

## Guía taller del proyecto

### Integrantes

- Juan Esteban Lizcano Drieno Cod=10432427165
- Carlos Balvino Narosajua Rodriguez Cod 10432426951
- Sherman Daniel Rodriguez Niño cod 11162426496
- Joseph Esteban Linares Niño : cod 11482425182

### 9) Resume el ciclo de la vida de construcción de un programa

\* El ciclo de la vida de construcción de un programa, Esta compuesto por varias etapas que guían el desarrollo de un software desde la concepción hasta su retiro.

Las etapas principales son:

- 1) **Recolección de requisitos**: Se identifican las necesidades y expectativas de los usuarios
- 2) **Análisis**: Se analizan los requisitos recopilados para entender el problema y definir las especificaciones del Software
- 3) **Diseño**: Se elabora la arquitectura del software incluyendo la estructura de datos, interfaces y módulos. Se crean diagramas y modelos que guiarán la implementación
- 4) **Implementación**: Se escribe el código del programa según las especificaciones del diseño. En esta etapa incluye la codificación y la integración de los diferentes códigos
- 5) **Pruebas**: Se realizan pruebas del código para identificar y corregir los errores que muestren
- 6) **Despliegue**: Se entrega el software a los usuarios y se instala en el entorno de producción
- 7) **Mantenimiento**: Se realizan actualizaciones y correcciones de errores después de un tiempo. También se puede incluir la adición de nuevas funcionalidades y la mejora del rendimiento del software
- 8) **Retiro**: Cuando el software ya no es útil o ha sido reemplazado, se retira del uso activo



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

Implica varios aspectos claves que ~~se~~ ayudan a comprender la situación y a definir la solución adecuada.

1) **Definición del problema:** Clarificar lo que exactamente se necesita resolver, esto incluye la identificación de la causa del error

2) **Contexto:** Entender el entorno en el que se representa el problema, incluyendo factores económicos y tecnológicos que puedan influir en la solución.

3) **Requisitos:** Determinar que requisitos deben de cumplirse para que la solución sea efectiva

4) **Partes interesadas:** Identificar a todas las personas o grupos que se verán afectados por el problema y la solución

5) **Limitaciones y restricciones:** Reconocer las limitaciones que pueden afectar la solución, recursos disponibles, tiempo, presupuesto y normativas

6) **Análisis de riesgos:** Evaluar los posibles riesgos asociados al problema y también con sus respectivas soluciones

c) Explique las etapas del proceso de solución de problemas

1) **Identificación del problema:** Reconocer el problema y definirlo claramente

2) **Análisis del problema:** Investigar y comprender las causas del problema

3) **Generación de Alternativas:** Proponer diferentes ideas para resolver el problema

4) **Evaluación de Alternativas:** Comparar las diferentes soluciones en función de criterios como viabilidad, costo, tiempo y efectividad

5) **Selección de la solución:** Elegir la alternativa que mejor se adapte a las necesidades y restricciones identificadas

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

\* Elementos a entregar a un cliente:

### 1) Documentación del proyecto

- Descripción del problema y los objetivos
- Especificaciones técnicas y funcionales

### 2) Código Fuente

- Archivos del código programado
- Comentarios y documentación dentro del código para facilitar el entendimiento

### 3) Ejemplos de uso

- Casos de prueba y ejemplos que muestran como el uso del software

### 4) Resultados y Salidas:

- Ejemplo de los resultados que el programa devuelve

### 5) Soporte y Mantenimiento

- Información sobre cómo contactar al soporte técnico
- Planes de mantenimiento, si se acordó uno previamente

### 6) Derechos y Licencias

- Clarificación sobre los derechos de uso del código y las aplicaciones creadas

### 7) Plazos y entregables

- confirmación de los plazos de entrega y los hitos alcanzados



## Tarea #1.

**Objetivo:** Identificar los aspectos que forman parte de un problema.

**El problema:** 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.

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.

**Cliente:**

Banco

**Usuario:**

Clientes del banco que utilizan los cajeros automáticos para retirar dinero.

**Requerimiento Funcional:**

El programa debe permitir al usuario autenticarse con sus datos (cédula, teléfono, contraseña, etc.) y seleccionar el monto a retirar dentro de los límites establecidos.

**Mundo del problema:**

Es necesario conocer la disponibilidad de efectivo en cada cajero, el límite de retiro por transacción y por día, así como el saldo disponible en la cuenta del usuario. También se debe considerar la posibilidad de fallas técnicas o errores en la conexión que podría afectar la transacción.

**Requerimiento No Funcional:**

El programa debe garantizar la seguridad de los datos del usuario mediante autenticación segura y cifrado de la información. Además debe ser rápido y eficiente, evitando tiempos de espera.



