

PROYECTO DE LA ASIGNATURA:

Integrantes:

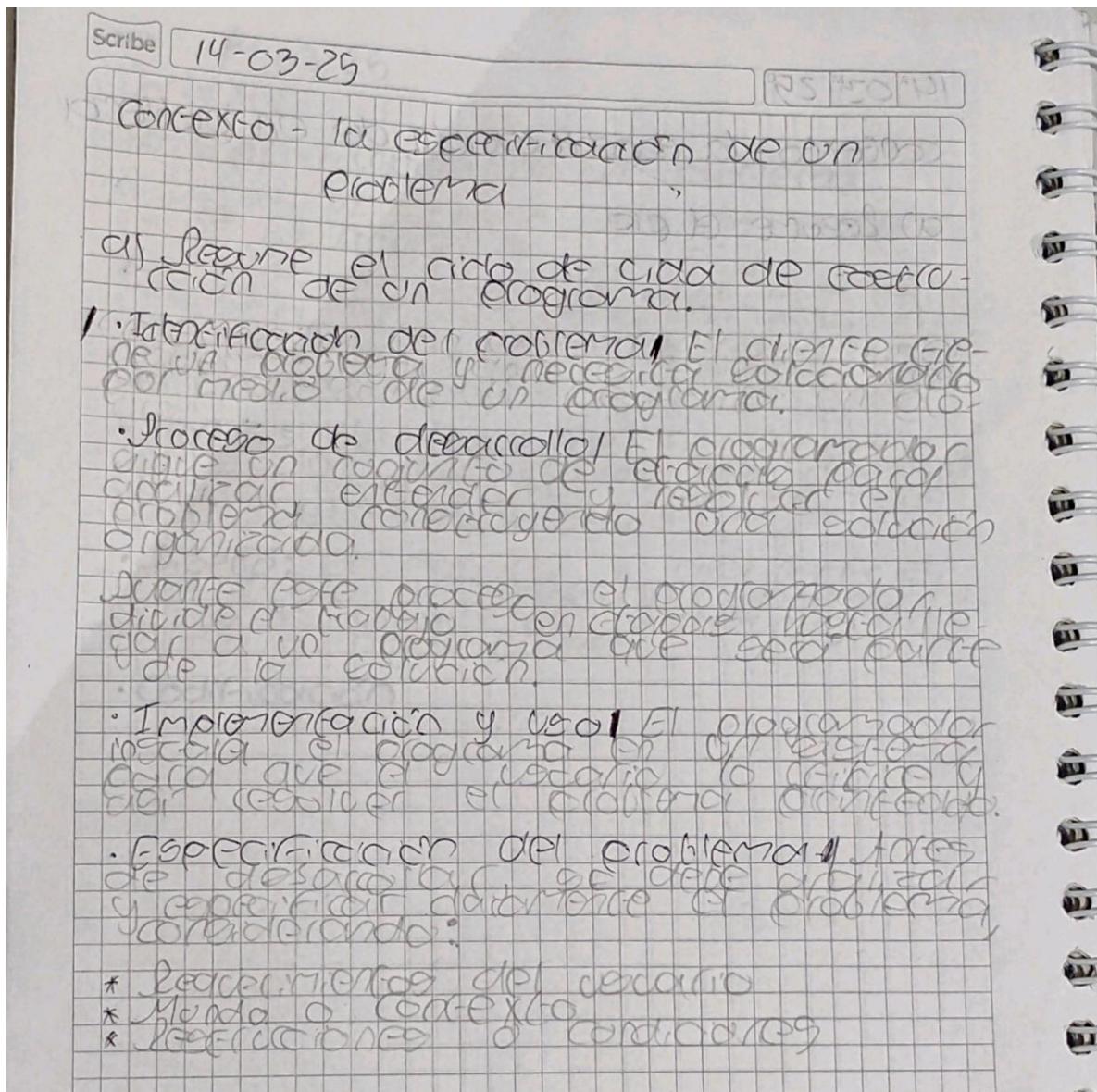
Javier Mateo Vargas Baquero

Sebastian Vaquiro

