

# TALLER DE PROGRAMACIÓN

## TAREA 1

---

### Informe del Modelo de Casos de Uso

---

#### GRUPO 14

#### Integrantes

Nombre	CI
Manuel Rodríguez	5097757-4
Guido Dinello	5021022-5
Cristian González	4894107-6
Francisco Mauri	5140198-4
Belen Olivera	5598984-3

#### Docente

Andrea Delgado
----------------

# Índice

<b>1. Introducción</b>	<b>4</b>
1.1. Propósito	4
1.2. Alcance	4
1.3. Estructura del Documento	4
<b>2. Actores y Casos de Uso</b>	<b>4</b>
2.1. Actores	4
2.2. Casos de Uso	4
<b>3. Comportamiento de Casos de Uso</b>	<b>7</b>
3.1. Alta de Usuario	7
3.1.1. Diagramas de Secuencia del Sistema	7
3.1.2. Contratos	7
3.2. Consulta de Usuario	9
3.2.1. Diagramas de Secuencia del Sistema	9
3.2.2. Contratos	9
3.3. Modificar Datos de Usuario	11
3.3.1. Diagramas de Secuencia del Sistema	11
3.3.2. Contratos	11
3.4. Alta de Actividad Turística	13
3.4.1. Diagramas de Secuencia del Sistema	13
3.4.2. Contratos	13
3.5. Consulta de Actividad Turística	15
3.5.1. Diagramas de Secuencia del Sistema	15
3.5.2. Contratos	15
3.6. Alta de Salida Turística	16
3.6.1. Diagramas de Secuencia del Sistema	16

3.6.2. Contratos . . . . .	17
3.7. Consulta de Salida Turística . . . . .	18
3.7.1. Diagramas de Secuencia del Sistema . . . . .	18
3.7.2. Contratos . . . . .	18
3.8. Inscripción a Salida Turística . . . . .	19
3.8.1. Diagramas de Secuencia del Sistema . . . . .	19
3.8.2. Contratos . . . . .	19
3.9. Crear Paquete de Actividades Turísticas . . . . .	21
3.9.1. Diagramas de Secuencia del Sistema . . . . .	21
3.9.2. Contratos . . . . .	21
3.10. Agregar Actividad Turística a Paquete . . . . .	22
3.10.1. Diagramas de Secuencia del Sistema . . . . .	22
3.10.2. Contratos . . . . .	22
3.11. Consulta de Paquete de Actividades Turísticas . . . . .	24
3.11.1. Diagramas de Secuencia del Sistema . . . . .	24
3.11.2. Contratos . . . . .	24
3.12. Alta de Departamento . . . . .	25
3.12.1. Diagramas de Secuencia del Sistema . . . . .	25
3.12.2. Contratos . . . . .	25

# 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 tres 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

Se presenta una descripción de los actores involucrados en el modelo de Casos de Uso y una descripción de alto nivel de los casos de uso del sistema. Cada caso de uso incluye una sección con los actores involucrados en él.

## 2.1. Actores

Se incluye una breve descripción de cada uno de los actores involucrados. Cada actor está descrito de la siguiente manera:

Actor	Administrador del Sistema
Descripción	Encargado de emplear el sistema y ejecutar los diferentes casos de uso.

## 2.2. Casos de Uso

Cada caso de uso está descrito de la siguiente manera:

Nombre	Alta de Usuario
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea dar de alta a un nuevo usuario en el sistema. Para ello indica si es un/a turista o un/a proveedor/a y sus datos básicos: nickname (único en la plataforma), nombre, apellido, correo electrónico (único en la plataforma) y fecha de nacimiento. Si es turista se debe ingresar además otro dato básico: la nacionalidad. Si es proveedor/a se ingresan además otros datos básicos: una descripción general y un link a su sitio web (este último opcional). Si el nickname o el correo electrónico se encuentran en uso por algún otro usuario, el sistema avisa al administrador, pudiendo éste corregir la información o cancelar el alta. Finalmente, el sistema da de alta al usuario.

Nombre	Consulta de Usuario
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea consultar el perfil de un usuario. Para ello el sistema muestra la lista de todos los usuarios y el administrador elige uno. Luego, el sistema muestra todos los datos básicos del usuario. Si es proveedor/a se muestra también la información básica de las actividades turísticas que ofrece y salidas asociadas. Si es turista se muestra también la información de las salidas a las que se inscribió. Si el administrador selecciona una actividad turística o una salida, se muestra la información detallada, tal como se indica en los casos de uso Consulta de Actividad Turística y Consulta de Salida Turística, respectivamente.

Nombre	Modificar Datos de Usuario
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea modificar el perfil de un usuario. Para ello el sistema muestra la lista de todos los usuarios y el administrador elige uno. Luego, el sistema muestra todos los datos básicos del usuario. El administrador puede editar todos los datos básicos, menos el nickname y el correo electrónico. Cuando termina la edición, el sistema actualiza los datos del usuario.

Nombre	Alta de Actividad Turística
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea dar de alta una actividad turística. En primer lugar, el sistema lista todos los proveedores y departamentos. El administrador elige un proveedor y un departamento, indicando también otros datos básicos de la actividad: nombre (único en la plataforma), descripción, duración (en horas), costo (en pesos uruguayos), ciudad y fecha de alta. En caso de que exista una actividad turística con dicho nombre, el administrador puede modificar los datos o cancelar el caso de uso. Finalmente, el sistema da de alta la actividad turística.

Nombre	Consulta de Actividad Turística
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea consultar una actividad turística. En primer lugar, el administrador indica el departamento donde se realiza la actividad turística y el sistema lista las actividades asociadas a dicho departamento. El administrador elige una de ellas y el sistema devuelve todos los datos de la actividad, incluyendo la lista de salidas y paquetes asociados. Si el administrador selecciona una salida o paquete asociado a esa actividad turística, se muestra la información detallada, tal como se indica en los casos de uso Consulta de Salida Turística y Consulta de Paquete de Actividades Turísticas, respectivamente.

Nombre	Alta de Salida Turística
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea dar de alta una salida para una actividad turística. En primer lugar, el administrador indica un departamento y el sistema lista las actividades turísticas asociadas al mismo. El administrador elige una de ellas e ingresa los datos básicos de la salida: nombre de la salida (único en la plataforma), fecha / hora / lugar de salida, máxima cantidad de turistas, y la fecha de alta. En caso de que exista una salida con el nombre ingresado, el administrador puede modificar los datos o cancelar el caso de uso. Finalmente, el sistema da de alta la salida turística.

Nombre	Consulta de Salida Turística
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea consultar una salida de una actividad turística. En primer lugar, el administrador indica el departamento donde se realiza la actividad y el sistema lista las actividades turísticas asociadas al mismo. El administrador elige una y el sistema lista todas las salidas asociadas. El administrador elige una de ellas y el sistema devuelve todos los datos de la salida turística seleccionada.

Nombre	Inscripción a Salida Turística
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea realizar una inscripción de un/a turista a una salida de una actividad turística. En primer lugar, el administrador indica el departamento donde se realiza la actividad turística y el sistema muestra las actividades asociadas al mismo. El administrador elige una y el sistema muestra los datos básicos de las salidas vigentes, si existen. Luego, el sistema muestra la lista de todos/as los/as turistas y el administrador selecciona el/la turista que quiere inscribir, la salida a la que lo/la quiere inscribir, la cantidad de turistas para la inscripción y la fecha de inscripción. En caso de que ya exista un registro de el/la turista a la salida turística o se haya alcanzado el límite máximo de turistas, el administrador podrá (dependiendo del caso): cambiar la salida seleccionada, cambiar el/la turista seleccionado/a o cancelar el caso de uso. Finalmente, el sistema realiza la inscripción de el/la turista a la salida en dicha fecha, registrando el costo de la inscripción.

Nombre	Crear Paquete de Actividades Turísticas
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea crear un paquete de actividades turísticas. En primer lugar, el administrador ingresa los datos básicos del paquete: nombre (único en la plataforma), descripción, período de validez en días, descuento (porcentaje) y fecha de alta. En caso de que exista un paquete de actividades turísticas con dicho nombre, el administrador puede modificar los datos que ingresó o cancelar el caso de uso. Finalmente, el sistema registra el paquete de actividades turísticas.

Nombre	Agregar Actividad Turística a Paquete
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador quiere agregar una actividad turística a un paquete existente. En primer lugar, el sistema lista los nombres de los paquetes que tiene registrados. El administrador selecciona un paquete de actividades turísticas. Luego, el administrador selecciona un departamento y el sistema lista las actividades turísticas que se realizan en dicho departamento que no forman parte del paquete seleccionado. El administrador selecciona una actividad turística. Finalmente, el sistema ingresa la actividad turística al paquete.

