

PROGRAMACIÓN I

AÑO 2025

Temas

- ✓ Presentación de la materia
- ✓ ¿Qué es la Informática?
- ✓ Etapas para la resolución de un problema con computadora
- ✓ ¿Qué es un Dato?
- ✓ Variables y Constantes
- ✓ Algoritmos + Datos = Programas
- ✓ Tipos de Datos: definición, clasificaciones y ejemplos
- ✓ Estructura de un programa en Pascal
- ✓ Clasificación de Lenguajes

Nos presentamos



Docentes

Teoría: María Virginia Ainchil

Práctica: JTP: César Estrebou

MODALIDAD de trabajo

CRONOGRAMA

Presentación de contenidos

Explicaciones
teóricas

Material de
estudio en IDEAS
(videoclases,
presentaciones, y
otros)
ASINCRÓNICO

- *Miércoles y viernes de 8 a 10 hs.*

TEORÍAS

Guías para realizar práctica

Clases de
práctica

Material de
estudio en IDEAS
ASINCRÓNICO

- *Lunes y miércoles de 10 a 12 hs*

PRÁCTICA

Objetivos



Programación 1 - Objetivos

- ✓ Analizar problemas resolubles con computadora, poniendo énfasis en la modelización, abstracción de funciones y en la descomposición funcional. Obtener una expresión sintética y precisa de los problemas, con una documentación de una metodología de trabajo por el alumno.
- ✓ Lograr una expresión simbólica, implementación y evaluación de algoritmos, orientando los mismos a la resolución de las partes (módulos) en que se descomponen los problemas, a partir de un paradigma procedural/imperativo.
- ✓ Comprender las nociones de estructuras de datos, tipos de datos y abstracción de datos.

Programación 1 - Objetivos

- ✓ Comprender las nociones vinculadas a la arquitectura y organización de la computadora. (PLAN DE ESTUDIOS PREVIO)
- ✓ Reconocer los conceptos de corrección y eficiencia de algoritmos.
- ✓ Combinar los elementos mencionados anteriormente a fin de completar el ciclo del problema a su solución con computadora, analizando simultáneamente algoritmos y datos.

Reglamento



Programación 1 - Para aprobar la cursada



Programación 1 - Aprobación

APROBACIÓN DE LA MATERIA

✓ EXAMEN FINAL

- Se toma todos los meses y ustedes deben anotarse según el calendario académico.
- Programación I toma los martes horario a considerar
- Info en el blog de la cátedra/cartelera virtual

✓ RÉGIMEN DE PROMOCIÓN

OPCIÓN A

- 50% asistencia a clases teóricas
- Aprobar el parcial en primera o segunda fecha
- 1 evaluación de promoción (en caso de plan viejo promedia con lo que obtienen en la promoción con el Prof. Cristina)

OPCIÓN B

- 50% asistencia a clases teóricas
- Aprobar el parcial en primera fecha
- Trabajo práctico con aplicación de temas de teoría con nota de 6 o más (ídem anterior)

Programación 1 - Régimen de promoción

Los alumnos que, **habiendo aprobado el parcial práctico (en primera o segunda fecha)**, no obtuvieran el mínimo de 6 como nota, podrán rendir un examen recuperatorio de la promoción si tuvieran nota mayor a 4.

Los alumnos que aprueben la promoción y **se encuentren inscriptos en el curso bajo esta modalidad**, tendrán registrada su nota al final del curso.

Programación 1 - Examen final

POR EXAMEN FINAL

Alumnos que rinden el examen final

Plazo de validez de la habilitación: La habilitación para rendir Examen Final, tendrá una validez de cuatro (4) semestres, inmediatos siguientes al del curso realizado.

Para rendir los exámenes finales existirán mesas examinadoras integradas por los Profesores de las Áreas/ Asignaturas. Los alumnos **deberán inscribirse en las fechas determinadas**, según el calendario académico.

Programación 1 - Con examen Final

✓ VER REGLAMENTO para mas detalles de la teoría

- ✓ Blog de la cátedra: <http://blogs.unlp.edu.ar/progra1/>
- ✓ Cartelera Virtual: <https://gestiondocente.info.unlp.edu.ar/cartelera>

FINALES – EN LA SEMANA DE FINAL MARTES con horario a confirmar en blog y cartelera

Programación I Información de la cátedra

En este curso trabajaremos con un entorno virtual de enseñanza y aprendizaje (IDEAS), para:

- Comunicarnos vía una mensajería interna
- Publicar las prácticas y materiales de estudio
- Publicar información de interés de la cursada

URL: <http://ideas.info.unlp.edu.ar>

Accesso a IDEAS



Programación I - Información de la cátedra

[INICIO](#)[REGISTRO](#)[ACERCA](#)[CONTACTO](#)[AYUDA](#)

INICIO DE SESIÓN

[INGRESAR](#)[RECUPERAR CONTRASEÑA](#)

Cabecera

PARTES DEL SISTEMA



CONTACTO AYUDA



Cecilia Sanz

csanz@lidi.info.unlp.edu.ar



Título

MIS CURSOS

COMO DOCENTE



Aprendiendo a resolver problemas - Demo

Profesor: Sanz Cecilia



COMO ESTUDIANTE



Analisis de Informacion Espacial del Sector Educativo

Profesor: Cas



Opciones para el usuario conectado:

- Buscar curso
- Editar Notificaciones

Listado de cursos



ideas

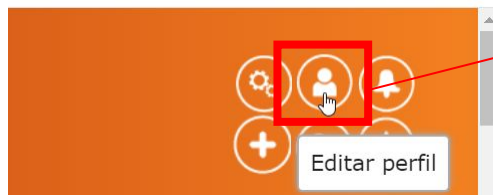
Integrando el mundo Digital
para Enseñar y Aprender Siempre



III-LIDI






NOTIFICACIONES



Importante para
recibir
notificaciones de
los cursos en tu
correo externo
(ejemplo, en tu
correo de Hotmail,
Gmail, yahoo o el
que hayas puesto
cuando te
registraste

Programación I Información de la cátedra

**SOLICITAR INSCRIPCIÓN EN
BUSCAR CURSO**

Buscar curso				
Ver área	Ingeniería Electrónica	Programación I		
Área ▼	Título del curso ▲	Profesor a cargo ▼	Fecha de inicio ▼	
Informática	Algorítmica y Programación I	Beatriz Depetris	14/08/2017	
Informática	ALGORITMICA Y PROGRAMACIÓN II	Beatriz Depetris	14/03/2016	
Informática	Programación I	Cecilia Sanz	12/03/2025	 Solicitar inscripción
Informática	Programación II- Segundo semestre 2020	Alejandro H. Gonzalez	02/09/2020	 Inscripción por parte del docente
Informática	Programación II-2019	Alejandro H. Gonzalez	07/08/2019	 Inscripción por parte del docente
Informática	Redictado-Programación II- 2020	Alejandro H. Gonzalez	09/03/2020	 Inscripción por parte del docente

1. Buscar Programacion I
2. Elegir el curso a cargo de Sanz, Cecilia
3. Hacer clic en Solicitar Inscripción

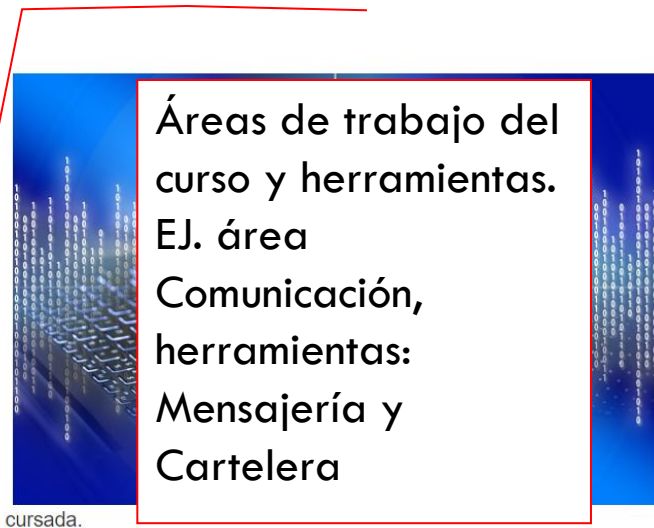
Programación I Información de la cátedra

Deben solicitar inscripción lo antes posible para que el JTP los habilite para ver el curso. EL curso lo verán como la imagen debajo, los íconos son atajos para ingresar por una herramienta determinada al curso (ej. ingresar a la Mensajería)



Hacer clic en el título del curso para Ingresar o en el panel debajo se elige a qué herramienta del curso entrar

Programación I Información de la cátedra



PROGRAMACIÓN I - 2024

Bienvenidos a este espacio de trabajo de la Asignatura Programación I

Hemos planificado utilizar este entorno con dos propósitos bien definidos:

1. Publicar el material de estudio y prácticas de la asignatura (Área de Inf. Gral y Contenidos)
2. Generar un espacio de comunicación alternativa, a través de las herramientas de Cartelera y Mensajería del área de Comunicación.

