



## INFORME DE AVANCES

### TRABAJO PRACTICO:

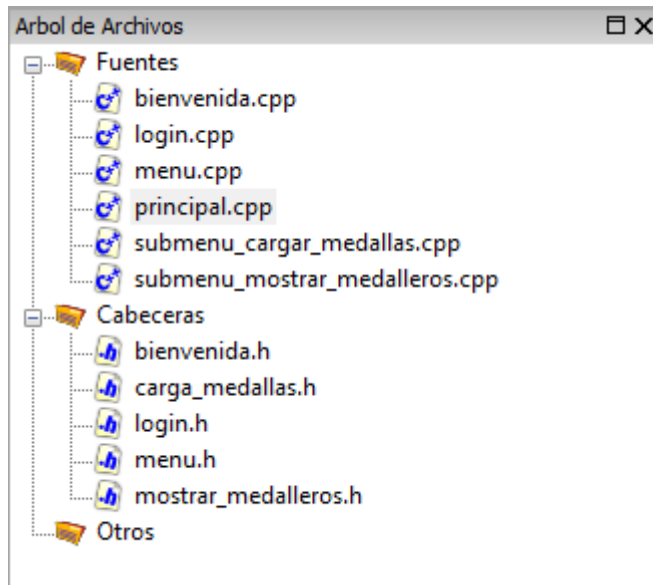
#### Medallero olímpico

**Catedra:** Algoritmos y Estructuras de Datos

**Integrantes:** Martinez Pedro Alejandro, Moretti Agustín

# Informe del Proyecto

Estructura del Proyecto:



## Archivo: `principal.cpp`

Descripción : Este archivo contiene la función principal del programa , que se encarga de iniciar la sesión del usuario y mostrar el menú principal.

Detalles :

- Inclusión de Cabeceras:

```
#include<iostream>
#include "login.h"
#include"menu.h"
using namespace std;
```

- Función Principal

```

#include<iostream>
#include <stdlib.h>
#include "login.h"
#include "menu.h"
#include "bienvenida.h"

using namespace std;

int main() {
    int user;

    //MENSAJE DE BIENVENIDA
    bienvenida();

    system("cls");
    //INICIAR SESION
    if(login(user)==true){
        system("cls");
        //MENU PRINCIPAL JUNTO CON LOS SUBMENUS
        menu_principal();
    }else{
        return 0;
    }
}

```

- Funcionalidad:
  - Llama a la función **bienvenida()** , muestra mensaje de bienvenida
  - Llama a la función **login()** para iniciar sesión, si es valido limpia la pantalla y ingresa a menu.
  - Llama a la función **menu\_principal()** para mostrar el menu principal seguido de que haya iniciado sesión.

### Archivo: menu.cpp

Descripción: Este archivo contiene la implementación de la función menú principal, que presenta el menú principal del sistema.

Detalles:

- Incluir Cabeceras

```

#ifndef MENU_H
#define MENU_H
#include<iostream>
#include<stdlib.h>
using namespace std;
void menu_principal();
#endif

```

- Función Principal:

```

void menu_principal(){
    system("color 60");
    bool validacion1=false;
    char opcion;
    char salir;
    do{
        //Visualizacion de menu
        cout << "Menu Principal"<<endl;
        cout << "===== " << endl;
        cout << "1. Cargar Medallas" << endl;
        cout << "2. Mostrar Medallero" << endl;
        cout << "X. Salir" << endl;
        cout << "Ingrese una opcion: ";
        cin >> opcion;
        cout<<endl;

        switch(opcion){
            case '1':{
                //Cargar Medallas
                cout<<"Carga de Medallas..."<<endl;
                cout<<endl;
                Sleep(1000);
                system("cls");
                cargar_medallas();
            }
            break;

            case '2':{
                // Mostrar Medallero
                cout<<"Mostrar Medalleros..."<<endl;
                cout<<endl;
                Sleep(1000);
                system("cls");
                mostrar_medalleros();
            }
            break;

            case 'X':{
                //Salida del Programa
                cout<<"Estas Seguro? (Y:Salir del programa No: Cualquier Tecla)"<<endl;
                cin>>salir;
                if(salir=='Y'){
                    validacion1=true;
                    cout<<"NOS VEMOS PRONTO....."<<endl;
                }
            }
            break;

            case 'x':{
                //Salida del Programa
                cout<<"Estas Seguro? (Y:Salir del programa No: Cualquier Tecla)"<<endl;
                cin>>salir;
                if(salir=='Y'){
                    validacion1=true;
                    cout<<"NOS VEMOS PRONTO....."<<endl;
                }
            }
            break;

            default:{
                //Invalido
                cout<<"Opcion invalida. Intente de nuevo"<<endl;
                cout<<endl;
                Sleep(1000);
                system("cls");
            }
            break;
        }
    } while(validacion1==false);
}

