

Proyecto Lógica Computacional

Contexto:

-La especificación de un problema:

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

Planeación: Primero, uno se sienta a pensar en qué quiere hacer. Se define el problema y se arman las ideas. Es como hacer un plan antes de construir una casa.

Análisis: Luego, se analizan los requisitos. Se habla con los usuarios para entender qué necesitan y se hacen listas de lo que debe tener el programa.

Diseño: Aquí se empieza a dibujar cómo va a lucir el programa. Se hacen diagramas y se decide la estructura, como si se estuviera diseñando los planos de la casa.

Implementación: En esta etapa, se pone todo en código. Los programadores escriben el software, como si estuvieran armando los ladrillos de la casa.

Pruebas: Despues, se prueba el programa a fondo. Se busca errores y se asegura que todo funcione bien, como revisar que no haya goteras.

Despliegue: Cuando todo está listo, se lanza el programa. Se le entrega a los usuarios.

Mantenimiento: Finalmente, hay que estar pendiente de cualquier problema y hacer mejoras.

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

Entender el problema: Hay que captar bien qué es lo que está fallando. Uno tiene que meterle cabeza al problema, como cuando un amigo tiene un problema y te lo cuenta y tú intentas entenderlo a fondo.

Reunir información: Se necesita recopilar datos. Se encuestan a los usuarios para saber cómo utilizan el programa y qué fallos han encontrado. Es como preguntarles a varios amigos qué les gusta y qué no de una película.

Definir los requisitos: Aquí hay que listar qué debe hacer el programa para solucionar el problema. Se trata de hacer una lista clara de las funciones necesarias, como cuando le dices a un desarrollador qué quieres que tenga tu app.

Identificar restricciones: Es importante saber qué limitaciones hay, como el presupuesto, el tiempo de entrega o las herramientas disponibles. Como cuando planificas un viaje y sabes que no puedes gastar más de cierto monto.

Analizar alternativas: Se miran diferentes maneras de resolver el problema. Se evalúan varias soluciones posibles y se analizan sus ventajas y desventajas, como decidir entre varias rutas para llegar a un lugar.

Establecer prioridades: Finalmente, se debe determinar qué es lo más urgente y qué puede esperar. Así se decide cuáles funcionalidades son más importantes para incluir en la primera versión del programa, como elegir qué platos son imprescindibles en una cena.

C) EXPLIQUE LAS ETAPAS DEL PROCESO DE SOLUCIÓN DE PROBLEMAS.

EL PROCESO DE SOLUCIÓN DE PROBLEMAS EN LÓGICA COMPUTACIONAL INCLUYE VARIAS ETAPAS. PRIMERO SE DEBE ENTENDER Y DEFINIR CLARAMENTE EL PROBLEMA Y QUÉ SE NECESA. LUEGO, SE ANALIZA EL PROBLEMA DIVIDIÉNDOLO EN PARTES MÁS PEQUEÑOS Y REVISANDO EJEMPLOS ESPECÍFICOS. DESPUÉS, SE DISEÑA UNA SOLUCIÓN CREANDO UN PLAN O ALGORITMO Y ELIGIENDO CÓMO ORGANIZAR LA INFORMACIÓN. LUEGO, SE IMPLEMENTA ESA SOLUCIÓN ESCRIBIENDO EL CÓDIGO Y HACIENDO PRUEBAS INICIALES. A CONTINUACIÓN, SE VERIFICA QUE TODO FUNCIONE CORRECTAMENTE MEDIANTE PRUEBAS MÁS EXHAUSTIVAS Y SE CORRIGEN ERRORES. TAMBién ES IMPORTANTE LO QUE SE HIZO PARA FUTURAS REFERENCIAS. AL FINAL, SE REVISA Y EVALÚA LA SOLUCIÓN CON EL FIN DE MEJORARLA.

D) CUÁLES SON LOS ELEMENTOS QUE SE DEBEN ENTREGAR A UN CLIENTE?