Para iniciar tu recorrido, debés hacer clic en la opción de **Propuesta** (del menú izquierdo) y ver allí el Programa, y el Reglamento. Luego en **Materiales y Actividades**, tendrás disponibles las teorías y las prácticas a medida que avance la

cursada.

MOTIVACIÓN



MOTIVACIÓN

¿Por qué elegimos esta carrera?
Compartimos este video

Video:

<https://www.youtube.com/watch?v=Y1HHBXDL9bg>

¿Qué es la Informática?





Es la **ciencia** que estudia el análisis y **resolución de problemas** utilizando **computadoras**.



La palabra **ciencia** se relaciona con una metodología fundamentada y racional para el estudio y resolución de los problemas. En este sentido la Informática se vincula especialmente con la Matemática y la Ingeniería.



Es la **ciencia** que estudia el análisis y **resolución de problemas** utilizando **computadoras**.



La **resolución de problemas** utilizando las herramientas informáticas puede tener aplicaciones en áreas muy diferentes tales como biología, comercio, control industrial, administración, robótica, educación, arquitectura, etc.



Es la **ciencia** que estudia el análisis y **resolución de problemas** utilizando **computadoras**.



Una **computadora** es una máquina digital y sincrónica, con cierta capacidad de cálculo numérico y lógico controlado por un programa almacenado y con probabilidad de comunicación con el mundo exterior y capacidad de almacenamiento de información.



La finalidad de la computadora es ayudar al hombre a realizar tareas repetitivas en menor tiempo y con mayor exactitud. La computadora no razona ni crea soluciones, sino que ejecuta una serie de órdenes que le proporciona el ser humano.

Objetivo de la Informática

Resolver problemas del mundo real utilizando computadoras.



*La computadora es una herramienta que podemos emplear en la resolución de problemas. La posibilidad de resolver un problema real con una computadora depende del **programa (software)** que carguemos en la máquina.*

HISTORIA de la
Informática

HISTORIA

The background of the slide is a historical map, likely a nautical chart, showing coastlines and geographical features. A magnifying glass is positioned over the map, focusing on a specific area. In the bottom right corner, a portion of a compass or sextant is visible, adding to the historical theme. The overall aesthetic is that of a historical document or a treasure map.

Historia de la Informática

En el Centro de Innovación y Transferencia de la Facultad (CIYTT), tenemos disponible un recorrido sobre algunas de estas figuras de la historia de la Informática. Video Innovática (juego de realidad virtual):

<https://drive.google.com/file/d/1f2tgDW5iFPYqr0OaP-s-pul6rd6YkTHa/view?usp=sharing>

Juego Albores:

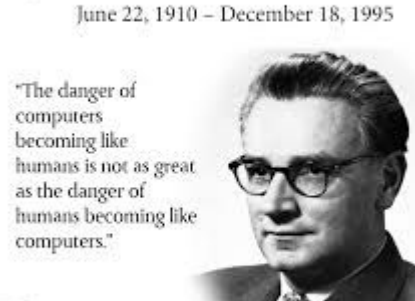
https://drive.google.com/file/d/1lICbNv1XWuVr_1HbS7aFROX4bPUeFS9x/view?usp=sharing



CHARLES BABBAGE

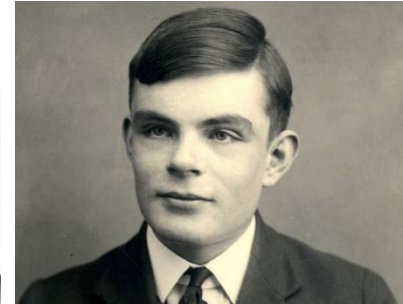


ADA LOVELACE



KONRAD SUZE □ Z3

1era computadora
moderna (1941)

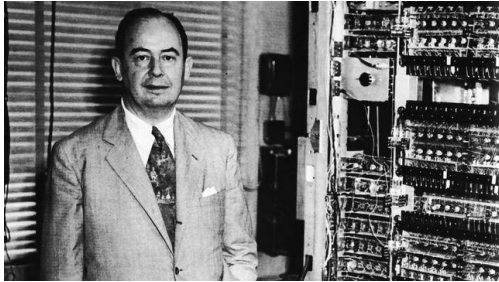


ALAN TURING

Historia de la Informática

¿Vemos un video?

Video: <https://www.youtube.com/watch?v=8tpxARw1X04>



JOHN VON NEUMANN
EDVAC



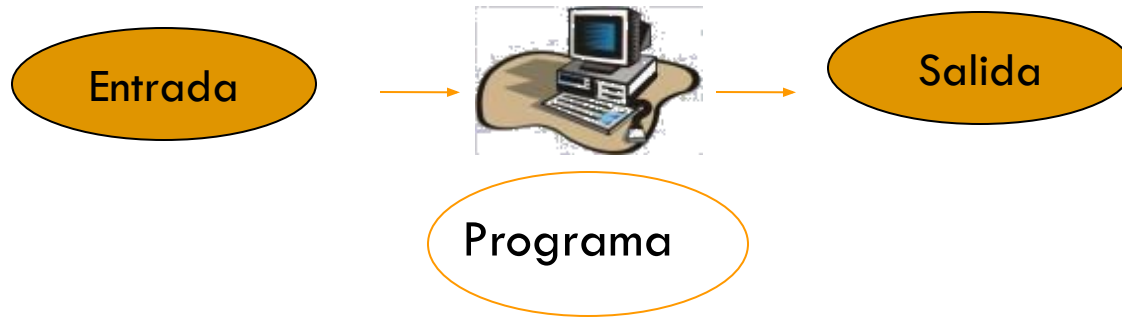
GRACE HOPPER
1ER COMPILADOR



ALAN KAY
INTERFACES GRÁFICAS

Otra visión sobre la computadora

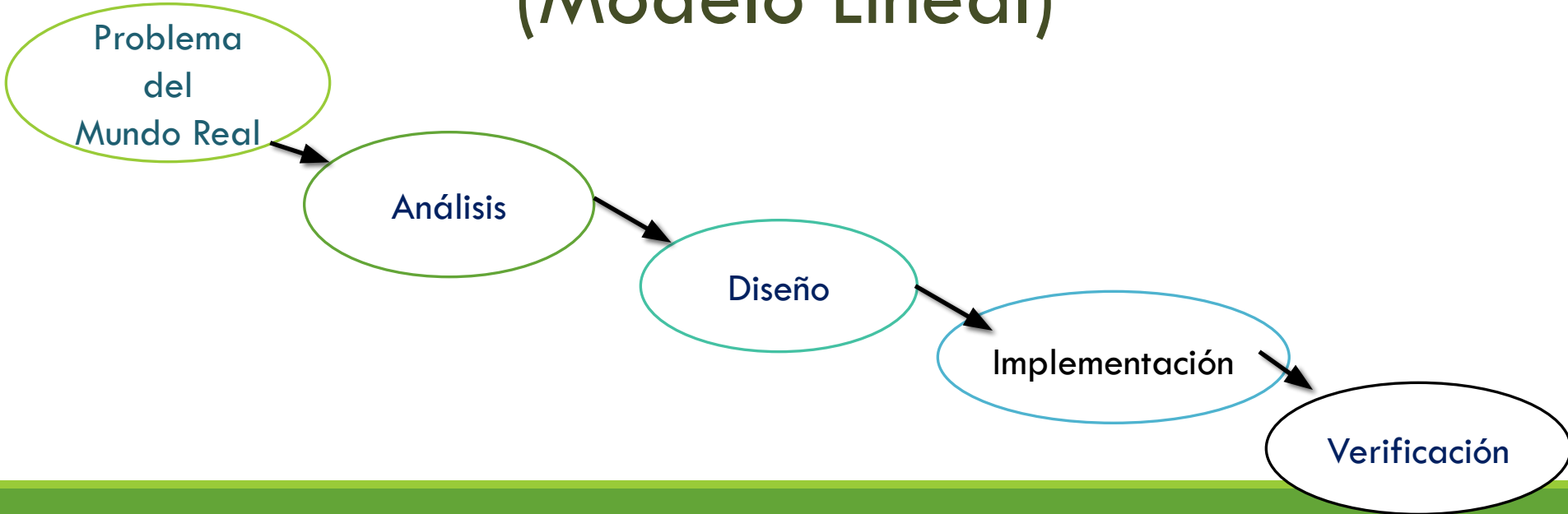
- La computadora es una máquina capaz de aceptar datos de entrada, ejecutar con ellos cálculos aritméticos y lógicos y dar información de salida (resultados), bajo control de un programa previamente almacenado en su memoria.



