

Actividad | #3 | Modelado UML.

Lenguaje Unificado de Modelado.

Ingeniería en Desarrollo de Software



TUTOR: Eduardo Israel Castillo Gracia.

ALUMNO: Manuel Marin Sanchez.

FECHA: 10/09/2025

Indicé.

Portada	1
Indicé	2
Introducción	3
Descripción	4
Justificación	5
Desarrollo.	
Actividad 1	
Definición de requisitos de casos de uso.....	6
Diagrama de Casos de Uso.....	7
Ficha de Especificaciones.....	8
Actividad 2	
Diagrama de Clases.....	9
Actividad 3	
Diagrama de Actividades.....	10
Diagrama de Componentes.....	11
Conclusión	12
Referencias	13

Introducción.

El modelado UML es una herramienta fundamental en la ingeniería de software para visualizar, especificar y documentar sistemas complejos. Este proyecto aplica el Lenguaje Unificado de Modelado (UML) al diseño del Sistema de Créditos y Compras de Coppel, integrando procesos clave como registro de clientes, consulta de buró crediticio, aprobación de créditos, compras y pagos. Mediante diagramas de casos de uso, clases, actividades y componentes, se representa tanto la estructura estática como el comportamiento dinámico del sistema. Este enfoque no solo facilita la comunicación entre desarrolladores, stakeholders y usuarios, sino que también sirve como blueprint para la implementación técnica. La documentación generada asegura que todos los requisitos funcionales y no funcionales sean capturados de manera precisa, reduciendo ambigüedades y garantizando que el sistema sea escalable, seguro y alineado con las necesidades del negocio. El proyecto demuestra cómo UML trasciende la teoría para convertirse en un pilar práctico en el desarrollo de software robusto.

Descripción.

El sistema modelado unifica los procesos de crédito y compras de Coppel, operando en tiendas físicas y en línea. Incluye módulos para registro de clientes, evaluación crediticia (consulta de buró), aprobación de créditos, compra a crédito, pagos y generación de estados de cuenta. Los diagramas de casos de uso definen las interacciones entre actores (clientes, empleados, asesores) y el sistema. El diagrama de clases detalla la estructura estática con clases como Cliente, Crédito, Compra y Pago, sus atributos, métodos y relaciones (asociaciones, herencia, composición). Los diagramas de actividades modelan flujos de procesos, como el proceso de compra o consulta de buró, mientras que el diagrama de componentes describe la arquitectura física, mostrando cómo se interconectan software y hardware. Este modelado integral asegura que el sistema sea comprensible para técnicos y no técnicos, y proporciona una base sólida para la fase de implementación.

Justificación.

La elección de UML para modelar el sistema de Coppel se justifica por su capacidad para abstraer la complejidad mediante visualizaciones estandarizadas. Los diagramas de casos de uso capturan requisitos funcionales desde la perspectiva del usuario, evitando malentendidos. Los diagramas de clases definen una estructura coherente y extensible, critical para un sistema que maneja datos sensibles (créditos, pagos). Los diagramas de actividades optimizan flujos operativos, identificando cuellos de botella o redundancias antes del desarrollo. El diagrama de componentes asegura que la infraestructura (servidores, bases de datos, interfaces) sea diseñada para escalar y integrarse con sistemas legacy. Además, UML es independiente de tecnologías específicas, permitiendo adaptaciones futuras. Este enfoque proactivo detecta errores de diseño en etapas tempranas, ahorrando costos y tiempo. Para un negocio como Coppel, donde la confiabilidad y seguridad son prioritarias, UML ofrece el rigor necesario para garantizar que el sistema cumpla con expectativas de desempeño, usabilidad y mantenibilidad.

Desarrollo: Actividad #1

Definición de requisitos funcionales y no funcionales.

