

Actividades Preliminares.

Contexto:

①. La especificación de un problema.

a) Resuma el ciclo de vida de construcción de un Programa.

Rta: El Ciclo de vida del desarrollo de Software (SDLC) se describen varias tareas necesarias para crear una nueva aplicación de Software. Para desarrollar esta es necesario que pase por varias etapas de medida que establecen los desarrolladores ya que estos agregan nuevas características y corrigen errores del Software.

- El ciclo de vida de el Software implica las fases de interconectadas que incluyen la planificación, el diseño, la implementación, la prueba, la entrega y mantenimiento.

1. Planificación: Se define los objetivos del proyecto y se elabora un plan detallado para el desarrollo del Software.

2. Diseño: Se realizan diseños arquitectónicos y técnicos del sistema, explicando como va a Funcionar.

3. Implementación: Se escribe el Código real del Software basado en las especificaciones dadas por los creadores.

4. Pruebas: Se hacen pruebas exhaustivas para Identificar y Corregir errores y asegurar que el Software funcione Correctamente.

5. Despliegue: Se implementa en el entorno de Producción para el uso por parte de los usuarios finales.

6. Mantenimiento: Para finalizar, el Software se mantiene y actualiza regularmente para corregir errores, también agregar nuevas características y garantizar su funcionalidad continua.

b) Explique los aspectos que hacen parte del análisis de un problema.

Rta:

1. Definición del Problema: Entender claramente cuál es el problema que se necesita resolver. También identificar los objetivos y los requisitos del mismo.
2. Análisis de requisitos: Aquí se determinan los requisitos necesarios para resolver el problema, tanto funcionales como no funcionales. (Funcionales - son los que deseamos hacer el programa y los no funcionales tiene restricciones en cuanto al tiempo de ejecución, uso de memoria entre otros).
3. Descomposición del Problema: Es donde se divide el problema en subproblemas más pequeños y manejables para facilitar la comprensión y solución.
4. Identificación de datos y estructuras de datos: Escoger qué datos serán necesarios para resolver el problema y cómo se va a almacenar.
5. Diseño de algoritmos: Desarrollar un plan paso a paso (algoritmo) para poder resolver el problema.
6. Identificación de casos especiales: Considerar los casos extremos y todas las posibles excepciones que podrían surgir durante la ejecución del programa.
7. Evaluación y Validación: Evaluar si el análisis y todo el diseño del algoritmo cumple con los requisitos iniciales y si resuelve el problema de manera efectiva.

C. Explique las etapas del proceso de solución de problemas

- La solución de problemas tiene varios componentes fundamentales los cuales son:

1. El diseño = En esta etapa se muestra la estructura de la solución, sus partes y sus relaciones, donde se define la estructura del programa para facilitar su mantenimiento. En esta etapa como se le debe dar una estructura al programa se debe realizar primero un algoritmo el cual nos especifica los pasos para resolver el problema, siguiente a eso se realiza un diagrama de flujo el cual representa una solución gráfica del algoritmo y por último se hace uso del lenguaje pseudocódigo como una semiestructura para organizar la solución.

2. Código fuente = En esta etapa el desarrollador del programa para la solución del problema escoge algún lenguaje de programación tales como Java, C, C# ó C++, normalmente se utiliza un lenguaje de programación que vaya enfocado a una programación orientada a objetos.

3. Archivos de construcción del programa = En estos archivos se hace énfasis en la explicación de la forma de uso para el código fuente para crear el código ejecutable, el cual es la versión traducida del código fuente a un lenguaje de máquina como el binario para que la computadora pueda entenderlo y ejecutarlo directamente, el programa que permite traducir el código fuente en código ejecutable se denomina compilador.

4. Pruebas = En esta etapa se hace entrega de un programa que funciona correctamente al cliente, para ello se hace uso de un programa que se encarga de probar que el programa desarrollado funciona, dicho programa funciona sobre un conjunto definido de datos y es capaz de validar que para esos datos predefinidos que a su vez simulan datos reales, el programa funciona bien.

d. ¿ Cuáles son los elementos que se deben entregar a un cliente?

La solución de un problema tiene tres partes:

- El diseño,
- El programa.
- Las pruebas de corrección del programa.

Estos son los elementos que se deben entregar al cliente.

En programación para identificar un problema a un cliente se deben de entregar los datos de entrada y salida, así como una descripción del problema.

e. Elabore la tarea No.1 (pag. 5 del texto guía), con el objetivo de identificar los aspectos que forman parte de un problema.

