

Programación 4

Informe del Modelo de Diseño

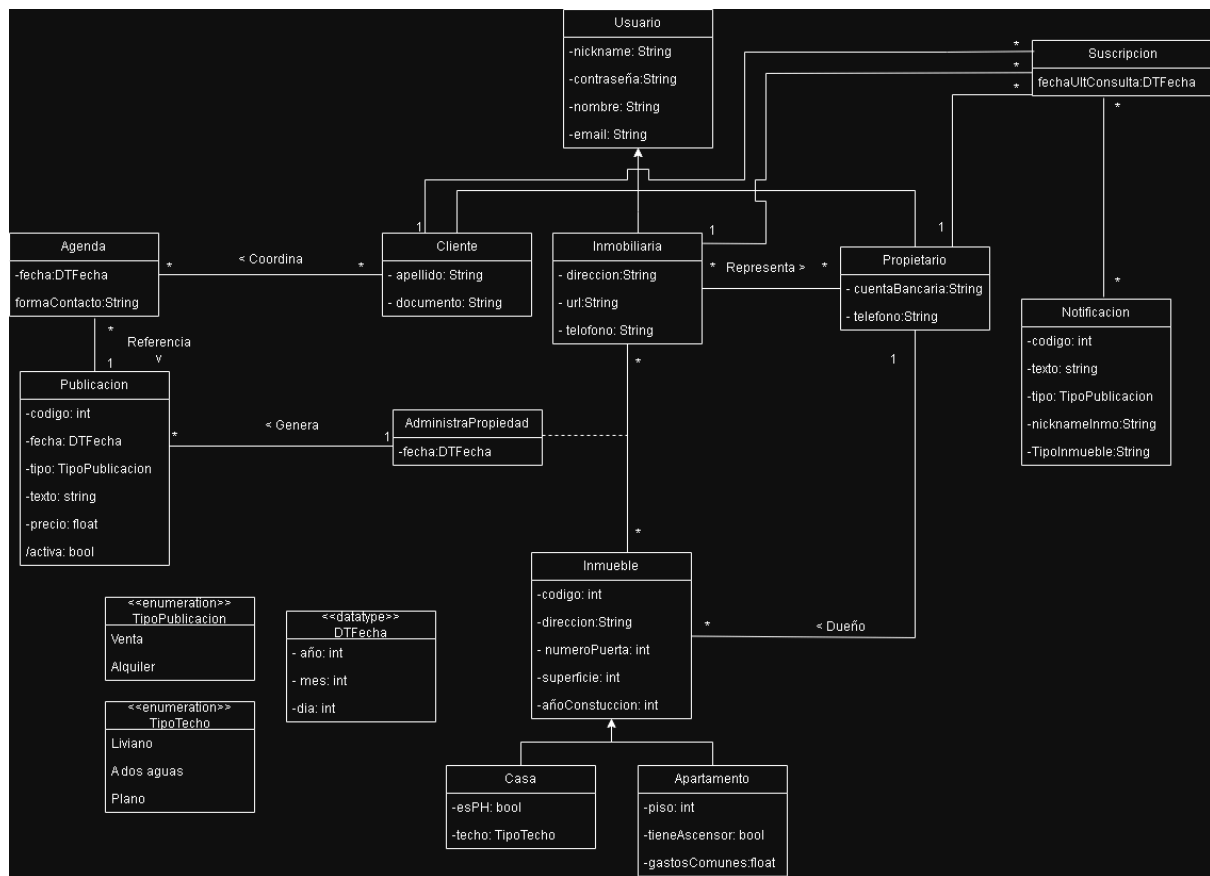
Grupo 66

Integrantes

Valentin Pedragosa	CI: 5.532.445-1
Matias Torres	CI: 5.485.366-5
Tomas Martinez	CI: 5.535.431-9
Camilo Morganti	CI: 5.443.356-4
Emiliano Esponda	CI: 5.491.971-6

