Imagen que contiene dibujo

Descripción generada automáticamente

REPORTE DE ESTADÍA

“SISTEMA PARA LA REALIZACIÓN DE COTIZACIONES DE SISTEMAS DE ALARMAS Y CCTV”

QUE COMO REQUISITO PARA OBTENER EL TÍTULO DE:

TÉCNICO SUPERIOR UNIVERSITARIO EN

TECNOLOGÍAS DE LA INFORMACIÓN ÁREA INFRAESTRUCTURA DE REDES DIGITALES

PRESENTA:

ALAN MAURICIO AGUIRRE ROBLES

ASESOR EMPRESARIAL:

ING. OMAR CORRAL ALVÍDREZ

ASESOR ACADÉMICO:

ING. LUIS IVÁN GARCÍA GONZÁLEZ

TUTOR ACADÉMICO:

ING. JOSÉ IRVING MARTÍNEZ SÁENZ

Agosto, 2020

**SISTEMA PARA LA REALIZACIÓN DE COTIZACIONES DE SISTEMAS DE ALARMAS Y CCTV**

Reporte Técnico de Estadías realizada por Alan Mauricio Aguirre Robles bajo la dirección del comité revisor indicado, aprobada por el mismo y aceptada como requisito parcial para obtener el título de:

**TÉCNICO SUPERIOR UNIVERSITARIO EN:**

TECNOLOGÍAS DE LA INFORMACIÓN ÁREA INFRAESTRUCTURA DE REDES DIGITALES

COMITÉ REVISOR:



OMAR CORRAL ALVÍDREZ

Asesor Empresarial



LUIS IVÁN GARCÍA GONZÁLEZ JOSÉ IRVING MARTÍNEZ SÁENZ

Asesor Académico Tutor Académico

**CONSTANCIA DE TERMINACIÓN DE ESTADÍA**

**Hgo. del Parral, Chih. a \_\_\_\_ de \_\_\_\_\_\_\_\_\_\_\_\_ de 20\_\_**

**ING. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**JEFE DE LA CARRERA DE \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PRESENTE. -**

En atención al convenio y/o acuerdo celebrado con la Universidad Tecnológica de Parral, hago constar que el(la) C. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, alumno(a) de la carrera de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, con número de matrícula \_\_\_\_\_\_\_\_\_\_\_\_; culminó con su programa de Estadías, en el que realizó el proyecto titulado **“\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”** en nuestras instalaciones, durante el periodo cuatrimestral comprendido del \_\_\_ de \_\_\_\_\_\_\_\_ al \_\_\_ de \_\_\_\_\_\_\_\_\_ del año en curso, cubriendo un total de 525 horas, de acuerdo al programa y plan de estudio de la carrera correspondiente, siendo su tutor por parte de esta empresa, el C. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Sin otro particular por el momento y en espera de que la información anterior cumpla con los requisitos de la Universidad Tecnológica de Parral, me despido de usted no sin antes ponerme a sus órdenes para cualquier aclaración.

**A T E N T A M E N T E**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(NOMBRE, FIRMA Y PUESTO DE PERSONA AUTORIZADA

POR LA EMPRESA PARA ASIGNAR LA LIBERACIÓN

DEL ALUMNO)

c.c.p. Departamento de Vinculación: Constancia de Servicio Social

# RESUMEN

ÍNDICE GENERAL

[RESUMEN i](#_Toc48754526)

[ÍNDICE DE FIGURAS iv](#_Toc48754527)

[I. INTRODUCCIÓN 1](#_Toc48754528)

[II. JUSTIFICACIÓN 2](#_Toc48754529)

[2.1. Delimitación del proyecto 3](#_Toc48754530)

[III. OBJETIVOS DEL PROYECTO 4](#_Toc48754531)

[3.1. Objetivo general 4](#_Toc48754532)

[3.2. Objetivos específicos 4](#_Toc48754533)

[IV. MARCO TEÓRICO 5](#_Toc48754534)

[4.1. Software. 5](#_Toc48754535)

[4.2. Base de datos. 5](#_Toc48754536)

[4.3 Sistemas gestores de bases de datos. 6](#_Toc48754537)

[4.4. Lenguaje SQL. 6](#_Toc48754538)

[4.5. MySQL. 7](#_Toc48754539)

[4.6. Modelo relacional. 7](#_Toc48754540)

[4.7. Java. 8](#_Toc48754541)

[4.8. JasperReports. 9](#_Toc48754542)

[4.9. JCalendar. 9](#_Toc48754543)

[4.10. MySQL Connector/J. 10](#_Toc48754544)

[4.11. Entorno de desarrollo integrado (IDE). 10](#_Toc48754545)

[4.12. NetBeans. 11](#_Toc48754546)

[4.13. Programación Orientada a Objetos. 12](#_Toc48754547)

[4.14. Modelo de desarrollo en cascada. 12](#_Toc48754548)

[4.15. UML. 14](#_Toc48754549)

[4.16. Diagrama de clases. 15](#_Toc48754550)

[4.17. Casos de uso. 15](#_Toc48754551)

[4.18. CCTV. 16](#_Toc48754552)

[V. DESARROLLO DE LA ESTADÍA 18](#_Toc48754553)

[5.1. Desarrollo de actividades 18](#_Toc48754554)

[5.1.1. La base de datos. 18](#_Toc48754555)

[5.1.2. La metodología de programación. 26](#_Toc48754556)

[5.1.3. La navegación en el programa. 38](#_Toc48754557)

[5.1.4. La hoja de cotizaciones. 48](#_Toc48754558)

[5.2. Actividades adicionales 49](#_Toc48754559)

[VI. ANÁLISIS DE LOS RESULTADOS 50](#_Toc48754560)

[6.1. Resultados obtenidos 50](#_Toc48754561)

[6.2. Conclusiones 50](#_Toc48754562)

[BIBLIOGRAFÍA 51](#_Toc48754563)

ÍNDICE DE FIGURAS

[Figura 1: Código de la tabla usuario. 19](#_Toc48754493)

[Figura 2: Código de la tabla categoría. 20](#_Toc48754494)

[Figura 3: Código de la tabla subcategoría. 21](#_Toc48754495)

[Figura 4: Código de la tabla producto. 22](#_Toc48754496)

[Figura 5: Código de la tabla paquete 23](#_Toc48754497)

[Figura 6: Código de la tabla empaquetado. 24](#_Toc48754498)

[Figura 7: Código de la tabla cotización 25](#_Toc48754499)

[Figura 8: Código de la tabla cotiza\_producto. 25](#_Toc48754500)

[Figura 9: Código de la tabla cotiza\_paquete. 26](#_Toc48754501)

[Figura 10: Estructura del paquete VO. 27](#_Toc48754502)

[Figura 11: Diagrama de clases de ProductoVO. 28](#_Toc48754503)

[Figura 12: Estructura del paquete DAO. 29](#_Toc48754504)

[Figura 13: Diagrama de clases de ProductoDAO. 30](#_Toc48754505)

[Figura 14: Estructura del paquete conexión. 31](#_Toc48754506)

[Figura 15: Diagrama de clases de Conexión. 31](#_Toc48754507)

[Figura 16: Estructura del paquete tables. 32](#_Toc48754508)

[Figura 17: Diagrama de clases de TablaProducto. 33](#_Toc48754509)

[Figura 18: Estructura del paquete images. 33](#_Toc48754510)

[Figura 19: Estructura del paquete frames. 34](#_Toc48754511)

[Figura 20: Diagrama de clases de productos, parte 1. 35](#_Toc48754512)

[Figura 21: Diagrama de clases de productos, parte 2. 36](#_Toc48754513)

[Figura 22: Diagrama de clases de productos, parte 3. 37](#_Toc48754514)

[Figura 23: Estructura del paquete cotizaciones. 37](#_Toc48754515)

[Figura 24: Diagrama de clases de generarCotizacion. 38](#_Toc48754516)

[Figura 25: Pantalla de inicio de sesión. 39](#_Toc48754517)

[Figura 26: Menú principal. 40](#_Toc48754518)

[Figura 27: Confirmación de salida. 41](#_Toc48754519)

[Figura 28: Pantalla de categorías. 41](#_Toc48754520)

[Figura 29: Pantalla de productos. 43](#_Toc48754521)

[Figura 30: Pantalla de paquetes. 44](#_Toc48754522)

[Figura 31: Pantalla de usuarios. 45](#_Toc48754523)

[Figura 32: Pantalla para hacer cotizaciones. 46](#_Toc48754524)

[Figura 33: La pantalla para ver cotizaciones. 47](#_Toc48754525)

**Sistema para la realización de cotizaciones de sistemas de alarmas y cctv**

# INTRODUCCIÓN

Sadcom es una empresa que se dedica a la instalación y mantenimiento a equipos de seguridad como alarmas, circuito cerrado de televisión y control de acceso. Lleva ofreciendo sus servicios en la ciudad de Parral por más de 15 años y siendo líder en cuanto a sistemas de seguridad para locales comerciales u hogares.

La empresa antes de realizar una instalación requiere hacer un análisis de los requerimientos del cliente dependiendo de sus necesidades, para así poder proponerle un costo estimado que tendría la compra del equipo necesario, la instalación de este y el mantenimiento periódico, en caso de ser necesario.

Con la realización de este proyecto se buscará hacer más eficiente el proceso de generación de cotizaciones, tanto para entregar al cliente una cotización objetiva y en un tiempo razonable, como para disminuir el tiempo que requiere la tarea de buscar y enlistar el equipo que se utilizará en la instalación, además de automatizar las respectivas cuentas.

El aspecto de brindar una cotización profesional resulta de suma importancia para la empresa, pues esta funciona como una carta de presentación del servicio y ayuda en gran parte a convencer al cliente de que realice la compra o contratación.

# JUSTIFICACIÓN

En el momento en que un cliente solicita más información sobre el servicio y se encuentra un paso más interesado sobre la contratación se vuelve necesario enviarle una cotización que ayude a clarificar el equipo que necesita y los precios de cada unidad, esto con la finalidad de que se sienta en confianza de rechazar la oferta si no le parece o aumentar su seguridad en la compra, al saber a ciencia cierta cuál es el costo que acumularía su compra; también es necesario mencionar que la entrega de una cotización resulta ser un componente de formalidad que proporciona profesionalismo y compromiso por parte de la empresa, repercutiendo directamente sobre su imagen.