Etapas para resolver problemas

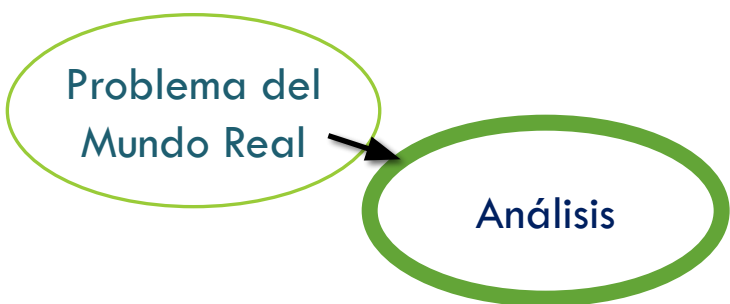


Etapas para la resolución de un problema mediante la computadora (Modelo Lineal)



Etapas para la resolución de un problema mediante la computadora (Modelo Lineal)

Problema del
Mundo Real



Análisis

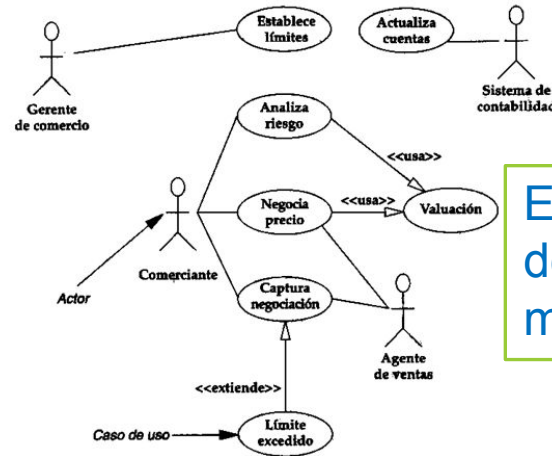
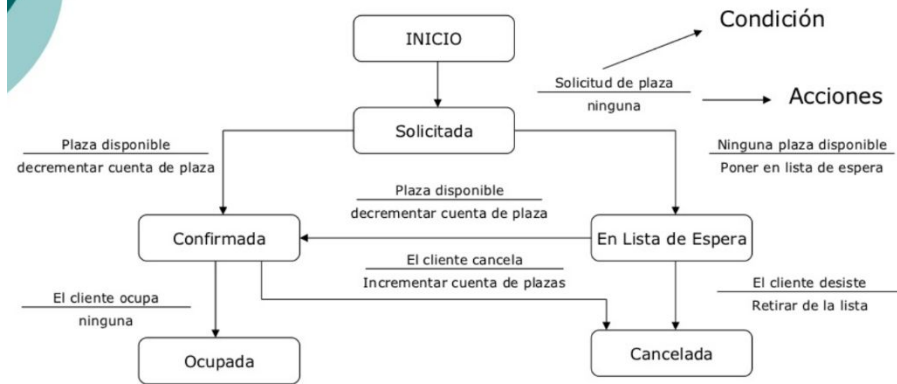
1

Se sintetizan los requerimientos del problema y se crea un **modelo** del problema, simplificando el contexto y los datos a representar. Análisis de entradas y salidas

¿A qué llamamos modelo?

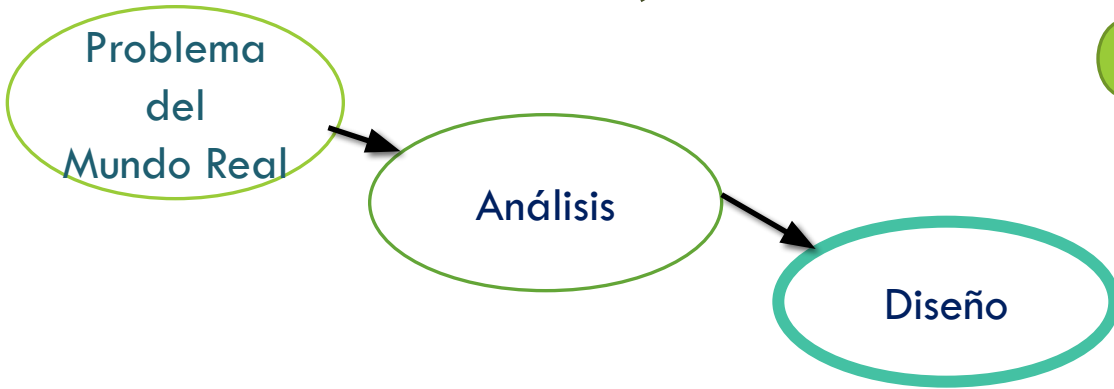
El término modelo es utilizado por los científicos en referencia a una representación simplificada de la realidad, que puede ser utilizada para hacer predicciones que no pueden ser testeadas por experimentación u observación

Ejemplo de un diagrama de transición de estados para reserva de Hotel (Utilizando forma UML).



Ejemplos: formas de mostrar un modelo

Etapas para la resolución de un problema mediante la computadora (Modelo Lineal)

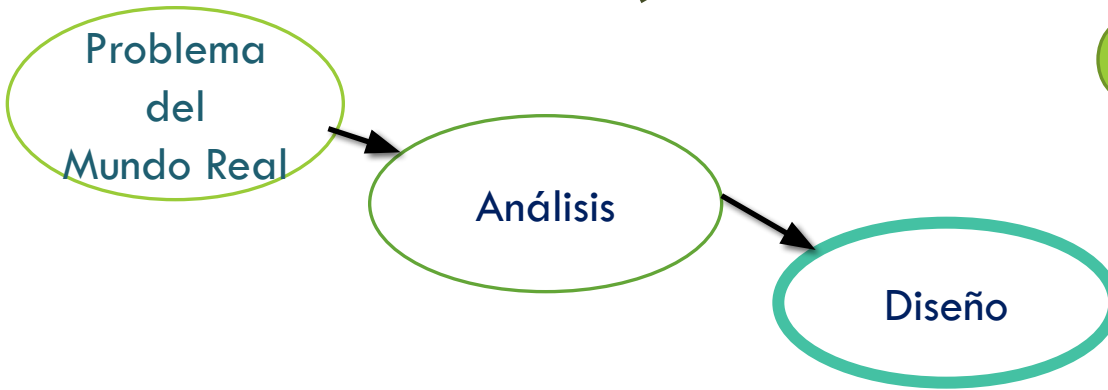


2

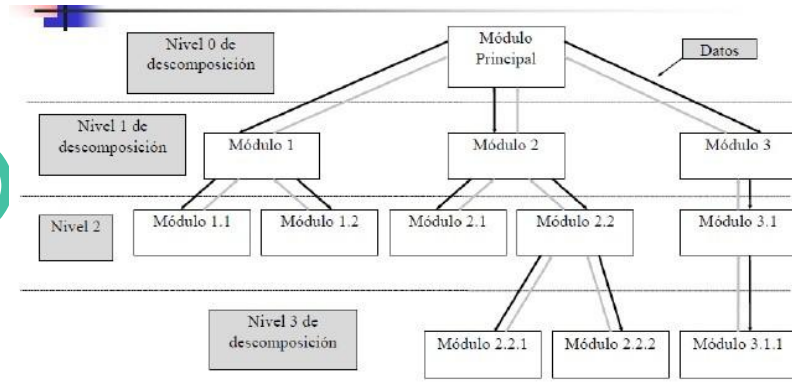
Se diseña una solución. En nuestro caso utilizaremos como estrategia la **Descomposición funcional**. Relacionado con la técnica de **Divide y vencerás**.