Docente: Federico Andrade

1 Realización de Casos de Uso

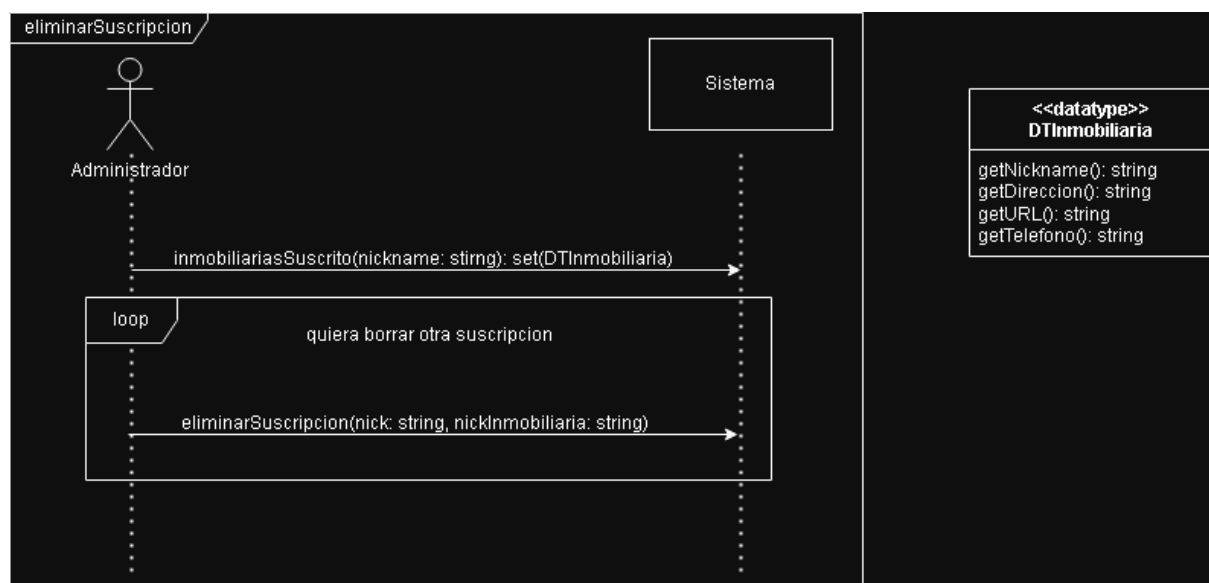


Una alternativa si los casos de uso fueran dados en la etapa de análisis hubiera sido agregar las clases suscripción y notificación como se ve en la imagen, pero al ser requerimientos introducidos en etapa de diseño, lo resolvimos en el DCD utilizando el patrón observer para evitar volver a la etapa de análisis.

La modificación que se debe hacer en el diagrama de secuencia de sistema del caso de uso Alta de Publicación, es agregar una nueva operación que se encargue de notificar a todos los usuarios suscritos a la inmobiliaria que está haciendo el alta de la publicación

Así quedarían los nuevos diagrama de secuencia del sistema:

Alta Publicacion:



Consulta de Notificaciones

Administrador

Sistema

notificacionesRecibidas(nickname:String):set(DTNotificacion)
EliminarNotificaciones(nickname:String)

<<datatype>>
DTNotificacion

Nickname:String
Codigo:int
Texto:String
TipoPub:TipoPublicacion
Tipo: TipoInmueble

Suscribirse a notificaciones

Administrador

Sistema

listarInmobiliariasnosuscrito(nickname:String):set(DTInmobiliaria)

