Proyecto: Cronómetro Pomodoro Personalizable

1. **Descripción del Contexto:** En la actualidad, la gestión eficiente del tiempo es un desafío constante, especialmente en actividades que requieren concentración sostenida. El método Pomodoro ha demostrado ser una estrategia efectiva para la productividad; sin embargo, las soluciones existentes, como aplicaciones y videos en línea, presentan limitaciones en términos de personalización.
2. **Definición del Problema o Necesidad:** Se ha identificado que muchas herramientas de Pomodoro no permiten una configuración flexible que se adapte a las necesidades individuales del usuario. En particular, no integran opciones de música personalizadas que favorezcan la concentración ni ofrecen intervalos de tiempo ajustables a las dinámicas específicas de cada persona. Como resultado, los usuarios deben alternar entre diversas aplicaciones y plataformas, generando interrupciones y reduciendo la eficiencia de sus sesiones de trabajo o estudio.
3. **Propuesta de Solución:** Se plantea el desarrollo de una plataforma web que permita configurar un cronómetro Pomodoro de manera completamente personalizable. Esta solución incluirá la posibilidad de ajustar los tiempos de trabajo y descanso según las preferencias del usuario, además de la integración con YouTube Music para seleccionar listas de reproducción acordes a cada sesión. Adicionalmente, se implementará un sistema de notificaciones sonoras al finalizar cada intervalo, acompañado de una funcionalidad que detenga o modifique la reproducción musical en función de la configuración establecida previamente.
4. **Justificación de la Solución:** La consolidación de estas funcionalidades en una única plataforma proporcionará una experiencia optimizada, reduciendo la necesidad de cambiar entre múltiples aplicaciones y minimizando las interrupciones. Esto permitirá a los usuarios mejorar su enfoque, maximizar su productividad y aprovechar al máximo cada sesión de trabajo, estudio o cualquier otra actividad que requiera concentración. En definitiva, esta solución representa un avance significativo en la eficiencia de la gestión del tiempo, adaptándose a las necesidades particulares de cada usuario.
5. **Modelado de Clases:**
   a. Clase Abstracta Temporizador: Es la estructura base del temporizador.

      Atributos:
      - duracion: int (Duración en minutos del temporizador).
      - estado: str (Estado actual del temporizador: "activo", "pausado", "detenido", "descanso").
      - tiempo_restante: int (Tiempo restante en el ciclo actual, en segundos, calculado en horas, minutos y segundos).

      Métodos Abstractos:

      - iniciar(): Inicia el temporizador.
      - pausar(): Pausa el temporizador.
      - reiniciar(): Reinicia el temporizador al estado inicial.

- actualizar_tiempo(): Disminuye progresivamente el tiempo restante, considerando el estado actual del temporizador.
- mostrar_tiempo(): Muestra el tiempo restante en horas, minutos y segundos.

b. Clase PomodoroTimer (Hereda de Temporizador): Implementa el temporizador Pomodoro.

Atributos:
- duracion_trabajo: int (Duración del intervalo de trabajo en minutos).
- duracion_descanso: int (Duración del intervalo de descanso en minutos).
- ciclos: int (Número de ciclos Pomodoro a ejecutar).

Métodos:

- iniciar(): Implementa la lógica para iniciar un temporizador Pomodoro.
- cambiar_estado(): Alterna entre los modos de trabajo y descanso.
- ejecutar_ciclo(): Controla el flujo de los ciclos de trabajo y descanso.

c. Clase Música: Gestiona la reproducción de música y la integración con YouTube Music.

Atributos:
- playlist_actual: str (URL o ID de la lista de reproducción activa).
- estado_reproduccion: str (Estado de reproducción: "reproduciendo", "pausado", "detenido").

Métodos:

- cargar_playlist(url: str): Carga una nueva lista de reproducción.
- reproducir(): Inicia la reproducción de la playlist.
- pausar(): Pausa la música actual.
- detener(): Detiene la reproducción.

d. Clase Notificacion: Maneja las alertas sonoras y visuales cuando un ciclo termina.

Atributos:
- sonido_alerta: str (Ruta o URL del sonido de alerta).
- mensaje: str (Mensaje a mostrar en la notificación emergente).

Métodos:

- enviar_notificacion(): Muestra una alerta visual en la interfaz.
- reproducir_sonido(): Ejecuta el sonido de alerta.

- mostrar_mensaje_por_estado(estado: str): Muestra un mensaje dependiendo del estado actual del temporizador.

e. Clase Usuario: Representa al usuario y sus preferencias dentro del sistema.

Atributos:
- nombre: str (Nombre del usuario).
- configuracion_timer: Temporizador (Configuración personalizada del cronómetro).
- playlist_favorita: str (Playlist predeterminada del usuario).

Métodos:

- guardar_preferencias(): Almacena las configuraciones del usuario.
- cargar_preferencias(): Recupera la configuración previa del usuario.

f. Clase Interfaz: Controla la interacción del usuario con la aplicación web.

Atributos:
- timer: Temporizador (Instancia del cronómetro que se visualiza y controla desde la UI).
- musica: Musica (Instancia del reproductor de música integrado).

Métodos:

- solicitar_configuracion_inicial(): Pregunta al usuario sobre la duración del temporizador y la selección de la playlist.
- mostrar_tiempo(): Muestra el tiempo restante en pantalla.
- actualizar_estado(): Cambia dinámicamente la interfaz según el estado del temporizador.
- recibir_input_usuario(): Captura la configuración ingresada por el usuario.

6. Relaciones entre Clases

- **Herencia:** PomodoroTimer hereda de Temporizador, implementando su comportamiento específico.
- **Asociación:** Usuario posee una referencia a Temporizador y Musica, lo que permite la personalización de la experiencia.
- **Composición:** Interfaz depende completamente de Temporizador y Musica, ya que sin ellos no puede funcionar.
- **Agregación:** PomodoroTimer utiliza Notificacion para alertar al usuario, pero estas pueden existir de manera independiente.