

Programación 4

Informe del Modelo de Casos de Uso

Grupo 32 Integrantes

Bernardino Texeira
Camilo de León
Carla Favale
Pablo Martinez

CI: 4.137.616-7
CI: 4.694.454-3
CI: 4.530.607-9
CI: 5.170.824-5

Docente: Jorge Corral

Indice

Indice.....	3
1Introducción.....	4
1.1Propósito.....	4
1.2Alcance.....	4
1.3Estructura del Documento.....	4
2Actores y Casos de Uso.....	5
2.1Actores.....	5
2.2Casos de Uso.....	5
3Comportamiento de Casos de Uso.....	5
3.1Nombre del Caso de Uso.....	5
3.1.1Diagramas de Secuencia del Sistema.....	5
3.1.2Contratos.....	5

1 Introducción

1.1 Propósito

El propósito de este documento es brindar una descripción general del Modelo de Casos de Uso.

1.2 Alcance

El informe del Modelo de Casos de Uso presenta una descripción de los casos de uso definidos, los actores y las asociaciones entre estos. También contiene una especificación del comportamiento declarado en los casos de uso.

1.3 Estructura del Documento

El documento está dividido en cuatro secciones. La segunda sección presenta una descripción de los actores y casos de uso contenidos en el modelo junto con las relaciones existentes entre ellos. Por último, la tercera sección presenta la especificación del comportamiento de cada caso de uso descrito.

2 Actores y Casos de Uso

2.1 Actores

Actor	Usuario socio
Descripción	Este actor es capaz de reservar consultas, devolverlas y consultar su estado en el sistema.

Actor	Usuario médico
Descripción	Este actor es capaz de listar todas las representaciones estandarizadas de diagnósticos, ingresar diagnósticos a una consulta determinada y obtener el historial de un paciente.

Actor	Usuario administrativo
Descripción	Es el encargado de ingresar o reactivar a otro usuario, obtener información de los usuarios que ingresó o reactivó, ingresar nuevos medicamentos al sistema al igual que nuevas representaciones estandarizadas de diagnósticos, listar las representaciones estandarizadas de diagnósticos existentes en el sistema, registrar una consulta y obtener la cantidad de consultas existentes por categoría de diagnóstico.

2.2 Casos de Uso

Nombre	Iniciar Sesión
Actores	Usuario
Descripción	<p>El caso de uso comienza cuando el Usuario desea iniciar sesión en el sistema Para esto ingresa su ci.</p> <p>Si se trata del usuario administrativo por defecto, el Sistema solicita su contraseña y si es correcta se asigna la sesión al usuario.</p> <p>Si es la primera vez que entra al sistema, el Sistema le solicita una contraseña indicando las reglas de caracteres y largo a seguir. Si la contraseña ingresada es válida, se activa al usuario, se almacena la contraseña y se asigna la sesión al Usuario.</p> <p>Si no es la primera vez, el Sistema solicita su contraseña y, si es correcta y el Usuario se encuentra activo, se asigna la sesión al Usuario. En caso de estar inactivo se despliega el mensaje "Usuario inactivo, no se puede realizar el inicio de sesión."</p> <p>En todos los casos, si se ingresa una contraseña no válida o incorrecta, el Usuario podrá volver a ingresar otra o cancelar el Caso de Uso.</p>

Nombre	Cerrar Sesión
Actores	Usuario
Descripción	El caso de uso permite cerrar la sesión iniciada por un Usuario. Para ello previamente debe haber una sesión asignada.

Nombre	Alta/Reactivación de Usuario
Actores	Usuario administrativo
Descripción	<p>El caso de uso comienza cuando un Usuario Administrativo que ha iniciado una sesión desea dar de alta o reactivar a un Usuario en el Sistema. Para esto el Usuario Administrativo ingresa en primer lugar una ci.</p> <p>En caso de que no exista un usuario con esa ci, procede a ingresar el nombre y apellido, sexo, fecha de nacimiento, y la o las categorías (Médico, Administrativo, Socio) para efectuar un alta.</p> <p>En caso de que ya exista un usuario con esa ci, el Sistema le devuelve el resto de los datos del usuario (nombre y apellido, sexo, edad y las categorías) así como si el usuario está activo o no. En caso de que este inactivo y concuerde con los datos del usuario que desea reactivar, el usuario administrativo puede realizar la reactivación.</p> <p>En cualquier otro caso, el usuario administrativo podrá volver a ingresar otra ci o cancelar el Caso de Uso.</p> <p>En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, o realiza el alta o la reactivación correspondiente.</p>

Nombre	Usuarios dados de alta y reactivados
Actores	Usuario administrativo
Descripción	<p>El caso de uso comienza cuando un Usuario Administrativo que ha iniciado una sesión desea obtener sus usuarios dados de alta al sistema. El sistema lista los usuarios dados de alta y reactivados por el usuario administrativo, mostrando para cada elemento la fecha del alta o reactivación, el tipo de operación (alta o reactivación), nombre y apellido, ci, edad y si se encuentra activo.</p>

Nombre	Alta medicamento
Actores	Usuario administrativo
Descripción	<p>El caso de uso comienza cuando un Usuario Administrativo que ha iniciado una sesión desea dar de alta un nuevo medicamento en el Sistema, ingresando el nombre y controlando que no exista otro medicamento con el mismo nombre en el sistema. En caso de existencia podrá volver a ingresar otro nombre o cancelar el Caso de Uso. Luego el Sistema da de alta al nuevo Medicamento.</p>

Nombre	Alta de representación estandarizada de diagnósticos
Actores	Usuario administrativo
Descripción	<p>El caso de uso comienza cuando un Usuario Administrativo que ha iniciado una sesión desea dar de alta nuevas representaciones estandarizadas de diagnósticos.</p> <p>En primer lugar obtiene la lista de las categorías de los diagnósticos. Si encuentra la categoría correspondiente a la representación del diagnóstico que desea agregar, la selecciona. De lo contrario, agrega una nueva categoría indicando la letra y la etiqueta de la categoría, quedando esta nueva categoría como seleccionada.</p> <p>Posteriormente pasa a agregar la representación del diagnóstico en la categoría previamente seleccionada. Para ello ingresa solamente la parte del número (2 cifras) correspondiente al código y la etiqueta del diagnóstico.</p> <p>En caso de que ya exista una representación estandarizada con dicho código o etiqueta, el usuario podrá volver a ingresar otro número y/o etiqueta. Con la categoría seleccionada, puede agregar tantas representaciones de diagnósticos correspondiente a ella como lo desee.</p> <p>Si lo desea, puede agregar tantas representaciones de diagnósticos en distintas categorías como el Usuario Administrativo lo desee siguiendo los pasos descriptos.</p> <p>Finalmente, el Sistema da de alta las nuevas representaciones estandarizadas de diagnósticos así como las nuevas categorías, si corresponde.</p> <p>En todo momento el usuario podrá cancelar el Caso de Uso.</p>

Nombre	Listar representaciones estandarizadas
Actores	Usuario administrativo, Usuario médico
Descripción	<p>El caso de uso comienza cuando un Usuario Administrativo o un Usuario Médico que ha iniciado una sesión desea ver las representaciones estandarizadas presentes en el sistema. A continuación, el sistema debe listar las representaciones indicando el código y la etiqueta para cada representación ordenadas por los códigos dentro de la categoría y agrupándolas por las categorías dando su letra y etiqueta ordenadas por las letras.</p> <p>Ejemplo:</p> <p>A Categoría-A</p> <p> A01 Etiqueta-A01</p> <p> ...</p> <p> ANN Etiqueta-ANN</p> <p> ...</p> <p>...</p> <p>P Categoría-P</p> <p> ...</p> <p> P03 Etiqueta-P03</p> <p> ...</p> <p>...</p>

Nombre	Reserva consulta
Actores	Usuario socio
Descripción	El caso de uso comienza cuando un Usuario Socio que ha iniciado una sesión desea agendar una nueva consulta. Para esto se listan todos los médicos del sistema, el usuario elije uno, e indica la fecha y hora para la consulta. Luego el Sistema da de alta la nueva reserva de consulta.

Nombre	Devolución de reserva
Actores	Usuario socio
Descripción	El caso de uso comienza cuando un Usuario Socio que ha iniciado una sesión desea devolver una reserva. Para esto se listan todas las reservas activas (cuya consulta no ha ocurrido aún) a la fecha para el socio. El socio elije una de ellas y el sistema la da de baja.

Nombre	Registro de consulta
Actores	Usuario administrativo
Descripción	El caso de uso comienza cuando un Usuario Administrativo que ha iniciado una sesión desea registrar la consulta de un paciente. Para esto el usuario administrativo ingresa la ci del médico y del paciente. Si la consulta es común proporciona además la fecha de la reserva registrando que el paciente asistió a la consulta y la fecha de la consulta. Al relevar este caso de uso se identificó que no puede existir más de una reserva ni consulta de un socio para un mismo médico para una fecha dada. Por otro lado, si la consulta es de emergencia el usuario administrativo registra la fecha de la consulta así como el motivo de la misma.

Nombre	Alta diagnósticos de una consulta
Actores	Usuario médico
Descripción	El caso de uso comienza cuando un Usuario Médico que ha iniciado una sesión desea dar de alta los diagnósticos de una consulta. El médico obtiene la lista todas las consultas de ese día indicando la ci del paciente de cada una de ellas. El médico selecciona la consulta indicando la ci del paciente. Seleccionada la consulta, el médico agrega los diagnósticos y para cada uno de ellos, si corresponde, los tratamientos. Para cada diagnóstico que desea agregar, el médico solicita la lista de las representaciones estandarizadas de diagnósticos y selecciona una (primero se le muestra las categorías y selecciona la categoría que le interesa, luego se le muestran los diagnósticos de la seleccionada y termina seleccionando el diagnóstico deseado. Debe poder recomenzar esta forma de selección para facilitarle la búsqueda). Elegida la representación estandarizada del diagnóstico, el médico puede complementar el diagnóstico con su propia descripción. Si corresponde, el médico indica para ese diagnóstico los tratamientos a seguir. Para ello indica el tipo de tratamiento, una descripción y según el tipo el resto de la información. Luego el Sistema da de alta los nuevos diagnósticos para la consulta.

