

# Práctica 5. La Agenda

## Solución

---

1 REQUISITOS DEL NEGOCIO	3
2 ANÁLISIS	7
2.1 Sumario	7
2.2 Meta	7
2.3 Stakeholders (Participantes)	7
2.3 Características básicas de la aplicación	7
2.4 Casos de Uso	8
2.4.1 Diagramas	8
2.4.2. Agregar Contacto	9
2.4.3. Agregar Cita	10
2.4.3. Buscar Cita	11
3 DISEÑO	12
3.1 Diagrama de clases conceptual	12
3.3 Diagrama de secuencia Insertar Contacto	13
3.3 Diagrama de secuencia Insertar Cita	14

## 1 REQUISITOS DEL NEGOCIO

<b>REQ01</b>	La agenda almacena los datos de contacto de personas	
Como	dueño de la agenda	
Quiero	registrar los datos de mis contactos	
Para	poder consultarlos y de esta forma facilitar la comunicación con ellos	
	Criterio Aceptación 01 - A	
	Dado	un nuevo contacto
	Cuando	lo añada a la agenda
	Entonces	podré consultarlo
	Criterio Aceptación 01 - B. Advierte de posibles contactos duplicados	
	Dado	un nuevo contacto duplicado de uno ya existente
	Cuando	se intenta añadir a la agenda
	Entonces	muestra una advertencia

<b>REQ02</b>	La agenda permite borrar contactos.	
Como	Dueño de la agenda	
Quiero	Mantener mis contactos actualizados eliminando los contactos antiguos o repetidos	
Para	facilitar mis búsquedas y consultas	
	Criterio Aceptación 02 - A	
	Dado	un contacto existente en la agenda
	Cuando	lo borre
	Entonces	no podrá consultarse

<b>REQ03</b>	Los contactos se clasifican en dos tipos: contactos personales y profesionales.	
Como	dueño de la agenda	
Quiero	diferenciar entre contactos personales y profesionales	
Para	facilitar mis búsquedas	
	Criterio Aceptación 03 - A. Un contacto personal puede ser también un contacto profesional.	
	Dado	un amigo que además es compañero de trabajo
	Cuando	lo clasifique en la agenda
	Entonces	será clasificado como tanto contacto profesional como personal

<b>REQ04</b>	La agenda permite crear citas con un contacto almacenado en la agenda	
Como	dueño de la agenda	
Quiero	poder citar a mis contactos directamente desde la agenda	
Para	agilizar la reserva de mi tiempo y de salas	
	Criterio Aceptación 04 - A. La cita es personal o profesional en función del tipo de contacto	
	Dado	un contacto de un determinado tipo
	Cuando	lo cite en la agenda
	Entonces	La cita será del mismo tipo que el contacto.
	Criterio Aceptación 04 - B. Advierte el solapamiento de citas	
	Dada	una cita que se solapa temporalmente sobre otras ya existentes
	Cuando	se guarda la cita en la agenda
	Entonces	advertirá del solapamiento
	Criterio Aceptación 04 - C. La búsqueda de citas se realiza por fecha y/o	

	los datos de contacto: teléfono, nombre y apellido	
	Dada	Una fecha (futura o pasada) o un contacto
	Cuando	se buscan las citas
	Entonces	Se devuelven todas las citas con los criterios coincidentes

<b>REQ05</b>	Advierte al usuario de una una cita planificada cuando queda menos de un plazo configurable	
Como	dueño de la agenda	
Quiero	que la agenda me recuerde las citas futuras con la suficiente antelación	
Para	no olvidar asistir y, además, avisar con antelación al contacto de mi posible retraso o no asistencia.	
	Criterio Aceptación 05 - A.	
	Dada	Un cita con un contacto y un tiempo de aviso preconfigurado
	Cuando	Falte el tiempo de aviso preconfigurado para la cita
	Entonces	La agenda mostrará un aviso indicando la futura cita con el contacto.

