

Unidad 1

Introducción general

Introducción a la algoritmia basado en Python

Unidad 1 - Introducción general

Contenido

1. Pensamiento secuencial y algorítmico.
2. ¿Qué es un algoritmo?
3. ¿Qué es programar?
4. Introducción al concepto de memoria
5. Introducción al concepto de procesamiento
6. ¿Qué es un diagrama de flujo?
7. ¿Qué es pseudocódigo?
8. Configuración del espacio de trabajo

Autoevaluación del día 1 (memoria, procesamiento y diagramas de flujo).

Actividad 1

Actividad 1

Acerca del pensamiento requerido en la programación

¿Por qué programar?

¿Para qué sirve programar?

Pensamiento secuencial / algorítmico / lógico

¿De qué se trata?

Es una forma de pensamiento que se basa en la secuenciación de acciones/tareas, es decir, la acomodación de acciones/tareas de manera sucesiva, como lo hacemos en una lista de tareas, una receta y muchas otras situaciones cotidianas.

1. Observar la carta
2. Seleccionar el producto
3. Solicitar el producto
4. Esperar
5. Disfrutar :)



Pensamiento secuencial / algorítmico / lógico

¿Qué permite lograr?

Esta forma de pensamiento permite transformar situaciones cotidianas en secuencias claramente definidas, donde cada paso de la secuencia se identifican 3 componentes:



Componentes de una tarea / acción - Entrada

¿Qué son las entradas?

Llamaremos de esa manera a la información (datos) que requiere una tarea para poder iniciarse, es decir, son en la metáfora de una receta... el listado de ingredientes.

También pueden ser reglas, indicadores, dimensiones u otro aspecto relevante para el proceso.

Bases de datos

Sensores

API
(Servicios en
internet)

Acciones del
usuario

Componentes de una tarea / acción - Proceso

¿Qué es el proceso?

Es la forma en que llamamos al conjunto interno de acciones que permite transformar las entradas en la salida(spoiler-alert) esperada. En nuestra metáfora de cocina son las acciones que ejecutamos sobre los ingredientes para transformarlos en nuestro plato (salida).



Operaciones

Verificaciones

Transformaciones

También pueden ser procedimientos establecidos, modelos de análisis, fórmulas u otro conjunto de acciones ya probadas como capaces de entregar determinado resultado.

Componentes de una tarea / acción - Salida

¿Qué es la salida?

Es la más importante de los componentes, refleja el objetivo mismo de la tarea que estamos realizando, aunque hemos mostrado entradas y proceso previamente, lo usual sería que identificamos primero la salida requerida y después el proceso que se requiere para lograr, incluyendo las entradas o datos que requiere para poder obtener dicho resultado.

Resultados

Información
procesada

Mensajes

Gráficos

Valores en
general

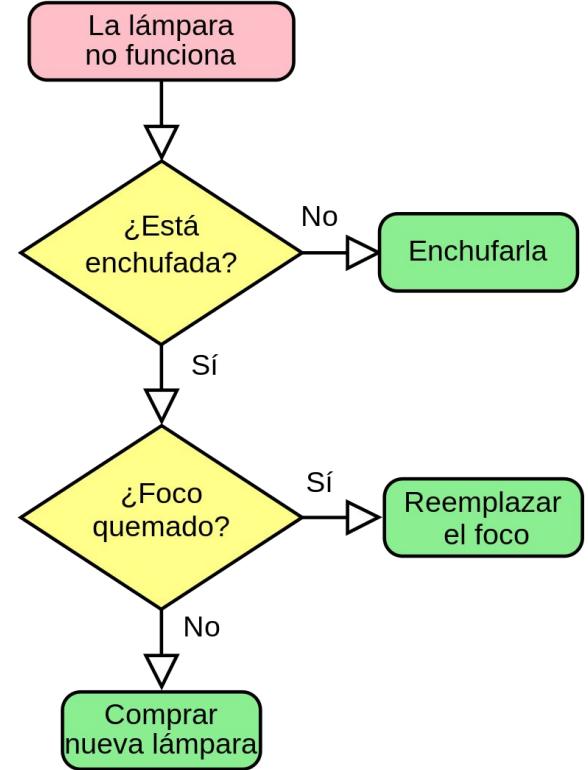
En resumen...

