

Mariana Agudelo Salazar - A00398722
Sharik Camila Rueda Lucero - A00399189
Natalia Vargas Parra - A00398706

ENGINEERING DESIGN METHOD

Phase 1: Problem Identification

- *Problem/Need:*

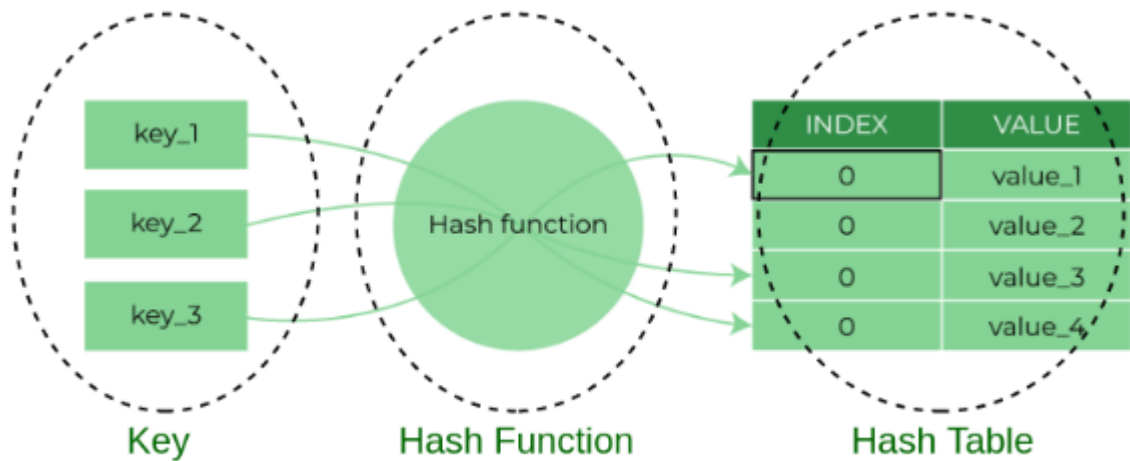
There is a need for a task and reminder management system that allows users to efficiently add, organize, and manage their to-do's and reminders.

- *Key Functionalities:*

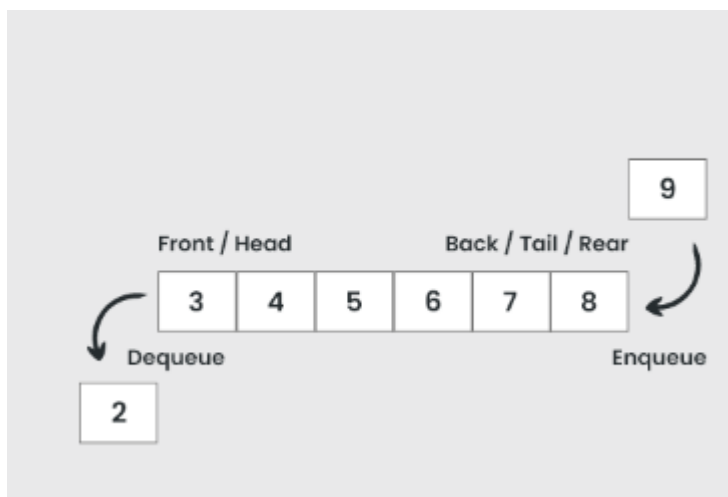
- Store tasks and reminders using a hash table for efficient data retrieval.
- Create a user-friendly interface for users to interact with the system.
- Implement sorting functionality for tasks and reminders by deadline or priority.
- Categorize tasks into "Priority" and "Non-priority".
- Use a priority queue (heap) to manage tasks based on their level of importance.
- Handle non-priority tasks using a First-In-First-Out (FIFO) approach.
- Implement an "Undo" feature to allow users to reverse their actions using a Last-In-First-Out (LIFO) stack.

Phase 2: Research and Gather Necessary Information

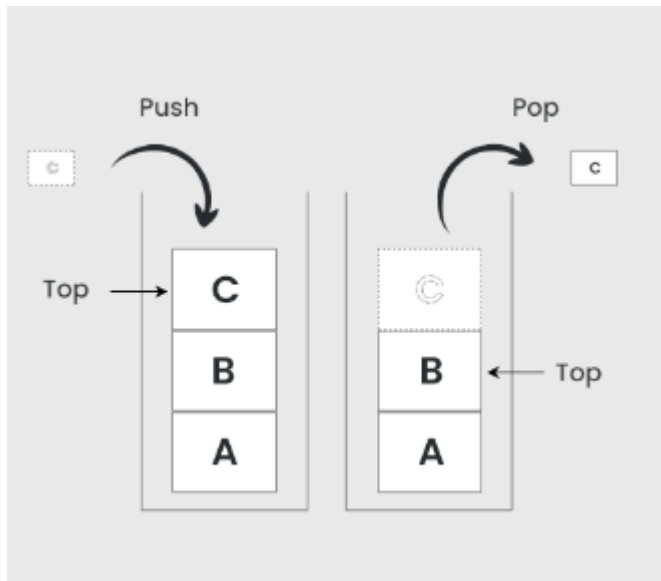
- *Task:* Is a piece of work to be done or undertaken.
- *Reminder:* A thing that causes someone to remember something
- *Hash Table:* Is a data structure that stores and organizes data by assigning “shortcuts” (unique keys) to items (values). They operate with hash functions that quickly translate each key into an index (slot number) where the value will be kept. In other words, it maps the key to the value, which makes it great for easily finding data in a computer program. A great example of hash tables are dictionaries, you have the first letter of the word as a key so you just have to look where the B's are. However this may cause collisions, when two values have the same key, in order to fix that, hash tables use buckets that could be arrays, linked lists, etc (they work like mini dictionaries).