<b>REQ06</b>	Una cita puede estar aceptada, pendiente de confirmación o realizada	
Como	dueño de la agenda	
Quiero	poder etiquetar una cita como aceptada, pendiente de confirmación o realizada	
Para	ayudarme a planificar mis citas y reuniones.	
	Criterio Aceptación 06 - A	
	Dada	un nueva cita para un contacto
	Cuando	guarde la cita
	Entonces	se guardará la cita como pendiente de confirmación

	Criterio Aceptación 06 - B	
	Dada	una cita etiquetada como “pendiente de confirmación”
	Cuando	el paciente confirme la cita
	Entonces	el dueño de la agenda cambiará su estado a “aceptada”
	Criterio Aceptación 06 - C	
	Dada	una cita “aceptada”
	Cuando	el tiempo exceda a su finalización
	Entonces	cambiará automáticamente a “realizada”

## **2 ANÁLISIS**

### **2.1 Sumario**

El problema plantea la creación de una agenda. En dicha agenda se distinguen dos tipos de contactos, los personales y los profesionales que tendrán características comunes y otras propias del tipo. Por otra parte, en la agenda se pueden crear citas a partir de los contactos.

### **2.2 Meta**

Ayudar en la vida cotidiana o profesional de cada persona, ofreciendo una manera de organizar sus contactos y sus citas al mismo tiempo, uniendo los contactos con las citas. Para ello permite crear, modificar y eliminar citas y contactos. Además debe poder buscar los contactos y las citas.

### **2.3 Stakeholders (Participantes)**

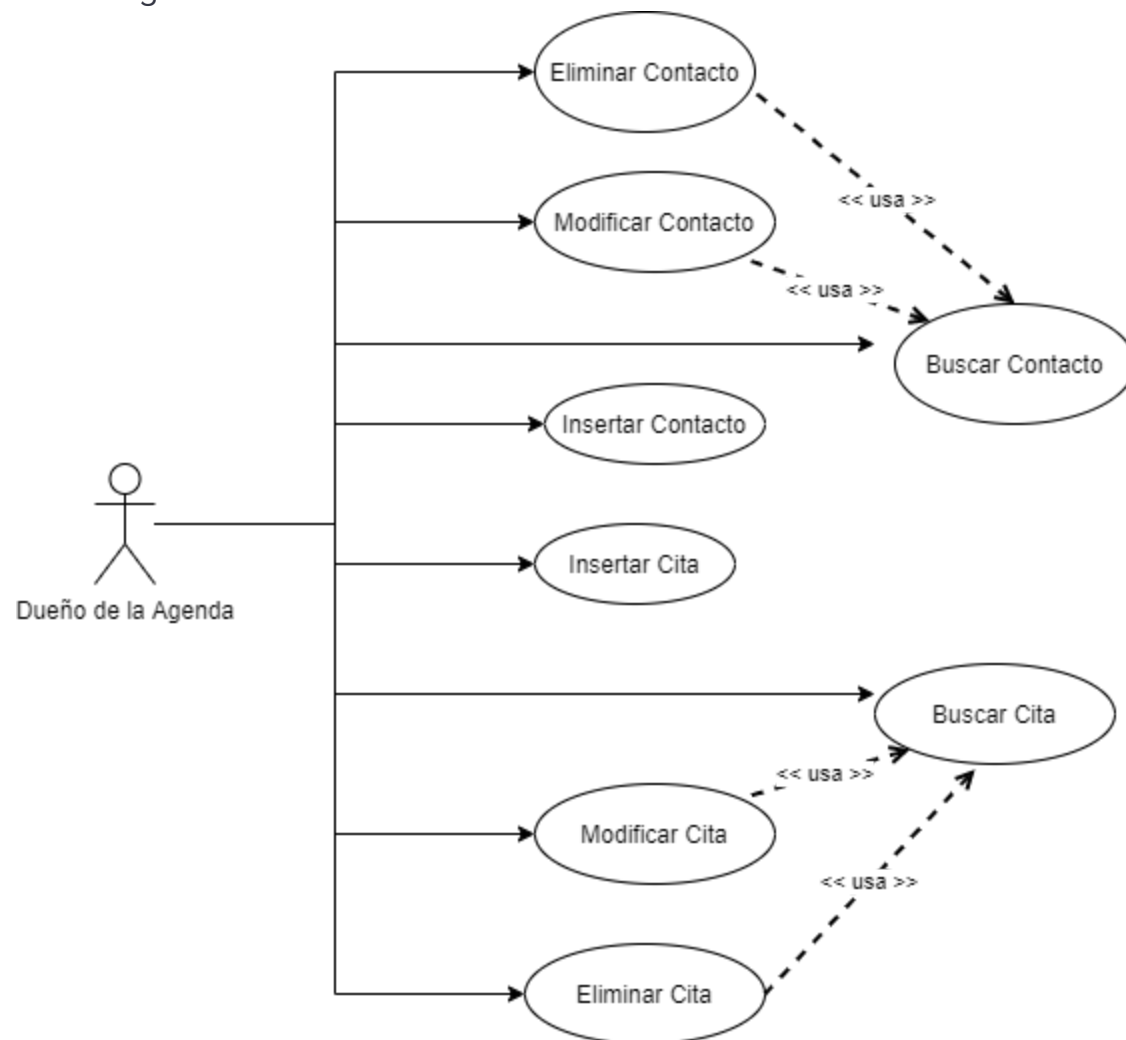
La agenda puede ser de interés para cualquier persona. El usuario habitual es un profesional que tiene muchas citas con otros de otras empresas.