AL ENTREGAR UN PROYECTO, SE DEBEN INCLUIR VARIOS ELEMENTOS CLAVES: UN INFORME QUE EXPLIQUE EL TRABAJO REALIZADO Y CÓMO SE RESOLVIÓ EL PROBLEMA, DOCUMENTACIÓN CLARA QUE DETALLA CÓMO USAR EL PRODUCTO O SERVICIO, ALQUILER MATERIAL DE SOPORTE COMO MANUALES O GUIAS, Y ACCESO A LA SOLUCIÓN FINAL, TAMBIÉN ES ÚTIL PROPORCIONAR INFORMACIÓN SOBRE EL MANTENIMIENTO Y SOPORTE FUTURO, ASI COMO UN RESUMEN DE LOS RESULTADOS OBTENIDOS Y LAS RECOMENDACIONES DE USO.

E. 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.

CUENTAS	BANCO	LA ENTIDAD QUE SOLICITA EL DESARROLLO DEL SOFTWARE
USUARIO	CLIENTE DEL BANCO	LA PERSONA QUE INTERACTUA DE MANERA DIRECTA CON EL CAJERO
REQUERIMIENTO FUNCIONAL	RETIRO DE EFECTIVO	EL SISTEMA DEBE PERMITIR AL USUARIO RETIRAR UNA CANTIDAD ESPECIFICA DE DINERO DE SU CUENTA, SIEMPRE Y CUANDO HAYA FONDOS SUFICIENTES.
	CONSULTA DE SALDO	EL SISTEMA DEBE MOSTRAR AL USUARIO EL SALDO DISPONIBLE EN SU CUENTA
MUNDO DEL PROBLEMA	CAJEROS AUTOMÁTICOS	DISPOSITIVOS FÍSICOS DISTRIBUIDOS GEGRÁFICAMENTE QUE INTERACTÚAN CON EL USUARIO Y EL SISTEMA CENTRAL DEL BANCO
	CUENTAS BANCARIAS	ENTIDADES DIGITALES QUE ALMALENAN INFORMACIÓN SOBRE LOS FONDOS DEL CLIENTE
	SISTEMA CENTRAL DEL BANCO	BASE DE DATOS QUE CONTIENE LA INFORMACIÓN DE TODAS LAS CUENTAS Y TRANSACCIONES
REQUERIMIENTO NO FUNCIONAL	SEGURIDAD	EL SISTEMA DEBE GARANTIZAR LA CONFIDENCIALIDAD DE LA INFORMACIÓN DEL CLIENTE Y PROTEGERLA CONTRA FRAUDES
	DISPONIBILIDAD	EL SISTEMA DEBE ESTAR DISPONIBLE LAS 24H DEL DIA, Y LOS 7 DIAS DE LA SEMANA
DESEMPEÑO		LAS TRANSACCIONES DEBEN REALIZARSE DE MANERA RÁPIDA Y EFICIENTE
ESCALABILIDAD		EL SISTEMA DEBE PODER MANEJAR UN AUMENTO EN EL NÚMERO DE USUARIOS Y TRANSACCIONES
MANTENIBILIDAD		EL SOFTWARE DEBE SER FÁCIL DE ACTUALIZAR Y DE MODIFICAR

ADICIONAL: EL SISTEMA NECESITARÁ UN MECANISMO SEGURO PARA IDENTIFICAR AL USUARIO. (POR EJEMPLO, TARJETA O PIN).

- SE TENDRÁN QUE ESTABLECER LÍMITES DIARIOS O MENSUALES PARA LAS TRANSACCIONES, RETROS Y CONSIGNACIONES.

- EL SISTEMA DEL CAJERO DEBERÁ ESTAR INTEGRADO CON OTROS SISTEMAS DEL BANCO, COMO EL SISTEMA DE CONTABILIDAD
- ES IMPORTANTE CONSIDERAR LA DISTRIBUCIÓN DE LOS CAJEROS, BUSCAR BUENOS PUNTOS GEOGRÁFICOS PARA GARANTIZAR LA COMUNICACIÓN EFICIENTE CON EL SISTEMA CENTRAL DEL BANCO