Nombre	Obtener historial paciente
Actores	Usuario médico
Descripción	El caso de uso comienza cuando un Usuario Médico que ha iniciado una sesión desea obtener el historial de un Socio. Para esto el médico ingresa la ci correspondiente a un socio al que solicita su historia. Como resultado obtiene la ci, nombre, apellido y fecha de nacimiento del Socio, y para cada médico con el cual tuvo una consulta se muestran las fechas de las consultas y tipo de consulta (común o emergencia), el nombre y apellido del médico, la representación estandarizada y la descripción del diagnóstico así como la descripción de cada uno de los tratamientos asignados para cada diagnóstico.

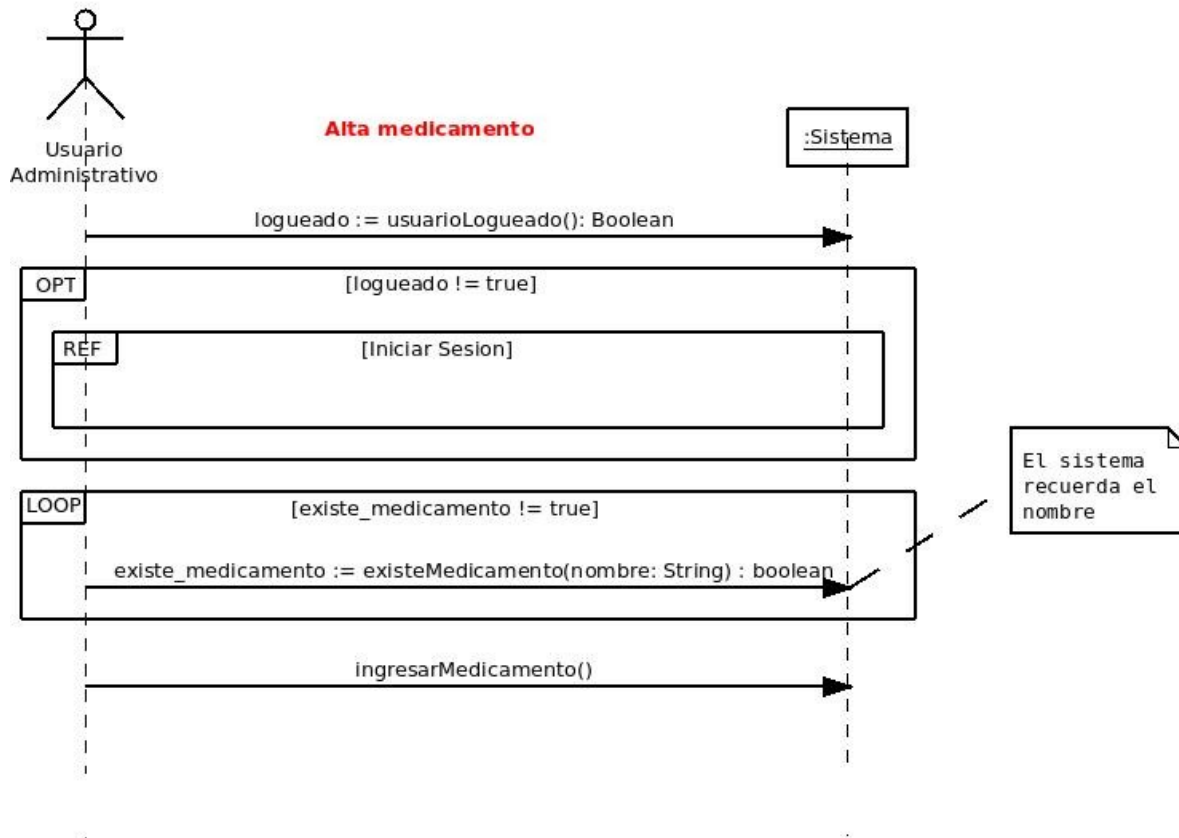
Nombre	Cantidad de consultas por categoría de diagnostico
Actores	Usuario médico
Descripción	El caso de uso comienza cuando un Usuario Administrativo que ha iniciado una sesión desea obtener la cantidad de consultas médicas por categoría de diagnóstico. Dadas 2 fechas (desde y hasta), el usuario espera la cantidad de consultas médicas realizadas entre esas 2 fechas (inclusivas) por cada categoría de diagnostico. Puede no especificarse una de las fechas o ninguna (en cuyos casos se asumen como “desde el comienzo de los tiempos” y/o el “fin de los tiempos”).

Nombre	Estado de situación de un socio
Actores	Usuario socio
Descripción	El caso de uso comienza cuando un Usuario Socio que ha iniciado una sesión desea obtener su situación para no quedar inactivo. Para esto el socio quiere saber la cantidad de reservas a las que no asistió a su consulta así como el detalle de las reservas activas (cuyas consultas aun no han ocurrido).

3 Comportamiento de Casos de Uso

3.1 Alta medicamento

3.1.1 Diagramas de Secuencia del Sistema



3.1.2 Contratos

Nombre	Existe medicamento
Operación	<code>existeMedicamento(nombre:String):boolean</code>
Entrada	nombre indica el nombre del medicamento que se desea dar de alta en el sistema
Salida	Un booleano que indica si ya existe en el sistema un medicamento cuyo nombre coincide con el nombre que se desea ingresar
Descripción	Esta operación verifica si ya existe un medicamento con el nombre dado
Excepciones	Si no se cumple pre1 se lanza la excepción <code>ExNoEstaLogueado</code> .

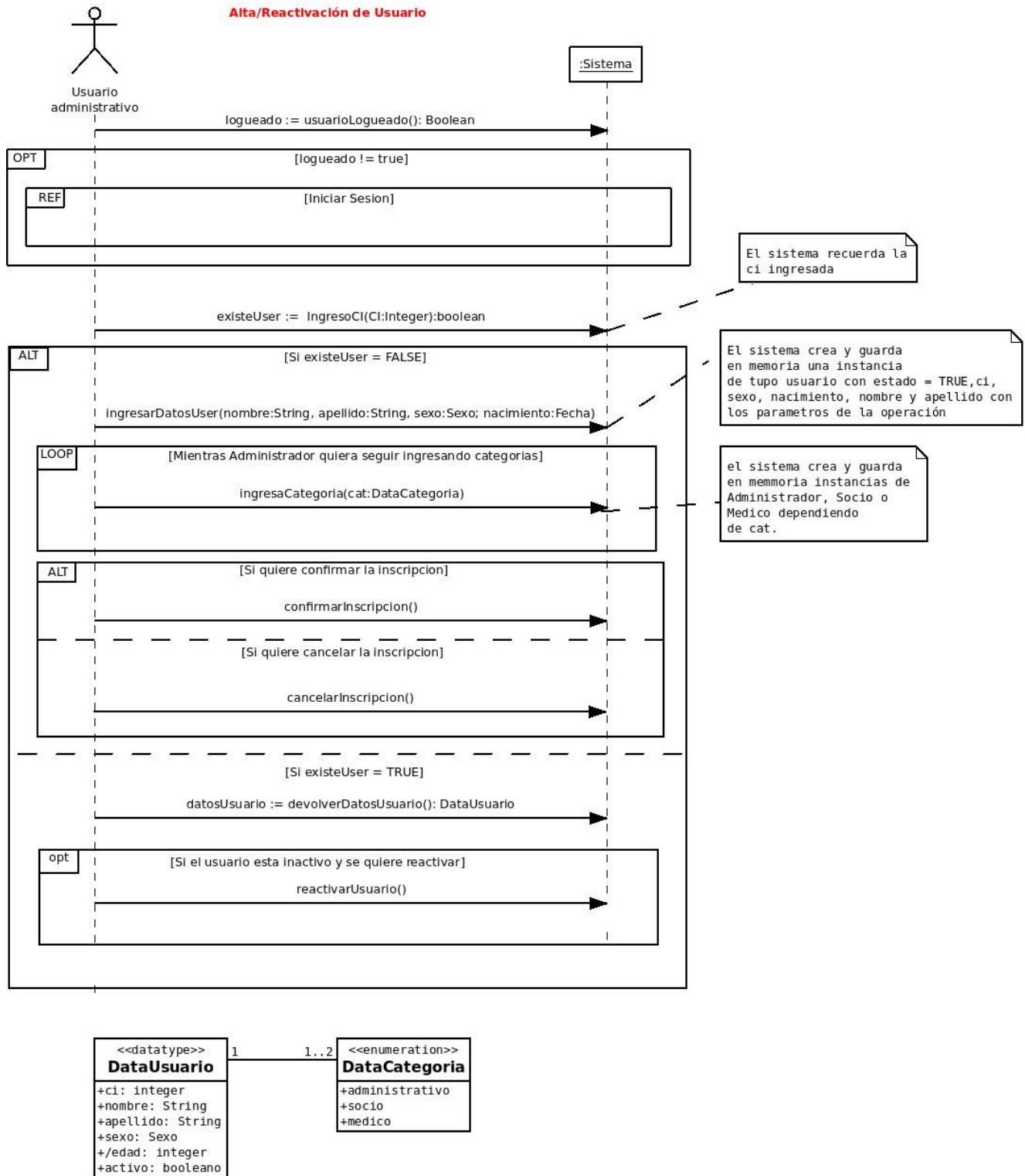
Precondiciones y Postcondiciones	
Pre1:	Existe en la memoria del sistema un usuario logueado de tipo administrativo.
Post1:	Existe en la memoria del sistema el nombre ingresado
Post2:	Si existe en el sistema una instancia de Medicamento con nombre = nombre se devuelve TRUE. En caso contrario se devuelve FALSE.

