

# Técnicas de Programación

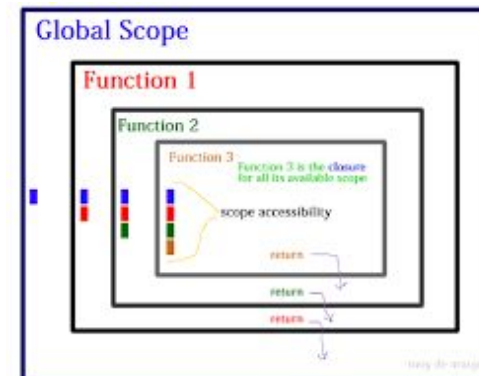
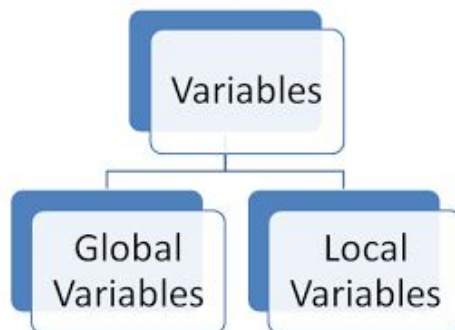
## CFL Programador full-stack

*Ámbito Variable, Legibilidad, Arreglos (Repaso)*

# Ámbito de las Variables

Al utilizar funciones se establece un límite para el alcance de las variables

- **Variables Locales:** Son aquellas que se encuentran dentro de un método. El valor se confina al método en el que está declarada
- **Variables Globales:** Son las que se definen o están declaradas en el algoritmo principal. Pueden utilizarse en cualquier método
- Se debe intentar crear métodos con variables locales y pocos parámetros para favorecer la reutilización y el mantenimiento del software



# Buenas Prácticas de Programación

- Entender el problema, diseñar una estrategia, implementar
- Nombres representativos de variables y métodos
- Código claro, comprensible, etc.
- Identación en las estructuras de control
- Comentarios en el código
- *//Así se comenta en JavaScript, con las dos barras*



# Buenas Prácticas de Programación

- Usar métodos
- No duplicar código
- Dividir el problema en sub problemas
- Construir el código tan simple como sea posible
- Que el código funcione no significa que esté bien programado



