

Algoritmos y Estructuras de Datos

Tecnicatura en Análisis de Sistemas
ISFDYT N° 210 - La Plata 2020



Temas que veremos en esta clase

Conceptos Básicos

Definiciones

- Algoritmo
- Programa
- Computadora
- Especificación

Historia



Algoritmos y Estructuras de Datos I

Objetivo

Aprender a programar en lenguajes imperativos

- Especificar problemas

Es describirlos en un lenguaje formal

Algoritmos y Estructuras de Datos I



- Pensar algoritmos para resolver los problemas

En esta materia nos concentramos en programas para tratamiento de secuencias



Algoritmos y Estructuras de Datos I

- Razonar acerca de estos algoritmos

Para obtener una visión abstracta del cómputo

Poder definir un manejo simbólico y utilizar herramientas para demostrar propiedades de nuestros programas

Probar la corrección de un programa respecto de su especificación

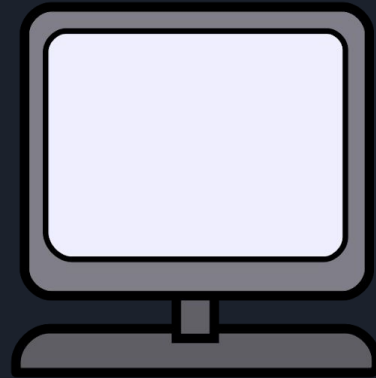
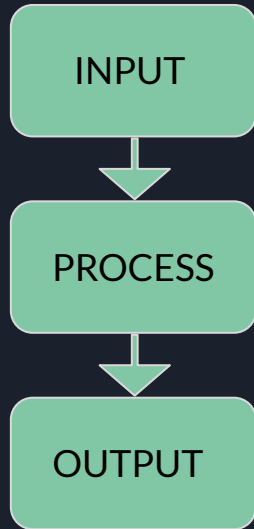
¿Qué es una computadora?



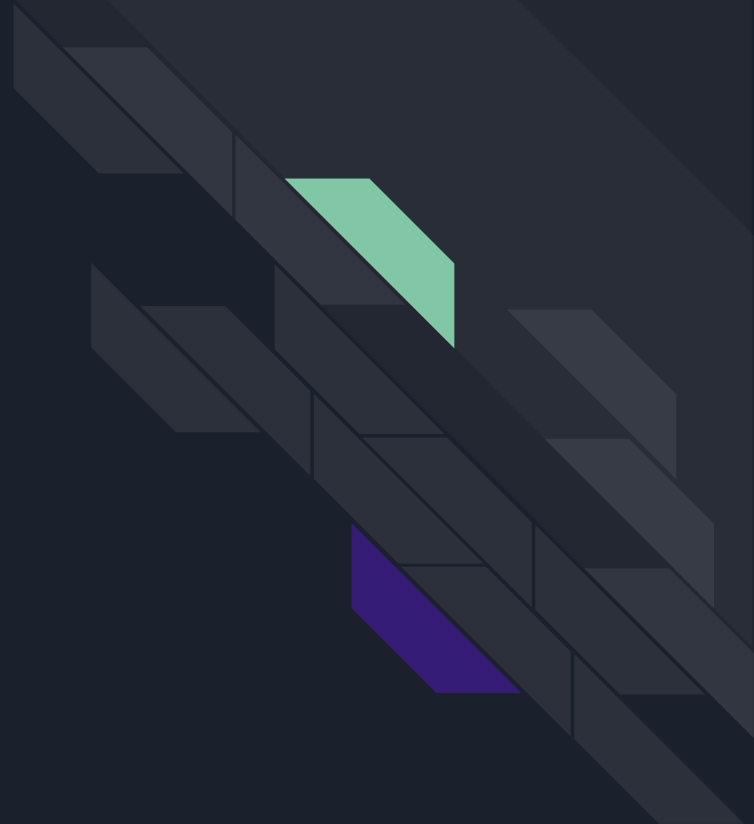
Una computadora es una máquina o autómeta que procesa datos automáticamente de acuerdo con un programa almacenado.

1. Es una máquina o autómeta. [Definición de Autómeta](#)
2. Su función es procesar datos, y estos términos deben entenderse en sentido amplio.
3. El procesamiento se realiza en forma automática.
4. El procesamiento se realiza siguiendo un algoritmo.
5. El algoritmo se convierte en un programa
6. Este programa está almacenado en una memoria interna de la misma computadora.

¿Qué es una computadora?



Algoritmos



Un poco de historia: Al-Juarismi



El origen del término **algoritmo** surgió a mediados del siglo IX cuando el distinguido matemático y astrónomo [Mohammed Ibn Musa-al-Khwarizmi](#) conocido generalmente **Al-Juarismi** desarrolló en su obra "Al-yebr-mugabata" fórmulas para posibilitar que con un número limitado de procesos fuese posible resolver ecuaciones de primer y segundo grado.

Algoritmo se refería originalmente sólo a las reglas de la aritmética con números arábigos. Fue en el siglo XVIII cuando se expandió su significado para abarcar en su definición a toda clase de procedimientos utilizados con el propósito de resolver problemas o realizar determinadas tareas.

