

Introducción a la Algorítmica y Programación (3300)

Prof. Ariel Ferreira Szpiniak - aferreira@exa.unrc.edu.ar
Departamento de Computación
Facultad de Cs. Exactas, Fco-Qcas y Naturales
Universidad Nacional de Río Cuarto

Teoría 1

Introducción



2017 Lic. Ariel Ferreira Szpiniak

Novedades

Paro Docente en la Universidad

La Asociación Gremial Docente de la UNRC decidió en asamblea que el lunes 6 y el martes 7 de marzo se realice un paro sin asistencia a los lugares de trabajo.

El motivo fue la propuesta realizada por el Gobierno el 23 de febrero:

- Un ajuste mensual en base a la inflación publicada por el INDEC a partir de marzo.
- El Ministerio de Educación y Deportes considera que los salarios docentes no perdieron poder adquisitivo en los últimos tres años.

Actividad: participar de la Marcha educativa junto a todos los sindicatos de la educación el lunes 6 de marzo a partir de las 10hs en la CABA.



2017 Lic. Ariel Ferreira Szpiniak

Novedades

Otros reclamos relacionados a la Educación

Ley de Financiamiento Educativo:

- Paritaria Nacional Docente (art 10).
- Reune al Estado Nacional, a los Estados provinciales y a los gremios de la enseñanza
- Debe fijar el piso salarial docente en todo el país para lo que se denomina el “cargo testigo”.
- Dispone del Fondo de Compensación Salarial Docente financiado por el Gobierno Nacional para ororgar justicia distributiva en todas las Provincias. Sin paritaria nacional el fondo queda sin uso.



2017 Lic. Ariel Ferreira Szpiniak

Novedades

Otros reclamos relacionados a la Educación

- Subejecución presupuestaria 2016:
 - Nacional: casi 20%
 - Ministerio de Educación y Deportes: 12% (12.094 millones).
- Poner un techo a las paritarias: 18% como aumento testigo. La paritaria docente es testigo para los demás gremios.
- Inflación 2016 proyectada en el Presupuesto: 25%, real: más del 40%.
- Inflación 2017 proyectada en el Presupuesto: 20%, real: 25% por lo menos
- Desprestigiar el reclamo:
 - Rompehuelgas: voluntarios docentes ¿?
 - Cuestionar a sindicalistas: caso Sec. Gral. de SUTEBA (Baradel) por no ser docente (demostró que sí lo és). ¿Ministro?

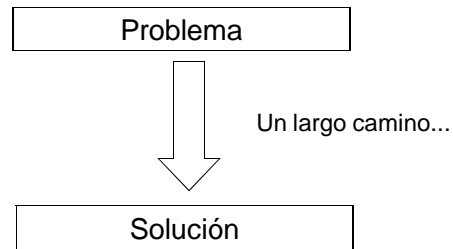


2017 Lic. Ariel Ferreira Szpiniak

Problemas

Solución mediante computadoras

No todos los problemas pueden ser resueltos por una computadora.



2017 Lic. Ariel Ferreira Szpiniak

5

Problemas

Pasos para solucionar un problema

• **Análisis**

El problema debe ser claramente especificado y entendido.

• **Diseño**

Construcción de una solución general del problema.

• **Implementación**

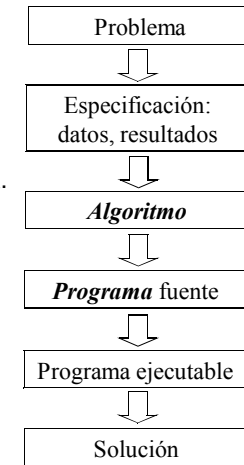
Traducción del algoritmo a un lenguaje de programación de alto nivel.

• **Compilación**

Traducción del programa a un lenguaje entendido por la computadora.

• **Ejecución y Prueba**

Corrida y prueba de funcionamiento del programa en la computadora.



2017 Lic. Ariel Ferreira Szpiniak

6

Pasos para solucionar un problema

Análisis

- El problema debe estar bien definido para poder obtener una solución satisfactoria.
- Los datos de entrada y los resultados deben ser precisamente descriptos.

Preguntas concernientes a la entrada:

- ¿Cuáles y cuántos son los valores de entrada?
- ¿Cómo se llaman esos datos?
- ¿Cuáles son valores válidos de entrada?
- Dibujo o esquema que permita entender mejor el problema (optativo)

Preguntas concernientes a los resultados (salida):

- ¿Cuáles y cuántos son los valores del resultado?
- ¿Cómo se llaman esos datos?
- ¿Cuáles son valores válidos del resultado?

Relación entre datos de entrada y resultados:

(fórmulas, ecuaciones, etc.)



2017 Lic. Ariel Ferreira Szpiniak

7

Pasos para solucionar un problema

Análisis (cont.)

