

UNIDAD 1: INTRODUCCIÓN A LOS ALGORITMOS Y MANEJO DE ARREGLOS

PRIMERA SEMANA



Mi vida es un algoritmo





Temario

- Concepto de algoritmo.
- Notación algorítmica.
- Clases, objetos y métodos.
- Estructuras de datos.

Concepto de Algoritmo

Es un conjunto ordenado y finito de operaciones que permite hallar la solución a un problema.



Concepto de Algoritmo

Reglas para efectuar algún cálculo, bien sea a mano o, más frecuentemente, en una máquina.

Ejemplos:

- Sumas, restas, multiplicación y división
- [Devocionario Anglicano](#)
- Algoritmo de euclides



Muhammad ibn Musa al-Khwarizmi
Matemático Persa del Siglo IX



Concepto de Algoritmo

- La ejecución de un algoritmo no implica una decisión subjetiva, o hacer uso de la intuición y creatividad.
 - ¿Una receta de cocina es un algoritmo?
 - ¿Procedimientos que efectúen elecciones aleatorias son algoritmos?
 - Algoritmos probabilísticos





Concepto de Algoritmo

- Cuando se utiliza un algoritmo para calcular la respuesta de un problema concreto, lo normal es suponer que las reglas nos darán, si se aplican correctamente, la respuesta correcta.
 - Sin embargo en algunas circunstancias usamos ***algoritmos aproximados***.
 - Se especifica el error que estamos dispuestos a tolerar.
 - O también, ***Algoritmos heurístico*** o ***heurística***.
 - No podemos controlar el error, pero si podemos estimar su magnitud.

Ejemplo de un algoritmo

Algoritmo de la multiplicación de enteros



Algoritmo para multiplicar dos números positivos

$$\begin{array}{r} 981 \\ 1234 \\ \hline 3924 \\ 2943 \\ 1962 \\ 981 \\ \hline 1210554 \end{array}$$

(a)

$$\begin{array}{r} 981 \\ 1234 \\ \hline 981 \\ 1962 \\ 2943 \\ 3924 \\ \hline 1210554 \end{array}$$

(b)

Figura 1.1. Multiplicación (a) americana (b) inglesa



Algoritmo para multiplicar dos números positivos

981	1.234	1.234
490	2.468	
245	4.936	4.936
122	9.872	
61	19.744	19.744
30	39.488	
15	78.976	78.976
7	157.952	157.952
3	315.904	315.904
1	631.808	<u>631.808</u>
		1.210.554

Regla:

- Dividir el valor de la columna izquierda entre 2, ignorando los restos, hasta que sea un 1
- A la vez se duplica el valor de la columna derecha
- Se suman los valores de la columna derecha siempre que su valor de la izquierda no sea par

Figura 1.2. Multiplicación à la russe



Algoritmo para multiplicar dos números positivos

	Multiplicar		Desplazar	Resultado
i)	09	12	4	108 . . .
ii)	09	34	2	306 .
iii)	81	12	2	972 .
iv)	81	34	0	<u>2754</u>
				1210554

Figura 1.3. Multiplicación de 0981 por 1.234 mediante divide y vencerás

- Ambos números tienen el mismo número de cifras y que sea potencias de 2. Si es necesario aumentamos ceros a la izquierda.

Representación de algoritmos

Por lo general se utiliza Pseudocódigo o diagramas de flujo



Representación de algoritmos

Ejemplo 1:

- Dados tres números enteros mayores a cero, considere la operación que permita calcular el promedio de los tres, considerando el punto a favor.
- Escriba el algoritmo para dicha operación e indique las especificaciones.



Pseudocódigo

Ejemplo 1:

- Dados tres números enteros mayores a cero, considere la operación que permita calcular el promedio de los tres, considerando el punto a favor.
- Escriba el algoritmo para dicha operación e indique las especificaciones.

ESPECIFICACIONES:

Objetivo:	Calcular el promedio de tres números
Entrada:	n1, n2, n3
Precondición:	$n1 > 0$, $n2 > 0$, $n3 > 0$
Salida:	Prom
Postcondición:	Variable Prom almacena el promedio de los números

ESTRATEGÍA

- 1) Sumar los tres números y asignarlo a una variable
- 2) Dividir la suma entre 3 y capturar la parte entera en la variable Prom
- 3) Capturamos el modulo o resto entre 3 y lo comparamos con el valor de 2
- 4) Incrementamos Prom en uno, si el resto coincide con el valor de 2



Pseudocódigo

Ejemplo 1:

- Dados tres números enteros mayores a cero, considere la operación que permita calcular el promedio de los tres, considerando el punto a favor.
- Escriba el algoritmo para dicha operación e indique las especificaciones.

