

# Proyecto Final de Informática

Julian Rodríguez Vega, Dayanna Lugo Vargas y Maria Fernanda Marin

Agosto 2021

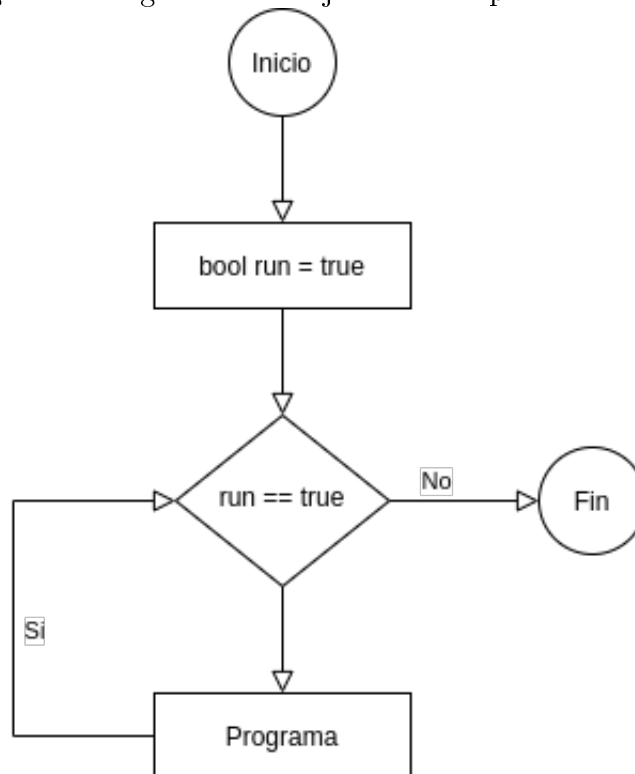
## 1. Menú

Iniciando con el menú de inicio, con un ciclo `do while` o un ciclo `while`, junto a una variable de tipo booleana, es posible programar un programa por consola que pueda continuar corriendo constantemente.

```
int main(){  
    bool run = true;  
    do {  
        // Programa  
    } while(run);  
    return 0;  
}
```

Dentro de este ciclo sera necesario cambiar el valor de `run`, para salir del ciclo y terminar la ejecución del programa.

Figura 1: Diagrama de Flujo del Ciclo para un Menú



Con una variable programa se captura que programa el usuario desea ejecutar, esta variable es evaluada con un switch envez de agrupar multiples condicionales.

