

Programación

básico

The logo consists of a solid blue rectangle with the word "CODEAR" in white, uppercase, sans-serif font centered within it. This rectangle is positioned inside a larger white rectangle that has a thin blue border. The blue rectangle is offset slightly from the top-left and bottom-left corners of the white rectangle, creating a layered effect.

CODEAR

qué es programar?

Programar es darle una orden a la computadora y que esta simplemente lo ejecute.

COMO PROGRAMAR

Hablandole a la computadora en un idioma que esta comprenda.

1

Software + Hardware

Hardware



software



¿Conoces algún otro software?

EJEMPLOS DE SOFTWARE



A solid blue square containing the white number '2' in a large, sans-serif font.

¿Cómo se hace un Software?

Paso a Paso



3

¿Qué es lógica de programación?

**Es el razonamiento lógico que se debe hacer
para resolver un problema.**

4

¿Qué es algoritmo?

Es escribir en una secuencia, la lógica.

EXEMPLO

Piense en la lógica de cómo
bañarse.

EXEMPLO BAÑARSE (LÓGICA)


Básicamente es entrar en la ducha, abrir el agua, regular la temperatura, utilizar el jabón, enjuagar y listo.

EXEMPLO BAÑARSE (ALGORITMO)

- 1) Entrar al baño
- 2) Sacarse la ropa
- 3) Abrir la ducha
- 4) Mojar el cuerpo
- 5) Pasar jabón por cuerpo
- 6) Enjuagar
- 7) Secar cuerpo
- 8) Vestirse con ropa limpia
- 9) Salir del baño

EXEMPLO BAÑARSE (ALGORITMO)

- 1) El baño está disponible?
- 2) Si, prosiga | si no, volver otra hora. ❌
- 3) Entrar al baño
- 4) Sacarse la ropa
- 5) Poner la ropa sucia a lavar
- 6) Abrir la ducha
- 7) Regular temperatura del agua
- 8) Ajustar la temperatura del agua
- 9) Si la temperatura es buena, comenzar el baño. Si no, ajustar la temperatura de nuevo
- 10) Mojar el cuerpo
- 11) Pasar el jabón por cuerpo
- 12) Enjuagar
- 13) Agarrar la toalla
- 14) Secar cuerpo
- 15) Vestirse con ropa limpia
- 16) Salir del baño

- 
- 1) Entrar en el baño
 - 2) Sacar la ropa
 - 3) Abrir la ducha
 - 4) Mojar el cuerpo
 - 5) Pasar el jabón en el cuerpo
 - 6) enjuagar
 - 7) Secar el cuerpo
 - 8) Vestir ropa limpia
 - 9) Salir del baño

EXEMPLO SUMA 2 EDADES

- 1) leer la primer edad
- 2) leer la segunda edad
- 3) hacer la suma de las edades

5

¿Qué es pseudocódigo?

Pseudocódigo es una forma genérica de escribir un algoritmo, utilizando un lenguaje simple sin necesidad de conocer la sintaxis de ningún lenguaje de programación.