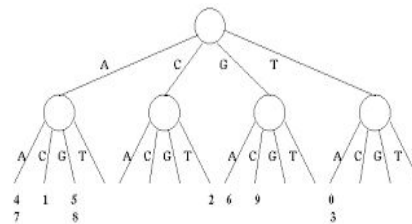
# Estructuras de Datos

## Forma particular de organizar datos



- Estructuras que permiten **COLECCIONAR** elementos

- GUARDARLOS
- RECORRERLOS
- MANIPULARLOS

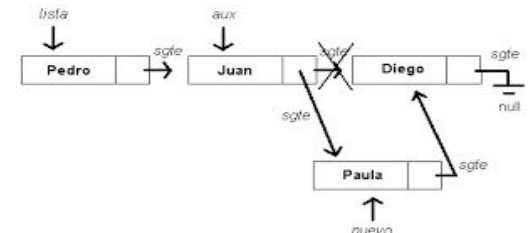


- Estructuras

- **LISTAS**
- **COLAS**
- **PILAS**
- **ARBOLES**

- Operaciones básicas

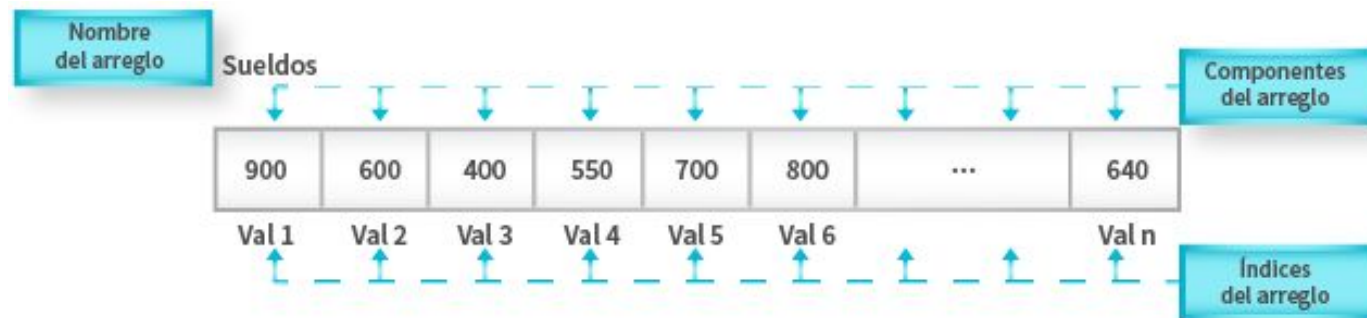
- **COLOCAR**
- **OBTENER**



# Estructuras de Datos

## *Arreglos/Listas/Vectores*

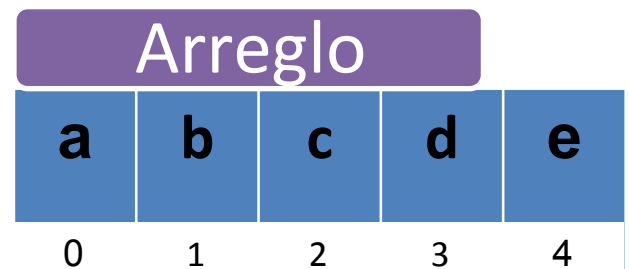
- Los arreglos son estructuras de datos homogéneas (todos sus datos son del mismo tipo)
- Permiten almacenar un determinado número de datos
- Tiene muchos elementos, y a cada uno de ellos se acceden indicando que posición se quiere usar (un índice)



# Estructuras de Datos

## *Arreglos/Listas/Vectores*

- Lista = Array
- Los elementos deben ser del mismo tipo de dato
- Zero-based (arreglos de base cero) -> Índices comienzan en 0
- La cantidad de elementos total = Length será igual al número del último elemento más 1
- Propiedades:
  - ELEMENTO o ITEM: a, b, c, d, e
  - LONGITUD: 5
  - INDICE o SUBINDICE: 0, 1, 2, 3, 4



**Longitud = Length = 5**

# Técnicas de Programación

## CFL Programador full-stack

*Estructuras de Datos y Métodos (Ejercicios)*



# Estructuras de Datos y Métodos

## *Producto Escalar*

- Cargue dos arreglos de dimensión N números (la cantidad es ingresada por el usuario)
- Calcule el producto escalar entre los dos arreglos:

Si  $A < a_1, b_1, c_1 >$  y  $B < a_2, b_2, c_2 >$

El producto escalar entre A y B en función de sus componentes está dado por:

$$A \cdot B = a_1 a_2 + b_1 b_2 + c_1 c_2$$

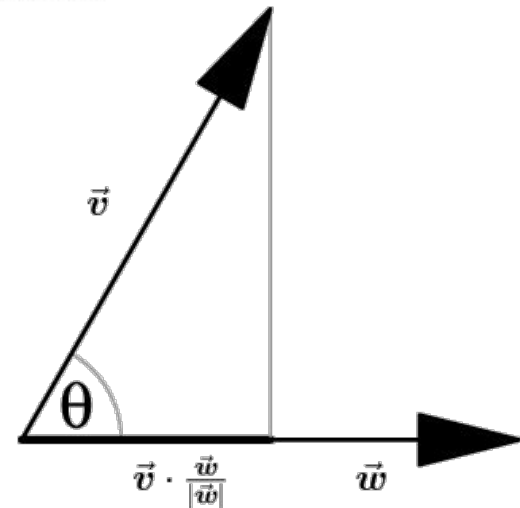
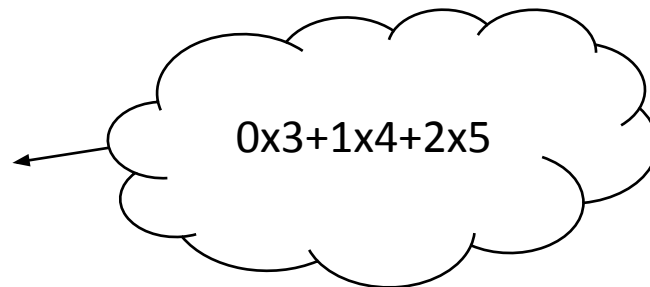
Ejemplo:

n = 3

v1 = 0, 1, 2

v2 = 3, 4, 5

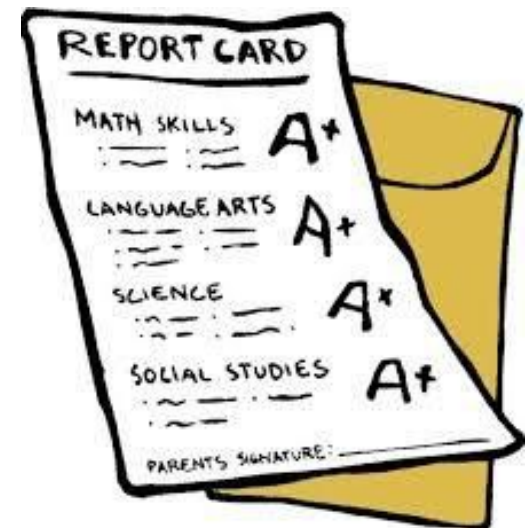
producto = 14



# Estructuras de Datos y Métodos

## *Promedio Escolar*

- Desarrolle un algoritmo que permita cargar alumnos y sus notas en los tres trimestres
- Se debe permitir obtener el promedio anual (es decir, de sus tres notas) de un alumno (ingresado por el usuario)
- Luego de resolverlo, pensar en aprovechar métodos y discutir como representar la información



# Estructuras de Datos y Métodos

## *Cine*

- Diseñar un algoritmo que recorra las butacas de una sala de cine y determine cuántas butacas desocupadas hay
- Suponga que para modelar este problema, se utiliza un arreglo con valores lógicos
  - La presencia de un valor verdadero (true) en el arreglo indica que la butaca está ocupada
  - La presencia de un valor falso (false) en el arreglo indica que la butaca está desocupada



# Estructuras de Datos y Métodos

## *Multiplicación*

- Implemente un método llamado “multiplicarArreglo” que recibe como parámetros tres arreglos de Enteros del mismo tamaño
- Los dos primeros arreglos contienen los números que se quieren multiplicar
- El tercer arreglo almacena el cálculo de la multiplicación de cada posición de los dos arreglos
- Utilice este método para multiplicar los siguientes cuatro arreglos de tres elementos

v1: [1, 2, 3]

v2: [4, 5, 6]

v3: [2, 1, 2]

v4: [1, 2, 1]

vResultado (v1\*v2\*v3\*v4): [8, 20, 36]

aproveche las ventajas de métodos  
para resolver el ejercicio



# Estructuras de Datos y Métodos

## *Jardín*

- El jardín infantil necesita un programa para poder asignar cursos a las aulas
- Se cuentan con tres aulas: **Azul**, **Verde** y **Amarilla**
- Cada aula cuenta con una capacidad diferente (es decir, un número de bancos)
- La aula **Azul** tiene 40 bancos, la **Verde** 35 y la **Amarilla** 30
- Dado un número de infantes ingresado por el usuario, el programa deberá determinar el aula que minimice la cantidad de bancos vacíos
- La salida del algoritmo es el color que identifica al aula asignada

Ejemplo: si la cantidad de personas de un curso es 34, entonces el aula a asignar será la Verde. El aula Amarilla no puede ser asignada porque la cantidad de personas es menor a la cantidad de bancos disponibles. El aula Azul es descartada porque quedan más bancos inutilizados que en el aula Verde (6 versus 1).