Nombre	Ingresar medicamento
Operación	ingresarMedicamento()
Entrada	-
Salida	-
Descripción	Esta operación ingresa un medicamento en el sistema con el nombre previamente ingresado.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoEstaLogeado. Si no se cumple pre2 se lanza la excepción NombreMedicamentoNoIngresado. Si no se cumple pre3 se lanza la excepción MedicamentoExistente.

Precondiciones y Postcondiciones	
Pre1:	Existe en la memoria del sistema un usuario logueado de tipo administrativo.
Pre2:	Existe en la memoria del sistema el nombre previamente ingresado.
Pre3:	No existe en el sistema una instancia de medicamento con nombre = nombre.
Post1:	Se crea en el sistema una instancia de Medicamento con nombre = nombre

3.2 Alta de usuario

3.2.1 Diagramas de Secuencia del Sistema



3.2.2 Contratos

Nombre	Ingreso CI
Operación	<code>ingresoCI(CI:integer):boolean</code>
Entrada	Un integer que representa a la cédula del usuario al que se quiere dar de alta o reactivar.
Salida	Retorna un booleano que indica si ya existe un usuario en el sistema con la misma CI ingresada.
Descripción	Dada la CI del usuario la operación valida si el mismo está ya registrado en el sistema.
Excepciones	Si no se cumple pre1 se lanza la excepción <code>ExNoEstaLogeado</code> .

Precondiciones y Postcondiciones	
Pre1: Existe en la memoria del sistema un usuario logueado de tipo administrativo.	
Post1: Existe en la memoria del sistema la CI ingresada.	
Post2: Si existe en el sistema una instancia de Usuario cuya CI = CI se retorna TRUE y FALSE en caso contrario.	

Nombre	Ingresar datos usuario
Operación	<code>IngresarDatosUsuario(nombre:String, apellido:String, sexo:Sexo, nacimiento:Fecha)</code>
Entrada	El string nombre indica el nombre del usuario a dar de alta en el sistema. El string apellido indica el nombre del usuario a dar de alta en el sistema. El dato enumerado sexo indica el sexo del usuario a dar de alta en el sistema. Nacimiento, de tipo Fecha, indica el la fecha de nacimiento del usuario a dar de alta en el sistema.
Salida	-
Descripción	Esta operación almacena en la memoria del sistema nombre, apellido, sexo y fecha de nacimiento de un usuario.
Excepciones	Si no se cumple pre1 se lanza la excepción <code>ExNoEstaLogeado</code> .

Precondiciones y Postcondiciones	
Pre1: Existe en la memoria del sistema un usuario logueado de tipo administrativo.	
Post1: Existe en la memoria del sistema el nombre, apellido, sexo y fecha de nacimiento del usuario a ingresar.	

Nombre	Ingresar categoria
Operación	ingresarCategoria(cat: String) ????
Entrada	cat indica el Rol del usuario.
Salida	-
Descripción	Esta operación almacena en la memoria del sistema el rol ingresada para el usuario que se quiere registrar.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoEstaLogeado.

Precondiciones y Postcondiciones	
Pre1: Existe en la memoria del sistema un usuario logueado de tipo administrativo.	
Post1: El rol se recuerda en la memoria del sistema.	

Nombre	Confirmar inscripción
Operación	confirmarInscripcion()
Entrada	-
Salida	-
Descripción	Esta operación da de alta a un usuario con los datos recordados en el sistema.
Excepciones	Si no se cumple la pre1 se lanza la excepción ExNoEstaLogeado. Si no se cumple la pre2 se lanza la excepción DatosFaltantes.

Precondiciones y Postcondiciones	
Pre1: Existe en la memoria del sistema un usuario logueado de tipo administrativo.	
Pre2: Existe en la memoria del sistema la CI, nombre, apellido, sexo, fecha de nacimiento y categorías previamente ingresadas.	
Post1: Existe en el sistema una instancia de Usuario con CI = CI, nombre = nombre, apellido = apellido, sexo = sexo, edad = edad, categorías = categorías y estado = inactivo.	

Nombre	Cancelar inscripción
Operación	cancelarInscripcion()
Entrada	-
Salida	-
Descripción	Esta operación cancela el caso de uso y borra de la memoria los datos almacenados.
Excepciones	Si no se cumple la pre1 se lanza la excepción ExNoEstaLogeado.

Precondiciones y Postcondiciones	
Pre1: Existe en la memoria del sistema un usuario logueado de tipo administrativo.	
Post1: Se elimina de la memoria del sistema las instancia de Usuario (CI, nombre, apellido, sexo, edad, categorías y estado) y se cancela el caso de uso.	

Nombre	Devolver datos usuario
Operación	DevolverDatosUsuario():DataUsuario
Entrada	-
Salida	Un DataUsuario conteniendo los datos del usuario.
Descripción	Esta operación devuelve nombre, apellido, sexo, edad, categorías y estado del usuario cuya cédula fue previamente ingresada.
Excepciones	Si no se cumple la pre1 se lanza la excepción ExNoEstaLogeado. Si no se cumple la pre2 se lanza la excepción DatosFaltantes. Si no se cumple la pre3 se lanza la excepción UsuarioInexistente.

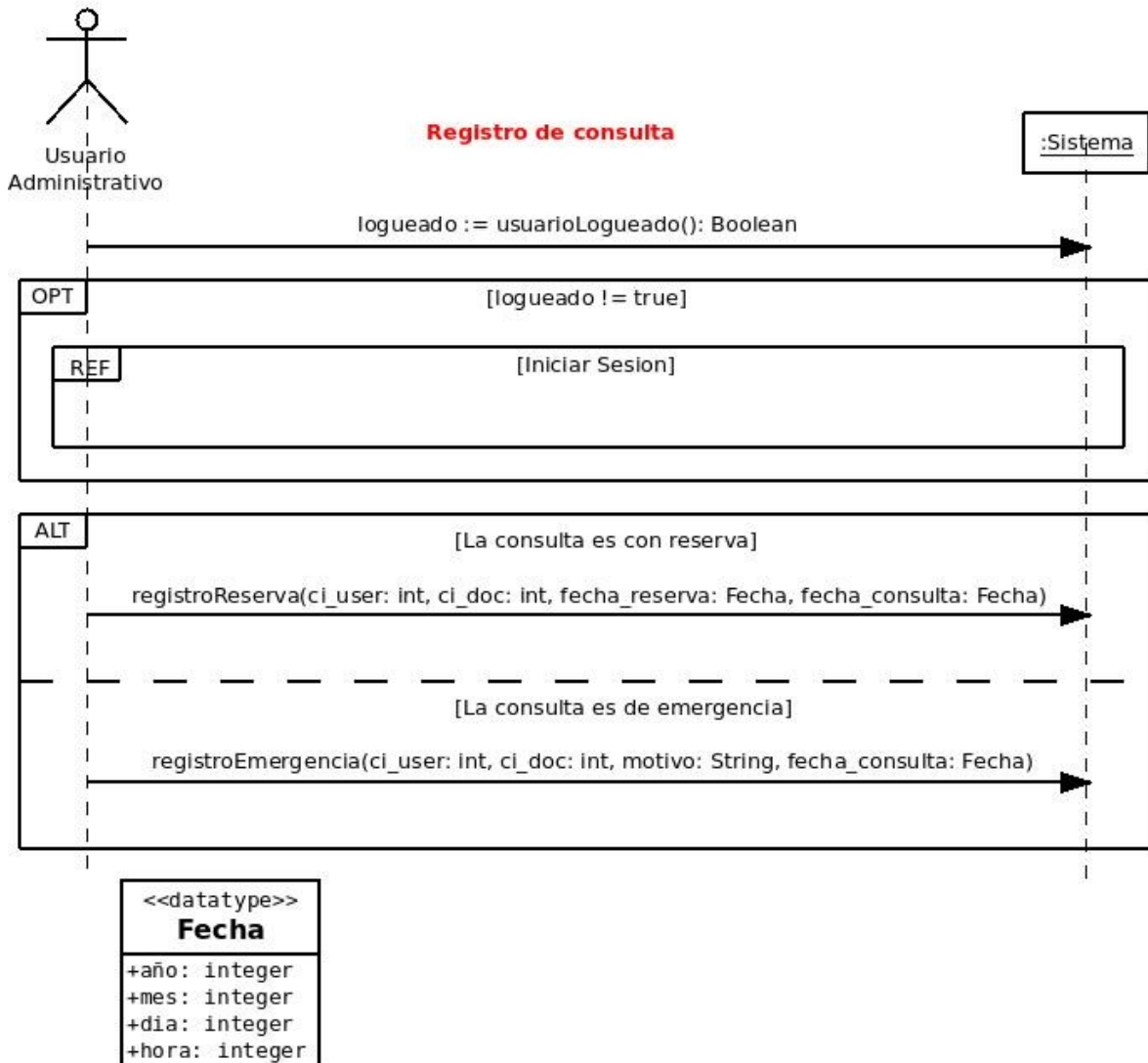
Precondiciones y Postcondiciones	
Pre1: Existe en la memoria del sistema un usuario logueado de tipo administrativo.	
Pre2: Existe en la memoria del sistema la CI previamente ingresada.	
Pre3: Existe en el sistema una instancia de usuario con CI = CI	
Post1: El sistema devuelve un DataUsuario con los datos nombre, apellido, sexo, edad, categorías y estado del usuario	

Nombre	Reactivar usuario
Operación	reactivarUsuario()
Entrada	-
Salida	-
Descripción	Esta operación reactiva el usuario cuya cédula fue previamente ingresada en el sistema
Excepciones	Si no se cumple la pre1 se lanza la excepción ExNoEstaLogeado. Si no se cumple la pre2 se lanza la excepción DatosFaltantes. Si no se cumple pre3 se lanza la excepción UsuarioInexistente. Si no se cumple pre4 se lanza la excepción UsuarioYaActivo.