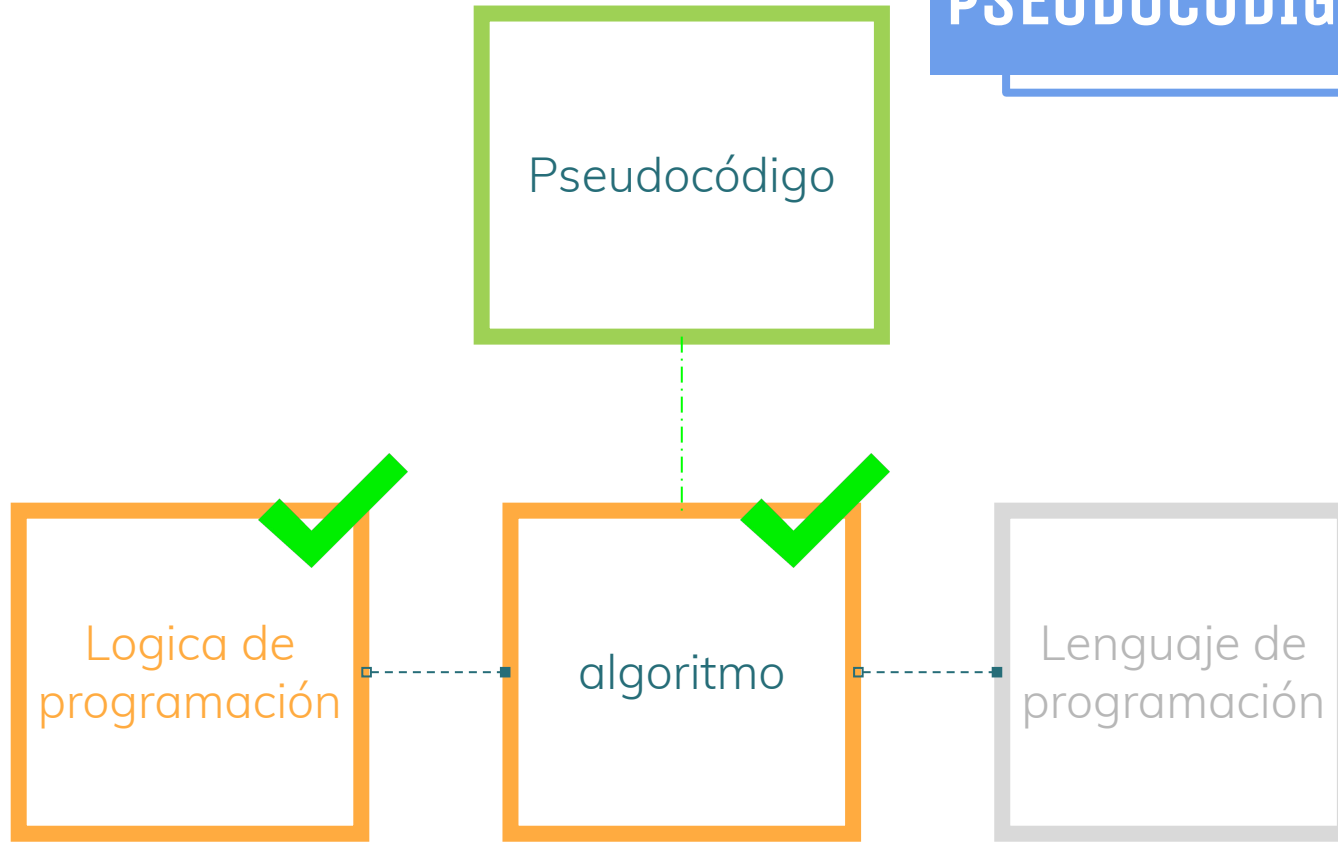
PSEUDOCÓDIGO

Pseudocódigo

✓
Logica de
programación

✓
algoritmo

Lenguaje de
programación



No es necesario escribir el algoritmo y luego pasar a un pseudocódigo o lenguaje de programación.

