

**PRIMER INFORME DE LA BASE DE DATOS DEL SISTEMA LOAN  
MANAGER**

**POR  
JUAN AMARANTO**

**DOCENTE  
BRAYAN ARCOS**

**INSTITUTO TECNOLOGICO DEL PUTUMAYO  
TECNOLOGIA EN DESARROLLO DE SOFTWARE  
DESARROLLO DE BASE DE DATOS  
MOCOA PUTUMAYO**

**13/09/2024**

Introducción .....	3
Metodología.....	3
Herramientas Utilizadas .....	4
Procedimientos.....	4
Desarrollo de consultas:.....	4
Desarrollo.....	5
Esquema de la Base de Datos .....	5
Relaciones entre las Tablas.....	6
Consideraciones de Diseño .....	6
ConsultasSQL.....	7
1. Consultas Realizadas: Detalles de las Consultas Ejecutadas .....	1
2. Resultados de Consultas: Salidas Obtenidas de las Consultas .....	1
3. Explicación de Consultas: Explicación de la Lógica Detrás de Cada Consulta	1
Diseño.....	8
1. Modelo de Datos: Diagramas Entidad-Relación (ERD), Normalización, Cardinalidad, ER Simbolización.....	9
2. Consideraciones de Diseño: Elección de Claves Primarias, Relaciones entre Tablas, Nombre de la Base de Datos y su Descripción, Entidades .....	11
Análisis y Discusión .....	13
Interpretación de Resultados .....	13
Referencias.....	13

## Introducción

Este informe tiene como objetivo documentar el desarrollo y la implementación del sistema LOAN MANAGER, una solución diseñada para gestionar de manera eficiente los procesos de préstamo y cobro a nivel local. Al enfocarnos en el ámbito local, buscamos proporcionar una herramienta útil para simplificar las transacciones financieras y fortalecer las relaciones comerciales en nuestra comunidad.

El sistema LOAN MANAGER fue desarrollado utilizando MySQL como base de datos, y este informe profundizará en el diseño de la base de datos, las consultas utilizadas y las estrategias de optimización implementadas. Nuestro objetivo es no solo presentar la solución técnica, sino también explicar las razones detrás de las decisiones de diseño y los beneficios que esta herramienta aporta a los usuarios. Al compartir este conocimiento, buscamos contribuir al avance de la comunidad de desarrolladores y fomentar el uso de tecnologías de base de datos para mejorar la gestión financiera.

## Metodología

El desarrollo del sistema LOAN MANAGER se llevó a cabo siguiendo una metodología que combinó la investigación teórica con la práctica. A continuación, se detallan las herramientas utilizadas y los procedimientos seguidos en cada etapa del proyecto.

#### Herramientas Utilizadas

- **MySQL:** Sistema de gestión de bases de datos relacionales utilizado para almacenar y gestionar la información del sistema. MySQL fue seleccionado por su eficiencia, fiabilidad y amplia comunidad de usuarios.
- **Workbench:** Herramienta gráfica de administración de bases de datos que facilitó el diseño, la creación y la gestión de la base de datos de LOAN MANAGER. Workbench permitió visualizar de manera intuitiva las tablas, las relaciones y las consultas.
- **draw.io:** Herramienta de diagramación utilizada para crear diagramas de entidad-relación (ER) que representaron visualmente la estructura de la base de datos. Los diagramas ER fueron fundamentales para comprender y comunicar el diseño de la base de datos a otros miembros del equipo.

#### Procedimientos

##### 1. Investigación:

- **Análisis del sector:** Se realizó una investigación exhaustiva sobre las prácticas comunes de gestión de préstamos y cobros en la ciudad, identificando los principales desafíos y necesidades de los usuarios.

##### 2. Diseño de la base de datos:

- **Identificación de entidades:** Se identificaron las entidades relevantes para el sistema, como clientes, préstamos, pagos, etc.
- **Definición de atributos:** Se determinaron los atributos de cada entidad y se establecieron las relaciones entre ellas.
- **Creación de diagramas ER:** Se utilizaron diagramas ER para representar visualmente la estructura de la base de datos y validar el diseño.

