LICENCIATURA EN INFORMATICA

PROGRAMADOR UNIVERSITARIO

INGENIERÍA EN INFORMÁTICA

ELEMENTOS DE COMPUTACIÓN Y LÓGICA

2023

Lic. Mariela A. Velázquez - Dr. Leonardo Albarracin

Universidad Nacional de Tucumán

2020 - AÑO DEL GENERAL MANUEL BELGRANO

Contenido

Unidad 1. Proposiciones Simples y Compuestas. Representación simbólica. Conectivos lógicos: Negación, Conjunción, Disyunción, Condicional, Bicondicional. Tablas de verdad. Jerarquía de conectivos. Clasificación de fórmulas lógicas por su significado: Tautologías, contradicciones y contingencias. Implicación y equivalencia lógica. Leyes Lógicas

Unidad 2. Razonamientos. Componentes: premisas, conclusión. Reglas de inferencia. Métodos de demostración: directo e indirecto. Validez de un razonamiento.Consistencia de Premisas.

Unidad 3. Lógica de Predicados. Funciones proposicionales. Universo del discurso. Representación simbólica. Cuantificadores: Cuantificador universal y cuantificador existencial. Alcance de un cuantificador. Variables libres y variables vinculadas. Proposiciones categóricas. Negación de proposiciones cuantificadas. Equivalencia de proposiciones cuantificadas universalmente y existencialmente. Razonamientos. Reglas de especificación universal y existencial. Reglas de generalización universal y existencial.

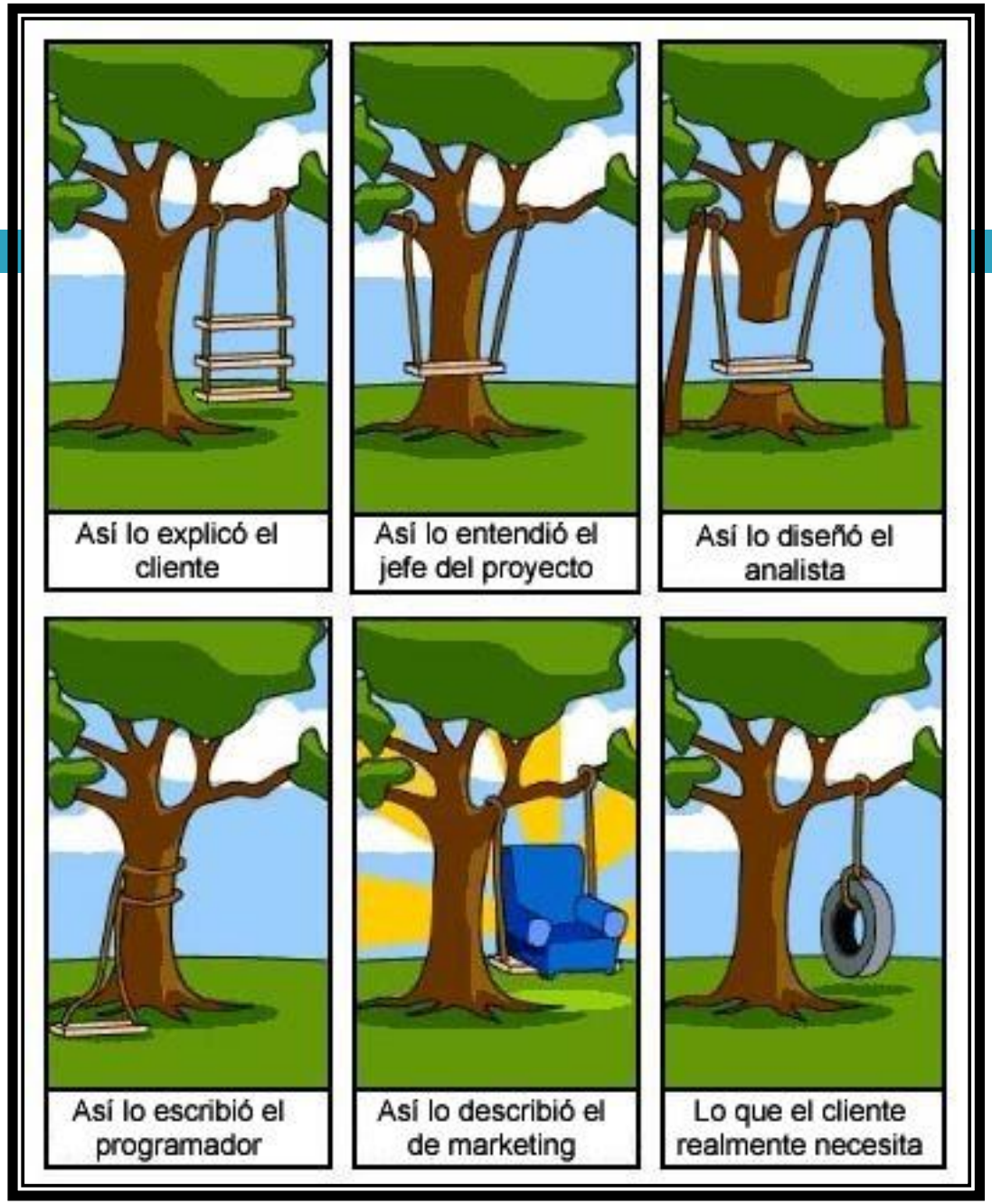
Unidad 4. Álgebras Booleanas y circuitos combinatorios. Propiedades de los circuitos combinatorios. Funciones Booleanas.