PSEUDOCÓDIGO

Logica de
programación



algoritmo



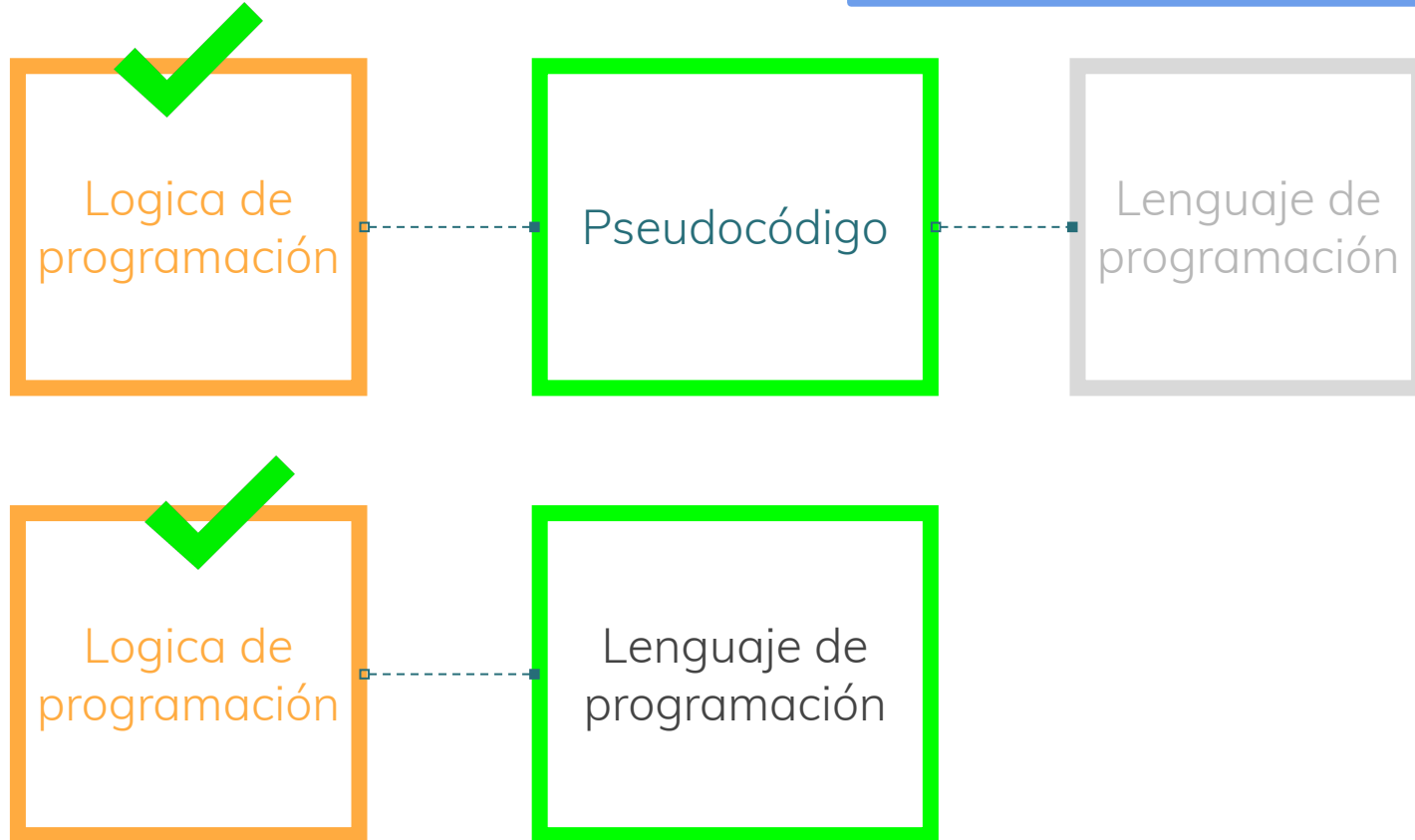
Pseudocódigo

Lenguaje de
programación



Podemos pensar en la lógica y ya escribir en un pseudo-código o incluso en un lenguaje de programación.

PSEUDOCÓDIGO



EJ: SUMA 2 EDADES

- 1) leer el primer edade
- 2) leer el segundo edade
- 3) Crear un espacio en la memoria de una computadora con nombre de resultado
- 4) Ese espacio llamado resultado tiene que recibir
→ $\text{Edade 1} + \text{Edade 2}$
- 5) Imprimir en una pantalla el resultado

