

# Proyecto SQL

Coderhouse



Gianfranco Ganicoche

# INTRODUCCIÓN

En este documento, se realizará una detallada descripción de todos los pasos necesarios para la creación de una base de datos relacional, pensada y diseñada para llevar adelante la operativa diaria de una empresa financiera que preste dinero a distintos clientes, así como tener control sobre las distintas variables y analizar patrones de comportamiento.

Además, para facilitar, hacer más eficiente y segura la manipulación de la base de datos, se crearon Funciones, Vistas, Store Procedures y Triggers, de modo que cualquier usuario pueda adentrarse en este proyecto sin arriesgar la integridad de las BD.

Hoy en día, es vital para cualquier empresa contar con información clave para la toma de decisiones, por eso almacenarla y tenerla organizada de forma útil y funcional se vuelve vital para el funcionamiento general del negocio.

## OBJETIVO

El objetivo principal de esta base de datos es tener toda la información relacionada a la solicitud de créditos, qué ocurre antes, qué ocurre durante y qué ocurre después del proceso. Es por esto que se almacenan datos del cliente, de los créditos y de los empleados, para tener una visión global del funcionamiento de la empresa. También se almacenarán los datos de los Canales por los que se pueden solicitar los créditos, o la forma en la que estos se pagan.

En un área como es la de los préstamos, se vuelve vital también almacenar la información sobre los rechazos, por lo que las solicitudes que no son finalizadas o que son rechazadas también quedarán almacenadas en las tablas.

## HERRAMIENTAS UTILIZADAS

- **Sistema de Gestión de Bases de Datos:**

El sistema de gestión de base de datos utilizado fue MySQL, fundamental para todo lo relacionado la creación, manipulación y administración de la BD.

- **Lenguaje de Programación SQL:**

Con el objetivo de interactuar con la BD, se utilizó el lenguaje de programación SQL. Con este, se desarrollaron funciones, vistas, store procedures y triggers, entre otros.

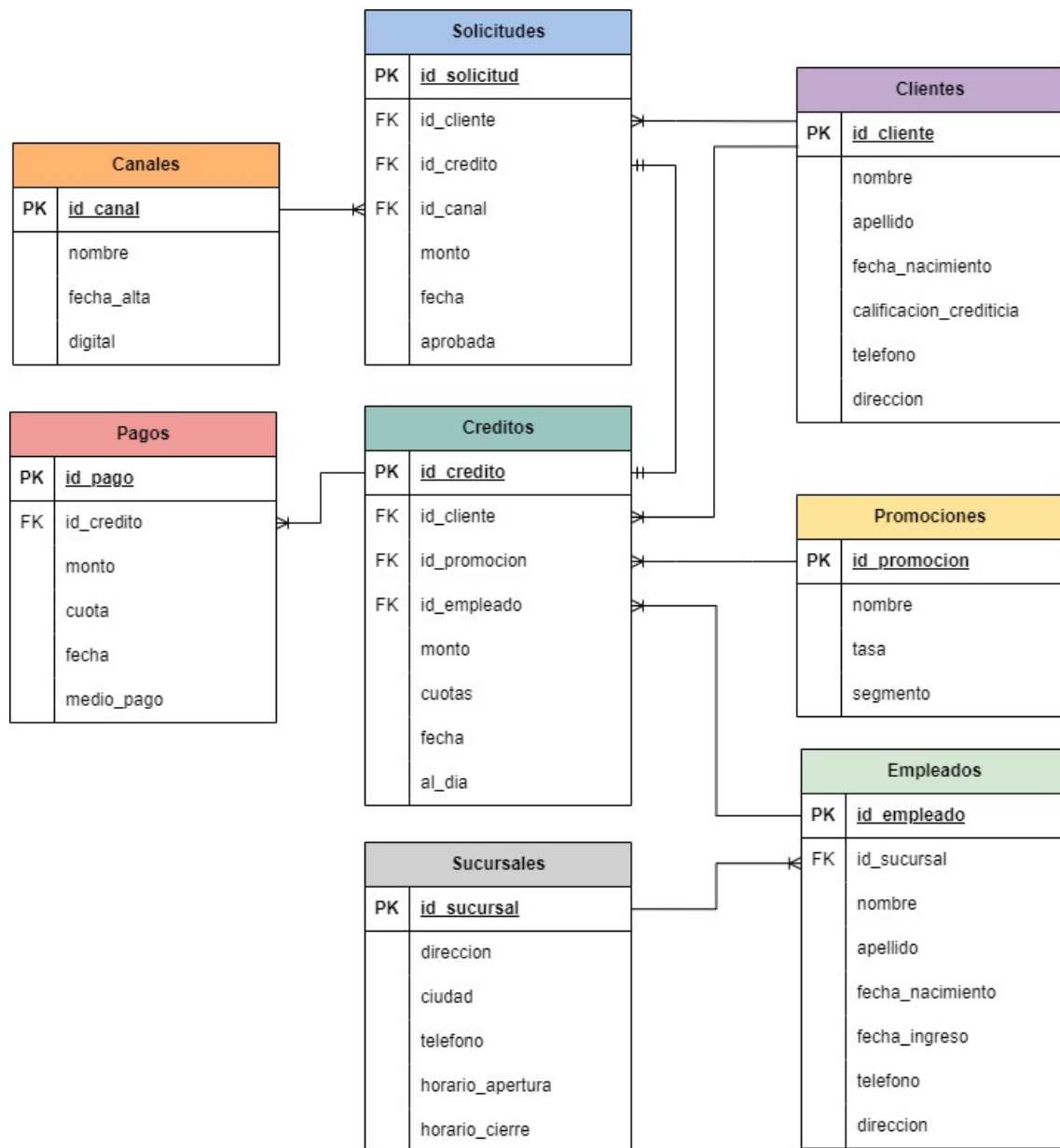
- **Herramientas de Modelado de Datos:**

