

# CS1112: Programación II

Unidad 3: Punteros

Sesión de Laboratorio - 6A

**Profesores:**

**María Hilda Bermejo** [mbermejo@utec.edu.pe](mailto:mbermejo@utec.edu.pe)

**Estanislao Contreras** [econtreras@utec.edu.pe](mailto:econtreras@utec.edu.pe)

**Jorge Villavicencio** [jvillavicencio@utec.edu.pe](mailto:jvillavicencio@utec.edu.pe)

**Edson Mendiola** [emendiola@utec.edu.pe](mailto:emendiola@utec.edu.pe)

**Ian Paul Brossard** [ibrossard@utec.edu.pe](mailto:ibrossard@utec.edu.pe)

**Jose Chavez** [jchaveza@utec.edu.pe](mailto:jchaveza@utec.edu.pe)

**Julio Yarasca** [jyarascam@utec.edu.pe](mailto:jyarascam@utec.edu.pe)

**Percy Quevedo** [pquevedo@utec.edu.pe](mailto:pquevedo@utec.edu.pe)

**Wilder Nina** [wnina@utec.edu.pe](mailto:wnina@utec.edu.pe)

**José Fiestas** [jfiestas@utec.edu.pe](mailto:jfiestas@utec.edu.pe)

**Material elaborado por:**

**Maria Hilda Bermejo, Ruben Rivas**



# Índice:

- Unidad 3: Punteros
  - Definicion
  - Manejo de memoria dinámica
  - Arreglos dinámicos unidimensionales
  - Arreglos dinámicos bidimensionales

# 3.3

Unidad 3:  
Matrices estáticas (arreglos de  
arreglos)  
Matrices dinámicas

UTEC

# Logro de la sesión:

Al finalizar la sesión, los alumnos podrán:

- Desarrollar programas utilizando punteros.
- Desarrollan programas utilizando matrices dinámicas

# Matrices dinámicas

# Ejemplo 1:

Elabore un programa que solicite las dimensiones de una matriz y permita mostrar un menú con varias opciones:

1. Generar matriz e imprimirla
2. Mostrar sólo números pares
3. Mostrar sólo números impares
4. Fin

Opción 1: Permite solicitar al usuario las dimensiones de la matriz, el valor de las filas y columnas deben ser mayores a 2. Luego se generarán aleatoriamente los valores para casillero de la matriz con valores enteros entre 100 y 200 inclusive. Luego mostrará la matriz.

Opción 2: Permite imprimir la matriz mostrando sólo números pares y en las posiciones donde se encuentran número impares se mostrará el valor cero.

Opción 3: Permite imprimir la matriz mostrando sólo números impares y en las posiciones donde se encuentran números pares se mostrará el valor cero.

## Ejemplo de salida:

```
Ingrese cantidad de filas:6
Ingrese cantidad de columnas:7

      MENU
      ----
      1. Generar matriz aleatoria
      2. Mostrar solo Pares
      3. Mostrar solo Impares
      4. Salir
      Elija opcion (1-4):1
```

142	200	142	158	126	198	145
161	156	122	130	127	197	112
155	189	102	100	113	116	119
180	185	159	164	109	105	106
174	124	150	170	127	136	132
138	194	138	117	115	176	142



```
MENU
----
1. Generar matriz aleatoria
2. Mostrar solo Pares
3. Mostrar solo Impares
4. Salir
Elija opcion (1-4): 2
```

142	200	142	158	126	198	0
0	156	122	130	0	0	112
0	0	102	100	0	116	0
180	0	0	164	0	0	106
174	124	150	170	0	136	132
138	194	138	0	0	176	142

```
MENU
----
1. Generar matriz aleatoria
2. Mostrar solo Pares
3. Mostrar solo Impares
4. Salir
Elija opcion (1-4):3
```

0	0	0	0	0	0	145
161	0	0	0	127	197	0
155	189	0	0	113	0	119
0	185	159	0	109	105	0
0	0	0	0	127	0	0
0	0	0	117	115	0	0

```
MENU
----
1. Generar matriz aleatoria
2. Mostrar solo Pares
3. Mostrar solo Impares
4. Salir
Elija opcion (1-4):4
```

Process finished with exit code 0



## Ejemplo 2:

Elabore un programa que permita al usuario ingresar las dimensiones para una matriz cuadrada (valide que filas y columnas sean iguales) y con esta información el programa realice lo siguiente:

1. Llenar datos a la matriz de manera aleatoria, con números enteros entre 0 y 99.
2. Imprimir la matriz.
3. Rotar las filas de arriba hacia abajo e imprimir la matriz resultante.
4. Rotar las columnas de izquierda a derecha e imprimir la matriz resultante.