Braianth Rodríguez

Brayan Parra

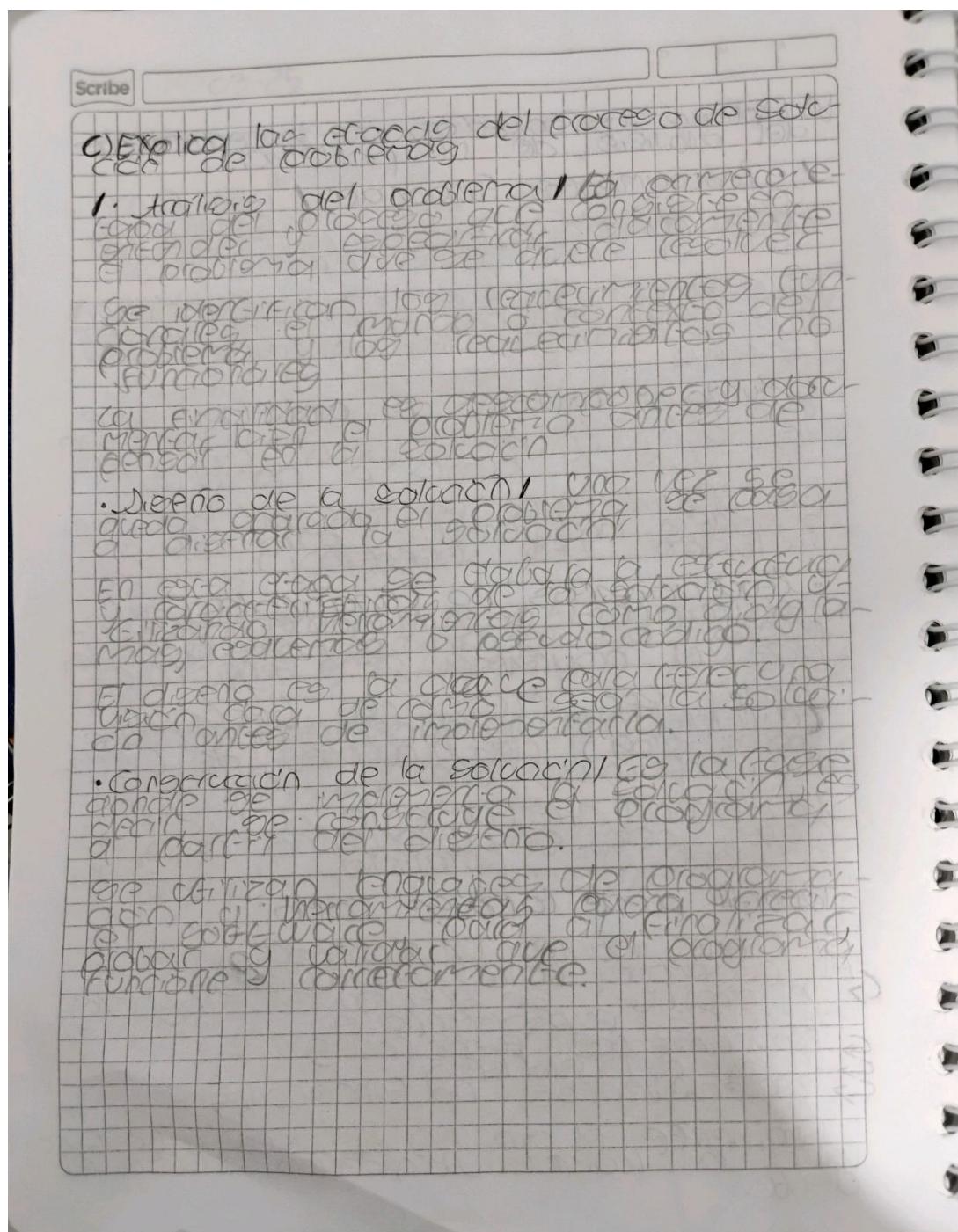
- a. Resuma el ciclo de vida de construcción de un programa.



