

 ESCUELA POLITÉCNICA NACIONAL Programación I	LABORATORIO No. 3
NOMBRE: Danny Paúl Ponce Guacalés.	FECHA DE ENTREGA: 20/06/2025

PRÁCTICA DE LABORATORIO No. 3
TEMA: Estructuras de Control y Repetición

1. OBJETIVOS

- Desarrollar un programa en C++ que implemente principios de Código Limpio.
- Gestionar una encuesta de artistas favoritos para un festival de música mediante un menú interactivo.
- Utilizar arreglos estáticos, estructuras de control y repetición.
- Garantizar un código legible, modular y robusto.

2. TEORÍA

En la práctica 3 de laboratorio, se utilizaron estructuras de control como if-else para la toma de decisiones, así como las validaciones y la estructura switch para gestionar las opciones del menú principal del programa. Las estructuras de repetición (for, while, do-while) fueron esenciales para recorrer los arreglos, con el fin de dar validación continua de entradas de usuario hasta que fueran correctas, y para el bucle principal del menú interactivo. Los arreglos estáticos permitieron almacenar colecciones de datos de tamaño fijo, como los nombres de los artistas, sus géneros musicales y sus respectivos votos. Finalmente, la implementación se basó en el uso de funciones modulares, lo cual es una característica clave del "Código Limpio", permitiendo que el programa sea más legible, fácil de mantener y reutilizable.

3. DESARROLLO

1. Lo primero que se realizó fue definir las estructuras de datos necesarias para almacenar la información de los artistas, incluyendo incluso un enum para los diferentes géneros musicales y una constante para el número máximo de artistas permitido. Los datos de los artistas (nombres, géneros y votos) se almacenaron en arreglos estáticos.
2. Luego, para una mayor organización en el código, de esta manera aplicando los principios del clean code, el programa se dividió en cinco funciones principales, cada una con un objetivo específico, para mejorar la modularidad y legibilidad del código.

 <p>ESCUELA POLITÉCNICA NACIONAL Programación I</p>	<p>LABORATORIO No. 3</p>
<p>NOMBRE: Danny Paúl Ponce Guacalés.</p>	<p>FECHA DE ENTREGA: 20/06/2025</p>

3. Luego de crear estas 5 funciones se creó una función main que actúa como el punto de entrada del programa, la cual esta encargada de inicializar los artistas y presenta un menú interactivo al usuario, que se repite hasta que el usuario elige salir.
4. Se implementaron validaciones robustas para garantizar que las entradas del usuario sean correctas. Esto incluyó la verificación de nombres vacíos o duplicados, el cumplimiento del límite de artistas (20), y que las opciones de menú, género y voto estén dentro de los rangos válidos. Se utilizaron cin.clear() y cin.ignore() para manejar entradas inválidas y limpiar el buffer de entrada, evitando errores inesperados para ello se investigó sobre estas funciones.

4. PROGRAMA IMPLEMENTADO

```

Encuesta_de_artistas.cpp > ...
1 #include <iostream>
2 #include <string>
3 #include <cctype> // Para tollover el cual convierte mayusculas en minusculas
4 #include <limits> // Para std::numeric_limits<int>
5
6 using namespace std;
7
8 // Definición del enum para los géneros musicales
9 enum GeneroMusical {
10     Ninguno = 0,
11     Pop = 1,
12     Cumbia = 2,
13     Salsa = 3,
14     Rock = 4,
15     Reggaeton = 5,
16     Otro = 6
17 };
18
19 const int MAXIMO_ARTISTAS = 20; // Límite de artistas
20
21 void inicializarArtistas(string nombres[], GeneroMusical generos[], int votos[], int& numero_de_artistas);
22 bool esNombreValido(const string& nombre, const string nombres[], int numero_de_artistas);
23 void registrarArtista(string nombres[], GeneroMusical generos[], int votos[], int& numero_de_artistas); //
24 void registrarVoto(const string nombres[], int votos[], int numero_de_artistas); //
25 void mostrarEstadisticas(const string nombres[], const GeneroMusical generos[], const int votos[], int numero_de_artistas); //

```

Figura 1. En la imagen se muestra la declaración del tipo enumerado GeneroMusical para clasificar los artistas, la constante que define el límite máximo de artistas en el sistema, la inclusión de bibliotecas que nos servirán para el programa y los prototipos de las funciones.

