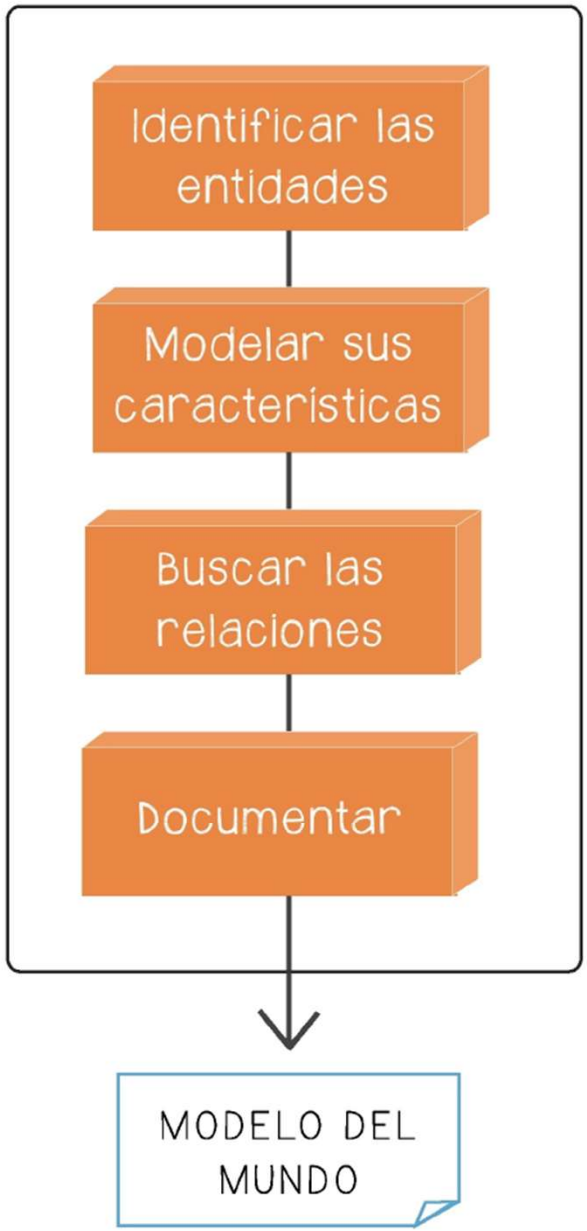
Modelo del mundo del problema



Modelo del mundo del problema

Identificar las entidades o clases

En programación orientada a objetos “POO”, las entidades del mundo se denominan clases, y serán los elementos básicos del diseño y la posterior implementación.

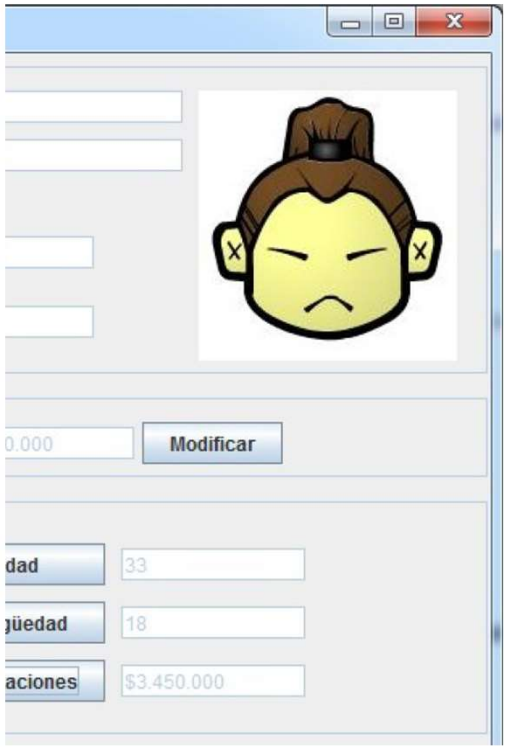
Ejemplo de entidades del simulador bancario

