Programación 4

Informe del Modelo de Diseño

Grupo 1
Integrantes

Alexis Gutiérrez	**CI:** 5.536.385-7
Gregory Fernández	**CI:** 5.610.839-5
Christian Donaley	**CI:** 5.165.275-5
Lucas Fierro	**CI:** 5.457.786-5
Valentino Báez	**CI:** 5.502.219-0

Docente: Federico Andrade

1 Análisis de los impactos de los nuevos requerimientos

1.1 Impacto en el modelo de dominio

La agregación de dichos requerimientos afectó muy poco el modelo de dominio antes establecido, se agregaron dos asociaciones:

1- Se genera una nueva relación *Suscrito* entre Cliente e Inmobiliaria el cual representa la suscripción de una instancia Cliente a una instancia Inmobiliaria.
2- Se genera una nueva relación *Suscrito* entre Propietario e Inmobiliaria el cual representa la suscripción de una instancia Propietario a una instancia Inmobiliaria.

Se agrega además un DataType:

- DTNotificacion
  Atributos:
    1- nicknameInmobiliaria: String
    2- código: int
    3- texto: String
    4- tipoPub: TipoPublicacion
    5- tipoInmueble: TipoInmueble

Y tres atributos en total:

1- Se agrega un nuevo atributo al concepto Cliente, *notificaciones,* que representa una lista con los DTNotificacion de las instancias de Publicacion generadas por las instancias de las inmobiliarias a las que el cliente está suscrito.
2- Se agrega un nuevo atributo al concepto Propietario, *notificaciones*, que representa una lista con los DTNotificacion de las instancias de Publicacion generadas por las instancias de Inmobiliaria a las que el Propietario está suscrito.
3- Se agrega un nuevo atributo al concepto Inmobiliaria, *suscriptores*, que representa una lista con los nickname de cada instancia de Usuario que está suscrita a la instancia de Inmobiliaria en cuestión.

Estas nuevas asociaciones, los nuevos atributos y conceptos no causaron repercusiones significativas en el desarrollo hasta el momento del sistema, no hizo falta volver a diseñar el modelo de dominio y solo se realizaron pequeños cambios y agregaciones.
Los casos de uso nuevos quedan satisfechos ya que las asociaciones realizadas permiten realizar las suscripciones y notificaciones entre las clases correspondientes.
Luego los atributos permiten almacenar la información generada a partir de dichas relaciones.
Por último DTNotificacion permite mostrar la información contenida en cada notificación que reciba tanto los propietarios como los clientes.

En conclusión en cuanto al modelo de dominio, las agregaciones no causaron problemas significativos, y se pudo satisfacer los nuevos requerimientos correctamente.

1.2 Impacto en los DSS

No hubo cambios en dichos diagramas pero sí en el contrato de altaPublicacion, donde se agregó una postcondición:

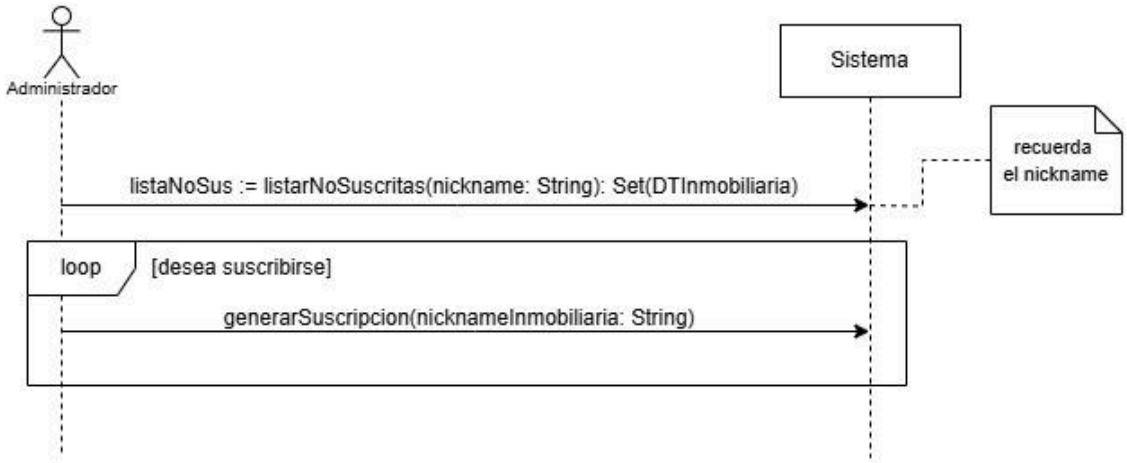
- Se realiza la notificación correspondiente a todos los usuarios que se encuentren suscritos a la inmobiliaria cuyo *nickname* coincide con el valor *nicknameInmobiliaria* añadiendo a cada uno de ellos el código *código* al set *notificaciones*, agregando la notificación a la lista de notificaciones de los clientes y propietarios correspondientes.

