

Programación 4

Informe del Modelo de Casos de Uso

Grupo 46

Integrantes

Alesina, Ignacio Federico

CI: 5.160.423-1

Cabrera, Angel Joel

CI: 5.341.119-9

Marín, Matias

CI: 5.140.893-2

Munhos, Luciana

CI: 5.304.346-1

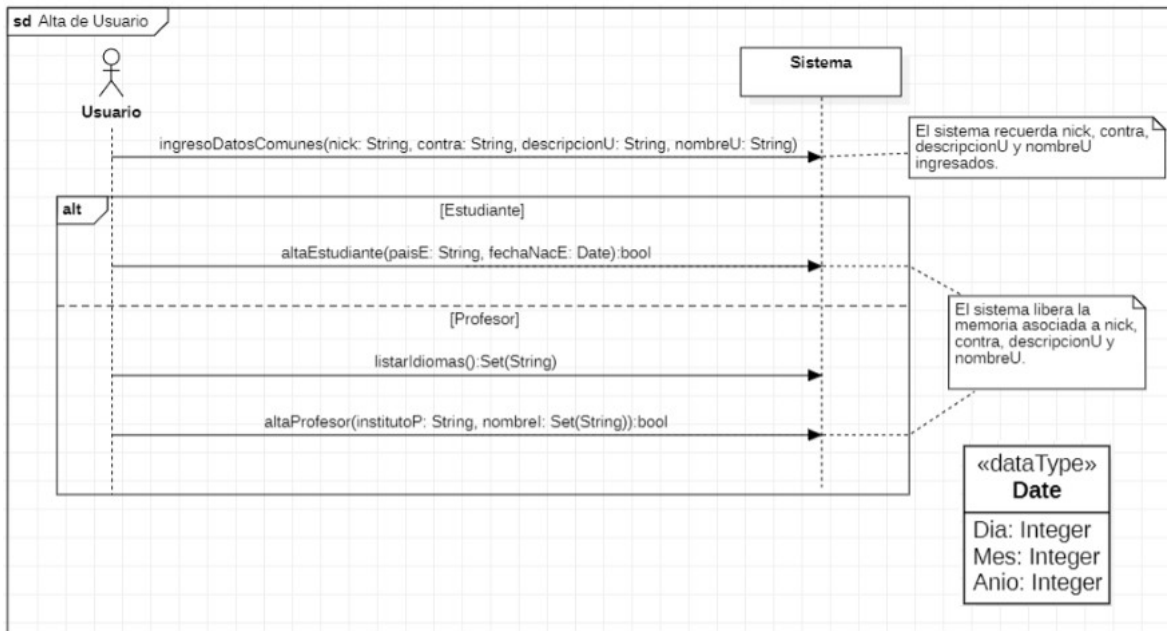
Roca, Manuel

CI: 5.258.563-6

Docente: Sebastián Pizard

Alta de usuario

Diagramas de Secuencia del Sistema



Contratos

Nombre	ingreso Datos Comunes
Operación	ingresoDatosComunes(nick: String, contra: String, descripcionU: String, nombreU: String)
Entrada	El parámetro <code>nick</code> es el string a colocar en el atributo Nickname del nuevo usuario a crear. El parámetro <code>contra</code> es el string a colocar en el atributo Contraseña del nuevo usuario a crear. El parámetro <code>descripcionU</code> es el String a colocar en el atributo Descripción del nuevo usuario a crear. El parámetro <code>nombreU</code> es el string a colocar en el atributo Nombre del nuevo usuario a crear
Salida	No corresponde.
Descripción	El procedimiento recibe parámetros, los cuales debe guardar en memoria.

Precondiciones y Postcondiciones
Pre1: El parámetro <code>contra</code> debe de tener al menos 6 caracteres.
Post1: El sistema recuerda los valores de los strings <code>nick</code> , <code>contra</code> , <code>descripcionU</code> y <code>nombreU</code> .

Nombre	alta Estudiante
Operación	<code>altaEstudiante(paise: String, fechaNacE: Date):bool</code>
Entrada	El parámetro <code>paise</code> es el string a colocar en el atributo <code>Pais</code> de la nueva instancia de <code>Estudiante</code> a crear. El parámetro <code>fechaNacE</code> es el string a colocar en el atributo <code>FechaNacimiento</code> de la nueva instancia de <code>Estudiante</code> a crear.
Salida	Retorna <code>true</code> si la operación se pudo realizar de forma correcta. Retorna <code>false</code> si existe en el sistema una instancia de <code>Usuario</code> cuyo atributo <code>Nickname</code> coincida con el valor de <code>nick</code> guardado en el sistema.
Descripción	El procedimiento genera una nueva instancia de <code>Estudiante</code> en el sistema, en el caso de que el <code>Nickname</code> a emplear en dicha nueva instancia coincida con el atributo <code>Nickname</code> de alguna otra instancia de <code>Usuario</code> en el sistema, se muestra un error. Luego libera la memoria de los valores recordados previamente.

Precondiciones y Postcondiciones	
<p>Pre1: El sistema recuerda <code>nick</code>, <code>contra</code>, <code>descripcionU</code> y <code>nombreU</code>.</p> <p>Post1: Si ya existía una instancia de <code>Usuario</code> cuyo atributo <code>Nickname</code> = <code>nick</code>, se indica un error. En caso contrario, existe una nueva instancia de <code>Estudiante</code> con los siguientes atributos:</p> <ul style="list-style-type: none"> • <code>Nickname</code> = <code>nick</code>, • <code>Contraseña</code> = <code>contra</code>, • <code>Descripcion</code> = <code>descripcionU</code>, • <code>Nombre</code> = <code>nombreU</code>, • <code>Pais</code> = <code>paise</code> • <code>FechaNacimiento</code> = <code>fechaNacE</code>. <p>Post2: Se libera la memoria asociada a <code>nick</code>, <code>contra</code>, <code>descripcionU</code> y <code>nombreU</code>.</p>	

Nombre	Listar Idiomas
Operación	<code>listarIdiomas():Set(String)</code>
Entrada	No corresponde
Salida	Retorna el conjunto formado por el atributo <code>Nombre</code> de todas las instancias de <code>Idioma</code> del Sistema
Descripción	Muestra el atributo <code>Nombre</code> de todas las instancias de <code>Idioma</code> del Sistema

