METODO DE INGENIERIA INTEGRADORA 1

Daniel Montezuma A00382231

Diego Mueses A00382021

Computación y estructuras discretas

FASE1:

1. The system will be implemented to carry out the admissions, discharges, and records of the people, using a database as an aid that will allow the reception staff to know if the patient is found, what conditions he has, and if they have entered to any of the laboratory units. Patients will be treated according to their condition or priority to avoid serious problems.
2. We want to manage entries and exits in a clinical laboratory, doing the search or admission, and assigning a priority according to their condition. There are two laboratory units which are: Hematology and General Purpose. The people at the reception of the clinical laboratory will be carrying out the admissions and discharges of the people in any of the clinical laboratory units, in turn, they will be able to select in which laboratory unit to attend and which person to attend. In case of making a mistake, the receptionists of the clinical laboratory will have an undo option to be able to reverse the entry or exit.

3.1 The system Will show an option to register patients, in which we requested the basic data of the patient like: Name, last name, age, id, genre etc.

* 1. The system will show an option to search for a patient to make it easier for the receptionists to locate patients.
  2. The system Will show an option to entry a patient, this Will help to send patients to any of the laboratory units in the clinical laboratory.
  3. The system Will show an option for show the queue in the laboratory, this Will help the receptionists to know the priority in a queue in any unit of the laboratory.
  4. The system will show the option to attend patients, this option will be useful for receptionists to attend patients according to their priority. Attending a patient will remove a patient from the queue.
  5. The system will show the undo option, which will be useful to reverse an exit or entry action by the receptionist.

FASE2:

To solve the problems of the clinical laboratory we will make use of the different data structures in java and a database that will be updated every time an entry or exit is made by the receptionists.

HASHTABLE

A Hashtable is a data structure that uses a key as an identifier to do the respective functionality such as search or delete. We use Hashtable as our database, which will be used to search for patients or register them.

PRIORITY QUEUE

A priority queue is used to process objects according to their priority. Items in the priority queue will be sorted based on their priority. We used the priority queue data structure as our "Queue" in the different laboratory units, so that it allowed us to attend to patients according to their priority or condition.

STACK

The Stack data structure is basically a stack as its name implies. This works with LIFO principle (Last in first out). We use stack as a queue that works parallel to the priority queue in the lab units, but thanks to its LIFO principle, we use it to add the functionality of the undo button in the program

FASE3:

To reduce code lines, we could only use the priority queue data structure and add a delete method to its functionalities for when the patient is discharged due to any situation.

FASE4:

\* Use the stack as our “Queue”, since it uses a LIFO principle, which would make it impossible for receptionists to attend to the first patients to arrive on a busy day.

\* Hastable as a "queue" would not be a good option either, since it would make the patient care process very difficult due to its priority.

\* Using Stack as a database would not be a viable idea either, since this would make the process of searching for a patient by its LIFO principle difficult, to search for a patient who entered first, we would have to remove all those who arrived before him, to find it.

FASE5:

Criterion A efficiency. We can see it as a solution with better efficiency than the others, for example, using the priority queue data structure as our queue in the laboratories, is more efficient than if we used Stacks or HastablesCriterio

B Usability. It refers to how intuitive and easy it is to use the program.

Criterion C Completeness. It refers to the fact that we have a solution that finds all the solutions stated in the problem. Thanks to the use of different data structures in the code we can efficiently found a solution to all.