

Método de la ingeniería

[Subtítulo del documento]



1 de octubre de 2023

[Nombre de la compañía]

[Dirección de la compañía]

Tabla de contenido

[Problem Identification 2](#_Toc147187413)

[Problem Definition: 2](#_Toc147187414)

[Functionalities to consider: 2](#_Toc147187415)

[Information Gathering 2](#_Toc147187416)

[Definitions: 2](#_Toc147187417)

[Search for Creative Solutions 2](#_Toc147187418)

[Requirement Analysis: 2](#_Toc147187419)

# Problem Identification

### Problem Definition:

A task and reminder management system should be designed to allow users to add, organize and manage their tasks and reminders.

### Functionalities to consider:

- Store tasks and reminders with the help of a hash table.

- Design an interface that allows users to add, modify and delete tasks and reminders (tasks and reminders will be sorted by due date or priority).

- Manage task priorities between priority and non-priority.

- Implement a method to undo actions performed by a user in the system.

# Information Gathering

### Definitions:

Hash table:

Hash Table is a data structure which stores data in an associative manner. In a hash table, data is stored in an array format, where each data value has its own unique index value. Access of data becomes very fast if we know the index of the desired data.

Stack:

A Stack is a Last in First Out (LIFO) data structure. It supports two basic operations called push and pop. The push operation adds an element at the top of the stack, and the pop operation removes an element from the top of the stack.

# Search for Creative Solutions

# Requirement Analysis:

|  |  |
| --- | --- |
| Client |  |
| User | Students of Icesi |
| Functional Requirements | * R1: Add a task or a reminder * R2: Modify a task or a reminder * R3: Delete a task or a reminder * R4: Show the list of tasks and reminders * R5: Undo student actions |
| Problem Context | A task and reminder management system should be designed to allow users to add, organize, and manage their tasks and reminders.  It should also allow users to undo previously performed actions. |
| Non Functional requirements | * Usage of data structures such as hash, queue, and stack. * Test cases implementation to ensure the correct operating of the program. * The program must be sustainable and secure. * The program must be efficient |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | R1: Add a task or a reminder | | | |
| Summary | The program must allow to add the tasks and reminders | | | |
| Input | **Input name** | **Data type** | | **Valid condition** |
| title | String | |  |
| description | String | |  |
| limitDate | Date | | dd-mm-yyyy |
| priority | Boolean | | true or false |
| Result or Postcondition | The system verifies that the task or reminder is added successfully | | | |
| Output | **Output name** | | **Data type** | **Format** |
| message | | String | “The task / reminder has been successfully added |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | R2: Modify a task or a reminder | | | |
| Summary | The system must allow the user to modify the content of an added task. | | | |
| Input | **Input name** | **Data type** | | **Valid condition** |
| title | String | | Can not be empty |
| description | String | | Can not be empty |
| limitDate | Date | | dd-mm-yyyy |
| priority | Boolean | | Priority or No priority |
| Result or Postcondition | The system verifies the modifications and updates the information of the task. | | | |
| Output | **Output name** | | **Data type** | **Format** |
| message | | String | “The task:” title” Has been successfully modified |
| message | | String | “The reminder:”title” Has been successfully modified |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | R3: Delete a task and/or reminder | | | |
| Summary | The system must allow the user to delete any of the tasks they have created. | | | |
| Input | **Input name** | **Data type** | | **Valid condition** |
| title | String | | Can not be empty |
| description | String | | Can not be empty |
| limitDate | Date | | dd-mm-yyyy (A date that hasn’t yet passed) |
| priority | Boolean | | Priority (true) or No priority (false) |
| Result or Postcondition | The system verifies the deletion of a task/reminder, and updates the list of tasks and reminders. | | | |
| Output | **Output name** | | **Data type** | **Format** |
| message | | String | “The task:” title” Has been successfully deleted |
| message | | String | “The reminder:”title” Has been successfully deleted |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier and Name | R4: See a list of tasks and reminders | | | |
| Summary | The system must allow the user to see their list of tasks and reminders whenever they want. It should show them like a list. | | | |
| Input | **Input name** | **Data type** | | **Valid condition** |
| n/a | n/a | | n/a |
| Result or Postcondition | The user will be able to see their list of tasks and reminders | | | |
| Output | **Output name** | | **Data type** | **Format** |
| Tasks | | Task | List |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier and Name | R5: Undo actions | | | |
| Summary | The system must allow the user to undo their last action within the system. This undo functionality should revert the last action performed, whether it was adding, modifying, or deleting a task or reminder. | | | |
| Input | **Input name** | **Data type** | | **Valid condition** |
| n/a | n/a | | n/a |
| Result or Postcondition | The system reverts the last action performed by the user, restoring the previous state of the task and reminder list. | | | |
| Output | **Output name** | | **Data type** | **Format** |
| message | | String | "The last action has been successfully undone." |