La evolución del escritorio

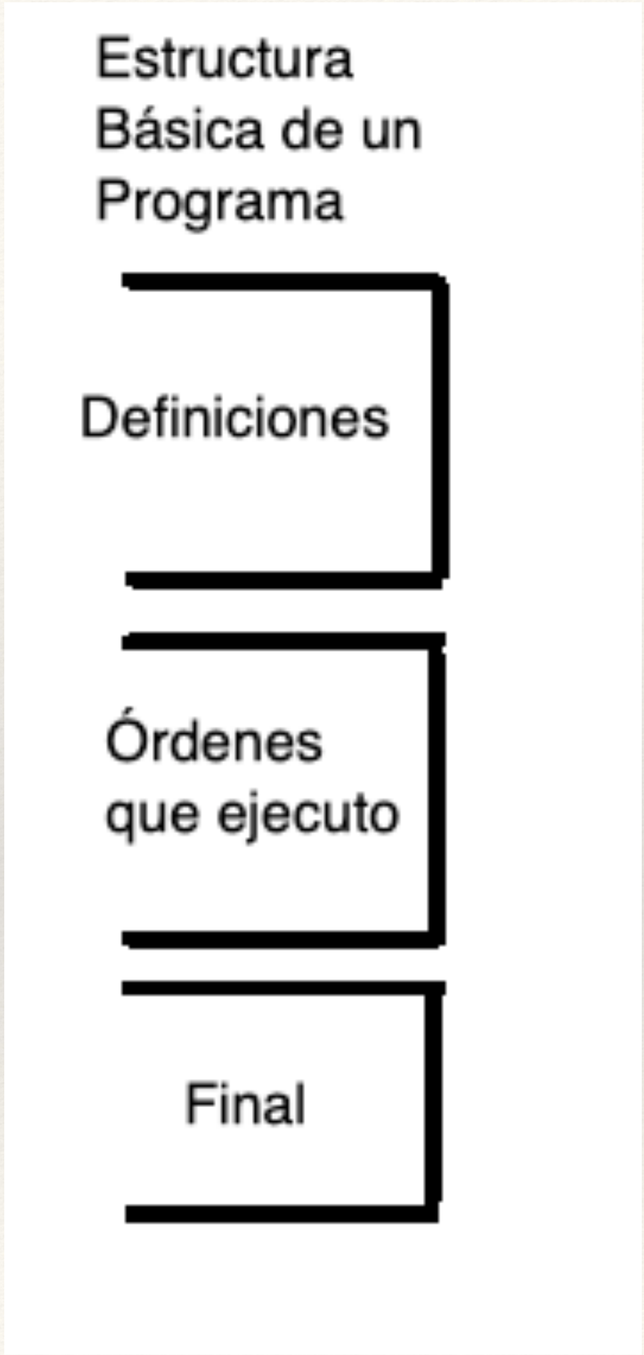
www.youtube.com/watch?v=uGI00HV7Cfw

Otros Lenguajes

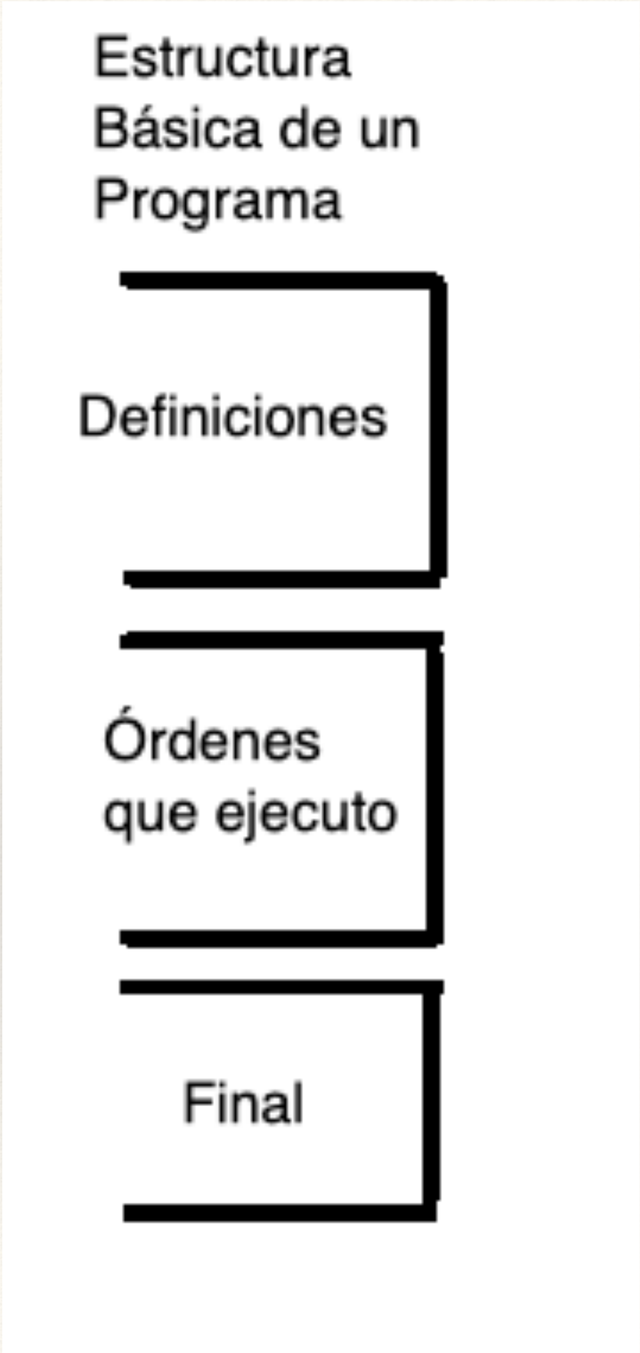
- En informática, todo cambia muy rápido
- En general son ciclos de 5 años, tanto en Hardware como en Software (quizás estemos en el comienzo de un ciclo).
- Hay que ser versátil y adaptable.