##### 3. Implementación de la base de datos:

- **Creación de las tablas:** Se crearon las tablas en MySQL siguiendo las especificaciones de los diagramas ER.
- **Definición de las claves primarias y foráneas:** Se establecieron las claves primarias y foráneas para garantizar la integridad de los datos.
- **Creación de índices:** Se crearon índices en los campos más utilizados para mejorar el rendimiento de las consultas.

#### Desarrollo de consultas:

- **Consultas básicas:** Se desarrollaron consultas para insertar, actualizar y eliminar datos, así como para realizar consultas simples.
- **Consultas complejas:** Se crearon consultas más elaboradas para generar reportes personalizados y realizar análisis de datos.

- **Optimización de consultas:** Se analizaron las consultas para identificar cuellos de botella y se aplicaron técnicas de optimización, como la utilización de índices y la reducción de la cantidad de datos procesados.

## Desarrollo

La base de datos **loan\_manager** ha sido diseñada como una solución integral para gestionar los procesos de préstamo y cobro en un entorno local. Su estructura modular y flexible permite almacenar y gestionar de manera eficiente una amplia variedad de información relacionada con clientes, negocios, préstamos y transacciones.

### Esquema de la Base de Datos

La base de datos se compone de las siguientes tablas interrelacionadas:

- **Direccion:** Contiene la información geográfica de las direcciones, incluyendo barrio, calle, carrera y una descripción detallada. Esta tabla sirve como referencia para ubicar tanto a los clientes como a los negocios.
- **Negocio:** Almacena los datos de las entidades comerciales involucradas en los préstamos, como su nombre, teléfono, descripción, propietario, dirección y estado legal. Esta tabla permite establecer una relación entre los clientes y sus lugares de trabajo.
- **Trabajo:** Describe el tipo de actividad o ocupación que desarrolla un cliente en un negocio específico.
- **Usuario:** Guarda la información de los usuarios del sistema, incluyendo sus roles (administrador, cobrador, etc.), datos de contacto y credenciales de acceso. Los usuarios pueden ser tanto empleados de la entidad financiera como los propios clientes.
- **Cliente:** Contiene los datos personales y de contacto de los clientes, así como información sobre su situación laboral y financiera. Esta tabla es el núcleo de la base de datos, ya que cada préstamo está asociado a un cliente.
- **Prestamo:** Almacena los detalles de cada préstamo otorgado, incluyendo el monto, la tasa de interés, las fechas de inicio y vencimiento, el plan de pagos, el saldo pendiente y el estado actual del préstamo (activo, cancelado, finalizado).
- **Pago:** Registra los pagos realizados por los clientes, relacionándolos con el préstamo correspondiente. Esta tabla permite llevar un control detallado del historial de pagos de cada cliente.
- **Garantia:** Almacena la información sobre las garantías ofrecidas por los clientes para respaldar los préstamos, como bienes inmuebles, vehículos o avales.

- **CobradorPrestamo:** Establece una relación entre los cobradores y los préstamos asignados a cada uno, permitiendo llevar un seguimiento del proceso de cobro.

### Relaciones entre las Tablas

Las tablas de la base de datos **loan\_manager** están interconectadas a través de **claves foráneas**, creando un conjunto de relaciones que reflejan los vínculos existentes entre los datos. Estas relaciones permiten garantizar la integridad de la información y facilitan la realización de consultas complejas.

A continuación, se detallan algunas de las relaciones más importantes:

- **Un cliente puede tener una dirección asociada:** La tabla **Cliente** se relaciona con la tabla **Direccion** a través de la clave foránea DireccionID. Esto permite conocer la dirección física de cada cliente.
- **Un cliente puede estar vinculado a un negocio:** La tabla **Cliente** también se relaciona con la tabla **Negocio** a través de la clave foránea NegocioID. Esta relación indica el lugar de trabajo del cliente, si aplica.
- **Un préstamo está asociado a un cliente:** La tabla **Prestamo** se relaciona con la tabla **Cliente** a través de la clave foránea ClienteID. Esto establece un vínculo entre cada préstamo y el cliente al que se otorgó.
- **Un préstamo puede tener múltiples pagos:** La tabla **Pago** se relaciona con la tabla **Prestamo** a través de la clave foránea PrestamoID. Esta relación permite registrar todos los pagos realizados por un cliente para un préstamo específico.
- **Un préstamo puede tener múltiples garantías:** La tabla **Garantia** se relaciona con la tabla **Prestamo** a través de la clave foránea PrestamoID. Esta relación indica las garantías ofrecidas por el cliente para respaldar un préstamo.
- **Un cobrador puede estar asignado a varios préstamos:** La tabla **CobradorPrestamo** se relaciona con las tablas **Usuario** (a través de CobradorID) y **Prestamo** (a través de PrestamoID). Esta relación permite asignar a cada cobrador los préstamos que debe gestionar.
- **Un negocio puede tener múltiples trabajos asociados:** La tabla **Trabajo** se relaciona con la tabla **Negocio** a través de la clave foránea NegocioID. Esto indica los diferentes tipos de trabajo que se realizan en un negocio.

### Ejemplo de Relación:

Consideremos la relación entre las tablas **Cliente** y **Prestamo**. Cada cliente puede tener múltiples préstamos, pero un préstamo solo puede estar asociado a un cliente. Esta es una relación de **uno a muchos**. La clave primaria de la tabla **Cliente** (ClienteID) se convierte en una clave foránea en la tabla **Prestamo**.

### Consideraciones de Diseño

Al diseñar esta base de datos, se tuvieron en cuenta los siguientes aspectos:

- **Normalización:** Se aplicaron las normas de normalización para evitar la redundancia de datos y garantizar la integridad de la información.
- **Flexibilidad:** La estructura de la base de datos permite adaptarse a futuras ampliaciones y cambios en los requisitos del negocio.
- **Rendimiento:** Se utilizaron índices para acelerar las consultas y mejorar el rendimiento del sistema.
- **Seguridad:** Se implementaron medidas de seguridad para proteger la información confidencial de los clientes.

## Consultas Típicas

La base de datos **loan\_manager** permite realizar una amplia variedad de consultas, como:

- Obtener la lista de clientes con préstamos vencidos.
- Calcular el saldo total de préstamos por cliente.
- Generar reportes de cobranza.
- Analizar la rentabilidad de los préstamos.

## Consultas SQL

1. Consulta simple, uso de una condición.

```
-- Consulta simple
SELECT * FROM Cliente WHERE Cliente.ClienteID = 1;
```

Result Grid								
Filter Rows: <input type="text"/>								
#	ClienteID	Nombre	Apellido	Documento	Telefono	DireccionID	NegocioID	TrabajoID
1	1	Jorge	Martínez	2233445566	3002233445	1	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Consulta avanzada, uso de múltiples condiciones.

-- Consulta avanzada

```
SELECT * FROM Prestamo WHERE Prestamo.PrestamoID <> 0
AND Prestamo.Monto > 1000 AND Prestamo.Estado = "Activo";
```

ClienteID	FiadorID	Monto	TasaInteres	FechaInicio	FechaFin	MontoPagc	FrecuenciaPagc	Saldo	Tipo	Estado
1	2	5000.00	5.00	2024-01-01	2024-12-31	500.00	Mensual	4500.00	1	Activo
2	3	3000.00	4.50	2024-02-01	2024-08-01	500.00	Quincenal	2500.00	0	Activo
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 3. Uso de LIKE:

-- Uso de LIKE

```
SELECT * FROM Cliente
WHERE LOWER(Cliente.Nombre) LIKE '_o%';
```

#	ClienteID	Nombre	Apellido	Documento	Telefono	DireccionID	NegocioID	TrabajoID
1	1	Jorge	Martínez	2233445566	3005648123	1	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 4. Uso de BETWEEN:

-- Uso de BETWEEN

```
SELECT * FROM Prestamo WHERE Prestamo.ID IS NOT NULL
AND Prestamo.Saldo BETWEEN 1000 AND 3000
```

#	PrestamoID	ClienteID	FiadorID	Monto	TasaInteres	FechaInicio	FechaFin	MontoPagc	FrecuenciaPagc	Saldo
1	2	2	3	3000.00	4.50	2024-02-01	2024-08-01	500.00	Quincenal	2500.00
2	3	3	1	1500.00	6.00	2024-03-01	2024-09-01	300.00	Mensual	1200.00
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