¿Qué busca este tipo de pensamiento?

Convertir situaciones “cotidianas” de negocio en secuencias de pasos que tienen una estructura como esta:



¿Qué es un algoritmo?

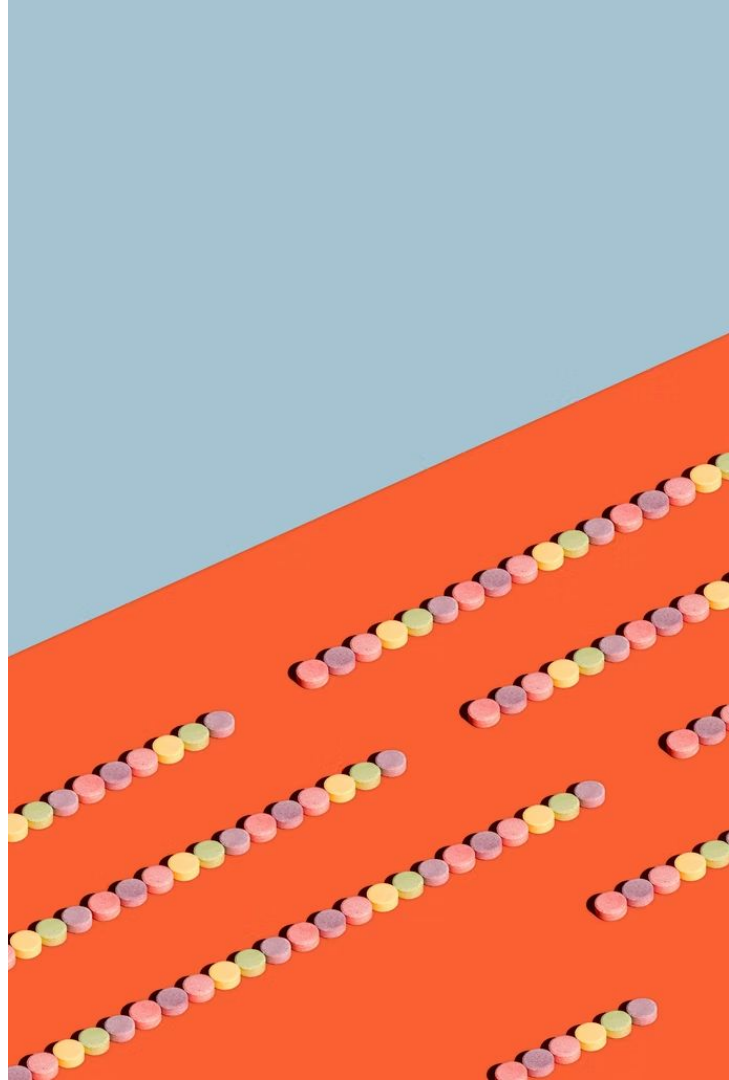


¿Qué es un algoritmo entonces?

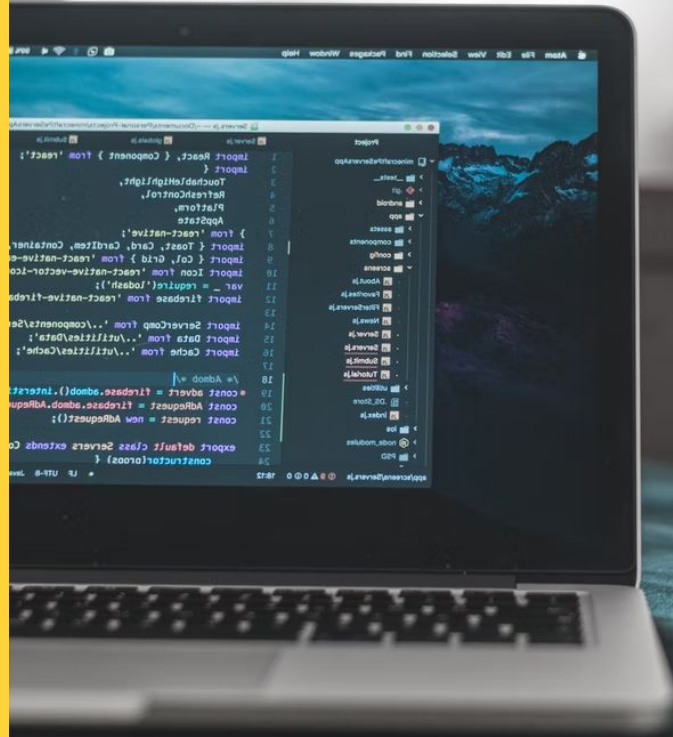
Dice en wikipedia algo como...

