

Proyecto final

Descripción de la temática: Call Center de Cobranza

Este proyecto se enfoca en la creación de una base de datos robusta y eficiente para gestionar las operaciones de un call center de cobranza. La temática elegida aborda la necesidad de organizar, rastrear y optimizar el proceso de recuperación de deudas, fundamental para la salud financiera de cualquier negocio que ofrezca crédito.

Introducción

Descripción General

El proyecto final consiste en el diseño e implementación de una base de datos relacional para un call center de cobranza. Esta base de datos será el corazón de las operaciones diarias, permitiendo una gestión integral de los deudores, los agentes de cobranza, las llamadas realizadas, los acuerdos de pago y el seguimiento de cada caso.

Contexto o Idea Principal

La idea principal de este proyecto es proporcionar una herramienta tecnológica que centralice la información y automatice procesos clave dentro de un call center de cobranza. Su función en el mundo real es optimizar la recuperación de deudas, mejorar la eficiencia de los agentes, garantizar el cumplimiento de normativas y ofrecer una visión clara del rendimiento de las operaciones de cobranza. En esencia, permitirá a la organización gestionar de forma más efectiva sus cuentas por cobrar y reducir la morosidad.

Objetivos

Metas Específicas

Centralizar la información de deudores y deudas: Consolidar todos los datos relevantes de los clientes y sus obligaciones financieras en un único sistema.

Automatizar el registro de interacciones: Permitir el registro rápido y preciso de cada llamada, mensaje o acuerdo de pago con los deudores.

Facilitar la asignación y seguimiento de casos: Habilitar la asignación eficiente de deudores a agentes específicos y el seguimiento del progreso de cada caso.

Generar reportes de gestión: Proporcionar herramientas para la creación de informes sobre la productividad de los agentes, la efectividad de las estrategias de cobranza y el estado general de las deudas.

Mejorar la toma de decisiones: Ofrecer datos claros y actualizados que permitan a la gerencia tomar decisiones estratégicas para optimizar la cobranza.

Áreas a Cubrir

La base de datos gestionará los siguientes aspectos:

Gestión de deudores: Información personal, historial de pagos, detalles de las deudas.

Gestión de deudas: Montos adeudados, fechas de vencimiento, tipo de deuda, historial de movimientos.

Gestión de agentes: Datos de los agentes de cobranza, rendimiento individual, casos asignados.

Gestión de interacciones: Registro de llamadas (fecha, duración, resultado), notas de la interacción, acuerdos de pago.

Gestión de acuerdos de pago: Detalles de los planes de pago acordados, fechas de cumplimiento, montos.

Generación de reportes: Rendimiento de agentes, estado de cartera, efectividad de estrategias de cobranza, análisis de morosidad.

Optimización de procesos: Estandarización del flujo de trabajo de cobranza y reducción de errores manuales.

Impacto

Estas metas resolverán el problema de la desorganización y la ineficiencia en la gestión de cobranza. Al centralizar la información y automatizar procesos, el call center podrá:

Reducir el tiempo y el esfuerzo dedicados a tareas administrativas.

Aumentar la tasa de recuperación de deudas.

Mejorar la productividad de los agentes.

Proveer información precisa para la toma de decisiones estratégicas.

Disminuir los errores humanos en el registro de datos.

Garantizar un seguimiento efectivo de cada caso de cobranza.

Situación problemática

Identificación del Problema

La necesidad de implementar una base de datos para un call center de cobranza surge de la complejidad inherente a la gestión de un gran volumen de deudores y sus respectivas deudas. Sin un sistema adecuado, la información se dispersa, las interacciones se pierden y el seguimiento se vuelve ineficiente, lo que impacta directamente en la capacidad de la empresa para recuperar sus activos.

Deficiencias del Sistema Actual

Actualmente, muchos call centers de cobranza operan con sistemas fragmentados o, en algunos casos, con registros manuales o hojas de cálculo. Las principales deficiencias son:
Registros en papel o sistemas manuales: Esto conlleva a la pérdida de información, errores de transcripción, dificultad para actualizar datos y un acceso lento a la información.

Dispersión de la información: Los datos de deudores, deudas e interacciones pueden estar almacenados en diferentes lugares, lo que dificulta obtener una visión completa del caso.

Falta de seguimiento automatizado: No hay un sistema que recuerde automáticamente los próximos pasos o los acuerdos de pago, lo que puede llevar a incumplimientos o a perder oportunidades de cobro.

