

PROYECTO DE LOGICA COMPUTA- CIONAL

Grupo: #4

Horario: 7:00 - 9:00 a.m / Martes y Jueves

Integrantes: Andrea Cala
Valentina Gómez
Samuel Rivas

ACTIVIDAD

a) Resuma el ciclo de construcción de un programa.

El ciclo de vida de un programa es básicamente una receta que lleva el paso a paso para crearlo. Primero se analiza qué se necesita hacer, luego se diseña la solución, entonces palabras se planea como se va a programar. Despues se escribe el código en el programa, luego se prueba para asegurarse que no tenga errores, despues se implementa y para terminar se da la opción de dejarlo y con el tiempo implementan mas cosas.

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

Cuando tenemos un problema en la programación, no solo es agregar mas código, si no que primero debemos el problema y que es lo que necesita y sus restricciones, luego se identifican los datos de entrada y sus salidas esperadas. Tambien se analizan los pasos necesarios para llegar a su solución y que herramientas o métodos se necesitarían aplicar en ese caso.

c) Explique las etapas del proceso de solución de problemas.

Resolver un problema en programación es como armar un rompecabezas. Primero se entiende el problema, luego se planea una solución que se puede representar con diagramas de flujo o código, se escribe el código basado en una planificación y se prueba para asegurarse si funciona bien y si se cometen errores, se corrige y si todo está bien, se optimiza para ser más eficiente.

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

Cuando se termina un programa y se da a un cliente, se entregan varias cosas.

- ✓ El código fuente

- ✓ El programa ejecutable
- ✓ La documentación
- ✓ Manuales del usuario.
- ✓ Mantenimiento o soporte.

E. Elabore la Tarea N° 1 (Pag 5 del texto guía), con el objetivo de identificar los aspectos que forman parte de un problema.

Tarea 1: Identificar los aspectos que forman parte de un problema.

problema: Un banco quiere crear un programa para manejar sus cajeros automáticos. Dicho programa sólo debe permitir retirar dicho 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 imaginar la manera de completarlo.

Usuario: Los clientes del banco que usan los cajeros automáticos para retirar dinero y consultar saldo.

requerimiento funcional : - permite a los usuarios retirar dinero.
- permite consultar el saldo de la cuenta.

Mundo del problema : - El banco necesita un sistema seguro y confiable
- los cajeros deben estar conectados a una base de datos que tenga en banco.
- mantener reglas para evitar fraude, un ejemplo el límite de monto en un retiro.

requerimiento no funcional - Seguridad
- Una disponibilidad alta del sistema
- Para alguna duda o sugerencia tener una respuesta rápida

f. Elabore la Tarea No. 2 (pag 13), con el objetivo de identificar los requerimientos funcionales de un programa

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 fundamentales.

REQUERIMIENTO FUNCIONAL 1	NOMBRE	Crear cuenta bancaria
	RESUMEN	Permite registrar una nueva cuenta en el sistema
	ENTRADAS	Nombre del cliente, Tipo de cuenta (ahorros / corriente), Depósito inicial
	RESULTADO	Cuenta creada con un número de cuenta único
REQUERIMIENTO FUNCIONAL 2	NOMBRE	Realizar depósito
	RESUMEN	Permite al cliente depositar dinero en su cuenta
	ENTRADAS	Número de cuenta, monto que va a depositar
	RESULTADO	Saldo actualizado con el nuevo monto (dinero depositado)

REQUERIMIENTO FUNCIONAL 3	NOMBRE	Retirar dinero
	RESUMEN	Permite al cliente hacer retiros de su cuenta
	ENTRADA	Numero de cuenta, monto que va a retirar
	RESULTADO	Saldo actualizado o mensaje de fondos insuficientes

g. Elabore la Tarea No. 3 (pag 14), con el objetivo de identificar los requerimientos funcionales de un problema