Luego de tener el diseño de la base de datos, y para dejarla plasmada en un diagrama se utilizó DRAW.IO, una plataforma para diseñar distintos esquemas, flujos y tablas, de forma muy intuitiva y amigable con el usuario.

- **GitHub:**

Por último, para almacenar los distintos pasos del proyecto, se utilizó un repositorio de GitHub, creado especialmente para el seguimiento de este curso.

## DIAGRAMA ENTIDAD – RELACIÓN



## DESCRIPCIÓN DE TABLAS

Tabla	Campo	Nombre del campo	Llave	Tipo de dato	Valor SQL
Clientes	id_cliente	Número identificador del cliente	Primaria	Número entero	int
	nombre	Nombre del cliente		Cadena de caracteres	varchar (30)
	apellido	Apellido del cliente		Cadena de caracteres	varchar (30)
	fecha_nacimiento	Fecha de nacimiento del cliente		Fecha	date
	calificacion_crediticia	Calificacion crediticia del cliente		Número entero	int
	direccion	Direccion del cliente		Cadena de caracteres	varchar (100)
	telefono	Telefono del cliente		Cadena de caracteres	varchar (30)

Tabla	Campo	Nombre del campo	Llave	Tipo de dato	Valor SQL
Creditos	id_credito	Número identificador del credito	Primaria	Número entero	int
	id_cliente	Número identificador del cliente	Foránea	Número entero	int
	id_promocion	Número identificador de la promoción	Foránea	Número entero	int
	id_empleado	Número identificador del empleado que autorizó el crédito	Foránea	Número entero	int
	id_solicitud	Número identificador de la solicitud	Foránea	Número entero	int
	monto	Monto del crédito		Valor monetario	money
	cuotas	Cuotas del crédito		Número entero	int
	fecha	Fecha del crédito		Fecha y hora	datetime
	al_dia	El crédito está al día?		True or False	boolean

Tabla	Campo	Nombre del campo	Llave	Tipo de dato	Valor SQL
Solicitudes	id_solicitud	Número identificador de la solicitud	Primaria	Número entero	int
	id_cliente	Número identificador del cliente	Foránea	Número entero	int
	id_canal	Número identificador del canal	Foránea	Número entero	int
	monto	Monto solicitado		Valor monetario	money
	fecha	Fecha de la solicitud		Fecha y hora	datetime
	aprobada	La solicitud fue aprobada?		True or False	boolean

Tabla	Campo	Nombre del campo	Llave	Tipo de dato	Valor SQL
Pagos	id_pago	Número identificador del pago	Primaria	Número entero	int
	id_credito	Número identificador del credito	Foránea	Número entero	int
	monto	Monto pago		Número decimal	money
	cuota	Cuota paga		Número entero	int
	fecha	Fecha de pago		Fecha y hora	datetime
	medio_pago	Medio de pago		Cadena de caracteres	varchar (30)

