

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES



INFORME DE PROYECTO FINAL

TÍTULO: Desarrollo de un Simulador de Cajero Automático en Chorrillos -
Lima

CURSO: TÉCNICAS DE PROGRAMACIÓN ORIENTADA A OBJETOS

DOCENTE DEL CURSO: Ing. José Anicama Silva

AUTORES:

- Ramos Alva, Rodrigo Raúl Alonso (100% participación)
- Santillana Olivares, Sebastián Rodolfo (30% participación)
- Portocarrero Huamán, Kiara Lidia (100% participación)

CHORRILLOS –

PERÚ 2025-1

Contenido

Información del Proyecto	4
Aprobaciones	4
1. Descripción del problema	5
2. Antecedentes / referencias	5
3. Restricciones del proyecto	5
3.1. Identificación y análisis de todas las restricciones, criterios y suposiciones	5
3.2. Generación y análisis de múltiples soluciones	5
3.3. Criterios para seleccionar la solución <i>(dentro de las restricciones económicas, ambientales, sociales, políticas, éticas, de seguridad, entre otras.)</i>	5
4. Alcance del producto / Software	5
5. Objetivo general	6
6. Objetivos específicos	7
7. Diagrama de clases y relaciones	11
8. Requerimientos funcionales	12
9. Reglas de negocio	14
10. Requerimientos no funcionales	15
11. Glosario	15
12. Bibliografía	1

INDICE DE ANEXOS

1. Diagrama Ishikawa

https://lucid.app/lucidspark/3c2bacb9-c61d-4316-9f94-df29dbe6a49b/edit?viewport_loc=552%2C91%2C2354%2C1659%2C0_0&invitationId=inv_4641e6d1-5f40-4e71-b884-026165ae49f8

1. Descripción del problema

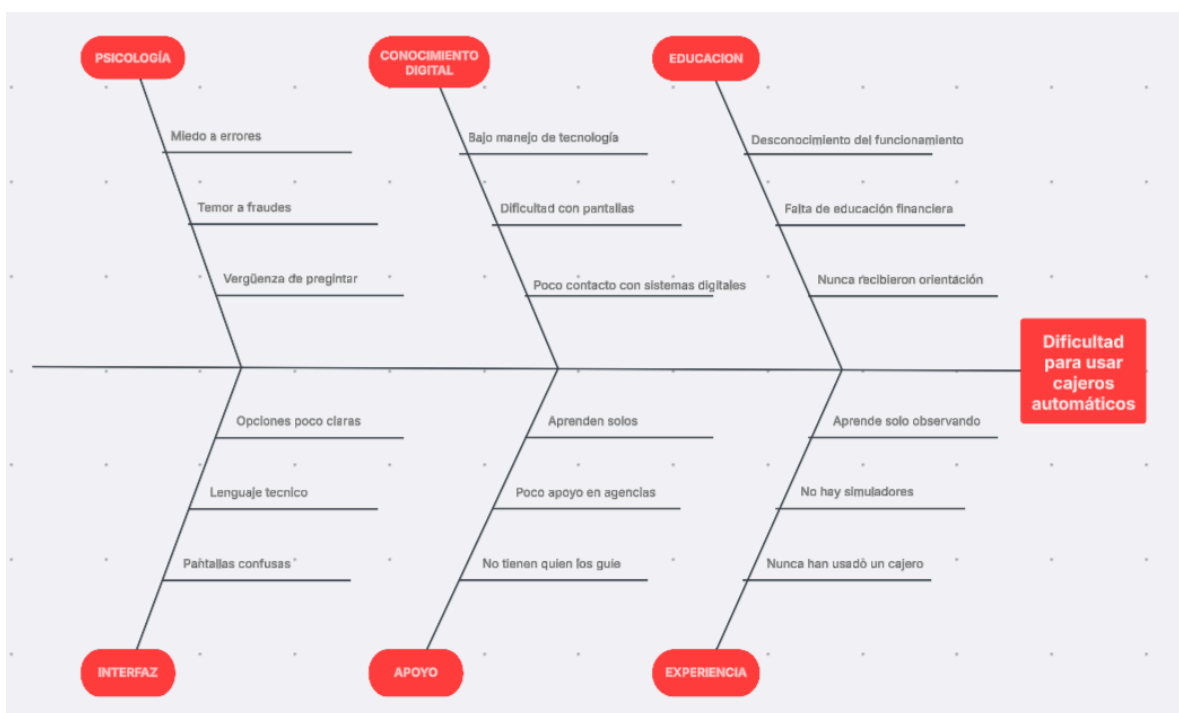
Problema: Falta de experiencia previa financiera en jóvenes y adultos mayores

En el distrito de Chorrillos, ubicado en Lima, muchas personas principalmente adultos mayores y jóvenes con poca experiencia digital enfrentan dificultades para utilizar cajeros automáticos.

