METODO DE INGENIERIA INTEGRADORA 1

Daniel Montezuma A00382231

Diego Mueses A00382021

Computación y estructuras discretas

FASE1:

1. Se implementará el sistema para llevar a cabo con los ingresos, egresos y registros de las personas según sea el caso, usando como ayuda una base de datos que permitirá al personal en recepción, saber si se encuentra el paciente, que condiciones tiene, y si ha ingresado a algunas de las unidades del laboratorio. Se atenderán a los pacientes según su condición o prioridad para evitar problemas más graves.
2. Se desea realizar la administración de entradas y salidas en un laboratorio clínico, realizando su búsqueda o ingreso según sea el caso, y asignándole una prioridad según su condición, se cuenta con dos unidades de laboratorio los cuales son: Hematología y Propósito General. Las personas en la recepción del laboratorio clínico serán los encargados de realizar los ingresos y egresos de las personas en cualquiera de las unidades del laboratorio clínico, a su vez, estos podrán seleccionar en que unidad del laboratorio atender y a que persona atender. En caso de cometer un error, los recepcionistas del laboratorio clínico contaran con una opción de deshacer para poder reversar el ingreso o egreso según sea el caso.
3. 3.1 El sistema deberá mostrar una opción de registro de pacientes, en la que se solicitaran datos básicos del paciente como: nombre, apellido, edad, documento, genero etc.
   1. El sistema mostrar una opción de buscar un paciente para facilitar la tarea a los recepcionistas a la hora de ubicar a los pacientes.
   2. El sistema mostrará la opción de realizar ingreso, que servirá para enviar al paciente a algunas de las unidades del laboratorio.
   3. El sistema mostrará la opción de mostrar fila en laboratorio, la cual servirá para guiar al recepcionista y saber el estado de la cola en alguna de las unidades del laboratorio.
   4. El sistema mostrara la opción de atender paciente, esta opción será útil para que los recepcionistas atiendan a los pacientes según su prioridad. Al atender un paciente este será eliminado de la fila.
   5. El sistema mostrara la opción de deshacer, que servirá para revertir una acción de egreso o ingreso por el recepcionista según sea el caso.

FASE2:

Para resolver los problemas del laboratorio clínico haremos uso de las diferentes estructuras de datos en java y una base de datos que se actualizara cada que se realiza un ingreso o egreso por parte de los recepcionistas.

HASHTABLE

Un Hashtable es una estructura de datos que utiliza una llave o clave como identificador para hacer las respectivas funcionalidades como buscar o eliminar. Usamos Hashtable como nuestra base de datos, la cual servirá para buscar los pacientes o registrarlos.

PRIORITY QUEUE

Una cola de prioridad es usada para procesar objetos según su prioridad. Los elementos en la cola de prioridad serán ordenados en base a su prioridad. Usamos la estructura de datos priority queue como nuestra “Fila” en las diferentes unidades del laboratorio, de manera que nos permitía atender a los pacientes según su prioridad o condición.

STACK

La estructura de datos Stack es básicamente una pila como su nombre lo indica. Esta funciona de manera LIFO (Last in first out).Usamos stack como una fila que funciona paralelo al priority queue en las unidades del laboratorio, pero esta gracias a su principio de LIFO, la utilizamos para agregar la funcionalidad del botón deshacer en el programa.

FASE3:

Para resumir más el código se podría hacer uso únicamente de la estructura de datos priority queue y añadir a sus funcionalidades un método de eliminar para cuando el paciente sea egresado por cualquier situación.

FASE4:

\* Usar el stack como nuestra “Fila”, ya que esta utiliza un principio de LIFO, lo cual imposibilitaría a los recepcionistas atender a los primeros pacientes en llegar en un día de mucha atención.

\* Hastable como “fila” tampoco sería una buena opción ya que dificultaría mucho el proceso de atención a pacientes por su prioridad.

\* Usar Stack como base de datos tampoco seria una idea viable, ya que este dificultaría el proceso de buscar a un paciente por su principio de LIFO, para buscar un paciente que entro de primero, tendríamos que sacar a todos los que llegaron antes que él, hasta encontrarlo.

\* Usar Priority Queue como base de datos del sistema no seria viable, ya que para buscar un paciente que no tenga una prioridad alta, tendríamos que buscar y sacar a todos los pacientes antes que el que tienen una prioridad mayor hasta llegar a él.

FASE5:

Criterio A eficiencia. La podemos ver como una solución con mejor eficiencia que las demás, por ejemplo, el utilizar La estructura de datos como nuestra fila en los laboratorios, es mas eficiente que si utilizaramos Stacks o Hastables

Criterio B Usabilidad. Se refiere a lo intuitivo y fácil que resulta usar el programa.

Criterio C Completitud. Se refiere a que tengamos una solución que encuentre todas las soluciones planteadas en el problema. Gracias al uso de diferentes estructuras de datos en el código podemos entrar de manera eficiente una solución a todos.