Informática
Curso 2024

Facultad de Ingeniería
Universidad de la República

Introducción

- La tarea principal de una computadora es *ejecutar programas*.
- Para que ?
  - Procesar información.

Programación de Computadoras

- En el nivel más simple consiste en ingresar en la computadora una secuencia de órdenes para lograr un cierto objetivo
  - Elaboración de programas informáticos
- Es el arte de hacer que una computadora haga lo que nosotros queramos.
- Puede ser tan corto como de una sola instrucción o tan largo como de varios millones de instrucciones.

Programa Informático

- Pensemos en una receta:
  - Un grupo de instrucciones que le dicen al cocinero cómo preparar un determinado plato.
  - Describe los ingredientes (los datos) y la secuencia de pasos (el proceso) necesarios para convertir los ingredientes en una torta.
- Un programa tiene un concepto muy similar.

Programa Informático

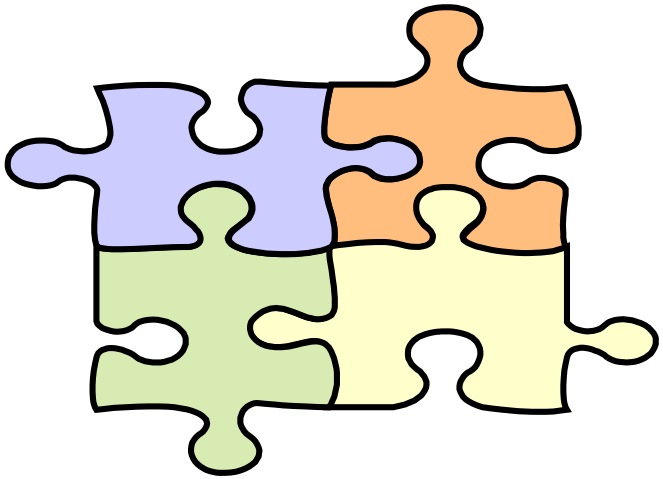
- Se compone de:
  - Los **datos** que forman parte del problema a resolver o que se requieren para resolverlo.
  - Las **acciones** necesarias para resolver el problema, y que procesan los datos.

Metodología de programación
Resolución de Problemas

- El proceso de resolución de un problema por medio de una computadora se compone de tres pasos básicos:
  - **Análisis del problema** y soluciones para el mismo.
  - Encontrar o **Diseñar un algoritmo** que resuelva el problema.
  - **Codificación** en un lenguaje de programación.

Análisis del problema

1. Identificar y partir en sub-problemas
2. Repetir **1.** hasta que los sub-problemas tengan solución conocida



Diseñar un Algoritmo

- Un ***algoritmo*** es un procedimiento detallado paso por paso para resolver un problema.
- Las **instrucciones** deben ser **claras y sin ambigüedades** y lo bastante específicas para ejecutarse y **terminarse** en un número **finito de pasos**.
  - Andar en bicicleta
  - Receta de cocina
  - Obtener el máximo de una lista de números

Diseñar un Algoritmo

- Un algoritmo es una secuencia de pasos
  - Precisos
    - Indicar el orden de realización de cada paso.
  - Definidos
    - Si se sigue más de una vez, obtiene el mismo resultado cada vez.
  - Finitos
    - Tiene fin: un número determinado de pasos

Diseñar un Algoritmo

- Algoritmo computacional
  - Se define como cualquier procedimiento computacional bien definido que toma valores como ***entrada*** y produce valores de ***salida***.
  - Son una secuencia de pasos que transforma datos de entrada en datos de salida.

Diseñar un Algoritmo

- Un algoritmo se puede expresar de distintas formas
  - Lenguaje Natural
  - En forma gráfica, usando diagramas de flujo
  - Utiizando un **pseudo-código**.