Precondiciones y Postcondiciones
<p>Pre1: Existe en la memoria del sistema un usuario logueado de tipo administrativo.</p> <p>Pre2: Existe en la memoria del sistema la CI previamente ingresada.</p> <p>Pre3: Existe en el sistema una instancia de usuario con CI = CI.</p> <p>Pre4: El estado del usuario con CI = CI es inactivo.</p> <p>Post1: El estado del usuario cuya CI = CI es pasa a ser activo.</p>

3.3 Registro de consulta

3.3.1 Diagramas de Secuencia del Sistema



3.3.2 Contratos

Nombre	Usuario logueado
Operación	usuarioLogueado(): Boolean
Entrada	-
Salida	Un booleano que indica si en el sistema existe un usuario logueado actualmente.
Descripción	Indica si en el sistema existe un usuario logueado.
Excepciones	-

Precondiciones y Postcondiciones
Pre: -
Post: -

Nombre	Registro reserva
Operación	registroReserva(ci_user: int, ci_doc: int, fecha_reserva: Fecha, fecha_consulta: Fecha): void
Entrada	ci_user indica la cédula del usuario socio al cual se le va a registrar la reserva, la ci_doc indica la cédula del usuario médico el cual atiende dicha consulta, la fecha_reserva indica la fecha en la cual se reservó la consulta y fecha_consulta indica el día en cual se efectuó la consulta.
Salida	-
Descripción	Esta operación certifica que un usuario asistió a la consulta que había reservado.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoEstaLogueado. Si no se cumple pre2 se lanza la excepción NoExisteUsuario

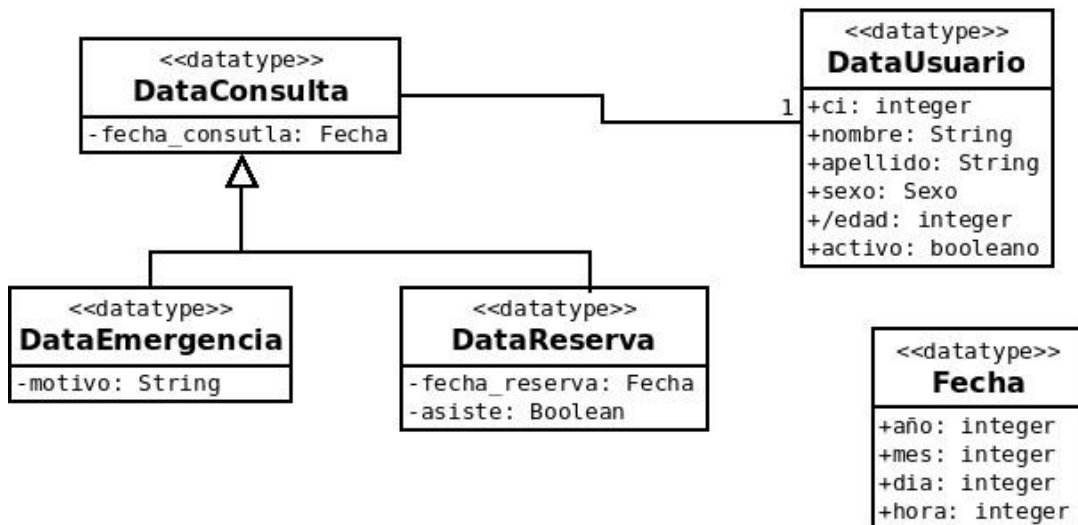
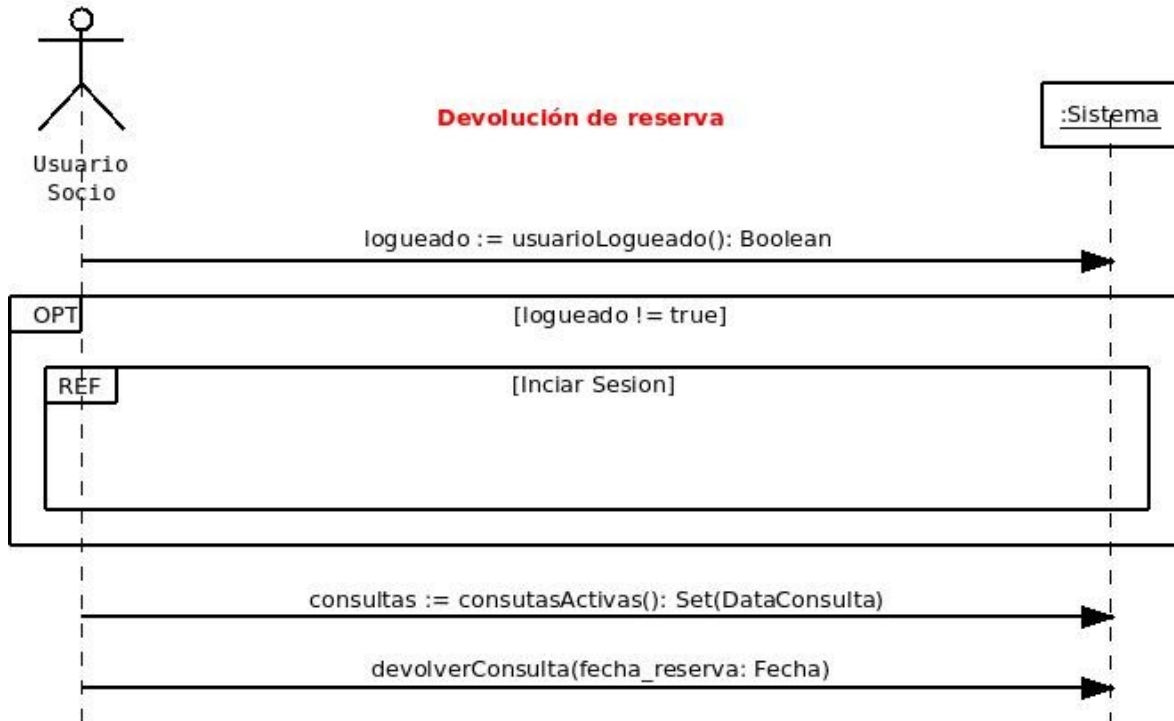
Precondiciones y Postcondiciones
Pre1: Existe en la memoria del sistema un usuario logueado de tipo administrativo.
Pre2: En el sistema deben existir una instancia de Usuario de tipo socio cuya ci == ci_user y otra instancia de Usuario de tipo médico cuya ci == ci_doc
Pre3: En el sistema debe existir una instancia de consulta con reserva con fechahora_reserva == fecha_reserva y fechahora_consulta == fecha_consulta.
Post: En el sistema se modifica el atributo asiste a TRUE de la instancia de consulta con reserva que tenga fechahora_reserva == fecha_reserva, fechahora_consulta == fecha_consulta, tenga un link a una instancia de Usuario de tipo socio con ci == ci_user y un link a una instancia de usuario de tipo médico con ci == ci_doc.

Nombre	Registro emergencia
Operación	registroEmergencia(ci_user: int, ci_doc: int, motivo: String, fecha_consulta: Fecha): void
Entrada	ci_user indica la cédula del usuario socio al cual se le va a registrar la reserva, la ci_doc indica la cédula del usuario médico el cual atiende dicha consulta, motivo indica el motivo por el cual se asiste a la emergencia y fecha_consulta indica el día en cual se efectuó la consulta.
Salida	-
Descripción	Esta operación ingresa una nueva consulta de emergencia con los datos ingresados como parámetro.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoEstaLogueado. Si no se cumple pre2 se lanza la excepción NoExisteUsuario

Precondiciones y Postcondiciones
<p>Pre1: Existe en la memoria del sistema un usuario logueado de tipo administrativo.</p> <p>Pre2: En el sistema deben existir una instancia de Usuario de tipo socio cuya ci == ci_user y otra instancia de Usuario de tipo médico cuya ci == ci_doc</p> <p>Post: En el sistema se crea una la instancia de consulta de emergencia que tenga fechahora_consulta == fecha_consulta, se crea un link entre la instancia de Usuario de tipo socio con ci == ci_user y un link a una instancia de usuario de tipo médico con ci == ci_doc con la consulta de emergencia antes creada.</p>

3.4 Devolución de reserva

3.4.1 Diagramas de Secuencia del Sistema



3.4.2 Contratos

Nombre	Consultas activas
Operación	consultasActivas(): Set(DataConsulta)
Entrada	-
Salida	Es un conjunto de DataConsultas que contienen la información de todas las consultas activas que hay en el sistema para el día actual.
Descripción	Esta operación devuelve todas las consultas que estén activas en el sistema en el día actual.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoEstaLogueado.

Precondiciones y Postcondiciones

Pre1: Existe en la memoria del sistema un usuario logueado de tipo socio.

Post1: En la memoria del sistema se crea un conjunto que contiene instancias de DataConsultas.

Nombre	Devolver consulta
Operación	devolverConsulta(fecha_reserva: Fecha)
Entrada	Fecha_reserva indica la fecha de la consulta que se desea devolver.
Salida	-
Descripción	Esta operación elimina del sistema una consulta ingresada por el usuario.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoEstaLogueado. Si ni se cumple pre2 se lanza la excepción NoExisteConsulta.