Dificultad para generar reportes: Extraer información relevante para evaluar el rendimiento o tomar decisiones estratégicas es un proceso manual, lento y propenso a errores.

Ineficiencia de los agentes: Los agentes dedican una cantidad considerable de tiempo a buscar información en lugar de interactuar con los deudores.

Solución Propuesta

La nueva base de datos SQL resolverá estos problemas al:

Centralizar toda la información: Creando un repositorio único y confiable para los datos de deudores, deudas, agentes e interacciones.

Estandarizar los procesos de registro: Asegurando que la información se capture de manera uniforme y completa.

Automatizar el seguimiento: Implementando funcionalidades para programar recordatorios de pago y seguimiento de casos.

Facilitar la generación de reportes en tiempo real: Proporcionando cuadros de mando y reportes predefinidos que permitan una visión clara del estado de la cobranza.

Mejorar la eficiencia operativa: Al reducir el tiempo de búsqueda de información y automatizar tareas repetitivas, los agentes podrán enfocarse en la recuperación de deudas.

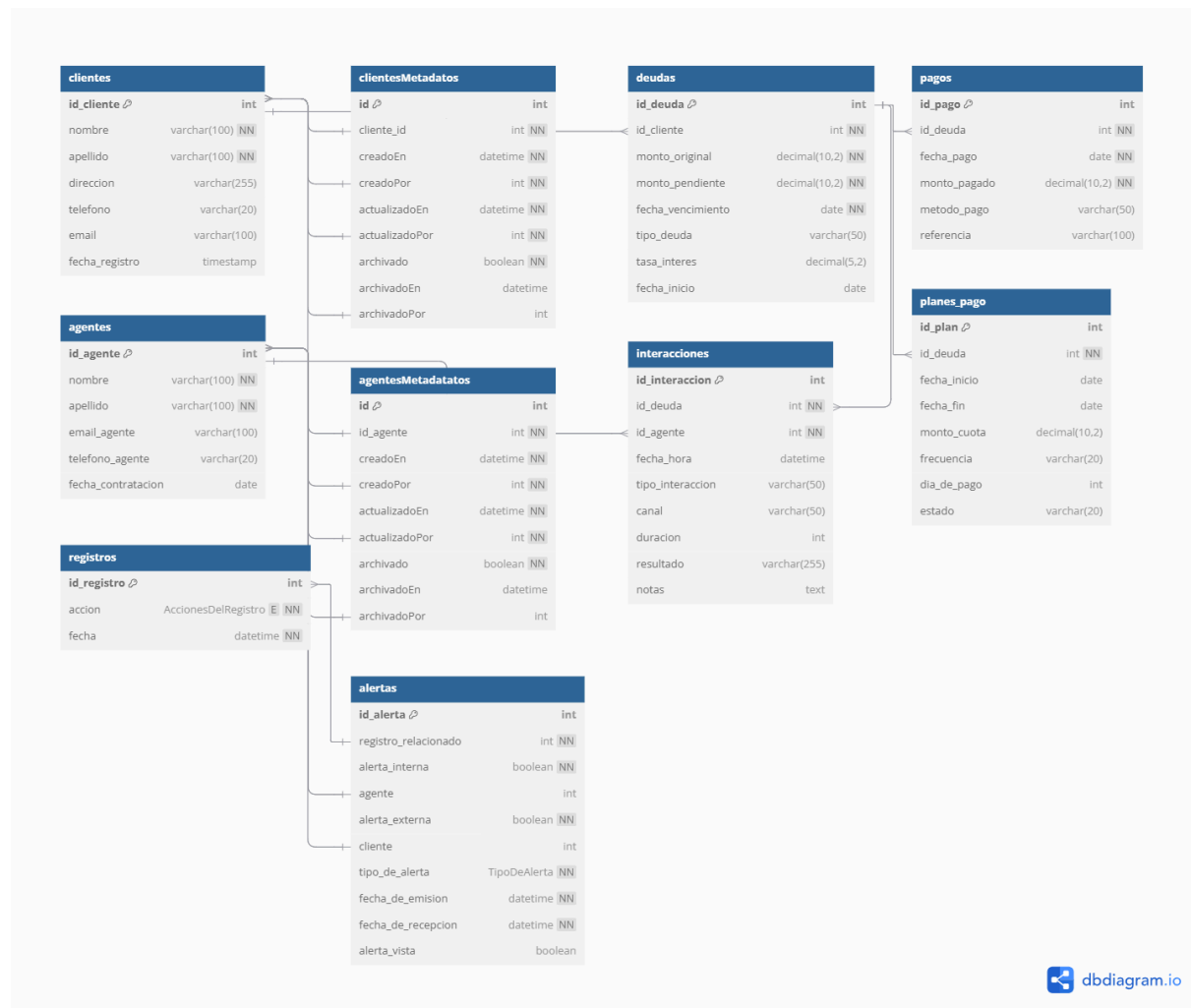
Reducir costos operativos: Minimizará la necesidad de recursos manuales y reducirá los errores que pueden generar costos adicionales.

