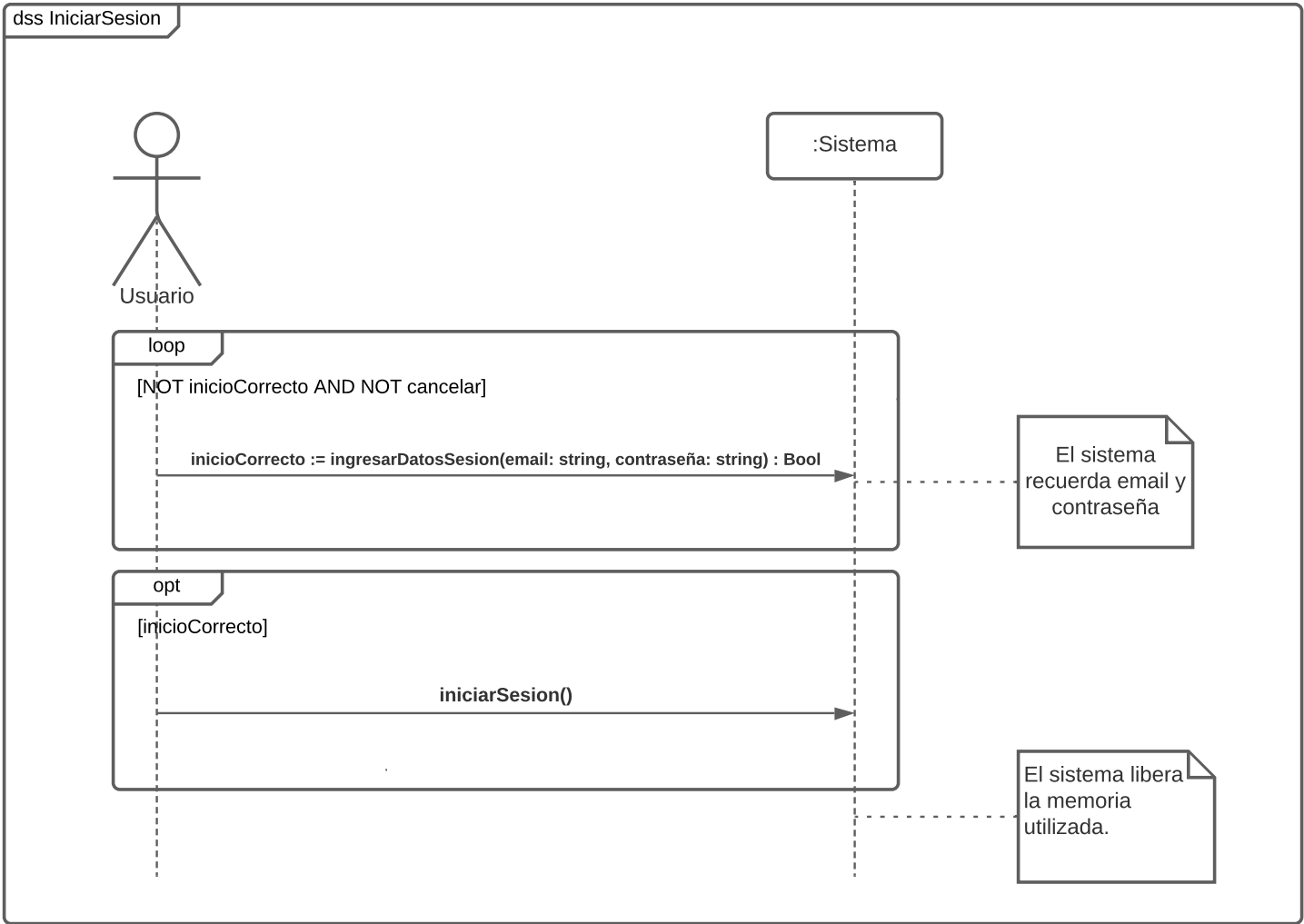
