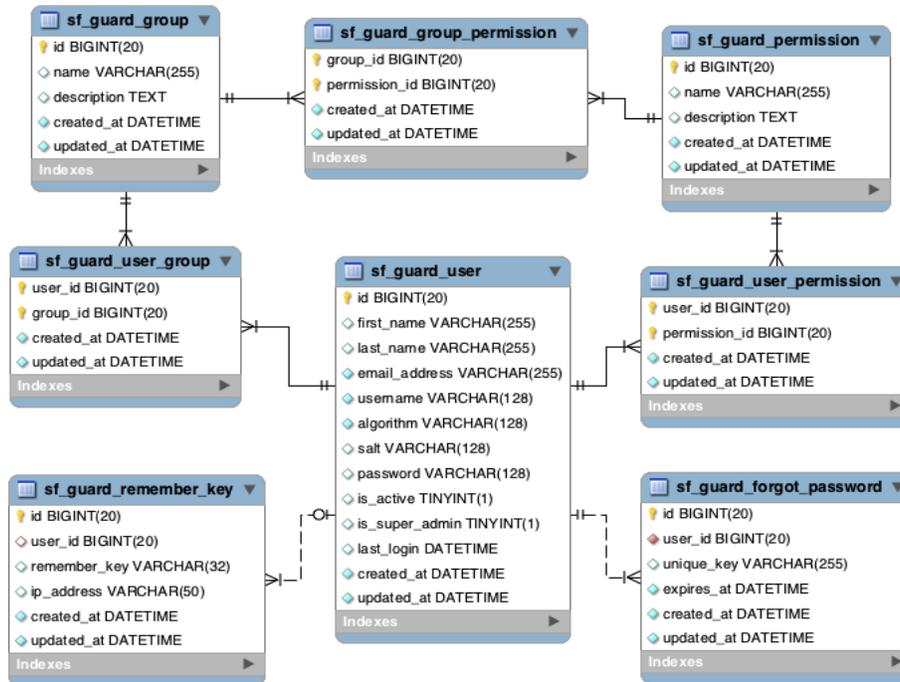


Figura 2.3 Base de datos relacional (Cobo, 2014)

En la figura número. 2.3 se observa un ejemplo de una base de datos relacional elaborada en workbench y los tipos de cardinalidad, uno a muchos o uno a uno.

Modelo jerárquico: estas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que tiene padre es llamado raíz y a los nodos que tienen hijos se les conocen como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos compartidos permitiendo crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

En la figura número. 2.4 se muestra un ejemplo de una base de datos jerárquica, donde se visualiza que la tabla curso que encabeza y los hijos de esta son oferta, requisito que a su vez se alimenta de profesor y estudiante.

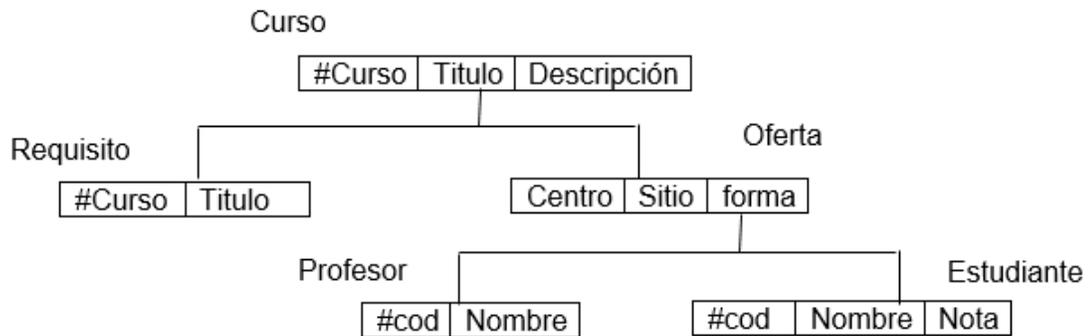


Figura 2.4 Base de datos Jerárquica (Cobo, 2014)

Algunas características de este modelo son:

- Los registros están dispuestos en forma de árbol y no pueden existir ciclos
- Los registros solo pueden estar relacionados mediante relaciones uno a uno o uno a muchos
- Cuando se elimina un registro padre se borran todos sus hijos. (Cobo, 2014)

Modelo multidimensional: Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de cubos OLAP (*OnLine Analytical Processing* o Procesamiento Analítico en Línea), básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

Se pueden visualizar estructuras multidimensionales como cubos de datos dentro de cubos de datos. Cada cara del cubo se considera una dimensión de los datos.

En la figura número 2.5 se muestran que en cada dimensión se puede representar una categoría, como productos, clientes, fecha.

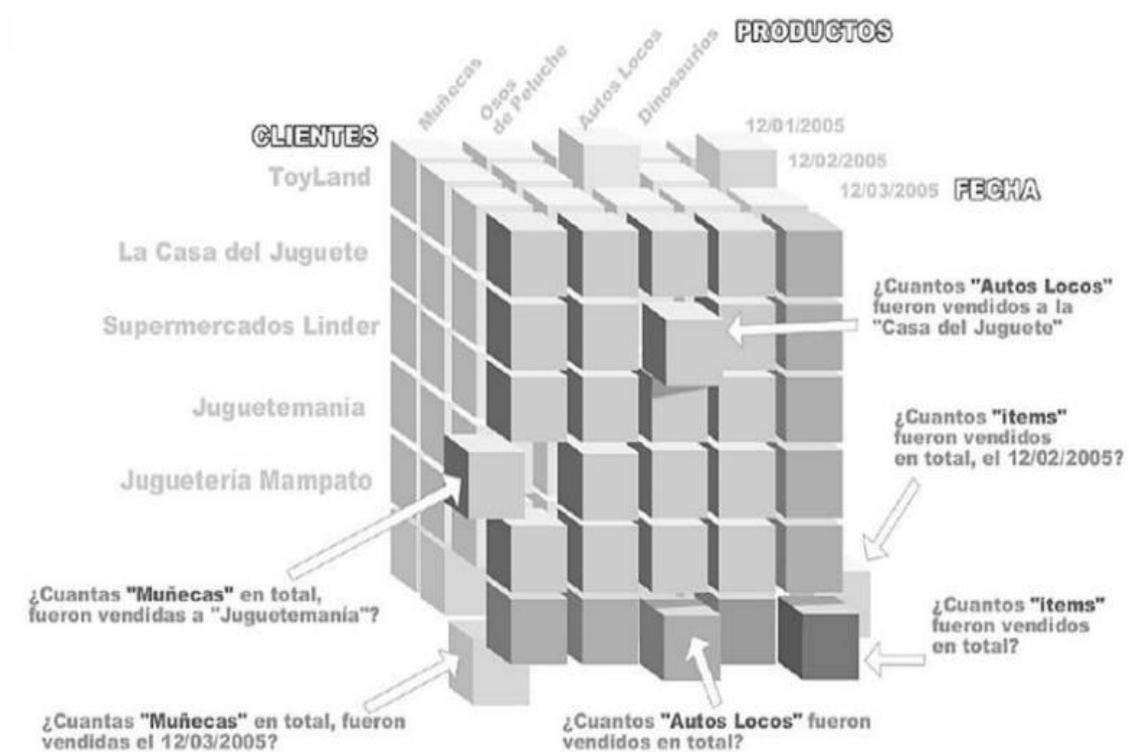


Figura 2.5 Base de datos multidimensional (wordpress, 2016)

Modelo de red de base de datos. Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de *nodo*: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

Estas bases de datos basan su comunicación principalmente en lo que es la arquitectura cliente servidor, para poder tener una comunicación dentro de una red LAN (*Local Area Network* o Red de Área Local) y poder compartir información con

los equipos de los usuarios que se encuentren conectados dentro de la red, en la figura 2.6 se muestra gráficamente el modelo en red. (Suarez, 2013).



Figura 2.6 Modelo en red de base de datos (wordpress, 2016)

2.7. Sistemas manejadores de base de datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (Avila, 2016)

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

2.7.1. Características

Algunas de las características con las que cuenta un sistema gestor de base de datos son:

Abstracción de la información. Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

Independencia. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

Redundancia mínima. Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

Seguridad. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

Integridad. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

Respaldo y recuperación. Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

Control de la concurrencia. En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

2.7.2. ACID

En bases de datos se denomina ACID a las características de los parámetros que permiten clasificar las transacciones de los sistemas de gestión de bases de datos. (Wikimedia, 2016)

Atomicidad: Si una operación consiste en una serie de pasos, todos ellos ocurren o ninguno, es decir, las transacciones son completas.

Consistencia: Es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto, se ejecutan aquellas operaciones que no van a romper las reglas y directrices de Integridad de la base de datos. La propiedad de consistencia sostiene que cualquier transacción llevará a la base de datos desde un estado válido a otro también válido. "La Integridad de la Base de Datos nos permite asegurar que los datos son exactos y consistentes, es decir que estén siempre intactos, sean siempre los esperados y que de ninguna manera cambien ni se deformen.

De esta manera podemos garantizar que la información que se presenta al usuario será siempre la misma."

Aislamiento: es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información sea independiente y no generen ningún tipo de error. Esta propiedad define cómo y cuándo los cambios producidos por una operación se hacen visibles para las demás operaciones concurrentes. El aislamiento puede alcanzarse en distintos niveles, siendo el parámetro esencial a la hora de seleccionar SGBDs.

Durabilidad: Es la propiedad que asegura que, una vez realizada la operación, ésta persistirá y no se podrá deshacer, aunque falle el sistema y que de esta forma los datos sobrevivan de alguna manera. (Wikimedia, 2016).

2.8. Herramientas de mapeo

2.8.1. Hibernate

Hibérnate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL (Hibernate, 2016).

2.8.1.1. Características

Es software libre, distribuido bajo los términos de las licencias GNU y Licencia Pública General Reducida (LGPL por sus siglas en inglés, *Lesser General Public License*) y uno de sus mayores atractivos lo constituye el manejo de consultas y su trabajo con bases de datos.

Hibernate, como todas las herramientas de su tipo busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la Programación Orientada a Objetos (OOP por sus siglas en inglés, *Object Oriented Programming*).

Hibérnate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL, generará las sentencias de este y liberará al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la

portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Este ofrece también un lenguaje de consulta de datos llamado *Hibernate QueryLanguage* (HQL por sus siglas en inglés), al mismo tiempo que una *Application Programming Interface* (API por sus siglas en inglés, Interfaz de programación de aplicaciones) para construir las consultas programáticamente (conocida como "criteria"). (Hibernate, 2016)

2.8.1.2. Ventajas

- a) Nos permite desarrollar mucho más rápido.
- b) Permite trabajar con la BD por medio de entidades en vez de *Querys*.
- c) Nos ofrece un paradigma 100% orientado a objetos.
- d) Elimina errores en tiempo de ejecución.
- e) Mejora el mantenimiento del software. (Hibernate, 2016)

2.8.2. Java Persistence API

Java Persistence API, más conocida por JPA es desarrollado para la plataforma Java EE, su primer lanzamiento fue 11 de mayo del 2006 con la versión 1.0 esta fue parte del proceso de la comunidad de java (Computing, 2016).

La versión JPA 2.0 fue lanzada el 10 de diciembre del 2009 mientras las de JPA 2.1 el 22 de abril del 2013. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos siguiendo el patrón de mapeo objeto-relacional.

La Java Persistence API proporciona a los desarrolladores de Java un recurso de mapeo de objetos / relaciones para administrar datos relacionales en aplicaciones Java. La persistencia de Java consta de cuatro áreas:

- La API de persistencia de Java.
- El lenguaje de consulta.
- La API de criterios de persistencia de Java.
- Metadatos de asignación de objetos / relacional.

Tabla 2.1 Comparativa Hibernate vs Java Persistence API

Característica	Hibernate	Java Persistence API
Estándar	Hibernate es una implementación de un estándar.	Cada proveedor debe proporcionar una implementación.
Portabilidad	Depende de JBOSS (Un servidor de aplicaciones java EE).	Depende del JBOSS.
Rendimiento	Los resultados de hibernate siempre están entre los mejores.	Aún no existen suficientes comparativas y resultados frente a las otras opciones.
Flexibilidad	Gran capacidad de trabajar en varios servidores.	Gran flexibilidad a la hora de elegir nuestro servidor.
Base de datos	Capacidad para trabajar con bases de datos NoSQL	No tienen la capacidad para trabajar con NoSQL.

2.9. Arquitectura cliente – servidor

En un sistema cliente-servidor, el usuario interactúa con un programa que se ejecuta en su computadora local (por ejemplo, un navegador Web o una aplicación basada en telefonía). Éste interactúa con otro programa que se ejecuta en una computadora remota.

La computadora remota proporciona servicios, como acceso a páginas Web, que están disponibles a clientes externos. Este modelo cliente-servidor es un modelo arquitectónico muy general de una aplicación. No está restringido a aplicaciones distribuidas a través de varias máquinas. También se puede usar como un modelo de interacción lógica donde cliente y servidor operan en la misma computadora.

En una arquitectura cliente-servidor, una aplicación se modela como un conjunto de servicios que proporcionan los servidores. (Sommerville, 2011).

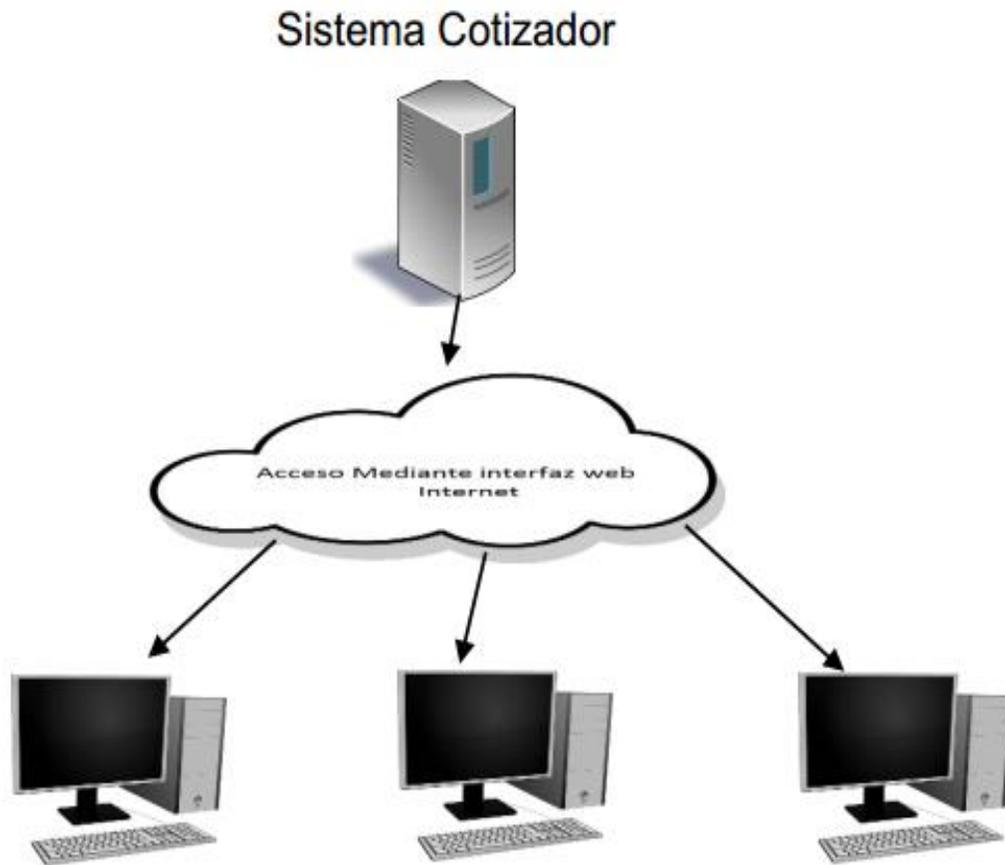


Figura 2.7 Arquitectura cliente servidor (Sommerville, 2011).

2.9.1. Ventajas

- Permite que la información se procese cerca de donde se ha generado
- Dado que las funciones del software quedan repartidas entre varias máquinas, es posible utilizar PC, o estaciones de trabajo para los clientes
- El crecimiento del hardware puede ser gradual:
- Facilita el uso de interfaces graficas de usuario y aplicaciones multitarea.

2.10. Modelo V

En este ambiente utilizaremos y llevaremos a cabo el modelo “V”, el cual es un modelo que tiene los siguientes objetivos:

- Minimizar los riesgos del proyecto.
- Mejorar y garantizar la calidad del proyecto.

En general es un modelo robusto y completo, y puede producir software de mayor calidad.

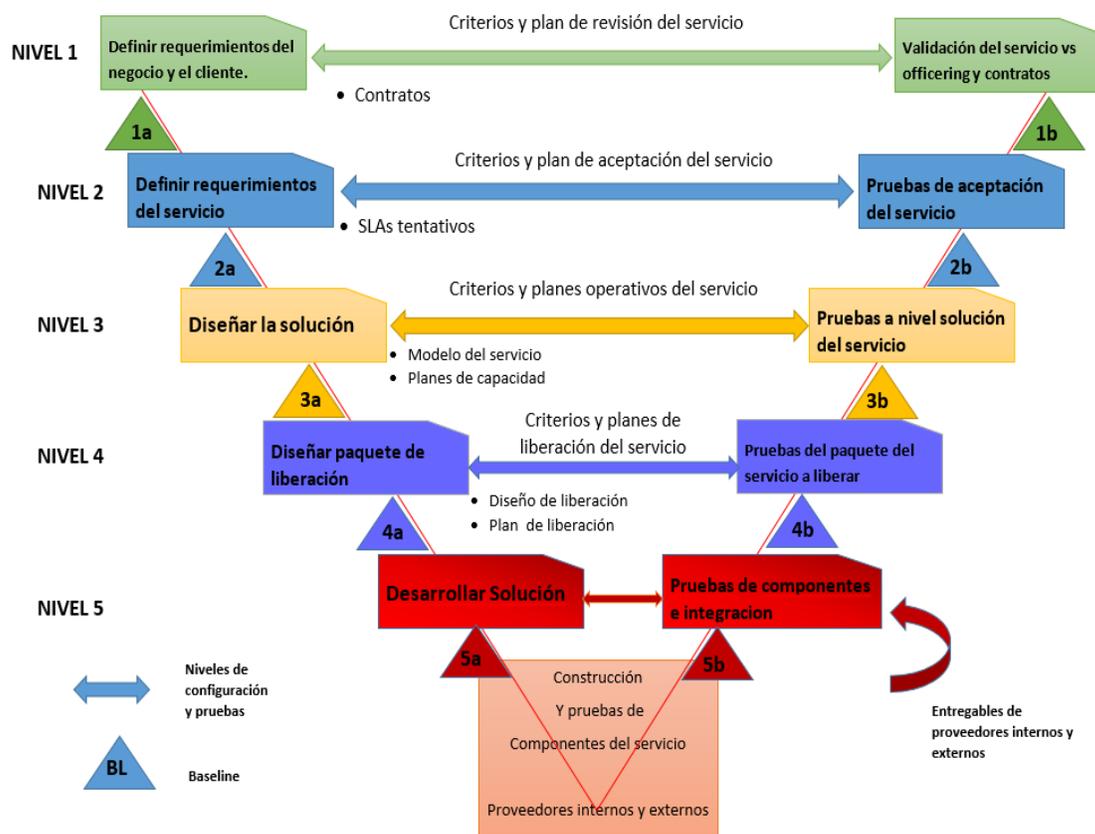


Figura 2.8 Modelo V (Martinez & Barrio, 2016)

En la figura número 2.8 se muestran las fases del modelo, del cual se toma de base para poder crear nuestro propio modelo, dependiendo de las necesidades y especificaciones del software, a continuación, explicaremos el modelo:

- Lado izquierdo: especificaciones de los requerimientos del servicio hasta el detalle del diseño del servicio.
- Lado derecho: se focaliza en las actividades de validación que se llevan a cabo en contra de las especificaciones definidas a la izquierda.
- A cada paso de la izquierda, hay implicación directa con la parte equivalente en el lado derecho.
- Dentro de cada ciclo de desarrollo repetitivo se pueden aplicar los conceptos del modelo V sobre los requerimientos de aprobación de estabilidad contra el diseño, con cada diseño repetitivo considerado para el grado de integridad y competencia que justificara el lanzamiento al cliente para juicio y valoración.

Dentro del modelo “V” están las siguientes pruebas:

- Pruebas de Componente: el cual tiene como objetivo localizar defectos y probar el funcionamiento del software, por ejemplo, datos válidos y erróneos al ingresar al sistema, y como responde el sistema ante dichas pruebas.
- Pruebas de Integración: En cual se debe checar detalladamente el flujo de la información entre los módulos.
- Pruebas de Sistema: en el cual se debe de revisar el funcionamiento del software, donde tiene que constar que cumpla con los requisitos funcionales y no funcionales.

Pruebas de Aceptación: En esta prueba es donde, el cliente nos confirma si el software cumple con todo lo que pidió y nos confirma de igual manera si es exitoso o no.

2.11. Evaluación del software

Durante el desarrollo hasta el término del sistema se debe realizar una serie de técnicas de evaluación del software, para detectar faltas y fallos.

En la fase de pruebas de ciclo de vida del software hay diferentes pruebas para medir la calidad de sistema que se va a implementar.

2.11.1. Pruebas de software

Hacer pruebas es una actividad que tiene el objetivo de evaluar y mejorar la calidad del producto, identificando defectos y problemas (Abram & Moore, 2004).

Las pruebas del software consisten en verificar el comportamiento de un programa dinámicamente a través de un grupo finito de casos de prueba, debidamente seleccionados, típicamente ámbito de ejecuciones infinito y en la relación al comportamiento esperado. En particular:

- **Dinámicamente:** Significa que hacer pruebas siempre supone ejecutar el programa con entrada de datos (valorados). Para precisar, es preciso afirmar que la entrada de valores no es siempre suficiente para definir una prueba, dado que un sistema complejo y no determinista podría tener diferentes comportamientos con las mismas entradas de datos, dependiendo del estado en el que se encuentre.
- **Finito:** Incluso en programas sencillos, teóricamente podría haber tantas pruebas que realizar, que hacer pruebas exhaustivas podría llevar meses o años. Hacer pruebas siempre supone un compromiso entre recursos y calendarios de trabajo limitados.
- **Seleccionados:** LA diferencia esencial entre las distintas técnicas de pruebas propuestas se encuentra en cómo se escoge el conjunto de pruebas. La forma de identificar el criterio de selección de pruebas más del mantenimiento del software aborda los temas relacionados con el mantenimiento del software. (Abram & Moore, 2004)

2.11.1.1. Pruebas de integridad

Las pruebas de integridad de base de datos son pruebas de los métodos y procesos utilizados para acceder y gestionar datos, para asegurar que los métodos de acceso, los procesos y las reglas de los datos funcionan como se espera y durante el acceso a la BD los datos no se corrompan, sean borrados, modificados o creados de forma inesperada (Globe Testeng, 2017).

2.11.1.2. Restricciones de integridad

Proporcionan un medio de asegurar que los cambios que se hacen en la base de datos, no resulten en una pérdida de la consistencia de los datos (Abraham Silberschatz, 2006).

Existen tres tipos de restricciones de integridad:

- De dominio.
- De las entidades.
- Referencial.

Restricciones de dominio

Las restricciones de dominio especifican que el valor de cada atributo A debe ser un valor atómico del dominio $dom(A)$ para ese atributo. Los tipos de datos asociados a los dominios por lo general incluyen los tipos de datos numéricos estándar de los números enteros (como entero- corto, entero, entero-largo) y reales (flotante y flotante de doble precisión) (Elmasri & Navathe, 2007).

Restricciones de entidad

Las restricciones de integridad de entidad declaran que el valor de ninguna clave principal puede ser NULL. Esto se debe a que dicha clave se emplea para identificar tuplas individuales en una relación.

