

Proyecto 11. Cine.

Actividades preliminares.

Contexto

La especificación de un problema:

A. Resume el ciclo de vida de construcción de un problema.

El ciclo de vida de construcción de un problema empieza con el cliente y un problema que requiere uso del computador para solucionarlo, el programador sigue una secuencia de etapas donde se realizan distintas tareas con objetivos específicos, después de todo el proceso se obtiene la solución al problema, el programa hace parte de la solución, y finalmente el programa se instala en el computador del cliente.

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

- Requerimientos funcionales

Es una operación que el programa que se va a construir debe proveer al usuario, y que está directamente relacionada con el problema que se quiere resolver.

- Mundo del problema

El objetivo es entender el mundo en el que ocurre el problema y recopila toda la información necesaria para que el programador pueda escribir el programa.

• Requerimiento no funcional

Están muchas veces relacionadas con restricciones sobre la tecnología que se debe usar, el volumen de los datos que se debe manejar o la cantidad de usuarios.

C. Explique las etapas del proceso de solución de problemas.

• Análisis del problema

El objetivo de esta etapa es entender y especificar el problema que se quiere resolver.

• Diseño de la solución

El objetivo es detallar, usando algún lenguaje (planos, dibujos, ecuaciones, diagramas, texto, etc.), las características que tendrá la solución antes de ser construida.

• Construcción de la solución

Tiene como objetivo implementar el programa a partir del diseño y probar su correcto funcionamiento.

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

- El diseño
- El programa
- Las pruebas de corrección del programa
- Manual de usuario.

Experiencia

E. Elabore la tarea # 1.

Un banco quiere crear un programa para manejar sus cajeros automáticos. Dicho programa solo debe permitir retirar dinero y consultar saldo de una cuenta.

Ciente

El banco

Usuario

Los clientes del banco

Requerimiento funcional

R1: El programa debe permitir retirar dinero de su cuenta bancaria a los clientes del banco. (se usa tarjeta y clave).

R2: El programa debe permitir consultar el saldo de su cuenta bancaria a los clientes del banco. (se usa tarjeta y clave).

Mundo del problema.

El banco debe tener una base de datos, la cual debe tener los datos de cada persona, como su saldo, límites de retiro diarios, etc. Esta base de datos se tiene que conectar con los cajeros y los cajeros siempre deben estar conectados a la red para enviar y recibir información de transacciones y movimientos en tiempo real.

Requerimiento no funcional.

Debe haber una seguridad que se proteja con una clave y que este todo cifrado, todo debe ser en tiempo real, debe estar disponible siempre, y la red debe ser lo más estable posible.

F. Elabore la tarea # 2

Un simulador bancario, identifique y especifique tres requerimientos funcionales.

RF1.	Nombre	Realizar un depósito	Identificador R1, "realizar" para describir la operación que se va a hacer.
	Resumen	Permite al usuario realizar un depósito en el simulador.	
	Entradas	Autenticarse en la app, número de cuenta al que se va a depositar, monto a depositar.	
	Resultado	El depósito se realizará, y el nuevo saldo será mostrado.	
RF2.	Nombre	Pedir un préstamo	
	Resumen	Permite al usuario entrar a la opción "pedir préstamo" dentro de la app.	
	Entradas	Autenticarse en la app, Meses que va a estar activo el préstamo, monto solicitado a préstamo, número de cuenta al que se va a depositar el préstamo.	
	Resultado	El préstamo se realizará si el cliente es aprobado para el crédito, se actualizará el dinero y el nuevo saldo será mostrado, si no es aprobado el crédito, el saldo será el mismo de antes.	
RF3.	Nombre	Cancelar tarjeta	
	Resumen	Permite al usuario cancelar tarjeta de débito o crédito por robo, perdida u otra razón, esto dentro de la app.	
	Entradas	Autenticarse en la app, seleccionar tarjeta a cancelar, seleccionar la razón de la cancelación.	
	Resultado	La tarjeta de crédito o débito será cancelada, a petición del usuario.	

