ALGORITMOS Y ESTRUCTURAS DE DATOS

Sección: SV31

Presentación del Trabajo Final

Hito 2

Profesor del curso:

Walter Cueva Chavez

Integrantes:

- Ramos Terrones, Breydi
- Mamani Rojas, Brayan German

Lima, junio del 2020

Introducción

En la actualidad, el análisis de los datos es un factor importante en la toma de decisiones de las empresas, motivo por el cual, la obtención de datos y la gestión de los mismos se ha convertido en una pieza fundamental en la competición de las empresas. El uso de datos, en estos últimos años, se ha ido aumentando exponencialmente, debido al avance en las capacidades tecnológicas y a la llegada de internet.

Estos sistemas de información, actualmente se basan en base de datos (BD) y sistemas de bases de datos (SGBD). De este modo, se ha logrado consolidar millones de datos de información que son almacenados para su posterior análisis. Por ello, el presente proyecto consistirá en desarrollar un mini-SGDB, un mini motor de Base de datos que almacena archivos, siendo esta una estructura de datos tabular bidimensional con ejes dispuestos en filas y columnas.

Problema

Desarrollar un mini gestor de base de datos, que permita manejar una abundante cantidad de datos, de forma dinámica. Además de contar con opciones que permitan realizar el análisis de datos de forma eficaz, filtro de datos, ordenamiento, búsquedas y similares.

Objetivos

- Desarrollar un mini gestor de base de datos, enfocado en la programación orientada a objetos. Considerando que la generación o creación de los datos será de forma dinámica.

- Elaboracion de algoritmos que permitan gestionar los datos, para su posterior análisis. Es decir, brindar opciones que permitan ordenar, buscar y gestionar los datos de forma eficiente, con el menor coste computacional posible.

Alcance de Proyecto

El presente proyecto tiene como fin emplear todos los temas presentados durante el presente ciclo, con el fin de demostrar lo aprendido. Para lo cual, este consta de tres entregables, los cuales se definen en el punto “Plan de trabajo”. Los recursos que vamos a utilizar para desarrollar este proyecto son:

- Hash table
- Árboles Binario
- Algoritmos de búsqueda y ordenamiento
- Programación orientada a objetos

Dichos recursos se desarrollaran en un periodo de cuatro semanas, los que están adjuntos a al cronograma de trabajo.

Marco conceptual

Estructura de datos:

Con estructura de datos, nos referimos a los medios en que se pueden organizar los datos, de modo que el uso de los mismo sea eficiente, de este modo, se logra que el manejo de enormes cantidades de datos tengan el menor coste computacional posible. Para ello, es necesario la implementación de algoritmos, de tal modo que estén diseñados para gestionar datos en un panorama dado. Entre las estructuras de datos más conocidas se encuentran los arreglos, árboles, listas, pilas, colas, Hash Tables. Cada una de estas tiene sus ventajas como desventajas, por ello, es esencial utilizar cada una de ellas para un fin específico, teniendo en cuenta sus limitaciones.

Base de Datos

Para entender una base de datos, es necesario identificar a qué nos referimos por darto. Un dato es un elemento, ya sea una palabra o número que independiente al contexto no tiene significado. De este modo, una base de datos, consiste en una colección o agrupación de múltiples datos relacionados entre sí, en un marco concreto, esto permite gestionar esta inmensa cantidad de datos, de tal modo que se puedan almacenar de forma organizada, para su posterior análisis.

Hash table

En informática , una tabla hash (mapa hash) es una estructura de datos que implementa un tipo de datos abstracto de matriz asociativa , una estructura que puede asignar claves a valores . Una tabla hash usa una función hash para calcular un índice, también llamado código hash , en una matriz de cubos o ranuras , desde donde se puede encontrar el valor deseado. Durante la búsqueda, la clave se codifica y el resumen resultante indica dónde se almacena el valor correspondiente.