En [matemáticas](#), [lógica](#), [ciencias de la computación](#) y disciplinas relacionadas, un **algoritmo** es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades.² Dado un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución

Secuencia de pasos ordenados (proceso) que permite que dado un conjunto de entradas se obtenga una salida (solución).



¿Qué es programar entonces?



Actividad 2

Actividad 2

Acerca de los conceptos de memoría y procesamiento

Concepto de memoria

En un contexto de introducción a la programación



Concepto de memoria

¿Qué sabemos hasta ahora?

Sabemos que vamos a construir **secuencias ordenadas** para **resolver problemas** a través de la **obtención de resultados (salida)** identificando los **procesos a ejecutar** y las **entradas** o datos requeridos.

¿Qué papel juega la memoria en este proceso?



Concepto de procesamiento

En un contexto de introducción a la
programación



Concepto de procesamiento

¿Cómo se conecta con lo que sabemos?

En términos generales los computadores **están hechos para el procesamiento**, esto se traduce en la capacidad de **realizar operaciones** matemáticas (hacen los cálculos por nosotros).

Nuestras secuencias (algoritmos) contienen tareas y **esas tareas requieren de operaciones, la resolución de dichas operaciones es el procesamiento.**

¿Cómo trabaja en compañía de la memoria?

¿La memoria permite el procesamiento? (concepto)

¿Si quisiéramos sumar 2 números?

En el algoritmo para sumar 2 números, vamos a tener 2 entradas (una para cada número) y una salida que será el resultado de operar dichos números (adición de unidades).



Actividad 3

Actividad 3

Acerca de los diagramas de flujo

Diagramas de flujo y su utilidad al aprender a programar

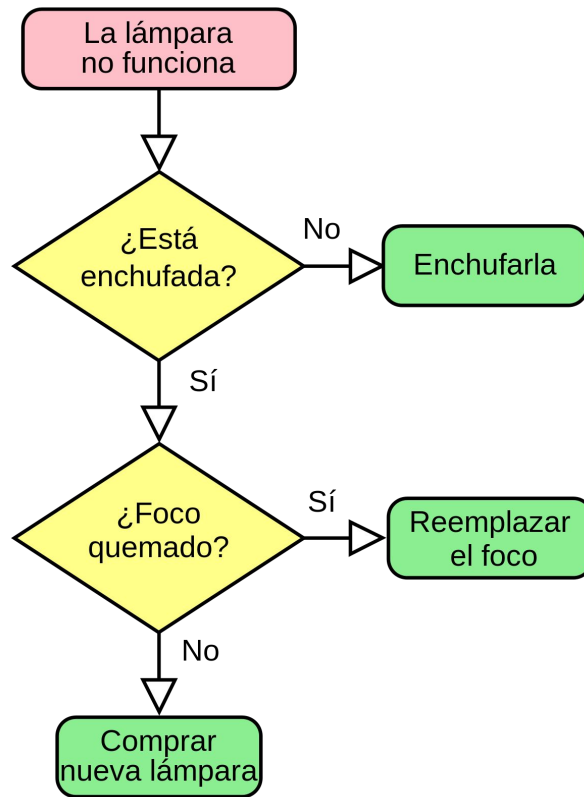


¿Qué es un diagrama de flujo?

¿Cómo se conecta con lo que sabemos?

Son esquemas que **representan secuencias**, permiten gestionar el camino que toma nuestra secuencia si existe más de uno.