REQUERIMIENTO FUNCIONAL 1	OBJETIVO:	Crear habilidad en la identificación y especificación de requerimientos funcionales. Para el caso de estudio 3, un programa para manejar un triángulo, identifique y especifique tres requerimientos funcionales
	NOMBRE	Determinar tipo de un triángulo
	RESUMEN	Identificar si el triángulo es equilatero, isosceles o escaleno
	ENTRADAS	Lado 1, Lado 2, Lado 3
	RESULTADO	Tipo de triángulo identificado

REQUERIMIENTO FUNCIONAL 2	NOMBRE	Calcular perimetro
	RESUMEN	Suma los tres lados del triangulo para hallar el valor del perimetro
	ENTRADAS	Lado 1, Lado 2, Lado 3
	RESULTADO	Valor del perimetro
REQUERIMIENTO FUNCIONAL 3	NOMBRE	Calcular area
	RESUMEN	Usa la formula de Base por Altura sobre 2
	ENTRADAS	Lado 1, Lado 2, Lado 3
	RESULTADO	Valor del area

h. Elabore la Tarea No. 9 (pag 17), con el objetivo de identificar las entidades del mundo del problema

OBJETIVO: Identificar las entidades del mundo para el caso de estudio 3. Un programa que maneje un triángulo.

TAREA 4:

Lea el anuncioado del caso y trate de quarse por los sustantivos para identificar las entidades del mundo del problema

NOMBRE	DESCRIPCION
ENTIDAD	Triángulo Representa un triángulo con sus lados, ángulos y el tipo. Es la entidad principal
ENTIDAD	Lado Representa la longitud de cada uno de ellos. Cada triángulo tiene 3 lados, cuyos valores determinan su tipo y características
ENTIDAD	Cálculo Permite realizar operaciones para determinar el tipo, calcular su perímetro y área.

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

Si no identificamos correctamente las entidades del mundo, el programa puede no representar de manera adecuada la realidad del problema. Esto puede llevar a errores en los cálculos, dificultades para estructurar el código y problemas al intentar ampliar el programa en el futuro. Además, podríamos omitir aspectos claves, lo que haría que nuestro sistema no pueda diferenciar correctamente tipos de triángulos o calcular su área de manera precisa.

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

Para decidir si un elemento es una entidad o solo una característica de ella, se debe analizar si tiene propiedades propias y si puede existir sin depender completamente de otra entidad. Si solo describe un aspecto de otra entidad sin tener una identidad propia, entonces es solo una característica. Por ejemplo, "Lado" es una entidad porque cada lado tiene un valor independiente y es esencial para definir un triángulo. En cambio, el "color" del triángulo no es una entidad, sino una característica que no afecta su estructura matemática.

i. Elabore la Tarea No. 5 (pag 20), con el objetivo de identificar las entidades de un caso de estudio

OBJETIVO: Identificar las características de las entidades del caso de estudio 2; un simulador bancario.

TAREA 5:

Para cada una de las 5 entidades identificadas en el caso de estudio del simulador bancario, identifique los atributos, sus valores, y escriba la clase en UML.

Clase: Cuenta Bancaria

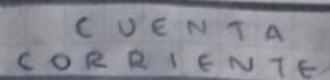
ATRIBUTO	VALORES POSIBLES	DIAGRAMA UML
Número de cuenta	Único, numérico no repetido	<p>CUENTA BANCARIA</p> <p>Número de cuenta</p>
Tipo de cuenta	Ahorros, Corriente	<p>Tipo de cuenta</p>
Saldo	Monto en la cuenta (≥ 0)	<p>Saldo</p>

Clase: Cuenta Corriente

ATRIBUTO VALORES POSIBLES

DIAGRAMA UML

Numero de cuenta	Unico, numerico, no repetido
Saldo	Monto en la cuenta (≥ 0)
Límite de crédito	Monto maximo permitido