 <p>ESCUELA POLITÉCNICA NACIONAL Programación I</p>	LABORATORIO No. 3
NOMBRE: Danny Paúl Ponce Guacalés.	FECHA DE ENTREGA: 20/06/2025

```
// Función para inicializar artistas predefinidos
void inicializarArtistas(string nombres[], GeneroMusical generos[], int votos[], int& numero_de_artistas){
    // Usar índices del 0 al 4 para los 5 artistas predefinidos
    nombres[0] = "Bad Bunny";
    generos[0] = Reggaeton;
    votos[0] = 0;

    nombres[1] = "Karol G";
    generos[1] = Reggaeton;
    votos[1] = 0;

    nombres[2] = "Shakira";
    generos[2] = Pop;
    votos[2] = 0;

    nombres[3] = "Juanes";
    generos[3] = Rock;
    votos[3] = 0;

    nombres[4] = "Marc Anthony";
    generos[4] = Salsa;
    votos[4] = 0;

    numero_de_artistas = 5; // Establece el número inicial de artistas
}
```

Figura 2. Muestra la implementación de la función encargada de cargar los 5 artistas predefinidos en el sistema, incluyendo sus nombres, géneros iniciales y votos en cero.

```
// Función para validar el nombre de un artista
bool esNombreValido(const string& nombre, const string nombres[], int numero_de_artistas){
    // Valida si el nombre está vacío
    if(nombre.empty()){
        cout << "Debe ingresar un nombre." << endl;
        return false;
    }

    // Convierte el nombre ingresado a minúsculas para comparación
    string nombre_artistas_minusculas = "";
    for(char a : nombre){
        nombre_artistas_minusculas += tolower(a);
    }

    // Compara con los nombres existentes (ignorando mayúsculas/minúsculas)
    for(int i = 0; i < numero_de_artistas; ++i){
        string nombre_artista_predefinido_minusculas = "";
        for(char a : nombres[i]){
            nombre_artista_predefinido_minusculas += tolower(a);
        }
        if (nombre_artistas_minusculas == nombre_artista_predefinido_minusculas){
            cout << "El artista " << nombre << " ya existe." << endl;
            return false;
        }
    }
    return true; // Nombre válido
}
```

Figura 3. En la figura se muestra la lógica de la función utilizada para validar los nombres de los artistas, verificando que no estén vacíos y que no sean duplicados de artistas ya existentes, realizando una comparación que ignora mayúsculas y minúsculas.



ESCUELA POLITÉCNICA NACIONAL
Programación I

LABORATORIO
No. 3

NOMBRE: Danny Paúl Ponce Guacalés.