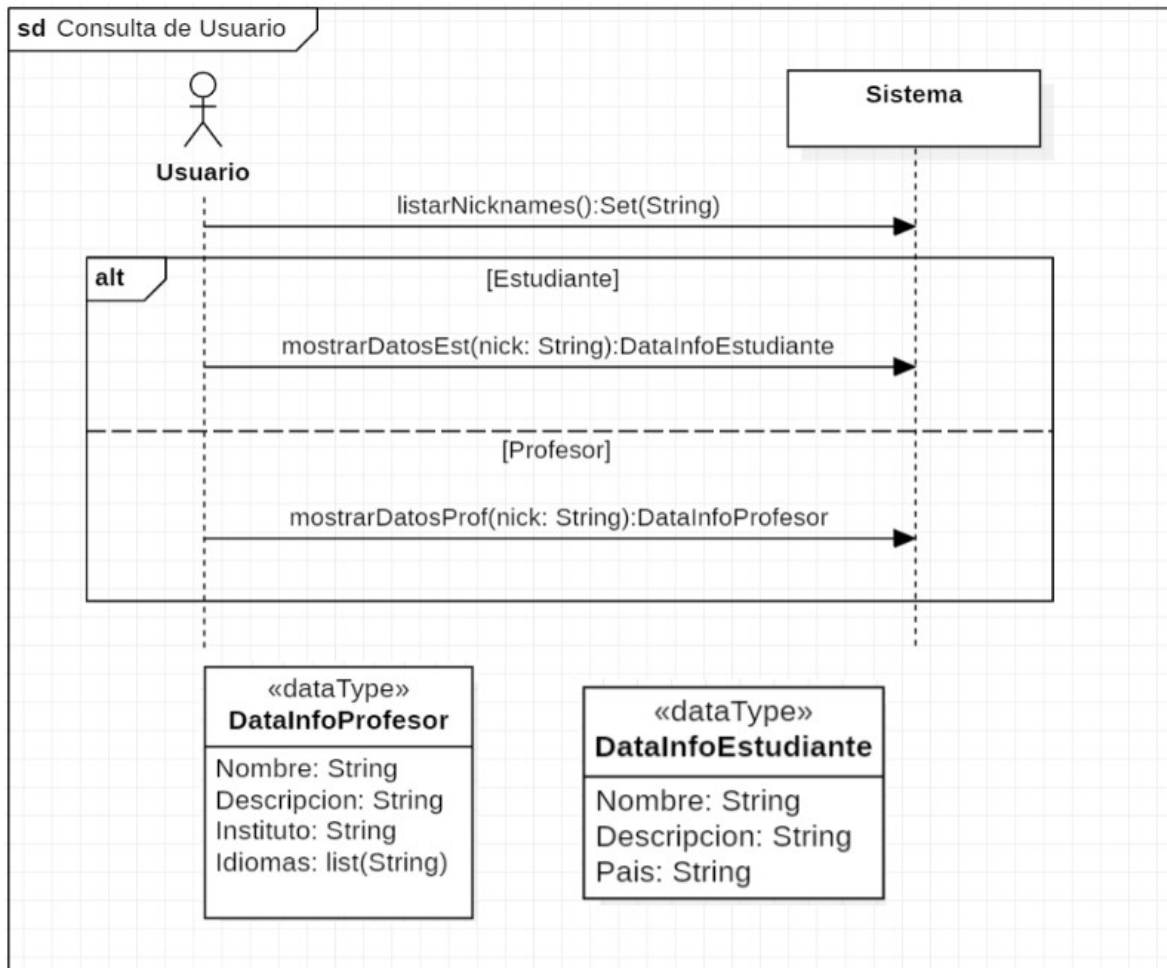
Nombre	Alta de Profesor
Operación	<code>altaProfesor(institutoP:String,nombreI:Set(String)) :bool</code>
Entrada	El parámetro <code>institutoP</code> es el string a colocar en el atributo <code>Instituto</code> de la

	nueva instancia de <code>Profesor</code> a crear. El parámetro <code>nombreI</code> es el conjunto de strings que equivalen a los idiomas que el profesor al crear su nueva instancia está apto para enseñar.
Salida	Retorna <code>true</code> si la operación se pudo realizar de forma correcta. Retorna <code>false</code> si existe en el sistema una instancia de <code>Usuario</code> cuyo atributo <code>Nickname</code> coincida con el valor de <code>nick</code> guardado en el sistema.
Descripción	El procedimiento genera una nueva instancia de <code>Profesor</code> en el sistema, en el caso de que el <code>Nickname</code> a emplear en dicha nueva instancia coincida con el atributo <code>Nickname</code> de alguna otra instancia de <code>Usuario</code> en el sistema, se muestra un error. Luego libera la memoria de los valores recordados previamente

Precondiciones y Postcondiciones	
<p>Pre1: El sistema recuerda <code>nick</code>, <code>contra</code>, <code>descripcionU</code> y <code>nombreU</code>.</p> <p>Pre2: El parámetro <code>nombreI</code> debe estar incluido en el conjunto retornado por la operación <code>listarIdiomas()</code>.</p> <p>Post1: En el caso que ya existiese una instancia de <code>Usuario</code> cuyo atributo <code>Nickname</code> tiene el valor <code>nick</code>, se indica el error. En caso contrario, entonces existe una nueva instancia de <code>Profesor</code> con atributos:</p> <ul style="list-style-type: none"> • <code>Nickname = nick</code>, • <code>Contraseña = contra</code> • <code>Descripción = descripcionU</code> • <code>Nombre = nombreU</code>, • <code>Instituto = institutoP</code>. <p>Además por cada string <code>a</code> perteneciente a <code>idiomas</code>, existe un link entre la nueva instancia de <code>Profesor</code> creada y la instancia de <code>nombreI</code> cuyo atributo <code>Nombre</code> coincide con dicho string <code>a</code>.</p> <p>Post2: Se libera la memoria asociada a <code>nick</code>, <code>contra</code>, <code>descripcionU</code> y <code>nombreU</code>.</p>	