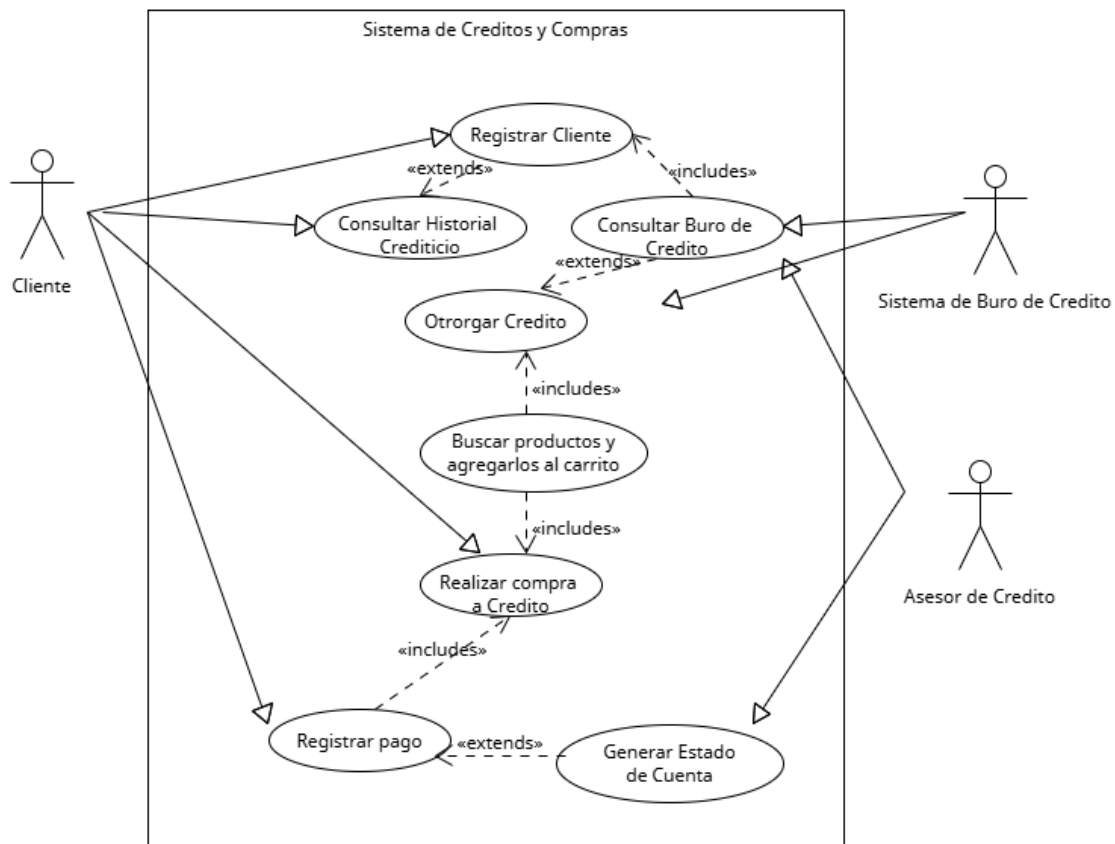
Requisitos Funcionales:

1. RF1: Registrar cliente nuevo (datos personales, historial inicial).
2. RF2: Consultar buró de crédito del cliente.
3. RF3: Otorgar crédito (aprobación/rechazo basado en buró).
4. RF4: Realizar compra a crédito (física o e-commerce).
5. RF5: Registrar pagos (presencial o en línea).
6. RF6: Consultar historial crediticio (cliente y empleados).
7. RF7: Generar estados de cuenta (mensual o a solicitud).

Requisitos No Funcionales:

1. RNF1: Seguridad (encriptación de datos, cumplimiento PCI-DSS).
2. RNF2: Disponibilidad 24/7 para e-commerce.
3. RNF3: Usabilidad (interfaz intuitiva para clientes y empleados).
4. RNF4: Escalabilidad (soporte para crecimiento de transacciones).
5. RNF5: Integración con sistemas legacy de Coppel.

Diagrama de Casos de Uso.



Ficha de Especificaciones.