Esta postcondición satisface los nuevos requerimientos ya que al realizarse las notificaciones correspondientes se entrega dicha información a los propietarios y clientes para que puedan consultarla cuando lo necesiten.

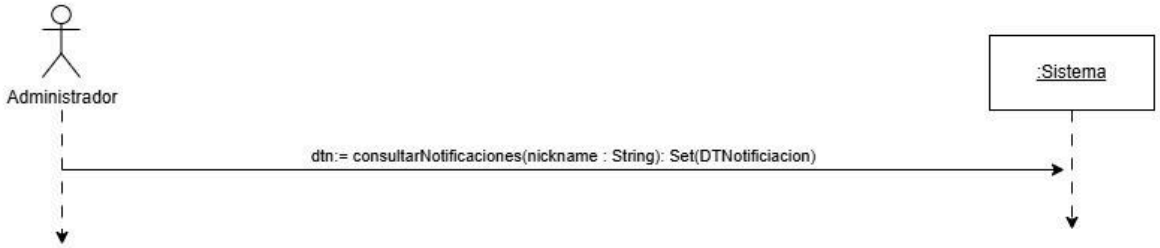
Si bien esto no modifica los DSS, sí conlleva que el diagrama de comunicación donde se representan las interacciones que realizan altaPublicacion tenga algunas interacciones más de las que hubieran si no se realizaran tales notificaciones.

1.3 DSS nuevos

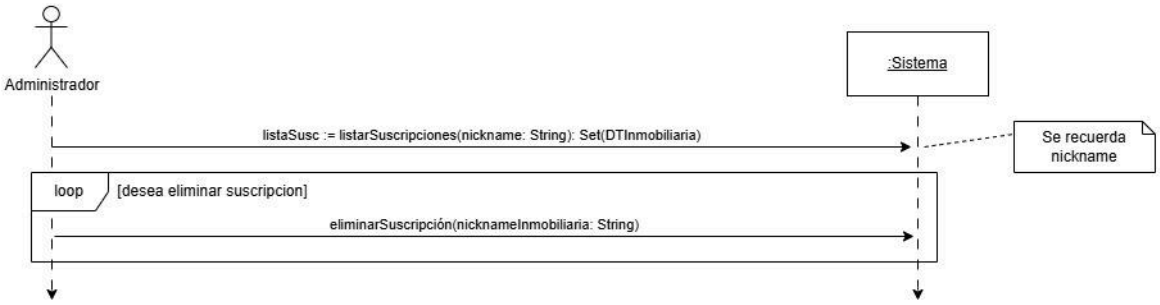
1.3.1 Caso de uso Suscribirse notificaciones



