

Proyecto
Lógica
Computacional

Brandon Santiago Faverro Alarcón
Juan David Gavira Acevedo
Juan David Sanabria Ramírez
Jonathan Andrés Triana Duvara

Universidad Antonio Nariño

Juan Martínez

Bogotá D.C

21 de septiembre de 2024



ACTIVIDADES PRÉELIMINARES

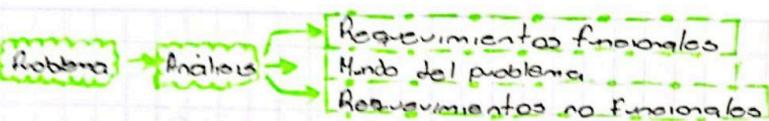
CONTEXTO

Brandon Forero
Juan García
Juan Sarabria
Andrés Triana

- La especificación de un problema:
 - a. Resuma el ciclo de vida de construcción de un programa.
 - b. Explique los aspectos que hacen parte del análisis de un problema.
 - c. Explique las etapas del proceso de solución de problemas.
 - d. ¿Cuáles son los elementos que se deben entregar a un cliente?

Solución

- a) Primero se debe analizar el problema para poder identificar objetivos, pasos y como el programa va a poder resolvélo, este paso implica recopilar información del cliente y de los usuarios a los que va destinado.
 - b) Aquí se estructura como se va a resolver el problema de manera estructurada, también el cómo estará la estructura del programa dividiendo la solución en diferentes módulos o componentes que puedan verificarse de manera individual.
 - c) En este punto empieza la codificación siendo este paso fundamental ya que el código debe ser actualizado y complicarlo o no seguir las normas de programación puede llevar a problemas futuros.
 - d) El código debe ser probado a fondo en busca de cualquier fallo y cualquier forma de aprovecharse siendo las formas más comunes las pruebas unitarias (por módulo), pruebas de integración (combinando módulos) y pruebas de sistema (todo el programa).
 - e) Este paso trata de probar el código de todo error lanzado durante la parte de prueba haciendo que funcione de manera correcta y pueda resolver el problema planteado.
 - f) Aquí se hace toda la documentación del programa como manuales, documentos y todo para que los usuarios pueden usarlo de manera correcta y en un futuro se pueda actualizar de la misma manera.
 - g) Mantener el programa de diversas formas como aumentando servidores o buscando errores que puedan ir saliendo durante el proceso esto siendo el paso más largo porque debe ser repetido durante toda la vida útil del programa.
- b) Explique los aspectos que hacen parte de un análisis de un problema.
- Analizar el problema para entenderlo bien. Es importante asegurarse de que se está abordando el problema correcto que el cliente necesita resolver, evitando malentendidos.
 - Dividir el problema en tres partes, esto facilita su comprensión y permite resolverlo paso a paso, simplificando el trabajo y haciendo más manejable.
 - Este método es útil en cualquier tipo de trabajo o problema. Aunque es común en computación, también se aplica a problemas de otras áreas.
 - Crear documentos para explicar la solución al cliente. Al finalizar el análisis, es esencial contar con herramientas que permitan comunicar la solución de manera clara y eficaz.



c) Explica los pasos del proceso de solución de un problema.

- ✓ Para desarrollar y profundizar muy bien sus etapas los podemos dividir en algunas partes:
- 1 El cliente ya sea persona natural o de una organización necesita la construcción de un programa. Y ésta realiza contacto con una empresa desarrolladora de software que pone a un profesional del campo a su disposición.
- 2 El profesional sigue un conjunto de pasos (proceso) para entender las necesidades del cliente y construir una solución de calidad.
- 3 Este profesional instala al programa que le convenga si el cliente puede usarlo para resolver definitivamente. Además, el cliente pone a alguien desde su gerencia dentro de su organización.
- **Especificaciones de un problema:** Iniciamos haciendo énfasis de que un programador no puede resolver un problema que no tiene conocimiento. Por esta razón, la primera etapa en todo proceso de construcción consiste en tratar de entender el problema, y expresar cualquier tipo de información, de tal manera, que cualquier otra persona del equipo de desarrollo pueda entender sin dificultad lo que espera el cliente de la solución. Para iniciar en los elementos de la especificación, vamos a hacer el paralelo con otras ingenierías. Considera el caso de un ingeniero civil que se enfrenta al problema de construir una carretera. Para eso debe tratar de identificar al menos tres aspectos del problema: los requerimientos del usuario (entre qué puntos quiere el cliente la carretera, cuántos curviles debe tener, etc).

