

Grupo #1

Juan Pablo Marquez - Nicol

a) Ciclo de vida de construcción de un programa.

El ciclo de vida de construcción de un programa se refiere a las fases que atraviesa un software desde su inicio hasta su finalización. El proceso consta de tres actividades principales:

Paso 1: Una persona u organización denominada cliente, tiene un problema y necesita la construcción de un programa para resolverlo. Para esto contacta una empresa de desarrollo de software que pone a su disposición un programador.

Paso 2: El programador sigue un conjunto de etapas, denominadas el proceso, para entender el problema del cliente y construir la manera organizada una solución de buena calidad, de la cual formará parte un programa.

Paso 3: El programador instala el programa que resuelve el problema en un computador y deja que el usuario lo utilice para resolver el problema.

b) Aspectos que hacen parte del análisis de un problema.

- Analizar un problema es tratar de entenderlo. Esta etapa busca garantizar que no tratemos de resolver un problema diferente al que tiene el cliente.
- Descomponer el problema en sus tres aspectos fundamentales, facilita la tarea de entenderlo.
- Además de entender el problema, debemos expresar lo que entendemos siguiendo algunas convenciones.

c) Las etapas del proceso de solución de problemas.

La primera etapa para resolver un problema es su análisis, para facilitar este proceso se descompone en sus tres partes: Análisis del problema, Diseño de la solución, Construcción de la solución.

Análisis del problema: El objetivo de este paso es entender el problema a tal punto que podamos explicarlo a un tercero, es recomendable consignar un documento donde se identifiquen los servicios que el programa debe ofrecer al usuario.

Diseño de la solución: Una vez entendido el problema diseñamos la solución, en esta detallamos las características que tendrá la solución, antes de ser construida esto nos dará una solución antes de comenzar el proceso de fabricación propiamente dicha.

Construcción de la solución: El último paso es escribir el código en el lenguaje de programación que establecimos, implementando el diseño que definimos en la etapa anterior, es importante, una vez terminada la construcción del programa, probarlo para comprobar su funcionamiento.

d) Elementos que se deben entregar a un cliente.

Al cliente se le entrega: El diseño, el programa y las pruebas de corrección del programa, es común que, además de los tres elementos citados anteriormente, también se incluyan un manual del usuario, que explique el funcionamiento del programa.

Tarea Identificar los aspectos que forman parte de un problema.

Cliente	El banco
Usuario	Clientes del banco
Requerimiento funcional	R1: El programa debe permitir retirar dinero. R2: El programa debe permitir consultar el saldo de cuenta.
Mundo del Problema	En el enunciado no está explícito, pero para poder resolver el problema, es necesario tener acceso a una base de datos que cuente con la información de las cuentas de los clientes.
Requerimiento no funcional	El único requerimiento no funcional mencionado en el enunciado es la instalación del programa en los diferentes cajeros automáticos.

TAREA 2 (F)

Requerimiento Funcional 1	Nombre	Registro de usuario
	Resumen	El sistema debe permitir a los usuarios registrarse proporcionando información personal y creando credenciales de acceso
	Entradas	Nombre, apellido, correo electrónico, número de identificación, contraseña
Requerimiento Funcional 2	Resultado	Usuario registrado en el sistema con credenciales válidas para iniciar la sesión
	Nombre	Deposito de dinero
	Resumen	El sistema debe permitir a los usuarios realizar depósitos en su cuenta bancaria dentro del simulador
Requerimiento Funcional 3	Entradas	Número de cuenta, monto a depositar
	Resultado	El saldo de la cuenta del usuario aumenta según el monto depositado
	Nombre	Retiro del dinero
Requerimiento Funcional 3	Resumen	El sistema debe permitir a los usuarios retirar dinero de su cuenta siempre que tengan saldo suficiente
	Entradas	Número de cuenta, monto a retirar
	Resultado	El saldo de la cuenta del usuario disminuye según el monto retirado, siempre que haya fondos disponibles

TAREA 5 (G)