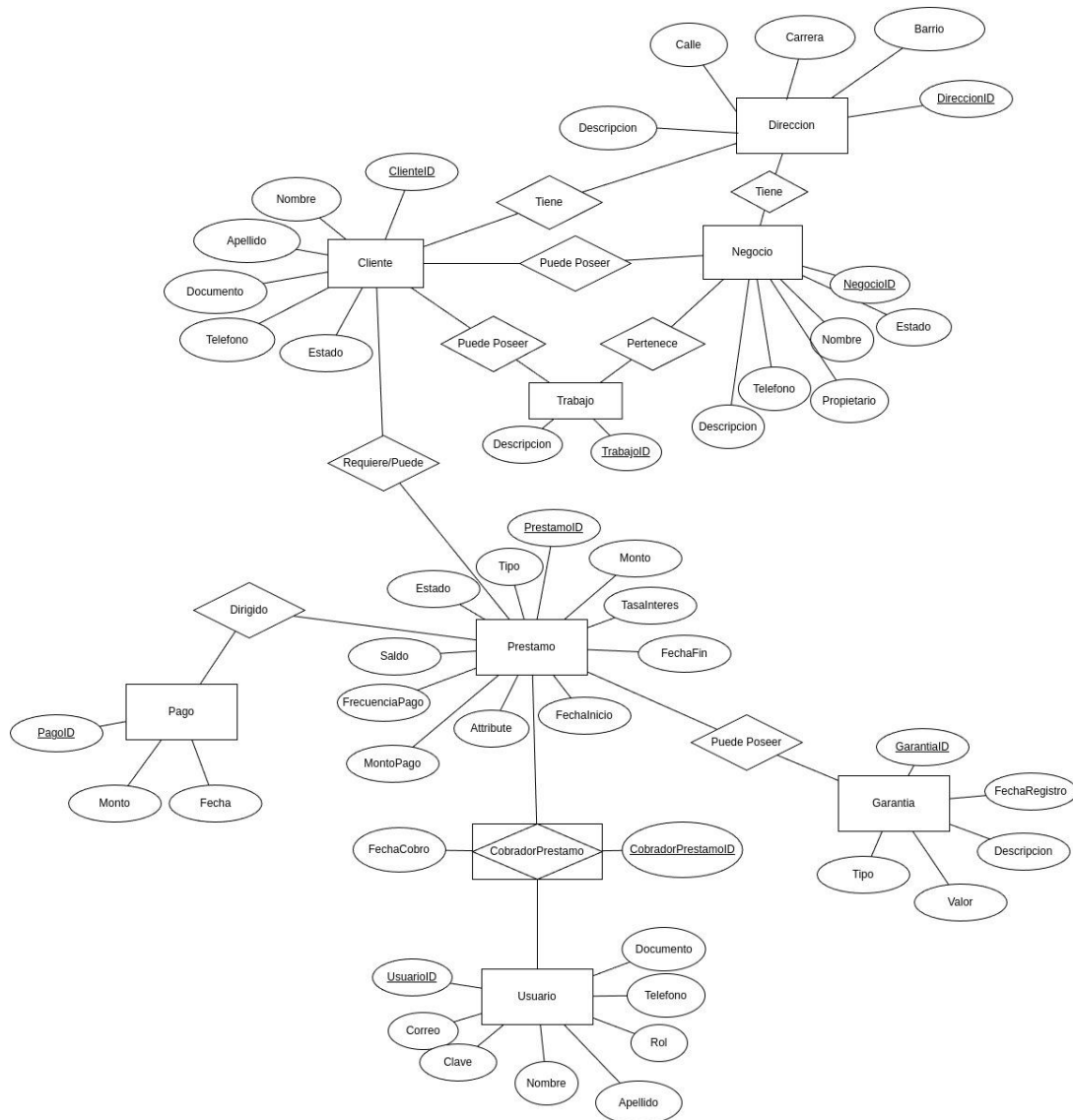
Diseño



## 1. Modelo de Datos: Diagramas Entidad-Relación (ERD), Normalización, Cardinalidad, ER Simbolización

### Modelo Entidad-Relación (ERD)

El diagrama Entidad-Relación (ERD) ilustra la estructura de nuestra base de datos y las relaciones entre las diferentes entidades. Este modelo muestra cómo las tablas están conectadas y ayuda a comprender la organización y las interacciones de los datos.



