

# Análisis de Algoritmos y Estructura de Datos

**Algoritmos y pseudocódigo**

Prof. Violeta Chang C

Semestre 2 – 2023



# Algoritmos y pseudocódigo

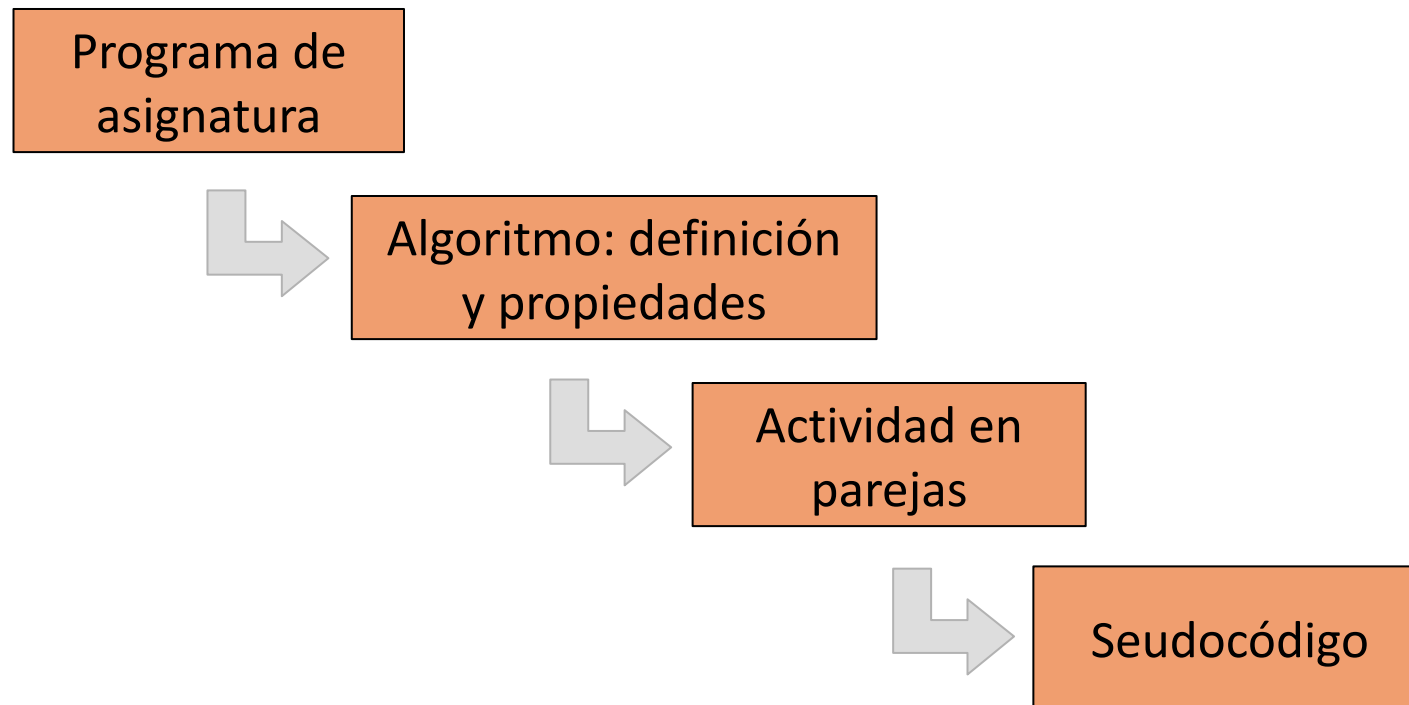
- **Contenidos:**

- Aspectos administrativos y generales de la asignatura
- Introducción a algoritmos
- Representación de un algoritmo

- **Objetivos:**

- Resolver dudas de la metodología de trabajo y evaluación de la asignatura
- Explicar concepto y características de un algoritmo
- Comprender necesidad de representación de un algoritmo
- Conocer un formato de pseudocódigo común en la construcción de algoritmos