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

- b) Explique los aspectos que hacen parte del análisis de un problema.
- 1. **Lequerimientos Funcionales / son funciones que el cliente necesita para que el sistema las realice**
En programación se manejan como procedimientos y funciones que el sistema debe proporcionar al cliente.
 - **Mundo o contexto del problema / son los datos y condiciones en los que el sistema opera.**
Se deben tener en cuenta las condiciones del entorno para que el sistema sea efectivo.
 - **Lequerimientos no Funcionales / son reglaciones o condiciones que limitan o dicen la manera en la que el programa debe funcionar.**
Pueden ser restricciones (tiempos, cantidad de usuarios simultáneos, precios, etc.) presupuestales, de calidad, rendimiento, etc.
No afectan directamente al mundo funcional pero si al entorno o a la calidad de servicio.
- También puede formarse como:
- El cliente plantea un problema
 - El programador realiza un análisis
 - Del análisis se comprenden los requerimientos
 - Lequerimientos Funcionales
 - Mundo del problema
 - Lequerimientos no Funcionales
- y finalmente el análisis se documenta para el cliente.

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



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

dijuales son los elementos que se
deben entregar a un cliente?

1. Al final del proceso de desarrollo
se debe entregar el diseño, los códigos
de corrección del programa, y el pro-
grama ya finalizado.

Con todo esto se dan los feri-
nados como los procesos visto
mañana.

- a. Elabore la Tarea No. 1 (pág. 5 del texto guía), con el objetivo de identificar los aspectos que forma parte de un problema.

Cliente	entidad bancaria
Usuario	usuarias banco
requerimiento funcional	el consultar saldo o el permiso de retirar dinero
Mundo problema	se requiere acceso a base de datos del banco; información del usuario, contraseña ID, numero de celular.
requerimiento no funcional	El aplicativo no puede manejar los cajeros automáticos, permite algunas interacciones con este pero no usa manejo ya que el cajero es un Sistema independiente y requiere usos manuales.

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

	<u>Nombre</u> cambiar clave de cuenta
	Resumen permite modificar la clave de la cuenta bancaria de un cliente.
Requerimiento funcional 1	Entradas - Nueva clave - Datos personales resultado La clave sea actualizado.
	<u>Nombre</u> Saber saldo de la cuenta
	Resumen permite saber el saldo asociado a una cuenta bancaria de un cliente
Requerimiento funcional 2	Entradas Datos personales - Número de cuenta resultado El saldo es de \$.....
	<u>Nombre</u> Solicitar prestamo bancario
	Resumen Permite solicitar un préstamo a la entidad bancaria en la cual está el cliente
Requerimiento funcional 3	Entradas - datos personales - cantidad de dinero - datos bancarios
	Resultados - Aprobado préstamo bancario - No aprobado préstamo bancario

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

Tarea 3	
Requerimiento Funcional	<p>Nombre: Determinar la clasificación del triángulo</p>
1	<p>Resumen: El programa debe identificar si un triángulo es equilátero, isósceles o escaleno en función de las longitudes de sus lados</p> <p>Entradas: Tres valores numéricos que representan los lados del triángulo</p> <p>Resultados: El programa nos dará como resultado la clasificación del triángulo (equilátero, isósceles o escaleno)</p>
2	<p>Nombre: Verificación de la existencia del triángulo</p> <p>Resumen: El programa deberá verificar si los tres lados ingresados pueden formar un triángulo</p> <p>Entradas: Tres valores numéricos que representan los lados del triángulo</p> <p>Resultados: un mensaje indicando si el triángulo formado es válido</p>

	<p>Nombre: calculo del area del triangulo</p>
Requerimiento funcional	<p>Resumen: El programador debe calcular el area del triangulo utilizando la formula de herón</p>
3	<p>Entradas: tres valores numericos que indiquen los lados del triangulo</p>
	<p>Resultado: un resultado numerico que representa el area del triangulo</p>

- d. Elabore la Tarea No. 4 (pág. 17), con el objetivo de identificar las entidades del mundo del problema.

Tarea 11

Nombre	Descripción	
Entidad	Triangulo	Representa la figura geométrica en si compuesta por tres lados y tres ángulos. El objeto inicial
Entidad	Lado	Cada uno de los tres segmentos de línea que forman el triángulo sus valores determinan el tipo y la existencia del triángulo
Entidad	Ángulo	Cada una de las tres aberturas formadas en la intersección de los lados pueden utilizarse para clasificar el triángulo

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

Rta: Al no identificarse estará incompleto y puede generar problemas como un código desorganizado, falta de precisión en el procesamiento de datos, mal análisis de los datos generando conflicto.