Cuando la persona encargada se va a dedicar a hacer una cotización necesita tomar un amplio catálogo con los precios y buscar cada producto en una larga lista de componentes ordenados por categoría, después, modificar una hoja de cálculo de manera manual introduciendo los datos de cada elemento que va a conformar la cotización para, finalmente, entregársela al cliente.

La realización del sistema busca hacer que el proceso mencionado anteriormente se haga de una manera más eficiente, puesto que el componente del tiempo es vital para cualquier empresa, porque el tiempo es dinero; agregándole el hecho de que si la cantidad de clientes aumenta también crece proporcionalmente el número de cotizaciones que se deben realizar, pudiendo a mermar la capacidad de procesamiento de las solicitudes y con ello la calidad de las mismas, viéndose afectada directamente la calidad del servicio.

## Delimitación del proyecto

Se realizará un sistema para computadora que permita construir una cotización y permita la entrega de esta en un formato almacenable y habilitado para su impresión.

Se contará con una base de datos en donde se encuentre almacenada la información más relevante de los productos que se ofrecen, también se podrán gestionar categorías para facilitar la búsqueda de elementos y su posterior adición a la cotización.

Se asegurará la confidencialidad del sistema concediendo el acceso solamente a usuarios que se encuentren registrados en la base de datos y cuenten con las credenciales correctas que se deberán ingresar en la primera pantalla del programa.

También, será posible gestionar paquetes que constan de un conjunto de productos y se podrán elegir como un elemento agregable a la cotización, permitiendo un acceso más directo a los productos que se venden más comúnmente.

El programa se ejecutará sobre una computadora personal con un software servidor de base de datos con cualquier sistema operativo que tenga instalada la máquina virtual de Java y en local, puesto que la empresa así lo requiere.

# OBJETIVOS DEL PROYECTO

## Objetivo general

Desarrollar un sistema para ordenador que permita realizar cotizaciones de una manera eficiente con la finalidad de mejorar los tiempos administrativos, brindar estimaciones objetivas al cliente, elevar la imagen de la empresa y aumentar las probabilidades de cierre de ventas; por medio de la aplicación de tecnologías de bases de datos, lenguajes de programación modernos y patrones de diseño.

## Objetivos específicos

* Proteger la confidencialidad de los datos mediante la utilización de métodos de autenticación por credenciales.
* Contar con una base de datos de los productos que la empresa ofrece.
* Permitir almacenar una imagen relativa a cada producto en la base de datos.
* Dar la posibilidad de modificar datos de los productos almacenados, así como eliminarlos.
* Permitir la creación y gestión de paquetes que consten de una agrupación de varios productos.
* Contar con una interfaz simple y limpia que permita al usuario su uso sin complicaciones.
* Presentar una hoja de cotización con un detalle de todos los productos, el subtotal y totales.
* Ofrecer un diseño sobrio y profesional en la hoja de cotización.
* Permitir la impresión y almacenamiento de las cotizaciones hechas.
* Facilitar la búsqueda y selección de productos para agregar a la cotización.
* Permitir la gestión de categorías de los productos para facilitar las búsquedas.
* Posibilidad de agregar, modificar y quitar usuarios con acceso al sistema.

# MARCO TEÓRICO

## 4.1. Software.

Como lo mencionan Moreno y Ramos (2014):

Por “software” entendemos al equipamiento o soporte lógico de un sistema informático. Lo constituye el conjunto de componentes lógicos y, por tanto, no tangibles y no físicos, necesarios para llevar a cabo una tarea específica en nuestro sistema. Es un componente imprescindible en todo sistema informático, que comunicará y dará órdenes al hardware para que se lleven a cabo todas las tareas que se el usuario del sistema le encomiende.

(p. 14)

## 4.2. Base de datos.

Hueso (2014) menciona sobre las bases de datos:

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. La base de datos es un conjunto de datos organizados en estructuras que se definen una sola vez y que se utiliza al mismo tiempo por muchos equipos y usuarios. En lugar de almacenarse en ficheros desconectados y de manera redundante, los datos en una base de datos están centralizados y organizados de forma que se minimice la redundancia y se facilite su gestión. La base de datos no pertenece a un equipo, se comparte por toda la organización. Además, la base de datos no solo contiene los datos de la organización, también almacena una descripción de dichos datos.

(p. 22)

## 4.3 Sistemas gestores de bases de datos.

Un sistema gestor de bases de datos es una herramienta informática que permite el acceso, creación y modificación de una base de datos, como dijo Hueso (2014):

El sistema de gestión de la base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma. Es una herramienta que sirve de interfaz entre el usuario y las bases de datos.

(p. 27)

## 4.4. Lenguaje SQL.

Es un lenguaje que se utiliza para la creación y manipulación de bases de datos, Reinosa, Maldonado, Muñoz, Damiano y Abrutsky (2012), lo definen como:

El lenguaje SQL se considera, por un lado, un lenguaje diseñado específicamente para la comunicación entre usuarios y, por otro con la base de datos para realizar todas las tareas requeridas para resolver los requerimientos como obtener información almacenada, realizar cálculos, modificar lo existente y agregar nuevas filas. La característica más destacada del lenguaje SQL es que es no procedimental, es decir, no se indica en sus sentencias cómo realizar la tarea, sino que se limita a describir el resultado buscado y queda todo el trabajo de resolver lo solicitado al servidor de bases de datos que, de acuerdo con su optimizador y con los metadatos (se los denomina así porque son datos acerca de los datos) del diccionario, cumplirá con la sentencia SQL provista.

(p. 105)

## 4.5. MySQL.

MySQL es un sistema de gestión de bases de datos muy conocido y utilizado alrededor del mundo, como lo menciona Hueso (2014):

Es un SGBD Open Source, lo que significa que es posible para cualquiera usar y modificar el software. Cualquiera puede bajar el software MySQL desde Internet y usarlo libremente. Es un sistema cliente/servidor que consiste en un servidor SQL multi-threaded (multihilo), que trabaja con diferentes programas y bibliotecas cliente, herramientas administrativas y un amplio abanico de interfaces de programación para aplicaciones (API).

(p. 20)

Así mismo, Reinosa et. al (2012) lo definen como:

MySQL surgió en una empresa sueca MySQL AB, en la década del noventa. En la actualidad, la empresa es subsidiada por Sun Microsystems de Oracle Corporation. MySQL es, sin duda, uno de los Sistemas de Gestión de Bases de Datos Relacionales (SGBDR) open source más difundido y utilizado. Se puede obtener bajo la licencia GNU GPL (Software Libre) o mediante la distribución comercial, que se diferencia en el soporte, las herramientas de monitoreo y en la posibilidad de incorporarlo en productos privativos.

(p. 336)

## 4.6. Modelo relacional.

El modelo relacional se utiliza como guía para el diseño lógico de una base de datos, como menciona Hueso (2014):

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales las filas (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.También sirven para representar el nivel externo (vistas) de una base de datos.

(p. 25)

## 4.7. Java.

Es un lenguaje de programación de los más utilizados en la actualidad, Sierra (2018) menciona sobre el mismo:

Desde su aparición a mediados de los 90, Java no ha hecho más que crecer y extenderse. Ya en su primera versión, incorporó una característica que comentaremos seguidamente y que hizo que tuviera gran aceptación por parte de la mayoría de las empresas de software del momento, se trata de la posibilidad de compilar una vez y ejecutar en cualquier parte, algo que sin lugar a duda fue una auténtica novedad en aquella época.

(p. 1)

Otra definición está dada por Ceballos (2006):

Java es un lenguaje de programación de alto nivel con el que se pueden escribir tanto programas convencionales como para Internet. Java incluye dos elementos: un compilador y un intérprete. El compilador (programa traductor) produce un código de bytes que se almacena en un fichero para ser ejecutado por el intérprete Java denominado máquina virtual de Java. Es fácil entender entonces que una de las ventajas significativas de Java sobre otros lenguajes de programación es que es independiente de la plataforma. Esto quiere decir que el código producido por el compilador Java puede transportarse a cualquier plataforma (Intel, Sparc, Motorola, etc.) que tenga instalada una máquina virtual Java y ejecutarse. Pensando en Internet esta característica es crucial ya que esta red conecta ordenadores muy distintos.

(p. 2)

## 4.8. JasperReports.

Es una librería que se implementa sobre Java para permitir la creación y modificación de documentos de texto, como se menciona en la página oficial de JasperSoft Community (2020):

La Biblioteca JasperReports es el motor de informes de código abierto más popular del mundo. Está completamente escrito en Java y puede utilizar datos provenientes de cualquier tipo de fuente de datos y producir documentos perfectos que se pueden ver, imprimir o exportar en una variedad de formatos de documentos, incluidos HTML, PDF, Excel, OpenOffice y Word.

(párr. 1)

## 4.9. JCalendar.

Es un complemento que se integra con Java y permite, como menciona Kai Todter (2020), elegir gráficamente una fecha en Java, está compuesto por otros elementos como JDayChooser, JMonthChooser y JDateChooser.

## 4.10. MySQL Connector/J.

Es un controlador que permite realizar la integración de MySQL con Java, Cuenca (2002), menciona sobre el mismo:

MySQL Connector/J es un driver nativo de Java que convierte las llamadas generadas por JDBC en el protocolo de red que utiliza la base de datos de Mysql. Permite al desarrollador trabajar con el lenguaje de programación Java y de esta forma construir programas que interactúan con Mysql.

(párr. 2)

## 4.11. Entorno de desarrollo integrado (IDE).

Es un conjunto de programas que sirven como herramientas para el programador, puesto que en un solo sistema se permite la edición de código, compilación y depuración, como menciona Maldonado (2007):

Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk u Objective-C. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación.

Los componentes que debe tener los IDEs son: Un editor de texto, compilador, intérprete, herramientas de automatización, un depurador, posibilidad de ofrecer un sistema de control de versiones, factibilidad para ayudar en la construcción de interfaces gráficas de usuarios.