```

- Funcionalidad:  
Menú de Opciones:

- '1': Llama a la función `mostrar_medalleros()` para mostrar los medalleros
- '2': Llama a la función `cargar_medallas()` para cargar medallas.
- 'X': Sale del sistema

#### Archivo: submenu\_mostrar\_medalleros.cpp

Descripción: Este archivo contiene la implementación de la función `mostrar_medalleros()`, que presenta un submenú para visualizar diferentes tipos de medalleros.

Detalles:

- Incluir Cabeceras:

```
#ifndef SUBMENU_MOSTRAR_MEDALLEROS_H
#define SUBMENU_MOSTRAR_MEDALLEROS_H
#include<iostream>
#include <windows.h>
using namespace std;

void mostrar_medalleros() ;

#endif
```

- Función Principal :

```

void mostrar_medalleros(){
    char opcion;
    bool validacion=false;
    system("color 60");
    do {
        // Visualización de menú
        cout << "Menu Juegos" << endl;
        cout << "===== " << endl;
        cout << "1. Medallero por Pais" << endl;
        cout << "2. Medallero por Deporte" << endl;
        cout << "3. Top Medallero" << endl;
        cout << "X. Volver al menu principal" << endl;
        cout << "Ingrese una opcion: ";
        cin >> opcion;
        cout << endl;

        switch(opcion) {
            case '1': {
                cout << "Funcionalidad en Desarrollo" << endl;
                cout << endl;
                Sleep(1000);
                system("cls");
                break;
            }
            case '2': {
                cout << "Funcionalidad en Desarrollo" << endl;
                cout << endl;
                Sleep(1000);
                system("cls");
                break;
            }
            case '3': {
                cout << "Funcionalidad en Desarrollo" << endl;
                cout << endl;
                Sleep(1000);
                system("cls");
                break;
            }
            case 'X': {
                cout << "Volviendo al Menu..." << endl;
                cout << endl;
                Sleep(1000);
                system("cls");
                validacion=true;
                break;
            }
            case 'x': {
                cout << "Volviendo al Menu..." << endl;
                cout << endl;
                Sleep(1000);
                system("cls");
                validacion=true;
                break;
            }
            default: {
                cout << "Opcion no valida. Intente de nuevo." << endl;
                cout << endl;
                Sleep(1000);
                system("cls");
                break;
            }
        }
    } while(validacion==false);
}

```

- Funcionalidad:  
Menú de Opciones:
  - '1' Medallero por país (en desarrollo)
  - '2' Medallero por Deporte (en desarrollo)
  - '3' Top Medallero (en desarrollo)
  - 'X' Volver al menú Principal

### Archivo: submenu\_cargar\_medallas.cpp

Descripción: Este archivo contiene la implementación de la función `cargar_medallas()`, que permite al usuario cargar medallas en el sistema.

Detalles:

- Incluir Cabeceras:

```
#ifndef CARGA_MEDALLAS_H
#define CARGA_MEDALLAS_H
#include<iostream>
#include <windows.h>
using namespace std;
void cargar_medallas();
#endif
```