Programación orientada a objetos

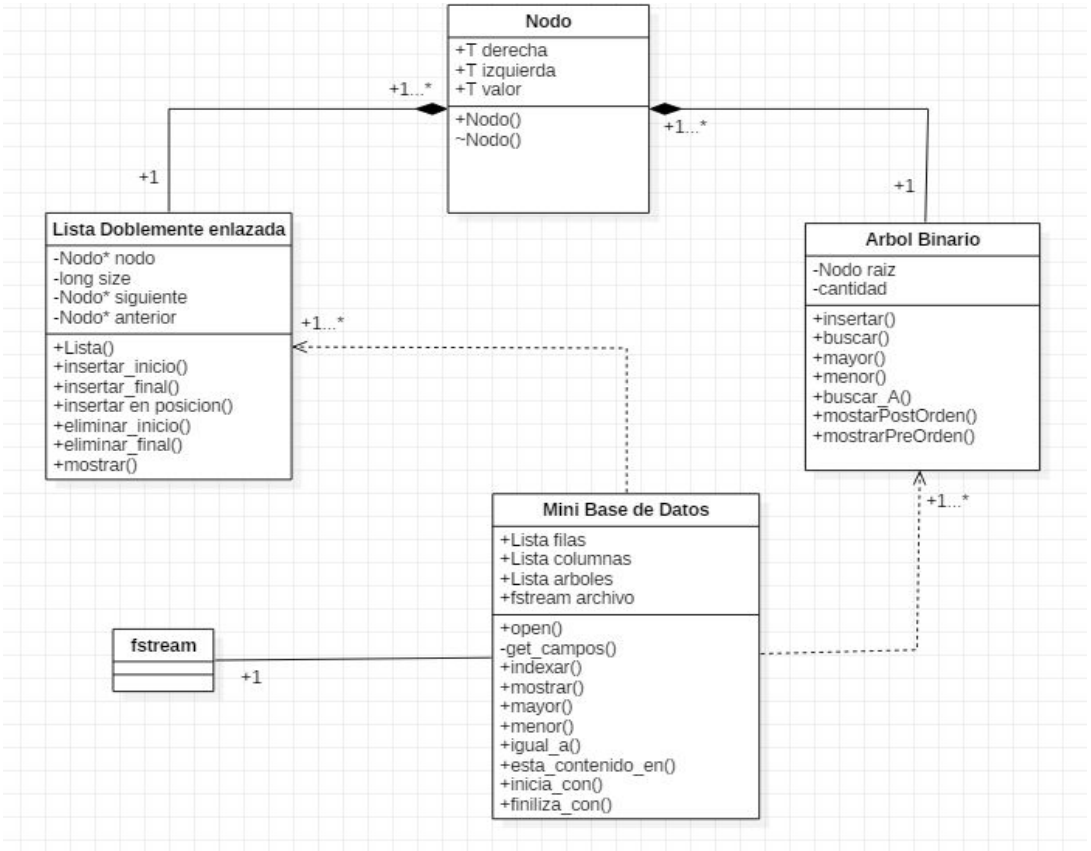
La Programación Orientada a Objetos (POO, en español; OOP, según sus siglas en inglés) es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial. Muchos de los objetos prediseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.

Está basada en varias técnicas del sexenio: herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Su uso se popularizó a principios de la década de 1990. En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos.

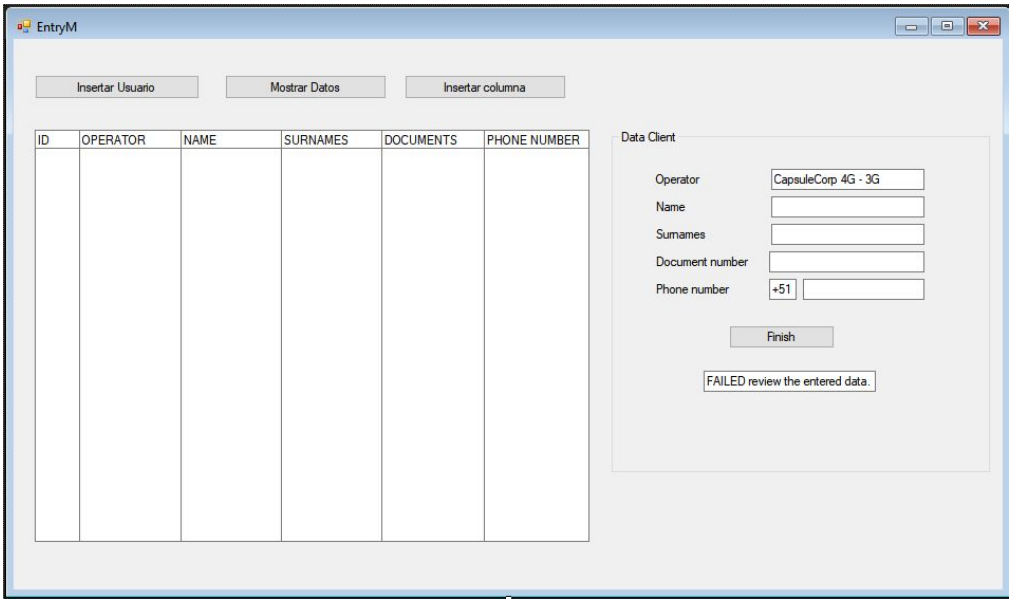
Algoritmos de ordenamiento

En computación y matemáticas un algoritmo de ordenamiento es un algoritmo que pone elementos de una lista o un vector en una secuencia dada por una relación de orden, es decir, el resultado de salida ha de ser una permutación —o reordenamiento— de la entrada que satisfaga la relación de orden dada. Las relaciones de orden más usadas son el orden numérico y el orden lexicográfico. Ordenamientos eficientes son importantes para optimizar el uso de otros algoritmos (como los de búsqueda y fusión) que requieren listas ordenadas para una ejecución rápida. También es útil para poner datos en forma canónica y para generar resultados legibles por humanos.

Diagrama de clases (Preliminar)



Diseño de interfaz de usuario.