La descomposición funcional de todas las acciones que propone el modelo nos ayudará a reducir la complejidad, a distribuir el trabajo y en el futuro a re-utilizar los módulos (es una forma de diseñar la solución, existen otras)

Etapas para la resolución de un problema mediante la computadora (Modelo Lineal)



2

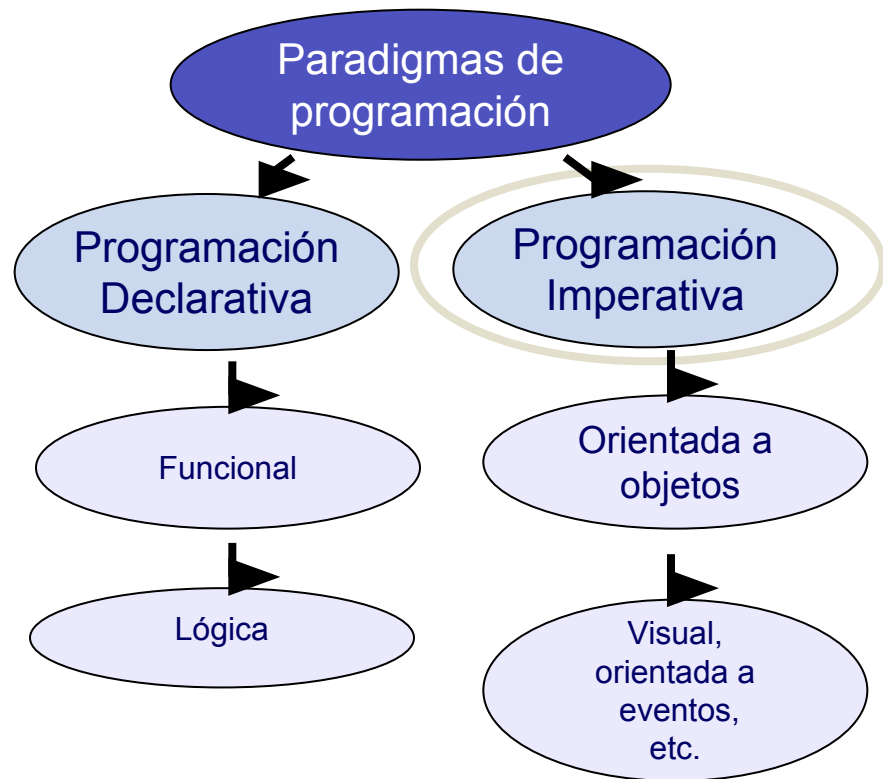


Paradigmas de Programación

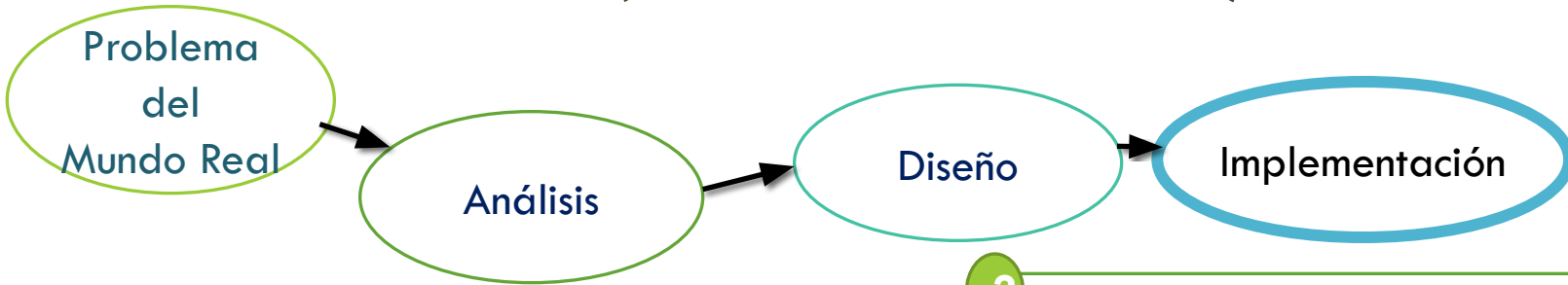
El diseño de la aplicación estará impactado por el paradigma de diseño y posterior de programación que se elija.

En una primera clasificación, se encuentran dos grandes grupos en donde es posible agrupar la mayoría de los paradigmas de programación conocidos hasta ahora.

En esta primera parte del curso trabajaremos bajo el paradigma imperativo/procedural.



Etapas para la resolución de un problema mediante la computadora (Modelo Lineal)

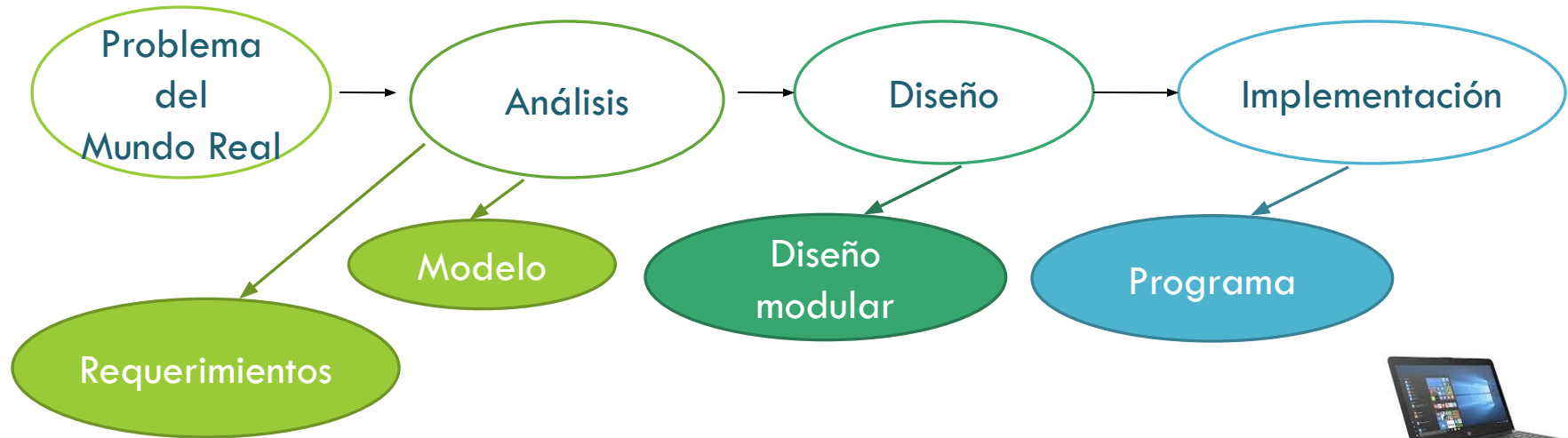


3

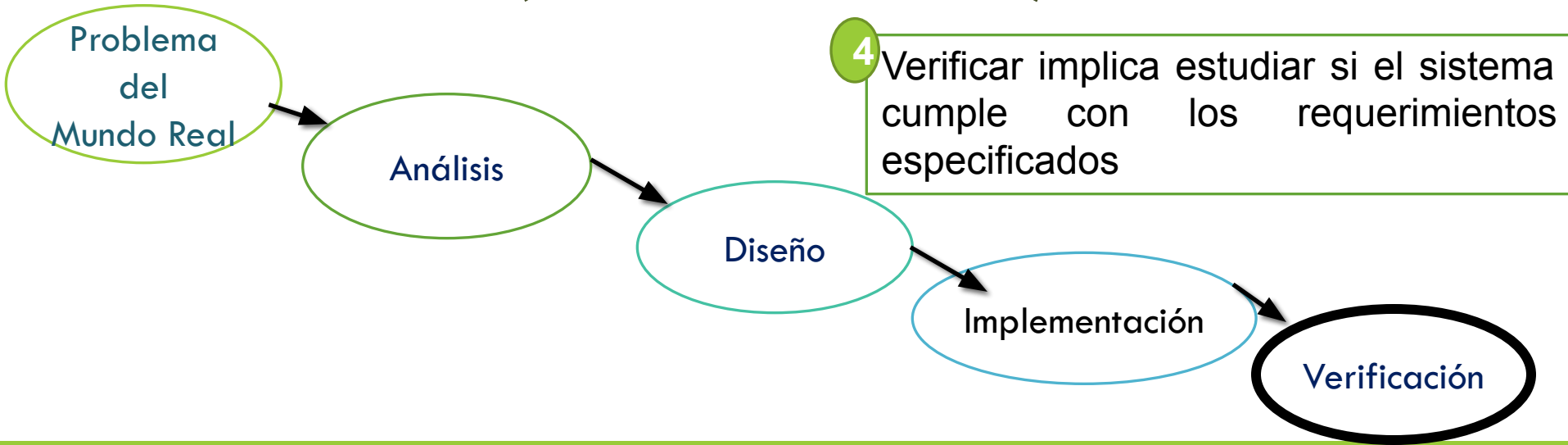
En esta etapa se deben escribir algoritmos en un lenguaje de programación y elegir la representación de los datos.