Nombre.	Registrar Cliente.
Autor.	Manuel Marin Sanchez.
Fecha.	30/08/2025.
Descripción.	Este escenario describe el proceso de registro de un nuevo cliente en el sistema de créditos y compras de coppel.
Actores.	Cliente, Empleado de tienda (opcional).
Precondiciones.	El cliente no debe estar registrado previamente en el sistema.
Flujo Normal.	<ol style="list-style-type: none"> 1. El cliente solicita el registro en tienda física o en línea. 2. El sistema muestra el formulario de registro solicitando: nombre, CURP, dirección, ingresos, etc. 3. Cliente completa formulario con los datos requeridos. 4. El sistema valida que los datos no estén duplicados y sean correctos. 5. Si la validación es exitosa, el sistema crea la cuenta del cliente y asigna un numero de credito inicial. 6. Se muestra un mensaje de confirmación y se inicia el proceso de consulta de buro de credito.
Flujo Alternativo.	<ol style="list-style-type: none"> 1. Datos inválidos: Si los datos están incorrectos o incompletos, el sistema solicita correcciones. 2. Cliente ya registrado: Si el cliente ya existe, sistema solicita validar datos o recuperar cuenta.
Postcondiciones.	Cliente queda registrado en el sistema y se habilita la opción para solicitar crédito.

Ficha de Especificaciones

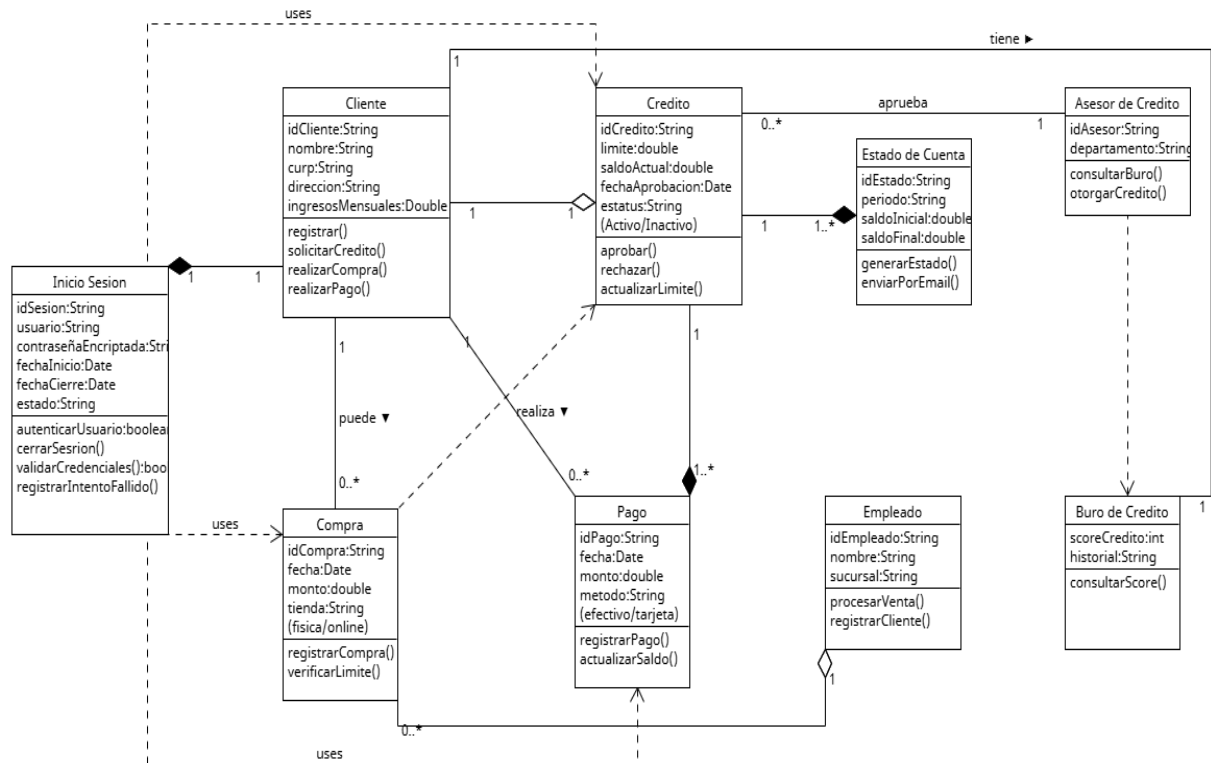
Nombre.	Realizar Compra a Crédito.
Fecha.	30/08/2025.
Autor.	Manuel Marin Sanchez.
Descripción.	Este escenario describe el proceso donde un cliente realiza una compra utilizando su crédito coppel, ya sea tienda física o en línea.
Actores.	Cliente, Empleado de Tienda (para ventas presenciales).
Precondiciones.	El cliente debe estar registrado y tener un crédito aprobado con límite disponible.
Flujo Normal.	<ol style="list-style-type: none"> 1. Cliente selecciona los productos a comprar. 2. Sistema calcula el total de la compra. 3. El cliente elige pagar con credito coppel. 4. El sistema verifica el limite de credito disponible. 5. Si el límite es suficiente, se confirma la compra. 6. El sistema registra la transacción, actualiza el saldo del crédito y genera un comprobante.
Flujo Alternativo.	<ol style="list-style-type: none"> 1. Limite de credito insuficiente: El sistema rechaza la compra y solicita otro medio de pago. 2. Error en transacción: Si falla el registro, se cancela la compra y se notifica al cliente.
Postcondiciones.	La compra queda registrada, el saldo se actualiza y genera un comprobante.

