PYTHON

- Diego Orozco Fonseca

IRSI | Information Risk & Security Institute

Actividad

01

- Expectativas del curso.
- Cuando me dicen programar, ¿qué pienso que es?.

IRSI Information Risk & Security Institute

02

Fundamentos de programación

IRSI Information Risk & Security Institute

“TODO EL MUNDO DEBERÍA A APRENDER A PROGRAMAR, PORQUE TE ENSEÑA A PENSAR”

—Steve Jobs

¿Qué es programar?

- Crear software usando un lenguaje de programación
- Darle instrucciones a la computadora
- Enseñarle a la computadora a hacer algo

¿Qué es programar?

Programar es muy similar a escribir una receta porque se debe de descomponer paso por paso el proceso de de cocinar un plato.

Los pasos pueden ser en español, inglés o cualquier idioma pero seguirán siendo las mismas.

¿Qué es programar?

Al desglose de un proceso en pasos detallados y ordenados le denominamos **Algoritmo**

Al fichero donde transcribimos estas instrucciones usando un **Lenguaje de Programación** concreto (C++, Python, Java) para que pueda ser ejecutado por una computadora, le llamamos **Programa**.

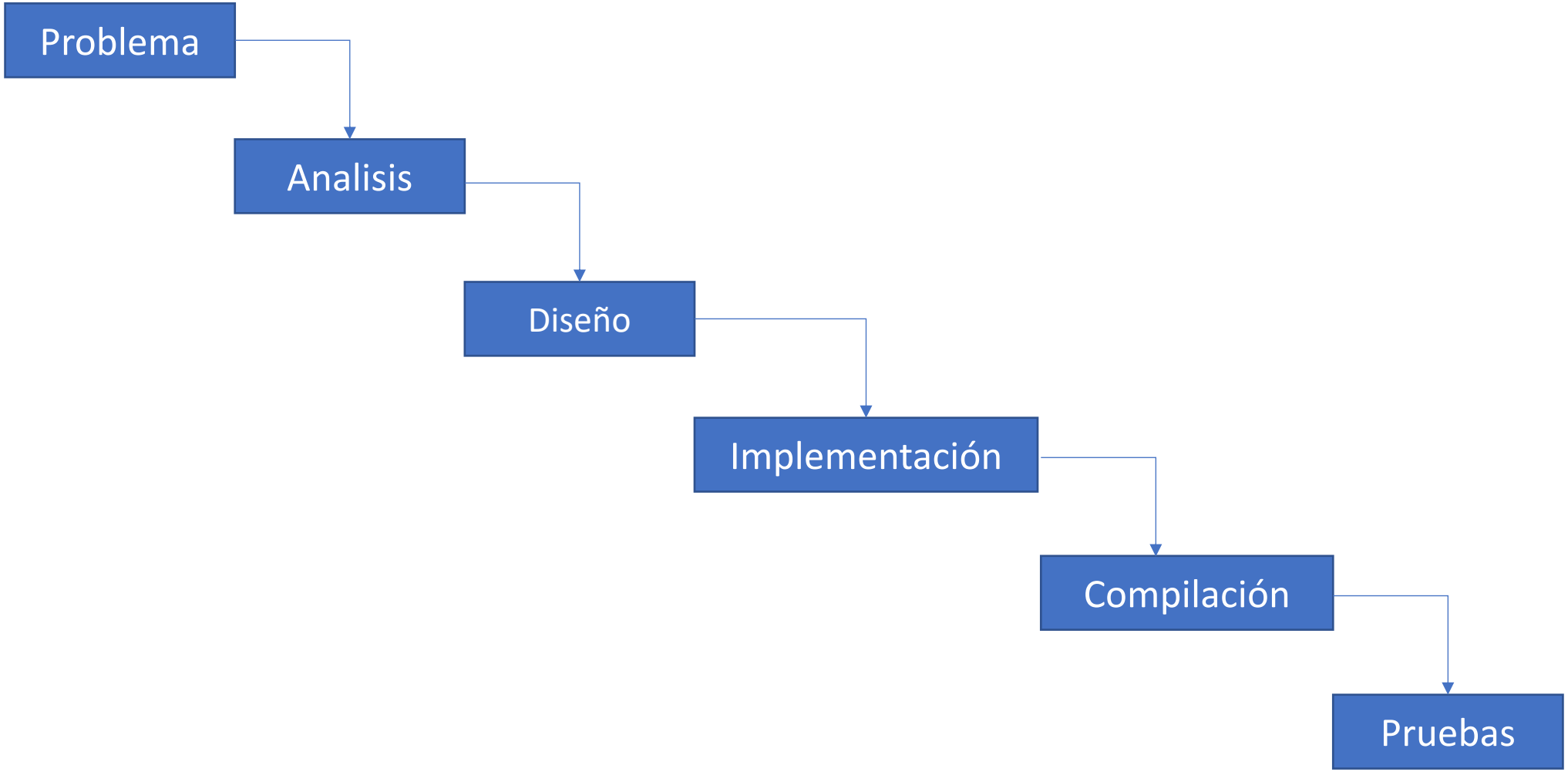
¿Qué es programar?