# Ruta de la sesión





DEPARTAMENTO DE  
**INGENIERÍA  
INFORMÁTICA**  
UNIVERSIDAD DE SANTIAGO DE CHILE

# Revisión de programa de asignatura



# Forma de trabajo

- Material docente disponible en uVirtual (código 13205).
- Contenido semanal: cátedra + refuerzo/ejercicios + laboratorio.
- Cátedra: martes y jueves
  - PEPs + incentivos/simulacros
- Laboratorio: viernes, en lenguaje C
  - Tareas + entregas semanales en clase



# Evaluación de cátedra

- POR con cátedra  $\geq 3.6$
- Cátedra y laboratorio se aprueban por separado
- Para aprobar el curso se debe aprobar AMBOS cátedra y laboratorio
- La exigencia académica en evaluaciones de cátedra y laboratorio es de 60%.

Cátedra (50%)

Pruebas escritas parciales	PEP1, PEP2, PEP3	33.3% c/u
Cat = promedio(PEP1, PEP2, PEP3)		

PEP1	martes 3 de octubre
PEP2	martes 14 de noviembre
PEP3	martes 12 de diciembre



# Evaluación de laboratorio

- TL: promedio de 70% mejores notas de entregas semanales
- Cátedra y laboratorio se aprueban por separado
- Para aprobar el curso se debe aprobar AMBOS cátedra y laboratorio
- La exigencia académica en evaluaciones de cátedra y laboratorio es de 60%.

Laboratorio (50%)

Tareas computacionales	L1, L2, L3	25% c/u
Trabajo en sesiones de laboratorio	TL	25%
Lab = promedio(L1, L2, L3, TL)		

Tarea 1	miércoles 04 de octubre
Tarea 2	miércoles 15 de noviembre
Tarea 3	miércoles 13 de diciembre



DEPARTAMENTO DE  
**INGENIERÍA  
INFORMÁTICA**  
UNIVERSIDAD DE SANTIAGO DE CHILE

# Algoritmo





# Consulta



- ¿Qué es un algoritmo?



- ¿Cómo debe ser un algoritmo?



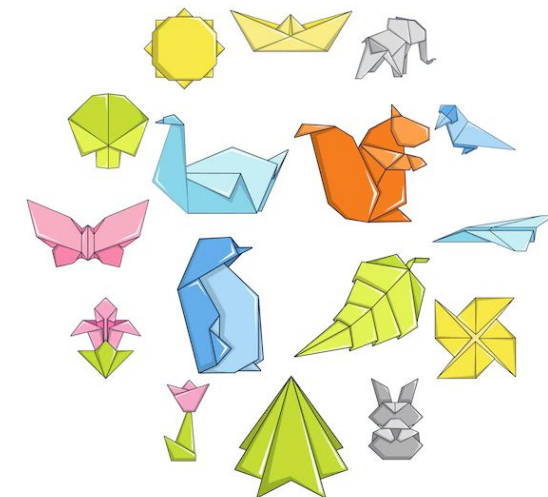
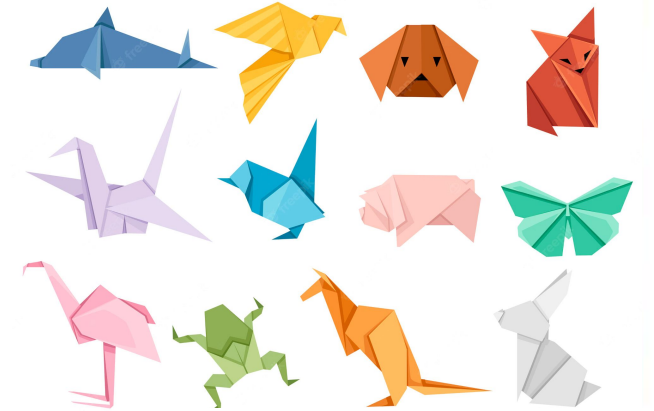
# Algoritmo: Definición

- Algoritmo:
  - Secuencia de instrucciones que se deben seguir para resolver un problema.
  - Procedimiento **bien definido**, que toma una entrada (valor, instancia) o un conjunto de entradas y **produce una salida** (o un conjunto de ellas) .
  - Serie de pasos computacionales que transforman una entrada en una salida.