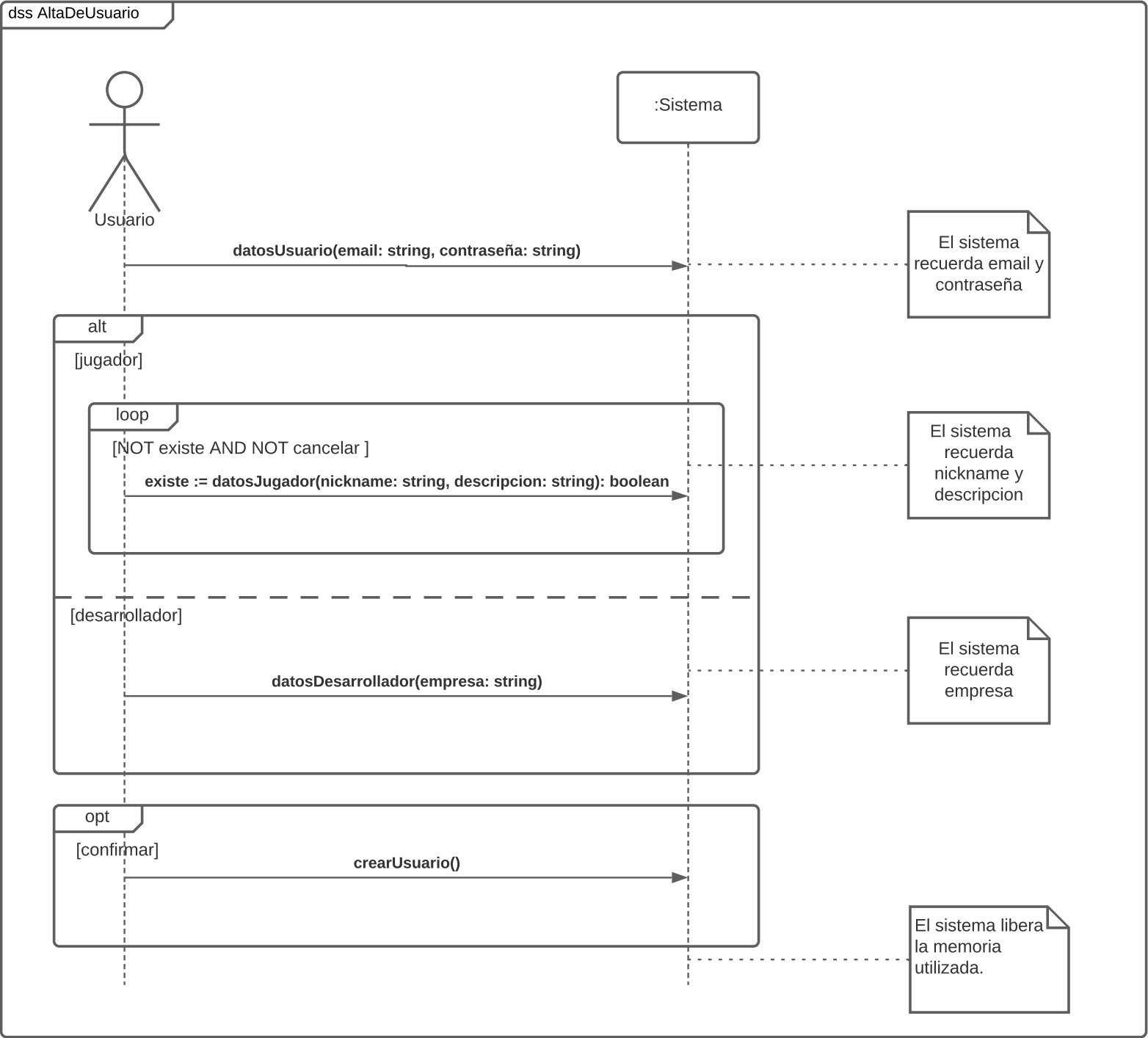
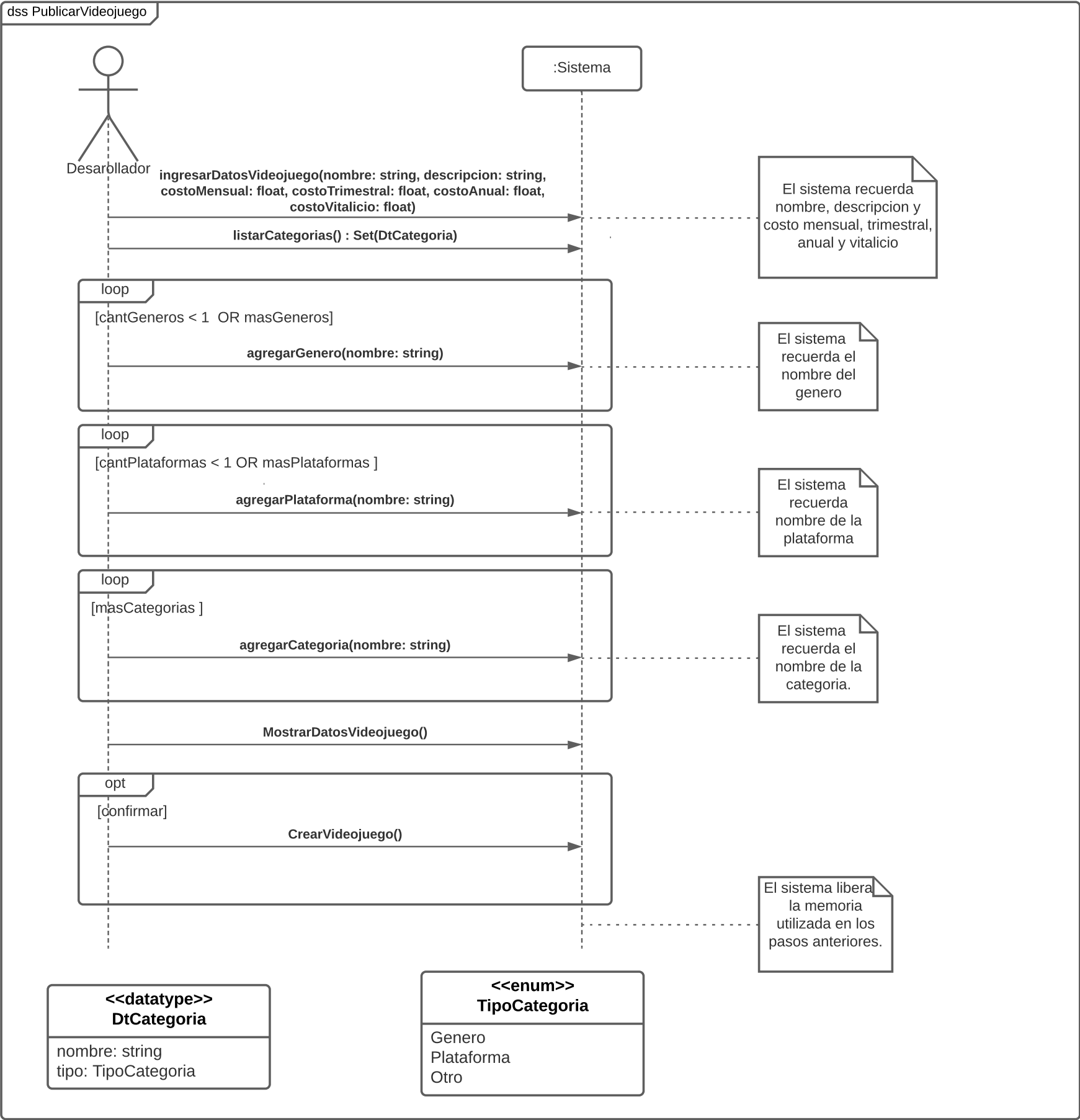
