COMPLEJIDAD TEMPORAL

Método para agregar una tarea

public String addTask(String Id, String name, String description, String dateLimit, int priority) {

        Task task = new Task(Id, name, description, dateLimit, priority);

        if (task.getPriority() == 0) {

            queue.add(task);

            agenda.insert(task.getId(),task);

        } else {

            priorityTasks.insert(task.getPriority(), task);

            agenda.insert(task.getId(),task);

        }

        return "Se agrego correctamente";

    }

1. Crear una instancia de **Task**: O(1)
2. Verificar la prioridad de la tarea: O(1)
3. Si la prioridad es igual a 0:
   * Agregar la tarea a **queue**: O(1)
   * Insertar la tarea en **agenda**: O(1)
4. Si la prioridad no es igual a 0:
   * Insertar la tarea en **priorityTasks**: O(1)
   * Insertar la tarea en **agenda**: O(1)

Dado que todas las operaciones son de tiempo constante, podemos sumarlas para obtener la complejidad total:

O(1) + O(1) + (O(1) + O(1)) = O(1)

Método para agregar un recordatorio

 public String addReminder(String Id, String name, String description, String dateLimit, int priority) {

        Reminder reminder = new Reminder(Id, name, description, dateLimit, priority);

        agenda.insert(Id, reminder);

        return "Se agrego correctamente";

    }

1. **Creación de la instancia de Reminder**:
   * La creación de una instancia de **Reminder** se realiza en tiempo constante (O(1)), ya que no depende del tamaño de los datos de entrada ni de ninguna estructura de datos. La operación se realiza en un tiempo fijo.
2. **Inserción en la estructura de datos agenda**:
   * La operación de inserción en **agenda** se realiza mediante la línea **agenda.insert(Id, reminder);**. La complejidad de esta operación depende de cómo esté implementada la estructura de datos **agenda**.
   * Si asumimos que la inserción en **agenda** se implementa de manera eficiente y es de tiempo constante (O(1)), entonces esta operación es O(1).

Dado que ambas operaciones son de tiempo constante (O(1)) en este análisis, la complejidad temporal total del método **addReminder** es O(1).