Estos **Programas** son un **conjunto de sentencias** escritas en un **lenguaje de programación** que le indican a la computadora

- **qué tareas debe realizar**
- **en qué orden**

a través de una serie de **instrucciones** que detallan completamente ese proceso sin **ambigüedad**.

Ciclo de vida de un programa



Especificaciones del problema: Importancia

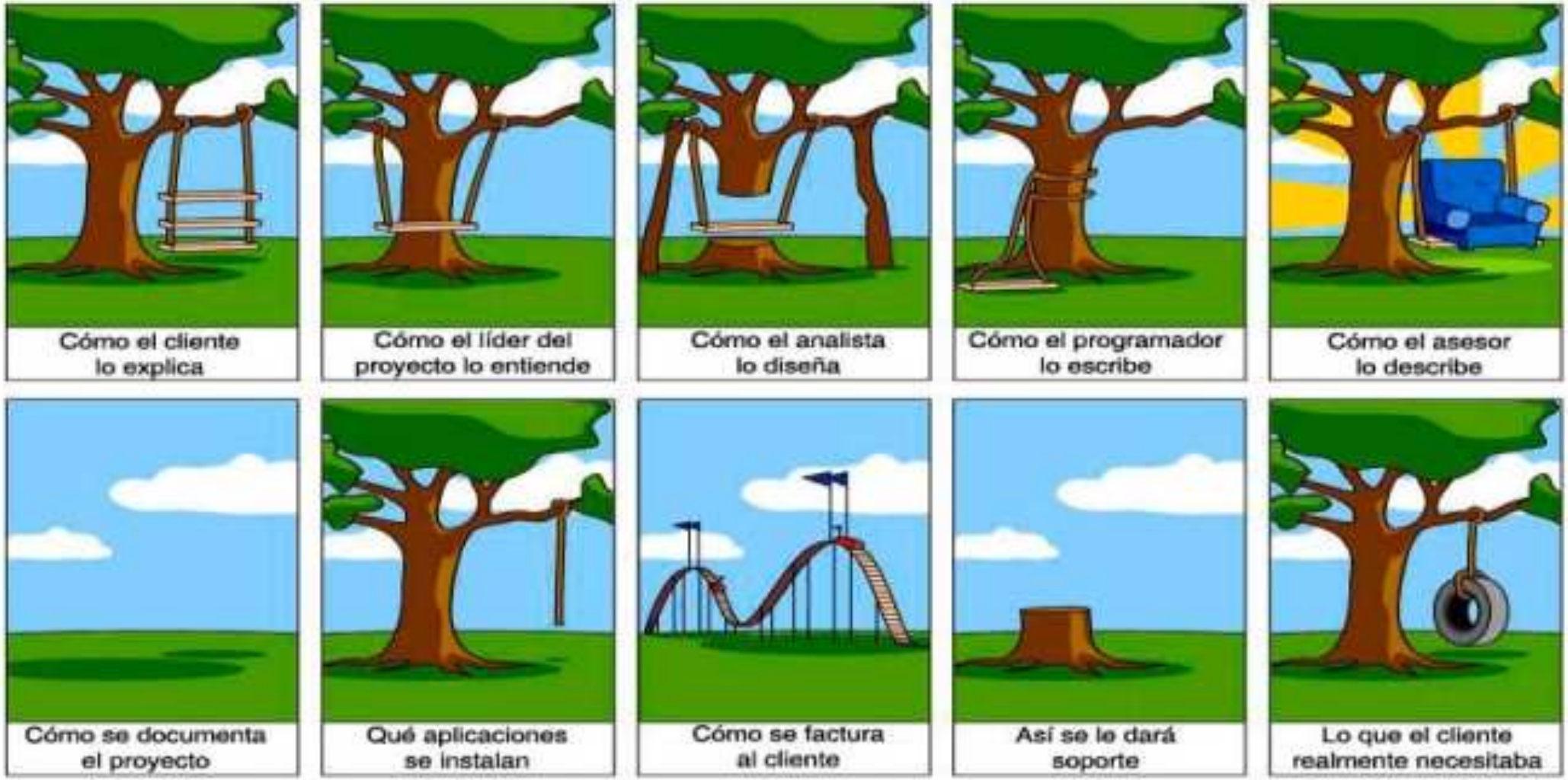
El objetivo principal del proceso de elaboración de un programa es solucionar un problema de la vida real.

