

# Programación 4

## Informe del Modelo de Casos de Uso

### Grupo 66

#### Integrantes

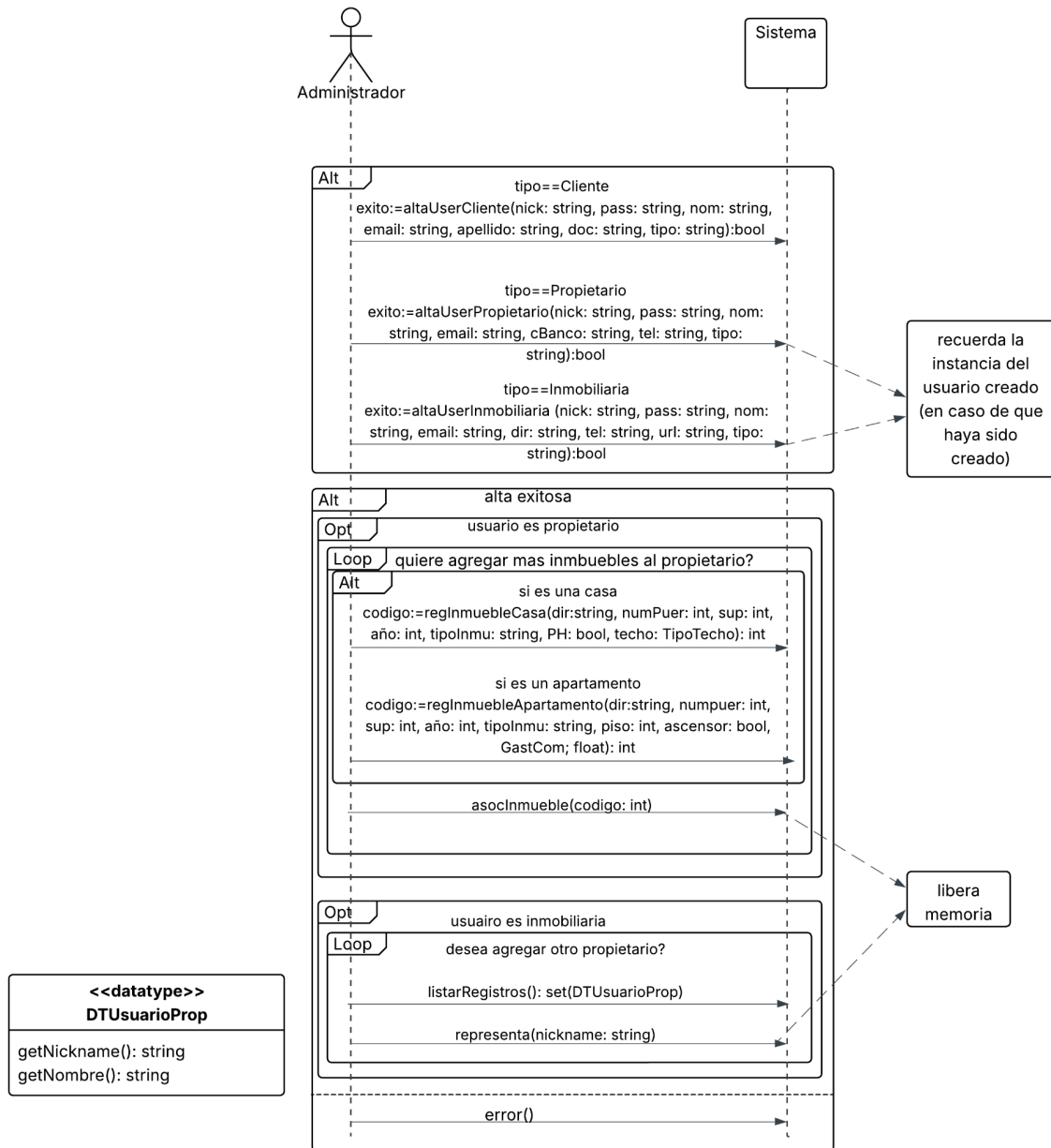
Valentin Pedragosa  
Matias Torres  
Tomás Martínez  
Emiliano Esponda  
Camilo Morganti