## Ejemplo de salida:

Ingrese cantidad de filas:6

Ingrese cantidad de columnas:6

46	87	46	86	42	91
50	27	24	6	98	72
34	24	76	85	9	13
2	25	65	78	27	75
10	89	89	46	75	41
69	62	59	92	65	33

Luego de rotar de arriba hacia abajo, la matriz queda asi:

69	62	59	92	65	33
46	87	46	86	42	91
50	27	24	6	98	72
34	24	76	85	9	13
2	25	65	78	27	75
10	89	89	46	75	41

Luego de rotar de izquierda a derecha, la matriz queda asi:

33	69	62	59	92	65
91	46	87	46	86	42
72	50	27	24	6	98
13	34	24	76	85	9
75	2	25	65	78	27
41	10	89	89	46	75

# Ejemplo 3:

Desarrolle un programa que permita multiplicar dos matrices de números enteros. El programa debe solicitar al usuario que se ingrese el orden de la matriz 1 y de la matriz 2. Debe verificar si matemáticamente es posible realizar la multiplicación de matrices.

Si no es posible el programa termina.

Si se puede realizar la multiplicación de matrices, entonces el programa debe realizar lo siguiente:

- Pedir espacio en el heap para la primera matriz.
- Pedir espacio en el heap para la segunda matriz.
- Generar datos de manera aleatoria para cada matriz.
- Imprimir ambas matrices.
- Realizar la multiplicación de matrices, dimensionando y generando la matriz resultado. Finalmente imprimir la matriz resultado.

Realice el programa utilizando funciones y distribuyendo en código en archivos \*.cpp y \*.h

## Ejemplo 1 de salida :

```
Dimensiones primera matriz
-----
Ingrese cantidad de filas:3
Ingrese cantidad de columnas:3

Dimensiones segunda matriz
-----
Ingrese cantidad de filas:3
Ingrese cantidad de columnas:3
```

```

      MATRIZ 1
-----
      6      3      4
      2      5      6
      3      9      7

      MATRIZ 2
-----
      1      6      5
      6      4      5
      3      2      5

      MATRIZ 3
-----
      36      56      65
      50      44      65
      78      68      95

Process finished with exit code 0
```

## Ejemplo 2 de salida:

```
Dimensiones primera matriz
-----
Ingrese cantidad de filas:3
Ingrese cantidad de columnas:4

Dimensiones segunda matriz
-----
Ingrese cantidad de filas:4
Ingrese cantidad de columnas:3
```

```

      MATRIZ 1
-----
      8      0      6      6
      7      0      5      1
      4      7      1      0

      MATRIZ 2
-----
      1      0      6
      1      9      5
      8      4      7
      1      8      2

      MATRIZ 3
-----
      62      72     102
      48      28      79
      19      67      66

Process finished with exit code 0
```

# Resumen

## **En esta sesión aprendiste a:**

1. Reconocer que es una matriz dinámica y que diferencia hay con una matriz estática.
2. Identificar las partes de una matriz dinámica.
3. Reproducir los pasos para crear arreglos y matrices en la memoria dinámica.
4. Reproducir los pasos para liberar la memoria dinámica en caso de una matriz dinámica.

# Bibliografía:

Deitel. P.J. and Deitel. H. M. (2016) C++ How to Program, Prentice Hall.

Stroustrup, Bjarne (2013). The C++ Programming Language, 4th Addison-Wesley.

Eckel, Bruce, 2000. Thinking in C++, Vol 1: Introduction to Standard C++, 2nd Edition, Prentice Hall



¡Nos vemos en la siguiente  
clase!

