

Apellido y Nombre: Villalba, Carla Camila.

Marco Teórico:

¿Por qué debemos usar Estructuras de Datos, que nos resuelve?

Una estructura de datos es una forma de organizar un conjunto de datos elementales, de manera eficiente, con el objetivo de facilitar su uso y manipulación. La forma en que se almacenan estos datos determina cómo se pueden recuperar. Utilizar estructuras de datos adecuadas permite realizar operaciones de manera eficiente, tanto en términos de tiempo como de espacio.

Por ejemplo, acceder a un elemento en un arreglo es muy rápido, mientras que insertar un elemento en una lista enlazada puede ser más eficiente que en un arreglo. Las estructuras de datos facilitan la organización lógica de los datos para que los algoritmos puedan manipularlos fácilmente. Además, ayudan a gestionar grandes volúmenes de datos, permitiendo que las aplicaciones escalen de manera efectiva.

Diferentes estructuras de datos están diseñadas para resolver distintos tipos de problemas. Por ejemplo, los grafos son ideales para representar redes y conexiones, mientras que las tablas hash son excelentes para búsquedas rápidas.

Las estructuras de datos son esenciales porque impactan directamente en la eficiencia y eficacia de los algoritmos, y son una parte fundamental del diseño y desarrollo de software.

¿Que es Un patrón Creacional, Estructural y Comportamiento?, dar ejemplos.

Los patrones de diseño son soluciones probadas y optimizadas a problemas comunes en el diseño de software. Se dividen en tres categorías principales: los patrones creacionales, que están diseñados para instanciar clases y objetos de manera eficiente; los patrones estructurales que se refieren a la estructura y composición de clases y objetos; y los patrones de comportamiento que se centran en la comunicación entre objetos.

Ejemplos:

*Patrones Creacionales: Singleton: Garantiza que una clase tenga solo una instancia y proporciona un punto de acceso global a ella. Ejemplo: Un gestor de configuración que se utiliza en toda la aplicación.

*Patrones Estructurales: Decorator: Añade responsabilidades adicionales a un objeto de manera dinámica. Ejemplo: Un sistema de notificaciones que puede enviar mensajes por correo electrónico, SMS, o ambos.

*Patrones de Comportamiento: Observer: Permite que un objeto notifique a otros objetos sobre cambios en su estado. Ejemplo: Un sistema de suscripción a eventos donde los suscriptores reciben actualizaciones automáticamente.

Que es un Pila y una Cola, dar ejemplos de APPS que las Implementen.

Son dos estructuras de datos. La pila sigue el principio FIFO,

"primero en entrar, primero en salir". En una cola, los elementos se agregan al final (cola) y se eliminan desde el frente (cabeza).