Requerimiento Funcional 1	Nombre	clasificación del triángulo
	Resumen	El sistema debe determinar el tipo de triángulo según las longitudes de sus lados (equilátero, isósceles, escaleno).
	Entradas	Longitud de los tres lados del triángulo
Requerimiento Funcional 2	Resultado	El sistema muestra si el triángulo es equilátero, isósceles, escaleno.
	Nombre	cálculo del perímetro
	Resumen	El sistema debe calcular el perímetro del triángulo sumando sus tres lados.
Requerimiento Funcional 3	Entradas	Longitud de los tres lados del triángulo
	Resultado	El sistema muestra el valor del perímetro del triángulo.
	Nombre	cálculo del área
Requerimiento Funcional 3	Resumen	El sistema debe calcular el área del triángulo utilizando la fórmula de Herón.
	Entradas	Longitud de tres lados del triángulo
	Resultado	El sistema muestra el área del triángulo calculada con la fórmula de Herón.

TAREA 4 (H)

	Nombre	Descripción
Entidad	Triángulo	Figura geométrica con tres lados y tres ángulos. Puede clasificarse según sus lados o sus ángulos.
Entidad	Lado	Segmento de línea que forma parte del triángulo. Se utilizan sus longitudes para clasificar el triángulo y calcular su perímetro.
Entidad	Ángulo	Espacio entre dos lados del triángulo. Se utilizan para clasificar el triángulo en acutángulo, rectángulo o obtusángulo.

Punto de reflexión: ¿Qué pasa si no identificamos bien las entidades del mundo?

- El modelo del sistema nos puede dar incorrecto, lo que esto afectaría su funcionalidad en el programa. Lo cual si no se considera correctamente el triángulo con sus lados y ángulos los cálculos serán incorrectos.

Punto de reflexión: ¿cómo decidir si se trata efectivamente de una entidad y no solo de una característica de una entidad ya identificada?

- Si es una entidad propia, si puede tener atributos o comportamientos específicos dentro del sistema.

TAREA 5 (I)

Clase: cuenta bancaria

Atributo	Valores Posibles	Diagrama UML
Número de cuenta	Número de 10 dígitos	<div>Cuenta Bancaria</div> <div>Número de cuenta, Saldo, titular</div>
Saldo	Valor numérico (positivo-negativo)	
Titular	Nombre del propietario de la cuenta	

Clase: cuenta corriente

Atributo	Valores Posibles	Diagrama UML
Límite de crédito	Valor numérico positivo	<div>Cuenta corriente</div> <div>Límite de crédito, comisión, descubierto</div>
comisión	Porcentaje aplicado por mantenimiento de cuenta	
Descubierto permitido	Si y No	

Clase: Cuenta Ahorros

Atributo	Valores Posibles
Tasa de Interés	Porcentaje anual aplicado al saldo
Movimientos permitidos	Número de retiros permitidos por mes
Monto Mínimo	Valor mínimo requerido para mantener la cuenta activa

Diagrama UML

```

classDiagram
    class CuentaAhorros {
        Tasa de Interés
        Movimientos Permitidos
        Monto Mínimo
    }
  
```

Clase = CDT

Atributo	Valores Posibles
Número de Certificado	Número Único de Identificación (10 dígitos)
Monto	Valor numérico (establecido por el banco)
Tiempo	30, 60, 90, 180, 360 o más

Diagrama UML

```

classDiagram
    class CDT {
        Número de certificado
        Monto
        Tiempo
    }
  
```

Clase = MES

Atributo	Valores Posibles
Nombre del mes	Enero, febrero, marzo, etc...
Día del mes	1, 2, 3, 4, 5, 6, 7, etc...
Bisiestro	Si o No

Diagrama UML

```

classDiagram
    class MES {
        Nombre del mes
        Día del mes
    }
  
```

Tarea 6.11

Si no se especifica desde qué estación se comienza el recorrido, cual es la estación de destino, el usuario no sabrá el camino correcto, si no se dice que la línea del metro el usuario podría equivocarse.

