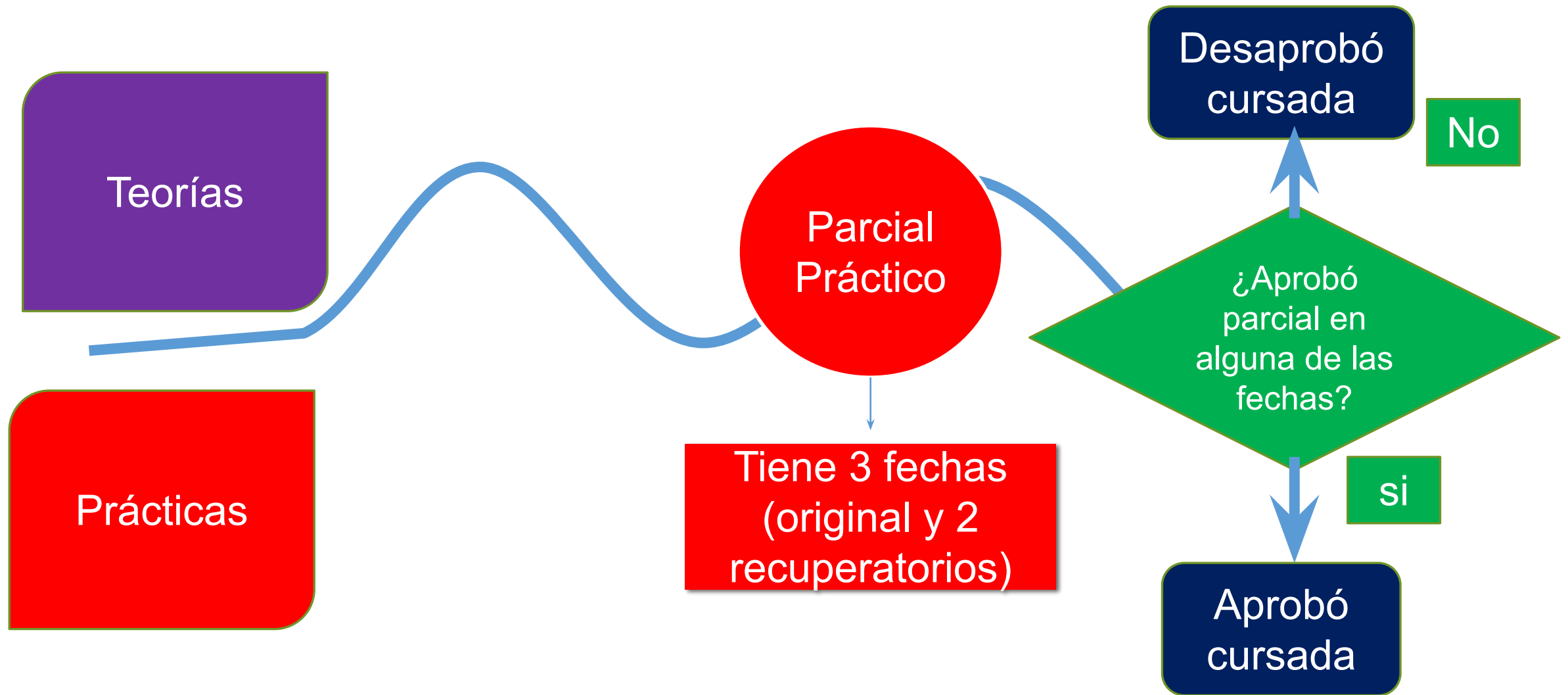


# Algoritmos y Programación I

# AyPI - Para aprobar la cursada



# AyPI - Aprobación

## APROBACIÓN DE LA MATERIA

```
graph TD; A[APROBACIÓN DE LA MATERIA] --> B[EXAMEN FINAL]; A --> C[RÉGIMEN DE PROMOCIÓN];
```

### ✓ EXAMEN FINAL

- Se toma todos los meses y ustedes deben anotarse según el calendario académico.
- Información de la cátedra en la cartelera virtual

### ✓ RÉGIMEN DE PROMOCIÓN

- Aprobar el parcial en primera o segunda fecha
- 1 evaluación de promoción con un recuperatorio

# Algoritmos y Programación I



Lograr que el alumno cuando termine el curso, posea conocimientos, métodos y herramientas para resolver distintos problemas con la computadora logrando:

- Analizar problemas, poniendo énfasis en la **modelización**, **abstracción** y en la **modularización** de los mismos.
- Obtener una expresión sintética, precisa y **documentada** de los problemas y su solución.
- Analizar y expresar correctamente algoritmos, orientando los mismos a la **resolución de las partes** (módulos) en que se descomponen los problemas.
- Introducir las nociones de **estructuras de datos**, **tipos de datos** y **abstracción de datos**.

# Algoritmos y Programación I



Lograr que el alumno cuando termine el curso, posea conocimientos, métodos y herramientas para resolver distintos problemas con la computadora logrando:

Combinar los elementos mencionados anteriormente a fin de que el alumno complete el ciclo del problema a su solución con computadora, con una metodología ordenada y documentada en la que se traten simultáneamente las estructuras de control y los datos y en el que se introduzcan **los conceptos de corrección y de eficiencia.**

# AyPI – Temas de la clase de hoy



- Análisis de problemas
- Definiciones Fundamentales
- Modelos + Datos = programa
- Tipos de datos
- Operaciones de lectura escritura
- Estructuras de control IF-THEN

# AyPI – Definiciones



## Informática

Es la **ciencia** que estudia el análisis y **resolución de problemas** utilizando **computadoras**.