La especificación del problema plantea:

- cuál es el problema qué se quiere resolver
- cuáles son las características esperadas de una posible solución

Los proyectos de desarrollo de *software* fallan debido a que se producen programas sin tener aún claro el problema y los requerimientos que tiene el usuario quien necesita del mismo.

Especificaciones del problema: Importancia



Análisis de un problema

IRSI Information Risk & Security Institute

Debe de ser comprendido en su totalidad.

Una buena práctica es descomponer el problema en **sub-problemas** o sub-tareas de **menor** complejidad.

Saber bien qué es lo que tengo que realizar.

Tips

Leer el problema varias veces
Establecer los datos del problema
Precisar el resultado que se desea lograr
Determinar la incógnita del problema
Organizar la información
Trazar una figura o diagrama

Diseño

Se define el algoritmo o secuencia de instrucciones necesarias para resolver un problema.

¿Qué se obtiene de esta etapa?

- características detalladas
- modo de ejecución
- modo de implementación
- definición de dependencias
- escoger y decidir las operaciones a efectuar
- eliminar los datos inútiles.

Implementación

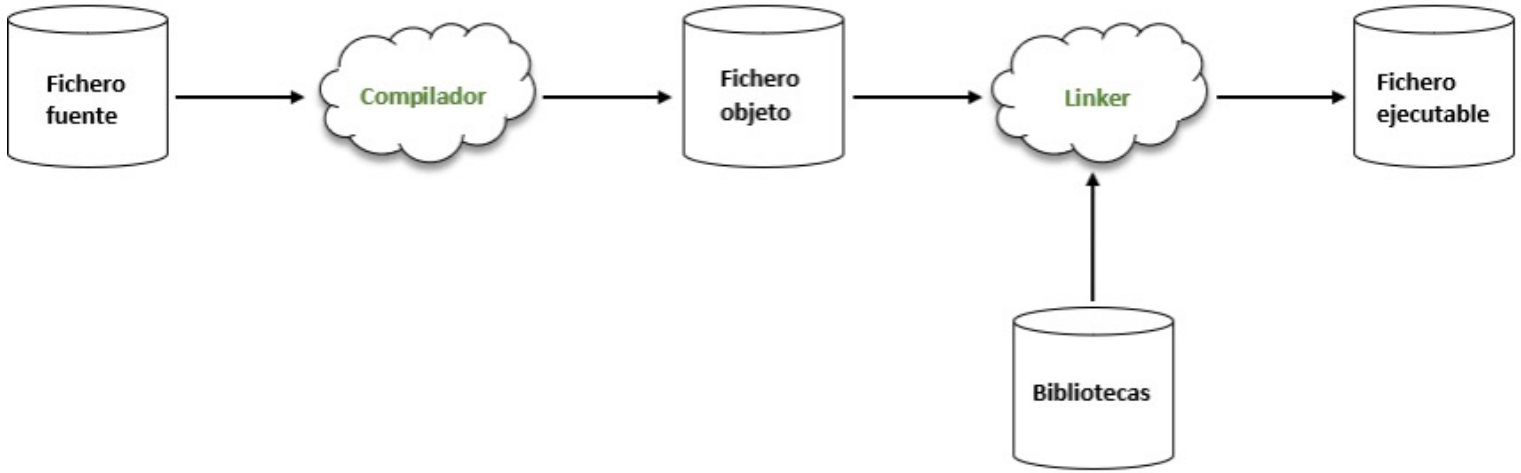
Es el proceso de escritura de las instrucciones o sentencias que va a ejecutar el programa, utilizando un lenguaje de programación.

Es llamado **código fuente**.

Compilación

Es un proceso mediante el cual el compilador produce, a partir del código fuente, un programa ejecutable para la computadora.

Al producto de este proceso normalmente se le llama programa binario o ejecutable.



