

Identificación del problema

Se necesita de un programa de software que ayude a los recepcionistas a de un laboratorio clínico a llevar a cabo la debida segmentación de los pacientes dependiendo a que modulo del laboratorio se dirigen y si necesitan de una oportuna atención prioritaria.

Recopilación de información

Hace falta saber cómo se les da prioridad a los enfermos, para esto, se investigó el método de triage que se utiliza en los hospitales, así como ejemplos de cómo se da prioridad los pacientes en los laboratorios clínicos. También se investigó la debida normativa para estas instituciones de salud.

Hospitales: metodología de triage

https://www.minsalud.gov.co/Normatividad_Nuevo/Resoluci%C3%B3n%205596%20de%202015.pdf

https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1137-66272010000200008#f1

https://udea.edu.co/wps/portal/udea/web/inicio/udea-noticias/udea-noticia/!ut/p/z0/fYw9D4JADlb_iguji6Yl46kqcTlyDgzHqXTRwgSpcgTuJP99DB-Pi0rwtT19AyAAAtjVyRZ7HUBJ-jvm62u3iRJugodKJVqk_Jah3vl-eLggPgfyAs8K3vMQUsxHrz9JB1MnhqHqWhSJH7dbW05qOnO7PiuWBykXp_Wy5lor7xyGVgnRnGElbm9YMIbGhhO6O-Qv2SbSL/

Laboratorios clínicos: ejemplos

<https://acreditacionensalud.org.co/wp-content/uploads/2021/08/Resolucion-207-de-2020-Adopcion-manual-acreditacion-laboratorio-V2.0.pdf>

<http://hospitalguarne.com/images/pdf/6.-Politica-de-Priorizacion.pdf>

<https://www.laboratoriotoxicologico.co/priorizacion-del-paciente/>

Requerimientos funcionales

Nomenclatura:

INGRE: Proceso de ingreso

ATEN: Tipo de atención

MON: Monitoreo

INGRE1: El personal de la clínica de poder ingresar los datos de un paciente, y estos serán guardados en la base de datos.

INGRE2: Si el paciente ya se encuentra en la base de datos el programa debe poder buscarlo y encontrarlo de la manera más eficiente posible.

INGRE3: En el caso requerirlo, el programa podrá deshacer la información suministrada de alguno de los pacientes ingresados en la base de datos.

ATEN1: Dependiendo de la información del paciente el programa debe poder decidir si recibirán una atención de tipo prioritario o general. Prioritario en caso de que el paciente: presente alguna enfermedad de base importante, sea adulto mayor o esté en embarazo.

Diseño de escenarios y casos de pruebas.

Diseño de los escenarios para la estructura de datos Hash Table.

Nombre	Clase	Escenario
stage1()	HashTableTest	Un objeto de la clase HashTable con tipo de dato String para el manejo de las claves y tipo de dato Patient para el manejo de los valores. Además, dicho objeto se encuentra vacío.
stage2()	HashTableTest	Un objeto de la clase HashTable con tipo de dato String para el manejo de claves y tipo de dato Patient para el manejo de los valores. Este objeto contiene 5 objetos de tipo HashNode que tienen el mismo tipo de dato para las claves y los valores que la clase HashTable . Los 5 objetos de tipo HashNode están compuestos por los siguientes datos: Objeto HashNode no.1: Clave: "1006048521", Valor: Patient = {

		<p>Nombre = "Jun", Id = "1006048521", Edad = 21, Dirección = "Candelaria", Correo = "david_jhun@outlook.es", Estado de embarazo = false, Enfermedad grave = false, Estado de discapacidad = false, Dependiente de oxígeno = false, Fecha de ingreso = Fecha }</p> <p>Objeto HashNode no.2: Clave: "1234567890", Valor: Patient = { Nombre = "Pepe", Id = "123456789", Edad = 30, Dirección = "Cali", Correo = "pepe@gmail.com", Estado de embarazo = false, Enfermedad grave = true, Estado de discapacidad = false, Dependiente de oxígeno = true, Fecha de ingreso = Fecha }</p> <p>Objeto HashNode no.3: Clave: "1122334455", Valor: Patient = { Nombre = "Cristian", Id = "1122334455", Edad = 18, Dirección = "Pasto", Correo = "cristian@outlook.com", Estado de embarazo = false, Enfermedad grave = false, Estado de discapacidad = false,</p>
--	--	--