FECHA DE
ENTREGA:
20/06/2025

```
// Función para registrar un nuevo artista
void registrarArtista(string nombres[], GeneroMusical generos[], int votos[], int& numero_de_artistas){
    // Valida si se ha alcanzado el límite de artistas
    if(numero_de_artistas >= MAXIMO_ARTISTAS){
        cout << "No se pueden registrar más artistas. Se alcanzó el límite de " << MAXIMO_ARTISTAS << "." << endl;
        return;
    }

    string nuevo_artista;
    cout << "Ingrese el nombre del artista que desea agregar: ";
    // Limpia el buffer de entrada antes de leer con getline (importante después de un cin >> numero)
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    getline(cin, nuevo_artista); // Lee la línea completa para nombres con espacios

    // Bucle para solicitar un nombre válido hasta que se cumpla la validación
    while (!esNombreValido(nuevo_artista, nombres, numero_de_artistas)){
        cout << "Por favor, ingrese un nombre valido: ";
        getline(cin, nuevo_artista); // Vuelve a leer el nombre
    }

    // Solicitud y valida el género musical
    int opcion_de_genero;
    do {
        cout << "Seleccione el género musical:" << endl;
        cout << "1. POP" << endl;
        cout << "2. CUMBIA" << endl;
        cout << "3. SALSA" << endl;
        cout << "4. ROCK" << endl;
        cout << "5. REGGAETON" << endl;
        cout << "6. OTRO" << endl;
        cout << "Ingrese el número del género (1-6): ";

        // Valida que la entrada sea un número y esté en el rango 1-6
        while (!(cin >> opcion_de_genero) || opcion_de_genero < 1 || opcion_de_genero > 6) {
            cout << "Entrada invalida. Por favor, ingrese un número entre 1 y 6: ";
            cin.clear(); // Limpia el estado de error de cin
            cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Ignora el resto de la línea
        }
    } while (false); // Este do-while se ejecutará solo una vez porque la validación interna ya maneja el bucle

    // Limpia el buffer después de leer el número del género, si se va a leer otra string
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    // Asignar el nuevo artista a los arreglos
    nombres[numero_de_artistas] = nuevo_artista;
    generos[numero_de_artistas] = static_cast<GeneroMusical>(opcion_de_genero); // Convierte int a enum
    votos[numero_de_artistas] = 0; // Inicializa los votos del nuevo artista
    numero_de_artistas++; // Incrementa el contador de artistas
    cout << "Artista " << nuevo_artista << " registrado exitosamente." << endl;
}
```

Figura 4. Se detalla la lógica para la función que permite al usuario ingresar un nuevo artista, incluyendo la validación del nombre y la selección interactiva de su género musical.

 <p>ESCUELA POLITÉCNICA NACIONAL Programación I</p>	LABORATORIO No. 3
NOMBRE: Danny Paúl Ponce Guacalés.	FECHA DE ENTREGA: 20/06/2025

```

// Función para registrar votos para un artista
void registrarVoto(const string nombres[], int votos[], int numero_de_artistas) {
    if (numero_de_artistas == 0) { // Maneja el caso de lista vacía
        cout << "No hay artistas registrados para votar." << endl;
        return;
    }

    int opcionVoto;
    do {
        cout << "Artistas disponibles para votar:" << endl;
        for (int i = 0; i < numero_de_artistas; ++i) {
            cout << (i + 1) << ". " << nombres[i] << endl;
        }
        cout << "Ingrese el numero del artista para votar (0 para salir): ";

        // Valida la entrada del voto
        while (!(cin >> opcionVoto) || opcionVoto < 0 || opcionVoto > numero_de_artistas) {
            cout << "Entrada invalida. Por favor, ingrese un numero entre 0 y " << numero_de_artistas << ":" ;
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }

        if (opcionVoto >= 1 && opcionVoto <= numero_de_artistas) {
            votos[opcionVoto - 1]++;
            cout << "Voto registrado para " << nombres[opcionVoto - 1] << endl;
        } else if (opcionVoto == 0) {
            cout << "Saliendo del registro de votos." << endl; // Opción para salir del bucle
        }
        // Limpiar el buffer si la siguiente operación no es un cin >> numero
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    } while (opcionVoto != 0); // Permite múltiples votos hasta que se ingrese 0
}

```

Figura 5. Se muestra el código de la función que gestiona el proceso de votación, mostrando la lista de artistas disponibles y permitiendo al usuario emitir votos por su artista preferido.

```

// Función para mostrar estadísticas de votos
void mostrarEstadisticas(const string nombres[], const GeneroMusical generos[], const int votos[], int numero_de_artistas) {
    if (numero_de_artistas == 0) { // Maneja el caso de lista vacía
        cout << "No hay artistas registrados para mostrar estadísticas." << endl;
        return;
    }

    cout << "--- Estadísticas de Votos por Artista ---" << endl;
    for (int i = 0; i < numero_de_artistas; ++i) {
        cout << nombres[i] << ": " << votos[i] << " votos" << endl; // Muestra nombre y votos
    }
    cout << "-----" << endl;
}

