

SEMINARIO DE PRÁCTICA DE INFORMATICA

ENTREGA N°2

Sistema de Gestión de Créditos Personales para Venta de Productos en Cuotas

CATEDRA - C - INF275 - EDH - PR - TUTORIAS CUATRIMESTRALES 2/25

Alumno: DAVIES ROBERTO OMAR

Profesor Titular Disciplinar: PABLO ALEJANDRO VIRGOLINI

Titular Experto: HUGO FERNANDO FRIAS

Fecha: 29/09/2025

CONTENIDO

Objetivo de esta actividad.....	4
Enunciado de la actividad.....	4
Consigna abierta.....	5
Los entregables a presentar son los siguientes:	5
Referencia:	6
Formato entregable	6
Desarrollo:	6
1. Título del proyecto	7
2. Introducción	7
3. Justificación	8
4. Objetivo general del proyecto.....	9
4.1. Objetivos específicos.....	9
4.2. Objetivos del sistema	9
5. Alcance	10
5.1. Límites y exclusiones:.....	10
5.2. Desde y hasta dónde llega el sistema:	10
6. Elicitación de requerimientos	11
6.1. Proceso de relevamiento	11
6.2. Identificación y priorización de requerimientos	12
6.3. Requerimientos funcionales.....	12
6.4. Requerimientos no funcionales	16
7. Modelo de Negocio	18
7.1. Actores principales	18
7.2. Procesos clave	18
7.3. Reglas de negocio.....	18
8. Propuesta de solución.....	20
9. Etapa de Análisis	21
9.1. Diagrama de casos de uso (UML)	21
9.2. Tabla: Casos de uso	21
9.3. Tabla de casos de uso <<include>> y <<extend>>	22
9.4. Fichas de casos de uso	22
9.5. Modelo de dominio (diagrama de clases simplificado).....	24
10. Etapa de Diseño.....	27
10.1. Descripción de la arquitectura general (MVC)	27

10.2.	Diseño de pantallas/interfaz	27
10.3.	Diagramas de secuencia	29
10.4.	Matriz de trazabilidad de requerimientos funcionales (RF).....	31
10.5.	Matriz de trazabilidad de requerimientos NO funcionales (RNF)	32
11.	Etapas de Implementación	33
11.1.	Tecnologías utilizadas.....	33
11.2.	Estructura de carpetas y clases principales.....	33
11.3.	Repositorio de código fuente	34
11.4.	Diagrama de despliegue.....	34
12.	Etapas de Pruebas.....	35
12.1.	Tipos de pruebas realizadas	35
12.2.	Plan de pruebas y casos de prueba	35
12.3.	Observaciones generales	37
13.	Modelo de datos y persistencia	38
13.1.	Modelo relacional de datos.....	38
13.2.	Diagrama Entidad-Relación (ER).....	38
13.3.	Definición de base de datos y scripts	39
13.4.	Consultas SQL de ejemplo	39
13.5.	Capturas de pantalla de las consultas tomadas de phpMyAdmin:	41
14.	Definiciones de comunicación	42
15.	Prototipo funcional	43
16.	Conclusiones y trabajo futuro	51
16.1.	Conclusiones.....	51
16.2.	Trabajo futuro	51
Bibliografía:		52

OBJETIVO DE ESTA ACTIVIDAD

Esta actividad busca que seas capaz de plantear una solución problemática que pueda resolverse mediante la realización de un proyecto informático.

Para llevar adelante este desafío, vas a poder aplicar muchos de los conceptos más importantes abordados durante el desarrollo del módulo 2, que recupera tu trabajo en materias troncales de la carrera.

Durante el proceso, lograrás aplicar tus conocimientos para alcanzar los siguientes objetivos:

- Reconocer y aplicar el proceso unificado de desarrollo.
- Utilizar UML para representar y comunicar los aspectos del sistema.
- Diseñar un modelo de datos relacional y realiza un diagrama entidad-relación.
- Emplear consultas SQL para gestionar una base de datos.
- Plantear los requerimientos de comunicación del sistema.

En este segundo TP deberás retomar los requerimientos planteados para el sistema a desarrollar y avanzar con el proyecto.

ENUNCIADO DE LA ACTIVIDAD

Los sistemas informáticos desempeñan un papel muy importante para la optimización de procesos en diversas áreas de cualquier organización. La tecnología brinda la posibilidad de automatizar tareas, recopilar y analizar datos a gran escala, agilizar procesos y proporcionar soluciones a desafíos complejos.

Te propongo que retomes lo que definiste en la actividad anterior y realices una entrega de acuerdo con lo solicitado en la consigna. Recuerda que puedes definir cualquier organización real y explorar oportunidades en relación con su seguridad, logística, gestión de inventarios, optimización de procesos industriales, análisis de datos, cuidado de la salud, la educación, o cualquier otra área.

Para continuar con la realización de un proyecto de desarrollo informático, se requiere aplicar de forma completa los modelos de análisis, diseño, implementación y pruebas del PUD. Es clave elegir adecuadamente los artefactos a presentar y utilizar correctamente el lenguaje unificado de desarrollo (UML).

En cuanto a la persistencia de los datos, se requiere elaborar un modelo relacional que represente las entidades, sus atributos y las relaciones específicas propias al sistema. Se deben aplicar los principios de normalización con el fin de garantizar que las tablas en la base de datos estén organizadas de manera eficaz y presentar un diagrama de entidad relación.

Paralelamente, se evalúan los requisitos de comunicación del sistema y se determina cómo se llevan a cabo las interacciones entre los diversos componentes del mismo.

CONSIGNA ABIERTA

Considera que continúas trabajando en la organización cuyo problema definiste resolver con un proyecto informático, y estás en el equipo de desarrollo que va a llevar adelante el mismo.

Recuerda que el proyecto informático definido debe ser distinto al considerado en la lectura y que el entregable final debe cumplir con los siguientes objetivos:

Realizar el análisis del modelo de negocios para definir y justificar el proyecto.

Aplicar el proceso unificado de desarrollo (PUD) para garantizar la calidad, escalabilidad y eficiencia en el ciclo de desarrollo.

Utilizar una base de datos MySQL para la persistencia de los datos.

Emplear Java como lenguaje de programación para el desarrollo del sistema.

A los fines del trabajo, para la implementación de la base de datos y el desarrollo con Java, puedes presentar un prototipo, que es “es la creación de un modelo operacional que incluya solo algunas características del sistema final” (Kendall & Kendall, 2011).

El concepto de operacional es clave, ya que no se trata de un simple modelo, sino que permite desarrollar módulos que se van integrando en la versión final del sistema.

Esta segunda actividad te permitirá aplicar de forma completa los modelos de análisis, diseño, implementación y pruebas del PUD. Es fundamental asegurar la trazabilidad del sistema, seleccionando adecuadamente los artefactos a presentar en cada etapa, utilizando correctamente el lenguaje unificado de desarrollo (UML) para representar y comunicar los diferentes aspectos del sistema.

A continuación, deberás diseñar un modelo de datos relacional que represente las entidades, atributos y relaciones específicas del sistema. Es importante aplicar los principios de normalización para asegurar que las tablas en la base de datos estén estructuradas de manera eficiente y plantear la realización de un diagrama entidad-relación. Recuerda asegurar que las estructuras de datos estén alineadas con el diseño del sistema y cumplan con los requisitos de integridad y rendimiento.

Finalmente, podrás definir cuáles son los protocolos y estándares adecuados para la comunicación entre el sistema y otros sistemas externos o la base de datos: entorno de red, infraestructura física, control de enlace de datos, etc.

LOS ENTREGABLES A PRESENTAR SON LOS SIGUIENTES:

- Etapa de análisis.
- Etapa de diseño.
- Etapa de implementación.
- Etapa de pruebas.
- Definición de base de datos para el sistema.
- Diagrama entidad-relación de la base de datos.
- Creación de las tablas MySQL.
- Inserción, consulta y borrado de registros.
- Presentación de las consultas SQL.
- Definiciones de comunicación.

REFERENCIA:

Kendall, K., & Kendall, J. (2011). Análisis y diseño de sistemas. Pearson Education.

FORMATO ENTREGABLE

La actividad propuesta debe ser elaborada en forma individual y respetar los criterios de presentación mencionados a continuación. Se debe realizar una entrega incremental, considerando las correcciones realizadas en la entrega previa.

El informe se debe presentar en un documento en formato PDF (papel A4, letra Calibri 11, espaciado simple). Debe incluir las respuestas a las consignas de forma prolija y ordenada, con su correspondiente explicación para justificar las decisiones adoptadas. Incluir una portada y respetar el siguiente criterio para el nombre del archivo:

«APELLIDO-NOMBRE»-AP2.PDF

La realización de todos los diagramas requiere utilizar una herramienta de libre elección, pero asegurando la prolijidad y claridad de presentación. Todos los diagramas deben incluirse en el documento a entregar con su correspondiente explicación.

Para la persistencia de los datos, deberás utilizar una base de datos MySQL, y para el desarrollo del prototipo, una IDE de Java. La explicación debe estar en el documento a entregar e incluir un enlace a la versión completa (código y archivos asociados) en Github: <https://github.com/Links to an external site>.

Se evaluará la precisión en las respuestas, capacidad de síntesis, redacción, ortografía y calidad de la presentación.

En caso de requerirlo, es importante revisar el cumplimiento de las normas de estilo American Psychological Association (APA).

DESARROLLO:

A continuación, se presenta el desarrollo completo del documento, estructurado y redactado conforme a las etapas y artefactos definidos por el Proceso Unificado de Desarrollo (PUD). Se abordan en detalle la problemática, la justificación, los objetivos, el análisis, diseño, implementación y pruebas del sistema, junto con los modelos, diagramas y evidencias correspondientes.

1. TÍTULO DEL PROYECTO

Sistema de Gestión de Créditos Personales para Venta de Productos en Cuotas

2. INTRODUCCIÓN

Este proyecto surge a partir de una necesidad detectada en una tienda de indumentaria y calzado de la ciudad de Trelew (Chubut), la cual administraba sus créditos de manera manual y fragmentada (planillas sueltas y registros informales), generando demoras, errores de cálculo y falta de trazabilidad. A partir de entrevistas y conversaciones con el personal operativo y administrativo de la tienda se realizó la elicitación de requerimientos, dando origen al presente sistema de gestión de créditos y pagos.

En el contexto actual, muchas pequeñas empresas y comercios minoristas enfrentan la necesidad de ofrecer a sus clientes la posibilidad de financiar sus compras a través de créditos personales y pagos en cuotas. Esta modalidad de financiación resulta fundamental para incrementar las ventas y fidelizar a los clientes, especialmente en escenarios económicos donde el acceso al crédito bancario es limitado para algunos sectores.

Sin embargo, la gestión manual de los créditos, vencimientos y pagos, realizada a menudo mediante planillas de cálculo o registros en papel, presenta desafíos significativos: dificultades para el seguimiento, errores en el cálculo de intereses, registros incompletos o desactualizados y una capacidad limitada para conocer el estado real de la cartera.

La ausencia de una herramienta informática adecuada genera problemas adicionales, como falta de información centralizada y oportuna para apoyar decisiones y evaluar riesgo operativo.

Ante esta problemática surge la necesidad de un sistema que organice y automatice la administración de créditos personales (clientes, créditos, cuotas, pagos y períodos de control). El desarrollo propuesto busca aportar información clara y en tiempo casi real, reducir errores y optimizar la gestión financiera interna. Asimismo, incorpora la noción de “lotes” para segmentar periodos operativos y facilitar la marcación de cuotas en mora. También se incorpora la emisión automática de un comprobante de alta de crédito y de recibos de pago, lo que mejora la trazabilidad documental.

3. JUSTIFICACIÓN

La necesidad de desarrollar este sistema surge de la brecha operativa evidenciada en el caso real de la tienda de indumentaria y calzado de Trelew y que se replica en numerosos comercios minoristas: la gestión manual y fragmentada de créditos, cuotas y pagos limita el control, incrementa riesgos de error y dificulta la obtención de información confiable. El uso de planillas o anotaciones informales impide un seguimiento preciso del estado de cada crédito, retrasa las cobranzas y reduce la trazabilidad frente a reclamos.

El sistema propuesto:

- Centraliza y resguarda la información (clientes, créditos, cuotas, pagos).
- Mejora la trazabilidad (evolución de créditos y cuotas, pagos, anulación controlada).
- Provee listados operativos consistentes (créditos activos, cuotas pendientes, historial de pagos).
- Reduce riesgos operativos (errores de cálculo, pérdida o duplicidad de registros).
- Sienta bases para escalabilidad futura (reportes analíticos, alertas, roles y auditoría).
- Indirectamente mejora la percepción de orden y confiabilidad frente a los clientes.
- Con la emisión del comprobante de crédito y el recibo de pago se mejora el respaldo documental interno y la claridad ante reclamos.

En síntesis, transforma un proceso manual en uno ordenado, seguro y escalable, generando valor operativo y estratégico.

4. OBJETIVO GENERAL DEL PROYECTO

Lograr una gestión eficiente, ordenada y confiable de los créditos personales otorgados a clientes, asegurando control integral, trazabilidad de la información y soporte a la toma de decisiones internas.

4.1. Objetivos específicos

- Centralizar la información de clientes, créditos, cuotas y pagos minimizando errores.
- Fortalecer el control de la cartera (estados de cuotas, detección de mora).
- Mejorar la capacidad de análisis operativo mediante listados consistentes.
- Optimizar tareas administrativas de alta de créditos, cobro y cierre de periodos.
- Incrementar la transparencia y seguridad básica del proceso, profesionalizando la gestión.
- Emitir comprobante de alta de crédito y recibos de pago para respaldo interno.

4.2. Objetivos del sistema

El sistema permitirá:

- Registrar, modificar, consultar y eliminar datos de clientes.
- Otorgar créditos definiendo monto, cantidad de cuotas y aplicando la tasa vigente.
- Generar automáticamente cuotas con cálculo de interés lineal y administrar su estado (pendiente, pagada, mora).
- Registrar pagos y anular pagos erróneos con reversión de estados.
- Emitir comprobante de alta de crédito.
- Emitir recibo por cada pago registrado.
- Consultar historial de créditos, cuotas y pagos por cliente.
- Agrupar créditos y cuotas en lotes (periodos) y efectuar cierre para marcar cuotas en mora.
- Proveer listados operativos básicos (créditos vigentes, cuotas pendientes, pagos realizados).
- Restringir el acceso mediante autenticación por contraseña (operativa y maestra).

5. ALCANCE

Incluye el desarrollo de un sistema de escritorio (Java + Swing) con persistencia MySQL para la gestión interna de créditos personales: administración de clientes, creación de créditos, generación de cuotas con interés, registro y anulación de pagos, control de estados y cierre de lotes con marcación de mora. Aporta comprobantes de crédito y pagos, y listados que permiten el seguimiento operativo de las operaciones.

5.1. Límites y exclusiones:

- Uso interno: no hay portal ni acceso directo para clientes externos.
- Sin gestión multiusuario con roles diferenciados (un único esquema de autenticación).
- Sin integración con pasarelas de pago, bancos o sistemas contables.
- Sin envío automático de recordatorios o notificaciones.
- Sin refinanciaciones ni reestructuración de créditos ya otorgados.
- Contraseñas almacenadas en texto plano (hashing pendiente).
- Sin auditoría detallada de operaciones (logs históricos no implementados).
- Reportes e indicadores gráficos fuera de la versión inicial (solo listados básicos).