# AyPI – Definiciones

Es la **ciencia** que estudia el análisis y **resolución de problemas** utilizando **computadoras**.



## Ciencia

Se relaciona con una metodología fundamentada y racional para el estudio y resolución de los problemas.

En este sentido la Informática se vincula especialmente con la Matemática y la Ingeniería



## Resolución

Se puede utilizar las herramientas informáticas en aplicaciones de áreas muy diferentes tales como biología, comercio, control industrial, administración, robótica, educación, arquitectura, etc.



## Computadora

Máquina digital y sincrónica, con cierta capacidad de cálculo numérico y lógico controlado por un programa almacenado y con probabilidad de comunicación con el mundo exterior. Ayuda al hombre a realizar tareas repetitivas en menor tiempo y con mayor exactitud. No razona ni crea soluciones, sino que ejecuta una serie de órdenes que le proporciona el ser humano



# AyPI – Definiciones



## Informática - Objetivo

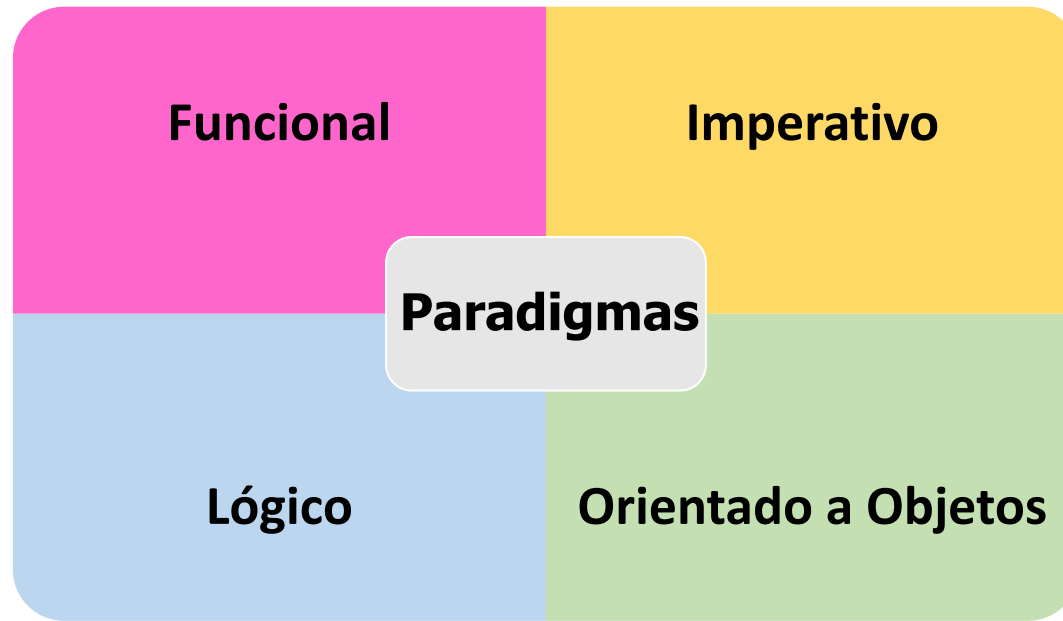
Resolver problemas del mundo real utilizando una computadora (utilizando un software)



## Informática - Problema

- Cuestión, asunto o proposición dudosa, que se trata de aclarar o resolver; requiere una solución.
- Enunciado encaminado a averiguar el modo de obtener un resultado cuando se conocen ciertos datos.

# AyPI – Paradigmas de programación



**Imperativo -  
procedural**

En general, los lenguajes de programación pueden ser clasificados a partir del modelo que siguen para DEFINIR y OPERAR información. Este aspecto permite jerarquizarlos según el paradigma que siguen.

# AyPI – Etapas en la resolución de problemas



## 1) Análisis del Problema:

¿Qué necesita el usuario? (¿y qué dice que necesita?)

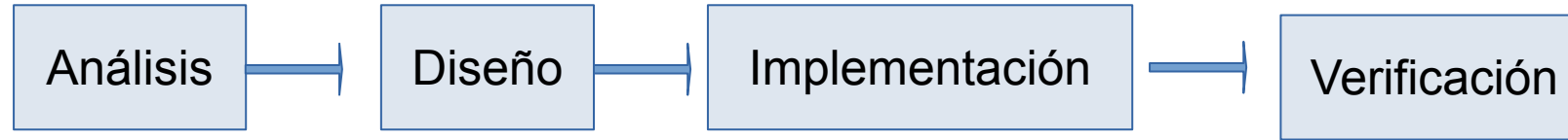
¿Con qué datos vamos a trabajar?

## 2) Diseño de Solución:

¿Qué partes que componen la solución del problema?

¿Cómo se relacionan estas partes?