(párrs. 3-5)

## 4.12. NetBeans.

Es uno de los entornos de desarrollo más populares para el desarrollo de aplicaciones con Java, Gómez y Moreno (2019), lo definen de la siguiente manera:

NetBeans es un entorno de desarrollo de uso libre que fue creado para utilizar lenguaje Java. Existe una variedad de módulos diseñados especialmente para hacerlo extensible. Esta cualidad hace que NetBeans sea lo suficientemente poderoso y escriba aplicaciones Java para escritorio, así como para dispositivos móviles. Cuenta con una interfaz gráfica para aplicaciones de escritorio rica en componentes y librerías de reutilización.

El editor de código, multilenguaje, por cierto, permite el coloreado habitual de los lenguajes de programación modernos, accesibilidad a las clases con sólo un clic del mouse, control de versiones, comprobaciones sintácticas y semánticas, plantillas reutilizables, entre otras características muy útiles para el desarrollo de software.

(p. 6)

## 4.13. Programación Orientada a Objetos.

Como lo menciona Pantaleo y Rinaudo (2015):

La Programación Orientada a Objetos (POO) surgió como una alternativa a la Programación Procedural y se basa en la descomposición de los problemas a partir de conceptos extraídos del dominio. Estos conceptos a la larga terminarán siendo objetos del modelo asociado al problema. La descomposición de los problemas en forma ortogonal a como lo hace la programación procedural es la diferencia esencial, en lugar de enfocarse en funcionalidades cada vez más simples, lo hace mediante los conceptos mencionados y la interacción entre ellos. Los otros dos mecanismos conductores de este paradigma son la abstracción en la determinación del alcance de los objetos y el establecimiento de jerarquías al generalizar sus comportamientos.

(p. 42)

## 4.14. Modelo de desarrollo en cascada.

Pantaleo y Rianudo (2015) mencionan sobre el desarrollo en cascada:

Es un modelo de desarrollo lineal secuencial. El proyecto de software es dividido en fases que deben procederse en forma secuencial. El proceso incluye una serie de etapas en el siguiente orden:

1. Definición del software: corresponde a la visión del producto, a sus aspectos desde el punto de vista comercial.
2. Análisis de requerimientos: implica el entendimiento del dominio del producto a ser desarrollado; esto es, funciones, comportamiento y relación con sistemas externos. El principal objetivo de esta fase es achicar los riesgos de negocio.
3. Diseño de la arquitectura: es la forma en que la solución se implementará.
4. Codificación: corresponde a la implementación de la solución de acuerdo a cómo se ha estipulado durante el diseño de la arquitectura.
5. Pruebas: deben asegurar que el producto satisfaga los requerimientos, es decir que cumplan con el comportamiento esperado. Los principales objetivos de esta fase son asegurar la calidad del producto y reducir los riesgos de falla de la aplicación en el entorno final de implantación.

Cuando finaliza cada una de las fases mencionadas, se realiza una revisión con el fin de determinar si está en condiciones de avanzar a la siguiente. Las fases no se solapan unas con otras.

Para poder aplicar esta metodología es conveniente que los requerimientos del cliente sean conocidos de antemano. Asimismo, es preferente un número reducido de requerimientos. Tengamos en cuenta que las pruebas de la aplicación serán llevadas a cabo cuando todo el desarrollo se encuentre finalizado. Encontrar un error en esta etapa provoca su corrección y la ejecución de las pruebas de regresión pertinentes. Por último, cuando el equipo de trabajo no tenga experiencia, la aplicación de esta metodología puede ser de gran utilidad dado que ayuda a minimizar el esfuerzo en vano.

(pp. 55-56)

## 4.15. UML.

El lenguaje UML, por su nombre en inglés *Unified Modeling Language,* es conocido como un lenguaje de modelado unificado, según Pantaleo y Rianudo (2015):

UML es un lenguaje que está compuesto por un conjunto de diagramas agrupados por un metamodelo que ayuda a especificar y diseñar el software de sistemas; particularmente software orientado a objetos. Por muchos años fue el estándar de hecho de la comunidad informática, aunque después de 6 años de su primera versión fue adoptado como tal.

De la misma manera que un constructor de casas o edificios antes de apilar ladrillos elabora un plano, un desarrollador de software antes de escribir código hace un plano que le permita evaluar alternativas de su diseño. Igual que una modista utilizando hojas de papel construye un molde para sus prendas en forma previa a cortar la tela y coserla, un desarrollador de software necesita elaborar un modelo que describa los diferentes aspectos del problema a resolver y de la solución propuesta. UML es un lenguaje de especificación y diseño que permite a los desarrolladores construir modelos para estudiar, analizar y decidir acerca de la estructura y el comportamiento que propondrá como solución y en definitiva cómo estará compuesto y cómo se comportará el código que implementará dicha solución.

(p. 110)

## 4.16. Diagrama de clases.

Pantaleo y Rianudo (2015) mencionan sobre los diagramas de clases:

El diagrama de clases muestra los componentes esenciales de un modelo orientado a objetos. En ellos se muestran las clases con algunos de sus atributos y métodos más importantes, en modo Dibujo (*Sketch*). Seguramente contaremos en el diseño de un sistema con varios gráficos en los cuales puede aparecer la misma clase. Cada uno de los diagramas debe haber sido construido con objetivos diferentes, y la clase en cuestión estará presente en cada uno de ellos jugando los diferentes roles que se le asignaron. La estrategia recomendada en la documentación de las clases es construir un diagrama por cada contexto del modelo y así serán manejables y legibles.

(p. 123)

## 4.17. Casos de uso.

Son diagramas UML que ejemplifican gráficamente la realización de cierta actividad por medio de interacciones entre actores y actividades, Pantaleo y Rianudo (2015), mencionan al respecto:

Cuando en etapas tempranas de un proyecto se busca definir su alcance, a partir de un listado de requerimientos funcionales se construye un diagrama de casos de uso. Estos modelan funcionalidades y sus relaciones; y su especificación textual especifica cómo serán implementados los requerimientos asociados. Asociamos cada funcionalidad a un actor, quien es el que a partir del uso del sistema genera el evento que ejecuta dicha funcionalidad. Estos diagramas son acompañados con la especificación textual de los pasos del caso de uso y prototipos de interfaces de usuario.

(p. 136)

## 4.18. CCTV.

Es un sistema que permite la vigilancia de espacios por medio de la utilización de cámaras y almacenamiento de video, como menciona el grupo Acacio (2016):

CCTV es el acrónimo de Circuito Cerrado de Televisión. Se trata de una instalación de componentes directamente conectados, lo que permite crear un circuito de imágenes que no puede ser visto por otra persona fuera de él. La diferencia con otras señales de televisión ordinarias es que éstas pueden ser vistas por cualquiera con una antena o dispositivo para recibirlas, mientras que en el CCTV la señal está restringida a la persona o personas autorizadas.

Los Circuitos Cerrados de Televisión se pueden personalizar para adaptarse a la naturaleza y necesidades de seguridad de cada cliente. Actualmente estos sistemas proporcionan imágenes de gran calidad tanto de día como de noche, además de ser muy fáciles de usar y de entender su funcionamiento.

Las cámaras CCTV están disponibles en varios sistemas: digitales, analógicos, con cable, sin cable, etc. No obstante, todos ellos comparten los mismos componentes: cámaras CCTV, una lente CCTV, monitor CCTV, cables que llevan la señal (en caso de que el sistema sea cableado), etc.

Las imágenes captadas por las cámaras CCTV son enviadas a un monitor CCTV y grabadas. Además, los recientes avances en tecnología permiten grabar el movimiento y notificar las incidencias de forma automática. De esta forma las cámaras solo graban cuando detectan movimiento, lo que permite ahorrar energía y espacio de almacenamiento. La notificación de eventos consiste en el envío de mensajes de texto o emails cuando se detecta alguna actividad.

(párrs. 1-2)

# DESARROLLO DE LA ESTADÍA

## Desarrollo de actividades

### La base de datos.

Una de las fases más importantes en el desarrollo de un sistema que satisfaga las necesidades del cliente, y de las más difíciles, es la abstracción de los datos y el diseño y estructuración de la base de datos. Esta debe estar pensada de modo de que se consuma el menor tamaño posible, haciéndose realidad llevando a cabo una buena elección de los tipos de datos, también, se debe procurar no tener datos redundantes, esto quiere decir que los datos que se encuentran en una tabla pertenezcan únicamente al objeto que representa en la realidad, a excepción de las llaves foráneas que hacen posible la vinculación entre entidades.

A continuación, se describirá la estructura de cada tabla de la base de datos, así como sus relaciones con las demás, también se dará una breve explicación de por qué la estructura elegida es la que se ajusta a las necesidades del cliente y permita que haya una conexión óptima con el sistema que se desarrollará posteriormente.

#### 5.1.1.1 La tabla usuario.

Antes de pensar en cómo funcionará internamente el sistema se necesita tener en cuenta la cuestión de seguridad respecto al ingreso al mismo, será necesario una tabla donde se tengan registrados los usuarios que podrán acceder al sistema.

En el caso muy particular de la empresa sadcom, el programa será utilizado únicamente por el gerente, que es quien hace las cotizaciones y conoce los precios de los productos. Por esto mismo, se ha pensado en crear una capa de acceso básica, con texto plano, no se necesita mucho, simplemente por si alguna persona llegara a ingresar a la computadora del gerente, para proteger la información que maneja el sistema.

Cabe mencionar que el software será instalado únicamente en la computadora del usuario principal, el gerente, el cual hará la función de servidor, para almacenar la base de datos y cliente, para recibir los datos almacenados.

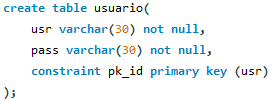


Figura 1: Código de la tabla usuario.

En la figura anterior se puede observar el código utilizado para la creación de la tabla usuarios la cual tendrá dos campos, “usr” para almacenar al usuario y “pass” para almacenar la contraseña. Ambos campos son texto plano, siendo requerido ingresar ambos para poder guardar un registro. Con la construcción de “usr” como llave primaria se asegura que no existan dos usuarios con el mismo nombre.