Precondiciones y Postcondiciones

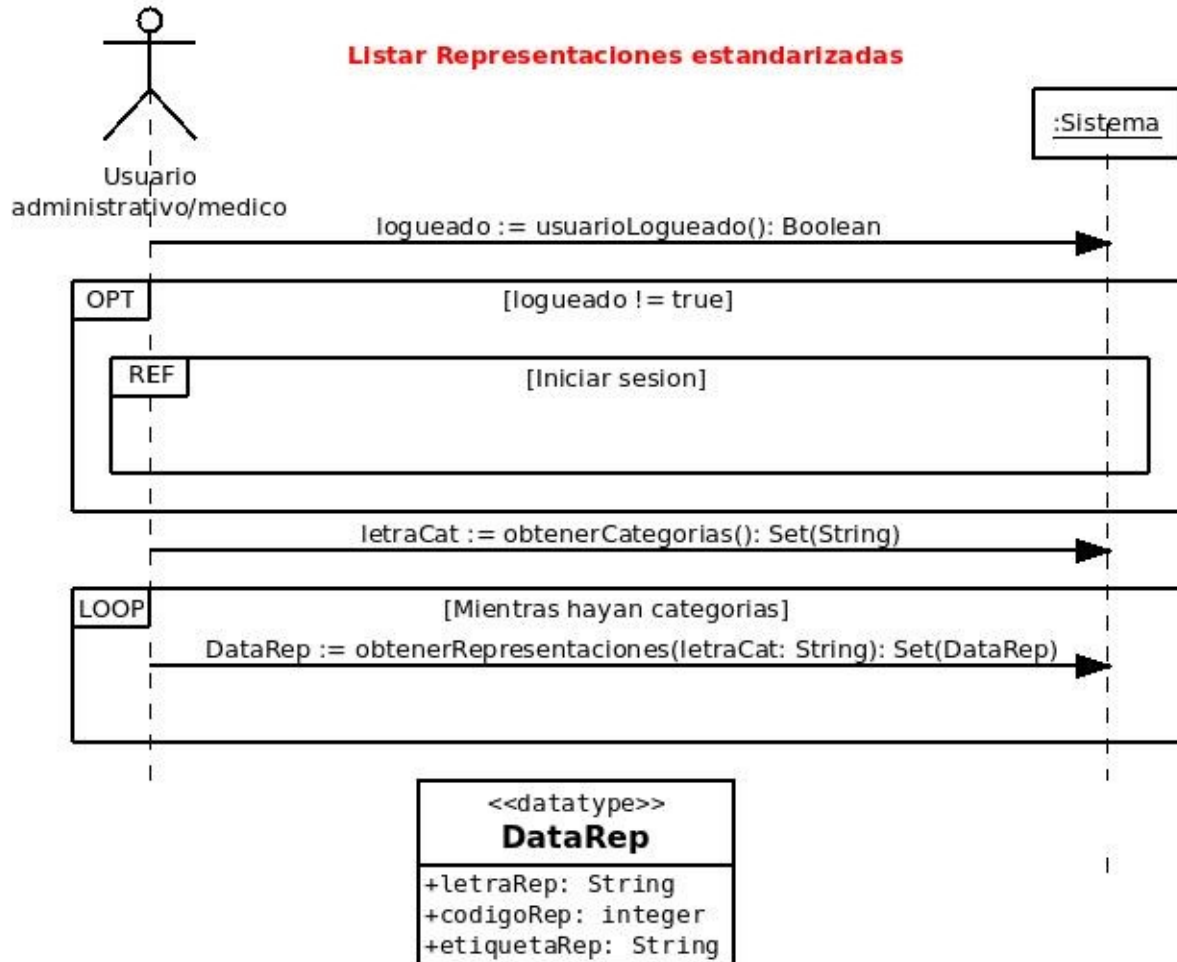
Pre1: Existe en la memoria del sistema un usuario logueado de tipo socio.

Pre2: En el sistema debe existir una instancia de Consulta con reserva cuyo atributo fecha_reserva == fecha_reserva y posea un link a una instancia de usuario con ci == ci del usuario logueado.

Post: En la memoria del sistema se elimina la instancia de consulta con reserva cuyo atributo fecha_reserva == fecha_reserva y posea un link a una instancia de usuario con ci == ci del usuario logueado.

3.5 Listar representaciones estandarizadas

3.5.1 Diagramas de Secuencia del Sistema



3.5.2 Contratos

Nombre	Obtener Categorías
Operación	ObtenerCategorias(): Set(String)
Entrada	-
Salida	Un conjunto de Strings donde cada una representa una de las categorías de representaciones ingresadas en el sistema
Descripción	Devuelve un conjunto de strings que indican las categorías de representaciones estandarizadas existentes en el sistema.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoEstaLogueado

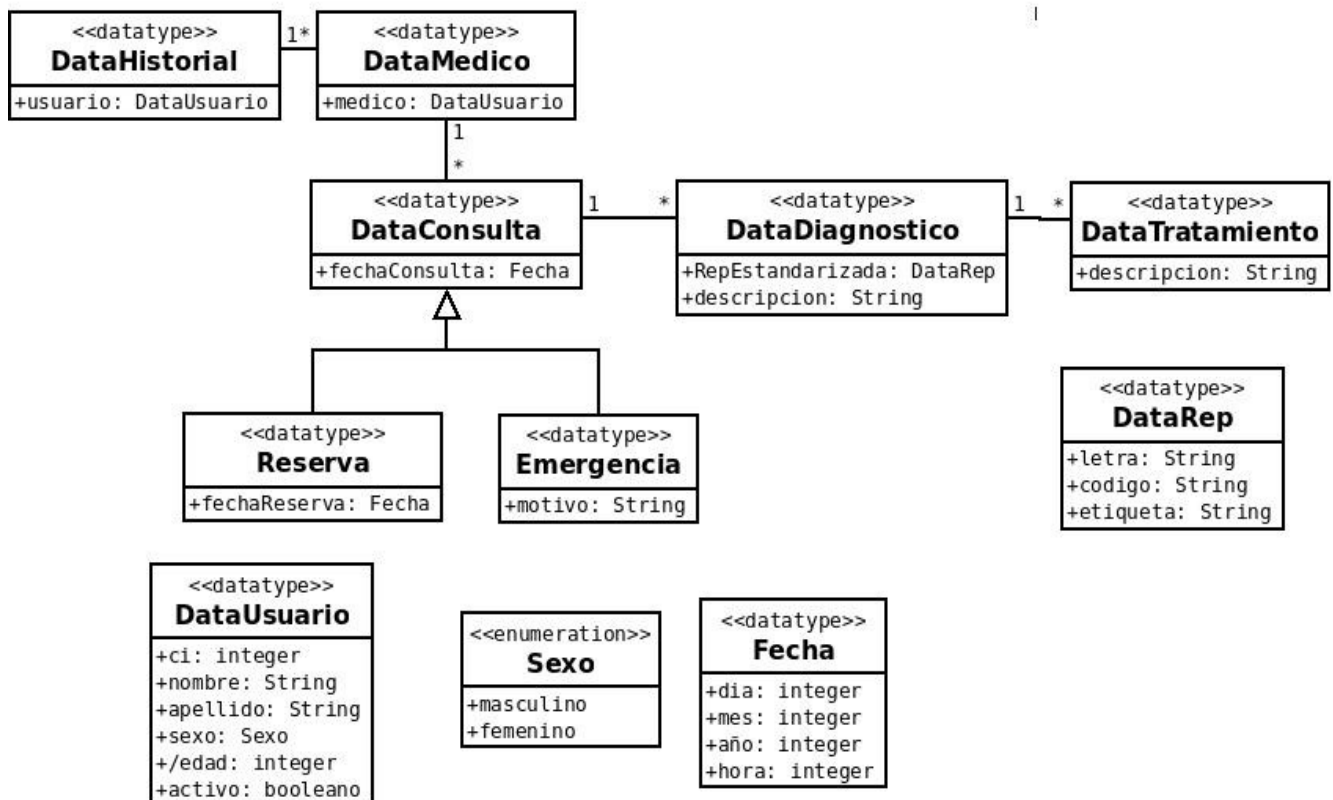
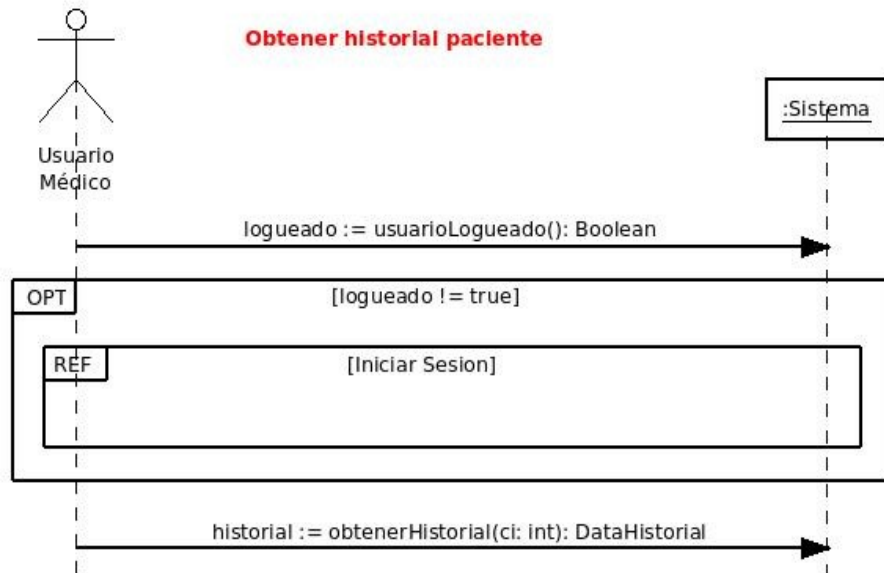
Precondiciones y Postcondiciones	
Pre1: Existe en la memoria del sistema un usuario logueado de tipo Administrativo o Médico.	
Post1: Se construyó y retornó un conjunto de Strings que representan cada una de las categorías de representaciones estandarizadas existentes en el sistema.	