Nombre	Consulta de Paquete de Actividades Turísticas
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador quiere consultar la información de un paquete de actividades turísticas. En primer lugar, el sistema lista los paquetes registrados. El administrador selecciona un paquete y el sistema devuelve sus datos básicos y las actividades turísticas que lo integran. En caso de seleccionar una actividad turística, se puede ver su información como indica el caso de uso Consulta de Actividad Turística.

Nombre	Alta de Departamento
Actores	Administrador del Sistema
Descripción	El caso de uso comienza cuando el administrador desea dar de alta un Departamento. Para ello, indica el nombre del departamento (único en la plataforma), una descripción y una URL. Si ya existe un departamento con igual nombre, el administrador puede reingresar los datos o cancelar el caso de uso. Finalmente, el sistema da de alta el departamento

### 3. Comportamiento de Casos de Uso

A continuación se presenta la especificación del comportamiento de cada caso, por medio de los Diagramas de Secuencia del Sistema (**sin memoria**) que sean necesarios, y los contratos de software de todas las operaciones del sistema incluidas en dichos diagramas.

#### 3.1. Alta de Usuario

##### 3.1.1. Diagramas de Secuencia del Sistema

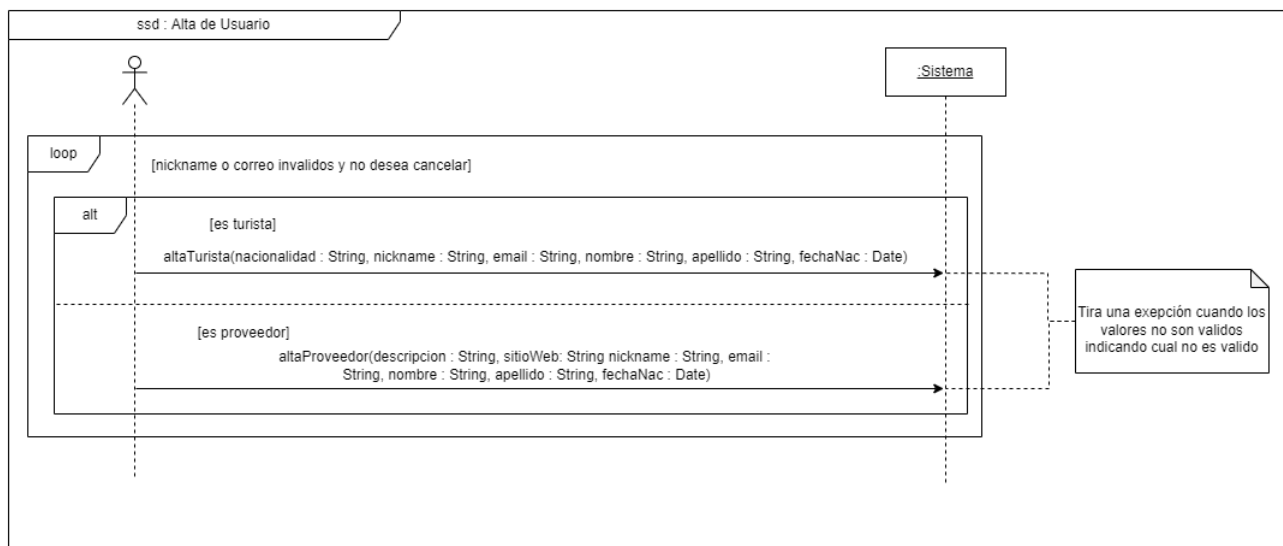


Figura 3.1: Modelo Conceptual del Sistema

##### 3.1.2. Contratos

<b>Operación</b>	<code>altaTurista(nacionalidad : String, nickname : String, email : String, nombre : String, apellido : String, fechaNac : Date)</code>
<b>Entrada</b>	nacionalidad, email, nombre, apellido, fechaNac son los atributos del Turista a dar de Alta.
<b>Salida</b>	No tiene
<b>Descripción</b>	Si los datos son validos se da de alta a un nuevo Turista en el Sistema sino tira una excepción.

**Precondiciones y Postcondiciones**

Pre: no tiene.

Post : Si existe en el sistema una instancia de Turista con atributo nickname igual al parámetro nickname, entonces corta la ejecución de la operación y lanza una excepción YaExisteUsuario

Post : Si existe en el sistema una instancia de Turista con atributo email igual al parámetro email, entonces corta la ejecución de la operación y lanza una excepción YaExisteUsuario

Post : Existe una nueva instancia de Turista en el Sistema con atributos igual a los parámetros proporcionados.

<b>Operación</b>	<code>altaProveedor(descripcion : String, sitioWeb: String, nickname : String, email: String, nombre : String, apellido : String, fechaNac : Date)</code>
<b>Entrada</b>	descripcion, sitioWeb, nickname, email, nombre, apellido, fechaNac son los atributos del Proveedor a dar de Alta.
<b>Salida</b>	No tiene
<b>Descripción</b>	Si los datos son validos da de alta a un nuevo Proveedor en el Sistema sino tira una excepción.

**Precondiciones y Postcondiciones**

Pre: no tiene

Post : Si existe en el sistema una instancia de Proveedor con atributo nickname igual al parámetro nickname, entonces corta la ejecución de la operación y lanza una excepción YaExisteUsuario

Post : Si existe en el sistema una instancia de Proveedor con atributo email igual al parámetro email, entonces corta la ejecución de la operación y lanza una excepción YaExisteUsuario

Post : Existe una nueva instancia de Proveedor en el Sistema con atributos igual a los parámetros proporcionados.