Pruebas

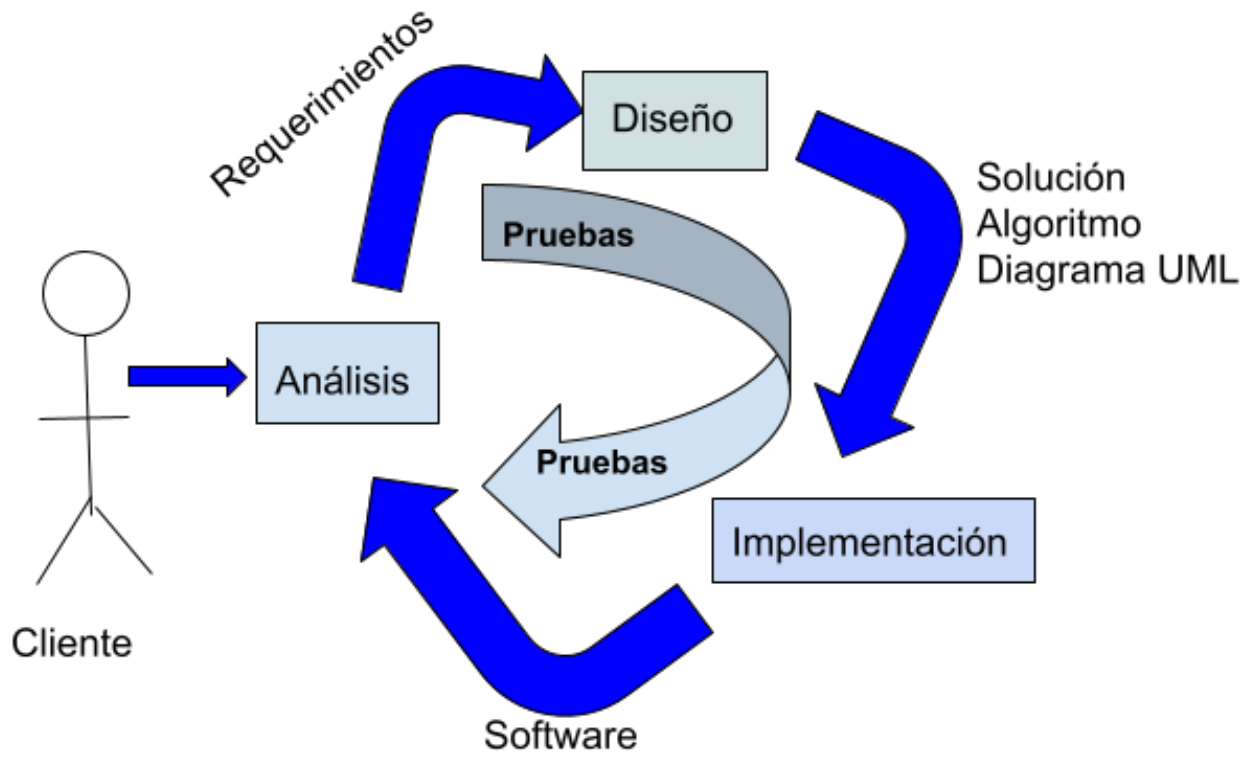
IRSI Information Risk & Security Institute

Es el proceso donde se verifica que el programa funciona correctamente.

Casos de prueba que debe pasar el programa para verificar que resuelve el problema inicial. Es recomendable que dichas pruebas contengan casos sencillos y casos complejos para así determinar la completa funcionalidad del programa.

Es recomendable, realizar los casos de prueba antes de iniciar con la etapa de programación o implementación.

Ciclo de Desarrollo



Algoritmo

Conjunto ordenado y finito de operaciones simples a través del cual podemos hallar la solución a un problema.

Antes de escribir el código de un programa hay que resolver con un algoritmo el problema que se nos plantea.

Algoritmo

Caracteristicas

Tienen inicio y fin

- todo algoritmo comienza en un estado inicial, una serie de datos específicos, y culmina con una solución o salida.

Funcionan en secuencia

- un algoritmo está compuesto por una serie de pasos ordenados.

Las secuencias son concretas

- cada paso es claro y no deja lugar a la ambigüedad.

La cantidad de pasos de un algoritmo es finita.