Esta limitación afecta su acceso a servicios financieros básicos como retiros de dinero, consultas de saldos o transferencias a seres queridos.

A su vez, este problema se agrava por la falta de educación tecnológica, el temor a cometer errores y el miedo a ser víctimas de fraudes, esto trae como consecuencia que estas personas eviten el uso de cajeros y prefieren la atención personalizada, lo que genera largas colas y demoras en agencias bancarias, especialmente en días de alta demanda como pagos de pensiones o bonos.

La ausencia de iniciativas tecnológicas inclusivas que permitan una educación práctica y segura empeora este problema, el cual afecta de forma recurrente a comunidades vulnerables como las de Chorrillo, donde hay una alta concentración de adultos mayores.



Por lo tanto, el problema que abordaremos en este trabajo es la falta de acceso y habilidades prácticas en el uso de cajeros automáticos por parte de adultos mayores y jóvenes con escasa experiencia financiera y tecnológica, proponiendo como solución el desarrollo de un simulador de cajero automático que les permita practicar en un entorno seguro, didáctico y accesible.

2. Antecedentes / referencias

A lo largo del tiempo, muchas personas han creado programas parecidos al simulador de cajero automático que queremos hacer. Estos programas se han usado en colegios, universidades y centros de capacitación para enseñar a las personas a cómo usar un cajero de forma segura y sin miedo.

Por ejemplo, Mead y Fisk (1998) hicieron un estudio donde mostraron que los adultos mayores aprenden mejor a usar los cajeros si practican con un simulador, de manera que pierden el miedo y entienden mejor los pasos. Más adelante, Kopeć et al. (2019) trabajaron con adultos mayores para crear un simulador en realidad virtual, donde ellos mismos ayudaron a decir cómo debía ser el programa para que les fuera fácil de usar. Esta idea también se aplicó para niños, Amat (2019) construyó un cajero con piezas electrónicas basado en Arduino y una pantalla para que los niños aprendan desde pequeños a manejar su dinero y a conocer cómo funcionan los cajeros. De igual manera, Forcheri y Molfino (1989) crearon un programa para enseñar matemáticas usando un cajero simulado, lo que hacía que aprender fuera más divertido.

Si hablamos de América Latina también hay investigaciones, como la de Carangui Cárdenas et al. (2017) ellos mostraron que los simuladores ayudan a los estudiantes a entender mejor temas de dinero y a aprender haciendo, no solo leyendo. Además, en internet hay programas parecidos que se pueden ver como ejemplo, como el de Dalbey (s. f.) que tiene un caso completo de un simulador de cajero donde se explica paso a paso cómo se construye.

Todos estos antecedentes ayudan a entender que el proyecto que queremos hacer es útil, posible y ya ha sido probado con buenos resultados, como también nos servirán como guía para planear mejor cómo será nuestro simulador.

3. Restricciones del proyecto

3.1. Identificación y análisis de todas las restricciones, criterios y suposiciones

Para realizar nuestro simulador de cajero automático hemos pensado en algunas cosas que podrían ser un problema (restricciones), cosas que queremos lograr (criterios) y cosas que creemos que pasarán (suposiciones):

Restricciones

- El programa debe funcionar sin internet
- Debe ser muy fácil de usar, incluso por personas que no saben mucho de computadoras
- No podemos gastar dinero porque es un proyecto escolar
- Tiene que funcionar en computadoras sencillas, como las que hay en escuelas o casas

Criterios

- Tiene que ser lo más parecido a un cajero real, con botones e instrucciones fáciles
- Debe permitir practicar operaciones como sacar dinero, depositarlo y ver saldo
- Debe ser un sistema amigable, interactivo y ordenado
- No debe contener errores para que así funcione de manera óptima

Suposiciones

- Los usuarios sabrán leer las instrucciones que aparecen en la pantalla
- Podrán usar el simulador con ayuda de un profesor, familiar o tutor
- Servirá para que las personas pierdan el miedo a usar un cajero real

