

**Ejercicio de clase:**  
**Análisis de requerimientos, Planeación y Pseudocódigo**

**Ana Sofia Mendez Ramirez**  
**Miguel Angel Moreno Villanueva**  
**Cristian Mateo Alvarez Posada**

**Universitaria Agustiniana**  
**Facultad de Ingeniería**  
**Tecnología en Desarrollo de Software**  
**2022**

### Lista de requerimientos:

1. Solicitarle a player su nombre
2. Mostrar el set de datos en forma de cartón de bingo (Filas y columnas)
3. Dar la opción de iniciar el juego
4. Eliminar los números del cartón que salgan en la "Ruleta"
5. Guardar en una lista aparte los número que se van eliminando}
6. Una vez el bingo se ha eliminado todos los números (Lo que significa que el set ha llegado a 0) se debe mostrar al jugador el orden en que se eliminaron los números.

## Análisis de requerimientos:

### Variables:

- playerName (string). La siguiente variable almacenará el nombre de jugador proporcionado por el usuario del programa.
- gameState (string). La siguiente variable almacenará la elección del jugador respecto al estado del juego, en donde el carácter '0' significa que el juego aun no se debe iniciar y el carácter '1' o cualquier otro carácter significa que el juego puede iniciarse
- i (int). La siguiente variable tiene el fin de funcionar como variable auxiliar a la hora de imprimir los valores recorridos dentro del set.
- j (int). La siguiente variable tiene el fin de funcionar como una variable auxiliar a la hora de almacenar los números del bingo que se eliminaran eventualmente y así asignarlos dentro de una lista.
- setNumbers (set). La siguiente variable permitirá almacenar las 32 posiciones del bingo, dichos números serán proporcionados por el desarrollador
- listDeleted (list). La siguiente variable permitirá el alojamiento en orden de los números que a medida que se ejecuta el código sean eliminados

### Entradas del software:

Teniendo en cuenta los datos proporcionados dentro los requerimientos, se reconoce que se deben realizar únicamente dos procedimientos correspondientes a la entrada ó ingreso de datos por parte del usuario, principalmente para controlar y determinar el rumbo que llevará el programa. Dichas entradas del software corresponden a:

- Ingreso de datos de tipo string, correspondientes al Nombre del jugador.
- Ingreso de datos de tipo string, correspondiente a una elección entre ejecutar o no el resto del programa.

## Procesos del software:

Con el fin de llevar a cabo los requerimientos proporcionados, resulta necesaria la creación de diversos procedimientos, entre los cuales se pueden mencionar:

- Se deberá crear y llenar con 32 datos una colección de datos de tipo set.
- Se tendrá que crear una estructura de control de tipo for, con la cual se busca permitir que se imprima en pantalla los datos contenidos dentro del set con un formato de “tabla” con dimensiones de 8x4.
- Se creará una colección de datos de tipo list para almacenar una serie de datos de manera ordenada.
- Se tendrá que crear una función, con la cual se elimina un dato al azar presente en el bingo, y a su vez envíe el dato eliminado a una variable, para posteriormente ser almacenado dentro de la colección de tipo list mencionada previamente.

## Salidas del software:

Como resultado de los entradas de datos proporcionadas por el usuario y los procesos llevados a cabo internamente, se obtienen una serie de salidas, destacando las siguientes dos salidas del software:

- Se imprimen los 32 datos almacenados en el set, con un formato de dimensiones de 8x4, simulando el aspecto del “cartón de juego” de un bingo.
- Se mostrará en pantalla un pequeño menú en el cual se le preguntará al usuario si desea ejecutar el “programa del bingo” o si por el contrario, prefiere cerrar el programa.
- Se imprime una lista ordenada, la cual presenta de forma concreta el orden en el cual fueron eliminados los datos de la colección de datos de tipo set.

## Planeación del software(¡Bingo!)

1. Mostrar por pantalla un mensaje de bienvenida.
2. Pedir al usuario que ingrese un nombre de jugador para proceder con el “bingo”, dicho dato se deberá almacenar en la variable: playerName (string).
3. Desplegar un pequeño menú, en el cual se le da la opción de elegir entre si ejecutar el programa “Bingo!” (1 o n) o de lo contrario (0) si desea cerrar el programa.
4. Pedir al usuario que ingrese el carácter correspondiente a su elección, dicho carácter se almacenará en la variable: gameState (string).
5. Se creará una colección de datos de tipo set, en la cual se ingresarán 9 números de manera aleatoria, dichos números serán proporcionados por el programador.
6. Se procede a crear un bucle de tipo for acompañado de la variable auxiliar i, con el cual se planea recorrer el set creado previamente, imprimiendo los datos contenidos en el set de modo de que pueda observar un formato de tabla ó cartón de 8x4.
7. A continuación se procede a eliminar un número aleatorio dentro de la colección de datos tipo set, dicho número a eliminar se almacenará en la variable auxiliar j (int)
8. Se crea un colección de datos de tipo list y posteriormente se asigna de manera ordenada (desde el primer número en ser eliminado hasta el último) el valor de cada uno de los número eliminados del set.
9. Una vez que se haya eliminado el último número del bingo, se procede a imprimir en pantalla la lista de números eliminados del bingo, conservando el orden de eliminación presentado a lo largo del “juego”