## Tarea #2:

**Objetivo:** Crear habilidad y especificación de requerimientos funcionales.  
 Para el caso de estudio Z, un simulador bancario, identifique y especifique tres requerimientos.

<b>Requerimiento Funcional #1</b>	<p><b>Nombre:</b> Consulta de saldo</p> <p><b>Resumen:</b> Permite al cliente verificar el monto disponible en su cuenta bancaria.</p> <p><b>Entradas:</b></p> <ul style="list-style-type: none"> <li>• Número de cuenta</li> <li>• Tipo de cuenta (DB o CD)</li> <li>• Documento de Identidad</li> <li>• Autenticación del usuario (Contraseña)</li> </ul> <p><b>Resultado:</b> El usuario podrá visualizar el saldo disponible en su cuenta.</p>
<b>Requerimiento Funcional #2</b>	<p><b>Nombre:</b> Depositar dinero en la cuenta</p> <p><b>Resumen:</b> Permite al cliente ingresar dinero a su cuenta bancaria.</p> <p><b>Entradas:</b></p> <ul style="list-style-type: none"> <li>• Número de cuenta bancaria</li> <li>• Monto a depositar</li> <li>• Tipo de cuenta (Débito o Crédito)</li> <li>• Medio de pago (Efectivo, transferencia)</li> </ul> <p><b>Resultado:</b> El saldo de la cuenta del cliente se actualizará con el monto depositado.</p>
<b>Requerimiento Funcional #3</b>	<p><b>Nombre:</b> Solicitar una tarjeta de crédito o débito</p> <p><b>Resumen:</b> Permite al cliente solicitar una nueva tarjeta bancaria, Débito o Crédito.</p> <p><b>Entradas:</b></p> <ul style="list-style-type: none"> <li>• Nombre completo del cliente</li> <li>• Fecha de nacimiento</li> <li>• Documento de identidad</li> <li>• Comprobante de ingresos</li> <li>• Dirección de domicilio</li> </ul> <p><b>Resultado:</b> El banco evaluará la solicitud y si es aprobada el cliente recibirá su tarjeta.</p>



#### Tarea #4.

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

Lea el enunciado del caso y trate de guiarse por los sustantivos para identificar las entidades del mundo del problema.

	Nombre	Descripción
Entidad	Triangulo	Representa la figura con sus lados y ángulos.
Entidad	Tipo de Triangulo	Clasificación de triángulo según lados y ángulos.
Entidad	Calculo Geométrico	Permite calcular área, perímetro y otras medidas.

¿Que pasa si no identificamos bien las entidades del mundo?

**BT1 //** si no identificamos bien las entidades del mundo tendremos una dificultad y mal manejo en la estructura del programa, eso también incluye los errores en la funcionalidad.

¿cómo decidir si se trata efectivamente de un entidad y no solo de una característica de una entidad y identificada?

**BT1 //** si un concepto puede existir por si mismo, significa que tiene múltiples atributos y eso puede relacionarse con otras entidades, entonces en este caso sería una entidad. pero si solo se describe algo dentro de una entidad, entonces solo sería una característica.

**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.

Requerimiento funcional #1	<p><b>Nombre:</b> Calcular el área del triángulo</p> <p><b>Resumen:</b> El programa calculará el área del triángulo utilizando la fórmula correspondiente con los datos ingresados por el usuario.</p> <p><b>Entradas:</b> <ul style="list-style-type: none"> <li>• Base (b)</li> <li>• Altura (h)</li> </ul> </p> <p><b>Resultado:</b> El programa mostrará el área del triángulo calculada con la fórmula <math>\text{Área} = \frac{b \cdot h}{2}</math></p>
Requerimiento funcional #2	<p><b>Nombre:</b> Determinar el tipo de triángulo</p> <p><b>Resumen:</b> El programa analizará las longitudes de los lados del triángulo y determinará a qué tipo pertenece.</p> <p><b>Entradas:</b> <ul style="list-style-type: none"> <li>• Lado a</li> <li>• Lado b</li> <li>• Lado c</li> </ul> </p> <p><b>Resultado:</b> El programa indicará el tipo de triángulo en función de sus lados y ángulos.</p>
Requerimiento funcional #3	<p><b>Nombre:</b> Calcular el perímetro del triángulo</p> <p><b>Resumen:</b> El programa calculará el perímetro del triángulo sumando las longitudes de sus tres lados.</p> <p><b>Entradas:</b> <ul style="list-style-type: none"> <li>• Lado a</li> <li>• Lado b</li> <li>• Lado c</li> </ul> </p> <p><b>Resultado:</b> El programa mostrará el perímetro del triángulo, calculado con la fórmula: <math>\text{perímetro} = a + b + c</math></p>



1. Elaborar base de datos (pag 107) con el objetivo de identificar las identidades del mundo.

1.1. ¿Cuáles son los atributos principales de la clase cuenta bancaria, y cuáles son sus valores posibles?

