



# Sintaxis de un Lenguaje de Alto Nivel

## Taller

### Introducción a la Programación Orientada a Objetos

Cada grupo debe resolver un problema, el cual le será asignado.

**Problema 1:** Realice un programa para jugar triqui. La interfaz del juego se despliega en consola como se muestra a continuación:

```
*****
**      TRIQUI      **
*****
|  1  |  2  |  3  |
-----
1 |  —  |  —  |  —  |
-----
2 |  —  |  —  |  —  |
-----
3 |  —  |  —  |  —  |
-----
Jugador 1 (X)
Ingrese Fila :
Ingrese Columna :
```

Cuando se despliega el tablero se muestra el numero de jugador que tiene el turno y el símbolo que le corresponde (X o O). En cada jugada se solicita la fila y la columna donde desea poner el símbolo. En cada turno se debe validar:

- Si la jugada que ingreso es válida (si la fila y columna esta dentro del rango y si la posición se encuentra vacía). Si no es válida debe volver a solicitar la jugada
- Si la jugada es válida, debe revisar si el jugador ganó

- Si la jugada es válida, debe revisar si ya no hay mas jugadas por realizar

Cuando finalice el juego se puede permitir la opción de guardar el resultado del juego en un archivo de texto. Este **podría** ser un estado final del juego:

```
*****
**      TRIQUI      **
*****
|  1  |  2  |  3  |
+-----+
1 | X  | O  | X  |
+-----+
2 | X  | O  | X  |
+-----+
3 | O  | X  | O  |
+-----+
Juego Empatado
```

Desea guardar el resultado del juego?

Si, escriba (S), No escriba (N):

Escriba nombre del archivo:

**Problema 2:** Realice un programa para jugar ahorcado. El programa cargará la palabra a buscar desde un archivo de texto. La interfaz del juego se despliega en consola como se muestra a continuación:

```
*****
**      AHORCADO      **
*****
*                      *
*                      *
*                      *
*                      *
*                      *
*                      *
*                      *
*                      *
*                      *
*                      *
*                      *
*                      *
*                      *
*****
Palabra: _ _ _ _ _ _ _ _ _ _
*****
```

Ingrese una letra:

Se solicita cada letra por consola y se debe validar:

- Si la letra ingresada es válida (un solo carácter alfanumérico). Si no es válida debe volver a solicitarla.
- Si la letra ingresada corresponde a una letra de la palabra a buscar. Si corresponde debe mostrarla en la interfaz.
- Si la letra no esta debe mostrar una parte del cuerpo (1. cabeza, 2. tronco, 3. brazo derecho, 4. brazo izquierdo, 5. pierna derecha, 6. pierna izquierda).
- Si adivina la palabra debe mostrar un mensaje que diga GANASTE.
- Si ya se mostraron todas las partes del cuerpo y falla, debe mostrar un mensaje que diga PERDISTE.

```
*****
**      AHORCADO      **
*****
*          *****   *
*          *X  X*     *
*          *****   *
*              *     *
*          *****   *
*          *   *   *   *
*        *   *   *   *
*              *     *
*          *   *     *
*          *   *     *
*          *   *     *
*          *   *     *
*****
*          PERDISTE   *
*****
```

**Problema 3:** Realice un programa que simule una caja registradora. La caja registradora almacena ventas, las cuales constan de un código, un valor total vendido, el valor del iva, correspondiente a 16 %, y un conjunto de productos. Cada producto tiene un nombre y un precio. La caja registradora permite las siguientes operaciones:

- Registrar una venta. Para esta operación se debe solicitar la cantidad de productos y registrar cada producto de la venta. El programa calcula automáticamente el valor total de la venta y el iva.
- Consultar una venta por código. Si la consulta retorna una venta, se debe tener la opción de eliminar la venta.
- Consultar todas las ventas dentro una rango de valor total vendido. Al realizar esta operación, se debe permitir guardar el resultado de la consulta en un archivo de texto.

- Consultar todas las ventas dentro de un rango de productos vendidos. Al realizar esta operación, se debe permitir guardar el resultado de la consulta en un archivo de texto.