- Identificar el lugar del problema, (el tipo de terreno, la cantidad de lluvia, la temperatura).

- Las restricciones y condiciones que plantea el cliente (el presupuesto máximo, que los pedidos no superen el \$1), serán una pérdida de tiempo y de recursos para el ingeniero civil, intentar construir la carretera si no ha entendido claramente los tres aspectos mencionados, y con estos errores el profesional puede perder la confianza del cliente. Para finalizar, cualquier problema, de cualquier campo se pueden dividir en estos 3 aspectos.

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

- Requerimientos funcionales: lo que el cliente espera en términos de la función de la solución (por ejemplo, un programa que resuelva mi tarea específica).
- Contenido de uso: El entorno en el que la solución será utilizada, como la oficina del cliente o una tarea.
- Requerimientos no funcionales: Restricciones o condiciones impuestas por el cliente, como tiempos de entrega, limitaciones técnicas o condiciones organizativas.

EXPERIENCIA

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

- 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 solo 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 medida de completarlo.
- Cliente = Un banco.
- Usuario = Usuarios del banco.
- Requerimiento funcional: Los cajeros automáticos deben permitir consultar el saldo de la cuenta y retirar dinero de ella.
- Mundo del problema: Se debe identificar el tipo de empresa bancaria, se debe conocer el tipo de divisa manejada en las regiones donde se ubican los cajeros automáticos, así como el funcionamiento de los cajeros automáticos antes de la instalación.
- Requerimiento no funcional: Conocer el monto por transacción máximo del cajero automático en base al límite del monto del usuario, tener un método de verificación del usuario del cajero automático, y conocer la cantidad de dinero disponible en el cajero.

f. Elabore la tarea No.2 (Pág. 13), con el objetivo de identificar los requerimientos funcionales de un problema.

	Nombre	T: Realizar transferencias entre cuentas.
	Resumen	El programa debe ser capaz de hacer que el usuario pueda transferir su plata a otra cuenta, garantizando que los fondos estén disponibles en las cuentas.
1	Requerimiento funcional	<ul style="list-style-type: none"> - Id de salida - Id de entrada - Cantidad a transferir - Mensaje del remitente - Autenticación del usuario - La transferencia es exitosa: El usuario recibe una notificación confirmando el envío y su correcto funcionamiento.
	Entradas	
	Resultado	<ul style="list-style-type: none"> - La transferencia falla: El usuario recibe una notificación de que el envío a fallado junto con la causa de este (fondos insuficientes, cuenta no válida, etc.), haciendo que el dinero no se envíe.
2	Nombre	CS: consultar saldo.
	Resumen	El programa debe ser capaz de mostrarle al usuario en tiempo real su saldo de manera precisa y entendible.
	Entradas	Número de cuenta.
	Resultado	El usuario debería poder ver en la pantalla el saldo retenido y el dinero total.
3	Nombre	GP: simulador de préstamos
	Resumen	El sistema debe permitir a los usuarios simular un préstamo bancario ingresando la cantidad a solicitar y el plazo deseado mostrando el plan de pagos y los intereses que se generan, durante el período.
	Entradas	<ul style="list-style-type: none"> - Cantidad del préstamo solicitado. - Plazo de pago (en meses o años). - Tasa del interés anual. - Tipo de préstamo (hipotecario, personal, automotriz, etc.).
	Resultado	El sistema debe dle muestra la información referente al préstamo como las cuotas de cada mes, los intereses y el pago total junto a la opción de aceptar o denegar el préstamo.