## Pseudocódigo(¡Bingo!)

Algoritmo bingo

    Escribir "Bienvenido a ¡Bingo!, donde podrá participar por asombrosos premios jugando bingo con nosotros"; // Dar la bienvenida al usuario a nuestro juego

    Escribir "Escriba aquí su nombre para empezar a jugar! "; // Mostrar por consola un mensaje que invite al usuario a ingresar su nombre para poder empezar

    Leer playerName; // Leer el nombre del usuario

// Crear un set llamado setNumber que almacenará las 32 posiciones del bingo

    Dimension setNumber[32];

// Llenar el bingo con las 32 números del bingo

```
setNumber[1] = 47;  
setNumber[2] = 62;  
setNumber[3] = 95;  
setNumber[4] = 3;  
setNumber[5] = 6;  
setNumber[6] = 14;  
setNumber[7] = 24;  
setNumber[8] = 46;  
setNumber[9] = 63;  
setNumber[10] = 37;  
setNumber[11] = 4;  
setNumber[12] = 89;  
setNumber[13] = 56;  
setNumber[14] = 43;  
setNumber[15] = 17;  
setNumber[16] = 22;  
setNumber[17] = 76;
```

```
setNumber[18] = 11;  
setNumber[19] = 43;  
setNumber[20] = 38;  
setNumber[21] = 10;  
setNumber[22] = 93;  
setNumber[23] = 33;  
setNumber[24] = 96;  
setNumber[25] = 27;  
setNumber[26] = 54;  
setNumber[27] = 75;  
setNumber[28] = 83;  
setNumber[29] = 23;  
setNumber[30] = 100;  
setNumber[31] = 15;  
setNumber[32] = 18;
```

```
// Imprimir por consola un saludo para el usuario y el cartón con el que el usuario jugará bingo
```

```
    Escribir "Hola ", playerName, " ¿Cómo estás?";  
    Escribir "Este es su cartón de juego con 32 posiciones!"
```

```
// Se hace un ciclo "for" para recorrer el set y poder mostrar por pantalla el cartón de juego
```

```
    Para i <- 1 Hasta 32 Con Paso 4 Hacer
```

```
        Escribir setNumber[i], " ", setNumber[i + 1], " ", setNumber[i + 2], " ", setNumber[i + 3];
```

```
// Se imprime el cartón de 8x4 en consola
```

```
    FinPara
```

```
    Dimension listDeleted[32]; // Declaración de una lista que almacenará los números eliminados
```

```
// Se le preguntará al usuario si desea empezar a jugar o no
```

```
    Imprimir "Está listo para empezar el bingo?"
```

```
    Imprimir "(1) Si"
```

```
Imprimir "(0) No: "  
Leer gameState;
```

```
// Se imprime por consola que se empezará el juego y se mencionan acotaciones importantes para jugar
```

```
Imprimir "Comencemos a jugar!";  
Imprimir "IMPORTANTE: Recuerde que los números que irán siendo eliminados serán representados en el cartón como un cero (0)"  
Imprimir ""
```

```
// Se hace un bucle while para comprobar que el usuario quiera seguir jugando o no
```

```
Mientras gameState = "1" Hacer
```

```
// Se hace un bucle for que hará que se "recorra" todo el set y de esta forma se irán eliminando números del setNumber
```

```
Para i <- 1 Hasta 32 Con Paso 1 Hacer
```

```
Imprimir "El número que está jugando es: ", setNumber[i]; // Se muestra por consola el número que será eliminado
```

```
listDeleted[i] = setNumber[i]; // Se almacena el número del set en la lista de los números eliminados
```

```
setNumber[i] = 0; // Y el número que fue eliminado pasa a ser cero para que sea más visible su eliminación en el cartón
```

```
Imprimir ""; // Se imprime como quedó el cartón con el nuevo cambio
```

```
Imprimir "Su cartón quedó de la siguiente manera: "
```

```
Para j <- 1 Hasta 32 Con Paso 4 Hacer
```

```
Escribir setNumber[j], " ", setNumber[j + 1], " ", setNumber[j + 2], " ", setNumber[j + 3];
```

```
// Se imprime el cartón de 8x4 en consola
```

```
FinPara
```

```
// Se le pregunta al usuario si quiere seguir participando del bingo
```

```
Imprimir "Desea continuar jugando?"
```

```
Imprimir "(1) Si"
```

```
Imprimir "(0) No: "
```

```
Leer gameState; // Se almacena la decision del usuario en la variable "gameState"
```

FinPara

// Se utiliza un if para comprobar si el juego ya termino

Si setNumber[32] = 0 Entonces

gameState = "0" // Se iguala la variable "gameState" a 0 ya que el juego ya terminó

Imprimir "El orden de los números eliminados fue:" // Se imprime por consola que le jugador ganó el bingo

Para j <- 1 Hasta 32 Con Paso 1 Hacer

Escribir listDeleted[j], " ", listDeleted[j + 1], " ", listDeleted[j + 2], " ", listDeleted[j + 3]; // Se imprime el cartón de

8x4 en consola

FinPara

Fin Si

Fin Mientras

FinAlgoritmo