Ejemplo de Problema: Calcule el área de un cuadrado

Preguntas concernientes a la entrada:

- ¿Cuáles y cuántos son los valores de entrada? Lado del cuadrado. Uno
- ¿Cómo se llaman esos datos? lado
- ¿Cuáles son valores válidos de entrada? Números enteros positivos

Preguntas concernientes a los resultados (salida):

- ¿Cuáles y cuántos son los valores del resultado? Área del cuadrado. Uno
- ¿Cómo se llaman esos datos? areaCuad
- ¿Cuáles son valores válidos del resultado? Números enteros positivos

Relación entre datos de entrada y resultados: $\text{areaCuad} = \text{lado} * \text{lado}$

- **Dato/s:** lado del cuadrado es número entero positivo (lado)
- **Resultado/s:** área del cuadrado es número entero positivo (areaCuad)
- **Relación entre datos de entrada y resultados:** $\text{area} = \text{lado} * \text{lado}$



2017 Lic. Ariel Ferreira Szpiniak

8

Problemas

Pasos para solucionar un problema

• Análisis

El problema debe ser claramente especificado y entendido.

• Diseño

Construcción de una solución general del problema.

• Implementación

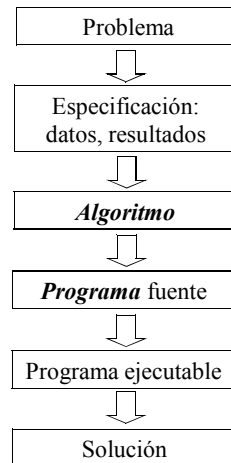
Traducción del algoritmo a un lenguaje de programación de alto nivel.

• Compilación

Traducción del programa a un lenguaje entendido por la computadora.

• Ejecución y Prueba

Corrida y prueba de funcionamiento del programa en la computadora.



Pasos para solucionar un problema

Diseño

Cuando los problemas adquieren cierta complejidad puede ser visto como la composición de varios (sub)problemas de menor complejidad.

Subproblemas:

- (1) lectura de datos
- (2) cálculo del área
- (3) exhibir resultados

Nota: en este caso la complejidad del problema no justifica los subproblemas. Es solo a modo de ejemplo.



Noción de Algoritmo

Acción, Entorno y Procesador

Consideremos los siguientes enunciados:

E1: 1/2 docena de huevos revueltos

- a) Romper seis huevos en un plato;
- b) Batir la clara y la yema con un tenedor;
- c) Calentar el aceite en una sartén al fuego;
- d) Cuando el aceite esté caliente, verter el contenido del plato;
- e) Sacar la sartén del fuego cuando el revuelto esté cocido.

E2: Cálculo de la media de dos números con una calculadora

- a) pulsar C ;
- b) teclear el primer número;
- c) pulsar + ;
- d) teclear el segundo número;
- e) pulsar % ;
- f) teclear 2;
- g) pulsar = . {Aparece el resultado}



Noción de Algoritmo

Acciones primitivas y descomposición

En los enunciados E1 y E2 hemos supuesto que el procesador sabía ejecutar las acciones enumeradas. En ese caso las acciones se denominan **primitivas**.

Definición: para un procesador dado, una acción es **primitiva** si el enunciado de dicha acción es suficiente para poder ejecutarla sin información suplementaria.

Si una acción no es primitiva debe ser descompuesta en dos o más acciones primitivas.

Definición: **descomponer** una acción -no primitiva- es encontrar una serie de **acciones primitivas** que realicen lo requerido por dicha acción.



Noción de Algoritmo

Ejemplo de descomposición

Supongamos que en el enunciado E1, el procesador es un niño y no comprende la acción “a) romper seis huevos en un plato”. Es necesario descomponer la acción.

a) romper seis huevos en un plato;

- | | |
|--|---|
| a1) poner seis huevos en la superficie de trabajo; | a11) tomar otro huevo de la superficie de trabajo; |
| a2) tomar un huevo de la superficie de trabajo; | a12) romperlo y verter su contenido en el plato; |
| a3) romperlo y verter su contenido en el plato; | a13) tirar las cáscaras a la basura; |
| a4) tirar las cáscaras a la basura; | a14) tomar otro huevo de la superficie de trabajo; |
| a5) tomar otro huevo de la superficie de trabajo; | a15) romperlo y verter su contenido en el plato; |
| a6) romperlo y verter su contenido en el plato; | a16) tirar las cáscaras a la basura; |
| a7) tirar las cáscaras a la basura; | a17) tomar el último huevo de la superficie de trabajo; |
| a8) tomar otro huevo de la superficie de trabajo; | a18) romperlo y verter su contenido en el plato; |
| a9) romperlo y verter su contenido en el plato; | a19) tirar las cáscaras a la basura; |
| a10) tirar las cáscaras a la basura; | |



2017 Lic. Ariel Ferreira Szpiniak

13

Noción de Algoritmo

Procesador

Definición: un procesador es la entidad responsable de ejecutar las **acciones primitivas**.

El procesador solo entiende las acciones primitivas, nada más.

El procesador puede ser una máquina, una persona, etc.