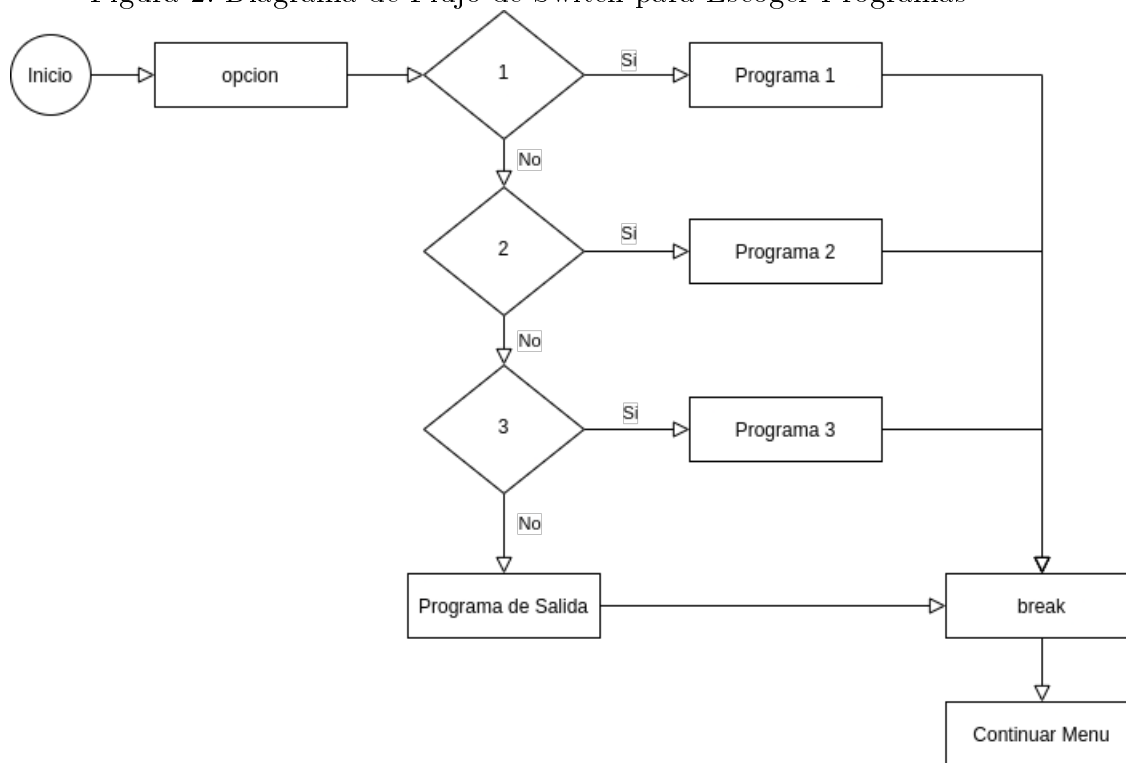
```
cout << "Programas_Disponibles" << endl;
cout << "1._Calculador_de_Promedio" << endl;
cout << "2._Contador_de_Digitos." << endl;
cout << "3._Sucesión_Numerica." << endl;

int programa;
cout << "Ingrese_una_opción:_";
cin >> programa;

switch(programa) {
    case 1:
        // Programa Calculador de promedio
        break;
    case 2:
        // Programa Contador de Digitos
        break;
    case 3:
        // Programa Susesión Numerica
        break;
    default:
        cout << "Esa_no_es_una_opción_valida";
}
```

En el caso `default` cuando el usuario no ingrese una de las 3 opciones, sera entonces dirigido a un ciclo para terminar la ejecución del programa. Cuando termine de ejecutarse el `switch`, el ciclo del menu inicial continuara ejecutando y el usuario podra escoger otro programa por ejecutar.

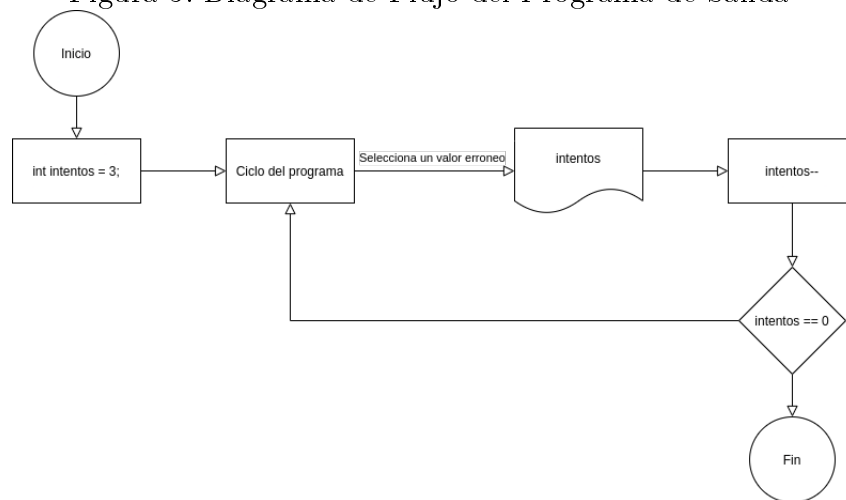
Figura 2: Diagrama de Flujo de Switch para Escoger Programas



## 2. Programas

Según la opción seleccionada, se ejecutara uno de los siguientes 3 programas, adicionalmente se solicita proporcionar un máximo de 3 intentos en caso de que un usuario ingrese un valor erróneo. Para este programa de salida se hace uso de un contador declarado antes del ciclo principal del programa, en caso de que el usuario ingrese un valor no definido en el switch de selección sera enviado al caso default, en donde se reduce el contador de intentos y se muestra una advertencia. Si el usuario gasta sus 3 intentos, la variable run seria modificada a false y el programa terminara de ejecutar. Para facilitar el uso del programa, este contador es reiniciado cuando el usuario selecciona cualquiera de los programas mostrados.