Unidad 5. Sistemas de numeración binario. Conversiones. Suma y resta de binarios. Sistema de numeración octal. conversiones. Sistema de numeración hexadecimal.

Unidad 6. Diseño de Algoritmos. La programación como una metodología. Diseño de Algoritmos. Métodos de refinamientos sucesivos. Lenguaje de Diseño de programas. La programación estructurada. Estructuras algorítmicas fundamentales. Formas de reducción de complejidad: secuenciación, análisis por casos, análisis Iterativo. Generalización del concepto de procedimiento: Acciones parametrizadas, funciones.

Unidad N° 6 – Parte 1: Diseño de Algoritmos. La programación como una metodología. Diseño de Algoritmos. Métodos de refinamientos sucesivos. Lenguaje de Diseño de programas. La programación estructurada. Estructuras algorítmicas fundamentales. Formas de reducción de complejidad: secuenciación, análisis por casos, análisis Iterativo.

Algoritmos


Así lo explicó el cliente
Así lo entendió el jefe del proyecto
Así lo diseñó el analista
Así lo escribió el programador
Así lo describió el de marketing
Lo que el cliente realmente necesita

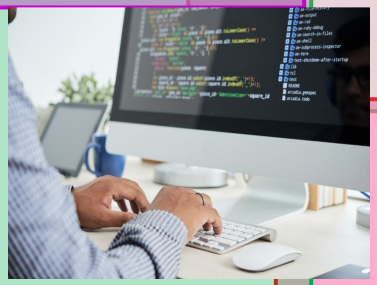
La programación como una metodología

El proceso de programación involucra una secuencia de etapas a cumplir:

Descripción del problema.

Diseño del algoritmo que resuelva el problema.

Codificación en un lenguaje que la computadora reconozca.