IRSI Information Risk & Security Institute

Caracteristicas

Se distinguen las siguientes acciones:

Entrada

- información de partida que necesita el algoritmo para arrancar

Proceso

- conjunto de todas las operaciones a realizar

Salida

- resultados obtenidos

Algoritmo

Caracteristicas

1) Compare A y B.
2) Si A es mayor o igual que B, entonces continúe con el paso 3, caso contrario, salte al 8
3) Compare A con C
4) Si A es mayor o igual que C, siga el paso 5, sino salte al 6
5) A es el mayor
6) Sino
7) C es el mayor
8) Si B es mayor que A
9) Compare B con C
10) Si B es mayor o igual que C ejecute el paso 11 sino salte al 12
11) B es el mayor
12) Sino
13) C es el mayor

IRSI Information Risk & Security Institute

Ejemplo

Preparar una taza de té

Entrada: tetera, taza, bolsa de té
Salida: taza de té

Inicio

Tomar la tetera
Llenarla de agua
Encender el fuego
Poner la tetera en el fuego
Esperar a que hierva el agua
Tomar la bolsa de té
Introducirla en la tetera
Esperar 1 minuto
Echar el té en la taza

Fin

IRSI Information Risk & Security Institute

Pseudocódigo

Describen un algoritmo de forma similar a un lenguaje de programación pero sin su rigidez, de forma más parecida al lenguaje natural.

- Son más compactos que los diagramas de flujo
- Más fáciles de usar ua que es muy similar al español
- Son más fáciles de transferir a un lenguaje de programación.

El pseudoćodigo no está regido por ningún estándar.

IRSI Information Risk & Security Institute

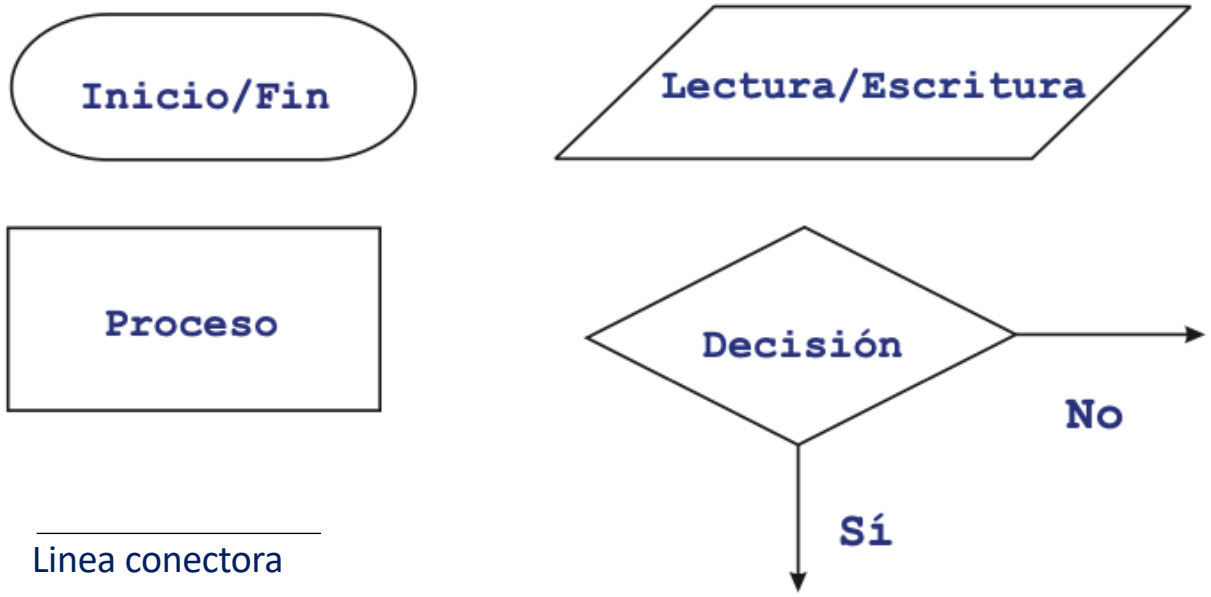
Pseudocódigo

Usaremos las palabras

- **INICIO/FIN** indica donde comienza y termina el algoritmo
- **LEER** para representar las acciones de lectura de datos (el programa recibe datos desde algún modo)
- **ASIGNAR (x ← y+z)**
- **SI … ENTONCES** para representar una condición/validación
- **ESCRIBIR** (el programa escribe información en algún medio)