Entidad	Descripción
CuentaBancaria	Es la entidad más importante del mundo del problema, puesto que define su frontera (todo lo que está por fuera de la cuenta bancaria no nos interesa). Es buena práctica comenzar la etapa de análisis tratando de identificar la clase más importante del problema. Cuando el nombre de la entidad es compuesto, se usa por convención una letra mayúscula al comienzo de cada palabra. En otra época se utilizaban el carácter "_" para separar las palabras (Cuenta_Bancaria) pero eso está pasado de moda.
CuentaCorriente	Este es otro concepto que existe en el mundo del problema. Según el enunciado una cuenta corriente forma parte de una cuenta bancaria, luego esta entidad está "dentro" de la frontera que nos interesa. Por ahora no nos interesan los detalles de la cuenta corriente (por ejemplo si tiene un saldo o si paga intereses). En este momento sólo queremos identificar los elementos del mundo del problema que están involucrados en los requerimientos funcionales.
CuentaAhorros	Este es el tercer concepto que aparece en el mundo del problema. De la misma manera que en el caso anterior, una cuenta bancaria "incluye" una cuenta de ahorros. Los nombres asignados a las clases deben ser significativos y dar una idea clara de la entidad del mundo que representan. No se debe exagerar con la longitud del nombre, porque de lo contrario los programas pueden resultar pesados de leer.
CDT	El nombre de esta clase se encuentra en mayúsculas, porque es una sigla. Otro nombre para esta clase habría podido ser el nombre completo del concepto: CertificadoDepositoTermino. En el lenguaje Java no es posible usar tildes en los nombres de los clases, así que nunca veremos una clase llamada CertificadoDepósitoTérmino.
Mes	El concepto de mes es el quinto y último concepto del caso de estudio. Esta será la clase que nos dirá en cuál mes de la simulación se encuentra la cuenta bancaria. No olvide que la simulación se hace mes a mes, y que existe un requerimiento funcional que permite avanzar un mes en la simulación. Si lee de nuevo el enunciado del caso, se dará cuenta de que todos los demás elementos del problema son características de las entidades anteriormente mencionadas o requerimientos funcionales.

Modelar características de las entidades

Ejemplo Atributos de la Clase Empleado

Atributo	Valores Posibles	Comentarios
nombre	Cadena de caracteres	La primera característica que aparece en el enunciado es el nombre del empleado. El valor de este atributo es una cadena de caracteres (por ejemplo, "Juan") Seleccionamos "nombre" como nombre del atributo. Es importante que los nombres de los atributos sean significativos (deben dar una idea clara de lo que una característica representa), para facilitar así la lectura y la escritura de los programas.
apellido	Cadena de caracteres	El segundo atributo es el apellido del empleado. Al igual que en el caso anterior, el valor que puede tomar este atributo es una cadena de caracteres (por ejemplo, "Pérez"). Como nombre del atributo seleccionamos "apellido". El nombre de un atributo debe ser único dentro de la clase (no es posible dar el mismo nombre a dos atributos).
sexo	Masculino o Femenino	Esta característica puede tomar dos valores: masculino o femenino. En esta etapa de análisis basta con identificar los valores posibles. Es importante destacar que los valores posibles de este atributo (llamado "sexo") no son cadenas de caracteres. No nos interesan las palabras en español que pueden describir los valores posibles de esta característica, sino los valores en sí mismos.
salario	Valores enteros positivos	El salario está expresado en pesos y su valor es un número entero positivo. Aquí estamos suponiendo que el valor del salario no tiene centavos.