¿Cómo hacemos si tenemos que aprender un lenguaje diferente? O al menos cómo hacemos para aprenderlo rápido.

Veamos un par de reglas que nos pueden ayudar, reconociendo estructuras comunes a todos los lenguajes de computación.



Tengo 3 estructuras principales



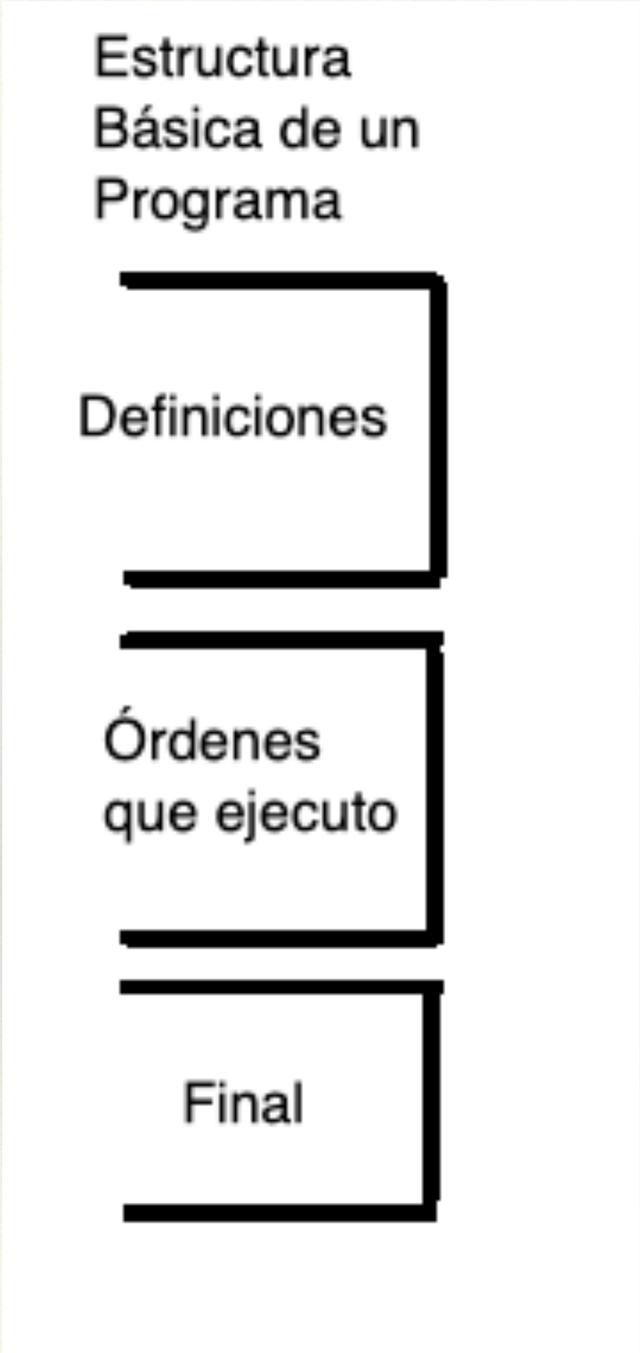
Tengo 3 estructuras principales

Definiciones

De que clase son cada una de las variables.