## 3.2. Consulta de Usuario

### 3.2.1. Diagramas de Secuencia del Sistema

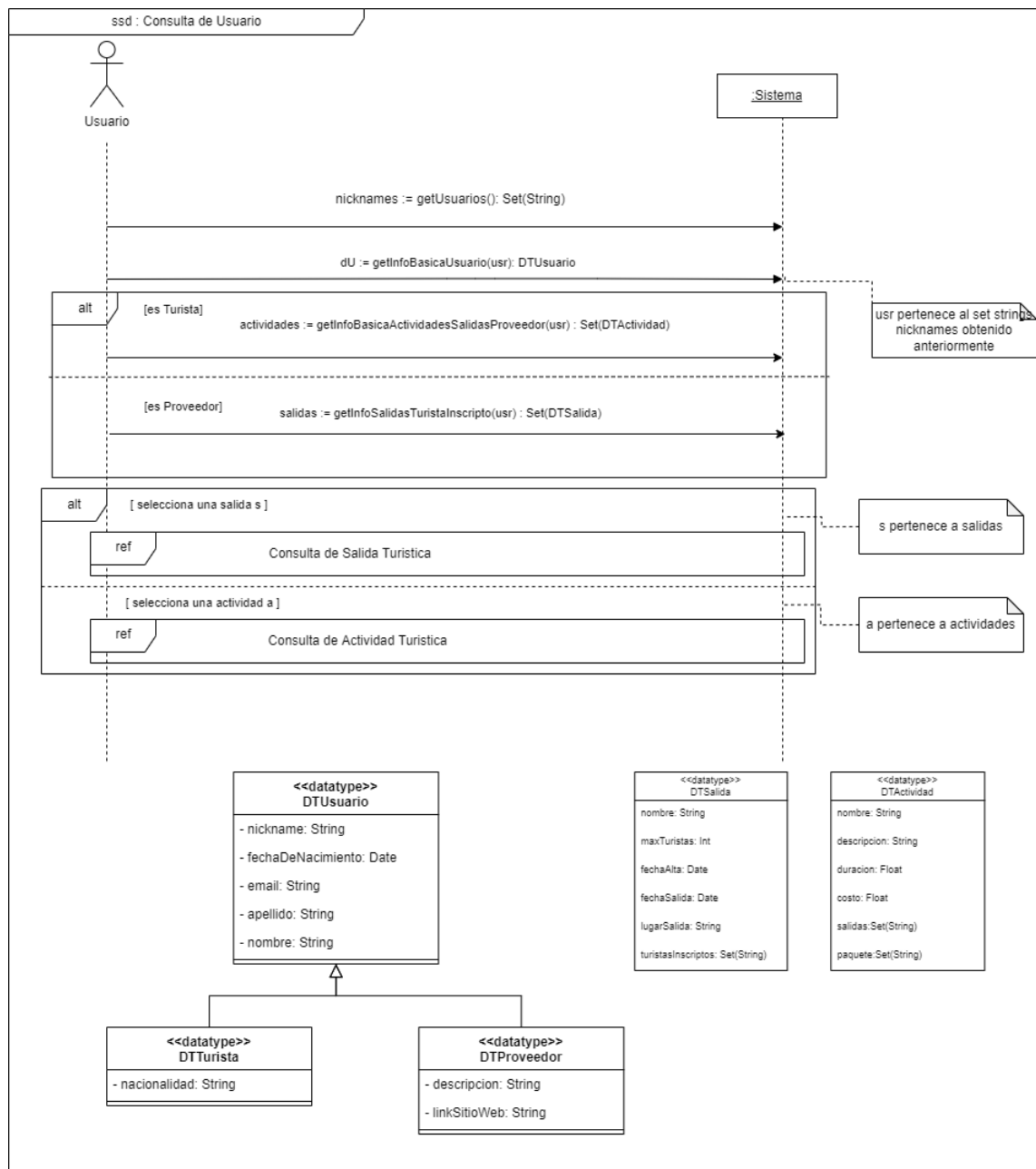


Figura 3.2: Modelo Conceptual del Sistema

### 3.2.2. Contratos

<b>Operación</b>	<code>getUsuarios():Set(String)</code>
<b>Entrada</b>	No corresponde
<b>Salida</b>	Colección con los nicknames de las instancias de Usuario registradas en el sistema.
<b>Descripción</b>	Busca y retorna los nicknames de los usuarios registrados

**Precondiciones y Postcondiciones**

Pre: no tiene

Post: Para cada instancia de Usuario registrada en sistema se recupera el valor de su atributo nickname, finalmente se crea un set con los valores recuperados y se lo retorna.

<b>Operación</b>	<code>getInfoBasicaUsuario(usr: String): DTUsuario</code>
<b>Entrada</b>	usr: nickname del usuario que se requiere la información
<b>Salida</b>	Dato de tipo DTUsuario, que contiene la información básica de una instancia de tipo Usuario registrada en el sistema, cuyo nickname coincide con usuario
<b>Descripción</b>	Busca y retorna la información del usuario de nickname usr.

**Precondiciones y Postcondiciones**

Pre: Hay una instancia de tipo Usuario registrada en el sistema cuyo valor en el atributo nickname coincide con usr (parámetro de la operación).

Post: Para la instancia de tipo Usuario guardada en el sistema cuyo valor en el atributo nickname coincide con el parámetro usr de entrada a la operación, se recupera los valores de los atributos, nickname, fechaDeNacimiento, email, apellido, nombre. En el caso de ser una instancia de Turista también se recupera el valor del atributo nacionalidad. En el caso de ser una instancia de Proveedor también se recupera los valores de los atributos descripcion y linkSitioWeb. Se crea un dato de tipo DTUsuario con dichos datos recuperados, luego se lo retorna.

<b>Operación</b>	<code>getInfoSalidasTuristaInscripto(usr: String) : Set(DTSalida)</code>
<b>Entrada</b>	usr es el valor del atributo nickname de la instancia de Turista para la cual se desea obtener la información de las Salidas a las cuales esta inscripto.
<b>Salida</b>	Es el conjunto de la información básica de la salidas a las cuales el turista de interés está inscripto.
<b>Descripción</b>	Devuelve la información básica de las salidas a las que el usuario de nickname usr esta inscripto.

**Precondiciones y Postcondiciones**

Pre : Existe en el sistema un Turista con atributo nickname igual a usr.

Post: para cada instancia de tipo SalidaTuristica que este asociada a la instancia de tipo IncripcionSalida que esta asociada a la instancia de tipo Turista con el valor del atributo nickname usr, se crea un dato de tipo DTSalida, finalmente se crea un set con los DTSalida creados y se lo retorna.

<b>Operación</b>	<code>getInfoActividadesSalidasProveedor(usr:String): Set(DTActividad)</code>
<b>Entrada</b>	usr es el valor del atributo nickname de la instancia de Proveedor para la cual se desea obtener la información de las Actividades que provee.
<b>Salida</b>	Conjunto de DTSalida que tiene la información básica de la actividades que el proveedor provee.
<b>Descripción</b>	Devuelve la información básica de las actividades a las que el usuario de nickname usr provee y las salidas asociadas.

#### Precondiciones y Postcondiciones

Pre : Existe en el sistema un Proveedor con atributo nickname igual al valor usr.

Post: para cada instancia de tipo ActividadTuristica que este asociada a la instancia de tipo Proveedor cuyo valor en el atributo nickname coincide con usr, se genera un dato de tipo DTActividad, que contiene además un set con los identificadores de las instancia de tipo SalidaTuristica asociadas a cada instancia ActividadTuristica.

### 3.3. Modificar Datos de Usuario

#### 3.3.1. Diagramas de Secuencia del Sistema

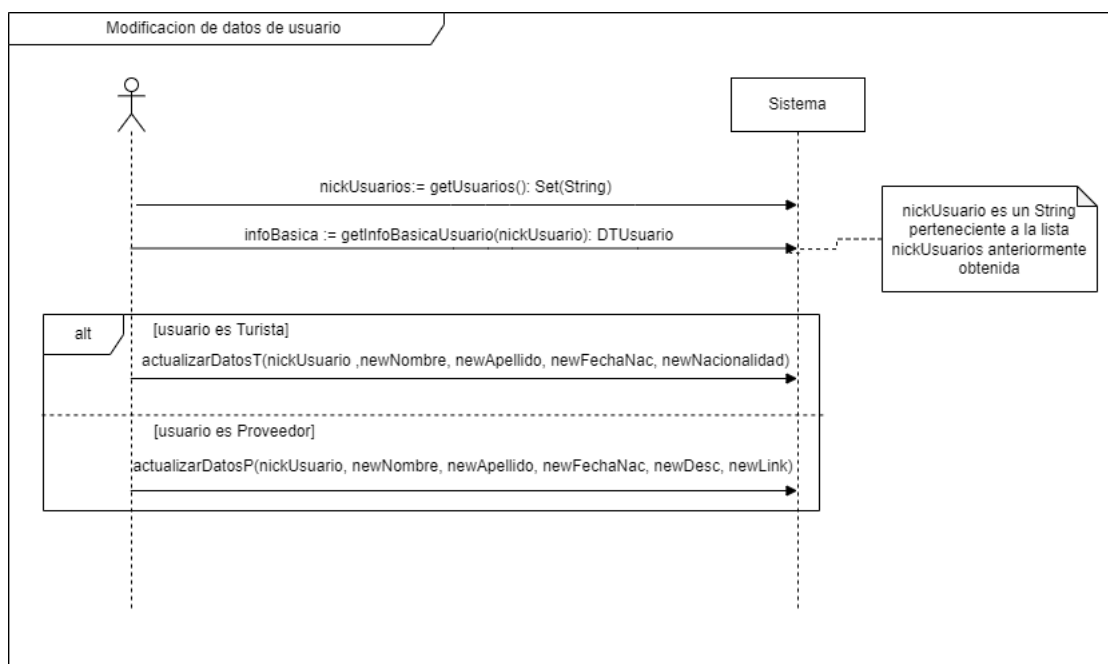


Figura 3.3: Modelo Conceptual del Sistema

#### 3.3.2. Contratos

<b>Operación</b>	<code>actualizarDatosT(nickUsuario: String, newNombre: String, newApellido: String, newFechaNac: Date, newNacionalidad: String)</code>
<b>Entrada</b>	nickUsuario, es el valor del atributo nickname de la instancia Turista "t" que se quiere actualizar; newNombre, es el valor por el cual se quiere cambiar el actual valor de nombre en "t"; newApellido, es el valor por el cual se quiere cambiar el actual valor de apellido en "t"; newFechaNac, es el valor por el cual se quiere cambiar el actual valor de fechaDeNacimiento de "t"; newNacionalidad, es el valor por el cual se quiere cambiar el actual valor de nacionalidad de "t".
<b>Salida</b>	No corresponde.
<b>Descripción</b>	Actualiza la información guardada en la instancia de tipo Turista cuyo nickname coincide con nickUsuario.

**Precondiciones y Postcondiciones**

Pre: Existe en el sistema una instancia de tipo Turista cuyo valor de nickname coincide con nickUsuario (parametro de la función).