G. Elabore la tarea # 3.

Un programa para manejar un triángulo

RF1.	Nombre	Mover el triángulo en el eje X, Y
	Resumen	Permitir al usuario mover el triángulo en un plano X, Y.
	Entradas	Coordenadas en el eje X, Coordenadas en el eje Y.
	Resultado	El triángulo será ubicado en el plano, según las coordenadas que haya seleccionado el usuario.
RF2	Nombre	Cambiar el tamaño del triángulo
	Resumen	Permitir al usuario cambiar el tamaño del triángulo a su gusto.
	Entradas	Tamaño base - Altura, tamaño pixeles.
	Resultado	El triángulo crecerá como el usuario haya elegido.
RF3.	Nombre	Rotar el triángulo
	Resumen	Permitir al usuario rotar el triángulo los grados que desee.
	Entradas	Grados de rotación que va a tener el triángulo.
	Resultado	El triángulo rotará, según como el usuario haya decidido.

H. Elabore la tarea # 4.

Un programa que maneje un triángulo.

	Nombre	Descripción
Entidad	Eje x, y triángulo	Esta representa las coordenadas que el triángulo tiene en un plano x, y .
Entidad	Rotación	Esta representa el ángulo de rotación del triángulo.
Entidad	Tamaño	Esta es la representación del factor de cambio de tamaño del triángulo.

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

El modelo para resolver el problema estará incompleto, lo que significa que el programa no será totalmente la solución al problema y afectará a sus funciones principales.

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

Se puede decidir partiendo de la idea de que una entidad tiene características propias, pudiéndose así describirse independientemente de otras entidades.

I. Elabore la tarea # 5

Identifique los atributos, valores posibles y escriba la clase en UML.

Clase: Cuenta bancaria

Diagrama UML

Atributo	valores posibles
Titularidad	Cadena de caracteres
Intereses	valores enteros entre 1 y 99
Costos	valores enteros positivos

Cuenta bancaria
Titularidad
Intereses
Costos

Clase: Cuenta corriente

Diagrama UML

Atributo	valores posibles
Intereses	valores enteros entre 1 y 99
Comisiones	valores enteros positivos
Tarjeta debito	valores enteros de 16 digitos

Cuenta corriente
Intereses
Comisiones
Tarjeta debito

Clase: Cuenta ahorros

Diagrama UML

Atributo	valores posibles
Saldo	valores enteros
Tasa interes	valores entre 1 y 99.
Número cuenta	valores enteros positivos

Cuenta ahorros
Saldo
Tasa interes
Número cuenta

Clase: CDT

Diagrama UML

Atributo	valores posibles
Monto	valores enteros positivos
Plazo	valores enteros positivos
Interés	valores entre 1 y 99.

CDT
Monto
Plazo
Intereses

Clase: Mes

Atributo	valores posibles	Diagrama UML
Horas	Valores ^{enteros} entre 1 y 24	^{Mes} Horas
Días	Valores enteros entre 1 y 31	días
Semanas	Valores enteros entre 1 y 4.	Semanas

J. Elabore la tarea # 6

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

R: / No se especifica cual boleto comprar, porque dudamos que solo exista una y sus características.

Como habla del mapa, se asume que el usuario tiene uno, pero ¿y si lo pierde, ¿dónde se puede guiar?

Si el usuario no se logra ubicar, ¿hay algún ayudante en la estación del metro?

En cierta parte las instrucciones se confunden en parte, pero fácilmente se pueden comprender con ayuda de alguien que ya sepa el proceso en el metro.

Accion

Enunciado

Se quiere construir un Programa que permita dibujar un triangulo

Esta figura geometrica esta definida por tres puntos, cada uno de los cuales tiene dos coordenadas X, Y . Este triangulo tiene, ademas, un color para las lineas y un color para el relleno. Un color, por su parte, esta definido por tres valores numericos entre 0 y 255 (estandar RGB por Red - Green - Blue). El primer valor numerico define la intensidad en rojo, el segundo en verde y el tercero el azul.

