



Universidad
Nacional
de Quilmes

Programación Objetos II

Trabajo Practico Final

1° cuatrimestre 2021

Equipo: TriKappa

- Nicolas Cundari → nicolasmcundari@gmail.com
- Gabriel Maugeri → g.r.maugeri@gmail.com
- Damian Peñalver → jdpenalver@hotmail.com

Profesor asignado: Diego Cano

Patrones de diseño utilizados

- Composite Pattern (adaptado)
- State Pattern
- Singleton Pattern
- Observer Pattern
- Template Method Pattern

Decisiones de diseño

Definimos el Sitio como una clase de instancia única, creemos que no debe ser posible que se creen más de una instancia de Sitio. El sitio conoce a los Usuarios y a través de estos a las Publicaciones. El Sitio también conoce al ObserverManager con quien interactúa para notificar a las aplicaciones externas que se suscriben a los diferentes eventos. El Sitio conoce también las clases que solo poseen descripciones (por Ej. TipoDeInmueble, MedioDePago, Servicio, etc.). El Sitio se encarga también de procesar pedidos de información como ser: topTenInquilinos, inmueblesLibres y tasaDeOcupacion.

Para los criterios de búsqueda creímos conveniente utilizar una adaptación de composite ya que tenemos un criterio de búsqueda “básico” obligatorio y además otros dos que son opcionales y se componen con este.

Sobre los usuarios creemos que no hay diferencias entre usuarios inquilinos y usuarios propietarios ya que indistintamente uno puede tomar el rol de otro y también puede ser los dos simultáneamente si es que publica una propiedad y reserva otra a la vez, por este motivo se modela en una única clase Usuario.

El Usuario posee dos listas, una con sus reservas y una con sus publicaciones.

Las Reservas tienen la característica de poder estar en diferentes estados en diferentes momentos y comportarse según su estado actual ante las acciones de aceptar, rechazar y cancelar. Por este motivo incorporamos el state pattern en esta clase y sus estados. Los estados a su vez implementan singleton pattern.

Tanto el Usuario como el Inmueble implementan la interfaz Calificable porque son los objetos del dominio sobre los cuales se puede hacer una calificación.

Una publicación puede tener distintos precios dependiendo de la temporada o si es fin de semana o no. Estas variaciones las implementamos con una lista de objetos PrecioDiaOcupacion que representa para cada día de la publicación su precio y además indica si ese día está disponible u ocupado.

La Publicación también posee un Inmueble y este la política de cancelación que aplica. Las distintas políticas de cancelación están modeladas como una jerarquía de clases e implementan `template method` para aplicar la cancelación.

Para finalizar el mecanismo de notificación a los sistemas y aplicaciones externas se modela con un `observer`. Los interesados se pueden suscribir a cancelaciones, bajas de precio y reservas. Para esto definimos una clase de instancia única (singleton) llamada `ObserverManager` que se encarga de notificar a los observadores suscriptos. La interacción se realiza entre la reserva cancelada y el Sitio y la publicación que bajo de precio y el Sitio, por último, el Sitio se encarga de interactuar con el `ObserverManager`.