La clase cuenta bancaria representa una cuenta genérica dentro de un sistema bancario. Algunos de sus atributos pueden ser:

Atributo	valores posibles
Nombre de cuenta	Cadena numérica única
titular	Nombre del cliente o asociado
Saldo	Valor numérico
tipo de moneda	como USD, EUR
Estado	activo, inactivo, bloqueado

- ¿Qué atributos deberían tener la funcionalidad de cuenta bancaria por lo que pueden incluir atributos específicos como:

Atributo	valores posibles
límite de sobregiro	valor numérico
comisión sobregiro	porcentaje
cheques asociados	{ 'Si' / 'No' }
tarifa de mantenimiento	valor numérico



3 cuáles son los atributos característicos de una cuenta ahorros dentro de simulador bancario

- la cuenta ahorro también hereda de cuenta bancaria pero con atributos que la diferencian

Atributo	valores posibles
Tasa de Interés	porcentaje %
Retiro permitido	'Si' 'No'
Monto mínimo de depósitos	valor numérico
bonificaciones	'Si' 'No'

- los CBT tienen características particulares:

Atributo	valores posibles
Monto Inicial	valor numérico
plazo	Entero
tasa de Interés	porcentaje %
Renovación automática	'Si' 'No'

- El mes es una clase auxiliar que podría utilizarse para gestionar operaciones bancarias dentro de un periodo determinado

Atributo	valores posibles
Nombre	Enero Febrero
Nº mes	Entero
Cantidad días	Entero



¿Cómo se representaría cada una de estas clases de un diagrama UML sin incluir las relaciones entre ellas?

1 El nombre de clase

2 los atributos de clase

3 Métodos o comportamientos (si hay)

¿Que diferencias existen entre atributos de cuenta corriente y cuenta ahorros en términos de funcionalidad bancaria?

- cuenta corriente permite sobregiros tiene comisiones y cobra cheque

cuenta ahorros tiene tasa de interés bonificación y un saldo mínimo

¿Cómo influye la clase Mes en la gestión de productos financieros dentro del simulador bancario?

permite calcular intereses mensuales para cuenta ahorros y CDT

define periodos como el cobro de comisiones en cuenta corriente

puede ayudar a la planificación de pagos y verificaciones

Es posible que algunos atributos sean comunes entre cuenta bancaria, corriente y ahorros

• N° cuenta, titular, Saldo, tipo de moneda, estado



## Punto 5

Las instrucciones de un algoritmo deben ser claras, precisas y no dejar espacio a interpretaciones ambiguas. Si el algoritmo para moverse en el metro de París no es lo suficientemente detallado, pueden surgir varios problemas:

### 1. Falta de información específica

- ¿Se indica el nombre exacto de las estaciones de origen y destino?
- ¿Se especifican las líneas y direcciones que deben tomarse?

### 2. Ambigüedad en las instrucciones

- Expresiones como "tomar la siguiente línea" pueden ser confusas si hay múltiples líneas.

Indicaciones como bajar en la tercera estación pueden generar errores si el usuario no cuenta bien.

### 3. Dependencia del sentido común

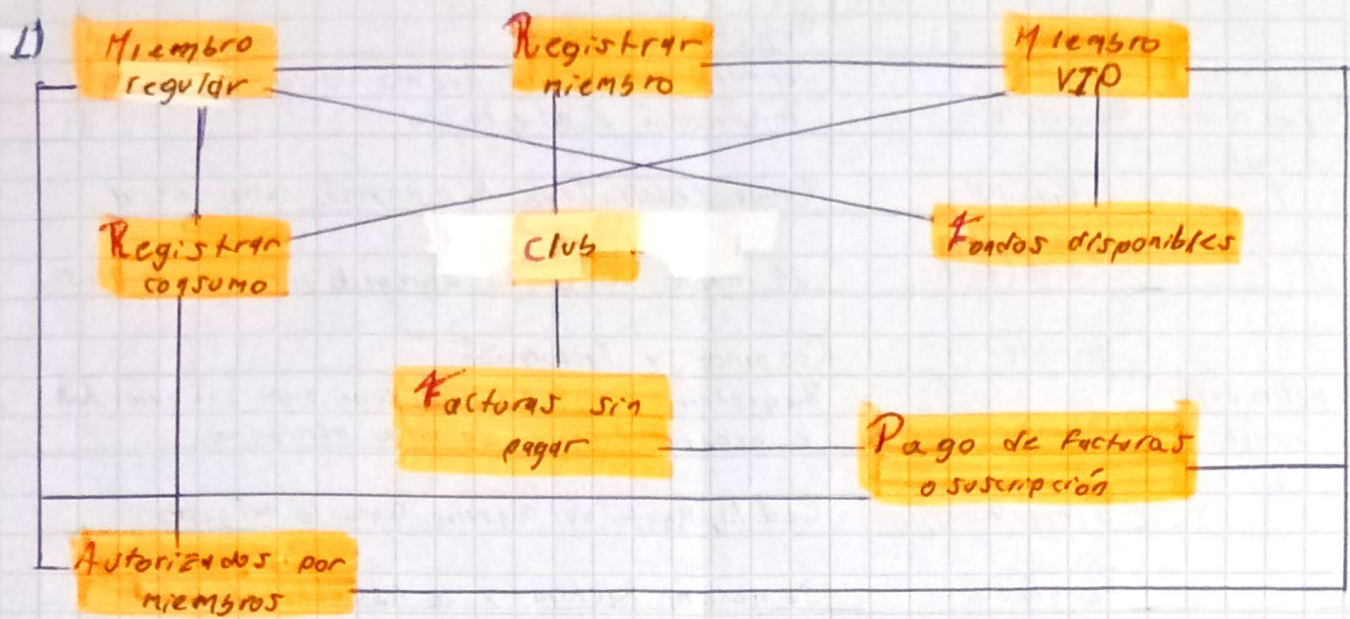
No todas las personas interpretan las instrucciones de la misma manera.