Por ejemplo, si dos o más tuplas tuvieran NULL en sus claves primarias, no seríamos capaces de diferenciarlas si intentamos hacer referencia a ellas desde otras relaciones (Elmasri & Navathe, 2007).

Restricciones de integridad referencial.

Las restricciones de integridad referencial están especificadas entre dos relaciones y se utiliza para mantener la consistencia entre tuplas de dos relaciones.

A menudo se desea asegurar que un valor que aparece en una relación para un conjunto de atributos determinado aparezca también en otra relación para un cierto conjunto de atributos (Elmasri & Navathe, 2007).

Integridad referencial en SQL

Las claves externas pueden especificarse como parte de la instrucción *create table* de SQL usando la cláusula *foreign key*. De manera predeterminada, una clave externa referencia los atributos que forman la clave primaria de la tabla referenciada (SQL también soporta una versión de la cláusula *references*, donde se puede especificar explícitamente una lista de atributos de la relación referenciada). Se usa la siguiente sintaxis para declarar que un atributo forma una clave externa:

foreign key (A1[, A2, . . . , An])

references R

Cuando se viola una restricción de integridad referencial, el procedimiento normal es rechazar la acción que provocó la violación. Sin embargo, la cláusula *foreign key* puede especificar que si una acción de borrado o de actualización de la relación a la que hace referencia viola la restricción, en lugar de rechazar la acción, hay que adoptar medidas para modificar la tupla de la relación que hace la referencia con objeto de restaurar la restricción. Considérese la siguiente definición de una restricción de integridad de la relación cuenta:

create table cuenta (...

foreign key (nombre-sucursal) **references** sucursal **on delete cascade on update cascade, ...)**

La cláusula ***on delete cascade*** asociada con la declaración de la clave externa, provoca que el borrado se realice en cascada. La cláusula ***on update cascade***, de forma similar realiza una actualización en cascada, si se modifica la clave primaria.

SQL también permite una acción diferente: establecer a nulo o darles un valor predeterminado a los atributos de la clave externa, de forma que se mantenga la integridad referencial, con la cláusula ***set default***.

Si hay una cadena de dependencias de claves externas entre varias relaciones, un borrado o una actualización en uno de sus extremos puede propagarse por toda la cadena, de ahí la denominación de actualizaciones o borrados en cascada (Elmasri & Navathe, 2007).

2.12. Estándar de evaluación de la calidad de software.

Por el gran avance tecnológico es necesaria la implementación de estándares de calidad para ofrecer un mejor producto.

Las aplicaciones de software deben ser evaluadas para descartar toda falla que pueda ser catastrófica, para ayudar a combatir todo tipo de error se cuenta con el estándar ISO/IEC 25000 es un estándar internacional con el objetivo de organizar, enriquecer y unificar las series que cubren dos procesos principales: especificación de requisitos de calidad del software y evaluación de la calidad del software soportada por la ISO 9126.

El estándar está dividido en cuatro partes las cuales dirigen, la realidad, métricas externas, métricas internas y calidad en métricas de uso.

En la (ISO/IEC 9126, 2001) se tiene la clasificación con un conjunto de características que se tomara en cuenta para la evaluación en las pruebas de este sistema.

- **Funcionalidad:** Es un conjunto de atributos que se relacionan con funciones y sus propiedades específicas.

- **Fiabilidad:** Es la capacidad del sistema para mantener su nivel de prestación bajo condiciones, es la probabilidad de que funcionalidad libre de fallos.
- **Eficiencia:** se refiere al conjunto de atributos que se relacionan entre el nivel de rendimiento del software y la cantidad de recursos utilizados bajo unas condiciones predeterminadas.
- **Usabilidad:** un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
- **Portabilidad:** la capacidad de un sistema para ser transferido de un entorno a otro.

2.13. Industria maderera y mercadeo

La industria maderera es el sector de la actividad industrial que se ocupa del procesamiento de la madera, desde su plantación hasta su transformación en objetos de uso práctico, pasando por la extracción, corte, almacenamiento o tratamiento bioquímico y moldeo. El producto final de esta actividad puede ser la fabricación de mobiliario, materiales de construcción o la obtención de celulosa para la fabricación de papel, entre otros derivados de la madera (Wikipedia, 2016).

Los productos de la industria maderera contribuyen a satisfacer algunas de las necesidades básicas de la sociedad moderna y a promover el bienestar del ser humano.

2.14. Venta de madera industrial

La madera es un producto proveniente de árboles, una materia para la fabricación de mobiliario, construcción de viviendas y una gran variedad de utensilios para diversos usos. La venta de estos productos depende a las demandas de diversos productos (vigas, vigas curvas, vigas I etc.) Se tiene una responsabilidad para el encargado de ventas ya que tiene en su responsabilidad contactar y convencer al cliente para vender los productos, Cada vez los proyectos que producen son más

grandes por lo que el encargado realiza cotizaciones en tiempo y forma. El proceso de venta de la madera industrial sigue los siguientes pasos:

- Contactar al cliente que necesita el producto
- Solicitar los requerimientos
- Mandar una pre cotización
- Hacer una visita, y corroborar información
- Visualizar el grado de dificultad
- Generar la cotización final.

Capítulo 3

Metodología

En este capítulo se describe la metodología utilizada para el desarrollo del modelo sistema de cotización, explicando cada una de las etapas realizada para su creación y manejo de información para la generación de una cotización.

3.1. Metodología de implementación

La prioridad de este trabajo es el desarrollo de un modelo de base de datos para un sistema de cotización, con el objetivo de reducir el trabajo de cotización, proporcionando un modelo para almacenar la información necesaria para generar la cotización y así como el resultado de la misma, tratando de brindar seguridad y eficacia en el manejo de los datos.

Se hace una combinación de la arquitectura MVC (Modelo Vista Controlador) y una metodología SCRUM para el desarrollo del modelo de forma específica.

La empresa de venta de madera industria tiene el departamento de ventas, que se encarga de salir y convencer al cliente a través de una cotización de sus productos. Para realizar este proceso se necesitan datos (información del cliente, materiales, personal, tratamientos, costos, fechas de entrega, total aproximado de días de trabajo, y el nombre de la personas que realiza la cotización), de modo que toda esta información se encuentra almacenada solo en un equipo de la persona que realiza la cotización, de igual manera el tiempo en que se realiza esta operación es muy tardado por la realización de operaciones, así que puede tardar hasta una semana para la entrega correcta de la misma cotización.

Observando esta problemática se toma en cuenta para el desarrollo de un sistema cotizador para la venta de madera industrial, generando un modelo de base de datos para el almacenamiento de datos e información.

3.2. Modelo vista controlador

En la figura número 3.1 se visualiza el patrón MVC con la separación entre los tres módulos, el módulo de vista, de control y el modelo mostrando la secuencia de interacción entre ambas, así mismo las herramientas que se emplean para la elaboración del sistema.

El patrón MVC facilita el desarrollo y funcionalidad del sistema ya que trabaja de forma independiente en cada módulo sin dejar a lado la funcionalidad.



Figura 3.1 Patrón de diseño con MVC (Fuente propia)

Modelo: Dentro de la secuencia del patrón de diseño el modelo proporciona la base de datos, la cual contiene la información que el controlador necesita para realizar operaciones de inserción, eliminación, consulta o actualización datos.

Este módulo contiene información referente a: clientes, usuarios, materiales, productos, tratamientos, acabados, maderas, servicios y cotizaciones.

Vista: En esta capa representa al modelo en un estado adecuado para el usuario, representa los procedimientos que el usuario realizar, le da facilidad de manejar y utilizar los datos que necesita para realizar alguna operación. Brindar la facilidad de realizar las operaciones invocando al controlador y a su vez al modelo. En la vista se presenta la interfaz gráfica de usuario donde permite acceder o salir del sistema, generar una nueva cotización, insertar o administrar datos, así como visualizar información deseada por el cliente.

Controlador: El controlador es el elemento más importante ya que gestiona y administra los eventos enviados por las interfaces y el modelo. Las interfaces hacen una petición a los JavaBean en java, estos se conectan y realizan las tareas necesarios mediante hibernate a la base de datos, la base de datos proporciona la información obtenida a hibernate, el cual responde a los JavaBean y estos a las vistas para las funciones de agregar, modificar, eliminar o buscar materiales, clientes, instalación y servicios, asimismo realiza el cálculo de las distintas operaciones correspondientes para la realización de una cotización, gestiona el permiso y acceso de cada usuario, y genera las búsquedas correspondientes así como los reportes.

3.3. SCRUM para el modelo de base de datos

Conforme al análisis correspondiente, se opta por la utilización de la metodología SCRUM, ya que se trata de una metodología ágil y flexible para la elaboración del sistema, que permite la planificación del producto de software, su seguimiento, verificación y termino.

Así como también permite realizar modificación de requerimientos funcionales o cambiar la prioridad en el inicio de cada sprint sin ningún problema.

En la figura núm. 3.2 se muestra la metodología de desarrollo basada en SCRUM, donde está conformada por la pila del producto que es donde se desglosa la entrevista con el cliente, la recopilación de requisitos a partir de las entrevistas

contenidas en la pila del sprint, se realiza el análisis, diseño, codificación, evaluación y las pruebas, teniendo iteraciones cada 30 días.

Durante la fase de implementación se verificará el funcionamiento del producto y una vez aprobada se procede a la entrega del proyecto.

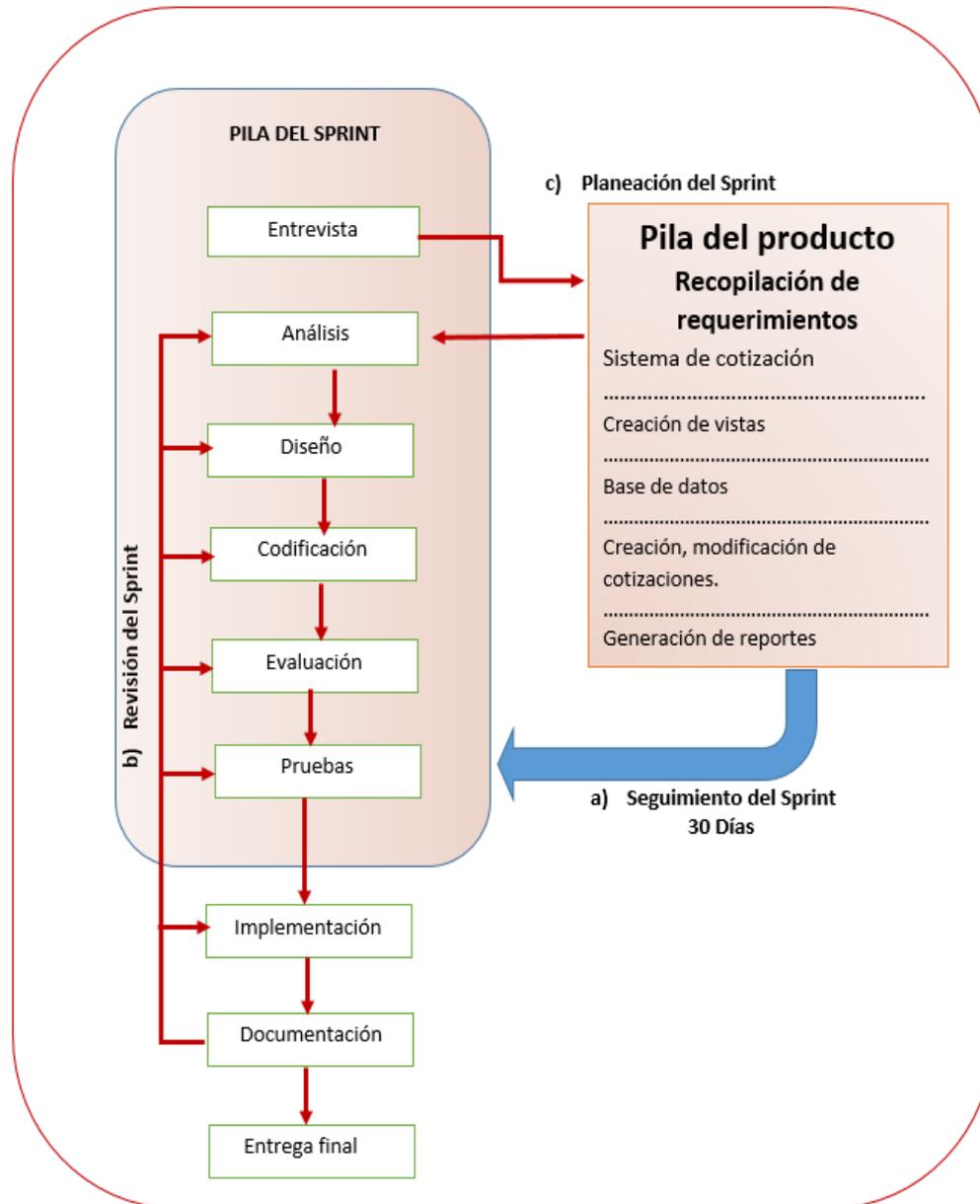


Figura 3.2 Metodología para el desarrollo del sistema cotizador basado en SCRUM (Suarez, 2013)

3.3.1. Planeación

Pila del producto: En este apartado especificamos los requerimientos funcionales y no funcionales, donde se tratan un conjunto de requerimientos para la funcionalidad correcta del sistema.

Pila del sprint: Está conformado por las entrevistas, análisis, diseño, codificación, evaluación y las pruebas, para realizar y cubrir con las necesidades de la pila del producto.

Tabla 3.1 Requerimientos para la pila de producto

Número	Nombre	Prioridad	Complejidad
001	Autenticar usuarios	Esencial	Complejo
002	Alta de usuarios	Esencial	Complejo
003	Listar usuarios	Útil	Mediano
004	Modificar usuarios	Esencial	Complejo
005	Eliminar usuarios	Esencial	Complejo
006	Alta de clientes	Esencial	Complejo
007	Listado de clientes	Útil	Mediano
008	Modificar de clientes	Esencial	Complejo
009	Eliminar de clientes	Esencial	Complejo
010	Alta de productos	Esencial	Complejo
011	Listar productos	Útil	Mediano
012	Modificar productos	Esencial	Complejo
013	Eliminar productos	Esencial	Complejo
014	Alta de materiales	Esencial	Complejo
015	Listar materiales	Útil	Mediano
016	Modificar materiales	Esencial	Complejo
017	Eliminar materiales	Esencial	Complejo
018	Alta de madera	Esencial	Complejo
019	Listar madera	Útil	Mediano
020	Modificar madera	Esencial	Complejo

Tabla 3.1 Requerimientos para la pila de producto (continuación)

Número	Nombre	Prioridad	Complejidad
021	Eliminar de madera	Esencial	Complejo
022	Alta de tratamiento	Esencial	Complejo
023	Listar tratamiento	Útil	Mediano
024	Modificar tratamiento	Esencial	Complejo
025	Eliminar tratamiento	Esencial	Complejo
026	Alta de acabado	Esencial	Complejo
027	Listar de acabado	útil	mediano
028	Modificar acabado	Esencial	Complejo
029	Eliminar acabado	Esencial	Complejo
030	Alta de personal	Esencial	Complejo
031	Listar de personal	útil	mediano
032	Modificar de personal	Esencial	Complejo
033	Eliminar de personal	Esencial	Complejo
034	Reportes número de cotizaciones	Esencial	Complejo
035	Reporte de numero de pie tablón vendidos	Esencial	Complejo
036	Generar cotización	Esencial	Muy complejo
037	Listar cotización	Esencial	Complejo
038	Modificar cotización	Esencial	Complejo
039	Eliminar cotización	Esencial	Complejo

3.4. Metodología para la elaboración del modelo de base de datos

Conforme a la arquitectura del patrón de diseño MVC, este trabajo comprende a la capa del modelo, en donde se almacenará información correspondiente a cotizaciones realizadas, así como datos importantes para la elaboración de la misma.

Tomando la metodología SCRUM se desprende el modelo para el desarrollo de la base de datos mostrada en la figura número 3.3.

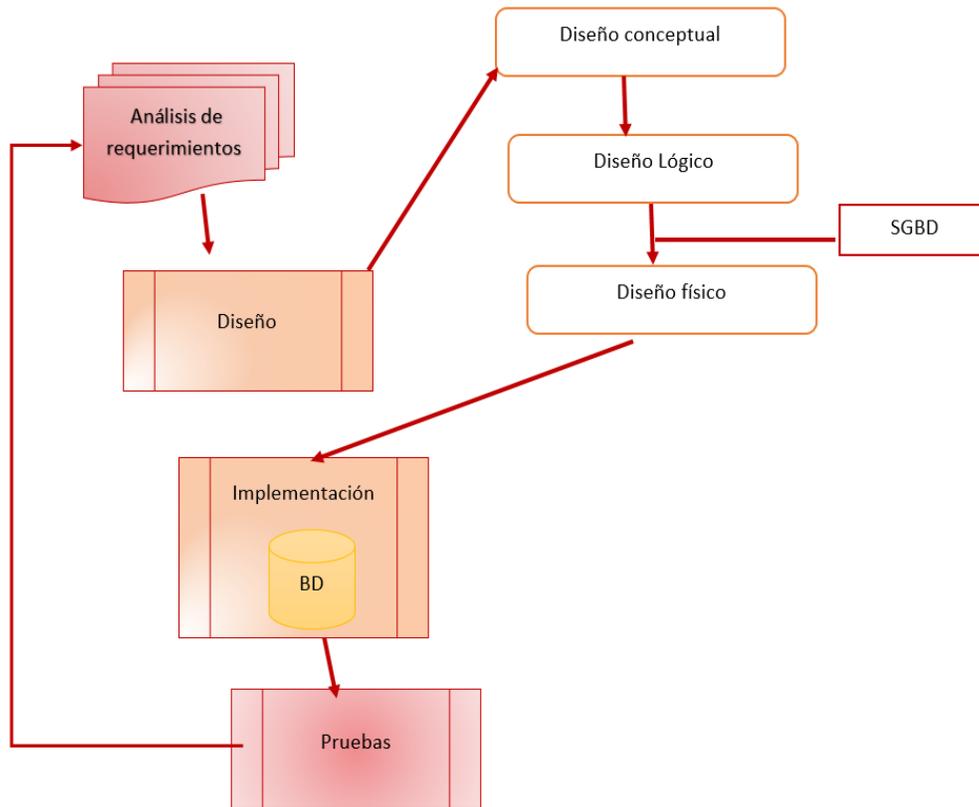


Figura 3.3 Modelo para la creación de la base de datos (Fuente propia)

3.5. Análisis de requerimientos

Durante esta fase se analizan los requerimientos obtenidos de las entrevistas, se conocen las expectativas del usuario identificando los grupos de usuarios reales y los entornos operativos y de proceso.

Las fases de análisis de problema son:

- Entrevista con los usuarios.
- Identificación de necesidades.
- Definir los datos necesarios para la aplicación y las funciones que debe contener la base de datos.

3.5.1. Diseño

- **Diseño conceptual:** Con la información analizada en el paso anterior se procede a plasmarla en términos de base de datos. En un nivel que el cliente la pueda entender es decir reflejamos como son los datos exponiéndolos en un diagrama entidad – relación.
- **Diseño Lógico:** Este diseño consiste en la transformación del diseño conceptual utilizando un modelo de datos lógico, en este caso el modelo lógico relacional. En este diseño nos acercamos a la generación de tablas para poder ser implementadas en un sistema gestor de base de datos.

Se utiliza MySql por su multiplataforma al trabajar en una amplia lista de sistemas operativos. Además de brindar un excelente rendimiento, flexibilidad y velocidad junto con su herramienta Workbench que permite la completa administración tanto de registros, permisos y conexiones.

- **Diseño Físico:** Establecemos como se van almacenar los datos, se elige el tipo, las estructuras de almacenamiento y acceso para alcanzar un rendimiento óptimo de las aplicaciones de la base de datos.

Dentro del diseño físico se implementa la base de datos en el SGBD y se define la estrategia de acceso.

3.5.2. Implementación del modelo

Es la fase donde se hace realidad la base de datos mediante la creación y compilación del diseño físico y transacciones a través de la aplicación para la implementación se procede a preparar un ambiente operacional para la realización de pruebas por separado y efectuar la evaluación al sistema integrado.

3.5.3. Pruebas

Aplicando las distintas pruebas para garantizar la funcionalidad y los objetivos de la base de datos tenemos. (Cloud, 2016).

- a) **Pruebas unitarias:** Donde se validará cada uno de los componentes de una solución. Este tipo de prueba se lleva a cabo durante la etapa de desarrollo.
- b) **Pruebas de integración:** Estas depende del éxito obtenido en las pruebas unitarias y debe lograr dos metas principales:
 - I. Garantizar que se puede construir y desplegar con éxito.
 - II. Asegurar que no surgen problemas durante la ejecución del trabajo: con este objetivo, una vez implementados y configurados, todos los trabajos deben ser ejecutados y los datos procesados.

La aplicación de esta prueba nos sirve para confirmar que el sistema una vez que los módulos están unidos la base de datos se ejecuta correctamente.

- c) **Pruebas de validación de datos:** Mediante esta prueba se someten los datos donde la debilidad más común es la seguridad de aplicación web, por falta de validación en cuanto a las entradas procedentes del cliente o del entorno de la aplicación. Esta debilidad conduce a casi todas las principales vulnerabilidades en aplicaciones como inyecciones sobre el intérprete, ataques locales/Unicode, sobre el sistema de archivo y desbordamiento de buffer.

Los datos procedentes de cualquier entidad o cliente externo nunca deberían ser confiables ya que estos alteran los datos. Esta prueba ha de ser llevada a cabo por un representante del negocio, ya que este perfil es quien mejor conoce los datos y puede validarlos con mayores garantías de éxito.

- d) **Pruebas de aceptación de usuario:** Su objetivo es asegurar que los datos que se proporcionan al usuario final cumplen con sus requerimientos y que lo mismo sucede con el sistema que se pone a su disposición.

- e) **Pruebas de rendimiento:** Esta prueba se ocupan de validar adecuadamente el rendimiento de la solución en condiciones de trabajo reales. Para ello en la prueba se consideran factores como la arquitectura de datos, configuración del hardware, escalabilidad del sistema o la complejidad de las consultas.

En caso que presente errores se realiza una retroalimentación regresando a analizar los requerimientos y realizar las correcciones.

Capítulo 4

Implementación

En este capítulo se presenta el proceso de desarrollo de la base de datos del sistema cotizador, siguiendo los pasos que se muestran en el modelo para la elaboración y diseño del modelo de la base de datos.

4.1. Análisis de requerimientos

Para el desarrollo del sistema se seleccionan los requerimientos y se analizan para agruparlos como requerimientos funcionales y no funcionales.

Los requerimientos obtenidos son un punto importante para la elaboración de cada fase del sistema, es decir lo que el sistema debe hacer y bajo qué circunstancias debe trabajar, dónde los requerimientos se definen en dos: requerimientos funcionales y requerimientos no funcionales, en las tablas número 4.1 y 4.2 se muestran los requerimientos básicos para la elaboración de la base de datos.

