

En este Curso ...

- Aprenderemos a usar el computador como una herramienta para resolver problemas de procesamiento de datos.
- Para eso tenemos que indicarle claramente lo que debe hacer.
- Debemos PROGRAMARLO !







Para programar debemos ...

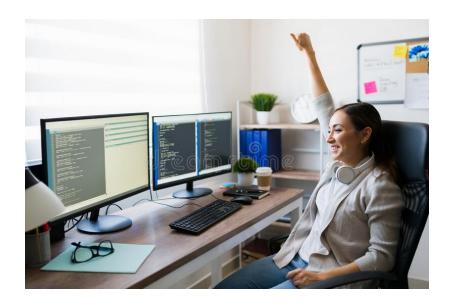
- Desarrollar habilidades de análisis, abstracción, clasificación y generalización de problemas.
- Aprender a **modelar y diseñar soluciones** que resuelvan problemas.
- Aprender un **lenguaje de programación** para escribir nuestra solución computacional.





Para ser un programador/a eficaz ...

- Debemos programar de manera formal, sistemática y rigurosa.
- Aplicando una **METODOLOGÍA** !







PASOS GENERALES PARA RESOLVER UN PROBLEMA

Entender el problema



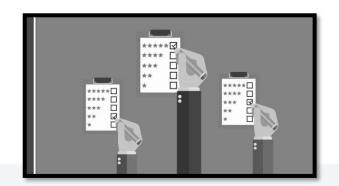
Ejecutar el Plan



Idear un Plan para Resolverlo



Evaluar la Solución







FASES EN LA RESOLUCIÓN DE UN PROBLEMA CON COMPUTADOR

- Análisis del Problema
- Diseño del Algoritmo
- Codificación
- Traducción y ejecución
- Verificación
- Depuración
- Mantención
- Documentación





FASES EN LA RESOLUCIÓN DE UN PROBLEMA CON COMPUTADOR

Análisis del Problema

• El problema se analiza teniendo presente las especificaciones de los requerimientos.

Diseño del Algoritmo

• Se diseña una solución que conduce a un algoritmo que resuelve el problema.

Codificación y traducción

• La solución se escribe en la sintaxis de un lenguaje de alto nivel y se obtiene un programa fuente que se traduce a continuación.

Ejecución , verificación y depuración

• El programa se ejecuta, se comprueba rigurosamente y se eliminan todos los errores que puedan aparecer.

Mantención

• El programa se actualiza y modifica todas las veces que sea necesario hasta cumplir con las necesidades de cambio de los usuarios.

Documentación

• Se escriben los informes de análisis, diseño y codificación, manuales de usuario y referencia.





PRIMER PROBLEMA

CÁLCULO ÁREA Y PERÍMETRO DE UNA CIRCUNFERENCIA

Pedro esta estudiando Geometría Euclidiana y necesita calcular el **Área** y **Perímetro** de una circunferencia. Él necesita un programa computacional que realice ambos cálculos.







ANÁLISIS DEL PROBLEMA

ANÁLISIS DEL PROBLEMA

En esta **FASE** debemos :

- Leer el enunciado del problema.
- Identificar qué datos de entrada se requieren (tipo de y cantidad).
- Identificar cuál es la salida deseada (tipo y cantidad).
- Identificar qué método produce la salida deseada.





ANÁLISIS

IDENTIFICAR
DATOS DE
ENTRADA

IDENTIFICAR
DATOS DE
SALIDA

IDENTIFICAR MÉTODO DE SOLUCIÓN





Problema Área y Perímetro de una Circunferencia

Pedro esta estudiando Geometría Euclidiana y necesita calcular el **Área** y **Perímetro** de una circunferencia. Él necesita un programa computacional que realice ambos cálculos.







Datos de Entrada

 Radio de la Circunferencia (valor de tipo real).

Datos de Salida

- Área de la Circunferencia (valor de tipo real).
- **Perimetro** de la Circunferencia (valor de tipo real).

Método de Solución

- Leer Radio de la Circunferencia
- Calcular Área de la Circunferencia
- Calcular Perímetro de la Circunferencia
- Escribir Resultados (Área y Perímetro).





DISEÑO DEL ALGORITMO

DISEÑO DEL ALGORITMO

- En esta **FASE** se debe diseñar el **algoritmo** que resuelve el problema.
- Un algoritmo es conjunto ordenado de pasos que indican las operaciones que se deben realizar para resolver un problema.