Programa_suma_dos_edades

EJ: SUMA 2 EDADES

Inicio

variables **edade1, idade2, resultado** : entero

*declarar variables
y constantes*

leer **edade1**

leer **edade2**

resultado \leftarrow **edade1** + **edade2**

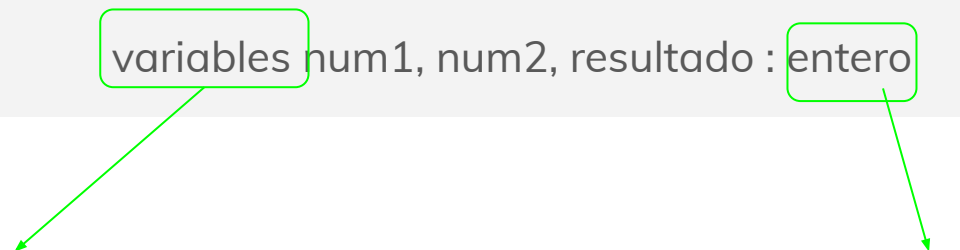
imprima **resultado**

Fin

EJ: SUMA 2 EDADES

Programa suma_dos_edades

variables num1, num2, resultado : entero



The diagram shows the variable declarations 'variables num1, num2, resultado : entero' from the code above. A green box highlights 'variables' with an arrow pointing to the 'VARIABLES' explanation box. Another green box highlights ': entero' with an arrow pointing to the 'ENTERO' explanation box.

VARIABLES - porque su contenido puede variar!
Por ejemplo:
edade1 puede ser 1, 2, 50, 100.

ENTERO - es el tipo !
Por ejemplo:
el edade1 hay que ser entero.
Hay otros tipos como decimal, caracteres(que son letras)

6

Variables x Constantes

Son nombres que creamos para representar un valor, para que así sea posible utilizar este mismo valor varias veces en esa lista de tareas, sin necesidad de reescribir todas las veces que sea necesario.

VARIABLES Y CONSTANTES

Programa_suma_plata

Inicio

constante **cuenta_ahorro**: real

variables **plata_atual, total**: real

cuenta_ahorro = \$1000.00

leer **plata_atual**

total \leftarrow **cuenta_ahorro** + **plata_atual**

imprima **total**

Fin

7

Tipos de Datos

Ejemplo: nombre, edad, sueldo ,sexo

Para una computadora esas informaciones son Datos.

Y los datos pueden tomar varios formatos.

Piense de la siguiente manera, el nombre es diferente de edad que, a su vez, es diferente de salario.

TIPO 1 : TEXTO

Los datos del tipo de texto representan una secuencia de uno o más caracteres. Normalmente se colocan entre comillas dobles.

Un ejemplo de dato del tipo texto:

→ **Nombre** ---> Juan Pablo

→ **Dirección** ---> Calle San Martín

Obs: espacios también cuentan como caracteres. O sea tenemos 11 caracteres en "Juan Pablo".

TIPO 2: ENTERO

Se representan por valores numéricos, tanto negativos, cuando positivos (sin decimales).

Un ejemplo de dato del tipo entero:

→ **Edad ---> 18**

→ **Departamento ---> 5**

TIPO 3 : REAL

Los datos del tipo real son valores numéricos, tanto positivos, como negativos que utilizan decimales,

Como ejemplo, podemos citar:

→ **Sueldo** ---> 13.434,52

→ **Precio** ---> 135,70

TIPO 1 : LÓGICO

Por último, el tipo lógico. Este tipo de datos normalmente está representado por alternativas:

SÍ o NO, VERDADERO o FALSO.

Los datos de tipo lógico también se pueden llamar Booleano.

Un ejemplo de dato lógico es:

→ **Sexo** ---> masculino o femenino

→ **Donante** ---> si o no

TIPO DE DATOS

Siendo así, quedó:

Nombre: TEXTO

Edad: ENTERO

Sueldo: REAL

Sexo: BOOLEANO

8

Estructura de selección y repetición

Formas y estructuras organizadas de hacer que una computadora pueda tomar decisiones o ejecutar varias veces una instrucción hasta que un límite preestablecido sea alcanzado, o una condición sea satisfecha, o el usuario intervenga.

Una computadora, diferente de los humanos todavía no puede pensar por su cuenta, siendo así, él necesita reglas bien definidas para su correcto funcionamiento.

Esto se logra a través de mucho entrenamiento y práctica con lógica de programación.

