



UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

ESCUELA DE INGENIERÍA DE SISTEMAS Y TELEMÁTICA

IMPLEMENTACIÓN DEL MODELO ENTIDAD RELACIÓN EXTENDIDO.

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE

INGENIERO DE SISTEMAS Y TELEMÁTICA

**AUTOR:** NATALIA KARINA JARRÍN CUESTA

**DIRECTOR:** ING OSWALDO MERCHÁN.

CUENCA, ECUADOR

2017

## **Dedicatoria**

Este trabajo quiero dedicárselo a Dios, mi luz, mi guía y mi camino, porque es Él quien me ha sostenido y me ha bendecido con todas las personas a las que hoy quiero dedicar esta meta cumplida.

Esteban, mi bebé, mi alegría, mi razón de vivir, quien con sus travesuras y ocurrencias me ha acompañado en la elaboración de este trabajo y es mi motor que me da fuerzas para seguir adelante.

A Holvin mi compañero, la persona con la que puedo contar siempre, mi roca, mi amor y mi mejor amigo.

A mis padres Jorge y Ana quienes han sido mi guía, mi ejemplo de responsabilidad respeto, honestidad, cariño y apoyo incondicional

A mis hermanos, Pao, mi compañera, mi apoyo, quien alegra mi vida con sus locuras y me anima para seguir adelante y a Jorge, quien desde durante los primeros años de mi vida ha estado siempre para guiarme y ayudarme.

A mis tíos Monse e Iván de quienes he recibido apoyo y cariño incondicional.

A mi Sammy mi amiga fiel.

Natalia Jarrín

## **Agradecimientos**

Mi gratitud con la Universidad del Azuay por la formación que recibí de esta institución, por los conocimientos para cumplir esta meta.

Gracias a los profesores que a lo largo de la carrera compartieron generosamente sus conocimientos técnicas y herramientas necesarias para empezar mi vida profesional.

Quiero agradecer a mi Director de tesis, Ingeniero Oswaldo Merchán por haberme guiado y ayudado en el desarrollo del presente trabajo investigativo.

A mi tribunal el Ingeniero Francisco Salgado y el Ingeniero Marcos Orellana por brindarme su guía en el desarrollo de este trabajo.

Gracias a mis compañeros por su amistad, apoyo, ánimo y compañía.

Gracias a todas las personas que ayudaron directa e indirectamente en la realización de este proyecto

Natalia Jarrín

## Índice de contenidos

Dedicatoria .....	ii
Agradecimientos.....	iii
Índice de contenidos.....	iv
Índice de imágenes .....	vii
Índice de tablas .....	ix
Resumen .....	xi
Abstract .....	xii
Introducción .....	13
Objetivos .....	14
Objetivo General .....	14
Objetivos específicos .....	14
Justificación .....	14
Alcance y resultados esperados .....	15
CAPÍTULO I: EJEMPLO DE APLICACIÓN DEL MODELO EER .....	16
1.1    Introducción.....	16
1.2    Estudio de caso.....	16
CAPÍTULO II: CONCEPTOS DEL MODELO EER .....	19
2.1    Introducción .....	19
2.2    Especialización y generalización .....	19
2.2.1    Superclases.....	19
2.2.2    Subclases.....	20
2.2.3    Especialización .....	21
2.2.4    Generalización.....	22
2.3    Jerarquías y entramado de especialización y generalización .....	26
2.4    Herencia múltiple .....	27
2.5    Categorías .....	28

2.6 Agregación y Asociación .....	29
2.7 Reducción a tablas del modelo EER .....	30
2.7.1 Mapeo de especialización y generalización .....	31
2.7.2 Mapeo de herencia múltiple.....	34
2.7.3 Mapeo de categorías.....	34
CAPÍTULO III: ANÁLISIS Y MAPEO DEL ESTUDIO DE CASO “DISTRIBUIDORA DE LIBROS” .....	36
3.1 Introducción .....	36
3.2 Análisis .....	36
3.2.1 Entidades y atributos .....	36
3.2.2 Generalizaciones y especializaciones.....	37
3.2.3 Relaciones.....	38
3.2.4 Resumen de entidades y relaciones.....	39
3.3 Modelo EER.....	42
3.4 Mapeo del modelo EER .....	43
3.4.1 Entidades.....	43
3.4.2 Relaciones.....	46
CAPÍTULO IV: IMPLEMENTACIÓN EN EL GESTOR DE BASES DE DATOS.....	48
4.1 Introducción .....	48
4.2 Diccionario de datos.....	48
4.2.1 Entidades.....	48
4.2.2 Relaciones.....	53
4.3 Resultados .....	55
4.3.1 Descripción.....	56
4.4 Conclusiones .....	91
REFERENCIAS .....	92
Bibliografía.....	92
Anexos.....	93
Anexo A: Script de la base de datos “distribuidoralibrosuno’.....	93

Anexo B: Aprobación del protocolo de trabajo de titulación. ....	100
Anexo C: Diseño del trabajo de titulación. ....	101

## Índice de imágenes

Figura 1. 1 Diagrama EER para el caso del mundo real "distribuidora de libros". ...	18
Figura 2. 1 Representación de superclases.....	20
Figura 2. 2 Representación de subclases. ....	21
Figura 2. 3 Representación de una especialización.....	22
Figura 2. 4 Clases de especialización de tipos de vendedores. ....	22
Figura 2. 5 Representación de la generalización de tipos de vendedores.....	23
Figura 2. 6 Representación de las restricciones de la especialización. ....	24
Figura 2. 7 Disyunción total. ....	24
Figura 2. 8 Disyunción parcial.....	25
Figura 2. 9 Solapamiento total. ....	25
Figura 2. 10 Solapamiento parcial. ....	26
Figura 2. 11 Representación de jerarquías y entramado de especialización y generalización. ....	27
Figura 2. 12 Representación de la herencia múltiple. ....	28
Figura 2. 13 Representación de categoría con el mismo atributo clave. ....	28
Figura 2. 14 Representación de categoría con distinto atributo clave .....	29
Figura 2. 15 Agregación de la relación 'TRADUCCION' .....	30
Figura 2. 16 Reducción a tabla de la subclase VENDEDORMEDIOTIEMPO con la opción 1.A. ....	31
Figura 2. 17 Reducción a tabla de la subclase VENDEDORTIEMPOCOMPLETO con la opción 1.A. ....	31
Figura 2. 18 Reducción a tabla de la subclase VENDEDOR con la opción 1.A.....	32
Figura 2. 19 Reducción a tabla de la subclase CLIENTE con la opción 1.A.....	32
Figura 2. 20 Reducción a tabla de la superclase PERSONAS con la opción 1.B. ....	32
Figura 2. 21 Reducción a tabla de la subclase AUTOR con la opción 1.B.....	32
Figura 2. 22 Reducción a tabla de la superclase PERSONAS y de sus subclases AUTOR y TRADUCTOR con la opción 2.A.....	33
Figura 2. 23 Reducción a tabla de la superclase PERSONAS y de sus subclases VENDEDOR y CLIENTE con la opción 2.B.....	34
Figura 2. 24 Reducción a tabla de la categoría DONADOR. ....	35
Figura 2. 25 Reducción a tabla de la categoría TIPOCONSTRUCCION.....	35

Figura 4. 1 Ventana principal. ....	56
Figura 4. 2 Menú "Mantenimientos". ....	57
Figura 4. 3 Ventana del mantenimiento de almacenes. ....	57
Figura 4. 4 Resultado de la búsqueda por código. ....	58
Figura 4. 5 Resultado de la búsqueda por dirección. ....	59
Figura 4. 6 Ventana del mantenimiento de autores. ....	59
Figura 4. 7 Resultado de la búsqueda por código. ....	60
Figura 4. 8 Resultado de la búsqueda por apellido ....	61
Figura 4. 9 Ventana del mantenimiento de clientes. ....	61
Figura 4. 10 Resultado de la búsqueda por código. ....	62
Figura 4. 11 Resultado de la búsqueda por apellido. ....	63
Figura 4. 12 Listado de clientes mayoristas. ....	63
Figura 4. 13 Listado clientes minoristas. ....	64
Figura 4. 14 Ventana del mantenimiento de editoriales. ....	64
Figura 4. 15 Resultado de la búsqueda por código. ....	65
Figura 4. 16 Resultado de la búsqueda por nombre. ....	66
Figura 4. 17 Ventana del mantenimiento de idiomas. ....	66
Figura 4. 18 Resultado de la búsqueda por código. ....	67
Figura 4. 19 Resultado de la búsqueda por descripción. ....	67
Figura 4. 20 Ventana del mantenimiento de libros. ....	68
Figura 4. 21 Resultado de la búsqueda por código. ....	69
Figura 4. 22 Resultado de la búsqueda por título. ....	69
Figura 4. 23 Ventana del mantenimiento de traductores. ....	70
Figura 4. 24 Resultado de la búsqueda por código. ....	70
Figura 4. 25 Resultado de la búsqueda por apellido. ....	71
Figura 4. 26 Ventana del mantenimiento de vendedores. ....	72
Figura 4. 27 Resultado de la búsqueda por código. ....	73
Figura 4. 28 Resultado de la búsqueda por apellido. ....	74
Figura 4. 29 Resultado de la búsqueda por almacén. ....	75
Figura 4. 30 Listado de vendedores a tiempo completo. ....	75
Figura 4. 31 Listado de vendedores a medio tiempo. ....	76
Figura 4. 32 Menú administrar libros. ....	76
Figura 4. 33 Ventana "Administrar información de libros" ....	77
Figura 4. 34 Venta "Ingresar traductor". ....	78
Figura 4. 35 Listado de la información del responsable de la traducción. ....	78

Figura 4. 36 Ventana "Ingresar stock por almacén" .....	79
Figura 4. 37 Mensaje de petición de actualización.....	79
Figura 4. 38 Ventana "Actualizar stock" .....	80
Figura 4. 39 Menú "Consultas". .....	80
Figura 4. 40 Ventana Consulta libro (opciones de búsqueda).....	81
Figura 4. 41 Listado por autor.....	82
Figura 4. 42 Resultado de la búsqueda por título.....	83
Figura 4. 43 Resultado de la búsqueda por código.....	84
Figura 4. 44 Listado por idioma. ....	85
Figura 4. 45 Menú "Pedidos". .....	86
Figura 4. 46 Ventana de ingreso de pedidos. ....	87
Figura 4. 47 Ventana buscar vendedor.....	88
Figura 4. 48 Ventana buscar cliente. ....	88
Figura 4. 49 Ventana agregar libros.....	89
Figura 4. 50 Ventana consultar pedidos.....	90
Figura 4. 51 Ventana datos del pedido. ....	91

## Índice de tablas

Tabla 3. 1 Resumen de entidades en el estudio de caso "distribuidora de libros" ...	40
Tabla 3. 2 Resumen de relaciones en el estudio de caso "distribuidora de libros" ..	41
Tabla 3. 3 Reducción a tabla de la clase PERSONAS. ....	43
Tabla 3. 4 Reducción a tabla de la clase AUTOR.....	43
Tabla 3. 5 Reducción a tabla de la clase TRADUCTOR. ....	43
Tabla 3. 6 Reducción a tabla de la clase VENDEDOR. ....	44
Tabla 3. 7 Reducción a tabla de la clase VENDEDORMEDIOTIEMPO. ....	44
Tabla 3. 8 Reducción a tabla de la clase VENDEDORTIEMPOCOMPLETO.....	44
Tabla 3. 9 Reducción a tabla de la clase CLIENTE.....	44
Tabla 3. 10 Reducción a tabla de la clase MINORISTA. ....	44
Tabla 3. 11 Reducción a tabla de la clase MAYORISTA. ....	44
Tabla 3. 12 Reducción a tabla de la clase LIBRO.....	45
Tabla 3. 13 Reducción a tabla de la clase EDITORIAL. ....	45
Tabla 3. 14 Reducción a tabla de la clase IDIOMA.....	45
Tabla 3. 15 Reducción a tabla de la clase ALMACEN.....	45

Tabla 3. 16 Reducción a tabla de la clase PEDIDO.....	45
Tabla 3. 17 Reducción a tabla de la relación ESCRIBE. ....	46
Tabla 3. 18 Reducción a tabla de la relación TRADUCCION. ....	46
Tabla 3. 19 Reducción a tabla de la relación TRADUCE. ....	46
Tabla 3. 20 Reducción a tabla de la relación EDITA.....	46
Tabla 3. 21 Reducción a tabla de la relación DETALLE. ....	46
Tabla 3. 22 Reducción a tabla de la relación TIENE.....	47
Tabla 4. 1 Diccionario de datos de la tabla PERSONAS. ....	48
Tabla 4. 2 Diccionario de datos de la tabla AUTOR.....	49
Tabla 4. 3 Diccionario de datos de la tabla TRADUCTOR. ....	49
Tabla 4. 4 Diccionario de datos de la tabla VENDEDOR. ....	49
Tabla 4. 5 Diccionario de datos de la tabla VENDEDORMEDIOTIEMPO. ....	50
Tabla 4. 6 Diccionario de datos de la tabla VENDEDORTIEMPOCOMPLETO.....	50
Tabla 4. 7 Diccionario de datos de la tabla CLIENTE. ....	50
Tabla 4. 8 Diccionario de datos de la tabla MINORISTA. ....	51
Tabla 4. 9 Diccionario de datos de la tabla MAYORISTA. ....	51
Tabla 4. 10 Diccionario de datos de la tabla LIBRO.....	51
Tabla 4. 11 Diccionario de datos de la tabla EDITORIAL. ....	52
Tabla 4. 12 Diccionario de datos de la tabla IDIOMA.....	52
Tabla 4. 13 Diccionario de datos de la tabla ALMACEN. ....	52
Tabla 4. 14 Diccionario de datos de la tabla PEDIDO.....	53
Tabla 4. 15 Diccionario de datos de la tabla ESCRIBE. ....	53
Tabla 4. 16 Diccionario de datos de la tabla TRADUCCION. ....	54
Tabla 4. 17 Diccionario de datos de la tabla TRADUCE. ....	54
Tabla 4. 18 Diccionario de datos de la tabla EDITA.....	54
Tabla 4. 19 Diccionario de datos de la tabla DETALLE. ....	55
Tabla 4. 20 Diccionario de datos de la tabla TIENE.....	55

## **Resumen**

El presente proyecto busca contribuir al modelado de las bases de datos a través del modelo entidad relación extendido, al exponer la funcionalidad del modelo en un estudio de caso determinado y su aplicación en un sistema de gestión de base de datos. Adicionalmente busca aplicar las características con las que cuenta el modelo entidad relación extendido (EER) y de esta forma, mejorar la comprensión del modelado de bases de datos, ya que, esta variación del modelo entidad relación (ER) es de fácil entendimiento y se acerca más al lenguaje humano, gracias a su notación gráfica la información y sus relaciones se comprenden de mejor manera.

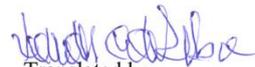
Se pretende sistematizar la información sobre el modelo EER, con ello aplicar dichos conceptos para definir de forma gráfica la estructura y funcionalidad de la base de datos y su aplicación en un sistema de gestión de base de datos (SGBD).

## Abstract

### ABSTRACT

The objective of this project was to contribute to data modeling through the Extended Entity-Relationship model by demonstrating its functionality in a particular case-study, and by applying it in a database management system. In addition, this study aimed at using the characteristics of the Extended Entity Relationship model (EER) in order to improve the understanding of database modeling, since this Entity Relationship model (ER) variation is easy to understand, and it is closer to human language. Hence, the information and its relations were better understood thanks to its graphic notation. The purpose was to systematize the EER model information, and in this way apply these concepts to graphically define the structure and functionality of database as well as its application in a Database Management System (DBMS).

  
UNIVERSIDAD DEL  
AZUAY  
Dpto. Idiomas

  
Translated by  
Lic. Lourdes Crespo

## Introducción

Al momento de diseñar una base de datos, los modelos conceptuales son una pieza vital dentro de las actividades, ya que por medio de estos se representa de forma gráfica los requerimientos de la misma. (Bajaj, 2010).

El modelo entidad relación extendido (EER) es un lenguaje que permite definir de forma gráfica la estructura y funcionalidad de una base de datos. Este modelo es una mejora del modelo entidad relación (ER) diseñado en 1976 por P.P Chen, usado como modelo base para el desarrollo de sistemas de bases de datos. Sin embargo en las décadas de los 80 y 90 aparecieron algunas extensiones para optimizarlo y mejorarlo. (Thalheim, 2009).

Existe una gran variedad de modelos conceptuales con los cuales es posible describir la estructura de datos de la base, sus restricciones de integridad y las operaciones de manipulación de datos; entre estos modelos los diseñadores y analistas de sistemas pueden optar por el modelo ER, que con sus mejoras y extensiones se obtiene el modelo EER, este modelo es de fácil aprendizaje, está más cerca del lenguaje humano que otros modelos y su notación gráfica ofrece un mejor entendimiento de la información. (Jain, Gore, & Singh, 2009).

Tanto el modelo ER como EER son capaces de representar estructuras complejas de la realidad por medio de entidades, es decir representan información del mundo real como personas, lugares, cosas, interacciones, entre otros, por medio de entidades y relaciones. Sin embargo el modelo EER incluye definiciones como subclase y superclase, además de los términos especialización, generalización, herencia múltiple y categoría, lo que facilita modelar de forma más precisa la realidad del sistema que se desea desarrollar. (Bagui, 2009), (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007).

En el presente trabajo se analizan los conceptos del modelo entidad relación extendido (EER). En el capítulo I se presenta un estudio del caso denominado “**distribuidora de libros**”, por medio del cual se ejemplifican los conceptos del modelo, sus restricciones, la estructura de datos, mapeado y diagramación. En el capítulo II se introduce los conceptos del modelo EER. En el capítulo III, se realiza el análisis del estudio de caso “**distribuidora de libros**” y su reducción a tablas o

mapeo. El capítulo IV contiene la Implementación en el gestor de bases de datos y conclusiones.

## **Objetivos**

### **Objetivo General**

Diseñar una base de datos usando el modelo entidad relación extendido e implementar en un sistema de gestión de base de datos (SGBD).

### **Objetivos específicos**

- Sistematizar información acerca del modelo EER.
- Definir el caso de estudio.
- Diseñar la base de datos.
- Implementar el modelo en un sistema de gestión de base de datos.

### **Justificación**

Para administrar de forma efectiva la información dentro de una base de datos es necesario que la misma sea: diseñada de forma correcta, que la información y sus relaciones sean entendibles, que el modelo utilizado sea de fácil comprensión para los involucrados. Si bien existe bibliografía especializada sobre los conceptos del modelo EER, no se encuentran estudios de caso en los cuales este se aplique, ni su relación con un software de gestión de bases de datos, es por ello que, se busca utilizar el modelo EER para el modelado e implementarlo en un sistema de gestión de base de datos.

## **Alcance y resultados esperados**

Sistematizar información acerca del modelo entidad relación extendido, al obtener un marco teórico de información sobre el modelo EER y sus características.

Definir el caso de estudio, detallando las clases que intervienen y sus relaciones.

Diseñar la base de datos, desarrollar el modelo entidad relación extendido sobre el caso de estudio, así como las tablas de las clases y relaciones dentro del modelo.

## CAPÍTULO I: EJEMPLO DE APLICACIÓN DEL MODELO EER

### 1.1 Introducción

En este capítulo se define el estudio de caso “**distribuidora de libros**”, el cual será utilizado para introducir los conceptos del modelo EER y cómo aplicarlos al diseñar una base de datos. En esta sección se determinan los requerimientos para el modelo, partiendo de los requisitos del usuario (mundo real), para posteriormente crear el esquema conceptual.

### 1.2 Estudio de caso

Se necesita diseñar una base de datos a partir del modelo EER con sus respectivas tablas para el manejo de la información y procesos de una distribuidora de libros, con almacenes ubicados en diferentes puntos, por lo que se precisa información sobre la ubicación de cada uno de ellos. Se maneja la siguiente información:

En cuanto a los libros se debe registrar su código, título, año de publicación, su editorial, idiomas en los cuales está disponible y precio de venta. Al mismo tiempo la distribuidora mantiene un registro de las editoriales, en donde consta su nombre, teléfono y ciudad.

Se busca también administrar la información como nombres, apellidos, fecha de nacimiento, teléfono, dirección y número de identificación tanto de autores, traductores, personal de vendedores y clientes, los cuales además de la información especificada en el punto anterior, tendrán características específicas para cada uno:

**Autores:** se registra su título profesional y nacionalidad.

**Traductores:** Es necesario conocer su nacionalidad

**Vendedor:** No existe rotación de personal en los diferentes almacenes de la distribuidora de libros, los datos que se deben registrar son la comisión fija que ganan por cada venta, la fecha en la cual se incorporó a trabajar y el almacén. Cabe señalar que los empleados pueden trabajar a medio tiempo o tiempo

completo, dependiendo del caso, se necesita conocer el número de horas, sueldo por hora y salario respectivamente.

Cliente: Existen dos tipos de clientes, mayoristas y minoristas, además cada cliente podrá acceder a un descuento especial para sus compras.

En el caso de que un vendedor del almacén necesite adquirir un libro, se lo registrará como cliente minorista.

La distribuidora también necesita mantener conocimiento sobre los idiomas en los que están traducidos los libros, por lo tanto se tiene una lista con la descripción o nombre del idioma, además se asume que un traductor realiza la transcripción del mismo a un determinado idioma, por lo que requiere a su vez información del responsable de dicha traducción, nombre, apellido, cédula o documento de identificación, nacionalidad, teléfono y dirección.

Finalmente los pedidos son realizados por los vendedores para su despacho, los mismos contienen información como: número de pedido, fecha del pedido, , la cantidad de libros por idioma, la dirección a la que debe ser enviado, el cliente que lo ha solicitado y el vendedor responsable del mismo.

En la Figura 1.1 se muestra el diagrama de un esquema EER para la base de datos “**Distribuidora de libros**”, de acuerdo a los requisitos del usuario. Por medio de este modelo se ejemplificarán los conceptos del EER.

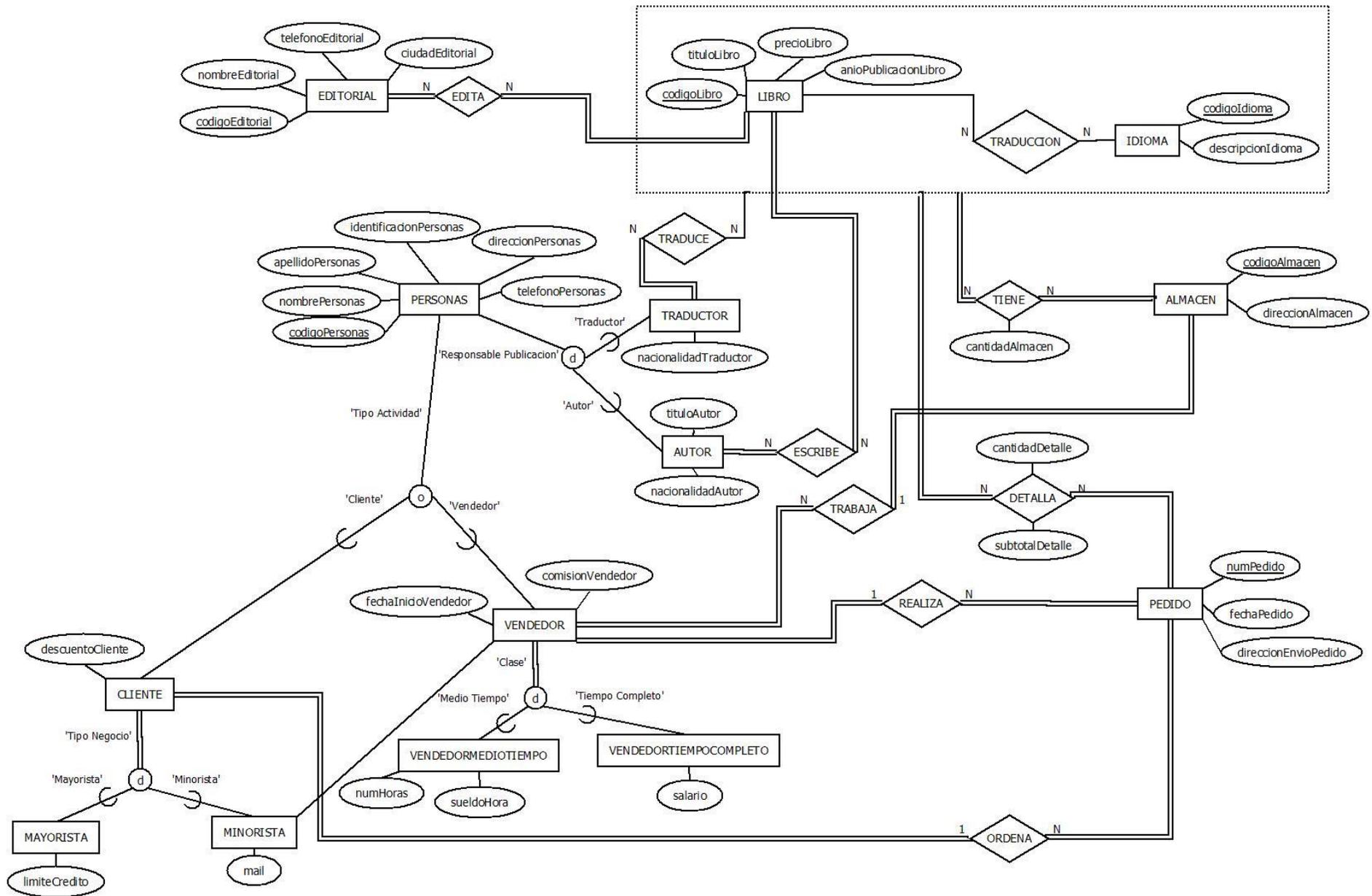


Figura 1. 1 Diagrama EER para el caso del mundo real "distribuidora de libros"

## CAPÍTULO II: CONCEPTOS DEL MODELO EER

### 2.1 Introducción

Este capítulo presenta y ejemplifica los conceptos adicionados al modelo ER para obtener el modelo EER a partir del estudio de caso “**distribuidora de libros**”, presentado en el capítulo I.

En la sección 2.2 se abordan los conceptos de especialización y generalización y sus restricciones, así como los conceptos de superclase, subclase y su relación. Posteriormente en la sección 2.3 se tratan los conceptos de jerarquías y entramado de especialización y generalización, mientras que en la sección 2.4 muestra el concepto de herencia múltiple. A continuación en la sección 2.5 se incorpora el concepto de la unión creando de esta forma las categorías. Finalmente en la sección 2.6 se ejemplifica por medio del estudio de caso “**distribuidora de libros**” las diversas opciones al momento de realizar el mapeo o reducción a tablas del modelo EER.

### 2.2 Especialización y generalización

Al introducir los conceptos de especialización y generalización se definen a la vez entidades llamadas superclases y subclases y cómo estas heredan atributos y se relacionan. (Connolly & Begg, 2005).

#### 2.2.1 Superclases

Es un tipo de entidad que posee uno o varios subgrupos de instancias, es decir de subclases. (Connolly & Begg, 2005). Por ejemplo, la superclase `PERSONA` con sus respectivas subclases `{CLIENTE, VENDEDOR, AUTOR, TRADUCTOR}` y a su vez cada una de estas puede convertirse en superclase al abarcar un conjunto de objetos mayor a los de sus subclases, como es el caso de `CLIENTE`, `{MINORISTA, MAYORISTA}`. Como se detalla en la Figura 2.1.

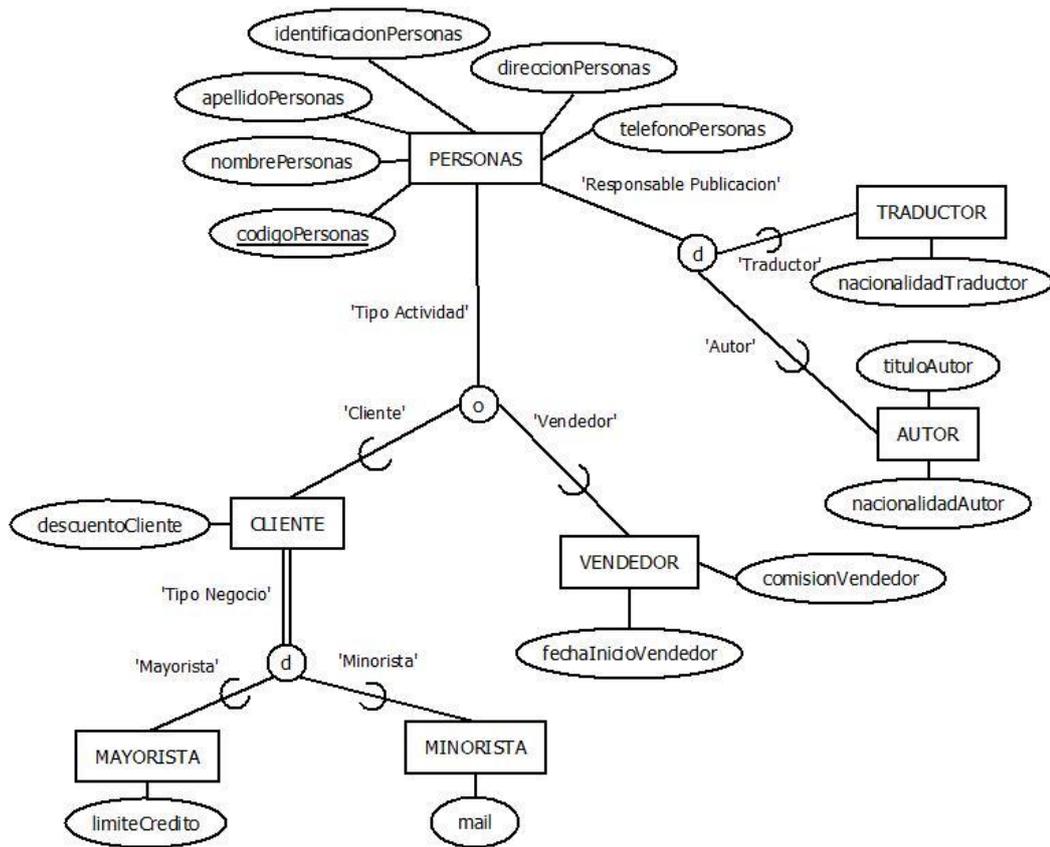


Figura 2. 1 Representación de superclases.