Nombre	Obtener Representaciones
Operación	obtenerRepresentaciones(letraCat:String): Set(DataRep)
Entrada	letraCat representa la letra de una de las categorías de representaciones existente en el sistema.
Salida	Un conjunto de instancias de DataRep que representa cada una de las Representaciones Estandarizadas, pertenecientes a la categoría representada por letraCat, existentes en el sistema.
Descripción	Devuelve las Representaciones Estandarizadas existentes en el sistema, que pertenecen a una categoría en particular.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoEstaLogueado Si no se cumple pre2 se lanza la excepción ExNoExisteCategoria

Precondiciones y Postcondiciones
Pre1: Existe en la memoria del sistema un usuario logueado de tipo Administrativo o Médico.
Pre2: Existe en el sistema al menos una instancia de diagnóstico que tenga la letra del atributo codigo == letraCat.
Post1: Se construyó y retornó un conjunto de instancias de DataRep que representan cada una de las representaciones estandarizadas existentes en el sistema, con la letra del atributo codigo = letraCat.

3.6 Obtener historial paciente

3.6.1 Diagramas de Secuencia del Sistema



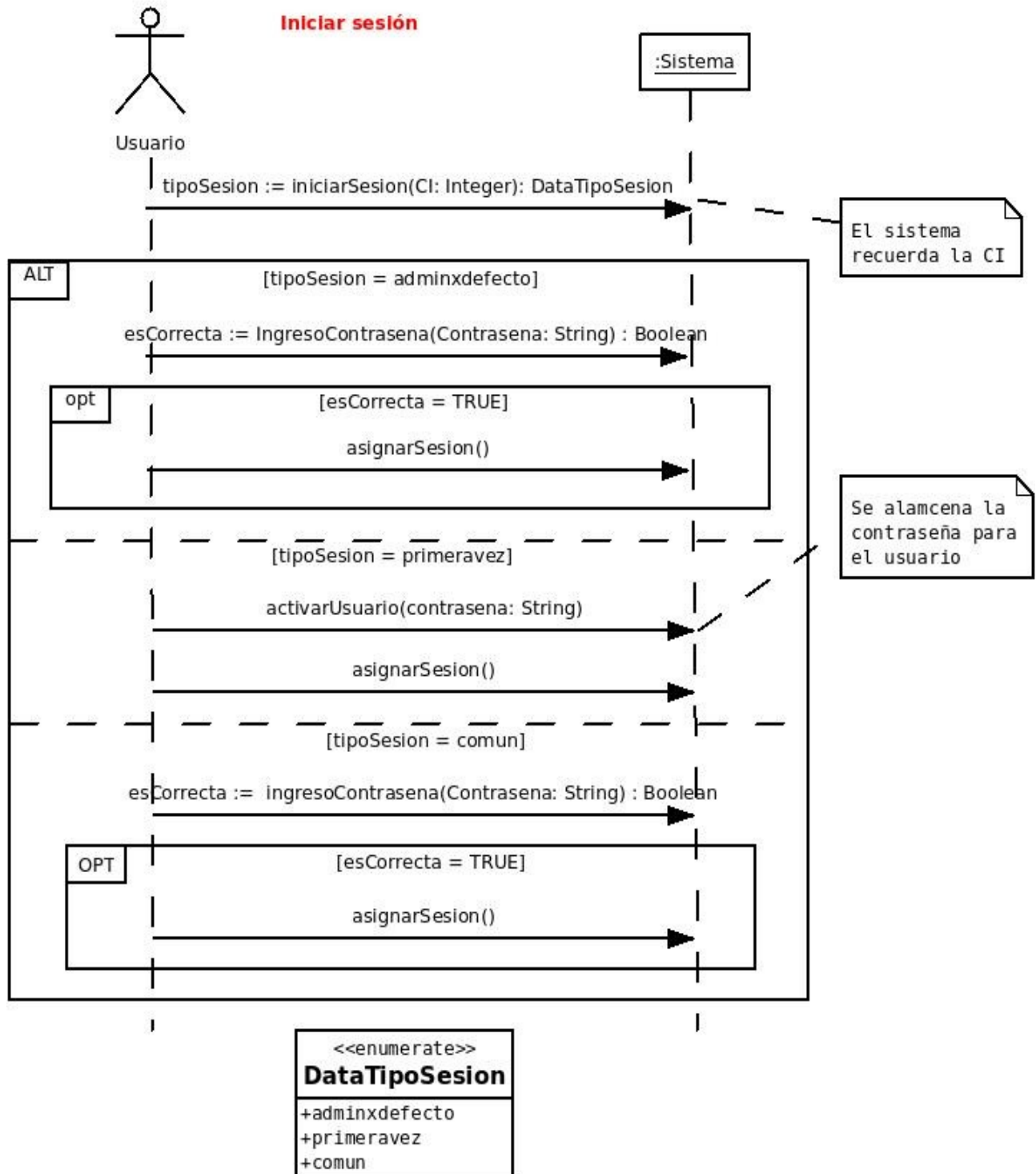
3.6.2 Contratos

Nombre	Obtener historial paciente
Operación	<code>obtenerHistorial(ci:Integer):DataHistorial</code>
Entrada	ci representa la cédula de identidad de un socio que está almacenado en la memoria del sistema.
Salida	Una instancia de DataHistorial que representa el historial médico del socio.
Descripción	Retorna la información del historial médico de un determinado socio para un determinado médico.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoEstaLogueado. Si no se cumple pre2 se lanza la excepción NoExisteUsuario

Precondiciones y Postcondiciones
<p>Pre1: Existe en la memoria del sistema un usuario logueado de tipo médico.</p> <p>Pre2: En el sistema debe existir una instancia de usuario cuya <code>ci == ci</code>.</p> <p>Post1: Se construyó y retornó una instancia de DataHistorial con la información del socio y un conjunto de instancias de DataMedicos con la información de un médico linkadas con un conjunto de instancias de DataConsulta que tiene la información de las consultas del socio con dicho médico; además la información de las consultas está linkada con un conjunto de instancias de DataDiagnostico con la información de los diagnósticos de dichas consultas. Ésta información está a su vez linkada con un conjunto de instancias de DataTratamientos con la descripción de los tratamientos.</p>

