**Método de diseño de ingeniería**

**Identificación del problema:**

Las necesidades propias del contexto del problema, junto con sus condiciones de resolución son las siguientes:

* El personal de la recepción de una Institución Prestadora de Salud precisa administrar el ingreso y egreso de pacientes al Laboratorio Clínico.
* No existe ningún sistema, dentro de la Institución Prestadora de Salud, que permita la administración del ingreso y egreso de los pacientes al Laboratorio Clínico.
* Existen personas con prioridad y personas sin prioridad.
* Las personas con mayor prioridad deben ingresar antes que las personas de menor prioridad.
* La solución debe garantizar la eficiencia en el registro de los pacientes.
* La solución debe garantizar la eficiencia en la búsqueda de pacientes.
* La solución debe permitir deshacer acciones de ingreso y egreso.

Definición del problema:

Una reconocida Institución Prestadora de Salud requiere un sistema que permita administrar el ingreso y egreso de pacientes en un laboratorio clínico; este debe permitir organizar los pacientes entre acceso general y prioritario para ser atendidos respectivamente.

**Recopilación de información:**

* Satck:

Es una lista o estructura de datos que permite almacenar y recuperar datos. Su principal característica es su modo de acceso a los elementos, el cual es el tipo L.I.F.O (Last in, First out, o Último en Entrar, Primero en Salir).

* Queue:

Es una estructura de datos, que sirve para almacenar elementos. Su característica principal es su modo de acceso a los elementos, el cual es el tipo F.I.F.O (First in, First Out, o Primero en Entrar, Primero en Salir).

* Colas de prioridad:

Es una estructura de datos, que sirve para almacenar elementos, y que funciona tal como lo hace una cola normal, pero incluye la diferencia de que, en esta, cada elemento almacenado tiene una prioridad asignada. Su principal característica radica en que los elementos con mayor prioridad salen antes, de la cola, que los de menor prioridad. Mientras que los elementos con prioridad igual salen, de la cola, siguiendo el orden de esta. Además de esto, es importante agregar que su implementación se realiza mediante los montículos.

* Montículo:

Es una estructura de dato, que sirve para almacenar elementos, que suele ser un arreglo que puede verse como un ABB (Árbol Binario de Búsqueda). En esta estructura, cada nodo del ABB es un elemento dentro del arreglo.

Es importante decir que en realidad no es un ABB, es solo un arreglo de elementos que nos imaginamos como un ABB. Además de esto, lo vemos como un ABB casi completo puesto que puede no estar lleno en su nivel más bajo. En caso de no estar lleno en su nivel más bajo, siempre va a estar lleno, desde la izquierda, hasta su nivel más bajo.

* Tabla hash:

Es una estructura de datos, que sirve para almacenar elementos, y que implementa otra estructura de datos llamada Diccionario. Esta estructura asocia claves con valores. Una de sus características principales es el tiempo promedio de sus operaciones es , sobre todo en la búsqueda, lo que la hace una opción muy viable cuando se busca la eficiencia en esta. Funciona transformando la clave con una función hash en un hash, un número que identifica la posición donde la tabla hash localiza el valor deseado.

* Linked-List:

Es una colección lineal de elementos de datos en donde cada elemento apunta a un elemento siguiente y, opcionalmente, a un elemento anterior. Es, en otras palabras, una colección de nodos que, al juntarse, constituyen una secuencia.

Esta estructura puede usarse para implementar otro tipo de estructuras tales como tablas hash, pilas, colas, etc.

**Búsqueda de soluciones:**

Alternativa 1:

Hacer uso de las colas de prioridad para asignar turno a los pacientes con mayor prioridad, colas para asignar turno a las personas que no tienen ninguna prioridad, después hacer uso de las tablas hash para agruparlos a todos. También hacer uso de las Stacks o pilas para implementar la opción de deshacer las acciones realizadas.

Alternativa 2:

Hacer uso de una sola cola de prioridad para ingresar a todos los pacientes y asignarles su turno, y tabla hash para guardar a todos los pacientes. También hacer uso de las Stacks o pilas para implementar la opción de deshacer la última acción realizada.

Alternativa 3:

Hacer uso de colas de prioridad para asignar a los pacientes con mayor prioridad su turno y así mismo usar colas sencillas para asignar a los demás pacientes en atención general. También hacer uso de las Stacks o pilas para implementar la opción de deshacer la última acción realizada.