Ficha de Especificaciones.

Nombre.	Consultar Buro de Credito.
Autor.	Manuel Marin Sanchez.
Fecha.	30/08/2025.
Descripción.	Este escenario describe la consulta de buro de credito para evaluar la solvencia de un cliente antes de otorgar o renovar un crédito.
Actores.	Asesor de crédito, Sistema de buro de credito.
Precondiciones.	Cliente debe estar registrado en el sistema y haber autorizado la consulta.
Flujo Normal.	<ol style="list-style-type: none">1. El asesor de crédito solicita la consulta desde el sistema.2. El sistema envía los datos del cliente (CURP, nombre completo) al Buro externo.3. El Buro externo devuelve el reporte crediticio.4. El asesor visualiza el reporte y decide si aprueba o rechaza el crédito.
Flujo Alternativo.	<ol style="list-style-type: none">1. Error de conexión con Buro: El sistema reintentar la conexión o notifica el fallo.2. Datos incorrectos: Si los datos del cliente son erróneos , se solicita la corrección.
Postcondiciones.	El reporte de Buro se almacena en el historial del cliente y esta disponible para futuras consultas.

Actividad #2

Diagrama de Clases.



Relaciones y Multiplicidad.

- Cliente (1) (1..*) Crédito
(Un cliente puede tener uno o más créditos)
- Cliente (1) (1..*) Compra
(Un cliente realiza una o más compras)
- Cliente (1) (1..*) Pago
(Un cliente realiza uno o más pagos)
- Empleado (1) (0..*) Compra
(Un empleado puede procesar cero o muchas compras)
- AsesorCrédito (1) (1..*) Crédito
(Un asesor aprueba uno o más créditos)
- Crédito (1) (1..*) EstadoCuenta
(Un crédito genera uno o más estados de cuenta)
- BuróCrédito (1) (1) Cliente
(Cada cliente tiene un reporte en el buró)

Actividad #3
Diagrama de Actividades.

