



universidad
cenfotec_
La U de la informática

Principios de Programación.

Variables, tipos de datos, valores y estándares.

Objetivos

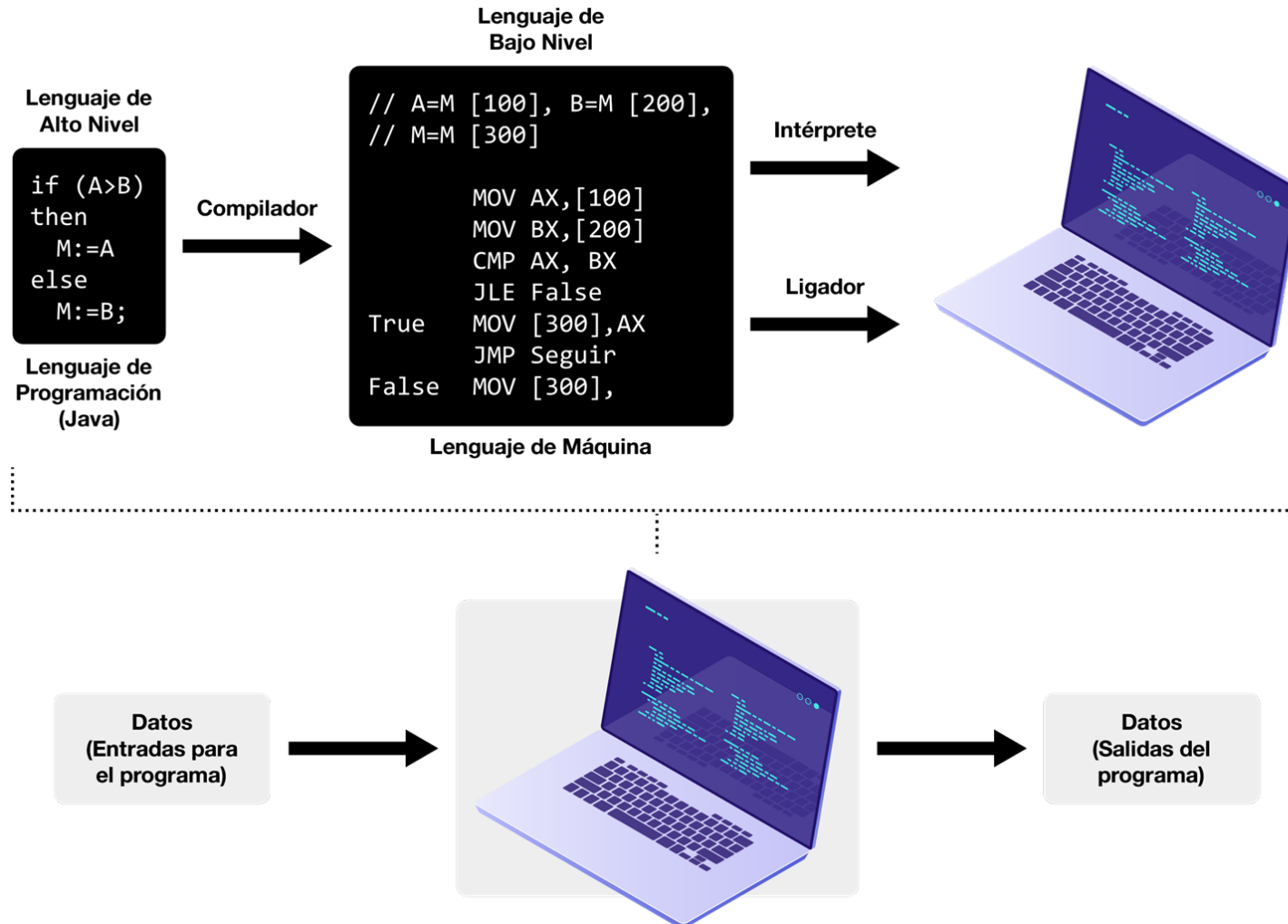
- Aprender la diferencia entre variable, tipo de dato y valor.
- Aprender el estándar para definir variables en Python.
- Aprender los diferentes tipos primitivos en Python.
- Representar en el lenguaje de programación Python cualquier algoritmo con estructuras secuenciales (operaciones básicas).

Repaso

Es importante que recuerde lo siguiente:

- Conceptos básicos de computación como compilador, intérprete y lenguaje de programación.
- Definición de algoritmo y programa.
- Los pasos necesarios para crear un programa estructurado.

