**Capítulo 2: Diseño de la Solución**

**1. Descripción de estructuras de datos y operaciones**

El sistema implementa diversas estructuras de datos dinámicas para simular la administración de procesos dentro de un entorno de sistema operativo simplificado. La **lista enlazada simple** se utiliza en el módulo del Gestor de Procesos para almacenar dinámicamente cada proceso con sus atributos: ID, nombre y prioridad. Esta estructura permite una modificación constante y eficiente sin necesidad de un tamaño fijo, lo cual resulta ideal para manejar múltiples procesos con variabilidad en el tiempo de ejecución.

Por otro lado, se emplea una **pila**, también basada en lista enlazada, en el módulo del Gestor de Memoria. Esta estructura tipo LIFO facilita el manejo ordenado de la asignación y liberación de bloques de memoria, simulando cómo un sistema operativo gestiona recursos temporales.

En el Planificador de CPU se combinan dos tipos de colas. Una **cola fija**, implementada con arreglo estático, que gestiona procesos de forma secuencial en base a su orden de llegada. También se utiliza una **cola prioritaria**, representada mediante una lista enlazada ordenada por prioridad descendente. Esta permite que los procesos con mayor prioridad sean seleccionados primero para su ejecución, simulando un esquema realista de planificación.

Cada estructura posee funciones específicas como inserción, eliminación, búsqueda, actualización e impresión, las cuales permiten un control completo sobre el estado y contenido de los procesos.

**2. Algoritmos principales**

**Pseudocódigo para agregar proceso**

En el documento, se muestra una imagen que contiene el pseudocódigo para insertar un nuevo proceso en la estructura de lista enlazada. Este algoritmo sigue una lógica clara: se crea un nuevo nodo, se asignan los datos del proceso y se enlaza al inicio de la lista. El procedimiento garantiza que los procesos se agreguen de forma dinámica sin afectar los ya existentes.

**Pseudocódigo para cambiar estado del proceso**

También se presenta, en formato imagen, el pseudocódigo correspondiente a la modificación del estado de un proceso. Este algoritmo recorre la lista de procesos hasta localizar el nodo con el ID correspondiente, y actualiza su campo de estado. Es fundamental para simular la transición de estados como “listo”, “en ejecución” o “finalizado” dentro del entorno del planificador.

**3. Diagramas de Flujo**

El documento incluye diagramas de flujo que representan visualmente las operaciones más importantes del sistema, como la inserción de procesos, ejecución de procesos con prioridad, y asignación/liberación de memoria. Cada diagrama permite comprender la lógica de ejecución paso a paso, incluyendo condiciones de validación y flujo de decisiones. Aunque están en formato de imagen, se puede identificar claramente el uso de decisiones (“¿la lista está vacía?”, “¿ID encontrado?”) y acciones secuenciales (“crear nodo”, “enlazar”, “ejecutar proceso”).

**4. Justificación del diseño**

La elección de estructuras dinámicas responde a la necesidad de flexibilidad y eficiencia en el manejo de procesos dentro del sistema. La **lista enlazada** permite un crecimiento ilimitado de procesos activos, con operaciones de inserción y eliminación de bajo costo computacional. La **pila** ofrece un control estructurado del uso de memoria temporal, asegurando que los bloques más recientes se liberen primero, tal como ocurre en escenarios reales.

En cuanto a la **cola fija**, su uso es adecuado para representar flujos sencillos de ejecución. No obstante, la implementación de una **cola prioritaria** añade un nivel más realista y avanzado al simular políticas de planificación donde no todos los procesos tienen el mismo peso. Esta dualidad mejora el rendimiento y permite gestionar diferentes tipos de tareas de forma organizada.

El diseño modular y bien segmentado facilita no solo la comprensión del código sino también su mantenimiento. Cada estructura tiene funciones dedicadas que pueden ser reutilizadas o modificadas sin afectar el resto del sistema, lo que promueve una alta cohesión y bajo acoplamiento, cualidades deseables en ingeniería de software.