<u>Tarea</u>	<u>Objetivo:</u> Identificar los aspectos que forman parte de un problema.
<u>Nº.1</u>	<u>El problema:</u> Un banco quiere crear un programa para manejar sus cajeros automáticos. Dicho programa sólo debe permitir retirar dinero y consultar el saldo de una cuenta. Identifique y discuta los aspectos que constituyen el problema. Si el enunciado no es explícito con respecto a algún punto, intente manejar la manera de completarlo.
<u>Cliente</u>	<u>El Banco</u>
<u>Usuario</u>	<u>Clientes del Banco.</u>

Requerimiento Funcional.

R₁ = El programa debe permitir al usuario retirar dinero de su cuenta (siempre y cuando tenga fondos suficientes).

R₂ = El programa debe permitir al usuario consultar el saldo actual de su cuenta.

Mundo del Problema.

Para resolver el problema es necesario conocer las cuentas bancarias que tienen los clientes en el banco para acceder a los cajeros automáticos; la red de comunicación que conecta a los cajeros automáticos con el sistema central del Banco; el sistema central del Banco que procesa las transacciones y gestiona la información de las cuentas y los clientes; las medidas de seguridad que se implementan para proteger la información de los clientes; los dispositivos (cajeros automáticos) que permiten a los clientes retirar dinero y consultar su saldo.

Requerimiento no Funcional.

R₁ = El programa debe ser capaz de manejar un gran número de transacciones y usuarios, sin afectar su rendimiento.

R₂ = El programa debe ser fácil de usar y de entender para los clientes.

R₃ = El programa debe ser compatible con diferentes sistemas operativos y dispositivos; el programa debe cumplir con las regulaciones y leyes bancarias.



ACE Scanner

F. Elabore la tarea No. 2 (pág. 13), con el objetivo de identificar los requerimientos funcionales de un problema.

Tarea 2. Objetivo: Crear habilidad en la identificación y especificación de requerimientos funcionales.

Para el caso de estudio 2, un simulador bancario, identifique y especifique tres requerimientos funcionales.

Requerimiento funcional No. 1	<u>Nombre:</u> Gestión de los productos bancarios. <u>Resumen:</u> Permitir al usuario realizar operaciones sobre los productos que conforman la cuenta. <u>Entradas:</u> <ul style="list-style-type: none">• Número de cuenta.• Tipo de cuenta.• Monto de la transacción.• Interés negociado (interés mensual negociado para el CDT).• Duración del CDT.• Acción (depositar, retirar, abrir, cerrar). <u>Resultado:</u> La cuenta del cliente se actualiza según la transacción realizada, mostrando cambios en el saldo, los productos bancarios y la información de la cuenta.
Requerimiento funcional No. 2.	<u>Nombre:</u> Simular los intereses y rendimientos. <u>Resumen:</u> Permitir calcular y mostrar el interés mensual que genera cada producto bancario.



ACE Scanner

<p><i>Requerimiento funcional No. 2.</i></p>	<p><u>Entradas:</u></p>	<ul style="list-style-type: none"> • Tipo de producto. • Saldo inicial. • Tasa de interés. • Período de tiempo.
	<p><u>Resultado</u></p>	<p>El programa calcula y muestra los intereses y rendimientos generados durante un período de tiempo específico.</p>
	<p><u>Nombre:</u></p>	<p>Se muestra el saldo final del producto bancario.</p>
	<p><u>Resumen:</u></p>	<p>Actualizar el saldo de la cuenta mes a mes.</p>
<p><i>Requerimiento funcional No. 3.</i></p>	<p><u>Entradas:</u></p>	<ul style="list-style-type: none"> • Saldo actual. • Intereses generados. • Depósitos. • Retiros. • Fechas de vencimiento de productos. • Actualización del saldo.
	<p><u>Resultado.</u></p>	<p>Saldo actualizado; reflejando los cambios realizados durante el mes.</p>



ACE Scanner

Caso 3 = Un programa para manejar un triángulo, identifique y especifique 3 requerimientos funcionales