# AyPI – Etapas en la resolución de problemas



## 3) Implementación:

Traducir la solución anterior a algún lenguaje de programación

## 4) Verificación:

Verificar que su ejecución conduce al resultado deseado, con datos representativos del problema real

# AyPI – ¿Por qué necesitamos lenguajes?



# AyPI – Algoritmo

Un algoritmo es una secuencia ordenada de pasos elementales, exenta de ambigüedades, que lleva a la solución de un problema dado en un tiempo finito.

Se denomina paso elemental aquel que no puede volver a ser dividido en otros más simples (Instrucción).

## Controlar y ajustar el desempeño de la impresora

Desde el panel de control puede realizar diversas tareas para mantener la impresora en buen estado de funcionamiento.

### Verificar los niveles de tinta

Para verificar los niveles de tinta:

1. Pulse **Home** (Inicio).
2. Pulse ▼ para seleccionar **Mantenimiento**, luego pulse **OK** (Aceptar).
3. Con **Verificar los niveles de tinta** seleccionado, pulse **OK** (Aceptar).

La pantalla LCD muestra los niveles aproximados de tinta en los cartuchos para que pueda saber cuándo debe reemplazarlos.

NOTA: Puede solicitar cartuchos de tinta en línea. Abra el software Home Center y seleccione **Solicitar consumibles**.

### Restablecer los parámetros predeterminados de fábrica

Para restablecer los parámetros predeterminados de fábrica:

1. Pulse **Home** (Inicio).
2. Pulse ▼ para seleccionar **Parámetros de la impresora**, luego pulse **OK** (Aceptar).
3. Pulse ▼ para seleccionar **Restablecer todos los parámetros**, luego pulse **OK** (Aceptar).
4. Cuando vea el mensaje: "Esta operación restablecerá los parámetros predeterminados de fábrica", pulse **OK** (Aceptar).

NOTA: Restablecer los parámetros predeterminados de fábrica no modifica la fecha y la hora, el idioma, el país/región ni los parámetros de red.

# AyPI – Algoritmos informáticos

Home / Cómo funciona el algoritmo de YouTube

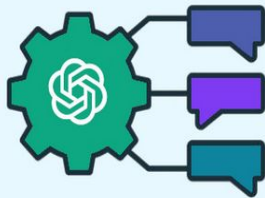
## Cómo funciona el algoritmo de YouTube



### ChatGPT: ¿Cómo funciona este algoritmo de PNL?

4 Abr • 11:17 am m de lecture

Data Science



DESARROLLO Y BASE DE DATOS NOTICIAS Y NUEVAS TECNOLOGÍAS

### GPS, Waze: ¿cómo funcionan sus algoritmos?

3 enero 2023 / Arnaud Guignant

Trabajar en el desarrollo de TI a menudo lleva a interesarse por el funcionamiento de las herramientas que se usan en el día a día. Este fue el caso en la carretera recientemente, con el lanzamiento de Waze.



# AyPI – Algoritmos más allá de la informática



## UCSG – IV Ciclo – Deber 01 – Algoritmo DE Reparación Tisular Y Cicatrización – Fisiopatología I

ALGORITMO DE REPARACIÓN TISULAR Y CICATRIZACIÓN

Asignatura

Fisiopatología (Medicina) 125 documentos

Universidad

Universidad Católica de Santiago de Guayaquil

## Algoritmo de reparación de PC

### Puesta en marcha

DE UN PC

Cuando nos enfrentamos al reto de poner en marcha un ordenador y con todas sus aplicaciones, encontramos dos caminos a solucionar. REPARACIÓN Y CONFIGURACIÓN

#### REPARACIÓN DEL EQUIPO

Si el equipo no enciende es un problema físico



**Paso 1**  
Revisión de corriente



**Paso 2**  
Comprobación componentes



**Paso 3**  
Accedemos a la BIOS

#### INSTALACIÓN Y CONFIGURACIÓN

Si el equipo enciende habrá que afrontar la instalación y/o configuración



**Paso 1**  
El S.O. arranca



**Paso 2**  
Comprobación de configuración



**Paso 3**  
Instalar las aplicaciones

# AyPI – PRE y POST condiciones

## PRE CONDICIÓN



**Es la información que se conoce como verdadera antes de iniciar el programa (ó módulo).**

## POST CONDICIÓN

**es la información que debería ser verdadera al concluir el programa (ó módulo), si se cumplen adecuadamente los pasos especificados.**