```

Figura 6. Presenta la implementación de la función que calcula y muestra el total de votos acumulados por cada artista registrado en el sistema

 <p>ESCUELA POLITÉCNICA NACIONAL Programación I</p>	LABORATORIO No. 3
NOMBRE: Danny Paúl Ponce Guacalés.	FECHA DE ENTREGA: 20/06/2025

```

int main() {
    // Declaración de los arreglos estáticos y el contador de artistas en main
    string nombredeartistas[MAXIMO_ARTISTAS];
    GeneroMusical generos_de_artistas[MAXIMO_ARTISTAS];
    int votos_para_artistas[MAXIMO_ARTISTAS];
    int numero_de_artistas = 0; // Contador de artistas actual

    // Inicializa los artistas predefinidos al inicio del programa
    inicializarArtistas(nombredeartistas, generos_de_artistas, votos_para_artistas, numero_de_artistas);

    int opcionMenu;
    do {
        // Muestra el menú principal al usuario
        cout << "\n--- MENU PRINCIPAL ---" << endl;
        cout << "1. Registrar nuevo artista" << endl; //
        cout << "2. Registrar voto" << endl; //
        cout << "3. Mostrar estadísticas" << endl; //
        cout << "4. Salir" << endl;
        cout << "Ingrese su opción (1-4): ";

        // Valida la entrada del menú
        while (!(cin >> opcionMenu) || opcionMenu < 1 || opcionMenu > 4) {
            cout << "Opción invalida. Por favor, ingrese un número entre 1 y 4: ";
            cin.clear(); // Limpia el estado de error de cin
            cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Ignora el resto de la línea
        }

        // Usa un switch para manejar las opciones del menú
        switch (opcionMenu) {
            case 1:
                registrarArtista(nombredeartistas, generos_de_artistas, votos_para_artistas, numero_de_artistas);
                break;
            case 2:
                registrarVoto(nombredeartistas, votos_para_artistas, numero_de_artistas);
                break;
            case 3:
                mostrarEstadisticas(nombredeartistas, generos_de_artistas, votos_para_artistas, numero_de_artistas);
                break;
            case 4:
                cout << "Saliendo del programa. ¡Hasta luego!" << endl;
                break;
        }
    } while (opcionMenu != 4); // El bucle continúa hasta que el usuario elija salir

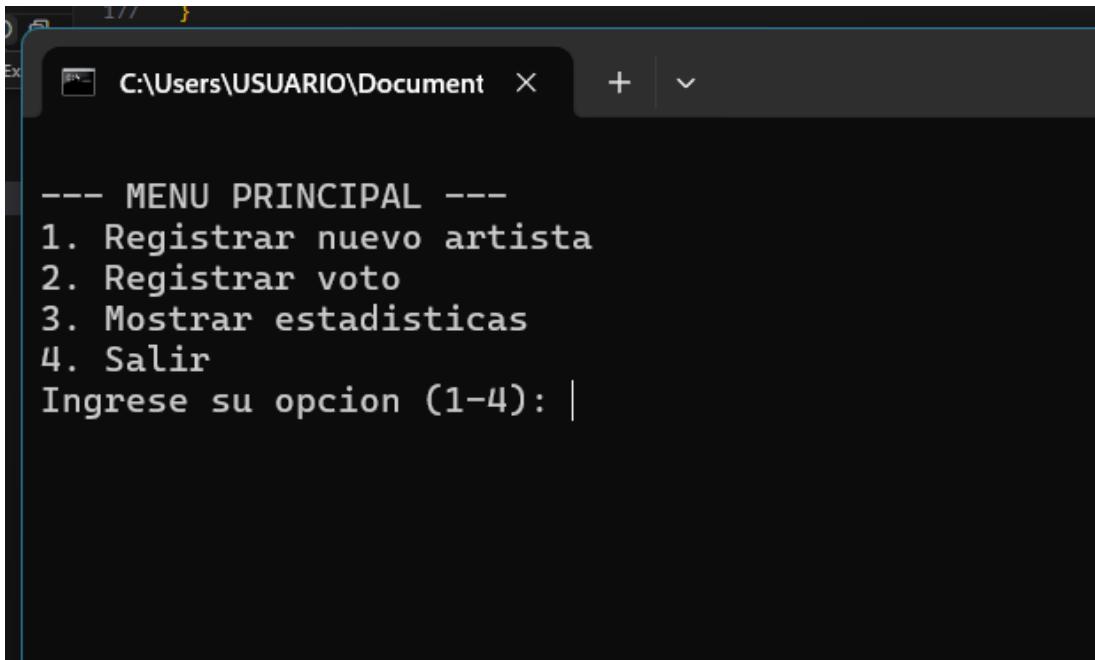
    return 0;
}