	Nombre	Dibujar el triángulo en pantalla
Requerimiento Funcional 1	Resumen	Permite al usuario ingresar los datos de X, Y para las coordenadas junto con los colores de linea y fondo según la selección del usuario.
	Entradas	- Valores X, Valores Y (para los tres puntos). - Color de líneas. - Color de fondo
	Resultado	Representación gráfica del triángulo según las coordenadas ingresadas y los colores seleccionados para las líneas y el fondo
	Nombre	Calcular las propiedades Geométricas del triángulo
Requerimiento Funcional 2	Resumen	Calcula y muestra el resultado del área, perímetro y altura del triángulo a partir de las coordenadas ingresadas por el usuario.
	Entradas	<ul style="list-style-type: none"> • Calcular la distancia entre las coordenadas (X, Y) de los tres lados del triángulo • Base del triángulo (1 de los lados)
	Resultado	<ul style="list-style-type: none"> • Valor de los lados del triángulo • Perímetro del triángulo • Área del triángulo • Altura del triángulo
	Nombre	Modificación del triángulo
Requerimiento Funcional 3	Resumen	Permite que el usuario pueda modificar tanto las coordenadas de los tres lados del triángulo como también el color del fondo y de las líneas
	Entradas	<ul style="list-style-type: none"> • Nuevas coordenadas (X, Y) de los puntos de el triángulo • Nuevo color de líneas y/o de fondo
	Salidas	<ul style="list-style-type: none"> • Triángulo Actualizado en la pantalla con nuevas coordenadas y colores • Cálculo Automático de sus propiedades.

Tarea 4

Objetivo = Identificar las entidades del mundo para el caso de estudio
3: Un programa que maneja un triángulo. Lee el enunciado del caso y trate de guiarse por los sustantivos para identificar las entidades del mundo

Problema

Nombre	Descripción
Entidad	Triángulo
Entidad	Usuario
Entidad	Sistema
<ul style="list-style-type: none"> Punto de reflexión = ¿Qué pasa si no identificamos bien las entidades del mundo? 	
<p>RII Si no se identifican bien las entidades del mundo, pueden surgir muchas dificultades en el desarrollo del sistema puesto que si los</p>	

entidades no estan bien definidas el diseño del programa por ende tambien puede ser dificil de mantener, lo que haria que en el procesamiento de datos las entidades no tengan relaciones coherentes, lo que puede llevar a la realizacion de calculos incorrectos o que el programa no pueda funcionar, por lo que se presentaron dificultades en la implementacion y mantenimiento del programa.

- Punto de reflexión: ¿Cómo decidir si se trata efectivamente de una entidad y no sólo por una característica de una entidad ya identificada?

RII Para que sea una entidad debemos tener en cuenta que esta debe ser un sustituto dentro del mundo del problema y representar un objeto con identidad propia. Es decir, no debe ser solo una característica o un requerimiento solicitado, sino algo que pueda existir por si mismo dentro del sistema.

→ Tarea No. 5 (Pag. 20)

Clase: CuentaBancaria

Atributo	Valores Posibles	Diagrama UML
Nombre titular	Cadena texto (String)	
Saldo	Número decimal o punto flotante (float o double)	- Nombre titular - Saldo - Clave de acceso
Clave de acceso	Cadena de texto (string) cualquier combinación de letras, números y símbolos.	

Clase: CuentaCorriente

Atributo	Valores Posibles	Diagrama UML
Saldo actual	Decimal (float o double)	
Tarjeta de débito	Entero (int) o Cadena de texto (String), cualquier combinación de letras, números y símbolos	- Saldo actual - Tarjeta de débito - Acceso en línea
Acceso en línea	Booleano (bool) Verdadero o Falso	

Clase: CuentaAhorros

Atributo	Valores Posibles	Diagrama UML
Monto mínimo	Decimal (float o double)	
Tasa de interés	Decimal (float o double) 0,00 a 100.000% Porcentaje	- monto mínimo - tasa de interés - Plazo de ahorro
Plazo de ahorro	Entero (int) o Cadena de texto (String)	

Clase CDT

Atributo	Valores Posibles	Diagrama UML
Estado del CDT	Cadena de texto (String) o numeración (enum)	
titular del CDT	Cadena de texto (String) Cualquier combinación de letras y espacios.	
Plazo	Entero (int)	<pre> classDiagram class CDT { Estado del CDT Titular del CDT -Plazo } </pre>

clase: mes.