Si el algoritmo dice "tome la línea correcta" pero el metro va a ambos sentidos podrá haber confusión, lo mejor será ser más específico y por ejemplo si dice "bajar a la cuarta estación", será confuso, por lo que mejor será nombrar todas las estaciones.

Si personas de otros países que no hablan el mismo idioma se les va a complicar, lo cual lo más apropiado será agregar frases e indicaciones cortas para que ellos puedan guiarse mejor en el sistema.

K) Enunciado

Se requiere crear un programa que permita simular el comportamiento de las cuentas bancarias de un cliente. Un cliente puede tener tres productos financieros básicos:

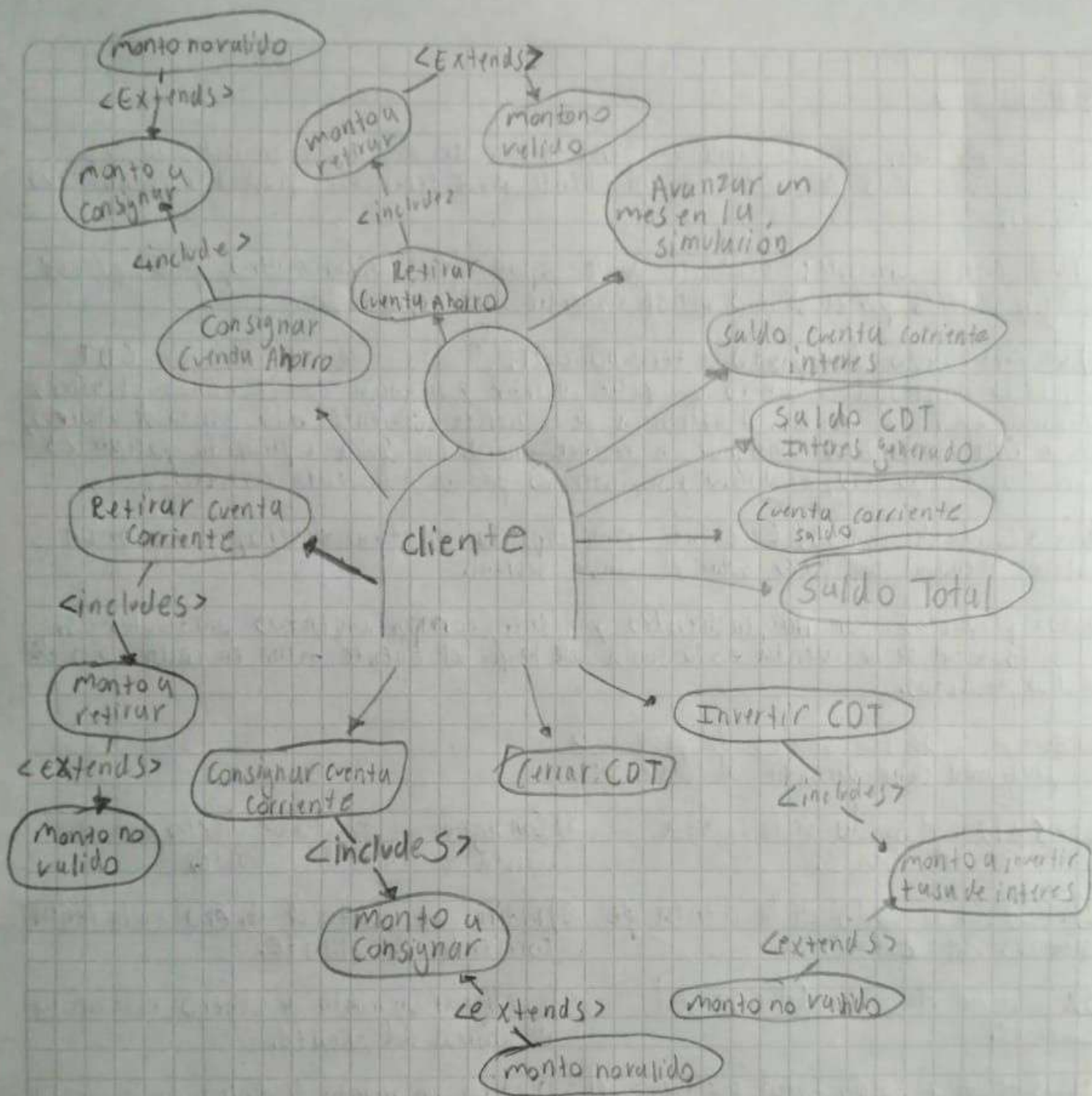
- 1) Una cuenta corriente: El cliente puede depositar o retirar dinero, pero no recibe ningún interés por el dinero que se encuentre allí depositado.
- 2) Un certificado de depósito a término CDT: Cuando el cliente abre un CDT, define la cantidad de dinero que quiere invertir y negocia con el banco el 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 cerrarlo, en cuyo caso, el dinero y sus intereses pasan a la cuenta corriente.
- 3) Una cuenta de ahorros: El cliente puede depositar o retirar dinero, y recibe un interés mensual del 0,6% sobre el saldo actual.