Hay enteros, reales o arreglos de varias dimensiones. Tengo que averiguar como se escriben las definiciones (por ahí no lo necesito).

Puede que tengo que indicar si cargo librerías (por ejemplo: Math, Numpy).



Asignaciones

Tengo que entender como se escriben las sentencias.

Donde empiezan, como se indica su final y es decir la gramática que utiliza el lenguaje. Por ejemplo, cuál es el símbolo que se usa para asignar:

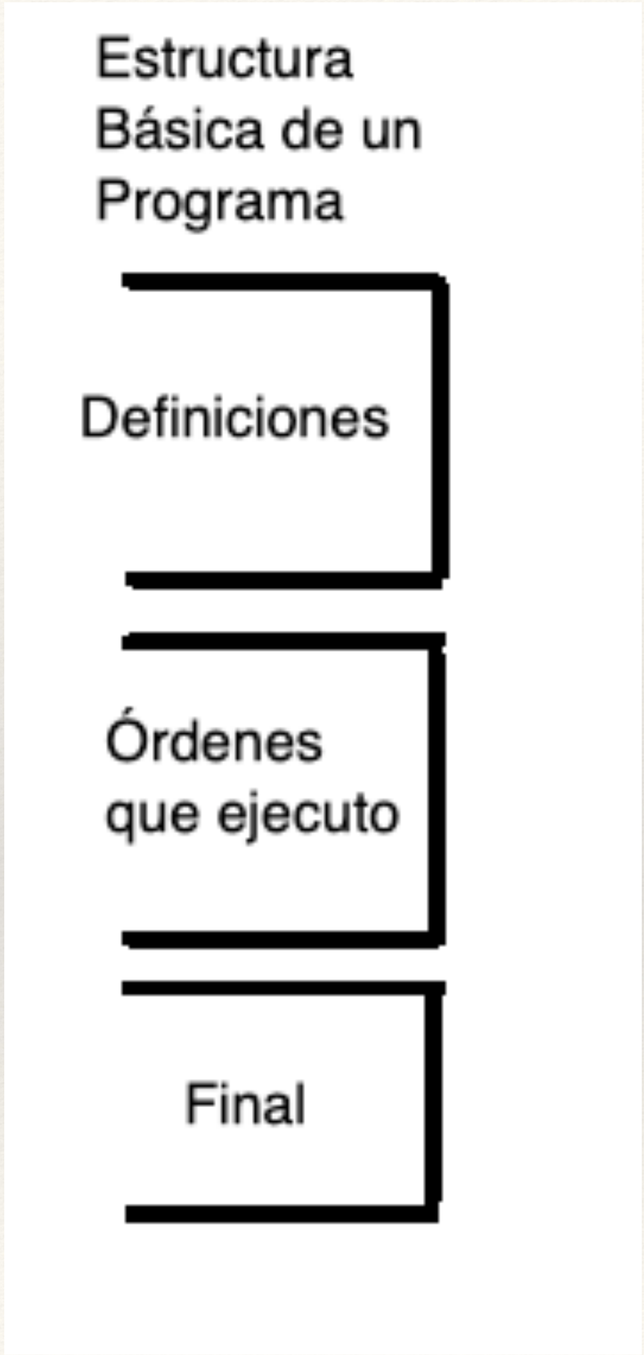
A=B+2

$A=$B+2; <— indico el final con ;

A:=B+2;

A <- B+2;