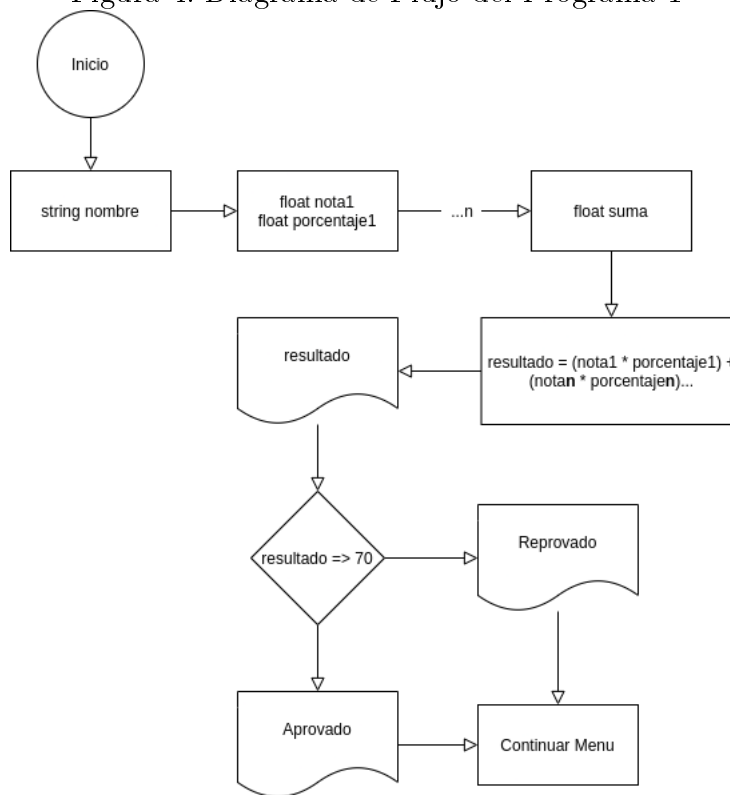
Figura 3: Diagrama de Flujo del Programa de Salida



## 2.1. Calculador de Promedio Ponderado

Este programa inicia solicitando información sobre el estudiante, luego pide un valor y un porcentaje para 6 instrumentos. Una vez se obtienen los 12 datos necesarios, el promedio ponderado es calculado directamente en una variable de tipo float llamada resultado. Finalmente se verifica si la variable resultado es mayor o igual a 70 para mostrar que el estudiante aprobo el curso o no.

Figura 4: Diagrama de Flujo del Programa 1



```
// Programa Calculador de promedio
string nombre;

cout << "Ingrese_un_nombre:_ " << " ";
cin >> nombre;
```

```

float notal;
float porcentaje1;
float resultado;

// Obtener valores
cout << "-_Ingrese_un_valor_para_el_primer_instrumento:_";
cin >> notal;
cout << "+_Ingrese_un_porcentaje_para_el_primer_instrumento:_";
cin >> porcentaje1;

// Calcular promedio ponderado
// Fuente: https://es.wikihow.com/calcular-el-promedio-ponderado
resultado = (notal * porcentaje1/100) + //resto de notas y porcentajes;

cout << endl;
cout << "Resultados:_ " << endl;
cout << "Calificación_final_es_" << resultado << endl;

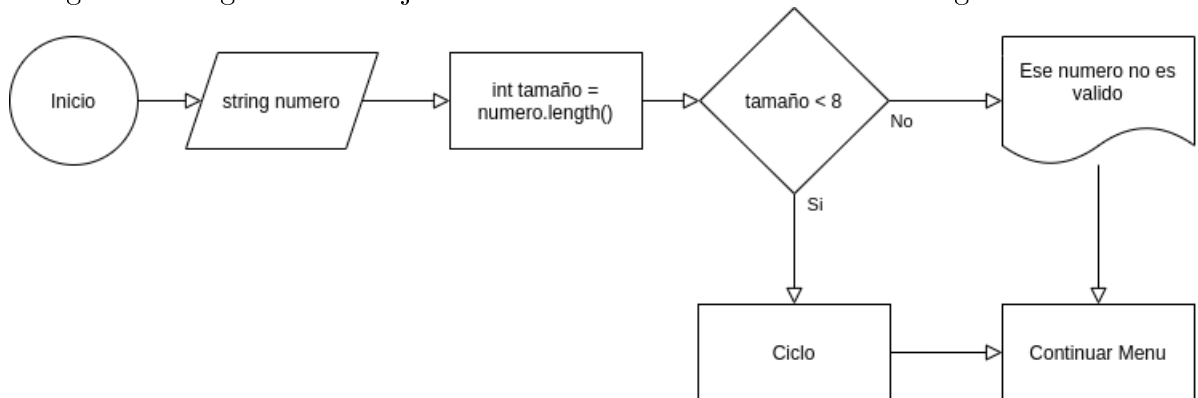
// Evaluar resultado
if(resultado >= 70) {
    cout << nombre << "_ha_aprobado_el_curso!" << endl;
} else {
    cout << nombre << "_ha_reprobado_el_curso" << endl;
}