Atributos de la clase Fecha

Atributo	Valores Posibles	Comentarios
dia	Valores enteros entre 1 y 31	La primera característica de una fecha es el día y puede tomar valores enteros entre 1 y 31. En los nombres de las variables no puede haber tildes, por lo que debemos contentarnos con el nombre "dia" (sin tilde) para el atributo.
mes	Valores enteros entre 1 y 12	La segunda característica es el mes. Aquí se podrían listar los meses del año como los valores posibles (por ejemplo, enero, febrero, etc.), pero por simplicidad vamos a decir que el mes corresponde a un valor entero entre 1 y 12.
anio	Valores enteros positivos	La última característica es el año. Debe ser un valor entero positivo (por ejemplo, 2001). Aquí nos encontramos de nuevo con un problema en español: los nombres de los atributos no pueden contener la letra "ñ". En este caso resolvimos reemplazar dicha letra y llamar el atributo "anio" que da aproximadamente el mismo sonido. Con las tres características anteriores queda completamente definida una fecha. Esa es la pregunta que nos debemos hacer cuando estamos en esta etapa: ¿es necesaria más información para describir la entidad que estamos representando? Si encontramos una característica cuyos valores posibles no son simples, como números, cadenas de caracteres, o una lista de valores, nos debemos preguntar si dicha característica no es más bien otra entidad que no identificamos en la etapa anterior. Si es el caso, simplemente la debemos agregar.

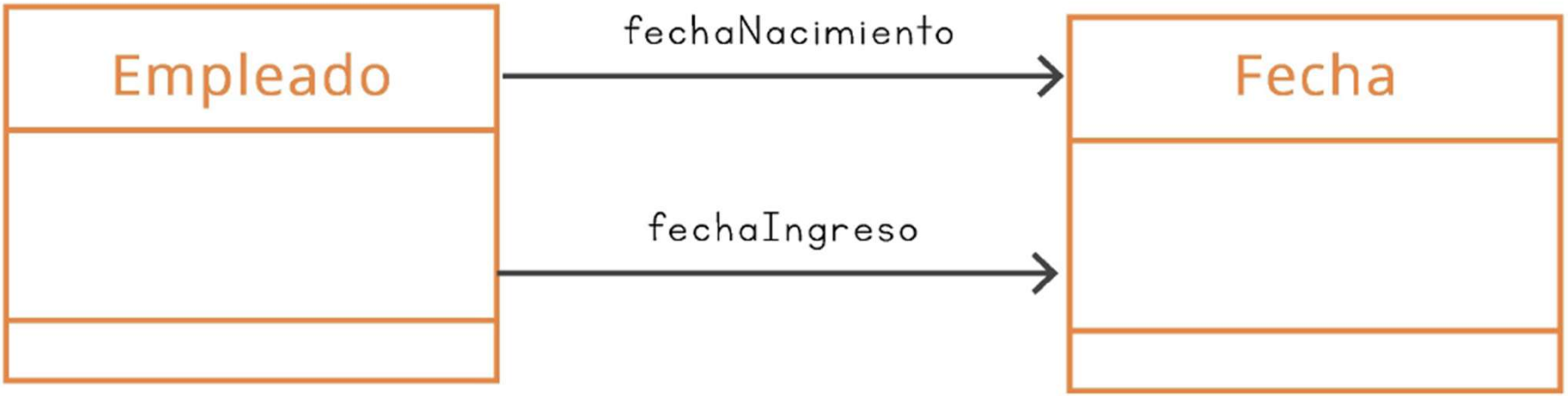
Relaciones entre clases

Debemos tratar de identificar las relaciones que existen entre las distintas entidades del mundo y asignarles un nombre.

Las relaciones se representan en UML (Lenguaje de Modelado Unificado) como flechas que unen las cajas de las clases y se denominan usualmente asociaciones.

El diagrama de clases en el cual se incluye la representación de todas las entidades y las relaciones que existen entre ellas se conoce como el modelo conceptual, porque explica la estructura y las relaciones de los elementos del mundo del problema.