### **2.3 Características básicas de la aplicación**

El usuario interactúa con un interface gráfico donde selecciona lo que quiere hacer. La aplicación no se ralentiza conforme aumenta el número de contactos y/o citas. El producto será una aplicación de escritorio. El patrón de presentación será Modelo - Vista - Presentador (Vista-Pasiva). La base de datos será persistida en ficheros de texto.

## 2.4 Casos de Uso

### 2.4.1 Diagramas



### 2.4.2. Agregar Contacto

<b>Identificador</b>	CU-01
<b>Nombre</b>	Agregar Contacto
<b>Meta</b>	Añadir un contacto nuevo a la agenda evitando duplicados



Actores	Usuario	
Precondición	no hay otro contacto con el mismo nombre o el mismo número de teléfono	
Flujo básico		
	Paso 1	El usuario indica que quiere añadir un nuevo contacto
	Paso 2	La agenda muestra los campos necesarios
	Paso 3	El usuario introduce el nombre, los apellidos, teléfono y tipo de contacto.
	Paso 4	La agenda comprueba que no hay otro contacto con el mismo nombre, apellidos o número.
	Paso 5	La agenda muestra el resto de datos específicos del tipo para terminar de completar los datos
	Paso 6	El usuario introduce los campos adicionales y confirma almacenamiento
Flujo alternativo		
	Paso 4	Incumplimiento del formato de alguno de los campos. La agenda informa del error y pide una corrección
	Paso 4	Existe otro contacto con el mismo nombre/apellidos o teléfono. La agenda avisa de un duplicado y permite corregir o mantener el duplicado.
Postcondición	El nuevo contacto se puede consultar en la agenda	

### 2.4.3. Agregar Cita

<b>Identificador</b>	CU-02
<b>Nombre</b>	Agregar Cita
<b>Meta</b>	añadir una nueva cita a un contacto
<b>Actores</b>	Usuario

Precondición	el usuario no tiene una cita a la misma hora. La agenda del contacto con sus citas no está disponible.	
Flujo básico		
	Paso 1	El usuario solicita añadir una cita
	Paso 2	El usuario selecciona un contacto
	Paso 3	La agenda solicita la fecha/hora y lugar.
	Paso 4	La agenda comprueba que no tiene otra cita a la misma hora y almacena la cita
	Paso 5	La agenda muestra los datos de la cita
Flujo alternativo		
	Paso 4	Incumplimiento del formato en alguno de los campos. La agenda informa al usuario sobre el error y pide corrección
	Paso 4	Existe otra cita a la misma hora. La agenda avisa al usuario. Permite solapar o solicita modificar la cita.
	Paso 4	El contacto introducido no ha sido encontrado. La agenda informa al usuario sobre el error y pide otro contacto
Postcondición	La cita se crea y se almacena	

