Universidad de Costa Rica

Escuela de ingeniería eléctrica

Estructuras abstractas de datos y algoritmos para ingeniería

Simulador de Cola para Parque de Atracciones

Autores:
Joseph Álvarez Sandí C20440
Joseph Castillo Torres C11795
Luis Fernando Rojas Morua B86941
Alejandro García Rucavado C03187

2 de Marzo de 2025

1. Uso de IA en el Proyecto

1.1. Manejo de Colas

Para el manejo de colas, se utilizó ChatGPT para tener un código funcional de referencia, para hacer una cola de grupos.

Entrada: *¿Cómo podría implementar en C++ una cola que maneje grupos. Siendo los grupos listas enlazadas donde cada nodo sea un visitante? .*

Anteriormente ya se había utilizado de entrada el código de las clases *Node.h*, *LinkedList.h* y *Queue.h*, por lo que la respuesta está basada en estos códigos proporcionados.

Salida:

```
#include <iostream>
using namespace std;

// Nodo para cada persona dentro de un grupo
class Person {
public:
    string name;
    Person* next;

    Person(string name) : name(name), next(nullptr) {}
};

// Nodo para representar un grupo en la cola
class Group {
public:
    int groupID;
    Person* head; // Apunta a la primera persona del grupo
    Person* tail; // Apunta a la ultima persona del grupo
    Group* next;  // Apunta al siguiente grupo en la cola

    Group(int groupID, string firstMember) :
        groupID(groupID), head(new Person(firstMember)), tail(head), next(nullptr) {}

    // Agregar una persona al grupo
    void addMember(string name) {
        Person* newPerson = new Person(name);
        tail->next = newPerson;  // La ultima persona apunta al nuevo
        tail = newPerson;        // Actualizar la ultima persona
    }
};
```

```
// Clase para la cola de grupos
class Queue {
private:
    Group* front; // Apunta al primer grupo en la cola
    Group* rear;  // Apunta al ultimo grupo en la cola
    int nextGroupID;

public:
    Queue() : front(nullptr), rear(nullptr), nextGroupID(1) {}

    // Agregar un nuevo grupo a la cola
    void addGroup(string name) {
        Group* newGroup = new Group(nextGroupID, name);
        if (!rear) {
            front = rear = newGroup;
        } else {
            rear->next = newGroup;
            rear = newGroup;
        }
        cout << name << " creo el grupo " << nextGroupID << endl;
        nextGroupID++;
    }


    // Atender al primer grupo en la cola
    void serveNext() {
        if (!front) {
            cout << "La cola esta vacia.\n";
            return;
        }
        Group* temp = front;
        front = front->next;
        cout << "Atendiendo al ";
        temp->showGroup();
        delete temp; // Liberar memoria
        if (!front) rear = nullptr; // Si la cola queda vacia
    }

};
```

Este código se utilizó de referencia para construir la clase *Queue.h*. El código final utilizado se encuentra en el archivo con ese nombre, y se encuentra comentado, explicando cada una de las partes del código.

1.2. Interfaz Gráfica