Relaciones entre clases



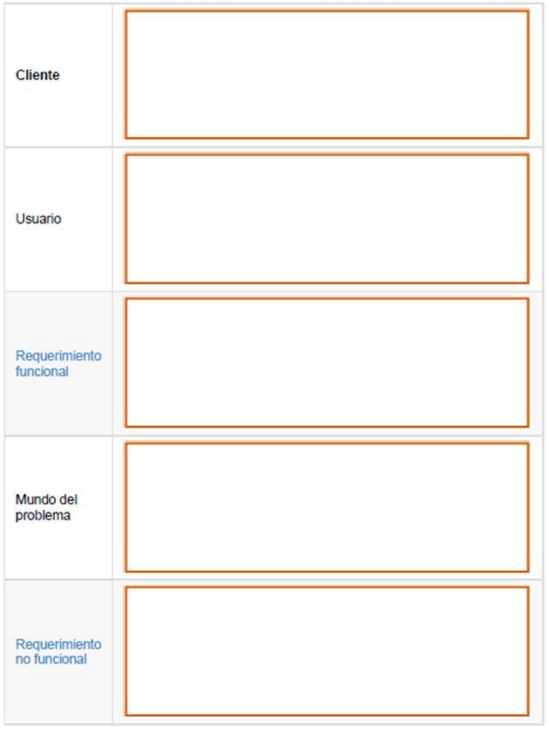
Ejercicio 1.

El problema: Un banco quiere crear un programa para manejar sus cajeros automáticos.

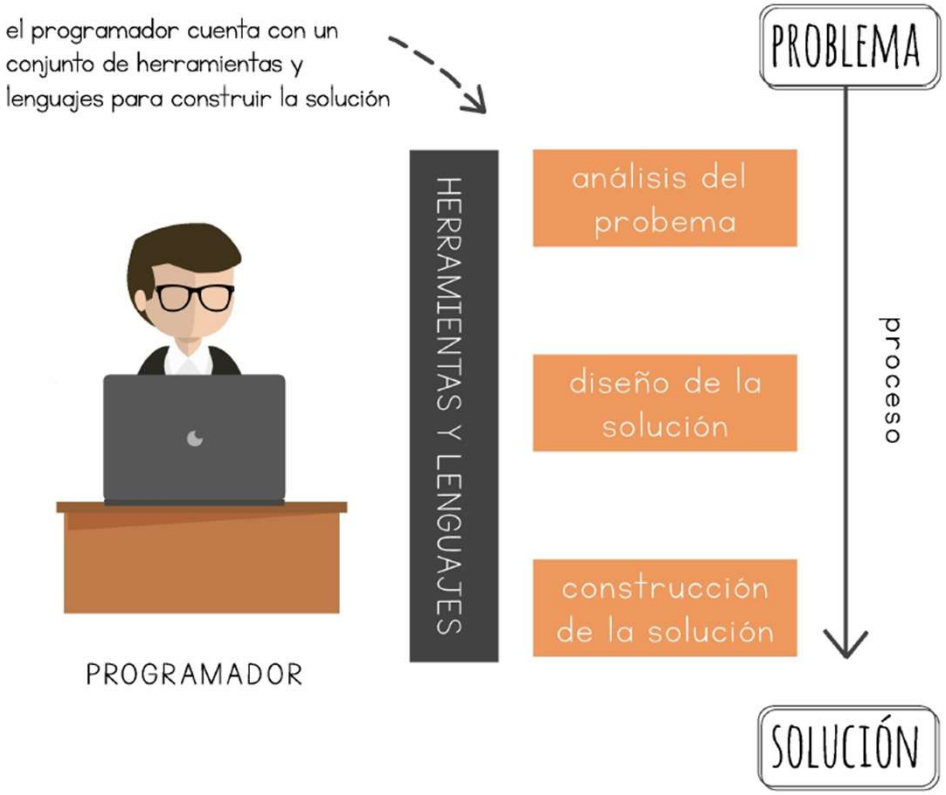
Dicho programa sólo debe permitir retirar dinero y consultar el saldo de una cuenta.

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.

Desarrollo


Cliente
Usuario
Requerimiento funcional
Mundo del problema
Requerimiento no funcional

Proceso y Herramientas



Proceso y Herramientas

Análisis del problema: El objetivo de esta etapa es entender y especificar el problema que se quiere resolver. Al terminar, deben estar especificados los requerimientos funcionales, debe estar establecida la información del mundo del problema y deben estar definidos los requerimientos no funcionales.

