

PROYECTO

MARIA JULIANA PEÑA MENDEZ
CARLOS SEBASTIAN NARANJO ARIAS.
ESTUDIANTE.

JUAN CARLOS MARTINEZ DIAZ
PROFESOR

LOGICA COMPUTACIONAL.
UNIVERSIDAD ANTONIO
NARINO.

2024.

ACTIVIDADES PRELIMINARES.

Contexto:

La especificación de un problema:

A. Resuma el ciclo de vida de construcción de un programa.

El ciclo de vida se puede resumir en 3 pasos los cuales son:

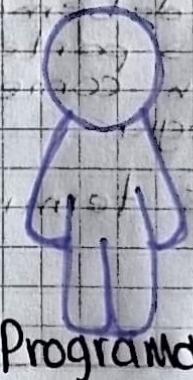
- Una organización o persona, denominada cliente, tiene un problema y necesita la construcción de un programa para resolverlo. Para esto contacta una empresa de desarrollo de software que pone a su disposición un programador.
- El programador sigue un conjunto de etapas, denominadas el proceso, para entender el problema del cliente y construir de manera organizada una solución de buena calidad, de la cual formara parte un programa.
- El programador instala el programa que resuelve el problema en un computador y deja que el usuario lo utilice para resolver el problema. Fíjese que no es necesario que el cliente y el usuario sean la misma persona. Piense por ejemplo que el cliente puede ser el gerente de producción de una fábrica y el usuario, un operario de la misma.

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

PROBLEMA → ANALISIS → Requerimientos funcionales.



cliente



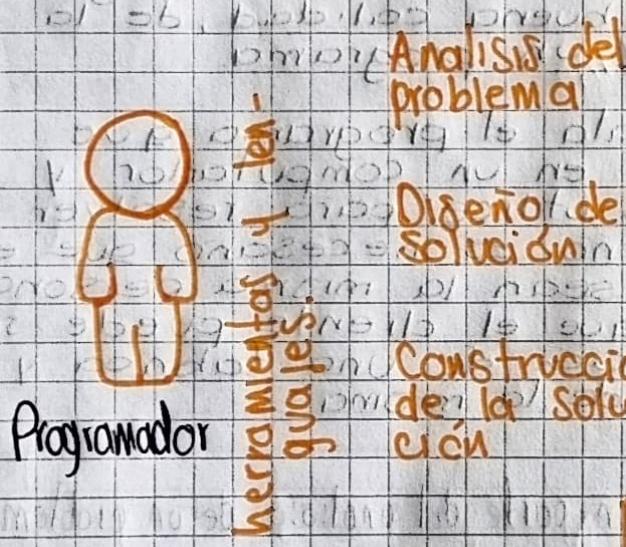
Programador.

→ Mundo del problema.

→ Requerimientos no funcionales

- * Analizar un problema es tratar de entenderlo. Esta etapa busca soluciones que no tratemos de resolver un problema diferente al que tiene el cliente.
- * Descomponer el problema en sus tres aspectos fundamentales, facilita la tarea de entenderlo: en cada etapa, nos podemos concentrar en solo uno de ellos, lo cual simplifica el trabajo.
- * Esta descomposición se puede generalizar para resolver todo tipo de problemas, no solo se utiliza en problemas cuya solución sea un programa de computador.
- * Además, de entender el problema, debemos expresar lo que entendemos siguiendo algunas convenciones.
- * Al terminar la etapa de análisis debemos generar un conjunto de documentos que contengan nuestro comprensión del problema. Con dichos documentos podemos validar nuestro trabajo, presentándoselo al cliente y discutiendo con él.

C.



Programador

heurísticas

Problema

Análisis del problema

Diseño de la solución

Construcción de la Solución

Solución

• La primera etapa para resolver un problema es analizarlo. Para facilitar este estudio, se debe descomponer el problema en 3 partes.

• Una vez que el problema se ha entendido y se ha expresado en un lenguaje que pueda entender sin ambigüedad, pasamos a la

etapa de diseño. Aquí debemos imaginarnos la solución y definir las partes que la van a componer. Es muy común comenzar esta etapa definiendo una estrategia.

