
Singleton

Kenneth Rodriguez

Resumen

Este proyecto busca ejemplificar el uso del patrón de diseño **Singleton** modelando una representación simple de un planificador de tareas de una computadora. Utilizando este patrón de diseño se busca mostrar sus ventajas y caso de uso.

Guía del Usuario

El programa busca representar la funcionalidad de un planificador de tareas del procesador de una computadora.

Al ser únicamente con fines demostrativos, el programa no cuenta con interacción con el usuario, de manera que todas las tareas a ejecutar ya se encuentran precargadas, y se esperaría el mismo comportamiento en cada ejecución.

El programa cuenta con 2 clases (explicadas con detalle más adelante): Planificador y Tarea. El Planificador es el Singleton de este programa, ya que solo existirá una instancia del mismo. El planificador se encargará de agendar las tareas que reciba, y posteriormente determinará el orden en que estas serán ejecutadas de acuerdo a su prioridad y orden de llegada.

Arquitectura del Sistema

El programa utiliza 2 clases: **Planificador** y **Tarea**.

La clase **Tarea** es una clase simple que representa un proceso en la computadora, como puede ser abrir un programa, conectar el equipo a la red, crear un archivo de texto, etc. Cuenta con un método ejecutar(), que representa el comportamiento a realizar de cada Tarea individual.

La clase **Planificador** es el singleton del programa, ya que solo existirá una instancia de esta clase en todo el programa. El singleton se encargará de agendar las tareas que reciba del programa y determinar el orden en que estas serán ejecutadas. Para esto, se

generó un enum de Prioridades alta, media y baja, y 3 colas de tareas correspondientes a cada prioridad.

Cuando el Planificador recibe una tarea, también recibe la prioridad de la misma, agregando así la tarea al final de la cola a la que pertenece.

Existen varias estrategias para determinar el orden en que se deben ejecutar las tareas pendientes, pero para este programa se utilizó un algoritmo sencillo. Para el proceso de ejecución, primero se ejecutan todas las tareas en la cola de prioridad alta hasta que no haya más tareas en la misma. Después se repite este proceso con la cola de prioridad media, y posteriormente con la cola de prioridad baja, hasta que no haya más tareas pendientes en el Planificador.

Conclusión

Este proyecto fue interesante y entretenido para mí, ya que no solo aprendí sobre patrones de diseño, sino que en mi investigación aprendí un poco más sobre el funcionamiento de un procesador.

Anteriormente ya había utilizado una implementación similar al Singleton, pero no de manera consciente. Aprender formalmente sobre este patrón de diseño fue algo que aprecié ya que me ayudó a aprender más sobre sus ventajas y casos de uso.