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. Los diseños nos van a permitir mostrar la solución antes de comenzar el proceso de fabricación propiamente dicho. Es importante destacar que dicha especificación es parte integral de la solución.

Construcción de la solución: Tiene como objetivo implementar el programa a partir del diseño y probar su correcto funcionamiento.

OJO!

Las etapas iniciales del proceso de construcción de un programa son críticas, puesto que cuanto más tarde se detecta un error, más costoso es corregirlo. Un error en la etapa de análisis (entender mal algún aspecto del problema) puede implicar la pérdida de mucho tiempo y dinero en un proyecto. Es importante que al finalizar cada etapa, tratemos de asegurarnos de que vamos avanzando correctamente en la construcción de la solución.

DISEÑO DE LA SOLUCIÓN

Algoritmo

conjunto de reglas operacionales inherentes a un cómputo". Se trata de un método sistemático, susceptible de ser realizado mecánicamente, para resolver un problema dado.

Conjunto de pasos lógico, coherente y ordenado que nos permite dar solución a un problema específico.

Algoritmo

Datos de entrada: un algoritmo tiene cero o más entradas, es decir la información que recibe el algoritmo al comenzar o, dinámicamente, mientras el algoritmo se ejecuta.

Procesamiento de datos: incluye las operaciones aritmético-lógicas, cuyo objetivo es obtener la solución del problema.

Salida de resultados: permite comunicar al exterior el resultado. Un algoritmo puede producir una o más salidas.

Características de un algoritmo

Entrada: definir lo que necesita el algoritmo

Salida: definir lo que produce.

No ambiguo: explícito, siempre sabe qué comando ejecutar.

Finito: El algoritmo termina en un número finito de pasos.

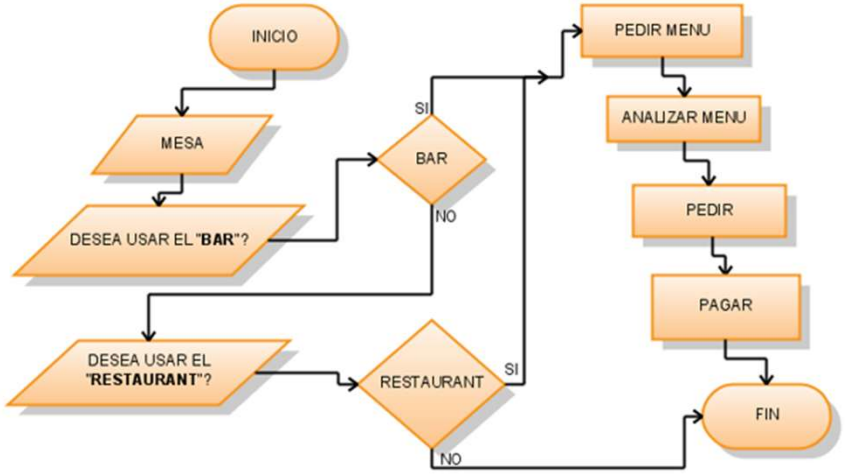
Correcto: Hace lo que se supone que debe hacer. La solución es correcta

Efectividad: Cada instrucción se completa en tiempo finito. Cada instrucción debe ser lo suficientemente básica como para que en principio pueda ser ejecutada por cualquier persona usando papel y lápiz.

General: Debe ser lo suficientemente general como para contemplar todos los casos de entrada.