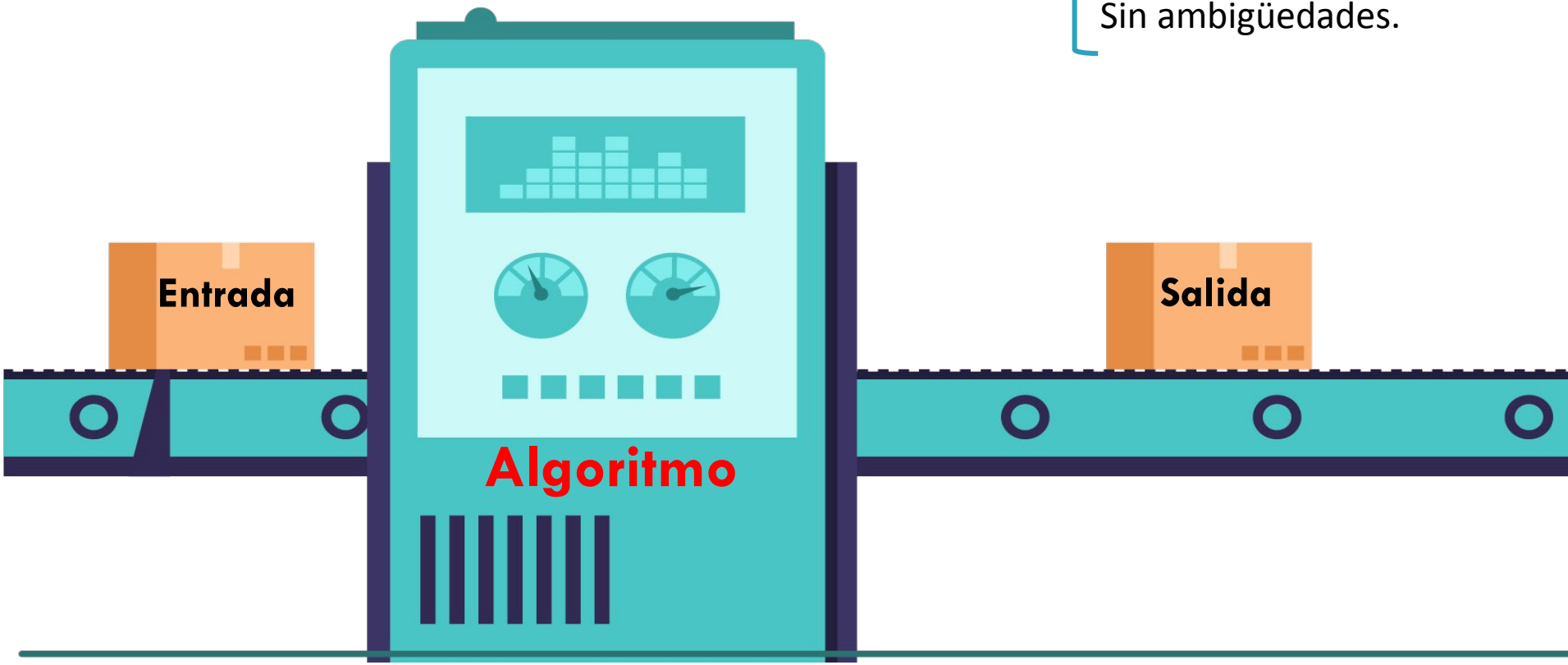
Corrección, pruebas y optimización del programa.

Documentación y mantenimiento del programa.

Descripción del problema

Debe partirse de una especificación del problema que sea:

- Clara.
- Correcta.
- Completa.
- Sin ambigüedades.



Diseño de un algoritmo para resolver el problema

Un Algoritmo es una descripción de la forma en que se debe realizar una tarea o un proceso, en una secuencia finita de pasos que se llevarán a cabo en un tiempo finito. El diseño del algoritmo se realiza usando un PSEUDOCÓDIGO.