2017 Lic. Ariel Ferreira Szpiniak

14

Noción de Algoritmo

Entorno de trabajo, variable y tipo

Entorno de trabajo

Definición: el entorno de trabajo es el espacio donde conviven los distintos utensilios que pueden ser usados por el procesador para ejecutar las **acciones primitivas**. El procesador modifica el entorno de trabajo mediante la ejecución de las acciones.

Dato

Definición: un dato es la **representación** de un **objeto del mundo real** mediante el cual podemos modelar aspectos del problema que se quiere resolver. Son los **valores** de información de los que se necesita **disponer** y en ocasiones **transformar**.



2017 Lic. Ariel Ferreira Szpiniak

15

Noción de Algoritmo

Entorno de trabajo, variable y tipo

Variable

Definición: son **datos** que tienen la **posibilidad de cambiar** su **valor** durante la ejecución del programa. Ejemplo: saldo

Constante

Definición: son **datos** que **no pueden cambiar** su **valor** durante la ejecución del programa. Ejemplo: Iva = 21

Tipo

Definición: es un **conjunto de valores posibles** que se encuentran ligados a un conjunto de operaciones para crearlos y manipularlos. Todo **dato**, tanto **constante** como **variable**, debe **pertenecer** a un **tipo**. Ejemplo: saldo \in Entero



2017 Lic. Ariel Ferreira Szpiniak

16

Algunos Tipos Simples

- **Entero:** es el conjunto de valores numéricos más simple de todos: ..., -2, -1, 0, 1, 2, ...

Operaciones: +, -, *, /, div, mod, =, <>, <, <=, >, >=

- **Lógico:** conjunto de valores lógicos: **Verdadero** y **Falso**.

Operaciones: y, o, no

- **Carácter:** conjunto de caracteres: letras minúsculas, mayúsculas, cifras, y signos especiales. 'M' 'm' '5' '%'

Operaciones: =, <>, >, <, >=, <=, ord, chr

- **Cadena:** conjunto de cadenas de caracteres: "promNotas" "5mentarios".

Operaciones: =, <>, >, <, >=, <=, +



Noción de Algoritmo Significado de la asignación

- El objetivo de la *asignación* es cambiar el valor almacenado en una **variable**.

$x \leftarrow e$ Guarda en **x** el valor de **e**

- Sintaxis: **<variable> \leftarrow <expresión>**

- Ejemplos:

Sean $i, j \in \text{Entero}$

$i \leftarrow 9$
 $j \leftarrow 3 + 4$

Deben coincidir los tipos de la variable y la expresión

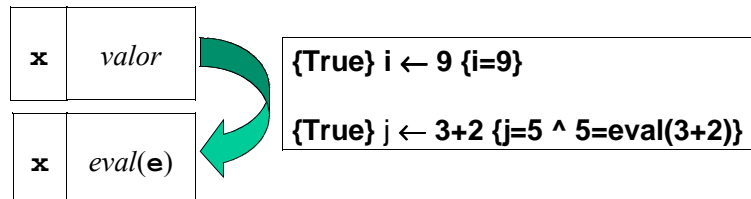


Noción de Algoritmo Significado de la asignación

Una asignación $x \leftarrow e$ es ejecutada siguiendo estos pasos:

1. Se evalúa la expresión **e**
2. Se reemplaza el valor almacenado en la variable **x**, por el valor de **e**.

$\{\text{True}\} x \leftarrow e \{x = X_0 \wedge X_0 = \text{eval}(e)\}$



Noción de Algoritmo Significado de la asignación

- Al producirse la asignación el valor anterior se pierde.
- La ocurrencia de una variable en el lado derecho de una asignación denota su *valor* actual.
- Una misma variable puede aparecer en la parte izquierda y derecha de una asignación.

Por ejemplo: $x \leftarrow x + 1$

Esto **NO** debe interpretarse como una ecuación matemática! Sólo significa que estamos usando el valor *actual* de la variable **x** para calcular su *nuevo* valor.



Algoritmo Definición

Dado un **procesador**, un **entorno de trabajo** y un tratamiento a ejecutar por dicho procesador sobre ese entorno, un **algoritmo** es el enunciado de una **secuencia finita de acciones** que realizan este tratamiento.

Otra definición: Un algoritmo es una sucesión finita de instrucciones o pasos no ambiguos que se pueden ejecutar en un tiempo finito para resolver un problema.



Algoritmo Definición (cont.)

Algoritmo: del árabe al-Jwarizmi, matemático del siglo IX

Características:

- Debe ser preciso e indicar el orden de realización de cada paso.
- Se debe obtener el mismo resultado cada vez que se aplica a los mismo datos.
- Se debe terminar en algún momento.

En la vida cotidiana empleamos algoritmos en multitud de ocasiones. También existen ejemplos de índole matemática (algoritmo de la división, Euclides, Gauss, Valor Medio, etc.)