Implementación

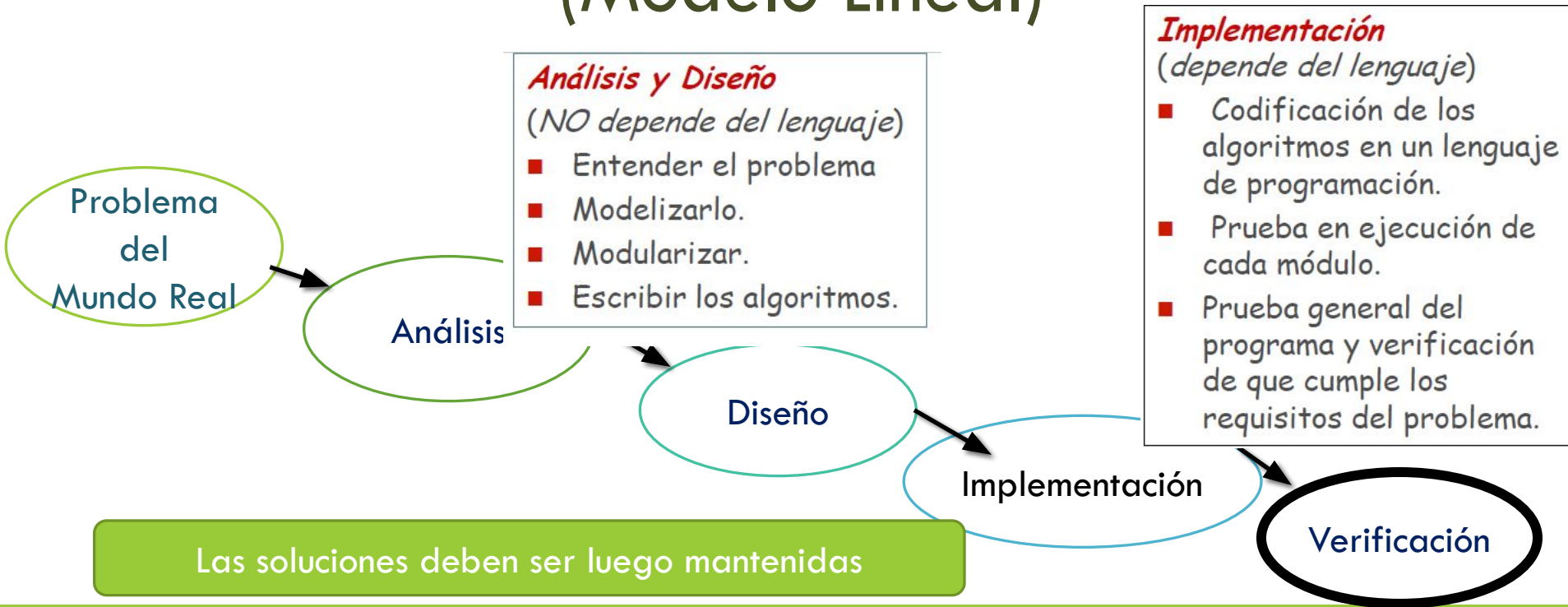
Una vez que se tiene la descomposición en módulos, debemos pasar a su implementación: esto requiere escribir **algoritmos** y elegir **datos**



Etapas para la resolución de un problema mediante la computadora (Modelo Lineal)



Etapas para la resolución de un problema mediante la computadora (Modelo Lineal)



¿Qué es un programa?



Definición de Programa

PROGRAMA = ALGORITMO + DATOS

Programa: Es un conjunto de instrucciones u órdenes ejecutables sobre una computadora, que permite cumplir con una función o tarea específica (dichas órdenes están expresadas en un lenguaje de programación concreto).

Definición de Programa

PROGRAMA = ALGORITMO + DATOS

Las componentes básicas de un programa son instrucciones y datos. Un programa está escrito en un lenguaje de programación.

¿Cuál es la diferencia entre algoritmo y programa?

¿Qué es un algoritmo?



Definición de Algoritmo

Algoritmo: especificación rigurosa de la secuencia de pasos (instrucciones) a realizar sobre un autómatas para alcanzar un resultado deseado en un tiempo finito. **No necesariamente está escrito en lenguaje de programación**

- ✓ Alcanzar el resultado en tiempo finito significa que suponemos que un algoritmo comienza y termina. Está implícito que el número de instrucciones debe ser también finito.
- ➡ ✓ Especificación rigurosa significa que debemos expresar un algoritmo en forma clara y unívoca.
- ✓ Si el autómatas es una computadora, tendremos que escribir el algoritmo en un lenguaje “entendible” y ejecutable por la máquina.

¿Qué es un dato?



Definición de Dato

PROGRAMA = ALGORITMO + DATOS

Dato: Es una representación de un objeto, aspecto, situación del mundo real mediante la cual podemos modelizar aspectos del problema que se quiere resolver con un programa sobre una computadora.

Por ejemplo: Un precio, una letra, un nombre, un mueble, una persona

Datos – Constantes y Variables

PROGRAMA = ALGORITMO + DATOS

Dato: Conceptualmente pueden ser **constantes** (no cambian durante la ejecución del programa), o **variables** (pueden cambiar su valor durante la ejecución del programa)



¿Qué es programar?

Para llegar a programar debemos:

- Elegir la representación adecuada de los datos del problema.

¿Qué datos y qué representación para cada uno, tendríamos en el problema de ejemplo?

- Elegir el lenguaje de programación a utilizar, según el problema y los recursos disponibles

Debemos decidir el lenguaje de programación adecuado para este caso – Lenguajes compilados (C, C++, Go, Pascal) e interpretados (Python, Javascript, PHP)

- Definir el conjunto de instrucciones en el lenguaje elegido, cuya ejecución ordenada lleva a la solución del problema

Se trata de escribir cada uno de los módulos definidos anteriormente en el lenguaje elegido... Esto es lo que aprenderemos en el curso.

ALGO SOBRE LENGUAJE PASCAL:

- Su creador fue Niklaus Wirth, en la Universidad Técnica de Zurich, en Suiza.
- El objetivo era crear un lenguaje de programación de alto nivel para enseñar programación estructurada.
- Rápidamente, Pascal fue usado en Estados Unidos y Europa, tanto como lenguaje de enseñanza y como lenguaje de propósito general.
- El nombre de Pascal fue elegido en honor a Blaise Pascal, un científico y matemático francés. Uno de sus logros se encuentra en la invención de la primera máquina de calcular mecánica del mundo.

ALGO SOBRE LENGUAJE PASCAL:

PROGRAMAR EN PASCAL:

¿VEMOS ALGUNOS EJEMPLOS?

Ejemplo 1

Ejemplos de Datos vistos

¿Qué pasa con los datos?

Programa en R-Info

programa nombre

procesos

proceso nombre

variables

comenzar

fin

variables

comenzar

fin

Tipos de datos:

numero

boolean

¿Qué son los tipos de datos?



Tipos de Datos



- Los algoritmos generalmente operan sobre datos de distinta naturaleza (números, letras, símbolos, etc.).
- Por lo tanto, los programas que implementan dichos algoritmos, necesitan alguna manera de representarlos.

Un *tipo de dato* es una clase de objetos de datos ligados a un conjunto de operaciones para crearlos y manipularlos.



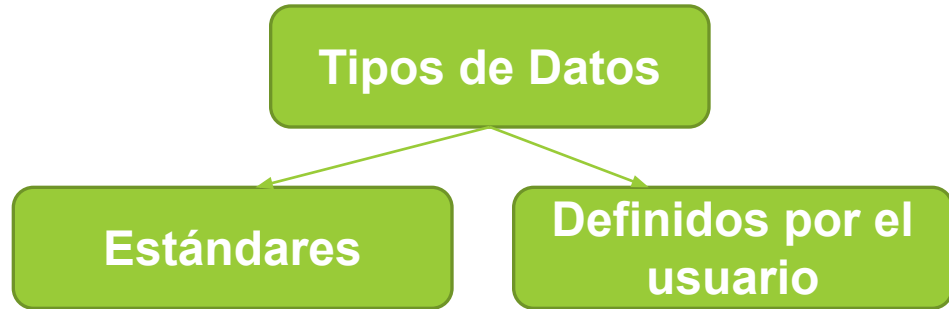
Tipos de Datos

Un *tipo de dato* es una clase de objetos de datos ligados a un conjunto de operaciones para crearlos y manipularlos.