Representación de algoritmos

Diagrama de flujo



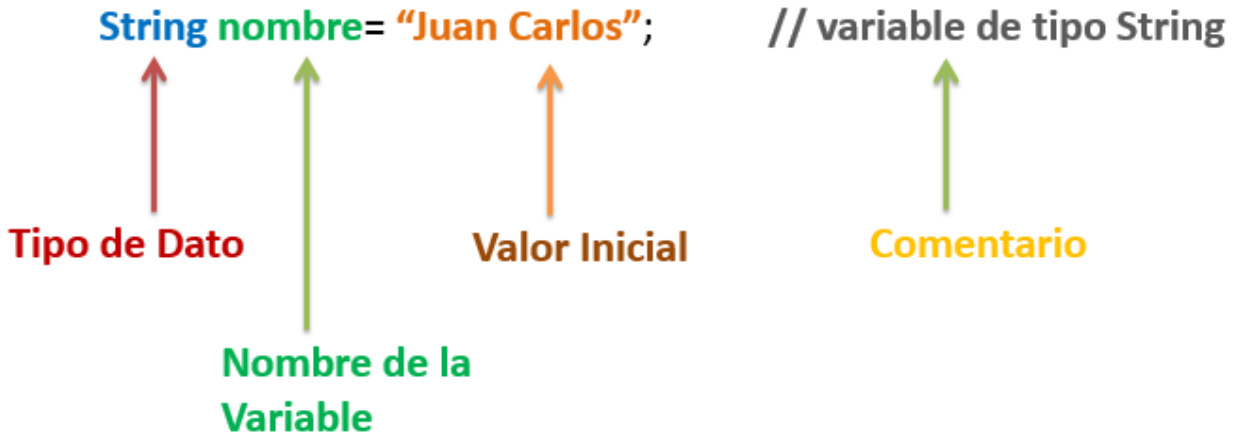
Variables

IRSI | Information Risk & Security Institute

¿Qué son las variables?

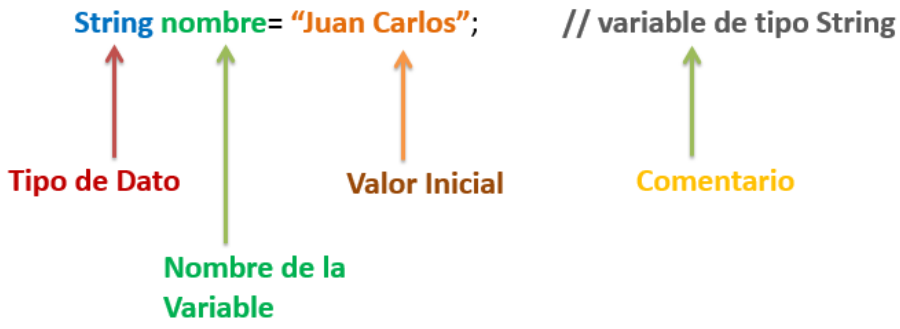
- Son espacios en memoria asignados para almacenar valores.

- Qué debe de contener una variable:
  - Nombre de la variable.
  - Tipo de dato



¿Cómo crear una variable?

- La creación de una variable requiere dos pasos:
  - declarar la variable
    - definición del tipo y del nombre de la variable.
  - inicializar la variable
    - es la asignación del valor inicial.



IRSI Information Risk & Security Institute

Declarar una variable

- Definición del tipo y del nombre de la variable.
  - age = 10

- Es posible declarar varias variables a la vez. Solo debe de ir separado por ,
- Pueden ser de diferente tipo
  - word, number, flag = "Diego", 7, False

IRSI Information Risk & Security Institute

Declarar una variable

Nombres

- Para los nombres de variables se puede usar los siguientes caracteres: ‘a’, ‘b’, …, ‘z’, ‘A’, ‘B’, …, ‘Z’, ‘0’, ‘1’, …, ‘9’, ‘_’.
- Para la primera letra del nombre de una variable no se puede utilizar un dígito.
- Utilice palabras significativas. Nombres deben dar una idea del uso que se pretende dar a cada una de las variables

Declarar una variable (cont)

Nombres

- No hay un estándar en Python para definir los nombres de las variables.
- Se recomienda usar:
  - camelCase
    - `myAge`
  - snake_case
    - `my_age`
- Sin importar cual tipo usan, solo se debe de usar uno alrededor de todo el código.

Otros tipos de variables

- Las variables pueden ser de tipo primitivo
  o
  pueden ser “referencias” a instancias.

- La inicialización de las instancias es con el nombre del tipo complejo (clase) y paréntesis al final.
  - Con esto se solicita la construcción de una instancia de la clase.

Por ejemplo:

```
my_person = Person()
```

Terminal

IRSI | Information Risk & Security Institute

IRSI Information Risk & Security Institute

Repaso