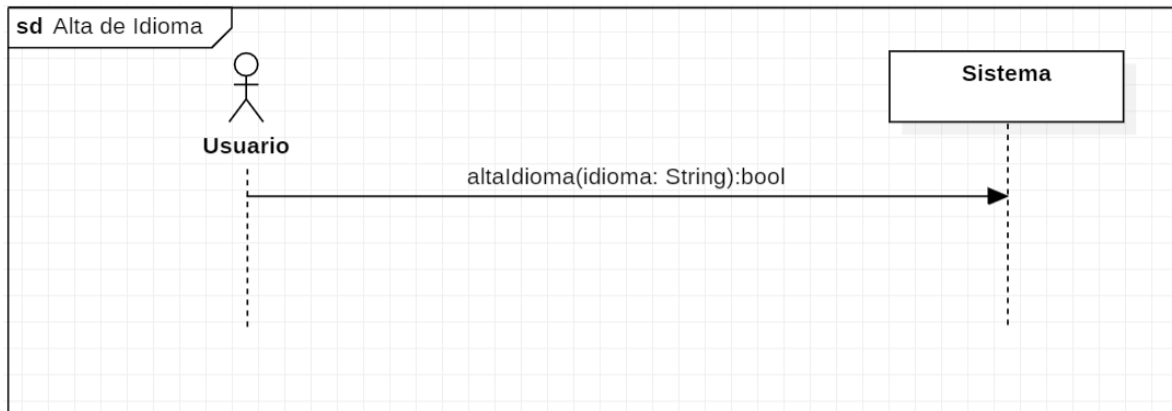
Consulta de usuario

Diagramas de Secuencia del Sistema



Alta de idioma

Diagramas de Secuencia del Sistema



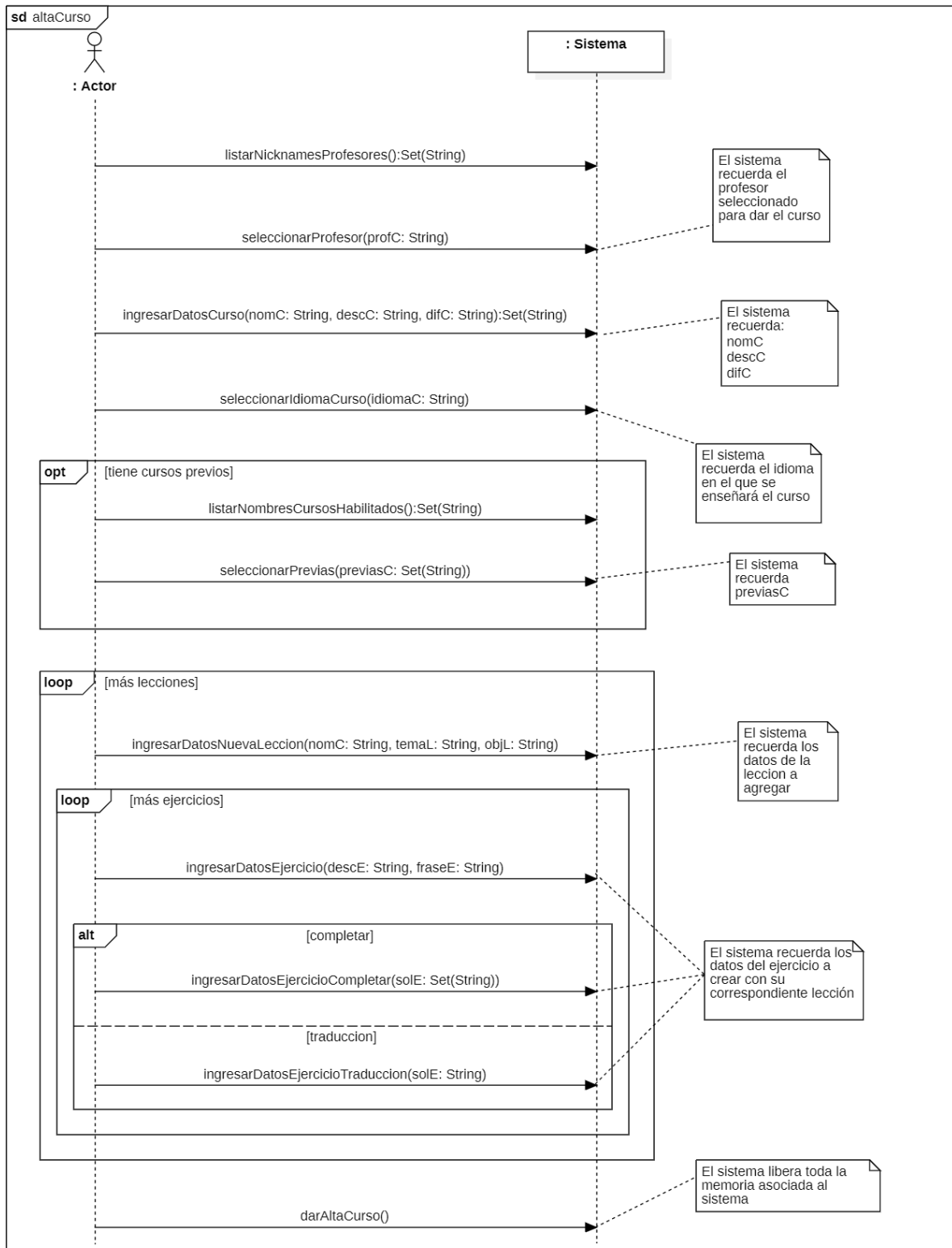
Consultar idiomas

Diagramas de Secuencia del Sistema



Alta de curso

Diagramas de Secuencia del Sistema



Contratos

Nombre	Listar Nicknames de profesores
Operación	<code>listarNicknamesProfesores() : Set(String)</code>
Entrada	Ninguno.
Salida	Set de String que contiene los atributos <code>Nickname</code> de todas las instancias de <code>Profesor</code> que existen en el sistema.
Descripción	Lista todos los Nicknames de los profesores existentes.

Precondiciones y Postcondiciones
Post1: Retorna un Set de String donde cada elemento es el atributo <code>Nickname</code> de una instancia de <code>Profesor</code> .

Nombre	Seleccionar profesor
Operación	<code>seleccionarProfesor(profC : String)</code>
Entrada	<code>profC</code> : un String que representa al <code>Nickname</code> de un profesor.
Salida	No corresponde.
Descripción	Asigna un profesor al curso a ser creado.

Precondiciones y Postcondiciones
Pre1: Existe en el sistema una instancia de <code>Profesor</code> con el atributo <code>Nickname = profC</code> .
Post1: El sistema recuerda la instancia de <code>Profesor</code> con atributo <code>Nickname = profC</code> .

Nombre	Ingresar datos del Curso
Operación	<code>ingresarDatosCurso(nomC : String, descC : String, difC : nivelDif) : Set(String)</code>
Entrada	<ul style="list-style-type: none">● <code>nomC</code>: un String que representa el nombre del curso a agregar.● <code>descC</code>: un String que representa la descripción del curso a agregar.● <code>difC</code>: un enumerado de tipo <code>nivelDif</code> que representa el nivel de dificultad del curso a agregar.
Salida	Set de String que contiene los nombres de todos los idiomas que sabe el profesor elegido que dictará el curso a agregar.

Descripción	Ingresa los datos correspondientes al curso que se quiere crear.
--------------------	--

Precondiciones y Postcondiciones
<p>Pre1: El sistema recuerda una instancia de Profesor .</p> <p>Pre2: No existe una instancia de Curso con atributo Nombre = nomC .</p> <p>Post1: El sistema recuerda nomC, descC y difC.</p> <p>Post2: Se retorna un Set de String que contiene los atributos Nombre de todas las instancias de Idioma asociadas a la instancia Profesor recordada por el sistema.</p>

Nombre	Seleccionar idioma de curso nuevo
Operación	seleccionarIdiomaCurso(idiomaC : String)
Entrada	IdiomaC: String que representa el nombre del idioma que se enseñará en el curso.
Salida	No corresponde.
Descripción	Enviar al sistema el idioma elegido por el usuario en el que se dictará el curso a agregar.

Precondiciones y Postcondiciones
<p>Pre1: El sistema recuerda una instancia de Profesor.</p> <p>Pre2: Existe una instancia de Idioma con atributo Nombre = idiomaC con un link se especializa a la instancia Profesor recordada por el sistema.</p> <p>Post1: El sistema recuerda la instancia de Idioma con atributo Nombre = idiomaC.</p>

Nombre	Listar nombres de cursos habilitados
Operación	listarNombresCursosHabilitados() : Set(String)
Entrada	No corresponde.
Salida	Un Set de String que contiene los nombres de todos los cursos habilitados.
Descripción	Muestra todos los nombres de los cursos habilitados en caso de que se quieran agregar cursos previos al curso a agregar.

Precondiciones y Postcondiciones
Post1: Se retornó un Set de String donde cada elemento que pertenece a la misma representa un

atributo `Nombre` de todas las instancias de `Curso` que están habilitados.

Nombre	Seleccionar previas de curso nuevo
Operación	<code>seleccionarPrevias (previasC : Set (String))</code>
Entrada	<code>previasC</code> : Set de String que representan los nombres de los cursos que se desean que sean previas del curso a agregar.
Salida	Un Set de String que contiene los nombres de todos los cursos habilitados.
Descripción	Agregar previas al curso nuevo.

Precondiciones y Postcondiciones

Pre1: Para todo String `e` del Set de String `previasC`, existe una instancia de `Curso` tal que su atributo `Nombre` = `e`.

Pre2: No existe String en la lista `previasC` tal que sea igual a `nomC` recordado por el sistema.

Post1: El sistema recuerda `previasC`.