Set A B+2



Comentarios

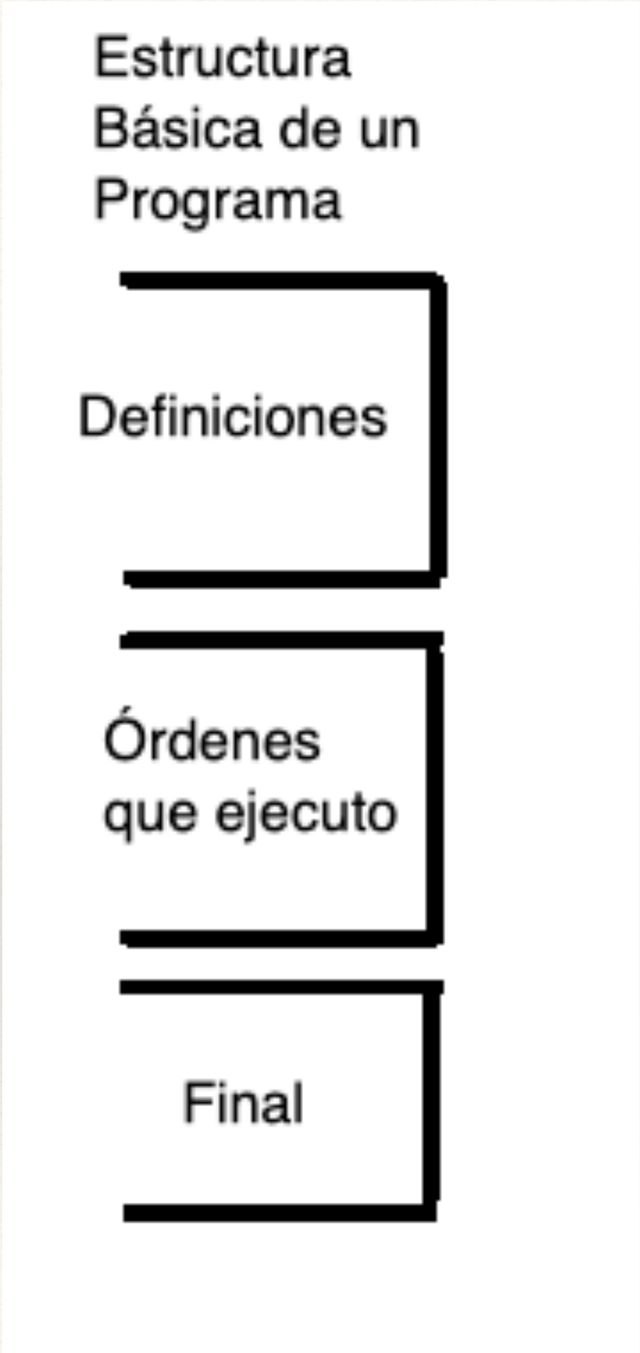
También tengo que ver como se escriben los comentarios, desde un símbolo que indica esta situación (c, #, !, %) o sentencias que abren y cierran comentarios:

<!— comentario (varios renglones) —>

/* Comentario comentario (varios renglones)*/

O un lenguaje tipo Markup

Que es una variante de Latex



Final

Son órdenes que indican que hacer cuando un programa termina, muy pocos en Fortran, por ejemplo: Close()

Ejecución

- El lenguaje podría ser compilado o interpretado.
- Podría existir un SDK (Software Development Kit).
- Hay sistemas que dividen el programa en celdas y cada celdas puede ejecutarse en forma independiente (ejemplo: python en una notebook).
- Hay programas para encontrar Bugs (bichos/errores) - Debugging.

Detalles de la asignaciones

Puede que las variables sean diferentes ya que mayúscula y minúsculas son diferentes: Mag, mag, MAg y MAG son variables diferentes.

Modos resumidos:

i=i+1. (Fortran)

i+=1. (Python)

i++ o (++i). (C, C++, Perl)

Ojo no es lo mismo a=b+(i++) que a=b+(++i)

a=a*10

a*=10

Ordenes y asignaciones

Variables Especiales:

Listas y conjuntos (mutables e inmutables): [1,2,3,4,59,49]

Diccionarios: variable(key) da un valor guardado

Arreglos - > Se usa [] en vez de (), puede que se usen al revés la indicación de filas y columnas

Objetos

Estructuras de Control - IF

IF()
Estilo Perl o C

```
if(algo) {  Pasa esto
}
else { Pasa lo otro
}
```

Estilo Python

```
if (algo):
    Pasa esto
else:
    Pasa lo otro
```

Se usan los símbolos:
<,>,<=,>=,==,!=
el "!" Indica negación,
El "==" indica igualdad

&& o & indican el operador AND
|| o | indican el OR

! Es el NOT

Estructuras de Control - loops

Estilo Perl o C

```
For (i=1; i<n; i++){
}
```

También existen

```
Do while() {
}
```

O la forma

```
While() do {
}
```

Funciones y subrutinas

Existen en todos los lenguajes, a veces solo se permiten funciones (pero ya no lo son en términos del Cálculo Matemático, que devuelven varios resultados que pueden no tener conexión entre ellos de ningún tipo)

Si el lenguaje es interpretado van a estar localizadas antes de ser llamadas.

Mientras que un lenguaje compilado como el Fortran da lo mismo su si están antes o después, y se las suele agregar como archivos aparte.

Repeat, Break and Next

```
repeat {
código
        IF (a<b) next
código
          If(c>x) break
código
}
```

DO, Cycle, Exit para Fortran 90/95

Try & Except

```
Try {
        Código 1
}

Except Error{
        Código 2
}
```

Pruebo si algo funciona o funcionó.

Si esto pasó corre el Código 1

Y si no lo hizo corre el Código 2

Sirve para tomar decisiones sobre errores que se pueden encontrar.