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

Rta: se puede definir si el concepto puede existir independiente y tiene propiedades propias

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

Tarea 5.

Objetivo: Identificar las características de las entidades del caso de estudio 2, un simulador bancario.

Para cada una de las cinco entidades identificadas en el caso de estudio del simulador bancario, identifique los atributos, sus valores posibles, y escriba la clase en UML. No incluya las relaciones que puedan existir entre las clases, ya que eso lo haremos en la siguiente etapa del análisis. Por ahora, trate de identificar las características de las entidades que son importantes para los requerimientos funcionales.

Atributo	Valores Posibles.	
Nº de Cuenta	5212-7512-3412-3456	Cuenta Bancaria
Titular de Cuenta	M. Molina.	# de cuenta Entidad. Titular de cuenta
Banco o Entidad	Av. Villas.	Diagrama UML.

* Clase: Cuenta - Bancaria.

Cuenta	Bancaria
# de cuenta	Entidad.

Atributo	Valores Posibles	
# de Cuenta	5212-7512-3412-3456	Cuenta Corriente
Banco	Bancolombia	# Cuenta Banco Pagos
Pagos	Depósito en cheque	Diagrama UML

* Clase: Cuenta Corriente

Cuenta	Corriente
# Cuenta	Pagos

Atributos	Valores Posibles.	
# de Cuenta	5212-7512-3412-3456	Cuenta Ahorros
Interés	% 0,8	# Cuenta Meta_Financiera Interés
Meta_Financiera	\$ 3'000 000	Diagrama UML

* Clase: Cuenta Ahorros

Cuenta	Ahorros
# Cuenta	Meta_Financiera

Atributos	Valores Posibles	
Interés	% 0,8	CDT
Rentabilidad	% 20	Plazo Rentabilidad
Plazo	12/03/2025	Interés

* Clase: CDT

CDT
Plazo
Rentabilidad

Diagrama UML

Atributo	Valores Posibles	
Diademanía	Lunes	
Fecha	27	
Nombre	Julio	Diagrama UML

↓ Clase: Mes

Mes
Diciembre
Nombre
Fecha

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

Tarea 6.

Objetivo: Reflexionar sobre el nivel de precisión que debe tener un algoritmo para evitar ambigüedades.

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. Pense por ejemplo si están completas.

¿Se preguntan para qué se integran de maneras distintas? ¿Estamos suponiendo que quien lo lee usa su "sentido común", o cualquier persona que lo use va a resolver siempre el problema de la misma manera?

→ El algoritmo como tal si está siendo enseñado a una persona adulta y con sentido común no tendría mucho problema en comprenderlo.

Pero más allá de eso al algoritmo le hace faltar determinismo pues no hay conciencia en los detalles más pequeños con sentido común. Por son muy claros con los detalles que se lograría cumplir. Por falta muy serillas. Pero sin él, no

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**.

Acción

Trabajo preliminar: Caso de estudio

Simulador bancario.

k)

Enunciado se desea crear un programa capaz de simular el comportamiento de las cuentas bancarias de un cliente, el cliente puede administrar tres productos financieros:

1. cuenta corriente.

2. cuenta de ahorros

3. certificado de depósito a término (CDT)

Requerimientos funcionales del proyecto

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 de 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 inversión en CDT