### Normalización

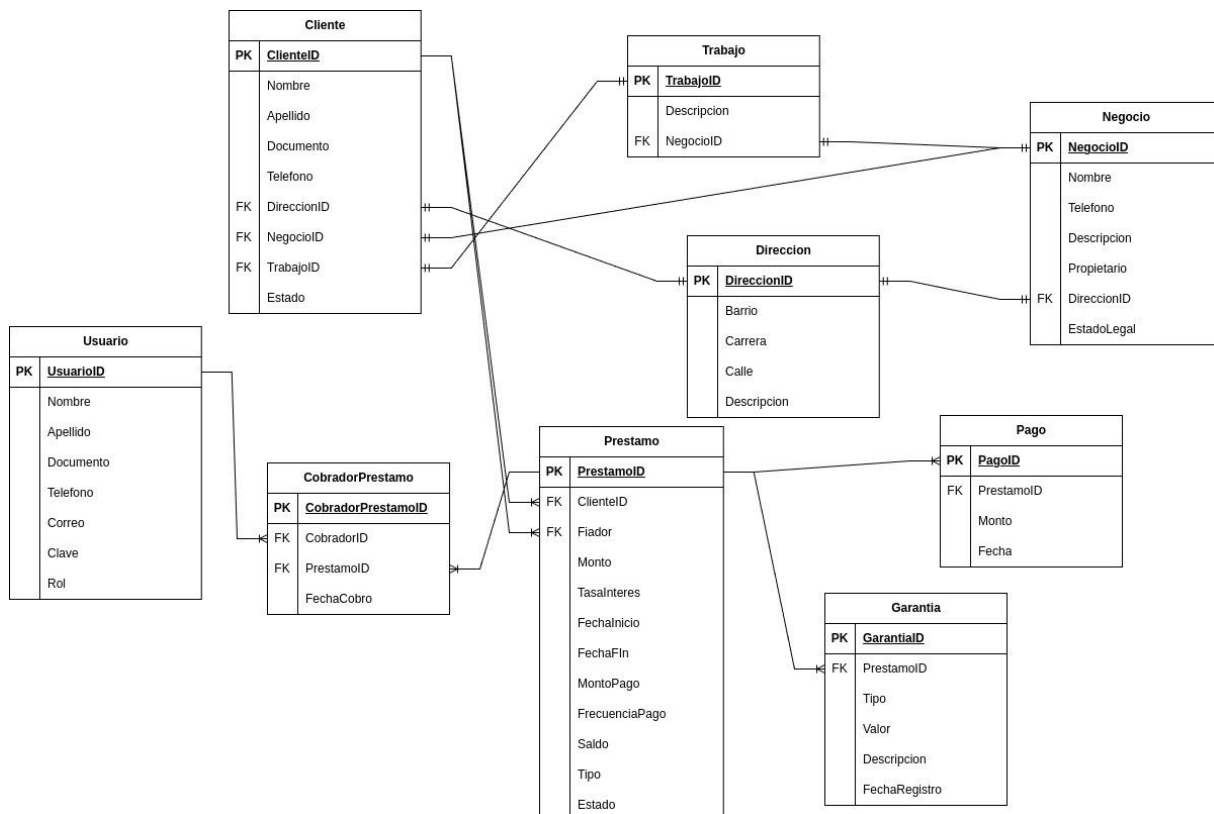
La base de datos ha sido diseñada siguiendo principios de normalización para asegurar la eficiencia y la integridad de los datos. La normalización se ha aplicado para evitar redundancias y dependencias anómalas. Las formas normales consideradas incluyen:

- **Primera Forma Normal (1NF):** Cada tabla tiene una clave primaria única y todos los atributos contienen solo valores atómicos.
- **Segunda Forma Normal (2NF):** Todos los atributos no clave dependen completamente de la clave primaria.
- **Tercera Forma Normal (3NF):** Todos los atributos no clave dependen directamente de la clave primaria, sin dependencias transitivas.