Algoritmo CalcularPromedio

Inicio

Suma \leftarrow n1 + n2 + n3

Prom \leftarrow Suma **divisionEntera** 3

Si (Suma **modulo** 3 = 2) entonces

Prom \leftarrow Prom + 1

Fsi

Fin

TRAZA DEL ALGORITMO

Considere los siguientes valores: n1 = 14, n2 = 24, y n3 = 36

<i>n1</i>	<i>n2</i>	<i>n3</i>	<i>Suma</i>	<i>Prom</i>	<i>Suma modulo 3</i>
14	24	36	74	24	2
				25	



Implementación en Java

Ejemplo 1:

- Dados tres números enteros mayores a cero, considere la operación que permita calcular el promedio de los tres, considerando el punto a favor.
- Escriba el algoritmo para dicha operación e indique las especificaciones.

```
public static int getAverage(int n1, int n2, int n3){  
    int sum = n1 + n2 + n3;  
    int avg = sum / 3;  
    if(sum % 3 == 2){  
        avg++;  
    }  
    return avg;  
}
```

TRAZA DEL ALGORITMO

Considere los siguientes valores: $n1 = 14$, $n2 = 24$, y $n3 = 36$

$n1$	$n2$	$n3$	Suma	Prom	Suma modulo 3
14	24	36	74	24	2
				25	



EJEMPLO 2


Ejemplo 2:

Calcular independientemente la suma de los números pares e impares comprendidos entre 1 y n
Escriba el algoritmo para esta operación e indique sus especificaciones.

ESPECIFICACIONES:

Objetivo:	Calcular suma de pares e impares
Entrada:	n
Precondición:	$n > 0$
Salida:	sumaPares, sumaImpares
Postcondición:	sumaPares y sumaImpares son mayores e iguales a cero

ESTRATEGÍA

Con $n = 6$		$\text{sumaPares} = 2 + 4 + 6 = 12$
		$\text{sumaImpares} = 1 + 3 + 5 = 9$

Pseudocódigo

Algoritmo CalcularSumasPI

Inicio

sumaPares \leftarrow 0

sumalImpares \leftarrow 0

Para $i = 1$ hasta n

Si ($i \bmod 2 = 0$) entonces

 sumaPares \leftarrow sumaPares + i

Si no

 sumalImpares \leftarrow sumalImpares + i

Fsi

FPara

Fin

Implementación en java

```
public static int[] getOddEvenSum(int n){
    int sumOdd = 0;
    int sumEven = 0;
    for(int i = 1; i <= n; i++){
        if(i % 2 == 0){
            sumEven = sumEven + i;
        }else{
            sumOdd = sumOdd + i;
        }
    }
    //Utilizamos un array para retornar los 2 valores
    int[] ans = {sumEven, sumOdd};
    return ans;
}
```

Estructura de Datos

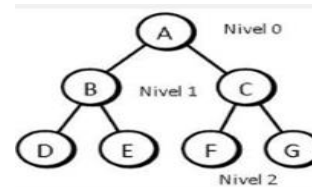
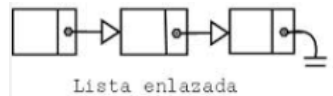
Modelo matemático que comprende elementos y funciones

Estructura de Datos

Es una forma de almacenar y organizar los datos, para facilitar el acceso y modificaciones. No hay una única estructura de datos que trabaje bien para todo propósito, es importante conocer las fortalezas y limitaciones de cada una de ellas.

Por ejemplo:

Los arreglos, las listas enlazadas, los árboles binarios, los grafos, etc.



CLASES – OBJETOS – MÉTODOS

CONCEPTO DE OBJETO

- Definición Formal (Grady Booch)
 - Un objeto es una entidad que tiene un **estado**, un **comportamiento** y una **identidad**.

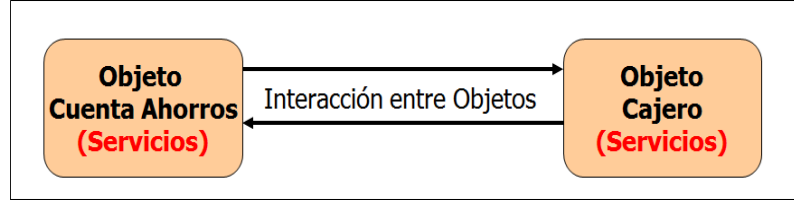


CONCEPTO DE OBJETOS...(Estado)

- Todos los objetos tienen características, propiedades o atributos.
 - Por ejemplo, el objeto Cuenta de Ahorro tiene como atributos: **Número Cuenta, Nombres, Dirección, Documento de Identidad.**
- Los **atributos** normalmente son estáticos, no cambian con el tiempo; mientras que los **valores** asociados a estos atributos son dinámicos, pueden cambiar con el tiempo.
- El **estado** de un objeto está dado por los valores actuales de cada una de sus propiedades o atributos.