Loop

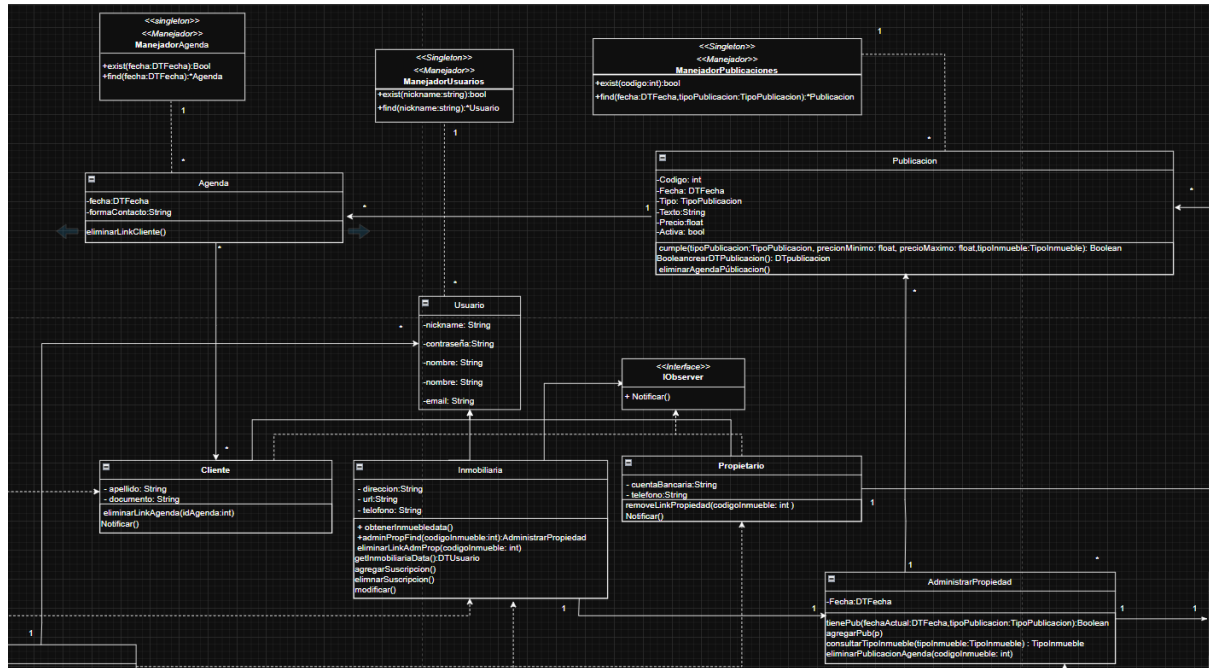
quiera suscribirse a otra inmobiliaria

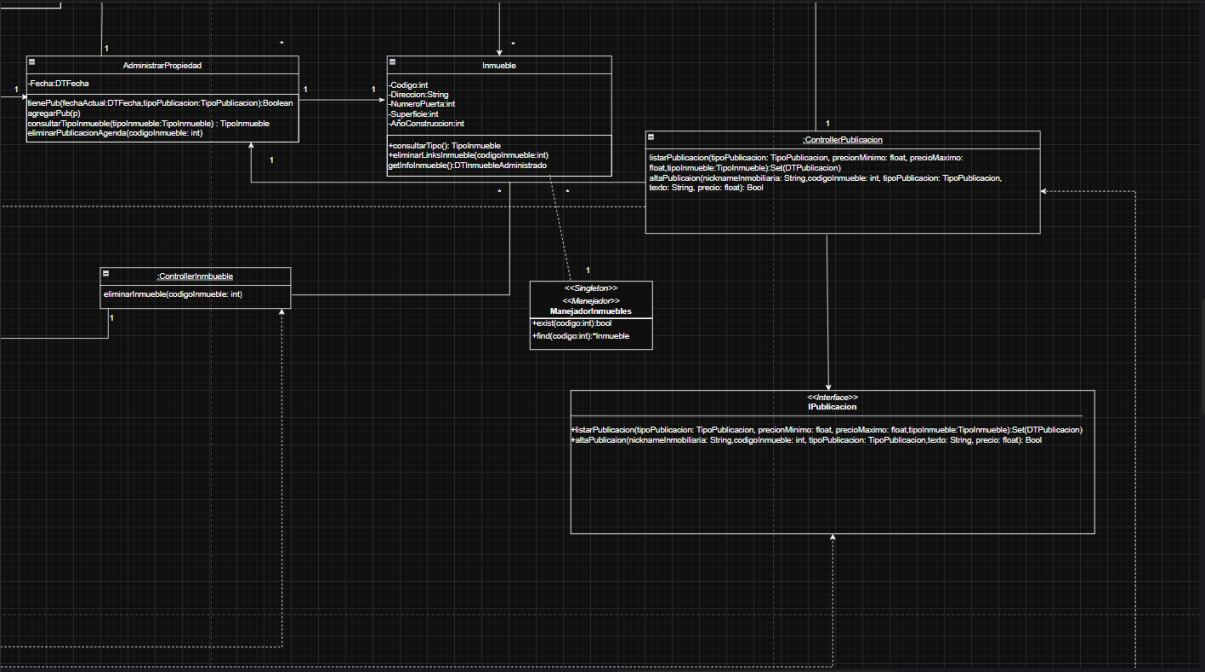