3.7 Iniciar sesión

3.7.1 Diagramas de Secuencia del Sistema



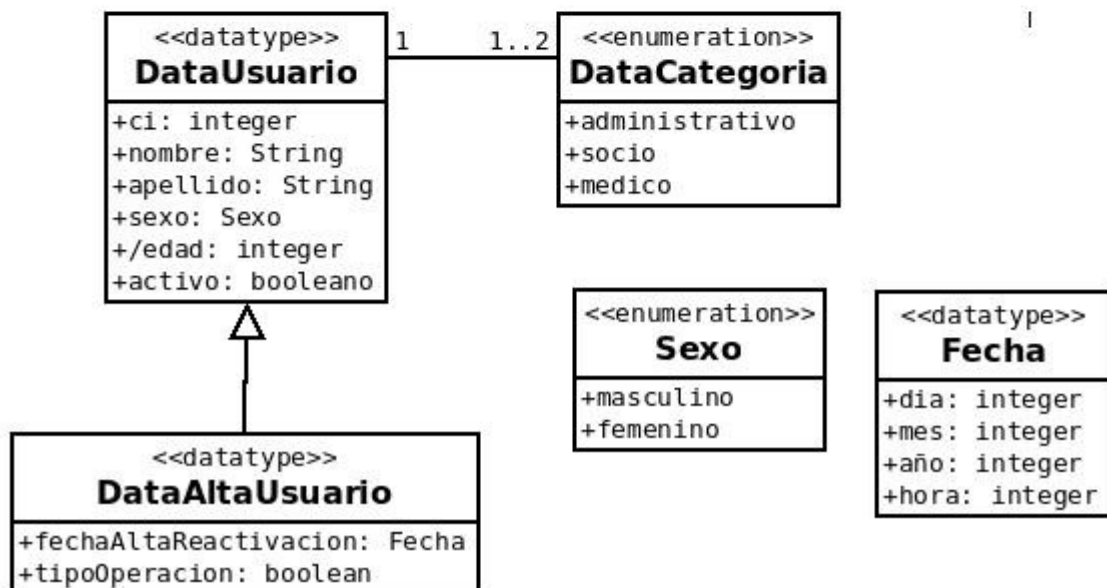
3.8 Cerrar sesión

3.8.1 Diagramas de Secuencia del Sistema



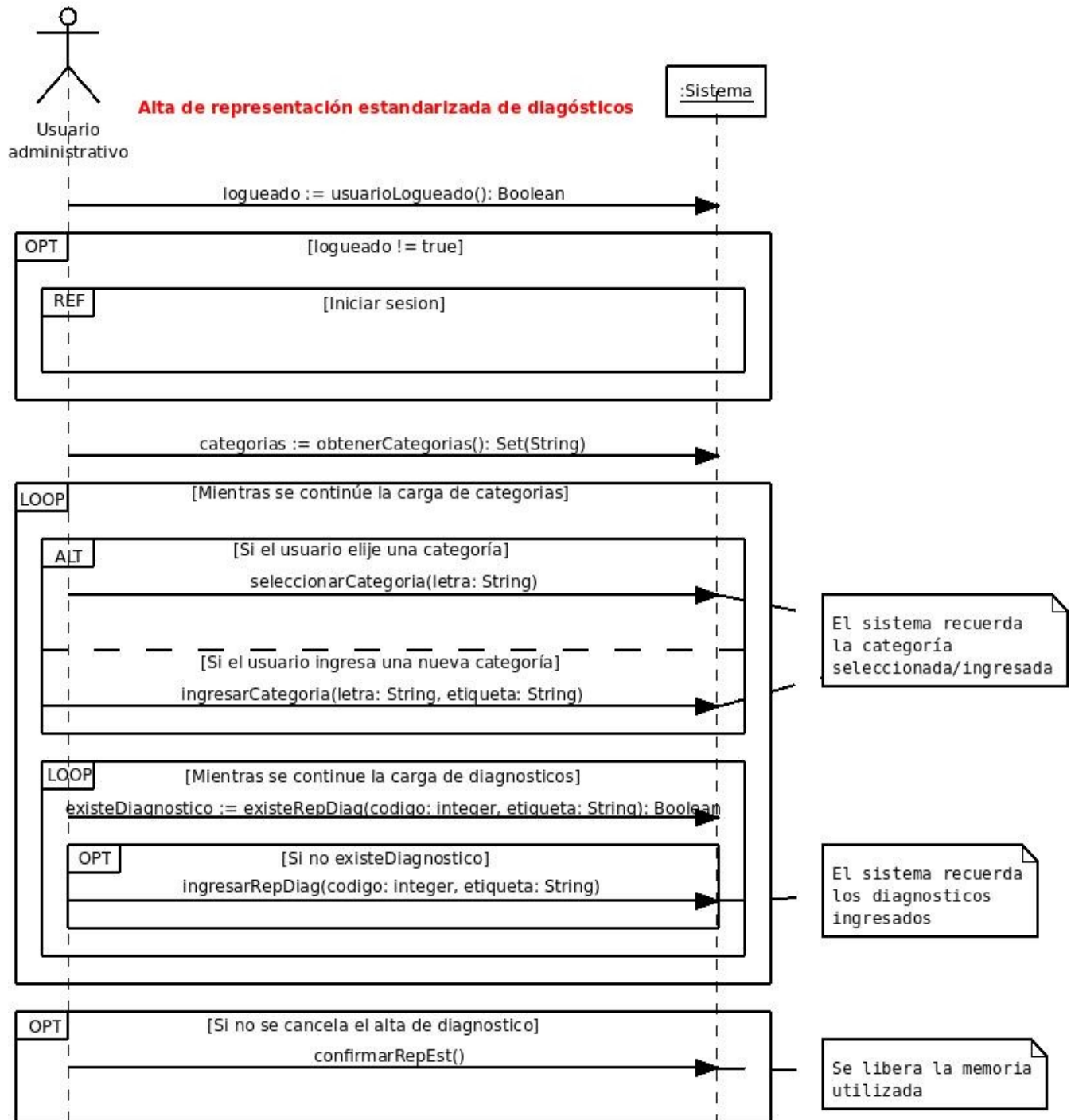
3.9 Usuarios dados de alta y reactivados

3.9.1 Diagramas de Secuencia del Sistema



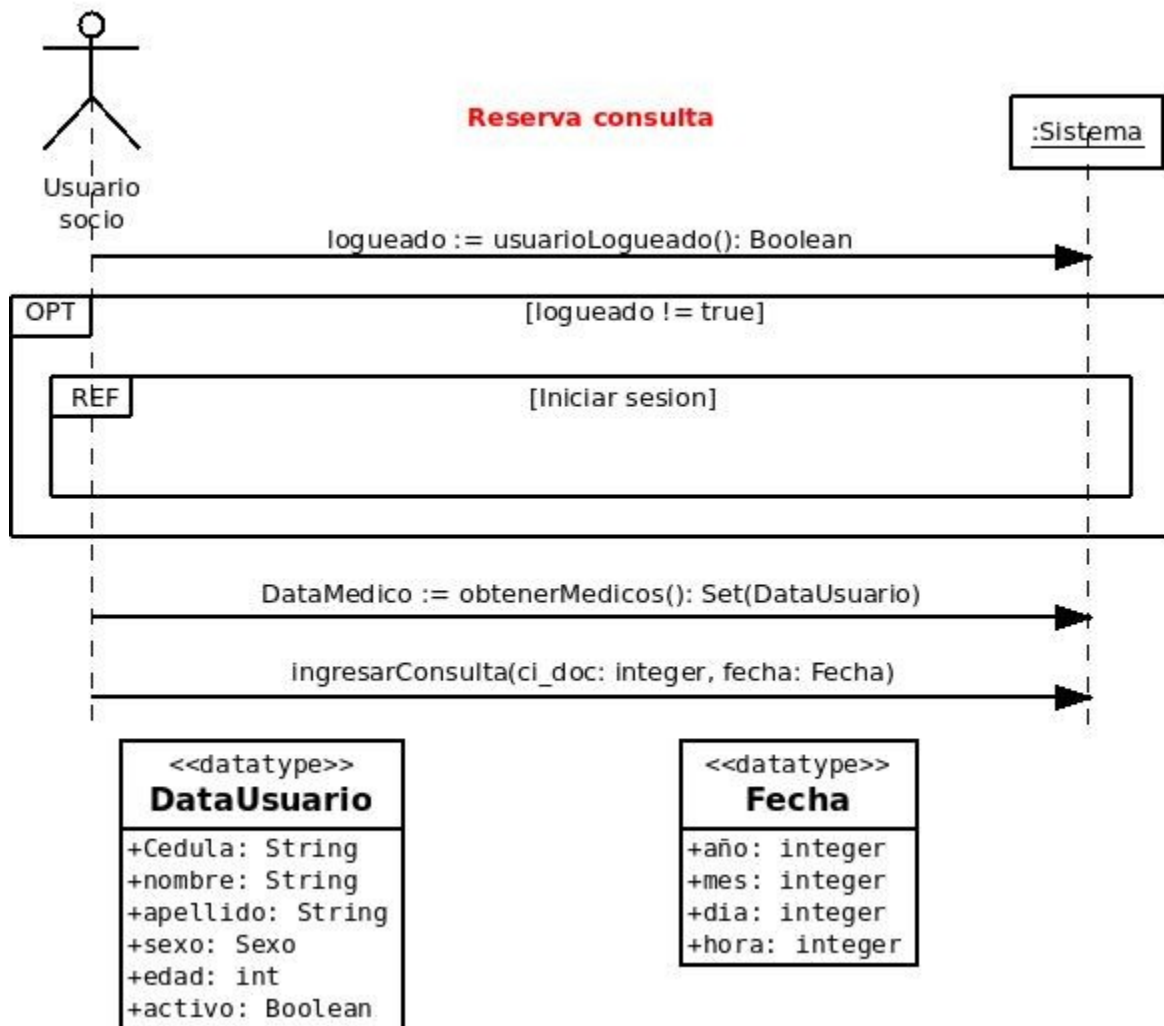
3.10 Alta representación estandarizada de diagnósticos