- *Queue*: Is a linear data structure that is open at both ends and operates in First In First Out order (FIFO). A great example of this is the queue at a bank, the first one will be attended first.



- *Stack*: Is a linear data structure that follows the particular order in the way operations are performed. As opposed to the queue, stack work Last In First Out (LIFO) or First In Last Out (FILO). A good example of this is a stack of books, the last book you put at the top of the stack will be the first one you take out.



- *Heap*: Is a special data structure based on tree structure where the tree is completely binary. Heaps can be of two types:
 - Max Heap: The key at the root must be the greatest among the children. This has to be true for all the subtrees of the heap.
 - Min Heap: The key at the root must be the minimum among the keys of the children. This has to be true for all the subtrees of the heap.

Sources:

- A dictionary.
- <https://www.geeksforgeeks.org/hash-table-data-structure/>

Phase 3: Search of Creative Solutions

Alternative 1. Create a voice assistant

This solution allows users to manage their tasks and reminders through a voice assistant. It allows adding and organizing these activities in a simple way. Also it offers the possibility of categorizing into “Priority” and “Non-Priority” tasks, making It is easy for the user to organize them.

Alternative 2. Create a To do application

This solution offers users the ability to manage their tasks and reminders through a To-Do application. The application is developed in Java and uses a hash table structure to store tasks and reminders. In addition, it provides users with key function such as adding, modifying and deleting tasks. It also uses stack and queue structures to categorize tasks into “Priority” and “Non-Priority”, improving the organization and efficiency of task management.

Alternative 3.Creative an artificial vision

This solution allows users to interact with the application to add, modify and delete tasks and reminders using sign language recognition designed by the programmer with the help of artificial intelligence. The application is developed in java and uses a linked list structure to store tasks and reminders. In addition, it offers the functionality to categorize tasks as “Priority” and “Non-Priority” using different structures offered by Java for better organization.

Phase 4: Idea Formulation to Preliminary Design

After a detailed analysis of the alternatives, the first option of Create a voice assistant is discarded because it doesn't satisfy the essential requirements of storing tasks and reminders in a hash table for more efficient access. This application also doesn't offer the requirements to delete tasks and reminders. Finally this app uses other different structures for categorizing tasks.

The detailed review of the other alternatives leads us to the following:

Alternative 2. Create a To do application

- The application fulfill with the necessary structures, such as has tables, stacks and queues, as well as organization functionalities, to provide the user with an efficient user experience
- The application has all the requirements requested by the customer
- The hash table structure complies with the complexity $O(1)$, has it is key which is the identifier an the value the task or reminder information
- The application has an interface that allows interact with user

Alternative 3.Creative an artificial vision

- The application has all the requirements requested by the client, but uses different structures to categorize tasks according to their priority and uses a LinkedList to manage tasks and reminders, which could have a higher level of complexity.

Phase 5: Evaluation and Selection of the Best Solution

According to the problem at hand the evaluation of the remaining alternatives goes as following:

Alternative 2: Create a To-Do Application

Pros:

- Uses hash tables, stacks, and queues for efficient task management.
- Meets all the requirements specified by the client.
- Hash table structure provides $O(1)$ complexity for quick access.

- Includes an interactive interface easy for the user to use.

Cons: Relies on the user's and client's usage, might not be usable for more complex methods or features.

Alternative 3: Create an Artificial Vision Application

Pros: Meets all client requirements but uses different structures for the tasks and uses linked lists for task and reminder management.

Cons: Using linked lists could introduce higher complexity.

Based on the evaluation, the 2 Alternative (the To-Do Application) is a better solution because it's the most efficient choice as it works closely with the client's requirements and offers a simpler data structure (hash table) for task management.

Phase 6: Reports and Specifications Preparation

In order to present the best chosen solution to the client a document presenting the specified pros of the solution will be written. This way the client and the possible stakeholders will be able to see and examine themselves the reasons why the Alternative 2 was chosen.