La interfaz de la aplicación se despliega en consola como se indica a continuación:

```
*****
***** CAJA REGISTRADORA *****
*****
* A) Registrar una venta *
* B) Consultar venta por codigo *
* C) Consultar ventas dentro de rango de valor vendido *
* D) Consultar ventas dentro de rango de productos vendidos *
* E) Salir *
*****
Ingrese la operacion que desea realizar :
```

**Problema 4:** Realice un programa que realice un algoritmo de *block matching*. Este algoritmo recibe un bloque de tamaño  $n \times n$  y busca en una matriz de tamaño  $m \times p$  cual es el bloque mas parecido usando una medida de distorsión. Las medidas de distorsión que debe implementar son:

- **Suma de diferencias absolutas:** La similitud entre un bloque determinado y un bloque en la posición  $(x, y)$  de una matriz es dado por:

$$SAD(x, y) = \sum_{i=0}^{x+n-1} \sum_{j=0}^{y+n-1} |\text{bloque}(i, j) - \text{matriz}(i + x, j + y)| \quad (1)$$

- **Error cuadrático medio:** La similitud entre un bloque determinado y un bloque en la posición  $(x, y)$  de una matriz es dado por:

$$MSE(x, y) = \frac{1}{n \times n} \sum_{i=0}^{x+n-1} \sum_{j=0}^{y+n-1} [\text{bloque}(i, j) - \text{matriz}(i + x, j + y)]^2 \quad (2)$$

El programa retorna la *SAD* menor, la *MSE* menor y las posiciones  $(x, y)$  en la matriz correspondiente a cada medida.

**Problema 5:** Realice un programa para administrar los contactos de una persona. Cada contacto tiene nombre, apellido y número de teléfono. El programa debe realizar las siguientes operaciones:

- Agregar Contacto. Al agregar, debe comprobar que el teléfono corresponda a un número.
- Eliminar Contacto.

- Modificar Contacto. Al modificar, debe comprobar que el teléfono corresponda a un número.
- Buscar contacto por teléfono. La búsqueda puede ser por número parcial, es decir, si se ingresa como valor a buscar el número (3165), debe retornar todos los contactos cuyo número de teléfono comience con esos valores.
- Buscar contacto por nombre. La búsqueda puede ser por nombre parcial, es decir, si se ingresa como valor de búsqueda (.^ng”), debe retornar todos los contactos cuyo nombre comience con estos caracteres.

La interfaz de la aplicación se despliega en consola como se indica a continuación:

```
*****
*****                                CONTACTOS                                *****
*****
* A) Agregar Contacto                                                         *
* B) Eliminar Contacto                                                         *
* C) Consultar Contactos por numero                                           *
* D) Consultar Contactos por nombre                                           *
* E) Salir                                                                     *
*****
Ingrese la operacion que desea realizar :
```

**Problema 6:** Realice un programa que indique si una matriz es un cuadro mágico. Un cuadro mágico es una matriz  $n \times n$  con las siguientes condiciones:

- Contiene los números desde 1 hasta  $n^2$ , todos diferentes entre sí.
- La suma de las filas, las columnas y ambas diagonales deben tener el mismo valor.

El programa carga la matriz desde un archivo de texto con el siguiente formato:

```
n
a(0,0) a(0,1) ... a(0,n)
a(1,0) a(1,1) ... a(1,n)
:      :      :      :
a(n,0) a(n,1) ... a(n,n)
```

donde  $n$  es el tamaño de la matriz y  $a_{(i,j)}$  corresponde al valor de la matriz en la fila  $i$  y la columna  $j$ . El programa debe validar que las dimensiones son correctas, sino debe mostrar un mensaje donde se indique que las dimensiones no concuerdan o que falta información. Al final, en consola se muestra la matriz y si es un cuadro mágico o no.

Escriba el código de C++ que solucione el problema, genere un archivo ejecutable para Windows (.exe) y comprima todo en un archivo .zip. En el campus estará un enlace habilitado para subir el archivo comprimido correspondiente a la solución del problema asignado hasta el **viernes 13 de abril a las 23:55**.