Post: Los valores de los atributos nickname, nombre, apellido, fechaDeNacimiento y nacionalidad de la instancia de tipo Turista cuyo nickname sea nickUsuario, cambian por los valores pasados como parámetros de la operación, menos el atributo nickname que se usa solo para encontrar la instancia en el sistema.

<b>Operación</b>	<code>actualizarDatosP(nickUsuario: String, newNombre: String, newApellido: String, newFechaNac: Date, newDesc: String, newLink: String)</code>
<b>Entrada</b>	nickUsuario, es el valor del atributo nickname de la instancia de Proveedor "p" que se quiere actualizar; newNombre, es el valor por el cual se quiere cambiar el actual valor de nombre en "p"; newApellido, es el valor por el cual se quiere cambiar el actual valor de apellido en "p"; newFechaNac, es el valor por el cual se quiere cambiar el actual valor de fechaDeNacimiento de "p"; newDesc, es el valor por el cual se quiere cambiar el actual valor de descripcion de "p"; newLink es el valor por el cual se quiere cambiar el actual valor de linkSitioWeb en "p".
<b>Salida</b>	No corresponde.
<b>Descripción</b>	Actualiza la información guardada en la instancia de tipo Proveedor cuyo nombre nickname coincide con nickUsuario.

**Precondiciones y Postcondiciones**

Pre: Existe en el sistema una instancia de tipo Proveedor cuyo valor de nickname coincide con nickUsuario (parametro de la función).

Post: Los valores de los atributos nickname, nombre, apellido, fechaDeNacimiento, descripcion y linkSitioWeb de la instancia de tipo Proveedor cuyo nickname sea nickUsuario, cambian por los valores pasados como parámetros de la operación, menos el atributo nickname que se usa solo para encontrar la instancia en el sistema.

### 3.4. Alta de Actividad Turística

#### 3.4.1. Diagramas de Secuencia del Sistema

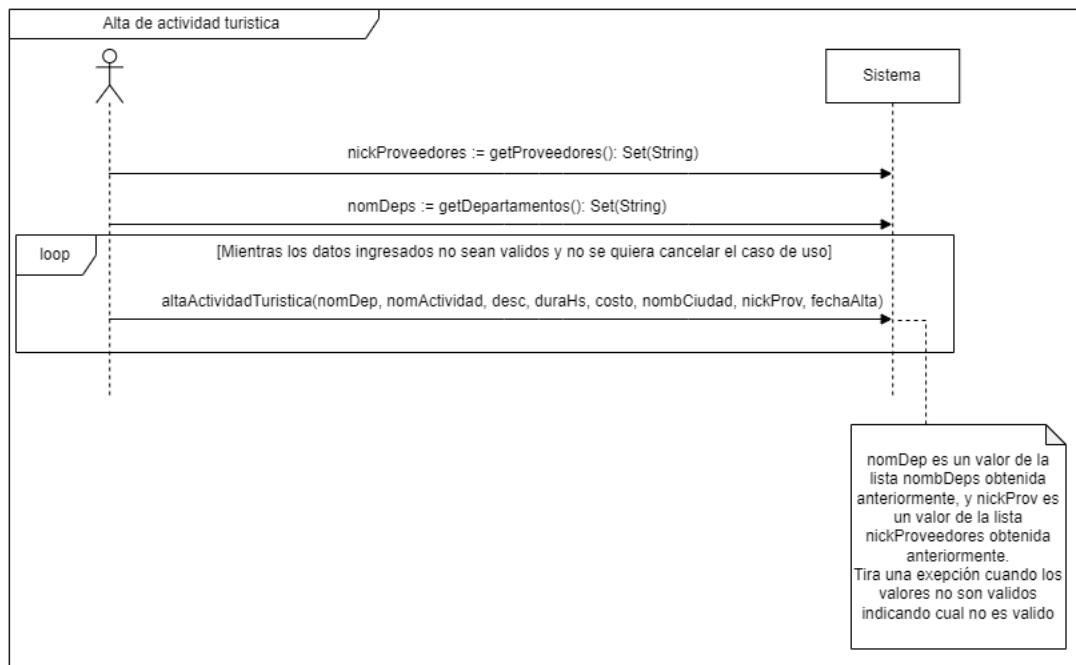


Figura 3.4: Modelo Conceptual del Sistema

#### 3.4.2. Contratos

<b>Operación</b>	<code>getProveedores(): Set(String)</code>
<b>Entrada</b>	No corresponde.
<b>Salida</b>	<code>Set(string)</code> de los identificadores de Proveedores en sistema.
<b>Descripción</b>	Retorna un set de strings, los identificadores (nickname) de cada proveedor en el sistema.

Precondiciones y Postcondiciones	
Pre: No tiene.	
Post: Se retorna un set de strings con el identificador de cada instancia de Proveedor en sistema.	

<b>Operación</b>	<code>getDepartamentos(): Set(String)</code>
<b>Entrada</b>	No corresponde.
<b>Salida</b>	<code>Set(string)</code> de los identificadores de Departamento en sistema.
<b>Descripción</b>	Retorna un set de strings, los identificadores (nombre) de cada departamento registrado en el sistema.

**Precondiciones y Postcondiciones**

Pre: No tiene.

Post: Se retorna un set de strings con el identificador de cada instancia de Departamento en sistema.

<b>Operación</b>	<code>altaActividadTuristica(nomDep: string, nomActividad: string, desc:sring, duraHs: integer, costo: float, nombCiudad: string, nickProv: string, fechaAlta: Date)</code>
<b>Entrada</b>	nomDep: nombre de un departamento ; nomActividad: nombre de una actividad turística ; desc: descripción de la actividad ; duraHs: duración en horas de la actividad ; costo: costo de la actividad ; nomCiudad: nombre de la ciudad donde se realiza la actividad ; nickProv: nickname del proveedor de la actividad ; fechaAlta; fecha en la que la actividad se dio de alta
<b>Salida</b>	No corresponde.
<b>Descripción</b>	Se da de alta una nueva ActividadTuristica.

**Precondiciones y Postcondiciones**

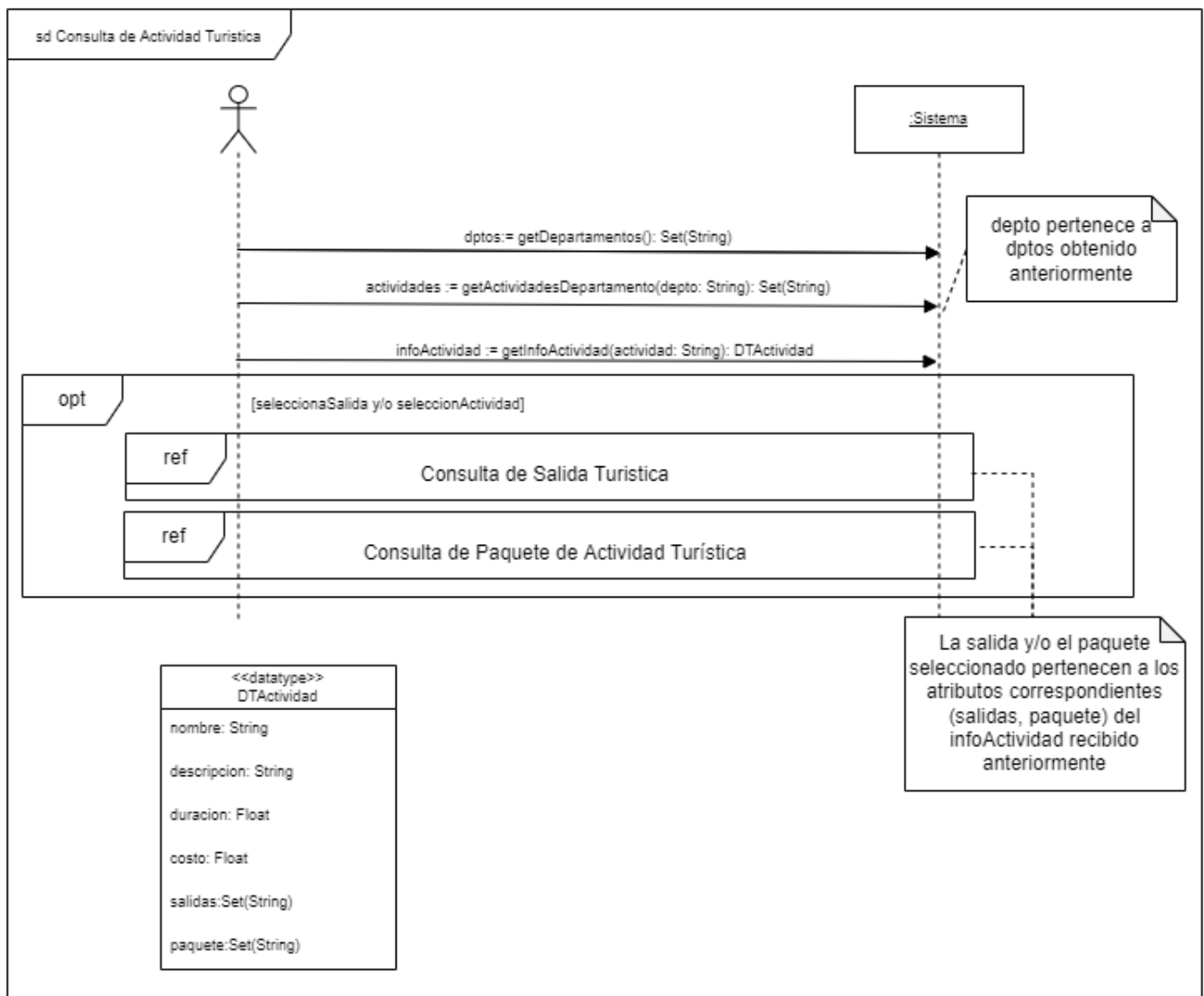
Pre: no tiene.

