

Universidad Mariano Gálvez de Guatemala

Facultad de Ingeniería Matemática y Ciencias Físicas

Campus Villa Nueva, Guatemala

Ingeniería en Sistemas de información y Ciencias de la computación

Ingeniero: Carlos Alejandro Arias

Curso: Física 1

Código de Curso: 012

Código de Carrera: 5090



Laboratorio 9

Byron Ignacio Salazar Orellana

Sección: A

No. Carné: 23-5096

Fecha: 15/05/2024

Introducción

Para el siguiente trabajo se presenta un programa el cual es un listado de tareas pendientes, en donde se utilizan pilas y colas para llevar el orden de desarrollo de las tareas. Este programa está hecho en lenguaje c++ y explica el código para que sea comprensible que es lo que realiza cada sección del mismo. Es de suma importancia la comprensión de estos métodos de ordenamiento de lista.

```

1  #include <iostream>
2  #include <stack>
3  #include <queue>
4  #include <string>
5  // Gracias a las clases stack y queue podremos usar las pilas y colas
6  // de forma respectiva. Las pilas se refieren a que el ultimo elemento
7  // en entrar es el primero en salir y las colas es el primer elemento en
8  // entrar es el primero en salir
9  struct Tarea { // Tarea se refiere a las tareas que vamos a ingresar
10     int id;
11     std::string descripcion;
12 };
13

```

Lo que hay que entender para trabajar con pilas y colas, es que estas nos sirven para llevar un listado de actividades, cosas o llevar el control de algo en específico, nos da la versatilidad de poder ir eliminando elementos según nuestra conveniencia. Las pilas cuentan con la función de finalizar o eliminar la última tarea ingresada a la lista. Por su parte las colas eliminan o finalizan la primera tarea ingresada a la lista.

```

int main() {
    std::stack<Tarea> pilaTareas; // se crea la pila de las tareas
    std::queue<Tarea> colaTareas; // se crea la cola de tareas
    int opcion; // opcion es lo que vamos a utilizar para el menu
    int idCounter = 1;

    do {
        std::cout << "Menu de Tareas" << std::endl; // este es el menú de tareas
        std::cout << "1. Agregar una tarea" << std::endl; // Con esta opción se añade una tarea a la lista
        std::cout << "2. Completar la ultima tarea" << std::endl; // Con esto utilizamos las pilas para completar la ultima tarea ingresada
        std::cout << "3. Completar la tarea mas antigua" << std::endl; // Con esto usamos las colas para completar la primer tarea ingresada
        std::cout << "4. Mostrar todas las tareas" << std::endl; // se muestra el listado de tareas
        std::cout << "5. Salir" << std::endl;
        std::cout << "Seleccione una opcion: "; // se nos pide que elijamos una opción (un numero del 1 al 5)
        std::cin >> opcion;
        std::cin.ignore();
    } while (opcion != 5);
}

```

Se crea la pila de las tareas y la cola de las tareas, así mismo se crea un menú con diferentes opciones para que el usuario pueda interactuar de una forma apropiada con la aplicación, donde se puede ingresar tarea, completar ultima tarea, completar primera tarea y mostrar las tareas que quedan pendientes.

```

switch (opcion) { //para el menu utilizaremos un switch case debido a la diversa cantidad de opciones del menu
case 1: { // se muestra el caso uno, que es el de añadir una tarea a la lista
// para ello se ingresa el nombre de la tarea y se genera dentro de la lista
std::string descripcion;
std::cout << "Ingrese la descripción de la tarea: ";
std::getline(std::cin, descripcion);

Tarea nuevaTarea = { idCounter++, descripcion };
pilaTareas.push(nuevaTarea);
colaTareas.push(nuevaTarea);

std::cout << "Tarea agregada correctamente." << std::endl;
break;
}
}

```

Para el menú usaremos un switch case, en donde están programadas todas las opciones, según el número ingresado al switch case. Para el primer caso, tendremos el de ingresar tareas, donde podremos ingresar tantas cuantas necesitemos, y se tomarán en cuenta para el listado.

```

case 2: { // La pila de tareas es la que nos permite completar la ultima tarea ingresarla
// por ende la ultima tarea se va a eliminar de la lista
if (!pilaTareas.empty()) {
Tarea tareaCompletada = pilaTareas.top();
pilaTareas.pop();
std::cout << "Tarea completada: " << tareaCompletada.descripcion << std::endl;
} else {
std::cout << "No hay tareas para completar." << std::endl;
}
break;
}