Nombre	Ingresar datos de nueva lección
Operación	<code>ingresarDatosNuevaLeccion (nomC : String, temaL : String, objL : String)</code>
Entrada	<ul style="list-style-type: none">● <code>nomC</code>: nombre del curso al que se le quiere agregar una lección.● <code>temaL</code>: tema del que tratará la lección a agregarse.● <code>objL</code>: objetivo de la lección a agregarse.
Salida	No corresponde.
Descripción	Ingresa los datos de la lección a crear.

Precondiciones y Postcondiciones

Post1: El sistema recuerda los datos de la lección a agregar.

Nombre	Ingresar datos de nuevo ejercicio
Operación	<code>ingresarDatosEjercicio (descE: String, fraseE: String)</code>
Entrada	<ul style="list-style-type: none">● <code>descE</code>: descripción del ejercicio a ser agregado.

	● fraseE: frase que contiene el ejercicio a ser agregado.
Salida	No corresponde.
Descripción	Ingresar datos sobre el ejercicio a agregar al sistema.

Precondiciones y Postcondiciones

Post1: El sistema recuerda los datos del ejercicio a crear.

Nombre	Ingresar datos de ejercicio de tipo completar
Operación	ingresarDatosEjercicioCompletar(solE: Set(String))
Entrada	● solE: solución del ejercicio a crearse
Salida	No corresponde.
Descripción	Crea un nuevo ejercicio del tipo completar.

Precondiciones y Postcondiciones

Post1: El sistema recuerda los datos del ejercicio de tipo completar a crear.

Nombre	Ingresar datos de ejercicio de tipo traducir
Operación	ingresarDatosEjercicioTraduccion(solE: String)
Entrada	● solE: solución del ejercicio a crearse.
Salida	No corresponde.
Descripción	Crea un nuevo ejercicio del tipo completar.

Precondiciones y Postcondiciones

Pre1: El sistema recuerda los datos del ejercicio de tipo traducir a crear.

Nombre	Dar de alta curso nuevo
Operación	darAltaCurso()
Entrada	No corresponde.
Salida	No corresponde.
Descripción	Da de alta el curso en el sistema.

Precondiciones y Postcondiciones

Pre1: El sistema recuerda una instancia de Profesor, una instancia de Idioma, los datos del curso a agregar (nomC, descC, difC), un set de Strings previasC, así como los datos de las lecciones y sus correspondientes ejercicios a agregar.

Post1: Existe una instancia C de Curso con atributos:

- Nombre = nomC
- Descripcion = descC
- Dificultad = difC

Post2: Se creó un link se enseña entre C y la instancia de Idioma recordada por el sistema.

Post3: Se creó un link crea entre C y la instancia de Profesor recordada por el sistema.

Post4: Se creó un link previa de entre C y todas las instancias de Curso que tienen como atributo Nombre un valor que se encuentra en el Set de String previasC recordado por el sistema.

Post5: Se creó una instancia de Leccion por cada datos de lecciones que recuerda el sistema y se creó un link parte de entre esa instancia y la instancia de Curso C.

Post6: Se creó una instancia de Completar por cada conjunto de datos de ejercicio de tipo completar que recuerda el sistema.

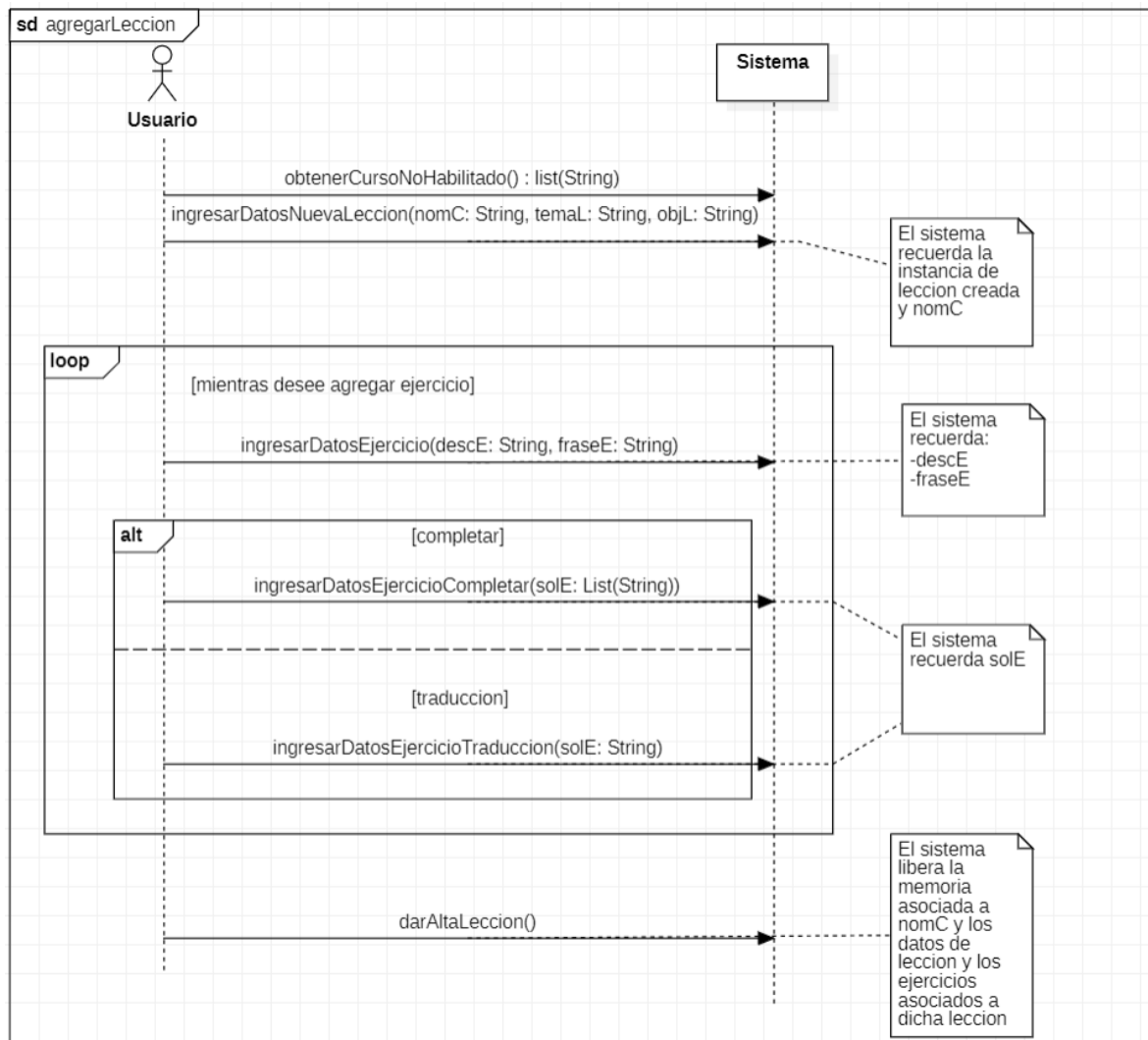
Post7: Se creó una instancia de Traducción por cada conjunto datos de ejercicio de tipo traducción que recuerda el sistema.

Post8: En el caso de que a una lección se le asignaran ejercicios, se creó un link parte de entre las instancias de lecciones creadas con sus correspondientes instancias de ejercicios creadas.

Post9: Se liberó toda la memoria del sistema.