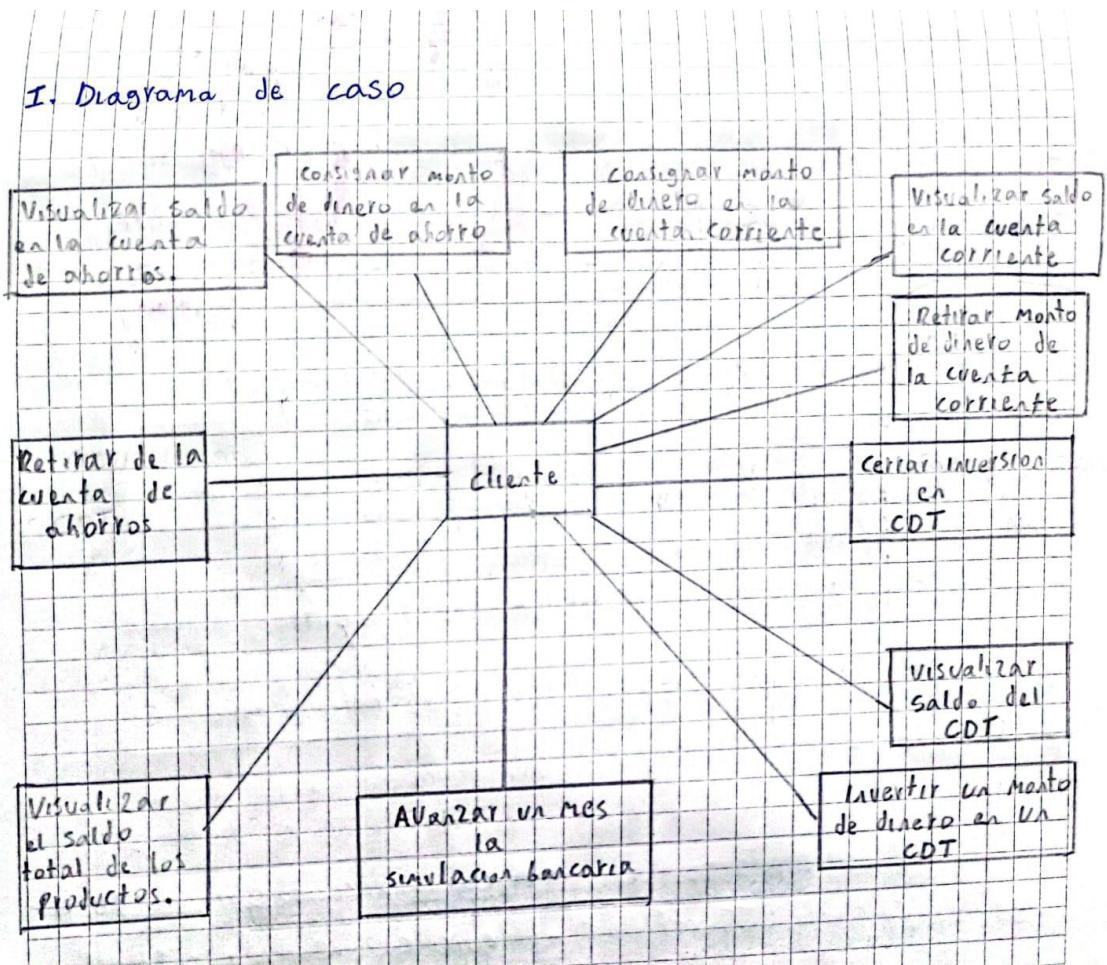
7. consignar dinero a la cuenta corriente del cliente.

8. retirar dinero de la cuenta corriente del cliente.

9. consignar dinero a la cuenta de ahorros del cliente.

10. retirar dinero de la cuenta de ahorros del cliente.

Dibuja el respectivo **diagrama de casos de uso** del ejemplo elegido (consulta "requerimientos funcionales" en 'Archivo').



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

mi el nombre de cada una de las clases y variables

clases mundo

Simulador Bancario

Variables.

- cedula : String
- Nombre : string
- mesActual : int
- inversion : CDT
- corriente : CuentaCorriente
- ahorros : CuentaAhorro

CDT

- valorInvertido; double
- interesMensual; double
- diasApertura; int

CuentaCorriente

- saldo : double

CuentaAhorros

- saldo : double
- interesMensual: double

Clase Interfaz

InterfazSimulador extends JFrame

- cuenta: SimuladorBancario

Atributos Interfaz

- PanelImagen: PanelImagen
- PanelAhorros: PanelAhorros
- PanelCorriente: PanelCorriente
- PanelCDT: PanelCDT
- PanelOpciones: PanelOpciones
- PanelSaldos : PanelSaldos
- PanelDatos: PanelDatoscliente