3.10.1 Diagramas de Secuencia del Sistema



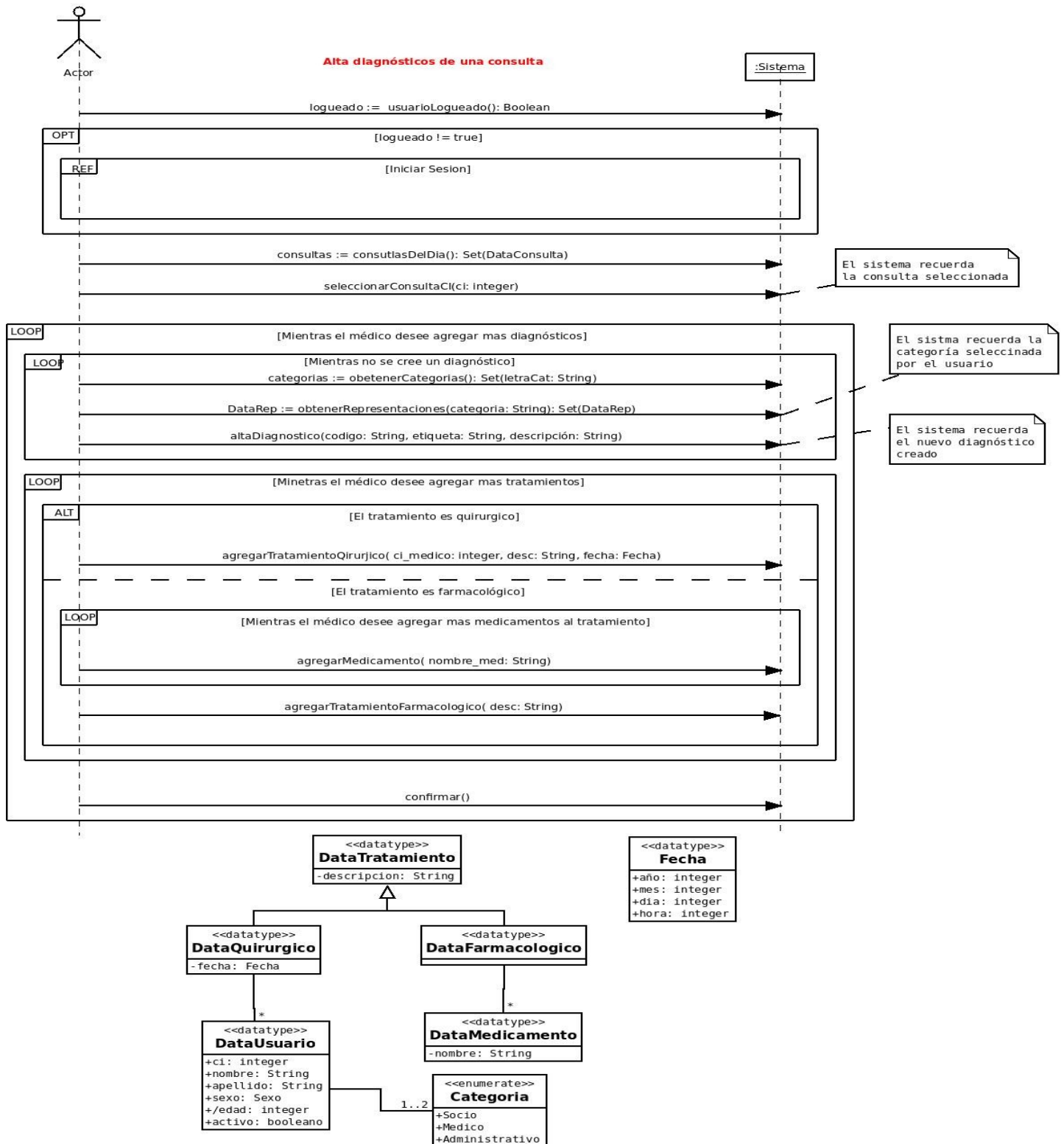
3.11 Reserva de consulta

3.11.1 Diagramas de Secuencia del Sistema



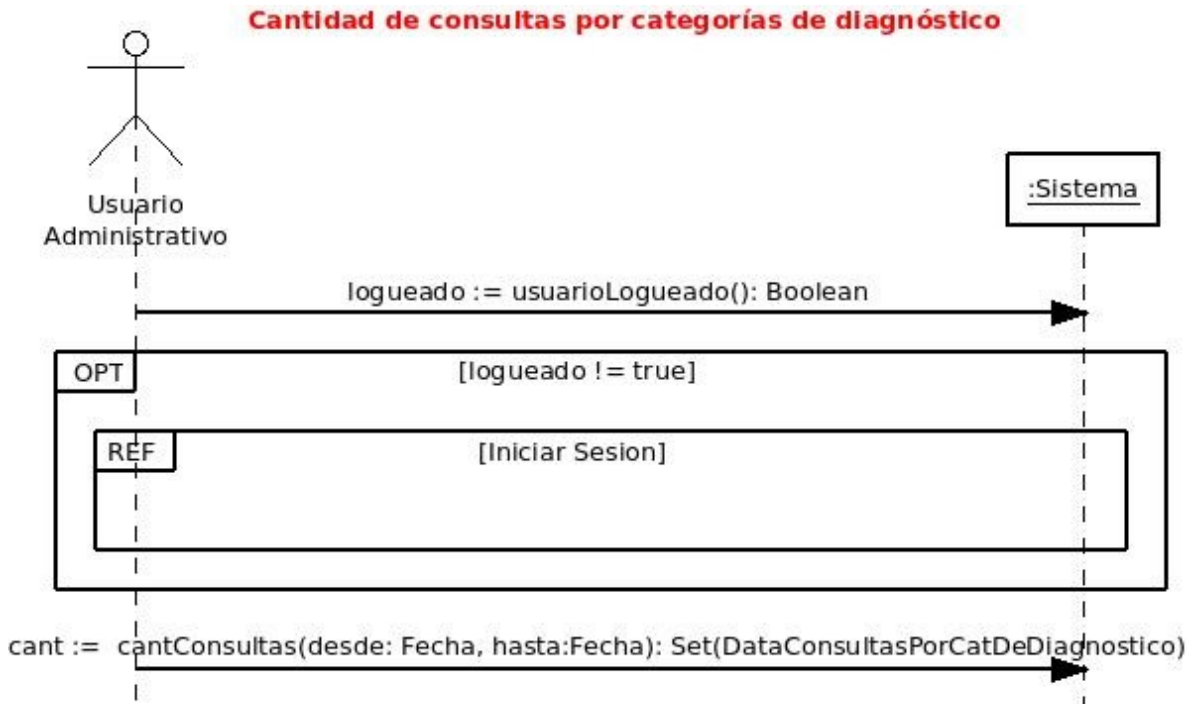
3.12 Alta diagnósticos de una consulta

3.12.1 Diagramas de Secuencia del Sistema



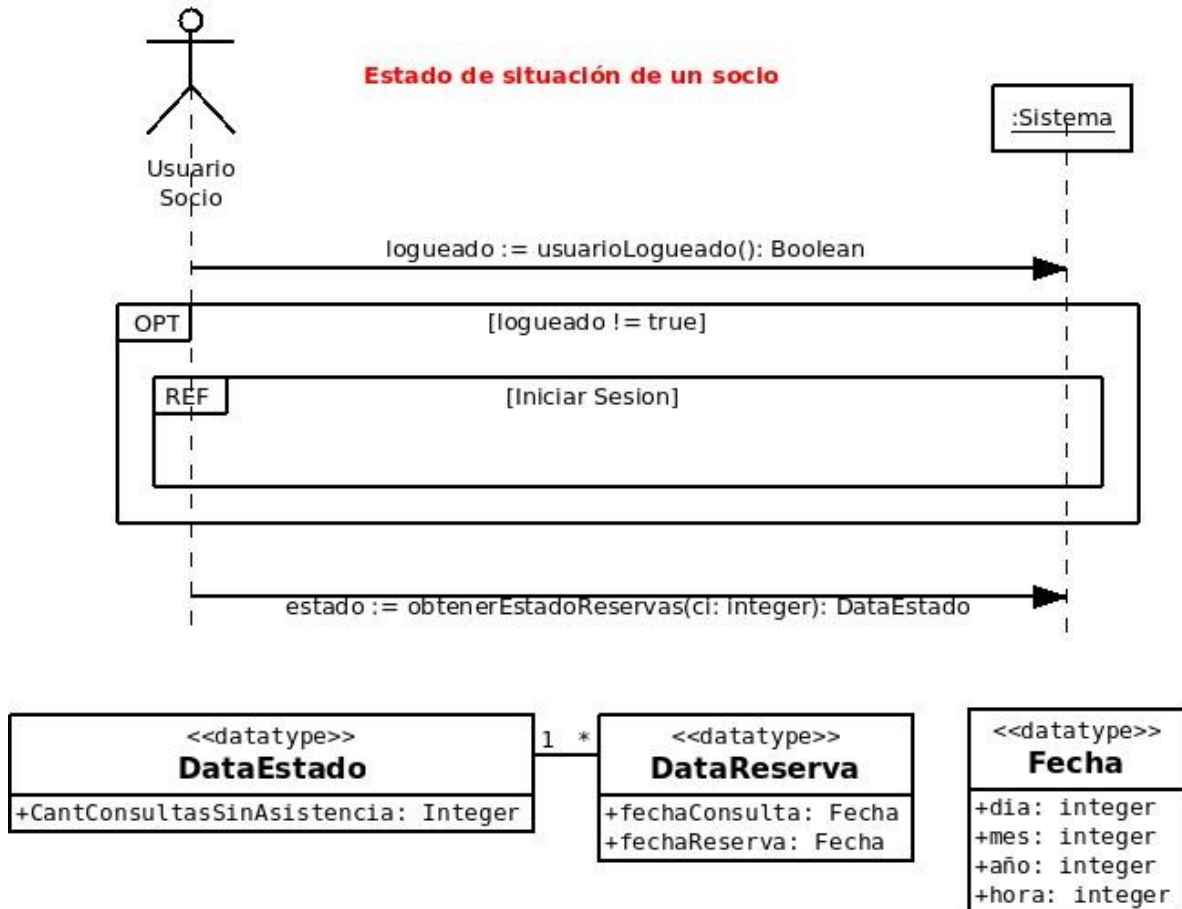
3.13 Cantidad de consultas por categoría de diagnóstico

3.13.1 Diagramas de Secuencia del Sistema



3.14 Estado de situación de un socio

3.14.1 Diagramas de Secuencia del Sistema



4 Anexo: Esqueletos

4.1 Alta usuario

```
// Operaciones del sistema
void ingresoCI(int CI);

bool existeUser(string CI);

void ingresarDatosUser(string nombre, string apellido, sexo
Sexo, nacimiento fecha);

void ingresaCategoria(DataCategoria cat);

void confirmarInscripcion();

void cancelarInscripcion();

DataUsuario devolverDatosUsuario(int CI);

void reactivarUsuario();

// Caso de uso
void altaUsuario(){

    if (!usuarioLogueado()){
        IniciarSesion();
    }

    existeUser := ingresoCI(CI);

    if (!existeUser){
        ingresarDatosUser(string nombre,
                           string apellido,
                           sexo Sexo,
                           nacimiento fecha);

        bool otraCategoria = true;
        while (otraCategoria){

            ingresaCategoria(cat);
            if (Si no desea ingresar otra categoría){
                otraCategoria = false;
            }
        }

        if(El usuario confirma el alta de usuario){
            confirmarInscripcion();
        }
        else if(El usuario cancela el alta de usuario){
            cancelarInscripcion();
        }
    }
}
```

```

    }
}

else{
    datosUsuario:= devolverDatosUsuario(CI);
    if(El usuario esta inactivo y se desea reactivar)
        reactivarUsuario()
}

}

```

4.2 Iniciar sesión

```

// Operaciones del sistema
DataTipoSesion iniciarSesion(int CI);
bool ingresoContrasena(String contrasena);
void asignarSesion();
void activarUsuario(String contrasena);

// Caso de uso
void IniciarSesion() {

    // Se pide la ci al usuario
    tipoSesion = iniciarSesion(ci);

    if (tipoSesion == adminxdefecto) {

        // Se pide la contraseña al usuario
        if(ingresoContrasena(contrasena)) {

            asignarSesion();

        }

    } else if (tipoSesion == primeravez) {

        // Se pide la contraseña al usuario
        activarUsuario(contrasena);

        asignarSesion();

    } else {

        // Se pide la contrasenia al usuario
        if (ingresoContrasena(contrasenia)) {

            asignarSesion();

        }

    }

}
}

```

4.3 Registro de consulta

```
// Operaciones del sistema
void registroReserva(ci_user:int,
                    ci_doc:int,
                    fecha_reserva:Fecha,
                    fecha_consulta:Fecha);

void registroEmergencia(ci_user:int,
                       ci_doc:int,
                       motivo:String,
                       fecha_consulta:Fecha);

// Caso de uso
void procedureRegistroConsulta() {

    if (!usuarioLogueado()) {
        IniciarSesion();
    }

    // Se pide tipoReserva al usuario
    if (tipoReserva == 'reserva') {

        registroReserva(ci_user, ci_doc, fecha_reserva, fecha_consul
ta);

    }
    else if (tipoReserva == 'emergencia') {

        registroEmergencia(ci_user, ci_doc, motivo, fecha_consulta);

    }

}
```