```

Figura 7. Finalmente se muestra el programa principal, en el que se juntan todas estas funciones, separadas por un menú interactivo.

5. SALIDAS DEL PROGRAMA

 ESCUELA POLITÉCNICA NACIONAL Programación I	LABORATORIO No. 3
NOMBRE: Danny Paúl Ponce Guacalés.	FECHA DE ENTREGA: 20/06/2025

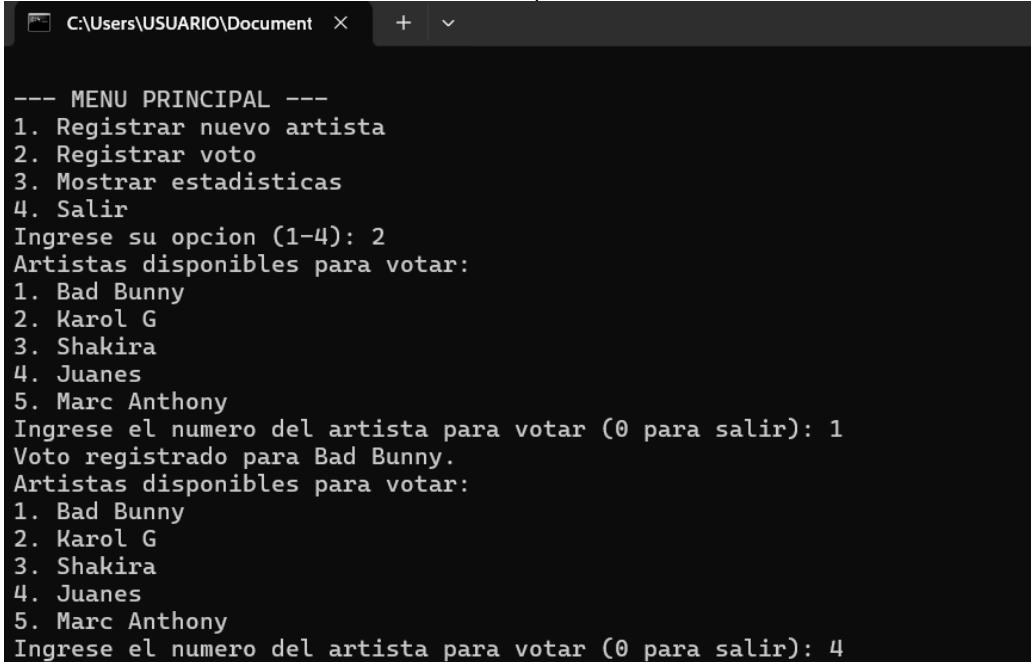


```

--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadísticas
4. Salir
Ingrese su opción (1-4): 

```

Figura 8. La imagen nos muestra el menú el cual navega entre las funciones del sistema, explícitamente mostrándonos 4.



```

--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadísticas
4. Salir
Ingrese su opción (1-4): 2
Artistas disponibles para votar:
1. Bad Bunny
2. Karol G
3. Shakira
4. Juanes
5. Marc Anthony
Ingrese el número del artista para votar (0 para salir): 1
Voto registrado para Bad Bunny.
Artistas disponibles para votar:
1. Bad Bunny
2. Karol G
3. Shakira
4. Juanes
5. Marc Anthony
Ingrese el número del artista para votar (0 para salir): 4

```

Figura 9. En la imagen se nos muestra la función de registrar votos, dando dos votos, 1 para el artista Bad Bunny y otra para el artista Juanes.