Los tipos de datos se caracterizan por:

- ✓ Un rango de valores posibles.
- ✓ Un conjunto de operaciones realizables sobre ese tipo.
- ✓ Una representación interna.

Tipos de Datos



- Los tipos de datos definidos por el **lenguaje** (primitivos o estándar) son provistos por el lenguaje y tanto la representación como sus operaciones y valores son reservadas al mismo.

- Los tipos definidos por el usuario, permiten definir nuevos tipos de datos a partir de los tipos simples.

Tipos de Datos

Definición de Tipo de datos simple

Los tipos de datos son simples si en un momento dado de la ejecución pueden contener sólo un valor de los posibles para el tipo.

Definición de Tipo de dato ordinal

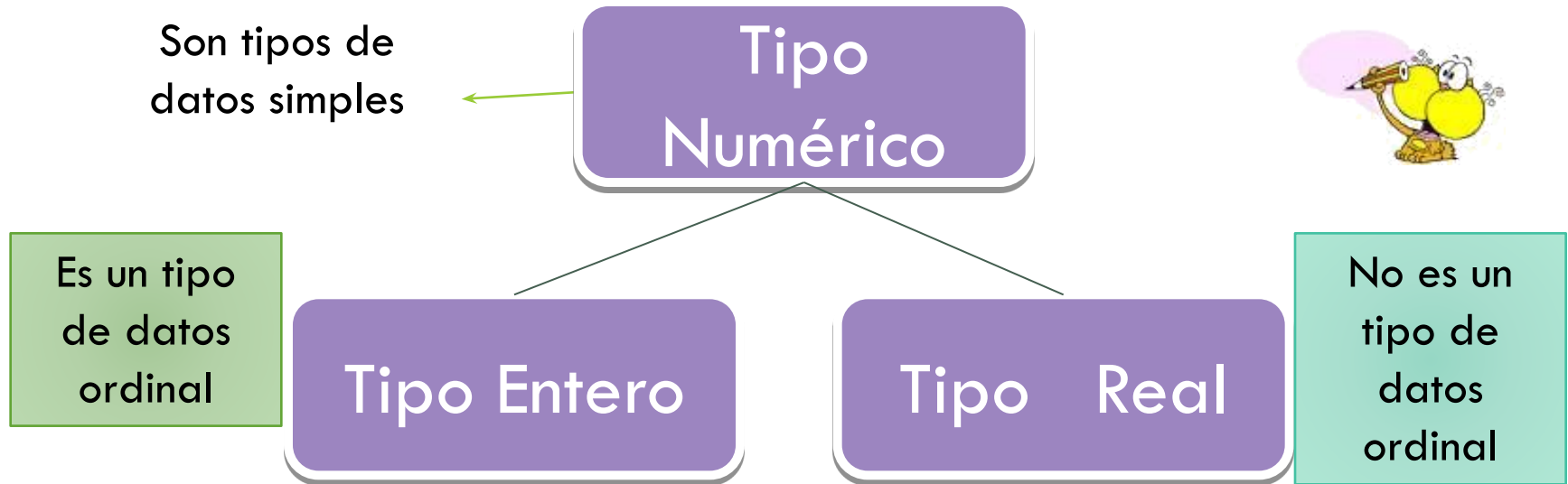
Los tipos de datos son ordinales si para cada valor de los posibles para el tipo, se puede decir cuál es su inmediato anterior y posterior.

Tipos de datos
numéricos



Tipos de Datos - Numérico

El tipo de datos numérico es el conjunto de valores que puede representarse de dos formas:



Tipos de Datos – Numérico - Entero

- Es simple y ordinal.
- Los elementos (dado que la memoria tiene una capacidad finita) son por ejemplo (dependiendo del compilador y la máquina donde se corra): -2^{32} a 2^{32} . Ejemplo: ..., -1, 0, 1, 2, etc.
- Los datos de este tipo no poseen decimales.
- ¿Ejemplos de datos enteros?

Tipos de Datos – Numérico - Real

- Es un tipo de datos simple que permite representar números con decimales.
- Los elementos de este tipo de datos se representan con cierta cantidad de bits para la parte entera y otra cierta cantidad para la parte decimal.
- ¿Ejemplos de datos reales?

Tipos de Datos – Numérico - Operaciones

ENTERO

- Suma (+)
- resta (-)
- Multiplicación (*)
- División entera (div)
- Módulo (mod)
- Operaciones relacionales

REAL

- Suma (+)
- resta (-)
- Multiplicación (*)
- División (/)
- División entera (div)
- Módulo (mod)
- Operaciones relacionales



Tipos de Datos – Numérico - Operaciones

Las expresiones que tienen dos o más operandos requieren reglas matemáticas que permitan determinar el orden de las operaciones. El orden de precedencia para la resolución, ya conocido, es:

1. operadores $*$, $/$
2. operadores $+$, $-$
3. operadores div y mod .



En caso que el orden de precedencia natural deba ser alterado, es posible la utilización de paréntesis dentro de la expresión.

$$6 + 2 * 4 = 14$$

$$(6 + 2) * 4 = 32$$

Tipos de Datos – Numérico - Operaciones

Además de los operadores matemáticos mencionados, el tipo de dato numérico posee operadores relacionales que permiten comparar valores.

Dichas relaciones son la **igualdad** ($=$), **desigualdad** ($<>$) y **de orden** ($<$, $<=$, $>$, $>=$).

Dependiendo del lenguaje de programación estos operadores tendrán diferente representación.

El resultado es del tipo de dato lógico. (Verdadero o Falso).



Tipos de Datos – Numérico - Operaciones

Ejemplos

$3 < 6$	Resultado: verdadero
$3 \leq 6$	Resultado: verdadero
$3 > 6$	Resultado: falso
$3 < > 6$	Resultado: verdadero
$5 \geq 2$	Resultado: verdadero

Ejemplo 2 en
Pascal

Tipos de datos lógico

FALSE
true

Tipos de Datos – Lógico

El tipo de dato lógico permite representar datos que pueden tomar solamente uno de dos valores. Este tipo de dato también es llamado tipo de dato boolean. Es un tipo de datos simple y ordinal.

Dichos valores son:

verdadero (true)

falso (false)

Se utiliza en situaciones donde se representan dos alternativas de una condición.

Tipos de Datos – Lógico -Operaciones

Los operadores lógicos o booleanos básicos son:

- negación (**not**),
- conjunción (**and**),
- disyunción (**or**).



El resultado de estas operaciones es el correspondiente a las conocidas tablas de verdad.

Tipos de Datos – Lógico -Operaciones

En cuanto a la precedencia entre los operadores lógicos se da:

1. negación (**not**),

2. conjunción (**and**),

3. disyunción (**or**).



`not (5>8) and (15<15)`

`not ((5>8) and (15<15))`

Tipos de datos
carácter



Tipos de Datos – Carácter



El tipo de dato carácter representa a elementos de un conjunto finito y ordenado de caracteres que la computadora reconoce.

Un dato de tipo carácter contiene sólo un carácter en un momento dado de la ejecución. Por esto es un tipo de **datos simple**. Además es un tipo de datos ordinal.

Los caracteres que reconocen las computadoras se normalizaron, entre otros, por un estándar llamado ASCII, el cual permite establecer un orden de precedencia entre los mismos.