# AyPI PRE y POST condiciones

Problema: sacar \$5.000 del cajero automático



Problema: realizar una videollamada por WhatsApp





**Un “dato” es la representación de un elemento u objeto del mundo real, que los algoritmos procesan o modifican.**

*Supongamos los siguientes problemas:*

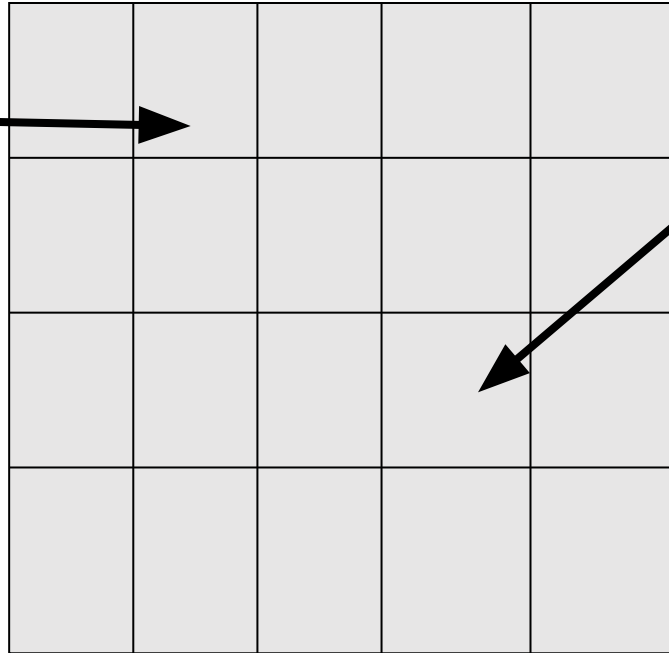
- Nieves fue de compras al shopping. Al salir, informar en **cuántos locales** entró.
- Sumar el **gasto total de todas las compras** realizadas por Nieves en el shopping.

# AyPI – Datos Variables y datos Constantes



**Variable NOMBRE**

Referencia una zona  
de memoria



**Constante NOMBRE**

Referencia una zona  
de memoria

**¿En qué se  
diferencian?**

# AyPI – Datos Variables y datos Constantes



## Variables

**Es una zona de memoria cuyo contenido puede cambiar su valor durante la ejecución del programa.**



## Constante

**Es una zona de memoria cuyo contenido NO puede cambiar su valor durante la ejecución del programa.**

# AyPI – Variables y Constantes



**¿Qué información resulta necesaria  
declararla como variables?**



**¿Qué información resulta útil declararla  
como constante?**

# AyPI – Estructura de un programa (En lenguaje Pascal)



Program nombre;

Const                      **Constantes del programa**

.....

módulos *{luego veremos como se declaran}*                      **Módulos del programa**

Var  
                              **Variables del programa**

begin  
    ...                      **Cuerpo del programa**  
end.



# AyPI – Variables



- Cada dato a representar (una variable) lleva un nombre asociado (un identificador).
- Cada variable debe tener un tipo asociado.
- Cada identificador será utilizado para almacenar el valor correspondiente.
- Notación de declaración: Identificador : tipo asociado
- Notación de asignación: Identificador := valor a guardar

```
program ejemplo1;  
var  
    { declaramos dos variables }  
    x, y : tipo;  
begin  
    { usamos las variables }  
    x := ...;  
    y := x;  
end.
```

# AyPI – Tipo de dato



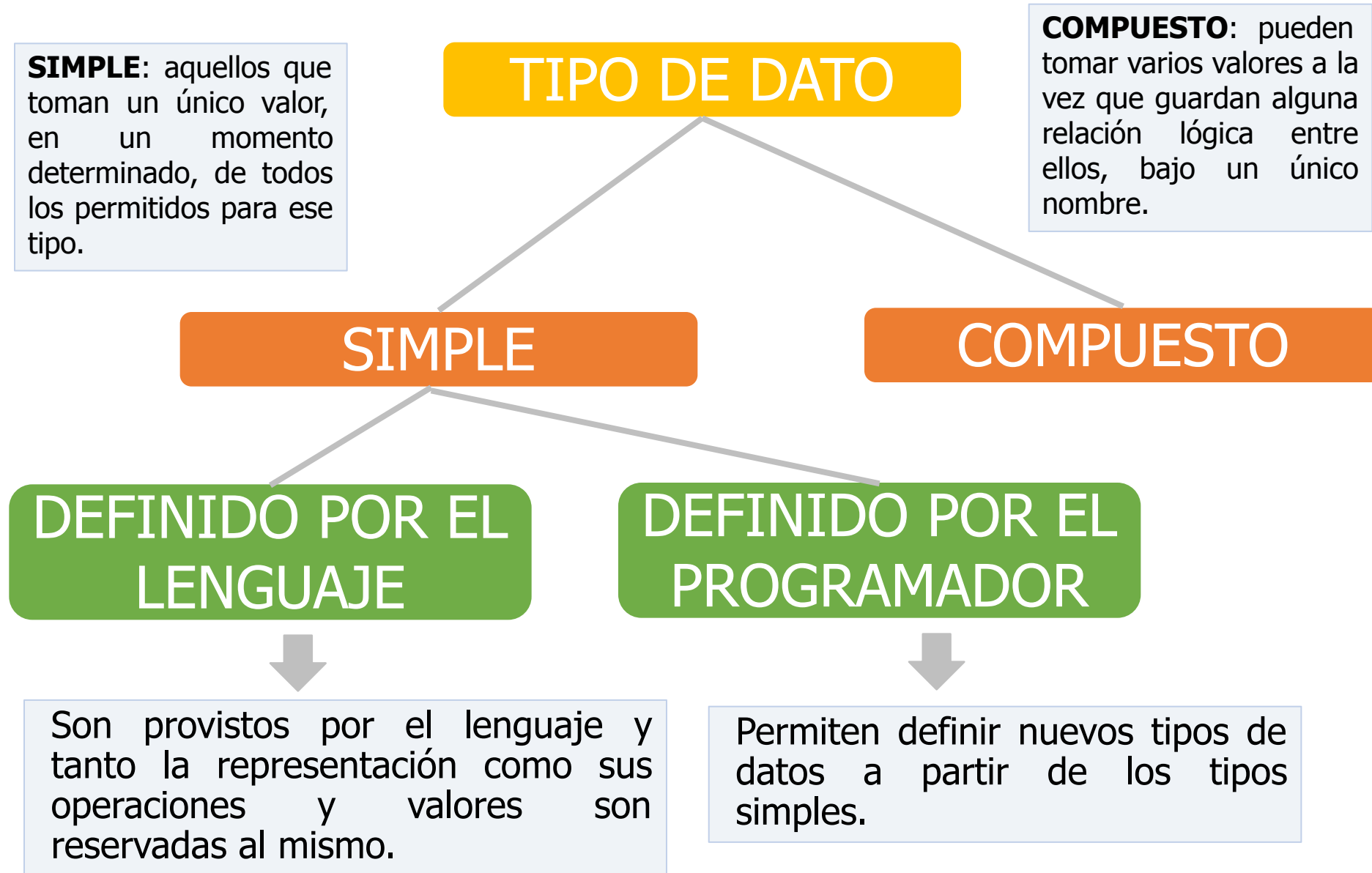
**Es una clase de objetos de datos ligados a un conjunto de operaciones para crearlos y manipularlos.**