### 2.2.2 Subclases

Son instancias que heredan atributos de una superclase y tienen características propias diferenciadas entre sí. (Connolly & Begg, 2005), (Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla, 2007) . Por ejemplo, {VENDEDOR MEDIO TIEMPO, VENDEDOR TIEMPO COMPLETO} son subclases de VENDEDOR, las cuales comparten características comunes como COMISION Y FECHA INICIO, sin embargo tienen atributos diferentes como NUMHORAS, SUELDOPORHORA y SALARIO, respectivamente. Como se muestra en la figura 2.2

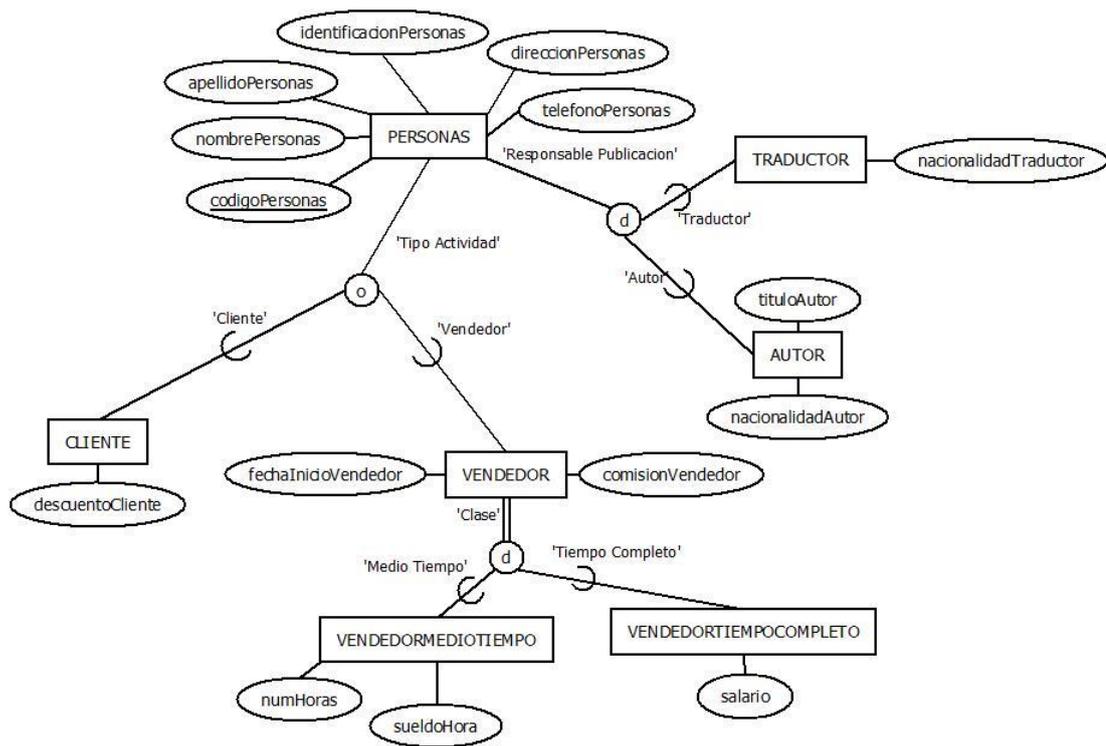


Figura 2. 2 Representación de subclases.

La relación entre una superclase y sus subclases se la denomina **clase/subclase**, la cual es una relación “**es un - es una**”. Por ejemplo, un **CLIENTE** es una **PERSONA** y **VENDEDOR** es una **PERSONA**. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007).

### 2.2.3 Especialización

La especialización define las subclases que pertenecerán a una determinada superclase, es posible tener varias especializaciones de una entidad, de acuerdo a las características que diferencian a los miembros de la misma. (Connolly & Begg, 2005) , (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007). Por ejemplo, una especialización de la clase **PERSONA**, según el tipo de actividad que realiza en la distribuidora, serían las subclases {**CLIENTE**, **VENDEDOR**} y otra especialización de la misma clase, de acuerdo al tipo de responsabilidad dentro de las publicaciones, podrían ser {**TRADUCTOR**, **AUTOR**}. Como se representa en la Figura 2.3

En un diagrama EER, la especialización se representa de la siguiente forma,

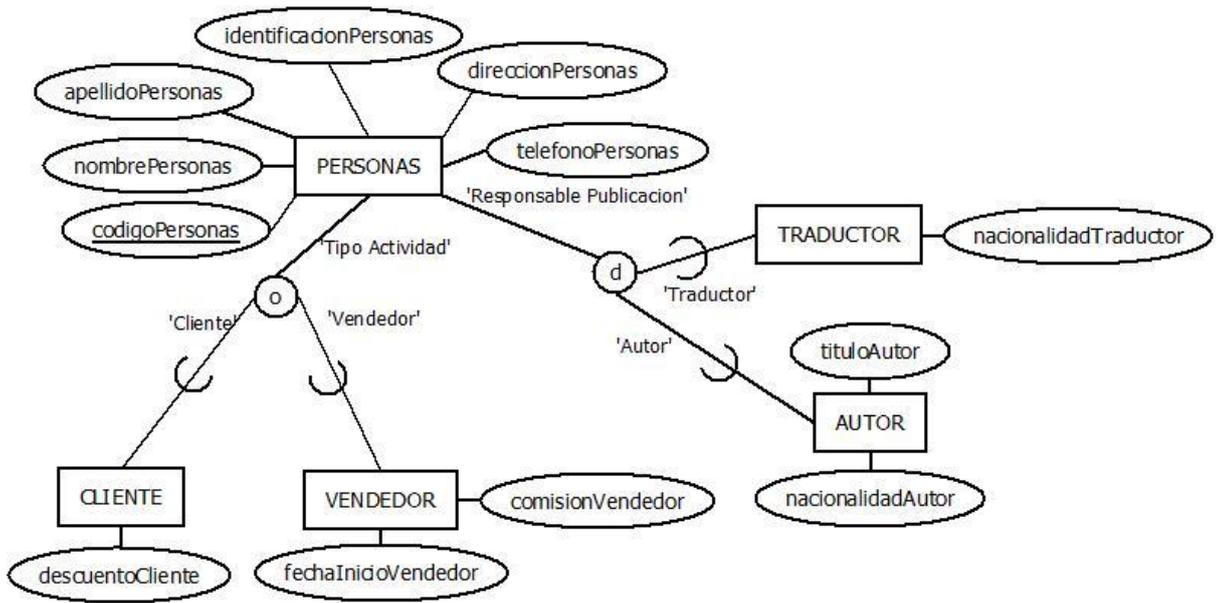


Figura 2. 3 Representación de una especialización

### 2.2.4 Generalización

Es un proceso en el cual se identifica una superclase al buscar atributos similares entre diversas entidades, es decir, la generalización es inversa a la especialización. (Connolly & Begg, 2005), (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007). Consideremos las clases {VENDEDOR MEDIO TIEMPO, VENDEDOR TIEMPO COMPLETO}, de la Figura 2.4, estas tienen atributos comunes como COMISION, FECHAINICIO y diferentes como NUMHORAS, SUELDOPORHORA, en el caso de VENDEDOR MEDIO TIEMPO y SALARIO, en el caso de VENDEDOR TIEMPO COMPLETO, en la generalización se minimizan o suprimen estas diferencias para obtener una superclase, en este caso VENDEDOR, la cual abarca los atributos comunes antes mencionados. Como se observa en la Figura 2.5.

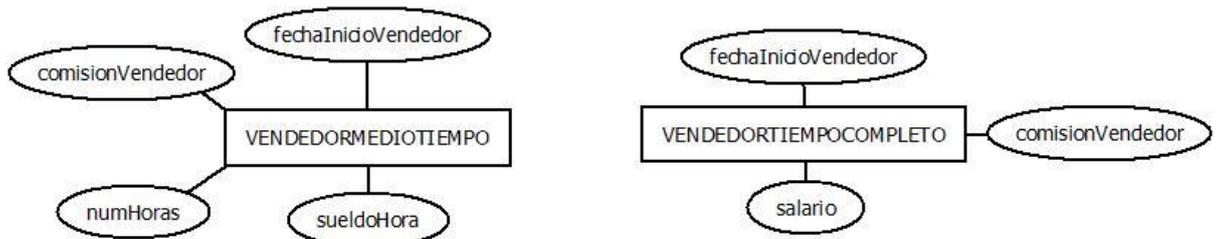


Figura 2. 4 Clases de especialización de tipos de vendedores.

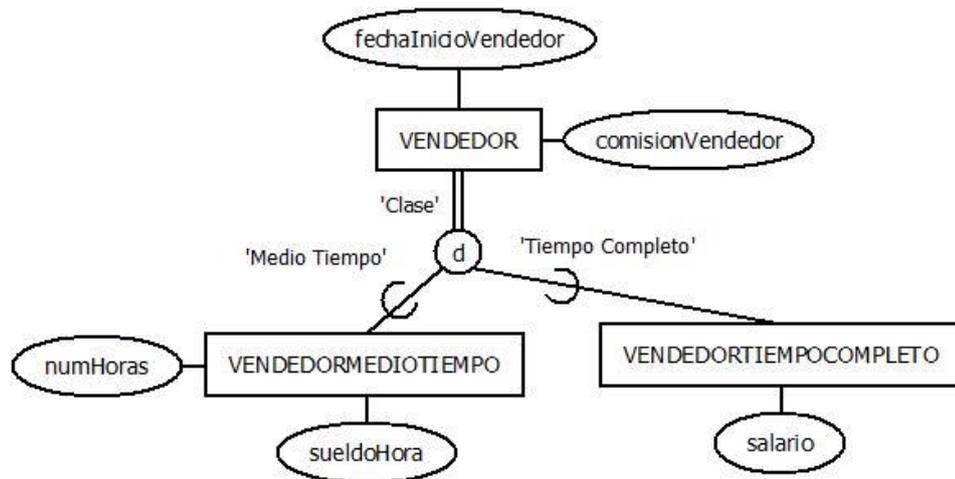


Figura 2. 5 Representación de la generalización de tipos de vendedores.

Tanto en la representación diagramática de la especialización, como en la de la generalización, se puede observar el símbolo U (subconjunto), el cual señala la dirección de la relación superclase/subclase. Dentro de la especialización se puede observar, por ejemplo, que las subclases {CLIENTE, VENDEDOR} y {AUTOR, TRADUCTOR} de la Figura 2.6, se encuentran conectadas a la superclase PERSONA por medio de un círculo, en donde se coloca el símbolo **d**, cuando se requiere representar la separación de las subclases de especialización, es decir que una entidad puede ser miembro solo de una de las subclases, esto se lo conoce como la **restricción de disyunción**, esta solamente ocurre cuando la superclase tiene más de una subclase. Por ejemplo, una PERSONA es un AUTOR o es un TRADUCTOR, sin embargo cuando se requiere representar la posibilidad de que las entidades puedan solaparse, formar parte de más de una de las subclases de especialización, se coloca la letra **o** en el círculo, indicando de esta forma la **superposición** de subclases. Por ejemplo en la Figura 2.6, una PERSONA es un VENDEDOR y también puede ser un CLIENTE. (Connolly & Begg, 2005), (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007).

Existe otro tipo de restricción dentro de la especialización, la **restricción de integridad**, como se representa en la Figura 2.6, permite indicar si se trata de una especialización **total** o **parcial**. Si la especialización es **total**, las entidades dentro de la superclase deben pertenecer como mínimo a una subclase, gráficamente se representa con una doble línea. Por ejemplo, cada CLIENTE debe ser MAYORISTA o MINORISTA. Cuando la especialización es **parcial**, las entidades de la superclase no se ven obligadas a pertenecer a ninguna de las subclases, su representación es

una línea sencilla. Por ejemplo en el caso de la especialización PERSONA, no todas las personas van a ser al mismo tiempo {CLIENTE – VENDEDOR} y {TRADUCTOR – AUTOR}.

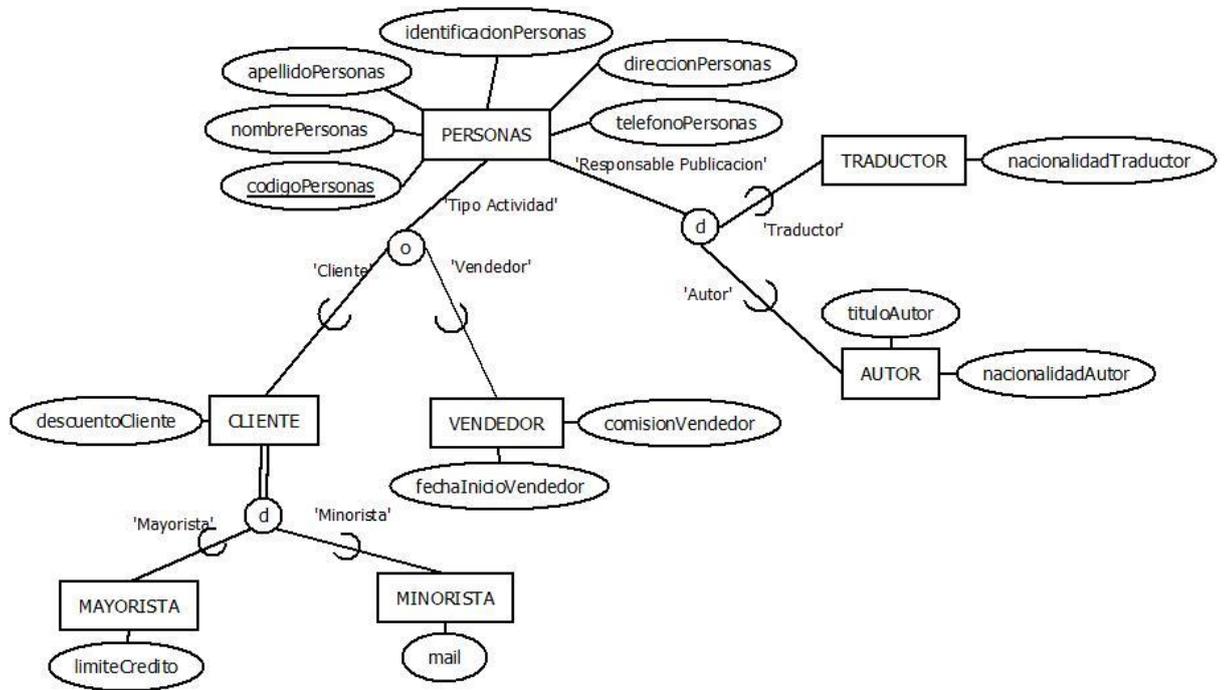


Figura 2. 6 Representación de las restricciones de la especialización.

Como conclusión se puede decir que existen cuatro tipos de especialización (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007):

- Disyunción total

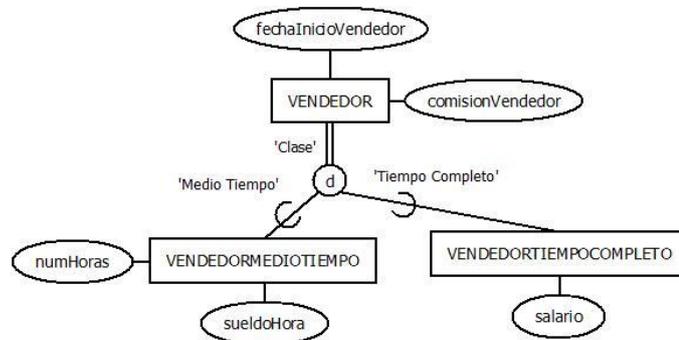


Figura 2. 7 Disyunción total.

- Disyunción parcial

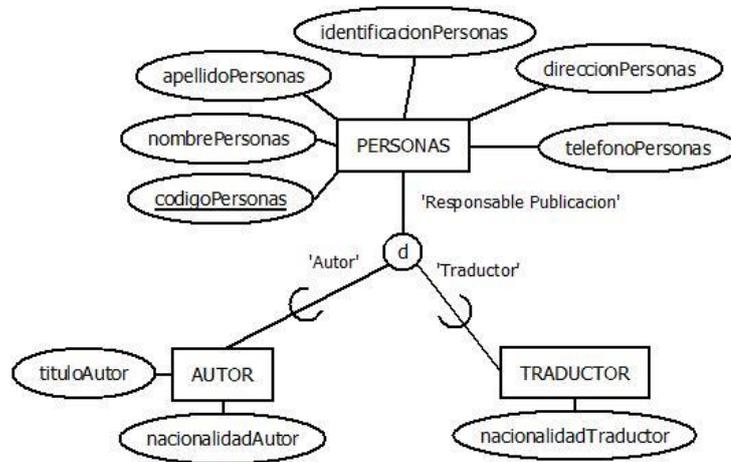


Figura 2. 8 Disyunción parcial.

- Solapamiento total

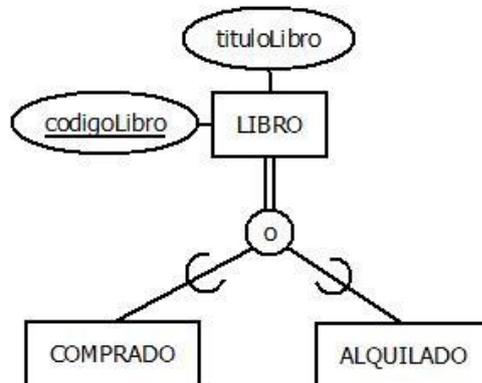


Figura 2. 9 Solapamiento total.

- Solapamiento parcial.

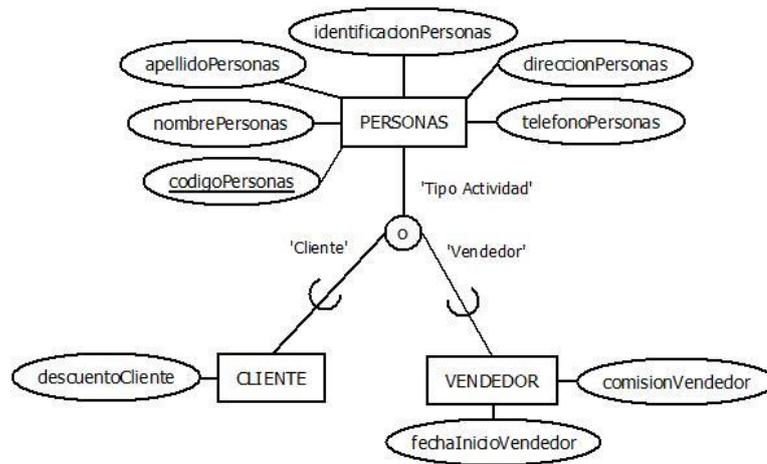


Figura 2. 10 Solapamiento parcial.

### 2.3 Jerarquías y entramado de especialización y generalización

En un diagrama de bases de datos se puede presentar una jerarquía o entramado de clases, esto ocurre cuando en una subclase podemos encontrar otras detalladas en ella. Por ejemplo en la Figura 2.11, VENDEDOR es una subclase de PERSONA, a la vez esta es superclase de las subclases VENDEDOR MEDIO TIEMPO y VENDEDOR TIEMPO COMPLETO. En la jerarquía de clases, dicha subclase puede tener un solo padre, es decir participar de una sola relación **clase/subclase**, como es el caso de MAYORISTA cuya única clase padre es VENDEDOR, sin embargo, cuando una subclase participa en más de una relación **clase/subclase** es un caso de entramado de especialización, como en el estudio de caso “**distribuidora de libro**” la subclase MINORISTA, la cual participa en la relación VENDEDOR y CLIENTE. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007)

Debido a la jerarquía o entramado de especialización se puede observar que el diagrama del modelo EER tiene forma de un árbol. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007). Estructurado por medio de las subclases y las superclases a las que pertenecen.

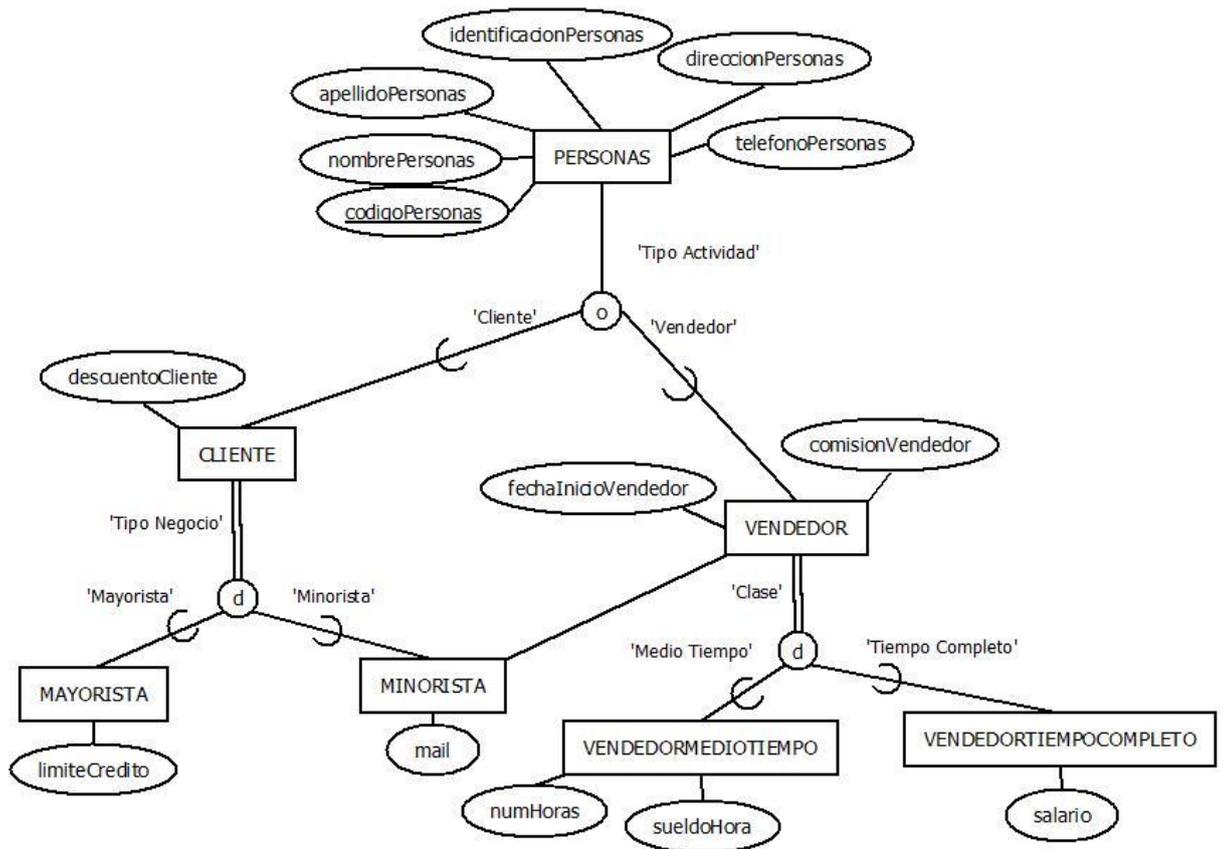


Figura 2. 11 Representación de jerarquías y entramado de especialización y generalización.

## 2.4 Herencia múltiple

En la especialización las subclases heredan los atributos de la superclase, no obstante pueden existir subclases que hereden atributos de varias clases, a esto se lo conoce como herencia múltiple y a la subclase que hereda se la denomina subclase compartida. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007) Por ejemplo, la subclase MINORISTA, de la Figura 2.12, la cual es una subclase compartida de CLIENTE y VENDEDOR.

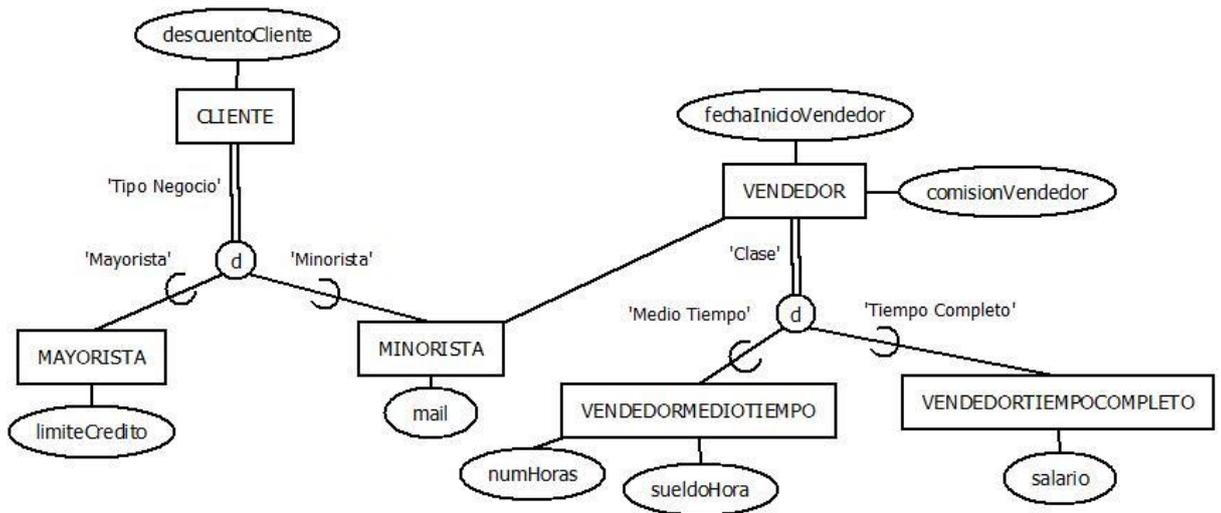


Figura 2. 12 Representación de la herencia múltiple.

## 2.5 Categorías

En un modelo EER se puede representar una subclase de tipo **unión** o **categoría** cuando esta hereda atributos de entidades de diferentes tipos, las cuales forman parte de dicha relación **superclase/subclase**. Las categorías también pueden ser parciales o totales, según la participación de la relación, esto se representa de forma similar a la especialización, por medio de líneas simples y dobles respectivamente. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007). Por ejemplo, la categoría TIPOCONSTRUCCION, de la Figura 2.13, la cual es una colección de las clases DEPARTAMENTO y CASA las cuales comparten el mismo atributo clave.

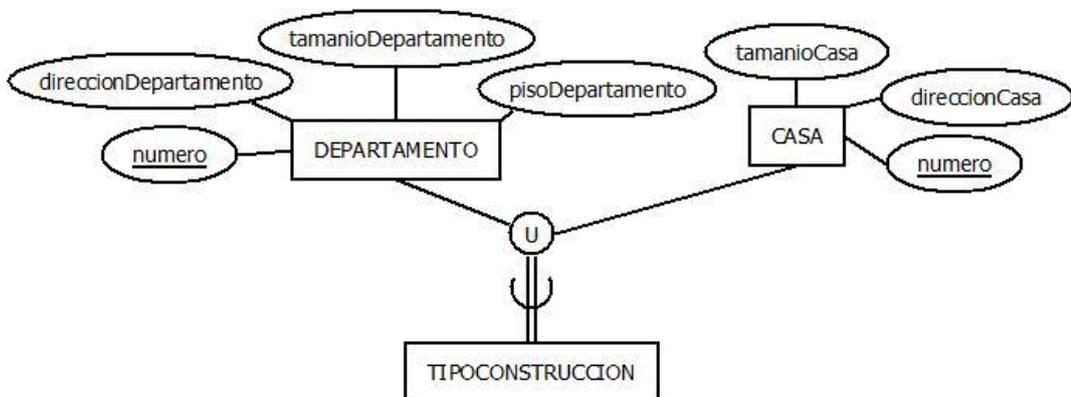


Figura 2. 13 Representación de categoría con el mismo atributo clave.

Pueden existir **categorías** cuyos atributos clave sean diferentes como es el caso de la Figura 2.14, en donde la categoría DONADOR es una colección de las clases PERSONA y EMPRESA.

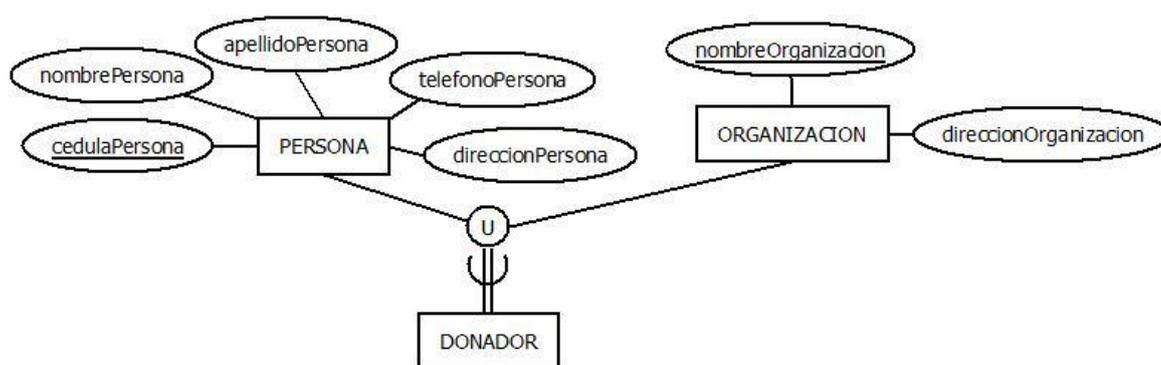


Figura 2. 14 Representación de categoría con distinto atributo clave

## 2.6 Agregación y Asociación

Es un concepto abstracto, el cual permite combinar entidades, es decir crear objetos compuestos gracias a sus entidades componentes, entre dichas entidades existe relación lo cual permite la creación de otra y por lo tanto se puede utilizar esta nueva entidad en una nueva relación. (Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla, 2007), (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007).