# Actividad: construcción de origami

En una hoja de cuaderno, escribir una secuencia de pasos para hacer un avión de papel.



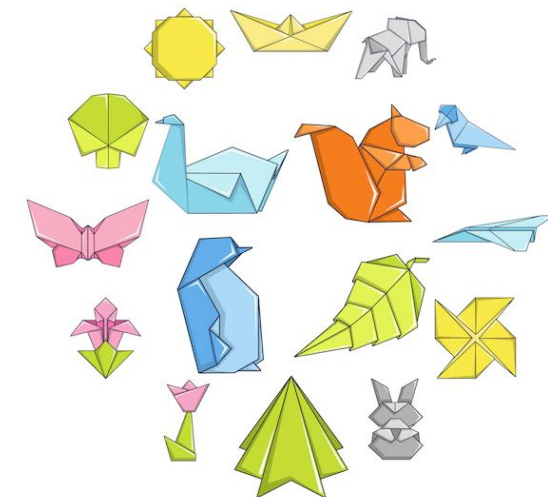
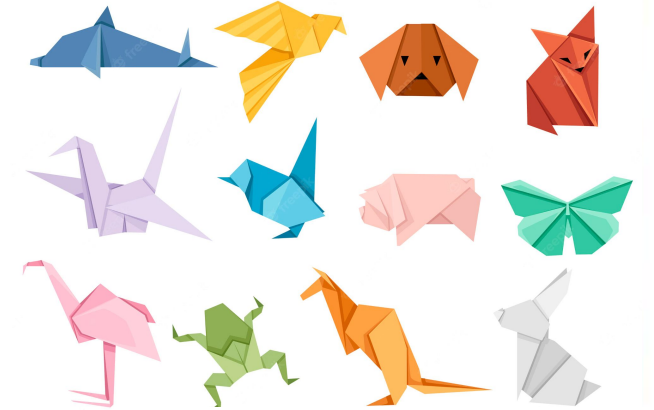


# Actividad: construcción de origami

En una hoja de cuaderno, escribir una secuencia de pasos para hacer un avión de papel.

Intercambiar la hoja con compañera(o) que esté más cerca y seguir los pasos (sin apoyo verbal)

¿se logró el resultado deseado?





# Propiedades de los algoritmos

- Un algoritmo es una **secuencia finita** de pasos que resuelve un **problema bien definido**.

Un algoritmo debe ser:



## PRECISO

Es decir, cada instrucción debe indicar claramente lo que se tiene que hacer.



## FINITO

Es decir, debe tener un número limitado de pasos.



## DEFINIDO

Es decir, debe producir los mismos resultados para las mismas condiciones de entrada.



# Algoritmo: Definición

- Los problemas computacionales deben ser **bien definidos**, acotados a un dominio determinado.
- Un algoritmo es correcto si para cada entrada se obtiene la salida correcta.
- Ejemplo: problema de ordenamiento
  - Entrada: secuencia de números  $n_i$
  - Salida: secuencia ordenada de números, tal que  $n_1 \leq n_2 \leq n_3 \dots$

¿Tipo de dato?  
¿Criterio de orden?



# Algoritmos: ejemplos de uso

- Son usados para resolver problemas en diversas áreas (por ejemplo):
  - **Internet:** pagerank, enrutamiento de paquetes, protocolos de comunicación, encriptación, recuperación de información, etc
  - **Comercio electrónico:** firma digitales, blockchain, seguridad, búsqueda y despliegue de información, transacciones, etc
  - **Industria:** optimización de recursos, inversiones, planificación de vuelos, alocaación de recursos computacionales, análisis de datos, etc
  - **Biología y medicina:** reconocimiento de imágenes, identificación de biomarcadores, seguridad y confidencialidad de los datos, análisis de señales, etc.
  - **Internet de las cosas:** sensores (zapatillas, jardín, puertas, perros, lavadoras, hornos, luces, basureros) + algoritmo de decisión