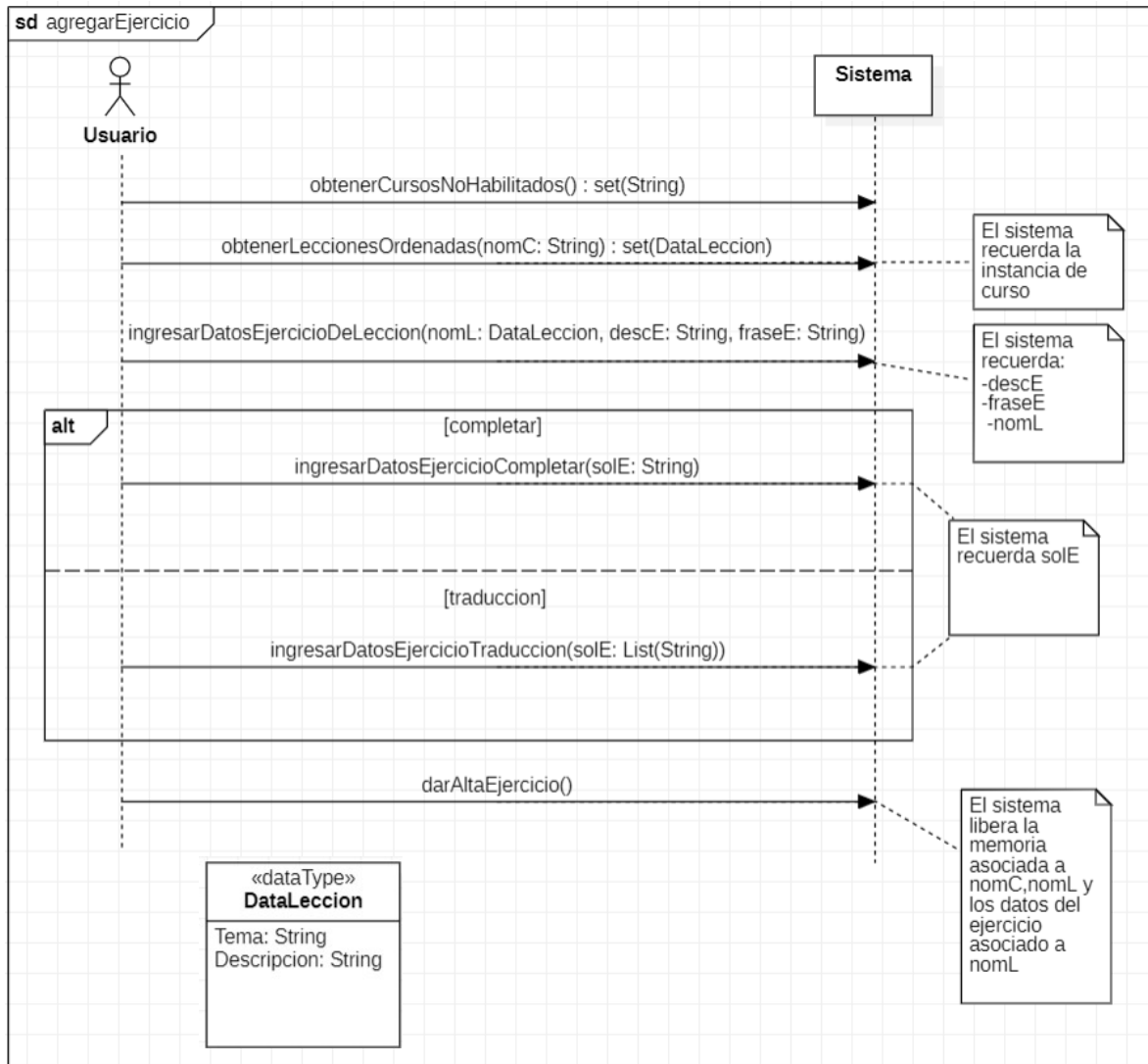
Agregar lección

Diagramas de Secuencia del Sistema



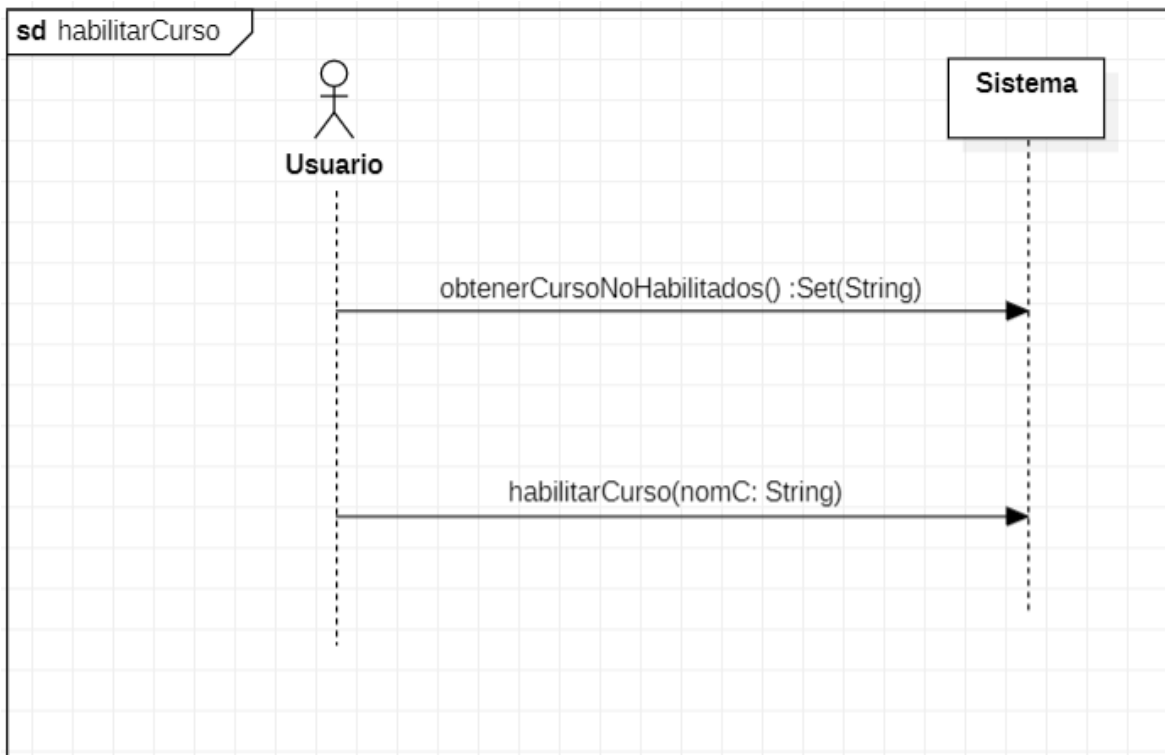
Agregar ejercicio

Diagramas de Secuencia del Sistema



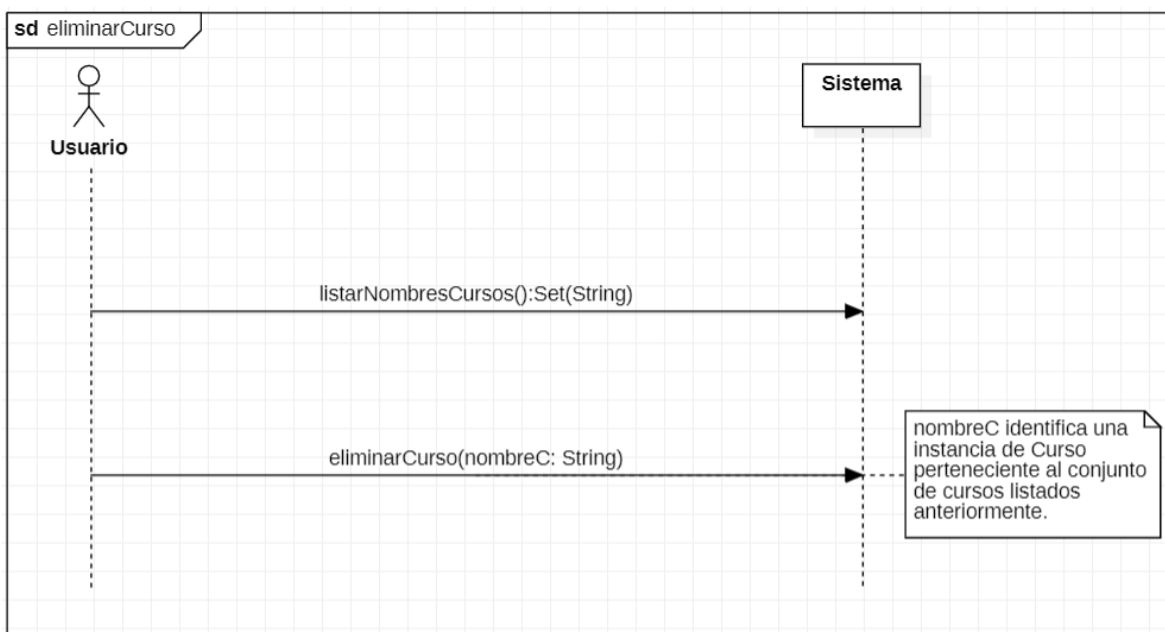
Habilitar curso

Diagramas de Secuencia del Sistema



Eliminar curso

Diagramas de Secuencia del Sistema



Contratos

Nombre	Listar Nombre de Cursos
Operación	<code>listarNombresCursos() : Set(String)</code>
Entrada	No corresponde.
Salida	Un Set de String con el nombre de cada instancia de <code>curso</code> que exista en el sistema.
Descripción	Retorna el nombre de todos los cursos existentes en el sistema.

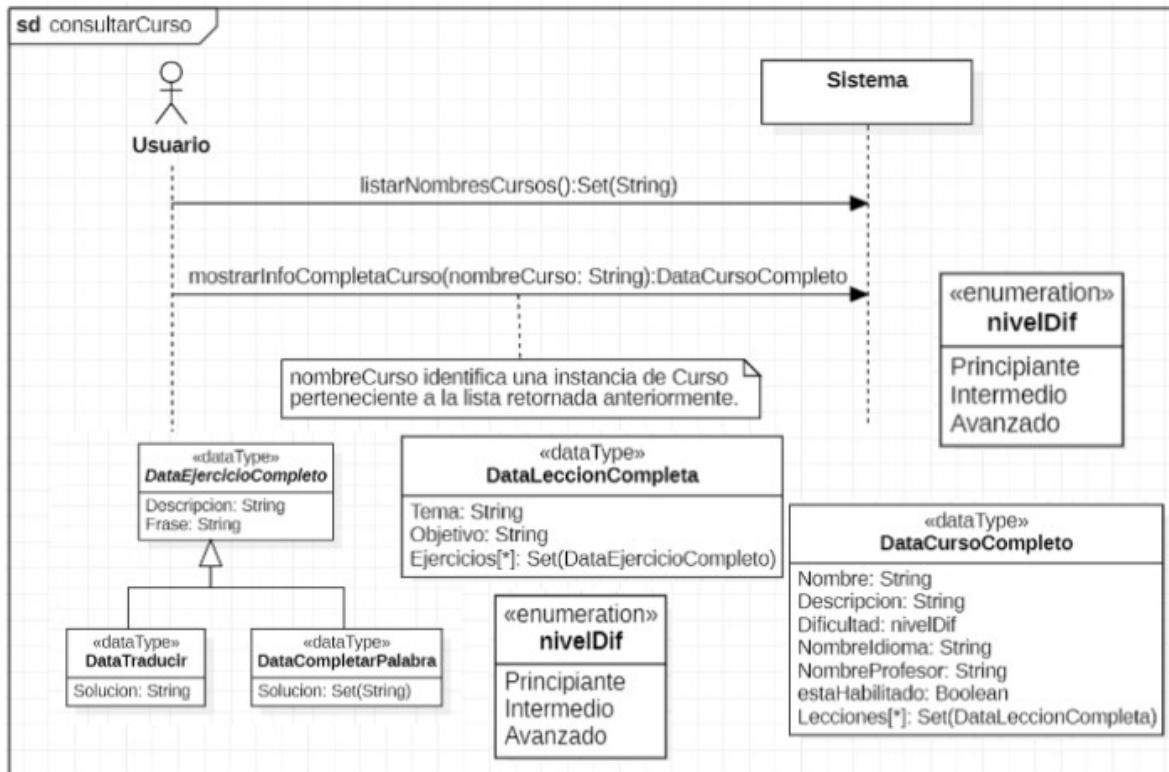
Precondiciones y Postcondiciones
Post1: Retorna un conjunto compuesto de elementos que representan el nombre de instancias de <code>Curso</code> existentes en el sistema.
Post2: Si no había ninguna instancia de <code>Curso</code> en el sistema, se retorna una lista vacía.

Nombre	Eliminar Curso
Operación	<code>eliminarCurso(nombreC: String)</code>
Entrada	<code>nombreC</code> representa el nombre que identifica una instancia de <code>Curso</code> en el sistema, que ha sido seleccionada para ser eliminada.
Salida	No corresponde.
Descripción	Elimina la instancia de <code>Curso</code> indicada, junto a todas las instancias de lecciones, ejercicios, inscripciones y registros de aprobaciones de los estudiantes, vinculadas con el curso.

Precondiciones y Postcondiciones
Pre1: Existe en el sistema una instancia de <code>Curso</code> con <code>Nombre = nombreC</code> .
Post1: Se eliminan las instancias de <code>Inscripción</code> de los estudiantes vinculadas al curso.
Post2: Se eliminan las instancias de <code>Progreso</code> que asocian las instancias de <code>Ejercicio</code> pertenecientes a la instancia de <code>Curso</code> con <code>Nombre = nombreC</code> , con estudiantes.
Post3: Se eliminan las instancias de <code>Ejercicio</code> vinculadas con el curso.
Post4: Se eliminan las instancias de <code>Leccion</code> vinculadas con el curso.
Post5: Se elimina la instancia de <code>Curso</code> con <code>Nombre = nombreC</code> .