• Cuando el diseño está terminado, pasamos a construir la solución.

d. Cuales son los elementos que se deben entregar a un cliente?

la solución de un problema tiene 3 partes; (1) el diseño, (2) el programa y (3) las pruebas de corrección del programa. Estos son los elementos que se deben entregar al cliente. Es común que, además de los 3 elementos citados anteriormente, la solución incluya un manual del usuario, que explique el funcionamiento del programa.

e. Elabore la tarea No.1 con el objetivo de identificar los aspectos que forman parte de un problema.

Problema Tarea No 1: 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.

| | |
|----------------------------|--|
| Cliente | Banco |
| Usuario | Usuarios de los cajeros automáticos. |
| Requerimiento funcional | Manejar sus cajeros para retirar y consultar saldo |
| Mundo del problema | Manejar todos los cajeros automáticos del banco |
| Requerimiento no funcional | El número de cajeros automáticos del banco |

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

| | | |
|--|--------|--|
| | Nombre | Visualizar el estado de la cuenta corriente del cliente. |
| | | |

| | | |
|-------------------------------|------------|---|
| Requerimiento funcional 1 | Resumen | Visualiza el saldo de la cuenta corriente del cliente. |
| | Entradas | Ninguna. |
| | Resultados | Se muestra el saldo de la cuenta corriente. |
| | Nombre. | Visualizar el saldo de la cuenta de ahorros del cliente. |
| Requerimiento funcional 2. | Resumen | Visualiza el saldo de la cuenta de ahorros del cliente |
| | Entradas | Ninguna |
| | Resultados | se muestra el saldo de la cuenta de ahorros y el interés de la cuenta |
| | Nombre. | Visualizar el saldo del CDT del cliente |
| Requerimiento funcional 3. | Resumen | Visualiza el Saldo del CDT por el cliente |
| | Entradas | Ninguna |
| | Resultados | se muestra el saldo del CDT y el interés del mismo |

Elabore la tarea N°1, con el objetivo de identificar los requerimientos funcionales de un problema.

| | | |
|--------------------------|------------|---|
| | Nombre. | Dibujar el triángulo en la interfaz. |
| Requerimiento funcional | Resumen | Dibuja el triángulo en la interfaz principal |
| 1 | Entradas | Ninguna |
| | Resultados | Se pinta el triángulo en la interfaz principal |
| | Nombre. | Combinar los puntos de triángulo |
| | Resumen | Cambios de puntos del triángulo |
| Requerimiento funcional | Entradas | Coordenadas x y y del P1 del triángulo Coordenadas x y y del punto 2 del triángulo Coordenadas x y y del punto 3 del triángulo |
| 2. | Resultados | Se pinta el triángulo con los nuevos puntos, se actualiza el área, el perímetro y la altura del triángulo. Si los nuevos puntos del triángulo son co-lineales se muestra un mensaje de error |
| | Nombre | Cambiar color relleno del triángulo |
| Requerimiento funcional. | Resumen | Cambia el color de relleno del triángulo. |

3.

Entradas Nuevo color de relleno

Resultados Se pinta triángulo con el nuevo color de relleno

h. Elabore la tarea No. 4, con el objetivo de identificar las entidades del mundo del problema.

| Nombre. | Descripción |
|---------|----------------------|
| Entidad | x y y |
| Entidad | Red Green Blue |
| Entidad | Lineas |

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

No se podía ejecutar el programa, ya que no contaba con todas las entidades diferentes

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

Es una entidad porque hace parte del mundo del problema.

i. Elabore la tarea No.5, con el objetivo de identificar las entidades de un caso de estudio

Clase: CuentaBancaria.

I(1) : 5/20/1

Atributos.

Valores posibles

Número de cuenta.

entre 0 y 9.

CuentaBancaria

de cuenta

Saldo Disponible.

entre 0 y 9.

Saldo disponible

Pin o clave.

entre 0 y 9.

Pin / clave.

Clase: CuentaCorriente.

Atributos.

Valores posibles.

CuentaCorriente.

Titular

Cadena de caracteres.

Titular

de cuenta

Valores numéricos

de cuenta

de cedula
titular.

Valores numéricos

de cedula

titular

Clase: CuentaAhorros.

Atributos

Valores Posibles

CuentaAhorros

Titular

Cadena de caracteres.

Titular

Ingreso
de dinero

Valores
numéricos

Ingreso de dinero

Retiro de
dinero.

Valores
numéricos.

Retiro de dinero

clase : CDT.

Atributos.