# Algoritmos: reflexionemos



- ¿Qué sucede si un algoritmo no está bien diseñado?
- ¿Puede un algoritmo matar a **una** persona? ¿Cómo?
- ¿Puede un algoritmo causar la muerte de **cientos** de personas?
- ¿Puede un algoritmo salvar vidas?
- ¿Puede un algoritmo mejorar la expectativa de vida de las personas?
- ¿Puede un algoritmo ser racista o clasista?
- ¿Puede un algoritmo ayudar a terminar con la economía?

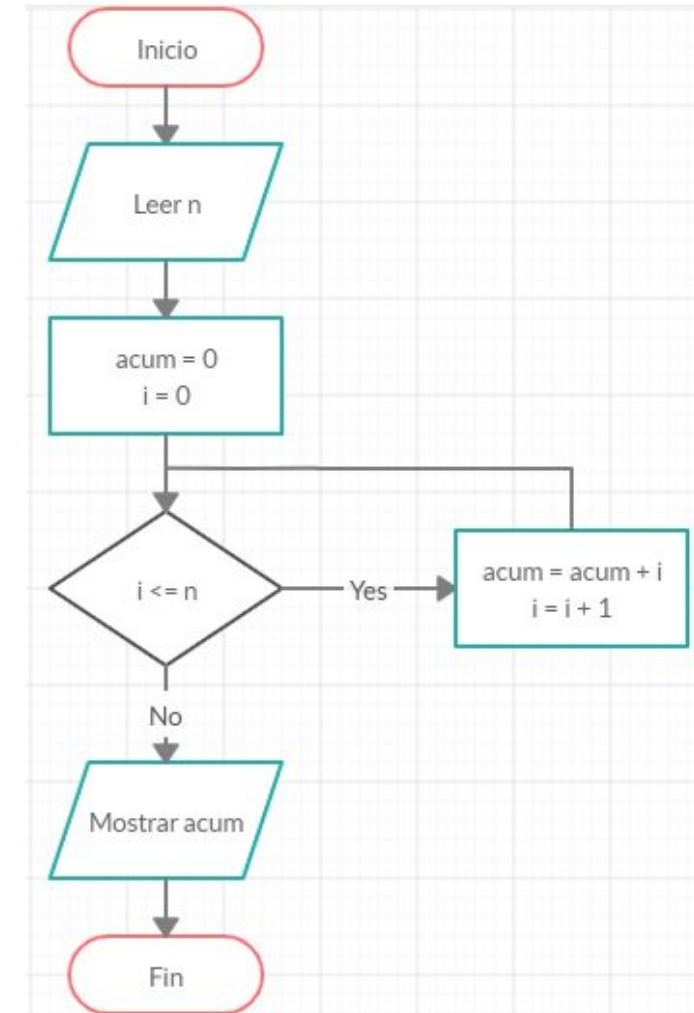


# Representación de algoritmos



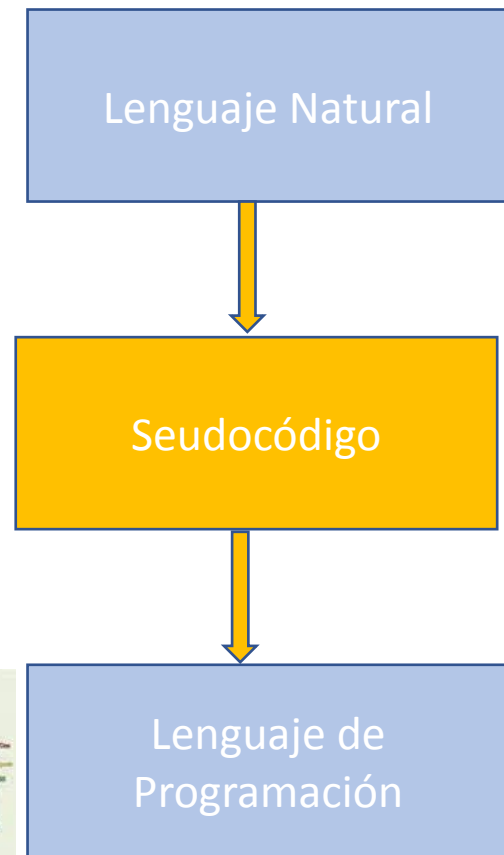
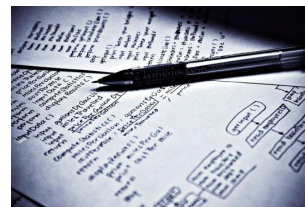
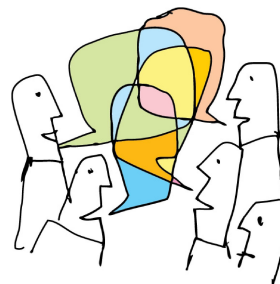
# Diagrama de flujo