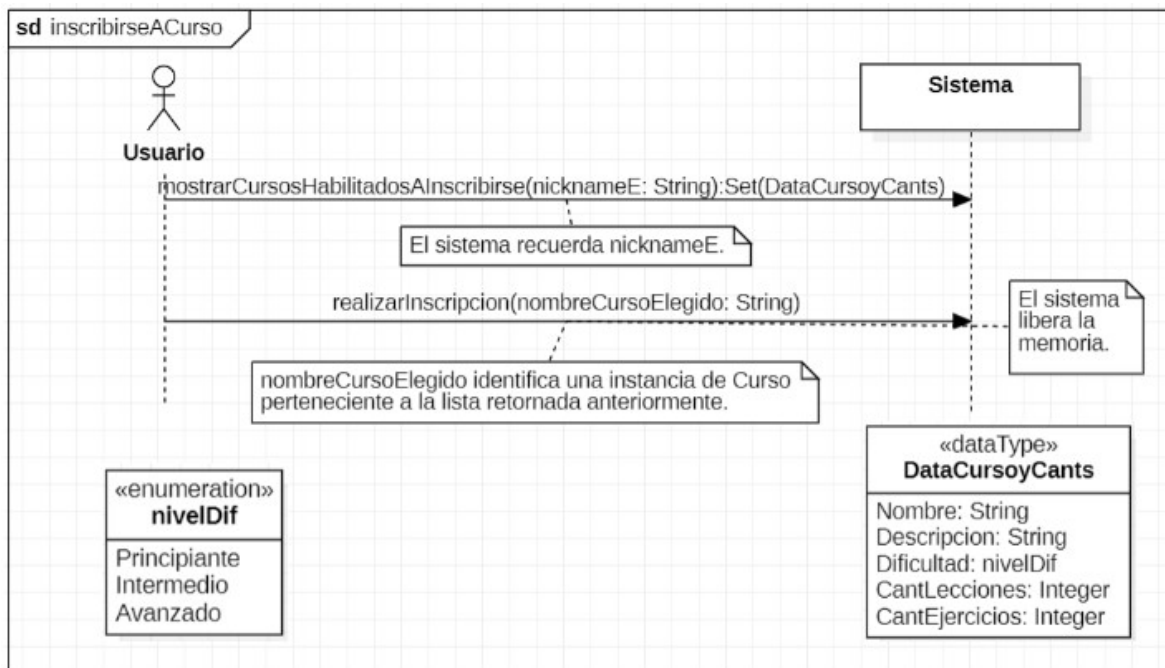
Consultar curso

Diagramas de Secuencia del Sistema



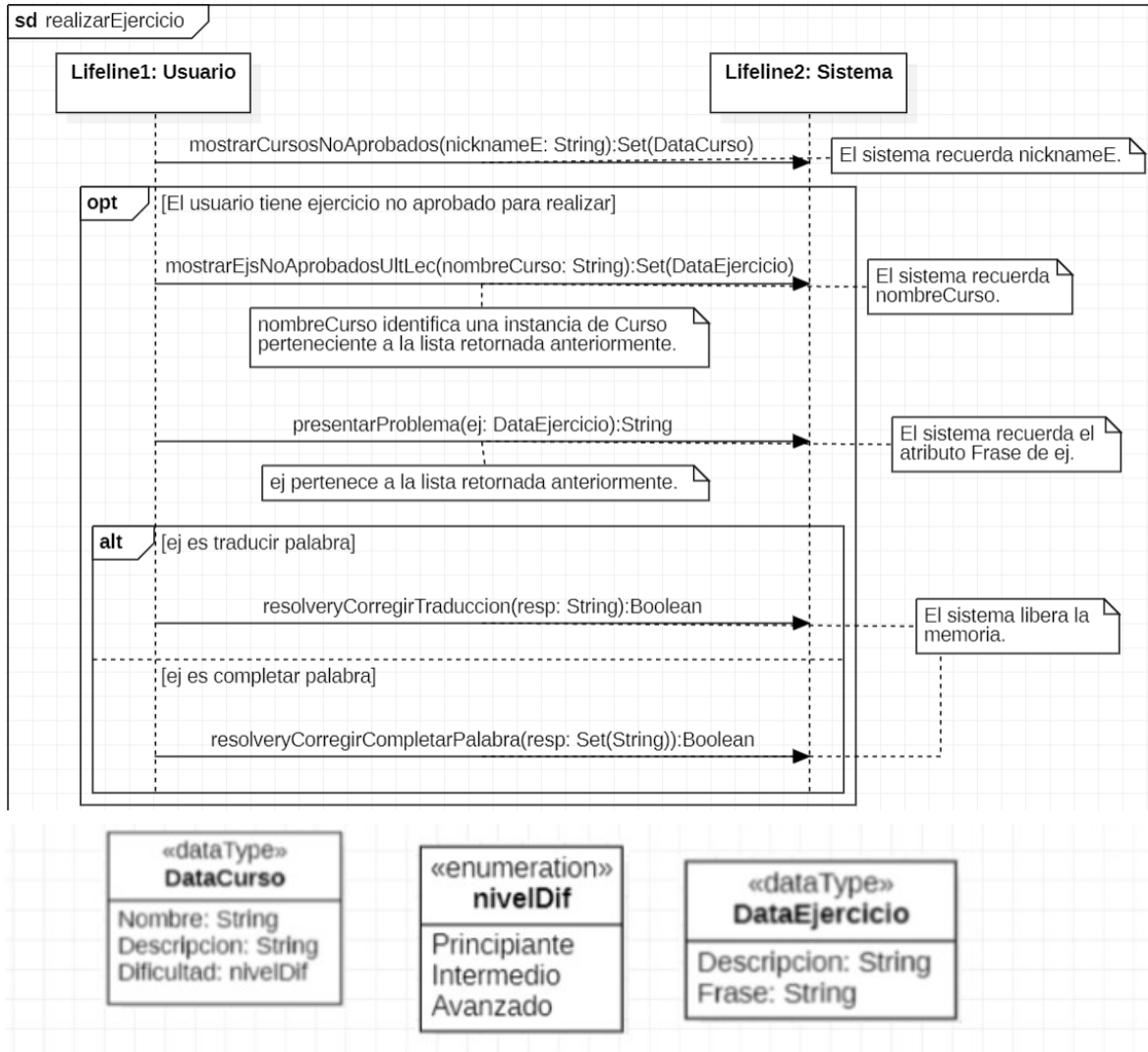
Inscribirse a curso

Diagramas de Secuencia del Sistema



Realizar ejercicio

Diagramas de Secuencia del Sistema



Contratos

Nombre	Mostrar cursos no aprobados
Operación	<code>mostrarCursosNoAprobados(nicknameE: String):Set(DataCurso)</code>
Entrada	<code>nicknameE</code> representa el <code>nickname</code> de un estudiante.
Salida	Un conjunto de las instancias de <code>Curso</code> asociadas mediante Inscripción al estudiante identificado por <code>nicknameE</code> , y que no ha aprobado.
Descripción	Dado el <code>nickname</code> de un estudiante, el Sistema muestra todos los cursos en los que el estudiante está inscripto y aún no ha aprobado.

Precondiciones y Postcondiciones
Pre1: <code>nicknameE</code> es el atributo <code>nickname</code> de una instancia de <code>Estudiante</code> registrado en el Sistema.
Post1: Se despliega un conjunto, cuyos elementos representan instancias de <code>Curso</code> en los que el estudiante, identificado por <code>nicknameE</code> , está inscripto y no ha aprobado.
Post2: El Sistema recuerda <code>nicknameE</code> .

Nombre	Mostrar los ejercicios no aprobados de la última lección de un curso
Operación	<code>mostrarEjsNoAprobadosUltLec(nombreCurso: String):Set(DataEjercicio)</code>
Entrada	<code>nombreCurso</code> es el identificador de una instancia de <code>Curso</code> en el Sistema.
Salida	Un conjunto de <code>datavalues DataEjercicio</code> , donde cada elemento representa una instancia de <code>Ejercicio</code> (ya sea de traducir o completar palabra) no aprobada de la última lección realizada por el estudiante, del curso identificado por <code>nombreCurso</code> . El estudiante es el identificado por el <code>nicknameE</code> en memoria del Sistema y el orden de lecciones respeta el establecido dentro del curso.
Descripción	Dado el nombre de un curso, el Sistema muestra los ejercicios no aprobados de la última lección realizada de ese curso, por el estudiante identificado por el <code>nicknameE</code> almacenado en memoria del Sistema.