CONCEPTO DE OBJETOS...(Comportamiento)

- Todos los objetos interactúan unos con otros. Esto significa que ningún objeto existe de forma aislada.
 - La interacción se efectúa en función a los **servicios** que ofrece cada objeto. Un servicio es alguna función (comportamiento) que puede efectuar un objeto.



- El comportamiento es cómo actúa y reacciona un objeto, en función a la interacción que efectúa sobre otros objetos o que efectúan otros objetos sobre él.

CONCEPTO DE OBJETOS...(Identidad)

- Un objeto en el mundo real es único y se diferencia de los otros objetos incluso de la misma clase.
 - Es posible identificar entre todas las propiedades del objeto, una propiedad que lo distingue de todos los demás objetos, esta propiedad es la que identifica el objeto.

CUENTA DE AHORROS	
Número Cuenta:	12345
Nombres:	Ana Paz
Dirección:	Av. Perú 123
Doc. Identidad:	23742723

No puede haber dos cuentas de ahorros con el mismo Número. En consecuencia, el Número de Cuenta **identifica** unívocamente al objeto

- La **identidad** es aquella propiedad o atributo de un objeto que los distingue de todos los demás objetos.

ENCAPSULAMIENTO

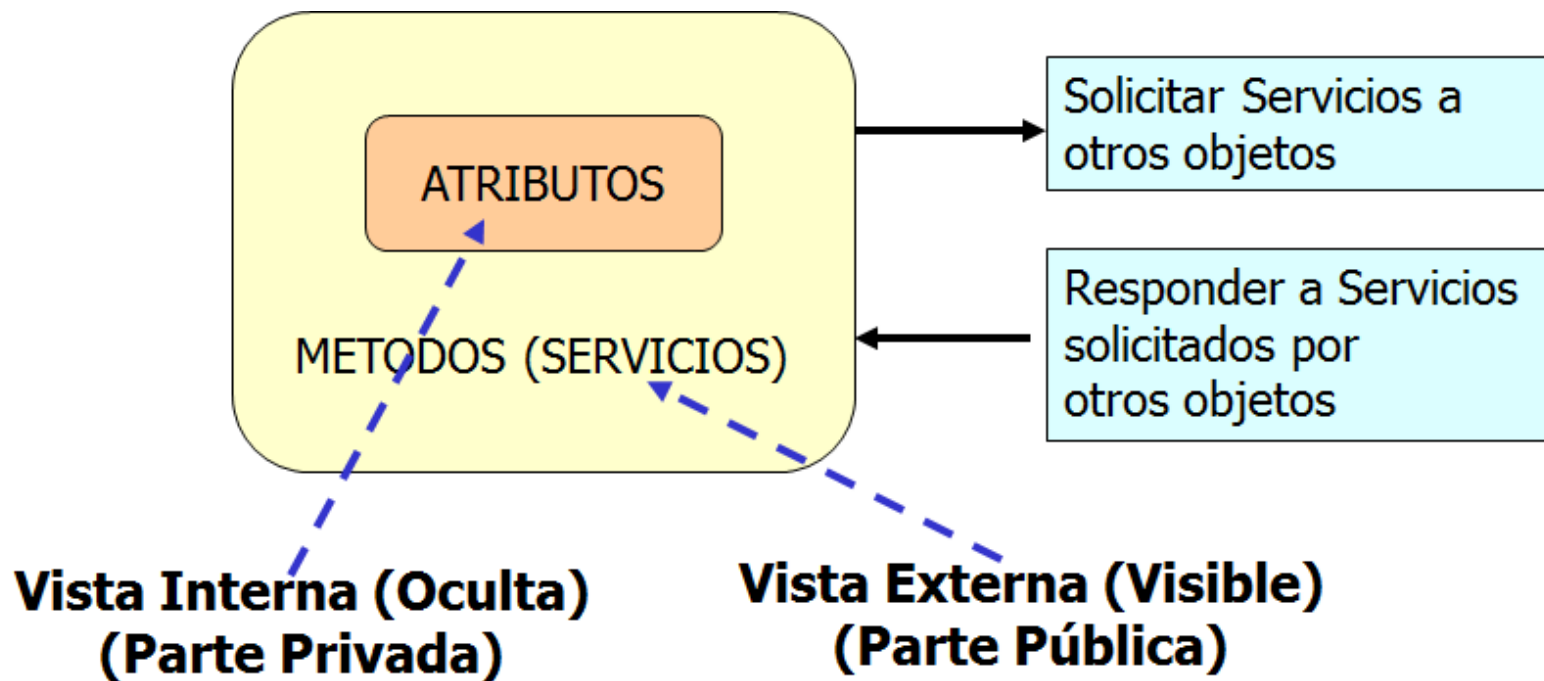
Objeto = Características + Comportamiento