3.2. **Generación y análisis de múltiples soluciones**

Antes de hacer el simulador hemos pensado en diferentes formas para ayudar a las personas a aprender a usar un cajero, estas son las ideas que tuvimos:

1. Crear un video explicativo en la plataforma de Youtube

Hacer un video donde se muestre cómo usar un cajero paso a paso

(Es fácil de hacer, pero la persona solo mira y no puede practicar)

2. Hacer un juego interactivo

Diseñar un juego donde el usuario gana puntos por hacer bien las operaciones de cajero

(Es divertido pero puede confundir a adultos mayores si parece un juego y no algo serio)

3. Elaborar un simulador como si fuera un cajero físico

Crear un programa donde la persona vea una pantalla parecida a la del cajero, con botones y opciones reales (Permite practicar sin miedo a equivocarse y se parece mucho a un cajero real)

Después de analizar estas tres ideas, creemos que la mejor es la número 3: "Elaborar un simulador como si fuera un cajero físico", ya que le permitirá al usuario aprender haciendo, sin riesgos y le ayudará a perder el miedo para una futura visita a alguna agencia.

3.3. **Criterios para seleccionar la solución**

Para elegir la mejor opción pensamos en los siguientes criterios:

Económicos:

- El simulador no debe costar dinero, lo haremos mediante herramientas gratuitas (NetBeans, LucidChart, GitHub)
- No se debería comprar nada, solo usar una computadora o laptop

Ambientales:

- No usaremos papel ni materiales físicos de manera que no contaminamos nada
- Solo se usa energía eléctrica de la computadora o laptop

Sociales:

- Ayuda a personas que normalmente no usan cajeros por miedo o falta de práctica
- Mejora su confianza para hacer trámites sin ayuda

Políticos:

- Ayuda a programas del gobierno que buscan incluir a adultos mayores en la tecnología
- Mediante campañas publicitarias se podría impulsar su uso

Éticos:

- No se pedirá datos personales
- El simulador es solo educativo, no se conectará a cuentas reales

Seguridad:

- Las cuentas son ficticias, por tanto, no hay peligro de perder dinero
- Es un espacio seguro para aprender

Por estas razones, decidimos hacer un simulador simple, sencillo de entender y seguro, que ayude a las personas a aprender cómo usar un cajero real

4. Alcance del producto / Software

Nuestro proyecto busca crear un programa educativo que parezca un cajero automático, este será usado por personas que no tienen mucha experiencia con la tecnología, especialmente adultos mayores y jóvenes. El programa ayudará a que practiquen de manera tranquila y sin ningún peligro, aprendiendo cómo se hacen operaciones como retirar dinero, ver el saldo o hacer depósitos

- Propósito u objetivo general

El propósito del software es enseñar de forma práctica el uso de cajeros automáticos permitiendo que los usuarios pierdan el miedo y se sientan seguros al momento de hacer operaciones reales, también busca que las personas se familiaricen con las pantallas y botones, tal como si estuvieran frente a un cajero de verdad

- Beneficios para la comunidad y organizaciones

Este simulador puede ser muy útil en colegios, centros de adultos mayores, talleres comunitarios o programas del gobierno, estos serían los principales beneficios:

- Ayuda a que las personas sepan manejar su propio dinero sin depender de otros
- Disminuye las colas en los bancos, porque menos gente pedirá ayuda si ya sabe usar el cajero
- Fomenta la inclusión digital, dándole una oportunidad a personas que normalmente no usan tecnología
- Al ser parte de campañas educativas de bancos o municipios, acerca los servicios financieros a quienes más lo necesitan

- Objetivos y metas del software

- Simular un cajero con pantallas fáciles de entender
- Incluir opciones reales como retiro, consulta y depósito
- Ser accesible desde cualquier computadora, sin necesidad de internet
- Servir como material de apoyo en programas de educación digital o financiera