### 2.4.3. Buscar Cita

<b>Identificador</b>	CU-03
<b>Nombre</b>	Buscar cita
<b>Meta</b>	Buscar una cita existente
<b>Actores</b>	Usuario
<b>Precondición</b>	La cita existe para un contacto

Flujo básico		
	Paso 1	El usuario indica que quiere buscar una cita
	Paso 2	La agenda muestra opciones para buscar una cita
	Paso 3	El usuario introduce la fecha y/o el nombre del contacto
	Paso 4	La agenda muestra la cita que coincida todos o parte de los campos de búsqueda.
Flujo alternativo		
	Paso 4	Incumplimiento del formato en alguno de los campos. La agenda informa al usuario de error y pide una corrección
	Paso 4	Los datos de búsqueda no coinciden con un contacto almacenado. Se muestra un aviso y algunos contactos similares.
Postcondición	Se muestra la cita que coincide con los campos buscados	

## 3 DISEÑO

### 3.1 Diagrama de clases conceptual

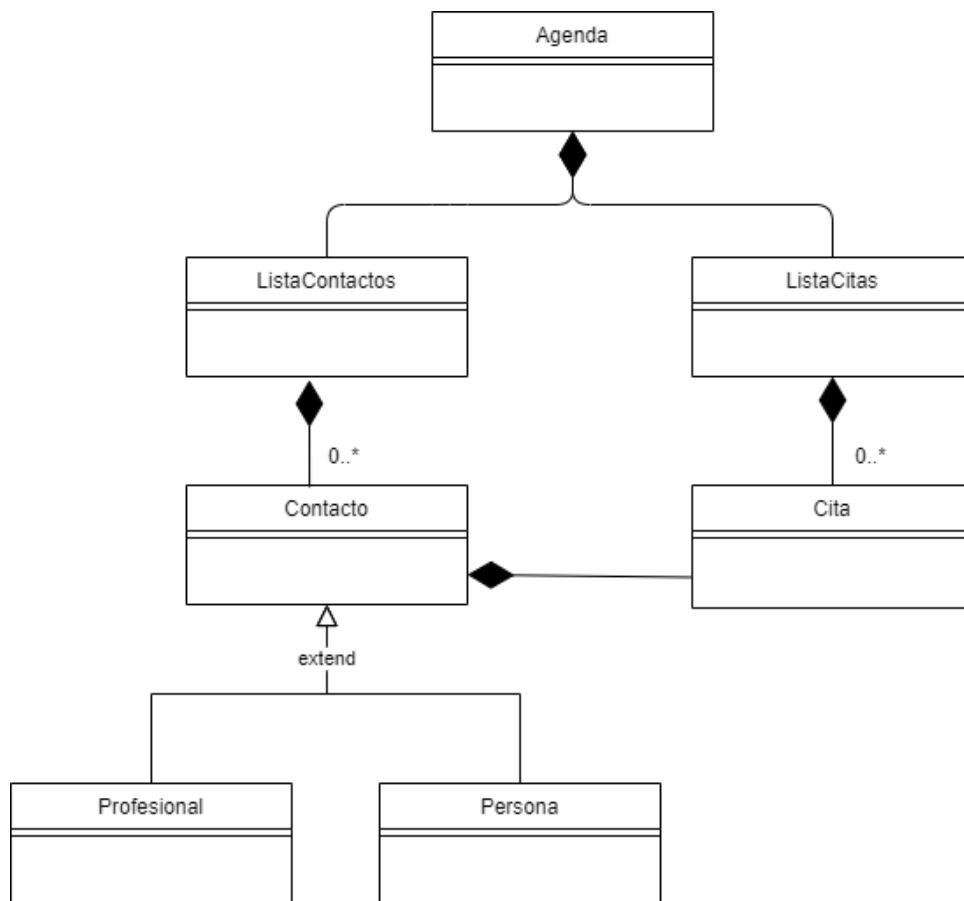
Debería existir la *clase* base *Contacto*, de la que extiendan *Personal* y *Profesional*.