Según los autores Elmasri & Navathe en su libro Fundamentos de Sistemas de Bases de Datos 2007, existen tres situaciones en las que se usa la agregación en el modelo EER:

- Cuando se crea un objeto complejo a partir de atributos de otro objeto.
- Al representar una agregación como una relación común.
- Al relacionar en un objeto agregado de nivel superior objetos ya relacionados entre sí. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007).

La relacion de agregación se define como **es una parte de – un componente de**. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007)

La asociación por su parte, es una abstracción la cual ayuda a asociar objetos que provienen de varias clases independientes y se define como: **está asociada con**. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007)

La diferencia entre asociación y agregación recae en que en una agregación la instancia y sus objetos son dependientes entre sí, es decir si se borra el objeto compuesto se eliminarán los objetos que lo componen, sin embargo, en la asociación si la instancia es borrada, los demás objetos seguirán existiendo. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007)

En la Figura 2.15 se puede observar la agregación de la relación 'TRADUCCION' por medio de la cual se conoce sobre los idiomas en los que está traducido un libro, requisito especificado en el enunciado del estudio de caso “**distribuidora de libros**”.

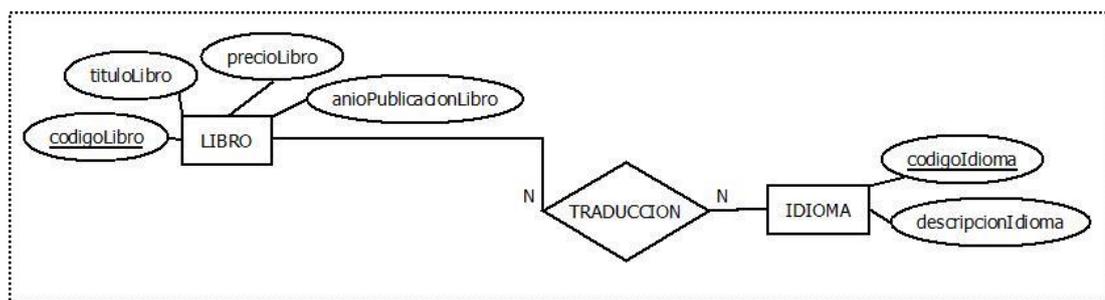


Figura 2. 15 Agregación de la relación 'TRADUCCION'

## 2.7 Reducción a tablas del modelo EER

Si bien se pueden seguir los pasos, para la reducción a tablas o mapeado, especificados en el modelo ER para la mayor parte del modelo EER, en el caso del mapeo de la especialización y generalización se presentan diversas alternativas así como es el caso de la herencia múltiple y de las categorías.

## 2.7.1 Mapeo de especialización y generalización

### 2.6.1.1 Alternativa 1: Varias relaciones

Opción 1.A: **Relaciones sólo de subclase**, se crea una relación por cada subclase, como restricción en esta opción se puede observar que únicamente se aplica si las subclases de especialización son **totales**, ya que en el caso de ser **parciales**, se generaría redundancia de datos en diferentes tablas, es decir las entidades se repetirían en varias relaciones. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007).

En el ejemplo 1 de las Figuras 2.15 y 2.16 se puede observar el mapeo de las subclases **totales**, VENDEDORMEDIOTIEMPO y VENDEDORTIEMPOCOMPLETO respectivamente.

VENDEDORMEDIOTIEMPO	codigoPersonas	identificacionPersonas	nombrePersonas	apellidoPersonas	telefonoPersonas	direccionPersonas	comision	fechaInicio	numHoras	sueldoHora
	Per001	2222222222	Natalia	Jarrín	2589632	San José	0.05	10/5/2010	4	10.5
	Per003	3333333333	Esteban	Molina	2589785	Av. 1 de Mayo	0.03	5/2/2015	4	10.5

Figura 2. 16 Reducción a tabla de la subclase VENDEDORMEDIOTIEMPO con la opción 1.A.

VENDEDORTIEMPOCOMPLETO	codigoPersonas	identificacionPersonas	nombrePersonas	apellidoPersonas	telefonoPers	direccionPersonas	comision	fechaInicio	salario
	Per002	6666666666	Andrés	Cepeda	2835698	Av. Loja	0.10	5/10/2009	2500
	Per005	7777777777	Víctor	Vintimilla	2456896	Remigio Crespo	0.15	4/12/2015	800

Figura 2. 17 Reducción a tabla de la subclase VENDEDORTIEMPOCOMPLETO con la opción 1.A.

En el ejemplo 2, de las Figuras 2.17 y 2.18, muestra el mapeo de las **subclases solapadas**, CLIENTE y VENDEDOR, se puede observar cómo se repiten los datos de *Mitch Davis*, ya que además de ser vendedor también puede realizar compras en la distribuidora.

VENDEDOR	codigoPersonas	identificacionPersonas	nombrePersonas	apellidoPersonas	telefonoPersonas	direccionPersonas	comision	fechalncio
	Per001	5555555555	Ana	Cuesta	2415263	SanJosé	0.05	10/5/2010
	Per002	4444444444	Mitch	Davis	2859583	Miami	0.03	5/2/2015

Figura 2. 18 Reducción a tabla de la subclase `VENDEDOR` con la opción 1.A.

CLIENTE	codigoPersonas	identificacion	nombrePersonas	apellidoPersonas	telefonoPersonas	direccionPersonas	descuentoCliente
	3	1111111111	Ernest	Hemingway	2831259	Av. Américas	0.10
	2	4444444444	Mitch	Davis	2859583	Miami	0.03

Figura 2. 19 Reducción a tabla de la subclase `CLIENTE` con la opción 1.A.

Opción B: **Relaciones de superclases y subclases**, se crean relaciones tanto para las subclases como para las superclases, esta forma de mapeo puede ser utilizada en especializaciones totales o parciales y solapadas o disjuntas, sin embargo esta opción crea demasiadas tablas del modelo. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007).

Esta opción es la que se utilizará para la reducción a tablas del estudio de caso “**distribuidora de libros**”, como se puede observar en la Figura 2.19 y 2.20

PERSONAS	codigoPersonas	identificacionPersonas	nombrePersonas	apellidoPersonas	telefonoPersonas	direccionPersonas
	Per001	1111111111	Ernest	Hemingway	0992515269	Av. Américas
	Per002	2222222222	Natalia	Jarrín	0985625896	San José
	Per003	3333333333	José	Alvarado	0985236652	Av. 1 de Mayo
	Per004	4444444444	Andrés	Cepeda	0896222463	Av. Loja
	Per005	5555555555	Víctor	Vintimilla	0978562322	Remigio Crespo
	Per006	6666666666	Lino	Novás Calvo	0984665893	Cuba
	Per007	7777777777	Esteban	Perez	0895555795	París

Figura 2. 20 Reducción a tabla de la superclase `PERSONAS` con la opción 1.B.

AUTOR	codigoAutor	nacionalidadAutor	tituloAutor
	Per001	Estadounidense	Lic. Comunicación

Figura 2. 21 Reducción a tabla de la subclase `AUTOR` con la opción 1.B.

### 2.6.1.2 Alternativa 2: Una sola relación

Opción 2.A: **Con un atributo de tipo**, se crea la tabla de la superclase, es decir una sola relación y se añade un atributo que ayuda a diferenciar a que subclase pertenece la tupla, a este se lo denomina **atributo tipo** o **discriminatorio**, esta opción de mapeo se la utiliza si las subclases son disjuntas, y puede generar demasiados valores nulos, en especial si se trata de una participación parcial en donde una entidad podría no pertenecer a ninguna subclase. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007).

Ejemplo 1: En la Figura 2.21, se realiza el mapeo de la superclase PERSONA y de sus subclases AUTOR Y TRADUCTOR, en este caso, el **atributo tipo** es *{tipoActividad}*, el cual permite conocer si se trata de un traductor o de un cliente y de acuerdo al tipo de actividad especificada llenar los datos para los atributos correspondientes según sea el caso, se puede observar que se generan varios valores nulos, en especial para Esteban Molina, quien no pertenece ninguna de las subclases.

PERSONAS	codigoPersonas	identificacionPersonas	nombrePersonas	apellidoPersonas	telefonoPersonas	direccionPersonas	tipoActividad	nacionalidadTraductor	tituloAutor	nacionalidadAutor
Per001		2222222222	Natalia	Jarrín	2589632	San José	TRADUCTOR	Estadounidense	NULL	NULL
Per002		3333333333	Ernest	Hemingway	2589785	Av. Américas	AUTOR	NULL	Lic.Comunicación	Ecuatoriano
Per003		4444444444	Esteban	Molina	2859583	Av. 1 de Mayo	NULL	NULL	NULL	NULL

Figura 2. 22 Reducción a tabla de la superclase PERSONAS y de sus subclases AUTOR y TRADUCTOR con la opción 2.A.

Opción 2.B: **Varios atributos tipo**, se crea una sola relación con los atributos de la superclase y de las subclases, por cada subclase, se crea un **atributo tipo booleano** que indica si pertenece o no a determinada subclase y de esta forma, una entidad que no pertenezca tendrá valores NULL para los atributos que no le corresponden. Esta forma de mapeo se la utiliza cuando se representan subclases solapadas o especializaciones disjuntas. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007).

Ejemplo 1: En el ejemplo de la Figura 2.22, se realiza el mapeo de la superclase PERSONA y de sus subclases VENDEDOR y CLIENTE, las cuales se pueden solapar, por lo tanto, un vendedor puede en algún momento convertirse en cliente, antes de

colocar los atributos de cada subclase, se crea un **atributo bandera**, en este caso  $\{clienteFlag\}$  y  $\{vendedorFlag\}$  respectivamente, y según sea el caso se llenan los datos de la tabla.

PERSONAS	codigoPersonas	identificacionPersonas	nombrePersonas	apellidoPersonas	telefonoPersonas	direccionPersonas	clienteFlag	descuentoCliente	vendedorFlag	comisionVendedor	fechalnicioVendedor
	Per001	1111111111	Ernest	Hemingway	2831259	Av. Américas	TRUE	0.10	FALSE	NULL	NULL
	Per002	2222222222	Natalia	Jarrin	2589632	San José	FALSE	NULL	TRUE	0.05	10/5/2014
	Per003	3333333333	Esteban	Molina	2589785	Av. 1 de Mayo	FALSE	NULL	FALSE	NULL	NULL
	Per004	4444444444	Mitch	Davis	2859583	Miami	TRUE	0.05	TRUE	0.10	4/5/2000

Figura 2. 23 Reducción a tabla de la superclase PERSONAS y de sus subclases VENDEDOR y CLIENTE con la opción 2.B.

### 2.7.2 Mapeo de herencia múltiple.

Para la reducción a tablas de las subclases compartidas se pueden utilizar todas las opciones de mapeo explicadas en el punto anterior, tanto las de la opción A como de la opción B, adicionalmente se debe tener presente que todas las clases tengan el mismo atributo clave ya que de otra forma la subclase se modela como una categoría.

### 2.7.3 Mapeo de categorías.

En la sección 2.5 Categorías, del presente capítulo, se explica el concepto de **categoría** o subclase de tipo **unión** mediante dos ejemplos:

El primer ejemplo “DONADOR” de la figura 2.14, es una categoría formada por las clases PERSONA y ORGANIZACIÓN, en este caso se presentan claves distintas ya que representan entidades diferentes, por lo que es necesario crear una **clave sustituta** o también llamada **clave subrogada** de igual forma se crea una tabla que representa a la **categoría** cuya llave principal será la antes mencionada **clave sustituta**, a la vez esta será una llave foránea que se hereda en cada relación de las superclases dentro de la **categoría**. (Elmasri & Navathe, Fundamentos de Sistemas de Bases de Datos, 2007) (Instituto Tecnológico Superior de Lerdo, 2010)

En la Figura 2.23 se crea la relación DONADOR, de la categoría DONADOR y se define el atributo {llaveDonador} como la llave principal y a la vez **clave sustituta**, la cual se agrega como llave foránea en las demás relaciones de la categoría.

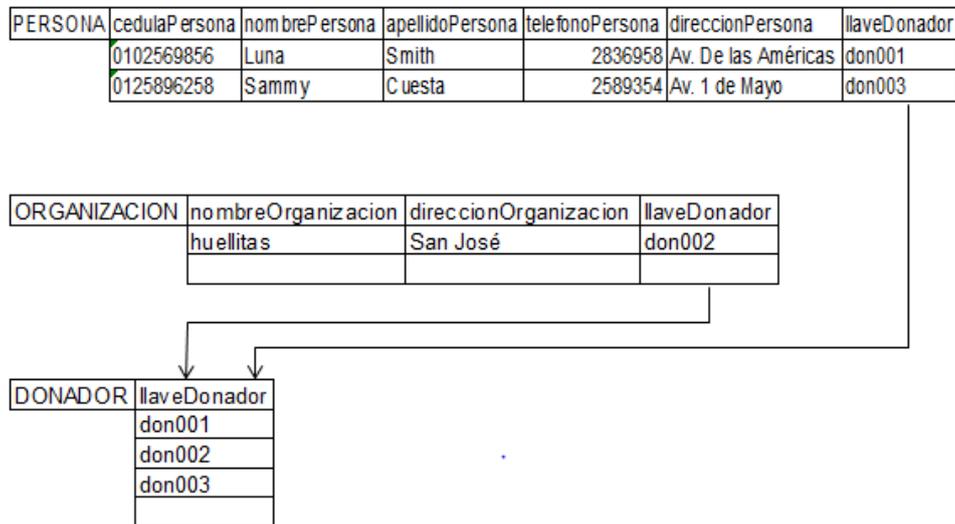


Figura 2. 24 Reducción a tabla de la categoría DONADOR.

El segundo ejemplo “TIPOCONSTRUCCION” de la Figura 2.13, la cual es una colección de las clases DEPARTAMENTO y CASA, comparten un mismo atributo clave por lo cual no es necesario crear la **clave sustituta** como se muestra en la Figura 2.24.

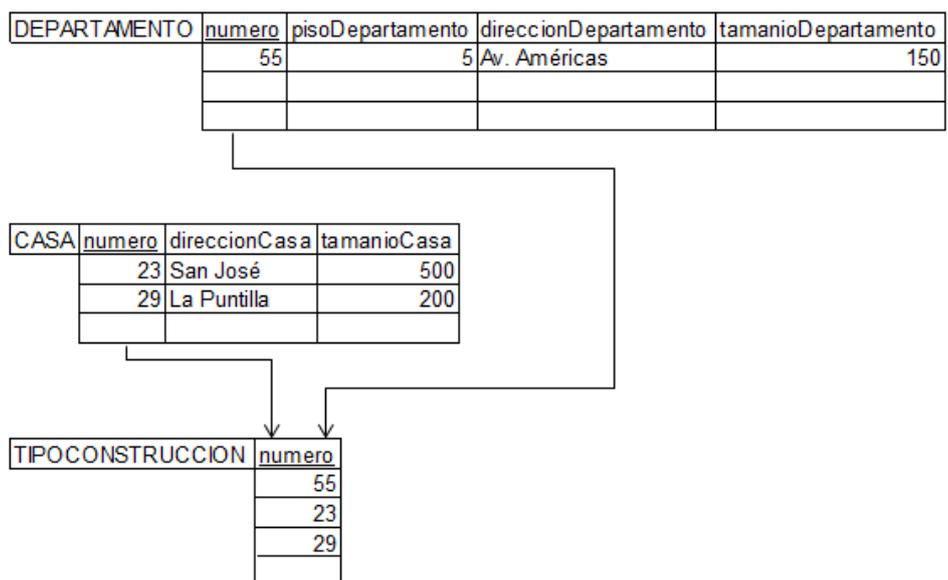


Figura 2. 25 Reducción a tabla de la categoría TIPOCONSTRUCCION.

## **CAPÍTULO III: ANÁLISIS Y MAPEO DEL ESTUDIO DE CASO “DISTRIBUIDORA DE LIBROS”**

### **3.1 Introducción**

En este capítulo se presenta el desarrollo del estudio de caso “**distribuidora de libros**”, basado en los conceptos exhibidos en el capítulo II del presente documento. Se realiza el análisis, se define el modelo EER con sus clases, relaciones y tablas.

### **3.2 Análisis**

Después de revisar los conceptos del modelo EER, para el estudio de caso “**distribuidora de libros**”, presentado en el capítulo I, se define lo siguiente:

#### **3.2.1 Entidades y atributos**

En el modelo constarán las clases: PERSONAS, AUTOR, TRADUCTOR, VENDEDOR, CLIENTE, VENDEDORTIEMPOCOMPLETO, VENDEDORMEDIOTIEMPO, MAYORISTA, MINORISTA, LIBRO, EDITORIAL, IDIOMA, ALMACEN y PEDIDO con sus respectivos atributos:

PERSONAS: `codigoPersonas`, la cual es la llave primaria de la clase, `nombrePersonas`, `apellidoPersonas`, `identificacionPersonas`, `direccionPersonas` y `telefonoPersonas`.

Las clases CLIENTE, VENDEDOR, TRADUCTOR, AUTOR son subclases de PERSONAS, por lo tanto heredan sus atributos.

AUTOR: `nacionalidadAutor`.

CLIENTE: `descuentoCliente`.

VENDEDOR: `fechaInicioVendedor` y `comisionVendedor`.

TRADUCTOR: nacionalidadTraductor.

Adicionalmente las clases MAYORISTA, MINORISTA, VENDEDORTIEMPOCOMPLETO, VENDEDORMEDIOTIEMPO son subclases de CLIENTE y VENDEDOR, respectivamente, y heredan sus atributos así como los que estas clases hayan heredado de PERSONAS.

VENDEDORTIEMPOCOMPLETO: salario.

VENDEDORMEDIOTIEMPO: numHoras, sueldoHora.

MAYORISTA: limiteCredito.

MINORISTA: mail.

LIBRO: codigoLibro, es la llave primaria de la clase, tituloLibro, precioLibro, anioPublicacionLibro.

EDITORIAL: codigoEditorial, es la llave primaria de la clase, nombreEditorial, telefonoEditorial, ciudadEditorial.

IDIOMA: codigoldioma, es el identificador de la clase, descripcionIdioma.

ALMACEN: codigoAlmacen, es el identificador de la clase, direccionAlmacen.

PEDIDO: numPedido, es la llave primaria de la clase, fechaPedido, direccionEnvioPedido.

### 3.2.2 Generalizaciones y especializaciones

En el modelo EER la clase PERSONAS es una **generalización** de las subclases CLIENTE, VENDEDOR, TRADUCTOR y AUTOR, ya que tienen atributos comunes entre sí, al mismo tiempo se pueden clasificar a las antes mencionadas especializaciones de acuerdo al tipo de actividad que desempeñan en el estudio de caso y a su tipo de responsabilidad en la publicación, es decir:

*'Tipo Actividad'*: se encuentran las clases CLIENTE y VENDEDOR, debido a que una PERSONA es un VENDEDOR y también puede ser un CLIENTE, en el círculo que conecta a las subclases con su respectiva superclase se coloca la letra **o**, señalando de esta forma una **superposición** de subclases.

*'Responsable Publicación'*: se encuentran las clases `TRADUCTOR` y `AUTOR`, en este caso una `PERSONA` es un `AUTOR` o es un `TRADUCTOR`, por lo tanto en el círculo se coloca una **d**, indicando una **restricción de disyunción**.

Las clases `CLIENTE` y `VENDEDOR`, a su vez, se convierten en generalizaciones: `CLIENTE`, es la superclase de las subclases `MAYORISTA` y `MINORISTA` mientras que `VENDEDOR` es la superclase de `VENDEDORTIEMPOCOMPLETO` y `VENDEDORMEDIOTIEMPO`, de igual forma para estas nuevas generalizaciones también existen restricciones en otras palabras:

Para la clase `CLIENTE`, la clasificación que se le dará a sus subclases será por *'Tipo Negocio'*, en este caso un `CLIENTE` es un `MAYORISTA` o es un `MINORISTA`, por lo tanto existe una **restricción de disyunción** y se colocará la letra **d**. Para la clase `VENDEDOR`, la clasificación será *'Clase'*, en donde un `VENDEDOR` es un `VENDEDORMEDIOTIEMPO` o un `VENDEDORTIEMPOCOMPLETO`, por lo que también se coloca una **d** en el círculo que une estas subclases con su superclase.

Adicionalmente según lo detallado en el enunciado del estudio de caso **"distribuidora de libros"** se necesitan las clases `EDITORIAL`, `PEDIDO`, `ALMACEN`, `LIBRO` e `IDIOMA` con sus respectivos atributos.

### 3.2.3 Relaciones

En cuanto a las relaciones, existen en el modelo las siguientes:

*'ORDENA'*: entre las clases `CLIENTE` – `PEDIDO`, de cardinalidad 1:N, un cliente ordena varios pedidos y varios pedidos son ordenados por un cliente.

*'REALIZA'*: entre las clases `VENDEDOR` – `PEDIDO`, de cardinalidad 1:N, un vendedor realiza varios pedidos y varios pedidos son realizados por un vendedor.

*'ESCRIBE'*: entre las clases `LIBRO` – `AUTOR`, de cardinalidad N:N, varios libros son escritos por varios autores y viceversa.

*'TRABAJA'*: entre las clases `VENDEDOR` – `ALMACEN`, de cardinalidad N:1, varios vendedores trabajan en un almacén y en un almacén trabajan varios vendedores, cumpliendo de esta forma el requisito del enunciado en donde se especifica que no existe rotación de vendedores entre los diferentes almacenes de la distribuidora.

'EDITA': entre las clases EDITORIAL - LIBRO, de cardinalidad N:N, varios libros son editados por varias editoriales y viceversa.

'TRADUCCION': entre las clases LIBRO - IDIOMA, de cardinalidad N:N, varios libros tienen traducción a varios idiomas y viceversa.

Sumado a esto, existe una agregación de la relación 'TRADUCCION', la cual ayuda a mantener conocimiento sobre los idiomas en los que está traducido un libro, requisito que consta en el enunciado del estudio de caso. Esta agregación interviene en las siguientes relaciones:

'TRADUCE': entre la clase TRADUCTOR y la agregación LIBRO - IDIOMA, de cardinalidad N:N, esta relación permite conocer el idioma de la traducción y el traductor responsable por dicha traducción.

'DETALLA': entre las clases PEDIDO y la agregación LIBRO - IDIOMA, de cardinalidad N:N, el atributo de la relación cantidadDetalle, maneja información sobre la cantidad de libros por idioma en un pedido.

'TIENE': entre las clases ALMACEN y la agregación LIBRO - IDIOMA, de cardinalidad N:N, el atributo de la relación cantidadAlmacen, contiene información sobre la cantidad de libros por idioma en un almacén.

### 3.2.4 Resumen de entidades y relaciones.

En la Tabla 3.1 y Tabla 3.2 respectivamente, se muestran las entidades y relaciones definidas en el presente capítulo para el estudio de caso “**distribuidora de libros**”.

Entidad	Descripción	Atributos	Hereda	Llave primaria
PERSONAS	Persona que interviene en el estudio de caso.	codigoPersonas, nombrePersonas, apellidoPersonas, identificacionPersonas, direccionPersonas, telefonoPersonas.	-	codigoPersonas
AUTOR	Autor de un libro	nacionalidadAutor, tituloAutor	PERSONAS	codigoAutor
TRADUCTOR	Traduce los libros	nacionalidadTraductor	PERSONAS	codigoTraductor

VENDEDOR	Vendedor que trabaja en un almacén	comisión, fechaInicio	PERSONAS	codigoVendedor
CLIENTE	Cliente que compra los libros	descuentoCliente	PERSONAS	codigoPersonas
VENDEDORTIEMPOCOMPLETO	Vendedor que trabaja tiempo completo en un almacén	salario	VENDEDOR	codTiempoCompleto
VENDEDORMEDIOTIEMPO	Vendedor que trabaja por horas en un almacén	numHoras, sueldoHora	VENDEDOR	codMedioTiempo
MAYORISTA	Cliente que realiza pedidos por cantidades grandes	limiteCredito	CLIENTE	codMayorista
MINORISTA	Cliente que realiza pedidos por menor	mail	CLIENTE	codMinorista
LIBRO	Publicación vendida por la distribuidora	codigoLibro, tituloLibro, precioLibro, anioPublicacionLibro	-	codigoLibro
EDITORIAL	Edita los libros	codigoEditorial, nombreEditorial, telefonoEditorial, ciudadEditorial	-	codigoEditorial
IDIOMA	Idioma en el que puede estar escrito un libro	codigoldioma, descripcionIdioma	-	codigoldioma
ALMACEN	Lugar donde se venden los libros	codigoAlmacen, direccionAlmacen	-	codigoAlmacen
PEDIDO	Pedido realizado por los clientes para comprar los libros	numPedido, fechaPedido, direccionEnvioPedido, codVendedor, codCliente.	-	numPedido

Tabla 3. 1 Resumen de entidades en el estudio de caso "distribuidora de libros"

<b>Relación</b>	<b>Entidades Involucradas</b>	<b>Atributos</b>
ESCRIBE	AUTOR (N) ... (N) LIBRO	-
TRADUCCION	IDIOMA (N) ... (N) LIBRO	-
AgregaciónTraducción	TRADUCCION {IDIOMA (N) ... (N) LIBRO}	-
TRADUCE	TRADUCTOR (N) ... (N) AgregaciónTraducción	-
EDITA	EDITORIAL (N) ... LIBRO (N)	-
DETALLA	PEDIDO (N) ... (N) AgregaciónTraducción	cantidadDetalle
TIENE	ALMACEN (N) ... (N) AgregaciónTraducción	cantidadAlmacen
TRABAJA	VENDEDOR (N) ... (1) ALMACEN	-
REALIZA	VENDEDOR (1) ... (N) PEDIDO	-

Tabla 3. 2 Resumen de relaciones en el estudio de caso "distribuidora de libros"

### 3.3 Modelo EER

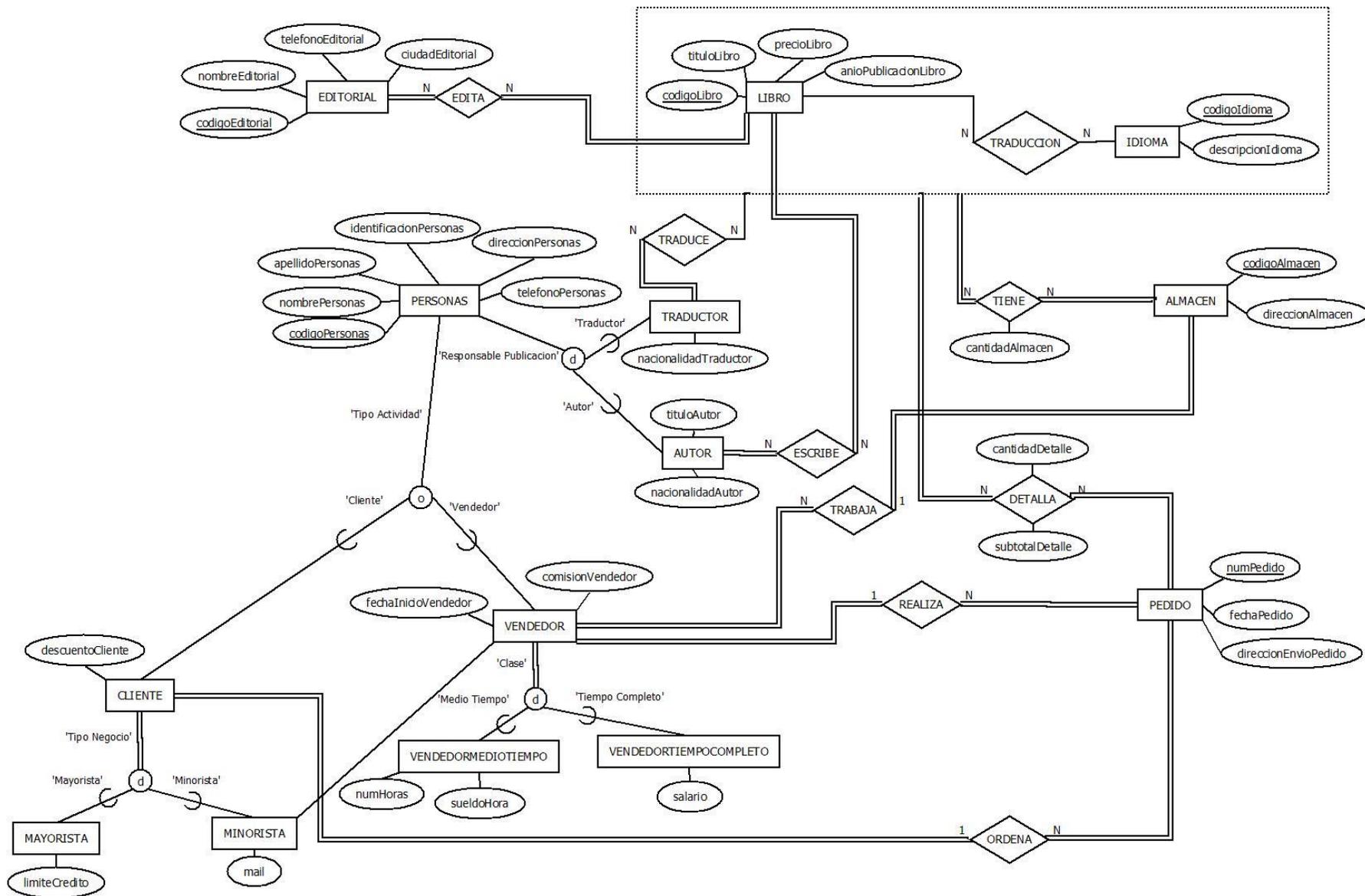


Figura 1.1 Diagrama EER para el caso del mundo real "distribuidora de libros".

### 3.4 Mapeo del modelo EER

Para la reducción a un esquema de base de datos relacional del estudio de caso “distribuidora de libros”, se ha optado por la Opción 1B: **Relaciones de superclases y subclases**, explicada en el capítulo II sección 2.6.1.1, se ha creado relaciones tanto para las subclases como para las superclases.

