



Computación 1

Ingeniería Forestal
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 queremos.

Programa Informático

- Es simplemente un conjunto de instrucciones que le indican a la computadora cómo llevar adelante una tarea en particular.
- 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.



1. Como lo explicó el cliente.



2. Como lo entendió el líder de proyecto.



3. Como fue diseñado por el analista.



4. Como fue programado el producto.



5. Como fue descrito por el consultor de negocio.



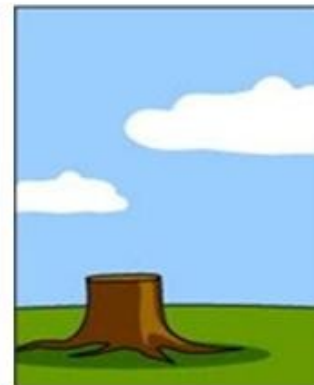
6. Lo documentado del producto y del proyecto.



7. Lo que se instaló para la operación.



8. Lo que se facturó.



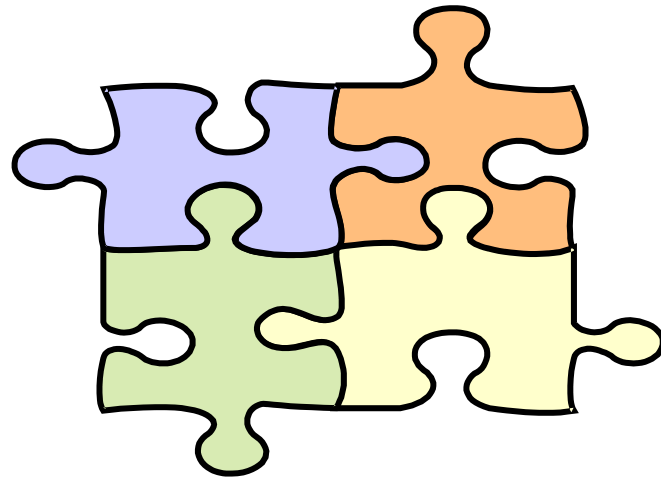
9. El soporte al producto.



10. Lo que en realidad necesitaba el cliente.

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 común divisor entre dos 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

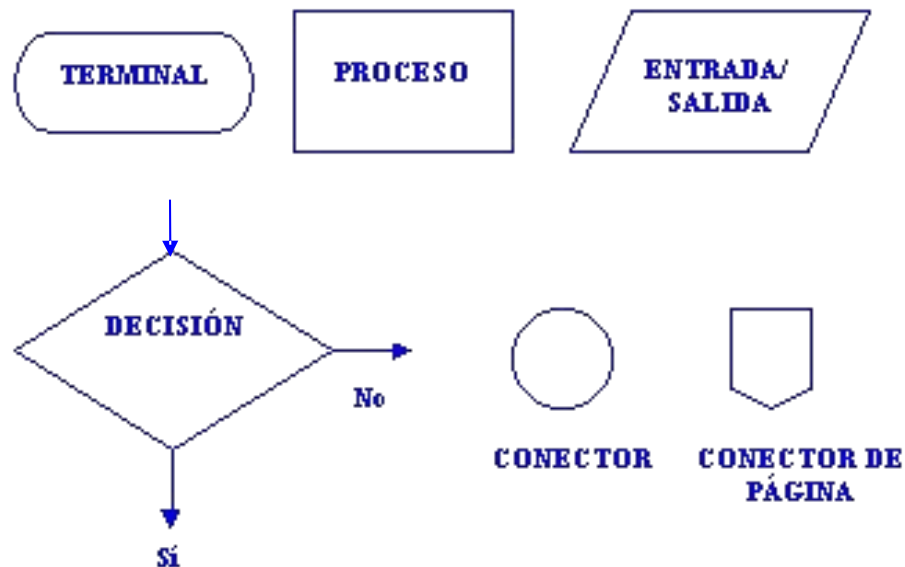
- Un algoritmo se puede expresar de distintas formas
 - 📖 Lenguaje Natural
 - 📖 En forma gráfica, usando diagramas de flujo
 - 📖 Utilizando un pseudo-código.