Algoritmo Estructura y Notación

Algoritmo <nombre>

Léxico

<declaración de variables, tipos,
acciones, funciones, etc>

Entorno de trabajo

Inicio

<secuencia de acciones>

Fin



Notación Algorítmica Acciones conocidas por el Procesador

nada No produce cambio alguno

x ← e Guarda en **x** el valor de **e**

Leer(x) Guarda en **x** el valor ingresado por teclado

Leer(w,z) Guarda en **w** el primer valor ingresado por teclado
y en **z** el segundo (se puede extender a más variables)

Escribir(e) Muestra por pantalla el valor de **e**. Donde **e** pueden
ser variables, texto, o variables y texto conjuntamente.



Noción de Algoritmo

Ejemplo

Algoritmo SumarDosNumeros

Léxico

$x, y, a \in \mathbb{Z}$ {números enteros}

Inicio

$x \leftarrow 15$

$y \leftarrow 2$

$a \leftarrow x+y$

Escribir(a)

Fin

comentario



2017 Lic. Ariel Ferreira Szpiniak

25

Pasos para solucionar un problema

Diseño (cont.)

Algoritmo AreaCuadrado

Léxico

$lado \in \mathbb{Z}^+$ {variable dato}

$areaCua \in \mathbb{Z}^+$ {variable resultado}

Inicio

Leer(lado)

$areaCua \leftarrow lado * lado$

Escribir('El area es: ', areaCua)

Fin

Nota: Observar la similitud entre este algoritmo y la descripción refinada de la solución del problema.

• Escribir('El area es: ', areaCua, ' metros cuadrados')
• Escribir('El area de un cuadrado de ', lado, ' metros de lado ', 'es: ', areaCua, ' metros cuadrados')



2017 Lic. Ariel Ferreira Szpiniak

26

Pasos para solucionar un problema

Implementación

Traducción del algoritmo a Pascal usando un editor de texto o un ambiente de desarrollo.

PROGRAM AreaCuadrado;

VAR

lado: Integer; {variable dato}

areaCua: Integer; {variable resultado}

BEGIN

Read(lado);

areaCua := lado * lado;

Writeln('El area es: ', areaCua)

END.

Writeln('El area es: ', areaCua, ' metros ')

Writeln('El area de un cuadrado de ', lado, ' metros de lado ', 'es: ', areaCua, ' metros cuadrados ')

Pascal posee el tipo Integer (entero), no permite diferenciar entre positivos y negativos.



2017 Lic. Ariel Ferreira Szpiniak

27

Pasos para solucionar un problema

Compilación y Ejecución

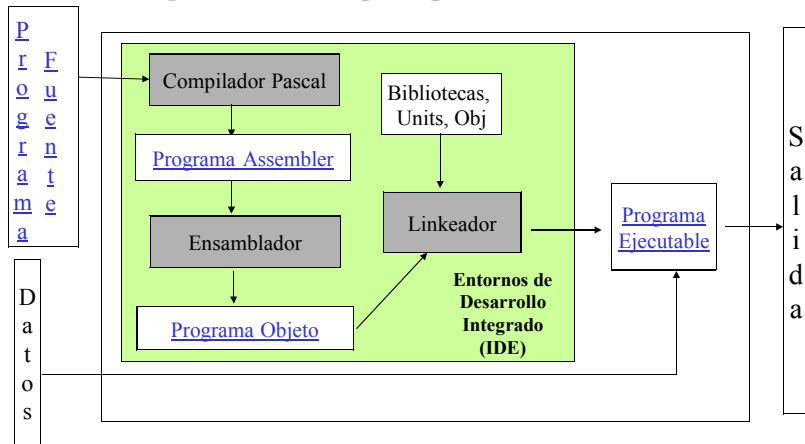
- Compilar el programa construido.
 - Detección de errores (sintácticos, ...).
- Ejecutar el resultado de la compilación.
 - Procesamiento de datos de entrada.
 - Detección de errores semánticos.
- Testeo
 - Probar el programa con una serie de valores de entrada y verificar que produce el resultado esperado en todos los casos.



2017 Lic. Ariel Ferreira Szpiniak

28

Pasos para solucionar un problema Compilación y Ejecución (cont.)



2017 Lic. Ariel Ferreira Szpiniak

29

Problemas Pasos para solucionar un problema

• Análisis

El problema debe ser claramente especificado y entendido.

• Diseño

Construcción de una solución general del problema.

• Implementación

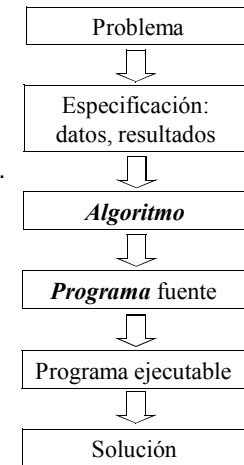
Traducción del algoritmo a un lenguaje de programación de alto nivel.

• Compilación

Traducción del programa a un lenguaje entendido por la computadora.

• Ejecución y Prueba

Corrida y prueba de funcionamiento del programa en la computadora.



