



MANUAL TÉCNICO FASE 1

Nombre: Angel Francisco Sique Santos

Carné: 202012039

INTRODUCCIÓN

En esta fase del proyecto Usac Games desea que se puedan implementar las funcionalidades del juego que permitirán gestionar la información de los usuarios y las skins del juego (fondo del tablero, imágenes de los barcos), por lo cual, a continuación se describen las estructuras y el funcionamiento a implementar.

REQUISITOS MÍNIMOS

- 500mb de disco duro
- 2gb de RAM
- Windows 7
- Equipo Intel Pentium o superior

OPCIONES DEL PROGRAMA

Menú principal

```
void menu(ListaUsuarios usuarios, ListaArticulos articulos, ColaTutorial tutorial, Cabecera cabecera)
{
    int opcion;
    string archivo;
    bool repetir = true;

    do
    {
        // Texto del menú que se verá cada vez
        cout << "===== " << endl;
        cout << "          Menu de Opciones" << endl;
        cout << "1. Carga Masiva" << endl;
        cout << "2. Registrar Usuario" << endl;
        cout << "3. Login" << endl;
        cout << "4. Reportes" << endl;
        cout << "0. Salir" << endl;
        cout << "===== " << endl;
        cout << "    Ingrese una opcion: ";
        cin >> opcion;

        switch (opcion)~
        } while (repetir);
    }
}
```

EL menú principal está en una función con un do while y dentro de este un switch que espera la opción elegida.

1. Carga Masiva

```
cout << "⇒Elija un archivo JSON para cargar: " << endl;
cin >> archivo;

{
    ifstream ifs(archivo);
    Json::Value usuariosObj;
    Json::Reader reader;

    reader.parse(ifs, usuariosObj);
    const Json::Value &usuariosJson = usuariosObj["usuarios"];
    for (int i = 0; i < usuariosJson.size(); i++)~

    ifstream ifs2(archivo);
    Json::Value articulosObj;
    reader.parse(ifs2, articulosObj);
}
```

Esta opción pide al usuario la ruta del archivo .json que contiene la información de los usuarios, artículos y el tutorial. Luego de leerlo guarda la información en sus respectivas estructuras.

2. Registrar Usuario

```
case 2:
    // Lista de instrucciones de la opción 2

    {

        string nick;
        string password;
        int monedas;
        int edad;
        cout << "Registrar usuario" << endl;
        cout << "⇒Nick: ";
        cin >> nick;
        cout << "⇒Password: ";
        cin >> password;

        cout << "⇒Edad: ";
        cin >> edad;
        if (usuarios.BuscarNick(nick) == false) ...
        }else{ ...

        cin.get();
        cout << endl;
        cout << endl;
    }

    break;
```

Esta opción le pide al usuario ingresar el nombre, la contraseña u la edad de un nuevo usuario para posteriormente crearlo.

3. Login

```
case 3:
    // Lista de instrucciones de la opción 3
    login(usuarios, tutorial, articulos);
    break;
```

Esta opción llama a una función llamada login.

```
void login(ListaUsuarios usuarios, ColaTutorial tutorial, ListaArticulos articulos)
{
    system("clear");
    cout << "===== " << endl;
    cout << "                LOGIN" << endl;

    cout << "⇒Nick: ";
    string nick;
    cin >> nick;
    cout << "⇒Password: ";
    string password;
    cin >> password;
    cout << endl;
    cout << endl;
    nodoUsuarios *usuarioActivo = usuarios.BuscarUsuario(nick, encriptarSHA256(password));
    if (usuarioActivo != NULL) ...
    else ...
}
```

La función login abre un menú donde se le pregunta el nick y la contraseña del usuario al que desea ingresar. Luego revisa el la lista circular con la función BuscarUsuario() esta función recorre la lista y si el nick y las contraseña coinciden devuelve el usuario que coincidió, del lo contrario devuelve NULL.

Si lo encuentra entra a otra función llamada menuLogin esta muestra un menú con las siguientes opciones:

```
cout << "          ⇒ Bienvenido " << usuarioActivo->nick << " ←" << endl;
cout << "===== " << endl;
cout << "1. Editar usuario" << endl;
cout << "2. Eliminar usuario" << endl;
cout << "3. Ver tutorial" << endl;
cout << "4. Ver articulos de la tienda" << endl;
cout << "5. Realizar movimientos" << endl;
cout << "6. Mostrar movimientos" << endl;
cout << "7. Cerrar sesion" << endl;
cout << "===== " << endl;
cout << endl;
cout << endl;
cout << "⇒Ingrese una opcion: ";
cin >> opcion;
cout << endl;
cout << endl;
```