5.2. Desde y hasta dónde llega el sistema:

Interviene desde el registro inicial de un cliente y el otorgamiento de un crédito hasta la gestión completa de sus cuotas (pagos, mora por cierre de lote) y la consulta de estados. No realiza cobranzas externas, ni integra ciclos contables o de tesorería ajenos al circuito interno. El diseño modular (capas Presentación–Control–DAO–DB) permite futuras ampliaciones funcionales y de seguridad. Cierra con la generación de comprobantes y recibos asociados a las operaciones internas.

6. ELICITACIÓN DE REQUERIMIENTOS

El proceso de relevamiento combinó técnicas cualitativas y observacionales para identificar necesidades operativas reales en la gestión de créditos personales en un comercio minorista de indumentaria y calzado de Trelew (Chubut). El objetivo fue transformar prácticas manuales (planillas dispersas y formularios aislados) en una solución unificada que asegure trazabilidad, integridad y eficiencia administrativa.

6.1. Proceso de relevamiento

Se aplicaron distintas acciones para entender cómo se trabaja actualmente y qué debía cubrir el sistema:

- Observación directa: se observaron las tareas diarias (altas de clientes, otorgamiento de créditos, registro de pagos y cierre del periodo). Se vieron tareas repetidas, controles que dependían solo de la memoria del personal y demoras por tener que revisar números a mano.
- Entrevistas: se realizaron entrevistas con dueños, personal administrativo y cajeros. Se recogieron problemas frecuentes: errores al calcular cuotas, dificultad para saber rápidamente cuánto queda por cobrar y pérdida de anotaciones sueltas.
- Revisión de documentos: se revisaron las planillas de Excel y los talonarios preimpresos usados cuando hay algún inconveniente con la computadora. Se detectaron riesgos: posibilidad de pisar datos, falta de un historial claro y ausencia de un enlace directo entre crédito, cuotas y pagos.
- Validación inicial del modelo: se presentaron las primeras ideas (entidades básicas y cálculo de interés lineal) y se ajustó el vocabulario. Se acordó usar el término “lote” para identificar cada periodo y luego poder marcar cuotas en mora al cerrarlo.

A continuación, se muestra el Excel utilizado por la tienda donde se ve como se registran los créditos, además, en otra hoja del Excel se guardan los datos de cada cliente con sus id:

B	C	D	E	F	G	H	I	J	K	L	M	N
ID CREDITO	ID_CLIENTE	RSOCIAL	NROCTA	DNI	FECHA	CUOTA	ATIDAD_C	VALOR_CUC	ENVIADO	NETO	MORA	A_VENCER
227328	1176	ABAD EXEQUIEL ESTEBAN	0830001031	1E+07	#####	4	4	\$ 122,30	\$ 122,30	\$ 429,00	\$ -	\$ -
228222	1176	ABAD EXEQUIEL ESTEBAN	0830001031	1E+07	#####	3	4	\$ 655,20	\$ 655,20	\$ 2.299,00	\$ -	\$ 655,20
229776	1176	ABAD EXEQUIEL ESTEBAN	0830001031	1E+07	#####	2	4	\$ 397,90	\$ 397,90	\$ 1.396,00	\$ -	\$ 795,80
229830	1176	ABAD EXEQUIEL ESTEBAN	0830001031	1E+07	#####	2	4	\$ 378,20	\$ 378,20	\$ 1.327,00	\$ -	\$ 756,40
230332	1176	ABAD EXEQUIEL ESTEBAN	0830001031	1E+07	6/6/2017	2	4	\$ 477,40	\$ 477,40	\$ 1.675,00	\$ -	\$ 954,80
231294	1176	ABAD EXEQUIEL ESTEBAN	0830001031	1E+07	#####	2	2	\$ 350,50	\$ 350,50	\$ 649,00	\$ -	\$ -
231442	1176	ABAD EXEQUIEL ESTEBAN	0830001031	1E+07	#####	1	4	\$ 511,30	\$ 511,30	\$ 1.794,00	\$ -	\$ 1.533,90
234476	1176	ABAD EXEQUIEL ESTEBAN	0830001031	1E+07	#####	3	3	\$ 656,70		\$ 1.759,00	\$ -	\$ 1.970,10
231566	6612	ABALLAY CECILIA SORAYA	0830001031	4E+07	#####	1	3	\$ 987,10	\$ 987,10	\$ 2.644,00	\$ -	\$ 1.974,20
104871	5442	ABALLAY JESUS GABRIEL (M)	FALTA	3E+07	#####	0	3	\$ 230,70	\$ 215,85	\$ 618,00	\$ 215,85	\$ -
229383	2953	ABITU MARCEL ALEJANDRA	0830021831	2E+07	#####	3	3	\$ 624,60	\$ 624,60	\$ 1.673,00	\$ -	\$ -
231219	2953	ABITU MARCEL ALEJANDRA	0830021831	2E+07	#####	2	3	\$ 421,10	\$ 421,10	\$ 1.128,00	\$ -	\$ 421,10
232292	2953	ABITU MARCEL ALEJANDRA	0830021831	2E+07	#####	1	3	\$ 479,40	\$ 479,40	\$ 1.284,00	\$ -	\$ 958,80
233988	2953	ABITU MARCEL ALEJANDRA	0830021831	2E+07	#####	2	2	\$ 323,50		\$ 599,00	\$ -	\$ 646,90
232151	1631	ABRAHAM LILIANA E.	0830016431	1E+07	#####	1	3	\$ 1.925,70	\$ 1.925,70	\$ 5.158,00	\$ -	\$ 3.851,40
230737	6189	ABRIGO DAIANA LUZ DE LA AI	0830001031	3E+07	#####	2	3	\$ 585,00	\$ 585,00	\$ 1.567,00	\$ -	\$ 585,00
228142	2909	ABRIGO EDGARDO J.	0830001031	2E+07	#####	3	3	\$ 539,80	\$ 539,80	\$ 1.446,00	\$ -	\$ -
229661	2909	ABRIGO EDGARDO J.	0830001031	2E+07	#####	2	3	\$ 2.157,10	\$ 2.157,10	\$ 5.778,00	\$ -	\$ 2.157,10
220179	6750	ABRIGO MARCOS ANTONIO	0830021831	2E+07	1/1/2016	0	5	\$ 634,20	\$ 4.018,50	\$ 634,20	\$ 4.018,50	\$ -
228598	5066	ABRIGO MARIA LUISA	0830021831	6E+06	6/5/2017	3	3	\$ 1.290,60	\$ 1.290,60	\$ 3.457,00	\$ -	\$ -
231914	5066	ABRIGO MARIA LUISA	0830021831	6E+06	6/7/2017	1	3	\$ 376,30	\$ 376,30	\$ 1.008,00	\$ -	\$ 752,60
75906	4464	ACEVEDO MAURO GABRIEL (FALTA	3E+07	6/5/2010	0	4	\$ 323,40	\$ 1.547,22	\$ 1.115,00	\$ 1.547,22	\$ -	\$ -
77717	4464	ACEVEDO MAURO GABRIEL (FALTA	3E+07	5/6/2010	0	3	\$ 271,40	\$ 1.715,40	\$ 727,00	\$ 1.715,40	\$ -	\$ -
80110	4464	ACEVEDO MAURO GABRIEL (FALTA	3E+07	#####	0	3	\$ 333,00	\$ 2.104,77	\$ 892,00	\$ 2.104,77	\$ -	\$ -
80538	4464	ACEVEDO MAURO GABRIEL (FALTA	3E+07	#####	0	1	\$ 26,00	\$ 54,79	\$ 26,00	\$ 54,79	\$ -	\$ -
211033012	5244	ACHAREZ SILVIA EDITH (M)	0830004131	2E+07	#####	0	1	\$ 461,40	\$ 656,71	\$ 461,40	\$ 656,71	\$ -
211033013	5244	ACHAREZ SILVIA EDITH (M)	0830004131	2E+07	#####	0	1	\$ 461,40	\$ 656,71	\$ 461,40	\$ 656,71	\$ -
211033014	5244	ACHAREZ SILVIA EDITH (M)	0830004131	2E+07	#####	0	1	\$ 461,40	\$ 656,71	\$ 461,40	\$ 656,71	\$ -
211033015	5244	ACHAREZ SILVIA EDITH (M)	0830004131	2E+07	#####	0	1	\$ 461,40	\$ 656,71	\$ 461,40	\$ 656,71	\$ -
211033016	5244	ACHAREZ SILVIA EDITH (M)	0830004131	2E+07	#####	0	1	\$ 461,40	\$ 656,71	\$ 461,40	\$ 656,71	\$ -
211033017	5244	ACHAREZ SILVIA EDITH (M)	0830004131	2E+07	#####	0	1	\$ 461,40	\$ 656,71	\$ 461,40	\$ 656,71	\$ -
211033018	5244	ACHAREZ SILVIA EDITH (M)	0830004131	2E+07	#####	0	1	\$ 461,40	\$ 656,71	\$ 461,40	\$ 656,71	\$ -
228592	1901	ACHERITOBEHERE MARISA	0830001031	2E+07	6/5/2017	3	3	\$ 280,00	\$ 280,00	\$ 750,00	\$ -	\$ -
232598	1901	ACHERITOBEHERE MARISA	0830001031	2E+07	#####	1	4	\$ 204,30	\$ 204,30	\$ 717,00	\$ -	\$ 612,90

También se adjunta un recibo que se emite e imprime desde otro Excel (se tapan datos personales del cliente por confidencialidad).

TIENDA TRELEW Rivadavia 1 - Tel: 285 - 4422621		RECIBO N° <input checked="" type="checkbox"/>	
		FECHA: 2025-09-27 17:37:08	
Documento de validación CUIT: 20-1174524-4 Ingresos Brutos: 90774 Ingresos Aca: 00/00/0000		COPIA	
Sr/Sra: [REDACTED] SUSANA			
Domicilio: [REDACTED] SARMIENTO			
Tel/Fax: 297 [REDACTED] DNI: [REDACTED]			
RECIBIMOS LA SUMA DE \$ 36,519.88 EN CONCEPTO DE: CREDITO: 587 RE Cuota: 2 de 4 Monto: \$ 36519.88			
TOTAL: \$ 36,519.88			
FIRMA:			
ACLARACION:			

6.2. Identificación y priorización de requerimientos

A partir de la evidencia recogida, se identificaron las siguientes necesidades principales:

- Automatizar el registro y actualización de datos de clientes, facilitando la búsqueda y gestión de información.
- Registrar la creación de créditos, parametrizando el monto, la cantidad de cuotas y la tasa de interés, y generando las cuotas automáticamente.
- Controlar el estado de cada cuota (pendiente, pagada, en mora), permitiendo el registro y seguimiento de pagos.
- Acceder de manera ágil al historial de créditos y pagos de cada cliente, así como a un resumen del historial de créditos.
- Agrupar operaciones en lotes mensuales para facilitar el cierre de períodos.
- Proteger la información mediante el control de acceso por contraseña.
- Emitir los comprobantes de las operaciones realizadas (créditos y pagos).
- A futuro, generar reportes periódicos que permitan analizar el desempeño financiero y facilitar la gestión de cobranzas.

Estos requerimientos fueron priorizados en función del impacto en la operatoria diaria, la reducción de riesgos y la mejora de la eficiencia administrativa. Se prestó especial atención a garantizar la trazabilidad de cada operación crediticia y la integridad de los datos, con el fin de proporcionar una base sólida para la toma de decisiones en la organización.

La elicitación de requerimientos se mantendrá como un proceso iterativo, de modo que nuevas necesidades o mejoras puedan ser incorporadas en futuras versiones del sistema.

6.3. Requerimientos funcionales

A continuación, se detallan los principales requerimientos funcionales identificados:

Requerimiento	RF01 – Ingreso al sistema
Actor	Usuario

Descripción	El usuario debe poder acceder al sistema ingresando una contraseña válida.
Precondiciones	El sistema debe estar instalado y en funcionamiento. El usuario debe conocer la contraseña asignada.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa la contraseña. 2. El sistema valida la contraseña contra la base de datos. 3. Si es correcta, permite el acceso.
Flujos alternativos	Si la contraseña es incorrecta, muestra mensaje de error y solicita reintento.
Postcondiciones	El usuario accede a las funcionalidades habilitadas del sistema.
Reglas de negocio asociadas	Solo usuarios que conozcan la contraseña pueden acceder al sistema.

Requerimiento	RF02 – Alta de cliente
Actor	Usuario
Descripción	Permite registrar un nuevo cliente en el sistema, ingresando sus datos completos.
Precondiciones	El usuario debe haber accedido al sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona “Alta de Cliente”. 2. Completa los campos requeridos (nombre, documento, dirección, teléfono, email). 3. El sistema valida que el documento no exista previamente. 4. El sistema almacena los datos en la base de datos.
Flujos alternativos	Si el documento ya existe, muestra mensaje de error y no permite el alta.
Postcondiciones	El cliente queda registrado y disponible para operaciones de crédito.
Reglas de negocio asociadas	El documento del cliente debe ser único en el sistema.

Requerimiento	RF03 – Modificación de cliente
Actor	Usuario
Descripción	Permite modificar los datos de un cliente existente.
Precondiciones	El cliente debe estar registrado previamente.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona el cliente a modificar. 2. El sistema muestra los datos actuales. 3. El usuario edita los campos y confirma los cambios. 4. El sistema valida y actualiza los datos en la base.
Flujos alternativos	Si los datos editados contienen un documento ya existente en otro cliente, muestra error y no permite la modificación.
Postcondiciones	El cliente queda actualizado en el sistema.
Reglas de negocio asociadas	El documento debe seguir siendo único.

Requerimiento	RF04 – Baja de cliente
Actor	Usuario
Descripción	Permite eliminar un cliente del sistema.

Precondiciones	El cliente debe estar registrado y seleccionado.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona al cliente a eliminar. 2. El sistema solicita confirmación. 3. Si el usuario confirma, elimina el registro.
Flujos alternativos	<p>Si el usuario cancela, no se efectúa la baja.</p> <p>Si el cliente tiene créditos vigentes no se permite la baja.</p>
Postcondiciones	El cliente es eliminado de la base.
Reglas de negocio asociadas	No se debe permitir la eliminación de clientes con créditos activos.

