

Programación

Licenciatura en Física

Dr. Manuel Pulido

¹ FaCENA, Universidad Nacional del Nordeste, Argentina

² CONICET, Argentina

³ Data Assimilation Research Centre, University of Reading, UK

Oficina: 1er Piso. FaCENA. Campus Libertad.

pulido@exa.unne.edu.ar

<https://pulidom.github.io/>

Horarios de clases

Jueves. 14-18 horas.

Viernes. 14-18 horas. Recuperatorio. Practicos. Consultas. (algun otro día disponible?).

Para regularizar: Aprobar dos parciales (60%)

Para promocionar: Aprobar los dos parciales con 80% o mas.

Lo único que requiere esta materia para aprobar es tiempo para leer y practicar en una computadora.

Disponibilidad del laboratorio: Se puede coordinar para que utilizen las computadoras del laboratorio de física.

Objetivos del curso

- ▶ Aprender a manejar una computadora.
- ▶ Aprender a pensar algorítmicamente.
- ▶ Aprender a programar (introducción).
- ▶ Aprender a leer datos, procesar datos, grabar datos y graficar (introducción).

Temas incluidos

- ▶ Conocimiento de sistema operativo linux.
- ▶ Manejo por ventanas y por terminal.
- ▶ Conocimiento básico de shell/bash (lenguaje terminal).
- ▶ Programación en python.

Bibliografía: Para cada tema hay bibliografía específica. Es una materia en la que se incentiva el 'googleo'.

¿Porque es importante la computación en la Lic. en Física?

Porque la computación en la Lic. en Física?

Actualmente, la computadora es una herramienta indispensable para el trabajo en nuestra carrera:

- ▶ Procesamiento de datos (experimentales/teóricos).
- ▶ Análisis y graficación de datos (experimentales/teóricos).
- ▶ Deducciones analíticas (álgebra simbólica).
- ▶ Procesamiento de textos (html, latex, word, etc).
- ▶ Simulaciones numéricas de problemas complejos (sin resolución analítica).

Pilares de la Física (y la ciencia en general)

- ▶ Experimentación u observación
- ▶ Teoría (deducciones a partir de primeros principios)
- ▶ Simulaciones

Que es una computadora?

Que es una computadora?

“Máquina” que permite realizar automaticamente una **serie de procesos lógicos y aritméticos** a través de una CPU (unidad de procesamiento central). Además dispone de canales de interacción entrada/salida (input/output) con:

1. Memoria RAM (random access memory). Muy **rápido** acceso pero se pierde al apagar el equipo. (Relativamente pequeña)
2. Memoria ROM (read only memory). Solo lectura (instrucciones complejas/firmware)
3. Disco rígido (o solid state drive SSDs). Acceso de escritura y lectura **lento** el contenido permanece si se apaga el equipo. (Relativamente grande)
4. Dispositivos varios: Monitor, teclado, parlantes, pen drive, etc.

Que es programar una computadora?

Que involucra programar?

- ▶ Determinar el problema científico. Fenómeno a estudiar (materia, sistemas complejos, biológicos, sociales, etc).
- ▶ Modelado matemático del proceso a estudiar (ecuaciones).
- ▶ Metodología.
 - ▶ 1. Diagrama de flujo (Receta de cocina).
 - ▶ 2. Modelado numérico de las ecuaciones.
 - ▶ 3. Implementación en lenguaje computacional.
 - ▶ 4. Debuging. Tests. Evaluaciones (casos simples).
- ▶ Simulaciones. (compilación y ejecución del programa).
- ▶ Interpretación de la solución.

Algoritmo

¿Qué es una receta de cocina?

Algoritmo

1. Orden secuencial. Las instrucciones se realizan en orden.
2. Preciso y definido. Cada paso o instrucción esta especificada sin ambigüedad.
3. **No se esperan peras de un olmo.** El algoritmo solo resuelve las instrucciones que se le dan (y no posee datos que no han sido asignados). La computadora no hace milagros
4. Entrada/Salida. Un algoritmo dispone de datos de entrada y de datos de salida (resultados).

Similitud con la receta de cocina.

Instrucciones

Una instrucción es una orden precisa que se le da a la computadora para que realice un determinado proceso.

Las computadoras/lenguajes de programación/ reconocen un muy limitado conjunto de instrucciones. Solo se puede utilizar el conjunto de instrucciones predefinido.

Lenguaje máquina y de alto nivel (C, fortran, perl, python, basic, etc).

Cada instrucción tiene una precisa sintaxis. Agregar un punto o una falta de este nos dará error.

Ejemplo 1

Realizar un algoritmo para calcular el área de un círculo.

$$\text{área} = \pi \times \text{radio}^2$$

¿Cual sería la receta de cocina para calcular el área?

Pseudo-código área de un círculo

1. Ingresar radio del círculo
2. Asignar $\pi = 3.141593$
3. Calcular $\text{área} = \pi \times \text{radio}^2$

Entonces un pseudo-código utiliza un lenguaje natural pero estableciendo en forma estricta el orden y cada instrucción requerida.

Diagrama de flujo

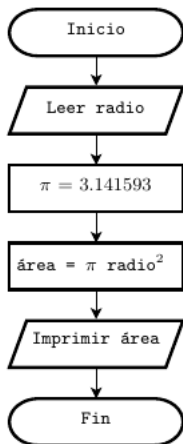


Diagrama de flujo: Es una representación gráfica de un algoritmo.

Cada tipo de instrucción tiene una **figura geométrica**.

Entre instrucciones se ponen flechas/vectores para representar el **sentido del flujo** (orden de ejecución de las instrucciones).

Ejemplo 2

Evaluar y graficar la funcion cuadratica: $y = ax^2$

Algoritmo evaluación función cuadrática

1. Ingresar el valor de a
2. Ingresar el valor inicial de x , x_0 .
3. Ingresar el valor de la resolución requerida dx .
4. Ingresar número de valores a evaluar n
5. Iniciar un ciclo/loop (de n ciclos)
6. Evaluo la función $y = ax^2$
7. Evaluo nueva coordenada $x = x_{viejo} + dx$
8. Termino el ciclo
9. Impresión/graficación del resultado.

Ejemplo 3

Transformar un número binario a su formato decimal

Algoritmo

1. Ingresar el número $b_n b_{n-1} \cdots b_3 b_2 b_1 b_0$. Como se determinan los dígitos?
2. Asignar $sum = 0$
3. Comienza ciclo i-ésimo $n + 1$.
4. Calcular $r_i = b_i 2^i$
5. Sumar $sum = sum + r_i$
6. Terminar ciclo
7. Imprimir resultado