Modelo de negocio

Descripción Abstracta de la Organización

La entidad que utilizará esta solución es una empresa o departamento de cobranza que gestiona una cartera de deudas de diferentes orígenes (financieras, minoristas, servicios, etc.). Esta organización se caracteriza por tener un equipo de agentes de cobranza que se comunican con los deudores a través de múltiples canales. Su operación depende en gran medida de la eficiencia en la gestión de la información y la capacidad de realizar un seguimiento proactivo de cada caso. La organización puede ser desde un pequeño call center interno de una empresa hasta un gran proveedor de servicios de cobranza externo.

Función y Propósito



Diagrama

Listado de Vistas más una descripción detallada, su objetivo, y qué tablas las componen.

1. vw_cartera_vencida – Vista de Cartera Vencida

- **Descripción:** Muestra las deudas vencidas con monto pendiente, incluyendo información del cliente, agente asignado, días de mora y saldo ajustado con intereses.
- **Objetivo:** Identificar deudas en mora y facilitar la gestión de cobranza.
- **Tablas involucradas:**
 - deudas, clientes, interacciones, agentes.

2. vw_efectividad_agentes – Vista de Efectividad de Agentes

- **Descripción:** Resume la gestión de cada agente: cuántas deudas ha gestionado, cuántas se liquidaron, cuánto ha recaudado y su promedio de duración por interacción.
- **Objetivo:** Evaluar el desempeño de los agentes de cobranza.

- *Tablas involucradas:*
 - agentes, interacciones, deudas, pagos.
3. vw_historial_pagos_cliente – Vista de Historial de Pagos por Cliente
- *Descripción:* Muestra el historial de pagos realizados por cliente, con tipo de deuda, método de pago y acumulado pagado hasta cada fecha.
 - *Objetivo:* Dar seguimiento detallado al progreso de pago de cada deuda.
 - *Tablas involucradas:*
 - clientes, deudas, pagos.
4. vw_estado_planes_pago – Vista de Planes de Pago con Estado
- *Descripción:* Muestra los planes de pago activos por cliente, indicando monto pendiente, total pagado y lo recaudado dentro del rango del plan.
 - *Objetivo:* Supervisar el cumplimiento y avance de los planes de pago acordados.
 - *Tablas involucradas:*
 - planes_pago, deudas, clientes, pagos.
5. vw_alertas_pendientes – Vista de Alertas Pendientes
- *Descripción:* Lista las alertas no vistas, clasificando al destinatario (cliente, agente o sistema), con información de la deuda y horas desde su emisión.
 - *Objetivo:* Facilitar el seguimiento de alertas importantes relacionadas a acciones pendientes.
 - *Tablas involucradas:*
 - alertas, registros, clientes, agentes, deudas.

Listado de Funciones que incluyan una descripción detallada, el objetivo para la cual fueron creadas y qué datos o tablas manipulan y/o son implementadas.

1. calcular_dias_mora – Calcular días de mora de una deuda
- *Descripción:* Calcula cuántos días han pasado desde la fecha de vencimiento de una deuda hasta el día actual.
 - *Objetivo:* Determinar automáticamente la cantidad de días en mora de una deuda para usar en reportes, alertas o cálculos financieros.
 - *Tablas involucradas:*
 - deudas (consulta de fecha_vencimiento)
2. calcular_saldo_interes – Calcular saldo pendiente con intereses
- *Descripción:* Calcula el saldo actualizado de una deuda considerando los intereses acumulados en función de los días de mora y la tasa de interés definida.
 - *Objetivo:* Proveer una forma automatizada de obtener el saldo real que debe el cliente incluyendo intereses por mora.
 - *Tablas involucradas:*
 - deudas (consulta de monto_pendiente, tasa_interes y fecha_vencimiento)
 - Además, invoca la función calcular_dias_mora() como parte del cálculo

Listado de Stored Procedures con una descripción detallada, qué objetivo o beneficio aportan al proyecto, y las tablas que lo componen y/o tablas con las que interactúa.