```

## 2.2. Programa Contador de Dígitos

El segundo programa en el enunciado solicita mostrar la cantidad de dígitos en un número y la suma de los mismos, este valor puede ser almacenado en una variable de tipo `string` la cual se encuentra compuesta por un conjunto de caracteres sobre los cuales se puede iterar. Para revisar la cantidad de caracteres dentro de una cadena de texto es posible utilizar la función `length()`, la cual es guardada en una nueva variable y usada para validar la cantidad de dígitos en el número con un condicional, en donde, de ser un número válido se continuará con el ciclo sino se mostrará un mensaje de error y el usuario será dirigido al menú principal.

Figura 5: Diagrama de Flujo de la Validación en el Contador de Dígitos



Para este ciclo se utilizará un ciclo `for` y una variable temporal para guardar el valor del carácter por el cual se está iterando en el ciclo, puesto que esta variable será de tipo `string`, será necesario convertirla a `integer` con la función `stoi` y así incrementar la variable con la suma de los dígitos.

```
cout << "Contador_de_Dígitos" << endl;
string numero;
// Capturar la entrada
cout << "Ingrese_un_numero:" << " ";
cin >> numero;
int tamaño = numero.length();
// Con la funcion length es posible obtener el tamaño de una cadena de t
if(tamaño < 8) {
    // Variables acumuladores
```

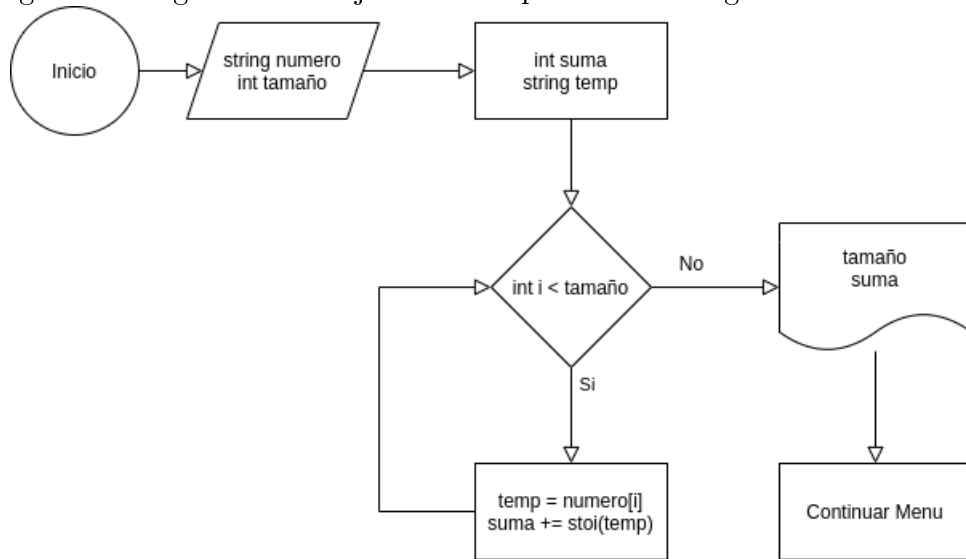


```

int suma = 0;
// Variable temporal para guardar el valor de un item dentro de la
string temp;
// Iterar por cada item en la cadena de texto
for(int i = 0; i < tamaño; i++){
    // Guardar el valor actual
    temp = numero[i];
    suma = suma + stoi(temp);
}
// Mostrar resultados
cout << "_Suma_de_los_digitos:_ " << suma << endl;
cout << "_Cantidad_de_digitos:_ " << tamaño << endl;
} else {
    cout << "Ese_numero_no_es_valido" << endl;
}