CI: 5.532.445-1  
CI: 5.485.366-5  
CI: 5.535.431-9  
CI: 5.491.971-6  
CI: 5.443.356-4

Docente: Federico Andrade

## Alta de usuario con memoria

### Diagramas de Secuencia del Sistema



*Contratos*

<b>Nombre</b>	<b>Alta de cliente</b>
<b>Operación</b>	<code>exito:=altaUserCliente(nick:string, pass:string, nom:string, email:string, apellido:string, doc:string, tipo:string):bool</code>
<b>Entrada</b>	nick: nickname del nuevo usuario. pass: contraseña del nuevo usuario. nom: nombre del nuevo usuario. email: email del nuevo usuario. apellido: apellido del nuevo usuario. doc: número de documento del nuevo usuario. tipo: tipo de usuario.
<b>Salida</b>	devuelve true si creo exitosamente, false en caso de que no
<b>Descripción</b>	Crea una instancia de cliente, carga sus datos correspondientes y lo da de alta si no existe otro usuario con el mismo nickname; en caso contrario devuelve false..

<b>Precondiciones y Postcondiciones</b>
<b>Pre:</b> pass debe contener al menos 6 caracteres. <b>Pre:</b> tipo debe ser cliente. <b>Post:</b> sí es creada, Existe una nueva instancia de cliente

<b>Nombre</b>	<b>Alta de propietario</b>
<b>Operación</b>	<code>altaUserPropietario(nick:string, pass:string, nom:string, email:string, cBanco:string, tel:string, tipo:string)</code>
<b>Entrada</b>	nick: nickname del nuevo usuario. pass: contraseña del nuevo usuario. nom: nombre del nuevo usuario. email: email del nuevo usuario. cBanco: cuenta bancaria del nuevo usuario. tel: teléfono de contacto del nuevo usuario. tipo: tipo de usuario.
<b>Salida</b>	Devuelve true si creo exitosamente, false en caso de que no
<b>Descripción</b>	Crea una instancia de propietario, carga sus datos correspondientes y, si no hay otra instancia propietario con el mismo nickname, es dado de alta y añadido al registro de propietarios; en caso contrario devuelve false..

Precondiciones y Postcondiciones
<p>Pre: pass debe contener al menos 6 caracteres.</p> <p>Pre: tipo debe ser propietario.</p> <p>Pre: tel debe ser un string de números.</p> <p>Post: sí es creada, Existe una nueva instancia de propietario</p> <p>Post: si la instancia es creada, se une al registro de propietarios.</p> <p>Post: sí es creada, se recuerda la nueva instancia de usuario.</p>

Nombre	Alta de inmobiliaria
Operación	altaUserInmobiliaria(nick:string, pass:string, nom:string, email:string, dir:string, tel:string, URL:string, tipo:string)
Entrada	<p>nick: nickname del nuevo usuario.</p> <p>pass: contraseña del nuevo usuario.</p> <p>nom: nombre del nuevo usuario.</p> <p>email: email del nuevo usuario.</p> <p>dir: dirección de la inmobiliaria.</p> <p>tel: teléfono de la inmobiliaria.</p> <p>URL: sitio web de la inmobiliaria.</p> <p>tipo: tipo de usuario.</p>
Salida	Devuelve true si creo exitosamente, false en caso de que no
Descripción	Crea una instancia de inmobiliaria, carga sus datos correspondientes y, si no hay otra instancia inmobiliaria con el mismo nickname, el usuario es dado de alta; en caso contrario devuelve false

Precondiciones y Postcondiciones
<p>Pre: pass debe contener al menos 6 caracteres.</p> <p>Pre: tipo debe ser inmobiliaria.</p> <p>Pre: tel debe ser un string de números.</p> <p>Post: sí es creada, Existe una nueva instancia de inmobiliaria</p> <p>Post: sí es creada, se recuerda la nueva instancia de usuario.</p>

<b>Nombre</b>	<b>Registro de Inmuebles casa</b>
<b>Operación</b>	<code>regInmuebleCasa(dir:string, numPuer:int, sup:int, año:int, tipoInmu:string, PH:bool, techo:TipoTecho):int</code>
<b>Entrada</b>	<p>dir: dirección del inmueble.</p> <p>numPuer: número de puerta del inmueble.</p> <p>sup: superficie del inmueble.</p> <p>año: año de construcción del inmueble.</p> <p>tipoInmu: tipo de inmueble.</p> <p>PH: si es propiedad horizontal.</p> <p>techo: tipo de techo.</p>
<b>Salida</b>	Código del inmueble.
<b>Descripción</b>	Crea una instancia inmueble, carga sus datos y la da de alta. Genera y devuelve un código numérico para el inmueble.