Diferencia cultural o de idioma pueden hacer que algo evidente para una persona no lo sea para otra.

### 4. Posibles interpretaciones múltiples

Si las instrucciones no especifican si se deben seguir ciertas señales o si hay alternativas en errores el usuario puede confundirse.





## M) Clase = Socio

Atributos = (cedula = string) (Nombre = string) (Fondos = float) (Tipo suscripción = string)  
(Facturas sin pagar = List[Factura]) (Personas autorizadas = List[Personas autorizadas])

Función = (Afiliar) (Agregar/Eliminar persona Afiliada) (Registrar consumo) (Pagar factura)  
(Aumentar Fondos)

## Clase = Factura

Atributos = (concepto = string) (valor = float) (Generado por = string)

Función = (Generar deuda/pago)

## Clase = Persona Autorizada

Atributos = (Nombre = string) (Socio Asociado = Socio)

Función = (Realizar consumo = valor: float, Concepto: string)

## Clase Club

Atributos = (Socios = List[Socios])

Función = (Afiliar socio = socio) (Buscar socio = cedula: string) (Verificar límite = vip)



N)

Requerimiento Funcional 1	Nombre	Registro de socios
	Resumen	Permite registrar nuevos socios con su información y saldo inicial
	Entradas	Nombre, Cédula, Tipo de suscripción, saldo inicial
	Resultados	Un nuevo socio es agregado a la base de datos
Requerimiento Funcional 2	Nombre	Consumos y facturación
	Resumen	Registrar los consumos de un socio con autorización y generar factura con saldo subyacente
	Entradas	Cédula, Monto del consumo, concepto del gasto
	Resultados	Se genera factura y se descuenta el saldo
Requerimiento Funcional 3	Nombre	Pago de facturas
	Resumen	Permite a los socios pagar las facturas pendientes con el saldo disponible
	Entradas	Cédula, pagar factura
	Resultados	La deuda se elimina y se actualiza el saldo
Requerimiento Funcional 4	Nombre	Gestión de Fondos
	Resumen	Permite agregar fondos a su saldo, respetando los límites de suscripción
	Entradas	Cédula, Monto a cargar
	Resultados	Se actualiza el saldo del socio

Idea = Sistema de Gestión del club



Requerimiento funcional 1	Nombre	Creación y administración de eventos
	Resumen	Permite registrar eventos con detalles de nombre, fecha, lugar y capacidad máxima
	Entrada	Nombre evento, fecha, lugar, capacidad
	Resultado	Se registra el evento en el sistema y se dispone.
Requerimiento funcional 2	Nombre	Registro de asistentes
	Resumen	Los usuarios pueden inscribirse en un evento y recibir confirmación
	Entrada	Nombre asistente, correo, evento que desea
	Resultado	Queda registrado el asistente y se envía confirmación
Requerimiento funcional 3	Nombre	Gestión de acceso
	Resumen	Controlar el ingreso de los asistentes del día mediante QR o lista de registro
	Entrada	Código QR o Nombre asistente
	Resultado	Se verifica registro y asistencia
Requerimiento funcional 4	Nombre	Evaluación y comentarios post-evento
	Resumen	Permite calificar el evento y dejar comentarios para futuras mejoras
	Entrada	ID evento, calificación (1-5 estrellas) comentario opcional
	Resultado	Se almacena la retroalimentación para mejorar en eventos futuros

Idc = Gestión de eventos