TABLA DE CARACTERES DEL CÓDIGO ASCII

1	25	49	73	97	121	145	169	193	217	241
2	26	50	74	98	122	146	170	194	218	242
3	27	51	75	99	123	147	171	195	219	243
4	28	52	76	100	124	148	172	196	220	244
5	29	53	77	101	125	149	173	197	221	245
6	30	54	78	102	126	150	174	198	222	246
7	31	55	79	103	127	151	175	199	223	247
8	32	56	80	104	128	152	176	200	224	248
9	33	57	81	105	129	153	177	201	225	249
10	34	58	82	106	130	154	178	202	226	250
11	35	59	83	107	131	155	179	203	227	251
12	36	60	84	108	132	156	180	204	228	252
13	37	61	85	109	133	157	181	205	229	253
14	38	62	86	110	134	158	182	206	230	254
15	39	63	87	111	135	159	183	207	231	255
16	40	64	88	112	136	160	184	208	232	PRESIONA
17	41	65	89	113	137	161	185	209	233	LA TECLA
18	42	66	90	114	138	162	186	210	234	Alt
19	43	67	91	115	139	163	187	211	235	MÁS EL
20	44	68	92	116	140	164	188	212	236	NUMERO
21	45	69	93	117	141	165	189	213	237	CORTESÍA DE:
22	46	70	94	118	142	166	190	214	238	REDEC
23	47	71	95	119	143	167	191	215	239	CREANDO
24	48	72	96	120	144	168	192	216	240	desde 1976

Tipos de Datos – Caracter

Algunos caracteres especiales: '!', '#', '\$', '%', □

Dígitos: '0', '1', '2', ..., '8', '9'

Letras mayúsculas: 'A', 'B', 'C', ..., 'Y', 'Z'

Letras minúsculas: 'a', 'b', 'c', ..., 'y', 'z'

¿Ejemplos de datos carácter?

Tipos de Datos – Caracter

Se debe tener en cuenta que no es lo mismo el valor entero 1 que el símbolo carácter '1'.

Un valor del tipo de dato carácter es sólo uno de los símbolos mencionados.

¿Si aplico la operación suma a un carácter con valor '4' y otro carácter con valor '3' que valor obtengo?

¿Si aplico la operación suma a un carácter con valor '4' y un número con valor 3 que valor obtengo?



Tipos de Datos – Caracter - Operaciones

Operador	Tipos de Operando	Tipo de resultado
=	Caracter	Lógico
<>	Caracter	Lógico
>	Caracter	Lógico
<	Carácter	Lógico
>=	Carácter	Lógico
<=	Caracter	Lógico



Tipos de Datos – Carácter - Operaciones

('b' = 'B') da como resultado falso

('c' < 'Z') da como resultado falso

('c' < 'z') da como resultado verdadero

('X' > '5') da como resultado verdadero

('?' < 'H') da como resultado verdadero

('4' = 4) ?

no puede evaluarse porque los operandos son de tipos distintos.

Tipos de Datos – Conclusión

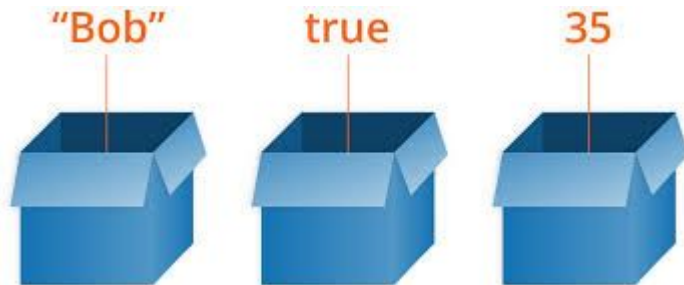
Como conclusión, dado que hay diferentes tipos de datos debe destacarse que:

Los diferentes tipos de datos deben especificarse y a esta especificación dentro de un programa se la conoce como **declaración**.



Una vez declarado un tipo, podemos asociar al mismo **variables**, es decir nombres simbólicos que pueden tomar los valores característicos del tipo.

VARIABLES



Recordando - Variables

Una variable es una zona de memoria cuyo contenido va a ser de alguno de los tipos mencionados anteriormente.

La dirección inicial de esta zona se asocia con el nombre de la variable.

Variable A
Referencia una
zona de
memoria





Variables

Llamaremos identificadores a los nombres descriptivos que se asocia a los objetos (constantes y variables) para abstraer dentro del programa su dirección real en memoria y su valor. En el ejemplo anterior, el identificador es A.



En Pascal, los identificadores están formados por letras, dígitos en cualquier orden y algunos símbolos especiales (excepto el primer carácter que debe ser una letra).

Constantes

Su valor no cambia durante la ejecución del programa

Se indica un valor en su declaración

Se declara con la palabra clave `const`

```
const nombre = valor;
```

donde `nombre` es el identificador que representa el nombre de la constante. El tipo de dato de la constante queda definido implícitamente por el tipo de dato de valor.

-

Variables

Su valor cambia durante la ejecución del programa

Se indica un valor en el programa

Se declara con la palabra clave `var`

```
var nombre : tipodedato;
```

donde `nombre` es el identificador que representa el nombre de la variable.

Programa en Pascal



¿Cómo es un programa ahora?

Programa nombre

Procesos

proceso nombre

variables

comenzar

fin

Variables

Comenzar

Fin

Program nombre;

const

N = 25 ; {N se asume de tipo de dato entero }

pi = 3.1416; {pi se asume de tipo de dato real }

c = 'C' ; {c se asume de tipo de dato carácter }

Var

cantidad: integer; {cantidad puede contener nro. entero}

total_cobrado: real; { total_cobrado número real }

estado: boolean; { estado valor lógico }

letra: char; { letra contiene un carácter }

Begin

End.

¿Cómo es un programa ahora?

- Los diferentes *tipo_de_variable* posibles **en Pascal** se corresponden con los tipos de datos vistos hasta el momento.
- ***integer*** tipo de dato numérico entero
- ***real*** tipo de dato numérico real
- ***boolean*** tipo de dato lógico
- ***char*** tipo de dato carácter

¿Cómo es un programa ahora?

Program nombre;

const

N = 25; {N se asume de tipo de dato entero }

Var

cantidad: integer; {cantidad puede contener nro. entero}

total_cobrado: real; { total_cobrado número real }

estado: boolean; { estado valor lógico }

letra: char; { letra contiene un carácter }

Begin

End.

Tipos de Lenguajes

- ✓ Algunos lenguajes, como Pascal, exigen que se especifique a qué tipo pertenece cada una de las variables
- ✓ Verifican que el tipo de los datos asignados a esa variable se correspondan con su definición.
- ✓ Esta clase de lenguajes se denomina *fuertemente tipados* (strongly typed).

Tipos de Lenguajes

- ✓ Otra clase de lenguajes, determina el tipo acorde al primer valor que se le asigne, se denomina ***auto tipados*** (***self typed***).
- ✓ Existe una tercera clase de lenguajes que permiten que una variable tome valores de distinto tipo durante la ejecución de un programa. Esta se denomina ***dinámicamente tipados*** (***dinamically typed***). En general son **lenguajes interpretados...**



Read y Write

¿Cómo asignamos valores a variables?

Readln:

se usa para leer datos (por defecto desde teclado) y asignarlos a las variables correspondientes.

¿Cómo asignamos valores a variables?

HASTA AHORA...

```
Program uno;
```

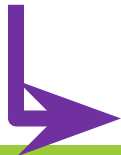
```
Var
```

```
    num:integer;
```

```
Begin
```

```
    num:= 8;
```

```
End.
```



ASIGNACIÓN

```
Program uno;
```

```
Var
```

```
    num:integer;
```

```
Begin
```

```
    readln (num);
```

```
End.
```



Entrada salida en Pascal

Writeln: se usa para mostrar información en pantalla. Puede ser, por ejemplo, el contenido de una variable. Pueden ser de tipo entero, real, char. Los datos a mostrar si son más de uno deben ir separados por coma.

Entrada salida en Pascal

```
program ejemploWrite;  
var  
  a, b: integer;  
Begin  
  Readln(a);  
  b:= a * 4;  
  write ('El valor de b es:' , b);  
End.
```

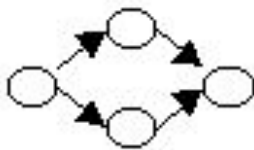
Estructuras de control

Construcciones estructurales en forma de grafo de flujo.

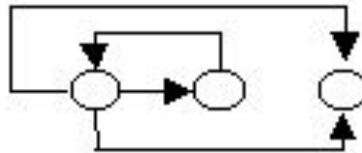
Secuencia



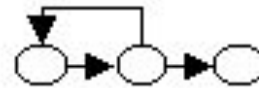
IF



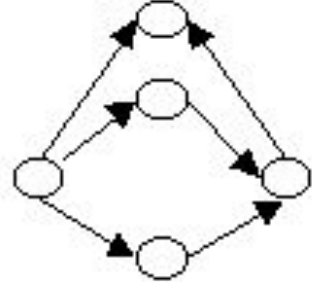
While



Until



Case



ESTRUCTURAS DE CONTROL

¿Qué son?