- Función Principal

```

#include "carga_medallas.h"
#include "menu.h"
void cargar_medallas() {
    system("color 60");
    char opcion;
    bool validacion=false;
    do {
        // Visualización del submenu
        cout << "Menu Cargar Medallas" << endl;
        cout << "-----" << endl;
        cout << "1. Carga por Pais" << endl;
        cout << "2. Carga por Deporte" << endl;
        cout << "3. Carga por Deporte Individual" << endl;
        cout << "4. Carga por Deporte Colectivo" << endl;
        cout << "5. Carga por Categoria Deportiva" << endl;
        cout << "X. Volver al Menu Principal" << endl;
        cout << "Ingrese una opcion: ";
        cin >> opcion;
        cout<<endl;

        switch(opcion) {
            case '1': {
                cout<<"Funcionalidad en desarrollo"<<endl;
                cout<<endl;
                Sleep(1000);
                system("cls");
                break;
            }
            case '2': {
                cout<<"Funcionalidad en desarrollo"<<endl;
                cout<<endl;
                Sleep(1000);
                system("cls");
                break;
            }
            case '3': {
                cout<<"Funcionalidad en desarrollo"<<endl;
                cout<<endl;
                Sleep(1000);
                system("cls");
                break;
            }
            case '4': {
                cout<<"Funcionalidad en desarrollo"<<endl;
                cout<<endl;
                Sleep(1000);
                system("cls");
                break;
            }
            case '5': {
                cout<<"Funcionalidad en desarrollo"<<endl;
                cout<<endl;
                Sleep(1000);
                system("cls");
                break;
            }
            case 'X':{
                cout << "Volviendo al Menu Principal..." << endl;
                cout<<endl;
                system("cls");
                validacion=true;
                break;
            }
            case 'x':{
                cout << "Volviendo al Menu Principal..." << endl;
                cout<<endl;
                system("cls");
                validacion=true;
                break;
            }
            default: {
                cout << "Opcion no valida. Intente de nuevo." << endl;
                cout<<endl;
                Sleep(1000);
                system("cls");
                break;
            }
        }
    } while(validacion==false);
}

```

- Funcionalidad:



- '1' Cargar por país (en desarrollo)
- '2' Carga por deporte (en desarrollo)
- '3' Carga por deporte Individual (en desarrollo)
- '4' Carga por deporte colectivo(en desarrollo)
- '5' Carga por categoría Deportiva(en desarrollo)
- 'X' Volver al menú principal

### Archivo: login.cpp

Descripción: Este archivo contiene la implementación de las funciones declaradas en **login.h**, incluyendo **esCapicua()**, **passwordgenerada()** y **login()**.

Detalles:

- Incluir Cabeceras

```
#ifndef LOGIN_H
#define LOGIN_H
#include<iostream>
#include <windows.h>
using namespace std;
bool esCapicua(int num) ;
int passwordgenerada(int user) ;
bool login(int user) ;
#endif
```

- Función Principal

```

#include "login.h"
bool esCapicua(int num)
{
    int original = num;
    int invertido = 0;
    while (num > 0)
    {
        invertido = invertido * 10 + num % 10;
        num /= 10;
    }
    return original == invertido;
}

int passwordgenerada(int user)
{
    int N, ultimaTres;
    N = user;
    N = (user % 10) + ((user / 100) % 10) * 10 + (user / 10000) * 100;
    int capicua = N + 1;
    while (!esCapicua(capicua))
    {
        capicua++;
    }

    ultimaTres = (user % 1000);

    int year = 2024; // Año actual
    return (year * capicua) + ultimaTres; // Fórmula para generar la contraseña
}

bool login(int user){
    system("color 60");
    int passworduser;
    int contIntentosLogin=0;
    bool user_valido = false;

    cout << "-----" << endl;
    cout << "          LOGIN          " << endl;
    cout << "-----" << endl;
    // Usuario
    do
    {
        cout << "Ingrese su Numero de Legajo: (5 digitos): ";
        cin >> user;

        if (user < 10000) {
            cout << "ID invalida. Pruebe con un numero mayor a 10000." << endl;
            Sleep(1000);
            system("cls");
        }
        else if (user > 99999) {
            cout << "ID invalida. Pruebe con un numero menor a 99999." << endl;
            Sleep(1000);
            system("cls");
        }

        user_valido = (user >= 10000 && user <= 99999);
    } while (user_valido == false);

    cout << "-----" << endl;
    int password = passwordgenerada(user);
    cout << "Password Generada ==> " << password << endl;
    do
    {
        cout << "Ingrese su password unica generada: ";
        cin >> passworduser;

        if (passworduser != password)
        {
            cout << "Password incorrecta, intente nuevamente" << endl;
            contIntentosLogin++;
            Sleep(1000);
            system("cls");
        }
    } while ((passworduser != password) && (contIntentosLogin < 3));

    if (contIntentosLogin == 3) {
        cout << "Limite de ingresos erroneos alcanzado. Por favor, reinicie el programa para reintentar." << endl;
        return false;
    }

    // INGRESADO AL SISTEMA CORRECTAMENTE
    if (passworduser == password)
    {
        cout << "-----" << endl;
        cout << "Bienvenido al sistema" << endl;
        cout << "-----" << endl;

        //IR A MENU PRINCIPAL
        Sleep(3000);
        return true;
    }
}

