

Informe e1

Patrón de diseño utilizado: Estado (habitaciones)

Para este ejercicio elegí el patrón estado para representar los posibles estados en los que se encuentra una habitación y su respectiva conducta en cada estado.

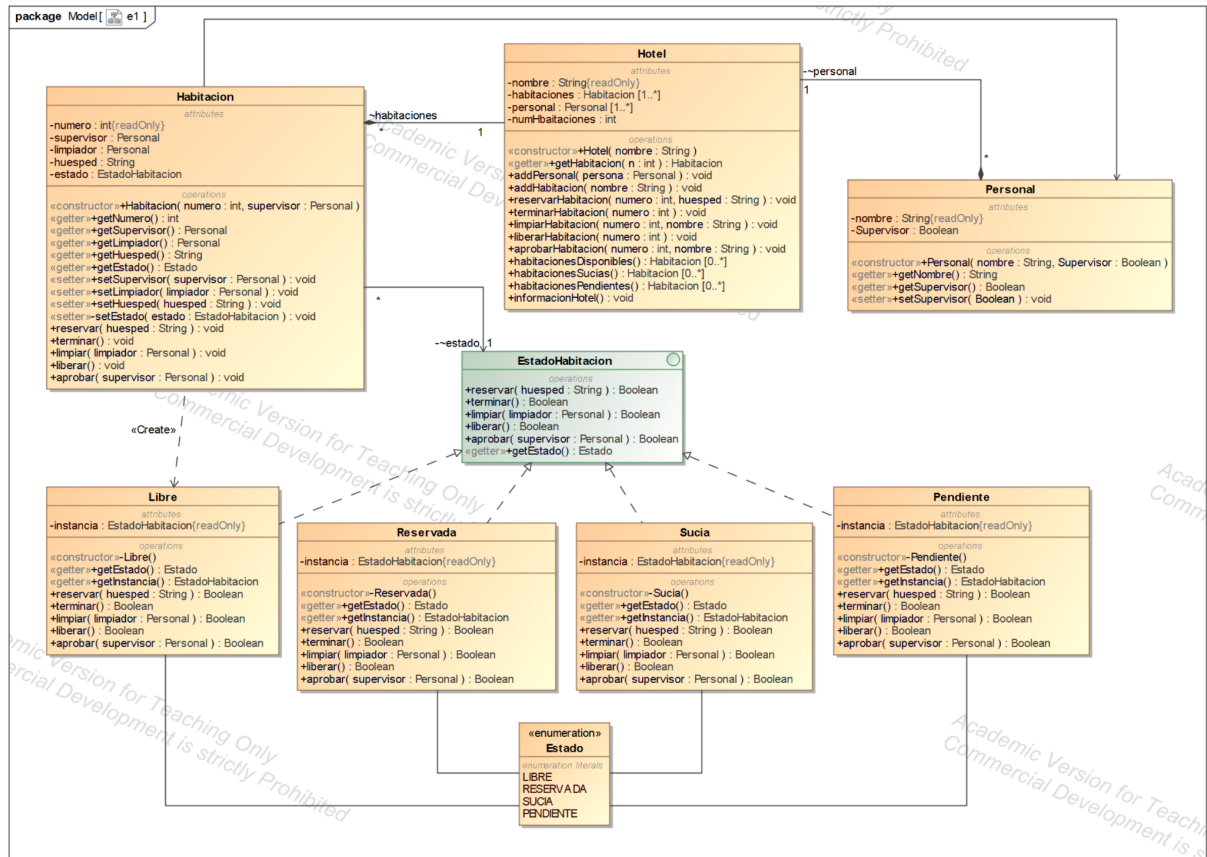
Una habitación puede pasar por diferentes estados (libre, ocupada, sucia y pendiente), pero solo podrá cambiar de estado en determinados casos y con acciones específicas, como pasar de reservada a libre mediante la acción de terminar habitación.

PRINCIPIOS SOLID:

Algunos ejemplos de los principios SOLID en este código incluyen:

- Principio de Responsabilidad Única (SRP): Cada clase tiene una única y bien definida responsabilidad. Por ejemplo, la clase reservar solo permite liberar o terminar la reserva de una habitación, y la clase sucia solo permite limpiar.
- Principio Abierto/Cerrado (OCP): El código está abierto para extensiones (agregando nuevos estados o funcionalidades) pero cerrado para modificaciones (el código existente no se modifica). Esto se logra utilizando una interfaz (EstadoHabitacion) y utilizando dicha interfaz en diferentes clases.
- Principio de Sustitución de Liskov (LSP): Los subtipos deben poder ser utilizados como sustitutos de su tipo principal. Esto se logra implementando la interfaz EstadoHabitacion en todas las clases.
- Principio de Segregación de Interfaces (ISP): Los clientes no deben ser forzados a depender de interfaces que no utilizan. Esto se logra teniendo diferentes métodos en la interfaz EstadoHabitacion que solo son implementados por las clases que los necesitan.
- Principio de Inversión de Dependencia (DIP): Los módulos de alto nivel no deben depender de módulos de bajo nivel. Ambos deben depender de abstracciones. Esto se logra utilizando la interfaz EstadoHabitacion y utilizandola en la clase Habitacion.

DIAGRAMA DE CLASES



DIAGRAMAS DINÁMICOS