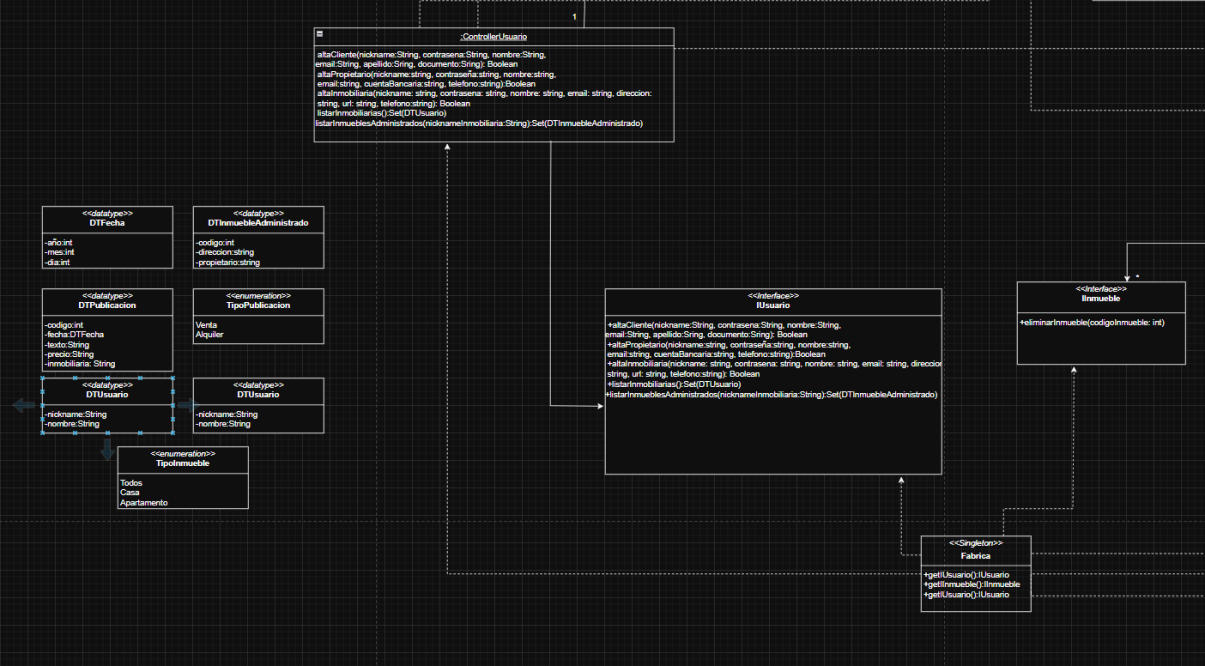
addSuscriptor(nickname:String,nicknameInmobiliaria:String)