Cantidad de dinero

Tiempo determinado

Dinero disponible.

Valores posibles

Valores numéricos

Valores numéricos

Valores numéricos

CDT

Cantidad de dinero

Tiempo determinado.

Dinero disponible.

Clase: Mes.

Atributo. Valores posibles

Mes

Valores numéricos.

Mes

Mes

Día

Valores numéricos.

Día

Año.

Año

Valores numéricos.

Año

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

Tarea 6 = Suponga que usted es la persona que va a utilizar el algoritmo anterior, pero impáuse en el metro de París. Identifique qué problemas podría tener con las instrucciones anteriores. Piense por ejemplo si están completas.

¿Se prestan para que se interpreten de maneras distintas? ¿Estamos suponiendo que quién lo lee va a "entender común", o cualquier persona que lo use va a resolver siempre el problema de la misma manera?

Reto:

- Compra de billetes = No especifica cómo compra el billete (é en efectivo, tarjeta?), ni qué tipo de billete se debe comprar. Se podrían incluir instrucciones sobre, como realizar la compra, qué formas de pago son aceptadas, y qué tipo de billete es el más adecuado según el viaje.
- Identificación en el mapa = La instrucción "identifique en el mapa del metro la estación donde está y a punto donde necesita ir" es algo general y podría ser confusa para alguien que no esté familiarizada con la disposición del mapa. Se podrían incluir detalles adicionales como identificar las líneas del metro (colores, números a letras) y cómo están organizados en el mapa.
- Localización de la estación más cercana al destino = No se menciona si la persona tiene algún mapa o aplicación para encontrar la estación más cercana al destino. Puedo no saber cómo hacerlo si no estoy familiarizado con la ciudad. Sugerir el uso de aplicaciones o mapas digitales, o cómo puede encontrar esta información en las estaciones del metro.
- Verificación líneas = Se dice "verifique si desde donde está hay alguna línea que pase por la estación de destino", pero no menciona qué hacer si no hay una línea directa o si es necesario hacer tránsitos entre líneas. Incluir instrucciones sobre como hacer tránsitos, qué estaciones permiten cambio de líneas y como identificar el orden correcto si hay varias líneas.
- Subido al metro = Se indica subir al metro en el orden de la línea identificada, pero no se especifica como verificar la dirección correcta del tren, ya que las líneas suelen tener dos direcciones opuestas. Aclarar que es importante verificar la dirección del tren en función de la última estación de la línea, lo cual se indica en los ordenes.

El algoritmo cubre los pasos generales, pero faltan detalles que podrían ser críticos para que alguien no se familiarice con el sistema de transporte parisino. Podría mejorarse agregando pasos más detallados, haciendo suposiciones simplificadas y proporcionando alternativas en caso de error.

ACTION =

Tiempo preliminar: Caso de estudio.

K. Estudia los siguientes aspectos del ejemplo seleccionado: Enunciado, Requerimientos funcionales (caso de uso) y el Modelo (clases de proyecto). A continuación, redacta el enunciado del problema y el nombre de cada uno de los requerimientos funcionales del proyecto.

Punto =

- Enunciado = Se quiere 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 lo siguiente fórmula.

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

• Listado requerimientos:

1. R1 - Visualizar la información del empleado
2. R2 - Modificar el salario del empleado
3. R3 - Calcular la edad del empleado
4. R4 - Calcular la antigüedad del empleado
5. R5 - Calcular las prestaciones del empleado
6. R6 - Cambiar el empleado.

1. Dibujo el respectivo diagrama de casos de uso del ejemplo elegido (consulta "requerimientos funcionales" en "Archivo").