Post : Si existe en el sistema una instancia de ActividadTuristica con atributo nombre igual al parámetro nomActividad, entonces corta la ejecución de la operación y lanza una excepción YaExisteActividadTuristica.

Post : Existe en el sistema una nueva instancia de ActividadTuristica con identificador nomActividad y con atributos de igual valor a los parámetros.

### 3.5. Consulta de Actividad Turística

### 3.5.1. Diagramas de Secuencia del Sistema



**Figura 3.5: Modelo Conceptual del Sistema**

### 3.5.2. Contratos

<b>Operación</b>	getActividadesDepartamento(dpto: String): Set(String)
<b>Entrada</b>	dpto, es el nombre del Departamento del cual se requiere la información de las actividades que tiene asociadas.
<b>Salida</b>	Retorna un set con el valor de los atributos nombre de las instancias de tipo Actividad-Turistica asociadas a la instancia Departamento cuyo atributo nombre es "dpto"
<b>Descripción</b>	Retorna un conjunto con los nombres de las actividades disponibles en el Departamento cuyo atributo nombre es "dpto".

**Precondiciones y Postcondiciones**

Pre: Existe en el sistema una instancia de tipo Departamento cuyo atributo nombre coincide con dpto parametro de la operación.

Post: Para cada instancia de tipo ActividadTuristica guardada en sistema que este asociada con la instancia de tipo Departamento cuyo atributo nombre es dpto, se recupera el valor del atributo nombre, luego se crea un set con dichos valores recuperados y lo retorna.

<b>Operación</b>	getInfoActividad(actividad: String): DTActividad
<b>Entrada</b>	actividad es el nombre de la Actividad de la que se requiere la información.
<b>Salida</b>	Retorna un dato de tipo DTActividad que contiene la información de la instancia Actividad cuyo nombre coincide con actividad.
<b>Descripción</b>	Retorna un dato con la información de la Actividad cuyo nombre es actividad.

**Precondiciones y Postcondiciones**

Pre: Existe en el sistema una instancia de tipo ActividadTuristica cuyo atributo nombre coincide con actividad parametro de la operación.

Post: Para la instancia de tipo ActividadTuristica guardada en sistema cuyo nombre coincide con el parametro de la operación actividad, se crea un dato de tipo DTActividad con su información y lo retorna.

## 3.6. Alta de Salida Turística

### 3.6.1. Diagramas de Secuencia del Sistema

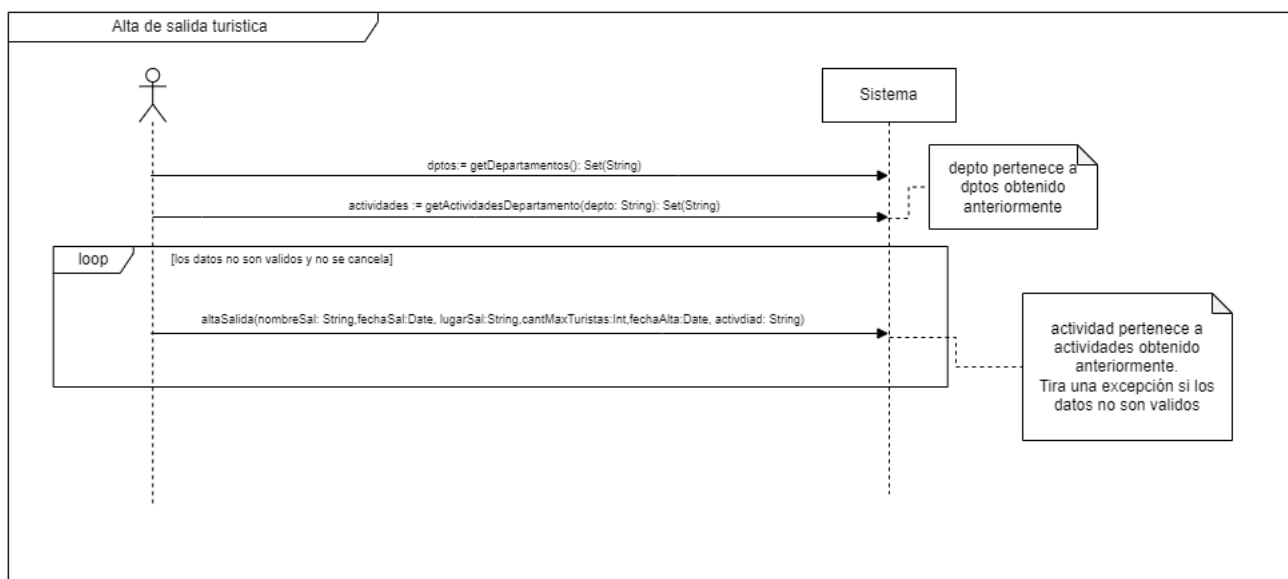


Figura 3.6: Modelo Conceptual del Sistema



### 3.6.2. Contratos

<b>Operación</b>	<code>altaSalida(nombreSal: String, fechaSal: Date, lugarSal: String, cantMaxTuristas: Int, fechaAlta: Date, actividad: String)</code>
<b>Entrada</b>	nombreSal: nombre de la nueva salida turistica ; fechaSal: fecha de la nueva salida ; lugarSal; lugar de la salida ; cantMaxTuristas: cantidad maxima de turistas en la salida ; fechaAlta: fecha de alta de la salida ; actividad: actividad asociada a la salida
<b>Salida</b>	No corresponde.
<b>Descripción</b>	Se crea una nueva salida turistica.

<b>Precondiciones y Postcondiciones</b>
---

Pre: No hay.

Post : Si existe en el sistema una instancia de SalidaTuristica con atributo nombre igual al parámetro nombre, entonces corta la ejecución de la operación y lanza una excepción YaExisteSalida.

Post : Existe una nueva instancia de SalidaTuristica en el Sistema con valor en los atributos igual a los parámetros. Dicha instancia esta asociada a la instancia de ActividadTuristica identificada con actividad (el valor del parámetro).