Tabla 4.1 Requerimientos funcionales para el sistema cotizador

Funcionales	
Requerimiento	Descripción
Administración de usuarios	Agregar, modificación, eliminación y listado de usuarios
Administración de clientes	Agregar, modificación, eliminación y listado de clientes
Administración de Materiales	agregar, modificación, eliminación y listado de materiales
Administración de Productos	agregar, modificación, eliminación y listado de productos
Administración de tratamientos	Agregar, modificación, eliminación y listado de tratamientos.
Administración de acabados	Agregar, modificación, eliminación y listado de acabados.
Administración de personal	Agregar, modificación, eliminación y listado de personal.
Administración de cotizaciones	Modificación, eliminación y listado de cotizaciones realizadas.
Operaciones	Almacenar los datos resultado de cada cálculo de en la cotización
Reportes	Crear reportes de numero de cotizaciones y número de pies tablón vendidos
Restricciones de consulta	Tener restricciones de consulta en los datos de la base de datos
Almacenar datos de la cotización	Almacenar los datos generados de cada cotización

Tabla 4.2 Requerimientos no funcionales para el sistema cotizador

Requerimiento	Descripción
NO FUNCIONALES	
Facilidad de uso	Los usuarios determinados para el uso del sistema tendrán la facilidad de llevar a cabo las operaciones que se integran en el sistema de una manera sencilla.
Escalabilidad	El sistema se desarrollará en una base de cambio, de tal forma que se agreguen al futuro más funcionalidades, modificar y eliminar funciones.
Flexibilidad	El sistema debe tener flexibilidad en el manejo de los tipos de datos.
Instalación	La instalación del sistema debe ser instalada de una manera fácil en el Hardware y Software especificados.
Seguridad	<p>La seguridad del sistema debe estar de acuerdo a los diferentes usuarios como son: administrador, operador y jefe de área.</p> <ul style="list-style-type: none"> • El acceso al sistema de ser de una manera fácil para los usuarios. • El control de acceso debe tener las restricciones de acuerdo a cada uno de los usuarios. • El sistema deberá rechazar accesos o modificaciones no autorizados.

Dentro de los requerimientos funcionales se hace uso del método para el desarrollo de la base de datos, las fases que la conforman son la obtención de requerimientos, recopilación y análisis, diseño de la base de datos, implementación y pruebas, en la tabla número 4.3 se describe cada fase y su resultado de cada fase.

Tabla 4.3 Descripción de las fases de la metodología del modelo de BD

FASE	DESCRIPCIÓN	Resultados
1	Obtención de requerimientos <ul style="list-style-type: none"> • Definición de requerimientos • Definición de actores 	Requerimientos del sistema y principales actores.
2	Recopilación y análisis <ul style="list-style-type: none"> • Definición de datos a utilizar 	Datos importantes para el uso del sistema
3	Diseño de la base de datos <ul style="list-style-type: none"> • Diseño conceptual • Diseño Lógico • SGBD • Diseño físico 	Los tres diseños diferentes que contiene la base de datos.
4	Implementación	Modelo de base de datos
5	Pruebas	Plan de pruebas

4.2. Definición de actores

En la empresa se lleva el proceso de venta de madera industrial para diferentes proyectos.

Para la obtención de todos los requerimientos del sistema se llevaron a cabo una serie de reuniones mensuales con clientes, para poder definir el personal involucrado con el sistema.

Después del análisis, se identificaron tenemos a la definición de los actores donde se encuentra el tipo de acceso que tiene cada uno, en la tabla 4.3 se describen cada uno de los actores que intervienen en el uso del sistema cotizador.

Tabla 4.4 Definición de actores

Código	Autor	Descripción
Admon	Administrador	Usuario administrador del sistema, el cual posee todos los permisos otorgados por el sistema a fin de realizar el mantenimiento del mismo.
Jefe Área	Jefe del área	Usuario administrador de un apartado específico, el cual tiene la capacidad de dar descuentos y modificaciones de algunos campos con restricción.
Operador	Operador	Funcionario del sistema responsable de realizar las cotizaciones y visualización de reportes.
Usuario	Usuario	Personal que tendrá acceso solo a descarga de cotizaciones.

Acorde con lo mencionado se puede notar que cada usuario tiene privilegios diferentes, por lo que en la base de datos se definirá una tabla de usuarios para que se pueda asignar el privilegio que tendrá dentro de la aplicación.

En la siguiente figura número 4.1 se muestra un diagrama de casos de uso del sistema cotizador general, donde contiene los casos de uso que se utilizan y las operaciones correspondientes, así como los privilegios que tiene cada uno de los usuarios.

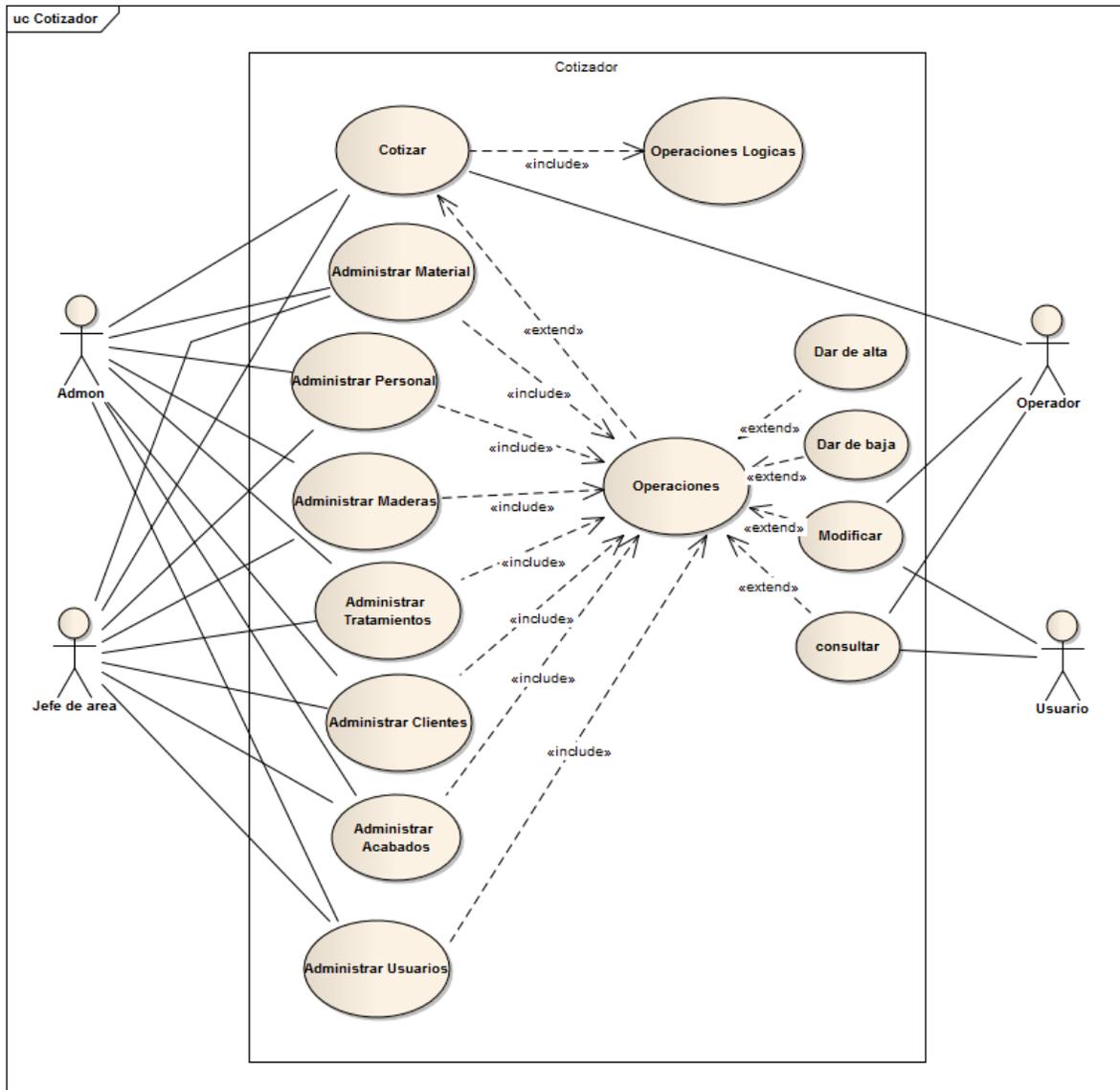


Figura 4.1 Diagrama de casos de uso del sistema general (Fuente propia)

En el diagrama de la figura número 4.2 se muestra las operaciones que solamente el administrador y el jefe de área puede realizar como son altas, modificaciones, eliminaciones y consultas en general.

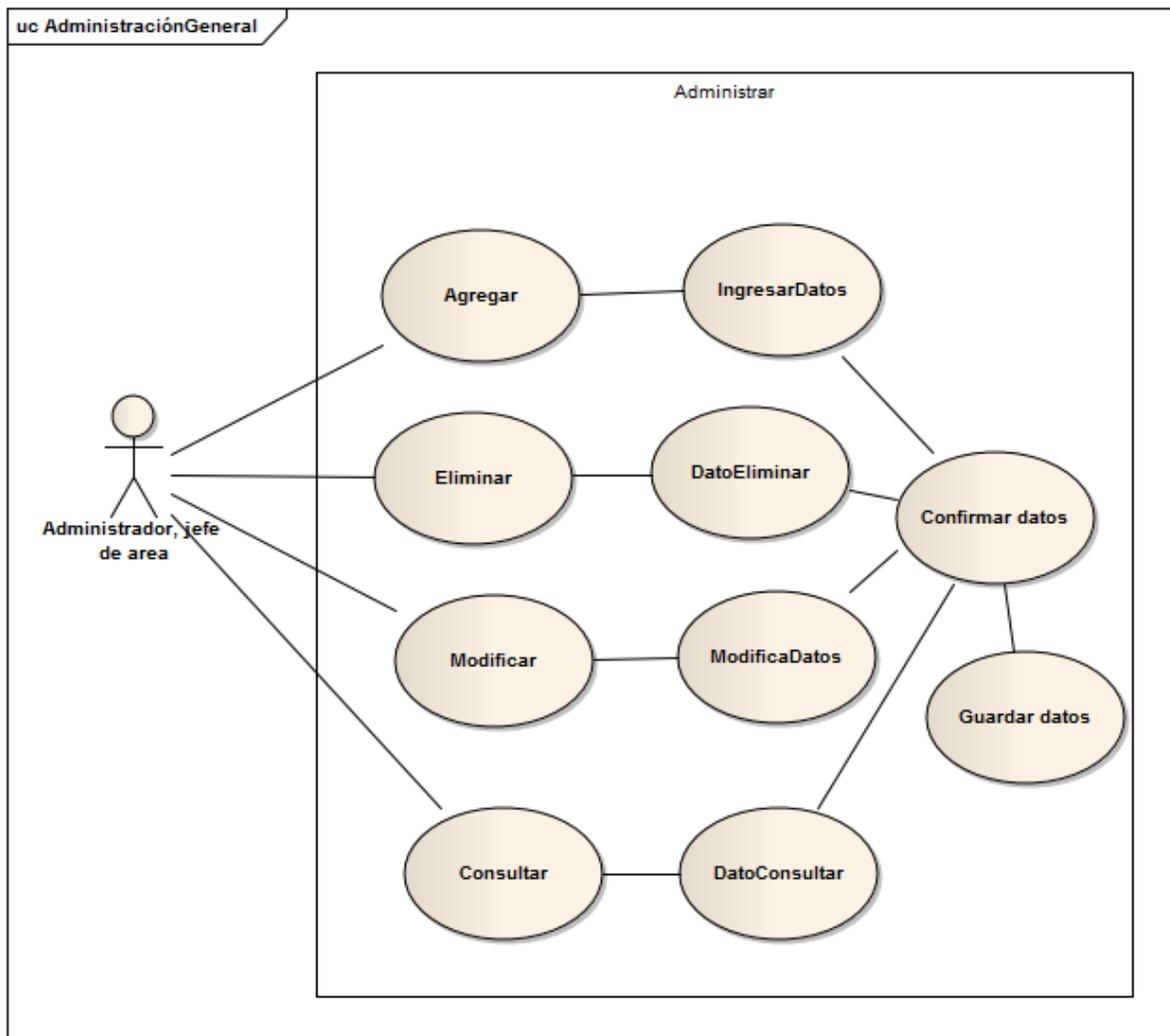


Figura 4.2 Diagrama de caso de usos de administración de operaciones (Fuente propia)

4.3. Análisis de requisitos

Se analizan los datos y estructura del modo en que operan los usuarios en donde se encuentran trabajando, para obtener en específico los datos importantes para la realización de la base de datos.

Con la hoja de cálculo que se muestra en la figura número 4.3 obteniendo datos importantes como logotipos, datos del cliente, celdas a llenar, datos de entrega, especificaciones del producto, grados de dificultad, etc.

Datos del cliente		Datos de entrega				
CLIENT	# CLIENT	CONTACTO	DIRECCIONES	TEL CONTACTOS Y CELL	EMAIL	
SHEYLA		SHEYLA CUB RECAMARA				
logo 	TOTAL IVA INCLUIDO	PRECIO TOTAL PROCESO MADERA	\$ 14,326	TOTAL TRABAJO EN PLANTA	\$ -	
	\$ 16,618	PRECIO TRANSPORT	\$ -	TOTAL TRABAJO EN OBRAS	\$ -	
		PRECIOS TOTAL	\$ 14,326	TOTAL ENGENERIA	\$ -	
		PT MADERA PROCESADO BRUT	295	IVA EN ESTA OFERTA	\$ 2,292	
	FECHA DE PROPUESTA	08/03/2016	NUMERO DE VIGAS	9	PROMEDIA VALOR PT NETO PROCESADO Y INSTALADO EN SU CASO	\$ 71
	DIA DE ENTREGA	12	ML DE VIGAS	20	PROMEDIA VALOR DE PROCESO DEL MADERA SOBRE PT BRUTO	\$ 48
	FECHA DE ENTREGA	20-mar.-16	M2 DE PISO / LAMBRIS	0		
			M2 DE TABLERO	0	NOMBRE DE HOMBRE EN EL EQUIPOS DE MANTENIMIENTO	0
	MANTENIMIENTO	#DIV/0!	INCLUIDO EL TRANSPORT.M.O		TOTAL MANTENIMIENTO UNIOTAFIA	#DIV/0!
	Tipo de madera					
VIGA LAMINADA	VIGA LAMINADA CURBA	VIGA DE ASERADRA TRABAJADA	VIGAS I	PISOS / LAMBRIS	TABLERO	
PT BRUTO	PT BRUTO	PT BRUTO	PT BRUTO	PT BRUTO	PT BRUTO	
295	0	0	0	0	0	
PT NETO	PT NETO	PT NETO	PT NETO	PT NETO	PT NETO	
201	0	0	0	0	0	

Total de pie tablon

Figura 4.3 Datos de caratula de cotización (Fuente del cliente)

Los datos que debe contener el modelo son obtenidos al analizar cada pestaña del archivo de Microsoft Excel®.

4.4. Planeación de base de datos

Con la información brindada y analizada se obtuvieron los datos necesarios para la creación de la base de datos, así como las tablas y relaciones que existen entre ellas.

Primero se definen las tablas que contendrá el cotizador mediante el diagrama entidad-relación, en esta se definen las entidades y los atributos para la elaboración del diagrama.

Para el modelado de los diagramas entidad relación se utiliza el formato del lenguaje de modelado unificado UML.

En la figura número 4.4 se muestran dos tablas realizadas con UML donde se representa la tabla de acabados con las relaciones hacia la tabla de detalle cotiza y el cotizador.

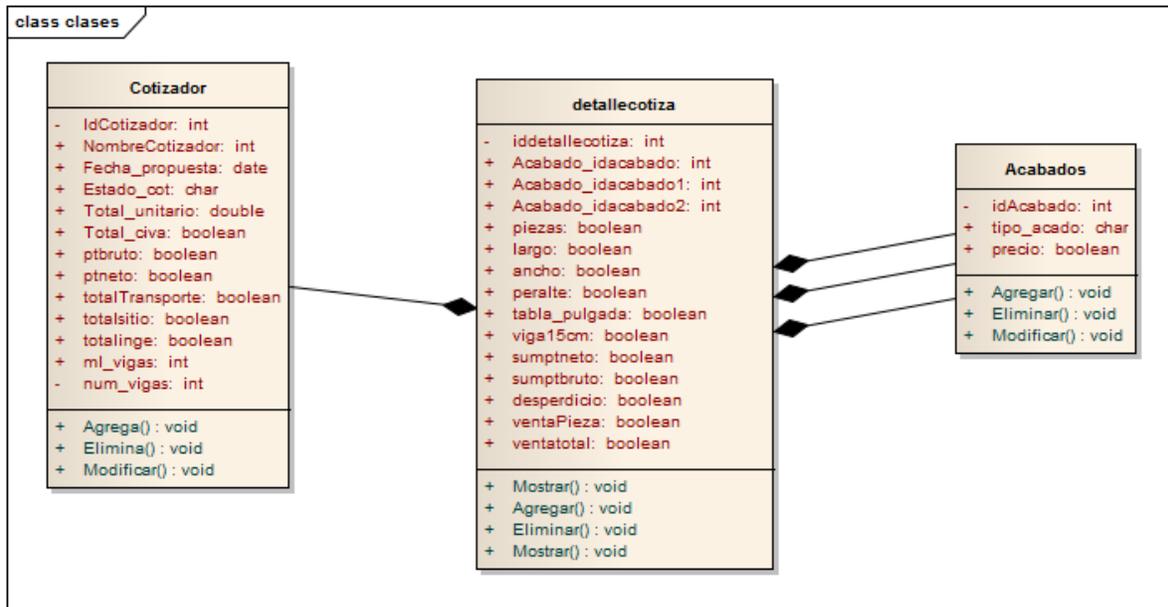


Figura 4.4 Tablas UML de cotizador, detallecotiza y acabados (Fuente propia)

4.5. Diseño de la base de datos

La elaboración del diseño de la base de datos se desarrolla a partir de tres diseños: conceptual, lógico y físico. En cada fase de diseño se integra cada uno de los datos obtenidos y se modela para la estructuración de los datos donde se crea y administra una cotización dentro de la base de datos.

Cada uno de los diseños tendrá el objetivo de obtener un conjunto de datos y operaciones sobre ellos, que nos permitan satisfacer las necesidades de la empresa al dar resultados precisos.

4.5.1. Diseño conceptual

Con respecto a la información obtenida se realiza la elaboración de un diseño conceptual donde se hace la representación de los datos y se capture las propiedades estáticas y dinámicas para satisfacer las necesidades de cada requerimiento; este diseño está integrado por 11 entidades con sus atributos, como se puede observar en la figura número 4.5, donde se muestra el diagrama entidad-relación del sistema, identificando cada una de las tablas que contendrán los datos necesarios para la debida elaboración de una cotización.

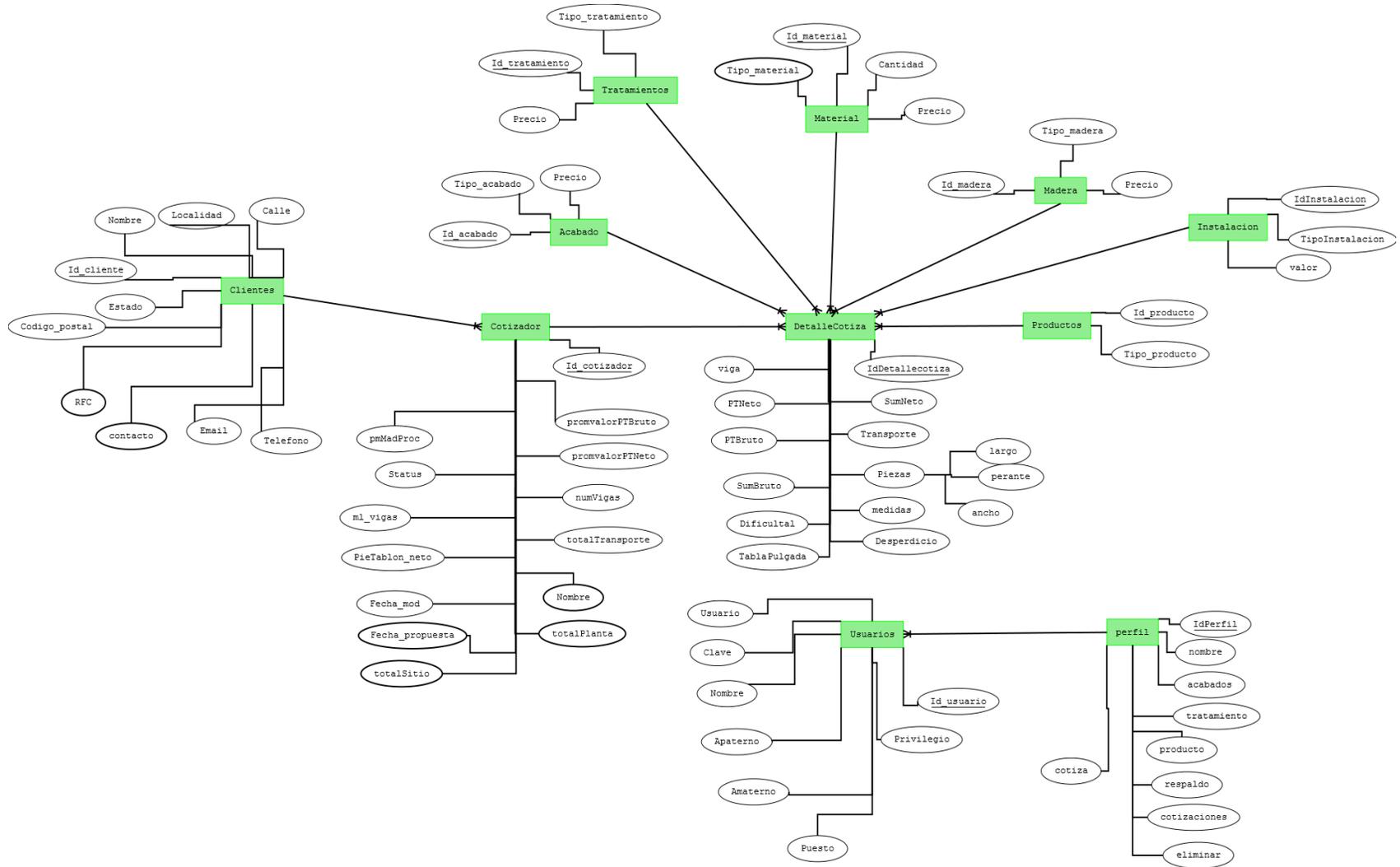


Figura 4.5 Diagrama entidad- relación del sistema cotizador (Fuente propia)

4.5.2. Diseño Lógico

Para el diseño lógico se parte del resultado del diseño conceptual mediante el modelo entidad relación, transformándolo en una estructura de datos de modelo relacional, donde se muestra un conjunto de esquemas de transacciones, Expresado los objetos de información, propiedades y relaciones entre objetos, así como las restricciones entre relaciones (estática) y las operaciones sobre los objetos o sus relaciones son el conjunto de operaciones (dinámica).

4.5.3. Diseño físico

Para el diseño físico se tienen en cuenta los diseños lógico y conceptual donde previamente se analizaron los datos para la estructura que tiene la base de datos, por consiguiente, se procede a integrar los datos de tal forma que se pueden implementar a un sistema gestor de base de datos relacionada con MySQL.

En este proceso es muy parecido al del diseño lógico de la base de datos aquí se detalla y se muestran los aspectos de eficiencia.

Para la elaboración de este diseño se hace uso de DBDL (*Database Design Language*) que nos sirve para la creación de las tablas que contiene la base de datos.