#### 3.4.1 Entidades

PERSONAS	codigoPersonas	identificacionPersonas	nombrePersonas	apellidoPersonas	telefonoPersonas	direccionPersonas
	Per001	1111111111	Ernest	Hemingway	0992515269	Av. Américas
	Per002	2222222222	Natalia	Jarrín	0985625896	San José
	Per003	3333333333	José	Alvarado	0985236652	Av. 1 de Mayo
	Per004	4444444444	Andrés	Cepeda	0896222463	Av. Loja
	Per005	5555555555	Víctor	Vintimilla	0978562322	Remigio Crespo
	Per006	6666666666	Lino	Novás Calvo	0984665893	Cuba
	Per007	7777777777	Esteban	Perez	0895555795	París

Tabla 3. 3 Reducción a tabla de la clase PERSONAS.

AUTOR	codigoAutor	nacionalidadAutor	tituloAutor
	Per001	Estadounidense	Lic. Comunicación

Tabla 3. 4 Reducción a tabla de la clase AUTOR.

TRADUCTOR	codigoTraductor	nacionalidadTraductor
	Per006	Cubano
	Per007	Francés

Tabla 3. 5 Reducción a tabla de la clase TRADUCTOR.

VENDEDOR	codigoVendedor	comision	fechaInicio	codigoAlmacenVendedor
	Per003	0.20	10/5/2015	Al001
	Per004	0.15	9/4/2000	Al001

Tabla 3. 6 Reducción a tabla de la clase VENDEDOR.

VENDEDORMEDIOTIEMPO	codMedioTiempo	numHoras	sueldoHora
	Per003	4	8.5

Tabla 3. 7 Reducción a tabla de la clase VENDEDORMEDIOTIEMPO.

VENDEDORTIEMPOCOMPLETO	codTiempoCompleto	Salario
	Per004	360

Tabla 3. 8 Reducción a tabla de la clase VENDEDORTIEMPOCOMPLETO.

CLIENTE	codigoCliente	descuentoCliente
	Per002	0.05
	Per005	0.00
	Per003	0.20

Tabla 3. 9 Reducción a tabla de la clase CLIENTE.

MINORISTA	codMinorista	mail
	Per005	p@hotmail.com
	Per003	n@hotmail.com

Tabla 3. 10 Reducción a tabla de la clase MINORISTA.

MAYORISTA	codMayorista	limiteCredito
	Per002	5000

Tabla 3. 11 Reducción a tabla de la clase MAYORISTA.

LIBRO	codigoLibro	tituloLibro	precioLibro	anioPublicacionLibro
	Lib001	El viejo y el mar	5.5	1952
	Lib002	Crepúsculo	10.2	2002

Tabla 3. 12 Reducción a tabla de la clase LIBRO.

EDITORIAL	codigoEditorial	nombreEditorial	telefonoEditorial	ciudadEditorial
	Edi001	Delbolsillo	4125963	España
	Edi002	Planeta	8596236	España

Tabla 3. 13 Reducción a tabla de la clase EDITORIAL.

IDIOMA	codigoldioma	descripcionIdioma
	Id001	Español
	Id002	Inglés
	Id003	Francés
	Id004	Alemán

Tabla 3. 14 Reducción a tabla de la clase IDIOMA.

ALMACEN	codigoAlmacen	direccionAlmacen
	Al001	Av. Loja
	Al002	Av. 1 de Mayo

Tabla 3. 15 Reducción a tabla de la clase ALMACEN.

PEDIDO	numPedido	fechaPedido	direccionEnvioPedido	codVendedor	codCliente
	Ped001	10/5/2016	Quito	Per004	Per002
	Ped002	10/6/2016	Cuenca	Per004	Per005
	Ped003	10/4/2016	Quito	Per004	Per003

Tabla 3. 16 Reducción a tabla de la clase PEDIDO.

### 3.4.2 Relaciones.

ESCRIBE	codAutor	codLibro
	Per001	Lib001
	Per001	Lib002

Tabla 3. 17 Reducción a tabla de la relación ESCRIBE.

TRADUCCION	codIdioma	codLibr
	Id001	Lib001
	Id003	Lib001
	Id002	Lib001
	Id003	Lib002

Tabla 3. 18 Reducción a tabla de la relación TRADUCCION.

TRADUCE	codTrad	codTraduceLibro	codTraduceldioma
	Per006	Lib001	Id001
	Per007	Lib001	Id003
	Per007	Lib001	Id002

Tabla 3. 19 Reducción a tabla de la relación TRADUCE.

EDITA	codEditorial	codEditaLibro
	Ed001	Lib001
	Ed002	Lib001
	Ed002	Lib002

Tabla 3. 20 Reducción a tabla de la relación EDITA.

DETALLA	numPed	codigoPedidoLibro	codIdm	cantidadDetalle
	Ped001	Lib001	Id001	200
	Ped002	Lib001	Id002	5
	Ped003	Lib001	Id003	20
	Ped003	Lib001	Id002	30

Tabla 3. 21 Reducción a tabla de la relación DETALLE.

TIENE	codAlmacen	codLibroPorAlmacen	codldmPorAlmacen	cantidadAlmacen
	Al001	Lib001	ld001	1000
	Al001	Lib002	ld002	200
	Al001	Lib001	ld002	20

Tabla 3. 22 Reducción a tabla de la relación TIENE.

## CAPÍTULO IV: IMPLEMENTACIÓN EN EL GESTOR DE BASES DE DATOS

### 4.1 Introducción

En este capítulo constan los diccionarios de datos de las tablas del modelo EER del estudio de caso “**distribuidora de libros**”, las mismas que fueron desarrolladas en el capítulo III del documento, estos diccionarios se utilizan como referente para crear las tablas en el SGBD. Finalmente se presentan los resultados de la implementación en JAVA y las conclusiones sobre el proyecto de investigación del modelo EER.

### 4.2 Diccionario de datos

#### 4.2.1 Entidades

PERSONAS	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codigoPersonas	Identificador de la persona	VARCHAR()	6	SI	NO	-	Dato alfanumérico único de 6 dígitos, formato Per000	-
	identificacionPersonas	Número de cédula, identificación o ruc según sea el caso	VARCHAR()	20	NO	NO	-	Dato numérico único	-
	nombrePersonas	Nombre de la persona ingresada al sistema	VARCHAR()	30	NO	NO	-	No se ingresan números	NULL
	apellidoPersonas	Apellido de la persona ingresada al sistema	VARCHAR()	30	NO	NO	-	No se ingresan números	NULL
	direccionPersonas	Dirección de domicilio de la persona	VARCHAR()	30	NO	NO	-	Dato alfanumérico	NULL
	telefonoPersonas	Número de contacto	VARCHAR()	10	NO	NO	-	Se ingresan solo números	NULL

Tabla 4. 1 Diccionario de datos de la tabla PERSONAS.

AUTOR	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codigoAutor	Identificador del autor	VARCHAR()	6	SI	SI	PERSONAS	Dato alfanumérico único de 6 dígitos, formato Per000	-
	nacionalidadAutor	Nacionalidad de origen del autor	VARCHAR()	20	NO	NO	-	No se ingresan números	NULL
	tituloAutor	Título de estudios del autor	VARCHAR()	20	NO	NO	-	No se ingresan números	NULL

Tabla 4. 2 Diccionario de datos de la tabla AUTOR.

TRADUCTOR	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codigoTraductor	Identificador del traductor	VARCHAR()	6	SI	SI	PERSONAS	Dato alfanumérico único de 6 dígitos, formato Per000	-
	nacionalidadTraductor	Nacionalidad de origen del traductor	VARCHAR()	20	NO	NO	-	No se ingresan números	NULL

Tabla 4. 3 Diccionario de datos de la tabla TRADUCTOR.

VENDEDOR	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codigoVendedor	Identificador del vendedor	VARCHAR()	6	SI	SI	PERSONAS	Dato alfanumérico único de 6 dígitos, formato Per000	-
	comision	Comisión que gana el vendedor	DOUBLE	-	NO	NO	-	Solo deben ingresarse datos numéricos	NULL
	fechaInicio	Fecha en la que comienza a laborar el vendedor en la distribuidora	VARCHAR()	10	NO	NO	-	Formato tipo fecha dd/mm/aaa	NULL
	codigoAlmacenVendedor	Código del almacén en donde trabaja el vendedor	VARCHAR()	5	NO	SI	ALMACEN	Dato alfanumérico único de 5 dígitos, formato AL000	-

Tabla 4. 4 Diccionario de datos de la tabla VENDEDOR.

<b>VENDEDOR MEDIO TIEMPO</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>LLP</b>	<b>LLF</b>	<b>Referencia</b>	<b>Dominio</b>	<b>Default</b>
	codMedioTiempo	Identificador del vendedor a medio tiempo	VARCHAR()	6	SI	SI	VENDEDOR	Dato alfanumérico único de 6 dígitos, formato Per000	-
	numHoras	Número de horas que trabaja el vendedor en el almacén	INT	11	NO	NO	-	Solo deben ingresarse datos numéricos	NULL
	sueldoHora	Valor monetario que gana el vendedor por hora	DOUBLE	-	NO	NO	-	Solo deben ingresarse datos numéricos	NULL

Tabla 4. 5 Diccionario de datos de la tabla **VENDEDORMEDIOTIEMPO**.

<b>VENDEDOR TIEMPO COMPLETO</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>LLP</b>	<b>LLF</b>	<b>Referencia</b>	<b>Dominio</b>	<b>Default</b>
	codTiempoCompleto	Identificador del vendedor a tiempo completo.	VARCHAR()	6	SI	SI	VENDEDOR	Dato alfanumérico único de 6 dígitos, formato Per000	-
	salario	Valor monetario que gana el vendedor por mes	DOUBLE	-	NO	NO	-	Solo deben ingresarse datos numéricos	NULL

Tabla 4. 6 Diccionario de datos de la tabla **VENDEDORTIEMPOCOMPLETO**.

<b>CLIENTE</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>LLP</b>	<b>LLF</b>	<b>Referencia</b>	<b>Dominio</b>	<b>Default</b>
	codigoCliente	Identificador del cliente	VARCHAR()	6	SI	SI	PERSONAS	Dato alfanumérico único de 6 dígitos, formato Per000	-
	descuentoCliente	Porcentaje de descuento que tiene el cliente en sus compras	DOUBLE	-	NO	NO	-	Solo deben ingresarse datos numéricos	NULL

Tabla 4. 7 Diccionario de datos de la tabla **CLIENTE**.

MINORISTA	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codMinorista	Identificador del comprador minorista	VARCHAR()	6	SI	SI	CLIENTE	Dato alfanumérico único de 6 dígitos, formato Per000	-
	mail	Correo electrónico de contacto del cliente	VARCHAR()	20	NO	NO	-	El correo debe ser ingresado con el símbolo @	NULL

Tabla 4. 8 Diccionario de datos de la tabla MINORISTA.

MAYORISTA	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codMayorista	Identificador del comprador mayorista	VARCHAR()	6	SI	SI	CLIENTE	Dato alfanumérico único de 6 dígitos, formato Per000	-
	limiteCredito	Valor monetario límite al cual accede el cliente mayorista	DOUBLE	-	NO	NO	-	Se deben ingresar solo valores numéricos	NULL

Tabla 4. 9 Diccionario de datos de la tabla MAYORISTA.

LIBRO	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codigoLibro	Identificador del libro	VARCHAR()	6	SI	NO	-	Dato alfanumérico único de 6 dígitos, formatoLib000	-
	tituloLibro	Nombre de la publicación	VARCHAR()	30	NO	NO	-	Se pueden ingresar números y letras	NULL
	precioLibro	Valor monetario al cual se vende la publicación	DOUBLE	-	NO	NO	-	Solo deben ingresarse datos numéricos	NULL
	anioPublicacionLibro	Año en el que se publicó el libro	INT	11	NO	NO	-	Solo deben ingresarse datos numéricos	NULL

Tabla 4. 10 Diccionario de datos de la tabla LIBRO.

EDITORIAL	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codigoEditorial	Identificador de la editorial	VARCHAR()	6	SI	NO	-	Dato alfanumérico único de 6 dígitos, formato Edi000	-
	nombreEditorial	Nombre de la editorial	VARCHAR()	20	NO	NO	-	Se pueden ingresar números y letras	NULL
	telefonoEditorial	Número de contacto	VARCHAR()	10	NO	NO	-	Solo deben ingresarse datos numéricos	NULL
	ciudadEditorial	Ciudad en donde se encuentra la editorial	VARCHAR()	10	NO	NO	-	Solo deben ingresarse letras	NULL

Tabla 4. 11 Diccionario de datos de la tabla EDITORIAL.

IDIOMA	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codigoldioma	Identificador del idioma	VARCHAR()	5	SI	NO	-	Dato alfanumérico único de 5 dígitos, formato Id000	-
	descripcionIdioma	Nombre del idioma	VARCHAR()	20	NO	NO	-	Solo se ingresan letras	NULL

Tabla 4. 12 Diccionario de datos de la tabla IDIOMA.

ALMACEN	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codigoAlmacen	Identificador del almacén en donde trabaja el vendedor	VARCHAR()	5	SI	NO	-	Dato alfanumérico único de 5 dígitos, formato AL000	-
	direccionAlmacen	Dirección donde se localiza el almacén	VARCHAR()	30	NO	NO	-	Se pueden ingresar letras y números	NULL

Tabla 4. 13 Diccionario de datos de la tabla ALMACEN.

PEDIDO	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	numPedido	Identificador del pedido	VARCHAR()	6	SI	NO	-	Dato alfanumérico único de 6 dígitos, formato Ped000	-
	fechaPedido	Fecha en la que se realizó el pedido	VARCHAR()	10	NO	NO	-	Formato tipo fecha dd/mm/aaa	NULL
	direccionEnvioPedido	Dirección de domicilio de la persona	VARCHAR()	30	NO	NO	-	Dato alfanumérico	NULL
	codVendedor	Identificador del vendedor	VARCHAR()	6	NO	SI	VENDEDOR	Dato alfanumérico único de 6 dígitos, formato Per000	-
	codCliente	Identificador del cliente	VARCHAR()	6	NO	SI	CLIENTE	Dato alfanumérico único de 6 dígitos, formato Per000	-

Tabla 4. 14 Diccionario de datos de la tabla PEDIDO.

#### 4.2.2 Relaciones.

ESCRIBE	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codAutor	Identificador del autor	VARCHAR()	6	SI	SI	AUTOR	Dato alfanumérico único de 6 dígitos, formato Per000	-
	codLibro	Identificador del libro	VARCHAR()	6	SI	SI	LIBRO	Dato alfanumérico único de 6 dígitos, formatoLib000	-

Tabla 4. 15 Diccionario de datos de la tabla ESCRIBE.

TRADUCCION	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codIdioma	Identificador del idioma	VARCHAR()	5	SI	SI	IDIOMA	Dato alfanumérico único de 5 dígitos, formato Id000	-
	codLibr	Identificador del libro	VARCHAR()	6	SI	SI	LIBRO	Dato alfanumérico único de 6 dígitos, formatoLib000	-

Tabla 4. 16 Diccionario de datos de la tabla TRADUCCION.

TRADUCE	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codTrad	Identificador del traductor	VARCHAR()	6	SI	SI	TRADUCTOR	Dato alfanumérico único de 6 dígitos, formato Per000	-
	codTraduceLibro	Identificador del libro	VARCHAR()	6	SI	SI	LIBRO	Dato alfanumérico único de 6 dígitos, formatoLib000	-
	codTraduceIdioma	Identificador del idioma	VARCHAR()	5	SI	SI	IDIOMA	Dato alfanumérico único de 5 dígitos, formato Id000	-

Tabla 4. 17 Diccionario de datos de la tabla TRADUCE.

EDITA	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codEditorial	Identificador de la editorial	VARCHAR()	6	SI	SI	EDITORIAL	Dato alfanumérico único de 6 dígitos, formato Edi000	-
	codEditaLibro	Identificador del libro	VARCHAR()	6	SI	SI	LIBRO	Dato alfanumérico único de 6 dígitos, formatoLib000	-

Tabla 4. 18 Diccionario de datos de la tabla EDITA.

DETALLA	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	numPed	Identificador del pedido	VARCHAR()	6	NO	SI	PEDIDO	Dato alfanumérico único de 6 dígitos, formato Ped000	-
	codigoPedidoLibro	Identificador del libro	VARCHAR()	6	NO	SI	LIBRO	Dato alfanumérico único de 6 dígitos, formatoLib000	-
	codldm	Identificador del idioma	VARCHAR()	5	NO	SI	IDIOMA	Dato alfanumérico único de 5 dígitos, formato Id000	-
	cantidadDetalle	cantidad de libros por idioma en un pedido	INT	11	NO	NO	-	Solo se ingresan números	-

Tabla 4. 19 Diccionario de datos de la tabla DETALLE.

TIENE	Atributo	Descripción	Tipo	Tamaño	LLP	LLF	Referencia	Dominio	Default
	codAlmacen	Identificador del almacén en donde trabaja el vendedor	VARCHAR()	5	NO	SI	ALMACEN	Dato alfanumérico único de 5 dígitos, formato AL000	-
	codLibroPorAlmacen	Identificador del libro	VARCHAR()	6	NO	SI	LIBRO	Dato alfanumérico único de 6 dígitos, formatoLib000	-
	codldmPorAlmacen	Identificador del idioma	VARCHAR()	5	NO	SI	IDIOMA	Dato alfanumérico único de 5 dígitos, formato Id000	-
	cantidadAlmacen	cantidad de libros por idioma en un almacén	INT	11	NO	NO	-	Solo se ingresan números	-

Tabla 4. 20 Diccionario de datos de la tabla TIENE.

### 4.3 Resultados

El script de creación de la base de datos del estudio de caso “**distribuidora de libros**” se encuentra en el Anexo A, a continuación se describen algunas tareas que se pueden llevar a cabo en el programa del estudio de caso, el cual fue

desarrollado con el lenguaje de programación JAVA, mediante la herramienta Netbeans IDE 8.2.

### 4.3.1 Descripción

En la ventana principal del programa del estudio de caso “**distribuidora de libros**”, constan los menús: mantenimiento, administrar libros, consultas y pedidos. (ver Figura 4.1).

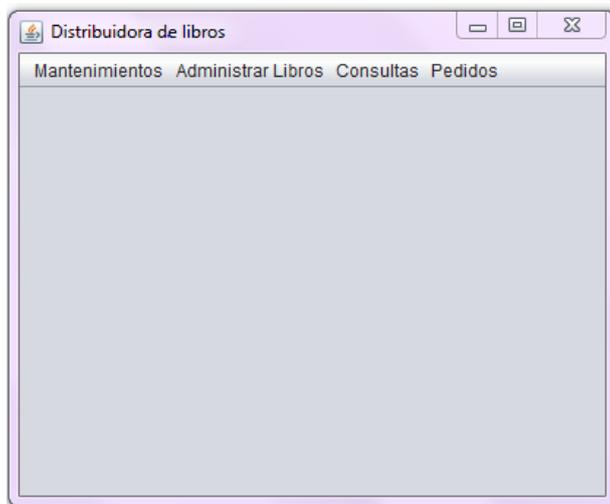


Figura 4. 1 Ventana principal.

#### 4.3.1.1 Menú Mantenimientos

En el menú “Mantenimientos” se realizan los ingresos, modificaciones, eliminaciones y búsquedas de almacenes, autores, clientes, tanto mayoristas como minoristas, editoriales, idiomas, libros, traductores y vendedores, los de medio tiempo o los de tiempo completo.

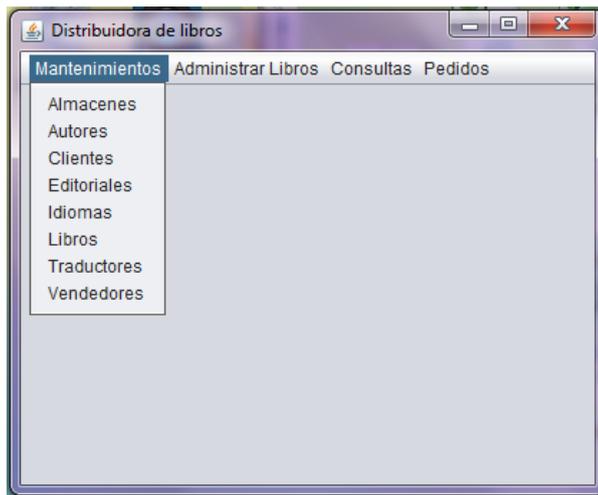


Figura 4. 2 Menú "Mantenimientos".

### Mantenimiento almacenes.

En la ventana de mantenimiento de almacenes se realiza el ingreso, modificación, eliminación, listar y consultas.

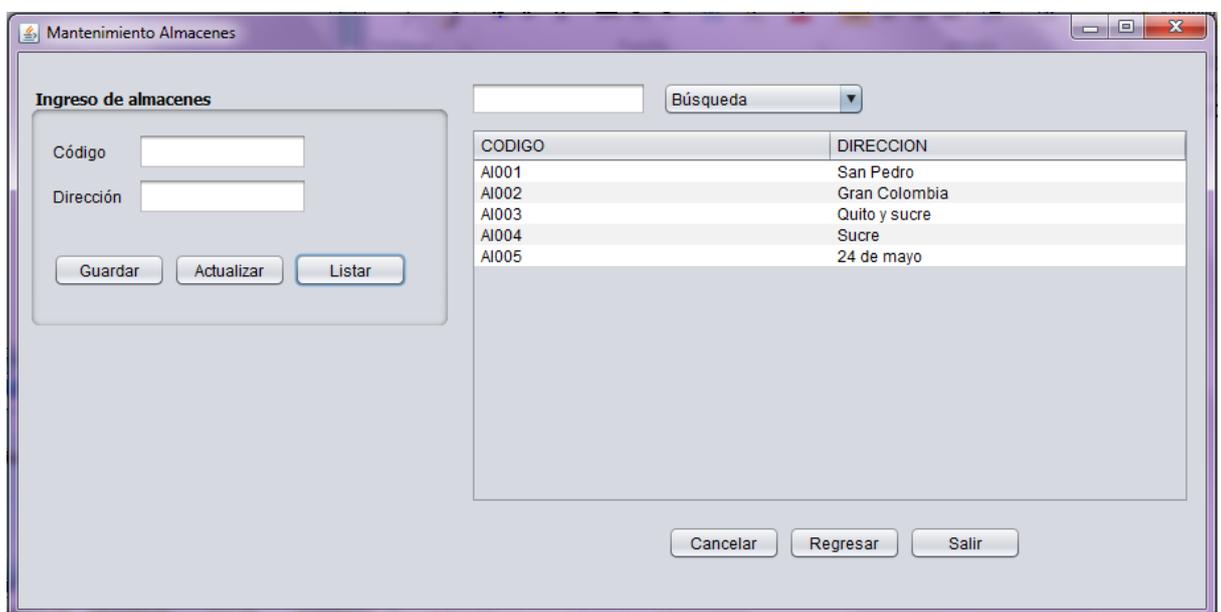


Figura 4. 3 Ventana del mantenimiento de almacenes.

Se pueden realizar dos tipos de búsqueda:

- Búsqueda por código:

Sentencia sql:

```
"SELECT *  
FROM almacen  
WHERE "+ "codigoAlmacen = '"+valor+"";
```

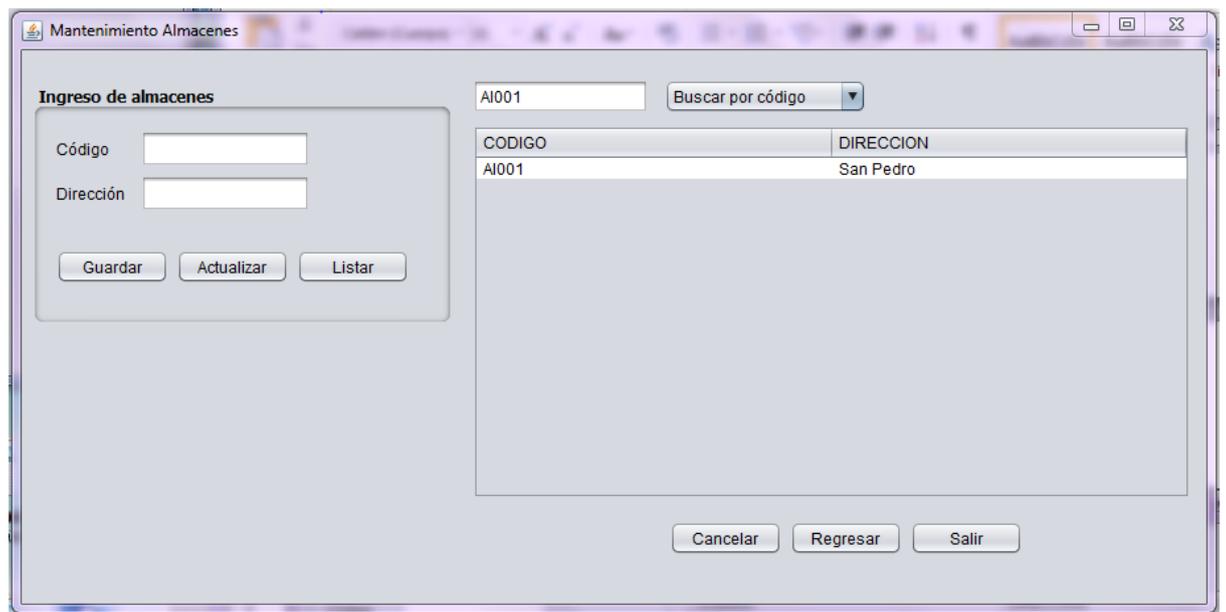


Figura 4. 4 Resultado de la búsqueda por código.

- Búsqueda por dirección

Sentencia sql:

```
"SELECT *  
FROM almacen  
WHERE "+ "direccionAlmacen LIKE '"+valor+"%";
```

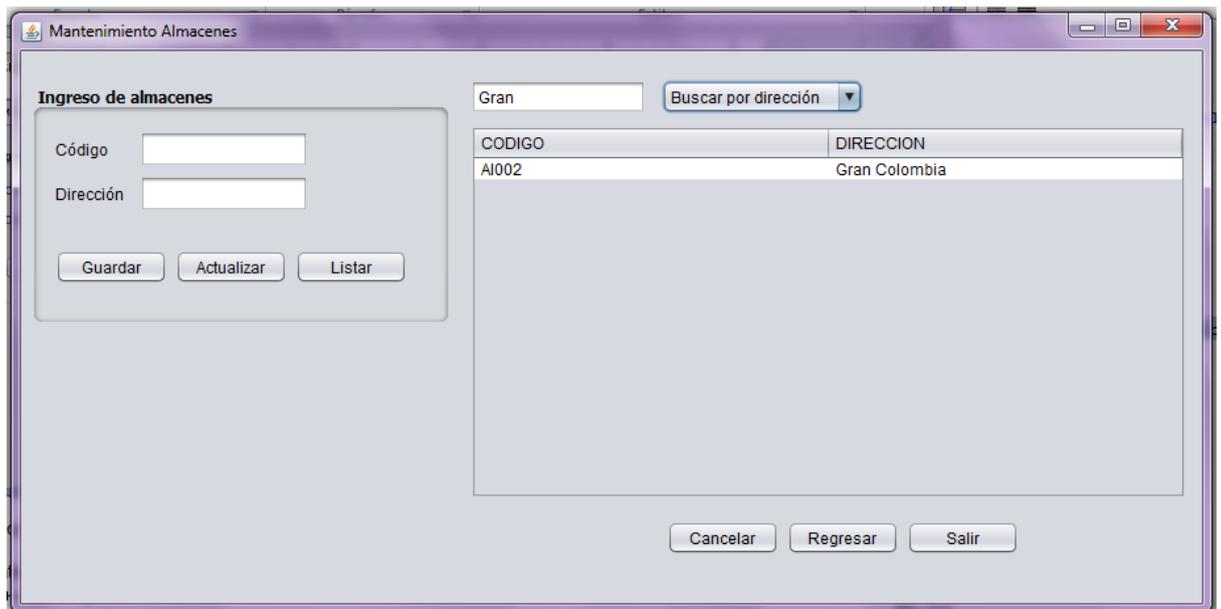


Figura 4. 5 Resultado de la búsqueda por dirección.

## Mantenimiento autores

En la ventana de mantenimiento de autores se realiza el ingreso, modificación, eliminación, listar y consultas.

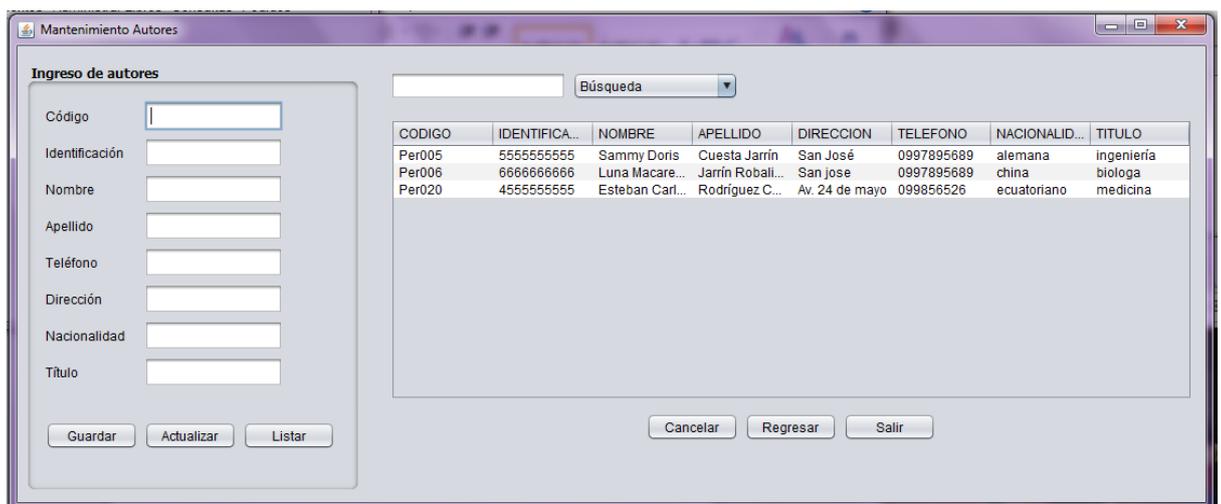


Figura 4. 6 Ventana del mantenimiento de autores.