Atributo	Valores Posibles	Diagrama UML
Año	Entero (int)	
Nombre del mes	Cadena de texto (String)	
numero de mes	Entero (int) 1-12.	<pre> classDiagram class mes { -Año -Nombre del mes -Número de mes } </pre>

j => Tarea N°. 6 (Pag 23)

Conclusiones:

1. Idioma, ya que para un extranjero sera muy difícil, porque si no maneja correctamente el idioma no encontrara la instrucción correcta.
2. Tampoco el nombre de la estación, ya que para una persona que vive allí es fácil porque todos los días puede transcurrir por este medio de transporte y conocerá el nombre de las estaciones. Pero para una persona que es nueva en este lugar es complicado.
3. las instrucciones están bien escritas pero antes de ejecutarlas toca estudiar primero el lugar a donde vamos.
4. El turista o yo como persona se me hace muy difícil no conocer el horario que tiene con mayor flujo, ya que uno como turista no conoce fácilmente estos datos.



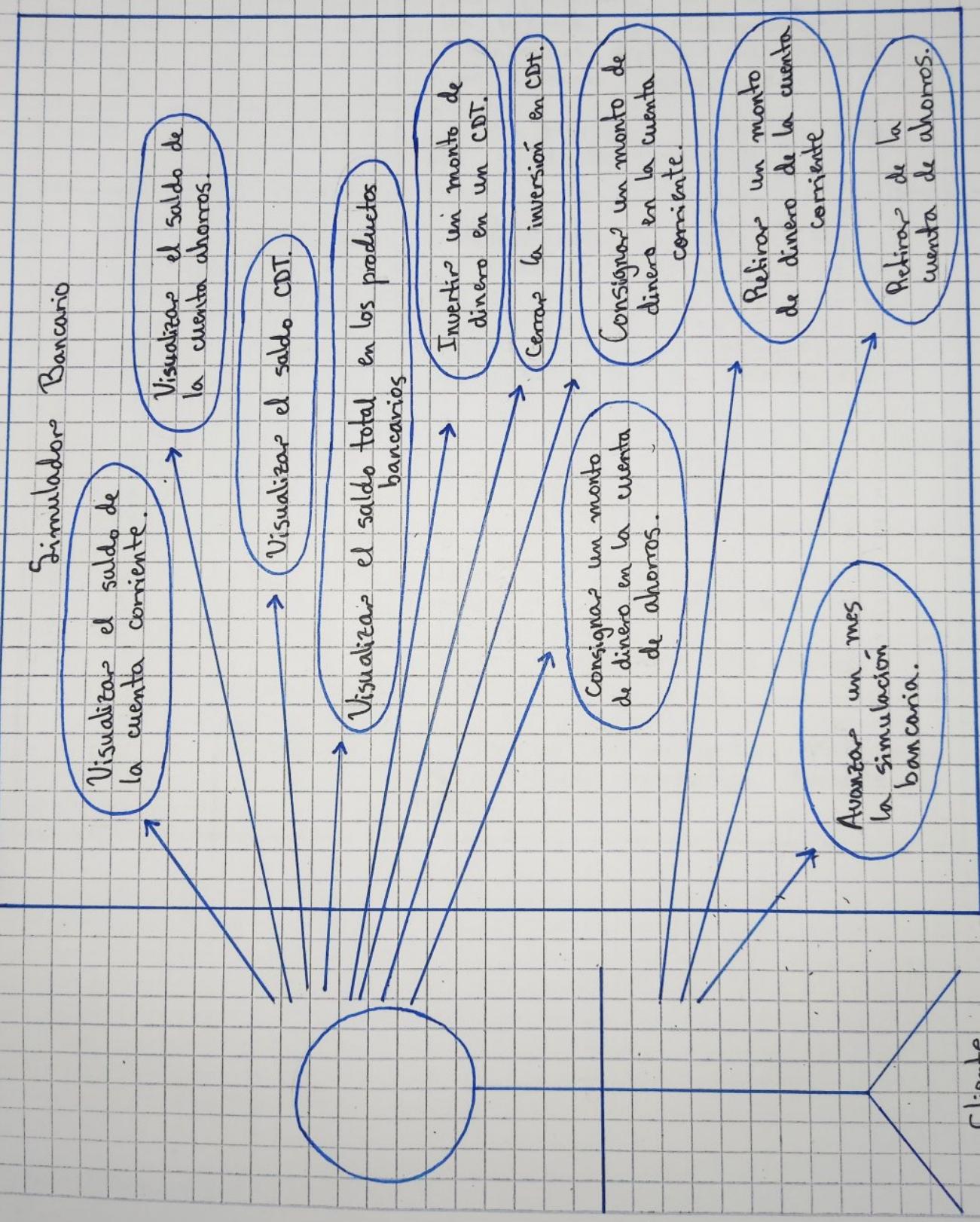
- K. En la vida crediticia de una persona es común el manejo de su dinero en diferentes escenarios que le permiten a la persona saber realmente si tiene un buen manejo de sus finanzas a través del tiempo, donde estos diferentes escenarios le brindan una ventaja en el presente como en el futuro.