<b>Precondiciones y Postcondiciones</b>
<p><b>Pre:</b> tipoInmu debe ser casa.</p> <p><b>Post:</b> se crea una nueva instancia de inmueble.</p> <p><b>Post:</b> el código generado debe ser único para cada instancia inmueble creada.</p>

<b>Nombre</b>	<b>Registro de Inmuebles apartamento</b>
<b>Operación</b>	<code>regInmuebleApartamento(dir:string, numPuer:int, sup:int, año:int, tipoInmu:string, piso:int, ascensor:bool, gc:float):int</code>
<b>Entrada</b>	<p>dir: dirección del inmueble.</p> <p>numPuer: número de puerta del inmueble.</p> <p>sup: superficie del inmueble.</p> <p>año: año de construcción del inmueble.</p> <p>tipoInmu: tipo de inmueble.</p> <p>piso: número de piso del apartamento.</p> <p>ascensor: el edificio posee ascensor.</p>

	gc: monto de gastos comunes.
<b>Salida</b>	Código del inmueble.
<b>Descripción</b>	Crea una instancia inmueble, carga sus datos y la da de alta. Genera y devuelve un código numérico para el inmueble.

<b>Precondiciones y Postcondiciones</b>	
Pre: tipoInmu debe ser apartamento.	
Post: se crea una nueva instancia de inmueble.	
Post: el código generado debe ser único para cada instancia inmueble creada.	

<b>Nombre</b>	Asociación de inmuebles
<b>Operación</b>	asoccInmueble(codigo:int)
<b>Entrada</b>	Recibe el código del inmueble.
<b>Salida</b>	-
<b>Descripción</b>	Asocia, mediante el código, al inmueble con su propietario..

<b>Precondiciones y Postcondiciones</b>	
Pre: el código debe estar asignado a un inmueble.	
Pre: existe en el sistema una instancia de propietario recordada.	
Post: existe un link entre el inmueble y el propietario.	
Post: libera memoria.	

<b>Nombre</b>	Lista de Propietarios
<b>Operación</b>	listarRegistros():set(DTUsuarioProp)
<b>Entrada</b>	-
<b>Salida</b>	Devuelve un set con los propietarios registrados.
<b>Descripción</b>	Muestra el nickname y nombre de los propietarios registrados.

<b>Precondiciones y Postcondiciones</b>	
Pre: hay al menos un propietario registrado.	

<b>Nombre</b>	Inmobiliaria representa propietario
<b>Operación</b>	representa(nickname:string)
<b>Entrada</b>	Recibe el nickname del propietario que la inmobiliaria representa.

<b>Salida</b>	-
<b>Descripción</b>	Asocia a la inmobiliaria con su representado..

**Precondiciones y Postcondiciones**

Pre: el nickname debe estar asignado a un propietario registrado.

Pre: existe en el sistema una instancia de inmobiliaria recordada.

Post: se crea una nueva asociación entre la inmobiliaria y el propietario.

Post: libera memoria.

<b>Nombre</b>	<b>Error</b>
<b>Operación</b>	<code>error()</code>
<b>Entrada</b>	No aplica
<b>Salida</b>	No aplica
<b>Descripción</b>	Informa que el sistema ha tenido un error en la alta de usuario

**Precondiciones y Postcondiciones**

Pre: el nickname debe estar asignado a un propietario registrado.