```

Figura 6: Diagrama de Flujo del Ciclo para Sumar Dígitos en un Numero



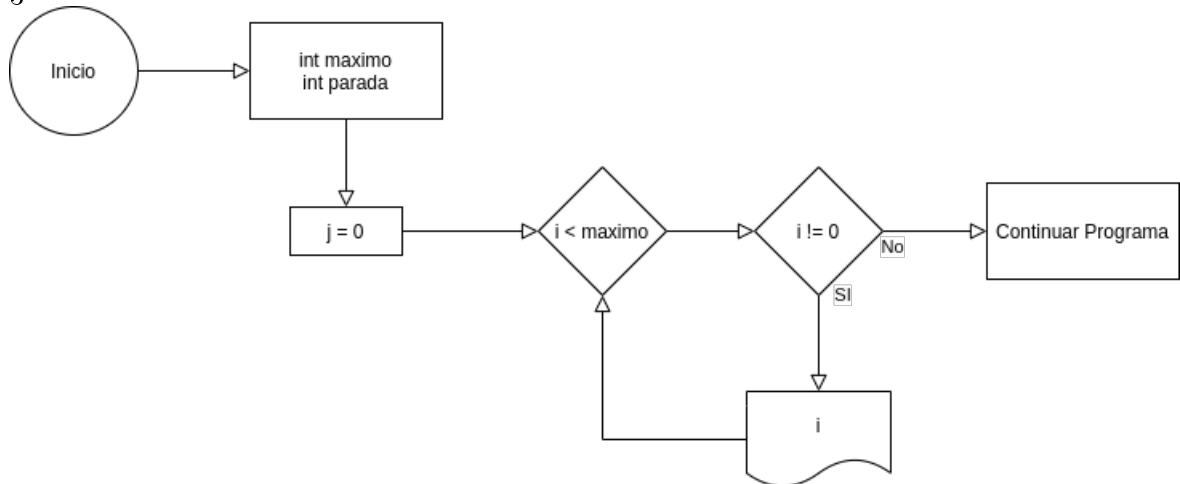
### 2.3. Sucesión Numérica

La sucesión numérica busca imprimir una sucesión números hasta una "parada", en donde se imprimaran los números ya recorridos y los siguientes números en la sucesión hasta llegar a la siguiente parada o hasta llegar a 20. Primero se inicia verificando si el numero actual por imprimir no es 0, esto se hace para evitar imprimir únicamente 0 y no romper con la secuencia numérica.

```
const int maximo = 20;  
// Indicador de cada cuanto se reinicia la secuencia  
const int parada = 5;  
for(int i = 0; i < maximo; i++) {  
// No imprimir la primera iteración de 0  
    if(i != 0){  
        // Imprimir numero  
        cout << i << " ";  
    }  
    // Imprimir numeros hasta llegar a un multiplo de la variable parada  
}  
cout << endl;
```

Figura 7: Diagrama de Flujo del Ciclo y Condicional Iniciales del Programa

3



Mediante una variable **integer** llamada parada, se revisa si un numero es múltiplo de este, para imprimir los números previos a este, esto es logrado mediante el operador **modulus** en un condicional.

```
// Imprimir todos los numeros hasta el numero actual
// Si el numero actual es multiplo de la variable parada
if((i %parada) == 0) {
    cout << endl;
    // Imprimir todos los numeros hasta el numero actual
    for(int j = 0; j < i+1; j++) {
        cout << j << " ";
    }
}
```

Figura 8: Diagrama de Flujo del Ciclo para Imprimir Todos los Números Antes de un Numero "x"