### 3.7. Consulta de Salida Turística

#### 3.7.1. Diagramas de Secuencia del Sistema

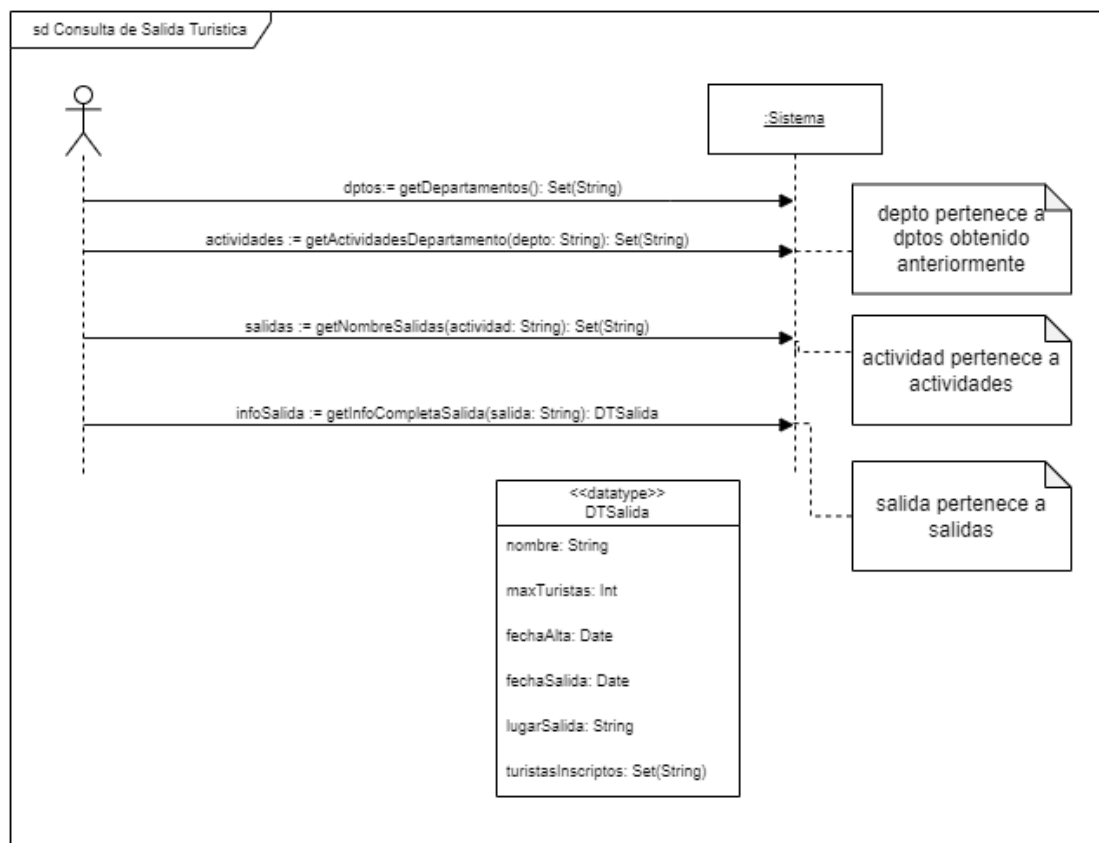


Figura 3.7: Modelo Conceptual del Sistema

#### 3.7.2. Contratos

<b>Operación</b>	<code>getNombreSalidas(actividad: String): Set(String)</code>
<b>Entrada</b>	actividad: Identifica una instancia de ActividadTuristica en el sistema.
<b>Salida</b>	Retorna un set de strings, los identificadores (nombre) de cada salida en el sistema asociada a la instancia de ActividadTuristica identificada por 'actividad'.
<b>Descripción</b>	Retorna los nombres de las salidas turísticas asociadas a una actividad turística dada.

##### Precondiciones y Postcondiciones

Pre: Debe existir en el sistema una instancia de ActividadTuristica identificada por 'actividad'.

Post: Se retorna un set de strings con el identificador de cada instancia de SalidaTuristica asociada a la instancia de ActividadTuristica identificada por 'actividad'.

<b>Operación</b>	<code>getInfoCompletaSalida(salida: String): DTSalida</code>
<b>Entrada</b>	salida: Identifica la salida que se quiere consultar.
<b>Salida</b>	Dato de tipo DTSalida, que contiene la información completa de una instancia de tipo SalidaTuristica registrada en el sistema, cuyo nombre coincide con salida.
<b>Descripción</b>	Busca y retorna la información de la salida de nombre salida.

#### Precondiciones y Postcondiciones

Pre: Hay una instancia de tipo SalidaTuristica registrada en el sistema cuyo valor en el atributo nombre coincide con salida (parámetro de la operación).

Post: Para la instancia de tipo SalidaTuristica guardada en el sistema cuyo valor en el atributo nombre coincide con el parámetro salida de entrada a la operación, se recupera los valores de los atributos, nombre, maxTuristas, fechaAlta, fechaSalida, lugarSalida, y un set de strings de turistasInscriptos(con el identificador de cada Turista asociado a la salida turistica). Se crea un dato de tipo DTSalida con dichos datos recuperados, luego se lo retorna.

## 3.8. Inscripción a Salida Turística

### 3.8.1. Diagramas de Secuencia del Sistema

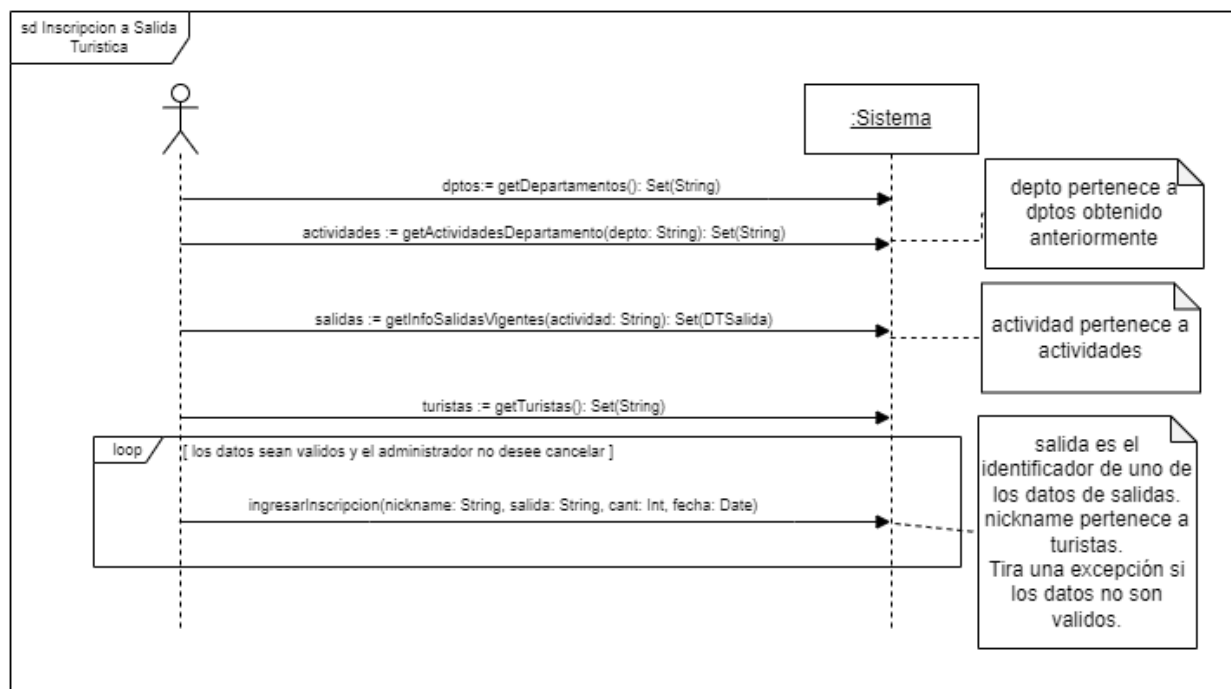


Figura 3.8: Modelo Conceptual del Sistema

### 3.8.2. Contratos

<b>Operación</b>	<code>getInfoSalidasVigentes(actividad: String): Set(DTSalida)</code>
<b>Entrada</b>	actividad: Identifica una instancia de ActividadTuristica en el sistema.
<b>Salida</b>	Conjunto de DTSalida correspondientes a las instancias de SalidaTuristica asociadas a la actividad ingresada.
<b>Descripción</b>	Retorna información sobre las salidas vigentes asociadas a una actividad dada.

**Precondiciones y Postcondiciones**

Pre: Debe existir en el sistema una instancia de SalidaTuristica cuyo atributo nombre es 'salida'.

Post: Para cada instancia de SalidaTuristica asociada a la instancia de ActividadTuristica con identificador 'actividad', si la salida está vigente, se crea un datatype DTSalida con la información de la misma.

<b>Operación</b>	<code>ingresarInscripcion(nickname: String, salida: String, cant: Int, fecha: Date)</code>
<b>Entrada</b>	nickname: Identifica una instancia de Turista en el sistema. salida: Identifica una instancia de SalidaTuristica en el sistema. cant: Representa la cantidad de turistas que se inscriben a la salida. fecha: Fecha en la que se realiza la inscripción.
<b>Salida</b>	No corresponde.
<b>Descripción</b>	Registra una inscripción de salida turística en el sistema con los datos proporcionados.

**Precondiciones y Postcondiciones**

Pre: Debe existir en el sistema una instancia de Turista cuyo atributo nickname es 'nickname'.

Pre: Debe existir en el sistema una instancia de SalidaTuristica cuyo atributo nombre es 'salida'.

Pre: El parámetro 'fecha' debe ser anterior al atributo fechaSalida en la instancia de SalidaTuristica identificada por 'salida'.