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

Requerimiento funcional 1	Nombre	AT: Calcular el área del triángulo.
	Resumen	El programa debe ser capaz de calcular el área de un triángulo proporcionado por la información del usuario usando la fórmula $\text{Área} = \frac{1}{2} * \text{Base} * \text{Altura}$.
	Entradas	<ul style="list-style-type: none"> - Base del triángulo (solo número positivo). - Altura del triángulo (solo número positivo). - El usuario podrá ver en pantalla el área del triángulo basado en los datos ingresados.
Requerimiento funcional 2	Resultado	<ul style="list-style-type: none"> - En caso de haber insertado un número negativo o el número cero el sistema le manda un mensaje de error indicando que solo se pueden números positivos.
	Nombre	TT: Determinar el tipo de triángulo según sus lados.
	Resumen	Dependiendo de las longitudes de los lados dados en la información enviada por el usuario, el programa deberá ser capaz de identificar el tipo de triángulo (equilátero, isósceles o escaleno).
Requerimiento funcional 3	Entradas	<ul style="list-style-type: none"> - Longitud del lado A (Valor numérico positivo). - Longitud del lado B (Valor numérico positivo). - Longitud del lado C (Valor numérico positivo). - El sistema identifica el tipo de triángulo basado en el sistema de equilátero (todos los lados iguales), isósceles (dos lados iguales) o escaleno (todos los lados diferentes) y muestra el resultado al usuario.
	Resultado	<ul style="list-style-type: none"> - En caso del usuario haber digitado números negativos, el número cero o una longitud imposible dando la creación del triángulo le llegara un mensaje de error.
	Nombre	CA: Calcular los ángulos del triángulo.
Requerimiento funcional 4	Resumen	El sistema debe permitir a los usuarios similar y debe ser capaz de calcular los ángulos internos con los datos de las longitudes dadas por el usuario esto será calculado por la ley del seno.
	Entradas	<ul style="list-style-type: none"> - Longitud del lado A (Valor numérico positivo). - Longitud del lado B (Valor numérico positivo). - Longitud del lado C (Valor numérico positivo). - El sistema calcula los tres ángulos internos del triángulo y los muestra en grados o radianes.
	Resultado	<ul style="list-style-type: none"> - Si los valores internos no forman un triángulo o tienen números negativos o cero, el sistema muestra un mensaje de error indicando que los lados no cumplen para la creación de un triángulo.

h. Elabore la Tarea No. 4 (Pág. 11) con el objetivo de identificar las entidades del mundo del problema.

- Objetivo: Identificar las entidades para el caso de un programa que maneje un triángulo.

Entidad	Nombre	Descripción
1	Puntos	✓ Los puntos son una parte esencial en el mundo de este enunciado ya que son los que dan la forma al triángulo, dándole sentido al programa.
2	Líneas	✓ Debido a que se le debe dar forma al triángulo, se conectan los puntos sobre ejes X y Y con los puntos dando la forma de triángulo.
3	Color	✓ El color es una parte que, aunque parezca innecesaria es esencial en el mundo de este problema ya que la mayor parte del programa se basa en la customización de las distintas partes del círculo.

Punto de reflexión:

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

✓ No tendríamos una vista plena y actualizada sobre lo que necesitamos en el problema, corriendo el riesgo de perder el enfoque e irnos por las ramas del objetivo, incluso llegando a solucionar un problema distinto al que está pasando el programa.

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

✓ Se puede explicar como que estos entidades del problema son partes de la solución que debemos dar, teniendo una gran relevancia en como analizamos y resolvemos el problema, poniéndose como puntos objetivos en la creación del programa.

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

✓ 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 pueden 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.