EJ: BAÑARSE

- 1) El baño está disponible?
- 2) Si esta prosiga | si no esta volver otra hora.
- 3) Entrar en el baño
- 4) Sacar la ropa
- 5) Poner la ropa sucia para lavar
- 6) Abrir la ducha
- 7) Regular temperatura del agua
- 8) Ajustar la temperatura del agua
- 9) **Si la temperatura es buena, comenzar el baño. Si no, ajustar la temperatura de nuevo**
- 10) Mojar el cuerpo
- 11) Pasar el jabón en el cuerpo
- 12) enjuagar
- 13) Tomar la toalla
- 14) Secar el cuerpo
- 15) Vestir ropa limpia
- 16) Salir del baño



Programa suma_dos_edades

Inicio

variables `edade1`, `edade2`, `resultado` : entero

leer **`edade1`**

leer **`edade2`**

`resultado` \leftarrow `edade1` + `edade2`

imprima `resultado`

Fin

ESTRUCTURA SELECCIÓN Y REPETICIÓN

comprobar si la
edad 1 y edad 2
son correctas

O sea, no son edades
negativas.

ESTRUCTURA SELECCIÓN Y REPETICIÓN

- 1) leer **edade1**
- 2) comprobar si la **edad1** es correcta. Si es correcta pasar para paso 3.
Si no, preguntar de nuevo en paso 1.
- 3) leer **edade2**
- 4) comprobar si la **edad2** es correcta. Si es correcta pasar para paso 5.
Si no, preguntar de nuevo en paso 4.
- 5) Hacer la suma

Si llegamos en el paso 5,
sabemos que todas edades
fueran verificadas, validadas.

Programa_suma_dos_edades

Inicio

variables **edade1**, **edade2**, **resultado** : entero

leer **edade1**

leer **edade2**

Si (**edade1** > 0) y (**edade2** > 0) **entonces**

resultado \leftarrow **edade1** + **edade2**

imprima **resultado**

Si no

Imprima "Por fa, ingrese edades correctas"

Fin_Si

Fin

IF

declarar variables
y constantes

Hay que validar si no
son edades
negativas.

Algoritmo_compra_manzana

Inicio

variables **plata_que_tengo**, **resto** : real

constante **manzana**: real

leer **plata_inicial**

manzana = 10.00

Mientras **plata_que_tengo** > **manzana** hacer

plata_que_tengo \leftarrow **plata_que_tengo** - **manzana**

imprima **plata_que_tengo**

Fin_Mientras

Fin

WHILE

*declarar variables y
constantes*



WHILE

Mientras plata_que_tengo > manzana hacer

 plata_que_tengo ← plata_que_tengo - manzana

 imprima plata_que_tengo

Fin_Mientras

Mientras plata_que_tengo > manzana hacer

 resto ← plata_que_tengo - manzana

 Plata_que_tengo ← resto

 imprima plata_que_tengo

Fin_Mientras

Ponemos 1 variable a más, que sería el resto.

Mientras plata_que_tengo > manzana hacer

plata_que_tengo \leftarrow plata_que_tengo - manzana

imprima plata_que_tengo

Fin_Mientras

WHILE

Ejemplo:

Tengo 25 pesos

Manzana = 10 pesos

Mientras 25 > 10 hacer

plata_que_tengo \leftarrow 25 - 10

imprima 15

Fin_Mientras

Mientras 25 > 10 hacer

plata_que_tengo \leftarrow 15 - 10

imprima 5

Fin_Mientras

Mientras 5 > 10 hacer



Mientras plata_que_tengo > manzana hacer

 resto ← plata_que_tengo - manzana

 Plata_que_tengo ← resto

 imprima plata_que_tengo

Fin_Mientras

WHILE

Ejemplo:

Tengo 18 pesos

Manzana = 10 pesos

Mientras 28 > 10 hacer

 resto ← 28 - 10

 Plata_que_tengo ← 18

 imprima 18

Fin_Mientras

Mientras 18 > 10 hacer

 resto ← 18 - 10

 Plata_que_tengo ← 8

 imprima 8

Fin_Mientras

Mientras 8 > 10 hacer



Programa compra_manzana

Inicio

variables **plata_que_tengo**, **ctd**: real

constante **manzana**: real

leer **plata_inicial**

manzana = 10.00

ctd+= o ctd++

Para (**plata_que_tengo** > **manzana**, **ctd** = 0, **ctd++**) hacer

plata_que_tengo ← **plata_que_tengo** - **manzana**

Imprimir **plata_que_tengo**

Fin_Para

imprima **ctd**

Fin

FOR

A cada vez que el lazo de repetición sucede a la variable ctd suma 1.

Ejemplos:

ctd+=2

ctd--



Para (plata_que_tengo > manzana, ctd = 0, ctd++) hacer

plata_que_tengo \leftarrow plata_que_tengo - manzana

imprima plata_que_tengo

Fin_Para

imprima ctd

FOR

Ejemplo:

Tengo 28 pesos

Manzana = 10 pesos

Para (28 > 10, ctd = 0, ctd++) hacer

plata_que_tengo \leftarrow 28 - 10

imprima 18

Fin_Para

imprima ctd 1

Para (18 > 10, ctd = 0, ctd++) hacer

plata_que_tengo \leftarrow 18 - 10

imprima 8

Fin_Para

imprima ctd 2

Para (8 > 10) hacer 

9

En dónde puedo escribir?

EN DÓNDE PUEDO ESCRIBIR?

- ❏ Block de notas
- ❏ Editores de textos
- ❏ IDE (ambiente integrado de desarrollo)

10

Lenguajes y Frameworks

LENGUAJES

Algunas

- ❏ Java
- ❏ C
- ❏ C++
- ❏ Python
- ❏ Ruby
- ❏ Go (Golang)
- ❏ PHP

Frameworks

Un conjunto de componentes que ayuda a desarrollar sitios de forma más rápida y fácil.

Cada lenguaje tiene un o más Frameworks.

LENGUAJES

❏ Java → Spring, Wizard

❏ Python → Django, Flask

❏ Ruby → Rails, Sinatra

❏ PHP → CakePHP, Laravel

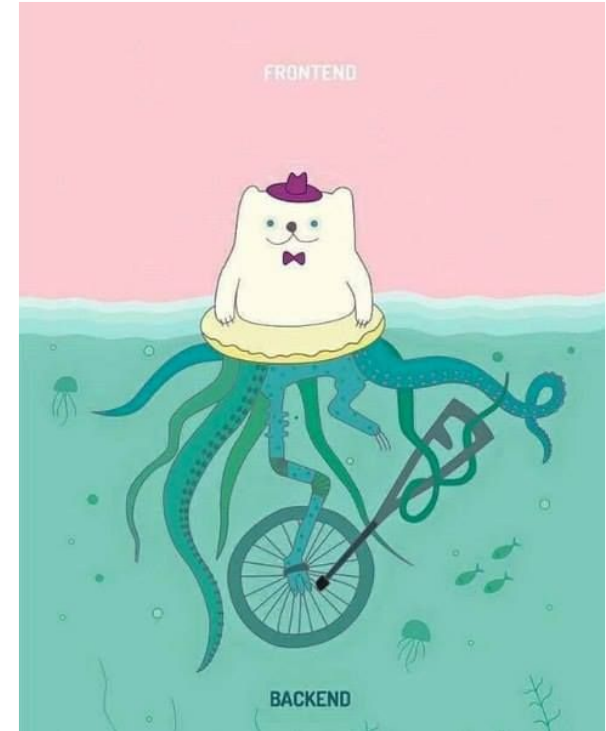
11

Backend x Frontend

El programador frontend es aquel que cuida de toda la parte de interacción con el usuario

ya el programador backend es el profesional que desarrolla el sistema que va a ser utilizado para gestión de los datos, sistema que tendrá interactividad con el usuario y utilizará la interfaz que fue desarrollado por el programador frontend.

BACKEND X FRONTEND



GRACIAS