F) UN SIMULACRO BANCARIO, IDENTIFIQUE Y ESPECIFIQUE TRES REQUERIMIENTOS FUNCIONALES

REQUERIMIENTO FUNCIONAL	ENTRADAS	RESULTADO
REALIZAR DEPÓSITO	EL USUARIO DEBE PODER INGRESAR UNA CANTIDAD DE DINERO A SU CUENTA	NUMERO DE CUENTA, CANTIDAD A DEPOSITAR
REALIZAR TRANSFERENCIA	EL USUARIO DEBE PODER TRANSFERIR UNA CANTIDAD DE DINERO A OTRA CUENTA SIN PROBLEMA	NÚMERO DE CUENTA ORIGEN, NÚMERO CUENTA DESTINO, CANTIDAD A TRANSFERIR
CONSULTAR SALDO	EL USUARIO DEBE PODER CONOCER EL SALDO DISPONIBLE EN SU CUENTA	NUMERO DE CUENTA

G) UN PROGRAMA PARA MANEJAR UN TRIÁNGULO, IDENTIFIQUE Y ESPECIFIQUE 3 REQUERIMIENTOS FUNCIONALES.

REQUERIMIENTO FUNCIONAL	ENTRADAS	RESULTADO
CÁLCULO DEL ÁREA	TRES LADOS O BASE Y ALTURA	VALOR NÚMÉRICO DEL ÁREA
CÁLCULO DEL PERÍMETRO	TRES LADOS	VALOR NÚMÉRICO DEL PERÍMETRO
CLASIFICACIÓN DE TRIÁNGULOS	TRES LADOS	CADENA DE TEXTO INDICANDO EL TIPO DE TRIÁNGULO

H) UN PROGRAMA QUE MANEJE UN TRIÁNGULO

ENTIDAD	
TRIÁNGULO	LA FIGURA GEOMÉTRICA DE TRES LADOS QUE ES EL OBJETO PRINCIPAL DE NUESTRO PROGRAMA
LADO	CADA UNA DE LAS LINEAS QUE CONFORMAN EL TRIÁNGULO
ÁNGULO	LA MEDIDA DE LA ABERTURA ENTRE DOS LADOS DE UN TRIÁNGULO
PUNTO	LA UBICACIÓN EXACTA DE UN VÉRTICE DEL TRIÁNGULO EN UN SISTEMA DE COORDENADAS
ÁREA	LA MEDIDA DE LA SUPERFICIE ENCERRADA POR EL TRIÁNGULO
PERÍMETRO	LA SUMA DE LAS LONGITUDES DE LOS LADOS DEL TRIÁNGULO
TIPO DE TRIÁNGULO	LA CLASIFICACIÓN DEL TRIÁNGULO SEGÚN LA LONGITUD DE SUS LADOS (EQUILÁTERO, ISÓSCELES, ESCALENO).

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

Clase: CuentaBancaria

Atributo	Valores Posibles	Diagrama UML
Número de Cuenta	Un número único que identifica la cuenta.	Cuenta Bancaria
Tipo de cuenta	Corriente, Ahorros	-númeroCuenta: int -tipoCuenta: string
Saldo	Un número decimal que representa el dinero disponible	-Saldo: float -fechaApertura: Date -cliente: Cliente
Fecha de apertura	Una fecha (dia,mes,año)	
Cliente asociado	El objeto Cliente al que pertenece la cuenta.	

Clase: CuentaCorriente

Atributo	Valores posibles	Diagrama UML
descubierto Permitido	Sí o No (booleano)	CuentaCorriente
tasaInteres	Un número decimal	-descubiertoPermitido: boolean -tasaInteres: float -númeroCheques: int
númeroCheques	Un número entero	
(Hereda de)	CuentaBancaria	

Clase: CuentaAhorros

Atributo	Valores posibles	Diagrama UML
tasaInteres	Un número decimal	CuentaAhorros
fechaVencimiento	Una fecha (dia,mes,año)	-tasaInteres: float -fechaVencimiento: Date
(Hereda de)	CuentaBancaria	

J. Elabore la Tarea No. 6 (pág. 23), con el objetivo de reflexionar sobre el nivel de precisión de un algoritmo.

Los problemas que se encontraron en el algoritmo son: Primero, no dice ni cómo comprar el tridente, si es sencillo o si toca recargar una tarjeta. También asume que uno sabe todo el mapa del metro de memoria, como si fuera un parisiense de toda la vida. Para mejorar su eficacia, es necesario detallar cada paso, ser más claros y más específicos, como si le estuvieras explicando a tu abuelita cómo llegar a la plaza de mercado, utilizando un lenguaje más claro y sencillo; se podría incluir elementos visuales, como mapas. Para mejorar la experiencia del usuario se podrían considerar opciones como crear una versión interactiva del algoritmo, colocandole datos en tiempo real sobre el servicio o permitir la personalización según las necesidades de cada usuario.