Tienen un rango de valores posibles

Tienen una representación interna

Tienen un conjunto de operaciones permitidas

# AyPI – Tipo de Dato - Clasificación





## DATO NUMÉRICO

Representa el conjunto de números que se pueden necesitar. Estos números pueden ser enteros o reales.

Tipo de datos  
**integer**  
**(entero)**

Es un tipo de dato simple, ordinal

Los valores son de la forma  
-10 , 200, -3000, 2560

Pueden tomar valores dentro de un rango predefinido dependiendo de la representación interna.



## Operaciones

### Operadores Matemáticos

- +
- -
- \*
- /

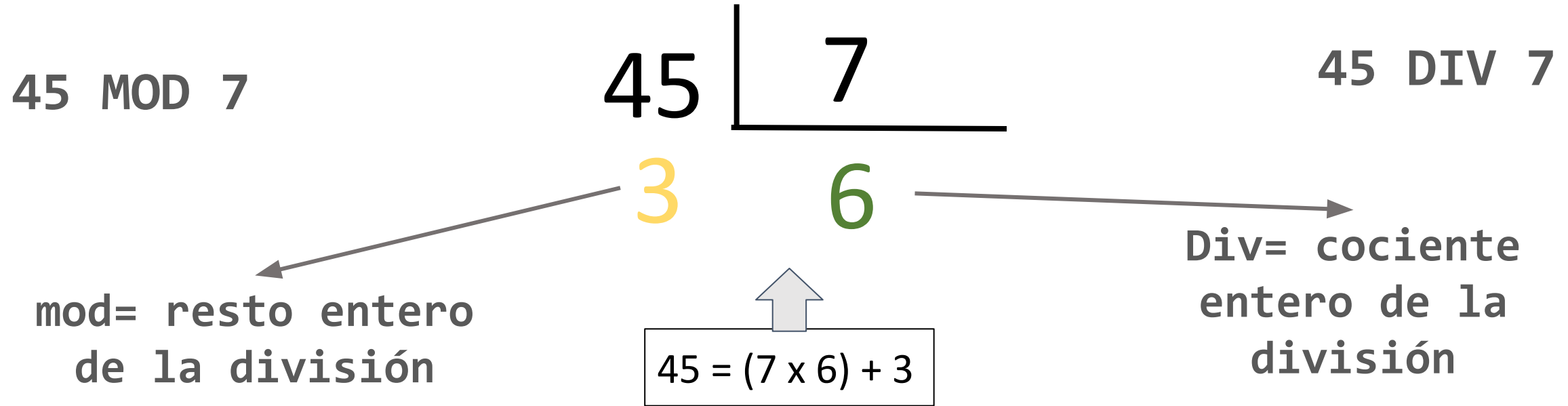
### Operadores Lógicos

- <
- >
- =
- <=
- >=

### Operadores Enteros

- Mod
- Div

¿Qué es div y mod?



```
program ejemplo;  
var a, b, c, d: integer;  
begin  
  a:= 45; b:= 7;  
  c:= a DIV b;  
  d:= a MOD b;  
end.
```



## DATO NUMÉRICO

**Representa el conjunto de números que se pueden necesitar. Estos números pueden ser enteros o reales.**

Tipo de datos  
**real**

Es un tipo de dato simple, permiten representar números con decimales

Los valores son de la forma

-10 , 200, -3000, 2560, 11.5, -22.89

Pueden tomar valores dentro de un rango predefinido dependiendo de la representación interna.



### Operaciones

#### Operadores Matemáticos

- +
- -
- \*
- /

#### Operadores Lógicos

- <
- >
- ==
- <=
- == >





Pensar tres casos en los cuales para representar los datos se debe utilizar un número real y no un entero.

Pensar un caso en el cual para representar el dato se debe utilizar un número entero y no un real.



Las expresiones que tienen dos o más operandos requieren reglas matemáticas que permitan determinar el orden de las operaciones.