Requerimiento	RF05 – Alta de crédito
Actor	Usuario
Descripción	Permite otorgar un nuevo crédito a un cliente, parametrizando monto, cantidad de cuotas y aplicando la tasa de interés vigente.
Precondiciones	El cliente debe estar registrado. Debe haber un lote abierto.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona al cliente. 2. Ingresa el monto y la cantidad de cuotas. 3. El sistema muestra la tasa de interés vigente. 4. El usuario confirma la operación. 5. El sistema registra el crédito y genera automáticamente las cuotas correspondientes. 6. El sistema genera el comprobante para imprimir.
Flujos alternativos	Si el usuario ingresa valores inválidos (monto o cuotas ≤ 0), muestra error.
Postcondiciones	<p>El crédito y sus cuotas quedan registrados y asociados al cliente.</p> <p>El comprobante del crédito queda listo para ser impreso.</p>
Reglas de negocio asociadas	<p>La tasa de interés es la vigente en la tabla “variables”.</p> <p>La cantidad de cuotas debe estar dentro del rango permitido (ejemplo: 1 a 12 cuotas). En un futuro este valor podrá ser configurado.</p>

Requerimiento	RF06 – Consulta y listado de clientes y créditos
Actor	Usuario
Descripción	Permite buscar / listar clientes, ver el listado general de todos los créditos y consultar los créditos (y sus cuotas) de un cliente específico.
Precondiciones	El usuario debe haber ingresado al sistema; deben existir clientes y/o créditos cargados.
Flujo principal	<ol style="list-style-type: none"> 1. Ingresa (opcional) criterios de búsqueda de clientes (nombre y/o documento). 2. El sistema muestra la tabla de clientes filtrados (o todos si no ingresa criterios). 3. Selecciona “Listar todos los créditos” para ver todos los créditos registrados. 4. Selecciona un cliente y elige “Créditos por cliente”. 5. El sistema muestra los créditos de ese cliente. 6. Selecciona un crédito y el sistema muestra sus cuotas (número, monto, estado y lote de vencimiento derivado).
Flujos alternativos	<p>A. Búsqueda sin resultados: mensaje indicativo.</p> <p>B. Cliente sin créditos: listado vacío con mensaje.</p>
Postcondiciones	El usuario dispone de la información solicitada (clientes, créditos y cuotas) para continuar otras operaciones.
Reglas de negocio asociadas	<p>Documento de cliente único (verificado en alta/modificación).</p> <p>Estados de crédito: vigente / cancelado.</p> <p>Lote de vencimiento = lote_origen + número_cuota.</p>

Requerimiento	RF07 – Registro de pago de cuota
Actor	Usuario
Descripción	Permite registrar el pago de una cuota, actualizando el estado de la cuota y del crédito si corresponde.
Precondiciones	Debe existir al menos una cuota pendiente para el cliente.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona al cliente y la cuota a pagar. 2. Elige método de pago y opcionalmente ingresa observaciones. 3. El sistema registra el pago y marca la cuota como pagada. 4. Si es la última cuota pendiente del crédito, cambia el estado del crédito a “cancelado”. 5. El sistema genera el comprobante del pago para imprimir.
Flujos alternativos	Si el usuario cancela, no se registra el pago.
Postcondiciones	El pago queda registrado y la cuota se actualiza. El comprobante del pago queda listo para ser impreso.
Reglas de negocio asociadas	Solo pueden pagarse cuotas en estado “pendiente”.

Requerimiento	RF08 – Consulta de historial y deuda por cliente
Actor	Usuario
Descripción	Visualiza, para un cliente, historial de créditos, sus cuotas y pagos, y un resumen de deuda (Deuda TOTAL, Deuda ACTUAL y Deuda en MORA) más el detalle de cuotas abiertas.
Precondiciones	Cliente existente; al menos un crédito o pagos asociados (si no, se informa vacío).
Flujo principal	<ol style="list-style-type: none"> 1. Selecciona un cliente. 2. El sistema muestra sus créditos. 3. Selecciona un crédito y se listan sus cuotas (número, monto, estado, lote de vencimiento derivado). 4. Accede a “Ver Deuda”. 5. El sistema calcula y muestra: <ul style="list-style-type: none"> • Deuda TOTAL = Σ cuotas pendiente o mora (todas). • Deuda ACTUAL = Σ cuotas pendiente o mora con $\text{lote_vencimiento} \leq \text{lote actual}$. • Deuda en MORA = Σ cuotas en mora. 6. Lista detalle de cuotas abiertas indicando si es futura ($\text{lote_venc} > \text{lote actual}$). 7. (Opcional) Refresca y se recalcula tras pagos o cierre de lote.
Flujos alternativos	A. Cliente sin créditos: mensaje “El cliente no posee créditos registrados”. B. Cliente sin cuotas abiertas: totales = 0.00 y detalle vacío.
Postcondiciones	Se visualiza historial y totales de deuda para gestión de cobranzas.
Reglas de negocio asociadas	$\text{lote_vencimiento} = \text{lote_origen} + \text{número_cuota}$. “Futura” si $\text{lote_vencimiento} > \text{lote actual}$. Solo cuotas pendiente o mora se incluyen en los totales. La mora se determina en el cierre de lote (RF09).

Requerimiento	RF09 – Cierre de lote y generación de siguiente
Actor	Usuario
Descripción	Permite cerrar el lote actual de créditos y abrir uno nuevo, actualizando el estado de las cuotas vencidas a “mora”.
Precondiciones	Debe haber un lote abierto.

Flujo principal	<ol style="list-style-type: none"> 1. El usuario confirma el cierre del lote. 2. El sistema registra la fecha de cierre y crea un nuevo lote con fecha de apertura actual. 3. Las cuotas vencidas y no pagadas se marcan como “mora”. Regla: (lote_origen + número_de_cuota) == nro_lote_cerrado. 4. El contador de lote en variables se actualiza.
Flujos alternativos	Si el usuario cancela, no se efectúa el cierre.
Postcondiciones	El nuevo lote queda abierto y el anterior, cerrado.
Reglas de negocio asociadas	El cierre de lote solo puede realizarse una vez por lote. Solo se permite abrir un nuevo lote al cerrar el anterior.

Requerimiento	RF10 – Cambio de contraseña
Actor	Usuario
Descripción	Permite modificar la contraseña de acceso al sistema.
Precondiciones	El usuario debe haber ingresado la contraseña actual correctamente.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede a la función de cambio de contraseña. 2. Ingresa la nueva contraseña y la confirma. 3. El sistema valida las entradas y actualiza la contraseña en la base de datos (por ahora la contraseña se guarda sin cifrado).
Flujos alternativos	Si las contraseñas no coinciden o quedan vacías, muestra error y solicita reingreso.
Postcondiciones	La nueva contraseña queda vigente para futuros accesos.
Reglas de negocio asociadas	La contraseña debe mantenerse en secreto y cumplir con las políticas de la organización.

6.4. Requerimientos no funcionales

Código	Descripción
RNF01	El sistema debe estar desarrollado en Java, con interfaz gráfica implementada en Swing.
RNF02	El sistema utilizará una base de datos MySQL para la persistencia de los datos.
RNF03	El acceso a los datos debe realizarse mediante un patrón Modelo-Vista-Controlador (MVC) y DAO para persistencia.
RNF04	El sistema debe estar disponible durante el horario operativo del comercio (lunes a sábado, 8 horas por día). Se acepta como máximo una interrupción no planificada de hasta 30 minutos por semana. Las tareas de mantenimiento o respaldo se realizan fuera de ese horario.
RNF05	El sistema debe garantizar la seguridad de los datos, solicitando contraseña al iniciar el programa.
RNF06	El sistema debe permitir realizar copias de respaldo periódicas de la base de datos las cuales se realizarán manualmente luego del cierre del comercio.
RNF07	El sistema debe almacenar todas las fechas en formato estándar internacional (ISO 8601).
RNF08	El sistema debe cumplir con estándares de calidad y buenas prácticas de desarrollo.
RNF09	El sistema debe ser escalable para soportar el crecimiento futuro de clientes y créditos.
RNF10	El sistema debe garantizar la confidencialidad de los datos de los clientes.
RNF11	El sistema debe estar instalado en la infraestructura propia de la organización (local).

Código	Descripción
RNF12	El sistema debe presentar una interfaz gráfica amigable, clara y de fácil uso para el usuario final.
RNF13	El sistema debe contar con mecanismos de recuperación ante fallos o inconsistencias de datos (rollback transaccional + carga manual desde talonario si hay caída).

7. MODELO DE NEGOCIO

El modelo de negocio describe la estructura operativa y lógica que sustenta la gestión de créditos personales en el comercio minorista. Este modelo identifica los actores involucrados, los procesos fundamentales y las reglas de negocio que rigen la operatoria del sistema.

7.1. Actores principales

- **Administrador/Dueño:** Supervisa el funcionamiento general del sistema, controla la concesión de créditos, el cierre de lotes y la emisión de listados.
- **Empleado/Cajero:** Gestiona la carga, modificación y baja de clientes, otorga créditos, registra pagos de cuotas y consulta información relevante para el día a día.
- **Cliente:** Persona física que solicita créditos para financiar la compra de productos en cuotas y realiza los pagos correspondientes.
- **Sistema de Gestión:** Plataforma informática que automatiza y centraliza la administración de clientes, créditos, cuotas, pagos y lotes.

7.2. Procesos clave

- **Registro y mantenimiento de clientes:** Alta, edición, búsqueda y baja de datos personales de los clientes.
- **Otorgamiento de créditos:** Creación de una operación crediticia, estableciendo monto, cantidad de cuotas y tasa de interés. Generación automática de las cuotas asociadas.
- **Administración y seguimiento de cuotas:** Control del estado de cada cuota (pendiente, pagada, en mora) y cálculo de vencimientos en base al lote de origen.
- **Registro de pagos:** Ingreso de pagos de cuotas, asignando método de pago y observaciones, con actualización automática de estados y emisión de recibos.
- **Gestión de lotes:** Agrupamiento de créditos y cuotas en lotes mensuales, cierre de lote y apertura del siguiente, actualización de estados de cuotas vencidas.
- **Consultas e informes:** Visualización de créditos y pagos por cliente, generación de listados de créditos, pagos, clientes.

7.3. Reglas de negocio

- **Tasa de interés:** El interés mensual aplicado a los créditos es el definido en la tabla de variables del sistema y puede ser actualizado solo por el administrador. Por ejemplo, si el interés es del 4% y se quiere pagar en 4 cuotas, el monto total a pagar se calcula sumando el interés según la cantidad de cuotas elegidas:

1 cuota: Valor de compra x 1.04

2 cuotas: Valor de compra x 1.08 / 2

3 cuotas: Valor de compra x 1.12 / 3

4 cuotas: Valor de compra x 1.16 / 4

- **Cantidad de cuotas:** Los créditos pueden otorgarse solo en un rango permitido de cuotas (de 1 a 12 cuotas).

- **Estado de cuotas:** Una cuota solo puede ser pagada si su estado es “pendiente”. Al cerrar un lote, todas las cuotas pendientes cuyo vencimiento haya pasado se marcan como “mora”.
- **Estado de créditos:** Un crédito se marca como “cancelado” automáticamente cuando todas sus cuotas han sido pagadas.
- **Eliminación de clientes:** No se permite eliminar clientes que tengan créditos vigentes o cuotas pendientes.
- **Seguridad y acceso:** El acceso al sistema está protegido por contraseña, y la modificación de credenciales solo puede realizarse por usuarios autenticados.
- **Integridad de datos:** Todo pago debe estar asociado a una cuota existente; no se permite registrar pagos sobre cuotas ya pagadas o anuladas.

8. PROPUESTA DE SOLUCIÓN

Se propone el desarrollo e implementación de una aplicación de escritorio para la gestión de créditos personales en un comercio minorista. El objetivo principal es ordenar y automatizar el registro, control y seguimiento de operaciones (clientes, créditos, cuotas, pagos y lotes), reduciendo tareas manuales y errores y ofreciendo información básica confiable para la operación diaria.

Elementos técnicos clave de la solución:

- Interfaz gráfica amigable: desarrollada en Java Swing, con formularios simples y tablas para consultas rápidas.
- Arquitectura modular: patrón Modelo–Vista–Controlador (MVC) apoyado en DAOs para el acceso a datos y separación clara de responsabilidades, facilitando mantenimiento y futuras extensiones.
- Persistencia: almacenamiento en MySQL (tablas normalizadas para clientes, créditos, cuotas, pagos, lotes y variables). Uso de transacciones en operaciones críticas (crear crédito + cuotas, registrar pago, cerrar lote, anular pago) para asegurar consistencia.
- Automatización de procesos clave: generación de cuotas con interés lineal, marcado automático de mora al cerrar lote, emisión de comprobante de alta de crédito y recibo de pago, actualización de estado del crédito al completarse todas las cuotas.
- Control de acceso: autenticación por contraseña (operativa o maestra). Seguridad avanzada (hash de contraseñas, roles, auditoría) prevista como mejora futura.

A nivel operativo el sistema permite: alta, búsqueda, modificación y baja de clientes; otorgamiento de créditos; consulta de créditos y cuotas; registro y anulación de pagos; emisión de comprobantes (crédito y pago); cierre y apertura de lotes con marcado de mora; y acceso a listados básicos (créditos vigentes, cuotas pendientes, historial de pagos).

La propuesta se plantea con un enfoque incremental: en mejoras futuras se podrán incorporar reportes agregados, exportaciones, notificaciones a clientes, roles con distintos permisos y mejoras de seguridad (hash de contraseñas y auditoría de acciones).

Esta solución reemplaza un manejo manual por un proceso centralizado y consistente, mejorando el control operativo y sentando una base para crecer funcionalmente sin rehacer la estructura principal.

9. ETAPA DE ANÁLISIS

9.1. Diagrama de casos de uso (UML)

El siguiente diagrama de casos de uso resume las principales funcionalidades que el sistema ofrece a los usuarios, así como los procesos automáticos que realiza el sistema:



9.2. Tabla: Casos de uso

Código	Nombre del caso de uso	Actor	Descripción breve
CU01	Ingresar al sistema	Usuario	El usuario accede al sistema ingresando la contraseña operativa o maestra.
CU02	Cambiar contraseña	Usuario	El usuario cambia la contraseña operativa desde el menú Sistema.
CU03	Alta de cliente	Usuario	El usuario registra un nuevo cliente en el sistema.
CU04	Modificar cliente	Usuario	El usuario modifica los datos de un cliente existente.
CU05	Eliminar cliente	Usuario	El usuario elimina un cliente del sistema (solo si no tiene créditos vigentes).
CU06	Buscar / Listar clientes	Usuario	El usuario busca o lista clientes por nombre, documento u otro criterio.

Código	Nombre del caso de uso	Actor	Descripción breve
CU07	Otorgar crédito	Usuario	El usuario otorga un nuevo crédito, asignando monto, cuotas e interés a un cliente y obtiene el comprobante.
CU08	Listar todos los créditos	Usuario	El usuario consulta el listado general de todos los créditos otorgados.
CU09	Consultar créditos por cliente	Usuario	El usuario consulta los créditos asociados a un cliente específico.
CU10	Registrar pago de cuota	Usuario	El usuario registra el pago de una cuota pendiente y el sistema emite el recibo de pago.
CU11	Anular pago	Usuario	El usuario puede anular un pago realizado, revirtiendo la cuota a pendiente.
CU12	Consultar pagos de un cliente	Usuario	El usuario consulta el historial de pagos realizados por un cliente.
CU13	Consultar deuda de un cliente	Usuario	El usuario consulta el resumen de deuda (total, actual, en mora) y detalle de cuotas abiertas de un cliente.
CU14	Cerrar lote	Usuario	El usuario cierra el lote actual, marca cuotas en mora y crea el siguiente.
CU15	Ver lote actual	Usuario	El usuario consulta los datos del lote operativo actual.