CARACTERÍSTICAS DE UN ALGORITMO

- **Finito:** tiene un número determinado de pasos.
- Preciso : indica el orden de realización de cada paso.
- Efectivo: al ejecutar la secuencia de pasos establecidos se obtiene el resultado esperado.





PARTES DE UN ALGORITMO

ENTRADA

Leer los Datos que se procesarán

PROCESO

Realizar
Operaciones
sobre los
Datos leídos

SALIDA

Escribir los resultados obtenidos





ALGORITMO PARA FREIR UN HUEVO

paso a paso 2





Problema Área y Perímetro de una Circunferencia

Pedro esta estudiando Geometría Euclidiana y necesita calcular el **Área** y **Perímetro** de una circunferencia. Él necesita un programa computacional que realice ambos cálculos.







Algoritmo en Lenguaje Natural

PASOS GENERALES PARA RESOLVER EL PROBLEMA

- 1. Leer valor del radio de la Circunferencia
- 2. Calcular el **área** de la Circunferencia usando la fórmula correspondiente.
- 3. Calcular el **perímetro** de la Circunferencia usando la fórmula correspondiente.
- 4. Escribir el valor del área de la Circunferencia
- 5. Escribir el valor del **perímetro** de la Circunferencia





ESCRITURA DEL ALGORITMO EN NOTACIÓN FORMAL

- Una vez establecida la secuencia ordenada de pasos que describen las operaciones generales que el computador debe realizar para llevar a cabo la tarea en lenguaje natural.
- Debemos escribir el algoritmo utilizando una técnica de representación formal y estandarizada, que puede ser:
 - **Descriptiva** : Pseudocódigo
 - **Gráfica** : Diagrama de Flujo





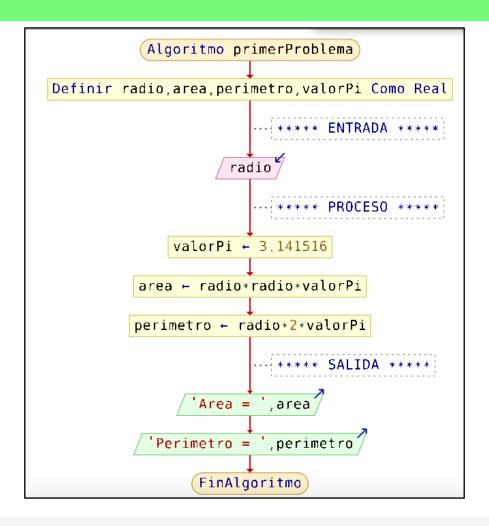
ALGORITMO EN PSEUDOCÓDIGO

```
Algoritmo primerProblema
 Definir radio, area, perimetro, valorPi Como Real
// ***** ENTRADA *****
Leer radio
// ***** PROCESO *****
valorPi <- 3.141516
 area <- radio * radio * valorPi
 perimetro <- radio * 2 * valorPi
 // ***** SALIDA *****
 Escribir 'Area = ', area
 Escribir 'Perimetro = ', perimetro
FinAlgoritmo
```





ALGORITMO EN DIAGRAMA DE FLUJO







CODIFICACIÓN Y TRADUCCIÓN

Codificación y Traducción

- En esta **FASE** debemos escribir cada paso de nuestro **algoritmo** como una **instrucción** equivalente en un **Lenguaje de Programación de alto nivel** para obtener un **programa fuente**.
- A continuación, el programa fuente debe ser traducido a lenguaje de máquina para poder ejecutarlo.
- Si el **programa fuente** tiene errores de **sintaxis** se deben identificar y corregir.





Codificación Algoritmo



```
1  from math import pi
2
3  # *** ENTRADA ***
4  radio = float(input())
5
6  # *** PROCESO ***
7  area = radio ** 2 * pi
8  perimetro = radio * 2 * pi
9
10  # *** SALIDA ***
11  print("Area = ", area)
12  print("Perimetro = ", perimetro)
```





Errores de Traducción

Se producen por el uso incorrecto de las <mark>reglas</mark> del lenguaje (errores de sintaxis)

```
Powered by mtrinket
< >
      main.py
                                   + 4
     from math import pi
                                              File "/tmp/sessions/b29c3c7464612848/main.py", line 7
                                                area := radio ** 2 * pi
     # *** ENTRADA ***
     radio = float(input())
                                            SyntaxError: invalid syntax
    # *** PROCESO ***
  7 area := radio ** 2 * pi
  8 perimetro := radio * 2 * pi
 10 # *** SALIDA ***
 11 print('Área = ', area)
 12 print('Perímetro = ', perimetro)
```





EJECUCIÓN, VERIFICACIÓN Y DEPURACIÓN

Ejecución, Verificación y Depuración

 En esta FASE se debe ejecutar el programa traducido, verificar que entregue los resultados esperados correctamente y en caso contrario depurarlo.