2017 Lic. Ariel Ferreira Szpiniak

30

Computadora

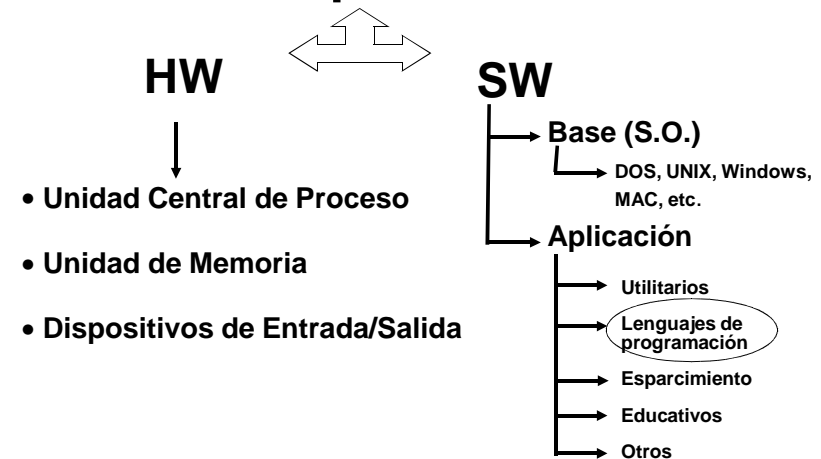
Definición: Máquina capaz de aceptar datos a través de un medio de entrada, procesarlos automáticamente bajo el control de un programa y proporcionar la información resultante a través de un medio de salida.



2017 Lic. Ariel Ferreira Szpiniak

31

Componentes



2017 Lic. Ariel Ferreira Szpiniak

32

HW Unidad Central de Proceso (UCP/CPU)

Controla el procesamiento de la información

Unidad de Control:

- Carga instrucciones en memoria
- Interpreta y
- Devuelve el resultado de la ejecución

Unidad Aritmética y Lógica:

- Proceso de operaciones aritméticas y lógicas
- Provee decisión a la Unidad de Control



2017 Lic. Ariel Ferreira Szpiniak

33

HW Unidad de Memoria

Memoria principal (o primaria):

- Conjunto de celdas de memoria direccionables vía un único nombre
- Acceso directo por referencia para:
 - Carga
 - Recuperación de información
- Programas residen en este tipo de memoria al ser ejecutados

Memoria secundaria:

- Permiten almacenar gran cantidad de información
- Información persistente
- Ejemplos: Discos duros, diskettes, CDRom, ZIP.



2017 Lic. Ariel Ferreira Szpiniak

34

HW Dispositivos de Entrada/Salida (Input/Output)

Los dispositivos de E/S permiten la comunicación ente el usuario y el computador.

Dispositivos de entrada:

- Teclados
- Ratón
- Lectores de disco
- etc.

Dispositivos de entrada/salida:

- Placa de red
- Modem
- Placa de sonido
- etc.

Dispositivos de salida:

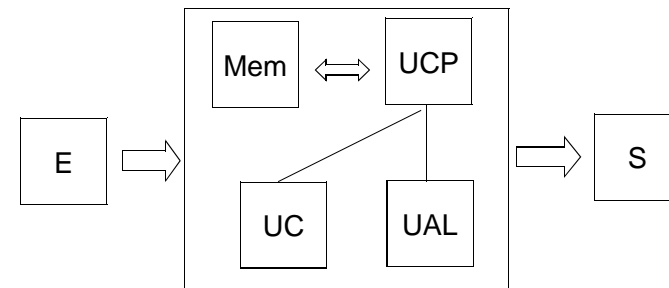
- Impresoras
- Pantalla
- etc.



2017 Lic. Ariel Ferreira Szpiniak

35

HW Modelo Refinado de Computadora



2017 Lic. Ariel Ferreira Szpiniak

36

SW

Lenguajes de Programación

Tipos de instrucciones ejecutadas por una computadora:

- de E/S: lectura/escritura de información
- Lógico-aritméticas
- Secuencia: Primero ejecutar una instrucción y luego otra
- Selección: si ... entonces ... sino ...
- Ciclo: Repetición de una secuencia de instrucciones
- Procedimiento: Grupo de instrucciones que pueden ser referenciadas y ejecutadas



2017 Lic. Ariel Ferreira Szpiniak

37

SW

Instrucciones para la computadora

- Contenido de memoria expresado en *bits* (dígitos binarios)
- Datos e instrucciones en el mismo lenguaje
- El lenguaje de máquina depende del diseño y del hardware del computador (por ej., diferentes máquinas pueden representar las instrucciones con códigos diferentes)



2017 Lic. Ariel Ferreira Szpiniak

38

SW

Lenguajes de bajo nivel

Lenguajes con un pobre nivel de abstracción, en el sentido de que sus instrucciones se asemejan mucho a las de máquina

Ejemplo: **Lenguaje de Ensamblado (Assembler)**

- En este tipo de lenguajes las instrucciones son nombradas por códigos mnemotécnicos. Por ejemplo: ADD, SUB, MPY, DIV, etc.
- Las instrucciones son traducidas a código de máquina mediante un *ensamblador*.