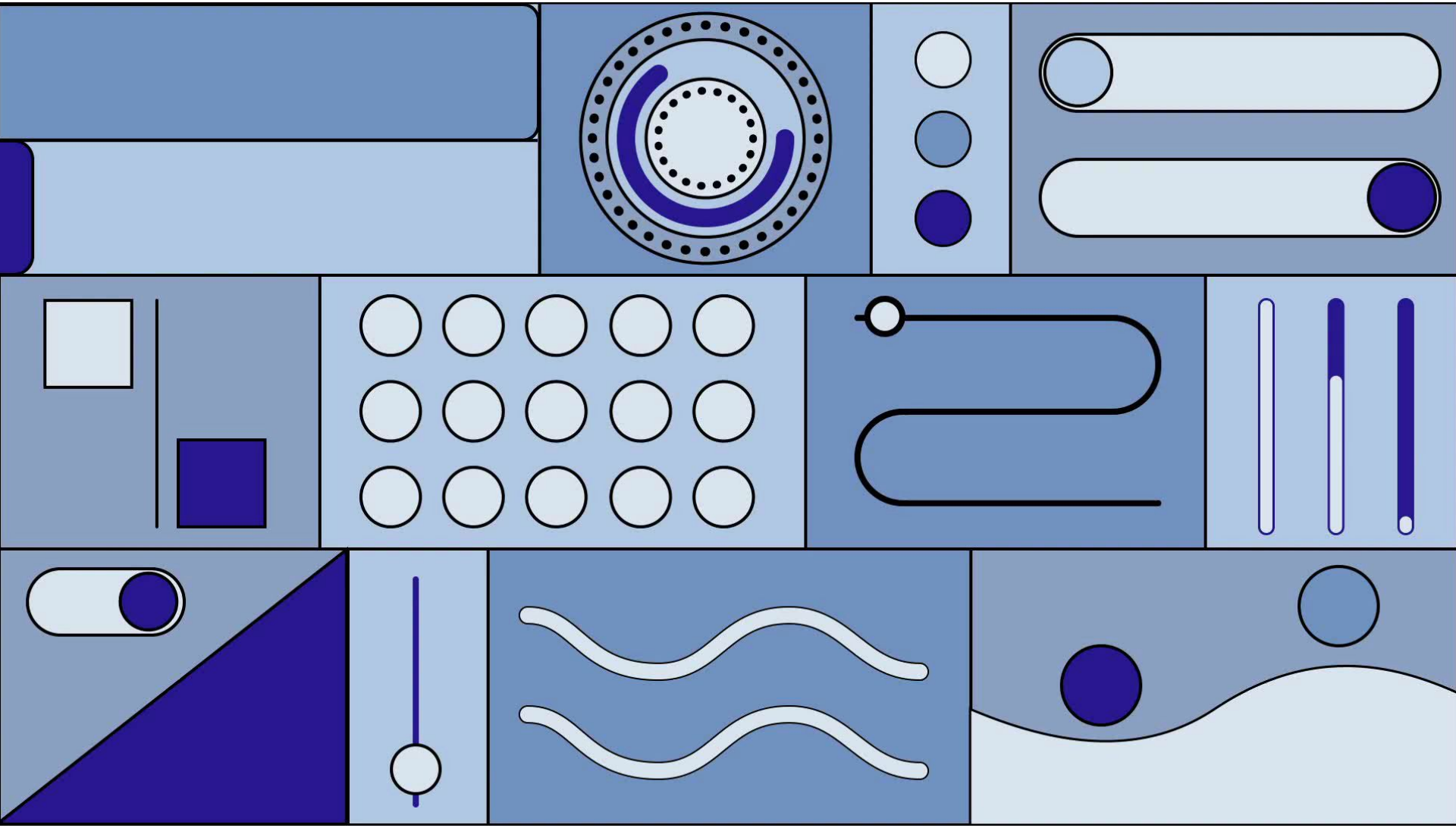
Neutro: Es independiente al lenguaje que se vaya a usar.
Completo: Permite expresar cualquier idea computacional.

Que el algoritmo sea:

Simple, es decir fácil de entender y de escribir.
Eficiente. Buen uso de los recursos de memoria y tiempo del proceso.

Para que usamos un algoritmo?

(https://youtu.be/EkObhToiseo)



Para que usamos un algoritmo?

(https://youtu.be/EkObhTo iseo)

Características del Pseudolenguaje

Cada Algoritmo

- Tiene un nombre que indica que tarea resolverá.