Post : Si el parámetro 'cant' es mayor a los plazos disponibles en la salida turística, se corta la ejecución de la operación y lanza una excepción PlazosExcedidos.

Post: Se crea una instancia de InscripcionSalida y se asocia con la instancia de Turista presente en el sistema identificada por 'nickname'. Además se asocia con la instancia de SalidaTuristica presente en el sistema identificada por 'salida'.

### 3.9. Crear Paquete de Actividades Turísticas

#### 3.9.1. Diagramas de Secuencia del Sistema

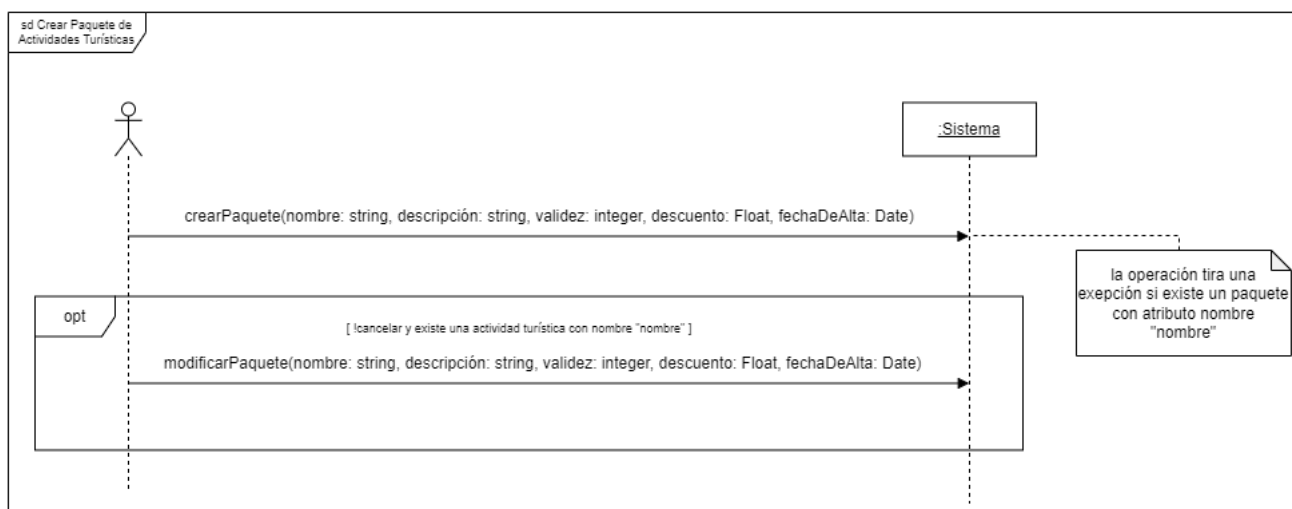


Figura 3.9: Modelo Conceptual del Sistema

#### 3.9.2. Contratos

<b>Operación</b>	<code>crearPaquete(nombre: string, descripción: string, validez: integer, descuento: Float, fechaDeAlta: Date)</code>
<b>Entrada</b>	nombre: el nombre del paquete nuevo ; descripción: la descripción del paquete nuevo ; validez: validez del paquete nuevo ; descuento: descuento del paquete nuevo ; fechaDeAlta: fecha de alta del paquete nuevo
<b>Salida</b>	No corresponde
<b>Descripción</b>	Crea un nuevo paquete turístico en el sistema

##### Precondiciones y Postcondiciones

Pre: No tiene

Post : Si existe en el sistema una instancia de `PaqueteTuristico` con atributo `nombre` igual al parámetro `nombre`, entonces corta la ejecución de la operación y lanza una excepción `YaExistePaqueteTuristico`

Post : Existe en el sistema una nueva instancia de `PaqueteTuristico` con identificador `nombre` y con atributos de igual valor a los parámetros.

<b>Operación</b>	<code>modificarPaquete(nombre: string, descripción: string, validez: integer, descuento: Float, fechaDeAlta: Date)</code>
<b>Entrada</b>	nombre: el nombre del paquete a modificar ; descripción: la descripción del paquete a modificar ; validez: validez del paquete a modificar ; descuento: descuento del paquete a modificar ; fechaDeAlta: fecha de alta del paquete a modificar
<b>Salida</b>	No corresponde
<b>Descripción</b>	Modifica un paquete turístico del sistema

#### Precondiciones y Postcondiciones

Pre: Existe una instancia de PaqueteTuristico con atributo nombre con el mismo valor que nombre (parámetro).

Post: Se modifican con los valores de los parámetros a los valores de la instancia de PaqueteTuristico con atributo nombre igual al parámetro nombre.

### 3.10. Agregar Actividad Turística a Paquete

#### 3.10.1. Diagramas de Secuencia del Sistema

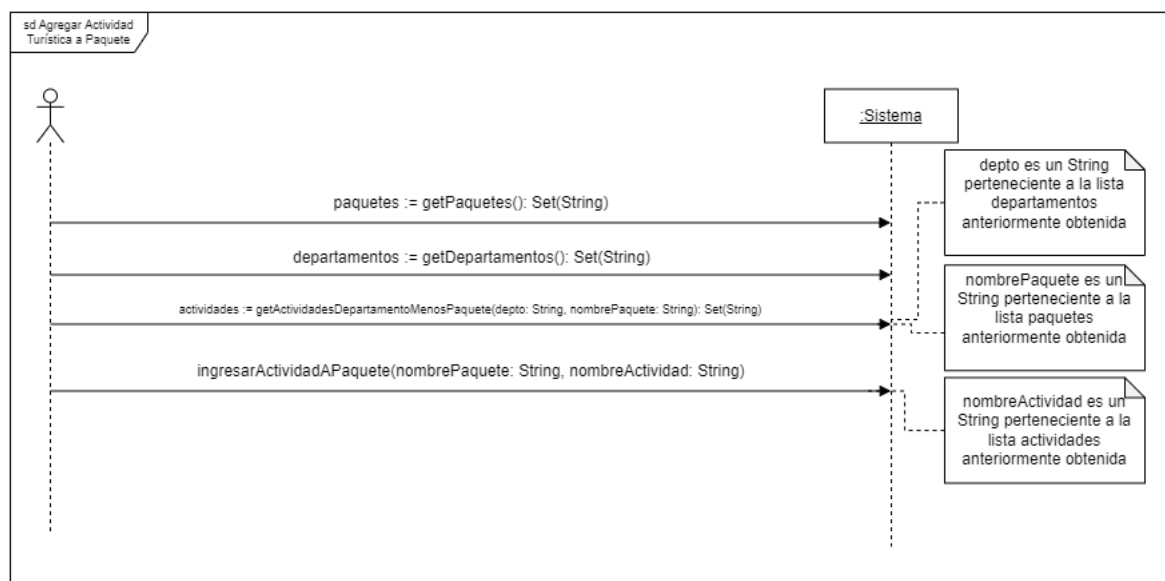


Figura 3.10: Modelo Conceptual del Sistema

#### 3.10.2. Contratos

<b>Operación</b>	getPaquetes(): Set(String)
<b>Entrada</b>	No corresponde.
<b>Salida</b>	Set(string) de los identificadores de PaqueteTuristico en sistema.
<b>Descripción</b>	Retorna un set de strings, los identificadores (nombre) de cada paquete registrado en el sistema.

#### Precondiciones y Postcondiciones

Pre: No tiene.

Post: Retorna un set de strings con el identificador de cada instancia de PaqueteTuristico en sistema.

<b>Operación</b>	<code>getActividadesDepartamentoMenosPaquete(depto: String, nombrePaquete: String): Set(String)</code>
<b>Entrada</b>	depto: nombre de un Departamento ; nombrePaquete: nombre de un PaqueteTuristico
<b>Salida</b>	Set(string) de los identificadores de ActividadTuristica en sistema. Excluyendo los identificadores de instancias de ActividadTuristica asociadas a la instancia de PaqueteTuristico con identificador nombrePaquete
<b>Descripción</b>	Retorna un set de strings con los identificadores de todas las actividades turisticas salvo por aquellas que ya se encuentran dentro del paquete turístico seleccionado.

**Precondiciones y Postcondiciones**

Pre: Existe una instancia de tipo Departamento identificada por depto guardada en el sistema.

Pre: Existe una instancia de tipo PaqueteTuristico identificada por nombrePaquete guardada en el sistema.

Post: Se retorna un set de strings con el identificador de cada instancia de ActividadTuristica en sistema que no esta asociada a una instancia de PaqueteTuristico con identificador nombrePaquete.