El Programa debe Permitir:

1. Dibujar el triangulo en la interfaz.
2. Cambiar los puntos del triangulo
3. Cambiar el color de relleno del triangulo
4. Cambiar el color de las lineas (bordes) del triangulo.
5. Mostrar el Perimetro del triangulo.
6. Mostrar el area del triangulo.
7. Mostrar la altura del triangulo.

Requerimientos Funcionales

Nombre = R1 - Dibujar el triangulo en la interfaz

Nombre = R2 - Cambiar los puntos del triangulo

Nombre = R3 - Cambiar color relleno del triangulo

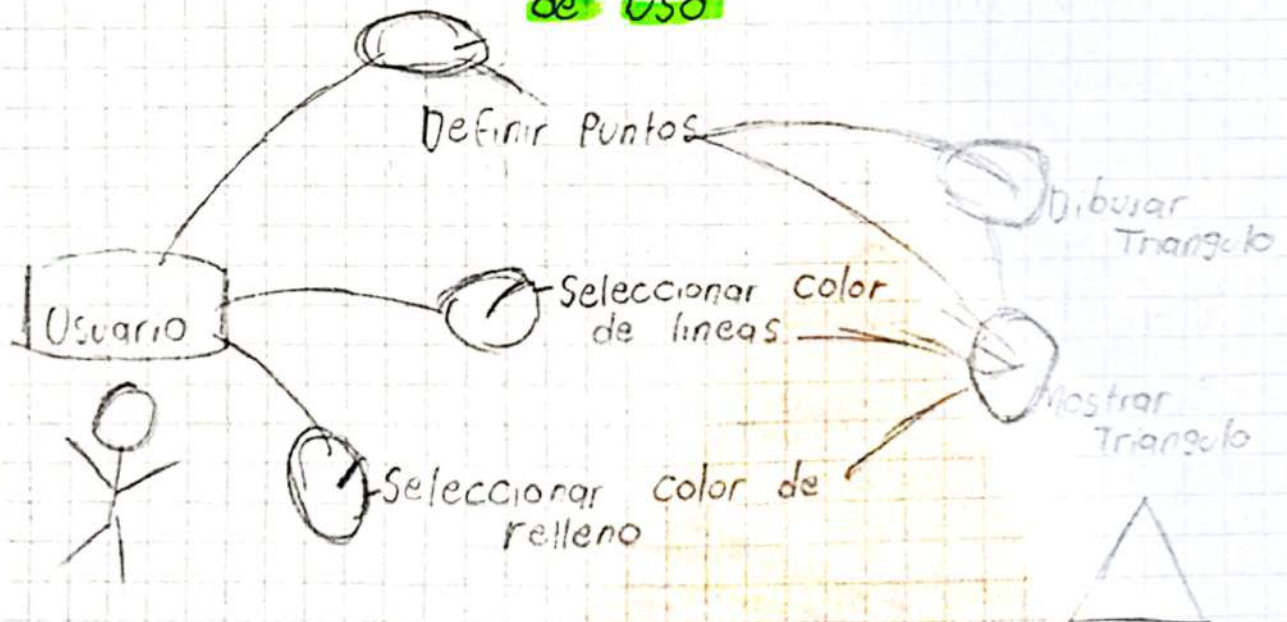
Nombre = R4 - Cambiar color de las lineas del triangulo