Se pueden realizar dos tipos de búsqueda:

- Búsqueda por código

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
      nombrePersonas, apellidoPersonas,
      direccionPersonas, telefonoPersonas,
      nacionalidadAutor, tituloAutor
FROM personas, autor
WHERE codigoPersonas = codigoAutor
      and "+ "codigoPersonas = '"+valor+"'";
```

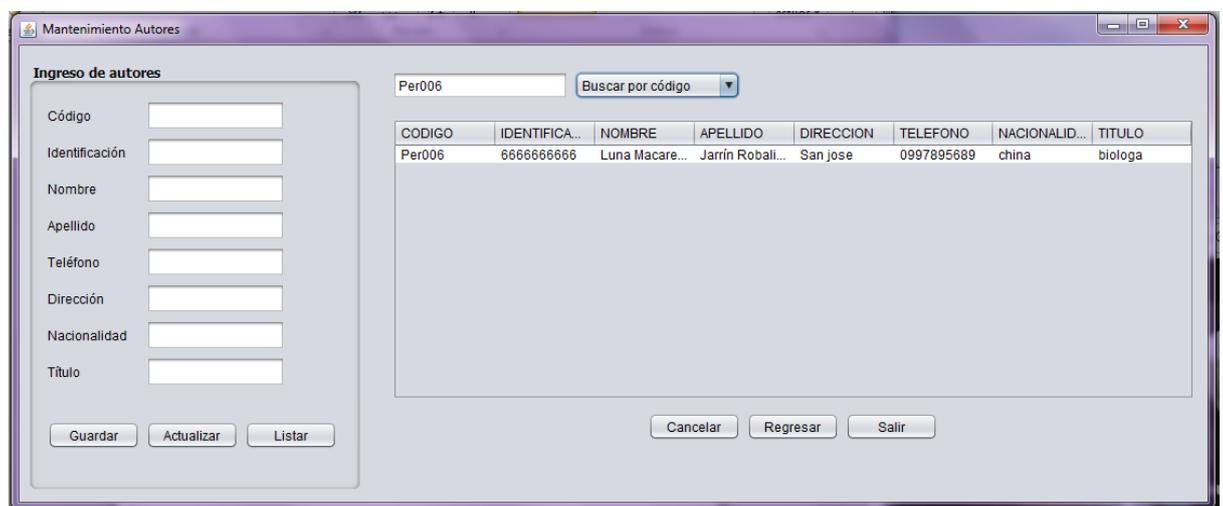


Figura 4. 7 Resultado de la búsqueda por código.

- Búsqueda por apellido

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
      nombrePersonas, apellidoPersonas,
      direccionPersonas, telefonoPersonas,
      nacionalidadAutor, tituloAutor
FROM personas, autor
WHERE codigoPersonas = codigoAutor
      and "+ "apellidoPersonas LIKE '"+valor+"%'";
```

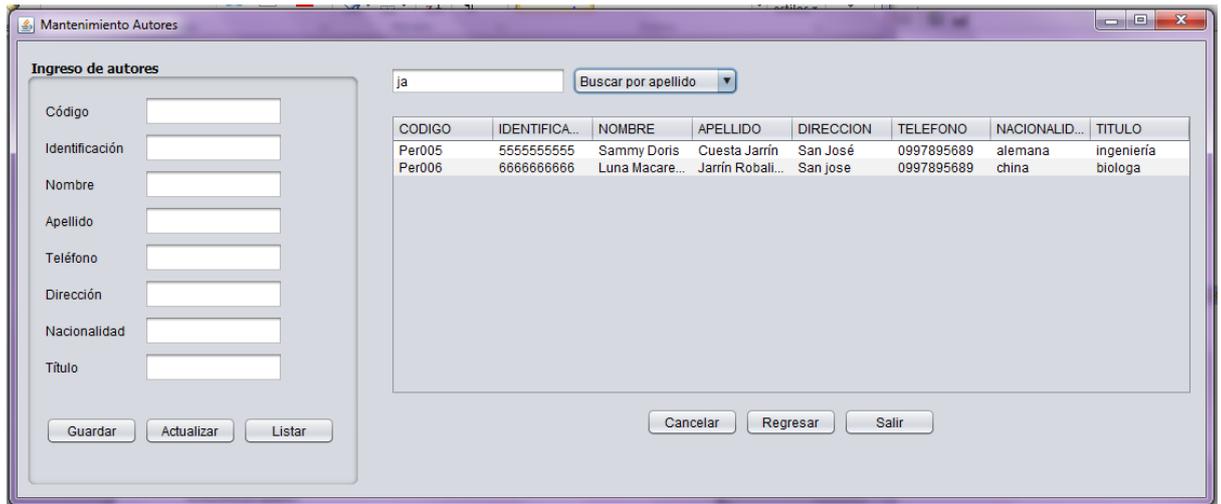


Figura 4. 8 Resultado de la búsqueda por apellido

## Mantenimiento clientes

En la ventana de mantenimiento de clientes se realiza el ingreso, modificación, eliminación, listar y consultas.

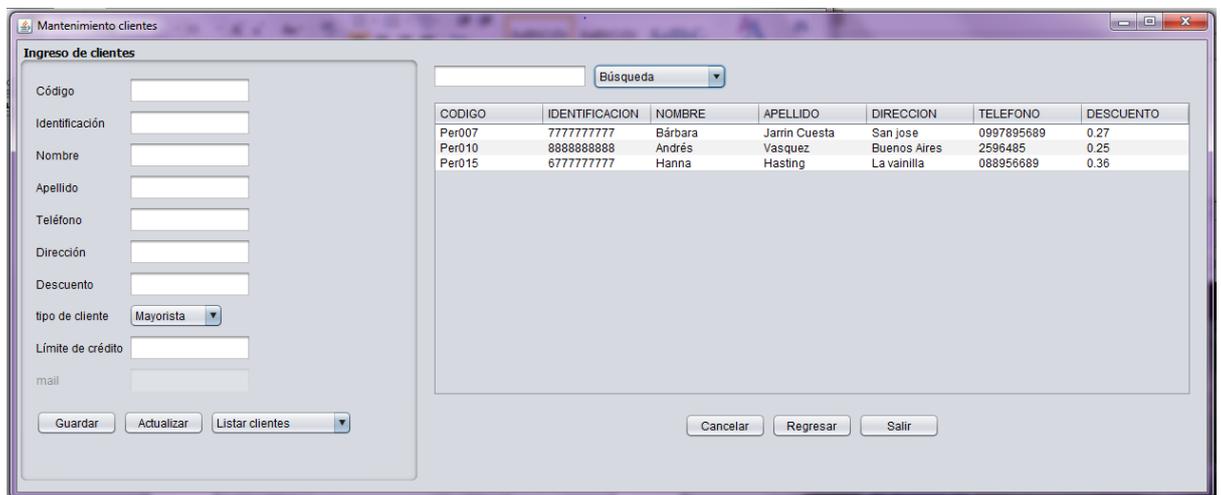


Figura 4. 9 Ventana del mantenimiento de clientes.

Se pueden realizar dos tipos de búsqueda y de listado:

- Búsqueda por código

Sentencia sql:

```

"SELECT codigoPersonas, identificacionPersonas,
        nombrePersonas, apellidoPersonas,
        direccionPersonas, telefonoPersonas,
        descuentoCliente
FROM personas, cliente
WHERE codigoPersonas = codigoCliente
      and "+ "codigoPersonas = '"+valor+"'";

```

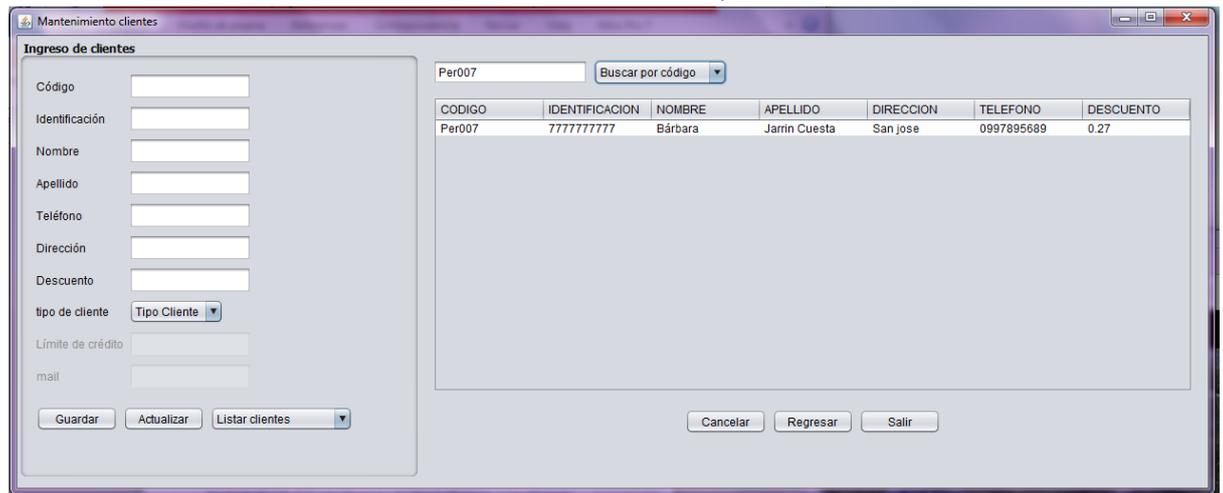


Figura 4. 10 Resultado de la búsqueda por código.

- Búsqueda por apellido

Sentencia sql:

```

"SELECT codigoPersonas, identificacionPersonas,
        nombrePersonas, apellidoPersonas,
        direccionPersonas, telefonoPersonas,
        descuentoCliente
FROM personas, cliente
WHERE codigoPersonas = codigoCliente
      and "+ "apellidoPersonas LIKE '"+valor+"%'";

```

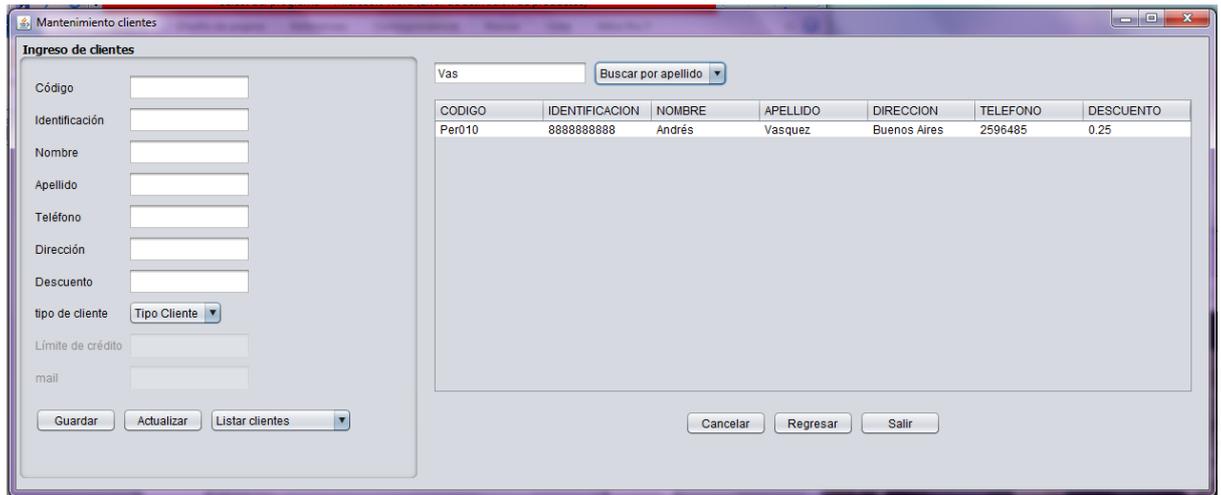


Figura 4. 11 Resultado de la búsqueda por apellido.

- Listar Clientes mayoristas

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
      nombrePersonas, apellidoPersonas,
      direccionPersonas, telefonoPersonas,
      descuentoCliente, limiteCredito
FROM personas, cliente, mayorista
WHERE codigoPersonas = codigoCliente
      and codigoPersonas = codMayorista";
```

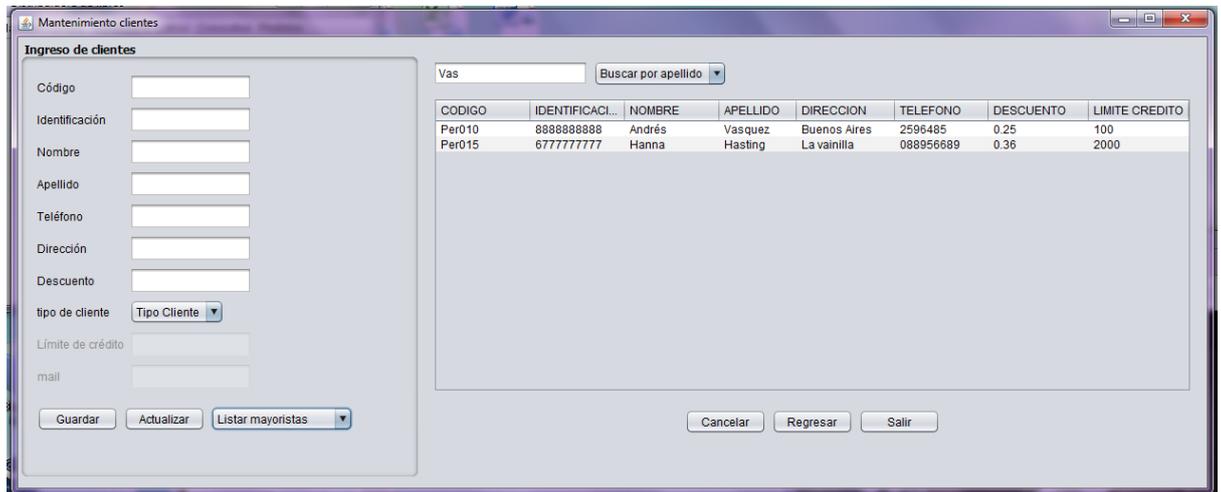


Figura 4. 12 Listado de clientes mayoristas.

- Listar clientes minoristas

Senetncia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
```

```

nombrePersonas, apellidoPersonas,
direccionPersonas, telefonoPersonas,
descuentoCliente, mail
FROM personas, cliente, minorista
WHERE codigoPersonas = codigoCliente
and codigoPersonas = codMinorista";

```

The screenshot shows a software window titled "Mantenimiento clientes". On the left, there is a form titled "Ingreso de clientes" with fields for: Código, Identificación, Nombre, Apellido, Teléfono, Dirección, Descuento, tipo de cliente (set to "Tipo Cliente"), Límite de crédito, and mail. Below the form are buttons for "Guardar", "Actualizar", and "Listar minoristas". On the right, there is a search bar with "Vas" and a dropdown "Buscar por apellido". Below that is a table with the following data:

CODIGO	IDENTIFICACI...	NOMBRE	APELLIDO	DIRECCION	TELEFONO	DESCUENTO	MAIL
Per007	7777777777	Bárbara	Jarrin Cuesta	San jose	0997895689	0.27	remibu@hot

At the bottom right of the window are buttons for "Cancelar", "Regresar", and "Salir".

Figura 4. 13 Listado clientes minoristas.

## Mantenimiento editoriales

En la ventana de mantenimiento de editoriales se realiza el ingreso, modificación, eliminación, listar y consultas.

The screenshot shows a software window titled "Mantenimiento editoriales". On the left, there is a form titled "Ingreso de editoriales" with fields for: Código, Nombre, Teléfono, and Ciudad. Below the form are buttons for "Guardar", "Actualizar", and "Listar". On the right, there is a search bar with a dropdown "Búsqueda". Below that is a table with the following data:

CODIGO	NOMBRE	TELEFONO	CIUDAD
Edi001	Aire ediciones	2897856	Madrid
Edi002	Marte ediciones	0998563696	Madrid

At the bottom right of the window are buttons for "Cancelar", "Regresar", and "Salir".

Figura 4. 14 Ventana del mantenimiento de editoriales.

Se pueden realizar dos tipos de búsqueda:

- Búsqueda por código

Sentencia sql:

```
"SELECT *  
FROM editorial  
WHERE "+ "codigoEditorial = '"+valor+'";
```

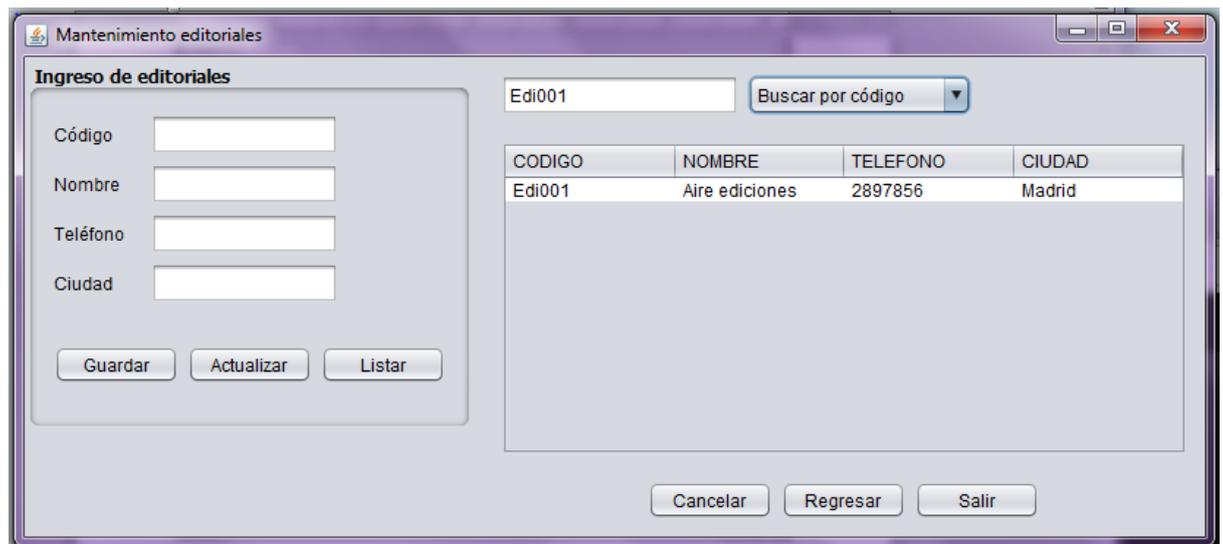


Figura 4. 15 Resultado de la búsqueda por código.

- Búsqueda por nombre de editorial

Sentencia sql:

```
"SELECT *  
FROM editorial  
WHERE "+ "nombreEditorial LIKE '%" + valor + "%'";
```

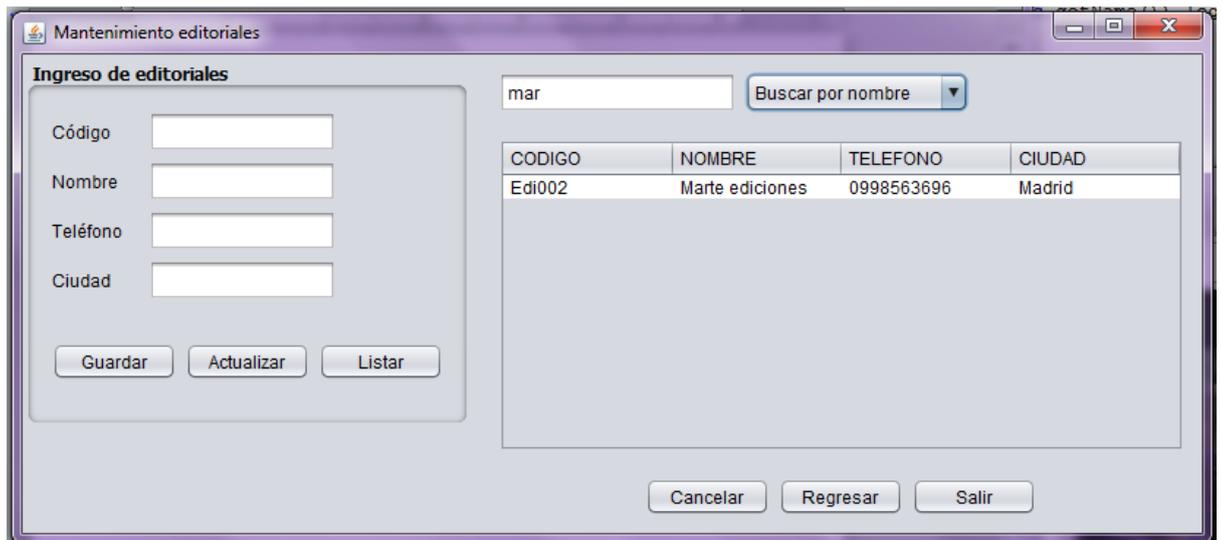


Figura 4. 16 Resultado de la búsqueda por nombre.

## Mantenimiento idiomas

En la ventana de mantenimiento de idiomas se realiza el ingreso, modificación, eliminación, listar y consultas.

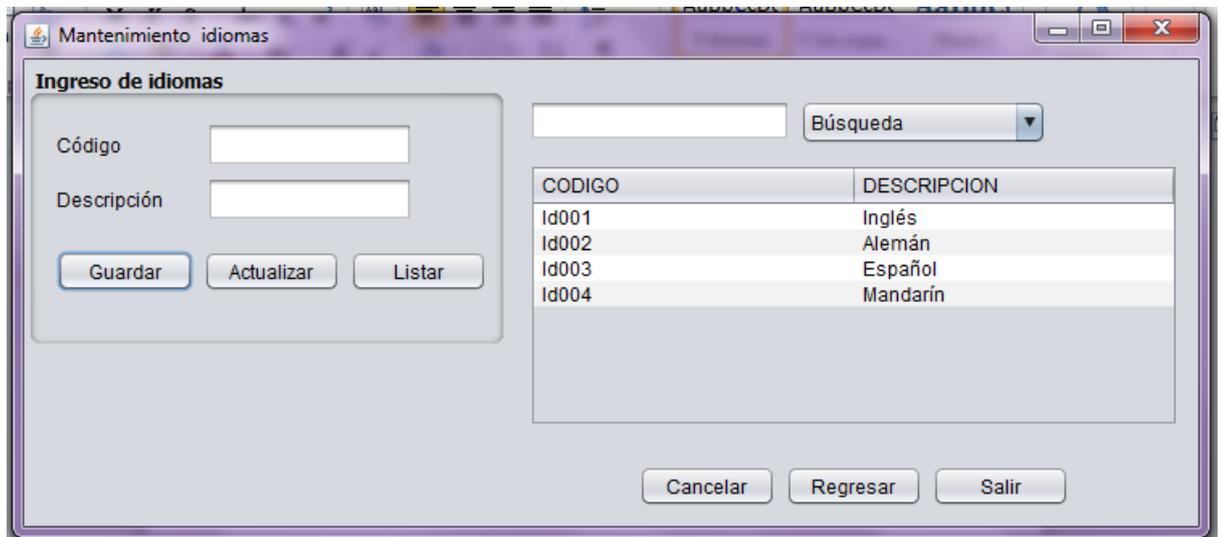


Figura 4. 17 Ventana del mantenimiento de idiomas.

Se pueden realizar dos tipos de búsqueda:

- Búsqueda por código

Sentencia sql:

```
"SELECT *  
FROM idioma  
WHERE "+ "codigoIdioma = '"+valor+"'";
```

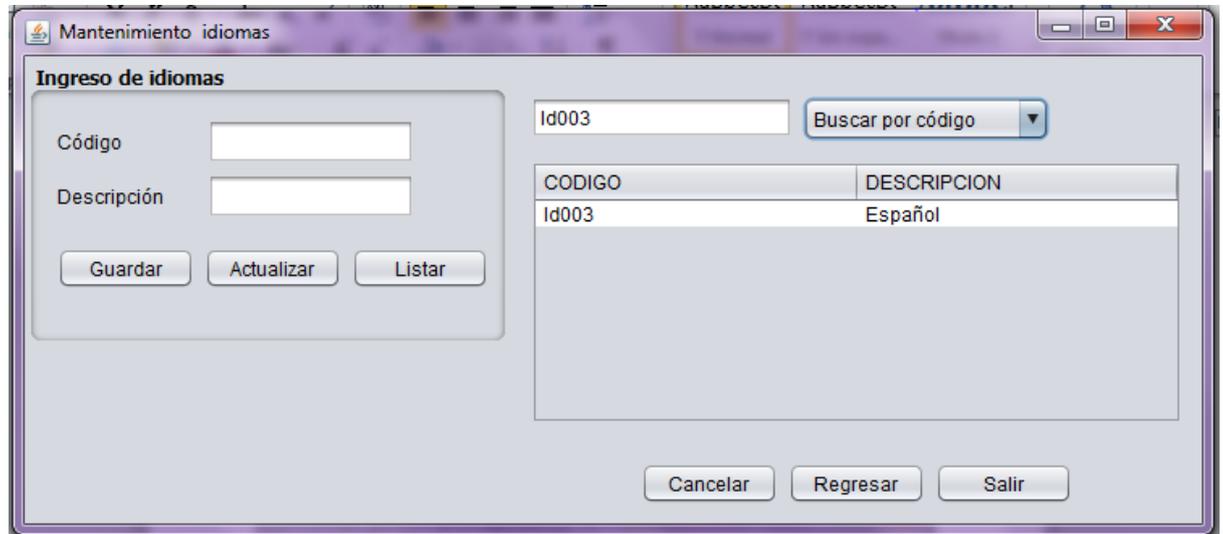


Figura 4. 18 Resultado de la búsqueda por código.

- Búsqueda por descripción de idioma

Sentencia sql:

```
"SELECT *  
FROM idioma  
WHERE "+ "descripcionIdioma LIKE '"+valor+"%'";
```

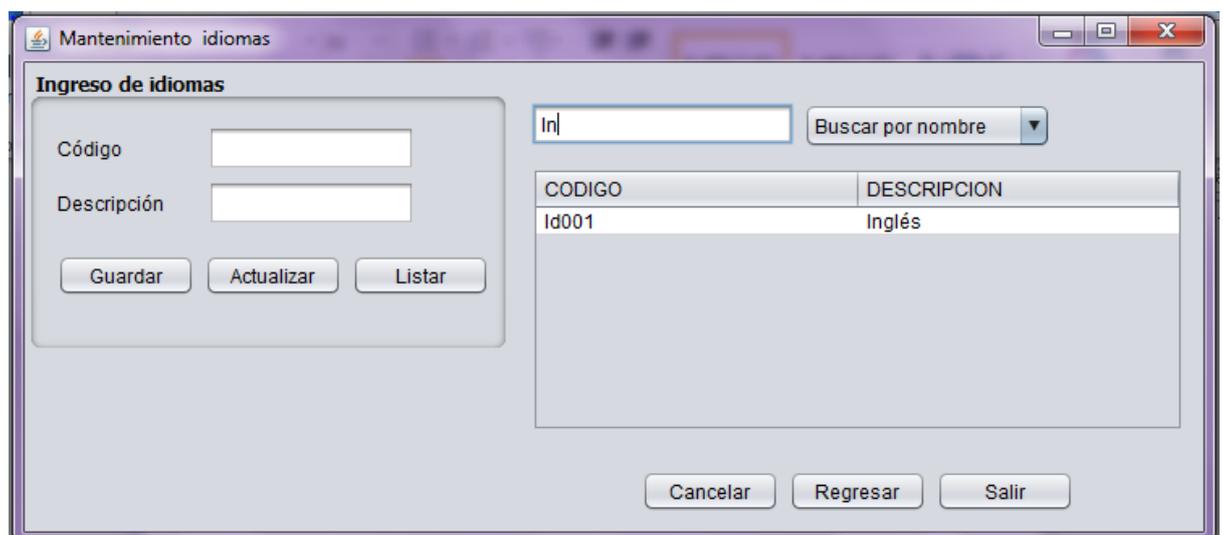
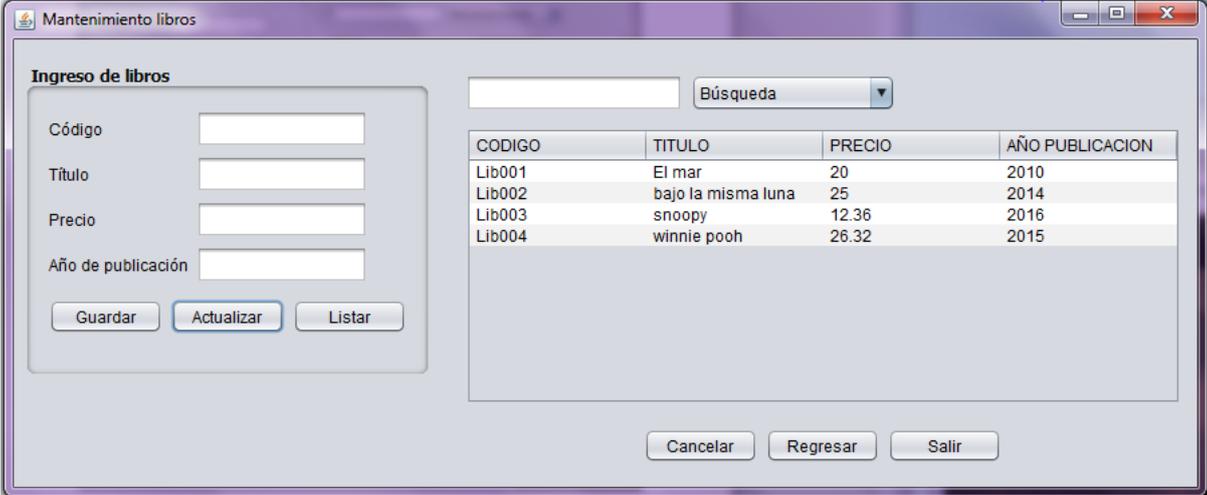


Figura 4. 19 Resultado de la búsqueda por descripción.