Repaso: Conceptos básicos de computación



Repaso: Algoritmos

Un **algoritmo** es una secuencia de instrucciones que, realizados con fidelidad, darán como resultado una tarea realizada.

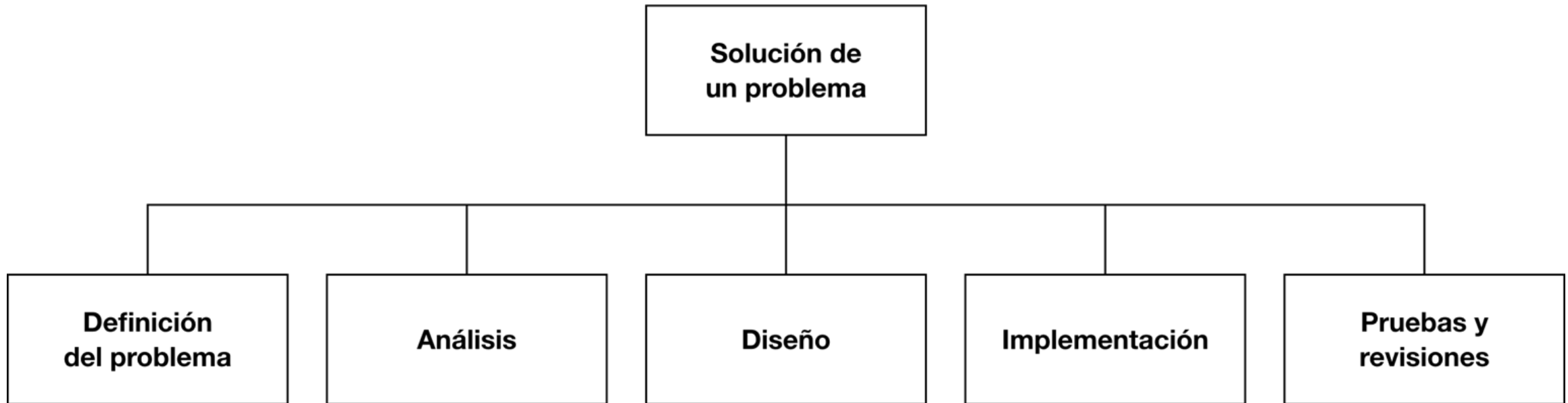
Los algoritmos son finitos, lo que significa que estas instrucciones tienen un inicio y un final.

Solicitar un préstamo:

1. Presentar documentos.
2. Esperar resolución.
3. Aprobaron el préstamo:
 Sí: Formalizar.
 No: Retirar documentos.



Repaso: Pasos para crear un programa estructurado



DOCUMENTACIÓN

The background is a dark blue gradient. It features a pattern of glowing green binary digits (0s and 1s) that are slightly out of focus, creating a sense of depth. Overlaid on this are several soft, out-of-focus circular bokeh lights in a teal or light blue color. A white rectangular border is positioned in the center of the image, enclosing the main text.

Variables, tipos de datos, valores y estándares

Tipos de datos simples en Python



A la hora de estructurar un **algoritmo** es posible que requiera utilizar datos muy distintos, como números enteros, números decimales, palabras, entre otros. Por ejemplo, para guardar la edad de una persona se requiere un número entero, para su salario se requerirá de un número con decimales y para su nombre una palabra.

Cada uno de estos datos tiene una representación concreta dentro de Python, sobre los cuales podemos ejecutar varias operaciones para interactuar y modificar sus valores.

Estas representaciones llevan como nombre **tipos de datos**.