- Representación gráfica de los pasos de un algoritmo
- Manera simple de mostrar el flujo de ejecución de los pasos de un algoritmo
- Define diferentes tipos de formas para representar operaciones en un algoritmo
- ¿Qué hace el siguiente diagrama de flujo?



# Seudocódigo

- Representación de los pasos de un algoritmo en un lenguaje de alto nivel
- **Se abstrae de los detalles de implementación**
- Es independiente de la máquina
- Usa expresiones cercanas al lenguaje natural y a los lenguajes de programación
- Permite adaptar pensamiento y describir los pasos para resolver un problema.
- Seudo: parece pero no es.





# Seudocódigo

- ¿Qué problemas podría presentar este pseudocódigo?

alg()

acum  $\leftarrow$  0

i  $\leftarrow$  0

mientras i  $\leq$  n hacer

acum  $\leftarrow$  acum+i

i  $\leftarrow$  i+1

devolver acum



# Seudocódigo

- ¿Qué problemas podría presentar este pseudocódigo?

alg()

acum  $\leftarrow$  0

i  $\leftarrow$  0

mientras i  $\leq$  n hacer

acum  $\leftarrow$  acum+i

i  $\leftarrow$  i+1

devolver acum

```
alg()  
  acum  $\leftarrow$  0  
  i  $\leftarrow$  0  
  mientras i  $\leq$  n hacer  
    acum  $\leftarrow$  acum+i  
  i  $\leftarrow$  i+1  
  devolver acum
```

```
alg(num n):num  
  acum  $\leftarrow$  0  
  i  $\leftarrow$  0  
  mientras i  $\leq$  n hacer  
    acum  $\leftarrow$  acum+i  
  i  $\leftarrow$  i+1  
  devolver acum
```

```
alg(num n):num  
  acum  $\leftarrow$  0  
  i  $\leftarrow$  0  
  mientras i  $\leq$  n hacer  
    acum  $\leftarrow$  acum+i  
  i  $\leftarrow$  i+1  
  devolver acum
```

```
num alg(num n):num  
  acum  $\leftarrow$  0  
  i  $\leftarrow$  0  
  mientras i  $\leq$  n hacer  
    acum  $\leftarrow$  acum+i  
  i  $\leftarrow$  i+1  
  devolver acum
```



# Seudocódigo

- ¿Qué problemas podría presentar este pseudocódigo?

alg()

acum  $\leftarrow$  0

i  $\leftarrow$  0

mientras i  $\leq$  n hacer

acum  $\leftarrow$  acum+i

i  $\leftarrow$  i+1

devolver acum

```
alg()  
  acum  $\leftarrow$  0  
  i  $\leftarrow$  0  
  mientras i  $\leq$  n hacer  
    acum  $\leftarrow$  acum+i  
  i  $\leftarrow$  i+1  
  devolver acum
```

```
alg(num n):num  
  acum  $\leftarrow$  0  
  i  $\leftarrow$  0  
  mientras i  $\leq$  n hacer  
    acum  $\leftarrow$  acum+i  
  i  $\leftarrow$  i+1  
  devolver acum
```

```
alg(num n):num  
  acum  $\leftarrow$  0  
  i  $\leftarrow$  0  
  mientras i  $\leq$  n hacer  
    acum  $\leftarrow$  acum+i  
  i  $\leftarrow$  i+1  
  devolver acum
```

```
num alg(num n):num  
  acum  $\leftarrow$  0  
  i  $\leftarrow$  0  
  mientras i  $\leq$  n hacer  
    acum  $\leftarrow$  acum+i  
  i  $\leftarrow$  i+1  
  devolver acum
```