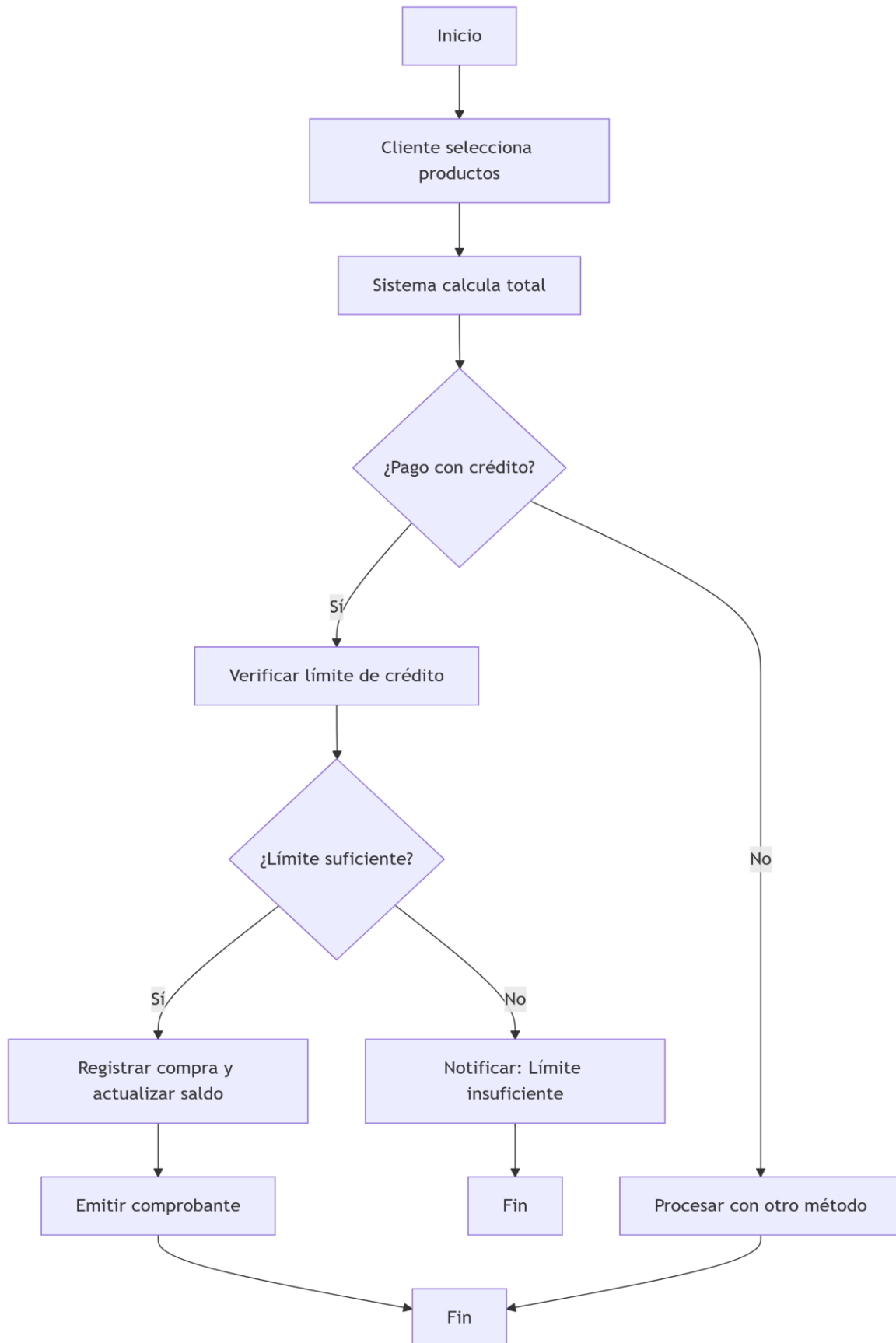
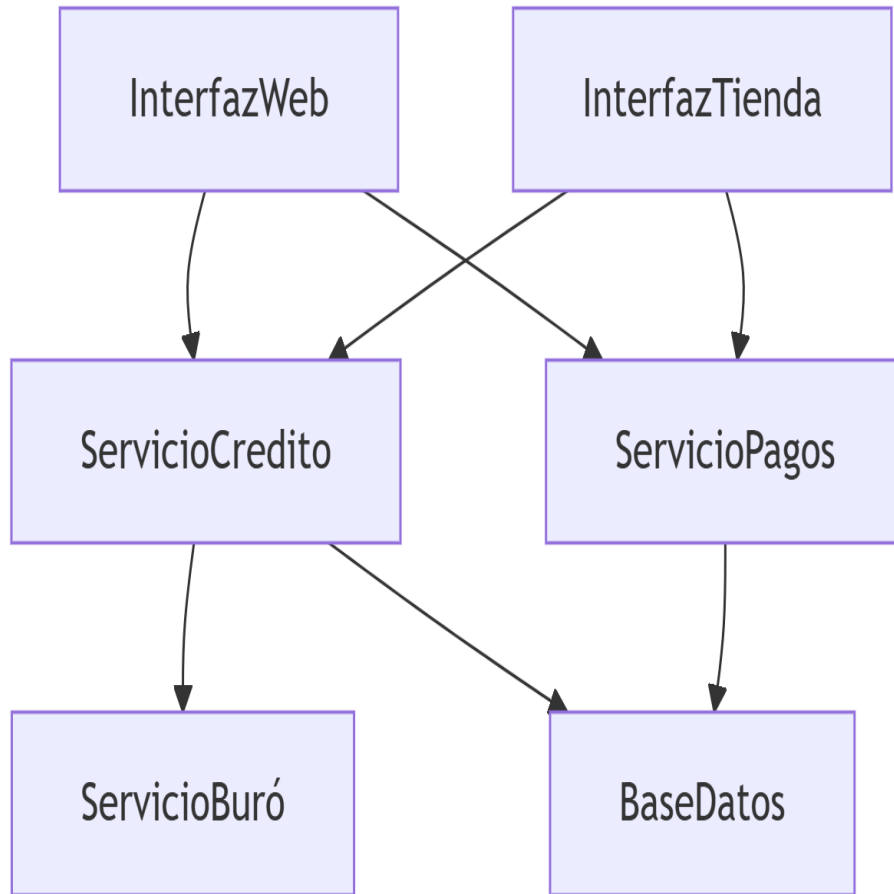