9.3. Tabla de casos de uso <<include>> y <<extend>>

Código	Nombre del caso de uso	Actor	Descripción breve
CU07a	Generar cuotas (<<include>>)	-	El sistema genera automáticamente las cuotas al otorgar un crédito.
CU07b	Emitir comprobante de crédito (<<include>>)	-	El sistema emite el comprobante de alta de crédito.
CU10a	Emitir recibo de pago (<<include>>)	-	El sistema emite un recibo por el pago de cuota registrado.
CU14a	Marcar cuotas en mora (<<include>>)	-	El sistema marca como "mora" las cuotas pendientes alcanzadas al cerrar el lote.

9.4. Fichas de casos de uso

A continuación, se presentan ejemplos de fichas completas para algunos de los casos más representativos.

CU01 – Ingresar al sistema

Código	CU01
Nombre	Ingresar al sistema
Actor	Usuario
Descripción	Permite al usuario acceder al sistema mediante el ingreso de una contraseña válida.
Precondiciones	El sistema debe estar instalado y la base de datos accesible; el usuario debe conocer la contraseña operativa o maestra.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa la contraseña. 2. El sistema valida la contraseña. 3. Si es válida, muestra el menú principal con las opciones habilitadas.
Flujos alternativos	Si la contraseña es incorrecta, el sistema muestra un mensaje de error y solicita el reingreso.
Postcondiciones	El usuario accede a las funcionalidades del sistema.
Reglas de negocio asociadas	Solo usuarios autenticados pueden operar el sistema.

CU03 – Alta de cliente

Código	CU03
Nombre	Alta de cliente
Actor	Usuario
Descripción	Permite registrar un nuevo cliente en la base de datos.
Precondiciones	El usuario debe estar autenticado.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede al formulario “Alta de Cliente”. 2. Completa nombre, documento y otros campos requeridos. 3. El sistema verifica que el documento no esté registrado (único). 4. Si es válido, almacena el cliente en la base de datos.
Flujos alternativos	Si el documento ya existe, el sistema muestra un error y no permite el alta.
Postcondiciones	El cliente queda registrado y disponible para operaciones.
Reglas de negocio asociadas	El documento debe ser único por cliente.

CU07 – Otorgar crédito

Código	CU07
Nombre	Otorgar crédito
Actor	Usuario
Descripción	Permite otorgar un crédito a un cliente, generando automáticamente las cuotas y emitiendo comprobante.
Precondiciones	Debe existir al menos un cliente; el lote debe estar abierto.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede al formulario de alta de crédito. 2. Selecciona un cliente, ingresa monto y cantidad de cuotas. 3. El sistema muestra la tasa de interés vigente. 4. El usuario confirma la operación. 5. El sistema crea el crédito, genera las cuotas y emite comprobante.
Flujos alternativos	Si el usuario ingresa valores inválidos (monto/cantidad), el sistema muestra error y no permite continuar.
Postcondiciones	El crédito y sus cuotas quedan registrados; se emite comprobante.

Reglas de negocio asociadas	Tasa y rango de cuotas según parámetros en tabla de variables.
-----------------------------	--

CU10 – Registrar pago de cuota

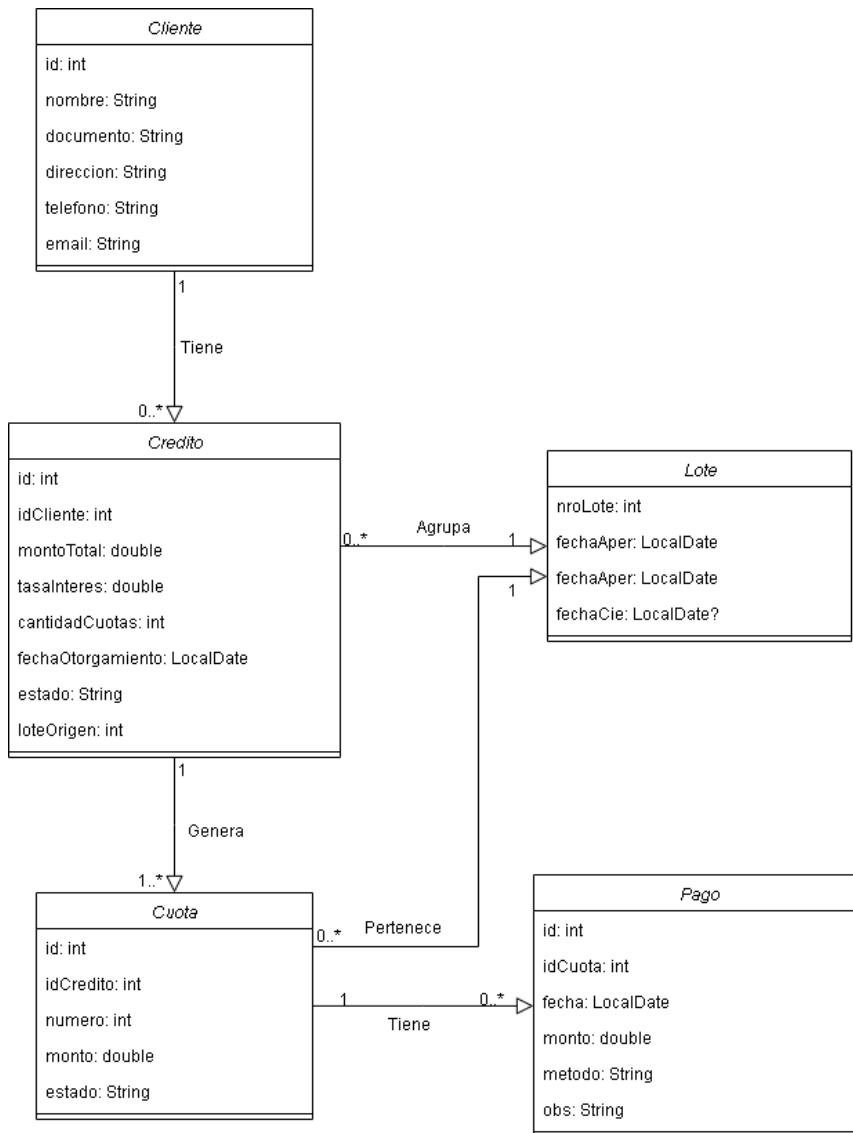
Código	CU10
Nombre	Registrar pago de cuota
Actor	Usuario
Descripción	Permite registrar el pago completo de una cuota pendiente y emitir el recibo correspondiente.
Precondiciones	El cliente debe tener al menos una cuota pendiente de pago.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede al panel de pago de cuota y selecciona cliente y cuota pendiente. 2. Ingresa método de pago y observaciones (opcional). 3. El sistema registra el pago, marca la cuota como pagada y emite el recibo. 4. Si era la última cuota del crédito, el crédito se marca como “cancelado”.
Flujos alternativos	Si el usuario cancela la operación, no se registra el pago.
Postcondiciones	El pago queda registrado; la cuota se marca como pagada; el recibo está disponible.
Reglas de negocio asociadas	Solo cuotas pendientes pueden pagarse; un pago por cuota.

CU13 – Consultar deuda de un cliente

Código	CU13
Nombre	Consultar deuda de un cliente
Actor	Usuario
Descripción	Permite consultar el resumen de deuda de un cliente (total, actual y en mora), así como el detalle de cuotas abiertas (pendiente/mora), indicando si son futuras.
Precondiciones	El cliente debe existir; puede o no tener créditos/cuotas abiertas.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona un cliente y accede a “Ver Deuda”. 2. El sistema calcula y muestra: <ul style="list-style-type: none"> - Deuda TOTAL (todas las cuotas pendiente/mora) - Deuda ACTUAL (pendiente/mora con vencimiento igual o anterior al lote actual) - Deuda en MORA (cuotas en mora) 3. El sistema lista el detalle de cuotas abiertas, indicando si son futuras (no vencidas aún).
Flujos alternativos	Si el cliente no posee créditos/cuotas abiertas, se muestra cero en los totales y detalle vacío.
Postcondiciones	El usuario visualiza el estado de deuda actual del cliente.
Reglas de negocio asociadas	Solo cuotas pendiente/mora se incluyen en los totales; cálculo de mora según cierre de lote.

9.5. Modelo de dominio (diagrama de clases simplificado)

El modelo de dominio identifica las principales entidades y sus relaciones. Se representa en el siguiente diagrama:



- **Cliente:** Persona que solicita créditos (atributos: id, nombre, documento, etc.).
- **Credito:** Crédito otorgado, asociado a un cliente (atributos: id, idCliente, monto, tasaInteres, cantidadCuotas, fecha, estado, lote).
- **Cuota:** Cuota de un crédito, con número, monto y estado.
- **Pago:** Registro de un pago realizado para una cuota específica.
- **Lote:** Agrupación de créditos/cuotas por período.

La relación entre Cuota y Lote (por vencimiento) no existe como FK directa en la base, sino que se deriva en tiempo de ejecución usando la fórmula: $\text{lote_vencimiento} = \text{crédito.lote_origen} + \text{número_cuota}$.

La clase Variables es un singleton y provee parámetros globales (contraseñas, interés mensual y contadores globales).

Relaciones entre clases principales

Cliente — Credito

- **Un Cliente puede tener muchos Créditos.**
- **Un Crédito pertenece a un único Cliente.**

- **Relación:**
 - Cliente (1) \leftrightarrow (0..*) Credito

Credito — Cuota

- **Un Crédito tiene muchas Cuotas.**
- **Cada Cuota pertenece a un único Crédito.**
- **Relación:**
 - Credito (1) \leftrightarrow (1..*) Cuota

Cuota — Pago

- **Una Cuota puede tener cero o más Pagos** (el sistema, por el momento, solo puede registrar un pago por cuota, pero se puede ampliar).
- **Cada Pago pertenece a una única Cuota.**
- **Relación:**
 - Cuota (1) \leftrightarrow (0..*) Pago

Credito — Lote

- **Un Crédito pertenece a un único Lote** (campo loteOrigen).
- **Un Lote agrupa muchos Créditos.**
- **Relación:**
 - Lote (1) \leftrightarrow (0..*) Credito

Cuota — Lote (lote de vencimiento)

- El vencimiento de la cuota se deriva de la lógica: **lote_vencimiento = credito.lote_origen + numero**
No es una FK directa, pero conceptualmente:
 - Lote (1) \leftrightarrow (0..*) Cuota (por vencimiento)

Tabla de relación:

Clase origen	Clase destino	Relación	Cardinalidad
Cliente	Credito	tiene	1 a 0..*
Credito	Cuota	genera	1 a 1..*
Cuota	Pago	tiene	1 a 0..*
Lote	Credito	agrupa	1 a 0..*
Cuota	Lote	pertenece a	1 a 1 (lote_origen + numero, lógica)

10. ETAPA DE DISEÑO

10.1. Descripción de la arquitectura general (MVC)

El sistema implementa el patrón de arquitectura Modelo-Vista-Controlador (MVC):

- **Modelo:** Clases Java que representan las entidades de negocio y persistencia, incluyendo: Cliente, Crédito, Cuota, Pago, Lote, y Variables.
Cada clase mapea directamente a una tabla principal de la base de datos.
- **Vista:** Interfaces gráficas implementadas en Java Swing. Incluyen paneles para:
 - Alta, modificación, baja y listado de clientes
 - Otorgamiento y listado de créditos
 - Registro, listado y anulación de pagos
 - Gestión y cierre de lotes
 - Cambio de contraseña
- **Controlador:** Clases que contienen la lógica de negocio y coordinan la interacción entre vista y modelo.
Ejemplos: ClienteController, CreditoController, PagoController, LoteController, PassController.
- **Persistencia:** El acceso a datos se realiza a través del paquete Dao, que implementa la comunicación con la base de datos MySQL mediante JDBC.
Cada entidad principal tiene su DAO correspondiente (ClienteDAO, CreditoDAO, etc.).

Esquema de módulos/componentes:

Módulo/Carpeta	Descripción
Controller/	Lógica de negocio y coordinación (uno por entidad principal)
Dao/	Acceso a datos (CRUD y lógica de persistencia por entidad)
Model/	Definición de entidades de dominio (POJOs)
View/	Interfaces gráficas para interacción con el usuario
MainWindow.java	Ventana principal y menú de navegación

10.2. Diseño de pantallas/interfaz

En esta etapa se definieron prototipos de baja fidelidad (wireframes) producidos con draw.io, cuyo objetivo es el de visualizar tempranamente la estructura de navegación (menú principal por dominios: Clientes, Créditos, Pagos, Lote y Sistema), delimitar los componentes esenciales de cada pantalla y validar con los usuarios involucrados la secuencia de tareas (alta de cliente → otorgamiento de crédito → pago de cuota).

Los bocetos NO representan aún decisiones finales de color, tipografía o mensajes, sino la distribución funcional mínima necesaria para cumplir los requerimientos elicitados.

Las figuras siguientes muestran algunas de las pantallas propuestas.

- **Panel de Login:** Autenticación por contraseña.

The login screen features a header bar with the text 'Sistema de Préstamos'. Below this, the instruction 'Ingrese la contraseña:' is displayed above a text input field. A blue button labeled 'Ingresar' is positioned below the input field. At the bottom of the page, the text 'Versión 1.0 - Sistema de Préstamos Personales' is visible.

- **Panel de Alta de Cliente:** Formulario para registrar clientes, con campos principales y botón de guardar.

The 'Alta de Cliente' form is located under the 'Clientes' tab of the 'Sistema de Préstamos Personales'. It contains five input fields labeled 'Nombre:', 'Documento:', 'Dirección:', 'Teléfono:', and 'Email:'. A blue 'Guardar' button is at the bottom.

- **Panel Modificar Cliente:** Mismo diseño que alta, con campos precargados y botón para guardar cambios.
- **Panel Baja Cliente:** Permite seleccionar un cliente existente y eliminarlo del sistema.

The 'Baja de Cliente' form is located under the 'Clientes' tab. It features a dropdown menu for 'Seleccionar Cliente:' with 'Juan Pérez (ID: 1)' selected, and a text input for 'ID de Cliente:' with the value '1'. A blue 'Eliminar' button is at the bottom.

- **Panel de Alta de crédito:** Selección de cliente, ingreso de monto, cuotas, y confirmación del otorgamiento.

The screenshot shows the 'Alta de Crédito' form within the 'Sistema de Préstamos Personales'. The form has a header with navigation tabs: 'Clientes', 'Créditos', 'Pagos', 'Lote', and 'Sistema'. The 'Alta de Crédito' section contains the following fields and controls:

- Cliente:** A dropdown menu showing 'Juan Pérez (ID: 1)'.
- Importe (capital):** A text input field containing '1000'.
- Cantidad de cuotas:** A text input field containing '4'.
- Interés mensual vigente:** A text input field containing '5,00 %'.
- Crear Crédito:** A blue button to submit the form.

- **Panel de Pagar Cuota:** Selección de cliente y cuota pendiente, registro de pago y emisión de recibo.