  ALGORITMO Sumar

- Puede tener o no una entrada: datos que necesitamos que el usuario ingrese para realizar una acción u operación.

  ENTRADA: num1, num2: enteros positivos

- Tiene una salida: datos que devuelve nuestro programa, en este caso nuestro algoritmo.

  SALIDA: suma: enteros positivos

- Toda entrada se debe LEER: todos los datos "ingresados" por nuestros usuarios se deben leer.

  LEER(num1, num2)

- Se puede asignar valores a las variables.

  Suma ← num1 + num2

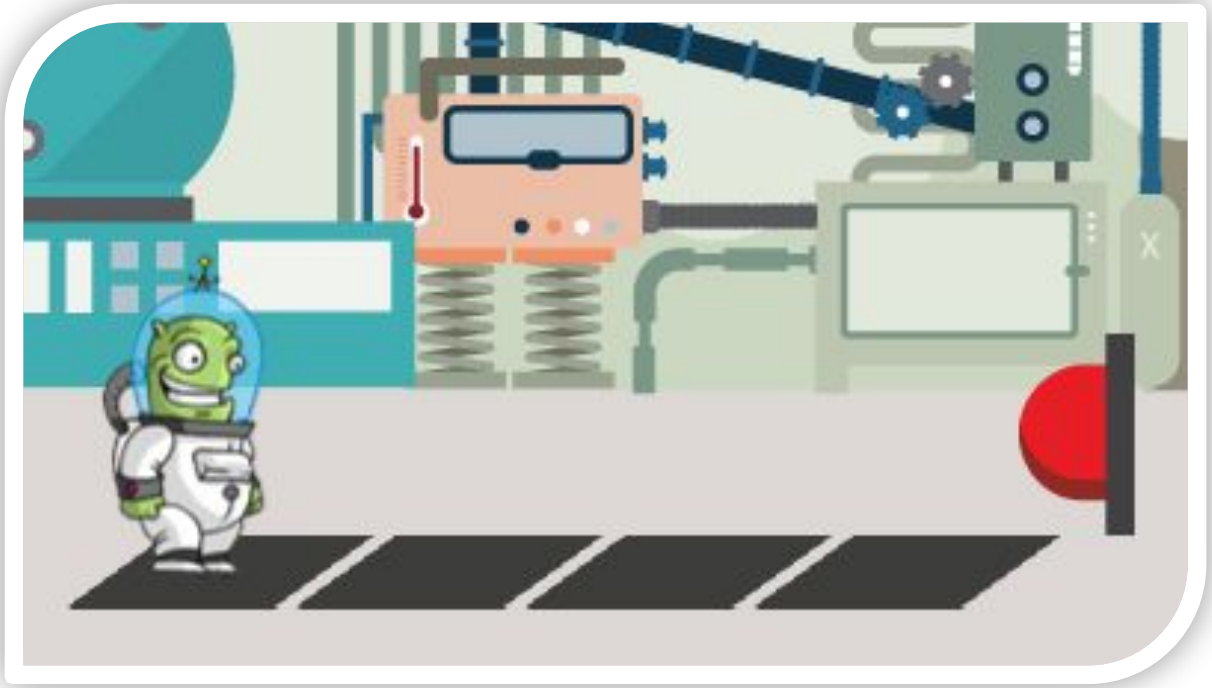
- Toda salida se debe ESCRIBIR: todos los datos que obtenemos de nuestro algoritmo se deben "mostrar".