Un banco requiere que se cree un programa que haga una simulación con respecto al tiempo de la cuenta de un cliente. Para llevar un control del cliente se requiere su nombre y número de cédula. El sistema requiere que una cuenta este constituida por tres productos financieros básicos: (1) Una cuenta corriente, (2) Una cuenta de ahorros y (3) un certificado de depósito a término (CDT).

Cada uno de los productos son independientes y tienen comportamientos particulares. En la cuenta corriente el cliente puede depositar o retirar dinero, pero sin recibir ningún interés por la cantidad o valor de dinero depositado. En la cuenta de ahorro, el cliente también puede depositar y retirar dinero, pero a diferencia de la cuenta corriente, esta cuenta recibe un interés del 0.67% mensualmente sobre el saldo actual. Cuando el cliente abre un CDT; define la cantidad de dinero que quiere invertir y negociar con el banco el interés mensual que va a recibir, a diferencia de la cuenta corriente o la cuenta de ahorros, en un CDT no se puede consignar ni retirar dinero. La única operación posible es cerrado, en cuyo caso, el dinero y sus intereses pasan a la cuenta corriente. Sin olvidar lo más importante y es que la suma de lo que tiene cada cliente en cada uno de estos productos es el saldo total de la cuenta.

Por lo tanto el sistema requiere el programa le permita al banco simular el manejo de los productos bancarios del cliente, brindandole las facilidades de ① visualizar el saldo de las cuentas, el CDT y por supuesto el saldo total que tiene el cliente en los productos del banco ② Consignar un monto de dinero tanto en la cuenta de ahorros como en la corriente ③ Retirar un monto de dinero en la cuenta de ahorros y/o la cuenta corriente ④ Invertir un monto de dinero en un CDT o cerrar la inversión del mismo y por supuesto esto basado en una simulación de tiempo mes a mes, para que el cliente pueda ver el resultado de sus movimientos bancarios y el rendimiento de sus inversiones.

1. Dibuja el respectivo diagrama de casos de uso del ejemplo elegido. (consulta " requerimiento Funcionales " en " Archivo ")



Punto m= Observa nuevamente el modelo conceptual del caso y escribe el nombre de cada una de las clases identificando sus respectivas variables atributos y funciones.

Simulador Banco

- cedula = String
- Nombre = String
- mes Actual = int
- + Simulador_banco (pcedula:String, pNombre:String)
- + dar nombre () : String
- + dar Cedula () : String
- + dar cuenta corriente () : Cuenta corriente
- + dar Cuenta Ahorros () : Cuenta Ahorros
- + dar CDT () : CDT
- + dar Mes Actual () : int
- + calcular Saldo Total () : double
- + invertir CDT (pMonto:double, pInteresMensual: double) : void
- + consignar cuentaCorriente (pMonto:double) : void
- + Consignar cuenta Ahorros (pMonto:double) : void
- + retirar CuentaCorriente (pMonto:double) : void
- + retirar CuentaAhorros (pMonto:double) : void
- + cerrar CDT () : void
- + metodo 1 () : String
- + metodo 2 () : String

Inversión

COT
- valorInvertido : double
- InteresMensual : double
- mesApertura : int
+ CDT
+ dasInteresMensual () : double
+ Invertir (pMontoInvertido : double, pInteresMensual : double, pMes : void)
+ calcularValorPresente (pMesActual : int) : double
+ Cerrar (pMesActual : int) : double

compte

Cuenta Corriente
- Saldo : double
+ cuenta corriente ()
+ darSaldo () : double
+ consignarMonto (pMonto:double) : void
+ retirarMonto (pMonto:double) : void

ahorros

Cuenta Ahorros
- Saldo : double
- InteresMensual : double
+ CuentaAhorros ()
+ darSaldo () : double
+ dasInteresMensual () : double
+ consignarMonto (pMonto:double) : void
+ retirarMonto (pMonto:double) : void
+ actualizarSaldoPorPasoMes () : void

n. Debes plantear 2 ideas de proyecto (problemas solubles y algorítmicos);

Idea 1:

Una tienda de ropa quiere crear un programa que le permita registrar las ventas y el material que se compra para abastecer la tienda; se debe registrar un catálogo sobre los productos que ofrece, y adicionalmente, el programa debe permitir saber cuánto dinero posee en caja.