 ESCUELA POLITÉCNICA NACIONAL Programación I	LABORATORIO No. 3
NOMBRE: Danny Paúl Ponce Guacalés.	FECHA DE ENTREGA: 20/06/2025

```

--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadísticas
4. Salir
Ingrese su opcion (1-4): 1
Ingrese el nombre del artista que desea agregar: Danny Ponce
Seleccione el genero musical:
1. POP
2. CUMBIA
3. SALSA
4. ROCK
5. REGGAETON
6. OTRO
Ingrese el numero del genero (1-6): 4
Artista 'Danny Ponce' registrado exitosamente.

--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadísticas
4. Salir
Ingrese su opcion (1-4): 2
Artistas disponibles para votar:
1. Bad Bunny
2. Karol G
3. Shakira
4. Juanes
5. Marc Anthony
6. Danny Ponce
Ingrese el numero del artista para votar (0 para salir): 6
Voto registrado para Danny Ponce.

```

Figura 10. La imagen nos muestra la función de agregar un nuevo artistas y luego le damos un voto.

 ESCUELA POLITÉCNICA NACIONAL Programación I	LABORATORIO No. 3
NOMBRE: Danny Paúl Ponce Guacalés.	FECHA DE ENTREGA: 20/06/2025

```

--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadísticas
4. Salir
Ingrese su opcion (1-4): 3
--- Estadísticas de Votos por Artista ---
Bad Bunny: 1 votos
Karol G: 0 votos
Shakira: 0 votos
Juanes: 1 votos
Marc Anthony: 0 votos
Danny Ponce: 1 votos
-----
```

Figura 11. La imagen nos muestra los datos estadísticos de los votos para los artistas, como se puede observar los votos para los 3 artistas a los que votamos anteriormente se guardaron con éxito.

```

C:\Users\USUARIO\Document × + ▾
--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadísticas
4. Salir
Ingrese su opcion (1-4): 1
Ingrese el nombre del artista que desea agregar: Shakira
El artista 'Shakira' ya existe.
Por favor, ingrese un nombre valido: shakira
El artista 'shakira' ya existe.
Por favor, ingrese un nombre valido: sHaKiRa
El artista 'sHaKiRa' ya existe.
Por favor, ingrese un nombre valido:
Debe ingresar un nombre.
Por favor, ingrese un nombre valido: |
```

Figura 12. La imagen nos muestra que pasa si tratamos de agregar un artista ya definido aún que hagamos combinaciones entre minúsculas y mayúsculas el sistema nos dice que ya está inicializado y también al tratar de ingresar un dato vacío el sistema no lo permite.

 <p>ESCUELA POLITÉCNICA NACIONAL Programación I</p>	<p>LABORATORIO No. 3</p>
<p>NOMBRE: Danny Paúl Ponce Guacalés.</p>	<p>FECHA DE ENTREGA: 20/06/2025</p>

```
--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadísticas
4. Salir
Ingrese su opcion (1-4): 1
No se pueden registrar más artistas. Se alcanzó el límite de 20.

--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadísticas
4. Salir
Ingrese su opcion (1-4): |
```