Diagramas de Flujo de Datos

Es la representación gráfica secuencial de un algoritmo o proceso donde se detallan los alcances, los actores, los puntos de decisión, las actividades y los posibles errores de flujo. Se utiliza en disciplinas como programación, economía, procesos industriales y psicología cognitiva



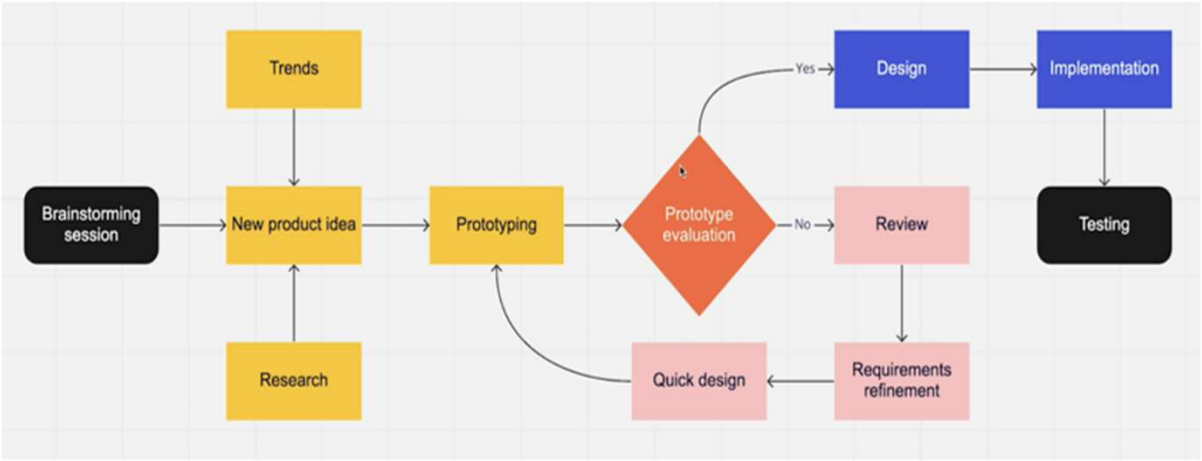
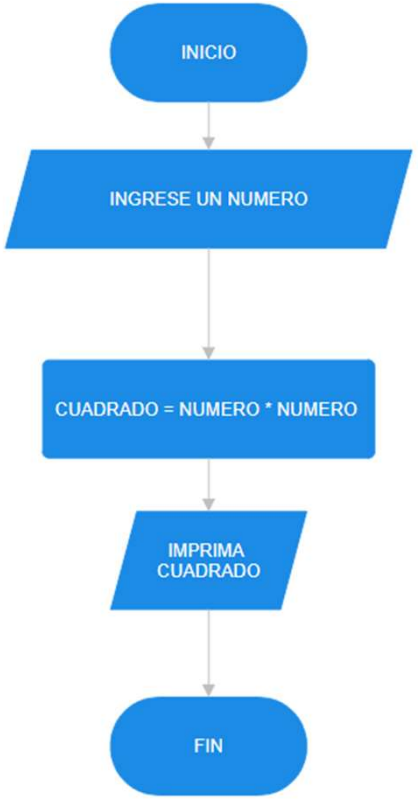


Diagrama de Flujo de Datos DFD

Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Linea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso



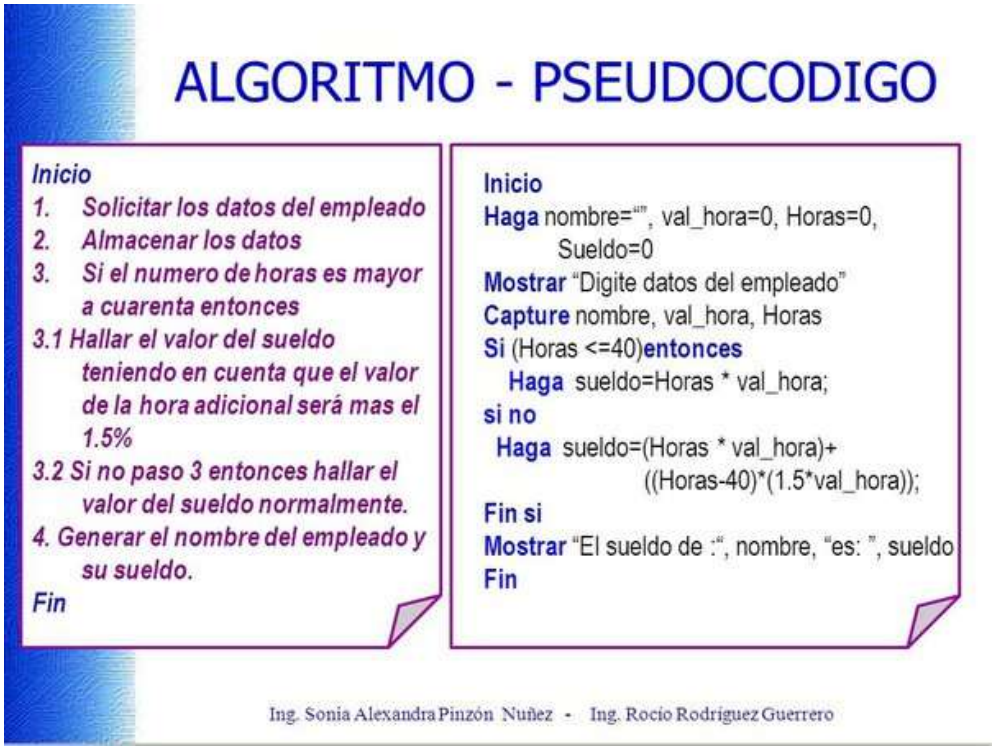
Tomado de: https://www.smartdraw.com/flowchart/simbolos-de-diagramas-de-flujo.htm

Pseudocódigo

```
1  Algoritmo PAR_IMPAR
2      Escribir 'Por favor, teclea un número'
3      Leer numero
4      Si numero MOD 2=0 Entonces
5          Escribir 'El número, ',numero,' es par'
6      SiNo
7          Escribir 'El número, ',numero,' es impar'
8      FinSi
9  FinAlgoritmo
```

```
Proceso Suma

    Definir A,B,C como Reales;

    Escribir "Ingrese el primer numero:";
    Leer A;

    Escribir "Ingrese el segundo numero:";
    Leer B;

    C <- A+B;

    Escribir "El resultado es: ",C;

FinProceso
```

Algoritmo vs pseudocódigo



Pre y postcondiciones de un algoritmo

- **Precondición**: es la información que se conoce como verdadera antes de comenzar el algoritmo.

- **Postcondición**: es la información que se conoce como verdadera después de finalizado el algoritmo, siempre que se cumpla con las precondiciones.

Ejemplo 1

Determinar el resto de la división entera entre dos números enteros N y M.

Precondición: N y M son números enteros. M debe ser distinto de 0.

Postcondición: el resultado será un valor entero comprendido entre 0 y M-1, que representa el resto de aplicar la división entera entre N y M.

Ejemplo 2

Determinar si el número 437 es primo.

Algoritmo: Dividir el número 437 entre cada uno de los números 1, 2, 3, 4,... 436. Si una de las divisiones es exacta (resto 0), entonces el número 437 no es primo, caso contrario es primo.

Precondición: No hay ningún requerimiento.

Postcondición: Se ha podido determinar si el número 437 es primo o no.

Referencias

Villalobos J. Fundamentos de Programación. Universidad de los Andes

Ruiz V. Manual de análisis y diseño de algoritmos. INACAP

Cisco. Fundamentos de Python 1.

Cartagena99. Fundamentos de programación

https://view.genially.com/65168bf30af28d0011da791d/horizontal-infographic-diagrams-proceso-de-compilacion

https://colab.research.google.com/drive/1xcBPcHlVT5gPOe2xPNYjcAPGDS1HKxa3#scrollTo=giq15p_ANzE_

https://intef.es/observatorio_tecno/pseint-programando-en-pseudocodigo/

https://basesdeda2s.data.blog/2019/03/26/tipos-de-datos-y-expresiones/

https://platzi.com/clases/3221-pensamiento-logico/50678-operadores-aritmeticos/

https://colab.research.google.com/drive/1ytmjdw3qM29er_4pUxbadJtC5nSXmVrX?usp=sharing