PanelAhorros extends JPanel implements ActionListener

- principal: InterfazSimulador

Atributos interfaz

- etiquetaSaldoAhorros: JLabel
- txtSaldoAhorros: JTextField
- btnConsignarCuentaAhorro: JButton
- btnRetirarCuentaAhorro: JButton

PanelCDT extends JPanel implements ActionListener

- principal: InterfazSimulador

Atributos interfaz

- etiquetaSaldoCDT: JLabel
- txtSaldoCDT: JTextField
- btnAbrirCDT: JButton
- btnCerrarCDT: JButton

PanelCorriente extends JPanel implements ActionListener

- txtSaldoCDT: JTextField
- btnCerrarCDT: JButton

PanelCorriente extends JPanel implements ActionListener

- principal: InterfazSimulador

Atributos de interfaz

- etiquetaSaldoCorriente: JLabel
- txtSaldoCorriente: JTextField
- btnConsignarCuentaCorriente: JButton
- btnRetirarCuentaCorriente: JButton

PanelDatosCliente extends JPanel

- etiquetaNombre: JLabel
- etiquetaCedula: JLabel
- txtNombre: JTextField
- txtCedula: JTextField

PanelOpciones extends JPanel implements ActionListener

- principal: InterfazSimulador

Atributos de interfaz

- opcion1: JButton

- opcion2: JButton

PanelSaldos. Extends JPanel Implements ActionListener

- principal: InterfazSimulador

Atributos de interfaz

- etiquetasaldo: JLabel - txtmes: JTextField

- etiquetames: JLabel - botonAvanzarMes: JButton

- txtsaldo: JTextField

clase pruebas

- SimuladorBancario Test

CDTTest

- cuenta: SimuladorBancario.

- cdt: CDT

CuentasAhorrosTest

- cuenta: CuentaAhorros

CuentaCorrienteTest

- cuenta: corriente

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

Página de gestión estudiantil

Página de gestión estudiantil		
Nombre	R7: actualizar nota del estudiante	
Resumen	permite modificar la calificación obtenida por un estudiante en una materia	
Requerimiento funcional 7	entradas - NUEVA nota - Nombre de estudiante	
resultado:	La nota ha sido actualizada correctamente	
Nombre	registrar estudiante	
Resumen	permite registrar en el sistema un nuevo estudiante	
Requerimiento funcional 2	entradas - Nombre completo, identificación, carrera, semestre	

Requerimiento funcional 2	Nombre completo, identificación, carrera, semestre
	se registró al estudiante exitosamente
Nombre	Registrar asistencia a clase
Resumen	permite realizar un registro de asistencia a un estudiante en una clase determinada
Requerimiento funcional 3	
	entradas - código de clase - lista de estudiantes
Resulado	

	Nombre	Crear horario académico
Resumen		permite generar un horario según las materias seleccionadas
Requerimiento funcional 4	entradas	- código de materias - días - horas de clase
Resulado		Se creó exitosamente el horario académico

Gestion de eventos

Idea 2

Gestion de eventos

	<p>Nombre Registrar eventos</p>
	<p>Resumen Permite agregar informacion de un evento o concierto a una base de datos.</p>
Requerimiento funcional 1	<p>entradas - Nombre del evento - fecha - Hora - ubicacion - capacidad maxima - artista principal.</p>
	<p>resultado Se registro correctamente el evento</p>
	<p>Nombre Venta de boletos</p>
	<p>Resumen registra la venta de boletos para un evento seleccionado</p>

Requerimiento funcional 2	entradas	- ID del evento - nombre del comprador - cantidad de boletos
	resultado	Se registró una venta
Resumen	Nombre	Asignar recursos para el montaje
	Resumen	permite registrar equipo y recursos para el evento
Requerimiento funcional 3	entradas	- tipo de recurso - cantidad - ubicación - tiempo de uso
	resultado	los recursos se asignaron correctamente
Resumen	Nombre	Registrar artistas contratados
	Resumen	Permite agregar a la base de datos, los datos de artistas contratados
Requerimiento funcional 4	entradas	- Nombre artista - Género musical - tarifa contratación - Duración del show
	resultado	El artista se registró correctamente