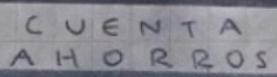
Numero de cuenta
Saldo
Límite de crédito

Clase: Cuenta Ahorros

ATRIBUTOS VALORES POSIBLES

DIAGRAMA UML

Numero de cuenta	Unico, numerico, no repetido
Saldo	Monto en la cuenta (≥ 0)
Tasas de interés	Porcentaje aplicado al saldo



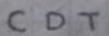
Numero de cuenta
Saldo
Tasa de interés

Clase: CDT

ATRIBUTO VALORES POSIBLES

DIAGRAMA UML

Numero de CDT	Unico, numerico, no repetido
Monto invertido	Monto total en el CDT
Tasa de interés	Porcentaje aplicado a la inversión



Numero de CDT
Monto invertido
Tasa de interés

Clase: Mes

ATRIBUTO

VALORES POSIBLES

DIAGRAMA UML

Número del mes

Entero entre 1 y 12

Nombre del mes

Enero, Febrero..., Diciembre

Días del mes

Cantidad de días
Según el mes

MES

Número del mes
Nombre del mes
Días del mes

II) Elabore la Tarea N°6 (pag. 23), con el objetivo de reflexionar sobre el nivel de presición de un algoritmo.

Tarea 6

Objetivo:

Reflexionar sobre el nivel de presición que debe ser usado un algoritmo para evitar ambigüedades.

Suponga que usted es la persona que va utilizar el algoritmo anterior, para moverse en el metro de París. Identifique qué problemas podría tener con las instrucciones anteriores. Piense por ejemplo si están completas.

¿Se prestan para que ser interpreten de maneras distintas? Estamos suponiendo que quien lo usa "usa su sentido común", o cualquier que lo use va a resolver el problema de la misma manera?

Solución

Problemas

Ya que yo no he estado en la Ciudad de París, no sé el idioma Francés ni manejar la moneda de allí se me podrían presentar varios problemas como:

1. En el primer paso indica que compre el tiquete en los puntos de venta, pero no me explica si en las máquinas automáticas o en las taquillas y ya que no sé francés tendría problemas para comunicarme con los vendedores o interactuar con las máquinas, no es preciso diciendo si el pago es en efectivo o se tiene que usar alguna tarjeta.
2. Otro problema que se presentaría es que no explica como usar el mapa del metro, porque podría ser confuso o complicado de entender si se cruzan varias líneas entre ellas y comprenderlo por su idioma.

3. Otro problema sería que en el punto 4 dice que verifique si hay una linea que pase por la estación del destino, pero no me da información de que hacer en caso que no halla una linea directa hasta el destino y toque hacer transbordos o si también hay estaciones cerradas o metros que no funcionen a ciertas horas.

Interpretación:

No se presta para interpretarse de otra forma ya que da el paso a paso con el fin de llegar al destino por medio del metro, pero si puede generar confusiones por los problemas mencionados anteriormente.

Sentido Común:

Si, el algoritmo asume que la persona que lo lee tiene sentido común y experiencia, ya que piensa que la persona sabe comprar el ticket en el metro parisino sin que se lo expliquen, también piensa que la persona sabe leer el mapa del metro y entiende sus conexiones, y sabe que hacer en caso en que hallan estaciones cerradas o retrasos.

Entonces cualquier persona que lo use no lo resolverá de la misma manera porque se puede encontraran experimentados con el uso del metro, del idioma y la moneda de París y personas con las cuales no sepan ninguno de estos y toque ser más específicos con el algoritmo.

- K) Estudia los siguientes aspectos del ejemplo seleccionado: Enunciado, Requerimientos funcionales (casos de uso) y el Modelo (clases del proyecto). A continuación, redacta el enunciado del problema y el nombre cada uno de los requerimientos funcionales del proyecto.