1. Si se entra a la opción editar usuario mostrara un pequeño menú donde se puede editar la edad, nick y contraseña del usuario con sesión iniciada.

```
cout << "===== " << endl;
cout << "1. Editar nick" << endl;
cout << "2. Editar edad" << endl;
cout << "3. Editar password" << endl;
cout << "4. Volver" << endl;
cout << "===== " << endl;
cout << "Ingrese una opcion: ";
cin >> opcion;
```

2. La opción eliminar usuario muestra una pregunta donde se le pide confirmar al usuario que realmente quiere eliminar el usuario, si responde que si se procede a eliminar de forma permanente el usuario con sesión iniciada.

```
cout << "Eliminar usuario" << endl;
cout << "¿Esta seguro que desea eliminar el usuario? (s/n): ";
char respuesta;
cin >> respuesta;
if (respuesta == 's')
{
    usuarios.EliminarUsuario(usuarioActivo);
    repetir = false;
    cout << "Usuario eliminado" << endl;
    cin.get();
    cout << endl;
    cout << endl;
}
else ...
```

3. La opción ver tutorial muestra en consola el tutorial ingresado en el archivo .json.

```
cout << "          Ver tutorial" << endl;
{
    tutorial.Imprimir();
    cin.get();
    cout << endl;
}
```

4. La opción Ver artículos de la tienda muestra en consola la tienda con los artículos ingresados en el .json

```
cout << "Ver articulos de la tienda" << endl;
{
    cout << "=====Monedas: " << usuarioActivo->monedas << endl;
    articulos.Imprimir();      ASique-4, hace 3 días • Commit
    cin.get();
    cout << endl;
}
```

5. La opcion realizar movimientos le pregunta al usuario las cooredenadas del movimiento, el usuario puede ingresar todos los movimientos que quiera, estos se guardan en una pila.

```
cout << "Realizar movimientos" << endl;
{
    ListaMovimientos *colaMovimientos;
    colaMovimientos = new ListaMovimientos();
    bool repetir = true;
    int i = 1;
    do
    {
        cout << "Movimiento numero: " << i << endl;
        cout << "Coordenad en x: ";
        int x;
        cin >> x;
        cout << "Coordenad en y: ";
        int y;
        cin >> y;
        cout << x << " " << y << endl;
        colaMovimientos->InsertarFinal(x, y);
        i += 1;
        cout << "Desea agregar otro movimiento? (s/n): ";
        char respuesta;
        cin >> respuesta;
        if (respuesta == 'n')-
    } while (repetir);

    usuarioActivo->lista.InsertarFinal(colaMovimientos);
    usuarioActivo->monedas += 1;
}
```

6. Mostrar movimientos muestra todas las pilas ingresadas en el usuario con sesión iniciada.

```
cout << "Mostrar movimientos" << endl;
{
    usuarios.MostrarMovimientos(usuarioActivo);
    cout << "Mostrar movimientos exitosamente" << endl;
    usuarioActivo->lista.CrearGraphviz();
    cin.get();
    cout << endl;
}
break;
```

7. La opción cerrar sesión regresa al menú principal y cierra la sesión del usuario activo.

4. Reportes

```
{
    cout << "===== " << endl;
    cout << "          Reportes" << endl;
    cout << "1. Reporte de usuarios" << endl;
    cout << "2. Reporte de articulos" << endl;
    cout << "3. Reporte de tutorial" << endl;
    cout << "0. Salir" << endl;
    cout << "===== " << endl;
    cout << "    Ingrese una opcion: ";
    cin >> i;
    switch (i) ...
} while (repetir1);
```

Esta opción abre un menú con opciones para realizar reportes.

La opción reporte usuario le pregunta al usuario el método en el que desea ordenar su lista en base a su edad para ser mostrada en consola y graficada con graphviz.

```
int i;
bool repetir2 = false;
cout << "===== " << endl;
cout << "1. Ordenar de maenra descendente" << endl;
cout << "2. Ordenar de manera ascendente" << endl;
cout << "0. Salir" << endl;
cout << "===== " << endl;
cin >> i;
do ...
```

La opción reportes artículos le pregunta al usuario el método en el que desea ordenar su lista en base a su precio para ser mostrada en consola y graficada con graphviz.

```
int i;
bool repetir2 = false;
cout << "===== " << endl;
cout << "1. Ordenar de maenra descendente" << endl;
cout << "2. Ordenar de manera ascendente" << endl;
cout << "0. Salir" << endl;
cout << "===== " << endl;
cin >> i;
do ...
```

La opción reporte tutorial muestra en consola y grafica usando graphviz el tutorial ingresado en el .json.


```
// Lista de instrucciones de la  
tutorial.Imprimir();  
tutorial.CrearGraphviz();  
break;
```

La opción salir regresa al menú principal.

5. Salir

Esta opción cierra el programa.