Nombre = R5 - Mostrar el Perimetro del triangulo

Nombre = R6 - Mostrar el area del triangulo

Nombre = R7 - Mostrar la altura del triangulo

Diagrama de Casos de Uso



Triangulo

```

+ Triangulo ()
+ dar Punto 1(): Punto
+ dar Punto 2(): Punto
+ dar Punto 3(): Punto
+ dar Color Lineas(): Color
+ dar Color Relleno(): Color
+ dar Perimetro(): double
+ dar Area(): double
+ dar Altura(): double
- Calcular Lado 1(): double
- Calcular Lado 2(): double
- Calcular Lado 3(): double
+ Calcular Angulo OPuesto (PX1: double, PY1: double, PX2: double, PY2: double, PX3: double, PY3: double): double
+ Cambiar Punto 1 (PX: double, PY: double): Void
+ Cambiar Punto 2 (PX: double, PY: double): Void
+ Cambiar Punto 3 (PX: double, PY: double): Void
+ Cambiar Color Relleno (pRojo: int, pVerde: int, pAzul: int): Void
+ Cambiar Color Lineas (pRojo: int, pVerde: int, pAzul: int): Void
+ metodo 1(): String
+ metodo 2(): String
    
```

Punto

```

- Punto 1
- Punto 2
- Punto 3

- X: double
- Y: double
+ punto (x: double, y: double)
+ dar X(): double
+ dar Y(): double
+ Cambiar X (pX: double): void
+ Cambiar Y (pY: double): void
    
```

Color

```

- Color Lineas
- Color Relleno

- rojo: int
- verde: int
- Azul: int

+ Color (pRojo: int, pVerde: int, pAzul: int)
+ dar Rojo(): int
+ dar Verde(): int
+ dar Azul(): int
+ Cambiar Rojo (pRojo: int): Void
+ Cambiar Verde (pVerde: int): Void
+ Cambiar Azul (pAzul: int): Void
    
```


Proyecto 11. Cine.

V. Debes plantear 2 ideas de proyecto

Idea 1.

RF1	Nombre	Crear una nueva tarjeta
	Resumen	Permite al usuario crear una nueva tarjeta cine.
	Entrada	Cédula propietario, dinero a ingresar en la tarjeta.
	Resultado	La tarjeta se creará con el depósito inicial del Cliente.
RF2	Nombre	Recargar la tarjeta cine
	Resumen	Permite al usuario recargar con dinero su tarjeta cine
	Entrada	Cédula propietario, Código tarjeta cine, dinero a recargar
	Resultado	La tarjeta cine se recargará con el depósito del Cliente
RF3	Nombre	Crear una reserva
	Resumen	El usuario podrá crear una reserva desde la app.
	Entrada	Cédula, correo, sillitas a elegir, película a ver
	Resultado	La reserva se realizará según los requerimientos del Cliente.
RF4	Nombre	Eliminar reserva actual
	Resumen	Permite al usuario eliminar una reserva existente
	Entrada	Cédula, correo, número de reserva
	Resultado	La reserva será eliminada.
RF5	Nombre	Pagar una reserva en efectivo
	Resumen	El usuario podrá pagar la reserva en efectivo
	Entrada	Cédula, correo, número pago, lugar pago (efectivo, suavol, etc)
	Resultado	La reserva será creada con un número de pago y se validará cuando el pago se haya hecho

Rf6	<p>nombre</p> <p>Resumen</p> <p>Entradas</p> <p>Resultado</p>	<p>Pagar la reserva con tarjeta cine</p> <p>El usuario podrá pagar su reserva con la tarjeta cine</p> <p>Cédula, correo, número de pago reserva, tarjeta cine.</p> <p>El usuario podrá pagar su reserva con su tarjeta cine y quedará reservado.</p>
Rf7	<p>nombre</p> <p>Resumen</p> <p>Entradas</p> <p>Resultado</p>	<p>Visualizar las sillas del cine.</p> <p>El usuario podrá revisar las sillas disponibles</p> <p>Pelicula a ver, horario, lugar.</p> <p>Se podrá visualizar las sillas disponibles para esa hora, sala y película.</p>
Rf8	<p>nombre</p> <p>Resumen</p> <p>Entradas</p> <p>Resultado</p>	<p>Visualizar dinero en caja</p> <p>El administrador podrá revisar el dinero en caja desde el app.</p> <p>Autenticación del administrador</p> <p>El administrador podrá revisar el dinero en caja después de haberse autenticado.</p>