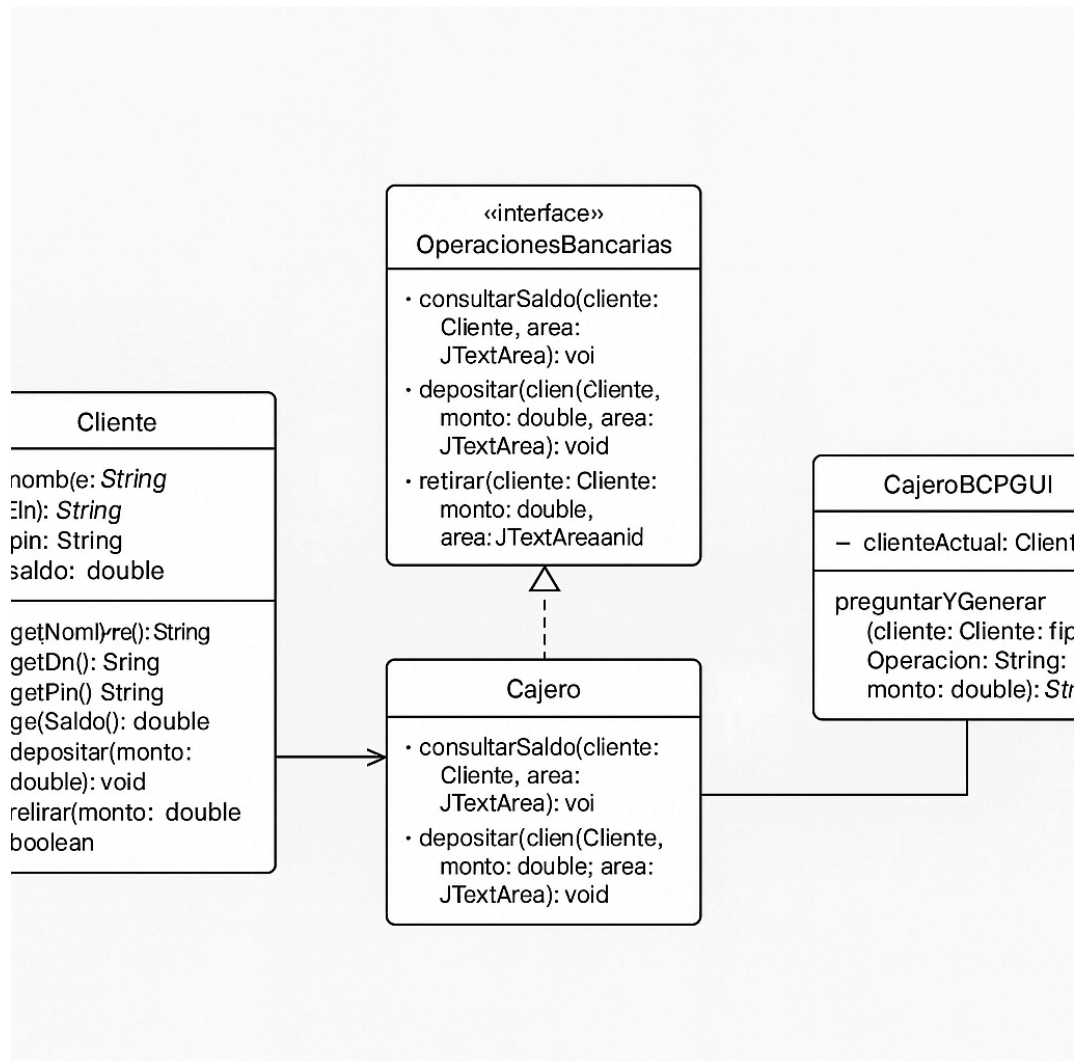
5. Objetivo general

- Desarrollar un simulador de cajero automático para personas con poca experiencia financiera o digital, como adultos mayores y jóvenes.
- Desarrollar un simulador de cajero automático que permita practicar operaciones bancarias básicas, con el fin de mejorar la confianza y el aprendizaje en el uso de servicios financieros digitales.

6. Objetivos específicos

- Hacer un programa que se vea y funcione como un cajero real
- Incluir funciones básicas como sacar dinero, ver el saldo y enviar dinero
- Crear pantallas fáciles de leer y con instrucciones claras
- Permitir que las personas practiquen sin miedo a equivocarse, con mensajes de ayuda y motivacionales
- Conectar el simulador con una base de datos para poder guardar información como saldos, retiros o depósitos a las cuentas ficticias

