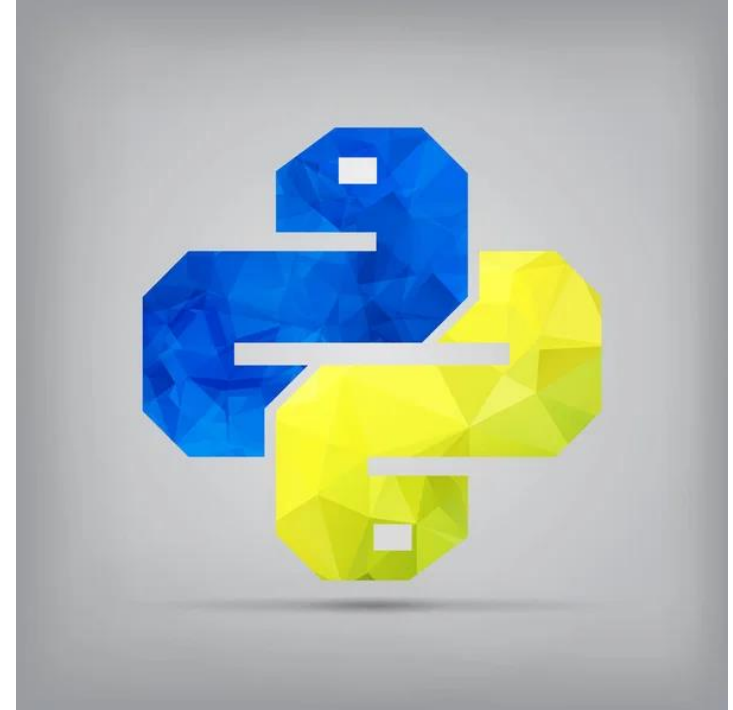




FUNDAMENTOS DE PYTHON

TEMARIO



TEMARIO

Modulo 1

- Algoritmos y programas
- Características de Python
- Primeras sentencias
- Tipos de datos, variables y funciones

Modulo 2

- Estructuras de control de flujo
- Programación orientada a objetos
- Clases
- Módulos

Modulo 3

- Manejo de archivos externos
- Conexión con base de datos
- Interfaces gráficas

TEMARIO

Modulo 1

- Algoritmos y programas
- Características de Python
- Primeras sentencias
- Tipos de datos, variables y funciones

- Familiarización con el lenguaje de programación
- Entender la sintaxis

Modulo 2

- Estructuras de control de flujo
- Programación orientada a objetos
- Clases
- Módulos

- Darle inteligencia a los programas
- Estructurar programas avanzados
- Automatización de rutinas en PC

Modulo 3

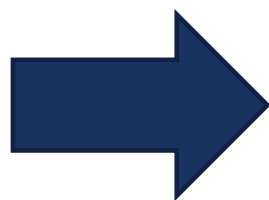
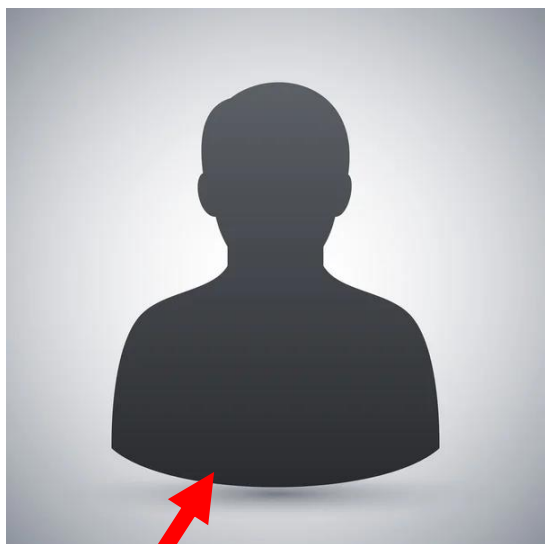
- Manejo de archivos externos
- Conexión con base de datos
- Interfaces gráficas

- Integración de programas con gestores de BBDD
- Visualización, generación de reportes, etc

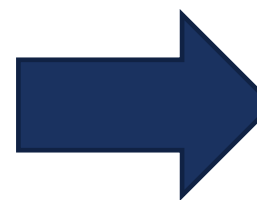
OBJETIVOS

PROGRAMACION

proceso → diseñar codificar dar mantenimiento Proteger → Código Fuente Programa



LENGUAJE DE PROGRAMACIÓN



EJEMPLO

El programador requiere que la computadora imprima en pantalla la utilidad de la empresa minera con los siguientes parámetros:

- Ingresos : \$10 M
- Costos: \$4.5 M



Instrucción (Código)

```
1 INGRESOS = 10
2 COSTOS = 4.5
3
4 UTILIDAD = INGRESOS - COSTOS
5
6 print("La utilidad es de la empresa minera es $" + str(UTILIDAD) + " millones.")
7
```

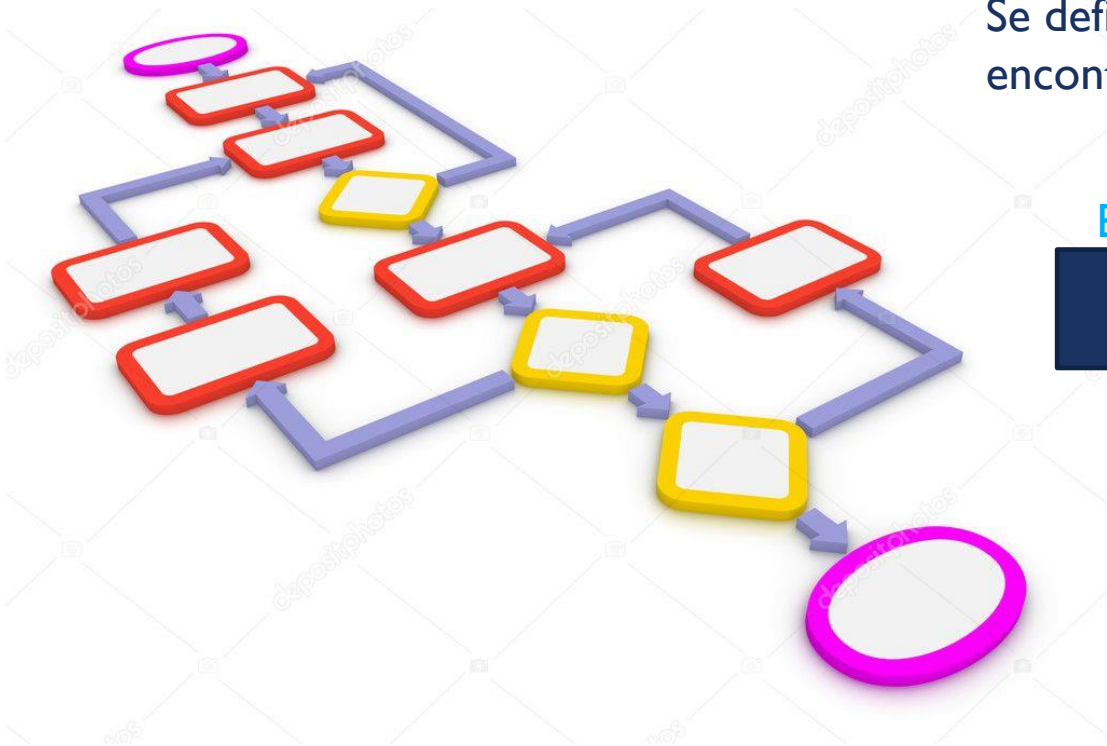


```
La utilidad es de la empresa minera es
$5.5 millones.
[Finished in 0.3s]
```

Resultado

ALGORITMOS

Se define como una lista finita de pasos ordenados que permite encontrar la solución a un problema determinado.



Entrada

Paso 1

Paso 2

Paso 3

Salida

TIPOS (EJEMPLO)

Resolveremos el problema de calcular la utilidad de una empresa minera presentado en el capítulo previo.

Lenguaje natural

Primero: Encontramos que por definición, la utilidad se define como la diferencia entre los ingresos y los costos de una empresa.

Segundo: Con los datos que se tienen, se reemplaza la fórmula.
 $10M - 4.5M = 5.5M$

Pseudocódigo

Proceso Utilidad

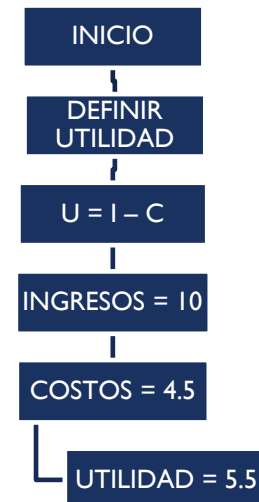
Definir Utilidad:
 $Utilidad = Ingresos - Costos$