El orden de precedencia para la resolución, ya conocido, es:

1. operadores **\***, **/**, **div** y **mod**
2. operadores **+**, **-**

En caso que el orden natural deba ser alterado, es posible la utilización de paréntesis dentro de la expresión.



```
program ejemplo;  
var a, b: integer; precedencia  
    c, d: real;  
begin  
    a:= 22;    b:= 5;  
    c:= a / b + 3 * 2;  
    d:= a / (b + 3) * 2;  
end.
```



Tipo de datos  
**boolean**  
**(lógico)**

## DATO LÓGICO

**Permite representar datos que pueden tomar dos valores verdadero o falso.**

Es un tipo de dato simple, ordinal

Los valores son de la forma  
true (verdadero)  
false (falso)

Operadores

- **AND** (conjunción)
- **OR** (disyunción)
- **NOT** (negación)

# AyPI – Dato lógico – Uso de operadores

- **X AND Y** : sólo es TRUE cuando tanto X como Y son TRUE. Cualquier otra combinación será FALSE
- **Y OR Y**: sólo es FALSE cuando tanto X como Y son FALSE. Cualquier otra combinación será TRUE
- **NOT X**: será TRUE cuando X sea FALSE, y será FALSE cuando X sea TRUE

```
program ejemploBoolean;  
var  
    x, y, z : boolean;  
begin  
    x := true;    y := false;  
    z := NOT (x AND y) OR y;  {¿qué valor toma la variable z?}  
end.
```



## DATO CHARACTER

**Representa un conjunto finito y ordenado de caracteres que la computadora reconoce. Un dato de tipo character contiene sólo un carácter.**

Tipo de datos

**char**  
**(caracter)**

Es un tipo de dato simple, ordinal

Los valores son de la forma

a B ! \$ L 4



## Operaciones

### Operadores Lógicos

- $<, <=$
- $>, >=$
- $=$
- $<>$

```
!"#$%&'()*+,-./
0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

Hay 95 caracteres ASCII imprimibles,  
numerados del 32 al 126.

La Tabla ASCII contiene todos los caracteres y el orden entre los mismos. <http://ascii.cl/es/>



```
program ejemploChar;  
var  
    car1, car2, car3, : char;  
    ok : boolean;  
begin  
    car1:= 'a';   car2:='D'; car3 := '1';  
    ok := car1 < car2;  
    ok := car1 >= car3;  
    ok := car1 = 'A';  
end.
```

**¿Qué resultado  
dan estas  
operaciones ?**

# AyPI – Estructura de un programa



Program nombre;

Const

N = 25;

pi = 3.14;

**Constantes del  
programa**

{módulos; luego veremos como se declaran}

var

edad: integer;

peso: real;

letra: char;

resultado: boolean;

**Variables del  
programa**

begin

edad:= 5;

peso:= -63.5;

edad:= edad + N;

letra:= 'A';

resultado:= letra = 'a';

**Cuerpo del programa**

end.





Los diferentes tipos de datos deben especificarse y a esta especificación dentro de un programa se la conoce como declaración.

Una vez declarado un tipo podemos asociar al mismo variables, es decir nombres simbólicos que pueden tomar los valores característicos del tipo.

Algunos lenguajes exigen que se especifique a qué tipo pertenece cada una de las variables. Verifican que el tipo de los datos asignados a esa variable se correspondan con su definición. Esta clase de lenguajes se denomina fuertemente tipados (**strongly typed**).

Otra clase de lenguajes, que verifica el tipo de las variables según su nombre, se denomina auto tipados (**self typed**).

Existe una tercera clase de lenguajes que permiten que una variable tome valores de distinto tipo durante la ejecución de un programa. Esta se denomina dinámicamente tipados (**dinamically typed**).

# AyPI – Operaciones de lectura/escritura



```
Program uno;
```

```
var
```

```
    edad: integer;
```

```
    valor: integer;
```

```
begin
```

```
    read (edad);
```

```
    valor:= edad + 15;
```

```
end.
```

**¿Cómo  
funciona el  
read?**



## READ

**Es una operación que contienen la mayoría de los lenguajes de programación. Se usa para tomar datos desde un dispositivo de entrada (por defecto desde teclado) y asignarlos a las variables correspondientes.**



```
Program uno;  
var  
    cant: integer;  
Begin  
    read (cant);  
End.
```

**El usuario ingresa un valor, y ese valor se guarda en la variable asociada a la operación read.**

# AyPI – OPERACIONES DE LECTURA/ESCRITURA



```
Program uno;
```

```
var
```

```
    edad: integer;
```

```
    valor: integer;
```

```
begin
```

```
    read (edad);
```

```
    valor:= edad + 15;
```

```
    write (valor);
```

```
end.
```

**¿Cómo  
funciona el  
write?**



## WRITE

Es una operación que contienen la mayoría de los lenguajes de programación. Se usa para mostrar el contenido de una variable, por defecto en pantalla.