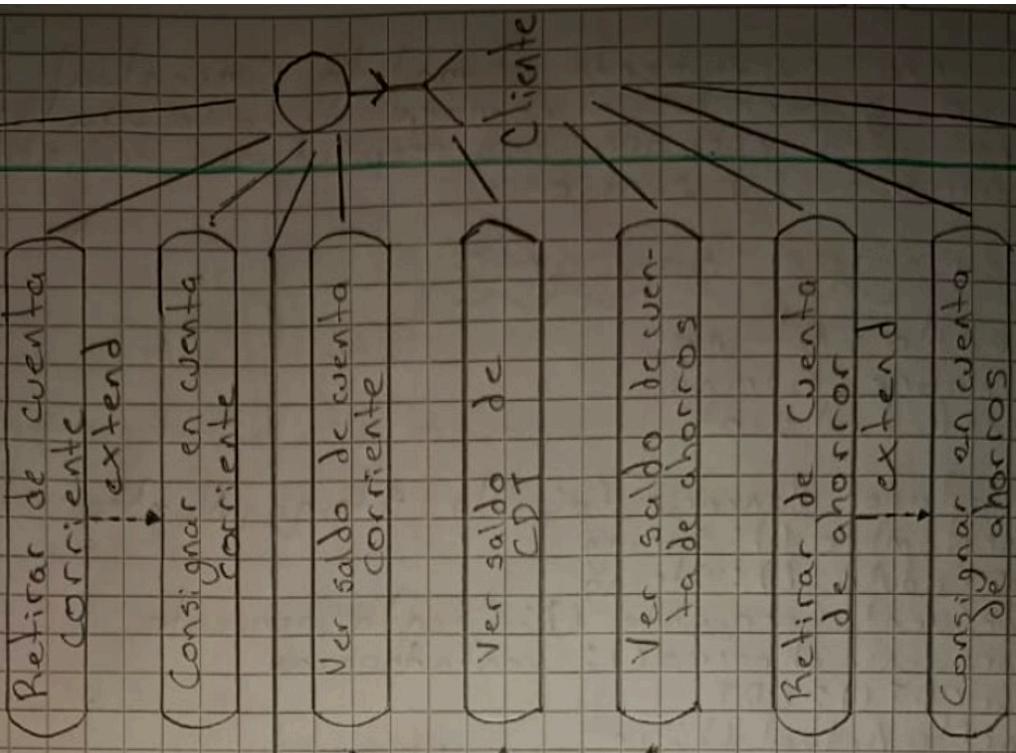
Enunciado Del Problema:

Se necesita desarrollar un Simulador Bancario que permita gestionar los productos financieros de un cliente: cuenta corriente, de ahorros y CDT. Sabiendo que cada producto tiene un comportamiento específico y el cliente pueda realizar depósitos, retiros, inversiones y simulaciones de intereses.

Requerimientos Funcionales:

1. Ver el saldo de la cuenta corriente, de ahorros y CDT
 2. Ver el saldo total de todos los productos financieros del cliente.
 3. Poder invertir dinero en un CDT, especificando el monto y la tasa de interés
 4. Poder cerrar un CDT y así transferir el monto y los intereses a la cuenta corriente
 5. Poder consignar y retirar dinero de la cuenta corriente.
 6. Poder consignar y retirar dinero de la cuenta de ahorros.
 7. Poder avanzar la simulación en un mes, aplicando sus intereses.
- 1) Dibuja el respectivo diagrama de casos de uso del ejemplo elegido (consulta "requerimientos funcionales" en "Archivo").