Definir Ingresos, Costos:
Ingresos = 10
Costos = 4.5

Reemplazar valores:
 $Utilidad = 10 - 4.5$

Escribir resultado:
Utilidad = 5.5

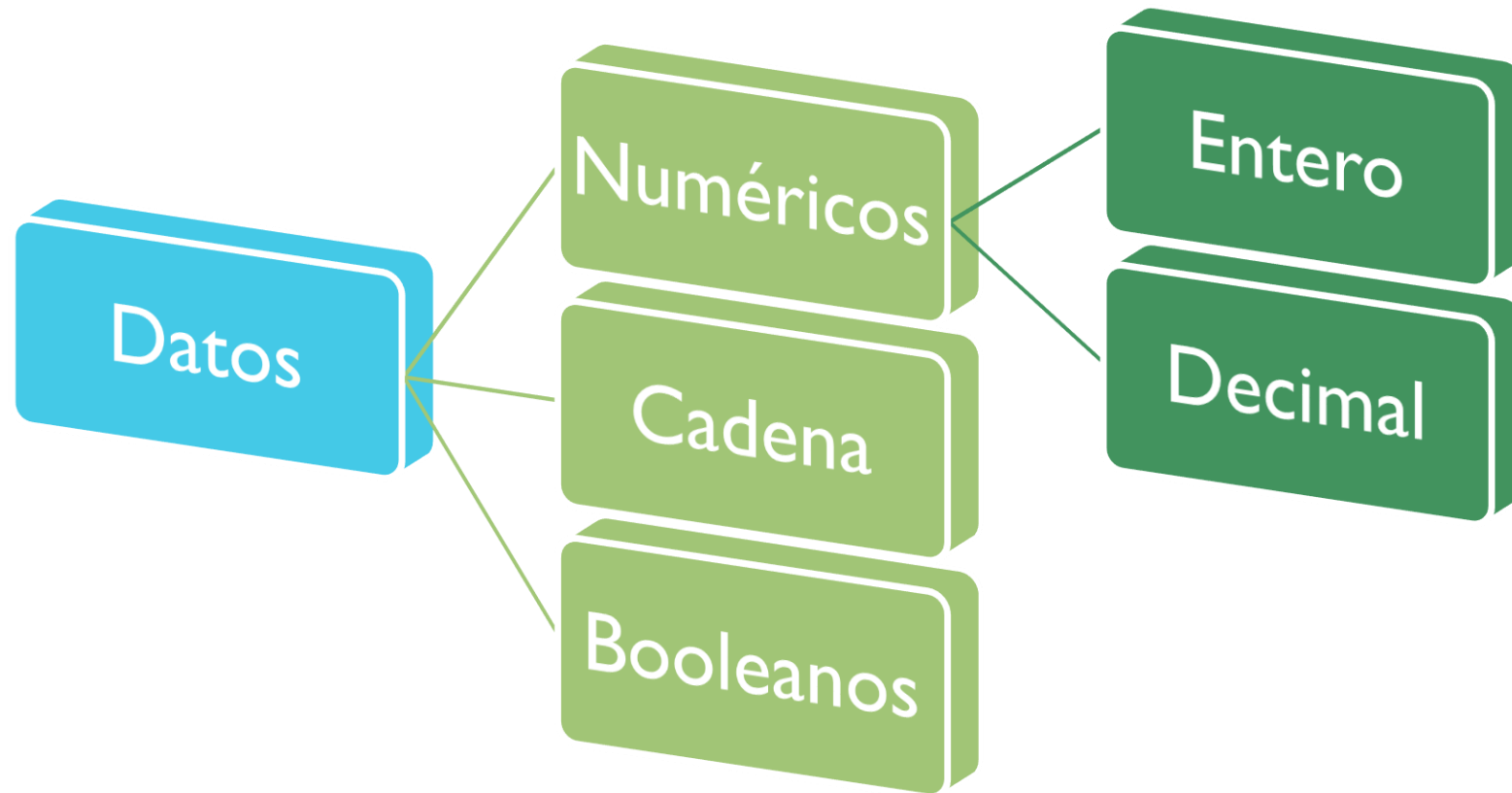
Diagrama de flujo



INSTALACIÓN

- **Python**
- **Editor de texto (Sublime Text)**

TIPOS DE DATOS



VARIABLES

Espacio en la memoria del ordenador donde se almacenará un valor que podrá cambiar durante la ejecución del programa.

factor_esponjamiento = 1.16

declaración de variable



asignación de variable



VARIABLES (REGLAS)

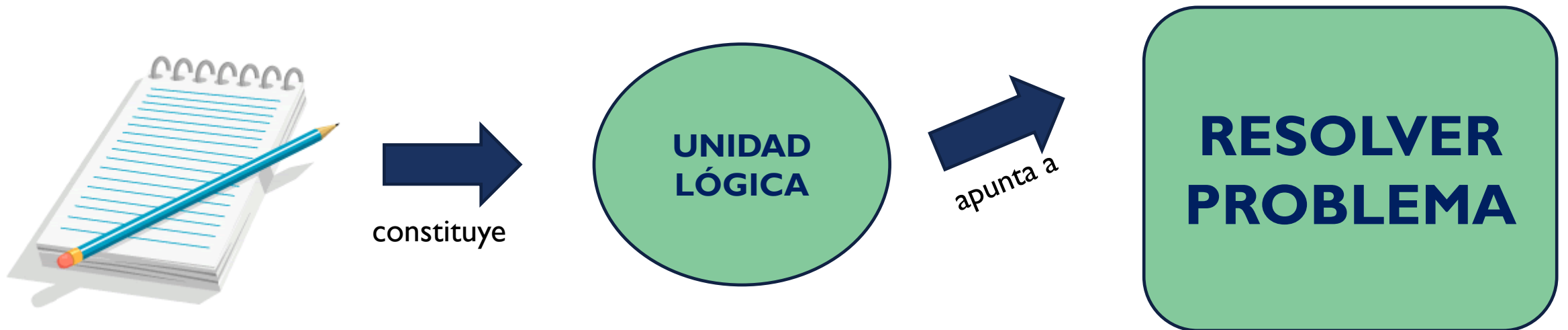
Tonelaje  **tonelaje**  **TONELAJE**

Ley cobre 

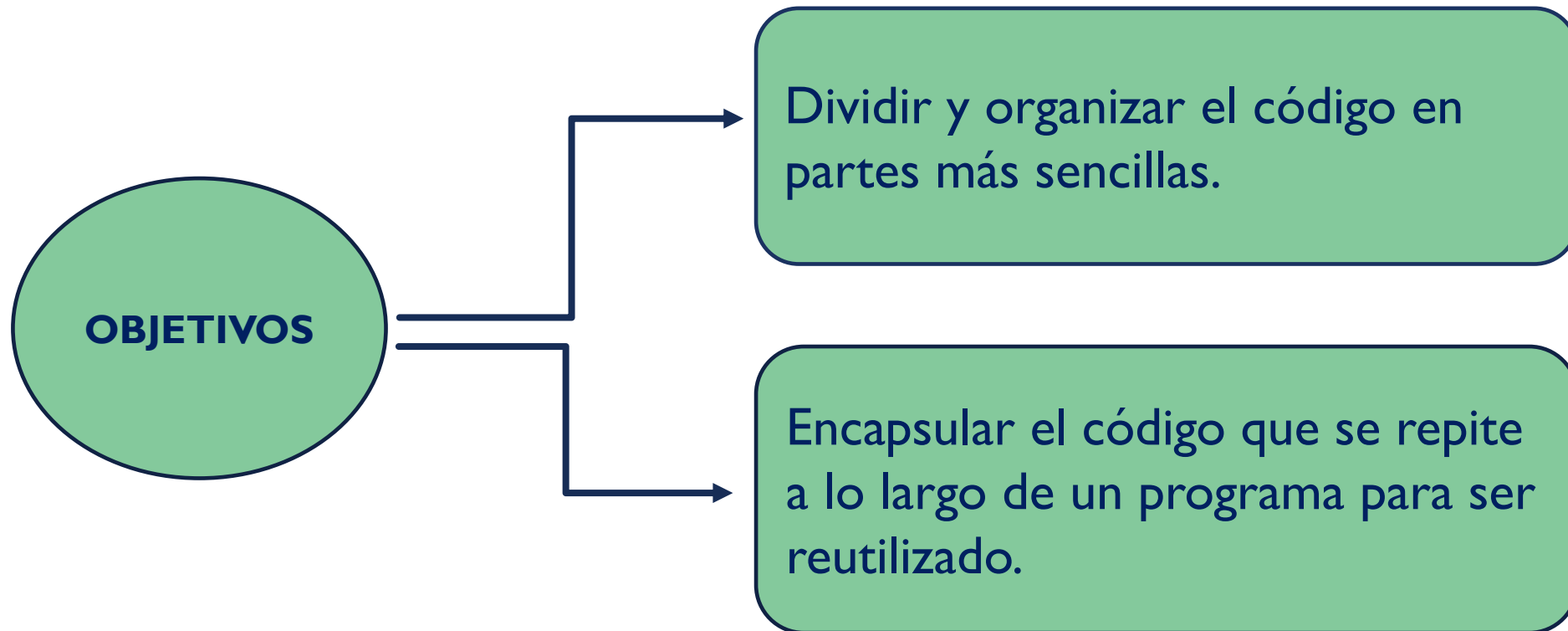
Ley_cobre 

FUNCIONES

Estructuras esenciales de código.



FUNCIONES



FUNCIONES

Sintaxis

def nombre_función (parámetros):

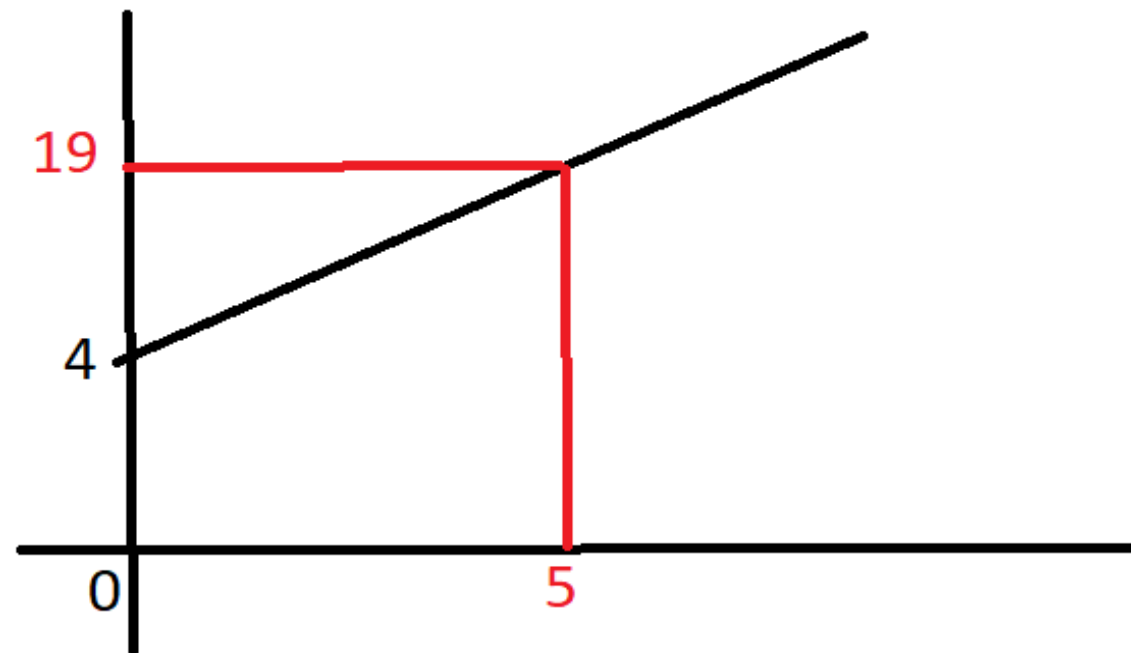
instrucciones

return (opcional)

* Una función siempre deberá ser llamada en algún lugar del código para que se ejecute.