<<datatype>>
DTInmobiliaria

getdireccion():sting
geturl():String
getTelefono():String

1.1 Estructura

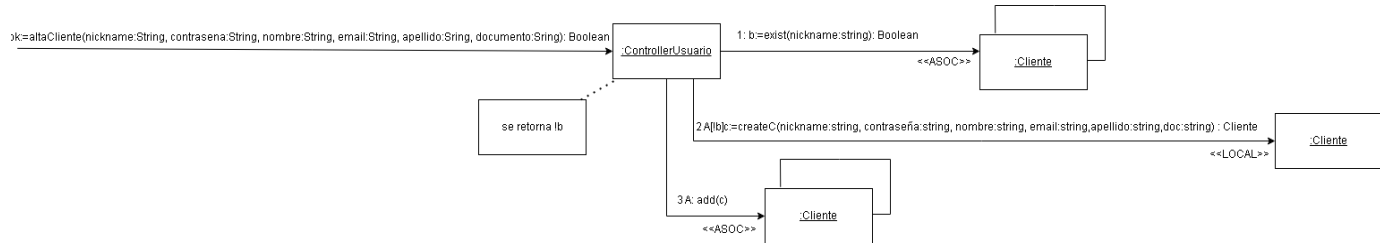




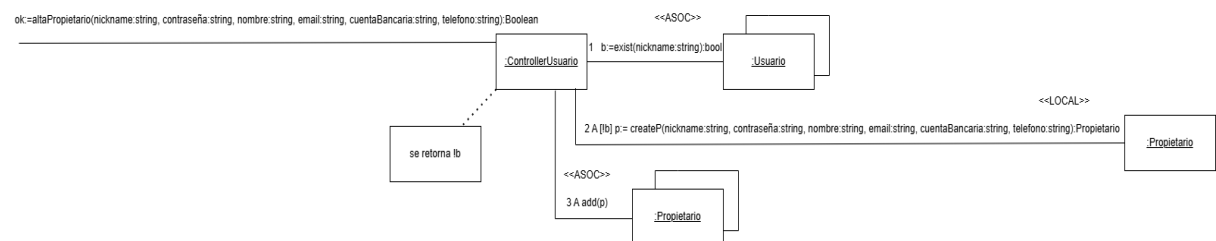
[https://app.diagrams.net/#G1m-FLgh3pNn_s7mISyPJzUHnPMIzmSMi9#%7B"pageId"%3A%7B"mvGABBS8F9LN_6VwxVgy"%7D](https://app.diagrams.net/#G1m-FLgh3pNn_s7mISyPJzUHnPMIzmSMi9#%7B) agregamos el link a el DCD por si no son lo suficientemente claras las distintas capturas.

1.1.1 Interacciones

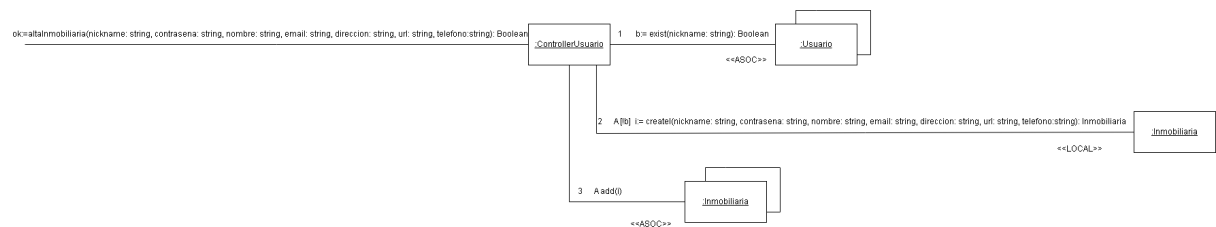
1.1.1.1 Diagramas de comunicación de la Alta Cliente



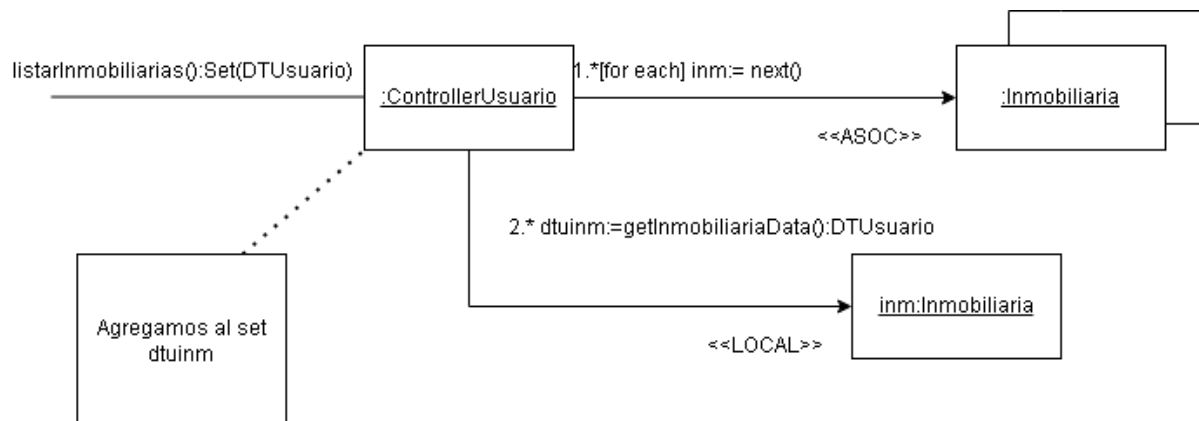
1.1.1.2 Diagramas de comunicación de la Alta Propietario