Representar el algoritmo formalmente

Pseudo-código

- El Pseudo-código es una representación de un programa en un lenguaje natural pero con formalidades propias de un lenguaje de programación
- Se tratará de escoger uno que ofrezca las mismas estructuras que el lenguaje en el que se prevé hacer la programación

Representar el algoritmo formalmente

Pseudo código: ejemplo

PALABRA	UTILIZACIÓN
ABRE	**Abre un archivo**
CIERRA	**Cierra un archivo**
CASO [ENOTROCASO]	**Selección entre múltiples alternativas. En otro caso, indica las acciones a realizar si no se cumple ninguno de los casos especificados.**
LEER	**Leer un dato del teclado**
ESCRIBE	**Visualiza un dato en pantalla**
HAZ	**Inicia la iteración HAZ – HASTA**
HASTA	**Cierra la iteración HAZ – HASTA**
MIENTRAS	**Inicia la iteración mientras**
PARA CADA	**Inicia un número fijo de iteraciones**
SI ENTONCES [SINO]	**Selección SI - ENTONCES – SINO**
INICIO	**Inicia un bloque de instrucciones**
FIN	**Finaliza un bloque de instrucciones**
NO	**Niega la condición que le sigue**
O	**Disyunción lógica**
Y	**Conjunción lógica**
{	**Inicio de comentario**
}	**Fin de comentario**
<=	**Asignación**

Ejemplo de pseudo-código

Colocar sobre en mesa

Problema: Elegir a dónde va el sobre según si es para Montevideo o para Canelones

No se sabe a priori para donde va.

Tomo el sobre

SI el sobre dice “Montevideo”

    Coloco en la mesa para Montevideo

SINO

    Coloco en la mesa para Canelones

FIN SI

Ejemplo de pseudo-código

Contar sobres en una bolsa

Problema: Contar sobres en una bolsa.

No se sabe a priori la cantidad de sobres.

Contar sobres en una bolsa:

Total de sobres es 0.

MIENTRAS haya sobres en la bolsa

    Saco un sobre

    Sumo 1 al total de sobres

FIN MIENTRAS

Ejemplo de pseudo-código

Contar sobres en bolsas

Problema: Contar sobres en bolsas.

No se sabe a priori la cantidad de bolsas ni de sobres.

Contar sobres en una bolsa:

Total de sobres es 0.

MIENTRAS haya sobres en la bolsa

    Saco un sobre

    Sumo 1 al total de sobres

FIN MIENTRAS

Contar todos los sobres en bolsas:

Total de sobres es 0.

MIENTRAS haya bolsas para contar

    Tomo una bolsa

    CS = Invocar Contar sobres en una bolsa

    Sumar CS al total de sobres

FIN MIENTRAS

Ejemplo de pseudo-código

Separar sobres de una bolsa

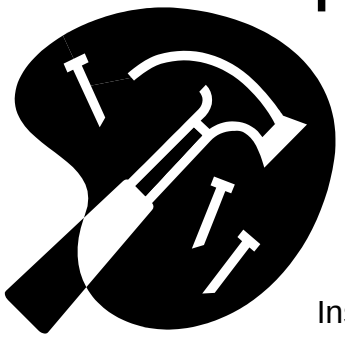
Problema: Separar sobres de una bolsa según va para Montevideo o para Canelones.

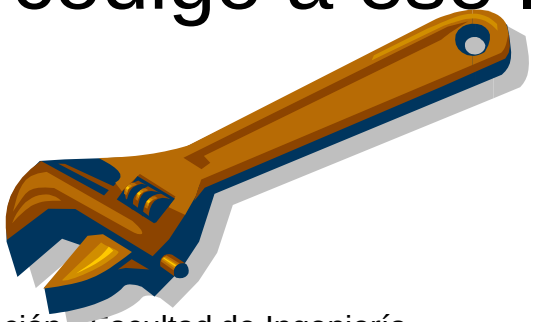
No se sabe a priori la cantidad de bolsas ni de sobres.

```
MIENTRAS haya sobres en la bolsa
        Tomo el sobre
        SI el sobre dice “Montevideo”
                Coloco en la mesa para Montevideo
        SINO
                Coloco en la mesa para Canelones
        FIN SI
FIN MIENTRAS
```

Codificar

- La ***codificación*** es el proceso de traducir un algoritmo en un lenguaje de programación.
- Escoger un lenguaje apropiado al tipo de aplicación
- Traducir el pseudo-código a ese lenguaje





Construcción de Programas

- Dos etapas
  - Programación
    - Análisis, planificación, diseño y construcción del programa
      - Se definen las acciones a realizar.
  - Ejecución
    - Puesta en práctica del mismo.
      - Las acciones se ejecutan.