**Transición de la formulación de ideas a los diseños preliminares:**

Como primera medida se opta por descartar la Alternativa número 3 debido a que la consideramos la más incompleta para implementar una solución sólida y factible haciendo uso de las buenas prácticas para una implementación de software, ya que al implementar solo dos, colas una de prioridad y una sencilla, sin hacer uso de una estructura de datos como la tabla hash, hace del algoritmo mucho mas pesado e ineficiente a largo plazo.

Revisión de las otras alternativas:

Alternativa 1:

* Es preciso y puede tener buenas prácticas en el proceso de su implementación. Sin embargo, puede llegar a ser extenso y propenso a caer en malas prácticas si no se tiene en cuenta cómo implementar las colas de prioridad de una manera correcta.
* Esta alternativa implica implementar una tabla hash propia, así como también la cola de prioridad, la cola, y la pila, para su correcto funcionamiento.

Alternativa 2:

* Al hacer uso de una sola cola de prioridad estamos delegando muchas funciones a una sola parte del sistema, lo que haría que nos encontráramos con malas prácticas, violando los criterios de diseño de software como lo son la partición de obligaciones.

**Evaluación y selección de la mejor solución:**

Criterios de evaluación:

* Criterio A.

Cumplimiento de los criterios de diseño.

* Criterio B.

Flexibilidad. Solución flexible y escalable a futuro.

* Criterio C.

Completitud. Hace uso de todo lo necesario para ser funcional y sostenible a futuro.

Modo de evaluación para cada una:

Numérica, 0 siendo el que menos cumple con el criterio y 5 si cumple completamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Criterio A | Criterio B | Criterio C | Total |
| Alternativa 1 | 4 | 4 | 5 | 13 |
| Alternativa 2 | 2 | 3 | 2 | 7 |

Selección:

De acuerdo con la evaluación anterior, se debe seleccionar la alternativa 1, ya que obtuvo la mayor puntuación de acuerdo con los criterios definidos.

**Especificación de requerimientos:**

**R1**: El programa deberá permitir al personal de recepción registrarse con sus datos básicos.

**R2**: El programa deberá permitir al personal de recepción iniciar sesión.

**R3**: El programa deberá permitir al personal de recepción ingresar al paciente con su información completa en una base de datos.

**R4**: El programa deberá permitir al personal de recepción registrar el ingreso del paciente en una base de datos.

**R5**: El programa deberá permitir al personal de recepción dirigir al paciente a una de las dos unidades del laboratorio (Hematología y Propósito general).

**R6**: El programa deberá permitir al personal de recepción buscar un paciente en la base de datos.

**R7**: El programa deberá permitir al personal de recepción asignar alguna prioridad a un paciente de la base de datos

**R8**: El programa deberá permitir al personal de recepción ingresar con prioridad a los pacientes prioritarios (aquellos que presenten alguna enfermedad de base importante, sean adultos mayores, estén en embarazo, etc. Esta información se extrae a partir de la base de datos del laboratorio).

**R9**: El programa deberá permitir al personal de recepción ingresar a los pacientes de acceso general por orden de llegada.

**R10**: El programa deberá permitir al personal de recepción hacer el egreso de los pacientes que anteriormente habían ingresado.

**R11**: El programa deberá permitir al personal de recepción deshacer la acción de ingresar o egresar paciente en el caso de una equivocación.

**R12**: El programa deberá contar con un panel que permitirá monitorear al personal de recepción en todo momento la lista de personas actualmente en el laboratorio.

**R13**: El programa deberá contar con un panel que permitirá monitorear al personal de recepción en todo momento el orden de atención de las personas en cada unidad.

**R14**: El programa deberá contar con un panel que permitirá monitorear al personal de recepción en todo momento el egreso de los pacientes para poder continuar con la atención de las siguientes personas.

**BONUS:**

**R15**: El programa deberá lograr que en un tiempo aleatorio entre 1 o 2 minutos se realice el egreso de un paciente dentro de las unidades.

**R16**: El programa deberá lograr que la persona atendida automáticamente por la unidad no aparezca en lista de espera, pero todavía deberá aparecer presente en el sistema para que posteriormente el personal de recepción realice el egreso de la persona manualmente.