Tabla	Campo	Nombre del campo	Llave	Tipo de dato	Valor SQL
Promociones	id_promocion	Número identificador de la promoción	Primaria	Número entero	int
	nombre	Nombre de la promoción		Cadena de caracteres	varchar (30)
	tasa	Tasa de interés de la promoción		Número decimal	decimal
	segmento	Segmento al que apunta la promoción		Cadena de caracteres	varchar (30)

Tabla	Campo	Nombre del campo	Llave	Tipo de dato	Valor SQL
Canales	id_canal	Número identificador del canal	Primaria	Número entero	int
	nombre	Nombre del canal		Cadena de caracteres	varchar (30)
	fecha_alta	Fecha de alta del canal		Fecha y hora	datetime
	digital	Es un canal digital?		True or False	boolean

Tabla	Campo	Nombre del campo	Llave	Tipo de dato	Valor SQL
Empleados	id_empleado	Número identificador del empleado	Primaria	Número entero	int
	id_sucursal	Número identificador de la sucursal donde trabaja el empleado	Foránea	Número entero	int
	nombre	Nombre del empleado		Cadena de caracteres	varchar (30)
	apellido	Apellido del empleado		Cadena de caracteres	varchar (30)
	fecha_nacimiento	Fecha de nacimiento del empleado		Fecha	date
	fecha_ingreso	Fecha de ingreso a la empresa del empleado		Fecha	date
	direccion	Dirección del empleado		Cadena de caracteres	varchar (100)
	telefono	Teléfono del empleado		Cadena de caracteres	varchar (30)

Tabla	Campo	Nombre del campo	Llave	Tipo de dato	Valor SQL
Sucursales	id_sucursal	Número identificador de la sucursal	Primaria	Número entero	int
	direccion	Dirección de la sucursal		Cadena de caracteres	varchar (100)
	ciudad	Ciudad de la sucursal		Cadena de caracteres	varchar (30)
	telefono	Teléfono de la sucursal		Cadena de caracteres	varchar (30)
	horario_apertura	Horario de apertura de la sucursal		Hora	time
	horario_cierre	Horario de cierre de la sucursal		Hora	time

# SCRIPTS SQL

- **Creación de Tablas:**

<https://github.com/gianfrancoganicoche/CursoSQL/blob/main/Script%20Creacion%20de%20Tablas%20-%20Entrega%201.txt>

- **Insert en las Tablas:**

<https://github.com/gianfrancoganicoche/CursoSQL/blob/main/Script%20Inserts%20-%20Entrega%202.txt>

- **Creación de Vistas:**

<https://github.com/gianfrancoganicoche/CursoSQL/blob/main/Script%20Vistas%20-%20Entrega%203.txt>

- **Funciones:**

<https://github.com/gianfrancoganicoche/CursoSQL/blob/main/Script%20Funciones%20-%20Entrega%204.txt>

- **Store Procedures:**

<https://github.com/gianfrancoganicoche/CursoSQL/blob/main/Script%20Store%20procedures%20-%20Entrega%205.txt>

- **Triggers:**

<https://github.com/gianfrancoganicoche/CursoSQL/blob/main/Script%20Triggers%20-%20Entrega%206.txt>



## Creación de vistas:

### 1. Vista Solicitudes rechazadas:

- **Objetivo:** Proporcionar una vista que muestre información detallada sobre las solicitudes de crédito que han sido rechazadas.
- **Tablas involucradas:** clientes, solicitudes, canales.
- **Campos mostrados:** Nombre del cliente, apellido, fecha de nacimiento, calificación crediticia, teléfono, monto solicitado, nombre del canal, y fecha de la solicitud.

### 2. Vista Solicitudes aprobadas:

- **Objetivo:** Proporcionar una vista que muestre información detallada sobre las solicitudes de crédito que han sido aprobadas.
- **Tablas involucradas:** clientes, solicitudes, canales.
- **Campos mostrados:** Nombre del cliente, apellido, fecha de nacimiento, calificación crediticia, teléfono, monto solicitado, nombre del canal, y fecha de la solicitud.

### 3. Vista Créditos mayores a \$20.000

- **Objetivo:** Proporcionar una vista que muestre información detallada sobre los créditos concedidos con un monto superior a \$20.000.
- **Tablas involucradas:** creditos, clientes, promociones.
- **Campos mostrados:** Nombre del cliente, apellido, monto del crédito, número de cuotas, tasa de interés, fecha de la concesión, segmento de la promoción y calificación crediticia del cliente.