2017 Lic. Ariel Ferreira Szpiniak

39

SW

Ejemplo de lenguaje Assembler

```
...
leal  U_SYSWIN32_OUTPUT,%edi
movl  %edi,-4(%ebp)
pushl  $.L7
pushl  -4(%ebp)
pushl  $0
call  FPC_WRITE_TEXT_SHORTSTR
pushl  -4(%ebp)
call  FPC_WRITELN_END
pushl  $.L4
call  FPC_IOCHECK
...
```

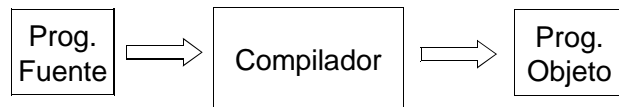


2017 Lic. Ariel Ferreira Szpiniak

40

SW Lenguajes de alto nivel

- Lenguajes con un mayor nivel de abstracción, en el sentido de que sus instrucciones asemejan al lenguaje natural.
- Pueden ser independientes de la máquina. Por lo tanto, pueden ser traducidos a distintos lenguajes de máquina.



Ejemplos

Pascal, C, C++, Java, Visual BASIC, COBOL, Fortran.



2017 Lic. Ariel Ferreira Szpiniak

41

SW Pascal

- Lenguaje de alto nivel que usaremos en la materia.
- Fue diseñado y desarrollado por N. Wirth alrededor del 1968. Estuvo orientado a la enseñanza de la programación estructurada.
- Este lenguaje permite expresar principios de programación y de diseño de solución de problemas en forma *abstracta* y *estructurada*.
- Una instrucción Pascal equivale a varias instrucciones de lenguaje de máquina.



2017 Lic. Ariel Ferreira Szpiniak

42

SW C

- Lenguaje de alto nivel que usaremos en la materia.
- Fue creado en 1972 por Dennis Ritchie, en los Laboratorios Bell, como evolución del lenguaje B.
- En 1983 el Instituto de Estándares Americanos estableció un estándar que definiera al lenguaje C, conocido como ANSI C.
- Los principales compiladores de C llevan implementado el estándar ANSI C.
- Utilizado para la implementación del sistema UNIX, y muchas aplicaciones complejas.



2017 Lic. Ariel Ferreira Szpiniak

43

SW Ejemplo de un programa Pascal

Este programa muestra por pantalla el mensaje “Hola mundo”.

```
PROGRAM HolaMundo;  
BEGIN  
    Writeln('Hola mundo')  
END.
```



2017 Lic. Ariel Ferreira Szpiniak

44

SW

Ejemplo de un programa C

Este programa muestra por pantalla el mensaje "Hola mundo".

```
#include <stdio.h>

int main()
{
    printf("Hola mundo");
    return 0;
}
```



2017 Lic. Ariel Ferreira Szpiniak

45

SW

Características de un buen programa

- Confiabilidad
- Adaptabilidad
- Legibilidad



2017 Lic. Ariel Ferreira Szpiniak

46

SW

Fracasos - Crisis del software

- **Sistema de Control de Tráfico Aéreo:** FAA-IBMcancelado: 144 M\$ - resto: atraso 5 años, 1.400 M\$
- **Sonda Mariner 1** (Venus) fallo: error de software
- **Taxi espacial:** 5 computadoras redundantes: demora 2 días ('81), un día ('85), pierde Intelsat 6 ('92),...
- **Sistema telefónico de AT&T:** corte de 1 día, error de tipos
- **Sistema de trenes alemán:** bloqueo de 1 día por Memoria llena
- **American Airlines+Marriott+Hilton+Budget1992:** Integración de reservas, abandonado, 165 M\$
- **6 fantasmas en el radar:** aeropuerto de San Francisco (EE.UU.) 9/1/01.
- **Tren noruego se detuvo el 31 Dic 2000:** el 2000 tenía 54 semanas en vez de 53.
- **Cohete ucraniano Tsiklon-3:** motores del cohete se apagaron a 367s. del despegue, 6 satélites (Ene/01)



2017 Lic. Ariel Ferreira Szpiniak

47

Bibliografía

- **Tutoriales, manuales y compilador Pascal:** en el sitio de la materia, Repositorio de archivos, carpeta Pascal.
- **introturbopascal.pdf** (Introducción a la noción de procesador, algoritmos, lenguajes y metodo de solución de problemas. Notación Pascal. No respeta nuestro estilo de programación)
- Scholl, P. y Peyrin, J.-P. "Esquemas Algorítmicos Fundamentales: Secuencias e iteración":
 - Introducción, Algoritmo, Léxico, Notación algorítmica (pags. 1 - 34)
 - Composición secuencial (pags. 35 - 55)
- Biondi, J. y Clavel, G. "Introducción a la Programación. Tomo 1: Algorítmica y Lenguajes":
 - Notación algorítmica (pags. 1 – 12)
 - Entorno, Tipos, Variables, Constantes, Notación algorítmica (pags. 13 – 34)
 - Composición condicional (35 - -53)
 - Procesadores, Lenguajes (pags. 127 – 140)
 - Introducción a Lógica (pags. 203 – 204)
 - Pascal (pags. 243 - 252)
- Wirth, N. "Algoritmos + Estruturas de Datos = Programas": Presentación y Prólogo muy interesantes. Tipos (pags. 1 – 12).
- Quetglás, Toledo, Cerverón. "Fundamentos de Informática y Programación"
 - <http://robotica.uv.es/Libro/Indice.html>
 - Capítulos 1 y 2