## Mantenimiento libros

En la ventana de mantenimiento de libros se realiza el ingreso, modificación, eliminación, listar y consultas.



The screenshot shows a window titled "Mantenimiento libros". On the left, there is a section titled "Ingreso de libros" with four input fields: "Código", "Título", "Precio", and "Año de publicación". Below these fields are three buttons: "Guardar", "Actualizar", and "Listar". On the right, there is a search bar with a "Búsqueda" dropdown menu. Below the search bar is a table with the following data:

CODIGO	TITULO	PRECIO	AÑO PUBLICACION
Lib001	El mar	20	2010
Lib002	bajo la misma luna	25	2014
Lib003	snoopy	12.36	2016
Lib004	winnie pooh	26.32	2015

At the bottom of the window, there are three buttons: "Cancelar", "Regresar", and "Salir".

Figura 4. 20 Ventana del mantenimiento de libros.

Se pueden realizar dos tipos de búsqueda:

- Búsqueda por código

Sentencia sql:

```
"SELECT *  
FROM libro  
WHERE "+ "codigoLibro = '"+valor+"";
```

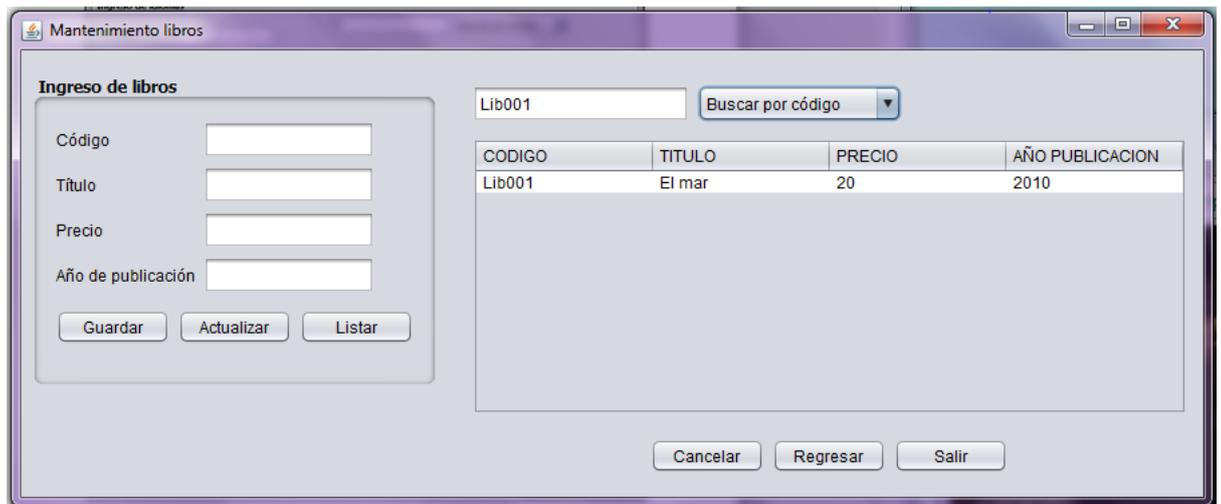


Figura 4. 21 Resultado de la búsqueda por código.

- Búsqueda por título

Sentencia sql:

```
"SELECT *
FROM libro
WHERE "+ "tituloLibro LIKE '%" + valor + "%'";
```

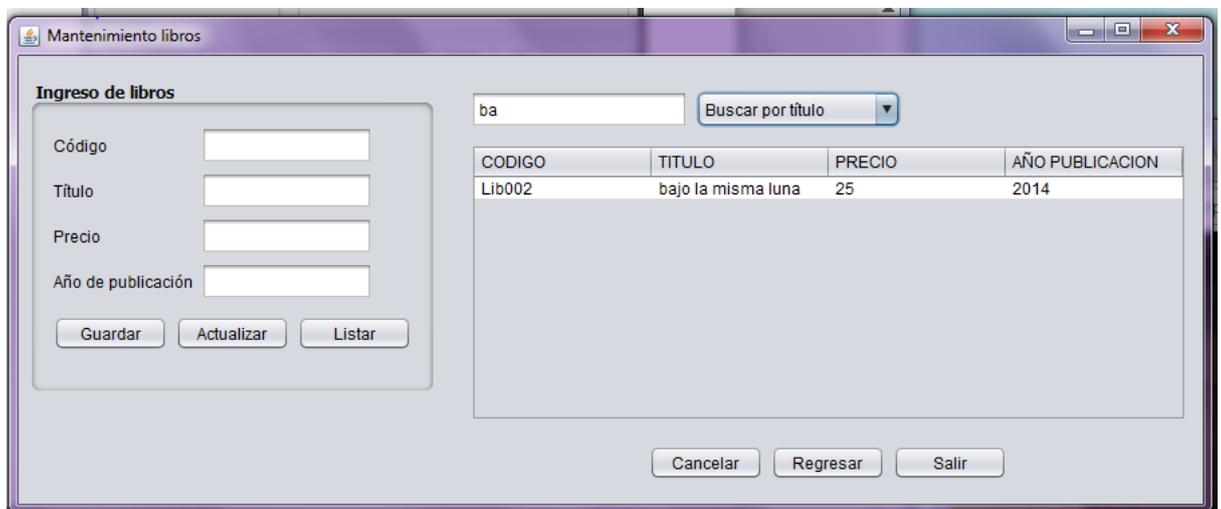


Figura 4. 22 Resultado de la búsqueda por título.

### Mantenimiento traductor.

En la ventana de mantenimiento de traductores se realiza el ingreso, modificación, eliminación, listar y consultas.

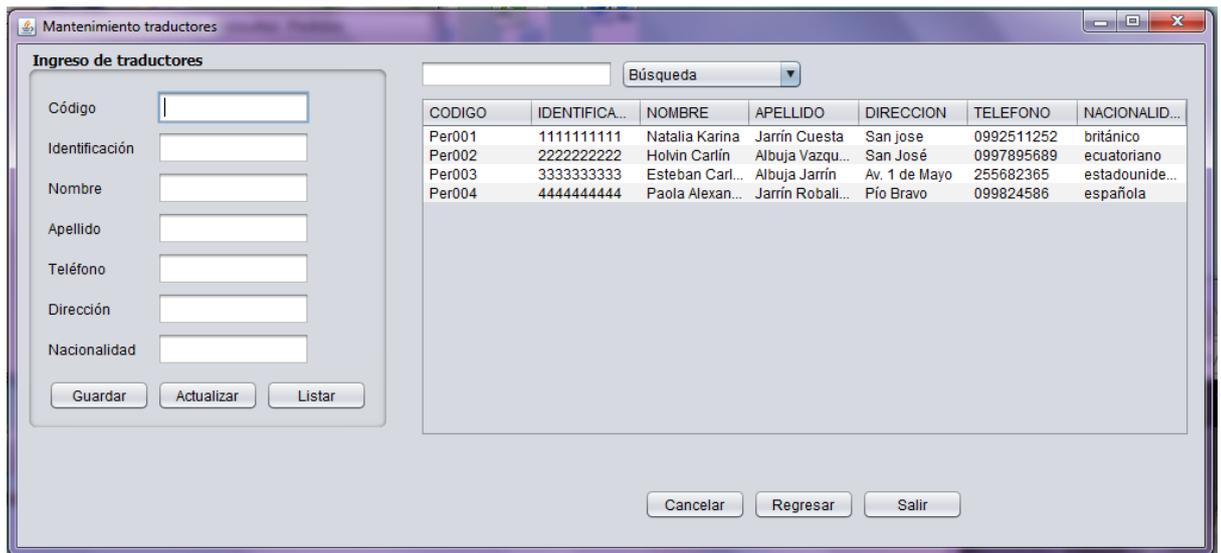


Figura 4. 23 Ventana del mantenimiento de traductores.

Se pueden realizar dos tipos de búsqueda:

- Búsqueda por código

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
      nombrePersonas, apellidoPersonas,
      direccionPersonas, telefonoPersonas,
      nacionalidadTraductor
FROM personas, traductor
WHERE codigoPersonas = codigoTraductor
      and "+ "codigoPersonas = '"+valor+"'";
```

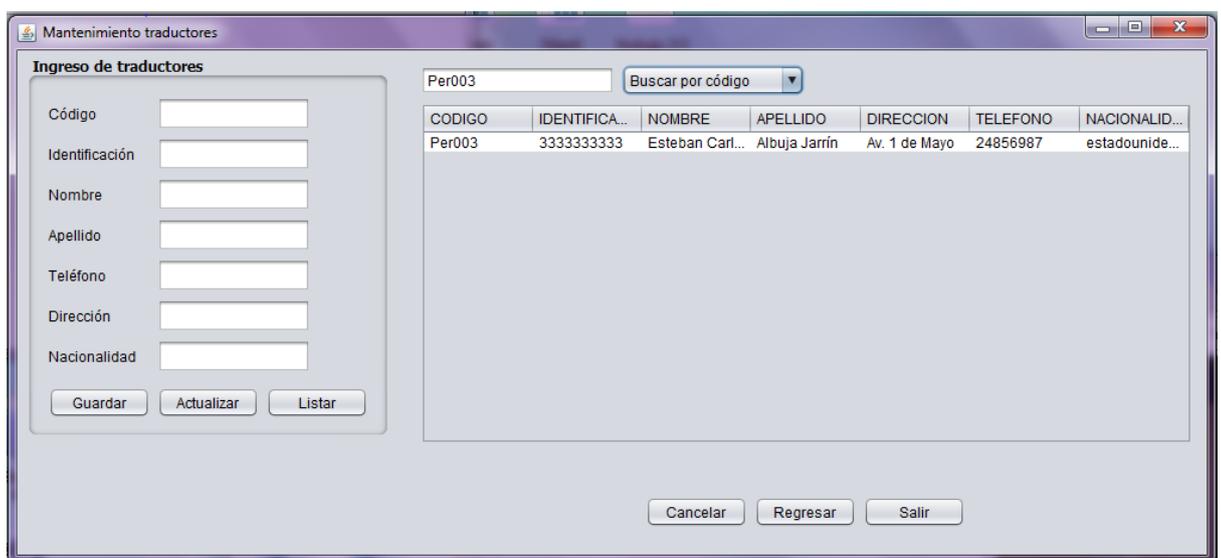


Figura 4. 24 Resultado de la búsqueda por código.

- Búsqueda por apellido

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,  
       nombrePersonas, apellidoPersonas,  
       direccionPersonas, telefonoPersonas,  
       nacionalidadTraductor  
FROM personas, traductor  
WHERE codigoPersonas = codigoTraductor  
       and "+ "apellidoPersonas LIKE '%" + valor + "%'";
```

CODIGO	IDENTIFICA...	NOMBRE	APELLIDO	DIRECCION	TELEFONO	NACIONALID...
Per002	2222222222	Holvin Carlin	Albuja Vazqu...	San José	0997895689	ecuatoriano
Per003	3333333333	Esteban Carl...	Albuja Jarrin	Av. 1 de Mayo	24856987	estadounide...

Figura 4. 25 Resultado de la búsqueda por apellido.

## Mantenimiento Vendedores

En la ventana de mantenimiento de vendedores se realiza el ingreso, modificación, eliminación, listar y consultas.

The screenshot shows a window titled "Mantenimiento vendedores" with a sub-header "Ingreso de vendedores". On the left is a form with fields for: Código, Identificación, Nombre, Apellido, Teléfono, Dirección, Comisión, Fecha de inicio, Almacén (dropdown menu showing "San Pedro"), tipo de vendedor (dropdown menu showing "Tipo Vendedor"), número de horas, sueldo por hora, and salario. At the bottom of the form are buttons for "Guardar", "Actualizar", and "Listar vendedores". On the right is a search bar with a "Búsqueda" dropdown and a table of vendors. The table has columns: CODIGO, IDENTIFICA..., NOMBRE, APELLIDO, DIRECCION, TELEFONO, COMISI..., FECHA INI..., and ALMACEN. Below the table are buttons for "Cancelar", "Regresar", and "Salir".

CODIGO	IDENTIFICA...	NOMBRE	APELLIDO	DIRECCION	TELEFONO	COMISI...	FECHA INI...	ALMACEN
Per013	999999999	Karina	Alvarado	San jose	0997895689	0.5	10/05/2015	Gran Colo...
Per014	122222222	Felipe	Alvarado Espinoza	San jose	0997895689	0.35	10/05/2015	Quito y sucre
Per016	233333333	Paola	Rodríguez Jarrín	Vega Muñoz	5885562	0.15	10/05/2016	Sucre
Per018	344444444	Jorge	Alvarez	Av 1 de ma...	25896593	0.89	10/5/1995	Sucre

Figura 4. 26 Ventana del mantenimiento de vendedores.

Se pueden realizar dos tipos de búsqueda y de listado:

- Búsqueda por código

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
nombrePersonas, apellidoPersonas,
direccionPersonas, telefonoPersonas,
comision, fechaInicio,
direccionAlmacen
FROM personas, vendedor, almacen
WHERE codigoPersonas = codigoVendedor
and codigoAlmacen = codigoAlmacenVendedor
and "+ "codigoPersonas = '"+valor+"";
```

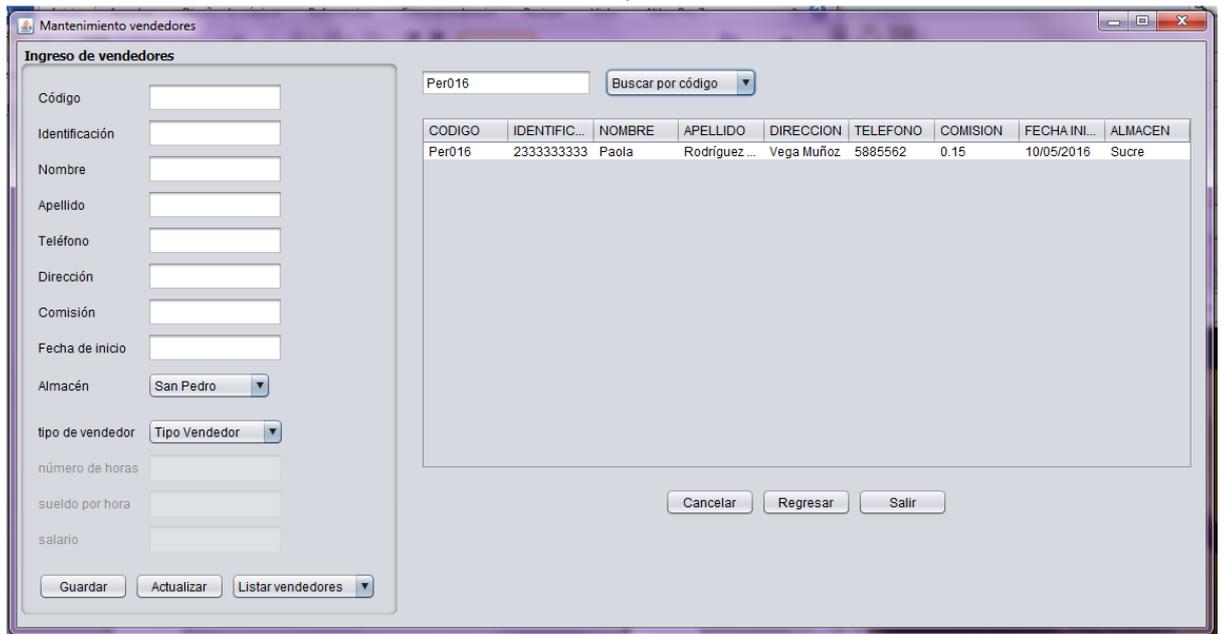


Figura 4. 27 Resultado de la búsqueda por código

- Búsqueda por apellido

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
      nombrePersonas, apellidoPersonas,
      direccionPersonas, telefonoPersonas,
      comision, fechaInicio, direccionAlmacen
FROM personas, vendedor, almacen
WHERE codigoPersonas = codigoVendedor
      and codigoAlmacen= codigoAlmacenVendedor
      and "+ "apellidoPersonas LIKE '%" + valor + "%'";
```

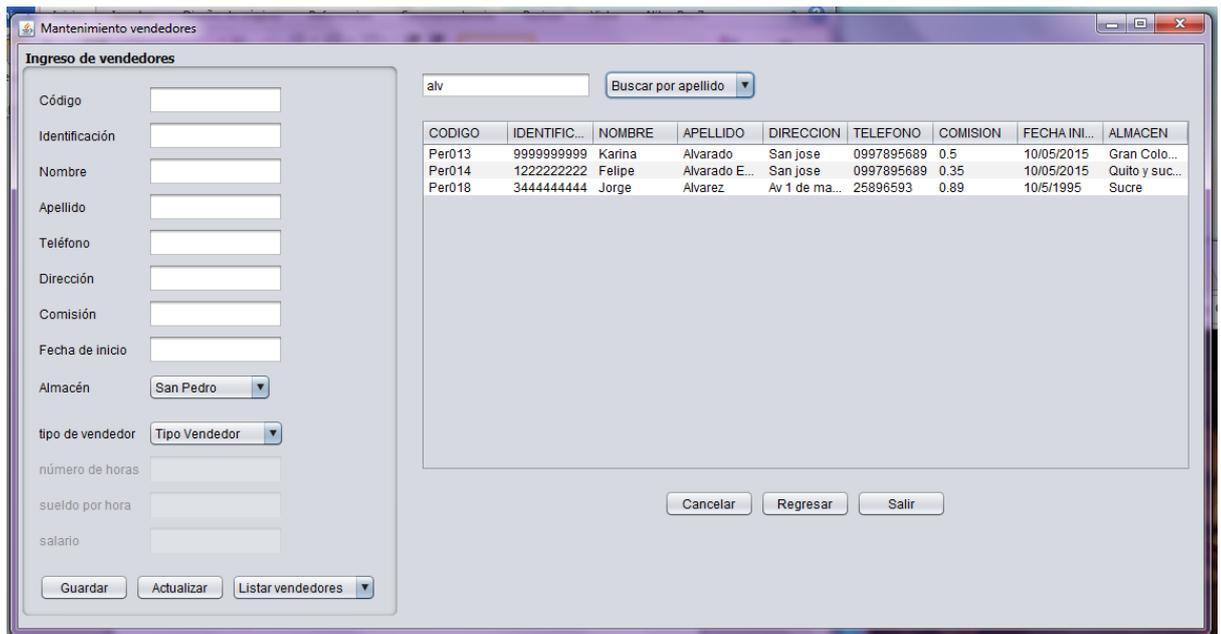


Figura 4. 28 Resultado de la búsqueda por apellido.

- Búsqueda por almacén

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
nombrePersonas, apellidoPersonas,
direccionPersonas, telefonoPersonas,
comision, fechaInicio, direccionAlmacen
FROM personas, vendedor, almacen
WHERE codigoPersonas = codigoVendedor
and codigoAlmacen= codigoAlmacenVendedor
and "+ "direccionAlmacen LIKE '%" + valor + "%'";
```

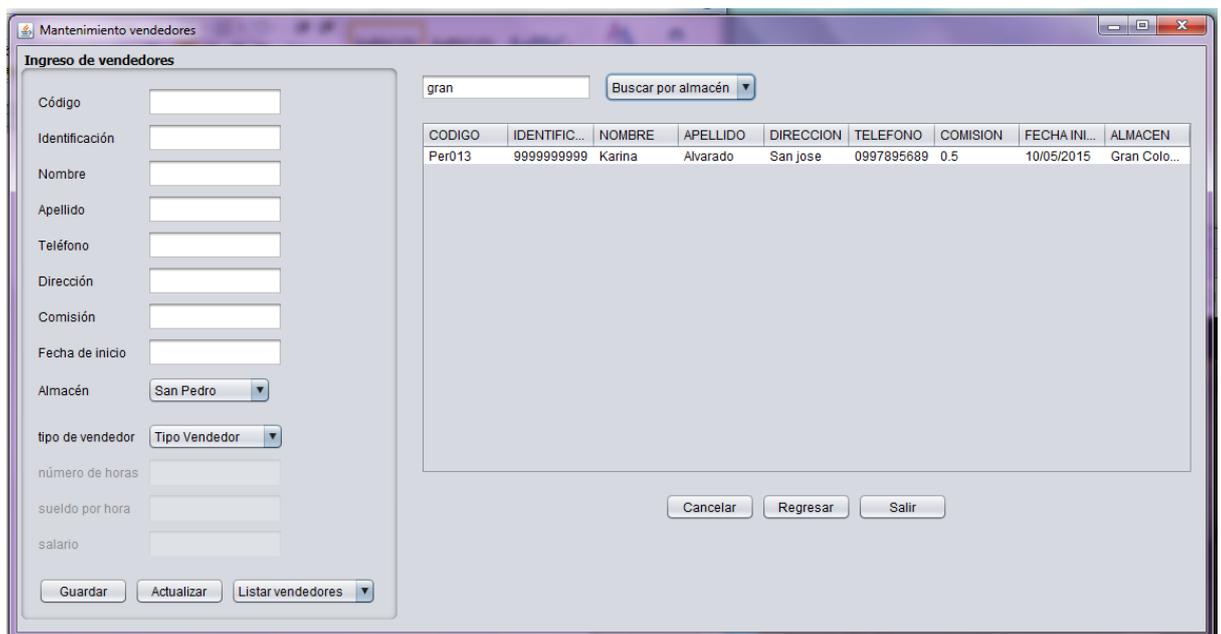


Figura 4. 29 Resultado de la búsqueda por almacén.

- Listar por vendedor tiempo completo

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
      nombrePersonas, apellidoPersonas,
      direccionPersonas, telefonoPersonas,
      comision, fechaInicio,
      direccionAlmacen, salario
FROM personas, vendedor, almacen, vendedortiemposcompleto
WHERE codigoPersonas = codigoVendedor
      and codigoAlmacen=codigoAlmacenVendedor
      and codigoPersonas= codTiempoCompleto";
```

CODIGO	IDENTIFI...	NOMBRE	APELLIDO	DIRECC...	TELEFO...	COMISION	FECHA L...	ALMACEN	SALARIO
Per013	9999999...	Karina	Alvarado	San jose	0997895...	0.5	10/05/2015	Gran Col...	3000

Figura 4. 30 Listado de vendedores a tiempo completo.

Listar por vendedor medio tiempo

Sentencia sql:

```
"SELECT codigoPersonas, identificacionPersonas,
      nombrePersonas, apellidoPersonas,
      direccionPersonas, telefonoPersonas,
      comision, fechaInicio, direccionAlmacen,
      numHoras, sueldoHora
FROM personas, vendedor, almacen, vendedormediotiempo
WHERE codigoPersonas = codigoVendedor
      and codigoAlmacen=codigoAlmacenVendedor
      and codigoPersonas= codMedioTiempo";
```

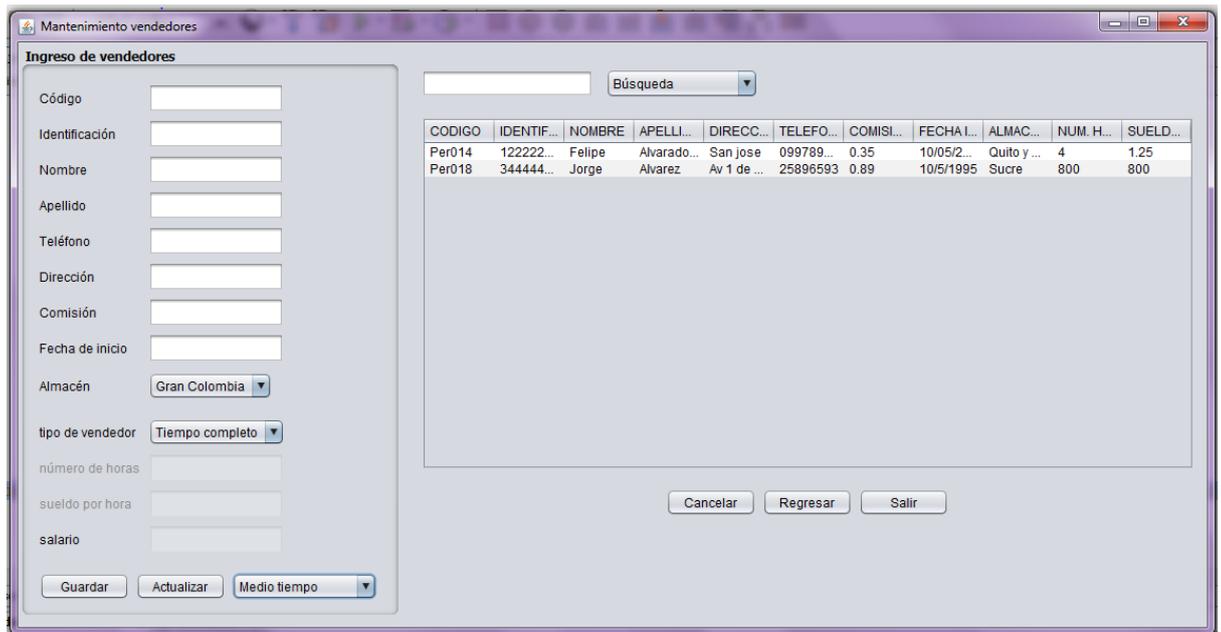


Figura 4. 31 Listado de vendedores a medio tiempo.

#### 4.3.1.2 Menú Administrar libros

En el menú "Administrar libros" se ingresa la información de los autores, idioma, editorial y traductor también se ingresa y modifica el stock existente en un almacén determinado de libros por cada idioma.

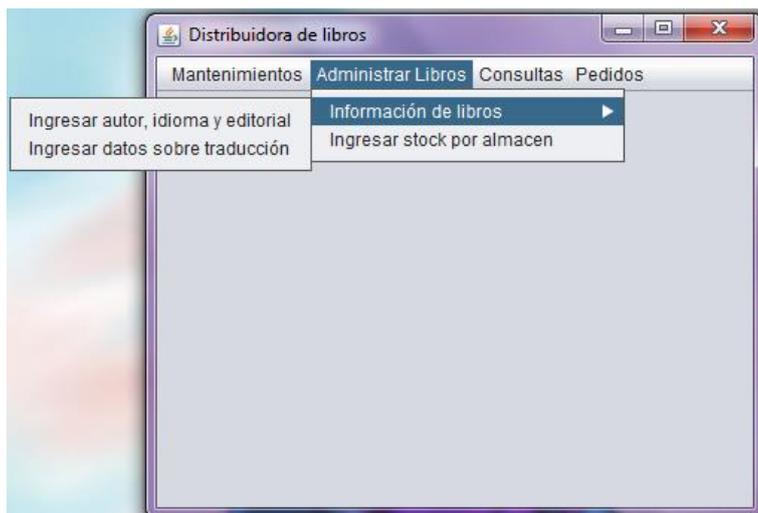


Figura 4. 32 Menú administrar libros.

## Información de libros.

### Ingresar autor, idioma y editorial

En la pantalla se listan los libros ingresados, al dar clic derecho se pueden agregar autor, editorial e idioma, los cuales ya están cargados en las listas.

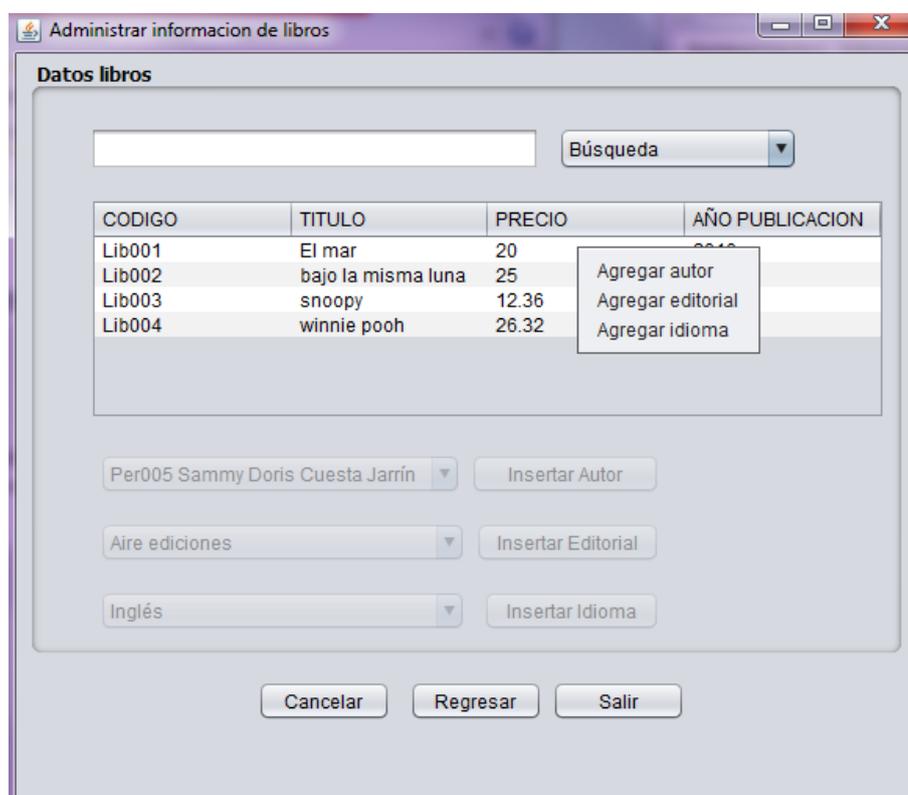


Figura 4. 33 Ventana "Administrar información de libros"

### Ingresar datos sobre traducción

En la pantalla se listan los libros ingresados en los idiomas en los que está disponible, al dar clic derecho se puede agregar al responsable de la traducción.