En la figura número 4.7 tenemos la forma en que se estructura una tabla para ser integrada en un sistema gestor de base de datos. Cada tabla se integra de su nombre, los campos con el tipo de dato necesario para poder almacenar la información que el cliente solicita.

```
CREATE TABLE IF NOT EXISTS `acabado` (  
  `idAcabado` int(11) NOT NULL AUTO_INCREMENT,  
  `tipo_acado` varchar(45) DEFAULT NULL,  
  `precio` float DEFAULT NULL,  
  PRIMARY KEY (`idAcabado`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

Figura 4.7 DBDL de acabados

En el anexo A se encuentran los DBDL de las tablas que componen el diseño físico de este modelo.

4.6. Implementación de modelo de base de datos

4.6.1. Mapeo de la base de datos

Para la integración de la base de datos con el módulo de control, se utiliza la herramienta *Hibernate* que nos permite realizar el mapeo objeto relacional de la BD, con el fin de convertir las tablas en objetos en lenguaje java para ser utilizados para el funcionamiento del sistema.

En la figura número 4.8 se muestra el mapeo de la base de datos con el módulo de control, dónde se visualiza en la parte izquierda de la imagen la carpeta de mapeo de cada tabla y en la parte derecha la definición de los atributos y sus relaciones.

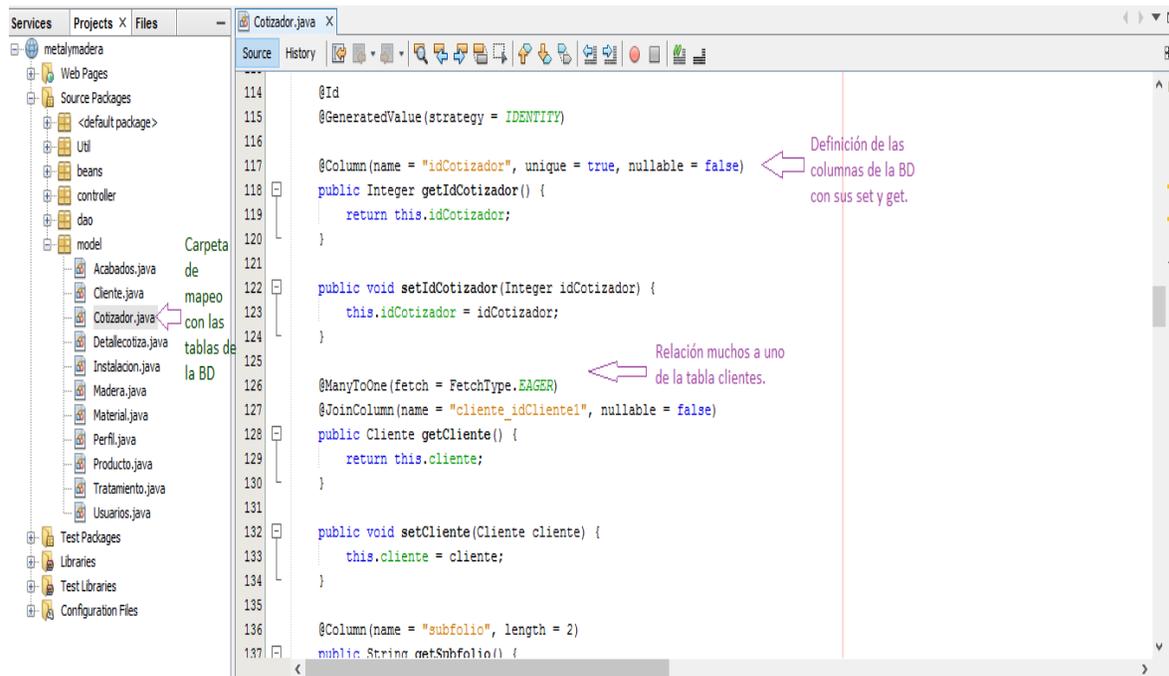


Figura 4.8 Mapeo de la base de datos (Fuente propia)

4.6.2. Módulo de creación de usuario

En el módulo de creación de usuarios, consiste en obtener la información de cada usuario y los permisos que tienen para realizar las operaciones, así el administrador del sistema dará paso al llenado de los campos para dar de alta al nuevo usuario

según el puesto que tenga o los privilegios asignados, los campos a llenar son: Nombre, apellido paterno, apellido materno, puesto, usuario, contraseña y el privilegio que tendrá según lo determine el administrador.

La correlación existente entre la base de datos y la interfaz se muestra en la figura 4.9

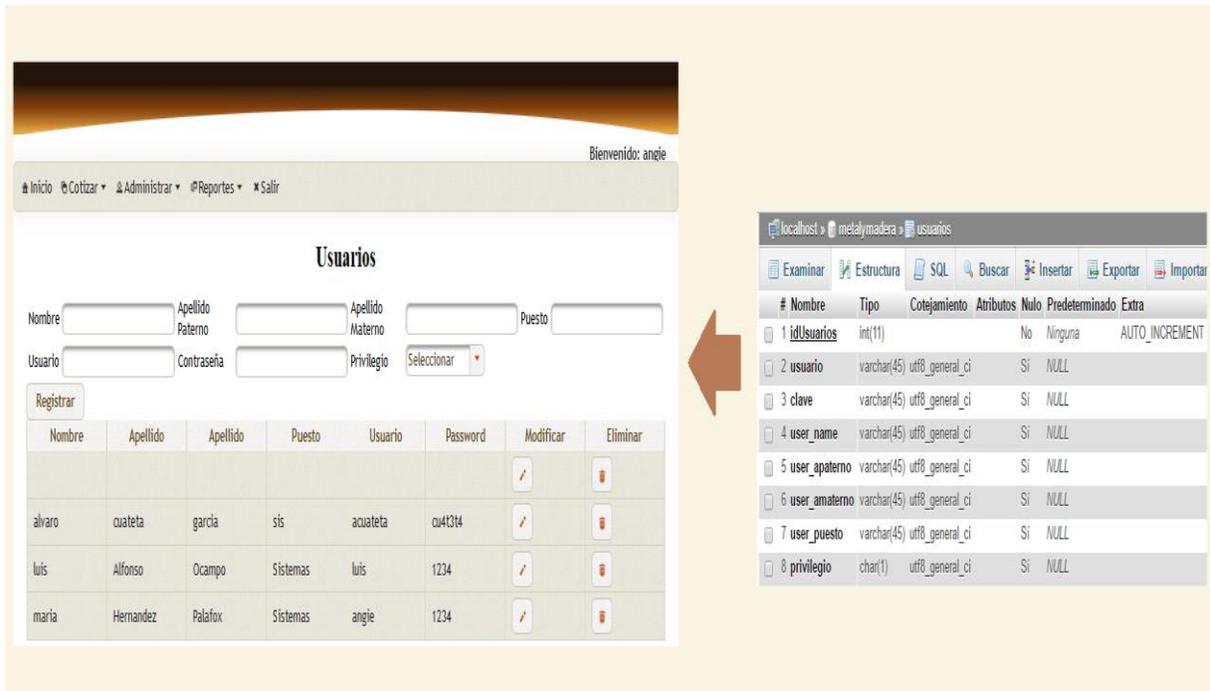


Figura 4.9 Correlación entre la interfaz y la base de datos (Interfaz elaborada por personal de diseño)

4.6.3. Modelo de administración de clientes

Para la creación de este módulo tenemos los datos del cliente, donde se dará de alta un nuevo cliente, se modificará, consultara y eliminará según el administrador crea necesario.

Para dar de alta un nuevo cliente se necesita los siguientes datos: El nombre de la empresa, nombre del contacto, apellido paterno, apellido materno del contacto, RFC ya sea de la persona física o moral, calle, número de la calle, localidad, estado, código postal, teléfono particular, celular, email.

En la figura 4.10 se muestra la interacción que existe la interfaz del cliente con la base de datos.

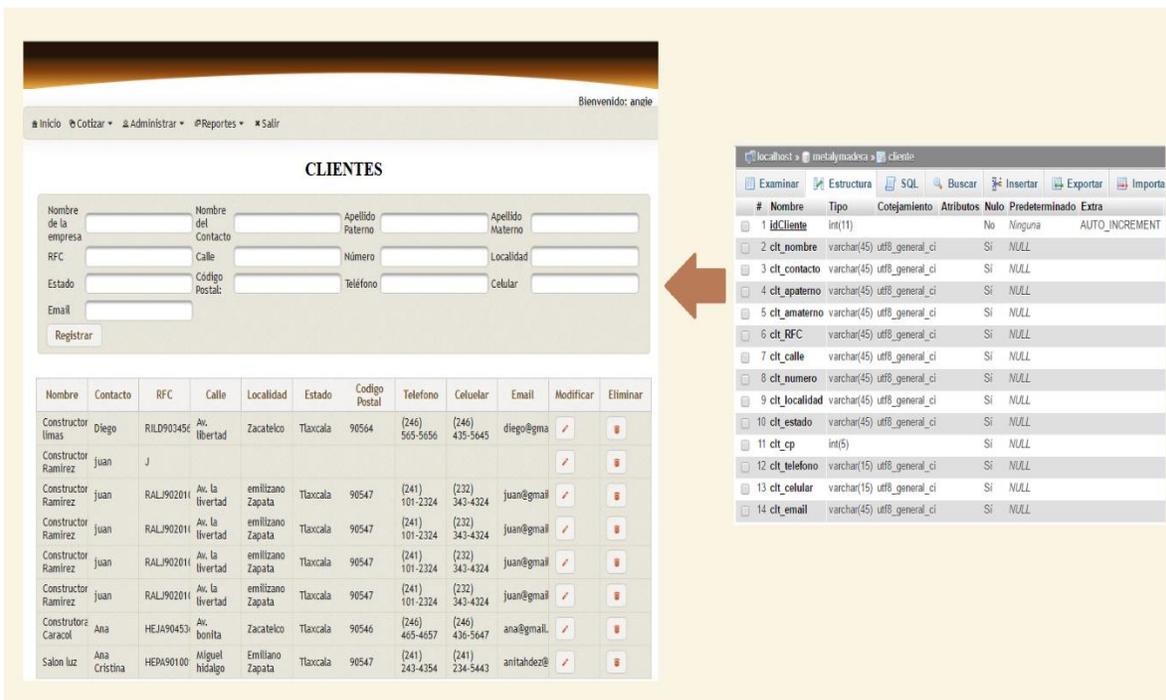


Figura 4.10 Iteración entre la base de datos e interfaz

4.6.4. Módulo de catálogos

Para crear una cotización se requiere tener tanto materiales, madera, acabados y tratamientos, por lo cual se crea un catálogo de estos donde se podrán administrar independientemente su información, teniendo solamente su nombre, y su precio para realizar las operaciones correspondientes.

En las figuras 4.11 muestra la iteración con la base de datos e interfaz de material, donde se visualiza la tabla que se utiliza y la forma gráfica en que se presenta



Figura 4.11 Iteración entre la BD e interfaz de usuario

4.6.5. Módulo de cotización

Para la creación de las tablas necesarios la información de clientes, madera, productos, acabados, tratamientos, donde se toman precios para las operaciones de una cotización.

Para dar de alta una cotización se llenará los datos siguientes: fecha, folio, cliente, cantidad de madera, tipo de madera, medidas de la madera, pies tablón neto, pies tablón bruto, tipos de acabados, tipo de tratamientos.

En la figura 4.12 se muestra las tablas que intervienen para poder realizar los cálculos correspondientes para poder entregar la cotización.

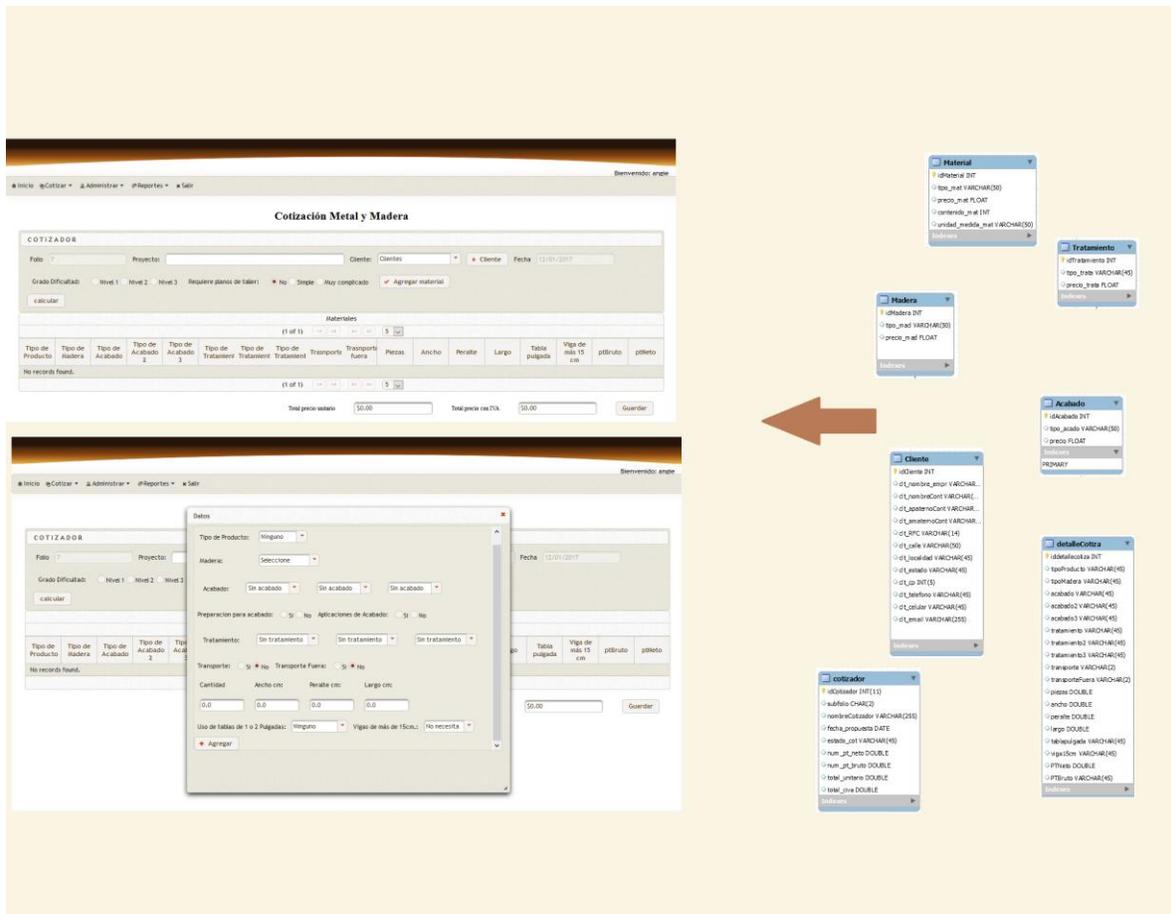


Figura 4.12 Correlación de la BD con interfaz del cotizador

4.6.6. Módulo de generación de cotización en PDF

Este módulo se necesita un conjunto de tablas para obtener la información adecuada para generar el PDF con la cotización que se le entregara al cliente; la información requerida es: datos del cliente, datos de requerimientos del cliente, detalles de la cotización.

Capítulo 5 Pruebas y Resultados

En este capítulo se presentan las pruebas y resultados del modelo de base de datos, teniendo las pruebas de funcionalidad, usabilidad, integridad y seguridad.

5.1. Plan de pruebas

5.1.1. Objetivo

Comprobar si el sistema cumple sus requisitos específicos que el cliente espera.

5.1.2. Alcance

Garantizar y controlar la calidad durante el proceso de desarrollo del sistema.

Tomando como referencia el modelo V mencionado en el capítulo 2, sección 2.10, se realizan las siguientes pruebas al sistema para probar su funcionamiento e integridad del mismo.

5.2. Prueba de funcionamiento

En la entrega del sistema se realizan pruebas de funcionamiento en la base de datos, este tipo de prueba examina si se cubren con las necesidades de funcionamiento acorde con las especificaciones entregadas.

En ellas se deben verificar si el sistema lleva a cabo correctamente todas las funciones requeridas, incluida la validación de los datos. Esta prueba se enfoca en las tareas y condiciones planeadas de error y de exploración, al mismo tiempo son

necesarias para evaluar la usabilidad y rendimiento para cada módulo del sistema conforme a la norma ISO / IEC / IEEE 25000 y ISO 2196 para la evaluación de la calidad de software.

5.2.1. Prueba de validación de formularios

El sistema de cotización cuenta con el llenado de varios formularios para el envío e ingreso de datos; en la prueba se realiza la validación de datos a la hora de ingresarlos al formulario, para asegurar que a la hora de guardar sean los correctos.

Esta prueba permite que checar la longitud de los campos que solo necesitan un número específico de caracteres. Permite además que los campos y los datos, la longitud de los datos que deben ingresar en las cajas de texto, la consistencia, el formato de los datos y seguridad de los datos mostrados e introducidos en los formularios de la interfaz.

En la figura 5.1 se muestra un test de prueba de formulario donde se presentan las preguntas para verificar el contenido que se ingresa en cada uno de los formularios.



Nombre: María de la Luz Pérez Fecha: 18/Mayo/2017

Sistema: sistema cotizador de madera industrial			
Cuestionario de evaluación de base de datos			
Dominio:	Base de datos		
Proceso:	Prueba de formulario		
Objetivo de control:	Verificar que el contenido que proviene y que se ingresa a la base de datos sea correcta.		
Cuestionario			
Pregunta	SI	NO	N/A
¿Los formularios a llenar tienen ayuda para que se agreguen los datos correctos?	X		
¿Los datos que se ingresan se guardan en la tabla correcta?	X		
¿Los campos del formulario corresponden a los de la tabla de la base de datos?	X		
¿El tipo de dato que se ingresa corresponde al tipo de datos de la base de datos?	X		
¿Las longitudes de los campos de texto son adecuados?	X		
¿Las etiquetas expresan lo que se debe insertar?	X		
¿Hay distinción entre campo opcional y obligatorio?	X		
¿Los datos son validados antes de ser guardados?	X		
¿Permite hacer correcciones en caso de errores?	X		
¿Los formularios permiten el autocompletado?	X		
¿Tiene formularios de validación de datos?	X		
¿Los campos de fechas te permite la inserción correctamente?	X		

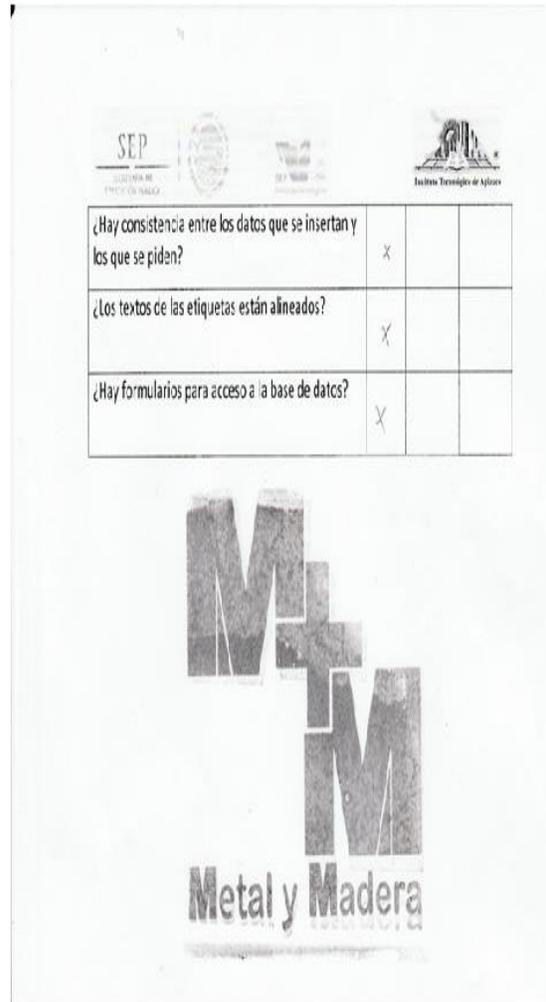


Figura 5.1 Prueba de formulario (Fuente propia)

5.2.2. Prueba de búsqueda

Este tipo de prueba permite verificar que los resultados de las búsquedas o consultas contengan la información solicitada por el usuario. Se valida contenido completo, errores ortográficos.

La figura 5.3 Se encuentra un *checklist* con el 100% de las búsquedas fueron correctas.

Nombre: María de la Luz Pérez Fecha: 13/Mayo/2017

Sistema: sistema cotizador de madera industrial			
Cuestionario de evaluación de base de datos			
Dominio:	Base de datos		
Proceso:	Prueba de búsqueda		
Objetivo de control:	Corroborar que las consultas que el usuario muestre la información adecuada.		
Cuestionario			
Pregunta	SI	NO	N/A
¿Cuándo realiza una consulta de alguna tabla obtiene la información correcta?	X		
¿La búsqueda se realiza de forma automática o intuitiva?	X		
¿Existe paginación en las consultas de datos?	X		
¿Tiene un indicador de cuantos registros se muestran?	X		
¿La búsqueda tiene alguna plantilla?	X		
¿El software cuenta con búsquedas especializadas?	X		
¿La consulta la puede visualizar claramente?	X		
¿Existe complicación en la muestra de datos?	/	X	
¿Los datos visualizados tienen errores en el formulario en que se muestra?		X	
¿Los datos e información mostrada se pueden guardar en un archivo PDF o doc.?	X		
¿Las consultas son acopladas a las necesidades del usuario?	X		
¿Los campos de texto contienen ayuda para saber que dato se debe buscar?		X	

Figura 5.2 Prueba de búsqueda

5.3. Prueba de usabilidad

Las pruebas realizadas en este rubro tienen la finalidad de verificar la facilidad de usar un sistema. Las pruebas de usabilidad verificaron el aprendizaje, es decir el usuario aprende fácilmente la funcionalidad de cada apartado del sistema, ver si las tareas básicas son realizadas a la primera vez que se tiene contacto con la aplicación.

Además, se vio la eficiencia, el manejo de errores y el grado de satisfacción para cada uno de los interesados.

5.3.1. Prueba de contenido

Esta prueba permite evaluar la claridad de datos su visualización de errores al proporcionar el sistema al usuario, ver si los datos son claros, completos y se observan sin algún error, así como su tabla o formulario correspondiente y a la hora de acceder a la base de datos no encuentre complicación alguna.

El *checklist* fue contestado por un usuario con los conocimientos sobre cotizaciones de madera industrial. En la figura 5.4 se muestra un *checklist* de la prueba de contenido, realizado por la gerente de ventas.



 Nombre: María de la Luz Pérez Fecha: 18/Mayo/2017

Sistema: sistema cotizador de madera industrial			
Cuestionario de evaluación de base de datos			
Dominio:	Base de datos		
Proceso:	Prueba de contenido		
Objetivo de control:	Verificar que el contenido que proviene y que se ingresa a la base de datos sea correcta.		
Cuestionario			
Pregunta	SI	NO	N/A
¿Se puede acceder con seguridad a la base de datos?	X		
¿Los tiempos de consulta son adecuados?	X		
¿La visualización de los datos se muestra en tablas con registros menores de 10?	X		
¿Existe algún tipo de formulario sin ayuda?		X	
¿Los datos que se insertan a través de formulario son claros?	X		
¿Es complicado la visualización de los datos?		X	
¿Se puede hacer respaldo de los datos que se visualizan?	X		
¿Muestra un mensaje de que las operaciones se han realizado correctamente en la base de datos?	X		
¿Los datos se muestran en orden?	X		
¿Los datos que se ingresan o se muestran tienen errores ortográficos?		X	
¿Los datos se muestran borrosos durante la navegación?	X	X	

Metal y Madera

Figura 5.3 Prueba de contenido (Fuente propia)

5.4. Prueba de integración

Esta prueba nos permite realizar la integración con los módulos de interfaz y verificar que todos los elementos que componen el sistema funcionen correctamente. Se centra esencialmente en probar la comunicación entre los componentes de los módulos de vista, control y modelo.

En la tabla 5.1 se muestran las pruebas que se realizaron para la integración del sistema, mostrando el número de requerimiento, la especificación del requerimiento, el tipo de prueba, descripción de la prueba, resultado esperado y el resultado obtenido siendo este correcto o incorrecto.

Tabla 5.1 Pruebas de integración

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P1	Autenticación de usuarios	Prueba de Estrés	Se debe ingresar el nombre de usuario y contraseña, para tener acceso al sistema.	Acceder al sistema con el usuario correspondiente	x	
P2	Alta de usuarios	Link	Dar la navegación correcta para el llenado del formulario.	Link bien direccionado	x	
P3	Alta de usuarios	Contenido	Validar que los campos del formulario reciban los tipos de datos correctos correspondientes a la BD.	Los campos deben cumplir con validación de contenido.	x	
P4	Alta de usuarios	Funcionabilidad	La función de guardar, realiza la validación de los datos y los almacena en la BD.	Datos almacenados en la base de datos correctamente.	x	
P5	Modificación de usuarios	Link	Dar la navegación correcta para la modificación de información del usuario.	Link bien direccionado.	x	
P6	Modificación de usuarios	Funcionalidad	La función de modifica, valida y actualiza registros del usuario en la BD.	Modificación de los registros en la BD correctamente.	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P7	Eliminación de usuarios	Link	Dar la navegación correcta para eliminar un registro de la tabla.	Link direccionado correctamente.	x	
P8	Eliminación de usuarios	Funcionabilidad	La función de eliminar, confirma y elimina el registro seleccionado.	Se realiza la validación, realiza la eliminación directa.	x	
P9	Alta clientes	Link	Dar el direccionamiento correcto al formulario agregar cliente.	Link direccionado correctamente.	x	
P10	Alta clientes	Funcionabilidad	La función de guardar, realiza la validación de los datos y los almacena en la BD.	Datos almacenados correctamente.	x	
P11	Alta clientes	Contenido	Validar que los campos del formulario reciban los tipos de datos correctos correspondientes a la BD.	Todos los campos cumplen con tipos de datos.	x	
P12	Modificación clientes	Link	Dar el direccionamiento correcto al formulario de modificar clientes.	Link direccionado correctamente.	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P13	Modificación clientes	Funcionalidad	La función actualiza los datos del formulario en la BD.	Registros modificados correctamente.	x	
P14	Eliminación clientes	Link	Dar el direccionamiento correcto a la ventana de confirmación.	Se envía el link a la ventana de confirmación.	x	
P15	Eliminación clientes	Funcionalidad	La función de eliminar registros de la tabla, confirma y elimina el registro de la BD.	Se valida y elimina el registro correctamente	x	
P16	Alta productos	Link	Dar el direccionamiento correcto al formulario de productos.	Link direccionado correctamente.	x	
P17	Alta productos	Funcionalidad	La función de guardar productos, realiza la validación de los datos y los almacena en la BD.	Datos almacenados correctamente.	x	
P18	Alta productos	Contenido	Validar que los campos del formulario reciban los tipos de datos correctos correspondientes a la BD.	Todos los campos cumplen con tipos de datos.	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P19	Modificación productos	Link	Dar el direccionamiento correcto al formulario modificar productos.	Link direccionado correctamente.	x	
P20	Modificación productos	Funcionalidad	La función actualiza los datos del formulario en la BD.	Registros modificados correctamente.	x	
P21	Eliminación productos	Link	Dar el direccionamiento correcto a la ventana de confirmación.	Se envía el link a la ventana de confirmación.	x	
P22	Eliminación productos	Funcionalidad	La función de eliminar registros de la tabla, confirma y elimina el registro de la BD.	No valida y solo elimina el registro directamente.	x	
P23	Alta materiales	Link	Dar el direccionamiento correcto al formulario de materiales.	Link direccionado correctamente.	x	
P24	Alta materiales	Funcionalidad	La función de guardar materiales, realiza la validación de los datos y los almacena en la BD.	Datos almacenados correctamente.	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P25	Alta materiales	Contenido	Validar que los campos del formulario reciban los tipos de datos correctos correspondientes a la BD.	Todos los campos cumplen con tipos de datos.	x	
P26	Modificación materiales	Link	Dar el direccionamiento correcto al formulario modificar materiales.	Link direccionado correctamente.	x	
P27	Modificación materiales	Funcionalidad	La función actualiza los datos del formulario en la BD.	Registros modificados correctamente.	x	
P28	Eliminación materiales	Link	Dar el direccionamiento correcto a la ventana de confirmación.	Se envía el link a la ventana de confirmación.	x	
P29	Eliminación materiales	Funcionalidad	La función de eliminar registros de la tabla, confirma y elimina el registro de la BD.	Se valida y se elimina registro.	x	
P30	Alta madera	Link	Dar el direccionamiento correcto al formulario de madera.	Link direccionado correctamente.	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P31	Alta madera	Funcionabilidad	La función de guardar madera, realiza la validación de los datos y los almacena en la BD.	Datos almacenados correctamente.	x	
P32	Alta madera	Contenido	Validar que los campos del formulario reciban los tipos de datos correctos correspondientes a la BD.	Todos los campos cumplen con tipos de datos.	x	
P33	Modificación madera	Link	Dar el direccionamiento correcto al formulario modificar madera.	Link direccionado correctamente.	x	
P34	Modificación madera	Funcionalidad	La función actualiza los datos del formulario en la BD.	Registros modificados correctamente.	x	
P35	Eliminación madera	Link	Dar el direccionamiento correcto a la ventana de confirmación.	Se envía el link a la ventana de confirmación.	x	
P36	Eliminación madera	Funcionalidad	La función de eliminar registros de la tabla, confirma y elimina el registro de la BD.	Si se elimina el registro	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P37	Alta tratamiento	Link	Dar el direccionamiento correcto al formulario de tratamiento.	Link direccionado correctamente.	x	
P38	Alta tratamiento	Funcionalidad	La función de guardar tratamiento, realiza la validación de los datos y los almacena en la BD.	Datos almacenados correctamente.	x	
P39	Alta tratamiento	Contenido	Validar que los campos del formulario reciban los tipos de datos correctos correspondientes a la BD.	Todos los campos cumplen con tipos de datos.	x	
P40	Modificación tratamiento	Link	Dar el direccionamiento correcto al formulario modificar tratamiento.	Link direccionado correctamente.	x	
P41	Modificación tratamiento	Funcionalidad	La función actualiza los datos del formulario en la BD.	Registros modificados correctamente.	x	
P42	Eliminación tratamiento	Link	Dar el direccionamiento correcto a la ventana de confirmación.	Se envía el link a la ventana de confirmación.	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P43	Eliminación tratamiento	Funcionalidad	La función de eliminar registros de la tabla, confirma y elimina el registro de la BD.	Valida y solo elimina el registro directamente.	x	
P44	Alta acabado	Link	Dar el direccionamiento correcto al formulario de acabado.	Link direccionado correctamente.	x	
P45	Alta acabado	Funcionalidad	La función de guardar acabado, realiza la validación de los datos y los almacena en la BD.	Datos almacenados correctamente.	x	
P46	Alta acabado	Contenido	Validar que los campos del formulario reciban los tipos de datos correctos correspondientes a la BD.	Todos los campos cumplen con tipos de datos.	x	
P47	Modificación acabado	Link	Dar el direccionamiento correcto al formulario modificar acabado.	Link direccionado correctamente.	x	
P48	Modificación acabado	Funcionalidad	La función actualiza los datos del formulario en la BD.	Registros modificados correctamente.	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P49	Eliminación acabado	Link	Dar el direccionamiento correcto a la ventana de confirmación.	Se envía el link a la ventana de confirmación.	x	
P50	Eliminación acabado	Funcionalidad	La función de eliminar registros de la tabla, confirma y elimina el registro de la BD.	Se valida y se elimina el registro únicamente.	x	
P51	Alta personal	Link	Dar el direccionamiento correcto al formulario de personal.	Link direccionado correctamente.	x	
P52	Alta personal	Funcionalidad	La función de guardar personal, realiza la validación de los datos y los almacena en la BD.	Datos almacenados correctamente.	x	
P53	Alta personal	Contenido	Validar que los campos del formulario reciban los tipos de datos correctos correspondientes a la BD.	Todos los campos cumplen con tipos de datos.	x	
P54	Modificación personal	Link	Dar el direccionamiento correcto al formulario modificar personal.	Link direccionado correctamente.	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P55	Modificación personal	Funcionalidad	La función actualiza los datos del formulario en la BD.	Registros modificados correctamente.	x	
P56	Eliminación personal	Link	Dar el direccionamiento correcto a la ventana de confirmación.	Se envía el link a la ventana de confirmación.	x	
P57	Eliminación personal	Funcionalidad	La función de eliminar registros de la tabla, confirma y elimina el registro de la BD.	Valida y solo elimina el registro directamente.	x	
P58	Alta cotización	Link	Dar el direccionamiento correcto al formulario de cotizar.	Link direccionado correctamente.	x	
P59	Alta cotización	Contenido	Validar que los campos del formulario reciban los tipos de datos correctos correspondientes a la BD.	Datos ingresados correctamente.	x	
P60	Alta cotización	Funcionalidad	La función de guardar madera, realiza la validación de los datos y los almacena en la BD.	La función genera la cotización correctamente.	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P61	Modificación cotización	Link	Dar el direccionamiento correcto al formulario de modificación	Se modifica correctamente.	x	
P62	Modificación cotización	Contenido	Validar que los campos modificados sean actualizados correctamente.	Se modifica correctamente.	x	
P63	Modificación cotización	Funcionalidad	La función de modificar la cotización, realiza el cambio de los datos en la BD	Se modifica correctamente.	x	
P64	Eliminar cotización	Link	Dar el direccionamiento correcto a la ventana de confirmación.	Se direcciona correctamente.	x	
P65	Eliminar cotización	Funcionalidad	La función de eliminar registros de la tabla, confirma y elimina el registro de la BD.	Eliminación correcta.	x	
P66	Generación de PDF de cotización	Link	Direccionar a la generación del pdf con la cotización realizada.	Se direcciona correctamente.	x	
P67	Generación de PDF de cotización	Funcionalidad	La función realiza la generación de pdf de la cotización realizada.	Se genera el pdf de la cotización adecuadamente	x	

Tabla 5.1 Pruebas de integración (Continuación)

N°	Especificación de Requerimientos	Tipo de Prueba	Descripción de Prueba	Resultado Esperado	Resultados obtenidos	
					Correcto	Incorrecto
P68	Reportes de cotizaciones	Link	Dar el direccionamiento correcto a la interfaz de reportes.	Da el direccionamiento correcto	x	
P69	Reportes de cotizaciones	Funcionalidad	La función de generar reportes de las cotizaciones existentes.	La función cumple con el reporte de cotización.	x	
P70	Respaldo de BD	Link	Dar el direccionamiento correcto a la interfaz de respaldo.	Da el direccionamiento correcto.	x	
P71	Respaldo de BD	Funcionalidad	La función de respaldo realiza la conexión a la BD y realiza el respaldo de la misma.	Respaldo correcto.	x	

De la información brindada en la tabla número 5.1 ofreció una visión de la integración del sistema, observando la totalidad de los requerimientos cumplidos, por lo que el sistema es óptimo para su uso.

En la figura número 5.4 se ilustra la integración de forma gráfica del patrón de diseño MVC.

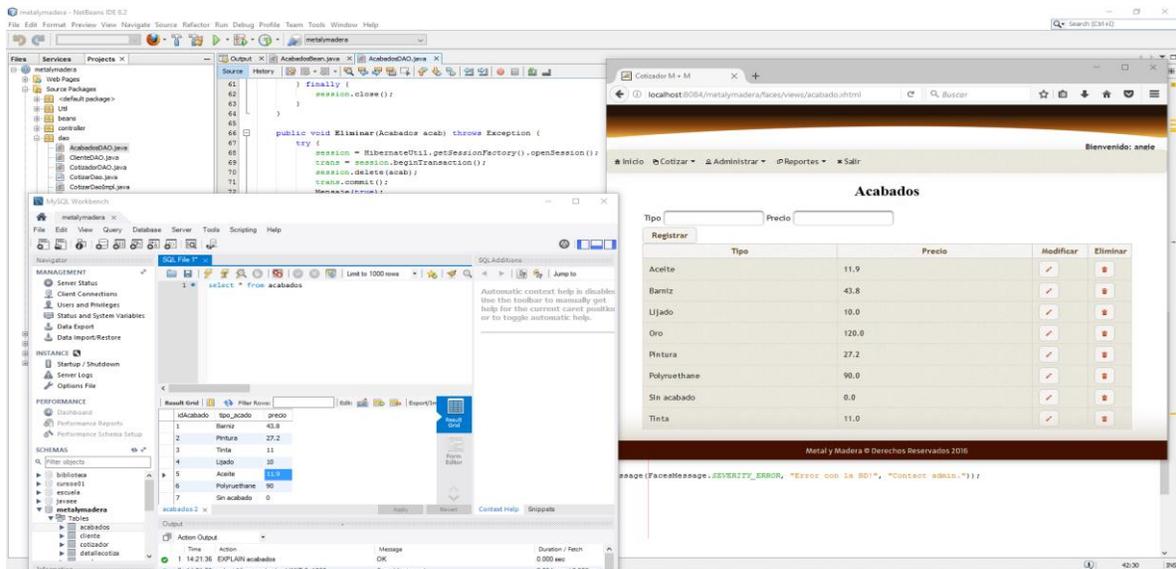


Figura 5.4 Integración del modelo MVC (Control e Interfaz elaborado por personal encargado)

5.5. Pruebas de seguridad

En estas pruebas es verificar que los mecanismos de protección estén incorporados al resguardo de los datos almacenados en la base de datos, para esta prueba se realizará el respaldo de la base de datos para garantizar que la información este protegida en casos de desastres, daños de disco, etc.



Figura 5.5 Respaldo de base de datos

En la figura número 5.5 se muestra la interacción de la interfaz de la cual se realizaron los respaldos a la base de datos.

En la figura número 5.6 se observa la carpeta en donde se han generado los respaldos de la base de datos, identificado por la fecha y la hora del respaldo.

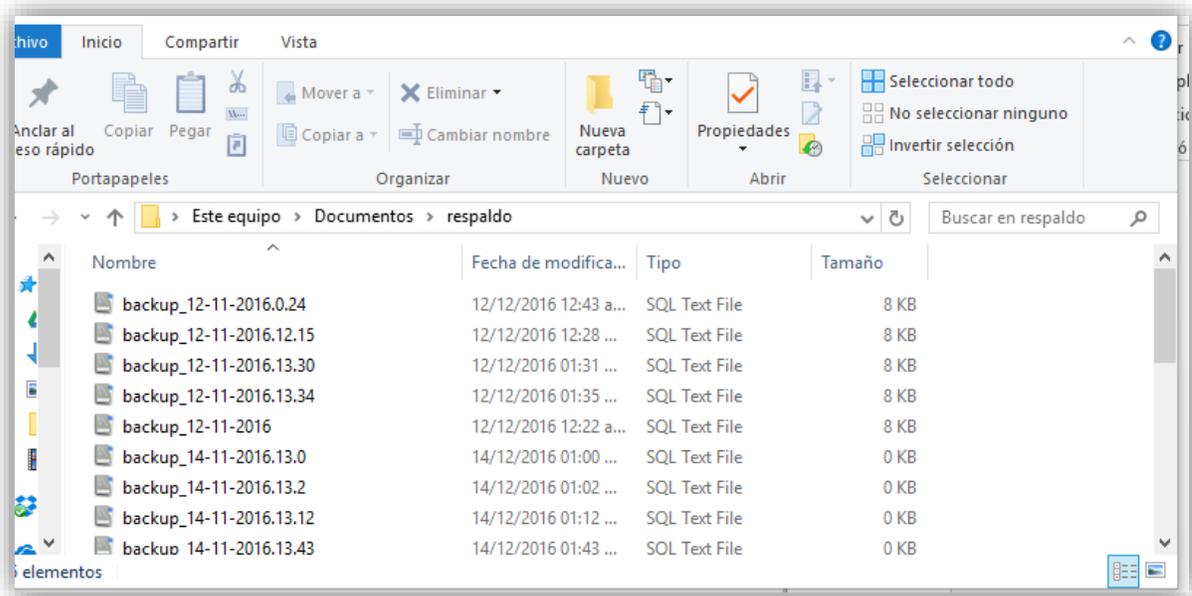


Figura 5.6 Carpeta con respaldos de la base de datos



Nombre: Maria de la luz Perez Fecha: 13/Mayo/2017

Sistema: sistema cotizador de madera industrial			
Cuestionario de evaluación de base de datos			
Dominio:	Base de datos		
Proceso:	seguridad		
Objetivo de control:	Verificar que los datos cuenten con la Seguridad adecuada para la elaboración del cotizador		
Cuestionario			
Pregunta	SI	NO	N/A
¿Se encuentra un administrador de sistemas en la empresa que lleve un control de los usuarios?	x		
¿Son gestionados los perfiles de estos usuarios por el administrador?	x		
¿Son gestionados los accesos a las instancias de la base de datos?	x		
Se renuevan las claves de los usuarios de la base de datos	x		
Se obliga el cambio de contraseña de forma automática.		x	
Se realizan copias de seguridad (diariamente, semanalmente, mensualmente.)?	x		
¿Posee la base de datos un diseño físico y lógico?	x		
¿Existe una instancia con copia del repositorio para el entorno de desarrollo?	x		
¿Los datos utilizados en el entorno de desarrollo, son reales?	x		
¿Los dispositivos que tienen las copias de seguridad, son almacenados fuera del edificio de la empresa?	x		
¿Cuándo se necesita restablecer la base de datos, se le comunica al administrador?	x		

Metal y Madera

Figura 5.7 Prueba de seguridad

5.6. Pruebas de rendimiento

En la siguiente prueba se centra en determinar la velocidad con la que opera la base de datos, dentro de las peticiones que el sistema solicita en las diferentes tareas y/o condiciones que en la prueba se requiera.

Este tipo de prueba tiene el objetivo que el usuario pueda utilizar el sistema de forma eficiente y al mismo tiempo se sienta satisfecho con la velocidad de la aplicación al momento de realizar tareas específicas, para este tipo de pruebas se realizan prueba de carga y prueba de inserción de datos.

5.6.1. Prueba de carga

Este tipo de prueba ayuda a que el sistema rinda adecuadamente y por consecuente verificando que el sistema no haya perdida de información o frustración al usuario al no poder realizar las diferentes funciones que la aplicación ofrece.

En la figura 5.8 se muestra la prueba de carga con la ayuda de la herramienta workbench donde se muestra que con 4 conexiones al servidor simultaneas y se tiene una utilización del 12% en el CPU y un índice de velocidad del 8.01 KB/s de respuesta en cada petición que se realizaba en el sistema.

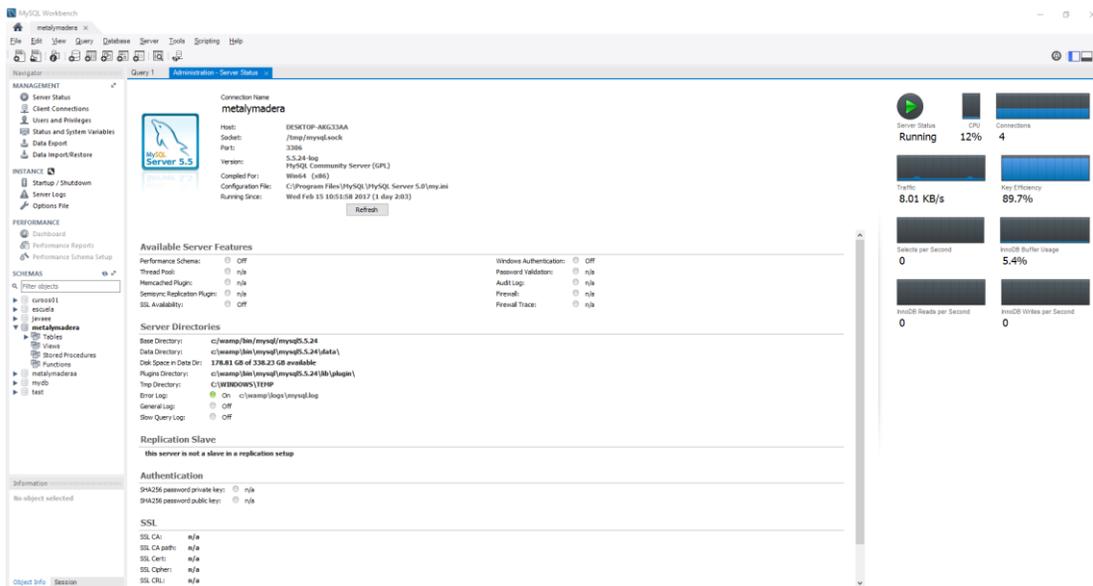


Figura 5.8 Prueba de rendimiento de la BD.

5.7. Prueba de integridad

Se realizaron pruebas de integridad de la base de datos para asegurar que las modificaciones hechas a la base de datos no provoquen la pérdida de consistencia de los datos.

5.7.1. Integridad de dominio

En la siguiente prueba nos permite corroborar que el valor ingresado al campo pertenece al dominio del atributo.

El diccionario de datos es la herramienta más óptima para corroborar que el dato ingresado corresponde al dominio establecido en la tabla número 5.2 se muestra el diccionario de datos de la entidad cliente especificando el nombre del campo, el tipo de dato, el tamaño, si es llave primaria, relación, la restricción de dominio y la descripción.

Tabla 5.2 Diccionario de datos de cliente

Nombre de la tabla: Cliente						
Nombre	Tipo	Lon	Pk	FK	Check	Descripción
idCliente	INT	10	X		0-9	Número identificador de clientes
clt_nombre	Varchar	45			A-Z	Nombre de la empresa
clt_contacto	Varchar	45			A-Z	Nombre del contacto directo
clt_apaterno	Varchar	45			A-Z	Apellido paterno del contacto
clt_amaterno	Varchar	45			A-Z	Apellido materno del contacto
clt_calle	Varchar	45			A-Z	Calle del contacto
clt_numero	Varchar	45			A-Z	Número del contacto
clt_localidad	Varchar	45			A-Z	Localidad del contacto o empresa
clt_estado	Varchar	45			A-Z	Estado del contacto o empresa
clt_cp	INT	5			0-9	Código postal
clt_telefono	Varchar	15			A-Z	Teléfono
clt_celular	Varchar	15			A-Z	Celular
clt_email	Varchar	45			A-Z	Correo electrónico

En la tabla 5.3 se visualiza la prueba de integridad de dominio de la tabla cliente, donde se encuentra: el nombre de campo, tipo de dato solicitado, tipo de dato ingresado calor esperado, valor recibido, resultado.

Tabla 5.3 Prueba de dominio

Nombre de la tabla: Cliente					
Nombre	Tipo de dato Solicitado	Valor ingresado	Valor Esperado	Valor recibido	Resultado
idCliente	INT	1234	Numérico	INT	Correcto
clt_nombre	Varchar	Constructivo M&M	Carácter	Varchar	Correcto
clt_contacto	Varchar	Luis	Carácter	Varchar	Correcto
clt_apaterno	Varchar	Carmona	Carácter	Varchar	Correcto
clt_amaterno	Varchar	Montiel	Carácter	Varchar	Correcto
clt_calle	Varchar	La libertad	Carácter	Varchar	Correcto
clt_numero	Varchar	34-B	Carácter	Varchar	Correcto
clt_localidad	Varchar	Zacatelco	Carácter	Varchar	Correcto
clt_estado	Varchar	Tlaxcala	Carácter	Varchar	Correcto
clt_cp	INT	90678	Numérico	Int	Correcto
clt_telefono	Varchar	(244)464-34-78	Carácter	Varchar	Correcto
clt_celular	Varchar	(246)123-34-56	Carácter	Varchar	Correcto
clt_email	Varchar	luiscmonti@gmail.com	Carácter	Varchar	Correcto

En la tabla número 5.4 se muestra los resultados de la evaluación de integridad de dominio.

Tabla 5.4 Resultados de prueba de dominio

Nombre de la tabla	Resultado
Cliente	Optimo
Cotizador	Optimo
Acabados	Optimo
Tratamiento	Optimo
Productos	Optimo
DetalleCotiza	Optimo

usuarios	Optimo
perfil	Optimo
instalación	Optimo
material	Optimo
madera	Optimo

5.7.2. Integridad de entidad

En la prueba de integridad de entidad se validan las entidades para que toda la tabla tenga clave primaria no null.

Tabla 5.5 Prueba de entidad

Nombre de la tabla	Clave primaria	Null	
		SI	NO
Cliente	idCliente		x
Cotizador	idCotizador		x
Acabados	IdAcabado		x
Tratamiento	idTratamiento		x
Productos	idProducto		x
DetalleCotiza	idDetalleCotiza		x
Usuarios	idUsuarios		x
Perfil	idPerfil		x
Instalación	idInstalacion		x
Material	idMaterial		x
Madera	iddMadera		x

En la tabla número 5.16 podemos observar que todas las tablas cuentan con una clave primaria que servirá como identificador entre relaciones, donde el valor de este campo no se permite que se inserte valor null.

5.7.3. Integridad referencial

En la prueba de integridad referencial se validó que las tablas con referencia tuvieran la sintaxis correspondiente para garantizar la integridad.

En la figura 5.9 Se muestra la sintaxis de la tabla cotizador donde se señalan cada elemento que corresponde a la restricción.