#### *4. Vista Información sobre Sucursales:*

- **Objetivo:** Proporcionar una vista que muestre información sobre las sucursales, incluyendo la cantidad de empleados en cada sucursal.
- **Tablas involucradas:** sucursales, empleados.
- **Campos mostrados:** Ciudad, dirección, teléfono, horario de apertura, horario de cierre y cantidad de empleados en cada sucursal.

#### *5. Vista de Ventas por cada Promoción:*

- **Objetivo:** Proporcionar una vista que muestre información sobre las ventas realizadas para cada promoción.
- **Tablas involucradas:** creditos, promociones.
- **Campos mostrados:** ID de la promoción, nombre de la promoción, monto total vendido en créditos y cuotas promedio.

## Funciones:

### 1. *Función para calcular la tasa de interés*

- **Objetivo:** Esta función tiene como objetivo calcular la tasa de interés para un crédito, basándose en el número de cuotas y el monto del crédito.
- **Datos o tablas manipuladas:** No manipula directamente ninguna tabla. Utiliza los parámetros de entrada (cuotas y monto) para realizar cálculos y devuelve un valor decimal que representa la tasa de interés.

### 2. *Función para calcular la edad:*

- **Objetivo:** La función tiene como objetivo calcular la edad de un cliente específico, proporcionando el ID del cliente como parámetro de entrada.
- **Datos o tablas manipuladas:** Accede a la tabla clientes para obtener la fecha de nacimiento del cliente usando el ID proporcionado. No realiza modificaciones en la base de datos, solo extrae información.

## Store Procedures:

### 1. Store Procedure para agregar un Canal de Venta

- **Objetivo/Beneficio:** Este procedimiento almacenado tiene como objetivo agregar un nuevo canal de venta a la base de datos. Al proporcionar un ID de canal, un nombre y un indicador digital, se inserta una nueva fila en la tabla canales con la fecha actual y la información proporcionada.
- **Tablas involucradas:** canales.
- **Interacción:** Interactúa directamente con la tabla canales para insertar la nueva información del canal.
- **Retorno:** Devuelve la información completa del canal recién insertado, ordenada por el ID del canal de manera descendente.

### 2. Store Procedure para ordenar una tabla

- **Objetivo/Beneficio:** Este procedimiento almacenado tiene como objetivo permitir la ordenación dinámica de la tabla credits según un campo específico y un tipo de orden (ascendente o descendente).
- **Tablas involucradas:** credits.
- **Interacción:** Interactúa con la tabla credits para recuperar los datos ordenados según los parámetros proporcionados.
- **Parámetros:** Se le proporciona el nombre del campo por el cual se desea ordenar (campo\_orden) y el tipo de orden (tipo\_orden).
- **Dinámico:** Utiliza la creación dinámica de SQL para construir y ejecutar la consulta SQL en función de los parámetros proporcionados.

Estos Store Procedures proporcionan funcionalidades específicas que pueden ser útiles en el contexto del proyecto, como agregar canales de venta de manera eficiente y ordenar dinámicamente los datos de créditos según diferentes criterios.

## PASOS A SEGUIR

- **Optimización y Rendimiento:**

Si bien en la planificación se ve la mayor cantidad de detalles posible, una vez que la base de datos esté en funcionamiento es importante realizar distintos análisis de rendimiento y optimización para garantizar que las consultas se ejecuten de manera eficiente, especialmente a medida que la base de datos crece en tamaño y complejidad.

- **Implementación de Seguridad:**

Revisar la autenticación de usuarios, controlar los accesos dependiendo de los roles y encriptar los datos, por ejemplo, para proteger la integridad y la confidencialidad de la información almacenada.

- **Monitoreo y Mantenimiento Continuo:**

Establecer procedimientos de monitoreo continuo para supervisar el rendimiento de la BD, identificar posibles problemas o cuellos de botella y realizar mantenimiento preventivo de la base de datos para garantizar su disponibilidad y confiabilidad a largo plazo.

- **Integración con Otros Sistemas:**

Integrar la base de datos con otros sistemas o aplicaciones dentro de la empresa resulta vital, como sistemas de gestión de relaciones con los clientes (CRM) o sistemas de contabilidad, para compartir datos de manera eficiente y mejorar la colaboración entre diferentes departamentos.

- **Análisis de Datos Avanzado:**

Explorar técnicas de análisis de datos avanzadas, como minería de datos, Business Intelligence o aprendizaje automático, para obtener información valiosa a partir de los datos almacenados y respaldar la toma de decisiones estratégicas dentro de la empresa.