#### 5.1.1.2. La tabla categoría.

Como es de esperarse, en la base de datos se debe permitir guardar productos, para en el momento en que sea necesario hacer una cotización solamente elegirlos de entre una lista; pero antes de poder insertar productos se necesita pensar en su clasificación, esto con mentalidad de facilitar la búsqueda de productos en el futuro. Por eso ha nacido la tabla categoría, para poder crear un producto es necesario que este haga referencia a una categoría y para tener la referencia es necesario tener primero la categoría.

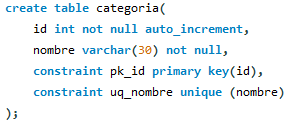


Figura 2: Código de la tabla categoría.

Se puede observar que se cuenta como clave primaria un campo llamado id, el cual servirá para uso interno del programa y se incrementará automáticamente conforme se vayan insertando registros, también cada categoría tiene un nombre, que será el identificador que un usuario utilizará. Como es lógico que no existan dos categorías con el mismo nombre se ha construido una clave única para ese campo, de no ser así se podrían insertar varias categorías con el mismo nombre, puesto que el control lo lleva la clave primaria y esa nunca se repite.

#### 5.1.1.3. La tabla subcategoría.

En el sistema cada producto pertenece a una categoría, pero dentro de cada categoría también hay categorías, por ejemplo, no es lo mismo hablar de cámaras como categoría, que de los dvr que es una subcategoría de la categoría cámaras.

Por eso y para lograr una clasificación más precisa de todos los productos, se ha decidido crear la tabla subcategoría, la cual contiene un id auto incrementable, el id de la categoría a la que pertenece, puesto que una subcategoría pertenece únicamente a una categoría y, finalmente, el nombre.

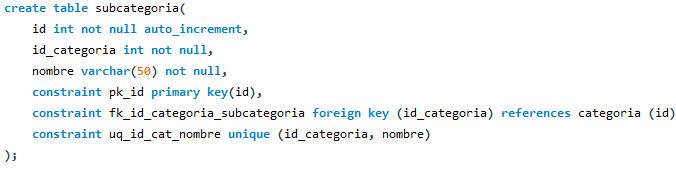


Figura 3: Código de la tabla subcategoría.

Se puede visualizar en el código que existe integridad referencial en la creación de id como llave primaria, el campo id\_categoria como único, permitiendo que exista solamente una subcategoría que se llame de determinada manera dentro de una categoría y la referencia a la categoría de la cual la tabla subcategoría es hija.

#### 5.1.1.4. La tabla producto.

En esta tabla se almacenará toda la información que corresponde a los productos vendidos por la empresa, existe un id para uso interno del sistema, también una descripción, se puede agregar el modelo del producto o en qué consiste, dependiendo de cómo se desea que aparezca en la hoja de cotización del cliente o como resulte más fácil la identificación para el usuario, también se almacena el precio con punto decimal, generalmente se ingresa el precio sin iva; un campo importante es la imagen, se ha decidido guardar este dato en la base de datos, aunque a veces no es lo más conveniente, porque resulta más sencillo manejarlo como un campo dentro del programa, de otra manera, se tendrían que agregar funciones extra para la importación de imágenes desde rutas. Finalmente se encuentra el campo de id\_subcategoría, un producto pertenece a una y solo una subcategoría, también a una categoría, pero por el hecho de que una subcategoría pertenece a una categoría se vuelve innecesario guardar ese dato en la tabla productos, puesto que esa información se sobreentiende y es accesible mediante la utilización de un join o una vista personalizada.

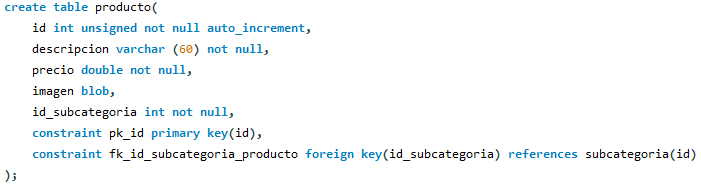


Figura 4: Código de la tabla producto.

#### 5.1.1.5. La tabla paquete.

Un paquete estará compuesto de un conjunto de productos, pero como no es relativo al paquete almacenar exactamente los productos que lo componen y debido a que puede ser más de uno, una cantidad indefinida, se ha decidido crear otra tabla que guarde la relación entre paquete y producto.

En esta tabla se puede encontrar el identificador único, una descripción alusiva al tipo de paquete del que se trate, el campo suma, que se refiere a un subtotal, el precio antes de aplicar un descuento, un descuento en porcentaje, por eso se eligió un campo entero sin signo y, finalmente, el campo precio, que guarda el resultado de aplicarle el descuento al subtotal.

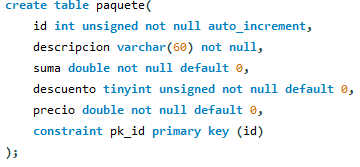


Figura 5: Código de la tabla paquete

#### 5.1.1.6. La tabla empaquetado.

Cada paquete guarda relación con los productos de los cuales está compuesto, esta tabla fusiona un producto en su paquete, es la que vincula el paquete a un producto, un producto puede pertenecer a varios paquetes y en un paquete puede haber varios productos.

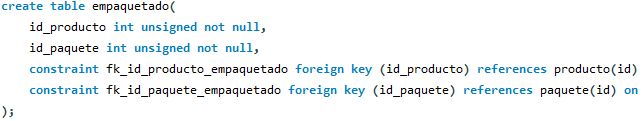


Figura 6: Código de la tabla empaquetado.

#### 5.1.1.7. La tabla cotización.

En la siguiente tabla se describirá la estructura de una cotización, entendida como un objeto independiente en la base de datos, con atributos solamente alusivos a la cotización, se cuenta con el campo identificador, una descripción, que dará una breve explicación sobre la finalidad de la cotización, la fecha en que fue realizada, el subtotal, que es la suma de los precios de los productos, el total, que es el precio al aplicar el iva; también se encuentra un campo llamado lugar, que se refiere a quién va dirigida la cotización, finalmente, un campo booleano llamado comprado el cual no se utilizará en el desarrollo de este sistema, sin embargo queda como base a un posible sistema de ventas.

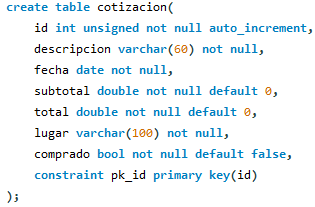


Figura 7: Código de la tabla cotización

#### 5.1.1.8. La tabla cotiza\_producto.

Como en una cotización se pueden agregar productos, pero al mismo tiempo paquetes, es necesario la creación de tablas intermedias que reúnan esa información, esto debido a la cardinalidad de relación n a n.

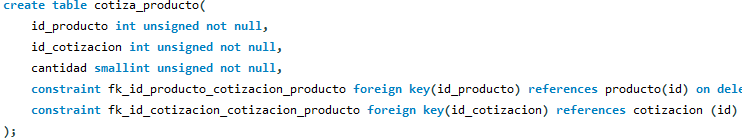


Figura 8: Código de la tabla cotiza\_producto.

Se puede decir que en esta tabla se crea un detalle de la cotización en cuanto a productos.

#### 5.1.1.9. La tabla cotiza\_paquete.

Al igual que en la tabla de cotización de productos, en esta tabla se vincula un paquete a una cotización, sirviendo a modo de detalle, al mismo tiempo, se guarda la cantidad de paquetes que se comprarán en la cotización, ya que se puede comprar más de uno del mismo tipo.

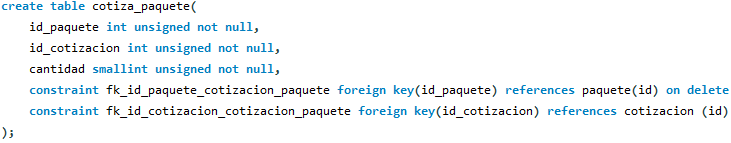


Figura 9: Código de la tabla cotiza\_paquete.

### La metodología de programación.

Se tiene pensado utilizar una versión modificada del modelo vista controlador para separar la programación por partes: por una parte el diseño de las pantallas, por otra la que tiene que ver con la conexión con la base de datos y el comportamiento separado de lo demás.

#### El paquete VO.

El objetivo de este paquete es crear una clase por tabla de la base de datos, la función de cada clase será fungir como recipiente de los datos, como estructura de datos para el almacenamiento de un objeto de la base de datos.

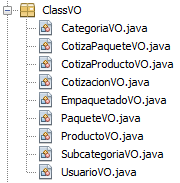


Figura 10: Estructura del paquete VO.

Cada clase tendrá como atributos las columnas de la tabla en cuestión de la base de datos, además se agregarán métodos como un constructor vacío, un constructor que permita inicializar todas las variables, el método toString, para utilizar el objeto directamente como texto. Una parte clave de estas clases es la implementación de métodos getters y setters, que servirán para obtener los valores almacenados en el objeto y establecerlos, respectivamente.

A continuación, se muestra un ejemplo de diagrama de clases para clarificar la estructura expuesta anteriormente:

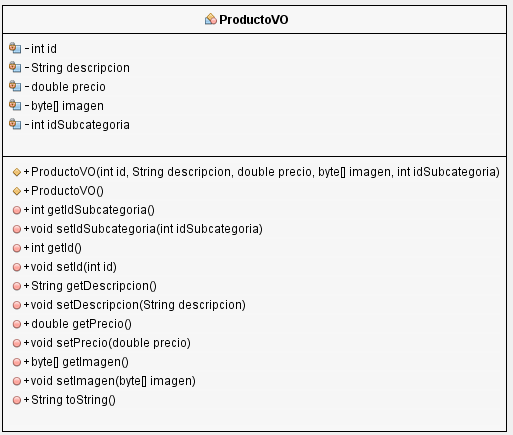


Figura 11: Diagrama de clases de ProductoVO.