EJEMPLO I

$$f(x) = 3x + 4$$



EJEMPLO II

Escribir una función que calcule la utilidad de un proyecto en minero

$$I = \text{Tonelaje} * \text{ley} * \text{RM} * \text{Precio}$$

$$C = \text{Tonelaje} * \text{Costo Unitario}$$

$$1 \text{ oz} = 31.1035 \text{ g}$$

CÁLCULO DEL BURDEN SEGÚN FÓRMULA KONYA

$$B = 3.15 \times De \times \sqrt[3]{\frac{de}{dr}}$$

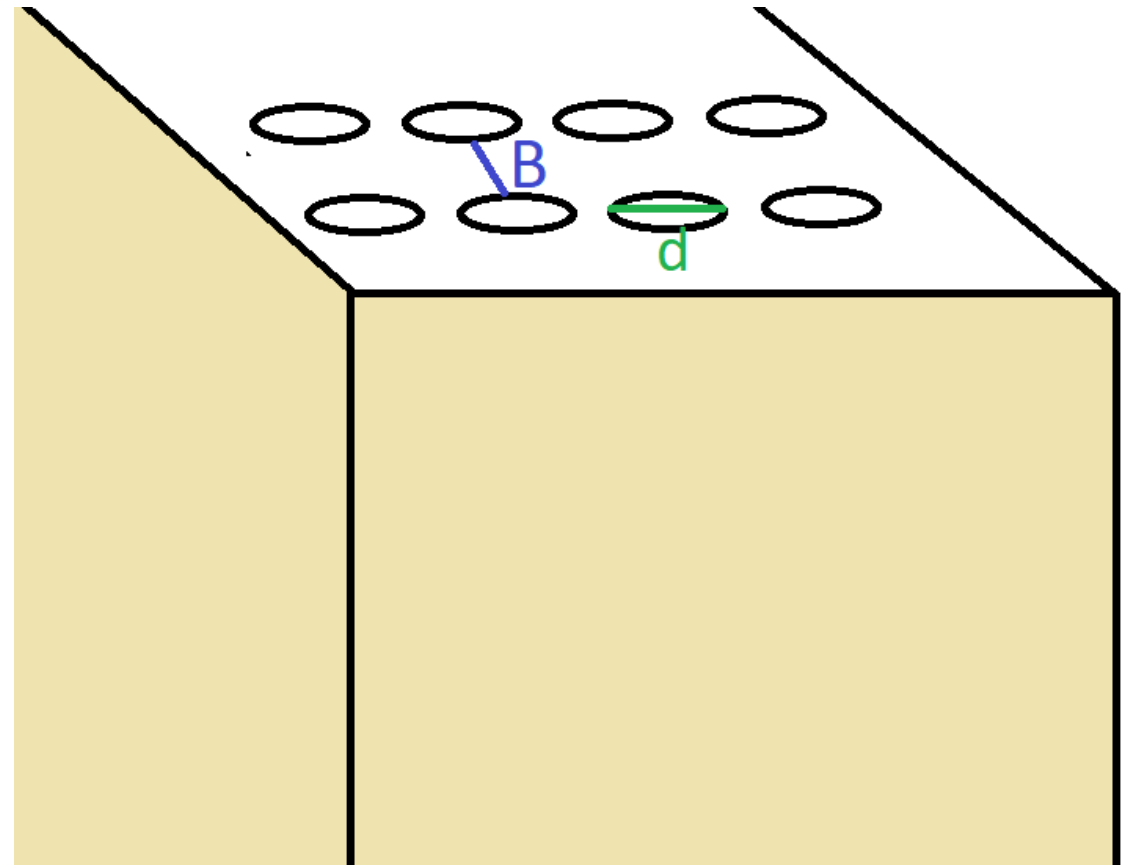
Donde:

B = burden (pies)

De = diámetro del explosivo (pulgadas)

de = densidad del explosivo ($\frac{g}{cm^3}$)

dr = densidad de la roca ($\frac{g}{cm^3}$)



EJEMPLO III

Hallar el burden de la malla de perforación cuadrada según la fórmula de Konya teniendo en consideración los siguientes datos:

Densidad roca: 2.3 g/cm^3 (caliza)

Densidad explosivo: 1.05 g/cm^3 (anfo aluminizado)

Diámetro del explosivo: 10.25 plg

ESTRUCTURAS DE DATOS - LISTAS

Perteneciente a la clase ***LIST***

Una de las estructuras más utilizadas por su naturaleza, dinamismo, fácil manejo y potencia.

Arreglo = Arrays {
tuplas
diccionarios

LAS LISTAS

LISTA_1 = [14, 25, 36, 12, 65]

Nombre de
la lista

Elemento de
la lista

Separador de
elementos

Corchetes de apertura y cierre
de la lista

CARACTERÍSTICAS

- Puede almacenar valores de diferentes tipos de datos.
- El número de elementos que tiene puede variar durante la ejecución de un programa.
- La posición de cada elemento está representado por un número entero llamado índice, el índice empieza desde 0.

LAS LISTAS

```
LISTA_2 = [ 1, 2.5, "Marian", False]
```

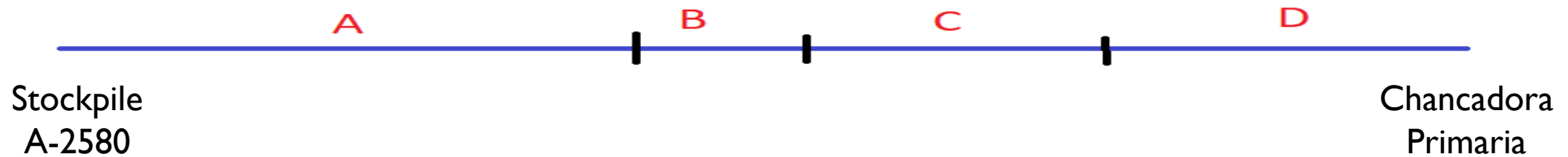
```
LISTA_2 [0] → 1
```

```
LISTA_2 [1] → 2.5
```

```
LISTA_2 [2] → "Marian"
```

```
LISTA_2 [3] → False
```

LISTAS (EJEMPLO)



Representados en las siguientes listas:

Segmentos = ['A', 'B', 'C', 'D']

Distancias = [100, 30, 60, 80] en km

Velocidades = [30, 85, 6, 110] en km/h

Hallar:

- **Imprimir en consola la velocidad del segmento C**
- **Imprimir en consola el tiempo empleado para completar el segmento B**
- **Hallar la velocidad media total**

ESTRUCTURAS DE CONTROL DE FLUJO


Es un bloque de código que permite agrupar instrucciones de manera controlada.

Estas estructuras se caracterizan por **modificar le flujo normal de un programa.**

- Estructuras de control condicionales
- Estructuras de control iterativas (bucles)


ESTRUCTURAS DE CONTROL DE FLUJO

SIN ESTRUCTURAS



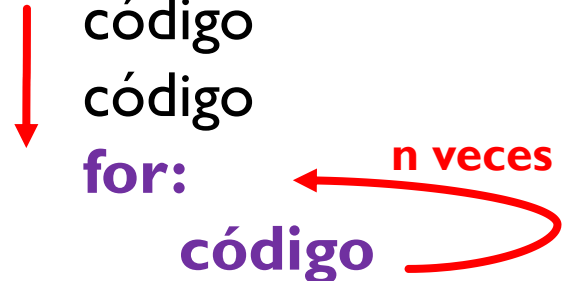
código
código
código
código
código
código
código
código
código
código

CONDICIONAL



código
código
código
if:
 código
else:
 código
código
código

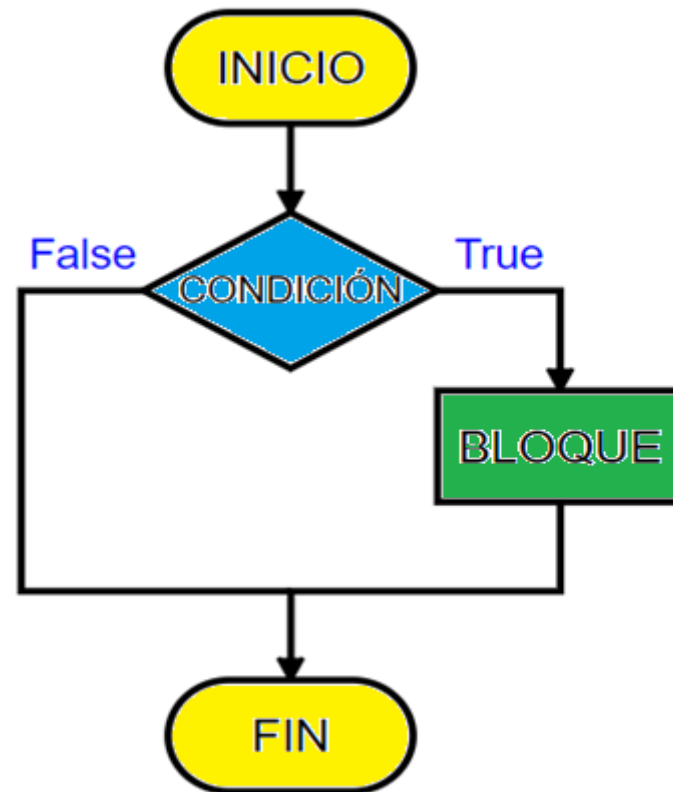
BUCLE



código
código
for: **n veces**
 código
código
código

CONDICIONALES

Permite que un programa ejecute unas instrucciones cuando se cumpla una condición.



CONDICIONALES

SINTAXIS

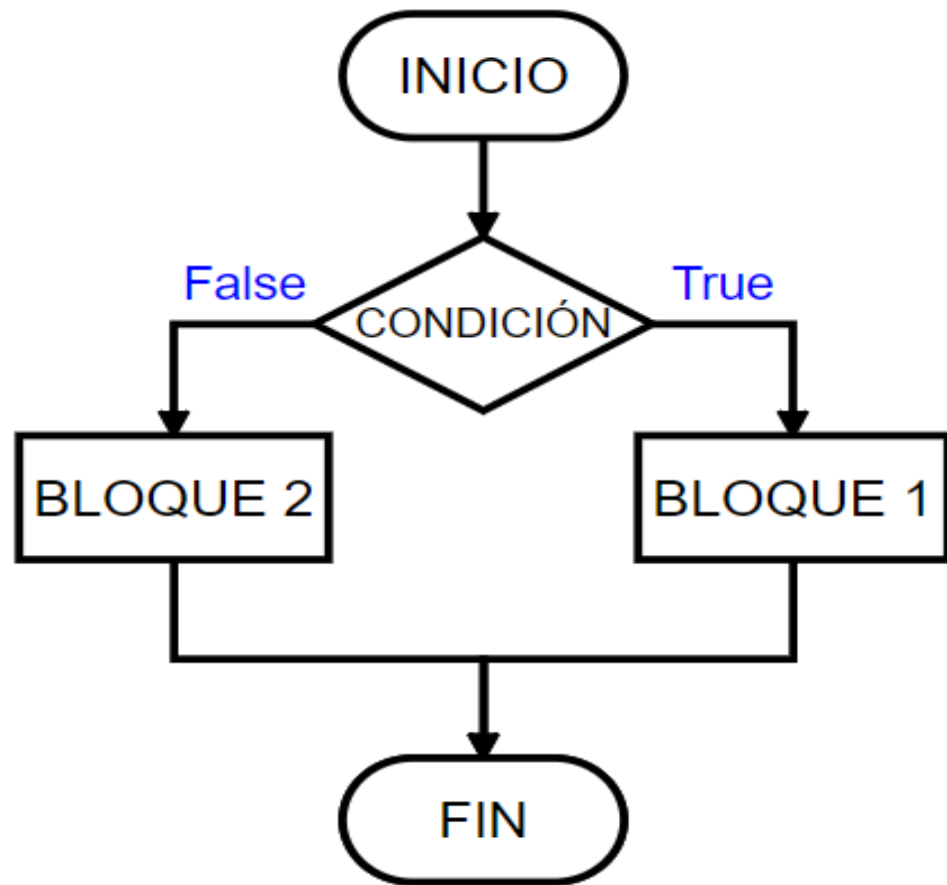
if condición:

código

EJEMPLO I

Supongamos que en una operación minera que produce oro, la ley de corte es de 0.65 g/ton. Elaboraremos un programa en el cual el usuario ingresará el valor de la ley oro y si la ley es mayor o igual que el valor de la ley de corte, el programa imprimirá “Es mineral”.