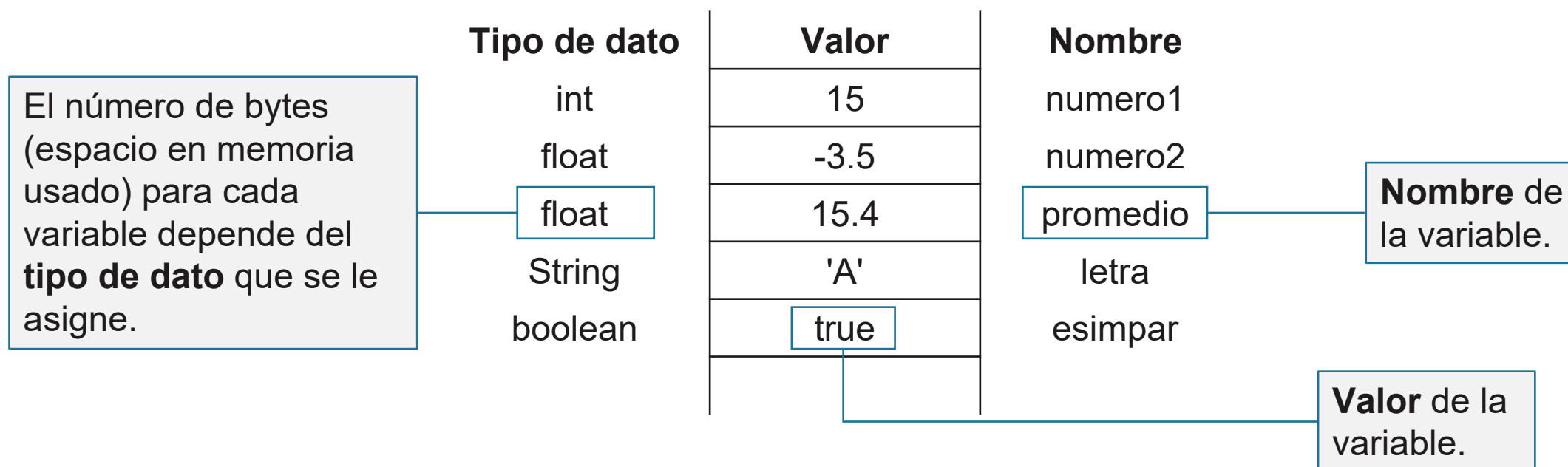
Tipos de datos en Python

Los diferentes tipos de datos en Java podemos clasificarlos en:

- Tipos de datos **simples o primitivos**: como los enteros, puntos flotantes, caracteres y de valor booleano.
- Tipos de datos **compuestos**: como los arreglos y matrices, los cuales serán estudiados posteriormente.
- Tipos de datos **definidos por el programador**: como las clases y sus relaciones, los cuales serán estudiados posteriormente.

Variables, tipos de datos y valores

Una vez se define el tipo de dato necesario en un algoritmo, este lo puede representar en un programa utilizando variables. Una **variable** es un espacio en memoria identificada por un nombre, que contiene un dato de cierto tipo (entero, real, carácter, etc.) y el contenido de la variable en un momento dado del tiempo es su **valor**.



Variables y tipos de datos simples en Python

En la siguiente tabla, en la primera columna se muestran algunos de los tipos de datos simples en Python, su descripción y ejemplos de declaración de variables con los respectivos tipos de datos:

Tipo	Descripción	Ejemplos
int	Números enteros.	<code>edad = 33</code> <code>anno_nacimiento = 1987</code> <code>cantidad_mascotas = 3</code>
float	Números con decimales o de punto flotante.	<code>imc = 23.40</code> <code>estatura = 1.70</code>
String	Caracteres o letras (se puede usar comillas simples o dobles).	<code>sistema_operativo = "Linux"</code> <code>nombre_completo = 'Ada Lovelace'</code>
boolean	Valores lógicos, verdadero o falso. <i>(Tanto los valores de True y False inician con mayúscula en Python)</i>	<code>buen_estudiante = True</code> <code>hace_calor = False</code>

Declaración de una variable

Para poder utilizar una variable es necesario **declararla**. Lo cual consiste en indicarle a Python el **identificador** (nombre) de la variable.

```
nombre
```

Esto separa un espacio en memoria y le permite a Python utilizar la variable en futuras operaciones. Al declararla, se le puede dar un valor inicial, esto se conoce como **inicialización**.

```
nombre = ""
```

Una vez declarada una variable, darle un valor en cualquier momento se conoce como **asignación** y si le asigna un nuevo valor este se sobre escribe y el valor anterior se pierde.

```
nombre = "Ada Lovelace"
```

Ejemplo de declaración de variables

```
1  nombre = ""  
2  edad = 0  
3  estatura = 0.0
```

En Python es requerido realizar el proceso de declaración y de inicialización como parte de la misma instrucción.

Python infiere el tipo de dato asociado a la variable, gracias al valor con el que se inicializó. Esto se conoce como tipado débil de datos, ya que no hay que indicar explícitamente el nombre del tipo.