No está tabulado  
Entradas no definidas



# Seudocódigo

- ¿Qué problemas podría presentar este pseudocódigo?

alg()

acum ← 0

i ← 0

mientras i ≤ n hacer

acum ← acum+i

i ← i+1

devolver acum

```
alg()  
  acum ← 0  
  i ← 0  
  mientras i ≤ n hacer
```

```
alg(num n):num  
  acum ← 0  
  i ← 0
```

Para evitar interpretaciones diferentes, se usará un formato definido para pseudocódigo

```
acum ← 0  
i ← 0  
mientras i ≤ n hacer  
  acum ← acum+i  
i ← i+1  
devolver acum
```

```
num alg(num n):num  
  acum ← 0  
  i ← 0  
  mientras i ≤ n hacer  
    acum ← acum+i  
  i ← i+1  
  devolver acum
```

No está tabulado  
Entradas no definidas



# Seudocódigo\*

Nombre del  
algoritmo

Entrada del  
algoritmo

Salida del  
algoritmo

```
alg(num n): num  
    acum ← 0  
    i ← 0  
    mientras i <= n hacer  
        acum ← acum+i  
        i ← i+1  
    devolver (acum)
```

Cuerpo del algoritmo,  
ordenado, indentado y con  
las instrucciones claras y  
definidas, de acuerdo al  
bloque que pertenece

Retorno del algoritmo  
que coincide con la  
salida definida

\*estos son aspectos generales, ver detalle





# Seudocódigo



¿Hacen lo mismo?

Alg1(num n)

```
    acum ← 0
    para i ← 0 hasta n
        acum ← acum + i
    escribir(acum)
```

Alg3(num n)

```
    acum ← (n * (n + 1)) / 2
    escribir(acum)
```

Alg2(num n)

```
    acum ← 0
    i ← 0
    mientras i ≤ n hacer
        acum ← acum + i
        i ← i + 1
    escribir(acum)
```



# Seudocódigo



¿Hacen lo mismo?

Alg1(num n)

```
    acum ← 0  
    para i ← 0 hasta n  
        acum ← acum + i  
    escribir(acum)
```

Alg3(num n)

```
    acum ← (n * (n + 1)) / 2  
    escribir(acum)
```

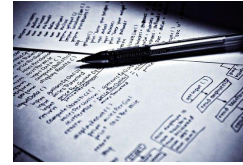
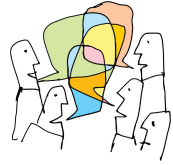
Alg2(num n)

```
    acum ← 0  
    i ← 0  
    mientras i ≤ n hacer  
        acum ← acum + i  
        i ← i + 1  
    escribir(acum)
```

¿Cuál es mejor?



# Contexto



solución 1



seudocódigo 1



solución 2



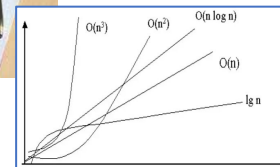
seudocódigo 2



solución 3



seudocódigo 3





# Algunos tipos de algoritmos

- Según la estructura:
  - Iterativos, recursivos
- Según su función :
  - De ordenamiento, de búsqueda, de encadenamiento adaptativo, de encadenamiento estático
- Según su forma de llegar a resultados
  - Probabilísticos/aleatorios/estocásticos, heurísticos, de escalada, voraces, deterministas



# Actividad de cierre



- Ir a [menti.com](https://www.menti.com) e ingresar código 3577 3179



- Próxima semana:
  - Complejidad de algoritmos