```
Program uno;  
var  
    cant: integer;  
Begin  
    read (cant);  
    cant:= cant + 1;  
    write (cant);  
End.
```

*¿Variantes  
del write?*

**El valor almacenado en la variable asociada a la operación write, se muestra en pantalla.**

# AyPI – Tipos de Datos

## WRITE

¿Cómo altero el  
orden de  
ejecución?

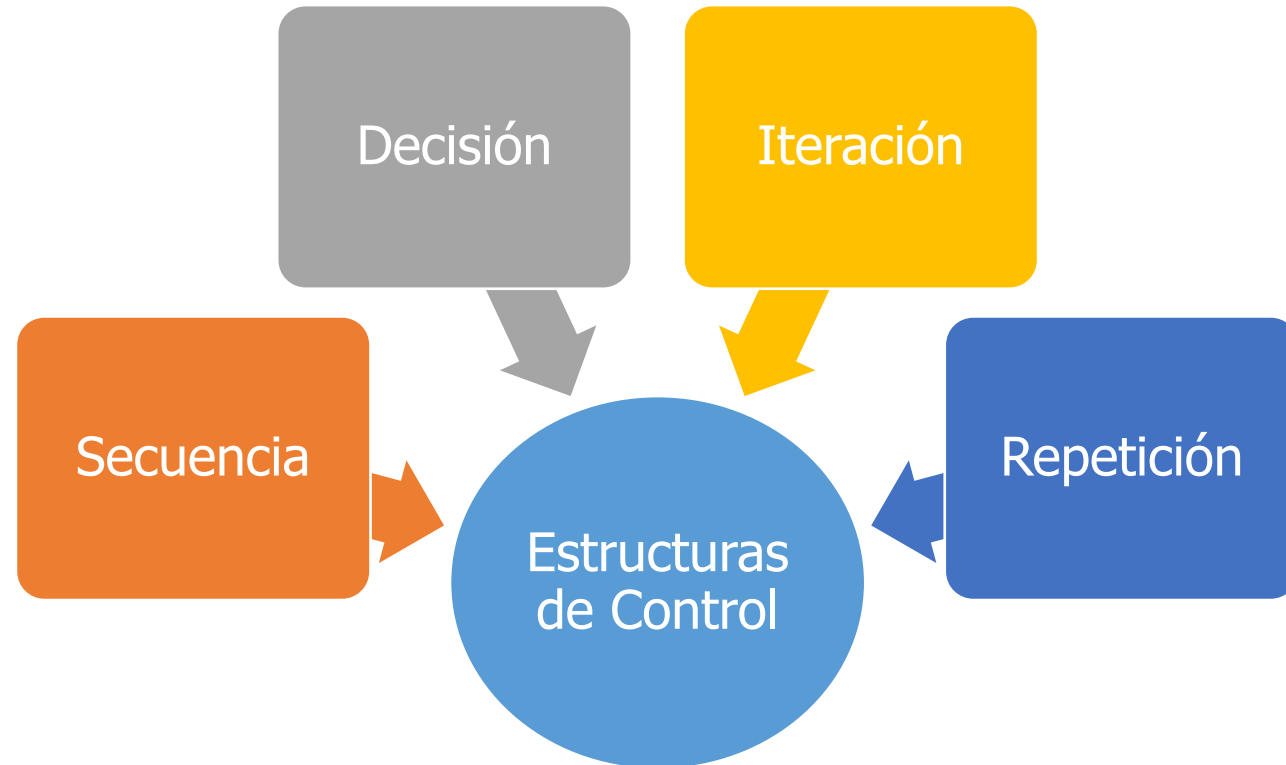
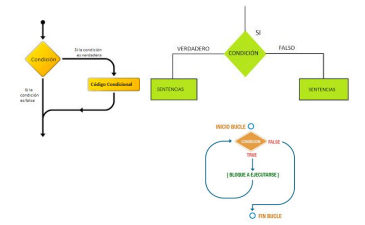


```
Program uno;  
var  
    ...  
Begin  
    ...  
    write ('texto');      Write ('Los valores ingresados son 0')  
  
    write (variable);     Write (num);  
  
    write ('texto',variable); Write ('El resultado es:',num);  
  
    write ('texto', resultado de una operación); Write ('El resultado  
End.                                             es:',num+4);
```

# AyPI – Estructuras de control



**Todos los lenguajes de programación tienen un conjunto mínimo de instrucciones que permiten especificar el control del algoritmo que se quiere implementar. Como mínimo deben contener: secuencia, decisión e iteración.**



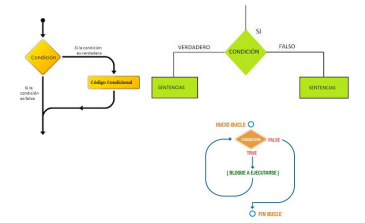
# AyPI – Estructuras de control



## SECUENCIA

La estructura de control más simple, está representada por una sucesión de operaciones (por ej. asignaciones), en la que el orden de ejecución coincide con el orden físico de aparición de las instrucciones.

```
Program uno;  
  