#### El paquete DAO.

En el paquete DAO se almacenarán todas las clases que se encarguen de interactuar con la base de datos, dentro de cada clase se encuentran los métodos para realizar modificaciones a la base de datos, eliminar registros, insertar datos. Se ha creado una clase por cada tabla de la base de datos.

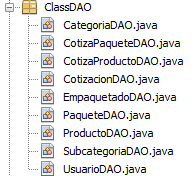


Figura 12: Estructura del paquete DAO.

En cada clase se incluyen métodos como insertar, modificar, eliminar, buscar respecto a ciertos parámetros, todo lo anterior haciendo uso de las clases VO descritas anteriormente.

A continuación, se muestra un diagrama de clases que hace más evidente la función de una clase del paquete DAO

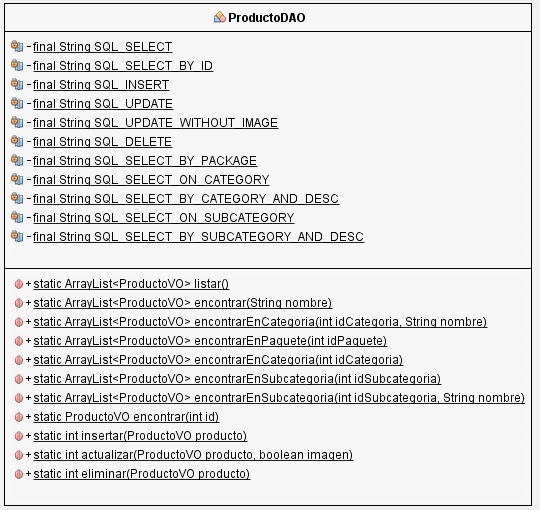


Figura 13: Diagrama de clases de ProductoDAO.

#### El paquete y la clase conexión.

El paquete conexión contiene únicamente la clase conexión, la cual sirve solamente para realizar la comunicación con la base de datos y retornar el objeto conexión que será utilizado por las clases DAO.



Figura 14: Estructura del paquete conexión.

También se han agregado métodos referentes al manejo de fechas dentro de la clase conexión, puesto que sirven para la conversión de objetos de fecha de java a objetos de fecha de sql. A continuación, se muestra el diagrama de clases de la clase conexión:

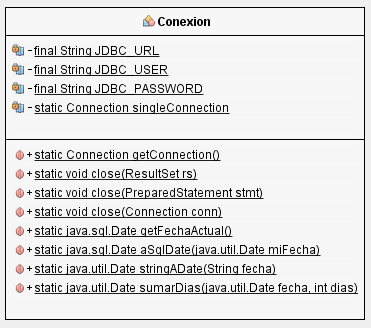


Figura 15: Diagrama de clases de Conexión.

#### El paquete tables.

Debido a que las tablas de productos y paquetes se usarán en distintas pantallas del programa, se ha decidido crear una clase por conjunto de tablas a utilizar de un tipo de objeto de la base de datos, para poder llenar las tablas desde cualquier parte del programa.

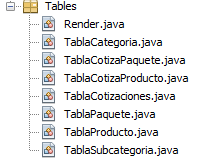


Figura 16: Estructura del paquete tables.

En estas clases se encuentran métodos que retornan tablas cargadas desde la base de datos dependiendo ciertos parámetros que sean ingresados. Enseguida se muestra un diagrama de clases de una clase de tabla.

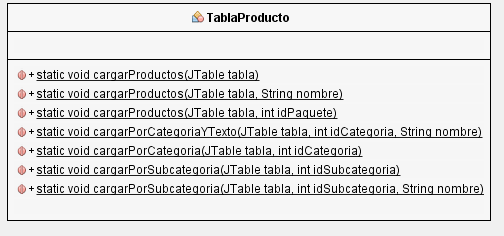


Figura 17: Diagrama de clases de TablaProducto.

#### El paquete images.

Dentro de este paquete se encontrarán todos los archivos de imágenes e íconos que pasarán a formar parte del diseño de las pantallas del sistema.

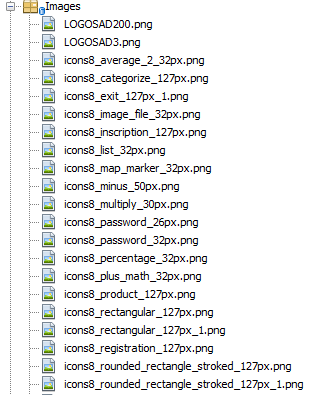


Figura 18: Estructura del paquete images.

#### El paquete frames.

Dentro de este paquete se albergarán las clases correspondientes a las pantallas del sistema, a la interfaz de usuario, además, dentro de ellas, se encontrará toda el comportamiento e interacción con las demás clases.

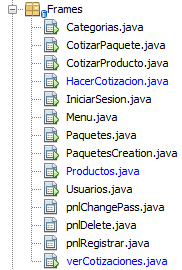


Figura 19: Estructura del paquete frames.

Mostrar un diagrama de clases completo de cualquiera de estas clases requeriría mucho espacio, puesto que se tienen abundantes atributos y muy diversos métodos, que realizan desde verificaciones de campos hasta funciones a grandes rasgos, como guardar un dato:

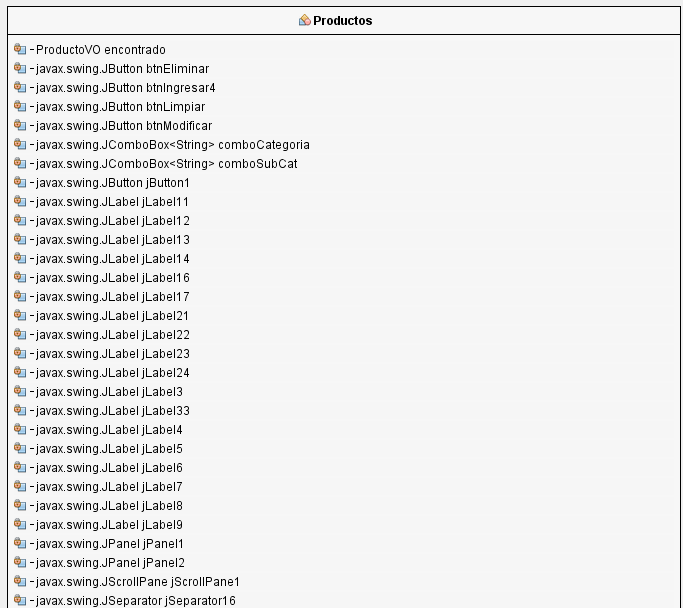


Figura 20: Diagrama de clases de productos, parte 1.



Figura 21: Diagrama de clases de productos, parte 2.



Figura 22: Diagrama de clases de productos, parte 3.

#### El paquete cotizaciones.

Dentro de este paquete se puede encontrar la clase que dirige la creación de los documentos arrojados para una cotización, además, los archivos del diseño de la cotización y los archivos tipo Jasper que son los generadores de los documentos de cotización.

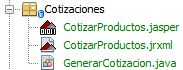


Figura 23: Estructura del paquete cotizaciones.

La clase GenerarCotización es la que se debe mandar a llamar si se desea crear un documento producto de una cotización, contiene los métodos para acceder a los archivos Jasper y jrxml, a continuación, se muestra su diagrama de clases:

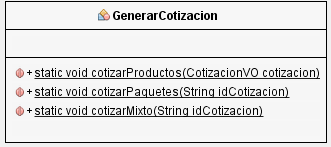


Figura 24: Diagrama de clases de generarCotizacion.

### La navegación en el programa.

A continuación, se explicará de manera breve cómo es la navegación entre pantallas del programa, y qué función se realiza en general dentro de cada una de ellas, también, se mencionará el por qué de ciertos elementos que han sido situados en las pantallas.

#### La pantalla de inicio de sesión.

La primera vista que se tiene del programa representa un formulario simple de inicio de sesión y sirve para proteger el acceso a personas no autorizadas.



Figura 25: Pantalla de inicio de sesión.

Se puede observar a la izquierda el logo de la empresa, en otro color el formulario de inicio de sesión, seguido de los campos de usuario y contraseña, acompañados por un ícono que hace alusión al contenido del campo, finalmente, un botón para verificar los datos y en caso de que sean correctos, mostrar el menú principal.

#### El menú principal.

Una vez que se ha ingresado al sistema se muestra el menú principal, el cual está conformado por una barra superior donde se encuentran los botones de minimizar y cerrar y los distintos apartados del cual el sistema está compuesto.



Figura 26: Menú principal.

Se puede ingresar a la ventana para hacer una cotización, para consultar las cotizaciones hechas anteriormente, para crear y gestionar categorías, para almacenar los productos, para gestionar los paquetes, para modificar e ingresar usuarios al sistema y para salir. Una vez que se hace clic en el apartado que se desea se abre la pantalla correspondiente, a excepción del apartado salir, que antes pregunta si realmente desea realizar esa acción.

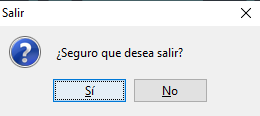


Figura 27: Confirmación de salida.

#### La pantalla de categorías.

Antes de ingresar productos es necesario crear categorías, si no faltará un campo para la creación del producto.

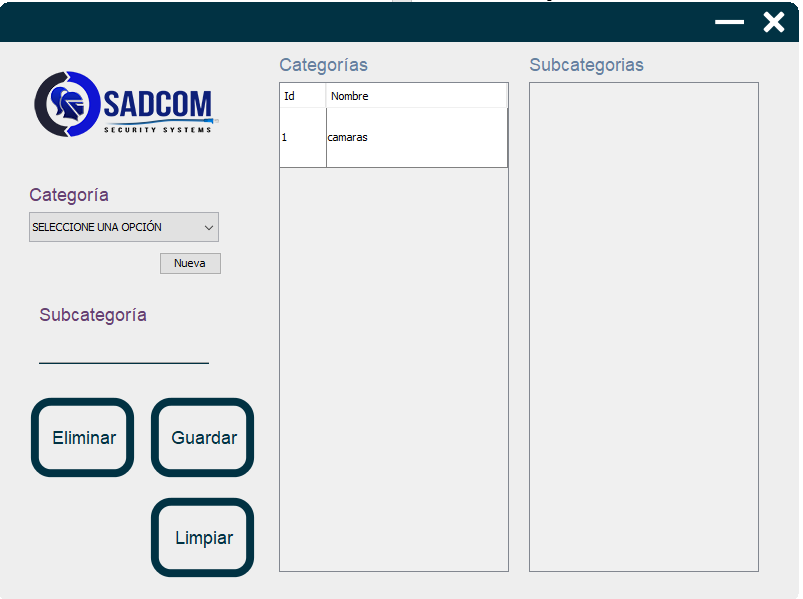


Figura 28: Pantalla de categorías.