1.3.2 Caso de uso Consultar notificaciones



1.3.3 Caso de uso Eliminar Suscripción



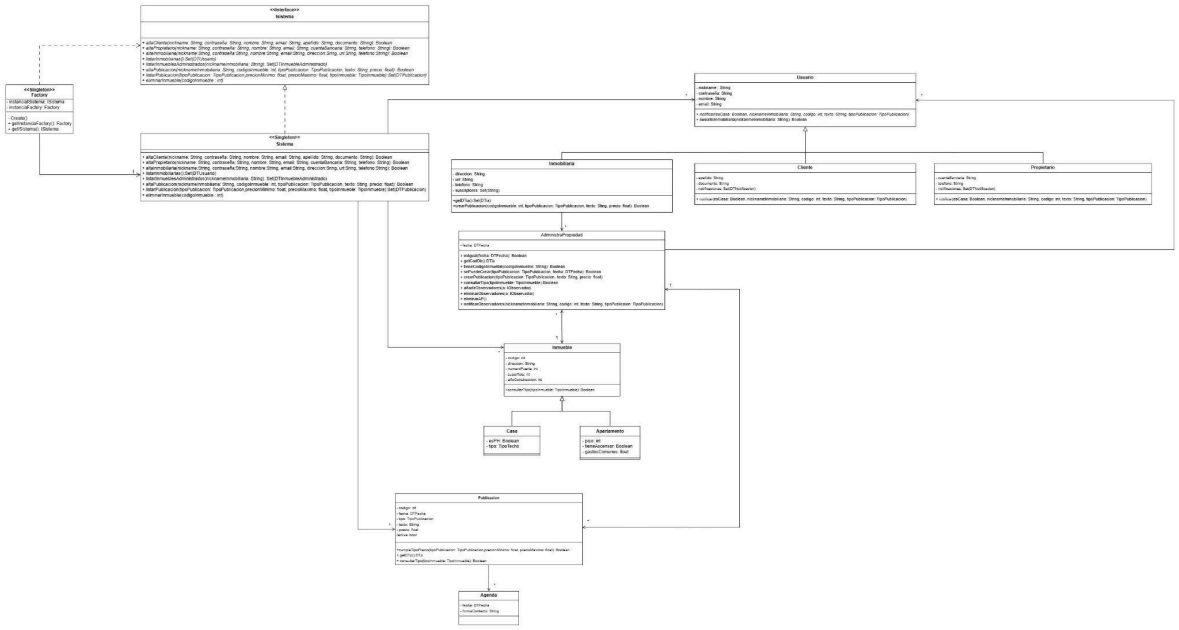
2 Realización de Casos de Uso

2.1 Alta de Usuario

Las operaciones realizadas son:

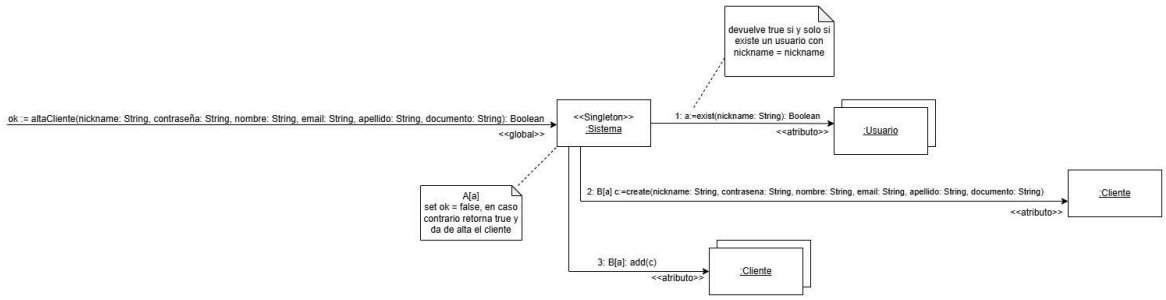
- altaCliente
- altaPropietario
- altaInmobiliaria

2.1.1 Estructura



2.1.2 Interacciones

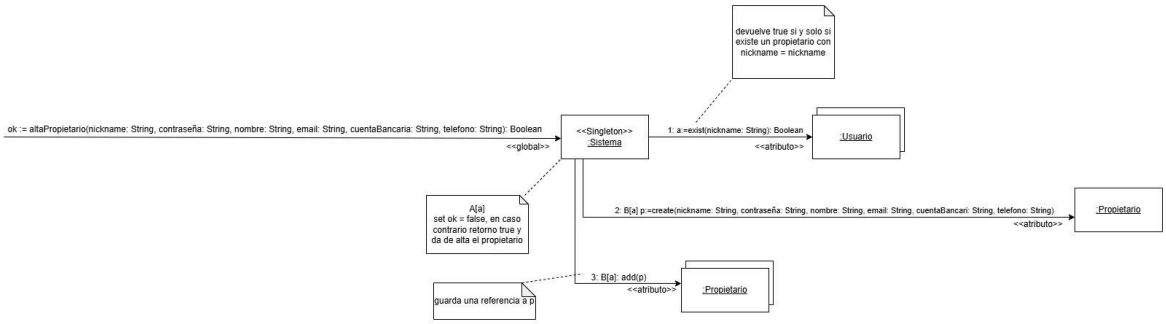
2.1.2.1 Diagrama de comunicación de altaCliente



Asignación de responsabilidades:

- Controlador de la operación: Sistema
- Creador de instancias de Cliente: Cliente