Contactos debería ser una colección de “*Contacto*”, para poder tener tanto contactos de tipo *Personal* como *Profesional*.

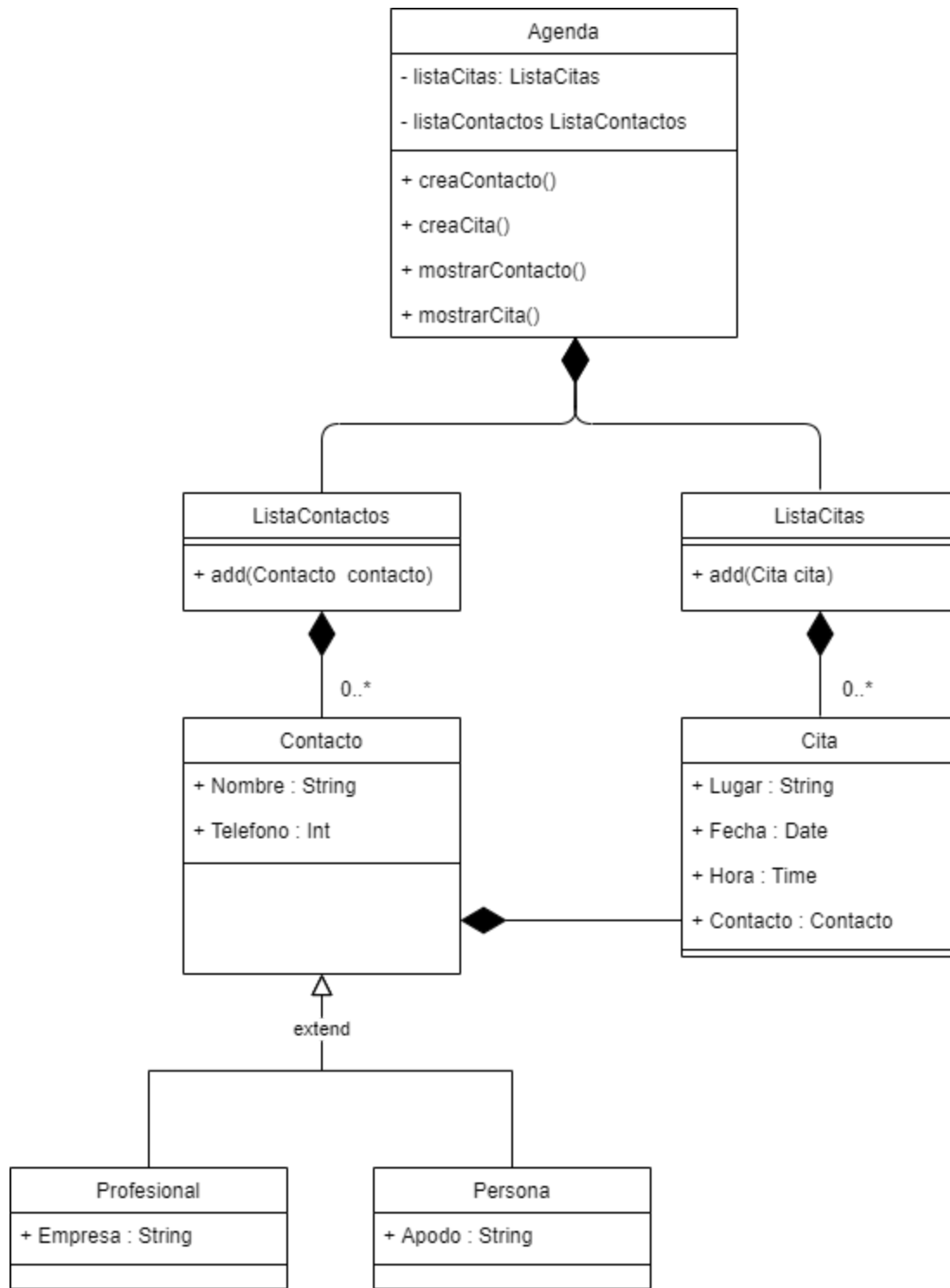
Si cada *Contacto* tuviera su colección de *Citas*, para listar todas las citas habría que buscar todos los contactos con citas. La alternativa sería definir una clase *ListaCitas* que fuese la colección de citas con cualquier contacto.