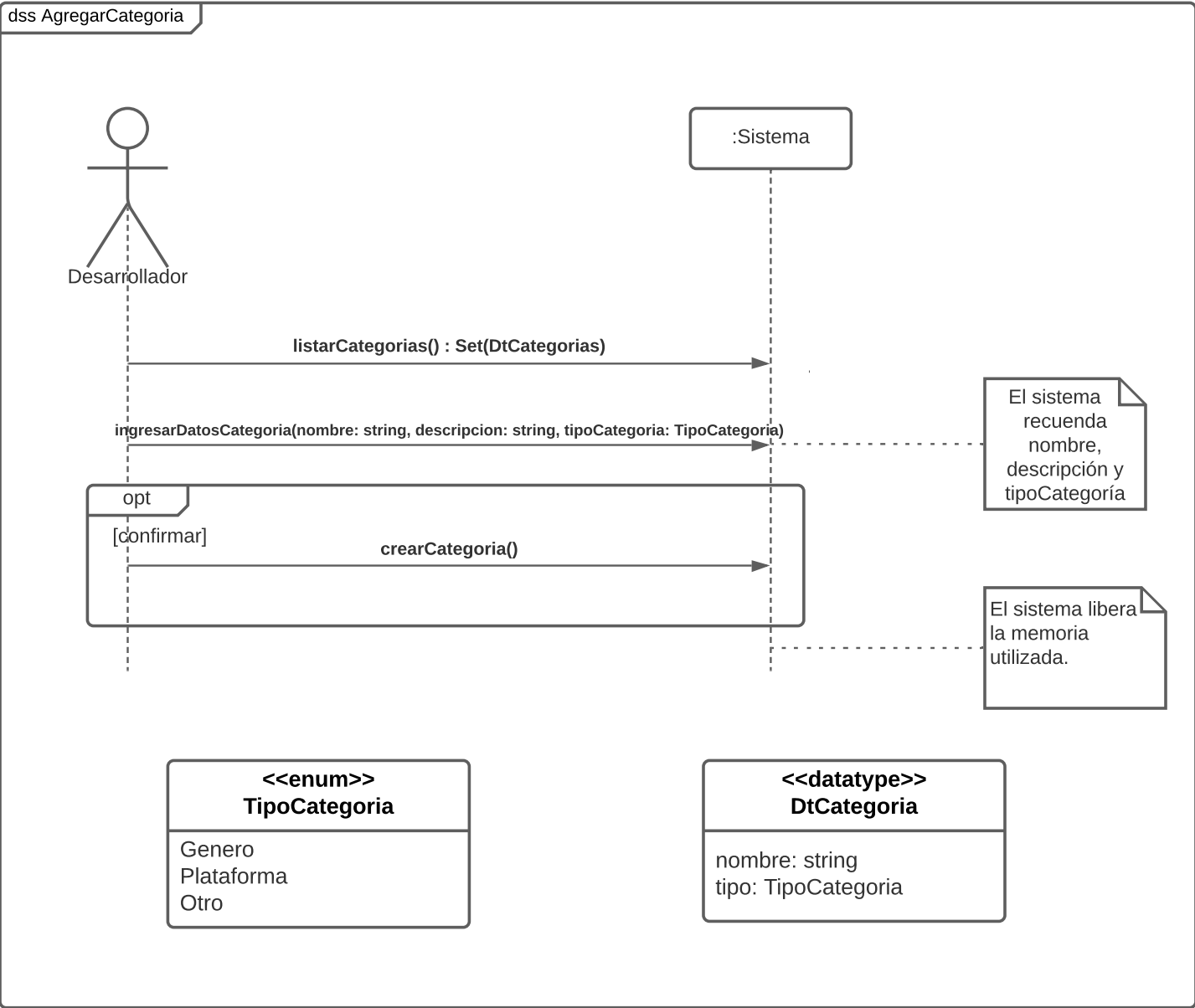