The screenshot shows the 'Pagar Cuota' form within the 'Sistema de Préstamos Personales'. The form has a header with navigation tabs: 'Clientes', 'Créditos', 'Pagos', 'Lote', and 'Sistema'. The 'Pagar Cuota' section contains the following fields and controls:

- Cliente:** A dropdown menu showing 'Juan Pérez (ID: 1)'.
- Cargar cuotas pendientes:** A blue button to load pending installments.
- Cuota:** A text input field containing '1'.
- Método de pago:** A text input field containing 'efectivo'.
- Pagar:** A green button to submit the payment.

- **Panel de Lote:** Visualización de lote actual, cierre de periodo y apertura de nuevo lote.

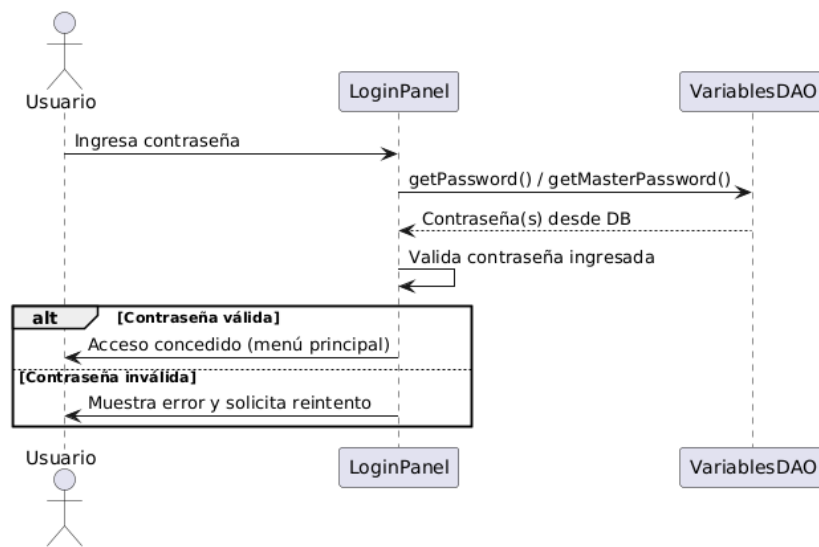
The screenshot shows the 'Lote Actual' form within the 'Sistema de Préstamos Personales'. The form has a header with navigation tabs: 'Clientes', 'Créditos', 'Pagos', 'Lote', and 'Sistema'. The 'Lote Actual' section contains the following information and controls:

- Lote Actual N°:** 4
- Abierto el:** 2025-08-01
- Estado:** ABIERTO
- Cerrar Lote y Crear Siguiente:** A red button to close the current lot and create the next one.

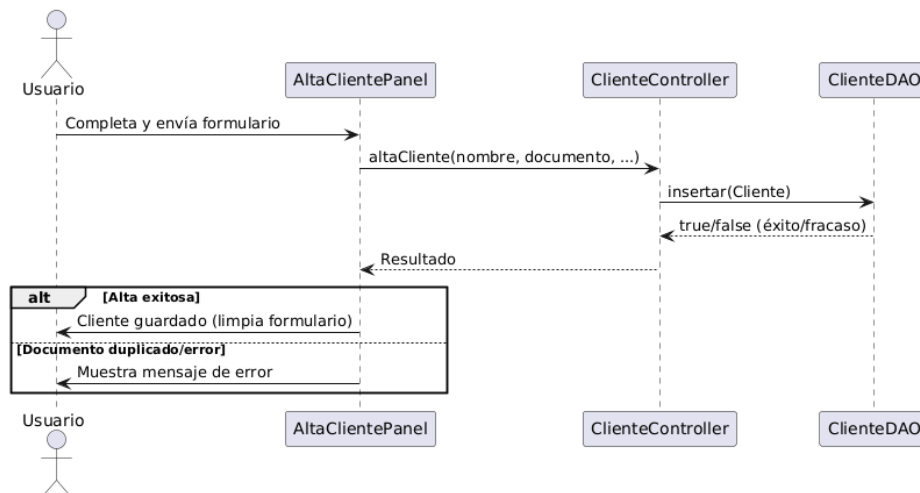
10.3. Diagramas de secuencia

Los diagramas de secuencia se realizaron utilizando la herramienta online www.planttext.com

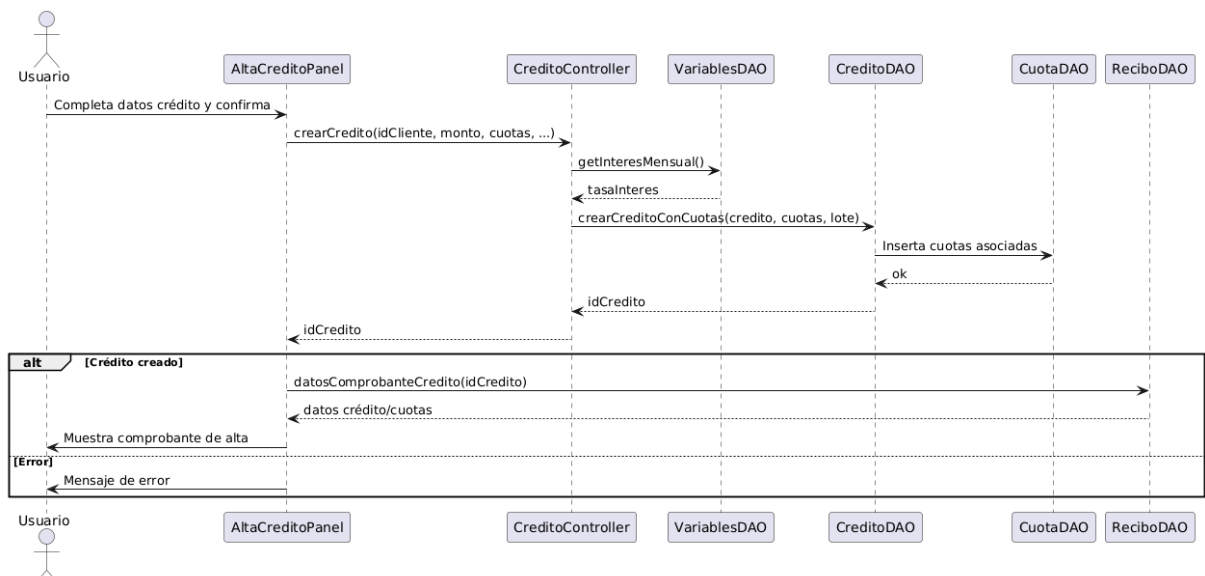
CU01 – Ingresar al sistema:



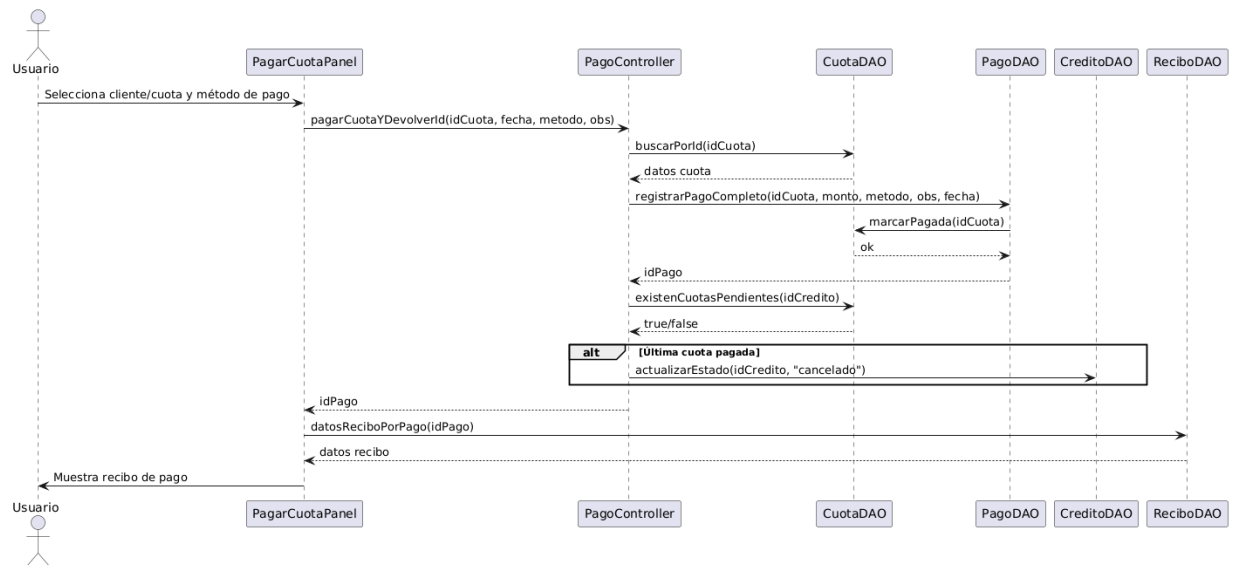
CU03 – Alta de cliente:



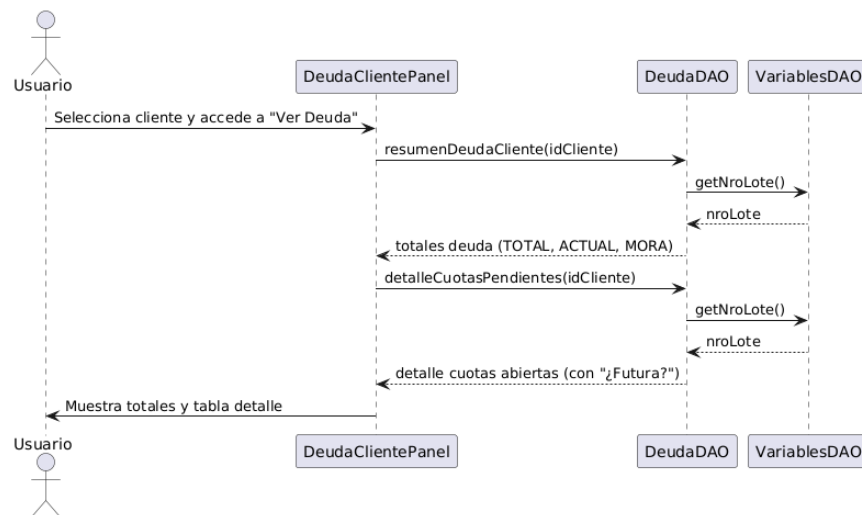
CU07 – Otorgar crédito:



CU10 – Registrar pago de cuota:



CU13 – Consultar deuda de un cliente:



10.4. Matriz de trazabilidad de requerimientos funcionales (RF)

Código RF	Requerimiento funcional	Casos de uso que lo implementan
RF01	Ingreso al sistema	CU01 Ingresar al sistema
RF02	Alta de cliente	CU03 Alta de cliente
RF03	Modificación de cliente	CU04 Modificar cliente
RF04	Baja de cliente	CU05 Eliminar cliente
RF05	Alta de crédito	CU07 Otorgar crédito
RF06	Consulta y listado de clientes y créditos	CU06 Buscar/Listar clientes CU08 Listar todos los créditos CU09 Consultar créditos por cliente
RF07	Registro de pago de cuota	CU10 Registrar pago de cuota
RF08	Consulta de historial y deuda por cliente	CU09 Consultar créditos por cliente

Código RF	Requerimiento funcional	Casos de uso que lo implementan
		CU12 Consultar pagos de un cliente CU13 Consultar deuda de un cliente
RF09	Cierre de lote y generación de siguiente	CU14 Cerrar lote
RF10	Cambio de contraseña	CU02 Cambiar contraseña

10.5. Matriz de trazabilidad de requerimientos NO funcionales (RNF)

Código RNF	Requerimiento No Funcional	Artefacto que lo implementa / evidencia
RNF01	El sistema debe estar desarrollado en Java, con interfaz gráfica implementada en Swing.	Código fuente (archivos .java en Model, View, Controller); paneles y ventanas en View/*.java
RNF02	El sistema utilizará una base de datos MySQL para la persistencia de los datos.	Script creditos.sql, clases Dao/*, archivo Dao/ConexionMySQL.java
RNF03	El acceso a los datos debe realizarse mediante un patrón Modelo-Vista-Controlador (MVC) y DAO para persistencia.	Estructura de carpetas (Model, View, Controller, Dao), README sección arquitectura, implementación en código fuente
RNF04	El sistema debe estar disponible durante el horario operativo del comercio.	Manual de uso, configuración de despliegue (no directamente implementado en código)
RNF05	El sistema debe garantizar la seguridad de los datos, solicitando contraseña al iniciar el programa.	Panel de Login (View/LoginPanel.java), validación en VariablesDAO, ficha CU01, README
RNF06	El sistema debe permitir realizar copias de respaldo periódicas de la base de datos.	Script SQL y manual de respaldo (indicaciones en README)
RNF07	El sistema debe almacenar todas las fechas en formato estándar internacional (ISO 8601).	Definición en tablas SQL (date), uso de java.time.LocalDate en Model y Dao
RNF08	El sistema debe cumplir con estándares de calidad y buenas prácticas de desarrollo.	Estructura del código, uso de patrones, comentarios, validación por docentes
RNF09	El sistema debe ser escalable para soportar el crecimiento futuro de clientes y créditos.	Diseño de tablas relacionales y claves, separación en capas, documentación en README
RNF10	El sistema debe garantizar la confidencialidad de los datos de los clientes.	Acceso restringido por contraseña, almacenamiento local sin exposición pública
RNF11	El sistema debe estar instalado en la infraestructura propia de la organización (local).	Instrucciones de instalación local, no requiere servicios cloud
RNF12	El sistema debe presentar una interfaz gráfica amigable, clara y de fácil uso para el usuario final.	Prototipos de interfaz (wireframes), paneles Swing, validación con usuarios, screenshots en documentación
RNF13	El sistema debe contar con mecanismos de recuperación ante fallos o inconsistencias de datos	Transacciones en operaciones críticas (crear crédito, pago, cierre de lote), rollback en DAOs, instrucciones de carga manual

11. ETAPA DE IMPLEMENTACIÓN

11.1. Tecnologías utilizadas

- Lenguaje de programación: Java SE 17+
- (Recomendado para compatibilidad y soporte a largo plazo; el proyecto es compatible con Java 8+, pero se sugiere Java 17 o superior).
- Interfaz gráfica: Java Swing
- Base de datos: MySQL 8.x
- Acceso a datos: JDBC utilizando DAOs específicos para cada entidad
- Control de versiones y distribución: Repositorio público en GitHub
- Herramientas adicionales: draw.io (wireframes), PlantUML (diagramas), IDE (Visual Studio Code)

11.2. Estructura de carpetas y clases principales.

El código fuente está organizado siguiendo el patrón MVC y la separación de responsabilidades:

- **/Controller/**
Controladores de la lógica de negocio y coordinación entre la vista y el modelo:
 - ClienteController
 - CreditoController
 - PagoController
 - LoteController
 - PassController (cambio de contraseña)
- **/Dao/**
Acceso a datos (CRUD y consultas especializadas) para cada entidad principal:
 - ClienteDAO, CreditoDAO, CuotaDAO, PagoDAO, LoteDAO, VariablesDAO, ReciboDAO, ConexionMySQL
- **/Model/**
Definición de entidades y objetos de dominio:
 - Cliente, Credito, Cuota, Pago, Lote, Variables
- **/View/**
Interfaces gráficas para la interacción con el usuario:
 - Panel de login
 - Paneles de clientes (alta, baja, modificación, búsqueda, listado, deuda)
 - Paneles de créditos (alta, listado por cliente, listado general, comprobante)
 - Paneles de pagos (registrar, anular, listar)
 - Paneles de lote (visualizar, cerrar lote)
 - Panel de cambio de contraseña
- **MainWindow.java**
Clase principal ubicada en la raíz del proyecto. Lanza la aplicación, administra la ventana principal y el menú de navegación.