CONDICIONALES BLOQUE ELSE (“SINO”)



if condición:

código 1

else:

código 2

CONDICIONALES 2

2 bloques if vs if---else

1. No se obliga a Python a evaluar siempre las dos condiciones. Eficiencia en tiempo de procesamiento.
2. Ahorro de código
3. Evita errores (que el valor no caiga en ninguna condición y termine no ejecutándose)

EJEMPLO 2 (SENTENCIA CONDICIONALES ANIDADAS)

Escribiremos un programa para asignar si el material se trata de mineral o desmonte dependiendo de dos propiedades (tipo de mineral y ley de cobre)

	Óxido	Sulfuro
$\geq 2.5\%$	MINERAL	MINERAL
$\geq 0.85\%$	MINERAL	DESMONTE
$< 0.85\%$	DESMONTE	DESMONTE

IF – ELIF – ELSE

Se utiliza cuando se tiene más de dos opciones.

SINTAXIS

if condición 1:
 código 1

elif condición 2:
 código 2

else:
 código 3

EJEMPLO 3 (BLOQUES IF – ELIF – ELSE)

Clasifique el material según la tabla adjunta:

	Óxido	Sulfuro	Argílico
$\geq 2.5\%$	MINERAL_1	MINERAL_2	DESMONTE
$\geq 0.85\%$	MINERAL_1	DESMONTE	DESMONTE
$< 0.85\%$	DESMONTE	DESMONTE	DESMONTE

ESTRUCTURAS DE CONTROL ITERATIVAS (BUCLES)

Son otra herramienta para alterar el flujo normal de un programa. Nos permite repetir una porción de código tantas veces como queramos.

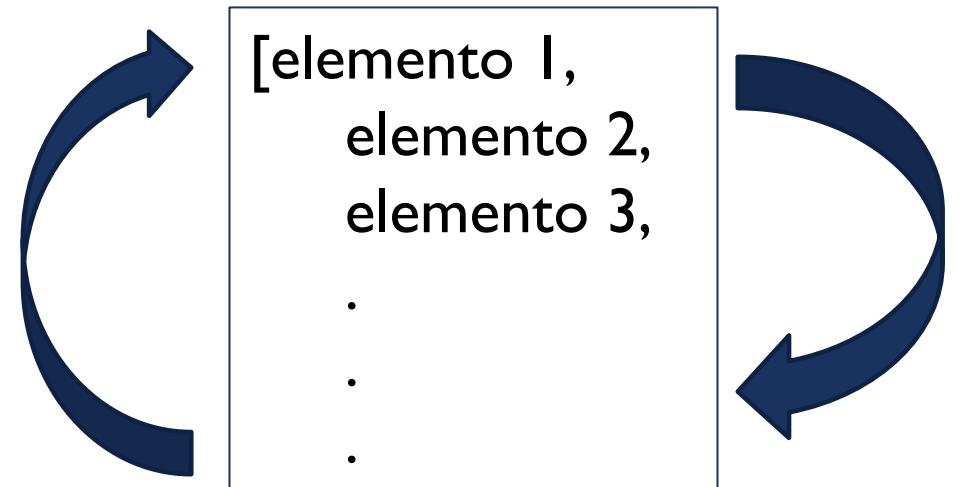
while

for

BUCLE FOR

Se utiliza para recorrer los elementos de un objeto iterable (lista, tupla, conjunto, diccionario) y ejecutar un bloque de código.

En cada paso de la iteración se tiene en cuenta a un único elemento del objeto iterable, sobre el cuál se pueden aplicar una serie de operaciones.



BUCLE FOR - SINTAXIS

```
for <elem> in <iterable>:  
    <Tu código>
```

EJEMPLO

Tonelaje	Ley Au
12,675	0.64
10,489	0.57
20,550	0.32
14,520	0.75
11,254	0.41
22,540	0.12
32,542	0.35
12,546	0.27
9,844	0.84
8,652	0.94

Según el cuadro de tonelajes y ley de oro adjunta, y con los siguientes datos:

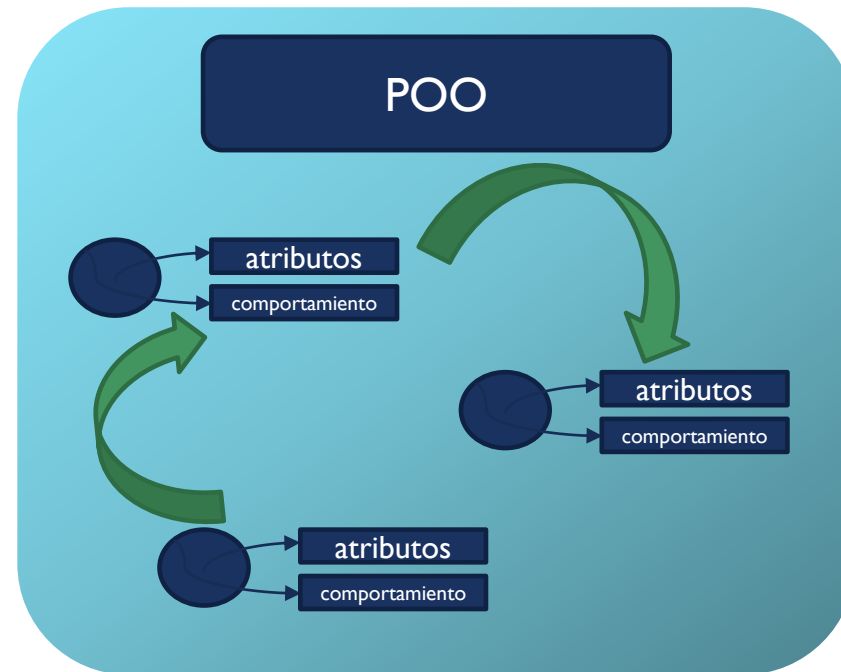
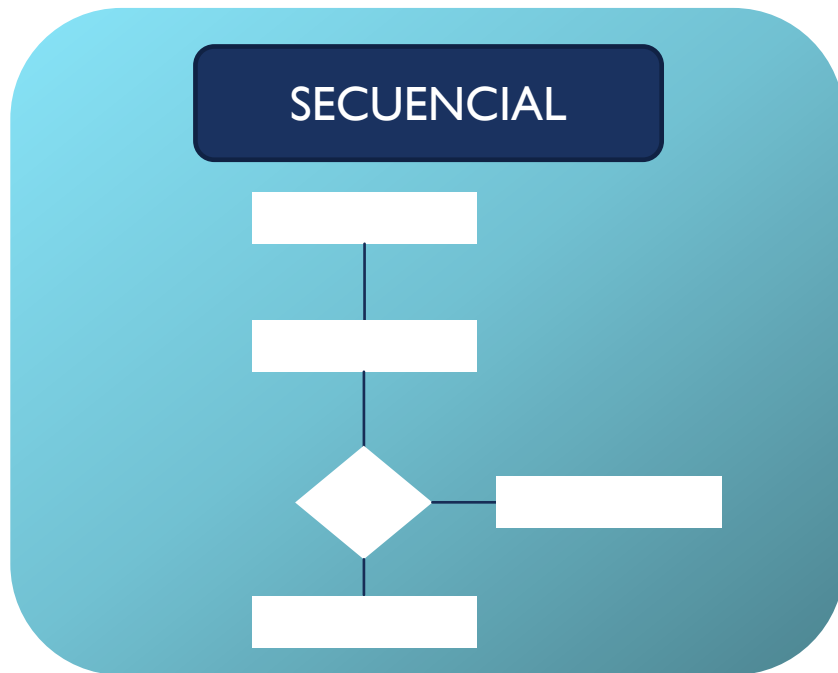
- Ley de corte: 0.45 g/t
- Recuperación metalúrgica: 70%
- Precio del oro: 1740 \$/oz
- Costo de ventas: 120 \$/oz
- Costo de minado: 4.5 \$/t
- Costo de procesamiento: 8 \$/t

Calcular:

- Tonelaje total de mineral y ley media
- Tonelaje total de desmonte
- Ingreso total
- Costo total
- Utilidad

POO - PARADIGMAS DE PROGRAMACIÓN

Paradigma: estilo o forma de programar



POO - OBJETO

Abstracción



Atributos

- Marca
- Capacidad
- Peso cargado
- Peso vacío
- Costo horario

Métodos

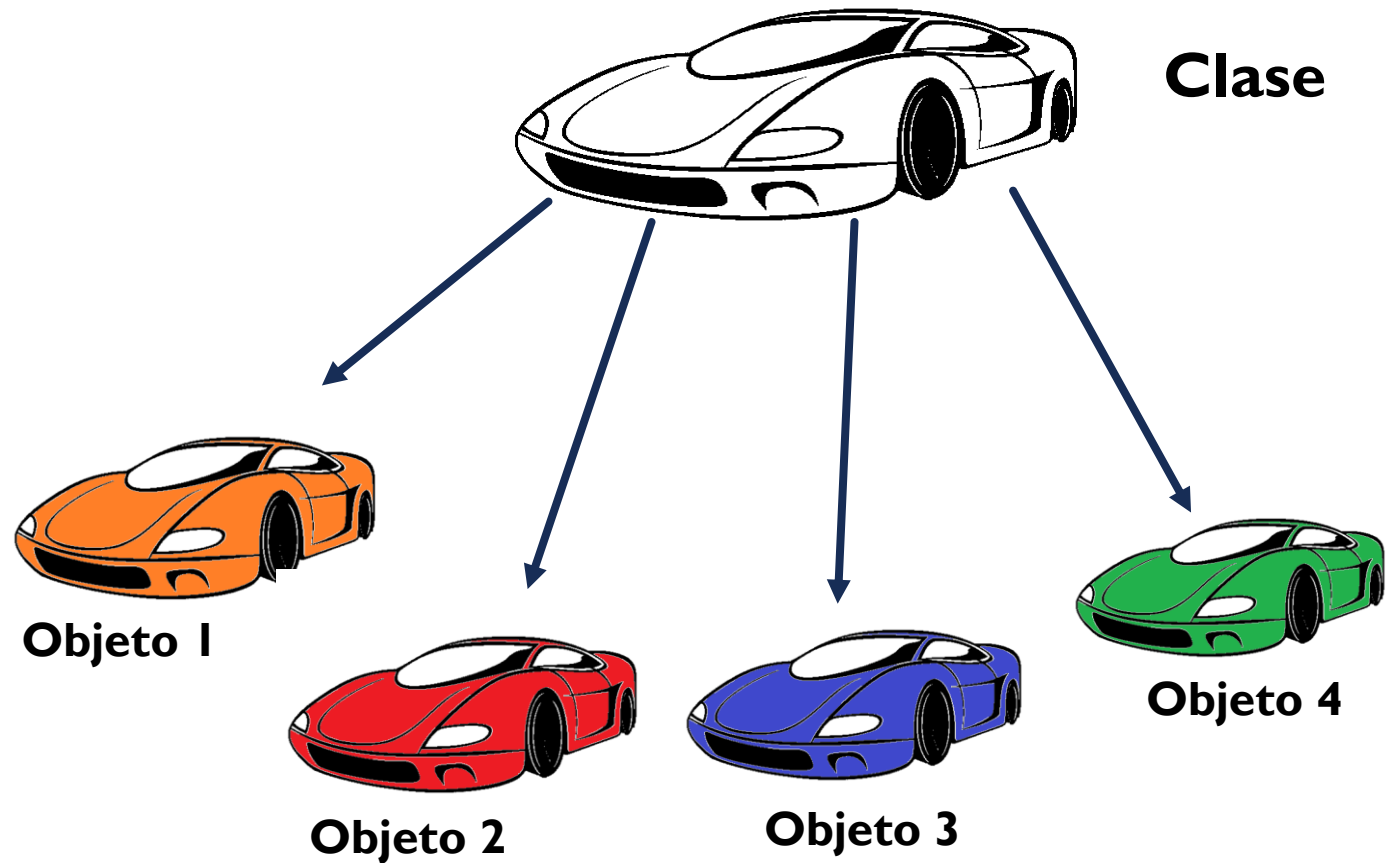
- Acelerar
- Frenar
- Estacionar
- Descargar

POO - CLASE

Todas la propiedades y métodos comunes a los objetos se encapsulan en clases.