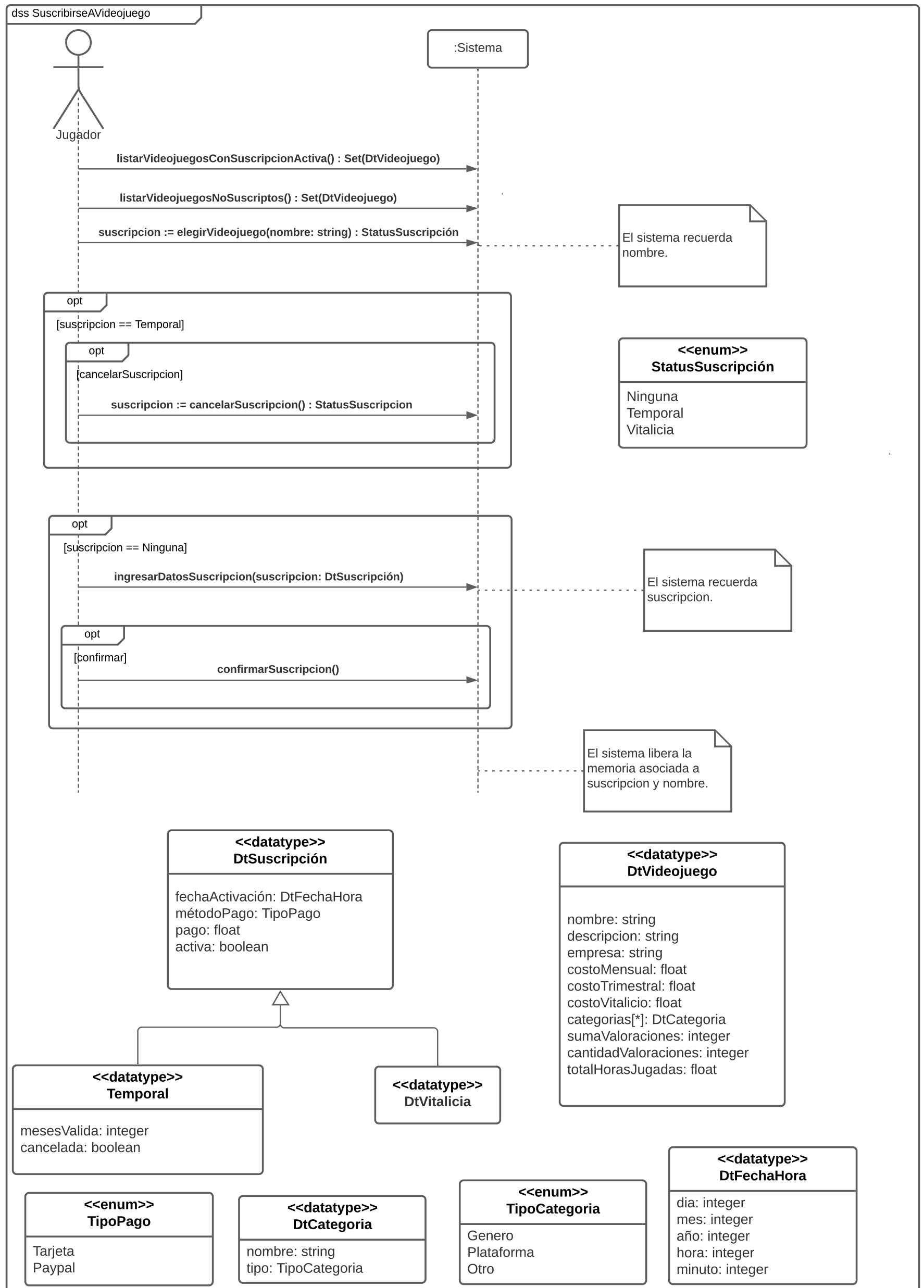
Programación 4