...  
var  
    num:integer;  
begin  
    read (num);  
    write (num);  
end.
```



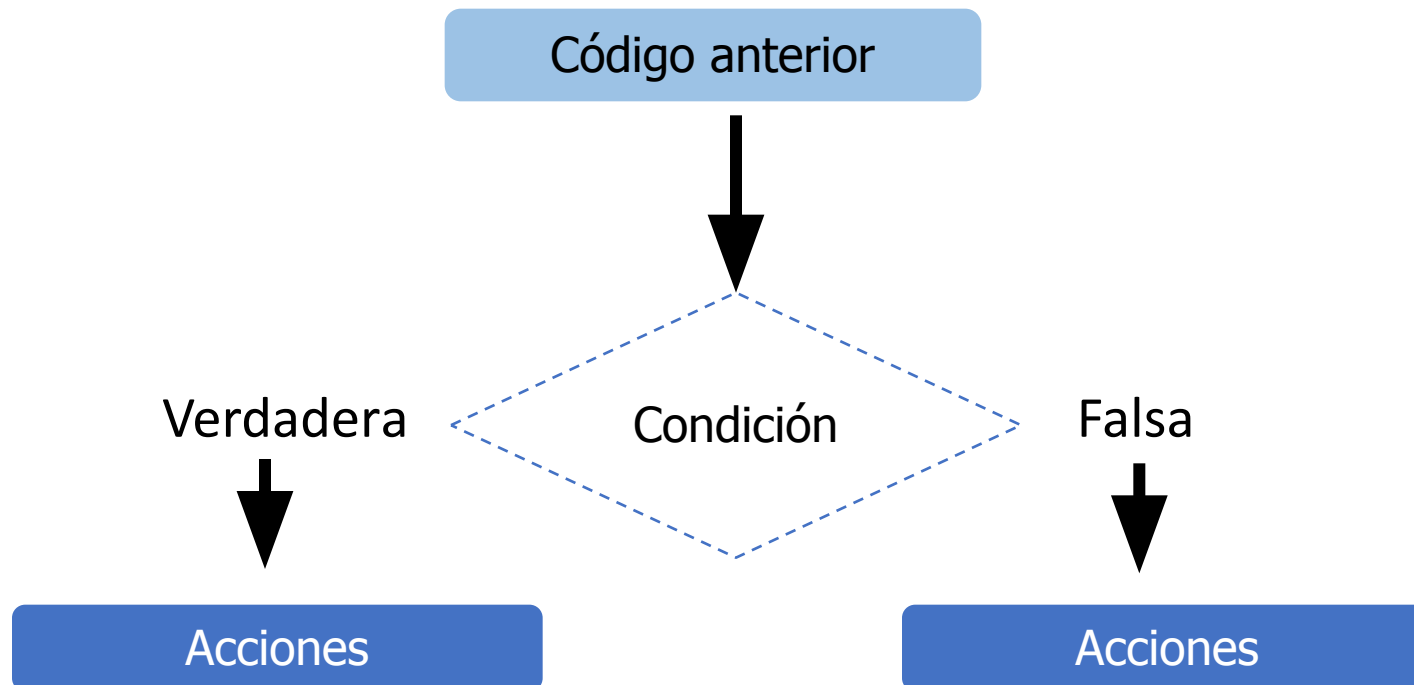


# AyPI – Estructuras de control

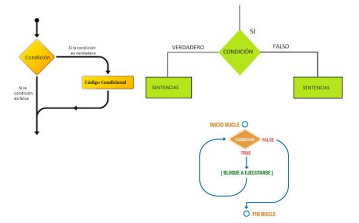


## DECISIÓN

En un algoritmo representativo de un problema real es necesario tomar decisiones en función de los datos del problema. La estructura básica de decisión entre dos alternativas es la que se representa simbólicamente:



¿Cómo es la sintaxis?



# AyPI – Estructuras de control

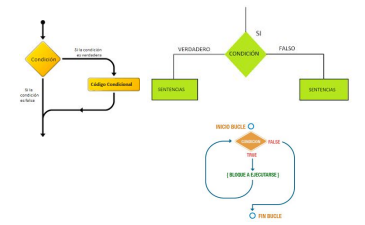
```
if (condición) then  
    acción;
```

  
más de  
una acción

```
if (condición) then  
    acción 1  
else  
    acción 2;
```

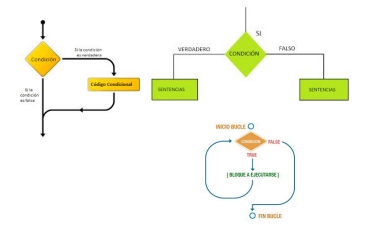
```
if (condición) then  
    begin  
        acción 1;  
        acción 2;  
    end  
else  
    acción 3;
```

## DECISIÓN



```
if (condición) then  
    begin  
        acción 1;  
        acción 2;  
    end;
```

```
if (condición) then  
    begin  
        acción 1;  
        acción 2;  
    end  
else  
    begin  
        acción 3;  
        acción 4;  
    end  
end;
```

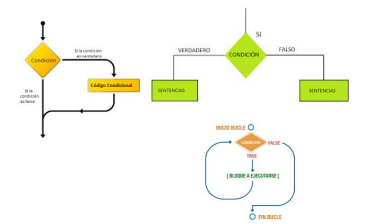


**Realice un programa que lea un carácter y determine si es una vocal minúscula.**

- Cómo leo un carácter
- Cómo veo si es vocal
- Cómo muestro el resultado

# AyPI – Estructuras de control

## DECISIÓN



Realice un programa que lea un carácter y determine si es una vocal minúscula.

```
Program uno;  
var  
    car:char;  
  
begin  
    read (car);      Leo un caracter  
    if ((car = 'a') or (car = 'e') or (car = 'i')  
        or (car = 'o') or (car = 'u'))      Veo si es vocal  
    then  
        write ("El carácter es vocal minúscula")  
    else      Informo el resultado  
        write ("El caracter no es vocal minúscula");  
end.
```

# AyPI – CONSULTAS