Diagrama de Actividades.



1. Identificar los componentes principales

Basado en el sistema de Coppel, los componentes clave son:

- InterfazWeb (para clientes en línea)
- InterfazTienda (para empleados en tiendas físicas)
- ServicioCredito (lógica de créditos)
- ServicioPagos (procesamiento de pagos)
- ServicioBuró (consulta externa al buró de crédito)
- BaseDatos (almacenamiento persistente)

2. Definir dependencias entre componentes

- InterfazWeb depende de ServicioCredito y ServicioPagos.
- InterfazTienda depende de ServicioCredito y ServicioPagos.
- ServicioCredito depende de ServicioBuró y BaseDatos.
- ServicioPagos depende de BaseDatos.

Conclusión.

Este proyecto demostró la utilidad del UML para diseñar sistemas robustos y escalables. Los diagramas creados (casos de uso, clases, actividades y componentes) no solo representan técnicamente el sistema de créditos de Coppel, sino que también encapsulan best practices de diseño orientado a objetos. En el ámbito laboral, estos modelos son esenciales para coordinación entre equipos multidisciplinarios y para la onboarding de nuevos desarrolladores. En la vida cotidiana, refuerzan habilidades de abstracción y planificación estratégica. La integración de vistas estáticas (clases) y dinámicas (actividades) asegura que el sistema sea tanto estructuralmente sólido como comportamentalmente eficiente. Futuras iteraciones podrían incluir diagramas de despliegue para infraestructura cloud o diagramas de estado para transiciones crediticias. En definitiva, el UML trasciende la notación gráfica: es un lenguaje universal que bridge entre ideas y implementación.

Referencias.

[Lenguaje Unificado de modelado #1 - Zoom](#)

[Lenguaje Unificado de modelado #2 - Zoom](#)

[Lenguaje Unificado de modelado #3 - Zoom](#)

https://github.com/ManuelMarin14/Manuel_Marin_Act2.git