Se puede observar, al igual que en todas las pantallas, la barra superior que contiene los botones de salir y minimizar. Empezando de arriba hacia abajo y de izquierda a derecha, se puede ver el logo de la empresa, el cual, al hacerle clic, retorna al menú principal; luego, se puede observar un comboBox que contiene las categorías del sistema, esto es para seleccionar en que subcategoría se guardará, en caso de querer crear una nueva categoría se hace clic sobre nueva. Se pueden ver los botones de eliminar y guardar, que cumplirán su función únicamente cuando los campos requeridos hayan sido llenados de manera satisfactoria. Más abajo se puede encontrar el botón limpiar, el cual regresará la pantalla a su estado inicial

También se pueden observar dos tablas: una llamada categorías y subcategorías; la tabla categorías mostrará las categorías que se encuentran dadas de alta en la base de datos, mientras que la tabla subcategorías solamente se cargará cuando se haga clic sobre una categoría de la tabla, dando a entender que los datos mostrados pertenecen a esa categoría seleccionada.

#### La pantalla de productos.

El diseño es muy similar a la pantalla anterior, se conserva la barra superior y el logo con su funcionalidad, pero se ha agregado una barra de búsqueda y un botón para modificar. La barra de búsqueda busca coincidencias por descripción y muestra todos los productos que cumplan con las palabras escritas, de no haber un texto para buscar, carga en la tabla todos los productos de la base de datos.

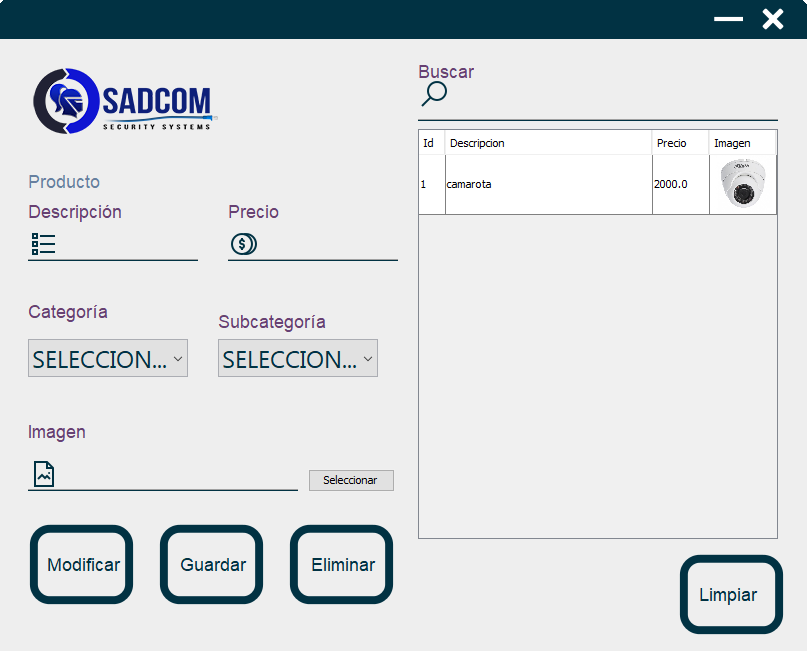


Figura 29: Pantalla de productos.

Una vez que se hace clic sobre la selección de categoría se cargan todas las categorías de la base de datos y una vez que se selecciona una categoría se cargan todas las subcategorías que le corresponden. En el campo de imagen se completa automáticamente la ruta de la imagen que se desea guardar, para guardar la imagen en la base de datos se debe buscar su localización con el botón seleccionar. Para modificar y eliminar es necesario primero encontrar el producto con la barra de búsqueda y luego hacer clic al elemento en la tabla, para que así se llenen automáticamente los campos de la izquierda y se puedan hacer las acciones antes mencionadas.

#### La pantalla de paquetes.

En la siguiente pantalla se mantienen elementos básicos de las demás, pero se quiere hacer hincapié en la barra de búsqueda y en las tablas de abajo, ya que se presenta una idea similar con categorías y subcategorías, nada más que ahora en la primera tabla se cargan los paquetes y en la otra los productos que lo constituyen. Los campos de subtotal y precio se cargan automáticamente, al no introducir ningún dato en el campo de descuento el programa asume que el descuento es cero, por lo tanto, no se hace indispensable aplicar descuento siempre; sin embargo, no se permite guardar un paquete sin antes llenar el campo descripción.

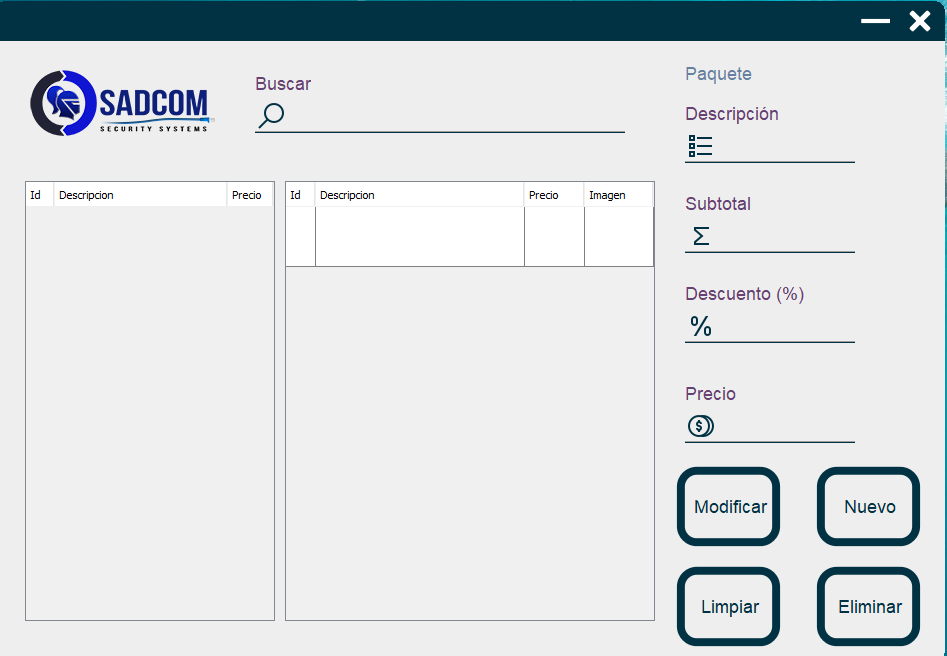


Figura 30: Pantalla de paquetes.

#### La pantalla usuarios.

En esta pantalla se muestra un menú lateral especificando los tipos de acciones que se pueden hacer respecto a los usuarios. Una vez que se ha elegido una opción de entre la lista, se carga en el lado derecho la porción de programa que va a permitir llevar a cabo tal acción.



Figura 31: Pantalla de usuarios.

#### La pantalla para hacer cotizaciones.

Es la pantalla más compleja del sistema, pues manda abrir pantallas para agregar productos y paquetes, primero es necesario seleccionar una fecha a la cual se quiera decir que fue hecha la cotización, por defecto se carga con la fecha del sistema, enseguida, es necesario llenar datos como la descripción de la cotización la cual habla brevemente sobre el objetivo de la cotización: un sistema de alarma o de cctv, por ejemplo; el iva al dejarse vacío se calculará como el 16 por ciento, o en caso de aumentar, se puede ingresar el valor manualmente. Los campos subtotal y total se llenan automáticamente, uno al sumar los paquetes y productos y otro al aplicar el iva sobre la sumatoria.

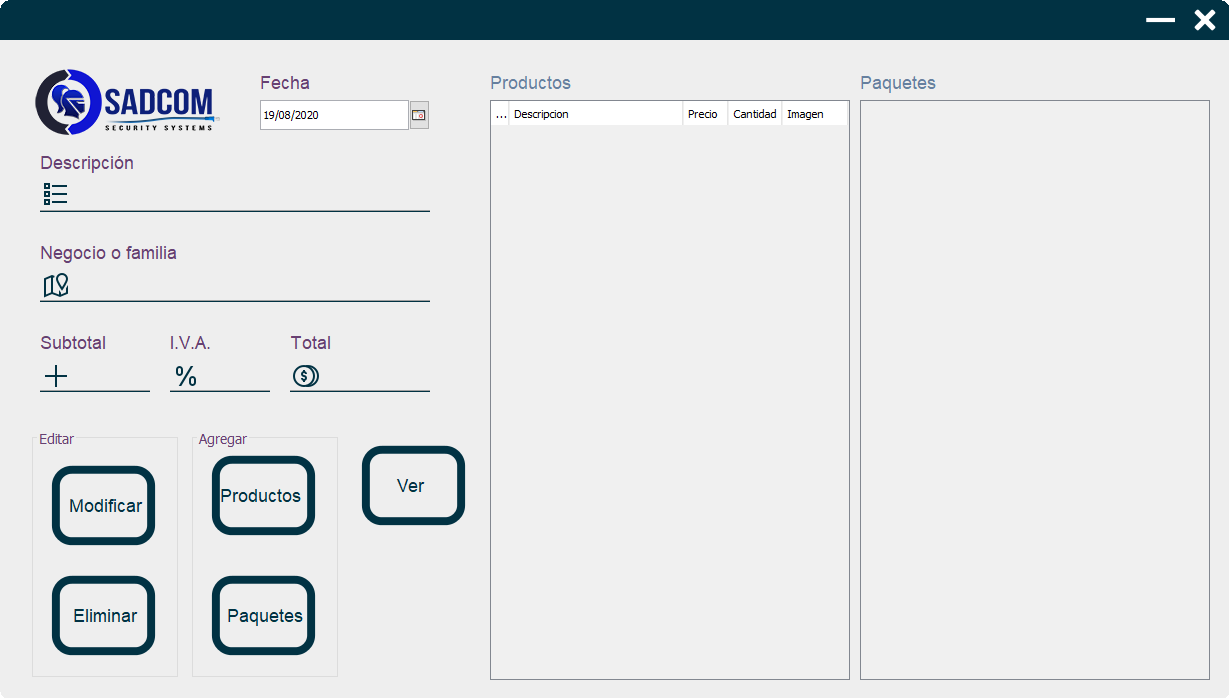


Figura 32: Pantalla para hacer cotizaciones.