7. Diagrama de clases y relaciones



8. Requerimientos funcionales

ID	Requerimiento Funcional	Relacionado a HU	Estado de Implementación	Observaciones
RF01	Validar acceso del cliente con DNI de 8 dígitos.	HU01	✓ Implementado	Controla el formato y permite 3 intentos antes de bloquear.
RF02	Permitir la consulta del saldo actual del cliente.	HU03	✓ Implementado	Mostrado correctamente desde la interfaz con botón.
RF03	Depositar un monto positivo y actualizar el saldo.	HU04	✓ Implementado	Valida números positivos, actualiza saldo y muestra resultado.
RF04	Retirar dinero solo con validación de PIN de 4 dígitos.	HU02	✓ Implementado	Solicita PIN y valida antes de ejecutar la transacción.
RF05	Validar que el cliente tenga saldo suficiente antes del retiro.	HU05	✓ Implementado	Previene sobregiros, muestra mensaje si no hay fondos.
RF06	Mostrar una interfaz gráfica con botones para las operaciones.	HU11	✓ Implementado	Uso de JFrame y JButton, diseño claro.
RF07	Generar un recibo en .txt con los datos de la operación bancaria.	HU06, HU09	✓ Implementado	Recibo incluye nombre, operación, monto, fecha, saldo y número único.
RF08	Preguntar al usuario si desea generar un recibo antes de crearlo.	HU07	✓ Implementado	Diálogo de confirmación JOptionPane funcional.
RF09	Abrir el recibo automáticamente en el Bloc de notas tras una operación.	HU08	✓ Implementado	Usa Runtime.getRuntime().exec("notepad archivo.txt").
RF10	Mostrar número de operación único en una ventana emergente tras la operación.	HU10, HU15	✓ Implementado	Número generado aleatoriamente y mostrado al cliente correctamente.

Funcionalidades

Funcionalidad 1: Autenticación del Cliente

Descripción: Validación del acceso mediante DNI y PIN. **Prioridad:** Alta **Acciones iniciadoras y comportamiento esperado:**

- El usuario ingresa su DNI de 8 dígitos.
- El sistema verifica el formato y limita a 3 intentos.
- Si es válido, solicita el PIN de 4 dígitos.
- Se muestra mensaje de error si el formato es inválido.

Requerimientos funcionales:

- **REQ-01:** El sistema debe verificar que el DNI tenga exactamente 8 dígitos numéricos.
- **REQ-02:** Limitar los intentos de ingreso a 3; bloquear en caso de exceder.
- **REQ-03:** Validar que el PIN sea de 4 dígitos numéricos.
- **REQ-04:** Mostrar mensajes claros ante errores de formato o ingreso fallido

Funcionalidad 2: Gestión de Transacciones Bancarias

Descripción: Permitir al usuario consultar saldo, realizar depósitos y retiros. **Prioridad:** Alta **Acciones iniciadoras y comportamiento esperado:**

- El usuario accede a la pantalla principal con botones claros.
- Puede elegir entre "Consultar Saldo", "Depositar", o "Retirar Dinero".
- El sistema muestra mensajes de éxito o error según la acción.

Requerimientos funcionales:

- **REQ-05:** Mostrar saldo actual correctamente al seleccionar "Consultar Saldo".
- **REQ-06:** Aceptar depósitos solo con montos positivos.
- **REQ-07:** Validar que el usuario tenga saldo suficiente antes de permitir el retiro.
- **REQ-08:** Actualizar el saldo después de cada operación.
- **REQ-09:** Generar un recibo en archivo .txt tras una operación exitosa.

Funcionalidad 3: Accesibilidad y Retroalimentación

Descripción: Garantizar que el sistema sea comprensible para adultos mayores y personas con poca experiencia digital. **Prioridad:** Alta **Acciones iniciadoras y comportamiento esperado:**

- Activación automática de guía por voz.
- Modo de alto contraste y textos grandes habilitados por defecto.
- Animaciones ilustran cada paso.
- El sistema registra errores frecuentes y genera reportes.

Requerimientos funcionales:

- **REQ-10:** Incluir subtítulos y guía por voz en todas las instrucciones.
- **REQ-11:** Mostrar textos en tamaño grande y contrastes adecuados.
- **REQ-12:** Permitir navegación mediante clic o toque.
- **REQ-13:** Registrar errores comunes del usuario para mejorar la experiencia.
- **REQ-14:** Emitir retroalimentación positiva tras cada paso completado con éxito.