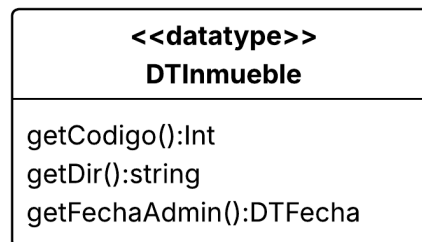
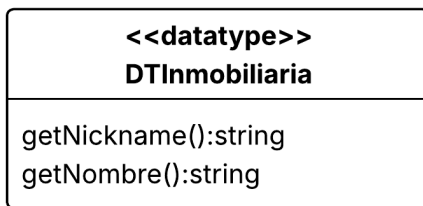
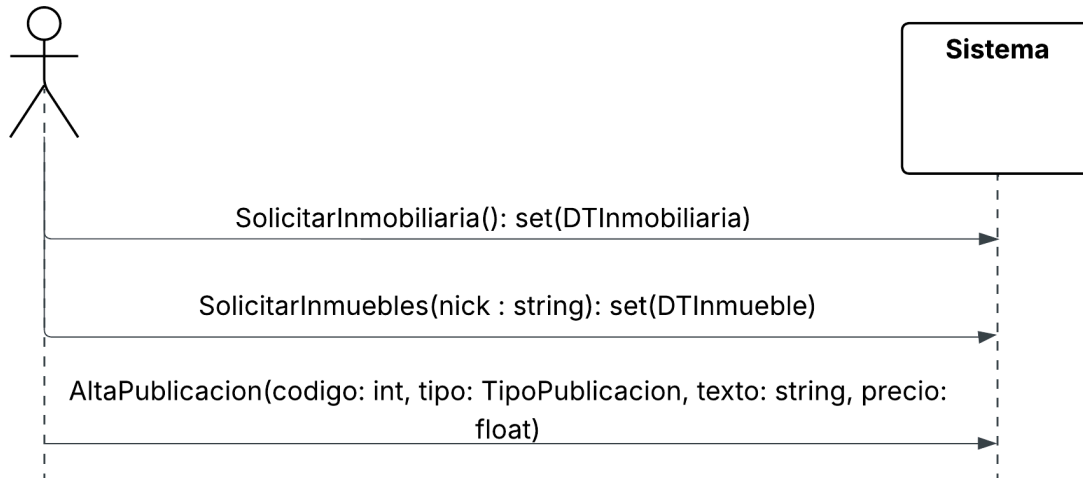
Pre: existe en el sistema una instancia de inmobiliaria recordada.

Post: se crea una nueva asociación entre la inmobiliaria y el propietario.

Post: libera memoria.

## Alta de Publicación

### Diagramas de Secuencia del Sistema



### Contratos

<b>Nombre</b>	<b>SolicitarInmobiliaria</b>
<b>Operación</b>	<code>solicitarInmobiliaria():set(DTInmobiliaria)</code>
<b>Entrada</b>	No aplica.
<b>Salida</b>	Set con todas las inmobiliarias.



<b>Descripción</b>	Devuelve un set con todas las inmobiliarias
--------------------	---

<b>Precondiciones y Postcondiciones</b>
<b>Pre:</b> Existe al menos una inmobiliaria cargada en el sistema.

<b>Nombre</b>	<b>SolicitarInmuebles</b>
<b>Operación</b>	solicitarInmuebles(nick:string):set(DTInmueble)
<b>Entrada</b>	nick: Nickname de la inmobiliaria a listar los inmuebles
<b>Salida</b>	Set de DTInmuebles administrados por la inmobiliaria con nickname <b>nick</b>
<b>Descripción</b>	Muestra los inmuebles administrados por la Inmobiliaria con nickname <b>nick</b>