## Cardinalidad y Simbolización ER

Los tipos de cardinalidad en nuestro modelo son:

- 1. Cliente (1) ---- (N) Tiene (N) ---- (1) Prestamo
- 2. Prestamo (1) ---- (N) Tiene (N) ---- (1) Garantia
- 3. Prestamo (1) ---- (N) Registrado en (N) ---- (1) CobradorPrestamo
- 4. Usuario (1) ---- (N) Registra (N) ---- (1) CobradorPrestamo
- 5. Cliente (1) ---- (N) Tiene (N) ---- (1) Pago
- 6. Negocio (1) ---- (N) Tiene (N) ---- (1) Cliente
- 7. Trabajo (1) ---- (N) Asociado a (N) ---- (1) Cliente
- 8. Direccion (1) ---- (N) Ubica (N) ---- (1) Negocio
- 9. Direccion (1) ---- (N) Ubica (N) ---- (1) Cliente

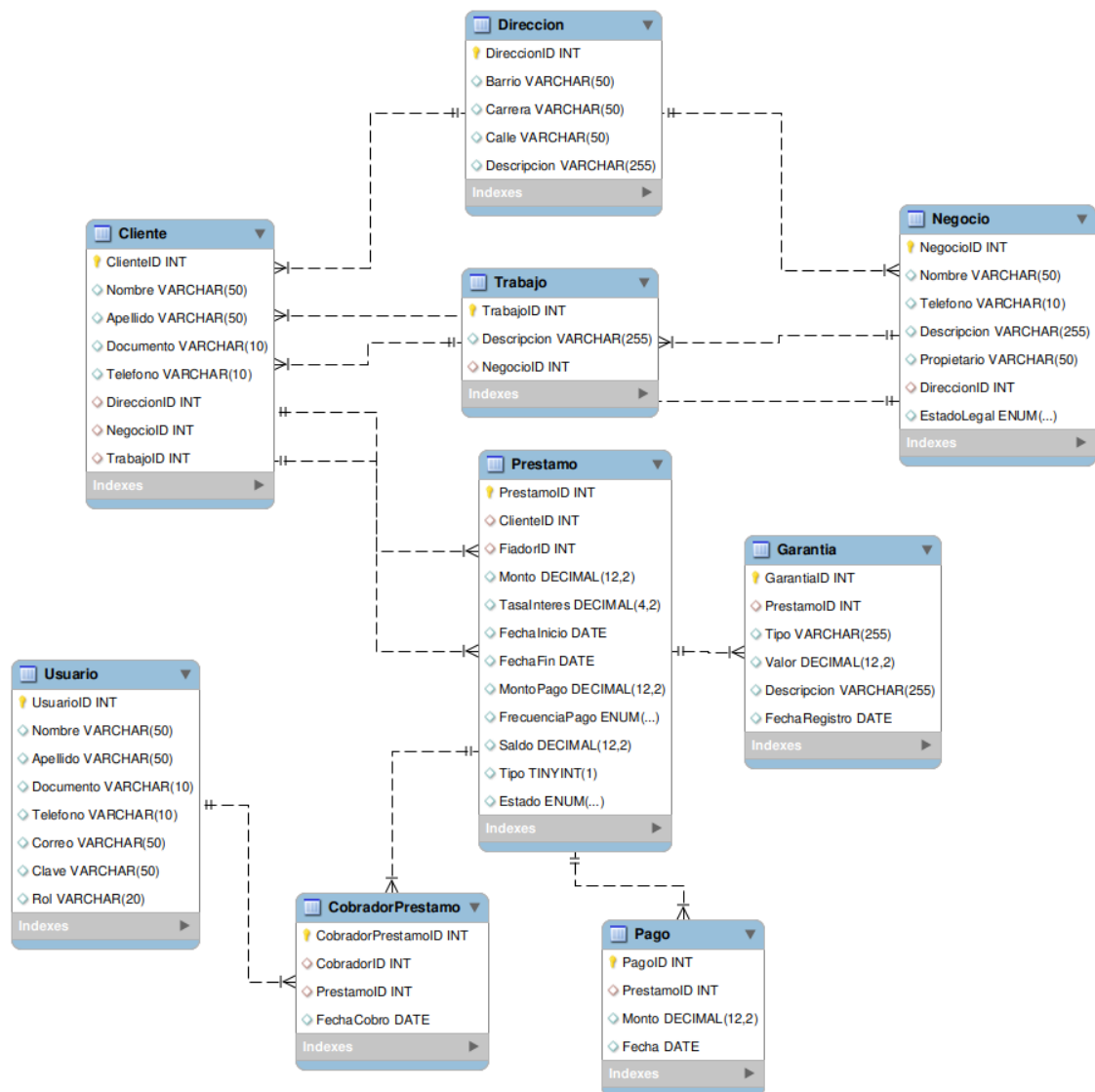


## 2. Consideraciones de Diseño: Elección de Claves Primarias, Relaciones entre Tablas, Nombre de la Base de Datos y su Descripción, Entidades

### Elección de Claves Primarias

Las claves primarias se han elegido para garantizar la unicidad de cada registro en las tablas. Las claves primarias en la base de datos son:

- **DireccionID:** Identificador único para cada dirección en la tabla Direccion.
- **NegocioID:** Identificador único para cada negocio en la tabla Negocio.
- **TrabajoID:** Identificador único para cada trabajo en la tabla Trabajo.
- **UsuarioID:** Identificador único para cada usuario en la tabla Usuario.
- **ClienteID:** Identificador único para cada cliente en la tabla Cliente.
- **PrestamoID:** Identificador único para cada préstamo en la tabla Prestamo.
- **PagoID:** Identificador único para cada pago en la tabla Pago.
- **GarantiaID:** Identificador único para cada garantía en la tabla Garantia.
- **CobradorPrestamoID:** Identificador único para cada registro de cobro en la tabla CobradorPrestamo.



## Entidades

Las entidades de la base de datos son:

- **Direccion:** Contiene la información sobre las direcciones de clientes y negocios.
- **Negocio:** Información sobre negocios que pueden estar asociados con clientes.
- **Trabajo:** Detalles sobre los trabajos realizados por los clientes en distintos negocios.