  ESCRIBIR(suma)

- **PARAR** Para indicar el final de un algoritmo, hacemos uso de la acción primitiva

Veamos un ejemplo:

https://pilasbloques.program.ar/online/#/desafio/1

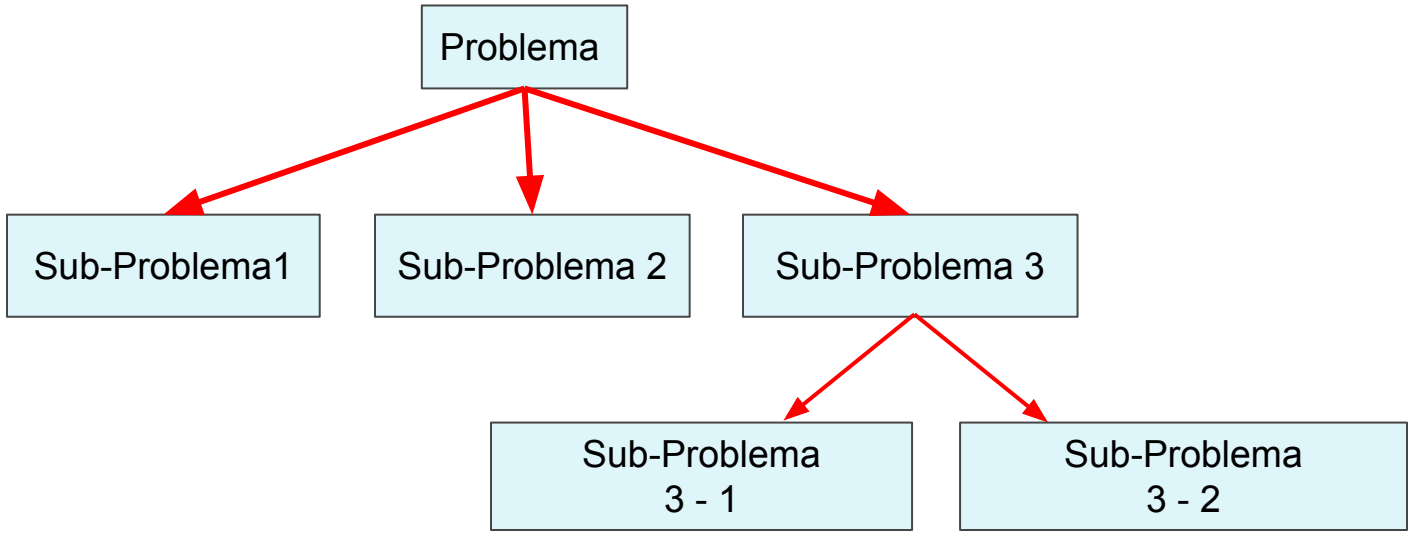


Técnicas de Diseño de Algoritmos

El proceso de adicionar más detalles a una solución de un problema se conoce como **REFINAMIENTO SUCESIVO**.

El método o técnica: **DIVIDE AND CONQUER** con la que se ataca un problema tiene la característica de ser una técnica ***TOP-DOWN.***

Es una estrategia que permite descomponer jerárquicamente un problema largo y complejo en subproblemas más pequeños y fáciles de resolver que el problema de partida.



Veamos un ejemplo de refinamiento:

https://pilasbloques.program.ar/online/#/desafio/2

Estructuras Algorítmicas Fundamentales

La programación es una actividad ordenada y disciplinada y para esto se debe describir adecuadamente el problema que queremos modelar.

Las **estructuras básicas** o fundamentales de la programación estructurada son:

- ✔ **SECUENCIACIÓN**
- ✔ **SELECCIÓN**
- ✔ **ITERACIÓN**

Las estructuras algorítmicas fundamentales permiten realizar un análisis descendente del problema y diseñar una solución por refinamientos sucesivos.

SECUENCIACIÓN

La secuenciación consiste en la descomposición del problema en una secuencia de acciones intermedias.

```
A1;
A2;
A3;
```

- Las acciones se ejecutan de a una a la vez.
- Cada paso se ejecuta una vez.
- Las acciones se ejecutan en el orden en el que están escritas.
- La terminación del último paso implica la terminación de la secuencia.

