



Facultad Regional  
**UTN** VILLA MARIA

# INSTANCIA EVALUATIVA N\*1

Algoritmos y Estructura de Datos

Profesor:  
Ing. Simieli Paola

Alumno: Federico Martin Villoria  
Legajo: 13906

- 1) La programación limita los procesos de resolución de problemas al ámbito de la informática. Utilizando diferentes lenguajes de código, variables o instrucciones de control como bucles. El pensamiento computacional es universal, ya que nos permite resolver problemas del mundo que nos rodea aplicando los conceptos informáticos. Se centra en la forma en que se aborda un problema, utilizando herramientas y técnicas. La principal diferencia entre el pensamiento computacional y la programación radica en su nivel de abstracción. El pensamiento computacional es un enfoque de alto nivel que se centra en el proceso mental y las estrategias para resolver un problema, mientras que el enfoque de bajo nivel de la programación se centra en la implementación concreta del código.  
Además, el pensamiento computacional es la base conceptual sobre la cual se construye la programación ya que proporciona el marco mental para abordar los problemas de manera sistemática y eficiente.  
Por otro lado, la programación es la implementación práctica de los pasos lógicos y estructurados para resolver un problema. Implica dominar un lenguaje de programación, utilizar una sintaxis y reglas específicas para escribir el código que ejecutara la computadora. La programación es el medio para dar vida a las ideas generadas mediante el pensamiento computacional.

#### Abstracción Procedimental:

- 1) Pedir filas y columnas.
  - a. Leer filas
  - b. Leer columnas
- 2) Al tener esos datos requerimos rellenar la matriz con los números que sea necesario cargar.

Para  $i < -1$  hasta filas hacer  
    Para  $< -j$  hasta columnas hacer  
        Leer Matriz[i,j]  
    FinPara  
FinPara
- 3) El problema nos pide que este procedimiento se debe pedir siempre y cuando no se ingresen 0 como filas y columnas. Entonces para ello necesitamos encapsularlo dentro del ciclo Mientras.

Leer filas  
Leer columnas  
  
Mientras filas  $\neq 0$  y columnas  $\neq 0$  hacer  
    Para  $i < -1$  hasta filas hacer  
        Para  $< -j$  hasta columnas hacer  
            Leer Matriz[i,j]  
        FinPara  
    FinPara

- 4) Ahora debemos realizar el cálculo de si un numero en determinada posición es un punto silla, para eso debemos hacer lo siguiente.

Para i<-1 Hasta filas Hacer

Para j<-1 Hasta columnas Hacer

Si MatrizA[i,j] > maximo\_fila[i] Entonces

maximo\_fila[i] <- MatrizA[i,j];

FinSi

Si MatrizA[i,j] < minimo\_columna[j] Entonces

minimo\_columna[j] <- MatrizA[i,j];

FinSi

FinPara

FinPara

Lo que obtenemos con ese procedimiento es el número máximo de la fila y el número mínimo de la columna, guardándolo en las variables correspondientes.

Para i<-1 Hasta filas Hacer

Para j<-1 Hasta columnas Hacer

Si MatrizA[i,j] > maximo\_columna[j] Entonces

maximo\_columna[j] <- MatrizA[i,j];

FinSi

Si MatrizA[i,j] < minimo\_fila[i] Entonces

minimo\_fila[i] <- MatrizA[i,j];

FinSi

FinPara

FinPara

Este siguiente segmento es la contraposición del problema, en el caso que se obtenga el menor número de la fila y el mayor número de la columna.

- 5) Luego donde estoy mas complicado es en el resultado de si la matriz tiene o no un punto de silla.

contador <- 0;

auxiliar <- 0;

Para i<-1 Hasta filas Hacer

Para j<-1 Hasta columnas Hacer

Si maximo\_fila[i] = minimo\_columna[j] o maximo\_columna[j] = minimo\_fila[i] Entonces

```

                                contador <- contador + 1;
                                FinSi
                                Si maximo_fila[i] <> minimo_columna[j] y maximo_columna[j]
<> minimo_fila[i] Entonces
                                    auxiliar <- auxiliar + 1;
                                    FinSi
                                FinPara
                                FinPara

                                Si contador >= 1 Entonces
                                    Escribir "Si";
                                FinSi
                                Si auxiliar >= 1 Entonces
                                    Escribir "No";
                                FinSi

```

Yo utilizo contadores como bandera, en caso de que si un contador se activa quiere decir que, si hay en alguna posición un punto silla o no, dependiendo de la variable que se active. Teniendo como prioridad "Si".h