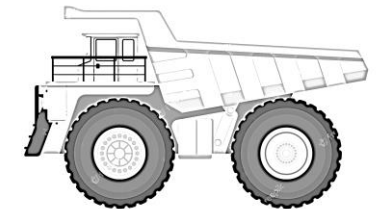
La clase es una plantilla o prototipo para crear objetos.

Instanciación de una clase



POO – INSTANCIA DE CLASE

También llamado EJEMPLAR DE CLASE



Clase
CAMIÓN

Instancia o ejemplarización de clase

POO – NOMENCLATURA DEL PUNTO

miAuto . color = “azul”
miAuto . peso = 1500
miAuto . alto = 150

PROPIEDADES

miAuto . arranca()
miAuto . frena()
miAuto . acelera()

MÉTODOS

POO – ENCAPSULACIÓN

- Es uno de los pilares de la programación orientada a objetos
- Permite regular el acceso a los métodos y atributos de una clase
- En cierta manera, enmascara la complejidad de una clase

EN PYTHON NO EXISTE. ES UN ESTILO. ES UNA NORMA. NO ES FUNCIONAL

DICCIONARIOS

Estructuras de datos

Clave : Valor

key value

Sintaxis

nombre_dic = { }

En esta estructura de datos se pueden almacenar valores enteros, flotantes, cadenas, clases, listas, tuplas e inclusive diccionarios.

DICCIONARIOS

Estructura de datos que nos permiten almacenar diferentes tipos de datos

CLAVE :VALOR

Los elementos almacenados NO ESTAN ORDENADOS

MODULOS

Un archivo con extensión .py .pyc que posee su propio espacio de nombres y que puede contener variables, funciones, clases e incluso otros módulos.

Sirven para organizar y reutilizar el código (POO)

MODULOS

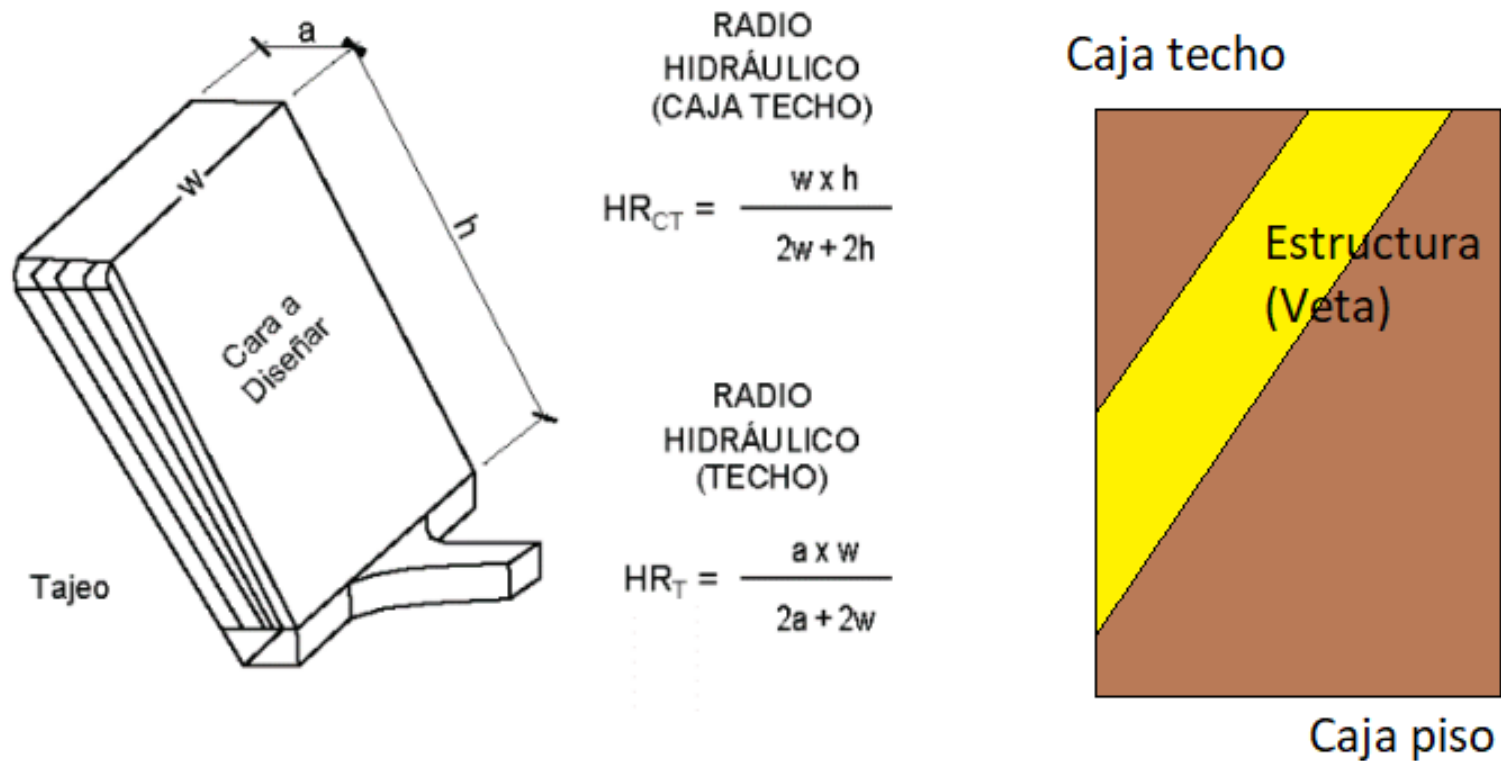
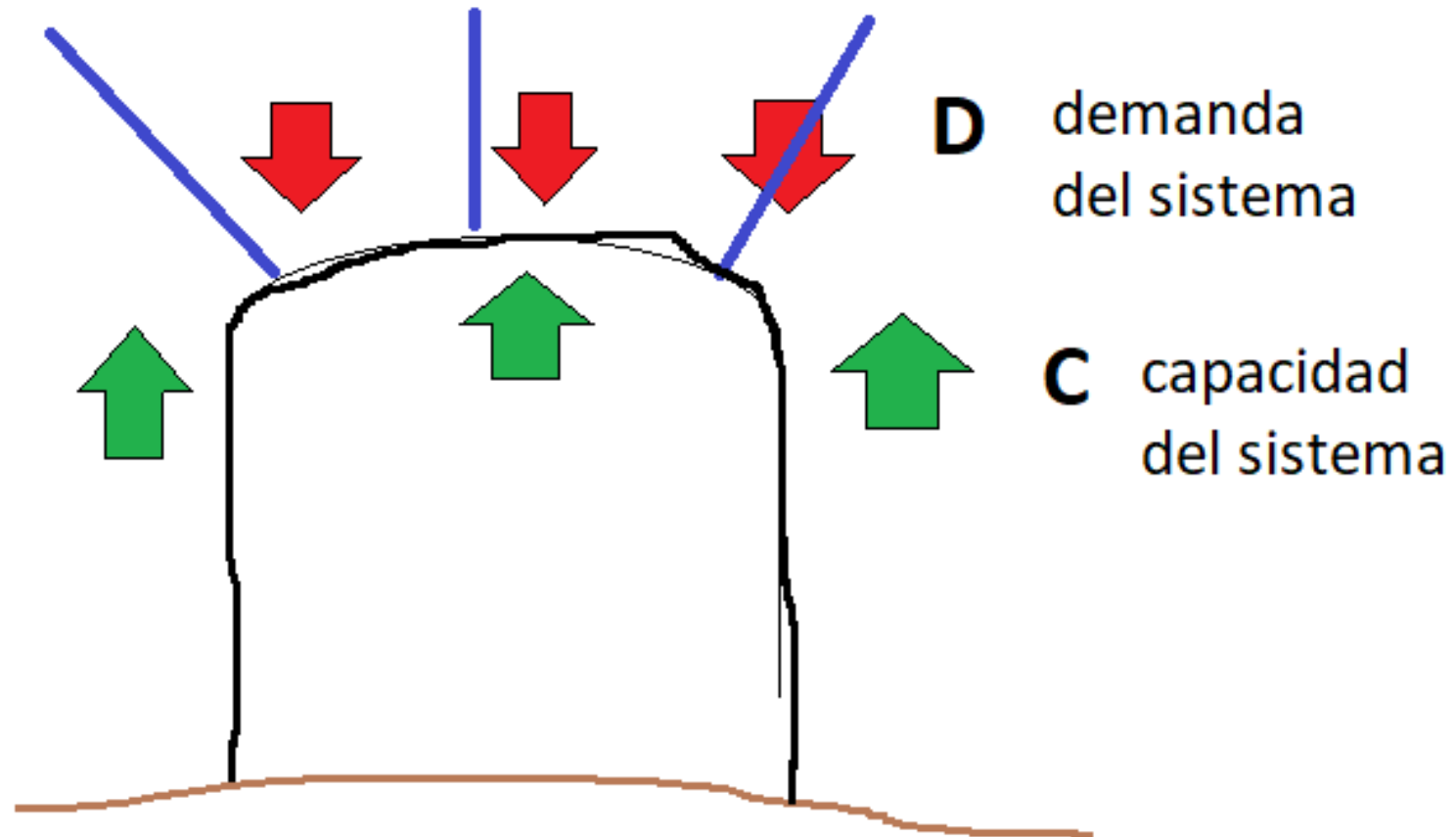


Figura 7-8. Radio Hidráulico (RH)

MODULOS



MODULO OS

El módulo OS en Python proporciona y expone los detalles y la funcionalidad del sistema.

Muy útil para automatizaciones elaborando rutinas con los archivos y carpetas del sistema operativo.

ARCHIVOS EXTERNOS

Persistencia de Datos

Posibilidad de guardar los datos que estamos trabajando en el programa, se puedan guardar.

Hasta ahora, cuando se cierra el programa se borra la información.