K) ESTUDIA LOS SIGUIENTES ASPECTOS DEL EJEMPLO SELECCIONADO:
ENUNCIADO, REQUERIMIENTOS FUNCIONALES (CASOS DE USO) Y MODELO
(CLASE DE PROYECTO)

TRIANGULO

ENUNCIADO, SE QUIERE CONSTRUIR UN PROGRAMA QUE PERMITA DIBUJAR UN TRIÁNGULO, ESTA FIGURA GEOMÉTRICA ESTÁ DEFINIDA POR TRES PUNTOS, CADA UNO DE LOS CUALES TIENE DOS COORDENADAS X, Y. ESTE TRIÁNGULO TIENE, ADEMÁS UN COLOR PARA LAS LINEAS Y UN COLOR PARA EL RELLENO.

UN COLOR, POR SU PARTE, ESTÁ DEFINIDO POR TRES VALORES NUMÉRICOS ENTRE 0 Y 25 (ESTÁNDAR RGB POR RED-GREEN-BLUE). EL PRIMER VALOR NUMÉRICO DEFINE LA INTENSIDAD EN ROJO, EL SEGUNDO EN VERDE Y EL TERCERO EN AZUL.

EL PROGRAMA DEBE PERMITIR

- DIBUJAR EL TRIÁNGULO EN LA INTERFAZ
- CAMBIAR LOS PUNTOS DEL TRIÁNGULO
- CAMBIAR EL COLOR DE RELLENO DEL TRIÁNGULO
- CAMBIAR EL COLOR DE LAS LINEAS