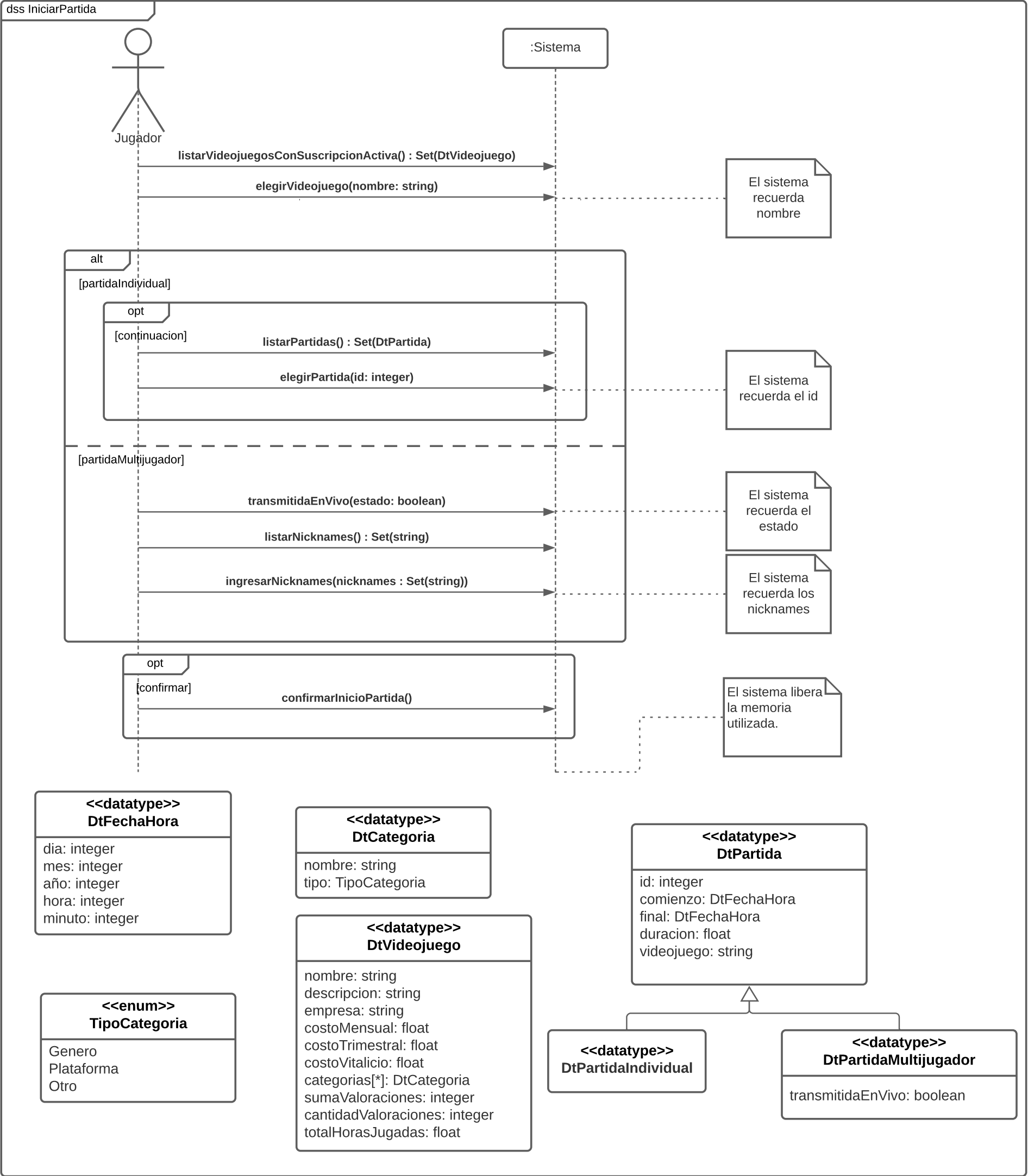
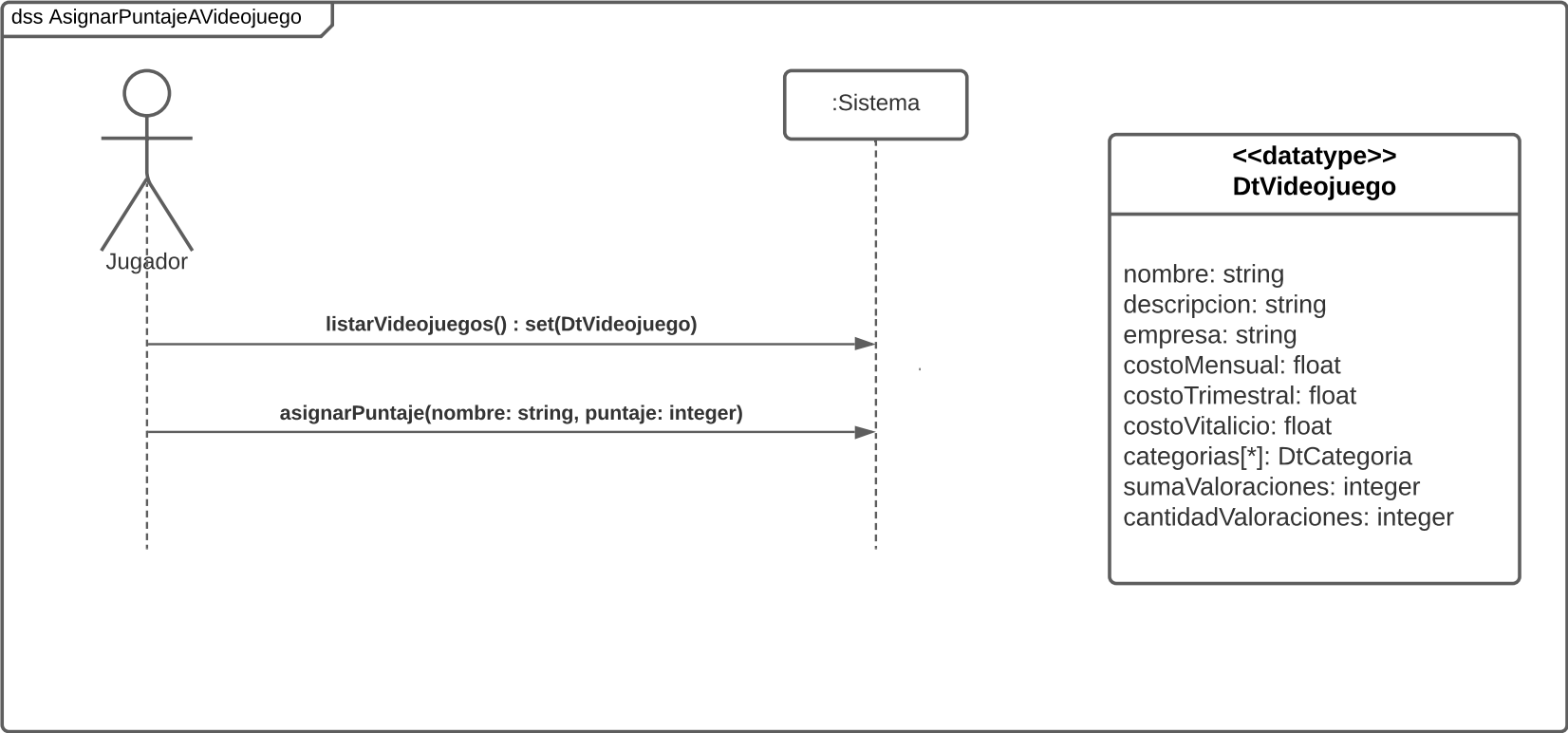