Los pasos a seguir son :

- Diseñar casos de prueba (normales / valores extremos / excepciones).
- Ejecutar el programa con cada caso de prueba verificando que los resultados obtenidos sean los esperados.
- Si los **resultados son** incorrectos, detectar los **errores** en el código y corregirlos (**depurar**).





Ejecución y Prueba del Programa

```
1  from math import pi
2
3  # *** ENTRADA ***
4  radio = float(input())
5
6  # *** PROCESO ***
7  area = radio ** 2 * pi
8  perimetro = radio * 2 * pi
9
10  # *** SALIDA ***
11  print("Area = ", area)
12  print("Perimetro = ", perimetro)
```

	Entrada	Esperado	Se obtuvo	
✓	1	Area = 3.141592653589793 Perimetro = 6.283185307179586	Area = 3.141592653589793 Perimetro = 6.283185307179586	✓
√	2	Area = 12.566370614359172 Perimetro = 12.566370614359172	Area = 12.566370614359172 Perimetro = 12.566370614359172	√
✓	3	Area = 28.274333882308138 Perimetro = 18.84955592153876	Area = 28.274333882308138 Perimetro = 18.84955592153876	√
√	4	Area = 50.26548245743669 Perimetro = 25.132741228718345	Area = 50.26548245743669 Perimetro = 25.132741228718345	✓
√	5	Area = 78.53981633974483 Perimetro = 31.41592653589793	Area = 78.53981633974483 Perimetro = 31.41592653589793	✓

Todas las pruebas superadas. <





Errores Comunes

- Errores de Lógica : Se producen en la lógica del programa, la fuente del error esta en el diseño del algoritmo. Son los más difíciles de detectar.
- Errores de Ejecución : Se producen por instrucciones que el computador comprende pero no puede ejecutar. Por ejemplo: una división por cero.





Ejecución y Prueba del Programa

```
1  from math import pi
2
3  # *** ENTRADA ***
4  radio = float(input())
5
6  # *** PROCESO ***
7  area = radio * 2 * pi
8  perimetro = radio * 2 * pi
9
10  # *** SALIDA ***
11  print("Area = ", area)
12  print("Perimetro = ", perimetro)
```

ERROR DE LÓGICA

	Entrada	Esperado	Se obtuvo	
×	1	Area = 3.141592653589793 Θ Perimetro = 6.283185307179586	Area = 6.283185307179586↔ Perimetro = 6.283185307179586	×
✓	2	Area = 12.566370614359172↔ Perimetro = 12.566370614359172	Area = 12.566370614359172↔ Perimetro = 12.566370614359172	√
×	3	Area = 28.274333882308138↔ Perimetro = 18.84955592153876	Area = 18.84955592153876↔ Perimetro = 18.84955592153876	×
×	4	Area = 50.26548245743669↔ Perimetro = 25.132741228718345	Area = 25.132741228718345↔ Perimetro = 25.132741228718345	×
×	5	Area = 78.53981633974483₽ Perimetro = 31.41592653589793	Area = 31.41592653589793 → Perimetro = 31.41592653589793	×

Your code must pass all tests to earn any marks. Try again.

Ocultar diferencias





FASES EN LA RESOLUCIÓN DE UN PROBLEMA CON COMPUTADOR - CURSO

Análisis del Problema

• El problema se analiza teniendo presente las especificaciones de los requerimientos.

Diseño del Algoritmo

•Se diseña una solución que conduce a un algoritmo que resuelve el problema.

Codificación y traducción

• La solución se escribe en la sintaxis de un lenguaje de alto nivel y se obtiene un programa fuente que se traduce a continuación.

Ejecución , verificación y depuración

•El programa se ejecuta, se comprueba rigurosamente y se eliminan todos los errores que puedan aparecer..