<b>Operación</b>	<code>ingresarActividadAPaquete(nombrePaquete: String, nombreActividad: String)</code>
<b>Entrada</b>	nombrePaquete: nombre de un paquete turistico ; nombreActividad: nombre de una actividad turistica
<b>Salida</b>	No corresponde.
<b>Descripción</b>	Agrega a la actividad turistica con identificador nombreActividad en el paquete turistico con identificador nombrePaquete.

**Precondiciones y Postcondiciones**

Pre: Existe una instancia de tipo ActividadTuristica identificada por nombreActividad guardada en el sistema.

Pre: Existe una instancia de tipo PaqueteTuristico identificada por nombrePaquete guardada en el sistema.

Post: Existe una asociación entre la instancia de tipo PaqueteTuristico con identificador nombrePaquete y la instancia de tipo ActividadTuristica con identificador nombreActividad.

### 3.11. Consulta de Paquete de Actividades Turísticas

#### 3.11.1. Diagramas de Secuencia del Sistema

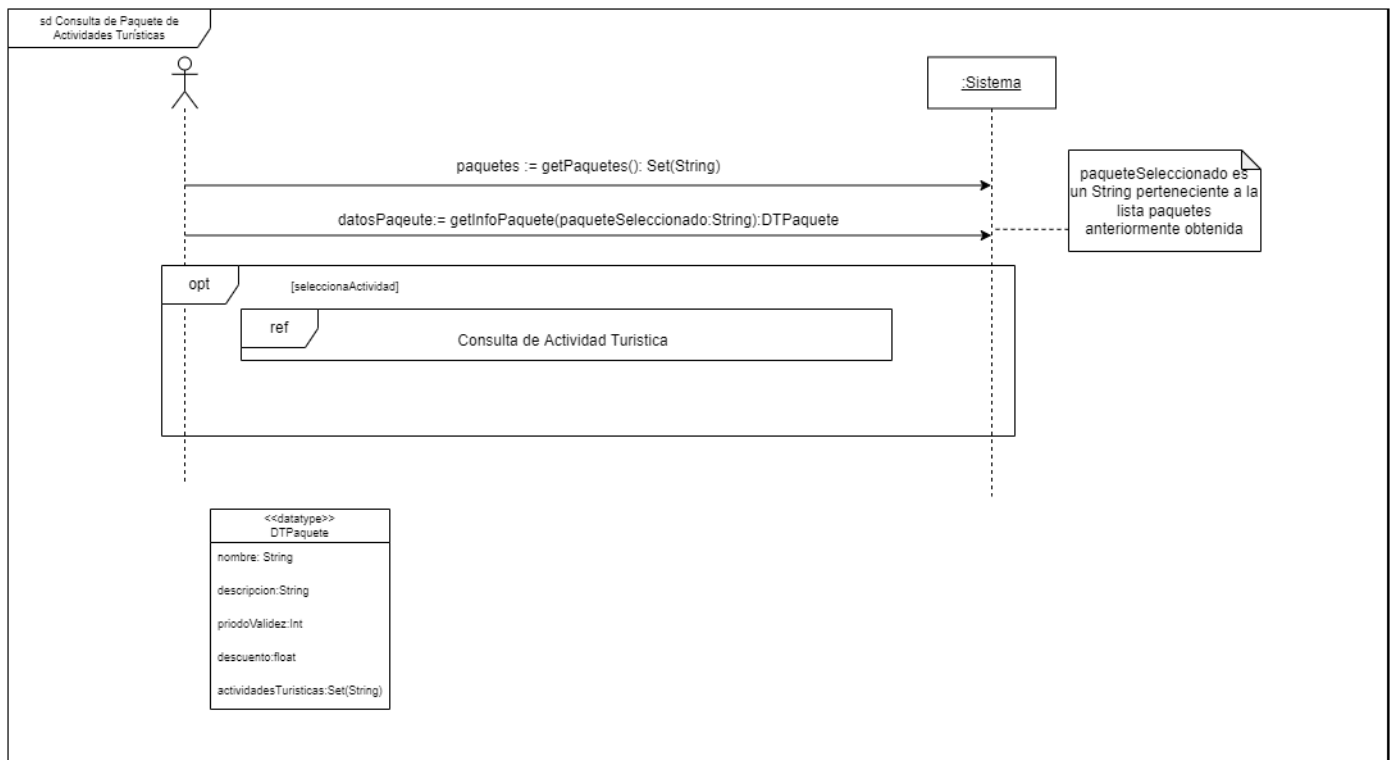


Figura 3.11: Modelo Conceptual del Sistema

#### 3.11.2. Contratos

<b>Operación</b>	<code>getInfoPaquete(paqueteSeleccionado:String):DTPaquete</code>
<b>Entrada</b>	<code>paqueteSeleccionado</code> : nombre de un paquete turístico
<b>Salida</b>	Retorna un DTPaquete con la información de la instancia PaqueteTuristico con identificador <code>paqueteSeleccionado</code> .
<b>Descripción</b>	Retorna un DTPaquete con la información del paquete turístico seleccionado.

##### Precondiciones y Postcondiciones

Pre: Debe existir en el sistema una instancia de PaqueteTuristico identificada por `paqueteSeleccionado`.

Post: Retorna un DTPaquete con la información de la instancia PaqueteTuristico con identificador `paqueteSeleccionado`.



## 3.12. Alta de Departamento

### 3.12.1. Diagramas de Secuencia del Sistema

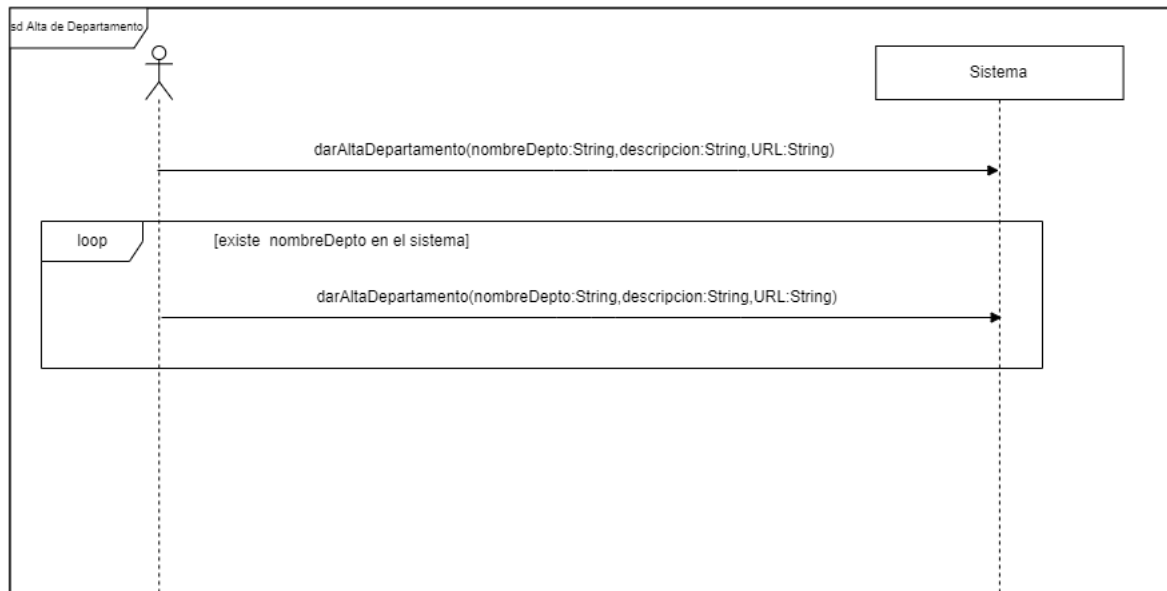


Figura 3.12: Modelo Conceptual del Sistema

### 3.12.2. Contratos

<b>Operación</b>	<code>darAltaDepartamento(nombreDepto:String, descripcion:String, URL:String)</code>
<b>Entrada</b>	nombreDepto: el nombre del nuevo departamento ; descripcion: la descripcion del nuevo departamento ; URL: URL del nuevo departamento
<b>Salida</b>	No corresponde
<b>Descripción</b>	Crea un nuevo departamento en el sistema

#### Precondiciones y Postcondiciones

Pre: No tiene

Post: Si existe en el sistema una instancia de Departamento con atributo nombre igual al parametro nombre, entonces corta la ejecución de la operación y lanza una excepción `YaExisteDepartamento`.

Post: Existe una nueva instancia de Departamento en el Sistema con valor de atributos igual a los parámetros proporcionados.