Requerimiento Funcional No. 1	Nombre: <u>Resumen.</u>	Registrar las ventas.
	<u>Entradas.</u>	<ul style="list-style-type: none">• Fecha de venta.• Producto vendido.• Cantidad vendida.• Precio de venta.• Forma de pago.• Cliente.
	<u>Resultado.</u>	Registro de ventas. La información de las ventas se registran y se guardan de manera exitosa.
Requerimiento Funcional No. 2	Nombre: <u>Resumen.</u>	Registrar compra de material.

Permite guardar información de cada compra.



ACE Scanner

<p><u>Requerimiento Funcional No. 2</u></p>	<p><u>Entradas:</u></p> <ul style="list-style-type: none"> • Fecha de compra. • Proveedor. • Producto. • Cantidad comprada. • Precio de compra. • Forma de pago. <p><u>Resultado:</u></p> <p>Registro de compra que permite realizar un seguimiento de las mismas.</p>
<p><u>Requerimiento Funcional No. 3</u></p>	<p><u>Nombre:</u></p> <p>Realizar catálogo de los productos.</p> <p><u>Resumen:</u></p> <p>Permite crear y editar un catálogo de productos.</p> <p><u>Entradas:</u></p> <ul style="list-style-type: none"> • Nombre del producto. • Descripción del producto. • Precio del producto. • Categoría del producto. • Imagen del producto. <p><u>Resultado:</u></p> <p>Catálogo de productos actualizado.</p>
<p><u>Requerimiento Funcional No. 4</u></p>	<p><u>Nombre:</u></p> <p>Gestionar el control de caja.</p> <p><u>Resumen:</u></p> <p>Permite registrar salidas y entradas de dinero, además, mostrar el saldo actualizado.</p> <p><u>Entradas:</u></p> <ul style="list-style-type: none"> • Tipo de transacción. • Fecha de transacción. • Monto de la transacción. • Descripción de la transacción. • Saldo inicial.

Resultado

Saldo actualizado de la caja. Permitiendo tener un control preciso y actualizado del dinero en la caja.



ACE Scanner

Idea N° 2= Una papelería necesita la creación de un programa que le permita llevar de manera organizada la gestión de ventas, control del inventario y clientes para mejorar su eficiencia, para así evitar errores en el control de sus productos.

	Nombre	Gestión y procesamiento de ventas
Requerimiento	Requerimiento	Permite al usuario ingresar el número de ventas del día y generar un reporte con el total de ingresos.
Funcional 1	Entrada	<ul style="list-style-type: none"> - Ingrese el número de ventas realizadas - Ingrese el valor de cada venta - Ingrese el nombre del producto - Cantidad total de ventas realizadas - Suma total de los valores de cada producto - Reporte final junto a un desglose por producto.
	Salida	
	Nombre	Gestión de productos e inventario
Requerimiento	Requerimiento	Permite al usuario ingresar el tipo de producto, clasificarlo en una categoría y asignarle detalles como Precio, código y productos disponibles.
Funcional 2	Entrada	<ul style="list-style-type: none"> - tipo de producto - Clasificación de categorías - Precio del producto - Código del producto - Productos disponibles
	Resultado	<ul style="list-style-type: none"> - cantidad total de productos disponibles divididos por categoría. - Valor unitario de cada producto - inventario Actualizado.
	Nombre	Registro de clientes
Requerimiento	Requerimiento	Permite al usuario ingresar los datos Personales de cada cliente junto con su historial de compras para mejorar la eficiencia y ofrecer promociones personalizadas.
Funcional 3		

	Entradas	- Nombre del cliente - Teléfono o correo electrónico - Historial de compras
	Resultado	- Base de datos de clientes organizada. - Reporte de compras por cliente. - Posibilidad de ofrecer descuentos.
	Nombre	Gestión de Proveedores
Requerimiento Funcional	Resumen	Permite registrar y gestionar proveedores para facilitar la reposición de productos y llevar manejo de calidad.
4	Entradas	- Nombre del proveedor - Contacto (Teléfono o correo electrónico) - Productos que suministra. - Historial de pedidos pasados.
	Resultado	- Base de datos de proveedores organizada - Registros de pedidos de cada proveedor - Optimización de abastecimiento de productos