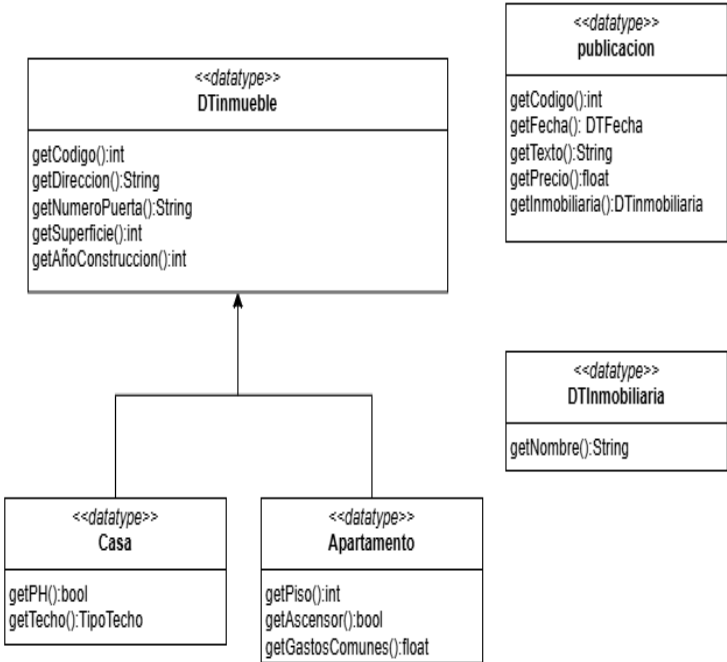
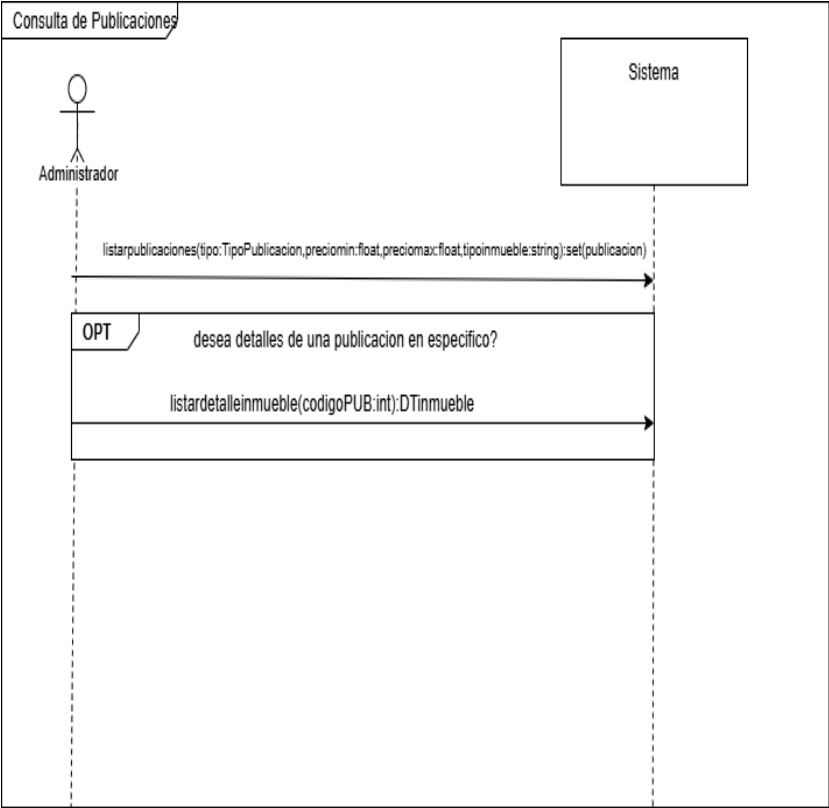
<b>Precondiciones y Postcondiciones</b>
<b>Pre:</b> Existe una inmobiliaria con nickname <b>nick</b>

<b>Nombre</b>	<b>AltaPublicacion</b>
<b>Operación</b>	altaPublicacion(codigo:int, tipo:TipoPublicacion, texto:string, precio:float)
<b>Entrada</b>	codigo: Código del inmueble a crear la publicación. tipo: Tipo de publicación. texto: Texto de la publicación. precio: Precio de la publicacion
<b>Salida</b>	-
<b>Descripción</b>	Si existe una publicación para la fecha y tipo no se da el alta y muestra mensaje de error, si no, da el alta con los datos dados, le asigna un código incremental generado por el sistema, la fecha del sistema y se pone como publicación activa. Por último crea el link con la instancia de AdministraPropiedad del inmueble con codigo <b>codigo</b> .

<b>Precondiciones y Postcondiciones</b>
<b>Pre:</b> Existe un inmueble con código <b>codigo</b>
<b>Post:</b> Existe una nueva instancia de Publicacion
<b>Post:</b> Existe un link entre la publicación creada y AdministraPropiedad

# Consulta de publicaciones

## Diagramas de Secuencia del Sistema



*Contratos*

<b>Nombre</b>	<b>ListarPublicaciones</b>
<b>Operación</b>	<code>ListarPublicaciones(tipo:TipoPublicacion,preciomin:float,preciomax:float,tipoinmueble:string):set(publicacion)</code>
<b>Entrada</b>	<p>tipo: es el tipo de la publicación</p> <p>preciomin: es el precio mínimo de las publicaciones</p> <p>preciomax: es el precio máximo de las publicaciones</p> <p>tipoinmueble: es el tipo del inmueble</p>
<b>Salida</b>	Set de publicaciones que cumplen con los filtros
<b>Descripción</b>	Filtrar y mostrar las publicaciones activas que coincidan con los intereses del administrador, tipo de publicación, rango de precio, y tipo de inmueble en caso de que se especifique