Diagrama clases =

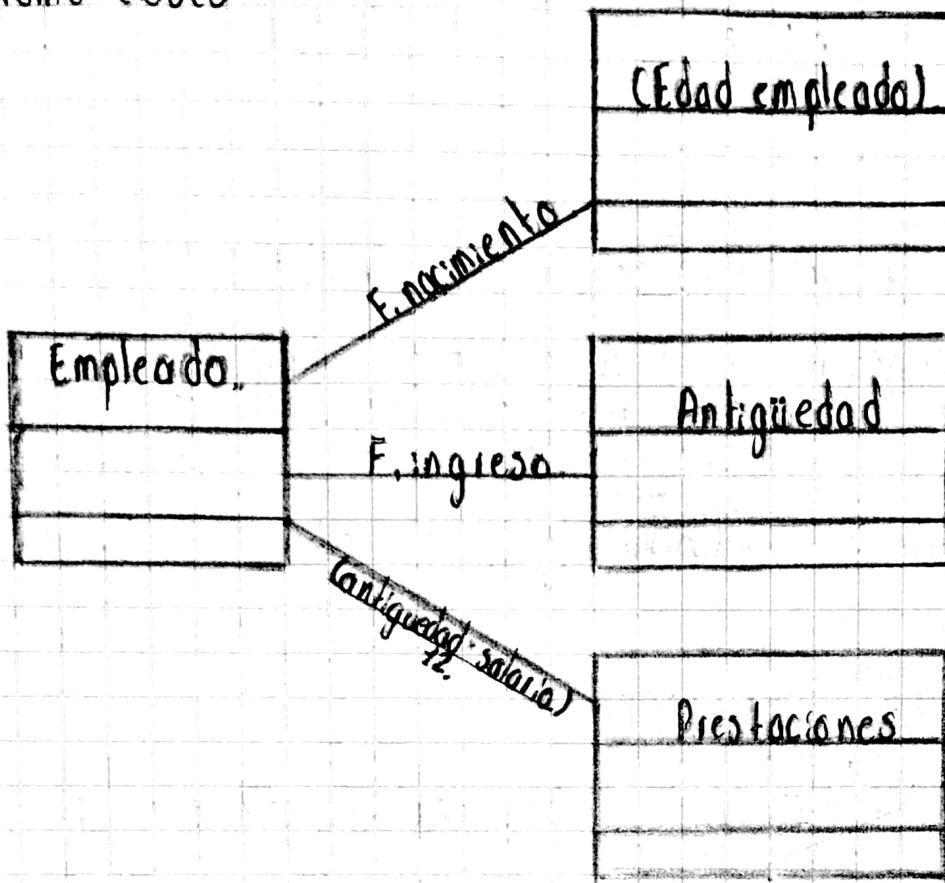
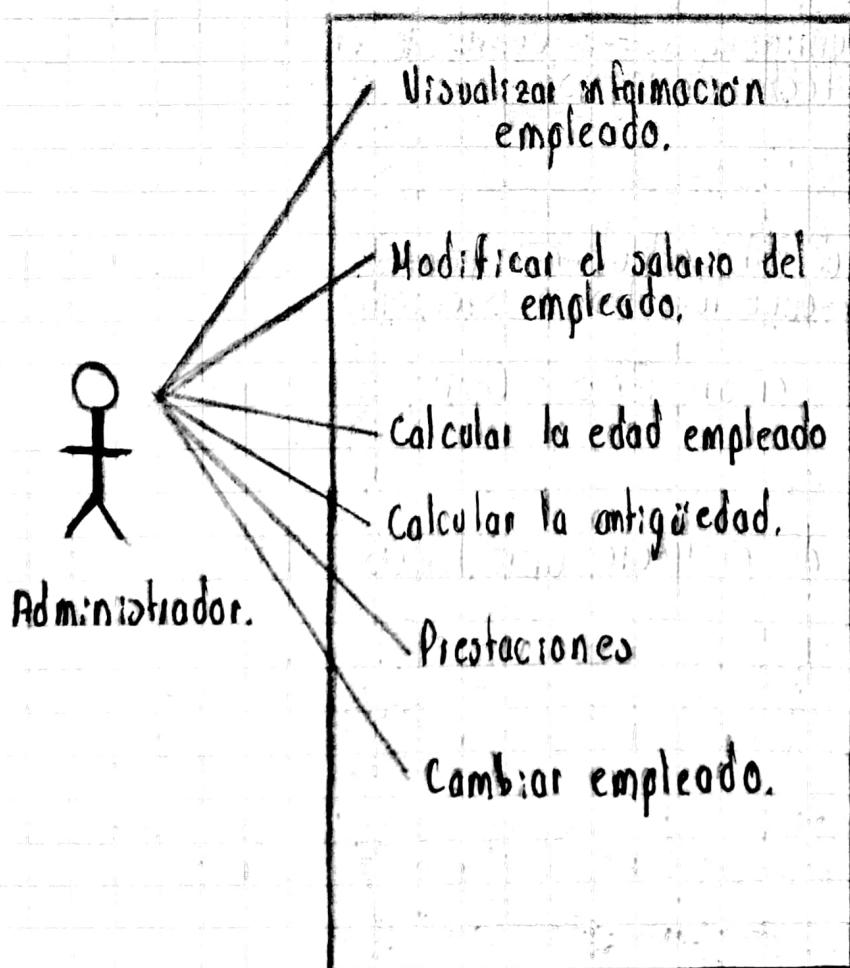