Estos productos son independientes y tienen comportamientos particulares. El saldo total de la cuenta es la suma de lo que el cliente tiene en cada uno de dichas productos.

Requerimientos funcionales del proyecto

El programa debe permitir al usuario:

- 1) Visualizar el saldo de la cuenta corriente del cliente.
- 2) Visualizar el saldo de la cuenta de ahorros del cliente.
- 3) Visualizar el saldo del CDT del cliente.
- 4) Visualizar el saldo total que tiene el cliente en los productos del banco.
- 5) Invertir un monto de dinero en un CDT.
- 6) Cerrar la inversión en CDT.
- 7) Consignar un monto de dinero en la cuenta corriente del cliente.
- 8) Retirar un monto de dinero en la cuenta corriente del cliente.
- 9) Consignar un monto de dinero en la cuenta de ahorros del cliente.
- 10) Retirar un monto de dinero en la cuenta de ahorros del cliente.
- 11) Avanzar un mes la simulación.



m) Modelo conceptual

① Cuenta bancaria

Atributos:

- Número de cuenta
- Titular
- Saldo
- Tipo de cuenta
- Banco
- Estado (Activa, inactiva, bloqueada)
- Fecha de apertura

Métodos:

- Depositar monto
- Retirar monto
- Transferir monto
- Consultar saldo
- Generar estado de cuenta
- Bloquear cuenta
- Desbloquear cuenta
- Cerrar cuenta

② Cuenta corriente

Atributos:

- Número de cuenta
- Titular
- Saldo
- Límite de sobregiro
- Banco
- Moneda

Métodos:

- Depositar monto
- Retirar monto
- Transferir monto
- Consultar saldo
- Generar estado de cuenta

③ Cuenta de Ahorro

Atributos:

- Número de cuenta
- Titular
- Saldo
- Tasa de interés
- Banco
- Fecha de apertura
- Límite de retiros
- Estado (Activa, inactiva, bloqueada)

Métodos:

- Depositar monto
- Retirar monto
- Transferir monto

- Consultar Saldo
- Generar estado de cuenta
- Aplicar intereses
- Verificar límites
- Bloquear cuenta
- Desbloquear cuenta
- Cerrar cuenta

④ Cuenta CDT

Atributos =

- Número de certificado
- Titular
- Monto invertido
- Tasa de interés
- Plazo (días, meses, años o hasta el vencimiento)
- Fecha de apertura
- Fecha de vencimiento
- Intereses generados
- Estado (Activo, Inactivo, bloqueado)
- Banco o Identidad financiera
- Renovación

Métodos =

- Consultar saldo
- Calcular intereses
- Generar estado CDT
- Renovar CDT
- Redimir CDT
- Cancelar anticipadamente
- Modificar renovación

⑤ Mes

Atributos =

- Nombre del mes
- Número del día del mes
- Cantidad de días
- Año asociado
- Días festivos
- Año bisiesto

Métodos

- Es año bisiesto
- Obtener cantidad de días
- Festivos
- Fin de semana
- Día laboral