11.3. Repositorio de código fuente

Todo el código fuente, scripts SQL y documentación asociada se encuentra disponible en el siguiente repositorio público:

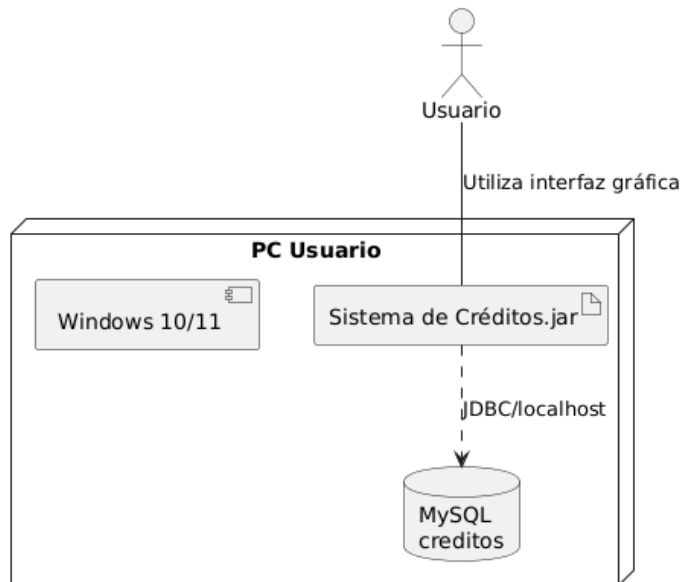
- <https://github.com/Titoyco/seminario>

11.4. Diagrama de despliegue

El siguiente diagrama de despliegue muestra la arquitectura física del sistema, incluyendo los componentes principales, las conexiones entre la aplicación de escritorio y la base de datos MySQL, y la ubicación de los archivos de respaldo:

El sistema se desplegará en una única PC con Windows 10/11. Esta máquina contendrá:

- El servidor de base de datos MySQL 8.x, donde se almacenará la base completa del sistema.
- El archivo de base de datos (creditos.sql) ya cargado y configurado.
- La aplicación Java de escritorio (archivo ejecutable .jar).
- El usuario interactúa directamente con la interfaz gráfica Java Swing, que se conecta localmente al servidor MySQL utilizando JDBC.
- Este esquema garantiza la simplicidad de instalación y operación, y permite ampliar a futuro el acceso multiusuario en red local si se requiere.



12. ETAPA DE PRUEBAS

12.1. Tipos de pruebas realizadas

- **Pruebas funcionales:** Alta, modificación y baja de clientes; otorgamiento de créditos; registro y anulación de pagos; cierre de lote y generación de listados.
- **Pruebas de integración:** Flujos completos entre diferentes módulos (por ejemplo, alta de cliente, otorgamiento de crédito, pago de cuota, consulta de listados).
- **Pruebas de interfaz:** Validación visual y de usabilidad en los paneles Swing.
- **Pruebas de base de datos:** Verificación de integridad referencial, restricciones de unicidad y funcionamiento de las claves foráneas.

12.2. Plan de pruebas y casos de prueba

A continuación, se detallan los casos de prueba seleccionados, las entradas consideradas, los valores frontera y el resultado esperado para los 5 casos de uso más representativos del sistema:

Caso de uso: CU01 – Ingresar al sistema

Nº	Caso de Prueba	Entrada	Valor límite/borde	Resultado esperado
1	Login correcto	Contraseña válida	Contraseña exacta	Acceso concedido, menú principal
2	Login incorrecto	Contraseña incorrecta	Error en 1 letra	Mensaje de error, reintento
3	Campo vacío	(sin ingresar nada)	Vacío	Mensaje de error, reintento
4	Uso de contraseña maestra	Contraseña maestra válida	Contraseña "master"	Acceso concedido
5	Múltiples intentos fallidos	Varias contraseñas erróneas seguidas	3 intentos seguidos	No bloquea, permite reintentar

Caso de uso: CU03 – Alta de cliente

Nº	Caso de Prueba	Entrada	Valor límite/borde	Resultado esperado
1	Alta válida	Todos los campos correctos	Nombre/documento normales	Cliente guardado
2	Documento duplicado	Documento ya existente en otro cliente	Igual a otro	Mensaje de error, no guarda

N°	Caso de Prueba	Entrada	Valor límite/borde	Resultado esperado
3	Campo obligatorio vacío	Falta nombre/documento/email	Vacío (cada campo)	Mensaje de error, no guarda
4	Longitud máxima nombre	Nombre de 100 caracteres	100 caracteres (límite tabla)	Cliente guardado
5	Longitud máxima documento	Documento de 50 caracteres	50 caracteres (límite tabla)	Cliente guardado
6	Longitud máxima email	Email de 100 caracteres	100 caracteres (límite tabla)	Cliente guardado

Caso de uso: CU07 – Otorgar crédito

N°	Caso de Prueba	Entrada	Valor límite/borde	Resultado esperado
1	Alta de crédito válida	Monto y cuotas válidos, cliente existente	Monto positivo, cuotas 1–12	Crédito y cuotas generados, comprobante
2	Cliente no seleccionado	Sin elegir cliente	Cliente null	Mensaje de error, no guarda
3	Monto igual a cero	Monto = 0	0	Mensaje de error, no guarda
4	Cuotas igual a cero	Cuotas = 0	0	Mensaje de error, no guarda
5	Cuotas en mínimo permitido	Cuotas = 1	1	Crédito generado, 1 cuota
6	Cuotas en máximo permitido	Cuotas = 12	12 (según regla)	Crédito generado, 12 cuotas
7	Cuotas fuera de rango	Cuotas = 13	13 (>12)	Mensaje de error, no guarda
8	Monto máximo permitido	Monto = 99999999.99	Límite decimal(12,2)	Crédito generado si dentro de límites

Caso de uso: CU10 – Registrar pago de cuota

N°	Caso de Prueba	Entrada	Valor límite/borde	Resultado esperado
1	Pago válido	Selección de cuota pendiente	Cuota en estado “pendiente”	Pago registrado, cuota pagada, recibo
2	Pago a cuota ya pagada	Selección de cuota “pagada”	Estado “pagada”	Mensaje de error, no registra
3	Método de pago vacío	No selecciona método	Vacío	Mensaje de error, no registra
4	Observaciones muy largas	Observaciones de 200 caracteres	200 caracteres	Pago registrado si dentro del límite
5	Última cuota del crédito	Pago a última cuota pendiente	Última cuota	Crédito pasa a estado “cancelado”

Caso de uso: CU13 – Consultar deuda de un cliente

N°	Caso de Prueba	Entrada	Valor límite/borde	Resultado esperado
1	Cliente con deuda	Cliente con cuotas pendientes y en mora	-	Muestra totales y detalle correctamente
2	Cliente sin créditos	Cliente sin créditos	-	Totales = 0, detalle vacío
3	Cliente con todas las cuotas pagadas	Cliente con créditos pero sin deuda	-	Totales = 0, detalle vacío
4	Lote actual en frontera	Cuotas con lote_vencimiento = lote actual	Límite de cálculo “actual”	Suma correcta en deuda “actual”
5	Todas cuotas futuras	Todas las cuotas lote_venc > lote actual	Todas “futuras”	Deuda total mayor que “actual”

12.3. Observaciones generales

- Todos los casos de prueba deben ejecutarse con la base de datos en estado conocido y restaurable, dejando evidencia (capturas o logs).
- Los valores límite y de frontera aseguran que el sistema responde correctamente ante entradas extremas o inesperadas.
- Los resultados esperados deben coincidir con los mensajes y estados definidos en la interfaz y la lógica del sistema.

13. MODELO DE DATOS Y PERSISTENCIA

13.1. Modelo relacional de datos

El sistema utiliza una base de datos relacional MySQL, estructurada en las siguientes tablas principales:

- **clientes:** almacena los datos personales de cada cliente.
 - **PK:** id (INT, autoincremental)
 - **UK:** documento (único)
- **creditos:** registra cada crédito otorgado a un cliente.
 - **PK:** id (INT, autoincremental)
 - **FK:** id_cliente → clientes(id)
 - **FK:** lote_origen → lotes(nro_lote)
- **cuotas:** contiene las cuotas generadas para cada crédito.
 - **PK:** id (INT, autoincremental)
 - **FK:** id_credito → creditos(id)
 - **UK:** (id_credito, numero) (una sola cuota N por crédito)
- **pagos:** registra los pagos asociados a cuotas.
 - **PK:** id (INT, autoincremental)
 - **FK:** id_cuota → cuotas(id)
- **lotes:** agrupa los créditos/cuotas por períodos.
 - **PK:** nro_lote (INT)
- **variables:** almacena parámetros globales y de control del sistema (contraseña, interés vigente, nro lote/credito actual).
 - **PK:** id=1 (única fila)

Justificación de la normalización

Todas las tablas cumplen al menos la **Tercera Forma Normal (3FN)**:

- No existen atributos multivaluados ni repetidos.
- Cada tabla contiene únicamente atributos que dependen de la clave primaria.
- En los casos de claves foráneas, las dependencias son plenas y no transitivas.
- Los campos de búsqueda frecuente (documento, lote, estado) cuentan con índices.

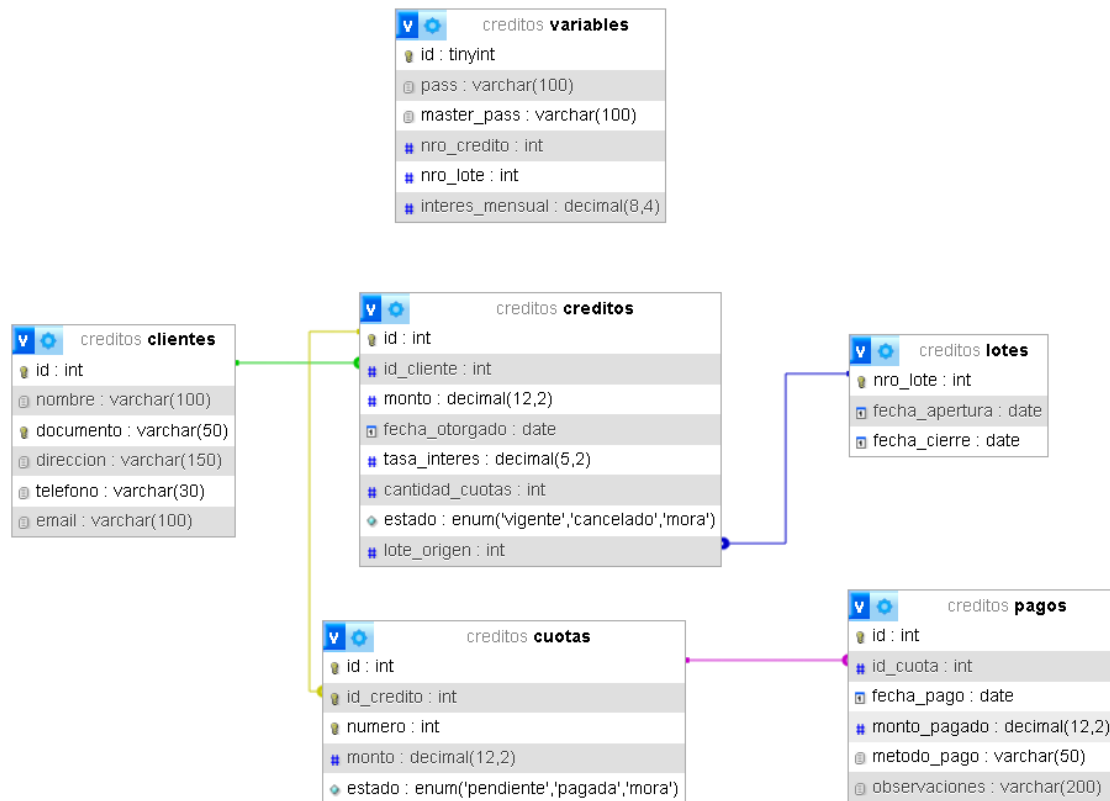
Relación con el modelo de dominio:

Cada entidad del modelo de dominio (Cliente, Credito, Cuota, Pago, Lote, Variables) corresponde directamente a una tabla en la base de datos. Las relaciones entre entidades en el modelo de clases se implementan mediante claves foráneas en el modelo relacional.

13.2. Diagrama Entidad-Relación (ER)

El siguiente diagrama ER (generado con phpMyAdmin de forma automática a partir de la creación de las tablas y la declaración de sus claves primarias y foráneas en el script `creditos.sql`) representa las entidades y relaciones centrales del sistema. No es un diagrama conceptual dibujado manualmente, sino la proyección directa del modelo físico implementado en MySQL. Las líneas entre tablas corresponden a las FOREIGN KEY definidas (con sus

cardinalidades implícitas 1:N). La tabla “variables” actúa como una tabla de configuración con una única fila y, por ello, no se materializan relaciones adicionales más allá de su uso lógico por la aplicación.



Explicación de las entidades y relaciones:

- Un Cliente puede tener varios Créditos (1:N).
- Un Crédito genera varias Cuotas (1:N).
- Una Cuota puede estar asociada a varios Pagos (1:N), aunque en este sistema por ahora es uno por cuota.
- Un Crédito pertenece a un Lote (N:1).
- Una Cuota pertenece a un Crédito y se agrupa por Lote de manera lógica (no física).
- La tabla variables centraliza los parámetros globales y los contadores.

13.3. Definición de base de datos y scripts

La estructura física y los datos de ejemplo se encuentran en el archivo [creditos.sql](#), incluido en el repositorio del proyecto. (el usuario definido es root y no tiene clave)

Correspondencia del script con el modelo:

- Las instrucciones CREATE TABLE y ALTER TABLE implementan las entidades y relaciones tal como se describen en el modelo de dominio y el diagrama ER.
- Se definen claves primarias, foráneas y restricciones de unicidad en los campos relevantes.
- La tabla “variables” asegura la gestión centralizada de parámetros del sistema.

