

# TP Final Programación Orientada a Objetos 2

## 1° cuatrimestre 2021

### Integrantes:

- Coria Lautaro. [lautacoria97@gmail.com](mailto:lautacoria97@gmail.com)
- Scaglioni Camila. [camila.pia.scaglioni@gmail.com](mailto:camila.pia.scaglioni@gmail.com)
- Segovia Ariel. [ariel-110595@hotmail.com](mailto:ariel-110595@hotmail.com)

### Introduccion

Decidimos organizar las tareas de manera que cada uno abordara una tematica en particular en lugar de cada integrante diseñar clases definidas. Al comienzo no fue sencillo pero a medida que cada uno hacia su aporte se iban abriendo las puertas a diferentes posibles soluciones. El primer gran conflicto fue aprender a trabajar con github, donde tuvimos problemas al resolver conflictos en el codigo, con el classpath (los tests y paquetes no se comportaban como tal) y hasta ultimo momento con la codificacion de caracteres, teniendo que alterar algunos nombres de variables para evitar el uso de “ñ”.

### Publicacion y Alquiler

Quizas la tematica mas simple en un comienzo hasta el momento de idear una estrategia para los diferentes precios que pueden existir para un inmueble dependiendo de las fechas elegidas por el propietario de dicho inmueble. Se delego la responsabilidad a una clase llamada PeriodoPrecio para que se encargue de guardar una determinada cantidad de fechas con su respectivo precio, el inmueble en si cuenta con un atributo precioPorDefecto, el cual se retorna si el dia solicitado no existe entre los periodos creados por el propietario.

### Busqueda de Inmuebles

Mas alla del propio buscador, se creo una clase busqueda donde el usuario puede ingresar los parametros que necesita y crea convenientes para encontrar los inmuebles que mas le convengan. El buscador opera sobre esta clase, realizando los filtros pertinentes sobre los inmuebles publicados en el sitio, uno a la vez.

### Ranking de Inmuebles y Propietarios

Se desarrollo un sistema de perfiles donde tanto usuarios (inquilinos y propietarios) como inmuebles pudiera recibir y guardar puntos, y en el caso de los usuarios, brindar puntos. La clase encargada de trabajar con las puntuaciones es la clase abstracta Perfil, dado que las clases hijas van a tener el mismo comportamiento en cuanto a los puntajes, variando unicamente en las categorias que cada uno contenga. Cabe destacar que para evitar codigo repetido, se creo la interfaz PuntuablePorEstadia que engloba el comportamiento de recibir puntaje para propietarios e inmuebles.

## Visualizacion y Reserva

Aquí es donde se dividen las responsabilidades de cada clase concreta que extiende a perfil para dar lugar a la visualización. En el caso del perfil del inmueble se desarrollo un composite donde un perfil contiene a otro (perfil de inmueble contiene a perfil de dueño del inmueble), de esta manera es posible llegar tanto a los comentarios sobre el propietario como a los del inmueble.

## Concrecion de una Reserva

En un principio se establecio el uso de un state para las reservas, dando lugar a los estados PendienteDeConfirmacion, Confirmado y Cancelado, pero esta decision se torno excesiva y se tomo la decision de que tanto inquilinos como propietarios guardaran sus reservas confirmadas en una colección específica y sus reservas pendientes de confirmacion en el caso de propietarios

## Administrador de Reservas para Inquilinos

Es una clase simple que solo realiza funciones sobre las reservas a cargo del inquilino o cancela las reservas. En este ultimo caso solo ejecutaria el disparador para dar inicio a dicha cancelacion de reserva.

## Administrador del sitio

Se creo una clase llamada Administrador Del Sitio que servirá para saber todos los inmuebles libres, la tasa de inmuebles, para dar de alta las categorías para el sitio (Propietario Inquilino e Inmueble) que luego serán calificadas, los servicios que serán seleccionados para publicar un inmueble y los tipos de inmueble que se utilizan en el sistema.

Para saber los 10 inquilinos que más alquilaron, se creó un método que implementa otro método que ordena la lista de todos los usuarios que alquilaron. El orden de la lista se hizo creando otra clase (CompararUsuarios) que implementa la interfaz Comparator. En esta clase se comparan las veces que alquilaron los usuarios, y así, con el metodo collections.sort podemos ordenar de mayor a menor, los inquilinos que mas veces alquilaron.

## Políticas de Cancelacion

Desarrollado a partir de un strategy donde cada reserva cuenta con una política de cancelacion designada. Por otro lado, se puede destacar que en la política de cancelacion intermedia se pueden ejecutar diferentes acciones dependiendo de la fecha actual y la fecha del primer día de la reserva. Podemos ver aquí, el uso de un strategy dentro de otro para evitar el uso de estructuras de control de manera anidada.

## Notificaciones

Se crearon 2 clases; Sitio Web (para sitios web como trivago, al mundo, etc) y AppUser (Aplicacion con la cual los usuarios se suscriben).

El sitio web, por el momento, esta interesado en las rebajas de precio de los inmuebles en los cuales esta interesado, es por eso que cada vez que el propietario actualiza el precio, y este es mas bajo que el anterior precio, se dispara un mensaje alertando que X inmueble esta a N precio.

Para la aplicación de celular, esta interesada en las cancelaciones de reservas de ciertos inmuebles. Cada vez que se libera un inmueble al cual este estaba interesado, se le envía un mensaje alertando que X inmueble se liberó.

Estas dos clases implementan una interfaz llamada Inotify, que a la vez, esta implementa otras 2 interfaces llamadas HomePagePublisher, que tiene el metodo Publish para alertar la baja de precios en inmuebles interesados, y PopUpWindow, que tiene el metodo popUp para alertar un inmueble cancelado.

## Reserva Condicional

Asi como la implementacion del sistema de puntajes dentro de perfil, se desarrollo la reserva condicional en base al uso de maps, donde cada llave es una reserva confirmada y el valor son las posibles reservas encoladas que puedan llegar a confirmarse si la reserva llave llegara a cancelarse.