

**Universidad Peruana de Ciencias Aplicadas**

**Carrera**: Ingeniería de Software

**Ciclo**: 2020-01

**Nombre del curso**: Algoritmos y estructura de datos

**Sección**: SV31

**Nombre del profesor**: Walter Cueva Chavez

 "Informe de Trabajo Final" 

**Alumnos:**

Diaz Torres, Sebastian

Toulier Funes, Sebastian

Lima - Perú

**Mini base de datos utilizando estructura de datos y algoritmos**

Introducción:

En el presente informe hablaremos acerca de un programa, el cual es una mini-SGDB, desarrollada en el lenguaje de programación C++. Esta solución nos ha servido para aplicar nuestros conocimientos acerca de la programación orientada a objetos y, a la vez, las estructuras de datos y algoritmos de ordenamiento aprendidos a lo largo del ciclo. Para ello, es necesario precisar las labores que realiza el computador. Uno de los objetivos de este es de procesar, almacenar y dar acceso a una gran cantidad de información. Por lo cual se usan estructuras de datos, dependiendo su finalidad pueden ser: listas, pilas, colas, arboles binarios, grafos, tablas hash, etc. Por otra parte, las estructuras de datos permiten construir, de forma eficiente, algoritmos para una gran variedad de problemas.

Objetivos:

El principal objetivo de este trabajo es construir tipos de datos abstractos y algoritmos, teniendo en cuenta las restricciones impuestas por los recursos computacionales. Otro objetivo de este trabajo es el fortalecer nuestros conocimientos en la implementación de diversas estructuras de datos, así como los algoritmos pertinentes, de tal forma que la solución funcione en su totalidad.

Marco conceptual:

Definición de conceptos

1.3.1 Base de datos

Una base de datos es una colección de datos organizada en un sistema de archivos electrónicos. Esta permite el acceso directo a los datos almacenados y, también, permite que un programa o conjunto de programas puedan gestionar los datos almacenados. Según Rafael Campus Paré:

*“…una base de datos de un SI es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del SI y de sus interrelaciones. Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos.”*

1.3.2 Estructura de datos

Una estructura de datos es la colección de datos, distribuidos es un esqueleto. El cual puede ser: Pilas, colas, arboles binarios, grafos, tablas hash, arreglos, vectores, etc. Según Luis Joyanes, en su libro *Fundamentos de programación,* se puede definir las estructuras de datos de la siguiente manera:

*“Una estructura de datos es una colección de datos que pueden ser caracterizados por su organización y las operaciones que se definen en ella. Las estructuras de datos son muy importantes en los sistemas de computadora. Los tipos de datos más frecuentes utilizados en los diferentes lenguajes de programación son: datos simples y datos estructurados.”*

1.3.3 Algoritmos

Thomas Cormen, en su libro *Introduction to algorithms,* da la siguiente definición:

*“Informalmente, un algoritmo es cualquier procedimiento computacional bien definido que toma algún valor, o conjunto de valores, como entrada y produce algún valor, o conjunto de valores, como salida. Por lo tanto, un algoritmo es una secuencia de pasos computacionales que transforman la entrada en la salida. También podemos ver un algoritmo como una herramienta para resolver un problema computacional bien especificado. El enunciado del problema especifica en términos generales la relación deseada de entrada / salida. El algoritmo describe un procedimiento computacional específico para lograr esa relación de entrada / salida.”*

Partiendo de esa definición, utilizaremos algoritos en este proyecto para realizar las funciones básicas de la gestión de datos, como también para ordenar los mismos.

Diagrama de clases:

Cronograma de trabajo:

|  |  |  |  |
| --- | --- | --- | --- |
| **N°** | **SEMANA** | **DURACIÓN** | **ACTIVIDAD** |
| 1 | 11 | 5 días | * Diagrama de clases * Definir el marco conceptual y los objetivos * Crear el repositorio en Github |
| 2 | 12 | 4 días | * Definir las estructuras de datos a usar * Definir el prototipo * Implementar las clases básicas |
| 3 | 13 | 4 días | * Realizar el diagrama de clases de segundo nivel * Realizar el diseño general del software * Implementar las funcionalidades del Mini-SGDB |
| 4 | 14 | 2 días | * Testear la solución y realizar los cambios pertinentes |
| 5 | 15 | 1 día | * Entrega del proyecto final y exposición |