40 REQUERIMIENTOS SOLICITADOS :

Interacción del usuario (1–10)

1. El sistema debe permitir simular el ingreso de tarjeta.
2. El sistema debe mostrar una pantalla inicial con opciones básicas.
3. El sistema debe ofrecer la opción de retirar dinero.
4. El sistema debe permitir la selección de cuenta.
5. El sistema debe simular la introducción de monto.
6. El sistema debe validar si el monto es válido.
7. El sistema debe mostrar mensaje de operación exitosa.
8. El sistema debe permitir consultar saldo.
9. El sistema debe ofrecer opción de transferencias.
10. El sistema debe permitir cancelar la operación.

Accesibilidad y soporte (11–20)

11. El sistema debe tener guía por voz.
12. El sistema debe tener textos grandes y legibles.
13. El sistema debe incluir subtítulos en todas las instrucciones.
14. El sistema debe permitir ajustar volumen.
15. El sistema debe tener ayuda visual con íconos.
16. El sistema debe mostrar una animación de cada paso.
17. El sistema debe permitir repetir las instrucciones.
18. El sistema debe tener modo contraste alto.
19. El sistema debe estar disponible en español.
20. El sistema debe ser operable mediante clic o toque.

Registro y evaluación (21–30)

21. El sistema debe registrar el avance del usuario.

22. El sistema debe emitir un resumen de operaciones realizadas.
23. El sistema debe tener cuestionarios de autoevaluación.
24. El sistema debe guardar las sesiones en modo offline.
25. El sistema debe mostrar retroalimentación positiva.
26. El sistema debe incluir preguntas frecuentes.
27. El sistema debe registrar errores frecuentes del usuario.
28. El sistema debe generar reportes de progreso.
29. El sistema debe permitir imprimir simulaciones (opcional).
30. El sistema debe permitir calificar la experiencia.

Funcionalidades adicionales (31–40)

9. El sistema debe simular cambio de clave.
10. El sistema debe incluir mensajes de advertencia realistas.
11. El sistema debe simular bloqueo por clave incorrecta.
12. El sistema debe tener opción de guía paso a paso.
13. El sistema debe permitir navegación libre y por módulo.
14. El sistema debe simular transacciones con recibo.
15. El sistema debe tener opción de seleccionar tipo de operación.
16. El sistema debe tener opción de práctica libre.
17. El sistema debe simular horario de cajero y límites.
18. El sistema debe tener retroalimentación visual de cada acción.

9. Reglas de negocio

Reglas de Negocio para el Simulador de Cajero Automático

1. Autenticación y Seguridad

El acceso al sistema requiere ingresar un DNI válido de 8 dígitos, con un máximo de 3 intentos.

Para realizar retiros, el cliente debe ingresar un PIN válido de 4 dígitos.

Si el formato del DNI o PIN es incorrecto, se debe mostrar un mensaje claro y bloquear el acceso tras 3 fallos.

2. Operaciones Bancarias

Las transacciones solo se ejecutan si hay saldo suficiente, evitando sobregiros.

Los depósitos deben ser numéricos y positivos; cualquier otro formato se rechaza.

El saldo del cliente debe actualizarse correctamente tras cada operación exitosa.

3. Recibos y Confirmaciones

Cada operación genera un recibo con número único de 10 dígitos.

El cliente puede decidir si desea recibo o no antes de que se genere.

Todos los recibos deben abrirse automáticamente en el Bloc de notas (Windows).

Se debe mostrar una ventana emergente con el número de operación tras cada transacción exitosa.

4. Accesibilidad y Soporte

El sistema debe funcionar sin conexión a internet.

La interfaz debe tener texto grande, alto contraste y subtítulos en todas las instrucciones.

El sistema debe ofrecer una guía por voz y retroalimentación visual en cada paso.

Se deben registrar errores frecuentes y el progreso del usuario, para generar reportes y evaluaciones.

5. Restricciones Técnicas

El simulador debe poder ejecutarse en computadoras de bajo rendimiento, como las que hay en escuelas o hogares.

No debe requerir instalación de software adicional.