Objeto = Información + Proceso

Objeto = Atributos + Métodos

- Las características y el comportamiento de un objeto es modelado en un compartimiento (cápsula) único e indivisible.
- Por encapsulamiento debemos entender que un objeto debe ser considerado como “un todo”, como una unidad atómica indivisible, que engloba las características y el comportamiento del objeto.

REPRESENTACION DE UN OBJETO



CLASES

- La **abstracción de clasificación** nos permite percibir que los objetos de la realidad están organizados como clases o tipos de objetos, caracterizados por propiedades comunes. Todo objeto pertenece a alguna clase de objetos.
- En la POO una clase es la implementación de un tipo de objetos. La POO identifica, implementa y utiliza tipos o clases de objetos del dominio del problema.
- **Una clase es como un molde que se utilizará para construir objetos.**

Clase = Implementación (*Atributos + Métodos*)
Comunes a todos los objetos de la clase

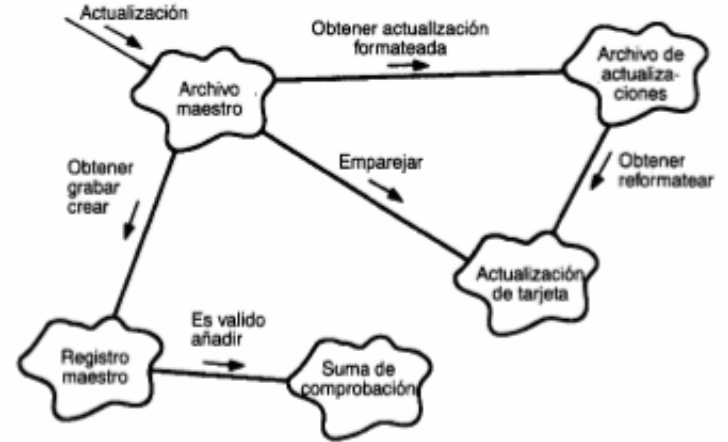
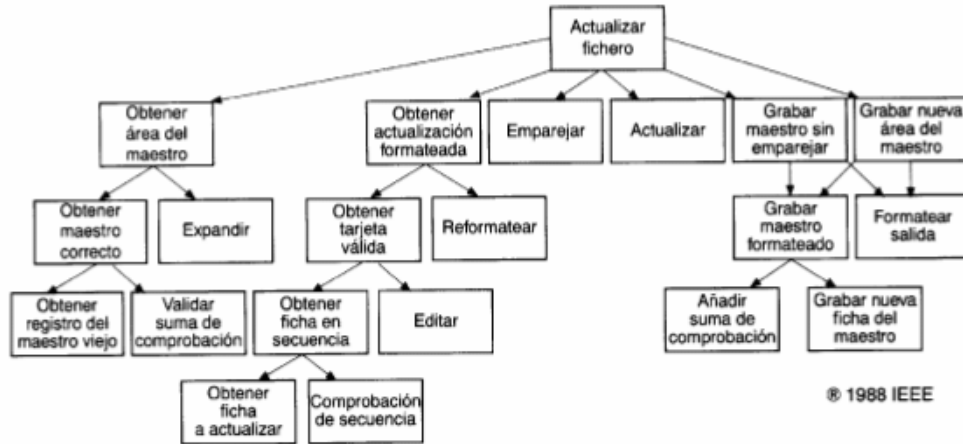
En la POO lo que realmente se implementa son las clases y no los objetos.
Los objetos se crean o construyen ("*tienen vida*") en tiempo de ejecución

CLASES...(Conceptos Asociados)