```
CREATE TABLE IF NOT EXISTS `cotizador` (
  `idCotizador` int(11) NOT NULL AUTO_INCREMENT,
  `subfolio` char(2) DEFAULT 'A',
  `nombreCotizador` varchar(255) DEFAULT NULL,
  `fecha_propuesta` date DEFAULT NULL,
  `fecha_mod` date DEFAULT NULL,
  `estado_cot` varchar(45) DEFAULT 'En proceso',
  `total_unitario` double DEFAULT NULL,
  `iva` double DEFAULT NULL,
  `total_civa` double DEFAULT NULL,
  `ptmadproc` double DEFAULT NULL,
  `promvalorPTbruto` double DEFAULT NULL,
  `promvalorPTneto` double DEFAULT NULL,
  `num_vigas` double DEFAULT NULL,
  `mlvigas` double DEFAULT NULL,
  `totaltransporte` double DEFAULT NULL,
  `totalplanta` double DEFAULT NULL,
  `totalsitio` double DEFAULT NULL,
  `totalinge` double DEFAULT NULL,
  `cliente_idCliente1` int(11) NOT NULL,
  `estatus` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`idCotizador`),
  KEY `fk_cotizador_cliente1_idx` (`cliente_idCliente1`),
  CONSTRAINT `fk_cotizador_cliente1` FOREIGN KEY (`cliente_idCliente1`)
REFERENCES `cliente` (`idCliente`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=utf8;
```

Figura 5.9 Sintaxis de creación de tabla con referencia

La columna correspondiente en la clave foránea (cliente_idCliente) y en la clave referenciada (idCliente) tienen los datos similares para que puedan ser comparados sin la necesidad de hacer una conversión de tipos.

5.7.3.1. Inserción de datos con clave foránea

En la prueba de inserción de datos con clave foránea, se tienen por dos tipos de prueba en consola, y en interfaz gráfica, donde se aprecia que los datos son insertados correctamente en cada tabla, en esta prueba todas las inserciones son exitosas, correspondiendo cada fila de la tabla con el dato solicitado.

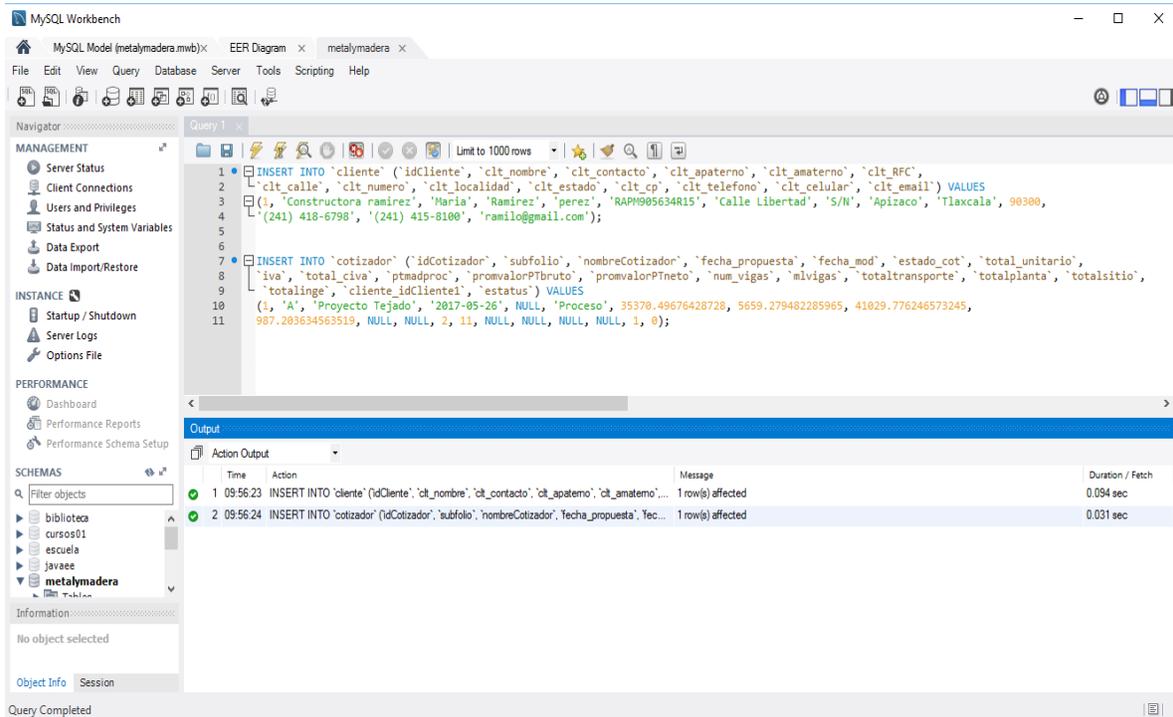


Figura 5.10 Inserción de datos desde consola.

En la figura número 5.10 se muestra una inserción de datos en tabla foránea teniendo como resultado el registro guardado correctamente desde la consola de MySql, y en la figura número 5.11 se muestra una inserción de datos desde la interfaz teniendo como resultado la inserción correcta.

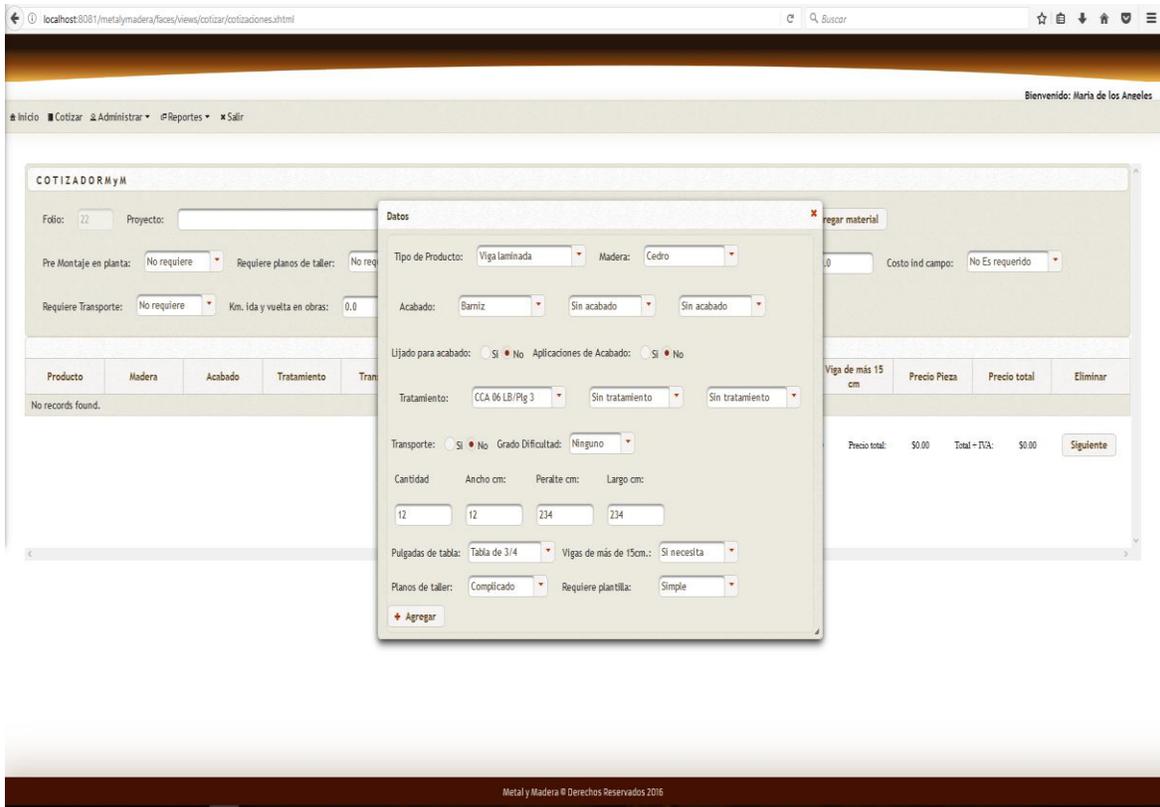


Figura 5.11 Inserción de datos desde interfaz.

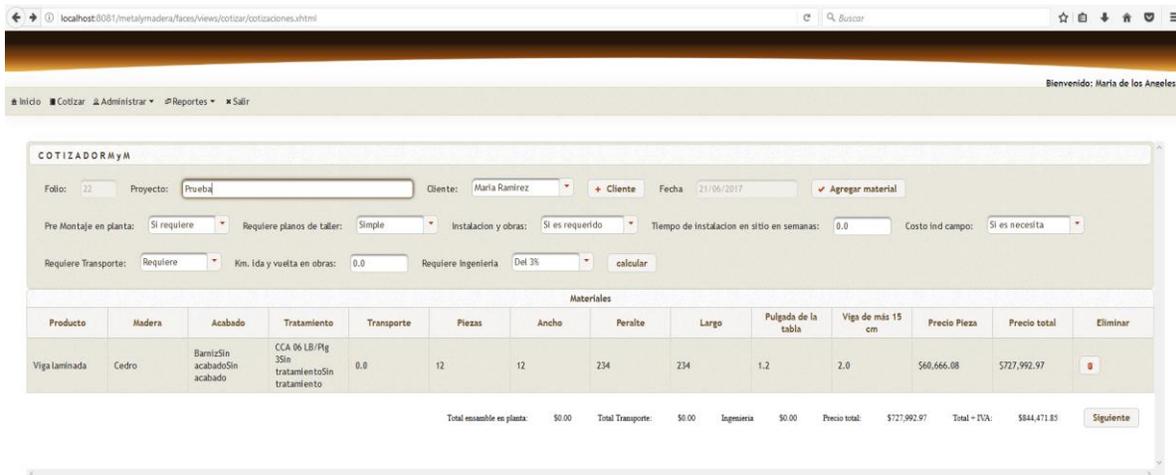


Figura 5.12 Inserción de datos

Durante las revisiones de cada sprint y en el momento de entregar el prototipo del sistema, se realizaron las pruebas de funcionalidad, rendimiento, seguridad, usabilidad e integridad, dando como resultado una retroalimentación para la mejora del mismo.

En la tabla número 5.6 se observan los resultados obtenidos durante la mitad de desarrollo del sistema y en la tabla número 5.7 se visualizan los resultados en la penúltima revisión.

Tabla 5.6 Resultado de pruebas en el sprint 5

Tipo de prueba	Porcentaje
Funcionalidad	50%
Rendimiento	80%
Seguridad	50%
Integración	70%
Usabilidad	70%

Tabla 5.7 Resultado de pruebas en el sprint 8

Tipo de prueba	Porcentaje
Funcionalidad	80%
Rendimiento	95%
Seguridad	90%
Integración	100%
Usabilidad	95%

Cabe señalar que durante el desarrollo del sistema se realizaron las pruebas necesarias para que en cada entrega se obtuvieran resultados favorables, en la tabla 5.8 se muestran los resultados al finalizar el sistema.

Tabla 5.8 Porcentajes de evaluación

Tipo de prueba	Porcentaje
Funcionalidad	100%
Rendimiento	100%
Seguridad	100%
Integración	100%
Usabilidad	100%
Integridad	100%

En la realización de estas pruebas se demuestra que los resultados obtenidos son satisfactorios para el uso del sistema, ya que al realizar este tipo de pruebas se garantiza la confianza del cliente, así como la seguridad de que los datos son guardados y utilizados adecuadamente.

Con las distintas pruebas ayudaron a detectar y corregir diferentes errores en tiempo oportuno teniendo los siguientes beneficios:

- Minimizar las fallas en operaciones básicas para la cotización.
- Un mejor manejo en la calidad de respuesta de una cotización.

Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones después de haber implementado el modelo de base de datos y la integración al sistema cotizador de madera industrial, permitiendo resolver la problemática definida en el apartado 1.1 del capítulo 1 donde se propone agilizar del proceso de elaboración de una cotización, cumpliendo con los objetivos del apartado 1.3, desarrollando un modelo de base de datos para la realización de cotizaciones de madera industrial.

6.1. Conclusiones

En el trabajo se realizó la creación e implementación de un modelo de base de datos para un sistema de cotización de la empresa de madera industrial. Este modelo proporciona al usuario el resguardo de información, así como el manejo de datos para generar cotizaciones de forma rápida, con mayor exactitud en los resultados y proporcionándole al usuario la confianza de que el sistema cumple con el fin para el cual fue creado.

La creación del modelo de base de datos se basó en el patrón de diseño Modelo Vista Controlador y la metodología ágil SCRUM, siendo estas herramientas suficientes para realizar el trabajo planteado.

Tomando en cuenta la metodología mencionada en el capítulo 3 se llevó a cabo el análisis, diseño, implementación y las pruebas para el esquema de la base de datos,

realizando la implementación del sistema de cotización con los módulos de vista y control, así como la aplicación de las pruebas respectivas.

Además, el uso de esta metodología brindó los recursos necesarios para trabajar en el diseño apropiado para el sistema, teniendo tres diseños para su desarrollo: Conceptual, físico y lógico.

Al integrar el sistema de cotización se utilizó el framework *Hibernate* facilitando la conexión y mapeo de los atributos entre la base de datos tradicional y el modelo de objetos que el sistema necesita para su funcionamiento, descrito en el apartado 4.6 del capítulo 4.

Tomando como base el modelo en V mencionado en el apartado 2.10 del capítulo 2, se realizaron las pruebas de funcionamiento, integración, rendimiento y búsqueda, garantizando la calidad del sistema.

Con el sistema desarrollado para las cotizaciones de la empresa de madera industrial, los usuarios pueden realizar las cotizaciones en tiempo y forma, con la información necesaria almacenada en la base de datos.

Por consiguiente, se puede concluir que con lo anterior se da respuesta a la pregunta de investigación afirmando que el modelo de base de datos es capaz de brindar información necesaria para realizar cotizaciones dando resultados precisos, gracias a que se realizan los cálculos utilizando todos los decimales de las operaciones, estos resultados son guardados en la base de datos donde se visualizarán en el momento que el cliente lo solicite.

Al implementar el sistema de cotización se da un gran paso en el avance tecnológico en el estado de Tlaxcala y el país, siendo el cotizador una herramienta que permite agilizar el proceso de ventas de metal y madera, al entregar cotizaciones de acuerdo a las necesidades del cliente, obteniendo más proyectos a construir y con ello generando más empleo para los ciudadanos del estado.

6.2. Trabajos futuros

Dentro del trabajo de investigación se presentan los siguientes temas donde puede existir interés para la ampliación de este trabajo.

- Seguridad: en caso de la información de la base de datos, implementar un método de cifrado para garantizar que los datos no sean leídos por terceros.
- Respaldo en la nube: proporcionando el resguardo de información en caso de un problema de desastre natural u otra índole.

6.3. Experiencia adquirida

Durante la elaboración de este trabajo desarrollé diferentes habilidades como:

- Trabajo en equipo.
- Utilizar una metodología ágil para sistemas de software.
- Trabajar bajo un patrón de diseño.
- Utilizar herramientas framework como son *Hibernate*.

Bibliografía

-
- A, C., S, D., V, R., & J., C. (2007). *Metodologías Ágiles*. Trujillo-peru.
- Abraham Silberschatz, H. F. (2006). *Fundamentos de bases de datos*. España: McGraw-Hill Companies.
- Abram, A., & Moore, J. W. (2004). *SWEBOK*. Los Alamitos, California: IEEE Computer Society.
- Alvarado, I. O. (2011). Medición de la eficiencia de la herramienta gráfica de usuario MyStarSQL, para el sistema gestor de base de datos MS SQL server 5.x con respecto a otras herramientas gráficas. *IEEE, Sección El Salvador*.
- Avila, K. (05 de Mayo de 2016). *¿Qué es un sistema gestor de base de datos o DGBD?* Obtenido de <http://www.cavsi.com/preguntasrepuestas/que-es-un-sistema-gestor-de-base-de-datos-o-sgbd/>
- Begum, S. (abril 8, 2016). Las pruebas utilizando la base de datos web y diagrama RE modelo de transición de estado. *Conferencia internacional sobre comunicación y procesamiento de señal*.
- Bergamaschi, S. (2015). La combinación de usuario y perspectiva de la base de datos de palabras clave para resolver consultas sobre bases de datos relacionales. . *Elsevier*, 1-19.
- Carrion, D. (2010). Un enfoque orientado al diseño de base de dato para la informacion geografica (SIG). *Springer*, 75-84.
- Cloud, L. (12 de Julio de 2016). *Blog sobre Bussiness Intelligence*. Obtenido de Test de calidad para Data Warehouse: <http://www.lantares.com/blog/test-de-calidad-para-data-warehouse-y-bases-de-datos>

- Cobo, A. (2014). *Diseño y programación de bases de datos*. Madrid, España: Visión Libros.
- Computing, O. C. (09 de 09 de 2016). *API de persistencia de Java*. Obtenido de Introducción al API de Java Persistence: <https://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html>
- Crookshanks, E. (2005). *Practical Enterprise Software Development Techniques*. Springer Science + Business Media New York.
- Dimitrieski, V. (2015). Conceptos y evaluación del enfoque entidad – relación extendido al enfoque para el diseño de base de datos en un entorno multi-paradigma en una herramienta de modelado de sistemas de información. *Elsevier*, 299-318.
- El-Hajj, W., & Brahim, G. B. (2016). Seguridad por construcción en desarrollo de aplicaciones web a través de anotaciones de la base de datos. *Elsevier*, 151-165.
- Herrera, H. A., & Valenzuela, C. R. (2016). NoSQL, la nueva tendencia en el manejo de datos. *Faculta de la ingenieria tecnologica investigación academica TIA*, 147-150.
- Hibernate, C. (20 de 05 de 2016). *Hibernate*. Obtenido de <http://hibernate.org/>
- ISO/IEC 9126. (2001). *International Standard, "Information technology Software product evaluation – Quality characteristics and guidelines for their use"*.
- Klochkv, Y., & Klochkova, E. (2016). Modelo de diseño de bases de datos en condiciones de recursos limitados. *h International Conference on Reliability, Infocom Technologies and Optimization IEEE*, 64-66.
- Manager, S. (2015). *Gestión de proyectos Scrum Manager*. Scrum Manager®.
- Martinez, R., & Barrio, P. (10 de Noviembre de 2016). *Sobre Pruebas de software y Servicios de TI*. Obtenido de <http://excelza.blogspot.mx/2009/12/sobre-pruebas-de-software-y-servicios.html>

- Morales, J. D. (12 de Noviembre de 2015). *IBM developerWorks*. Obtenido de Características y tipos de bases de datos: https://www.ibm.com/developerworks/ssa/data/library/tipos_bases_de_datos/
- Reenskaug, T. (27 de Mayo de 2007). *The original MVC reports*. Obtenido de The original MVC reports: http://folk.uio.no/trygver/2007/MVC_Original.pdf
- SCRUMGUIDES. (27 de Mayo de 2016). *¿Qué es SCRUM?* Obtenido de <http://www.scrumguides.org/history.html>
- Singh, P. (2015). Base de datos utilizando el cifrado de seguridad. *Conferencia Internacional sobre tendencias futuristas en análisis computacional y la gestión del conocimiento*, 253-258.
- Sommerville, I. (2011). *Ingeniería del software*. Madrid(España): PERSON EDUCACION S.A.
- Suarez, M. E. (2013). *Diseño e implementación de las bases de datos de un sistema de información para la administración de documentos de calidad*. Tlaxcala.
- Switzerland, A. M. (2015). *Managing Agile Strategy implementation organisation*. New York: Springer cham Heidelberg.
- Valdés, D. P. (26 de 05 de 2016). *¿Qué son las bases de datos?* Obtenido de <http://www.maestrosdelweb.com/principiantes/¿que-son-las-bases-de-datos>.
- Wikimedia, F. (01 de mayo de 2016). *wikipedia*. Obtenido de ACID: <https://es.wikipedia.org/wiki/ACID>
- wordpress. (01 de mayo de 2016). *Fundamentos de la administración de base de datos*. Obtenido de <https://basededatosig.wordpress.com/estructura-de-red/>

ANEXOS

Anexo A Diseño de DBDL

Lenguaje de diseño de base de datos.