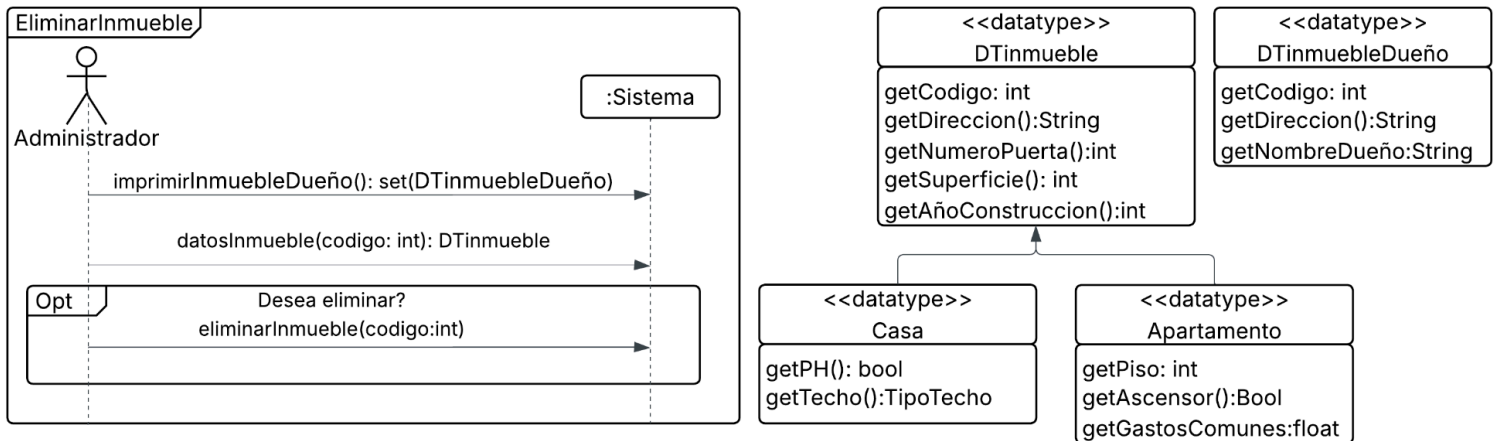
<b>Precondiciones y Postcondiciones</b>
<p><b>Pre:</b> Existe al menos una publicación activa.</p> <p><b>Pre:</b> <code>preciomin ≤ preciomax</code>.</p> <p><b>Pre:</b> <code>tipoinmueble</code> debe ser exactamente "Casas", "Apartamentos" o "Todos".</p>

<b>Nombre</b>	<b>ListarDetalleInmueble</b>
<b>Operación</b>	<code>ListarDetalleInmueble(codigo:int):DTInmueble</code>
<b>Entrada</b>	codigo: es el código de una publicación
<b>Salida</b>	Retorna toda la información del inmueble
<b>Descripción</b>	Muestra el detalle completo del inmueble asociado a la publicación de dicho código

<b>Precondiciones y Postcondiciones</b>
<b>Pre:</b> Existe una publicación activa con ese código

## Eliminar Inmueble

### Diagramas de Secuencia del Sistema



### Contratos

Nombre	Imprimir Inmuebles
Operación	<code>imprimirInmuebleDueño(): set (DTinmuelleDueño)</code>
Entrada	-
Salida	Set de DTinmuelleDueño
Descripción	Retorna el código, la dirección, el nombre del propietario para cada inmueble en el sistema.

Nombre	Listar Detalle Del Inmueble
Operación	<code>datosInmueble(codigo: int): DTinmuelle</code>
Entrada	codigo: Es el codigo del inmueble
Salida	Devuelve un DTinmuelle con información del inmueble
Descripción	Retorna toda la información del inmueble

<b>Nombre</b>	<b>Eliminar Inmuebles</b>
<b>Operación</b>	<code>eliminarInmueble(codigo: Int)</code>
<b>Entrada</b>	codigo: es el codigo de un inmueble
<b>Salida</b>	-
<b>Descripción</b>	Debe dejar inactivas todas las publicaciones del inmueble, las agendas asociadas al inmueble y borrar el inmueble.

<b>Precondiciones y Postcondiciones</b>
<b>Pre:</b> El inmueble debe existir <b>Post:</b> Se borra agenda <b>Post:</b> Se cambian los estados de las publicaciones activas <b>Post:</b> se borra inmueble