Laboratorio 2

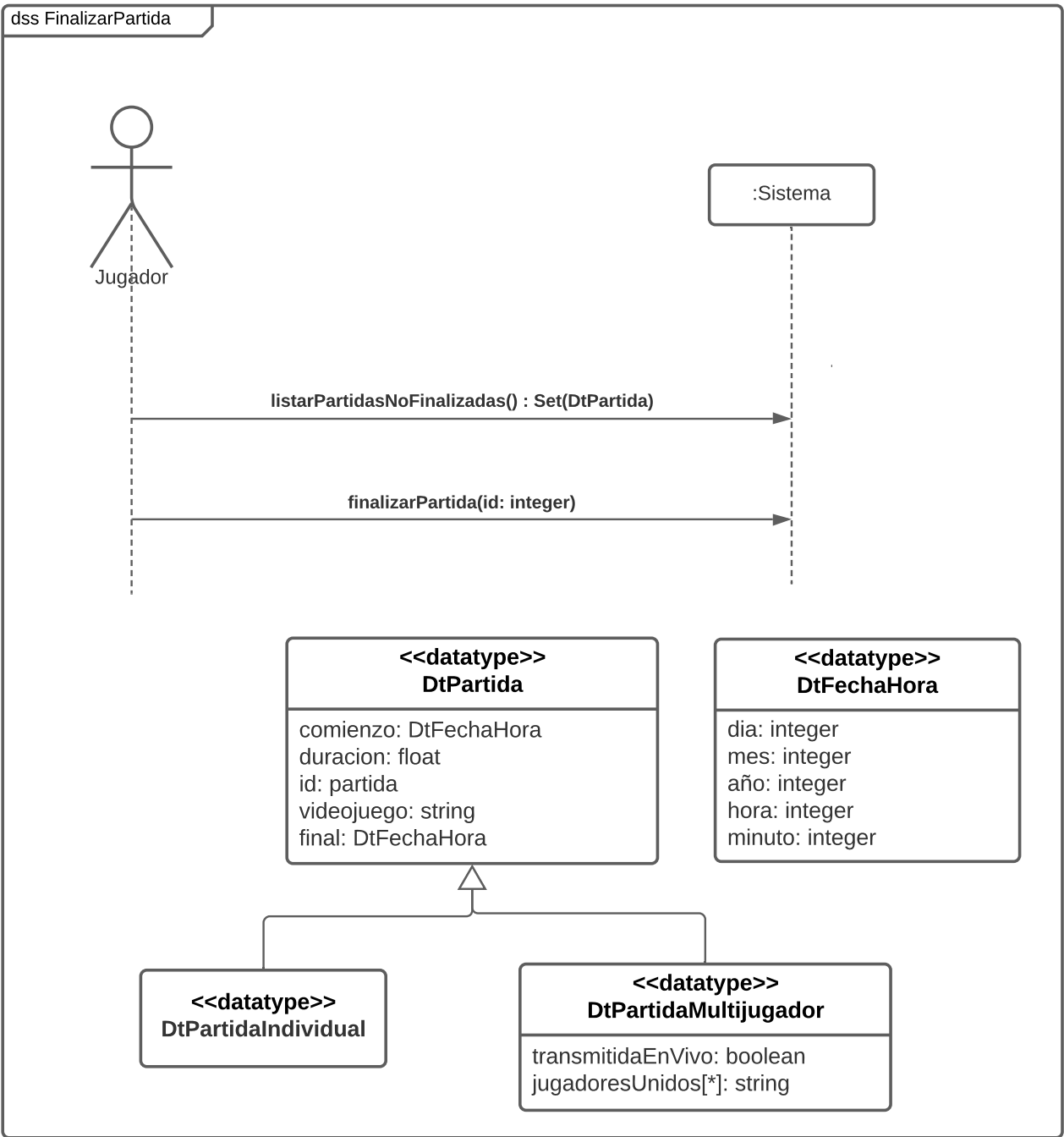