• Clase: Cuenta Bancaria

Atributo	Valores Posibles
Nombre	Cadena de caracteres
Apellido	Cadena de caracteres
Número de documento	Valores enteros y decimales

• Clase: Cuenta corriente

Atributo	Valores Posibles
Consignar	Valores enteros positivos
Retirar	Valores enteros positivos
Saldo corriente	Valores enteros y decimales

• Clase: Cuenta ahorro

Atributo	Valores posibles
Consignar	Valores enteros positivos
Retirar	Valores enteros positivos
Saldo ahorros	Valores enteros y decimales

• Clase: CDT

Atributo	Valores posibles
Abrir	Valores enteros entre 1 y 2
Cerrar	Valores enteros entre 1 y 2
Saldo CDT	Valores enteros y decimales

• Clase: Mes

Atributo	Valores posibles
Mes	Valores enteros positivos
Avanza Mes	Valores enteros entre 1 y 2
Total	Valores enteros y decimales

J. Elabore la Tarea No. 5 (pág 23), con el objetivo de reflexionar sobre Sobre el nivel de precisión de un algoritmo.

Conclusiones

- Los algoritmos deben ser claros y lo menos ambiguos posibles porque un algoritmo no es algo que puedas dejar a la interpretación, ya que son instrucciones y éstas deben ser lo más entendibles posibles para que quien las lee pueda seguir a la perfección.

¿Se presentan para que se interpreten de manera distinta?

- Aunque las instrucciones son claras tienen distintas formas de ser llevadas a cabo por ejemplo dice que en un mapa revise el metro al que me debo subir, pero no dice que tipo de mapa, si digital o físico y es verdad que son detalles menores pero esos detalles marcan la diferencia al final del día.

¿Estamos asumiendo 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 es bastante claro y entendible, entonces si, estamos asumiendo que quien lo lee y lo use tiene un mínimo de inteligencia o sentido común, pero existen miles de personas en el mundo y todas piensan diferente, alguna persona va a entender las instrucciones de manera distinta a la otra persona, pero todas deberían llegar al mismo resultado.

Objetivo:

- Entender la complejidad que tiene la tarea de escribir un algoritmo.

- Esta tarea es para ser desarrollada en parejas: (1) en el primer cuadrante haga un dibujo simple, (2) en el segundo cuadrante escucha las instrucciones para explicarle a la otra persona como hacer el dibujo, (3) lee las instrucciones a la otra persona, quien debe intentar seguir las sin ninguna ayuda adicional, (4) compare el dibujo inicial y el dibujo resultante.



ALGORITMO:

1. Dibuja la cabeza:
Haz un óvalo para la cabeza y divide la horizontal y vertical en el centro para dibujar la cara.
2. Dibuja el cuerpo:
Dibuja el torso con un rectángulo u otro óvalo y vuélvelo a dividir.
3. Dibuja los brazos y las piernas:
Empieza a agregar extremidades como brazos y piernas.
4. Agrega los detalles básicos:
Agrega la cara y los detalles de la persona como ropa.

El proceso de crear un logoritmo fácil y entendible fue más difícil de lo pensado, no por su extrema dificultad, sino que transmitir instrucciones a través de los puntos no es tarea fácil porque una malinterpretación puede arruinar o cambiar el resultado final

K. Estudia los siguientes aspectos del ejemplo seleccionado. Enunciado, requerimientos funcionales (clases del proyecto), luego, redacta el enunciado del problema y el nombre de los requerimientos funcionales del proyecto.

- **Enunciado:** Se requiere una aplicación que permita manejar la información de un empleado.
- Del empleado se maneja la siguiente información:
 - Nombre.
 - Apellido.
 - Género (femenino o masculino).
 - Fecha de nacimiento.
 - Foto.
 - Fecha de ingreso a la empresa.
 - Salario básico.

- La aplicación permite visualizar la información del empleado, y hacer los siguientes cálculos:

- Edad del empleado, utilizando la fecha de nacimiento del empleado.
- Antigüedad del empleado, utilizando la fecha de ingreso del empleado.
- Prestaciones a las que tiene derecho el empleado. Para este cálculo se debe usar la siguiente fórmula.