Agenda es una composición de colecciones de *Contacto* y *Cita*: *ListaContactos* y *ListaCitas*.

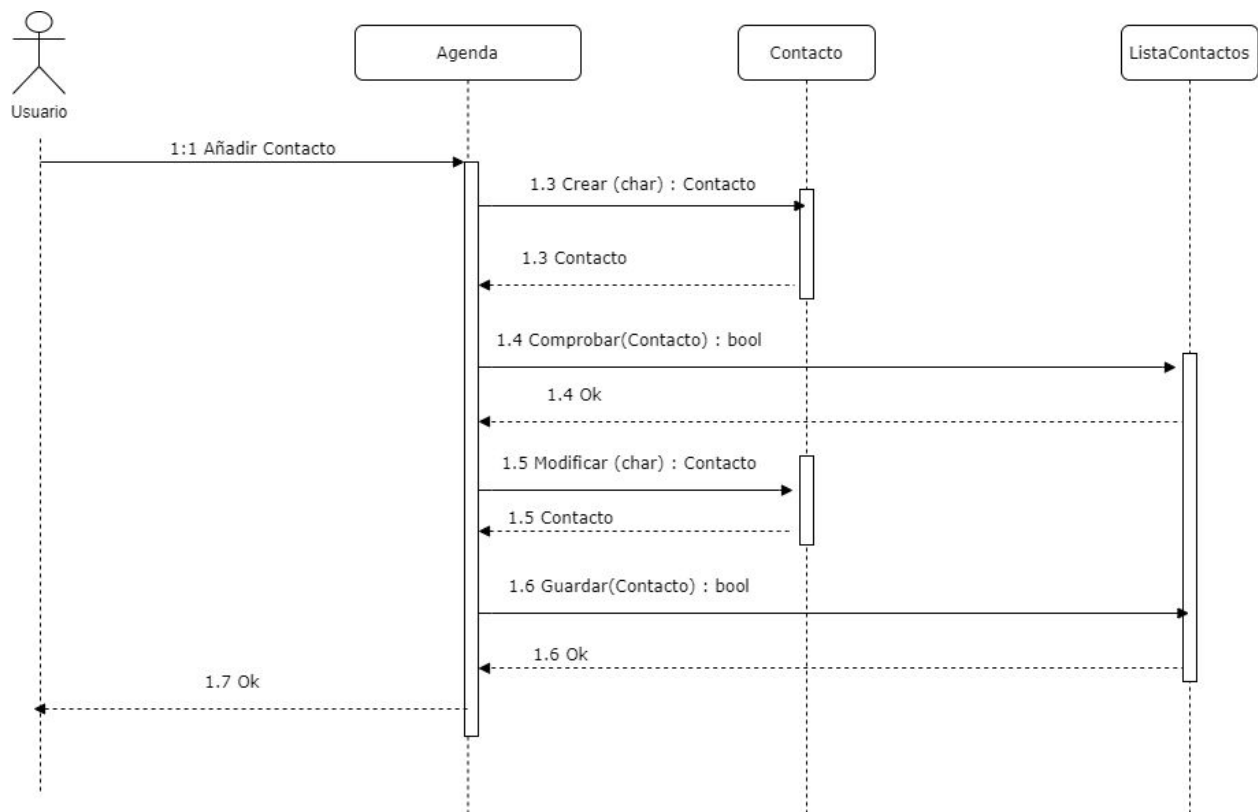
*Contacto* debe ser implementado como *abstract* o un *interface* por el **Principio de inversión de dependencia**: Las clases deben depender de abstracciones no de implementaciones.