ARCHIVOS EXTERNOS - OPCIONES

- Guardar la información en archivo externo
- Guardar la información en BBDD

ARCHIVOS EXTERNOS - PASOS

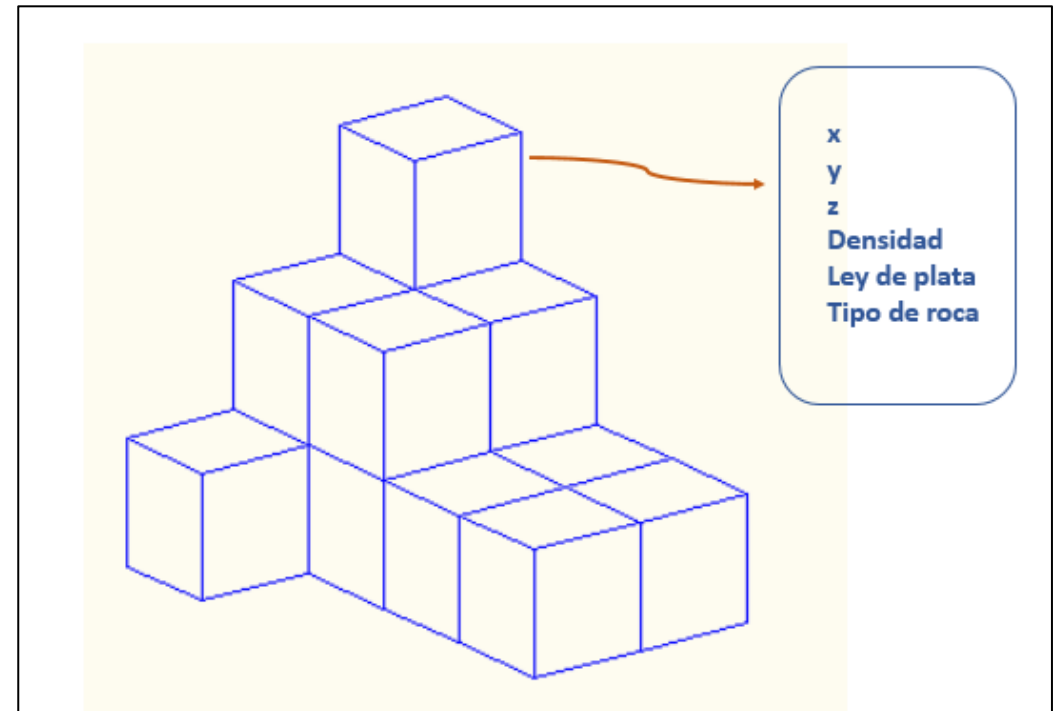
1. Crear archivo
2. Abrir archivo
3. Manipular archivo
4. Cerrar archivo

LIBRERÍAS

IO
CSV

MODELO DE BLOQUES

Un modelo de bloques es una representación simplificada de un yacimiento mineral y sus alrededores, que se puede considerar como una pila de “ladrillos” generados por computadora que representan pequeños volúmenes de roca en un depósito (compuesto por mineral y desechos). Cada “ladrillo” o celda contiene estimaciones de datos, como la ley del elemento, la densidad y otros valores de entidades geológicas o relacionados a factores de ingeniería.



EJERCICIOS

EJERCICIO 1: Encontrar la ley media de Au que tiene el modelo de bloques compartido

EJERCICIO 2: con los siguientes parámetros, encontrar el beneficio de minar todo el mineral del modelo. Considerar que no se debe minar el desmonte

$C_{\text{Minado}}: 2.50 \text{ \$/t}$

$C_{\text{Procesam}}: 4.30 \text{ \$/t}$

$\text{Precio}: 1600 \text{ \$/oz}$

$C_{\text{Venta}}: 250 \text{ \$/oz}$

$RM: 72\%$

BASE DE DATOS - ELEMENTOS

Equipos - BaseEjemplo - OpenOffice.org Table Data View

Archivo Editar Ver Insertar Herramientas Ventana Ayuda

IdEquipo	Nombre
1	LocosDelBalón
2	Imparables
3	SinMiedo FC
4	Los mejores
5	Luchadores
<	< Campo auto

TABLA

GESTOR DE BASE DE DATOS

Es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos. A través de este se maneja todo acceso a la base de datos con el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones.

ORACLE[®]



SQL – STRUCTURED QUERY LANGUAGE

Es el lenguaje de consulta más utilizado para manejar-administrar una base de datos

Consulta: es un objeto que proporciona una visión personal de los datos almacenados en las tablas ya creadas

MODELO DE BASE DE DATOS

Es una representación, usualmente gráfica, de estructuras de datos.

- Modelo de base de datos jerárquicos
- **Modelo relacional**
- Modelo de red
- Modelo de base de datos orientado a objetos
- **Modelo entidad relación**

MODELO ENTIDAD - RELACIÓN

Es el modelo de base de datos más sencillo.

Entidad

Representan cosas u objetos (ya sean reales o abstractos), que se diferencian claramente entre si.

Atributos

Los atributos definen o identifican las características de la entidad (es el contenido de la entidad).

MODELO ENTIDAD - RELACIÓN

ENTIDADES

Camiones

Polígonos de
minado

Descargas

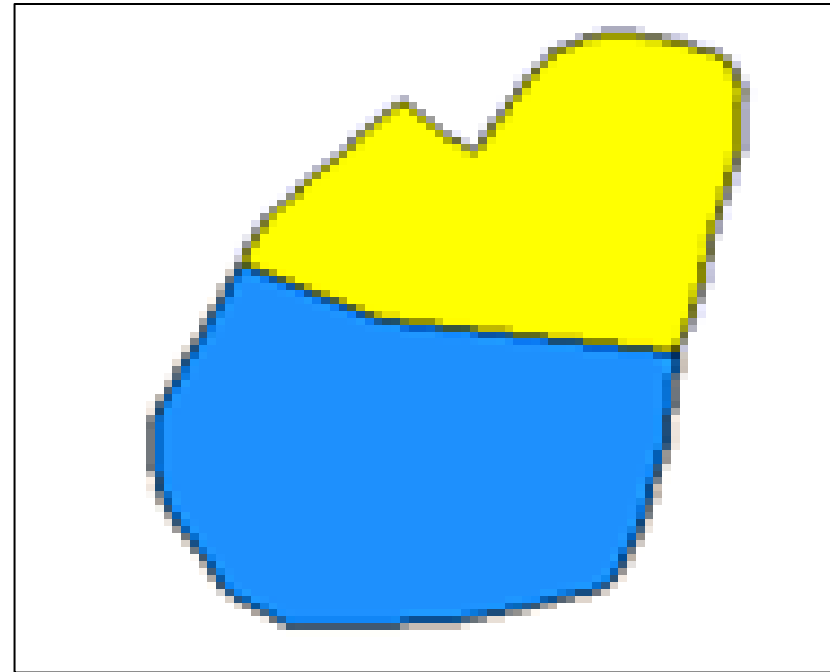
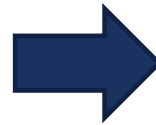
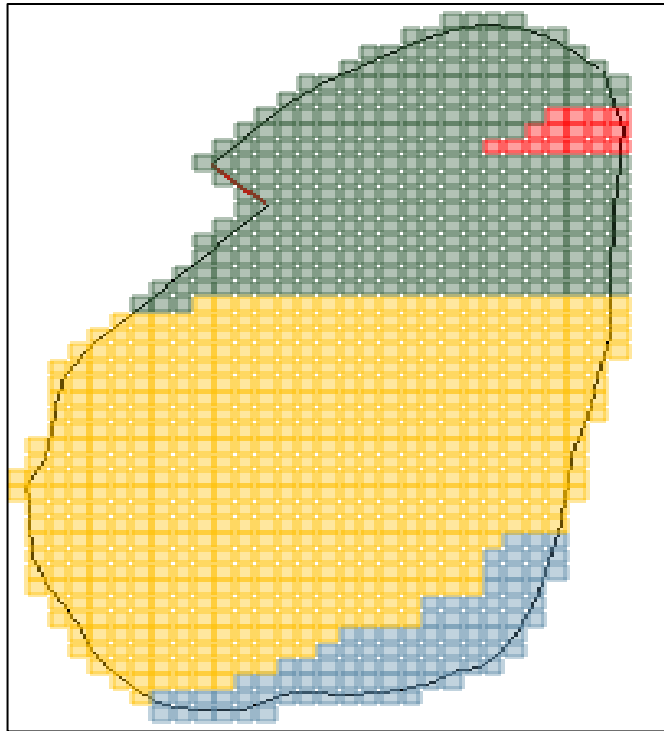
ATRIBUTOS

- N° Camión
- Marca
- Modelo
- Capacidad

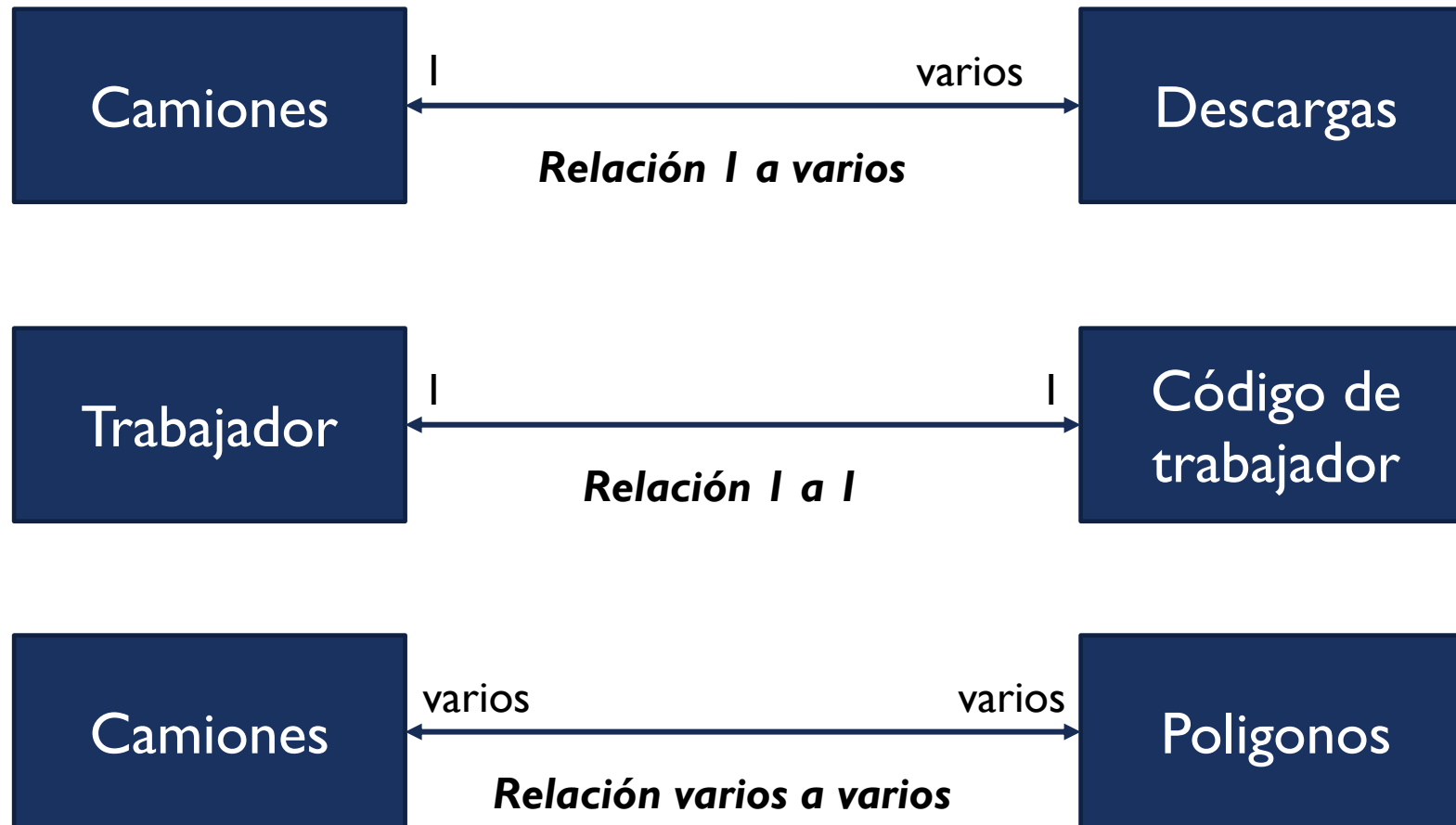
- Código
- Tipo
- Tonelaje
- Ley

- ID
- Fecha
- Destino
- N° Camión
- Código Pol

POLIGONO DE MINADO



MODELO ENTIDAD - RELACIÓN



TABLAS

	A	B	C	D	E
1	CAMION	ESTADO	ODOMETRO	CAPACIDAD	
2	V01	Operativo	178452	135	
3	V02	Operativo	225986	135	
4	V03	Operativo	98452	135	
5	V04	Stand-by	82459	135	
6	V05	Mant. Prev	45218	135	
7					
8					
9					
10					