```

CREATE TABLE IF NOT EXISTS `cotizador` (
  `idCotizador` int(11) NOT NULL AUTO_INCREMENT,
  `subfolio` char(2) DEFAULT 'A',
  `nombreCotizador` varchar(255) DEFAULT NULL,
  `fecha_propuesta` date DEFAULT NULL,
  `fecha_mod` date DEFAULT NULL,
  `estado_cot` varchar(45) DEFAULT 'En proceso',
  `total_unitario` double DEFAULT NULL,
  `iva` double DEFAULT NULL,
  `total_civa` double DEFAULT NULL,
  `ptmadproc` double DEFAULT NULL,
  `promvalorPTbruto` double DEFAULT NULL,
  `promvalorPTneto` double DEFAULT NULL,
  `num_vigas` double DEFAULT NULL,
  `mlvigas` double DEFAULT NULL,
  `totaltransporte` double DEFAULT NULL,
  `totalplanta` double DEFAULT NULL,
  `totalsitio` double DEFAULT NULL,
  `totalinge` double DEFAULT NULL,
  `cliente_idCliente1` int(11) NOT NULL,
  `estatus` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`idCotizador`),
  KEY `fk_cotizador_cliente1_idx` (`cliente_idCliente1`),
  CONSTRAINT `fk_cotizador_cliente1` FOREIGN KEY (`cliente_idCliente1`)
REFERENCES `cliente` (`idCliente`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=utf8;

```

DBDL para la tabla cotizador

```

CREATE TABLE IF NOT EXISTS `madera` (
  `idMadera` int(11) NOT NULL AUTO_INCREMENT,
  `tipo_mad` varchar(45) DEFAULT NULL,
  `precio_mad` float DEFAULT NULL,
  PRIMARY KEY (`idMadera`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

```

DBDL para tabla madera

```

CREATE TABLE IF NOT EXISTS `material` (
  `idMaterial` int(11) NOT NULL AUTO_INCREMENT,
  `tipo_mat` varchar(45) DEFAULT NULL,
  `precio_mat` float DEFAULT NULL,
  `cantidad` int(11) DEFAULT NULL,
  `unidad_medida_mat` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idMaterial`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

```

DBDL para tabla material

```

CREATE TABLE IF NOT EXISTS `personal` (
  `idPersonal` int(11) NOT NULL AUTO_INCREMENT,
  `tipo_perso` varchar(45) DEFAULT NULL,
  `sueldo_perso` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idPersonal`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

```

DBDL para tabla de personal

```

CREATE TABLE IF NOT EXISTS `producto` (
  `idProducto` int(11) NOT NULL AUTO_INCREMENT,
  `tipo_producto` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idProducto`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

```

DBDL para tabla productos