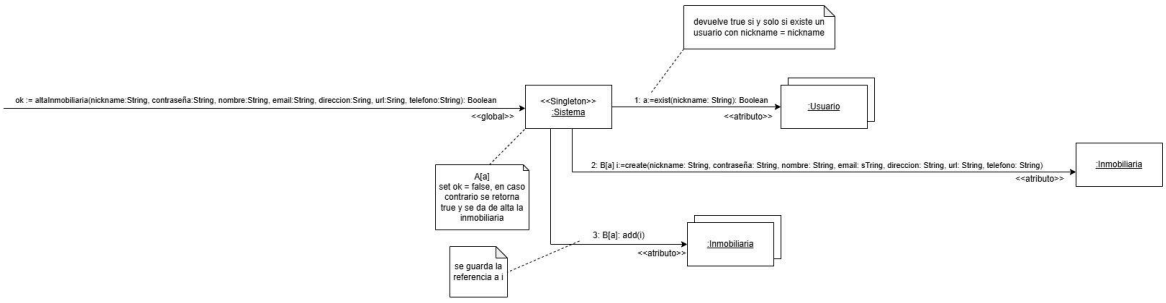
2.1.2.2 Diagrama de comunicación de altaPropietario



Asignación de responsabilidades:

- Controlador de la operación: Sistema
- Creador de instancias de Propietario: Propietario

2.1.2.3 Diagrama de comunicación de altaInmobiliaria



Asignación de responsabilidades:

- Controlador de la operación: Sistema
- Creador de instancias de Inmueble: Inmueble

2.2 Alta de Publicación

Las operaciones realizadas son:

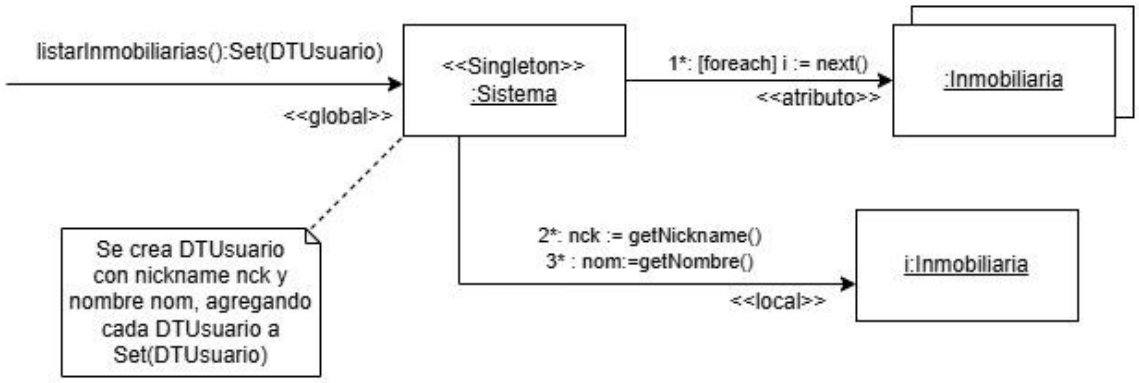
- listarInmobiliarias
- listarInmueblesAdministrados
- altaPublicacion

2.2.1 Estructura

La estructura realizada es la misma que en Alta de Usuario.

2.2.2 Interacciones

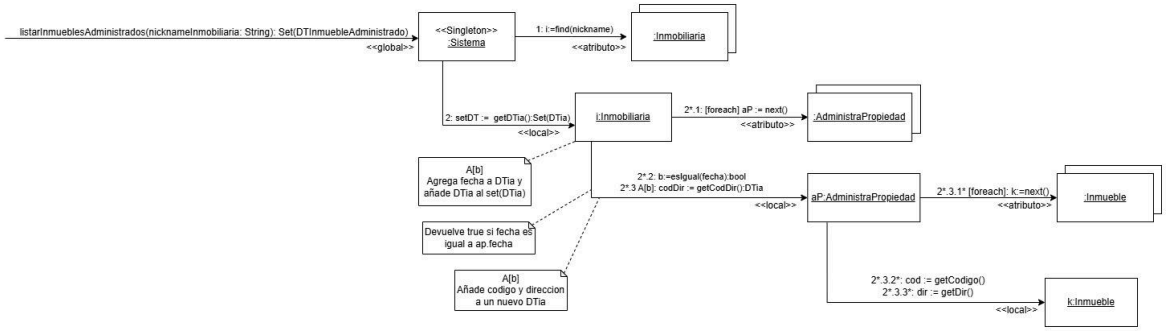
2.2.2.1 Diagrama de comunicación de listarInmobiliarias



Asignación de responsabilidades:

- Controlador de la operación: Sistema
- Experto en información de Inmobiliaria: Inmobiliaria

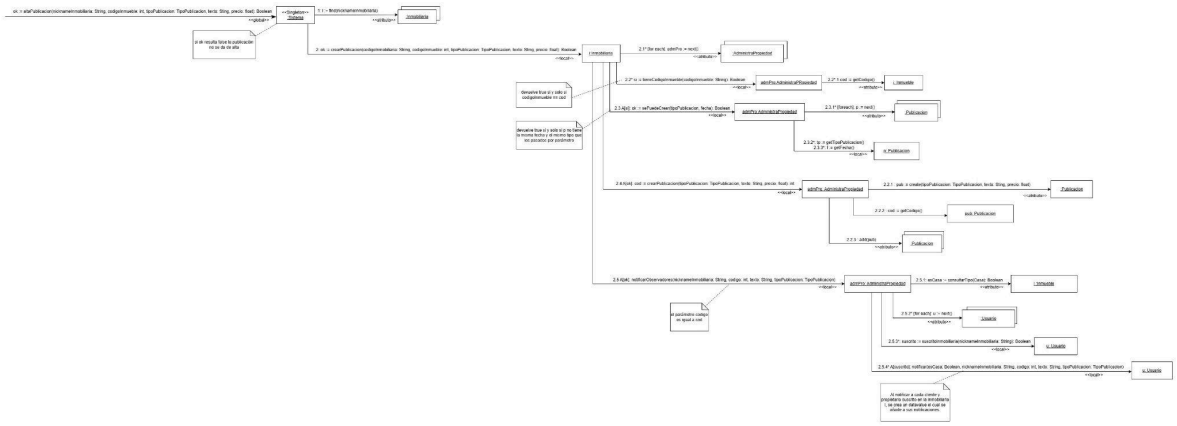
2.2.2.2 Diagrama de comunicación de listarInmueblesAdministrados



Asignación de responsabilidades:

- Controlador de la operación: Sistema
- Experto en información de los inmuebles administrados por Inmobiliaria: AdministraPropiedad
- Experto en información de Inmueble: Inmueble

2.2.2.3 Diagrama de comunicación de altaPublicacion



Asignación de responsabilidades:

- Controlador de la operación: Sistema
- Experto en información de Inmueble: Inmueble
- Experto en inmuebles administrados por Inmobiliaria: AdministraPropiedad
- Experto en información de las publicaciones: Publicacion
- Experto en crear las publicaciones: Publicacion
- Experto en notificar a los observadores: AdministraPropiedad

2.3 Consulta de publicación

Las operaciones realizadas son:

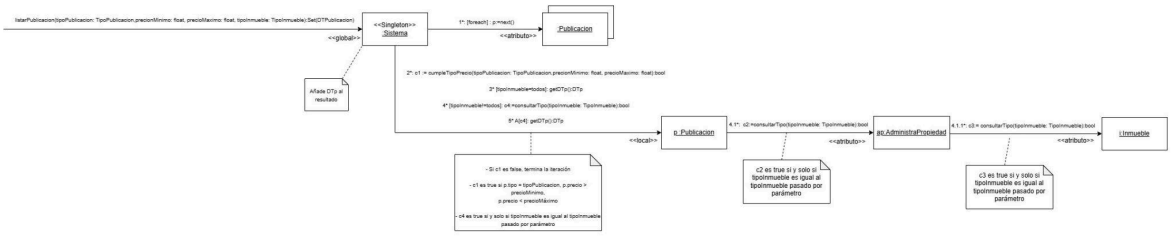
- listarPublicacion

2.3.1 Estructura

La estructura realizada es la misma que en Alta de Usuario.

2.3.2 Interacciones

2.3.2.1 Diagrama de comunicación de listarPublicacion



Asignación de responsabilidades:

- Controlador de la operación: Sistema
- Experto en información de los inmuebles: Inmueble
- Experto en información de Publicacion: Publicacion

2.4 Eliminar Inmueble

Las operaciones realizadas son:

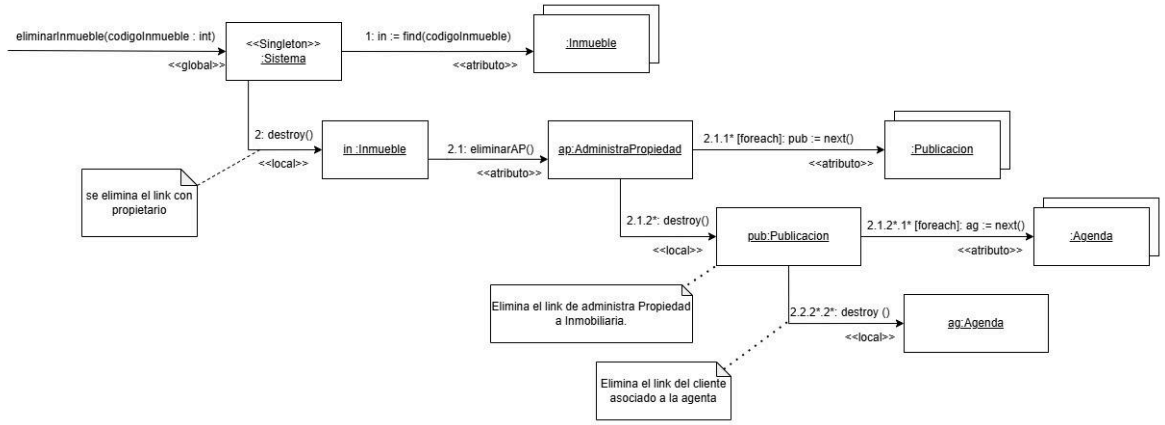
- eliminarInmueble

2.4.1 Estructura

La estructura realizada es la misma que en Alta de Usuario.

2.4.2 Interacciones

2.4.2.1 Diagrama de comunicación de eliminarInmueble



Asignación de responsabilidades:

- Controlador de la operación: Sistema
- Destructor de clientes: Cliente
- Destructor de Agenda: Agenda
- Destructor de Publicacion: Publicacion
- Destructor de Inmueble y AdministraPropiedad: Inmueble

3 Patrones de diseño

3.1 Patrón de diseño 1

Singleton

Optamos por el uso de Singleton en dos clases que consideramos totalmente necesarias, las cuales son Sistema y Factory. Para el caso de Sistema, su utilización es vital en nuestro enfoque de diseño, ya que debe haber una única instancia del sistema controlando las operaciones principales. A su vez, la implementación de Factory como Singleton nos ayuda a tener un control más preciso sobre quién obtiene instancias del sistema, siendo él, el único que puede conceder las mismas, teniendo además un acceso fácil sobre ellas.

Clases involucradas:

- Sistema, cumpliendo el rol de Singleton.
- Factory, cumpliendo el rol de una fábrica con instancia única.

3.2 Patrón de diseño 2

Factory

Decidimos utilizar Factory ya que debemos tener controladas las instancias realizadas sobre la clase Sistema, teniendo un acceso fácil y devolviendo instancias de la misma de forma coherente.

Clases involucradas:

- Factory, cumpliendo el rol de Factory.
- ISistema, cumpliendo el rol de IProveedor.
- Sistema, cumpliendo el rol del Proveedor concreto.

3.3 Patrón de diseño 3

Observer

La utilización de observer es sumamente necesaria para solucionar el problema de notificar a los suscriptores de las inmobiliarias cada vez que se realiza una publicación nueva. Utilizando observer se mitiga el problema de forma fácil y directa.

Clases involucradas:

- Usuario, cumpliendo el rol de Observer.
- Propietario, cumpliendo el rol de Observer Concreto1.
- Cliente, cumpliendo el rol de Observer Concreto2.
- Administra Propiedad, cumpliendo el rol de Subject.

4 Criterios Generales

A la hora de realizar el diseño, nos encontramos con el problema de organizar y saber delegar las responsabilidades de manejar las operaciones principales del sistema, las cuales van a interactuar directamente con el usuario y otras clases internas que las utilicen, debiendo ser de manera ordenada y sin generar conflictos entre las mismas. Para esto decidimos que el sistema sea único, actuando como controlador y tenga el control de todas las operaciones principales que se realicen.

De la mano de añadir un controlador que tenga las operaciones principales, decidimos utilizar una interfaz del mismo para encapsular los métodos y que las interacciones con la capa de presentación se realicen a través de la misma. Esto nos permite que no se conozca la implementación directa de las funciones que están en el sistema, además de tener una mejor modularidad y permitir hacer cambios en las operaciones principales del sistema sin tener que cambiar necesariamente todos los métodos asociados.

Por otro lado, las colecciones de instancias de Usuario, Inmueble y Publicacion, serán guardadas en Sistema.

5 DataType's usados