1. registrar_pago – Registrar pago y actualizar deuda

- *Descripción:* Registra un nuevo pago asociado a una deuda, incluyendo método y referencia.
- *Objetivo/Beneficio:* Automatiza el registro de pagos y delega la actualización del saldo pendiente a un trigger, garantizando integridad de datos.
- *Tablas involucradas:*
 - pagos (inserción directa)
 - deudas (afectada indirectamente mediante trigger para actualizar monto_pendiente)

2. crear_plan_pago – Generar plan de pago

- *Descripción:* Crea un plan de pago para una deuda, calculando fecha de fin según frecuencia (mensual, quincenal, otro).
- *Objetivo/Beneficio:* Facilita la programación estructurada de pagos, con lógica incorporada para el periodo del plan.
- *Tablas involucradas:*
 - planes_pago (inserción directa)
 - deudas (relación mediante id_deuda)

3. generar_reporte_morosidad – Reporte diario de morosidad

- *Descripción:* Genera alertas internas para los agentes sobre clientes con deudas vencidas, y registra esta acción en la tabla de eventos (registros).
- *Objetivo/Beneficio:* Automatiza notificaciones internas para la gestión activa de cuentas en mora y crea trazabilidad del proceso.
- *Tablas involucradas:*
 - registros (inserta un nuevo evento)
 - alertas (inserta nuevas alertas vinculadas al evento)
 - deudas, clientes, agentes, interacciones (consulta de datos para decidir a quién notificar)

Uso de Procedimientos Almacenados Los procedimientos almacenados permiten ejecutar secuencias de operaciones predefinidas.

- **registrar_pago**: Registra un pago y actualiza automáticamente el saldo pendiente de la deuda.
 - Ejemplo: `CALL registrar_pago(1, 500.00, 'Tarjeta', 'REF1001');`
- **crear_plan_pago**: Genera un plan de pagos estructurado para una deuda específica.
 - Ejemplo: `CALL crear_plan_pago(2, 83.58, 'Mensual', 5);`
- **generar_reporte_morosidad**: Crea alertas automáticas para deudas en mora.
 - Ejemplo: `CALL generar_reporte_morosidad();`

2. Uso de Funciones Las funciones realizan cálculos específicos y devuelven un valor.

- **calcular_dias_mora**: Devuelve el número de días que una deuda ha estado en mora.
 - Ejemplo: `SELECT calcular_dias_mora(1) AS dias_mora;`
- **calcular_saldo_interes**: Calcula el saldo pendiente de una deuda, incluyendo los intereses acumulados por mora.
 - Ejemplo: `SELECT calcular_saldo_interes(1) AS saldo_con_interes;`

3. Uso de Vistas Las vistas son tablas virtuales que simplifican consultas complejas, mostrando datos pre-filtrados o agregados.

- **vw_cartera_vencida**: Muestra todas las deudas actualmente en mora.
 - Ejemplo: `SELECT * FROM vw_cartera_vencida WHERE id_deuda = 1;`
- **vw_efectividad_agentes**: Proporciona métricas de rendimiento para los agentes (deudas gestionadas, recaudación, etc.).
 - Ejemplo: `SELECT * FROM vw_efectividad_agentes WHERE id_agente = 1;`
- **vw_historial_pagos_cliente**: Detalla el historial completo de pagos de cada cliente.
 - Ejemplo: `SELECT * FROM vw_historial_pagos_cliente WHERE id_cliente = 1;`
- **vw_estado_planes_pago**: Presenta el estado actual y el progreso de los planes de pago.
 - Ejemplo: `SELECT * FROM vw_estado_planes_pago WHERE id_deuda = 1;`
- **vw_alertas_pendientes**: Muestra las alertas generadas que aún no han sido marcadas como vistas.
 - Ejemplo: `SELECT * FROM vw_alertas_pendientes WHERE id_alerta = 1;`

Herramientas y tecnologías usadas:

- mysql workbench
- <https://dbdiagram.io> (Gráficas de tablas)
- <https://github.com> (Entrega y almacenamiento)
- <https://gemini.google.com> (Creación de datos)