FILA O REGISTRO

TABLA

COLUMNA O CAMPO

TABLAS – CLAVES

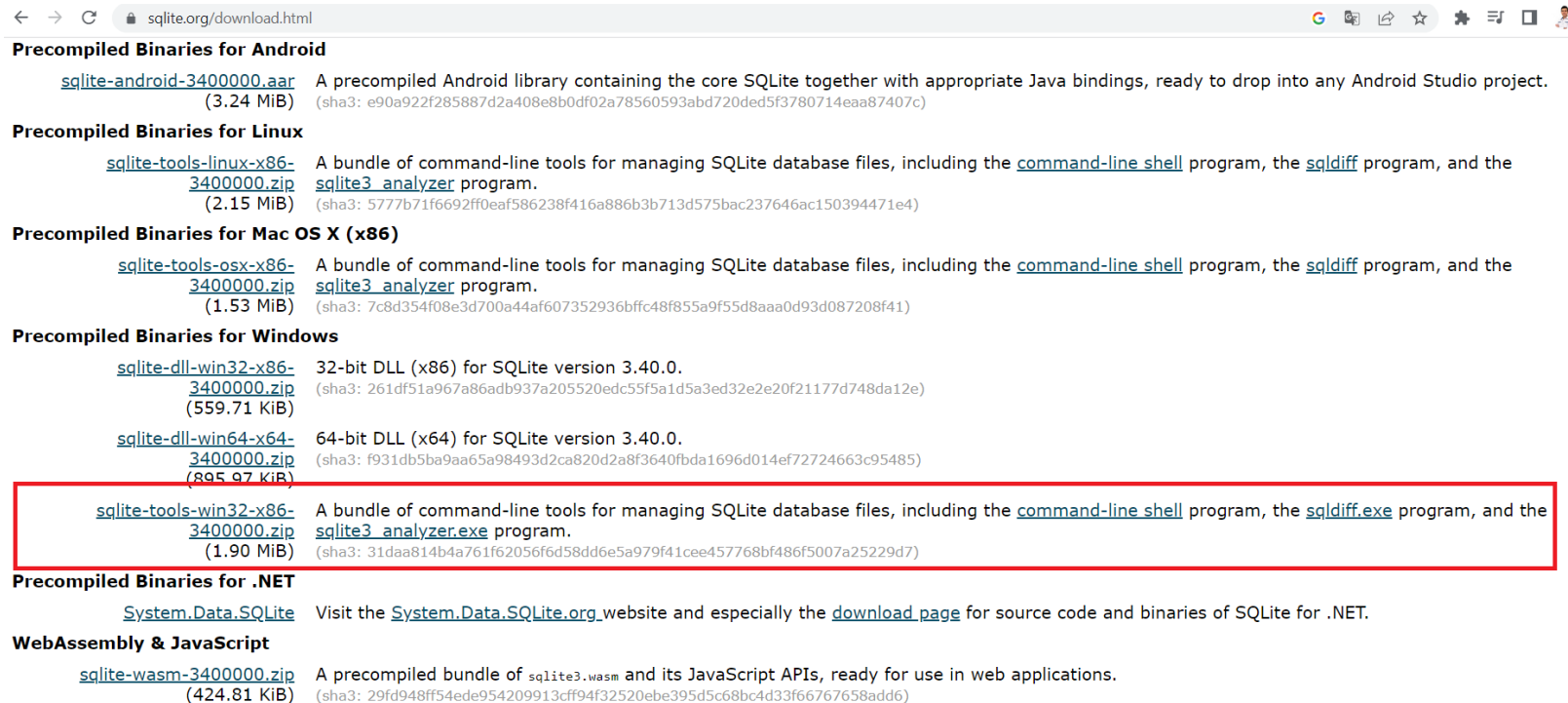
Claves: elementos que impiden duplicidad de registros. Una de las grandes desventajas que presentan otros modelos de organización y gestión de bases de datos.

Claves primarias: son los atributos (columnas) según el tipo de relación que se ha definido en la tabla.


Claves foráneas o externas: son claves presentes en tablas que no han concebido la misma.

INSTALACIÓN SQLITE 3

Link: <https://sqlite.org/download.html>



The screenshot shows the SQLite.org download page with the following content:

← → ↻ sqlite.org/download.html 

Precompiled Binaries for Android

[sqlite-android-3400000.aar](#) (3.24 MiB) A precompiled Android library containing the core SQLite together with appropriate Java bindings, ready to drop into any Android Studio project. (sha3: e90a922f285887d2a408e8b0df02a78560593abd720ded5f3780714eaa87407c)

Precompiled Binaries for Linux

[sqlite-tools-linux-x86-3400000.zip](#) (2.15 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff](#) program, and the [sqlite3_analyzer](#) program. (sha3: 5777b71f6692ff0eaf586238f416a886b3b713d575bac237646ac150394471e4)

Precompiled Binaries for Mac OS X (x86)

[sqlite-tools-osx-x86-3400000.zip](#) (1.53 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff](#) program, and the [sqlite3_analyzer](#) program. (sha3: 7c8d354f08e3d700a44af607352936bffc48f855a9f55d8aaa0d93d087208f41)

Precompiled Binaries for Windows

[sqlite-dll-win32-x86-3400000.zip](#) (559.71 KiB) 32-bit DLL (x86) for SQLite version 3.40.0. (sha3: 261df51a967a86adb937a205520edc55f5a1d5a3ed32e20f21177d748da12e)

[sqlite-dll-win64-x64-3400000.zip](#) (895.97 KiB) 64-bit DLL (x64) for SQLite version 3.40.0. (sha3: f931db5ba9aa65a98493d2ca820d2a8f3640fbd1696d014ef72724663c95485)

[sqlite-tools-win32-x86-3400000.zip](#) (1.90 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff.exe](#) program, and the [sqlite3_analyzer.exe](#) program. (sha3: 31daa814b4a761f62056f6d58dd6e5a979f41cee457768bf486f5007a25229d7)

Precompiled Binaries for .NET

[System.Data.SQLite](#) Visit the [System.Data.SQLite.org](#) website and especially the [download page](#) for source code and binaries of SQLite for .NET.

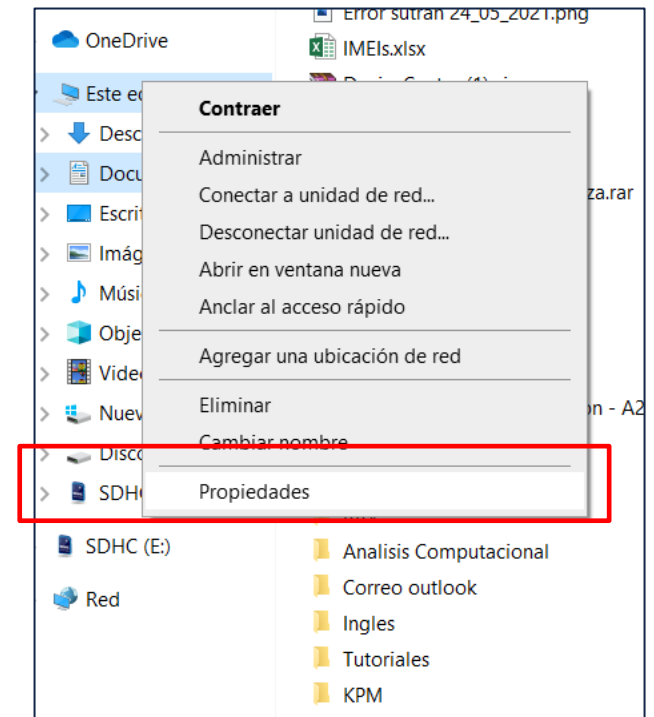
WebAssembly & JavaScript

[sqlite-wasm-3400000.zip](#) (424.81 KiB) A precompiled bundle of `sqlite3.wasm` and its JavaScript APIs, ready for use in web applications. (sha3: 29fd948ff54ede954209913cff94f32520e8e395d5c68bc4d33f66767658add6)

Seleccionar esta opción

INSTALACIÓN SQLITE 3

1. Crear la carpeta DBSQLite en el disco C
2. Mover el archivo descargado a la carpeta creada
3. Descomprimir el archivo
4. Click derecho sobre este equipo
5. Seleccionar Propiedades



INSTALACIÓN SQLITE 3

6. Seleccionar Configuración avanzada del sistema
7. Seleccionar Variables de entorno

Configuración relacionada

[Configuración de BitLocker](#)

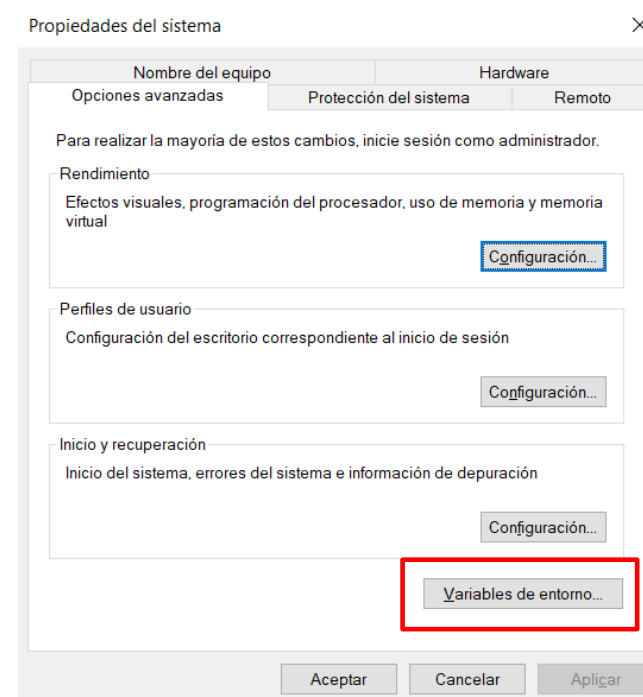
[Administrador de dispositivos](#)