		<p>Dependiente de oxígeno = true, Fecha de ingreso = Fecha }</p> <p>Objeto HashNode no.4: Clave: "5566778899", Valor: Patient = { Nombre = "Jose", Id = "5566778899", Edad = 40, Dirección = "Bogota", Correo = "jose@gmail.com", Estado de embarazo = false, Enfermedad grave = false, Estado de discapacidad = false, Dependiente de oxígeno = false, Fecha de ingreso = Fecha }</p> <p>Objeto HashNode no.5: Clave: "1011121314", Valor: Patient = { Nombre = "Luigi", Id = "1011121314", Edad = 25, Dirección = "Santander", Correo = "luigi@hotmail.com", Estado de embarazo = false, Enfermedad grave = false, Estado de discapacidad = false, Dependiente de oxígeno = true, Fecha de ingreso = Fecha }</p>
stage3()	HashTableTest	<p>Un objeto de tipo HashTable con tipo de dato String para el manejo de claves y valores. Dicho objeto contiene 26 objetos de tipo HashNode con tipo de dato String</p>

		<p>para el manejo de las claves y los valores.</p> <p>Las claves y valores de los objetos son los siguientes:</p> <p>{“456”, “Maria”}, {“789”, “Gonzalo”}, {“mario”, “Casimiro”}, {“luigi”, “Jose”}, {“135”, “Mario”}, {“pt”, “Jaime”}, {“p1”, “David”}, {“p2”, “Valery”}, {“David”, “Jun”}, {“Oclivia”, “Oclivia”}, {“Laura”, “Laura”}, {“Valentina”, “Valentina”}, {“Myeong gu”, “Myeong gu”}, {“Cubides”, “cubides”}, {“oscar”, “oscar”}, {“sebastian”, “sebastian”}, {“gonzalo”, “Varona”}, {“ramiro”, “ramiro”}, {“luisa”, “luisa”}, {“maine”, “maine”}, {“lucy”, “lucy”}, {“dorio”, “dorio”}, {“faraday”, “faraday”}, {“Adam”, “Adam Smasher”}, {“cortana”, “Cortana”}, {“117”, “John”}</p>
--	--	---

Diseño de los casos de prueba para la estructura de datos Hash Table.

Objetivo de la prueba: Comprobar el correcto funcionamiento del método insert de la clase HashTable .				
Clase	Método	Escenario	Valores de entrada	Resultado
HashTableTest	testInsert1()	stage1()	<p>Un objeto de tipo HashNode con clave: “1020304050” y valor:</p> <p>Patient = {</p> <p>Nombre = “Gloria”,</p> <p>Id = “1020304050”,</p> <p>Edad = 34,</p> <p>Dirección = “Medellin”,</p>	<p>El objeto HashNode es agregado de manera exitosa a la tabla.</p>

			Correo = "gloria@gmail.com", Estado de embarazo = true, Enfermedad grave = false, Estado de discapacidad = false, Dependiente de oxígeno = true, Fecha de ingreso = Fecha }	
--	--	--	---	--

Objetivo de la prueba: Comprobar el correcto funcionamiento del método **insert** de la clase **HashTable** cuando hay nodos agregados en la tabla.

Clase	Método	Escenario	Valores de entrada	Resultado
HashTableTest	testInsert2()	stage2()	Un objeto de tipo HashNode con clave: "2030405060" y valor: Patient = { Nombre = "Kevin", Id = "2030405060", Edad = 17, Dirección = "Villagorgona", Correo = "Ninguno", Estado de embarazo = false, Enfermedad grave = true, Estado de discapacidad = false,	El objeto HashNode fue agregado de manera exitosa a la tabla.