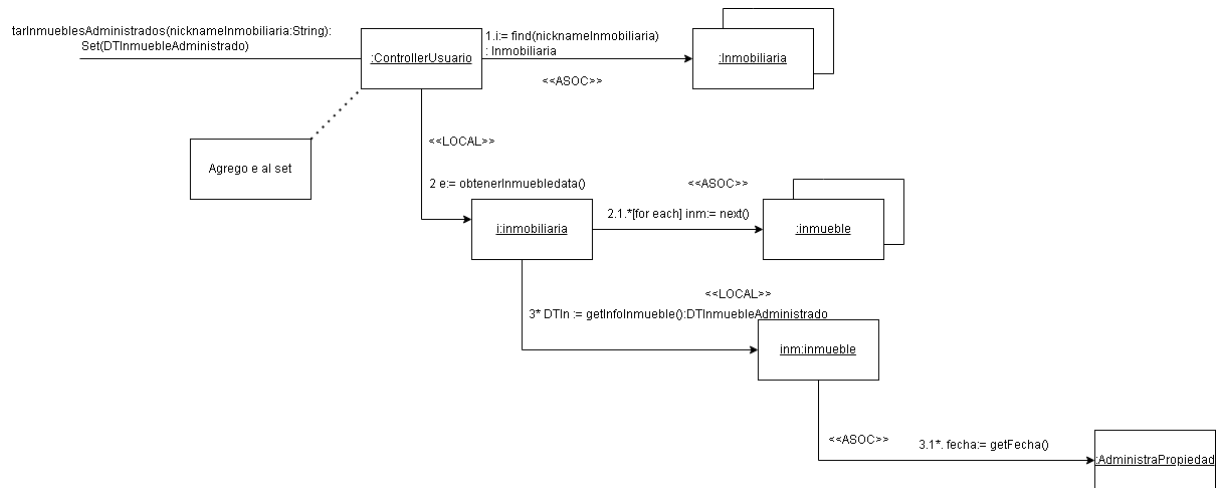
1.1.1.3 Diagramas de comunicación de Alta Inmobiliaria



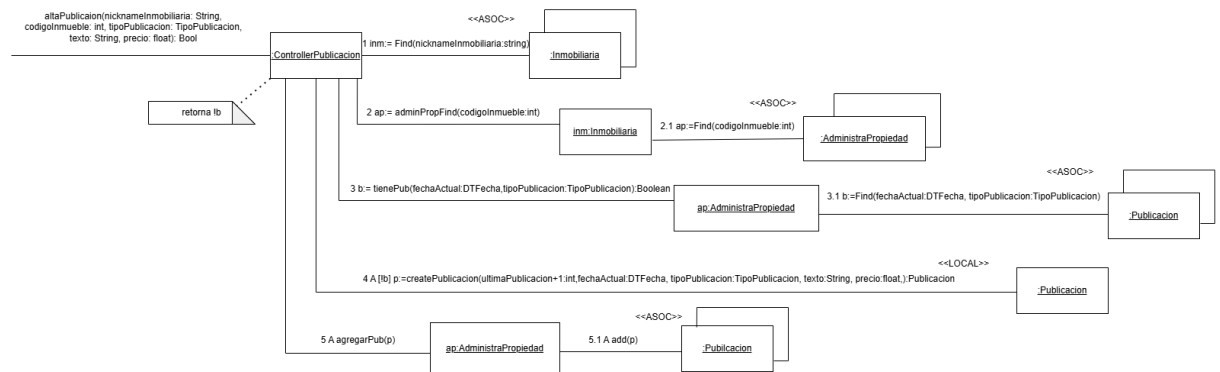
1.1.1.4 Diagramas de comunicación de la Lista Inmobiliarias