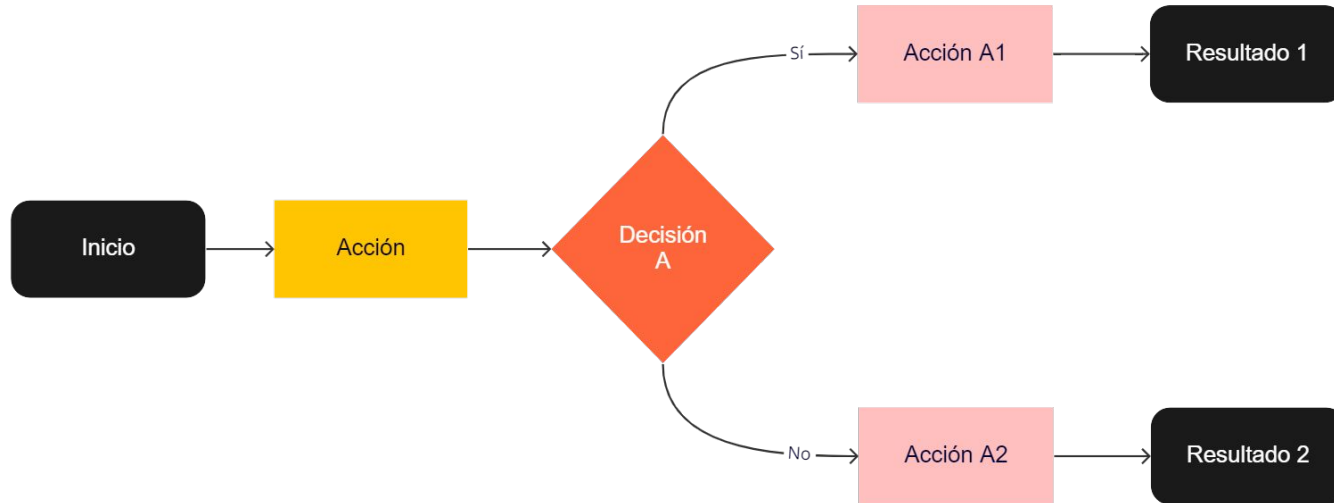
¿Cuáles son las reglas generales para construirlos?



¿Cómo construimos un diagrama de flujo?