			Dependiente de oxígeno = true, Fecha de ingreso = Fecha }	
--	--	--	---	--

Objetivo de la prueba: Comprobar el correcto funcionamiento del método **insert** de la clase **HashTable** cuando hay objetos agregados a la tabla y se requiere manejo de colisiones.

Clase	Método	Escenario	Valores de entrada	Resultado
HashTableTest	testInsert3()	stage3()	Cuatro objetos de tipo HashNode . Objeto número 1: Clave "123" y valor "Luigi". Objeto número 2: Clave "Kim" y valor "Kim". Objeto número 3: Clave "rebecca" y valor "rebecca". Objeto número 4: Clave "pilar" y valor "pilar".	Los objetos de tipo HashNode fueron agregados de manera exitosa con manejo de colisiones.

Objetivo de la prueba: Comprobar el correcto funcionamiento del método **search** de la clase **HashTable** cuando se introduce una clave.

Clase	Método	Escenario	Valores de entrada	Resultado
HashTableTest	testSearch1()	stage2()	Un objeto de tipo String con valor "1011121314".	Se retorna de manera exitosa el objeto de tipo HashNode con la clave introducida.

Objetivo de la prueba: Comprobar el correcto funcionamiento del método **search** de la clase **HashTable** cuando se introduce una clave y el objeto buscado es manejado por colisiones en una lista doblemente enlazada.

Clase	Método	Escenario	Valores de entrada	Resultado
HashTableTest	testSearch2()	stage3()	Un objeto de tipo String con valor "sebastian".	Se retorna de manera exitosa el objeto de tipo HashNode con la clave introducida cuando este fue manejado por colisiones en una lista doblemente enlazada.

Objetivo de la prueba: Comprobar el correcto funcionamiento del método **search** de la clase **HashTable** cuando se introduce una clave; el objeto buscado es manejado por colisiones y se agregan nuevos objetos a la tabla.

Clase	Método	Escenario	Valores de entrada	Resultado
HashTableTest	testSearch3()	stage3()	Un objeto de tipo String con valor "rebecca".	Se retorna de manera exitosa el objeto de tipo HashNode con la clave introducida cuando este fue manejado por colisiones en una lista doblemente enlazada y se agregaron nuevos objetos.

Objetivo de la prueba: Comprobar el correcto funcionamiento del método **delete** de la clase **HashTable** cuando se introduce una clave.

Clase	Método	Escenario	Valores de entrada	Resultado
HashTableTest	testDelete1()	stage2()	Un objeto de tipo String con	El objeto de tipo HashNode es eliminado

			valor "9334234323".	de manera exitosa de la tabla.
--	--	--	------------------------	--------------------------------------

Objetivo de la prueba: Comprobar el correcto funcionamiento del método **delete** de la clase **HashTable** cuando se introduce una clave y el objeto a eliminar es manejado por colisiones en una lista doblemente enlazada.

Clase	Método	Escenario	Valores de entrada	Resultado
HashTableTest	testDelete2()	stage3()	Un objeto de tipo String con valor "ramiro".	El objeto de tipo HashNode es eliminado de manera exitosa de la tabla cuando este estaba manejado por colisiones en una lista doblemente enlazada.

Objetivo de la prueba: Comprobar el correcto funcionamiento del método **delete** de la clase **HashTable** cuando se introduce una clave; el objeto a eliminar es manejado por colisiones en una lista doblemente enlazada y se agregan nuevos objetos a la tabla.

Clase	Método	Escenario	Valores de entrada	Resultado
HashTableTest	testDelete3()	stage3()	Un objeto de tipo String con valor "Kim".	El objeto de tipo HashNode es eliminado de manera exitosa de la tabla cuando este estaba manejado por colisiones en una lista doblemente enlazada y se agregaron nuevos objetos.