Figura 13. La imagen nos muestra que pasa una vez registramos 20 artistas y queremos añadir uno más.

```
};

const int MAXIMO_ARTISTAS = 20; // Límite de artistas

void inicializarArtistas(string nombres[], GeneroMusical generos[], int votos[], int& numero_de_artistas);
bool esNombreValido(const string& nombre, const string nombres[], int numero_de_artistas);
void registrarArtista(string nombres[], GeneroMusical generos[], int votos[], int& numero_de_artistas); //
void registrarVoto(const string nombres[], int votos[], int numero_de_artistas); //
void mostrarEstadisticas(const string nombres[], const GeneroMusical generos[], const int votos[], int numero_de_artistas); //

// Función para validar el nombre de un artista
bool esNombreValido(const string& nombre, const string nombres[], int numero_de_artistas){
    // Valida si el nombre está vacío
    if(nombre.empty()){
        cout << "Debe ingresar un nombre." << endl;
        return false;
    }
}
```

Figura 14. En la imagen se nos muestra que eliminamos la función inicializar artistas y comentamos los prototipos para observar como el programa funciona sin esta función

 ESCUELA POLITÉCNICA NACIONAL Programación I	LABORATORIO No. 3
NOMBRE: Danny Paúl Ponce Guacalés.	FECHA DE ENTREGA: 20/06/2025

```

C:\Users\USUARIO\Document < + >
Ingrese el numero del genero (1-6): 2
Artista 'dannyponce' registrado exitosamente.

--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadisticas
4. Salir
Ingrese su opcion (1-4): 2
Artistas disponibles para votar:
1. dannyponce
Ingrese el numero del artista para votar (0 para salir): 1
Voto registrado para dannyponce.
Artistas disponibles para votar:
1. dannyponce
Ingrese el numero del artista para votar (0 para salir): 0
Saliendo del registro de votos.

--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadisticas
4. Salir
Ingrese su opcion (1-4): 3
--- Estadisticas de Votos por Artista ---
dannyponce: 2 votos
-----
--- MENU PRINCIPAL ---
1. Registrar nuevo artista
2. Registrar voto
3. Mostrar estadisticas
4. Salir
Ingrese su opcion (1-4): |

```

Figura 15. En la imagen se nos muestra que es lo que ocurre cuando el programa se ejecuta sin la función que inicializa a los artistas predefinidos.

6. CONCLUSIONES

1. La implementación del programa demostró una sólida robustez, gestionando eficazmente las entradas inválidas y los casos límite. Se validaron aspectos cruciales como nombres vacíos o duplicados, opciones de menú o género fuera de rango, y el cumplimiento del límite de artistas. Esto aseguró que el programa no colapsara ante errores del usuario, sino que proporcionara mensajes claros y precisos, cumpliendo con las garantías de estabilidad y los criterios de robustez establecidos.
2. El desarrollo del código se adhirió a los principios de Código Limpio, manifestados en el uso de nombres descriptivos y la modularización en funciones específicas y concisas. La aplicación efectiva de estructuras de control (if-else, switch) y de repetición (for, while, do-while), junto con el manejo de arreglos estáticos y tipos enum, fue fundamental para la correctitud de las funcionalidades de registro, votación y estadísticas⁴⁴⁴. Este enfoque contribuyó directamente a la claridad y legibilidad general del programa.

 ESCUELA POLITÉCNICA NACIONAL Programación I	LABORATORIO No. 3
NOMBRE: Danny Paúl Ponce Guacalés.	FECHA DE ENTREGA: 20/06/2025

7. REFERENCIAS BIBLIOGRÁFICAS

freeCodeCamp. (2020, 2 enero). *How to Write Clean Code in C++*.

freeCodeCamp.org. <https://www.freecodecamp.org/news/how-to-write-clean-code-in-c/>

Moisset, D. (s. f.). Clase string (size - length - empty - at).

<https://www.tutorialesprogramacionya.com/cmasmasya/detallegioncepto.php?codigo=193>

Parzibyte. (2024, 19 julio). *Convertir cadena a mayúscula y minúscula en C++*.

Parzibyte's Blog.

<https://parzibyte.me/blog/2019/06/05/convertir-cadena-mayuscula-minuscula-cpp/>