- **Usuario:** Información sobre los usuarios del sistema, incluidos los roles como cobradores y administradores.
- **Cliente:** Datos sobre los clientes que solicitan préstamos.
- **Prestamo:** Información sobre los préstamos otorgados a clientes.
- **Pago:** Registra los pagos realizados para los préstamos.
- **Garantia:** Información sobre las garantías proporcionadas para asegurar los préstamos.

- **CobradorPrestamo:** Relaciona a los cobradores con los préstamos que gestionan.

## Análisis y Discusión

### Interpretación de Resultados

Los resultados obtenidos del sistema LOAN MANAGER reflejan una alineación efectiva con los objetivos del proyecto:

#### **Eficiencia en la Gestión de Préstamos y Pagos**

- **Resultado:** Consultas rápidas y precisas sobre préstamos, pagos y garantías.
- **Relación con los Objetivos:** Mejora la eficiencia en la gestión financiera, facilitando el seguimiento de transacciones y fortaleciendo las relaciones comerciales locales.

#### **Precisión en el Seguimiento de Garantías y Cobradores**

- **Resultado:** Asignación precisa de garantías y cobradores a préstamos.
- **Relación con los Objetivos:** Asegura transparencia y precisión en la gestión de préstamos, apoyando la transparencia financiera en la comunidad.

#### **Optimización y Rendimiento del Sistema**

- **Resultado:** Consultas optimizadas con buen rendimiento incluso con grandes volúmenes de datos.
- **Relación con los Objetivos:** Permite una gestión eficiente de datos, cumpliendo con el objetivo de ofrecer una herramienta efectiva para la gestión de préstamos y cobros.

#### **Facilidad de Uso y Accesibilidad**

- **Resultado:** Interfaz amigable y clara para los usuarios.
- **Relación con los Objetivos:** Facilita la gestión financiera para los usuarios locales, simplificando las transacciones y mejorando la experiencia general.

## Referencias

- <https://github.com/jcamilo-am/mysql-Juan-Amaranto>
- Asociación de Cobros y Prestamos del Putumayo.