$$\text{Prestaciones} = (\text{antigüedad} * \text{salario}) / 12$$

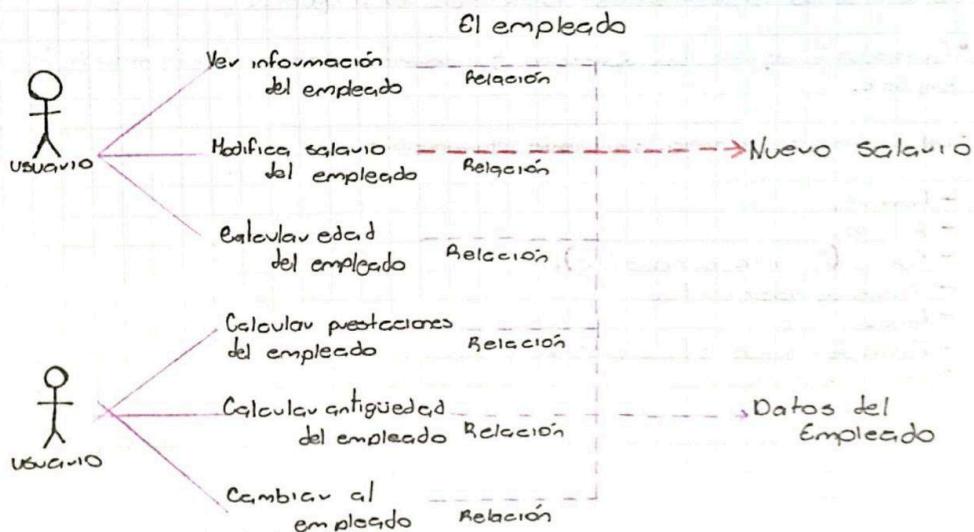
- El programa debe permitir:

- 1 Visualizar la información del empleado.
- 2 Modificar el salario del empleado.
- 3 Calcular la edad del empleado.
- 4 Calcular la antigüedad del empleado en la empresa.
- 5 Calcular las prestaciones del empleado.
- 6 Cambiar al empleado.

- Listado de requerimientos

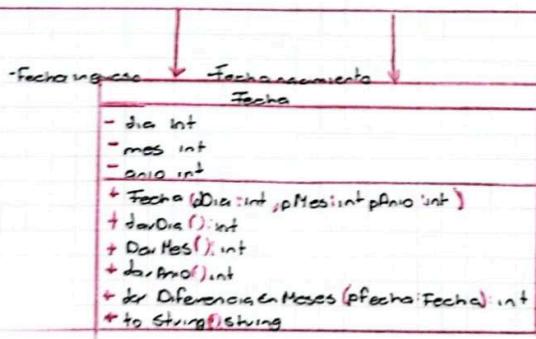
- Visualizar la información del empleado.
- Modificar el salario del empleado.
- Calcular la edad del empleado.
- Calcular la antigüedad del empleado.
- Calcular las prestaciones de un empleado.
- Cambiar al empleado.

t.



Empleados

```
- Nombre: String  
- Apellido: String  
- Genero: Int  
- Imagen: String  
- Salario: Double  
+ Empleado()  
+ darNombre(): String  
+ darGenero(): Int  
+ darFechaDeNacimiento(): String  
+ darImagen(): String  
+ darSalario(): Double  
+ darApellido(): String  
+ calcularEdad(): Int  
+ CalcularAntiguedad(): Int  
+ adquirirPrestaciones(): Double  
+ CambiarEmpleado(pNombre: String, pApellido: String, pGenero: Int, pfechaNacimiento: Fecha, pfechaIngreso: Fecha, pSalario: int,  
    pImagen: String) void  
+ cambiarSalario(sSalario: double) void  
+ darFechaActual(): Fecha  
+ Metodo1(): String  
+ Metodo2(): String
```



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

• Idea #1: Una página para organizar tareas para trabajadores o estudiantes.