SECUENCIACIÓN

Ejemplo: Leer dos números (num1 > num2) y calcular la suma y la resta:

```
ALGORITMO Operaciones

ENTRADA: num1,num2:números enteros
SALIDA: sum,resta:números enteros

A1. LEER (num1, num2)
A2. calcular-operaciones
A3. ESCRIBIR(sum,resta)
A4. PARAR
```

Variables

Tipo

Paso a refinar

```
A2. calcular-operaciones
      A21. sum ← num1 + num2
      A22. resta ← num1 – num2
```

El pseudocódigo es más fácil de cambiar y corregir que el código del programa.

ESTRUCTURA DE SELECCIÓN

Descomposición en dos subdominios excluyentes

Son utilizadas para tomar **decisiones lógicas**

```
SI condición  ENTONCES
        A1
SINO
        A2
Fin SI
```

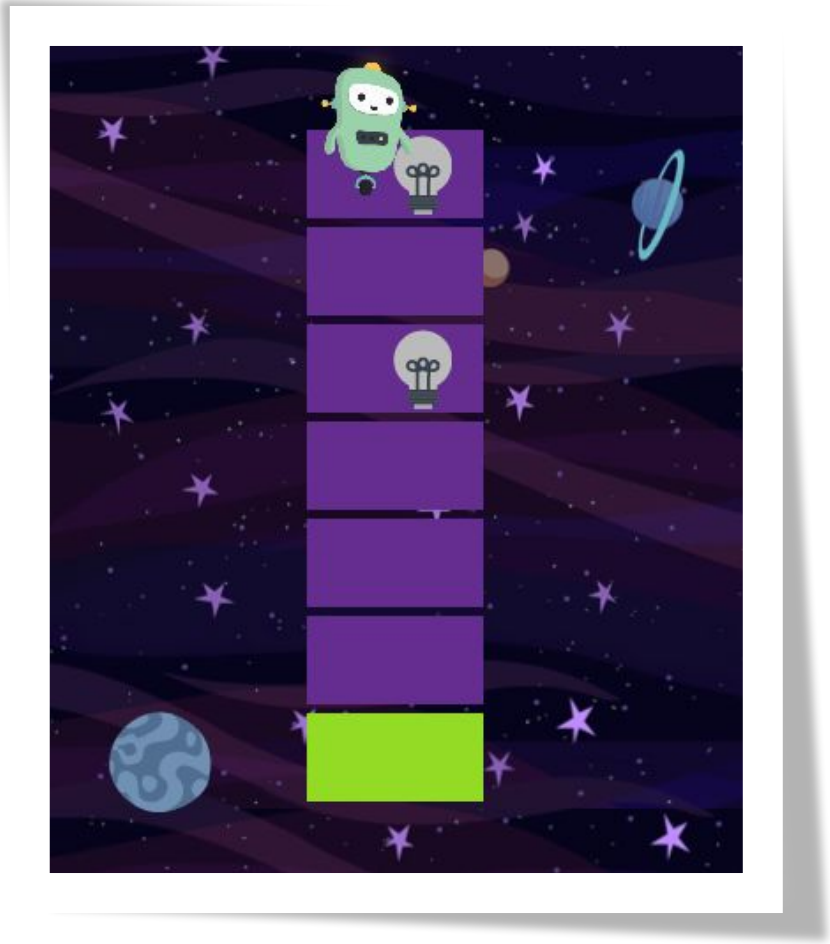
Descomposición en varios subdominios excluyentes

Se utiliza cuando hay más de dos alternativas para elegir. Se evalúa la expresión que puede tomar n valores, según el valor que la expresión tenga en cada momento se ejecutan la/s acción/es correspondiente/s

```
SEGÚN  expresión
      ∈ conjunto de valores1:
      A1;
      ∈ conjunto de valores2:
      A2;
      ∈ conjunto de valores3:
      A3;
      ...
      SINO          : An;
Fin SEGUN
```