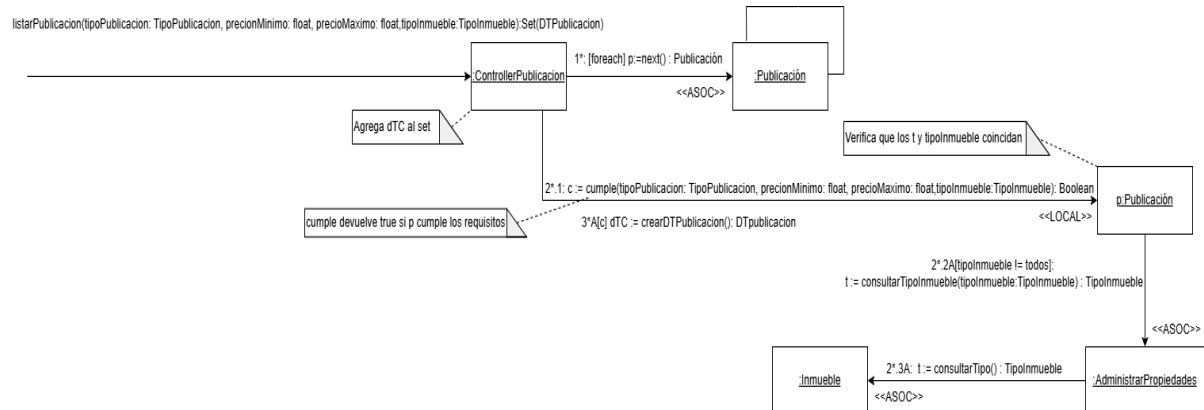
1.1.1.5 Diagramas de comunicación de la Listar Inmuebles Administrados



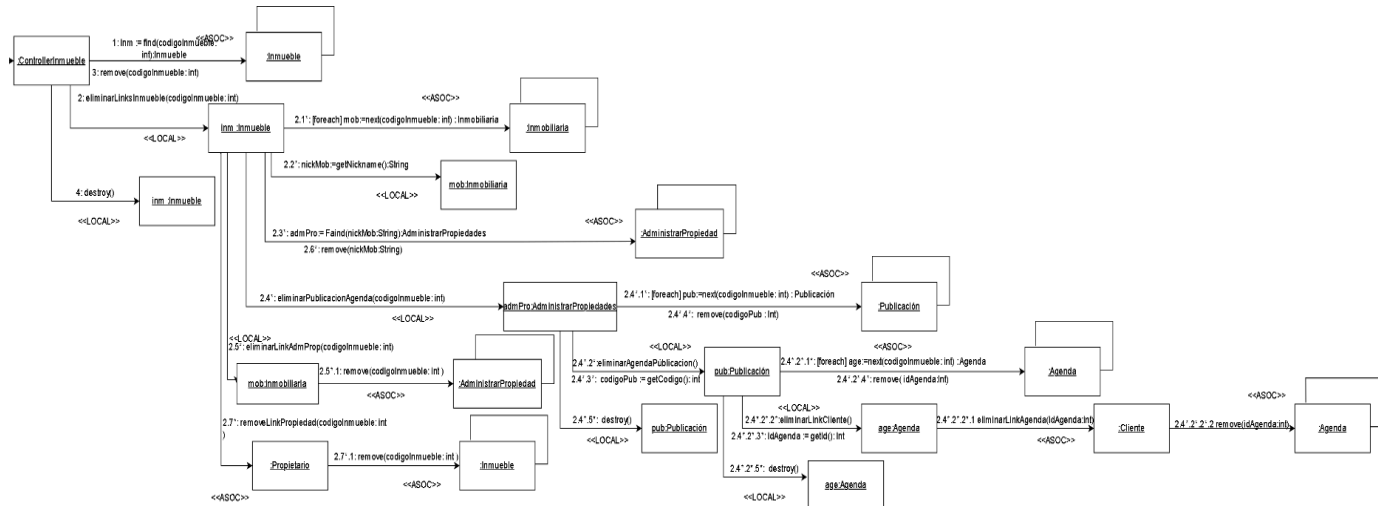
1.1.1.6 Diagramas de comunicación de la Alta Publicacion