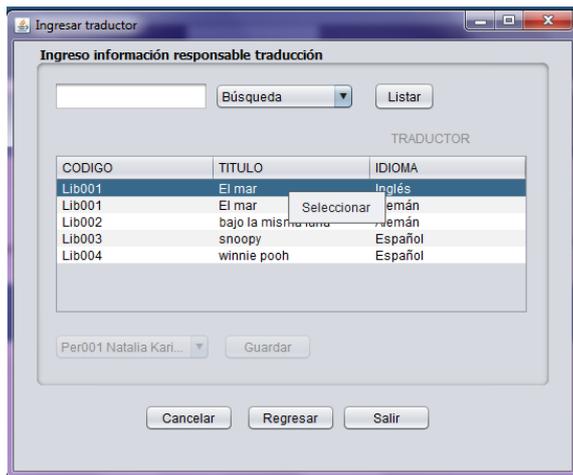


Figura 4. 34 Venta "Ingresar traductor".

Al dar clic en el botón listar se abre una nueva ventana en donde se muestra información del libro, idioma y su traductor.

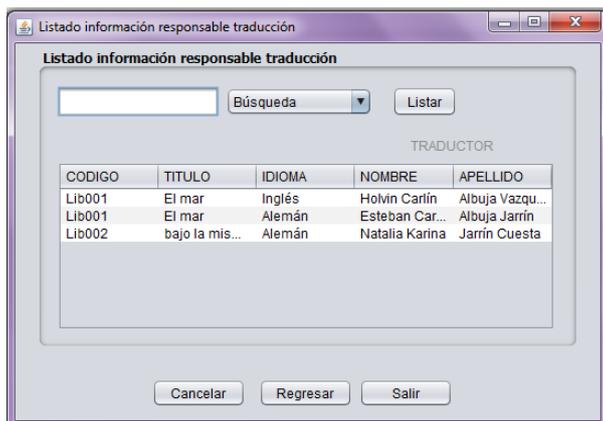


Figura 4. 35 Listado de la información del responsable de la traducción.

### Ingresar stock por almacén

En la pantalla se listan los libros ingresados y los idiomas en los cuales están disponibles, al seleccionar una publicación y dar clic derecho se puede ingresar la cantidad, se escoge el almacén en el cual está disponible

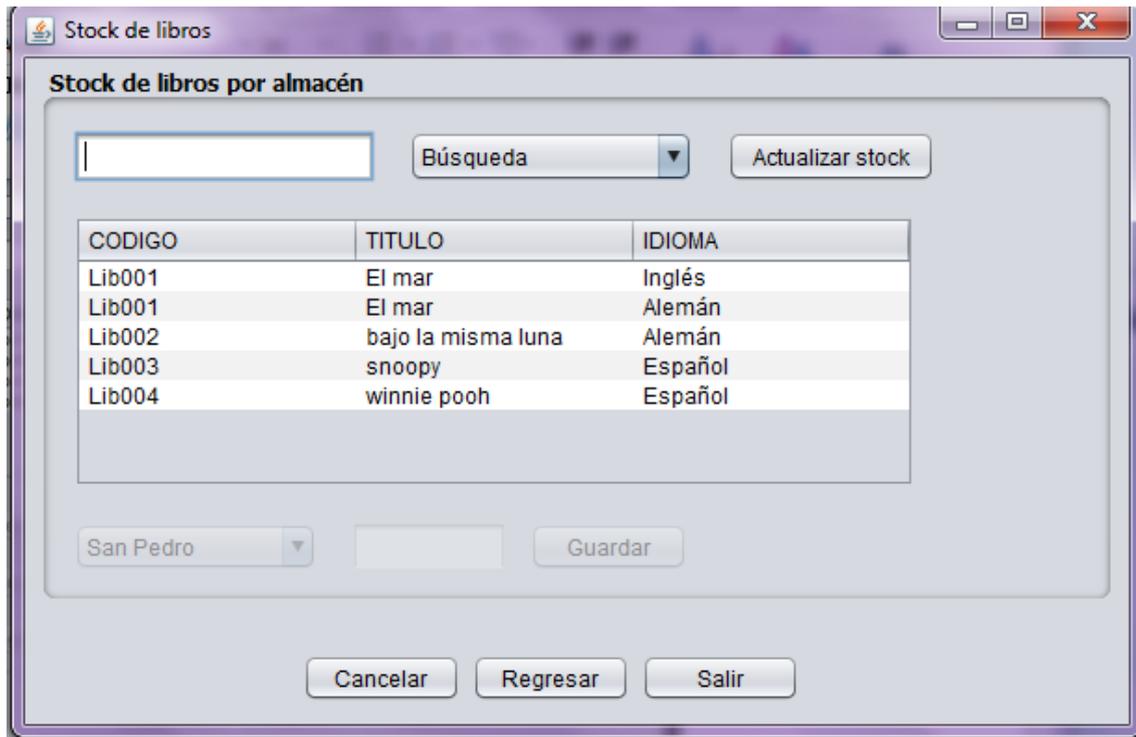


Figura 4. 36 Ventana "Ingresar stock por almacén"

Si el libro en un determinado idioma ya ha sido ingresado en el stock de un almacén se mostrarán dos mensajes, el primero indica que la llave primaria esta repetida y el segundo pregunta si desea actualizar el stock, al responder "SI" se muestra la tabla de libros por idioma y almacén con sus respectivas cantidades las cuales pueden ser actualizadas, también se puede actualizar el stock al dar clic en el botón "Actualizar stock".



Figura 4. 37 Mensaje de petición de actualización

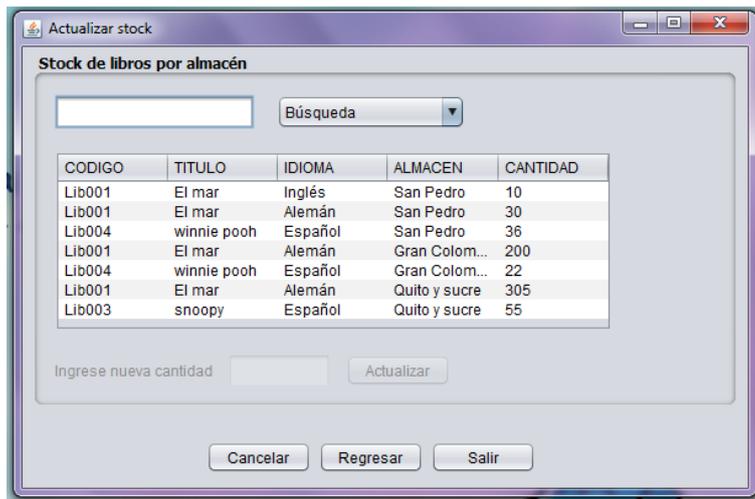


Figura 4. 38 Ventana "Actualizar stock"

#### 4.3.1.3 Menú Consultas

En el menú "Consultas" se pueden realizar búsquedas de los libros según diferentes criterios.

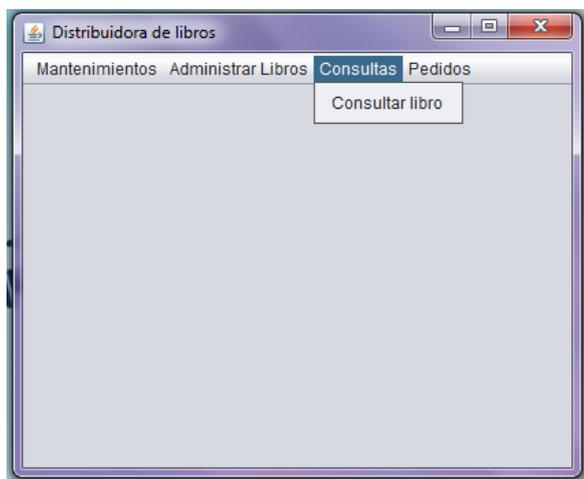


Figura 4. 39 Menú "Consultas".

#### Consultar libros

Se pueden realizar consultas por autor, título, código e idioma

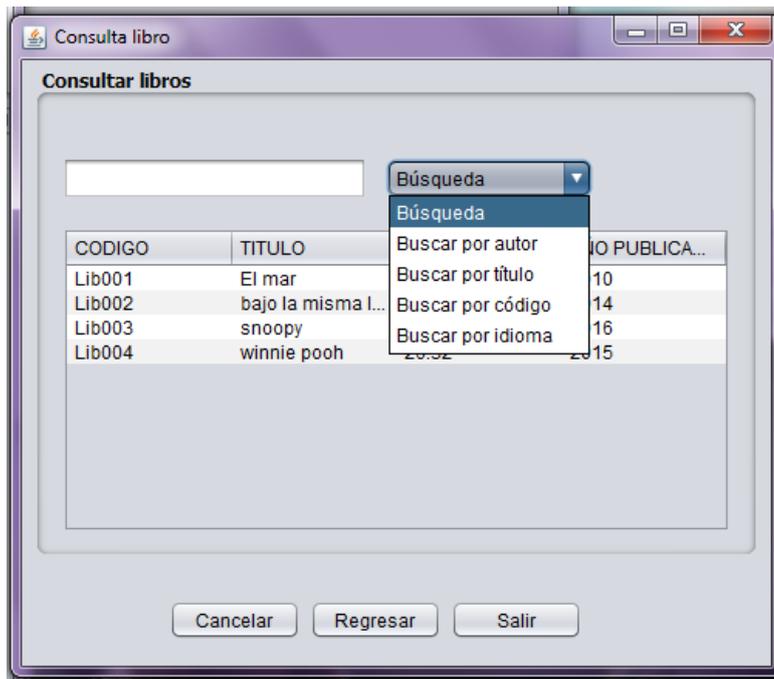


Figura 4. 40 Ventana Consulta libro (opciones de búsqueda)

- Buscar por autor

Se listan el código, título, precio, año de publicación y los nombres del o los autores de determinado libro al colocar el apellido del autor en el campo de texto se filtrará la tabla según lo escrito.

Sentencia sql=

```
"SELECT DISTINCT codigoLibro, tituloLibro, precioLibro,
                    anioPublicacionLibro, nombrePersonas,
                    apellidoPersonas
FROM personas, libro, autor, escribe
WHERE codigoPersonas = codAutor
    and codigoLibro = codLibro
    and "+ "apellidoPersonas LIKE '%" + valor + "%'";
```

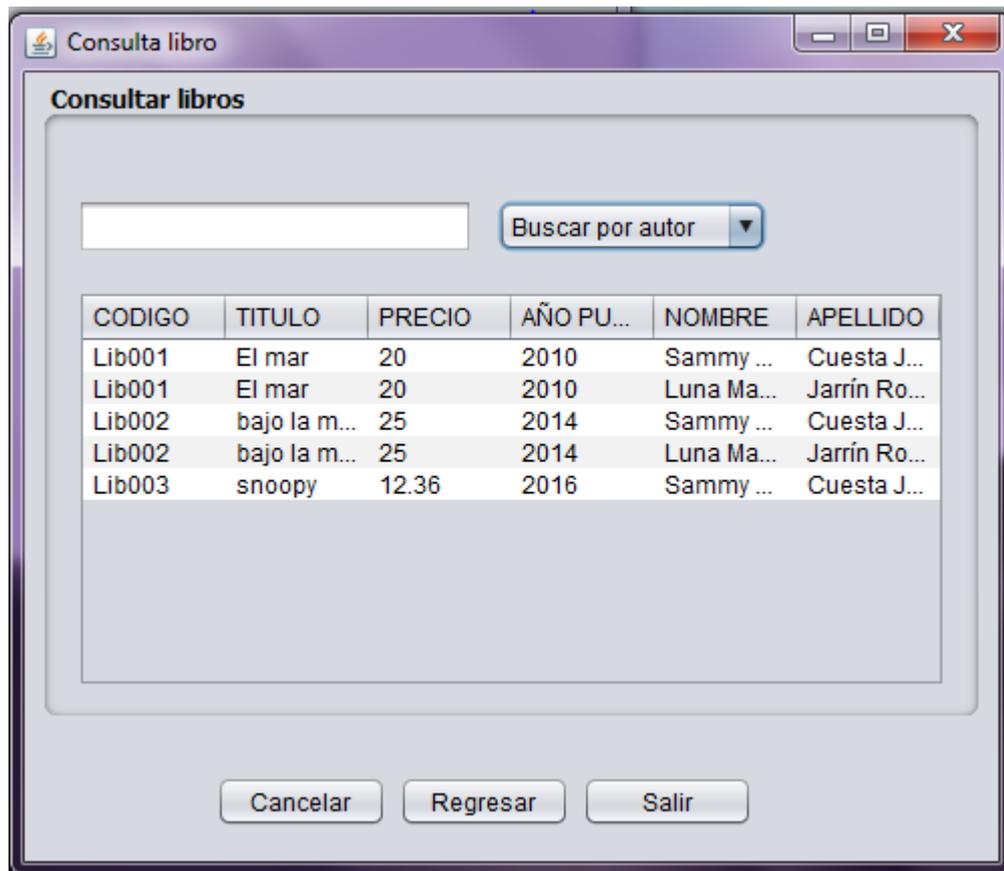


Figura 4. 41 Listado por autor.

- Buscar por título

Al ingresar el título en el campo de texto se filtra la tabla y muestra el código, título, precio y año de publicación del libro consultado

Sentencia sql:

```
"SELECT *
FROM libro
WHERE "+ "tituloLibro LIKE '%" + valor + "%'";
```

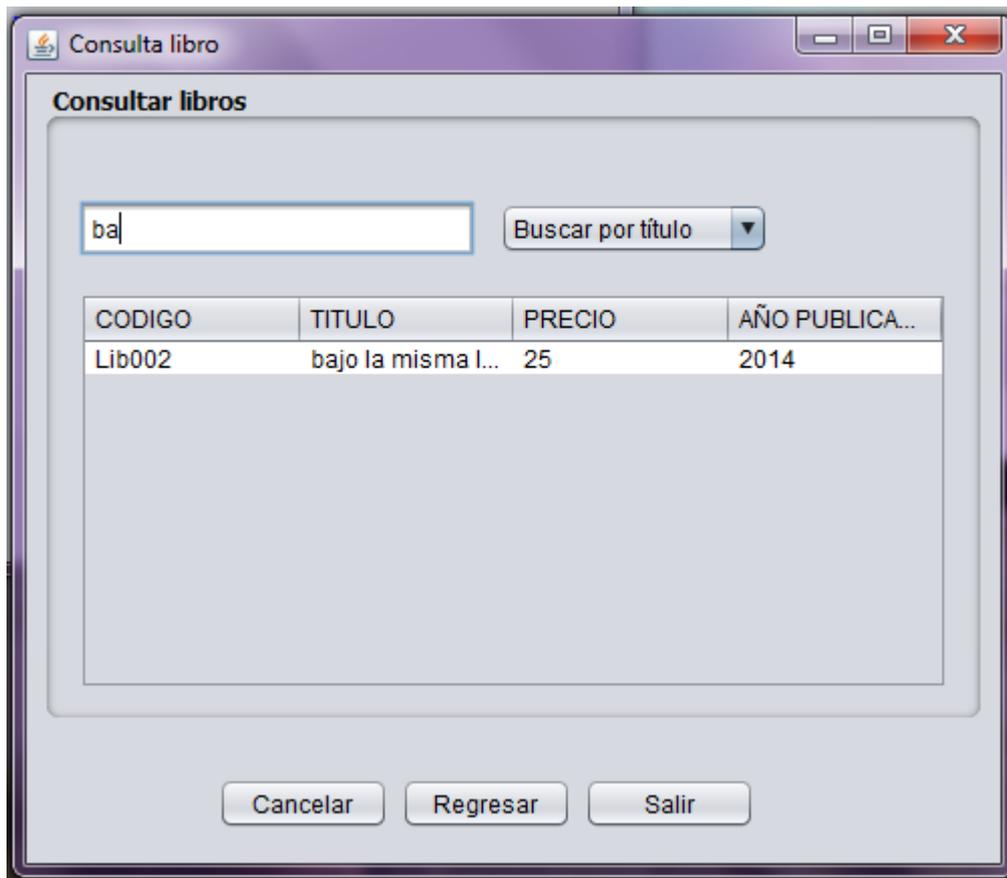


Figura 4. 42 Resultado de la búsqueda por título.

- Buscar por código

Al ingresar el código en el campo de texto se filtra la tabla y muestra el código, título, precio y año de publicación del libro consultado

sentencia sql:

```
"SELECT *  
FROM libro  
WHERE "+ "codigoLibro = '"+valor+'";
```

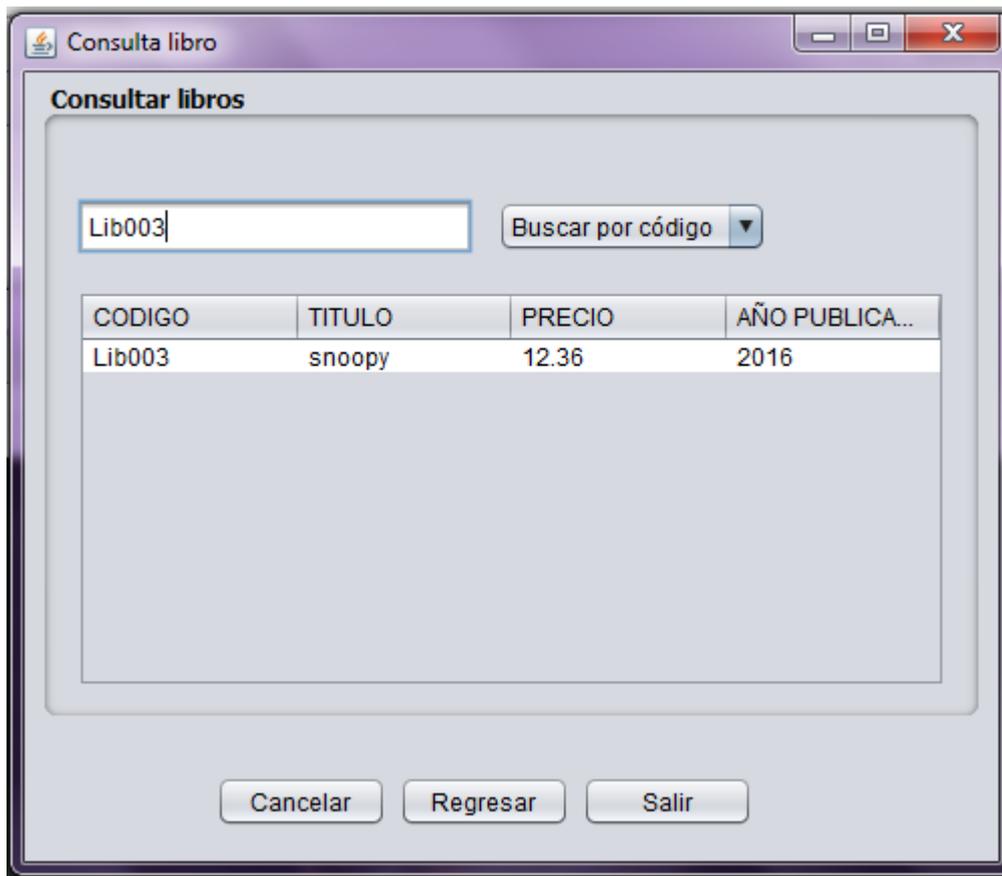


Figura 4. 43 Resultado de la búsqueda por código.

- Buscar por idioma

Se listan el código, título, precio, año de publicación e idioma en el que está disponible el libro, al colocar el idioma del cual se desea conocer las publicaciones disponibles en el campo de texto se filtrará la tabla según lo escrito.

Sentencia sql:

```
"SELECT codigoLibro, tituloLibro, precioLibro,
      anioPublicacionLibro, descripcionIdioma
FROM libro, traduccion, idioma
WHERE codigoLibro = codLibr
      and codigoIdioma = codIdioma
      and " + "descripcionIdioma LIKE '%" + valor + "%' ";
```

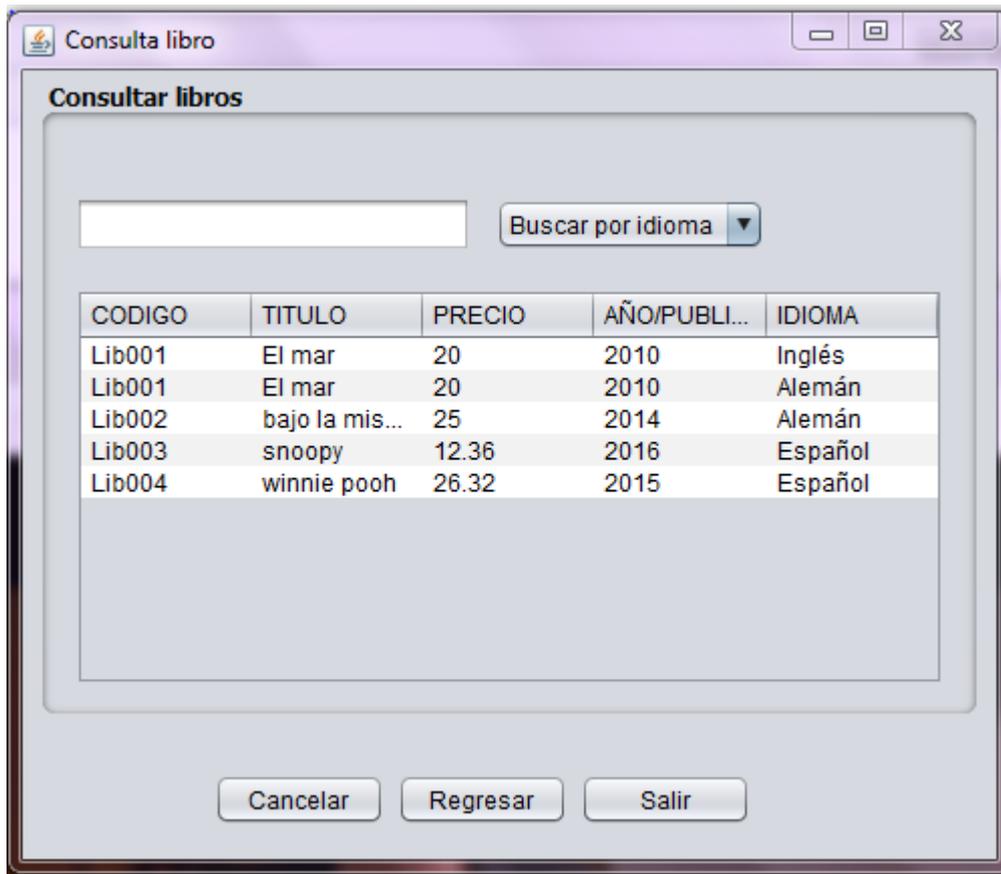


Figura 4. 44 Listado por idioma.

#### 4.3.1.4 Menú Pedidos

En el menú "Pedidos" se pueden ingresar y consultar pedidos

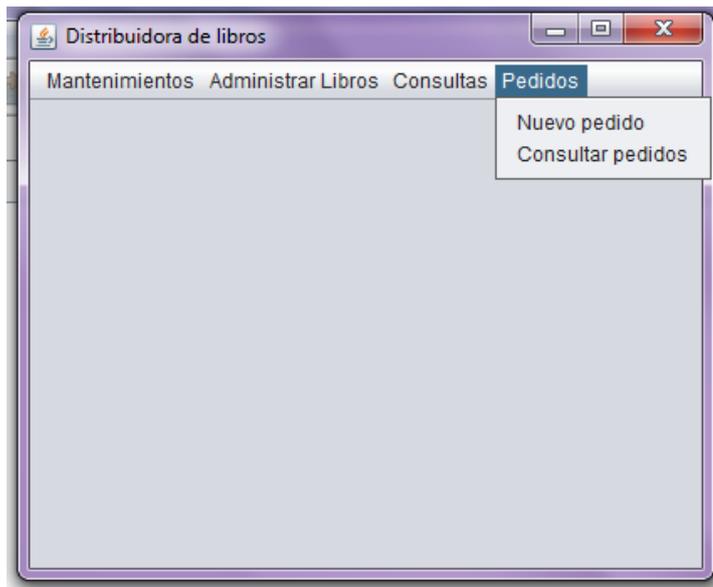


Figura 4. 45 Menú "Pedidos".

### **Ingreso nuevo pedido**

En la pantalla se presenta el formulario con los datos necesarios para ingresar un nuevo pedido, en cuanto al vendedor y cliente los datos se buscan al dar clic en los respectivos botones, y se selecciona el indicado de una tabla.

Ingreso nuevo pedido

### Cabecera pedido

Número de pedido

Fecha

Dirección

DATOS VENDEDOR DATOS CLIENTE

Código  Código

Nombre  Nombre

Apellido  Apellido

Almacén

### Detalle pedido

CANTIDAD	CODIGO	TITULO	COD_IDIO...	IDIOMA	PRECIO	SUBTOTAL

Figura 4. 46 Ventana de ingreso de pedidos.

## Buscar vendedor

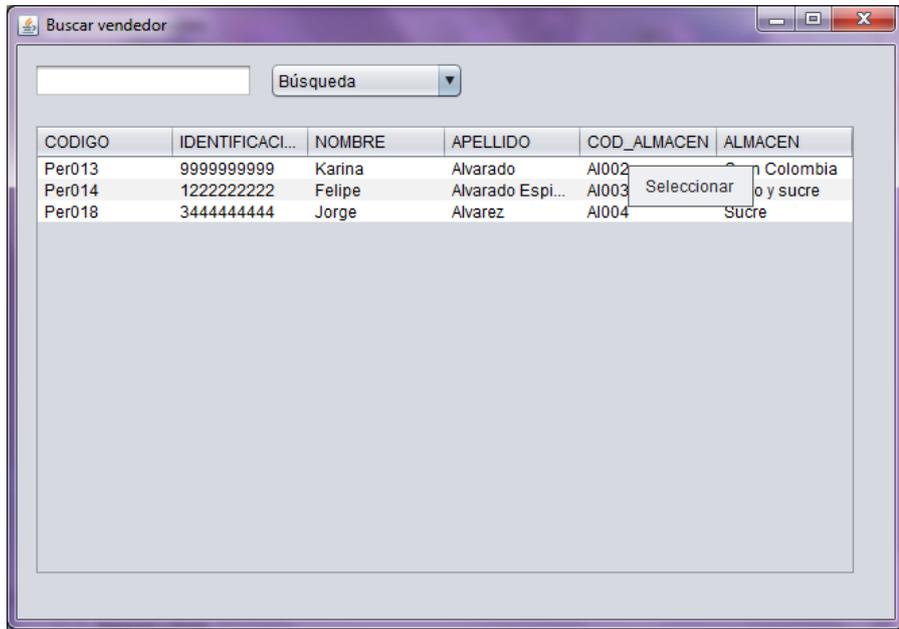


Figura 4. 47 Ventana buscar vendedor.

## Buscar cliente

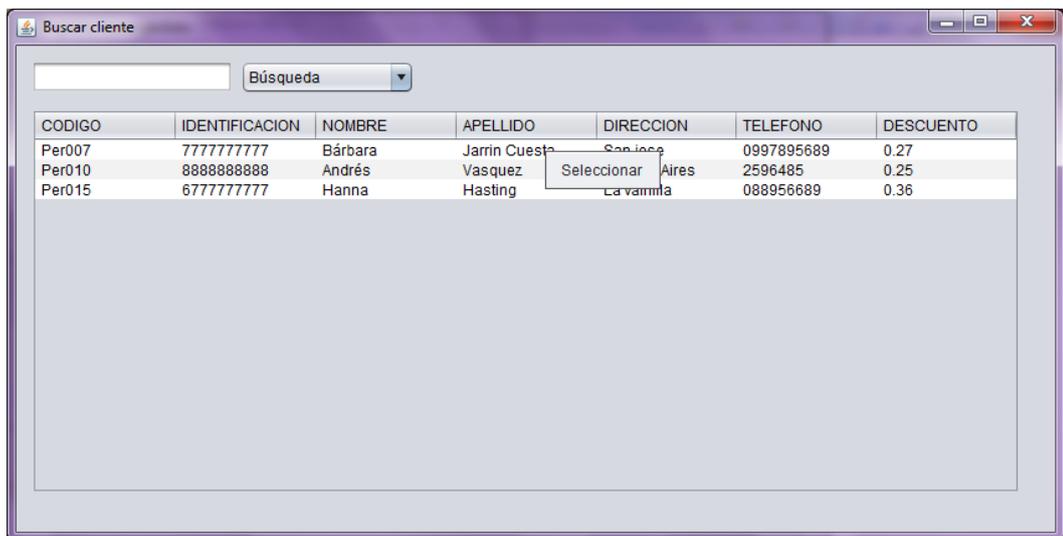


Figura 4. 48 Ventana buscar cliente.

Para la selección de los libros del pedido se da clic en el botón agregar libros, se despliega otra pantalla y al dar clic el botón Listar libros se muestran las publicaciones según el almacén en donde se está realizando el pedido, se ingresa la cantidad y una vez elegidos los libros se da clic en el botón REALIZAR PEDIDO.