```

CREATE TABLE IF NOT EXISTS `servicios` (
  `idservicios` int(11) NOT NULL AUTO_INCREMENT,
  `nombre_serv` varchar(45) DEFAULT NULL,
  `porcentaje` float DEFAULT NULL,
  PRIMARY KEY (`idservicios`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

```

DBDL para la tabla servicios

```
CREATE TABLE IF NOT EXISTS `tratamiento` (
  `idTratamiento` int(11) NOT NULL AUTO_INCREMENT,
  `tipo_trata` varchar(45) DEFAULT NULL,
  `precio` float DEFAULT NULL,
  PRIMARY KEY (`idTratamiento`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

DBDL para tabla de tratamiento

```
CREATE TABLE IF NOT EXISTS `usuario` (
  `idUsuario` int(11) NOT NULL AUTO_INCREMENT,
  `usuario` varchar(45) DEFAULT NULL,
  `clave` varchar(45) DEFAULT NULL,
  `user_name` varchar(45) DEFAULT NULL,
  `user_apaterno` varchar(45) DEFAULT NULL,
  `user_amaterno` varchar(45) DEFAULT NULL,
  `user_puesto` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idUsuario`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

DBDL para tabla de usuarios

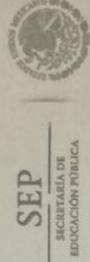
```
CREATE TABLE IF NOT EXISTS `perfil` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) DEFAULT NULL,
  `acabados` tinyint(1) DEFAULT '0',
  `clientes` tinyint(1) DEFAULT '0',
  `material` tinyint(1) DEFAULT '0',
  `madera` tinyint(1) DEFAULT '0',
  `tratamiento` tinyint(1) DEFAULT '0',
  `productos` tinyint(1) DEFAULT '0',
  `respaldo` tinyint(1) DEFAULT '0',
  `usuarios` tinyint(1) DEFAULT '0',
  `rcotizaciones` tinyint(1) DEFAULT '0',
  `oproduccion` tinyint(1) DEFAULT '0',
  `cotfecha` tinyint(1) DEFAULT '0',
  `eliminacot` tinyint(1) DEFAULT '0',
  `cotizar` tinyint(1) DEFAULT '0',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

DBDL para tabla de perfiles

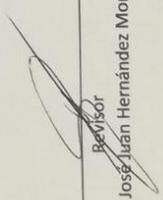
Anexo B Minutas de Sprint



BITÁCORA DE CONTROL
"SISTEMA INFORMÁTICO PARA LA
COTIZACIÓN DE VIGAS DE MADERA
PARA LA EMPRESA METAL Y MADERA"



Fecha	Persona que atendió la visita	Descripción de la visita	Observación hecha por parte de la empresa	Nombre de los participantes en la visita
19-01-2016	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatí Hernández	Primera entrevista realizada para conocer a detalles del sistema a realiza. Se recabo los primeros requisitos del sistema	La empresa proporciono que realizaban en las cotizaciones de los clientes, también sobre que contemplaban para llevar a cabo.	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatí Hernández Lic. Álvaro Cuateta García Ing. María de los A. Hernández Palafox Ing. Luis A. Ocampo Ake



Revisor
M. en C. José Juan Hernández Mora



Encargada del proyecto
Ing. María de los Angeles Hernández Palafox

Minuta A



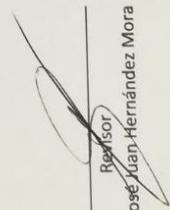
INSTITUTO TECNOLÓGICO DE APIZACO

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



**BITÁCORA DE CONTROL
"SISTEMA INFORMÁTICO PARA LA
COTIZACIÓN DE VIGAS DE MADERA
PARA LA EMPRESA METAL Y MADERA"**

Fecha	Persona que atendió la visita	Descripción de la visita	Observación hecha por parte de la empresa	Nombre de los participantes en la visita
04-02-2016	Lic. María de la Luz Pérez García	Se presenta una revisión a fondo de los requisitos para cotizar una Viga Recta, Viga l y Viga Curva	La empresa proporciono los puntos que se deben considerar para chorizar los productos.	Lic. María de la Luz Pérez García Lic. Álvaro Cuateta García Ing. María de los A. Hernández Palafox Ing. Luis A. Ocampo Ake



Revisor
M. en C. José Juan Hernández Mora



Encargada del proyecto
Ing. María de los Angeles Hernández Palafox

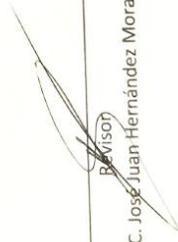
Minuta B



BITÁCORA DE CONTROL
"SISTEMA INFORMÁTICO PARA LA
COTIZACIÓN DE VIGAS DE MADERA
PARA LA EMPRESA METAL Y MADERA"



Fecha	Persona que atendió la visita	Descripción de la visita	Observación hecha por parte de la empresa	Nombre de los participantes en la visita
26-02-2016	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatl Hernández	Se le presentó a la empresa la primera documentación de los requerimientos recabados para realizar una cotización	La empresa agrego campos de transporte para acabados y tratamientos especiales.	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatl Hernández Lic. Álvaro Cuateta García Ing. María de los A. Hernández Palafox Ing. Luis A. Ocampo Ake



M. en C. José Juan Hernández Mora



Encargada del proyecto
Ing. María de los Angeles Hernández Palafox

Minuta C



INSTITUTO TECNOLÓGICO DE APIZACO

BITÁCORA DE CONTROL
"SISTEMA INFORMÁTICO PARA LA
COTIZACIÓN DE VIGAS DE MADERA
PARA LA EMPRESA METAL Y MADERA"



SEP
 SECRETARÍA DE
 EDUCACIÓN PÚBLICA

Fecha	Persona que atendió la visita	Descripción de la visita	Observación hecha por parte de la empresa	Nombre de los participantes en la visita
14-03-2016	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatí Hernández	Se entregó por parte de la empresa un documento que se genera para la cotización	La empresa pidió que se realizara un formato nuevo para imprimir las cotizaciones tomando en cuenta el original y actual formato.	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatí Hernández Lic. Álvaro Cuateta García Ing. María de los A. Hernández Palafox Ing. Luis A. Ocampo Ake

Revisó

M. en C. José Juan Hernández Mora

Encargada del proyecto

Ing. María de los Ángeles Hernández Palafox

Minuta D

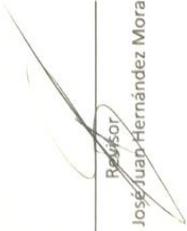


BITÁCORA DE CONTROL
"SISTEMA INFORMÁTICO PARA LA
COTIZACIÓN DE VIGAS DE MADERA
PARA LA EMPRESA METAL Y MADERA"



SEP
 SECRETARÍA DE
 EDUCACIÓN PÚBLICA

Fecha	Persona que atendió la visita	Descripción de la visita	Observación hecha por parte de la empresa	Nombre de los participantes en la visita
22-03-2016	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatl Hernández	Se hizo entrega de la primera caratula para las impresiones de cotizaciones	La empresa pidió que algunos campos sean eliminados para no mostrar al cliente y agregar nuevos campos, como PT neto, PT bruto y total unitario.	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatl Hernández Lic. Álvaro Cuateta García Ing. María de los A. Hernández Palafox Ing. Luis A. Ocampo Ake



Revisor
M. en C. José Juan Hernández Mora

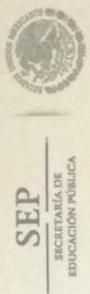


Encargada del proyecto
Ing. María de los Angeles Hernández Palafox

Minuta E



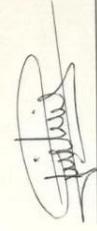
**BITÁCORA DE CONTROL
"SISTEMA INFORMÁTICO PARA LA
COTIZACIÓN DE VIGAS DE MADERA
PARA LA EMPRESA METAL Y MADERA"**



Fecha	Persona que atendió la visita	Descripción de la visita	Observación hecha por parte de la empresa	Nombre de los participantes en la visita
11-04-2016	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatí Hernández	Se hizo entrega de la caratula para las impresiones de cotizaciones	La empresa solicito generar una caratula para impresiones futuras de orden de producción.	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatí Hernández Lic. Álvaro Cuateta García Ing. María de los A. Hernández Palafox Ing. Luis A. Ocampo Ake



Revisor
M. en C. José Juan Hernández Mora



Encargada del proyecto
Ing. María de los Angeles Hernández Palafox

Minuta F



BITÁCORA DE CONTROL
"SISTEMA INFORMÁTICO PARA LA
COTIZACIÓN DE VIGAS DE MADERA
PARA LA EMPRESA METAL Y MADERA"



Fecha	Persona que atendió la visita	Descripción de la visita	Observación hecha por parte de la empresa	Nombre de los participantes en la visita
02-05-2016	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatl Hernández	Se hizo entrega del primer prototipo del sistema	La empresa pidió que se agregaran nuevos campos para llevar a cabo una cotización y se entregaron nuevos requisitos del sistema	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatl Hernández Lic. Álvaro Cuateta García Ing. María de los A. Hernández Palafox Ing. Luis A. Ocampo Ake



Revisor
M. en C. José Juan Hernández Mora



Encargada del proyecto
Ing. María de los Ángeles Hernández Palafox

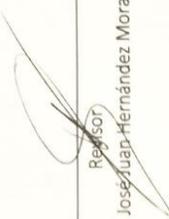
Minuta G



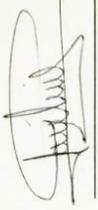
BITÁCORA DE CONTROL
"SISTEMA INFORMÁTICO PARA LA
COTIZACIÓN DE VIGAS DE MADERA
PARA LA EMPRESA METAL Y MADERA"



Fecha	Persona que atendió la visita	Descripción de la visita	Observación hecha por parte de la empresa	Nombre de los participantes en la visita
17-05-2016	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatli Hernández	Se hizo entrega de correcciones del prototipo de la interfaz y el análisis de los nuevos requerimientos	La empresa pidió que se agregaran nuevos campos para llevar a cabo una cotización y que los colores de la interfaz sean acorde a la empresa. No incluir imágenes en las cotizaciones.	Lic. María de la Luz Pérez García Cont. Edgar Cuahutecatli Hernández Lic. Álvaro Cuateta García Ing. María de los A. Hernández Palafox Ing. Luis A. Ocampo Ake



Revisor
M. en C. José Juap. Hernández Mora



Encargada del proyecto
Ing. María de los Ángeles Hernández Palafox

Minuta H

Anexo C Artículos publicados

ACADEMIA JOURNALS



Primer Congreso Internacional de Academia Journals
en Educación Superior Tecnológica Pública
Tlaxcala 2016

Certificado

otorgado a

Ing Luis Alfonso Ocampo Ake
Ing María de los Ángeles Hernández Palafox
M.C José Juan Hernández Mora
M.C Guadalupe Medina Barrera

por su artículo intitulado

Desarrollo de una aplicación enfocado en el modelo vista controlador para cotización de servicios en la empresa Metal y Madera Srl de C.V

(Artículo No. Tlax255)

el cual fue presentado en el Congreso desarrollado del 16 al 18 de marzo 2016 en Tlaxcala, México,
y publicado en el portal de Internet *AcademiaJournals.com*,
con número de registro ISSN 1946-5351, VOL. 8, NO. 2
y en el libro electrónico online con ISBN 978-1-939982-21-6 intitulado
Compendio de Investigación Academia Journals Tlaxcala 2016

Dr. Rafael Moras
Editor, Academia Journals
Profesor de Ing. Industrial, St. Mary's University



Desarrollo de una aplicación enfocado en el modelo vista controlador para cotización de servicios en una empresa de madera industrial

Ing. Luis Alfonso Ocampo Ake¹, Ing. María de los Ángeles Hernández Palafox²,
M en C. José Juan Hernández Mora³ y M en C. María Guadalupe Medina Barrera⁴

Resumen— Dentro del marco de desarrollo tecnológico, se encuentra la necesidad de desarrollar un sistema cotizador a la medida para una empresa de maderas industriales, para llevar a cabo el desarrollo de la aplicación se utiliza el modelo vista controlador (MVC) que permite separar el proyecto en tres diferentes módulos cada uno especializado en su área explicita, el controlador es el encargado de interpretar las solicitudes del usuario, la vista proyecta la salida gráfica o textual al usuario y por último el modelo encargado del comportamiento y los datos del dominio de aplicación, así mismo se complementa con una metodología de desarrollo ágil “SCRUM” el cual trabaja colaborativamente en equipo para obtener el mejor resultado de un proyecto.

Palabras clave— MVC, SCRUM, UML, MODELADO, SERVLET.

Introducción

Dentro de la empresa en la industria de la madera se realiza el proceso de la venta de vigas para llevar a cabo proyectos de diferentes magnitudes, al mismo tiempo se realizan proyectos de construcción para diferentes empresas en diferentes estados del país y en otros países. La empresa tiene un problema de la cual se necesita una solución, por lo que se propone la creación de un sistema donde ayude a realizar las cotizaciones para los clientes de una manera más rápida, eficiente y a la medida. Con el fin de facilitar al usuario la manipulación del sistema se contempla el uso del modelo vista controlador para la realización del mismo y con ayuda de una metodología ágil.

En el marco de la empresa se realizan el proceso de la venta de vigas para llevar a cabo proyectos de diferentes magnitudes, al igual se realizan construcciones para diferentes empresas en diferentes estados del país y en otros países como Alemania, Italia entre otros.

En la actualidad se maneja un sistema realizado en Excel en donde se apoyan para llevar a cabo la dicha acción, pero se contempla la mejorar del proceso con el desarrollo de un nuevo sistema de software, ya que los encargados de convencer al cliente desean un software que se pueda utilizar desde cualquier parte en donde se encuentren estando dentro y fuera de la empresa.

El sistema a crear se pretende establecer en un servidor web para tener acceso en diferentes puntos de los países en donde se encuentre el operador de dicho sistema mediante el uso de internet.

Se contemplan diferentes tipos de Usuarios y sus respectivas Contraseñas para el control al sistema de ventas, dentro de estos usuarios existe el operador se encargara de llevar a cabo las cotizaciones para algún cliente interesado, este sin permisos para hacer modificaciones de acuerdo a los precios de las ventas, el administrador del sistema además de poder llevar a cabo cotizaciones podrá tener acceso a la edición de un precio y la eliminación de un producto o servicios, así como también ambos usuarios podrá generar los reportes necesarios o de acuerdo a las necesidades de cada personal.

¹ Ing. Luis Alfonso Ocampo Ake Egresado del Instituto Tecnológico superior de Felipe Carrillo Puerto, estudiante de la Maestría en Sistemas Computacionales en el Instituto Tecnológico de Apizaco. ocampo.luis.alfonso@gmail.com (autor correspondiente).

² Ing. María de los Angeles Hernández Palafox Egresado del Instituto Tecnológico de Apizaco, estudiante de la Maestría en Sistemas Computacionales del mismo. mariaangeleshp@gmail.com (autor correspondiente).

³ M en C. José Juan Hernández Mora profesor de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco jhmora@itapizaco.edu.mx.

⁴ M en C. Ma. Guadalupe Medina Barrea profesor de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco. lupita_medina@hotmail.com.

Descripción del Método

Programación de un sistema web

Para el desarrollo del software se realizó un estudio profundo en el giro de la empresa para obtener los requerimientos y datos necesarios para el llevar a cabo el desarrollo del sistema.

Para llevar a cabo esta aplicación se debe contar con una seguridad de la información, organizada y mostrada, con el modelo de la aplicación que será capaz de mostrar información confiable apoyada de una base de datos propia, donde contendrá información donde almacene dichos datos.

Hay que considerar dentro del desarrollo de software las diferentes formas de describir la información y para ello existen tres diferentes visiones.

Visión conceptual: Es la que representa el problema real, define la estructura de toda la base.

Visión externa: es la vista final que tiene el usuario

Visión física: es la representación del almacenamiento de los datos, la estructura física de los datos.

En el proceso de diseño existen tres actividades esenciales durante el proceso de análisis:

1. Análisis del usuario: Se desarrolla una comprensión de las tareas que realiza, el entorno del trabajo, los sistemas que utiliza etc.
2. Prototipo del sistema: El diseño y desarrollo es un proceso iterativo, aunque los usuarios pueden hablar de los que necesitan en el sistema.
3. Evaluación de la interfaz: Evaluación más formalizada sobre las experiencias reales. [1]

Para utilizar el MVC⁵ efectivamente se debe entender la división del trabajo dentro de la triada. También se debe comprender cómo las tres partes se comunican entre sí y con otros puntos de vista activos y controladores; el intercambio de una pantalla, mouse, teclado y la visualización entre varias aplicaciones exige la comunicación y la cooperación para hacer el mejor uso del modelo. [2]

Desarrollo de una GUI⁶

El desarrollo de la interfaz se implementará utilizando un *servlet* de java. Un *servlet* es una aplicación que se ejecuta en el servidor aumentando su funcionalidad, facilitando al programador el desarrollo de aplicaciones al sistema, dejando de lado del servidor la ejecución de la aplicación. [3]

El usuario accedera desde un URL al dominio contenedor del *Servlet* al que ingresara con un nombre de usuario y contraseña asignados previamente, dentro de la integración se pueden realizar tareas de alta de materiales y servicios, operaciones básicas para el cálculo de la cotización, así mismo como bajas y modificaciones de los servicios.

El menú principal será desarrollado utilizando un *Framework* de java nombrado *Primefaces* para presentar de una manera amigable las funcionalidades. *Primefaces* es una librería de componentes para *JavaServer Faces (JSF)* de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. [4]

Las ventajas de la utilizar *primefaces* son:

- Soporte nativo a Ajax
- Es compatible con otras librerías de componentes como *JbossRichFaces*
- Uso de JavaScript no intrusivo
- Es un proyecto open source, activo y estable

En el modelado del sistema se contempla la herramienta MySQL ya que algunas de sus características son:

- Es muy destacable su velocidad de respuesta.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el sistema gestor de base de datos a cada caso concreto.
- Es altamente confiable en cuanto a estabilidad.
- Su administración se basa en usuarios y privilegios.
- Optimizado para equipos de múltiples procesadores. [5]

⁵ MVC: Modelo, Vista, Controlador

⁶ GUI: *Graphical user interface*, es un programa informático que actúa de interfaz de usuario.

En la figura 1 se presenta la integración de los componentes de cada etapa del modelo.



Fig1. Estructura General del Modelo Vista Controlador

En la Tabla 1 observamos la descripción de cada uno de los actores principales que manipularan el sistema, como se nota se cuenta con un administrador, una jefe de área y el operador que cada uno de ellos tienen diferentes privilegios dentro del uso del sistema.

Código	Autor	Descripción
Admon.	Administrador	Usuario administrador del sistema el cual posee todos los permisos otorgados por el sistema a fin de realizar el mantenimiento al sistema.
JefeArea	Jefe del área	Usuario administrador de un apartado específico el cual tiene la capacidad de dar descuentos, y modificaciones de algunos campos con restricción.
Operador	Operador	Funcionario del sistema responsable de realizar las cotizaciones y visualización de reportes.

Tabla 1.- Autores de caso de uso

Especificaciones para la generación de la base de datos.

Dentro del proyecto la seguridad es muy importante, por lo que se optó por la utilización de métodos de encriptación, que aseguran los datos guardados.

La seguridad UML permite formalizar las decisiones de control de acceso que depende de dos tipos de información

1. Información estática: es decir la asignación de roles y permisos a los usuarios y jerarquías de funciones

2. Información dinámica: la satisfacción de restricciones de autorización del estado actual del sistema. En esta metodología se utiliza seguridad UML para especificar políticas de seguridad más componentes modelos UML. [5]

Modelado de requerimientos

El proceso de ingeniería de software seguro implementa UML a fin de abordar las necesidades de un proyecto de software. El UML permite una modelación de los escenarios de una aplicación. Algunos de los diagramas más utilizados dentro del análisis son: Diagramas de casos de uso, de clases, actividades, secuencia etc. [6]

El modelo impulsado a la seguridad se ha convertido en un área de investigación durante la última década. Propone un nuevo conjunto de anotación que extiende el UML (Unified Modeling Language) La nueva notación comprende el conjunto de semántica necesaria de un estado basado en el lenguaje de modelado de seguridad y fue en gran parte gracias a su uso intuitivo y comprender proporcionado muy poca formación. Una ventaja de utilizar la notación es que puede ser especificado en meta modelo subyacente del estado - gráficos, por ende, permitiendo más en profundidad del análisis formal de los modelos. [5]

Resultados

Después de haber lleva el análisis de requerimientos y funcionalidades del sistema en base al modelo vista controlador, llegamos a los siguientes resultados.

La figura 2 nos muestra un diagrama general de casos de uso del sistema donde notamos los privilegios que permiten el acceso correspondiente al uso de operaciones que tiene el sistema.

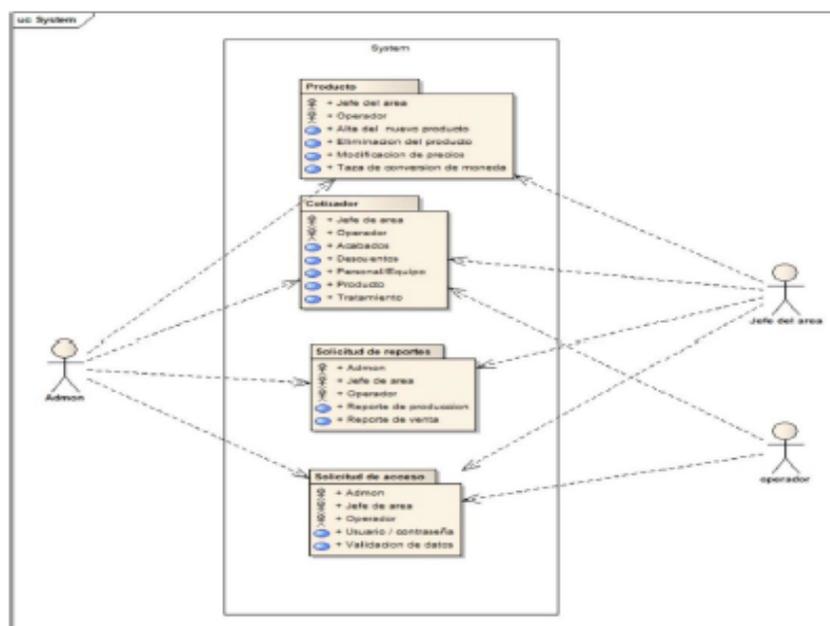


Fig2 Diagrama general de casos de uso

En la figura 3 se muestran algunas de las clases principales que utiliza el sistema de cotización, en este caso el material, los servicios y los usuarios tienen un rol muy importante para la generación de la cotización.

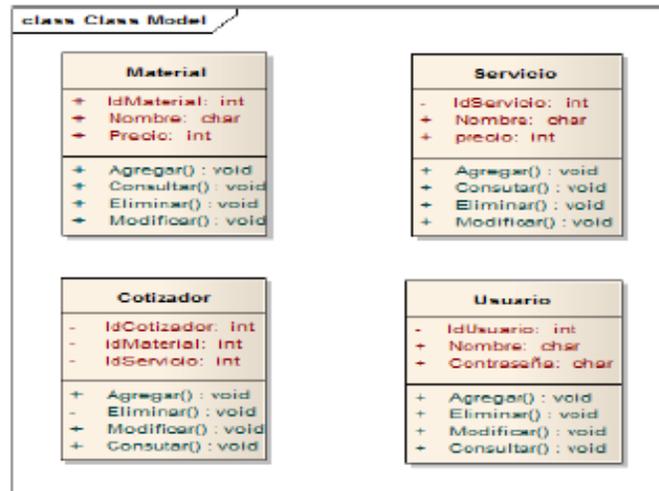


Fig. 3.- Diagrama de clases

Solo se presentaron algunos diagramas importantes del sistema ya que por reserva industrial no se pueden presentar todos los diagramas del sistema.