1.1.1.7 Diagramas de comunicación de la Listar Publicacion



1.1.1.8 Diagramas de comunicación de la Eliminar Inmueble



Se puede visualizar con más detalle los diagramas de comunicación en el siguiente link:
https://drive.google.com/drive/folders/14BPbLKBHLgw_BScS-8hpJgJ6HJFP1Ti8?usp=sharing

1.1.2 Patrones de diseño

1.1.2.1 Observer

Como los clientes y propietarios del sistema pueden suscribirse a las inmobiliarias que les resulten más interesantes, permitiéndoles recibir notificaciones cada vez se da de alta una nueva publicación para un inmueble que administra la inmobiliaria, se decidió utilizar el patrón Observer que prevé este tipo de situaciones.

Las clases involucradas y sus roles son:

- Inmobiliaria, que cumple el rol de Subject.
- IObserver, que cumple el rol de Observer.
- Cliente y Propietario, que cumplen el rol de Observadores concretos.

1.1.2.2 Factory

Dado que se decidió separar el sistema en distintos controladores especializados en el manejo de las operaciones relacionadas entre sí (esta decisión se justifica en la sección de los criterios grasp), para reducir el acoplamiento entre el consumidor de los controladores y el sistema, se optó por utilizar el patrón de diseño Factory que permite la visibilidad desde el consumidor de los controladores sin acoplarse directamente.

Las clases involucradas y sus roles son:

- Fabrica, que cumple el rol de fábrica.
- IUserario, que cumple el rol de interfaz de ControllerUsuario.
- Inmueble, que cumple el rol de interfaz de ControllerInmueble.
- IPublicacion, que cumple el rol de interfaz de ControllerPublicacion.
- ControllerUsuario, ControllerInmueble, ControllerPublicacion, cumplen el rol de controladores.
- El administrador del sistema sería quien ocuparía el rol de consumidor.

1.1.3.3 Singleton

En numerosas partes del diseño nos enfrentamos a la necesidad de limitar a una las instancia de una clase y que se pueda acceder a ella desde cualquier parte del sistema, en estos casos se resolvió utilizando el patrón Singleton, y se indica en el DCD con “<<Singleton>>” las clases que serán implementadas siguiendo este patrón, y como solo hay un único rol en este patron que lo cumple la misma clase a continuacion se detallan las clases que serán implementadas siguiendo el patrón:

- ManejadorPublicaciones
- ManejadorUsuarios
- ManejadorAgenda
- IObserver
- Fabrica

1.1.3 Criterios GRASP

Se utilizaron 3 controladores ControllerUsuario, ControllerPublicacion y ControllerInmueble lo cual nos ayudó a asignar las responsabilidades de manejar las operaciones del sistema. Se minimizó el acoplamiento entre clases, logrando dependencias mínimas, evitando también la comunicación entre controladores. Las responsabilidades se asignaron a las clases que tienen la información necesaria para cumplirlas. Por ejemplo, la clase Inmobiliaria fue la encargada de devolver DTUsuario con su propio nickname y nombre. Se buscó que cada clase tuviera una única responsabilidad bien definida y relacionada, logrando así una alta cohesión.

2 Criterios Generales

Algunas aclaraciones, si bien en las clases con patrón de singleton tiene el atributo de la instancia y su getter estos no están escritos en el dcd, porque se está dado por hecho que se implementan, como también algunas operaciones de los manejadores que suelen ser típicas a la hora de manejar colecciones como: find(), add(), create(), destroy(), remove(), exists(), member(), next(). Por último en los diagramas de comunicación cuando se accede a la colección desde una instancia de una clase con una relación de tipo 1 a *, se da por hecho que se está accediendo a la colección propia que contiene instancias con las que esta relacionada.