13.4. Consultas SQL de ejemplo

Se deja constancia que todas las operaciones de inserción, actualización y borrado se realizan con integridad referencial garantizada por las claves foráneas y las restricciones de la base de datos definidas en el script. A continuación, se presentan ejemplos de consultas SQL reales utilizadas en el sistema:

a) Insertar un nuevo cliente

```
INSERT INTO clientes (nombre, documento, direccion, telefono, email)
VALUES ('Juan Pérez', '12345678', 'Calle Falsa 123', '1122334455', 'juan.perez@mail.com');
```

b) Insertar un nuevo crédito y sus cuotas

(Las cuotas se generan automáticamente en la lógica de negocio y se insertan en transacción)

```
-- Insertar el crédito
INSERT INTO creditos (id_cliente, monto, fecha_otorgado, tasa_interes, cantidad_cuotas,
estado, lote_origen)
VALUES (1, 10000.00, '2025-09-27', 5.00, 3, 'vigente', 5);

-- Insertar las cuotas asociadas (por cada cuota)
INSERT INTO cuotas (id_credito, numero, monto, estado)
VALUES (21, 1, 3500.00, 'pendiente'), (21, 2, 3500.00, 'pendiente'), (21, 3, 3500.00,
'pendiente');
```

c) Registrar un pago de cuota

```
INSERT INTO pagos (id_cuota, fecha_pago, monto_pagado, metodo_pago, observaciones)
VALUES (1, '2025-09-27', 3500.00, 'efectivo', '');
```

d) Consulta de clientes

```
SELECT * FROM clientes;
```

e) Consulta de créditos por cliente

```
SELECT * FROM creditos WHERE id_cliente = 1;
```

f) Consulta de cuotas pendientes de un cliente

```
SELECT cu.id AS id_cuota, cu.numero, cu.monto, cr.id AS id_credito
FROM cuotas cu
JOIN creditos cr ON cr.id = cu.id_credito
WHERE cr.id_cliente = 1 AND cu.estado='pendiente' AND cr.estado='vigente'
ORDER BY cr.id DESC, cu.numero ASC;
```

g) Borrado físico de un cliente *(solo si no tiene créditos activos)*

```
DELETE FROM clientes WHERE id = 1;
```


13.5. Capturas de pantalla de las consultas tomadas de phpMyAdmin:

SELECT * FROM `clientes`

Perfilando [Editar en línea][Editar][Explicar SQL][Crear código PHP][Actualizar]

1

>>>

Mostrar todo

Número de filas: 25

Filtrar filas:

Ordenar según la clave:

Opciones extra

			id	nombre	documento	direccion	telefono	email				
<input type="checkbox"/>		Editar		Copiar		Borrar	1	Juan Pérez	12345678	Calle Falsa 123	1122334455	juan.perez@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	2	María López	22334455	Av. Siempre Viva 742	1133445566	maria.lopez@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	3	Carlos García	33445566	Diagonal 80 1002	1144556677	carlos.garcia@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	4	Laura Fernández	44556677	Ruta 8 km 58	1155667788	laura.fernandez@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	5	Luis Gómez	55667788	San Martín 350	1166778899	luis.gomez@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	6	Ana Torres	66778899	Belgrano 200	1177889900	ana.torres@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	7	Diego Ramírez	77889900	Mitre 88	1188990011	diego.ramirez@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	8	Juancho Perez	23432564	Islas Malvinas 123	11432292	jpereztw@hotmail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	9	Mauro Herrera	99001122	Moreno 450	1100112233	mauro.herrera@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	10	Julia Castro	10111213	Maipú 25	1111223344	julia.castro@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	11	Pedro Suárez	12131415	Itzaingó 78	1122334456	pedro.suarez@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	12	Cecilia Ríos	14151617	Colón 300	1133445567	cecilia.rios@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	13	Emilia Medina	16171819	9 de Julio 100	1144556678	emilia.medina@mail.com
<input type="checkbox"/>		Editar		Copiar		Borrar	14	Nicolás Vega	18192021	Santa Fe 222	1155667789	nicolas.vega@mail.com

SELECT * FROM credits WHERE id_cliente = 1;

Perfilando [Editar en línea][Editar][Explicar SQL][Crear código PHP][Actualizar]

Mostrar todo

Número de filas: 25

Filtrar filas:

Ordenar según la clave: Ninguna

Opciones extra

				id	id_cliente	monto	fecha_otorgado	tasa_interes	cantidad_cuotas	estado	lote_origen				
<input type="checkbox"/>		Editar		Copiar		Borrar	1		1	10000.00	2025-04-02	10.00	3	vigente	0
<input type="checkbox"/>		Editar		Copiar		Borrar	21		1	1000.00	2025-09-20	5.00	5	vigente	4
<input type="checkbox"/>		Editar		Copiar		Borrar	22		1	100000.00	2025-09-20	5.00	4	vigente	4
<input type="checkbox"/>		Editar		Copiar		Borrar	23		1	1000.00	2025-09-28	5.00	4	vigente	4

SELECT cu.id AS id_cuota, cu.numero, cu.monto, cr.id AS id_credito FROM cuotas cu JOIN credits cr ON cr.id = cu.id_credito WHERE cr.id_cliente = 1 AND cu.estado='pendiente' AND cr.estado='vigente' ORDER BY cr.id DESC, cu.numero ASC;

Perfilando [Editar en línea][Editar][Explicar SQL][Crear código PHP][Actualizar]

Mostrar todo

Número de filas: 25

Filtrar filas:

Ordenar según la clave: Ninguna

Opciones extra

id_cuota	numero	monto	id_credito
58	1	300.00	23
59	2	300.00	23
60	3	300.00	23
61	4	300.00	23
55	2	30000.00	22
56	3	30000.00	22
57	4	30000.00	22
49	1	200.00	21
50	2	200.00	21
51	3	200.00	21
52	4	200.00	21
53	5	200.00	21
3	3	3667.34	1

14. DEFINICIONES DE COMUNICACIÓN

En este proyecto, la aplicación se ejecuta de manera local en una única computadora con sistema operativo Windows 10/11.

La única comunicación relevante se produce entre la aplicación de escritorio Java (Swing) y el servidor de base de datos MySQL, ambos instalados en la misma máquina.

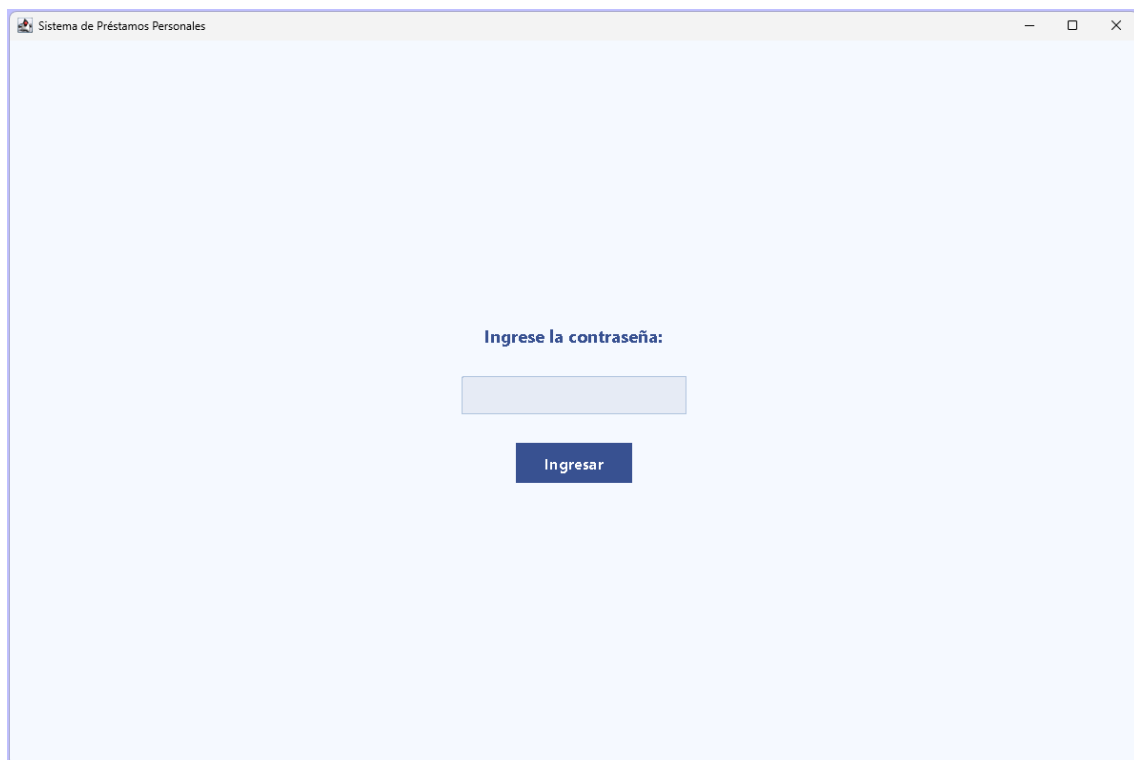
- **Tipo de comunicación:** Local (interprocesos en la misma PC)
- **Protocolo:** JDBC (Java Database Connectivity) mediante la URL de conexión a localhost.
- **Puerto utilizado:** 3306 (puerto predeterminado de MySQL, acceso local)
- **Seguridad:** El acceso está restringido por usuario y contraseña de MySQL y no se expone la base de datos a la red.
- **No existen conexiones externas** a otras computadoras, servidores remotos o servicios web, ni se utiliza red local ni Internet.

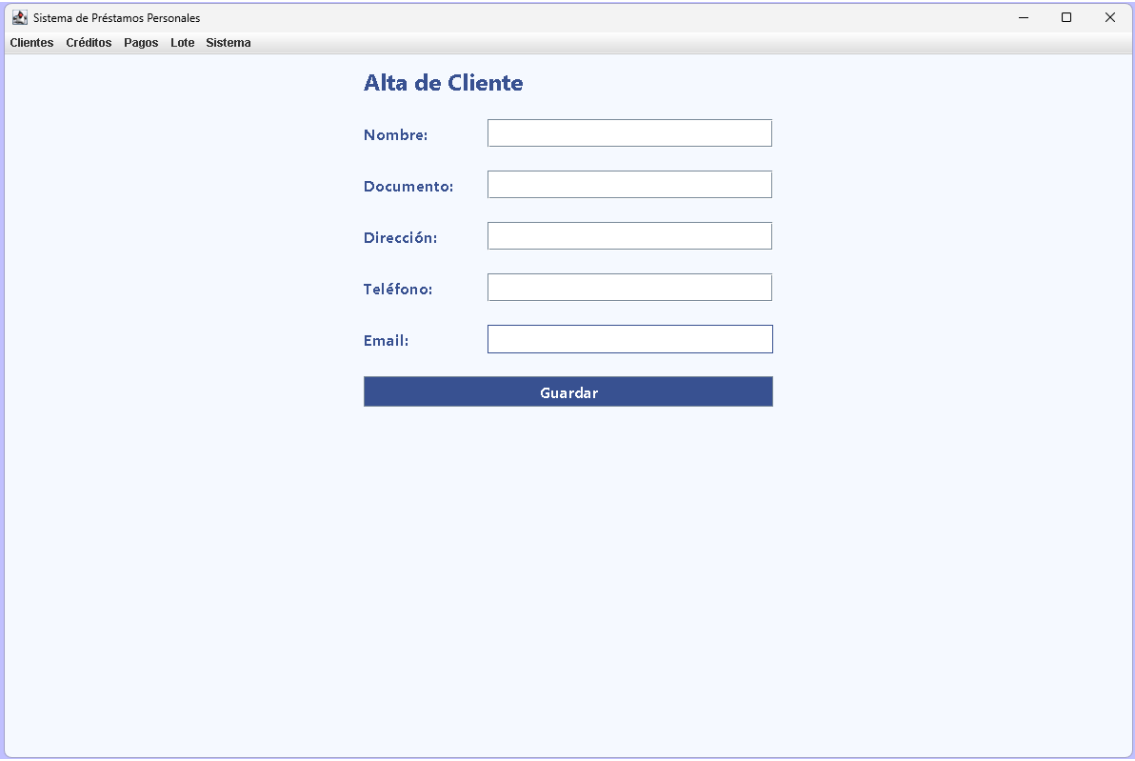
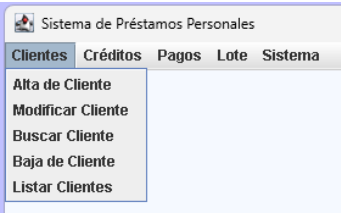
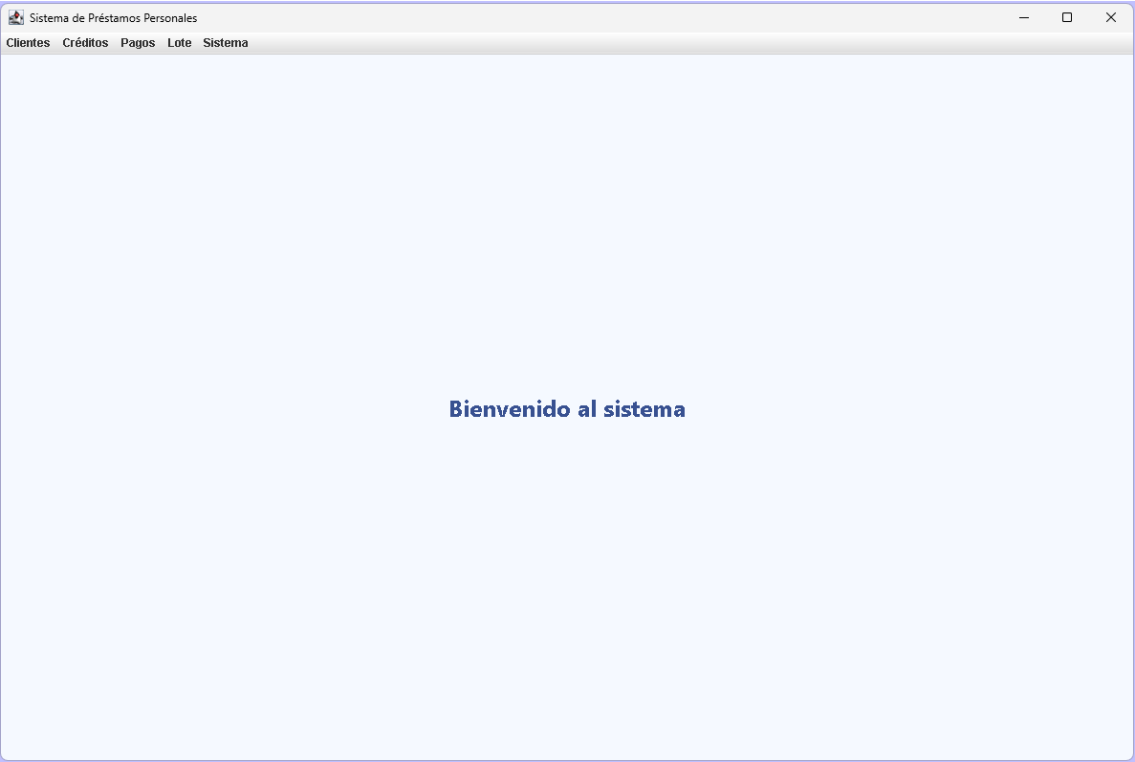
15. PROTOTIPO FUNCIONAL

La presente sección documenta el prototipo funcional de la interfaz de usuario desarrollado en Java Swing. El diseño se orientó a minimizar la curva de aprendizaje del personal de la tienda y a garantizar visibilidad inmediata del estado de clientes, créditos, cuotas y pagos. Cada pantalla se corresponde con requerimientos funcionales previamente elicitados y contribuye a reducir errores de registro, agilizar la operatoria y asegurar trazabilidad de las transacciones clave.