Por lo tanto se tiene que la variable ***nombre*** es de tipo *String*(texto), la variable ***edad*** es de tipo entera y la variable ***estatura*** es de tipo punto flotante.

Ejemplo de declaración de variables

```
1  nombre = ""
2  edad = 0
3  estatura = 0.0
4
5  nombre = "Ada Lovelace"
6  edad = 1852 - 1815
7  estatura = 1.62
```

Una vez que se **declara** e **inicializa** una variable, es posible modificar su contenido **asignando** un nuevo valor a la variable.

Es posible asignar cualquier valor que concuerde con el tipo de dato de la variable, como en el caso de la variable **nombre** se asigna un *String* concreto, o el resultado de una operación matemática, como en el caso de la variable **edad**.

Sintaxis del lenguaje

De la misma forma que existen reglas sintácticas y gramaticales en las lenguas naturales de comunicación entre seres humanos, los lenguajes de programación tienen pautas a seguir al momento de codificar.

Por ejemplo, en español una oración con algún error puede sonar extraña o con un significado distorsionado, en un lenguaje de programación un descuido en la sintaxis puede causar que el computador no logre ejecutar las instrucciones dadas.

Un ser humano tiene la capacidad de recuperar significado de una oración extraña, el proceso de compilación de una computadora no, debido a esto es importante que su código esté escrito de forma correcta.

La importancia de definir estándares

A través de los años, desarrolladores con gran experiencia han creado **estándares de programación** para promover la creación de código de calidad, para que aquellas personas que lean el programa logren entender fácilmente la solución dada.

Por ejemplo, existen muchas nomenclaturas establecidas para darle nombre a las variables, las cuales tienen el objetivo de facilitar la lectura y comprensión dentro del programa. Tres de esas nomenclaturas que se utilizarán a lo largo del curso son:

- **camelCase**: la primera letra de una variable irá en **minúscula** y si está compuesto por más palabras, estas irán en **mayúscula**. Por ejemplo: *primerNumero*, *sumaTotal*, *salarioNeto*.
- **PascalCase**: similar a camelCase, pero en este caso la primera letra de todas las palabras se escribe en **mayúscula**. Por ejemplo: *PromedioPonderado*, *ListaEstudiantes*, *ResultadoFinal*.
- **snake_case**: es un estilo de escritura donde los espacios se reemplazan por “_” y las letras van siempre en minúscula. Por ejemplo: *primer_numero*, *suma_total*, *salario_neto*

Estándares para nombre de las variables

Se recomienda que las variables cumplan con el siguiente estándar:

- Tenga un nombre significativo que represente su funcionalidad dentro del programa.
- Utilice una de las nomenclaturas existentes a la hora de definir el nombre de la variable.
- Si desea utilizar números, escribirlos al final del nombre de la variable.

porcentajeAsistencia

cantidadTotalEntregas

notas_estudiantes

Sintaxis para nombre de las variables

Existen **restricciones** a la hora de definir el nombre de las variables que, en caso de omitirlas, Python las detectará como un error en el código. Por ejemplo:

- Se debe iniciar el nombre de la variable con una letra. No se debe iniciar con un símbolo o número, *2estudiante*, *#Mes*, *%adultosMayores* son ejemplos de identificadores **inadecuados**.
- Python al utilizar palabras provenientes del inglés en su sintaxis, no permite el uso de tildes ni virgulillas (como en la letra Ñ) como *opción3*, *primerAño*, *id_Camión*.

A la hora de trabajar con variables en Python recuerde lo siguiente:

- Declarar la variable y asignarle un valor inicial genérico.
- Para variables de tipos *String* se puede usar comillas simples o dobles vacías. `""`, `''`
- Para variables de tipo `int`, se puede usar el `0`
- Para variables de tipo `float`, se puede usar `0.0`
- Para variables de tipo `boolean`, se puede inicializar en `False` (al menos que se requiera lo contrario según contexto).

A la hora de trabajar con variables en Python recuerde lo siguiente:

- Los tres procesos relacionados con variables, **declaración** cuando crea una variable y le da un nombre, **inicialización** cuando le da un valor inicial y **asignación** cuando en su programa asigna un nuevo valor a una variable.
- Utilizar las nomenclaturas y estándares mencionados a la hora de nombrar sus variables para asegurarse que sea fácil comprender la funcionalidad en el código.



universidad
cenfotec_
La U de la informática