Diagrama casos de uso =



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

Rta =

los clases que se usan en el problema son los siguientes =

- * Edad del empleado, utilizando la fecha de nacimiento del empleado.
- * Antigüedad del empleado, utilizando la fecha de ingreso del empleado.
- * Prestaciones o las que tiene derecho el empleado.

$$\text{Prestaciones} = (\text{Antigüedad} \times \text{Salario}) / 12$$

- Edad =

- * Atributos = Fecha.nacimiento = dato
- * Funciones = Calcula edad - devuelve la edad del empleado calculada a partir de la fecha.nacimiento.

- Antigüedad =

- * Atributos = Fecha.ingreso = dato
- * Funciones = Calcula antigüedad - devuelve la antigüedad del empleado calculado a partir de la fecha de ingreso.

- Prestaciones =

- * Atributos = Puede ser un valor calculado a partir de la clase antigüedad.
- * Funciones = Calcula y devuelve las prestaciones utilizando la formula.

$$\text{Prestaciones} = (\text{Antigüedad} \times \text{Salario}) / 12$$

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

Idea 1: Sistema de gestión de tareas. Creado para ayudar a los usuarios a organizar y gestionar sus tareas diarias de manera eficiente. Pueden crear, asignarles fechas de vencimiento y prioridades, y realizar seguimiento de su progreso.

Idea 2: Aplicación seguimiento de hábitos. La aplicación tiene como objetivo ayudar a los usuarios a establecer y mantener hábitos positivos en su vida diaria.

| | | |
|-------------------------|------------|---|
| Requerimiento Funcional | Nombre. | Crear Tarea |
| | Resumen | Permite a los usuarios añadir una nueva tarea a su lista. |
| | Entradas. | Título de la tarea, descripción, fecha de vencimiento. |
| | Resultado. | La tarea se añade a la lista de tareas del usuario. |
| Requerimiento Funcional | Nombre | Actualizar Tarea |
| | Resumen | Permite a los usuarios modificar los detalles existentes. |
| | Entradas | ID de la tarea, nuevos valores para el título, descripción. |
| | Resultados | La tarea se actualiza con la nueva información. |
| Requerimiento Funcional | Nombre | Eliminar Tarea. |
| | Resumen | Permite a los usuarios eliminar una tarea de su lista. |
| | Entradas | ID de la tarea. |
| | Resultados | La tarea es eliminada de la lista del usuario. |
| Requerimiento Funcional | Nombre | Listar Tareas |
| | Resumen | Muestra todos las tareas del usuario, con opción filtrado. |
| | Entradas | Filtros de entrada (opcional). |
| | Resultados | Se muestra una lista de tareas según los filtros aplicados. |

| | | |
|-------------------------|-----------|---|
| Requerimiento Funcional | Nombre | Añadir hábito |
| | Resumen | Permite a los usuarios registrar un nuevo hábito. |
| | Entradas | Nombre del hábito, frecuencia (diaria, semanal, mensual). |
| | Resultado | El hábito se añade a la lista de hábitos del usuario. |
| Requerimiento Funcional | Nombre | Marcar Hábito como completado |
| | Resumen | Permite a los usuarios marcar un hábito como completado. |
| | Entradas | ID del hábito, fecha. |
| | Resultado | Se actualiza el registro del hábito para indicar completado. |
| Requerimiento Funcional | Nombre | Visualizar progreso |
| | Resumen | Usuarios ver su progreso o seguimiento a lo largo del tiempo. |
| | Entradas | ID del hábito |
| | Resultado | Se genera un gráfico o una lista que muestra el progreso. |
| Requerimiento Funcional | Nombre | Eliminar hábito |
| | Resumen | Permite a los usuarios eliminar el hábito de su lista. |
| | Entrada | ID del hábito a eliminar. |
| | Resultado | El hábito es eliminado de la lista del usuario. |