Primer Algoritmo: Ada Byron

El primer caso de un algoritmo escrito para una computadora son las notas escritas por [Ada Byron](#), respetada matemática británica que en 1842 escribió un algoritmo para el motor analítico de [Charles Babbage](#).

Es, por esta razón, por lo que se considera a Ada Byron como la primera programadora de la historia. Sin embargo, dado que Babbage nunca terminó su motor analítico, el algoritmo que realizó jamás llegó a implementarse.





¿Qué es un Algoritmo?

Definición

Algoritmo es la especificación rigurosa de la secuencia de pasos (instrucciones) a realizar sobre un autómata para alcanzar un resultado deseado en un tiempo finito.

“Algoritmos, datos y programas” 2001 - Armando E. De Giusti - Prentice Hall

Algoritmos: Otra definición



Podemos entender a un algoritmo como la descripción de los pasos precisos para resolver un problema a partir de datos de entrada.

1. Es la descripción de los pasos a dar
2. Especifica una sucesión de pasos primitivos
3. El objetivo es resolver un problema
4. Un algoritmo típicamente trabaja a partir de datos de entrada
5. Un algoritmo produce información de salida



¿Qué es un programa?

Un **programa** es la descripción de un algoritmo en un lenguaje de programación.

- Corresponde a la implementación concreta de un algoritmo para ser ejecutado en una computadora
- Se describe en un lenguaje de programación

Especificación, algoritmo, programa



1. Especificación:

- ¿Qué problema tenemos?
- Habitualmente, dada en lenguaje formal.
- Es un contrato que da las propiedades de los datos de entrada y las propiedades de la solución.

2. Algoritmo: Descripción de la solución escrita para humanos.

- ¿Cómo resolvemos el problema?

Especificación, algoritmo, programa



3. **Programa:** Descripción de la solución para ser ejecutada en una computadora.

- ¿Cómo resolvemos el problema? Pero escrito en un lenguaje de programación

Especificación





Especificación de problemas

Una especificación es un contrato que define qué se debe resolver y qué propiedades debe tener la solución.

- Define el qué y no el cómo.

La especificación de un problema incluye un conjunto de parámetros: datos de entrada cuyos valores serán conocidos recién al ejecutar el programa.

Además de cumplir un rol “contractual”, la especificación del problema es insumo para las actividades de...

- Pruebas (testing)
- Verificación formal de corrección
- Derivación formal (construir un programa a partir de la especificación).

Parámetros y tipos de datos



La especificación de un problema incluye un conjunto de parámetros: datos de entrada cuyos valores serán conocidos recién al ejecutar el programa.

Cada parámetro tiene un tipo de datos.

Tipo de datos: Conjunto de valores provisto de ciertas operaciones para trabajar con estos valores.

Contratos (1)



Una especificación es un contrato entre el programador de una función y el usuario de esa función.

Ejemplo: calcular la raíz cuadrada de un número real.

¿Cómo es la especificación (informalmente, por ahora) de este problema?

Contratos (2)



Para hacer el cálculo, el programa debe recibir un número no negativo.

- Obligación del usuario: no puede proveer números negativos.
- Derecho del programador: puede suponer que el argumento recibido no es negativo.

El resultado va a ser la raíz cuadrada del número recibido.

- Obligación del programador: debe calcular la raíz, siempre y cuando haya recibido un número no negativo
- Derecho del usuario: puede suponer que el resultado va a ser correcto

Partes de una especificación (contrato)



1. Encabezado

2. Precondición o cláusula “requiere”

- Condición sobre los argumentos, que el programador da por cierta.
- Especifica lo que requiere la función para hacer su tarea.
- Por ejemplo: “el valor de entrada es un real no negativo”

3. Postcondición o cláusula “asegura”

- Condición sobre el resultado, que debe ser cumplida por el programador siempre y cuando el usuario haya cumplido la precondición.
- Especifica lo que la función asegura que se va a cumplir después de llamarla (si se cumplía la precondición).
- Por ejemplo: “la salida es la raíz cuadrada del valor de entrada”



¿Por qué escribir la especificación del problema?

Nos ayuda a entender mejor el problema

Nos ayuda a construir el programa

- Derivación (Automática) de Programas

Nos ayuda a prevenir errores en el programa

- Testing
- Verificación (automática) de Programas

Verificación (Automática) de Programas



Si tengo una especificación formal F , y un programa P que implementa la especificación F , se puede probar matemáticamente (como si fuera un teorema) que P implementa correctamente a F .

- Lo puedo probar manualmente
- Lo puedo probar semi-automáticamente
- Lo puedo probar automáticamente

Pero ¿Por qué no probarlo automáticamente siempre?

Die Entscheidungsproblem (The Halting Problem)



El Halting Problem es el problema de determinar si un programa P termina o no.

El problema de chequear si P implementa correctamente una especificación formal F puede codificarse como un programa

En 1936 Alan Turing demostró que el Halting Problem era indecidible.



Bibliografia

Obligatoria

“Algoritmos, datos y Programas” A.
De Giusti Prentice Hall 2001.

Opcional

“Fundamentos de Programación,
Algoritmos y Estructuras de datos”
Luis Joyanes Aguilar.

“Estructuras de datos en Java” Mark
Allen Weiss.