Definición de TDA y estructuras de datos a usar

Arboles binarios

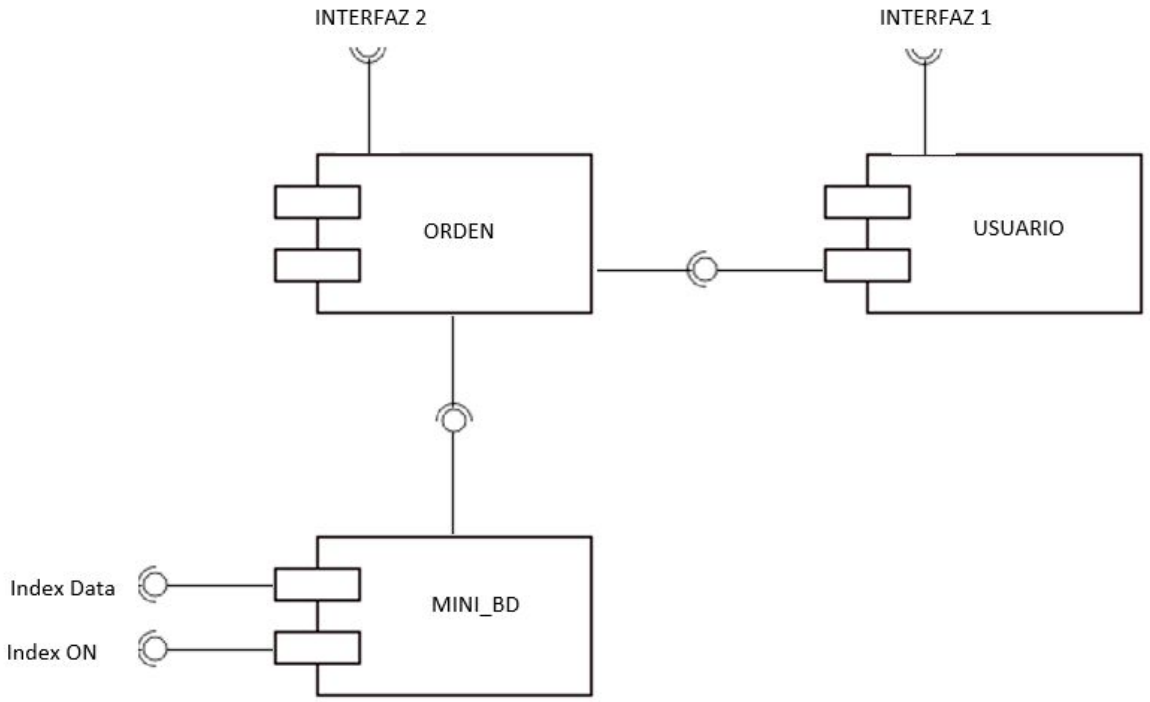
En ciencias de la computación, un árbol binario es una estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho. No pueden tener más de dos hijos (de ahí el nombre "binario"). Si algún hijo tiene como referencia a null, es decir que no almacena ningún dato, entonces este es llamado un nodo externo. En el caso contrario el hijo es llamado un nodo interno. Usos comunes de los árboles binarios son los árboles binarios de búsqueda, los montículos binarios y Codificación de Huffman.

Listas

En ciencias de la computación, una lista enlazada es una de las estructuras de datos fundamentales, y puede ser usada para implementar otras estructuras de datos. Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior. El principal beneficio de las listas enlazadas respecto a los vectores convencionales es que el orden de los elementos enlazados puede ser diferente al orden de almacenamiento en la memoria o el disco, permitiendo que el orden de recorrido de la lista sea diferente al de almacenamiento.

Una lista enlazada es un tipo de dato auto referenciado porque contienen un puntero o enlace (en inglés *link*, del mismo significado) a otro dato del mismo tipo. Las listas enlazadas permiten inserciones y eliminación de nodos en cualquier punto de la lista en tiempo constante (suponiendo que dicho punto está previamente identificado o localizado), pero no permiten un acceso aleatorio. Existen diferentes tipos de listas enlazadas: listas enlazadas simples, listas doblemente enlazadas, listas enlazadas circulares y listas enlazadas doblemente circulares.

Diseño de componentes



Plan de trabajo detallado (Proyecto, Milestones, Issues)

Para la entrega del trabajo final, este constará de tres entregables. A Continuación se detalla cada uno de ellos.

Entregable 1 - Hito 1	Entregable 2 - Hito 2	Entregable 3 - Hito 3
● Introducción ● Problema ● Objetivos ● Alcance de Proyecto ● Marco conceptual ● Diagrama de clases (preliminar) ● plan de trabajo detallado ● Cronograma	● Asignación de recursos ● Definición de requisitos. Diseño de interfaz de usuario ● Tipos de datos abstractos ● Selección de estructuras de datos ● Implementación de las estructuras de datos ● Diagrama de clases (Preliminar) ● código fuente(preliminar) ● Implementación de clases entidad ● Prototipo de la aplicación	● Diagrama de clases (version ● Final) ● Implementación de todas las funcionalidades del Mini-SGDB ● Diseño de archivos ● Conclusiones ● Referencias ● código fuente(Final) ● Implementación de almacenado de datos en archivos

Cronograma

Entregable	Semana 12	Semana 13	Semana 15
Hito 1	x	x	x
Hito 2		x	x
Hito 3			x