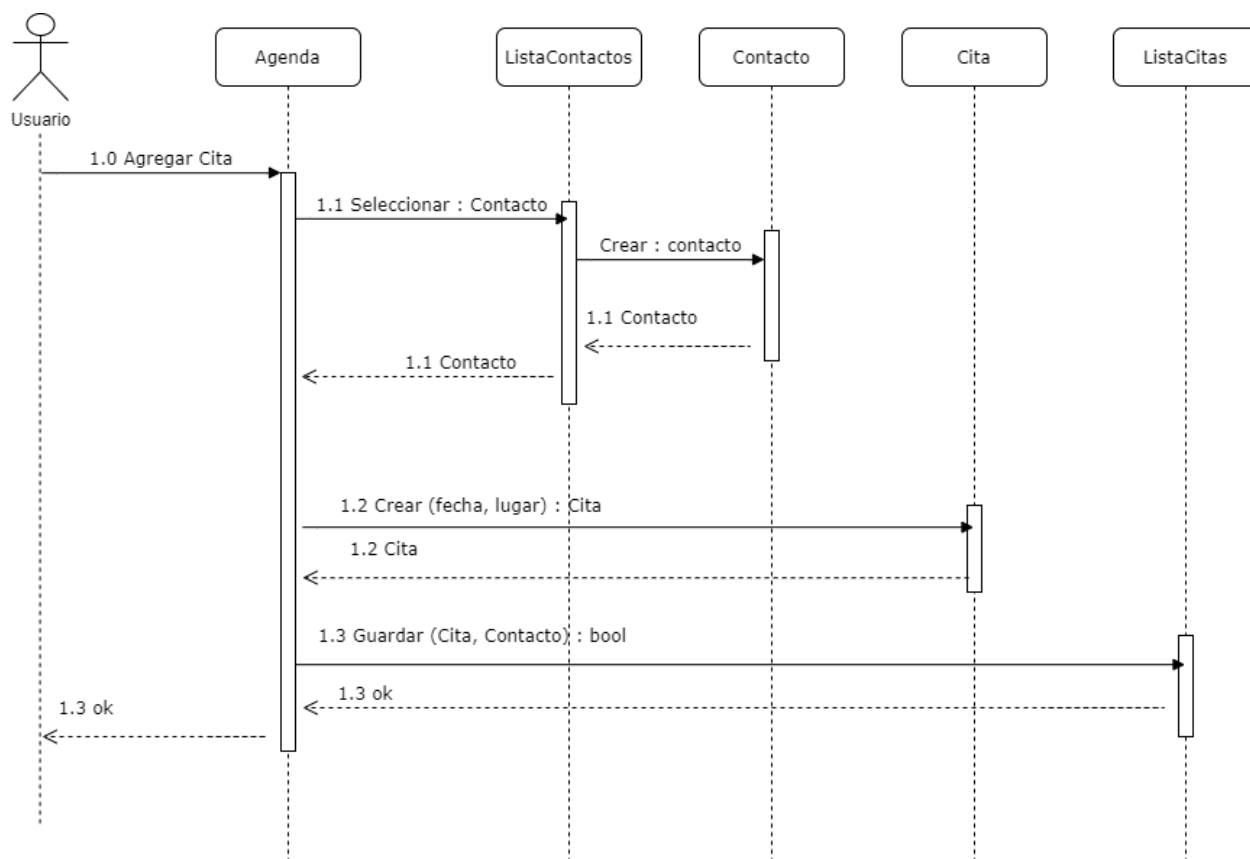
### 3.2 Diagrama de clases detallado



### 3.3 Diagrama de secuencia Insertar Contacto



### 3.3 Diagrama de secuencia Insertar Cita



### 3.3 Diagrama de secuencia Buscar Cita