Veamos un ejemplo de Selección:

https://pilasbloques.program.ar/online/#/desafio/17



ESTRUCTURA DE SELECCIÓN

Descomposición en dos subdominios excluyentes

Ejemplo: En un juego se lanza un dado si sale 1 o 6 se gana $200, en cualquier otro caso se pierde lo que tenía ganado hasta ese momento

```
Algoritmo DADOS
ENTRADA: dado: número entero positivo.
SALIDA: mensaje: cadena de caracteres.

A1.  LEER(dado)
A2.  SI ( (dado=1) o (dado= 6)) ENTONCES
           ESCRIBIR( "Gana $200")
     SINO
           ESCRIBIR ("Pierde todo")
A3.  PARAR.
```

Por esta rama entraría para los casos 2, 3 , 4, 5

ESTRUCTURA DE SELECCIÓN

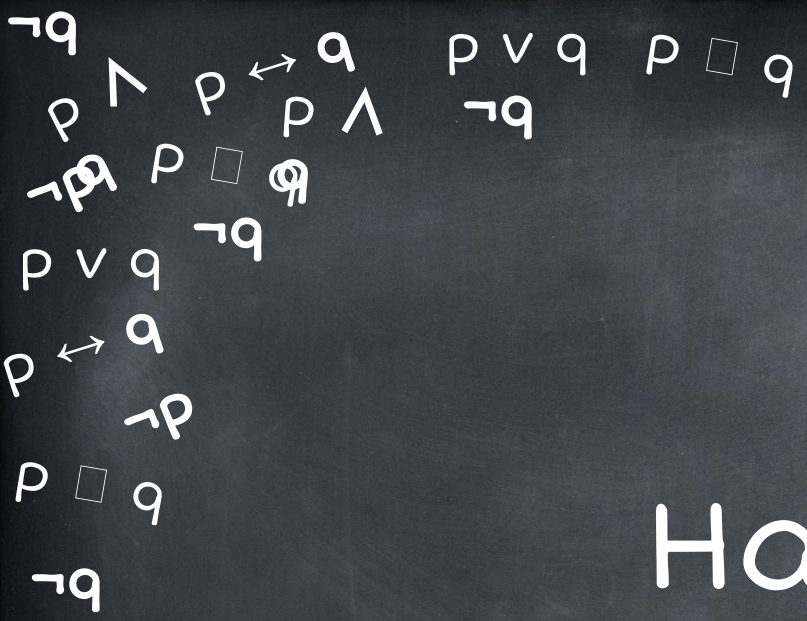
Ejemplo: Calcular el sueldo de un empleado según su antigüedad y su básico.

```
Algoritmo Sueldo

ENTRADA: básico: real positivo, antigüedad: entero positivo.
SALIDA: sueldo: real positivo.

A1. LEER(básico, antigüedad)
A2. SEGÚN antigüedad
       0 ≤ antigüedad < 1   : sueldo  ← básico;
       1 ≤ antigüedad < 5   : sueldo  ← 1.1 * básico;
       5 ≤ antigüedad < 10  : sueldo  ← 1.3 * básico;
       10 ≤ antigüedad < 20    : sueldo  ← 1.5 * básico;
    SINO   : sueldo  ← 2.2 * básico;
A3. ESCRIBIR(sueldo)
A4. PARAR.
```

EJERCICIO

En una librería céntrica los libros tienen un precio registrado, pero el precio final que debe pagarse será teniendo en cuenta lo siguiente: si el libro tiene menos de un año de antigüedad su precio coincide con el registrado. Si tiene entre uno y tres años deberá aplicarse un descuento del 15%. Si tiene más de tres años deberá aplicarse una rebaja del 25%. Además al precio calculado antes se lo debe aumentar en 10% si se paga con tarjeta.

Resuelva el problema usando un algoritmo con al menos dos niveles de refinamiento.



Hasta la próxima clase

p	q	p ∨ q
V	V	V
V	F	V
F	V	V
F	F	F