La navegación se organiza mediante un menú superior que agrupa las operaciones por dominio: Clientes, Créditos, Pagos, Lote y Sistema (configuración / sesión). El flujo típico inicia con la autenticación, continúa con la selección del módulo y culmina en una operación transaccional (alta, pago, cierre de lote) o en una consulta. Esta estructura disminuye la sobrecarga cognitiva y facilita la memorización de rutas de acceso.

A continuación, se presentan las capturas de cada módulo del prototipo funcional en su estado actual, el cual puede ser obtenido y probado desde el repositorio de GitHub <https://github.com/Titoyco/seminario>. En el mismo se encuentra además del código fuente, el archivo SQL para crear al base de datos con datos de prueba.





Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Modificar Cliente

Seleccionar Cliente:

Juan Pérez (ID: 1)

Nombre:

Juan Pérez

Documento:

12345678

Dirección:

Calle Falsa 123

Teléfono:

1122334455

Email:

juan.perez@mail.com

Guardar Cambios

Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Buscar Clientes

Nombre:

Ana

Documento:

Buscar

ID	Nombre	Documento	Dirección	Teléfono	Email
6	Ana Torres	66778899	Belgrano 200	1177889900	ana.torres@mail.com
26	Daiana Cabrera	42434445	Jujuy 17	1177889021	daiana.cabrera@mail.com

Modificar

Eliminar

Ver Préstamos

Ver Pagos

Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Baja de Cliente

Seleccionar Cliente:Juan Pérez (ID: 1)

ID de Cliente:1

Eliminar

Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Listado de Clientes

ID	Nombre	Documento	Dirección	Teléfono	Email
1	Juan Pérez	12345678	Calle Falsa 123	1122334455	juan.perez@mail.com
2	Maria López	22334455	Av. Siempre Viva 742	1133445566	maria.lopez@mail.com
3	Carlos García	33445566	Diagonal 80 1002	1144556677	carlos.garcia@mail.com
4	Laura Fernández	44556677	Ruta 8 km 58	1155667788	laura.fernandez@mail.com
5	Luis Gómez	55667788	San Martín 350	1166778899	luis.gomez@mail.com
6	Ana Torres	66778899	Belgrano 200	1177889900	ana.torres@mail.com
7	Diego Ramírez	77889900	Mitre 88	1188990011	diego.ramirez@mail.com
8	Juancho Perez	23432564	Islas Malvinas 123	11432292	jperezbv@hotmail.com
9	Mauro Herrera	99001122	Moreno 450	1100112233	mauro.herrera@mail.com
10	Julia Castro	10111213	Maipú 25	1111223344	julia.castro@mail.com
11	Pedro Suárez	12131415	Ituzaingo 78	1122334455	pedro.suarez@mail.com
12	Cecilia Ríos	14151617	Colón 300	1133445566	cecilia.rios@mail.com
13	Emilia Medina	16171819	9 de Julio 100	1144556678	emilia.medina@mail.com
14	Nicolás Vega	18192021	Santa Fe 222	1155667789	nicolas.vega@mail.com
15	Lautaro Sosa	20212223	Entre Ríos 500	1166778900	lautaro.sosa@mail.com
16	Camila Molina	22232425	Rivadavia 1200	1177889001	camila.molina@mail.com
17	Martín Navarro	24252627	Urquiza 99	1188990122	martin.navarro@mail.com
18	Valeria Álvarez	26272829	Alvear 15	1199001233	valeria.alvarez@mail.com
19	Kevin Paredes	28293031	Rawson 80	1100112344	kevin.paredes@mail.com
20	Carla Benítez	30313233	Castelli 200	1111223455	carla.benitez@mail.com
21	Franco Ruiz	32333435	Mendoza 1100	1122334566	franco.ruiz@mail.com
22	Agustina Giménez	34353637	Chacabuco 77	1133445677	agustina.gimenez@mail.com
23	Bruno Correa	36373839	Corrientes 100	1144556788	bruno.correa@mail.com
24	Lucía Ortiz	38394041	Sarmiento 412	1155667899	lucia.ortiz@mail.com
25	Tomás Aguirre	40414243	Catamarca 200	1166778910	tomas.aguirre@mail.com
26	Daiana Cabrera	42434445	Jujuy 17	1177889021	daiana.cabrera@mail.com
27	Gastón Molina	44454647	Formosa 80	1188990132	gaston.molina@mail.com
28	Natalia Cabrera	46474849	San Juan 110	1199001343	natalia.cabrera@mail.com
29	Matías Rivas	48495051	La Rioja 300	1100112454	matias.rivas@mail.com
30	Rocío Funes	50515253	Salta 222	1111223565	rocio.funes@mail.com
33	nuevo	132123	ppasp	123	asdqw@ads.com
34	vamos otro	99999	8888	7777	66666

Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Alta de Crédito

Créditos por Cliente

Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Alta de Crédito

Ciente:

Juan Pérez (ID: 1)

Importe (capital):

1000

Cantidad de cuotas:

4

Interés mensual vigente:

5.00 %

Crear Crédito

Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Ciente: Juan Pérez (ID: 1)

Cargar Créditos

ID	Monto	Tasa%	Cuotas	Fecha	Estado	Lote Origen
22	100000.0	5.0	4	2025-09-20	vigente	4
21	1000.0	5.0	5	2025-09-20	vigente	4
1	10000.0	10.0	3	2025-04-02	vigente	0

#	Monto	Estado	Lote Vencimiento
1	30000.0	pagada	5
2	30000.0	pendiente	6
3	30000.0	pendiente	7
4	30000.0	pendiente	8

Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Pagar Cuota

Listar Pagos por Cliente

Anular Pago

Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Pagar Cuota

Cliente:

Juan Pérez (ID: 1)

Cargar cuotas pendientes

Cuota:

-

Método de pago:

efectivo

Observaciones:

Pagar

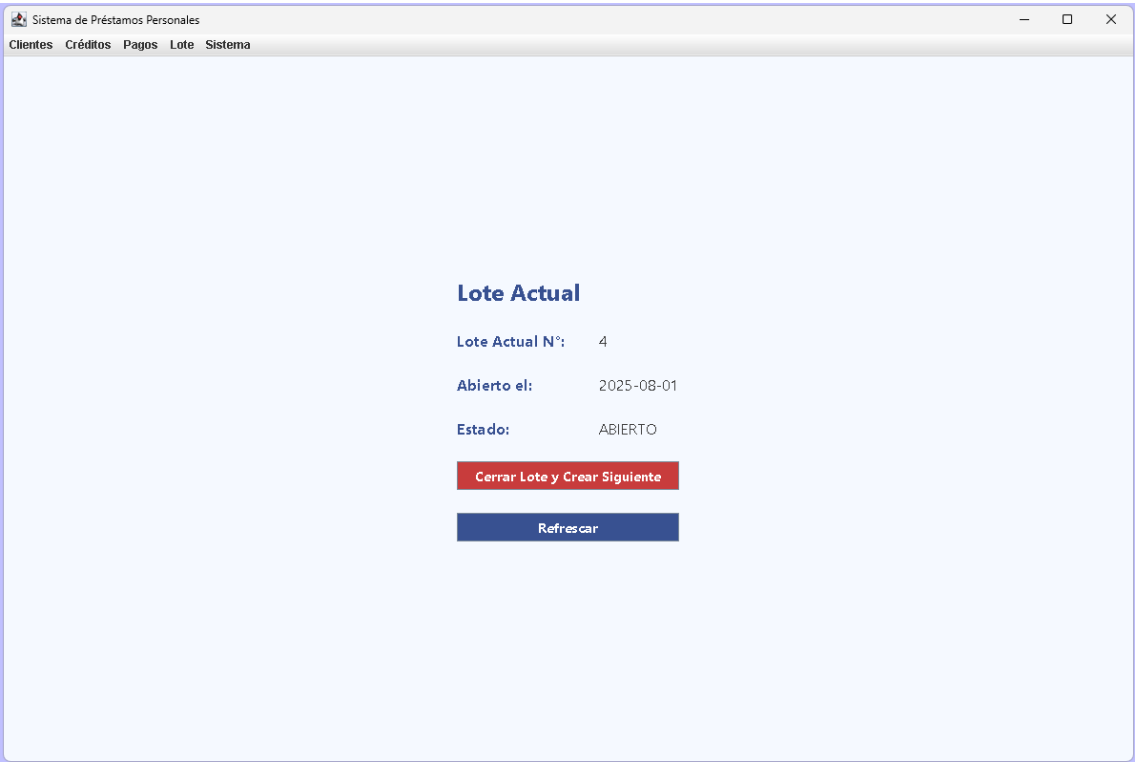
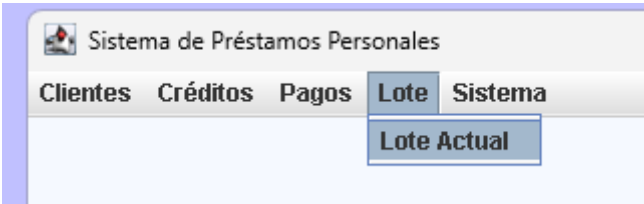
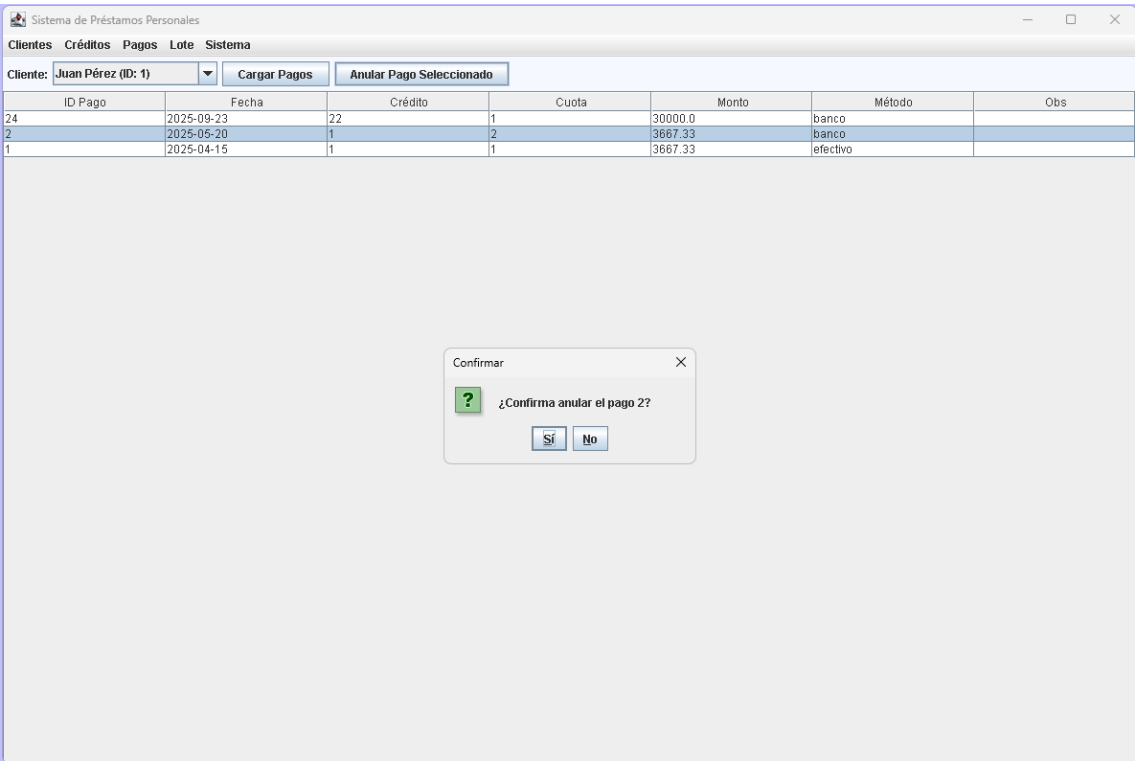
Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Cliente: Juan Pérez (ID: 1)

Cargar Pagos

ID Pago	Fecha	Crédito	Cuota	Monto	Método	Obs
24	2025-09-23	22	1	30000.0	banco	
2	2025-05-20	1	2	3667.33	banco	
1	2025-04-15	1	1	3667.33	efectivo	



Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Cambio de Contraseña

Nueva contraseña:

Confirmar contraseña:

Guardar

Sistema de Préstamos Personales

CientesCréditosPagosLoteSistema

Cambio de contraseña

Cerrar sesión

Bienvenido al sistema

16. CONCLUSIONES Y TRABAJO FUTURO

16.1. Conclusiones

El desarrollo del sistema de gestión de créditos permitió automatizar y optimizar los procesos administrativos de otorgamiento, seguimiento y cobranza en un comercio minorista.

A partir de la implementación del patrón Modelo-Vista-Controlador (MVC) y el uso de Java Swing con MySQL, se logró:

- Centralizar la información de clientes, créditos, cuotas y pagos en una única base de datos relacional.
- Mejorar la trazabilidad y control de las operaciones, disminuyendo errores manuales y redundancia en los datos.
- Facilitar la consulta y el registro de operaciones frecuentes mediante interfaces gráficas amigables y flujos claros.
- Garantizar la integridad referencial y la seguridad básica de los datos, mediante claves foráneas y autenticación por contraseña.
- Proveer una solución flexible, que puede ampliarse a entornos multiusuario en red local con mínimos ajustes de configuración.

El sistema cumple con los requerimientos funcionales y no funcionales planteados inicialmente, y fue validado mediante una serie de pruebas que cubren los principales casos de uso y valores frontera.

16.2. Trabajo futuro

Si bien el sistema cubre las necesidades actuales del comercio, existen oportunidades de mejora y evolución que pueden contemplarse a futuro, tales como:

- Incorporar reportes y estadísticas avanzadas (mora, ranking de clientes, análisis de cobros).
- Agregar funcionalidades de exportación de datos a PDF o Excel.
- Mejorar la seguridad de las contraseñas (hash, políticas de complejidad) y gestión de usuarios/roles.
- Implementar auditoría de operaciones y bitácora de cambios.
- Desarrollar mecanismos de backup y recuperación automáticos.
- Adaptar la aplicación para funcionamiento multiusuario en red local o, eventualmente, en la nube.
- Incluir notificaciones automáticas para recordatorio de vencimientos y cuotas en mora.
- Mejorar la interfaz gráfica a nivel estético y de experiencia de usuario.
- Optimizar el soporte para grandes volúmenes de datos y crecimiento futuro del negocio.

El sistema desarrollado constituye una base sólida sobre la cual pueden agregarse nuevas funcionalidades y mejoras, adaptándose a las necesidades cambiantes del comercio y del entorno tecnológico.

BIBLIOGRAFÍA:

- Kendall, K. y Kendall, J. (2011). Análisis y Diseño de Sistemas. Pearson Education.
- Oliveros, A. y Antonelli, L. (2015). Técnicas de elicitación de requerimientos. XXI Congreso Argentino de Ciencias de la Computación, Junín, Argentina.
- Saroka, R. (2002). Sistemas de Información en la era digital. Fundación OSDE.
- Sommerville, I. (2005). Ingeniería del software (Séptima ed.). Pearson-Addison Wesley.
- Stair, R. y Reynolds, G. (2010). Principios de Sistemas de Información (9a. ed.). Cenage Learning.
- Van Gigch, J. (1995). Teoría General de Sistemas. Trillas.