No se puede depender de servicios en línea o conexiones externas para su funcionamiento. **Requerimientos de interfaces externas**

Historias de usuario

ID	Historia de Usuario	Criterios de Aceptación
HU01	Como cliente, quiero ingresar mi DNI de 8 dígitos, para poder acceder de manera segura.	<ul style="list-style-type: none"> - El sistema solo permite avanzar si el DNI tiene exactamente 8 dígitos numéricos. - Se muestra un mensaje de error si el formato es inválido. - Se limita a 3 intentos.
HU02	Como cliente, quiero ingresar un PIN de 4 dígitos al retirar dinero, para proteger mi cuenta.	<ul style="list-style-type: none"> - Solo se permite el retiro si el PIN es correcto. - Si el PIN es incorrecto, se muestra un mensaje. - El retiro no se ejecuta sin validación de PIN.
HU03	Como cliente, quiero consultar mi saldo, para saber cuánto dinero tengo disponible.	<ul style="list-style-type: none"> - Se muestra el saldo actual del cliente en pantalla. - No se modifica el saldo. - Puede realizarse en cualquier momento tras ingresar al sistema.

HU04	Como cliente, quiero ingresar un monto a depositar, para aumentar mi saldo.	<ul style="list-style-type: none"> - El sistema acepta montos positivos numéricos. - Se actualiza correctamente el saldo del cliente. - Se notifica al usuario del nuevo saldo.
HU05	Como cliente, quiero retirar dinero solo si tengo saldo suficiente, para evitar sobregiros.	<ul style="list-style-type: none"> - El sistema verifica el saldo antes de retirar. - Si no hay fondos suficientes, se muestra un mensaje y no se procesa. - El saldo se actualiza correctamente si procede.
HU06	Como cliente, quiero recibir un recibo al realizar una operación, para tener evidencia.	<ul style="list-style-type: none"> - Después de una operación, se genera un archivo `.txt` con los detalles. - El recibo incluye nombre, operación, monto, fecha y saldo actual.
HU07	Como cliente, quiero decidir si deseo generar un recibo o no, para ahorrar papel.	<ul style="list-style-type: none"> - El sistema pregunta al usuario si desea el recibo. - Si elige "No", no se genera ningún archivo. - Si elige "Sí", se genera normalmente.
HU08	Como cliente, quiero que el recibo se abra automáticamente en el Bloc de notas, para revisar fácilmente los detalles.	<ul style="list-style-type: none"> - El archivo generado se abre automáticamente con `notepad` en Windows. - El contenido del recibo es legible y bien estructurado.
HU09	Como cliente, quiero que cada recibo tenga un número único de operación, para identificar o reportar la transacción fácilmente.	<ul style="list-style-type: none"> - Cada operación incluye un número único de 10 dígitos en el recibo. - Este número no se repite. - Está claramente visible en el recibo.

ID	Historia de Usuario	Criterios de Aceptación
HU01	Como cliente, quiero ingresar mi DNI de 8 dígitos, para poder acceder de manera segura.	<ul style="list-style-type: none"> - El sistema solo permite avanzar si el DNI tiene exactamente 8 dígitos numéricos. - Se muestra un mensaje de error si el formato es inválido. - Se limita a 3 intentos.
HU02	Como cliente, quiero ingresar un PIN de 4 dígitos al retirar dinero, para proteger mi cuenta.	<ul style="list-style-type: none"> - Solo se permite el retiro si el PIN es correcto. - Si el PIN es incorrecto, se muestra un mensaje. - El retiro no se ejecuta sin validación de PIN.
HU03	Como cliente, quiero consultar mi saldo, para saber cuánto dinero tengo disponible.	<ul style="list-style-type: none"> - Se muestra el saldo actual del cliente en pantalla. - No se modifica el saldo. - Puede realizarse en cualquier momento tras ingresar al sistema.

Interfaz Principal

- **Pantalla de bienvenida** con opciones básicas: consultar saldo, retirar dinero, depositar, salir.
- **Botones grandes y etiquetados claramente** usando **JButton**, diseñados para ser reconocibles por personas con poca experiencia digital.
- **Modo de alto contraste** activado por defecto para mayor legibilidad.
- **Texto en fuente grande** y guía por voz complementaria para reforzar comprensión.

Pantalla de Autenticación

- Campo para ingresar **DNI de 8 dígitos** con validación en tiempo real.
- Campo de ingreso de **PIN de 4 dígitos** protegido, con mensajes claros si se comete un error.
- Mensajes emergentes (**JOptionPane**) para mostrar intentos fallidos o acceso exitoso.

Simulación de Cajero

- Interfaz que **simula la apariencia física de un cajero real**.
- Mensaje de confirmación tras cada operación exitosa.