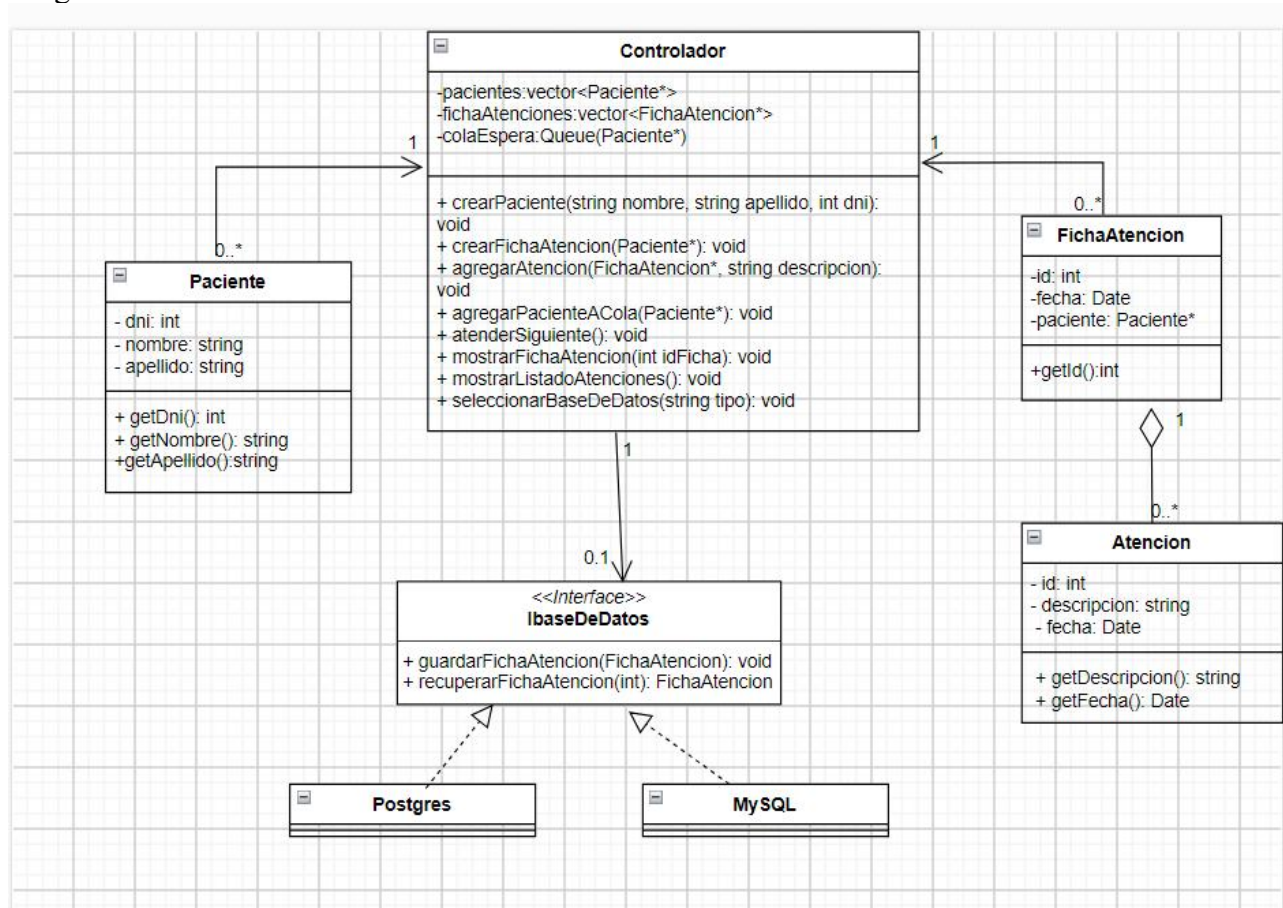
Un ejemplo de pila sería el historial de navegación que primero muestra el último sitio web que se visitó. y un ejemplo de cola sería la cola de personas en espera en un consultorio médico.

Cómo mediría el Rendimiento de un Algoritmo, que es BigO?

Medimos el rendimiento de un algoritmo según su complejidad algorítmica, es como medimos cuánto tiempo necesita un para hacer aquello para lo que fue pensado. Esto nos dice qué tan eficiente es. No se mide el tiempo en términos absolutos, sino en relación con el tamaño del problema. El tiempo se cuenta en pasos en pasos.

Big O, es una notación que sirve para explicar como se desarrollará la velocidad de un algoritmo en función del tamaño de un problema. es una notación para el peor caso, o el techo de crecimiento.

Diagrama de Clases:



Marco Practico

Un Centro de Salud Comunitario y Social de Músicos nos convoca para desarrollar una APP que Implemente la Atención a los Pacientes, Deberemos Realizar la Ficha de Atención, asignada al paciente y las Atenciones recibidas en el Centro de Salud, Deberemos almacenar en una Base de Datos (pudiendo seleccionar al momento de arrancar entre MySQL o Postgress) la ficha de Atención.

La misma debe permitir:

1. Crear la Ficha de Atención.
2. Atender los Servicios por Orden de Llegada de los Clientes
3. Cargar la Ficha de Atención.
4. Mostrar la Ficha de Atención.
5. Mostrar en una pantalla (consola) el listado de las Atenciones.



La aplicación deberá realizarse Implementando los Objetos Correspondientes.

Deberá mostrar por consola las Atenciones Realizadas

Se aconseja usar clases y un Array de Objetos e implementar la Estructura Correspondiente.

La persistencia deberá ser Implementada vía patrón de Software Correspondiente de forma Mock.

Nota: Tome nota de lo que se pide. Responda las consignas como se han establecido. No “haga” de más, tampoco de “Menos”,

Para aprobar será necesario contar con más del 60% de “cada uno” de los Marcos aprobados, representando 4 (Cuatro) en Nota (se distribuye ponderadamente).