Se pueden cambiar datos como la descripción o el negocio antes de salirse de la creación de la cotización. Al terminar de agregar productos se hace clic sobre ver y se arroja una hoja lista para imprimir con el detalle de la cotización. Las tablas de productos y paquetes muestran los elementos que se han agregado a la cotización de dichos tipos.

#### La pantalla para ver cotizaciones.

En esta pantalla se permite entrar a ver cotizaciones hechas anteriormente, primero es necesario encontrar la cotización respecto a los campos de búsqueda utilizándolos solos o en combinación, hacer clic sobre la cotización deseada y los campos se llenarán automáticamente, así mismo, se permite ver la hoja de cotización completa al hacer clic sobre ver.

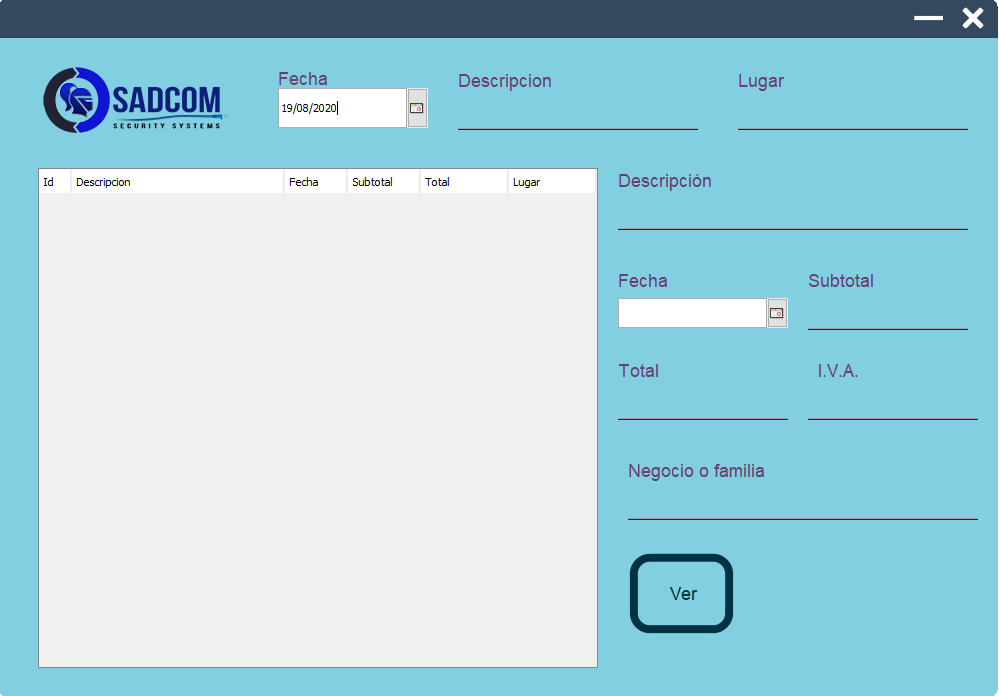


Figura 33: La pantalla para ver cotizaciones.

### La hoja de cotizaciones.

En el siguiente apartado se mostrará la creación de la hoja de cotizaciones variable creada gracias a la herramienta JasperReports, la cual permite pasar parámetros a una consulta a la base de datos y visualizar con un diseño agradable el resultado obtenido de la consulta.

En la parte superior se encuentra el logo de la empresa, seguido por atentamente, acompañado del nombre del lugar al cual la cotización va dirigida. Enseguida, se muestra un saludo y se presenta la cotización llamándola por el campo descripción según se almacenó en la base de datos. En la esquina superior izquierda se puede ver la fecha en la cual el documento fue emitido.

Después, se muestra un detalle de los productos incluidos o paquetes, a modo de tabla, empezando por la cantidad de productos, la descripción del producto, una imagen representativa del producto y, finalmente, el campo subtotal, obtenido al multiplicar la cantidad por el precio unitario.

También se muestra el número de página en la parte inferior, como un formato que indica que se encuentra en la página x de x totales. Debajo de la tabla se encuentra el total de la cotización, que se obtiene al sumar todos los subtotales.

En la parte inferior de la hoja se encuentra una breve despedida y las firmas del gerente de la empresa y la autorización del cliente, termina el saludo con una posdata indicando que los precios antes mencionados no incluyen el 16% del iva.

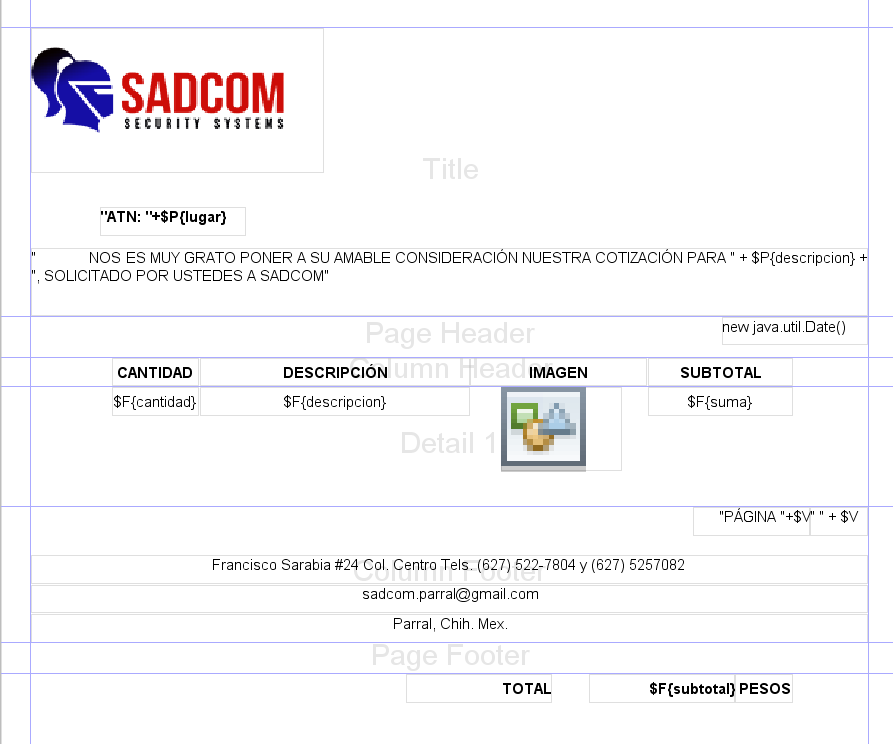


Figura 34: Diseño de la hoja de cotizaciones, parte 1.

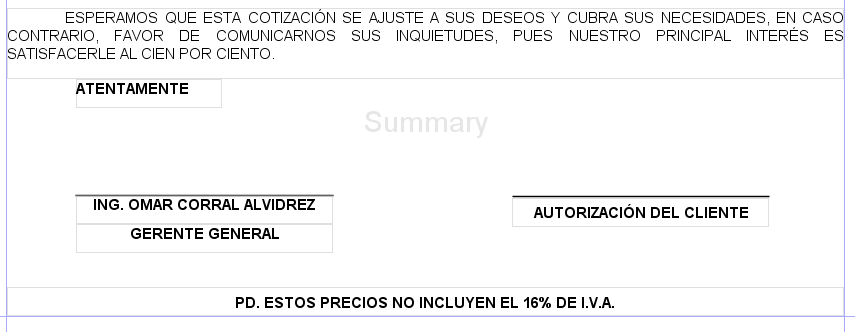


Figura 35: Diseño de la hoja de cotizaciones, parte 2.

En el marco de la página y para finalizar, se muestran los datos de contacto de la empresa, como calle, teléfonos de atención, un correo y la ubicación respecto al estado y municipio.

Cabe mencionar que en este diseño se cargan automáticamente los datos en función de los parámetros de entrada y con datos desde la base de datos, por lo que solo fue necesario hacer una hoja de cotizaciones para la totalidad de estas, ya que la hoja es solo un recipiente que será variable en cuanto a los datos que se alimenten automáticamente.

### Usabilidad y funciones del sistema.

A continuación, se describirán las funciones básicas del sistema y un paso por paso de cómo realizarlas.

#### Crear una categoría.

Para crear una categoría será necesario iniciar sesión, en el menú, hacer clic sobre el apartado de categorías y hacer clic en nueva, como se aprecia en la figura 36.

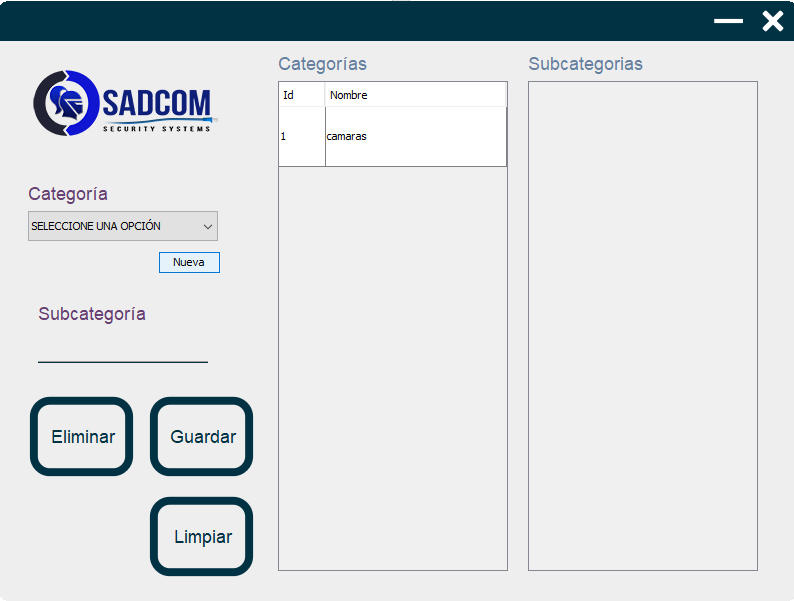


Figura 36: Nueva categoría.