Simulador Bancario



M) Observa nuevamente el modelo conceptual del caso y escribe el nombre de cada una las clases identificando sus respectivas variables (atributos) y funciones:

Simulador Bancario

- Cedula: String
- Nombre: String
- mesActual: int
- + Simulador Bancario (pCedula: String, pNombre: String)
- + darNombre (): String
- + darCedula (): String
- + darCuentaCorriente (): CuentaCorriente
- + darCuentaAhorros (): CuentaAhorros
- + darCDT (): CDT
- + darMesActual (): int
- + calcularSaldoTotal (): double
- + invertirCDT (pMonto: double, pInteresMensual: double): void
- + consignarCuentaCorriente (pMonto: double): void
- + consignarCuentaAhorros (pMonto: double): void
- + retirarCuentaAhorros (pMonto: double): void
- + avanzarMesSimulacion (): void
- + cerrarCDT (): void
- + metodo 1 (): String
- + metodo 2 (): String

Inversion - CDT

- valorInvertido: double
- interesMensual: double
- mesApertura: int
- + CDT ()
- + darInteresMensual (): double
- + invertir (pMontоЦnvertido: double, pInteresMensual: double, pMes: int): void
- + calcularValorPresente (pMesActual: int): double
- + cerrar (pMesActual: int): double

Corriente - Cuenta Corriente

- Saldo: double

+ CuentaCorriente ()

+ darSaldo (): double

+ consignarMonto (pMonto: double): void

+ retirarMonto (pMonto: double): void

Ahorros - Cuenta Ahorros

- Saldo: double

- interesMensual: double

+ CuentaAhorros ()

+ darSaldo (): double

+ darInteresMensual (): double

+ consignarMonto (pMonto: double): void

+ retirarMonto (pMonto: double): void

+ actualizarSaldoPorPasosMeses (): void

n) Debes plantear 2 ideas de proyecto (Problemas solubles y algorítmicos:

Idea 1:

App Para Gestionar Tareas y Recordatorios

A veces olvidamos cosas importantes, así que esta app nos ayudara a organizar las tareas y a recordarlas

Requerimiento funcional 1:

Nombre: Agregar tarea

Resumen: Permitir al usuario agregar una nueva tarea con un título y fecha límite

Entradas: Nombre de la tarea, fecha límite

Resultado: La tarea se guarda en la lista de pendientes.

* Requerimiento Funcional 2:

Nombre: Marcar una tarea como completada

Resumen: El usuario marca la tarea como echada para que desaparezca de la lista de pendientes.

Entradas: Selección de una tarea

Resultado: La tarea se mueve a la lista de completadas.

* Requerimiento Funcional 3:

Nombre: Configurar recordatorios

Resumen: Se puede establecer una notificación para recordar tareas importantes.

Entradas: Selección de tarea, hora del recordatorio o fecha

Resultado: La app envía un mensaje a la hora y fecha programada

* Requerimiento Funcional 4:

Nombre: Ver todas las tareas

Resumen: Muestra todas las tareas organizadas en "Pendientes" y "Completadas".

Entradas: Ninguna

Resultado: Se muestra la lista de tareas

Idea 2:

(Simulador De Presupuesto Personal)

Muchas veces gastamos dinero sin darnos cuenta, así que esta app ayuda a llevar un control del dinero.

* Requerimiento Funcional 1:

Nombre: Registrar un ingreso

Resumen: Permite al usuario ingresar dinero recibido, como salario o regalos.

Entradas: Monto del ingreso de saldo, ventas, etc

Resultado: El saldo total aumenta

* Requerimiento Funcional 2:

Nombre: Registrar un gasto

Resumen: Se pueden agregar gastos para ver en qué se está usando el dinero

Entradas: Monto del gasto de comida, transporte, ocio, etc.

Resultado: El saldo total disminuye

* Requerimiento Funcional 3:

Nombre: Ver el resumen financiero

Resumen: Muestra un resumen con el dinero disponible y los gastos

Resultado: Muestra un gráfico con ingresos, gastos y saldo total

DD | MM | AA

*Bequerimiento Fraccional 4:

Nombre: Establecer un presupuesto mensual

Resumen: El usuario puede fijar un límite de gasto para evitar quedar sin dinero

Entradas: monto maximo de gasto mensual

Resultado: La app avisa cuando el usuario se acerque al límite.