```

Para el segundo caso tenemos las pilas, las cuales eliminan o en este caso finaliza la última tarea ingresada a la lista, es un método que se basa en UEPS (Últimas en entrar, primeras en salir).

```

case 3: { // Las colas, caso contrario, nos permite completar la primer tarea ingresada a la lista
// luego de haberse completado esta, se eliminará de la lista.
if (!colaTareas.empty()) {
Tarea tareaAtendida = colaTareas.front();
colaTareas.pop();
std::cout << "Tarea atendida: " << tareaAtendida.descripcion << std::endl;
} else {
std::cout << "No hay tareas para atender." << std::endl;
}
break;
}
}

```

Para el tercer caso tenemos las colas, similarmente a las pilas, también finalizan las tareas, pero con la peculiaridad de eliminar la primera tarea ingresada, la cual se basa en el método PEPS (primeras en entrar, primeras en salir).

```
case 4: { // en el caso número 4 se nos mostraran las tareas que quedan pendientes
// todas las tareas ingresadas las cuales no se han sacado de la lista ni con pilas ni con colas
    if (!pilaTareas.empty()) {
        std::cout << "Lista de tareas pendientes:" << std::endl;
        std::queue<Tarea> copiaColaTareas = colaTareas;
        int numeroTarea = 1;
        while (!copiaColaTareas.empty()) {
            Tarea tarea = copiaColaTareas.front();
            copiaColaTareas.pop();
            std::cout << numeroTarea++ << ". " << tarea.descripcion << std::endl;
        }
        // en caso de no haber tareas se nos mostrará un mensaje en pantalla
    } else {
        std::cout << "No hay tareas pendientes." << std::endl;
    }
    break;
// para el ultimo programa se nos muestra la opcion de salir
}
```

Para el cuarto caso se nos muestra el listado de tareas pendiente, las cuales fueron ingresadas y no se han finalizado por ningún medio, ya sea por pilas o por colas.

```
case 5:
    std::cout << "Saliendo del programa." << std::endl;
    break;
default:
    std::cout << "Opción no válida. Intente nuevamente." << std::endl;
    break;
}

std::cout << std::endl;
} while (opcion != 5);

return 0;
}
```

Por último, tenemos la opción de salir del programa.

```
Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion:
```

Iniciando el programa esto es lo que nos saldrá al no haber ingresado ningún dato

```
D:\Pilas y Colas.exe X + v
Ingrese la descripcion de la tarea: Lavar platos
Tarea agregada correctamente.

Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion: 1
Ingrese la descripcion de la tarea: Ordenar mi cama
Tarea agregada correctamente.

Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion: 1
Ingrese la descripcion de la tarea: Tender la ropa
Tarea agregada correctamente.

Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion:
```

Ingresamos las tareas al listado para luego finalizarlas.

```
Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion: 4
Lista de tareas pendientes:
1. comer
2. Lavar platos
3. Ordenar mi cama
4. Tender la ropa
```

Podemos ver las tareas con el listado de tareas.

```
Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion: 3
Tarea atendida: comer

Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion: 4
Lista de tareas pendientes:
1. Lavar platos
2. Ordenar mi cama
3. Tender la ropa
```

Luego, podremos eliminar la primer tarea ingresada, en este caso era la tarea de comer.


```
Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion: 2
Tarea completada: Ordenar mi cama
```

```
Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion: 4
Lista de tareas pendientes:
1. Lavar platos
2. Ordenar mi cama
```

```
Menu de Tareas
1. Agregar una tarea
2. Completar la ultima tarea
3. Completar la tarea mas antigua
4. Mostrar todas las tareas
5. Salir
Seleccione una opcion:
```

Y también podremos finalizar tareas por medio de las pilas, en este caso fue la última tarea.

Conclusiones

Es de suma importancia dominar este tema, ya que es un método de suma utilidad y que los programadores pueden utilizar para facilitar la estructuración de listas por medio de pilas las cuales sacan de la lista el último elemento ingresado y las colas que eliminan el primer elemento ingresado, todo esto según nuestra conveniencia para una mejor estructura de un programa o llevar de una mejor forma una lista de actividades a realizar.