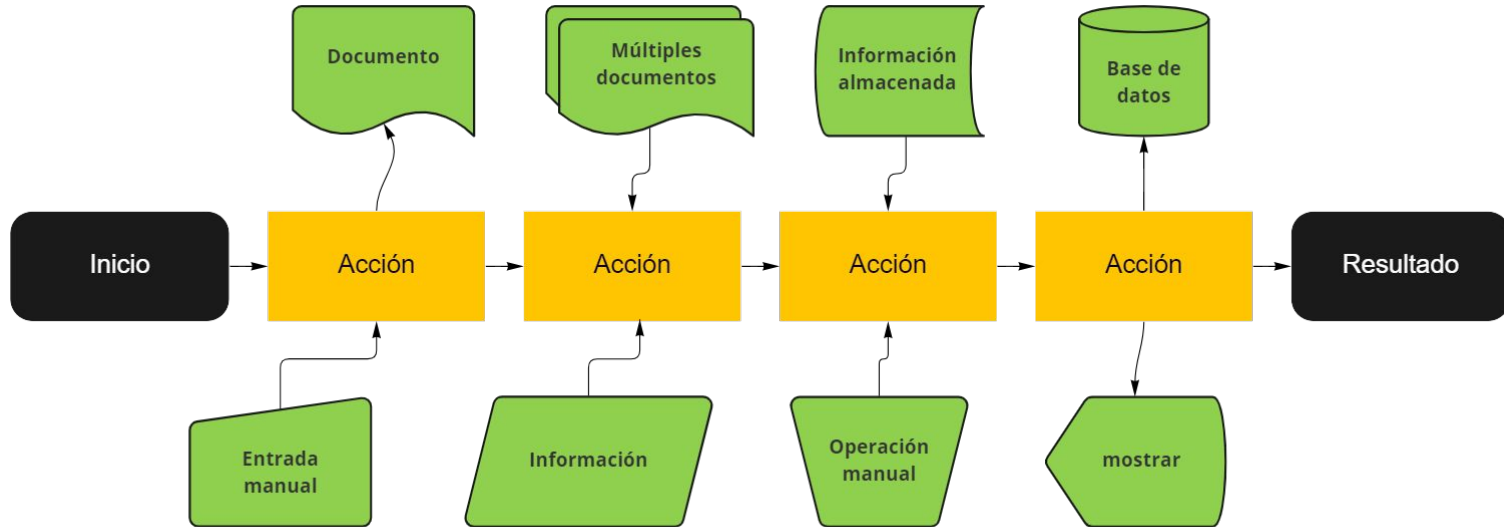
Convenciones básicas

Iniciadores, pasos, decisiones y flujos.



Otras figuras usadas

Convenciones para las entradas datos y operaciones de almacenamiento o gestionadas por el usuario



Es una ayuda para nuestro cerebro

¿Cómo pasar de cotidiano a algorítmico?

A través del análisis y el uso de esquemas que hacen “sencillo” el proceso de secuenciar un conjunto de tareas.

No siempre se hacen, pero mientras aprendes son bastante útiles.



Actividad 4

Actividad 4

Acercas del Pseudocódigo

Pseudocódigo

Un paso más, un dolor de cabeza
menos

Pseudocódigo estilo C:

```
subproceso funcion bizzbuzz
para (i <- 1; i<=100; i++) {
    establecer print_number a verdadero;
    Si i es divisible por 3
        escribir "Bizz";
        establecer print_number a falso;
    Si i es divisible por 5
        escribir "Buzz";
        establecer print_number a falso;
    Si print_number, escribir i;
    escribir una nueva línea;
}
```


¿Qué es el pseudocódigo?

¿Qué sabemos hasta ahora?

ya hemos avanzado en las **herramientas para construir nuestras secuencias**, sabemos que requerimos identificar **las salidas, las entradas y el proceso necesario**, también sabemos que es podemos usar diagramas de flujo en el proceso de construcción de nuestras secuencias.

¿Cómo aprovechamos el pseudocódigo en este camino?

¿Qué es el pseudocódigo?

Versión simplificada pero coherente

En ciencias de la computación, y análisis numérico, el **pseudocódigo**¹ (o lenguaje de descripción algorítmico) es una descripción de alto nivel compacta e informal² del principio operativo de un programa informático u otro algoritmo.

Utiliza las convenciones estructurales de un lenguaje de programación real,³ pero está diseñado para la lectura humana en lugar de la lectura mediante máquina,⁴

¿Cómo se ve esta forma de escritura y qué convenciones se usan?

Pseudocódigo

No estandarizado, pero explícito.

Veremos usualmente pseudocódigo en publicaciones científicas, libros de texto de programación u otras fuentes, es independiente de un lenguaje específico pero **contiene elementos comunes** en relación al control de flujo (hablaremos de esto en detalle más adelante).

¿Vemos algunos ejemplos?

Pseudocódigo

No estandarizado, pero explicito.

Procedimiento Ordenar (L)

↑ //Comentario : $L = (L_1, L_2, \dots, L_n)$ es una lista con n elementos//

$k \leftarrow 0$;

Repetir

↑ intercambio \leftarrow Falso;

$k \leftarrow k + 1$;

Para $i \leftarrow 1$ Hasta $n - k$ Con Paso 1 Hacer

↑ Si $L_i > L_{i+1}$ Entonces

↑ intercambiar (L_i, L_{i+1})

↓ intercambio \leftarrow Verdadero;

↓ Fin Si

↓ Fin Para

↓ Hasta Que intercambio = Falso;

Fin Procedimiento

Actividad 5

Actividad 5

Configuración del espacio de trabajo

Configuración del espacio de trabajo

¿Qué necesitamos?

Librerías del lenguaje (El núcleo de lenguaje)

IDE - Entorno integrado de desarrollo

Instalación de Anaconda

<https://www.anaconda.com/>