Precondiciones y Postcondiciones
Pre1: El Sistema recuerda <code>nicknameE</code> .
Pre2: <code>nombreCurso</code> identifica una instancia de <code>Curso</code> perteneciente a la lista retornada anteriormente.
Pre3: El usuario asociado a <code>nicknameE</code> almacenado en el sistema, tiene al menos un ejercicio no aprobado para realizar de dicho <code>Curso</code> c identificado por <code>nombreCurso</code> .

Post1: El Sistema recuerda `nombreCurso`.

Post2: Se despliega un conjunto donde cada elemento representa una instancia de ejercicio (ya sea `Completa` o `TraducirPalabra`) no aprobado, por el estudiante identificado por el `nicknameE` almacenado en la memoria del sistema, de la última lección que ha realizado del curso identificado por `nombreCurso`, al que está inscripto.

Nombre	Presentar un problema a realizar
Operación	<code>presentarProblema(ej: DataEjercicio):String</code>
Entrada	<code>ej</code> es el ejercicio que desea resolver el estudiante.
Salida	Se retorna la frase presentada que pertenece a <code>ej</code> .
Descripción	El Usuario selecciona el ejercicio que desea realizar y el Sistema presenta su problema, o sea, presenta la frase que se debe resolver.

Precondiciones y Postcondiciones

Pre1: El Sistema recuerda `nicknameE`.

Pre2: El Sistema recuerda `nombreCurso`.

Pre3: El usuario asociado a `nicknameE` almacenado en el sistema, tiene ejercicio no aprobado para realizar.

Pre4: `ej` pertenece a la lista retornada anteriormente.

Post1: El sistema presenta el problema elegido por el Usuario.

Post2: El sistema recuerda el atributo `Frase` del `DataEjercicio ej`, que representa al ejercicio que se realiza de la última lección sin aprobar del curso identificado por `nombreCurso` en memoria, por el estudiante identificado por `nicknameE` en memoria.

Nombre	Resolver y corregir ejercicio de traducción
Operación	<code>resolveryCorregirTraduccion(resp: String):Boolean</code>
Entrada	<code>resp</code> representa la respuesta del estudiante asociado al <code>nicknameE</code> almacenado en memoria, al problema presentado por el Sistema que está asociado al ejercicio representado por el atributo <code>Frase</code> del <code>DataEjercicio ej</code> en memoria.
Salida	Retorna <code>true</code> si la respuesta ingresada coincide con la solución del ejercicio seleccionado a resolver, de lo contrario <code>false</code> .
Descripción	El estudiante ingresa su respuesta al ejercicio de traducción. El Sistema compara la respuesta ingresada con la solución de dicho ejercicio y si coinciden, el ejercicio se marca como aprobado por el estudiante. Si dicho ejercicio era el ultimo para completar la lección, se marca la lección como aprobada. Si dicha lección era la ultima del curso, se marca el curso como aprobado por el estudiante.

Precondiciones y Postcondiciones
<p>Pre1: El Sistema recuerda <code>nicknameE</code>.</p> <p>Pre2: El Sistema recuerda <code>nombreCurso</code>.</p> <p>Pre3: El usuario asociado a <code>nicknameE</code> almacenado en el sistema, tiene ejercicio no aprobado para realizar.</p> <p>Pre4: El sistema recuerda el atributo <code>Frase</code> del <code>DataEjercicio ej</code> que representa al ejercicio que se realiza, de la última lección sin aprobar del curso identificado por <code>nombreCurso</code> en memoria, por el estudiante identificado por <code>nicknameE</code> en memoria.</p> <p>Pre5: El ejercicio, representado por el atributo <code>Frase</code> del <code>DataEjercicio ej</code> en memoria, es una instancia de <code>Traducir</code>.</p> <p>Post1: Si la respuesta ingresada coincide con el atributo <code>Solucion</code> del ejercicio (representado por el atributo <code>Frase</code> de <code>ej</code> almacenado en memoria), se asigna <code>Aprobado = true</code> en la instancia <code>Progreso</code> que asocia dicha instancia de <code>Traducir</code> y el estudiante identificado por <code>nicknameE</code> almacenado en memoria. Sino, dicho atributo <code>Aprobado</code> continúa en <code>false</code>, para dicha asociación. Si corresponde asignarle <code>true</code> a dicho <code>Aprobado</code>, y el ejercicio resuelto es el último para completar la lección por parte del estudiante, se marca la lección como aprobada por el mismo. Si dicha lección era la última del curso, se marca el curso como aprobado por el estudiante.</p> <p>Post2: Se libera la memoria del Sistema.</p>

Nombre	Resolver y corregir ejercicio de completar palabra
Operación	<code>resolveryCorregirCompletarPalabra (resp: Set (String)) : Boolean</code>
Entrada	<code>resp</code> representa la respuesta del estudiante identificado por <code>nicknameE</code> almacenado en memoria, al problema presentado por el Sistema que está asociado al ejercicio identificado por <code>Frase</code> de <code>ej</code> en memoria.
Salida	Retorna <code>true</code> si la respuesta ingresada coincide con la solución del ejercicio seleccionado a resolver, de lo contrario <code>false</code> .
Descripción	El estudiante ingresa su respuesta al ejercicio de completar palabra. El Sistema compara la respuesta ingresada con la solución del ejercicio y si coinciden, el ejercicio se marca como aprobado por el estudiante. Si dicho ejercicio era el ultimo para completar la lección, se marca la lección como aprobada. Si dicha lección era la ultima del curso, se marca el curso como aprobado por el estudiante.

Precondiciones y Postcondiciones
<p>Pre1: El Sistema recuerda <code>nicknameE</code>.</p> <p>Pre2: El Sistema recuerda <code>nombreCurso</code>.</p> <p>Pre3: El usuario asociado a <code>nicknameE</code> almacenado en el sistema, tiene ejercicio no aprobado para realizar.</p>

Pre4: El sistema recuerda el atributo `Frase` del `DataEjercicio ej` que representa al ejercicio que se realiza, de la última lección sin aprobar del curso identificado por `nombreCurso` en memoria, por el estudiante identificado por `nicknameE` en memoria.

Pre5: El ejercicio, representado por el atributo `Frase` del `DataEjercicio ej` en memoria, es una instancia de `CompletarPalabra`.

Post1: Si la respuesta ingresada coincide con el atributo `Solucion` del ejercicio (representado por los atributos `Descripcion` y `Frase` de `ej` almacenados en memoria), se asigna `Aprobado = true` en la instancia `Progreso` que asocia dicha instancia de `CompletarPalabra` y el estudiante identificado por `nicknameE` almacenado en memoria. Sino, dicho atributo `Aprobado` continúa en `false`, para dicha asociación. Si corresponde asignarle `true` a dicho `Aprobado`, y el ejercicio resuelto es el último para completar la lección por parte del estudiante, se marca la lección como aprobada por el mismo. Si dicha lección era la última del curso, se marca el curso como aprobado por el estudiante.

Post2: Se libera la memoria del Sistema.

Consultar estadísticas

Diagramas de Secuencia del Sistema