- MOSTRAR PERIMETRO
- EL ÁREA DEL TRIÁNGULO
- ALTURA DEL TRIÁNGULO

NOMBRE DE REQUERIMIENTOS

DIBUJAR EL TRIANGULO EN LA INTERFAZ

CAMBIAR LOS PUNTOS DEL TRIÁNGULO

CAMBIAR COLOR DE RELLENO DEL TRIÁNGULO

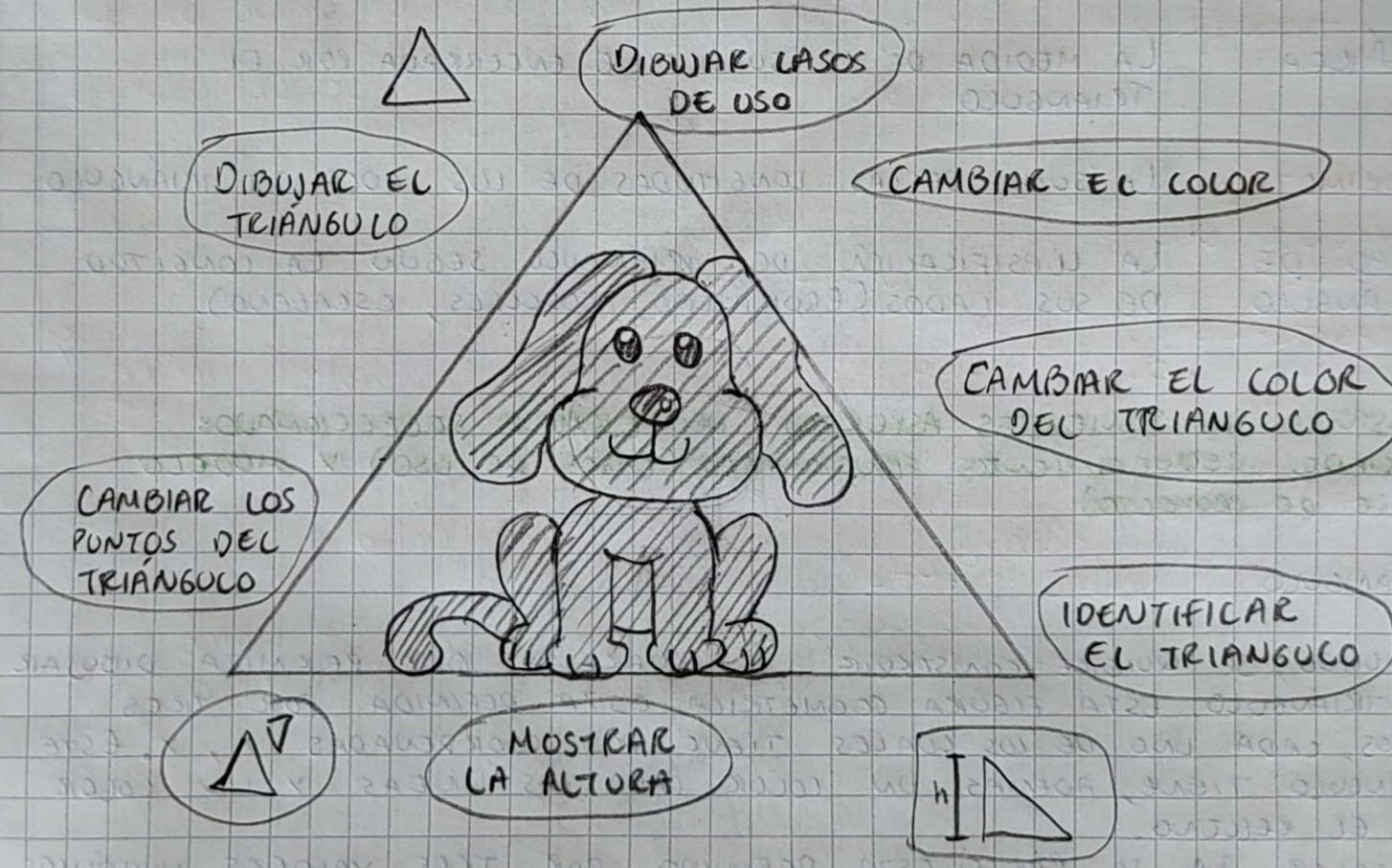
CAMBIAR COLOR DE LAS LINEAS DEL TRIÁNGULO

MOSTRAR EL ÁREA DEL TRIÁNGULO

MOSTRAR LA ALTURA DEL TRIÁNGULO

MOSTRAR EL PERIMETRO DEL TRIÁNGULO

(1)



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

Diseño de un Simulador Bancario

Clases Principales:

- Cliente: Representa al usuario con sus datos personales.
- Cuenta: Clase base para todas las tipos de cuentas.
- CuentaCorriente: Hereda de Cuenta, sin intereses.
- CuentaAhorros: Hereda de Cuenta, con intereses.
- CDT: Hereda de Cuenta, con plazo fijo y tasa de interés.

Relaciones:

Un cliente puede tener múltiples cuentas.

Funcionalidades:

Depositos: Aumentar el saldo de una cuenta.

Retiros: Disminuir el saldo de una cuenta.

Cálculo de intereses: Calcular intereses para cuentas de ahorro y CDT.

Renovación de CDT: Extender el plazo de un CDT.

n. Plantear 2 ideas de proyecto (problemas solubles y algorítmicos).

Idea 1:

	Nombre	Un sistema de recomendación de películas
Requerimiento Funcional 1	Resumen	Un programa que sugiere películas a un usuario basado en sus gustos y preferencias
	Entradas	Datos de películas (título, género, actores, director, sinopsis), perfil del usuario
	Resultado	Una lista de películas recomendadas personalizadas para el usuario.

Idea 2:

	Resumen	Una aplicación móvil o web que permita a los entrenadores de futsal registrar y analizar las estadísticas de los partidos de su equipo
Requerimiento Funcional 2.	Nombre	Futsal Pro
	Entradas	Datos de los jugadores, goles anotados, tarjetas amarillas y rojas, sustituciones, posesión del balón, tiros a puerta, etc.
	Resultado	Generación de estadísticas detalladas por jugador y por equipo (goles por partido promedio, de posesión, porcentaje de acierto en los tiros, etc.), visualización de gráficos y tablas comparativas.