Representar el algoritmo formalmente

Diagrama de flujos

Escoger una forma conveniente de representar el algoritmo

● Diagrama de flujos de datos

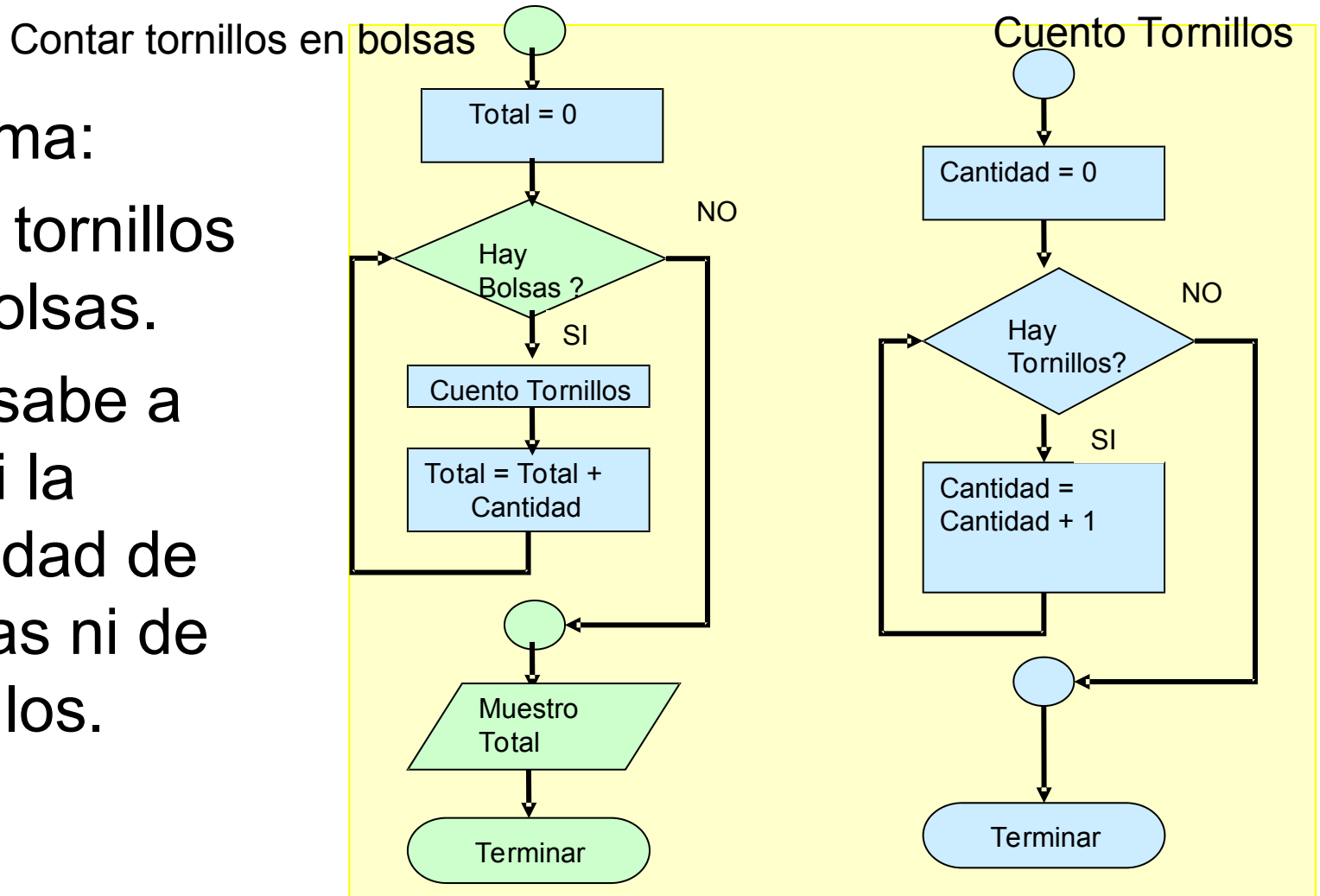


Ejemplo de diagrama de flujo

Problema:

Contar tornillos en bolsas.

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



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 [EN OTRO CASO]	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

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 pseudocódigo.

Codificación

- Escritura de la solución en algún lenguaje de programación.

Construcción de Programas

Compilación

- Traducción del programa escrito en el lenguaje de programación elegido, al lenguaje de máquina.

Ejecución

- Si no hay errores de sintaxis detectados en la Compilación

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