Requerimiento funcional 1	Nombre	Creación de tareas.
	Resumen	El programa debe tener la capacidad de dejar al usuario crear y administrar las tareas; tales como poner descripciones, poner fechas límite, recordatorios, etc.
	Entradas	<ul style="list-style-type: none"> ✓ Id de tarea. ✓ Descripción de la tarea. ✓ Fecha límite. ✓ Recordatorio. ✓ Participantes
	Resultado	El programa registra la tarea en una tabla de fácil acceso para el programador y los involucrados con toda la información.
Requerimiento funcional 2	Nombre	Asignación automática de tareas.
	Resumen	El programa debe ser capaz de registrar roles para cada uno y dar información relevante según su rol.
	Entradas	<ul style="list-style-type: none"> ✓ Participantes. ✓ Id de roles (líder, programador, diseñador, tester, supervisores, etc.) ✓ Permisos de roles.
	Resultado	El programa permite asignar roles y permisos específicos para estos datos. La opción de mirar y editar lo que el rol de mayor cargo le dio permiso de mirar y editar.
Requerimiento funcional 3	Nombre	Reporte de rendimiento.
	Resumen	El sistema debe pedir de manera constante reportes de rendimiento a los participantes para ver sus distintas avances y mantener un ritmo de trabajo estable.
	Entradas	<ul style="list-style-type: none"> ✓ Id de usuario. ✓ Fecha límite para el reporte. ✓ Id de reporte completo. ✓ Id de tareas completas. ✓ Tareas pendiente.
	Resultado	El programa debe poder pedirles a los participantes reportes de desempeño y avisar sobre estos mismos a los roles encargados de supervisarlos.
Requerimiento funcional 4	Nombre	Notificaciones de fechas de vencimiento.
	Resumen	El programa debe enviar correos y notificaciones a los participantes sobre la fecha de vencimiento del proyecto o la tarea que se quiere realizar.
	Entradas	<ul style="list-style-type: none"> ✓ Id de la tarea. ✓ Fecha límite de la tarea. ✓ Método de envío de notificación (correo electrónico, notificación en la aplicación, SMS) ✓ Id del usuario.
	Resultado	El programa puede enviar estas notificaciones en tiempo real según el estado de la tarea mientras está este incompleta.

• Idea #2: Base de datos de una biblioteca

	Nombre	Adición de libro.
Requerimiento Funcional 1	Resumen	El programa debe ser capaz de añadir un libro a la base de datos, basándose en diferentes aspectos de dicho libro, como el autor, género, id del libro y lugar físico donde se encuentra.
	Entradas	<ul style="list-style-type: none"> ✓ Nombre del libro. ✓ Nombre del autor. ✓ Género del libro. ✓ Estantería del libro. ✓ Id del libro.
	Resultado	El programa añade el libro a la base de datos.
	Nombre	Prestamo de libros.
Requerimiento Funcional 2	Resumen	El programa debe registrar cuando se prestó un libro de la biblioteca, asignar un plazo para la devuelta del libro y relacionarlo con la persona que lo pidió prestado.
	Entradas	<ul style="list-style-type: none"> ✓ Id del usuario. ✓ Id del libro. ✓ Fecha de préstamo. ✓ Fecha de entrega.
	Resultado	El programa asocia y registra el libro como prestado asociándolo a la id del usuario y mostrando la fecha de préstamo y entrega.
	Nombre	Retiro del libro.
Requerimiento Funcional 3	Resumen	El programa debe retirar un libro de la base de datos eliminando la entrada.
	Entradas	<ul style="list-style-type: none"> ✓ Id del libro. ✓ Motivo del libro. ✓ Fecha del retiro.
	Resultado	El programa retira al libro de la base de datos y muestra en pantalla el motivo y fecha del retiro.
	Nombre	Registro de identificaciones.
Requerimiento Funcional 4	Resumen	El programa debe ser capaz de añadir usuarios a su base de datos, asignándoles un id único y reconociéndolos como capaces de pedir préstamos de ciertos libros, entonces permitiéndoles pedir préstamos de ciertos libros en base a su edad.
	Entradas	<ul style="list-style-type: none"> ✓ Nombre del usuario. ✓ Número de identificación personal del usuario. ✓ Edad del usuario.
	Resultado	El programa debe añadir al usuario como nueva entrada en la base de datos, proporcionándole una id única e imponible, mostrándole a qué tipos de libros tiene acceso de préstamo.