Una vez presionado el botón de nueva aparece un cuadro de diálogo solicitando que se ingrese el nuevo nombre de la categoría.

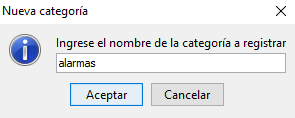


Figura 37: Introducir el nombre de la nueva categoría.

Ya que se ha guardado el nombre de la categoría la tabla de categorías se actualizará, mostrando el nuevo dato insertado.

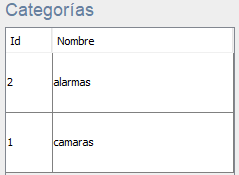


Figura 38: Tabla de categorías actualizada.

#### Crear una subcategoría.

Una vez dentro de la pantalla de categorías será necesario antes seleccionar la categoría a la cual pertenecerá la nueva subcategoría, esto se puede hacer de dos maneras, seleccionándolo directamente dentro del comboBox o haciendo clic sobre el elemento deseado en la tabla de categorías, lo cual llenará automáticamente el comboBox con la opción seleccionada. Cuando se hubo seleccionado la categoría, se ingresa el nombre de la subcategoría y se hace clic sobre el botón guardar, de ser satisfactoria la operación se mostrará un mensaje notificándolo.

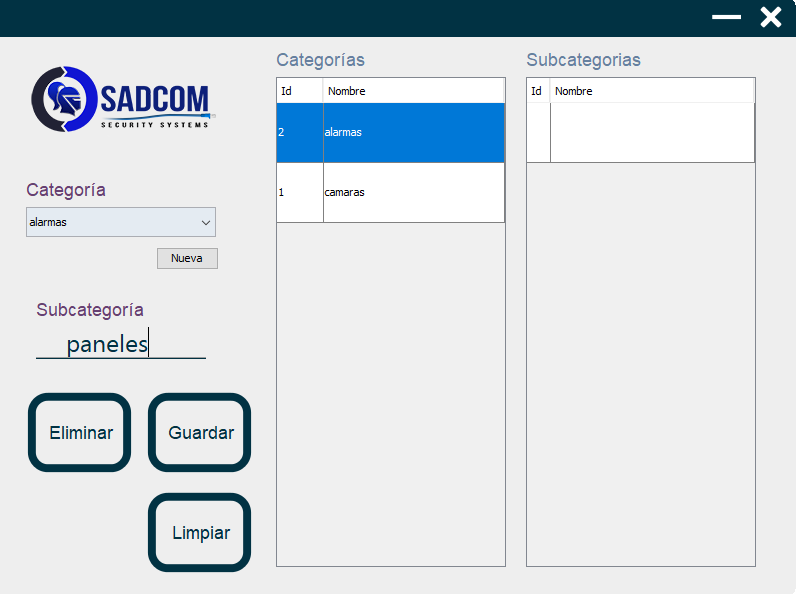


Figura 39: Registrar nueva subcategoría.

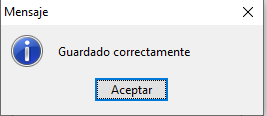


Figura 40: Subcategoría agregada satisfactoriamente.

Una vez que se ha guardado el cambio, se puede visualizar la categoría agregada formando parte de la tabla.

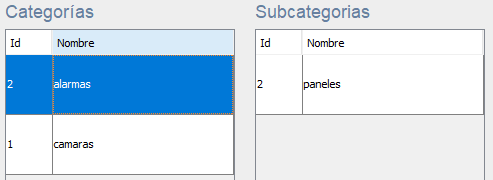


Figura 41: Tabla de subcategorías actualizada.

#### Eliminar categorías y subcategorías.

Dentro de la misma ventana, es posible eliminar categorías enteras y subcategorías. Es muy importante mencionar que al eliminar una categoría se eliminarán todas las subcategorías que tenía, del mismo modo, se eliminaran todos los registros de las demás tablas que tuvieran algo en común con la categoría en cuestión, por ejemplo, los productos y cotizaciones que pertenecían a esa categoría, lo mismo aplica al eliminar una subcategoría.

El procedimiento consiste en seleccionar solamente la categoría y dejar en blanco el campo de subcategoría, en caso de que se desee eliminar la categoría completa, antes de realizar la operación se muestra un cuadro de advertencia.

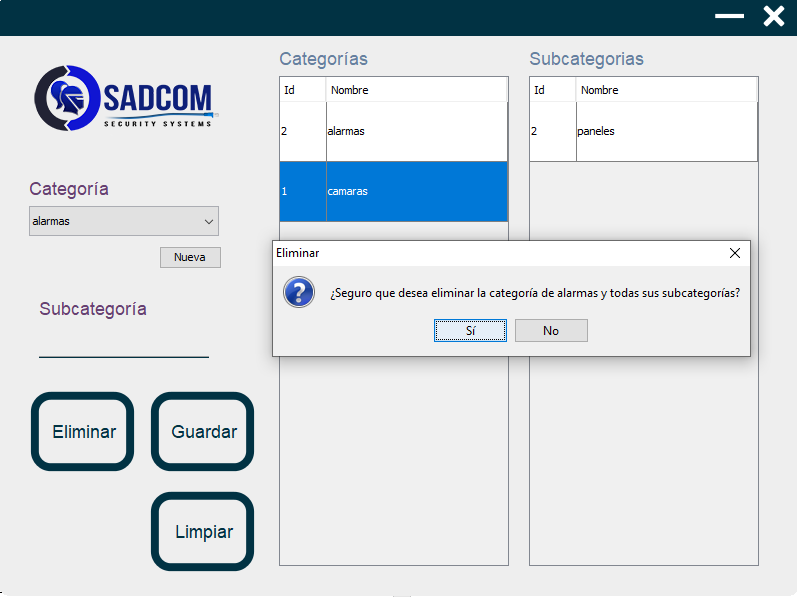


Figura 42: Eliminación de categoría.

Para eliminar una subcategoría es similar, solamente se tiene que ingresar el nombre de la subcategoría correctamente o en su caso, seleccionarla desde la tabla de subcategorías para que el campo se llene automáticamente, también se mostrará una ventana de confirmación. En caso de que cualquiera de los procedimientos sean realizados, las tablas correspondientes mostrarán los cambios.

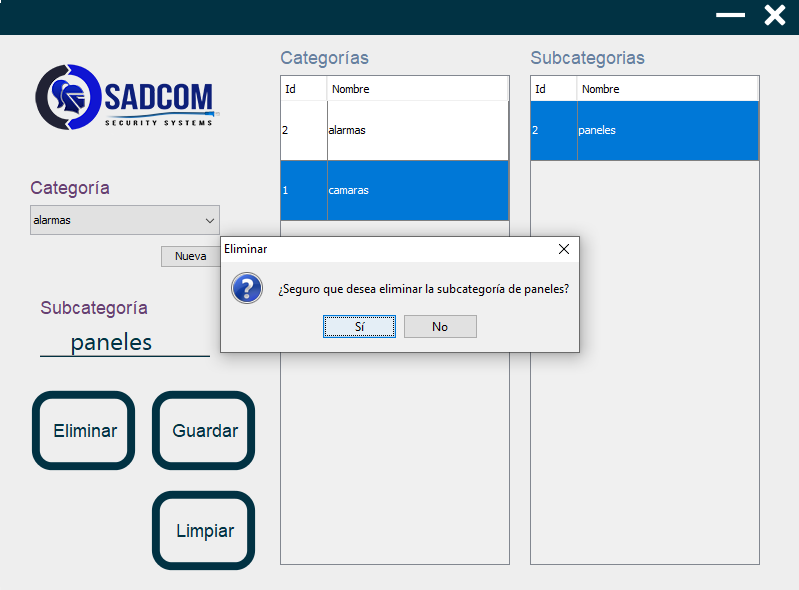


Figura 43: Eliminación de una subcategoría.

## Actividades adicionales

# ANÁLISIS DE LOS RESULTADOS

## Resultados obtenidos

## Conclusiones

# BIBLIOGRAFÍA

Acacio. (10 de Enero de 2016). *¿Qué es un sistema de cámaras CCTV?* Obtenido de Acacio: https://www.acacioseguridad.com/que-es-un-sistema-de-camaras-cctv/

Enrique Reinosa, C. M. (2012). *Bases de datos.* Buenos Aires: Alfaomega Grupo Editor Argentino.

Ibáñez, L. H. (2014). *Gestión de bases de datos .* Madrid: RA-MA.

Jaspersoft. (2020). *JasperReports Library*. Obtenido de Jaspersoft Community: https://community.jaspersoft.com/project/jasperreports-library

Maldonado, D. (3 de Septiembre de 2007). *¿Qué son los IDE de Programación?* Obtenido de El código K: https://elcodigok.blogspot.com/2007/09/que-son-los-ide-de-programacin.html

Moreno Nuñez, J., & Gómez Jimémez, E. (2019). *Fundamentos de programación Java con NetBeans .* Ciudad de México: Alfaomega Grupo Editor.

Pantaleo, G., & Rianudo, L. (2015). *Ingeniería de Software.* Buenos Aires: Alfaomega Grupo Editor Argentino.

Pérez, J. C. (2014). *ADMINISTRACIÓN DE SOFTWARE DE UN SISTEMA INFORMÁTICO.* Madrid: RA-MA, S.A. Editorial y Publicaciones .

Sierra, A. M. (2018). *Java Curso práctico de formación.* Ciudad de México: Alfaomega Grupo Editor.

Sierra, F. J. (2006). *Java 2. Lenguaje y aplicaciones.* Madrid: RA-MA Editorial.

Todter, K. (2020). *JCalendar*. Obtenido de ToEdter.