- **ACOPLAMIENTO**

- En la programación estructurada los módulos se escriben muchas veces en función de otros módulos. Por ejemplo, si el módulo B depende del módulo A, entonces cualquier cambio en el módulo A implica también una modificación en el módulo B.
- Esta dependencia se conoce como acoplamiento. Un programa con muchas dependencias tiene fuerte acoplamiento (calidad no deseada). Los buenos programas tienen pocas (o ninguna) dependencia, es decir, débil acoplamiento.
- La POO tiende al diseño e implementación de clases con débil acoplamiento

CLASES...(Conceptos Asociados)



CLASES...(Conceptos Asociados)

- **OCULTAMIENTO DE LA INFORMACIÓN**

- Es el proceso de ocultar todos los detalles internos de un objeto, y poner a disposición de los otros objetos solamente la lista de servicios que éste ofrece.
- **Es ver un objeto con una vista interna y una vista externa.**
- La **vista interna** no es visible ni accesible para otros objetos, esconde la estructura de los atributos y protege el acceso directo a ellos desde fuera del objeto. También es parte de la vista interna la implementación de los métodos, es decir el “**como**” están hechos.
- La **vista externa** es lo que ven los otros objetos, es la parte visible y accesible del objeto, está constituido por la lista de métodos del objeto, con una explicación de “**que**” hacen o “**para que**” sirven.

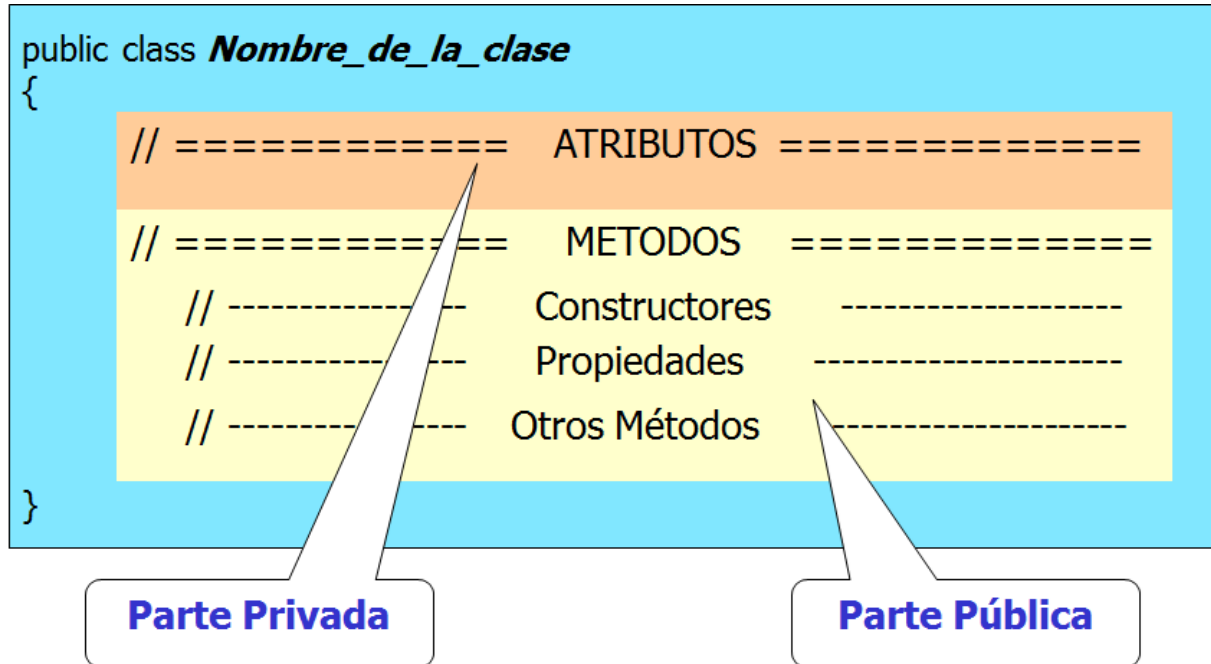
CLASES...(Conceptos Asociados)

- **INTERFACES**

- La interfaz es la vista externa del objeto. Es el conjunto de servicios que ofrece el objeto. Los otros objetos o el resto del programa pueden utilizar el objeto sólo mediante los servicios definidos en la interfaz.
- Se puede modificar la implementación interna de un objeto A, mientras no se cambie la interfaz del objeto A los otros objetos que interactúan con éste no se verán afectados.

IMPLEMENTACIÓN DE CLASES

- Esquema general de implementación de clases en Java, que se utilizará como patrón.





Referencias Bibliográficas

1. Brassard, G., Bratley, P., & Garcia-Bermejo, R. (1997). Fundamentos de algoritmia (Vol. 86). Madrid: Prentice Hall.