

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ciencias de la Ingeniería y Tecnología

Unidad Valle de las Palmas



Patrones de Software

Meta 4.4. Proyecto final: Gestor de citas aplicando patrones de Software

José Humberto Moreno Mejía

Noviembre 2024



Este sistema de gestión de citas es una herramienta que organiza y administra citas de manera eficiente. Está pensado para facilitar la vida de cualquier persona que necesite agendar, cancelar o eliminar citas. El sistema usa una ventana gráfica (hecha con **Tkinter**) donde puedes:

1. **Agregar citas automáticamente:**

- Se asignan a la fecha más cercana disponible y a una hora entre las 9:00 AM y las 5:00 PM. El sistema evita horarios ya ocupados.

2. **Eliminar citas:**

- Ingresa el ID de una cita y el sistema la elimina. Además, elimina automáticamente las citas que ya estén canceladas.

3. **Cancelar citas:**

- Cambia el estado de una cita a "Cancelada". Aunque no se elimina del sistema, esta cita ya no se puede modificar ni usar.

4. **Ver todas las citas:**

- Muestra una lista con el ID, tipo, detalles y estado de cada cita guardada.

Las citas se guardan en un archivo CSV, lo que asegura que no se pierdan incluso si cierras el programa.

### **¿Qué patrones de diseño usa el sistema?**

El sistema está construido usando **4 patrones de diseño**. Estos patrones ayudan a que el programa sea más organizado, fácil de entender y sencillo de mejorar en el futuro.

### **Fábrica (appointment\_factory.py)**

Este patrón se encarga de crear objetos Cita. Tú solo indicas el tipo de cita y los detalles, y el sistema la crea por ti. No necesitas preocuparte por cómo funciona el proceso por dentro. Hace que el código sea más ordenado y limpio. Además permite agregar nuevos tipos de citas sin afectar el funcionamiento actual.

### **Estado (appointment\_state.py)**

Este patrón permite que las citas cambien de comportamiento dependiendo de su estado. Permite manejar reglas específicas para cada estado. Es ideal para manejar citas que tienen estados diferentes.

### **Comando**

Cada acción principal (Agregar, Eliminar, Cancelar) se organiza como una tarea o comando. Esto ayuda a que el sistema sea más ordenado y fácil de entender.

Divide las acciones en partes claras y manejables. Se pueden agregar nuevas acciones en el futuro sin desordenar el código.

### **Observador**

Este patrón reacciona a eventos importantes y notifica al usuario. Por ejemplo, si eliminas o cancelas una cita, el sistema te avisa con un mensaje. Mejora la experiencia del usuario al mostrar mensajes claros sobre lo que está pasando en el sistema. Evita confusiones al informar al usuario sobre el éxito o fallo de sus acciones.