```

- Funcionalidad:
  - esCapicua(int num): verifica si un número es capicúa
  - passwordgenerada(int user): Genera una contraseña basada en el número de usuario
  - login(): Solicita el numero de legajo , genera una contraseña única que se necesita ingresar para validar los datos.

Archivo: pantallabienvvenida.cpp

Descripción: Para este apartado utilizamos una función **void logo()** que, mediante el uso de múltiples couts, muestre por pantalla el logo de bienvenida dibujado con distintos caracteres.

Antes de hacer la llamada de `logo()` en el `main`, establecemos un determinado tamaño de consola mediante la linea de codigo `system()`; para que el logo se muestre de forma correcta.

Luego se hace el llamado de la función `logo()`; y se pregunta mediante un `if` que, si la tecla ENTER fue presionada continúe con el programa.

### Detalles:

- **Inclusión de Cabeceras:**

```
#ifndef BIENVENIDA_H
#define BIENVENIDA_H
#include <stdlib.h>
#include <windows.h>
#include <iostream>

using namespace std;
void bienvenida();
void logo();
#endif
```

- **Función Principal:**

```

#include "bienvenida.h"

void logo() {
    cout<<endl;
    cout<<"|-----|<<endl;
    cout<<"| 333333333333 6666666666 9999 9999 00 11 |<<endl;
    cout<<"| 333333333333 6666 666 999999 9999 0 0 11 |<<endl;
    cout<<"| 3333 6666 666 9999 999 9999 00 11 |<<endl;
    cout<<"| 3333 6666666666 9999 9999999 11 |<<endl;
    cout<<"| 3333 6666 9999 999999 11 |<<endl;
    cout<<"| 3333 6666 9999 9999 111111 |<<endl;
    cout<<"|-----|<<endl;
    cout<<"| 000000 |<<endl;
    cout<<"| 2222222222 0000000000 222222 444444 |<<endl;
    cout<<"| 2222222222 0000000000 000000000000 2222222222 4444444444 |<<endl;
    cout<<"| 2222 0000 0000 222 222 444 444 |<<endl;
    cout<<"| 2222 0000 0000000000 222 222 444 444 |<<endl;
    cout<<"| 2222 0000 222 222 ** 000 0000 ** \\ 11 / |<<endl;
    cout<<"| 2222222222 ** 00000000 ** 4444444444 ** |<<endl;
    cout<<"| 2222222222 ** 444444 ** |<<endl;
    cout<<"|-----|<<endl;
    cout<<"|/*: SetConsoleTextAttribute(GetStdHandle (STD_OUTPUT_HANDLE),4); cout<<***** SetConsoleTextAttribute(GetStdHandle (STD_OUTPUT_HANDLE) |<<endl;
    cout<<"|-----|<<endl;
    cout<<"|<<endl;
    cout<<"|<<endl;
    cout<<"| MAILS DE LOS INTEGRANTES: |<<endl;
    cout<<"| Pedro Martinez (ISI B): pedromartinez966440@gmail.com |<<endl;
    cout<<"| Agustín Ignacio Moretti (ISI E): fugaribi.a@gmail.com |<<endl;
    cout<<"| Próximo integrante en futuro. |<<endl;
    cout<<"|<<endl;
    cout<<"| Pulse ENTER para continuar. |<<endl;
    cout<<"|<<endl;
    cout<<"| Version 1.0 - 2024. |<<endl;
    cout<<"|-----|<<endl;
}

}

void bienvenida() {
    system("mode con: cols=136 lines=40");
    logo();

    if(GetAsyncKeyState(VK_RETURN)) {
        /*CONTINUACION DEL PROGRAMA*/
    }

    system("pause");
}

```