2017 Lic. Ariel Ferreira Szpiniak

2017 Lic. Ariel Ferreira Szpiniak

Características de la asignatura

▪ **Carreras:** Analista en Computación – Profesorado en Ciencias de la Computación – Licenciatura en Ciencias de la Computación.

▪ **Régimen de regularidad:**

- 80% de asistencia a las clases teóricas en aula común en cada cuatrimestre.
- 80% de asistencia a las clases prácticas en aula común en cada cuatrimestre.
- 80% de asistencia a las clases de laboratorio en cada cuatrimestre.
- Presentación de 1 ejercicio resuelto en Lenguaje Pascal en los trabajos prácticos en que así se indique.
- Un trabajo evaluable individual por parcial, realizado en el Laboratorio, que aportará un 10% de la valoración total del parcial respectivo.
- Aprobar cuatro (4) exámenes parciales, dos en primer cuatrimestre y dos en el segundo. Habrá cuatro recuperatorios, uno por cada parcial.
- Aprobar un trabajo integrador en lenguaje C a mitad de año, y un proyecto a fin de año.

▪ **Asignación de horas semanales:**

Teóricos: 4hs. Consulta: 2hs.

Prácticos: Aula común: 4hs. Consulta: 2hs. **Laboratorio:** 2hs. Consulta: 2hs.

▪ **Exámenes parciales:** escritos e individuales. Dos exámenes parciales durante el primer cuatrimestre y dos durante el segundo. Fechas de cada examen: a definir antes del 29/03.

▪ **Examen final:** individual. Oral.



2017 Lic. Ariel Ferreira Szpiniak

53

Horarios

Teóricos

ARIEL FERREIRA SZPINIAK (aferreira@exa.unrc.edu.ar):

Lunes de 16 a 18hs, Jueves de 16 a 18hs.

Prácticos

Mañana

Comisión 1 -

Aula común: ROJO, GUILLERMO (grojo@exa.unrc.edu.ar):

Lunes de 10 a 12hs, Miércoles de 10 a 12hs.

Laboratorio: BENGOLEA, VALERIA (valebengolea@gmail.com):

Miércoles de 8 a 10hs.

Comisión 2 -

Aula común: LUIS CHAVEZ (lchavez@dc.exa.unrc.edu.ar)

Lunes de 10 a 12hs, Miércoles de 8 a 10hs.

Laboratorio: BENGOLEA, VALERIA (valebengolea@gmail.com):

Miércoles de 10 a 12hs.

Tarde

Comisión 3 -

Aula común: JORGE GUAZZONE (jguazzone@exa.unrc.edu.ar):

Lunes de 18 a 20hs, Miércoles de 16 a 18hs.

Laboratorio: BENGOLEA, VALERIA (valebengolea@gmail.com)

LUIS CHAVEZ (lchavez@dc.exa.unrc.edu.ar):

Miércoles de 18 a 20hs.

Laboratorio: Aula
101 del Pabellón 2



2017 Lic. Ariel Ferreira Szpiniak

54

Aulas

<https://sisinfo.unrc.edu.ar/bedepub/>

Día	Horario	Asignatura	Docente a cargo	Aula /Pabellón
Lunes	10 a 12	C1 – Práctico	Guillermo Rojo	
Lunes	10 a 12	C2 – Práctico	Luis Chávez	
Lunes	16 a 18	Teórico	Ariel Ferreira Szpiniak	
Lunes	18 a 20	C3 – Práctico	Jorge Guazzone	
Miércoles	08 a 10	C1 – Laboratorio	Valeria Bengolea	
Miércoles	08 a 10	C2 – Práctico	Luis Chávez	
Miércoles	10 a 12	C1 – Práctico	Guillermo Rojo	
Miércoles	10 a 12	C2 – Laboratorio	Valeria Bengolea	
Miércoles	16 a 18	C3 – Práctico	Jorge Guazzone	
Miércoles	18 a 20	C3 – Laboratorio	Luis Chávez	
Jueves	16 a 18	Teórico	Ariel Ferreira Szpiniak	



2017 Lic. Ariel Ferreira Szpiniak

55

Objetivos

- Capacidad para encontrar soluciones informáticas a problemas mediante la modelización disciplinada de soluciones y descomposición en módulos.
- Capacidad para traducir eficientemente algoritmos al lenguaje de programación Pascal y C.
- Habilidad en el uso del lenguaje Pascal y C.
- Habilidad en el uso de buenos hábitos de programación.
- Capacidad para documentar técnicamente los programas desarrollados (análisis, diseño, implementación, prueba, manuales para el usuario, etc).
- Habilidad en el uso de herramientas básicas de desarrollo de software.
- Capacidad para trabajar en grupo.