Conclusiones y trabajos futuros

Para concluir podemos hacer mención de que en el uso de la metodología, el modelado y la utilización del modelo vista controlador, proporcionan las herramientas adecuadas en la obtención de información para el desarrollo del sistema, donde se contemplan aspectos y necesidades que el usuario final necesita.

Al igual concluimos que con la información obtenida se pueden generar un sistema web para realizar las operaciones necesarias para generar una cotización.

Referencias

- [1] I. SOMMERVILLE, Ingeniería del software, Madrid España: Pearson Educación S.A., 2005.
- [2] P. Steve Burbeck, «Applications Programming in Smalltalk-80: How to use Model-View-Controller (MVC)» by S. Burbeck, 1992.
- [3] J. H. w. W. Crawford, Java Servlet Programming, 2000.
- [4] M. A. G.-R. Fernando Pech-May, «Desarrollo de Aplicaciones web con JPA, EJB, JSF y primefaces.» 2004.
- [5] M. G. G. Ó. P. M. Luis Alberto Casillas Santillán, Base de Datos en MySQL, UOC, 2003.
- [6] M. E. attar, «Extending the UML Statecharts Notation to Model Security Aspects.» *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, p. 661, JULY 2015.
- [7] G. R. J. y. J. I. Booch, «Unified Modeling Language User Guide.» *Addison-Wesley Object Technology Series*, 2005.

N° 1096



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE CELAYA



CONGRESO INTERNACIONAL DE INVESTIGACIÓN DE ACADEMIA JOURNALS.COM, CELAYA 2016

OTORGAN EL PRESENTE

CERTIFICADO

A

ING. MARÍA DE LOS ÁNGELES HERNÁNDEZ PALAFOX
ING. LUIS ALFONSO OCAMPO AKE
M. EN C. JOSÉ JUAN HERNÁNDEZ MORA
M. EN C. MARÍA GUADALUPE MEDINA BARRERA

POR SU PARTICIPACIÓN CON LA PONENCIA TITULADA
**DESARROLLO DE UNA APLICACIÓN UTILIZANDO LA
METODOLOGÍA SCRUM Y EL PATRÓN DE DISEÑO
MVC PARA LA COTIZACIÓN DE SERVICIOS EN UNA
EMPRESA DE MADERA INDUSTRIAL**

LA CUAL FUE PRESENTADA LOS DÍAS 9, 10 Y 11 DE NOVIEMBRE DE 2016,
EN EL INSTITUTO TECNOLÓGICO DE CELAYA, GUANAJUATO, MÉXICO.
PUBLICADA EN EL PORTAL DE INTERNET CELAYA.ACADEMIAJOURNALS.COM
MEMORIA ONLINE CON ISSN 1946-5351, VOL. 8, NO. 5 Y LIBRO DIGITAL CON
NÚMERO DE REGISTRO ISBN 978-1-939982-26-1, CON CÓDIGO DE BARRAS.

DR. RAFAEL MORAS
EDITOR, ACADEMIAJOURNALS.COM
PROFESOR DE INGENIERÍA INDUSTRIAL Y ADMINISTRATIVA
ST. MARY'S UNIVERSITY, SAN ANTONIO, TX. EEUU

ING. ALEJANDRO ALVAREZ BARCENAS
COORDINADOR GENERAL DEL
CONGRESO INTERNACIONAL DE INVESTIGACIÓN
ACADEMIA JOURNALS, CELAYA 2016

CEL1321



Desarrollo de una aplicación utilizando la metodología SCRUM y el patrón de diseño MVC para la cotización de servicios en una empresa de madera industrial

Ing. María de los Ángeles Hernández Palafox¹, Ing. Luis Alfonso Ocampo Ake²,
M. en C. José Juan Hernández Mora³ y M. en C. María Guadalupe Medina Barrera⁴

Resumen— El siguiente trabajo aborda el tema sobre el uso de la metodología SCRUM trabajando en conjunto con el patrón de diseño MVC, para la elaboración de un modelo de bases de datos, para un sistema cotizador a la medida de madera industrial. Con la ayuda de la metodología y el patrón de diseño, permite separar en tres módulos el sistema cada uno con su área explícita, el modelo fue desarrollado paralelamente e implementado con la vista y control del sistema.

Palabras clave— Metodología SCRUM, Patrón de diseño MVC (Modelo - vista- controlador), Modelo de base de datos, sistema cotizador.

Introducción

Hoy en día la mayoría de las aplicaciones contienen un modelo de base de datos, para la administración y resguardo de la información, así para el funcionamiento del sistema. Para la aplicación de cotización de servicios de la empresa de madera industrial, tienen la necesidad de guardar sus datos para la manipulación de los mismos, por lo que se desarrolla un modelo con ayuda de la metodología SCRUM y el patrón de diseño MVC, el MVC (Modelo- vista- controlador) permite la separación del sistema por módulos y trabajar conjuntamente con un equipo de trabajo, con la metodología nos permite la obtención de requerimientos así como la evaluación del sprint en un ciclo de tiempo definido. (Steve Burbec, 1992).

En este caso la base de datos trabaja en conjunto con el módulo de vista y el módulo de control, lo que esto permite optimizar el tiempo de entrega de la aplicación.

Complementando con la competencia que existe en el mercado, se tiene la necesidad de desarrollar un sistema que sea rápido eficiente y eficaz y así la empresa cuente con una ventaja para aumentar su productividad y satisfacer las necesidades de sus clientes en el menor tiempo posible. Para el desarrollo se requieren metodologías que garanticen la fiabilidad del software y las metodologías ágiles han ganado bastante popularidad desde hace algunos años, en la ingeniería del software juega un papel muy importante en el desarrollo, portabilidad, mantenibilidad, funcionalidad, fiabilidad y productividad del software (Urquiza Yllescas J. F., 2010).

Descripción del Método

Definición del problema

En la empresa de madera industrial realiza el proceso de venta de vigas de madera laminada, para proyectos de diferentes magnitudes. La empresa tiene el problema de necesitar un sistema cotizador que ayude a realizar este proceso de manera rápida y eficiente.

En la actualidad se maneja un sistema realizado en Excel en donde se apoya para llevar a cabo las cotizaciones solicitadas, al finalizar la cotización se procede a la impresión, entrega y guardo en diferentes carpetas.

Al manejar un número considerable de impresiones, la empresa tiene el problema de: alteración de cotizaciones, extravío de información, consulta lenta, duplicado de información.

Para la problemática se contempla la creación de una nueva aplicación donde los usuarios puedan almacenar toda esta información en una base de datos, el sistema estará en un servidor web donde se tenga acceso en cualquier punto del país para que el usuario realice la operación de forma rápida y segura.

Para la seguridad dentro del sistema los usuarios tendrán un nombre de identificación y una contraseña para que cada usuario tenga los privilegios que su puesto lo demande.

¹ Ing. María de los Ángeles Hernández Palafox Egresado del Instituto Tecnológico de Apizaco, estudiante de la Maestría en Sistemas Computacionales del mismo. mariaangeleshp@gmail.com (autor corresponsal).

² Ing. Luis Alfonso Ocampo Ake Egresado del Instituto Tecnológico superior de Felipe Carrillo Puerto, estudiante de la Maestría en Sistemas Computacionales en el Instituto Tecnológico de Apizaco. ocampo.luis.alfonso@gmail.com

³ M en C. José Juan Hernández Mora profesor de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco.

⁴ M en C. María Guadalupe Medina Barrea profesor de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco.

Metodología propuesta

Un factor importante para la aplicación de cotización es optimizar el acceso que se tiene a los datos, la distribución y el manejo de los mismos, brindando seguridad y eficiencia en el manejo a la hora de obtención de la información para realizar una operación por ejemplo una búsqueda, eliminación, actualización o inserción. Para que este proceso se realice adecuadamente se utiliza el patrón de diseño MVC que nos permite dividir el sistema permitiendo su desarrollo de forma independiente en cada capa sin dejar a lado la funcionalidad y la metodología SCRUM establecemos las etapas para el desarrollo de cada módulo.

SCRUM, se basa en la teoría del control empírico de los procesos, emplea un enfoque iterativo e incremental para optimizar la previsibilidad y controlar los riesgos (scrumguides, 2016).

En la figura 1 nos muestra la metodología que se sigue para el desarrollo de la aplicación, a partir de SCRUM, donde está conformada por la pila del producto que es donde se desglosa la entrevista con el cliente, la recopilación de requisitos y en la pila de sprint se realiza el análisis, diseño, codificación, evaluación y pruebas teniendo iteraciones de 30 días.

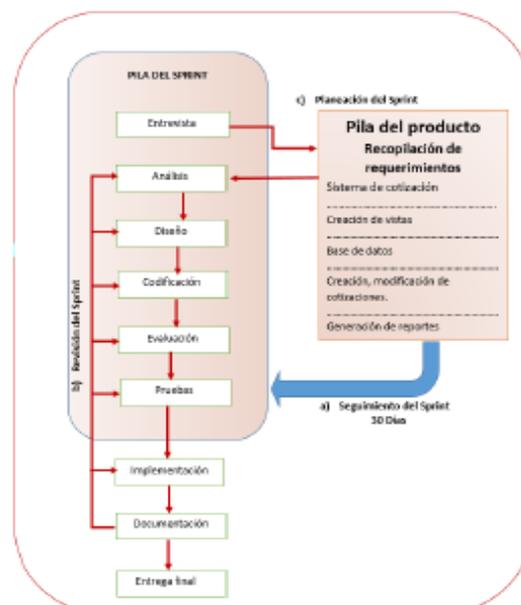


Figura: 1 Metodología para el desarrollo de la base de datos, bajo la metodología SCRUM

Para el desarrollo de la base de datos tomando como base el patrón de diseño MVC y la metodología SCRUM siguiendo las fases de planeación y seguimiento se desprende específicamente la metodología de diseño para el modelo de base de datos mostrada en la figura 2

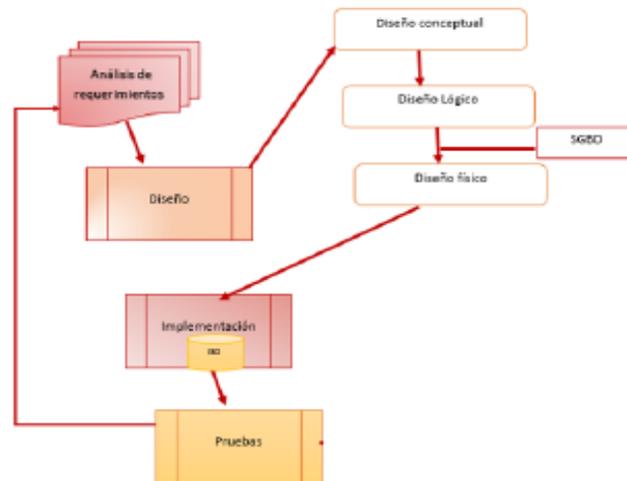


Figura: 2 Metodología para la creación de la base de datos

En el proceso de la base de datos se realiza el diseño en tres fases (Costa, 2015):

Diseño conceptual: Con la información analizada en el paso anterior, se procede a plasmarla en términos de base de datos, en un nivel que el cliente la pueda entender, es decir reflejamos como son los datos exponiéndolos en un diagrama entidad – relación, esta información se muestra en la figura 3.

Diseño lógico: En este diseño es la transformación del diseño conceptual, utilizando un modelo de datos lógico en este caso el modelo lógico relacional. En este diseño nos acercamos a la generación de tablas para poder ser implementadas en un sistema gestor de base de datos.

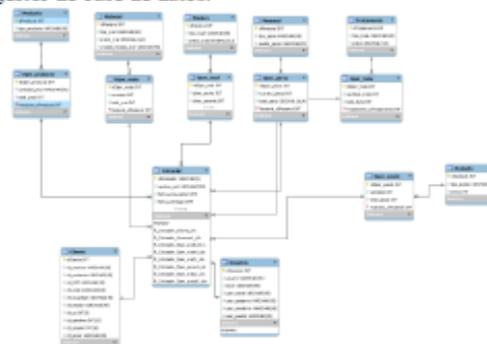


Figura: 3 Diagrama Entidad-Relación para sistema cotizador

Diseño físico: dentro de este diseño establecemos como se van a almacenar los datos y se elige el tipo de datos correspondiente, se definen las estructuras de almacenamiento y de acceso para alcanzar un rendimiento óptimo de las aplicaciones de la base de datos, dentro del diseño físico se implementa la base de datos en un sistema gestor de base de datos (SGBD) y se define la estrategia de acceso.

El sistema gestor de base de datos: Sistema donde se plasmará de diseño lógico es una herramienta donde se consideran diferentes factores técnicos, económicos y beneficios, para el rendimiento oportuno a la necesidad del cliente.

En el modelado del sistema se contempla la herramienta MySQL ya que algunas de sus características son (attar, 2015):

- Es muy destacable su velocidad de respuesta.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el sistema gestor de base de datos a cada caso concreto.
- Es altamente confiable en cuanto a estabilidad.
- Su administración se basa en usuarios y privilegios.
- Optimizado para equipos de múltiples procesadores.

En la figura 4 se muestra la integración de las tres capas del patrón de diseño MVC, donde la capa de vista del usuario realiza una petición al controlador, este la procesa encargándose de llamar a los datos que tiene la base de datos del modelo una vez que tiene la información, se encarga de terminar el proceso y muestra la información al usuario.



Figura: 4 MVC del sistema cotizador.

Resultados

Para el desarrollo de la base de datos, se trabajó con iteraciones con revisión cada 30 días, el equipo de desarrollo para el sistema es integrado por 3 personas, una persona para el desarrollo de cada módulo, en la revisión de cada iteración se analizaba que cada requerimiento se concluyera satisfactoriamente.

La administración de la base de datos se realizó con el software *workbench* es una herramienta de base de datos, donde nos permite desarrollar y administrar las bases de datos, además de permitir crear, ejecutar y optimizar consultas SQL. (Oracle Corporation, 2016).

En la figura 5 se realiza una consulta a la base de datos desde consola mostrando el rendimiento y estabilidad de los datos.

```

C:\Users\j...>mysql -u root -p
mysql> use cotizadora;
mysql> select * from productos;
+-----+-----+-----+-----+-----+-----+
| producto | precio | cantidad | fecha_producto | fecha_producto | cantidad |
+-----+-----+-----+-----+-----+-----+
| 1 | 1000 | 100 | 2016-11-09 | 2016-11-09 | 100 |
| 2 | 2000 | 50 | 2016-11-10 | 2016-11-10 | 50 |
| 3 | 3000 | 20 | 2016-11-11 | 2016-11-11 | 20 |
| 4 | 4000 | 10 | 2016-11-12 | 2016-11-12 | 10 |
| 5 | 5000 | 5 | 2016-11-13 | 2016-11-13 | 5 |
+-----+-----+-----+-----+-----+-----+
mysql> select * from cotizaciones;
+-----+-----+-----+-----+-----+-----+
| cotizacion | fecha_producto | fecha_producto | cantidad |
+-----+-----+-----+-----+-----+-----+
| 1 | 2016-11-09 | 2016-11-09 | 100 |
| 2 | 2016-11-10 | 2016-11-10 | 50 |
| 3 | 2016-11-11 | 2016-11-11 | 20 |
| 4 | 2016-11-12 | 2016-11-12 | 10 |
| 5 | 2016-11-13 | 2016-11-13 | 5 |
+-----+-----+-----+-----+-----+-----+

```

Figura: 5 consulta de la base de datos.

En la figura 6 se muestra los resultados de la integración con los módulos de vista donde el usuario interactúa con el sistema.

En la figura 7 Se muestra la integración con la capa de control para la manipulación de los datos para realizar las diferentes cotizaciones que los clientes soliciten.

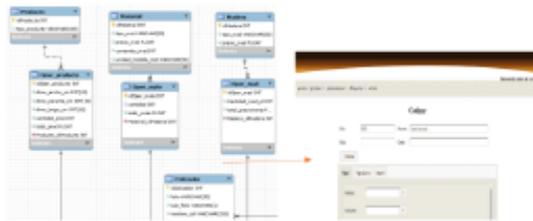


Figura: 6 Integración de la base de datos con interfaz.

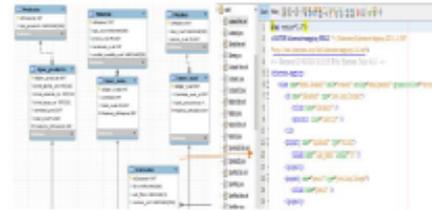


Figura: 7 Integración de la base de datos con el controlador.

Conclusiones

La experiencia de trabajar en un equipo utilizando el patrón de diseño MVC y la metodología SCRUM para el desarrollo de una base de datos para el sistema cotizador, demuestra que el trabajo es favorecido para encontrar resultados en un periodo de tiempo menor, donde a partir de los sprint revisados cada 30 días nos permite resultados más eficaces, flexibles, adaptables a cambios y resultados anticipados.

Aunque en algunas ocasiones surgieron dificultades por la falta de conocimiento, la metodología nos permitió que en cada revisión se reforzaran los puntos débiles o los que no cumplían con un requisito en específico.

Lo más importante es la comunicación en el grupo de trabajo y con el cliente para que cada requisito funcione correctamente, respecto con los resultados obtenidos al aplicar SCRUM con el patrón de diseño MVC son los cambios y observaciones en tiempo y forma para el correcto desarrollo del proyecto.

Referencias

Oracle Corporation. (20 de abril de 2016). *workbench*. Obtenido de <https://www.mysql.com/products/workbench/>

attar, M. (JULY 2015 de 2015). Extending the UML statecharts Notation to model security aspects. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 661.

Costa, D. C. (2015). *Introducción al diseño de base de datos*.

scrumguides. (27 de mayo de 2016). *¿Que es scrum?* Obtenido de <http://www.scrumguides.org/history.html>

Steve Burbeck, P. (1992). *Applications Programming in Smalltalk-80: How to use Model-View-Controller (MVC)*. by S. Burbeck.

Urquiza Yllescas J. F., M. M. (2010). Las metodologías ágiles y las arquitecturas de software. *coloquio Nacional de investigación*.



EL TECNOLÓGICO NACIONAL DE MÉXICO
Y EL INSTITUTO TECNOLÓGICO DE APIZACO

OTORGAN LA PRESENTE

CONSTANCIA

A

MARÍA DE LOS ÁNGELES HERNÁNDEZ
PALAFOX

POR OBTENER EL PRIMER LUGAR
CON EL PROYECTO
“SISTEMA COTIZADOR DE MADERA INDUSTRIAL”
EN LA CATEGORÍA III. SISTEMAS DE INFORMACIÓN
EN EL MARCO DEL SEMINARIO DE TECNOLOGÍAS DE LA
INFORMACIÓN Y COMUNICACIONES 2017
DE ESTA INSTITUCIÓN

APIZACO, TLAX., MAYO 24 DE 2017

MTRO. FELIPE PASCUAL ROSARIO AGUIRRE
DIRECTOR DEL IT DE APIZACO



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO



Fecha de certificación:
14/03/2015
Régimen de certificación:
Punto de registro:
Procedimiento:
Certificado número: 14/03/2015



Fecha de certificación:
08/03/2015
Régimen de certificación:
Punto de registro:
Procedimiento:
Certificado número: 08/03/2015

Anexo D Estancias



METAL Y MADERA CONSTRUCTIVOS J&M, S. DE R.L. DE C.V.
R.F.C. MMC 0312186J0

Tlaxcala, Tlaxcala a 09 de Enero del 2017

Asunto: Liberación de estancias

MTRO. FELIPE PASCUAL ROSARIO AGUIRRE
DIRECTOR
INSTITUTO TECNOLÓGICO DE APIZACO
PRESENTE

A través de este medio informo a usted que el C. Ing. María de los Ángeles Hernández Palafox alumno de la Maestría en Sistemas Computacionales con No de control M10370131 realizo sus estancias técnicas con el objetivo de "Desarrollar un modelo para el software de cotización e integrar los conocimientos obtenidos en aula con estancia en la empresa" durante el periodo comprendido del 02 Febrero del 2016 a 06 de Enero del 2017.

Sin más por el momento reciba un cordial saludo

ATENTAMENTE

NOE ESPINO TELLEZ
ADMINISTRACION Y FINANZAS



Metal y Madera

LARDIZABAL No. 17 DPTO. 4
COL. CENTRO
TLAXCALA, TLAXCALA, MEXICO C.P.90000
TEL: 52 (246) 46 20214
Email: espino@metalmadera.mx
www.metalmadera.mx



METAL Y MADERA CONSTRUCTIVOS J&M, S. DE R.L. DE C.V.
R.F.C. MMC 0312186J0

Tlaxcala, Tlaxcala a 15 de Marzo del 2016

MTRO. FELIPE PASCUAL ROSARIO AGUIRRE
DIRECTOR
INSTITUTO TECNOLÓGICO DE APIZACO
PRESENTE

A través de este medio informo a usted que el C. Ing. María de los Ángeles Hernández Palafox alumno de la Maestría en Sistemas Computacionales con No de control M10370131 ha sido aceptado para desarrollar el proyecto con el objetivo de "Desarrollar un modelo para el software de cotización e integrar los conocimientos obtenidos en aula con estancia en la empresa"

Durante su estancia técnica en esta empresa estará en el Departamento de ventas, cuyo responsable a cargo del C.P Edgar CuahutecatI Hernández que ocupa el cargo de Director General, durante el periodo comprendido del 02 Febrero del 2016 a 01 de Julio del 2016 con un horario de 09:00 a 17:00 de lunes a miércoles

Sin más por el momento reciba un cordial saludo

ATENTAMENTE

C.P Noé Espino Téllez

Administración y Finanzas

Metal y Madera

Carretera libre Apizaco-Tlaxco No. 66 C. Col. Agrícola de Dolores,
Tetla de La Solidaridad, Tlax. C.P 90430

TEL: 52 (241) 41 23105

Email: metalymadera@metalymadera.com.mx

Anexo E Carta de satisfacción



METAL Y MADERA CONSTRUCTIVOS J&M, S. DE R.L. DE C.V.
R.F.C. MMC 0312186J0

Tlaxcala, Tlaxcala a 30 de mayo del 2017
Asunto: Constancia de satisfacción.

MTRO. FELIPE ROSARIO AGUIRRE
DIRECTOR DEL INST. TEC. DE APIZACO
PRESENTE

AT'N: DR. JOSÉ FEDERICO CASCO VÁSQUEZ
JEFE DE LA DIV. DE EST DE POSGRADO E INV

Sirva la presente para enviarle un cordial saludo y notificarle que posterior a la recepción del proyecto de tesis del **Ing. María de los Ángeles Hernández Palafox**, alumna de la Maestría en sistemas Computacionales con el número de control **M10370131**, del a institución que usted destacadamente dirige, se incluyó en el proyecto que lleva como título:

"Desarrollar un modelo para el software de cotización e integrar los conocimientos obtenidos en aula con estancia en la empresa"

Siendo este desarrollado bajo la dirección del M. en C. José Juan Hernández Mora catedrático de la citada maestría.

En virtud de que se han cubierto satisfactoriamente los objetivos establecidos del citado proyecto.

Tenemos a bien dar constancia de que dicho proyecto de tesis

Cubre y satisface las expectativas planteadas al inicio de este proyecto

Agradeciendo ampliamente sus atenciones quedo de ustedes.

ATENTAMENTE

NOE ESPINO TELLEZ
ADMINISTRACION Y FINANZAS

Metal y Madera

LARDIZABAL No. 17 DPTO 4
COL. CENTRO
TLAXCALA, TLAXCALA, MEXICO C.P. 90000
TEL: 52 (246) 46 20214
Email: espino@metalmadera.mx
www.metalmadera.mx