Construcción de Programas

- Proceso de construcción
  - **Definición del problema**
    - Definirlo en términos sencillos.
    - Determinar entradas y salidas.
    - Identificar si tiene una solución conocida.
  - **Diseño de la solución**
    - Empleando algún método y herramientas: diagramas, lenguaje natural o pseudocodigo.
  - **Codificación**
    - Escritura de la solución en algún lenguaje de programación.

Construcción de Programas

- **Ejecución**
  - Si no hay errores de sintaxis detectados
- **Codificación**
  - Nuevamente si hay errores de sintaxis detectados en la Compilación
- **Diseño y Codificación**
  - Nuevamente si hay errores en la solución diseñada que se detectan en la Ejecución del programa.

Objetivos de la programación

- **Exactitud** en la realización de la tarea
  - Tiene que satisfacer la especificación exactamente.
    - Simplicidad. Elegir el algoritmo o técnica más simple disponible.
- **Eficiencia**:
  - Tiempo de ejecución
  - Uso de memoria RAM
  - Uso de otros recursos (gráficos, acceso a disco, de comunicación, etc.)

Objetivos de la programación

- **Claridad** del código fuente
  - Un programa es necesariamente tan complejo como el algoritmo que describe.
  - Se logra a través de:
    - Separación lógica en partes comprensibles que reflejen la distinción entre los temas que describen, y su presentación en una secuencia lógica que refleje las relaciones entre ellas.
    - Selección de las características del lenguaje.
    - Selección de las palabras usadas para denotar los objetos y conceptos involucrados
    - Inclusión de comentarios

Programa Informático

- Un programa se compone de
  - Datos
    - Son la representación dentro de la computadora de aspectos de la realidad.
    - Servirán para ser procesados y producir resultados.
  - Instrucciones.
    - Son la parte central del programa.
    - Manipulan los datos, realizan cálculos, muestran los resultados, etc.

Datos

- Tipos de datos
  - Enteros
  - **Reales**
  - Caracteres
  - **Palabras** (Strings)
  - **Vectores**
  - **Booleans** (verdadero, falso)
  - Otros

Datos

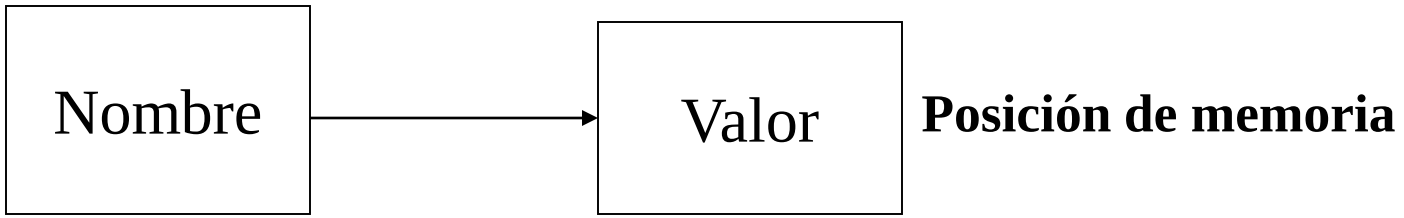
- Variables
  - Representan un valor posible y que puede variar tantas veces como se desea.

Variables

- Son datos cuyo valor asociado o significado puede cambiar durante la ejecución del programa.
- Su cambio de valor no es arbitrario, sino producto de la ejecución de ciertas sentencias en el programa.
- Son posiciones de memoria a las que asignamos un nombre y a través de las cuales podremos almacenar y recuperar datos.

Variables

- Por cada variable se reserva una posición en la memoria donde se aloja su valor corriente.
- Tal posición es solo accedida a través del nombre de la variable.



Variables

- Son introducidas mediante declaraciones de la forma:

```
<variable> = <valor>;
```

- Ejemplo: `x = 3.1416;`

Condiciones

- Los algoritmos con frecuencia presentan situaciones en las que se deben proporcionar acciones alternativas que pueden o no realizarse, dependiendo de los datos de entrada, reflejándose el cumplimiento o no de una determinada condición.

Condiciones

- Diseñar un algoritmo para calcular el salario semanal de un empleado que trabaja por horas.
- La empresa paga la hora trabajada el doble de la hora normal por todas las horas trabajadas mayores a 40.

```
LEER(pago_por_hora)
LEER(horas)
SI horas > 40
ENTONCES
    paga = pago_por_hora * 40 + 2 * pago_por_hora*(horas - 40)
SINO
    paga = pago_por_hora * horas
FIN_SI
```