2017 Lic. Ariel Ferreira Szpiniak

56

Bibliografía

- ★ Scholl, P. y Peyrin, J.-P. "Esquemas Algorítmicos Fundamentales: Secuencias e iteración", Barcelona, Ed. Masson, 1991.
- ★ Biondi, J. y Clavel, G. "Introducción a la Programación. Tomo 1: Algorítmica y Lenguajes", 2ª ed., Barcelona: Masson, 1985.
- ★ Clavel, G. y Biondi, J. "Introducción a la Programación. Tomo 2: Estructuras de Datos", 2ª ed., Barcelona: Masson, 1985.
- De Guisti, A. "Algoritmos, datos y programas. Con aplicaciones en Pascal, Delphi y Visual Da Vinci. Prentice Hall.
- Wirth, N. "Algoritmos + Estructuras de Datos = Programas". Ediciones del Castillo, 1980.

Bibliografía para la implementación en Pascal

- ★ Joyanes Aguilar, L. "Programación en Turbo Pascal". Mc Graw Hill.
- Grogono, P., "Programación en Pascal", Wilmington, Addison-Wesley, 1996.
- Wirth, N. and K. Jensen, "Pascal: User Manual and Report", 4ª ed., New York, Springer-Verlag, 1991 (traducción de la primera edición "Pascal: Manual del Usuario e Informe", Buenos Aires, El Ateneo, 1984).

Bibliografía Secundaria

- Lucas, M., J.-P. Peyrin y P. Scholl, "Algorítmica y Representación de Datos. Tomo 1: Secuencia, Automata de estados finitos", Barcelona, Ed. Masson, 1985.
- Aho, A., J. Hopcroft and J. Ullman, "Data Structures and Algorithms", Reading MA, Addison-Wesley, 1987 (traducción al castellano "Estructuras de Datos y Algoritmos", Addison-Wesley, 1988).



2017 Lic. Ariel Ferreira Szpiniak

57

Aula virtual

Utilizaremos un aula virtual para publicar:

- **Novedades generales**
- **Materiales digitales utilizados para desarrollar las clases teóricas, prácticas y proyectos.**
- **Entrega de trabajos prácticos**
- **Calendario de actividades**
- **Notas de parciales**
- **Foros de consulta**

Sitios Web de la asignatura (Campus Virtual SIAT):

<http://www.siat.unrc.edu.ar>

- **Registrarse en el Campus Virtual SIAT**
- **Ingresar al Campus e inscribirse en la materia: desde el menú principal del Campus, opción Inscripción On Line (Alumno)**
- **Clave de inscripción: 2017**



2017 Lic. Ariel Ferreira Szpiniak

58

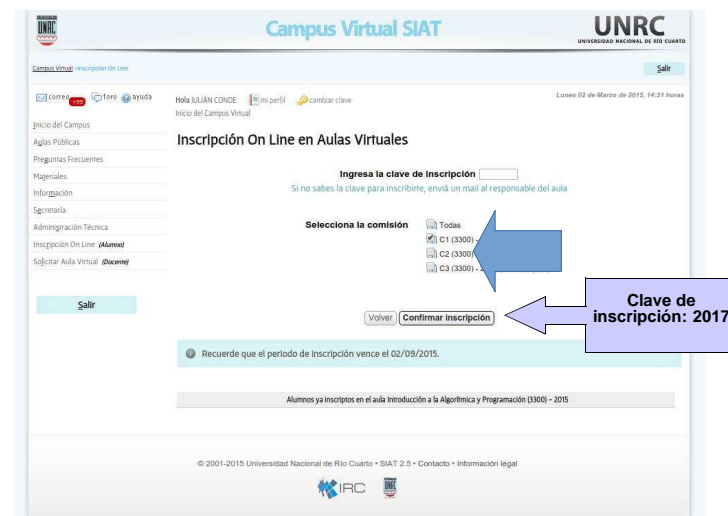
Aula virtual



2017 Lic. Ariel Ferreira Szpiniak

59

Aula virtual



2017 Lic. Ariel Ferreira Szpiniak

60

Citar/Atribuir: Ferreira, Szpiniak, A. (2017). Teoría 1: Introducción. Introducción a la Algorítmica y Programación (3300). Departamento de Computación. Facultad de Cs. Exactas, Fco-Qcas y Naturales. Universidad Nacional de Río Cuarto.

Usted es libre para:

Compartir: copiar y redistribuir el material en cualquier medio o formato.

Adaptar: remezclar, transformar y crear a partir del material.

El licenciente no puede revocar estas libertades en tanto usted siga los términos de la licencia.

Bajo los siguientes términos:



Atribución: Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciente.



Compartir Igual: Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted podrá distribuir su contribución siempre que utilice la misma licencia que la obra original.

<https://creativecommons.org/licenses/by-sa/2.5/ar/>