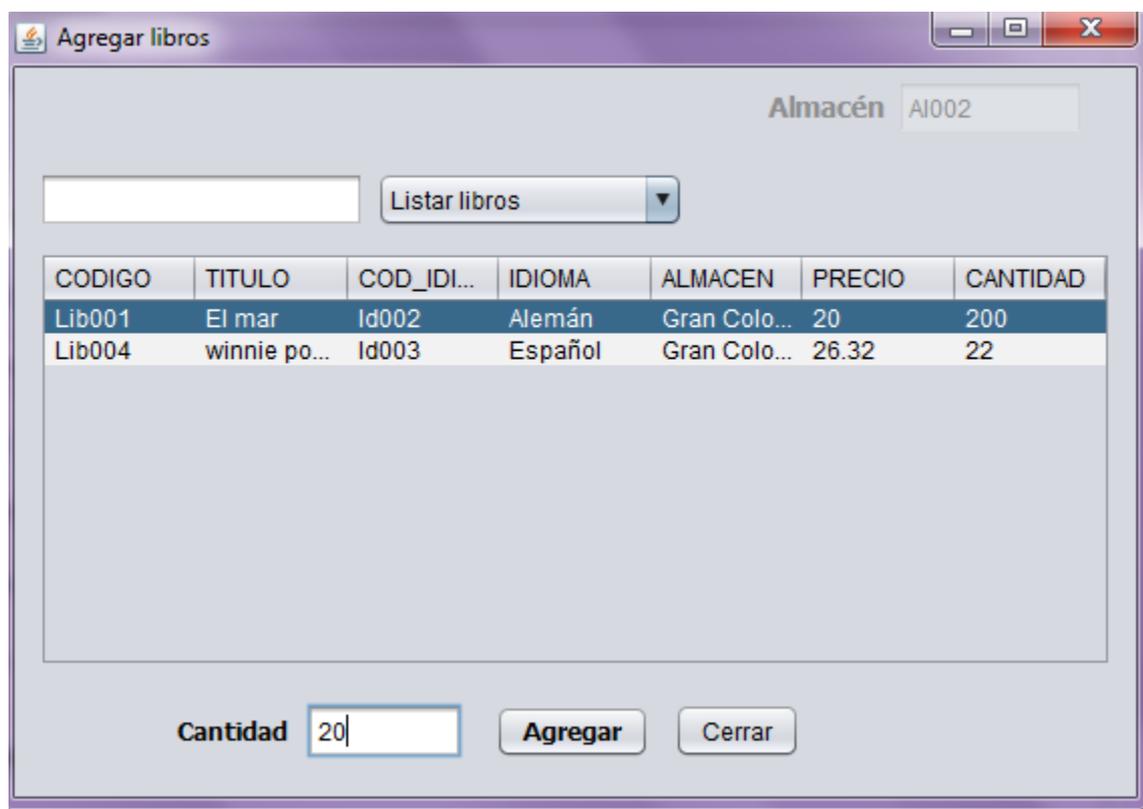


Figura 4. 49 Ventana agregar libros.

Sentencia sql para guardar en la tabla pedido y detalla, respectivamente:

```
"INSERT INTO pedido(numPedido, fechaPedido,
direccionEnvioPedido,
codVendedor, codCliente)VALUES(?,?,?,?);"
```

```
"INSERT INTO detalla(numPed, codigoPedidoLibro, codIdm,
cantidadDetalle)VALUES(?,?,?,?);"
```

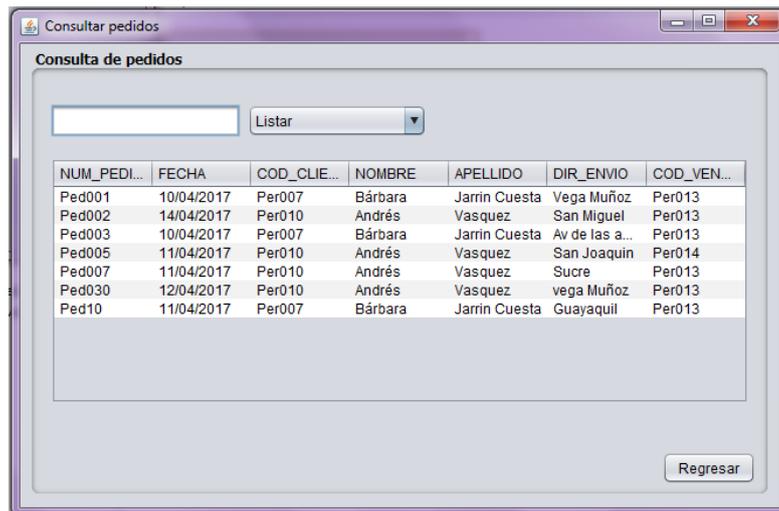
### Consultar pedidos

En la pantalla se muestran las cabeceras de los pedidos realizados, al señalar uno y dar clic derecho se despliega la opción mostrar, lo abre otra ventana , en esta se

muestra la cabecera y al dar clic en Mostrar detalle se listan los libros dentro del pedido.

Sentencia sql que obtiene los datos de la cabecera del pedido.

```
"SELECT numPedido, fechaPedido, codCliente, nombrePersonas,
        apellidoPersonas,direccionEnvioPedido, codVendedor
FROM pedido, personas
WHERE codigoPersonas= codCliente";
```



NUM_PEDI...	FECHA	COD_CLIE...	NOMBRE	APELLIDO	DIR_ENVIO	COD_VEN...
Ped001	10/04/2017	Per007	Bárbara	Jarrin Cuesta	Vega Muñoz	Per013
Ped002	14/04/2017	Per010	Andrés	Vasquez	San Miguel	Per013
Ped003	10/04/2017	Per007	Bárbara	Jarrin Cuesta	Av de las a...	Per013
Ped005	11/04/2017	Per010	Andrés	Vasquez	San Joaquin	Per014
Ped007	11/04/2017	Per010	Andrés	Vasquez	Sucre	Per013
Ped030	12/04/2017	Per010	Andrés	Vasquez	vega Muñoz	Per013
Ped10	11/04/2017	Per007	Bárbara	Jarrin Cuesta	Guayaquil	Per013

Figura 4. 50 Ventana consultar pedidos.

Sentencia sql que contiene los datos del detalle del pedido seleccionado.

```
"SELECT cantidadDetalle, codigoPedidoLibro, tituloLibro,
        codIdm, descripcionIdioma, precioLibro
FROM libro, detalla, idioma
WHERE codigoLibro = codigoPedidoLibro
and codigoIdioma = codIdm
and "+ "numPed ='"+valor+"' ";
```

CANTIDAD	CODIGO	TITULO	COD_IDIOMA	IDIOMA	PRECIO
10	Lib001	El mar	Id002	Alemán	20
2	Lib004	winnie pooh	Id003	Español	26.32

Figura 4. 51 Ventana datos del pedido.

#### 4.4 Conclusiones

En el presente trabajo investigativo se diseñó una base de datos usando el modelo entidad relación extendido (EER), la cual fue implementada en el sistema de gestión de bases de datos MySQL 5.7.15 y su aplicación se desarrolló en el lenguaje de programación JAVA, utilizando la herramienta Netbeans IDE 8.2.

Se aborda cada uno de los conceptos de modelo EER, utilizando un estudio de caso para facilitar la comprensión al lector y de esta forma presentar una alternativa eficiente para el manejo de información de una base de datos, ya que el modelo mejora la capacidad expresiva del esquema conceptual a un alto nivel de abstracción.

Al aplicar los conceptos del modelo EER para el diseño de la base de datos del estudio de caso “Distribuidora de libros” se obtuvo un modelo conceptual de fácil comprensión. Tanto la información requerida como las relaciones involucradas en este, se han logrado expresar con bastante exactitud, manteniendo una visión global del caso; de igual forma la base de datos creada tiene integridad referencial y no existe redundancia de datos.

## REFERENCIAS

### Bibliografía

Bagui, S. (2009). Mapping OWL to the Entity Relationship and Extended Entity Relationship models. *Int. J. Knowledge and Web Intelligence*, 125-149.

Bajaj, A. (2010). *The Effect of Abstraction of Constructs in Data Models on Modeling Performance: An Exploratory Empirical Study*. Tusla: AMCIS Proceedings.

Connolly, T. M., & Begg, C. E. (2005). *Sistemas de bases de datos, Un enfoque práctico para el diseño, implementación y gestión*. Essex: Pearson Education.

Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla. (Febrero de 2007). *Diseño de base de datos*. Sevilla, España.

Elmasri, R., & Navathe, S. B. (2007). *Fundamentos de Sistemas de Bases de Datos*. Madrid: Pearson Educación.

Instituto Tecnológico Superior de Lerdo. (2010). *MODELO ENTIDAD-RELACION EXTENDIDO*. Lerdo, México.

Thalheim, B. (2009). *Extended Entity-Relationship Model*. Kiel.

## Anexos

### Anexo A: Script de la base de datos "distribuidoralibrosuno"

```
-- MySQL dump 10.13  Distrib 5.7.12, for Win64 (x86_64)
--
-- Host: localhost    Database: distribuidoralibrosuno
--
-----
-- Server version 5.7.15-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `almacen`
--

DROP TABLE IF EXISTS `almacen`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `almacen` (
  `codigoAlmacen` varchar(5) NOT NULL,
  `direccionAlmacen` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`codigoAlmacen`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Lugar donde se venden
los libros';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `autor`
--

DROP TABLE IF EXISTS `autor`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `autor` (
  `codigoAutor` varchar(6) NOT NULL,
  `nacionalidadAutor` varchar(20) DEFAULT NULL,
  `tituloAutor` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`codigoAutor`),
  CONSTRAINT `codigoAutor` FOREIGN KEY (`codigoAutor`) REFERENCES
`personas` (`codigoPersonas`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Autor de un libro';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `cliente`
--

DROP TABLE IF EXISTS `cliente`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `cliente` (
  `codigoCliente` varchar(6) NOT NULL,
  `descuentoCliente` double DEFAULT NULL,
  PRIMARY KEY (`codigoCliente`),
  CONSTRAINT `codigoCliente` FOREIGN KEY (`codigoCliente`)
REFERENCES `personas` (`codigoPersonas`) ON DELETE CASCADE ON UPDATE
CASCADE
```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Cliente que compra los
libros';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `detalla`
--

DROP TABLE IF EXISTS `detalla`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `detalla` (
  `numPed` varchar(6) NOT NULL,
  `codigoPedidoLibro` varchar(6) NOT NULL,
  `codIdm` varchar(5) NOT NULL,
  `cantidadDetalle` int(11) NOT NULL,
  `subtotalDetalle` double,
  KEY `numPed_idx` (`numPed`),
  KEY `codigoPedidoLibro_idx` (`codigoPedidoLibro`),
  KEY `codIdm_idx` (`codIdm`),
  CONSTRAINT `codIdm` FOREIGN KEY (`codIdm`) REFERENCES `idioma`
(`codigoIdioma`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `codigoPedidoLibro` FOREIGN KEY (`codigoPedidoLibro`)
REFERENCES `libro` (`codigoLibro`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `numPed` FOREIGN KEY (`numPed`) REFERENCES `pedido`
(`numPedido`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Relación entre pedido,
los libros y su idioma';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `edita`
--

DROP TABLE IF EXISTS `edita`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `edita` (
  `codEditorial` varchar(6) NOT NULL,
  `codEditaLibro` varchar(6) NOT NULL,
  PRIMARY KEY (`codEditorial`,`codEditaLibro`),
  KEY `codEditorial_idx` (`codEditorial`),
  KEY `codEditaLibro_idx` (`codEditaLibro`),
  CONSTRAINT `codEditaLibro` FOREIGN KEY (`codEditaLibro`)
REFERENCES `libro` (`codigoLibro`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `codEditorial` FOREIGN KEY (`codEditorial`) REFERENCES
`editorial` (`codigoEditorial`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Relación entre
editorial y libro';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `editorial`
--

DROP TABLE IF EXISTS `editorial`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `editorial` (
  `codigoEditorial` varchar(6) NOT NULL,
  `nombreEditorial` varchar(20) DEFAULT NULL,
  `telefonoEditorial` varchar(10) DEFAULT NULL,
  `ciudadEditorial` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`codigoEditorial`),
  UNIQUE KEY `codigoEditorial_UNIQUE` (`codigoEditorial`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Edita los libros';
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Table structure for table `escribe`
--

DROP TABLE IF EXISTS `escribe`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `escribe` (
  `codAutor` varchar(6) NOT NULL,
  `codLibro` varchar(6) NOT NULL,
  PRIMARY KEY (`codAutor`,`codLibro`),
  KEY `codAutor_idx` (`codAutor`),
  KEY `codLibro_idx` (`codLibro`),
  CONSTRAINT `codAutor` FOREIGN KEY (`codAutor`) REFERENCES `autor`
(`codigoAutor`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `codLibro` FOREIGN KEY (`codLibro`) REFERENCES `libro`
(`codigoLibro`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Relacion entre el
autor y su libro';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `idioma`
--

DROP TABLE IF EXISTS `idioma`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `idioma` (
  `codigoIdioma` varchar(5) NOT NULL,
  `descripcionIdioma` varchar(20) NOT NULL,
  PRIMARY KEY (`codigoIdioma`,`descripcionIdioma`),
  UNIQUE KEY `codigoIdioma_UNIQUE` (`codigoIdioma`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Idioma en el que puede
estar escrito un libro';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `libro`
--

DROP TABLE IF EXISTS `libro`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `libro` (
  `codigoLibro` varchar(6) NOT NULL,
  `tituloLibro` varchar(30) NOT NULL,
  `precioLibro` double DEFAULT NULL,
  `anioPublicacionLibro` int(11) DEFAULT NULL,
  PRIMARY KEY (`codigoLibro`),
  UNIQUE KEY `codigoLibro_UNIQUE` (`codigoLibro`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Publicación vendida
por la distribuidora';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `mayorista`
--

DROP TABLE IF EXISTS `mayorista`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `mayorista` (
  `codMayorista` varchar(6) NOT NULL,
  `limiteCredito` double DEFAULT NULL,
  PRIMARY KEY (`codMayorista`),
  CONSTRAINT `codMayorista` FOREIGN KEY (`codMayorista`) REFERENCES
`cliente` (`codigoCliente`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Cliente que realiza
pedidos por cantidades grandes';

```

```

/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `minorista`
--

DROP TABLE IF EXISTS `minorista`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `minorista` (
  `codMinorista` varchar(6) NOT NULL,
  `mail` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`codMinorista`),
  CONSTRAINT `codMinorista` FOREIGN KEY (`codMinorista`) REFERENCES
`cliente` (`codigoCliente`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Cliente que realiza
pedidos por menor';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `pedido`
--

DROP TABLE IF EXISTS `pedido`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `pedido` (
  `numPedido` varchar(6) NOT NULL,
  `fechaPedido` varchar(10) DEFAULT NULL,
  `direccionEnvioPedido` varchar(30) DEFAULT NULL,
  `codVendedor` varchar(6) NOT NULL,
  `codCliente` varchar(6) NOT NULL,
  PRIMARY KEY (`numPedido`),
  UNIQUE KEY `numPedido_UNIQUE` (`numPedido`),
  KEY `codVendedor_idx` (`codVendedor`),
  KEY `codCliente_idx` (`codCliente`),
  CONSTRAINT `codCliente` FOREIGN KEY (`codCliente`) REFERENCES
`cliente` (`codigoCliente`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `codVendedor` FOREIGN KEY (`codVendedor`) REFERENCES
`vendedor` (`codigoVendedor`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Pedido realizado por
los clientes para comprar los libros';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `personas`
--

DROP TABLE IF EXISTS `personas`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `personas` (
  `codigoPersonas` varchar(6) NOT NULL,
  `identificacionPersonas` varchar(20) NOT NULL,
  `nombrePersonas` varchar(30) DEFAULT NULL,
  `apellidoPersonas` varchar(30) DEFAULT NULL,
  `direccionPersonas` varchar(30) DEFAULT NULL,
  `telefonoPersonas` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`codigoPersonas`),
  UNIQUE KEY `codigoPersonas_UNIQUE` (`codigoPersonas`),
  UNIQUE KEY `identificacionPersonas_UNIQUE`
(`identificacionPersonas`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Persona que interviene
en el estudio de caso';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `tiene`
--

```

```

DROP TABLE IF EXISTS `tiene`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client  = utf8 */;
CREATE TABLE `tiene` (
  `codAlmacen` varchar(5) NOT NULL,
  `codLibroPorAlmacen` varchar(6) NOT NULL,
  `codIdmPorAlmacen` varchar(5) NOT NULL,
  `cantidadAlmacen` int(11) NOT NULL,
  PRIMARY KEY
(`codAlmacen`, `codLibroPorAlmacen`, `codIdmPorAlmacen`),
  KEY `codAlmacen_idx` (`codAlmacen`),
  KEY `codLibroPorAlmacen_idx` (`codLibroPorAlmacen`),
  KEY `codIdmPorAlmacen_idx` (`codIdmPorAlmacen`),
  CONSTRAINT `codAlmacen` FOREIGN KEY (`codAlmacen`) REFERENCES
`almacen` (`codigoAlmacen`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `codIdmPorAlmacen` FOREIGN KEY (`codIdmPorAlmacen`)
REFERENCES `idioma` (`codigoIdioma`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `codLibroPorAlmacen` FOREIGN KEY (`codLibroPorAlmacen`)
REFERENCES `libro` (`codigoLibro`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Relación entre
almacén, libro e idioma';
/*!40101 SET character_set_client  = @saved_cs_client */;

```

```

--
-- Table structure for table `traduccion`
--

```

```

DROP TABLE IF EXISTS `traduccion`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client  = utf8 */;
CREATE TABLE `traduccion` (
  `codIdioma` varchar(5) NOT NULL,
  `codLibr` varchar(6) NOT NULL,
  PRIMARY KEY (`codIdioma`, `codLibr`),
  KEY `codIdioma_idx` (`codIdioma`),
  KEY `codLibr_idx` (`codLibr`),
  CONSTRAINT `codIdioma` FOREIGN KEY (`codIdioma`) REFERENCES
`idioma` (`codigoIdioma`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `codLibr` FOREIGN KEY (`codLibr`) REFERENCES `libro`
(`codigoLibro`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Relación entre el
libro y los idiomas en los que está escrito';
/*!40101 SET character_set_client  = @saved_cs_client */;

```

```

--
-- Table structure for table `traduce`
--

```

```

DROP TABLE IF EXISTS `traduce`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client  = utf8 */;
CREATE TABLE `traduce` (
  `codTrad` varchar(6) NOT NULL,
  `codTraduceLibro` varchar(6) NOT NULL,
  `codTraduceIdioma` varchar(5) NOT NULL,
  PRIMARY KEY (`codTrad`, `codTraduceLibro`, `codTraduceIdioma`),
  KEY `codTrad_idx` (`codTrad`),
  KEY `codTraduceLibro_idx` (`codTraduceLibro`),
  KEY `codTraduceIdioma_idx` (`codTraduceIdioma`),
  CONSTRAINT `codTrad` FOREIGN KEY (`codTrad`) REFERENCES
`traductor` (`codigoTraductor`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `codTraduceIdioma` FOREIGN KEY (`codTraduceIdioma`)
REFERENCES `idioma` (`codigoIdioma`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `codTraduceLibro` FOREIGN KEY (`codTraduceLibro`)
REFERENCES `libro` (`codigoLibro`) ON DELETE CASCADE ON UPDATE
CASCADE

```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Relación entre el
traductor libro e idioma';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `traductor`
--

DROP TABLE IF EXISTS `traductor`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `traductor` (
  `codigoTraductor` varchar(6) NOT NULL,
  `nacionalidadTraductor` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`codigoTraductor`),
  CONSTRAINT `codigoTraductor` FOREIGN KEY (`codigoTraductor`)
REFERENCES `personas` (`codigoPersonas`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Traduce los libros';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `vendedor`
--

DROP TABLE IF EXISTS `vendedor`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `vendedor` (
  `codigoVendedor` varchar(6) NOT NULL,
  `comision` double DEFAULT NULL,
  `fechaInicio` varchar(10) DEFAULT NULL,
  `codigoAlmacenVendedor` varchar(5) DEFAULT NULL,
  PRIMARY KEY (`codigoVendedor`),
  KEY `codigoAlmacenVendedor_idx` (`codigoAlmacenVendedor`),
  CONSTRAINT `codigoAlmacenVendedor` FOREIGN KEY
(`codigoAlmacenVendedor`) REFERENCES `almacen` (`codigoAlmacen`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `codigoVendedor` FOREIGN KEY (`codigoVendedor`)
REFERENCES `personas` (`codigoPersonas`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Vendedor que trabaja
en un almacén';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `vendedormediotiempo`
--

DROP TABLE IF EXISTS `vendedormediotiempo`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `vendedormediotiempo` (
  `codMedioTiempo` varchar(6) NOT NULL,
  `numHoras` int(11) DEFAULT NULL,
  `sueldoHora` double DEFAULT NULL,
  PRIMARY KEY (`codMedioTiempo`),
  CONSTRAINT `codMedioTiempo` FOREIGN KEY (`codMedioTiempo`)
REFERENCES `vendedor` (`codigoVendedor`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Vendedor que trabaja
por horas en un almacén';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `vendedortiemposcompleto`
--

DROP TABLE IF EXISTS `vendedortiemposcompleto`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;

```

```

/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `vendedor_tiempo_completo` (
  `codTiempoCompleto` varchar(6) NOT NULL,
  `salario` double DEFAULT NULL,
  PRIMARY KEY (`codTiempoCompleto`),
  CONSTRAINT `codTiempoCompleto` FOREIGN KEY (`codTiempoCompleto`)
REFERENCES `vendedor` (`codigoVendedor`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Vendedor que trabaja
tiempo completo en un almacén';
/*!40101 SET character_set_client = @saved_cs_client */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2017-04-16 13:59:16

```

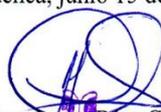
## Anexo B: Aprobación del protocolo de trabajo de titulación.

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay

### CERTIFICA:

Que, el Consejo de Facultad en sesión del 10 de junio de 2016, conoció la petición de la estudiante **NATALIA KARINA JARRIN CUESTA** con código 48884, quien presenta su trabajo de titulación denominado: **“IMPLEMENTACION DEL MODELO ENTIDAD RELACION EXTENDIDO”** previa a la obtención del título de Ingeniera de Sistemas y Telemática. El Consejo de Facultad acoge el informe de la Junta Académica y aprueba el diseño. Designa como *Director al ingeniero Oswaldo Merchán Manzano* y como miembros del *Tribunal Examinador a los ingenieros Francisco Salgado Arteaga y Marcos Orellana Cordero*. La peticionaria tiene un plazo de **SEIS MESES** para presentar su trabajo de titulación, esto es hasta el 10 de diciembre de 2016.

Cuenca, junio 15 de 2016



Dra. Jenny Ríos Coello  
Secretaria de la Facultad de  
Ciencias de la Administración

AZUAY  
FACULTAD DE  
ADMINISTRACION  
SECRETARIA

## Anexo C: Diseño del trabajo de titulación.

Universidad del Azuay  
Ingeniería de Sistemas y Telemática  
Metodología de la Investigación  
Propuesta de proyecto de titulación (capstone Project)



### 1. Datos generales

1.1. Nombre del estudiante: Natalia Karina Jarrín Cuesta.

1.1.1. Código: 48884.

1.1.2. Contacto: teléfono: 0992511252, correo electrónico: [natasha510@hotmail.com](mailto:natasha510@hotmail.com).

1.2. Director sugerido: Ing. Oswaldo Leonardo Merchán Manzano.

1.2.1. Contacto: [omerchan@uazuay.edu.ec](mailto:omerchan@uazuay.edu.ec).

1.3. Co-director sugerido:

1.3.1. Contacto:

1.4. Asesor metodológico: Ing. Francisco Salgado Arteaga.

1.5. Tribunal designado:

1.6. Aprobación:

1.7. Línea de Investigación de la carrera: Informática de computadoras.

1.7.1. Código UNESCO 1203.17 Informática

1.7.2. Tipo de trabajo: Proyecto de investigación

1.8. Área de estudio: Base de datos, programación.

1.9. Título propuesto: Implementación del Modelo Entidad Relación Extendido.

1.10. Estado del proyecto: El proyecto es nuevo y trata la investigación del modelo entidad relación extendido y de su implementación, el mismo es parte del estudio realizado por el Ing. Oswaldo Merchán.

### 2. Contenido

#### 2.1. Motivación de la investigación:

En la actualidad existen grandes cantidades de información la cual debe ser almacenada y administrada de forma eficiente y eficaz, por lo que la sociedad actual tiende a sistematizarla y estandarizarla. Una forma de lograr estos objetivos, es la utilización de sistemas computacionales y dentro de estos el uso de las bases de datos como repositorios para dicha información. Como estudio se propone el uso del modelo entidad relación extendido (EER) para diseñar la base de datos aplicada a un caso determinado, a la vez exponer cómo este modelo integra conceptos mejorados de los que permite el modelo entidad relación tradicional (ER).

## 2.2. Problemática:

Para administrar de forma efectiva la información dentro de una base de datos es necesario que la misma sea: diseñada de forma correcta, que la información y sus relaciones sea entendible; que el modelo utilizado sea de fácil comprensión para los involucrados. Si bien existe bibliografía especializada sobre los conceptos del modelo EER, no se encuentran casos de estudio en los cuales éste se aplique, ni su relación con un software de gestión de bases de datos, es por ello que se busca utilizar el EERM para el modelado e implementarlo en un sistema de gestión de base de datos.

## 2.3. Resumen:

El presente proyecto busca contribuir al modelado de las bases de datos a través del modelo entidad relación extendido, al exponer la funcionalidad del modelo en un caso de estudio determinado y su aplicación en un sistema de gestión de base de datos. Adicionalmente busca aplicar las características con las que cuenta el modelo EER y de esta forma, mejorar la comprensión del modelado de bases de datos, ya que, esta variación del modelo ER es de fácil entendimiento y se acerca más al lenguaje humano, gracias a su notación gráfica la información y sus relaciones se comprende de mejor manera.

El presente proyecto pretende sistematizar la información sobre el modelo EER, con ello aplicar dichos conceptos para definir de forma gráfica la estructura y funcionalidad de la base de datos y su aplicación en un SGBD.

## 2.4. Indagación exploratoria y base conceptual:

Al momento de diseñar una base de datos, los modelos conceptuales son una pieza vital dentro de las actividades, ya que por medio de estos se representa de forma gráfica los requerimientos de la misma. (Bajaj, 2010).

El modelo EER es un lenguaje que permite definir de forma gráfica la estructura y funcionalidad de una base de datos. Este modelo es una mejora del modelo entidad relación (ER) diseñado en 1976 por P.P Chen, usado como modelo base para el desarrollo de sistemas de bases de datos. Sin embargo en las décadas de los 80's y 90's aparecieron algunas extensiones para optimizarlo y mejorarlo. (Thalheim, 2009).

Aunque existe una gran variedad de modelos conceptuales que permiten realizar esta tarea, entre los preferidos por los diseñadores y analistas de sistemas se encuentran el modelo ER, que con sus mejoras y extensiones se obtiene el modelo EER. Debido a que, este modelo es de fácil aprendizaje, está más cerca del lenguaje humano que otros modelos y su notación gráfica permite un mejor entendimiento de la información. (Jain, Gore, & Singh, 2009).

Tanto el modelo ER como EER permiten representar estructuras complejas de la realidad por medio de entidades, es decir representan información del mundo real como personas, lugares, cosas, interacciones, entre otros por medio de entidades y relaciones. Sin embargo, el modelo EER puede representar clases también llamadas generalizaciones y subclases o especializaciones, lo que facilita modelar de forma más precisa la realidad del sistema que se desea desarrollar. (Bagui, 2009).



UNIVERSIDAD DEL  
AZUAY

## 2.5. Objetivo general:

Diseñar una base de datos usando el modelo entidad relación extendido e implementar en un sistema de gestión de base de datos (SGBD).

## 2.6. Objetivos específicos:

- Sistematizar información acerca del modelo EER.
- Definir el caso de estudio.
- Diseñar la base de datos.
- Implementar el modelo en un sistema de gestión de base de datos.

## 2.7. Metodología:

El presente proyecto es un estudio descriptivo y bibliográfico acerca del modelo entidad relación extendido. Para cumplir cada uno de los objetivos planteados se desarrollarán las siguientes actividades:

- Sistematizar información acerca del modelo entidad relación extendido.

Actividades:

Revisión de la bibliografía sobre el EER.

- Definir el caso de estudio.

Actividades:

Analizar la información sobre el caso de estudio.

- Diseñar la base de datos.

Actividades:

Realizar el diseño de la base de datos utilizando el modelo EER.

Reducir el modelo a tablas.

- Implementar el diseño en un sistema de gestión de base de datos.

Actividades:

Crear las tablas con sus restricciones de integridad de datos en SGBD.

Generar consultas y actualizaciones a la base de datos.

## 2.8. Alcances y resultados esperados:

- Sistematizar información acerca del modelo entidad relación extendido.

Entregables:

Obtener un marco teórico de información sobre el modelo EER y sus características.

- Definir el caso de estudio.

Entregables:

Enunciado sobre el caso de estudio, definiendo las clases que intervienen y sus relaciones.

- Diseñar la base de datos.

Entregables:

Modelo entidad relación extendido sobre el caso de estudio a desarrollar

Tablas de las clases y relaciones dentro del modelo.

Edición autorizada de 20.000 ejemplares  
Del 798.501 al 798.600

Nº

0748154

- Implementar el diseño en un sistema de gestión de base de datos.

Entregables:

Codificación en lenguaje de definición de datos y creación de consultas y mantenimiento en el lenguaje de manipulación de datos.

**2.9. Supuestos y riesgos:** No existen.

**2.10. Presupuesto:** No existe.

**2.11. Financiamiento:** El costo del proyecto será asumido en un 100% por la Autora.

**2.12. Esquema tentativo**

Capítulo Uno

El modelo entidad relación extendido y sus características

Capítulo Dos

Caso de estudio

Capítulo Tres

Modelo entidad relación extendido y generación de tablas

Capítulo Cuatro

Implementación en el gestor de bases de datos



2.14. Referencias:

- Bagui, S. (2009). Mapping OWL to the Entity Relationship and Extended Entity Relationship models. *Int. J. Knowledge and Web Intelligence*, 125-149.
- Bajaj, A. (2010). The Effect of Abstraction of Constructs in Data Models on Modeling Performance: An Exploratory Empirical Study. *Tusla: AMCIS Proceedings*.
- Jain, S. K., Gore, M. M., & Singh, G. (2009). An Experimental Study to Compare ER/EER and OO Models. *International Journal of Computing Science and Communication Technologies*, 221-228.
- Thalheim, B. (2009). *Extended Entity-Relationship Model*. Kiel.

2.15. Firma de responsabilidad:

  
Natalia Karina Jarrín Cuesta

2.16. Firma de responsabilidad:

  
Ing. Oswaldo Merchán

2.17. Fecha de entrega: Jueves 2 de junio de 2016