[Escritorio remoto](#)

[Protección del sistema](#)

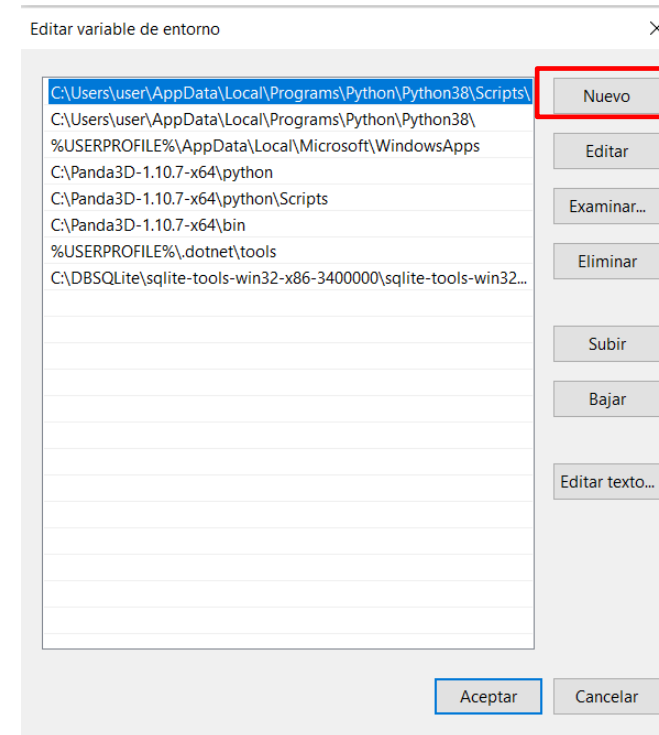
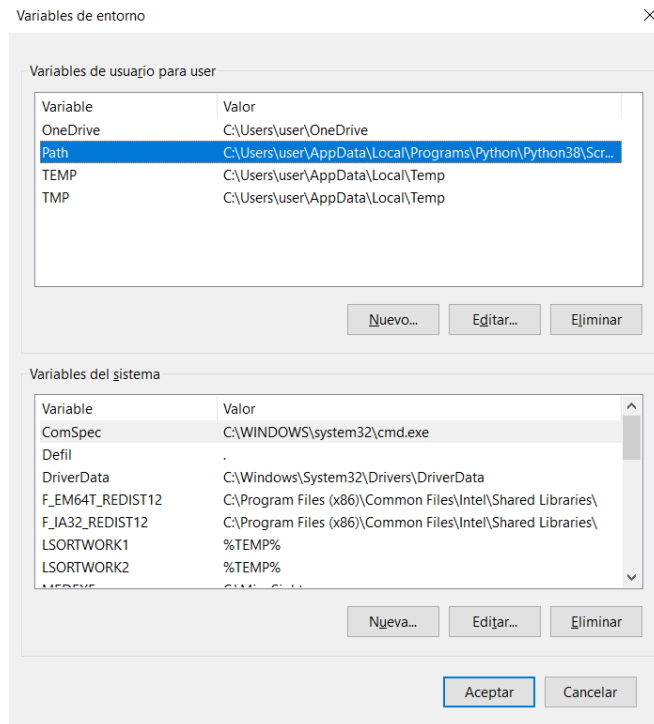
[Configuración avanzada del sistema](#)

[Cambiar el nombre de este equipo
\(avanzado\)](#)



INSTALACIÓN SQLITE 3

9. Doble click sobre la Variable Path
10. Seleccionar Nuevo



INSTALACIÓN SQLITE 3

11. Escribir la ruta de la carpeta descargada. En mi caso es:

<C:\DBSQLite\sqlite-tools-win32-x86-3400000\sqlite-tools-win32-x86-3400000>

12. Presionar Aceptar en todas las ventanas

INSTALACIÓN DB BROWSER

Link: <https://sqlitebrowser.org/dl/>

Downloads

(Please consider sponsoring us on Patreon 😊)

Windows

Our latest release (3.12.2) for Windows:

- [DB Browser for SQLite - Standard installer for 32-bit Windows](#)
- [DB Browser for SQLite - .zip \(no installer\) for 32-bit Windows](#)
- [DB Browser for SQLite - Standard installer for 64-bit Windows](#)
- [DB Browser for SQLite - .zip \(no installer\) for 64-bit Windows](#)

Windows PortableApp

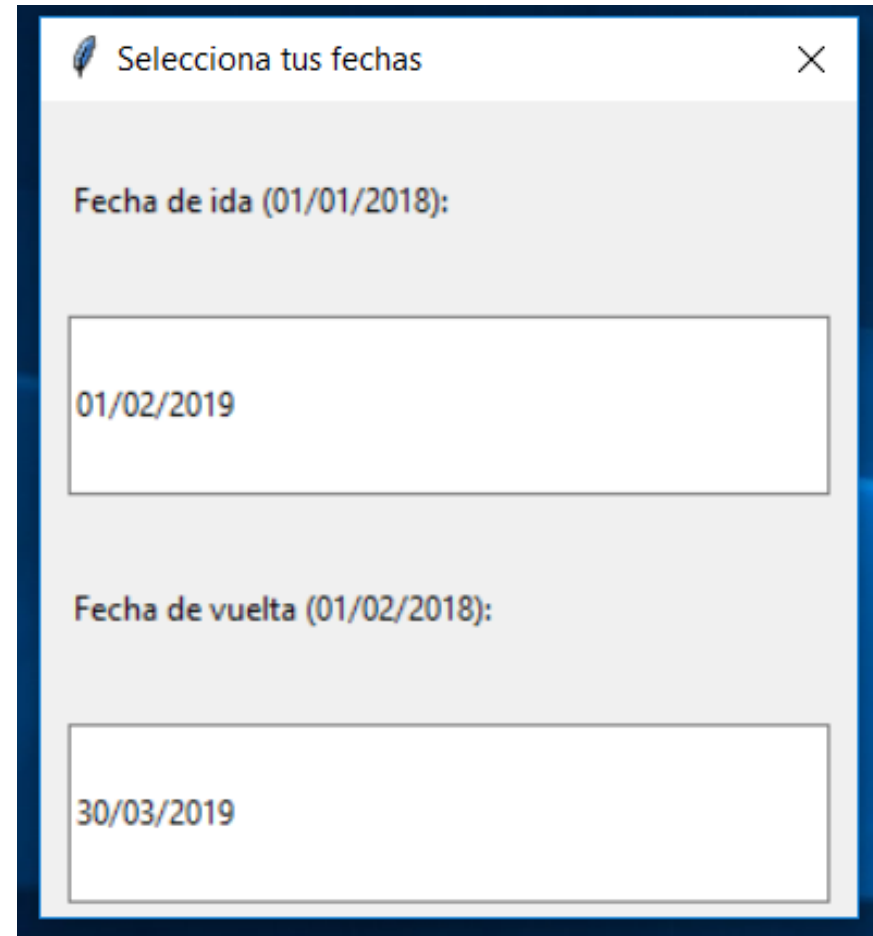
- [DB Browser for SQLite - PortableApp](#)

INTERFACES GRÁFICAS

INTERMEDIARIO

Programador - Usuario

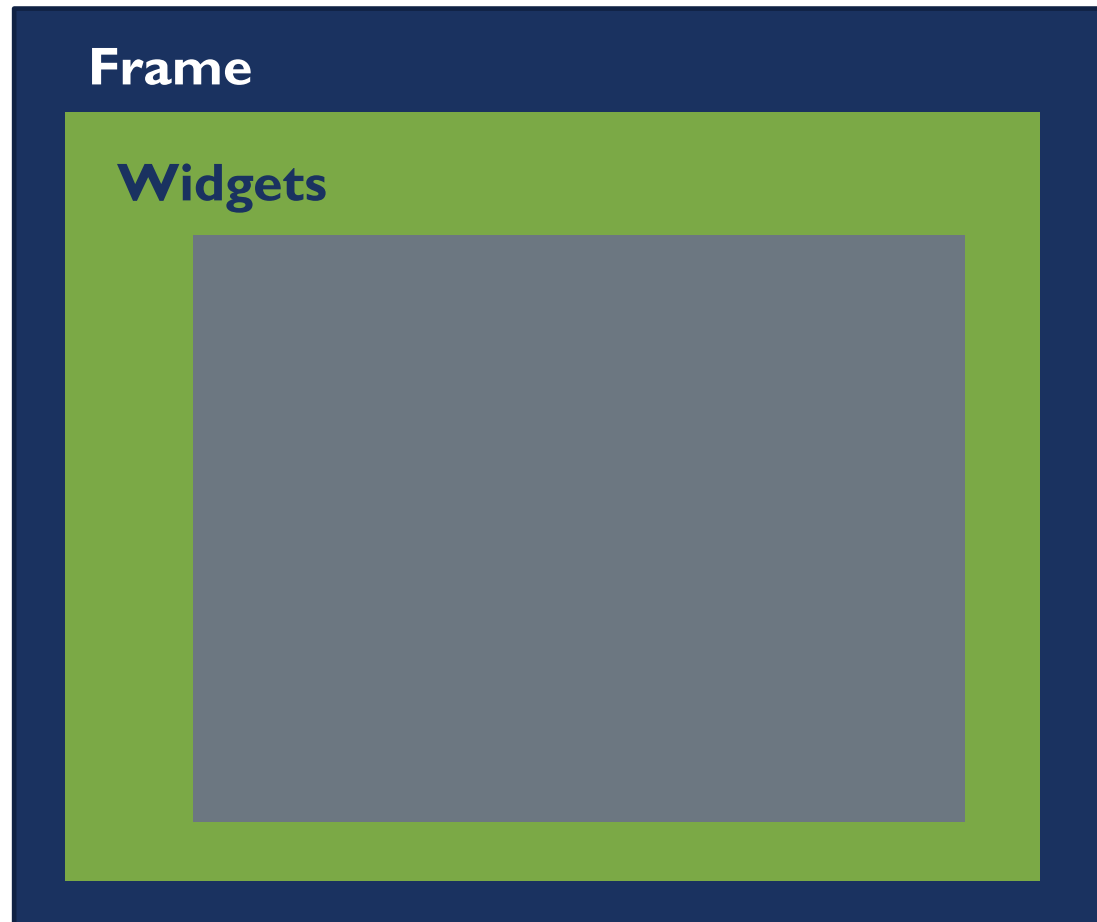
Librería: Tkinter



The image shows a Tkinter dialog box titled "Selecciona tus fechas" (Select your dates). The dialog has a close button (X) in the top right corner. It contains two input fields. The first field is labeled "Fecha de ida (01/01/2018):" and contains the date "01/02/2019". The second field is labeled "Fecha de vuelta (01/02/2018):" and contains the date "30/03/2019".

INTERFACES GRÁFICAS - ESTRUCTURA

Raíz





GRACIAS

ALGUIEN@EJEMPLO.COM