Estructuras de Control - Concepto

Todos los lenguajes de programación tienen un conjunto mínimo de instrucciones que permiten especificar el control del algoritmo que se quiere implementar. Permiten cambiar el flujo de ejecución de un programa

- Veremos que este conjunto **debe contener como mínimo:**
 - ✓ secuencia
 - ✓ decisión / selección
 - ✓ iteración



EJEMPLOS DE CÓMO ESTAS SITUACIONES REFLEJAN LA NECESIDAD DE CONTAR CON ESTRUCTURAS DE CONTROL

De pequeños



Si me prestás ese juguete, te invito a mi cumple



Mientras no me dan la mamadera, lloro



Para cada bloque que tengo, lo tomo y lo apilo

EJEMPLOS DE CÓMO ESTAS SITUACIONES REFLEJAN LA NECESIDAD DE CONTAR CON ESTRUCTURAS DE CONTROL

De más grandes



Si me llama, entonces se lo cuento



Para cada materia de la carrera, debo aprobar el final o la promoción

EJEMPLOS DE CÓMO ESTAS SITUACIONES REFLEJAN LA NECESIDAD DE CONTAR CON ESTRUCTURAS DE CONTROL

De adultos



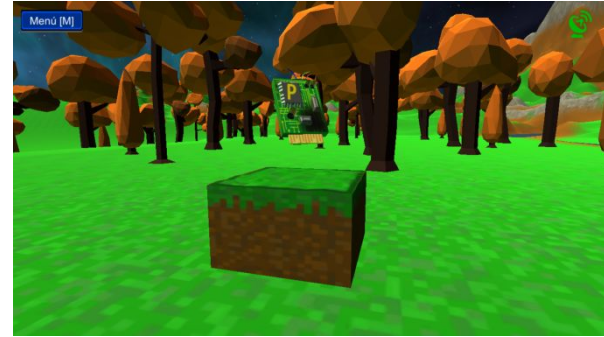
Si el porcentaje de ganancias es mayor a 50, reforzaré la inversión.
Si no deberé estudiar el problema



Mientras haya lugar en la valija pongo ropa

Juego ASTROCÓDIGO

www.astrocodigo.com



Estructuras de Control en Pascal

Estructuras de Control en Pascal

Secuencia



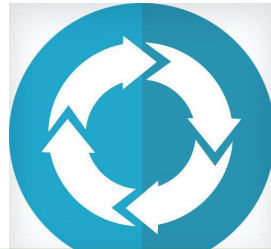
Decisión



Selección



Repetición



Iteración

Estructuras de Control - SECUENCIA

- La forma de ejecución más simple, está representada por una ***sucesión de operaciones*** (por ej. asignaciones), en la que el orden de ejecución coincide con el orden físico de aparición de las instrucciones.

8	100	108
a	b	res



```
Program ejemplo;  
var  
  a, b, res: integer;  
begin  
  readln (a);  
  b:= 100;  
  res := a + b;  
end;
```

Estructuras de Control - Motivación

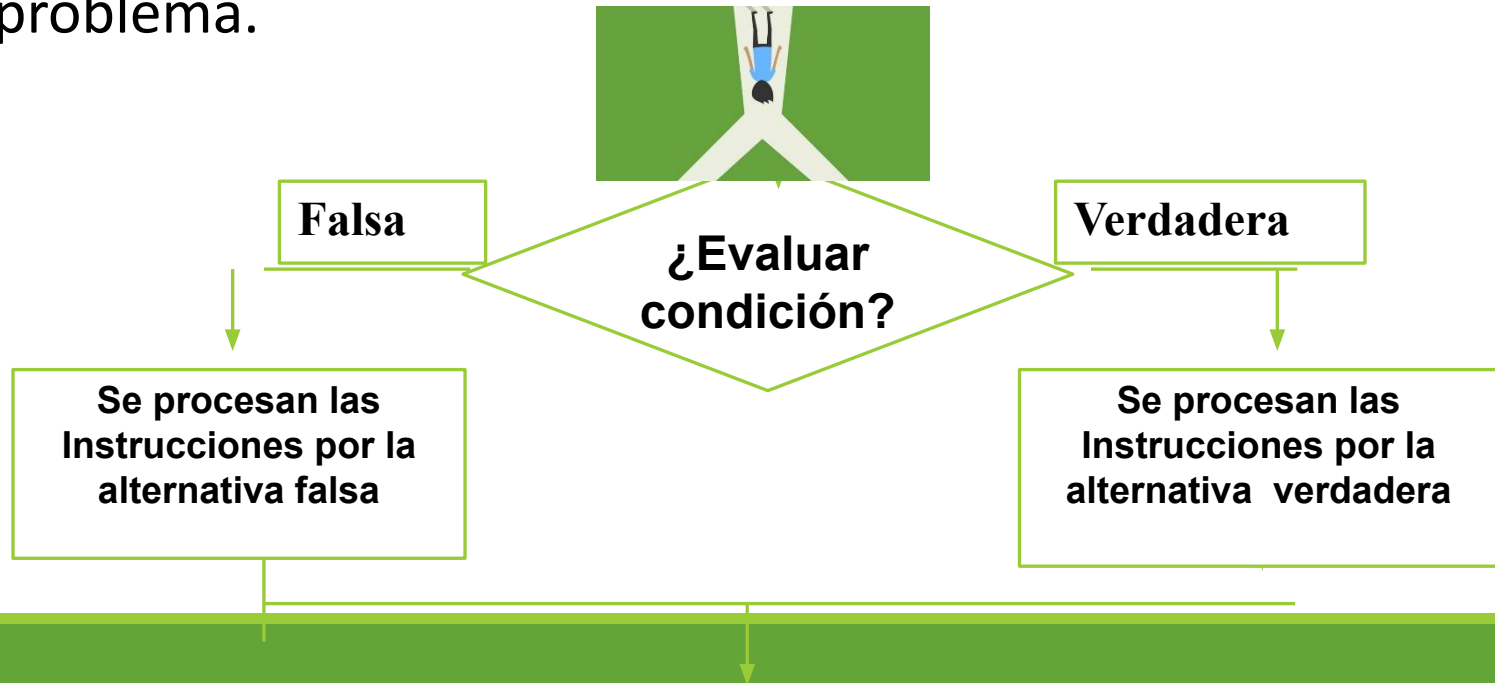
En un algoritmo representativo de un problema real es prácticamente imposible que todo sea secuencial.

Decisión



Estructuras de Control - DECISIÓN

- Es necesario tomar ***decisiones*** en función de los datos del problema.



Estructuras de Control - DECISIÓN

EN PASCAL...

if (condición) **then**

 accion

if (condición) **then**

 accion

else

 accion

EN PASCAL...

if (condición) **then**

begin

 accion

 accion

end

else

 accion

Decisión



Ejemplos

Estructuras de Control - DECISIÓN

Ejemplo I: Escriba un algoritmo que le permita determinar e informar si un número leído desde teclado es par o impar.

¿Cómo determino si es par?



¡Tú puedes hacerlo!



Estructuras de Control - DECISIÓN

```
Program uno;  
Var resto, num1:integer;  
Begin  
    readln (num1);  
    resto:= num1 MOD 2;  
    if (resto = 0)  
    then writeln ('Número par')  
    else writeln ('Número impar');  
End.
```

```
Program uno;  
Var  
    num1:integer;  
Begin  
    readln (num1);  
    if (num1 MOD 2 = 0)  
    then writeln ('Número par')  
    else writeln ('Número impar');  
End.
```

Estructuras de Control - DECISIÓN

Ejemplo II: Escriba un programa que lea dos números e informe el número más grande.

Estructuras de Control - DECISIÓN

Program uno;

Var

num1, num2: integer;

Begin

readln (num1);

readln (num2);

if (num1 > num2)

then writeln ('El mayor es: ', num1)

else writeln ('El mayor es: ', num2)

End.

*Qué informa el programa si
los números ingresados son
iguales?*

Estructuras de Control - DECISIÓN

Program dos;

Var

num1, num2: integer;

Begin

readln (num1);

readln (num2);

if (num1 > num2) **then** writeln ('El mayor es: ', num1)

else if (num2 > num1) **then** writeln ('El mayor es: ', num2)

else writeln('Los números son iguales');

End.

Resumiendo lo visto

- ✓ Etapas para la resolución de un problema
- ✓ Qué es un algoritmo y qué es un programa
- ✓ Qué es un dato
- ✓ Qué son los tipos de datos
- ✓ Algunos Tipos de datos en Pascal: numéricos, lógicos, y carácter
- ✓ Variables y constantes. Ejemplos en un Programa en Pascal
- ✓ Pascal como lenguaje fuertemente tipado