Pantallas de Transacción

- **Consulta de Saldo:** muestra el monto actual en pantalla.
- **Depósito y Retiro:** validación de montos, confirmaciones y actualizaciones del saldo.
- **Mensajes de error amigables** si el monto ingresado no es válido.

Generación de Recibos

- Pregunta al usuario si desea generar un recibo.
- Si acepta, se crea un archivo **.txt** con la información de la operación (nombre, monto, fecha, saldo, número de operación).
- El recibo se **abre automáticamente en el Bloc de notas** para facilitar revisión.

Interfaz de Accesibilidad

- **Guía por voz automática** explicando cada paso.
- **Subtítulos** en todas las instrucciones.
- **Animaciones didácticas** que muestran visualmente cómo realizar una acción.
- Navegación por **clic o toque**, según dispositivo.
- Posibilidad de **repetir instrucciones** si el usuario lo solicita.

Requerimientos no funcionales

Requerimientos No Funcionales del Simulador de Cajero Automático

1.- Rendimiento y Compatibilidad

- El sistema debe funcionar en computadoras de bajo rendimiento, como las disponibles en escuelas o hogares.
- No debe requerir conexión a internet para operar (modo offline por defecto).
- Todas las operaciones deben completarse en menos de 2 segundos, incluyendo generación de recibos y validación de datos.

2.- Usabilidad

- La interfaz debe ser intuitiva, con botones claramente etiquetados y funciones agrupadas visualmente.
- Los textos deben estar escritos en lenguaje sencillo, evitando jerga técnica.
- La navegación debe poder realizarse con clics o toques, sin necesidad de teclado avanzado.

Glosario

Descripción de términos y siglas necesarias para el entendimiento del documento de requerimientos de software.

JOptionPane	Componente de Java que permite mostrar mensajes emergentes al usuario.
Validación de Datos	Proceso de verificar que la información ingresada por el usuario cumple con los formatos requeridos.
Accesibilidad	Diseño que permite usar el sistema fácilmente, especialmente para adultos mayores o personas con discapacidad.
Modo Contraste Alto	Configuración visual que aumenta la diferencia entre colores para facilitar la lectura.
Guía por Voz	Instrucciones sonoras que explican paso a paso cómo utilizar el simulador.
Número de Operación	Código único generado para identificar cada transacción realizada.
Notepad	Programa de edición de texto en Windows donde se abren automáticamente los recibos generados.
Retroalimentación	Mensajes que indican si una acción fue exitosa o si debe corregirse algún error.
Operación Bancaria	Acción dentro del simulador como consultar saldo, retirar o depositar dinero.
Commit	Registro de cambios realizados al código en GitHub.
Branch (Rama)	Versión paralela del proyecto utilizada para implementar nuevas funcionalidades sin afectar la principal.
Interacción del	Secuencia de acciones que el usuario realiza para operar el

Usuario

simulador.

19. Bibliografía

Amat, A. N. (2019). *Arduino based automated teller machine (ATM) for children financial learning* [Trabajo de grado, Universiti Teknologi PETRONAS]. UTPedia.
<https://utpedia.utp.edu.my/20884/>

Carangui Cárdenas, L. R., Cajamarca Criollo, O. A., & Mantilla Crespo, X. A. (2017). Impacto del uso de simuladores en la enseñanza de la administración financiera. *Innovación Educativa (México, DF)*, 17(75), 103–122. <https://www.redalyc.org/articulo.oa?id=179454112006>

Dalbey, J. (s. f.). *ATM simulator: Software engineering case study*.
<https://users.csc.calpoly.edu/~jdalbey/SWE/CaseStudies/ATMSim/>

Forcheri, P., & Molfino, M. T. (1989). The teller simulator: An automatic problem solver for educational use. En H. Maurer (Ed.), *Computer assisted learning* (Vol. 360, pp. 117–131). Springer. https://doi.org/10.1007/3-540-51142-3_55

Kopeć, W., Wichrowski, M., Kalinowski, K., Jaskulska, A., Skorupska, K., Cnotkowski, D., & Kornacka, M. (2019). *VR with older adults: Participatory design of a virtual ATM training simulation*. arXiv. <https://arxiv.org/abs/1911.00466>

Mead, S., & Fisk, A. D. (1998). Measuring skill acquisition and retention with an ATM simulator: The need for age-specific training. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 53(6), P343–P353. <https://doi.org/10.1093/geronb/53B.6.P343>