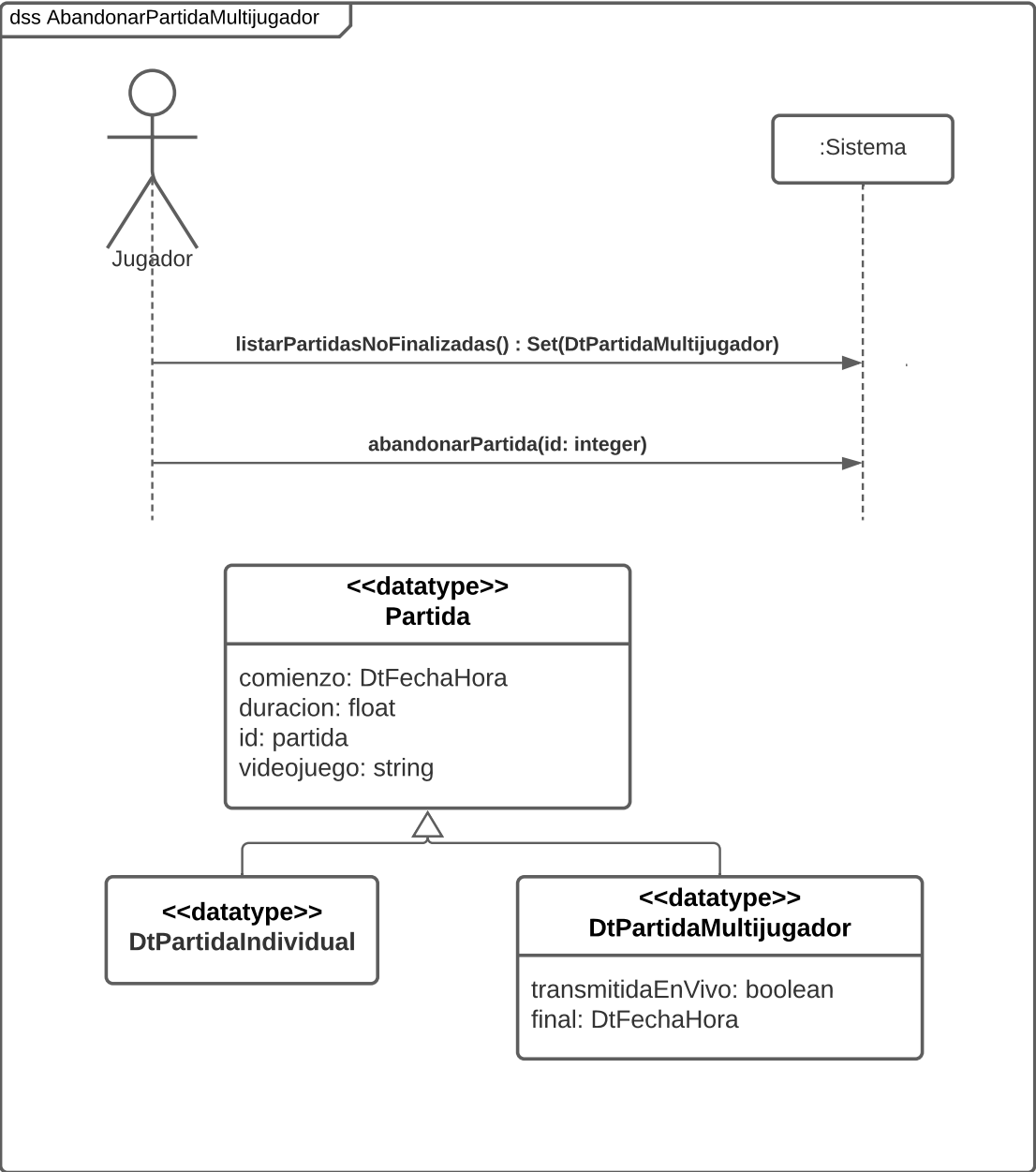
Roberto de Armas
Bruno Fernandez
Agustin Koltukian
Rafael Schol
Agustina Martinez

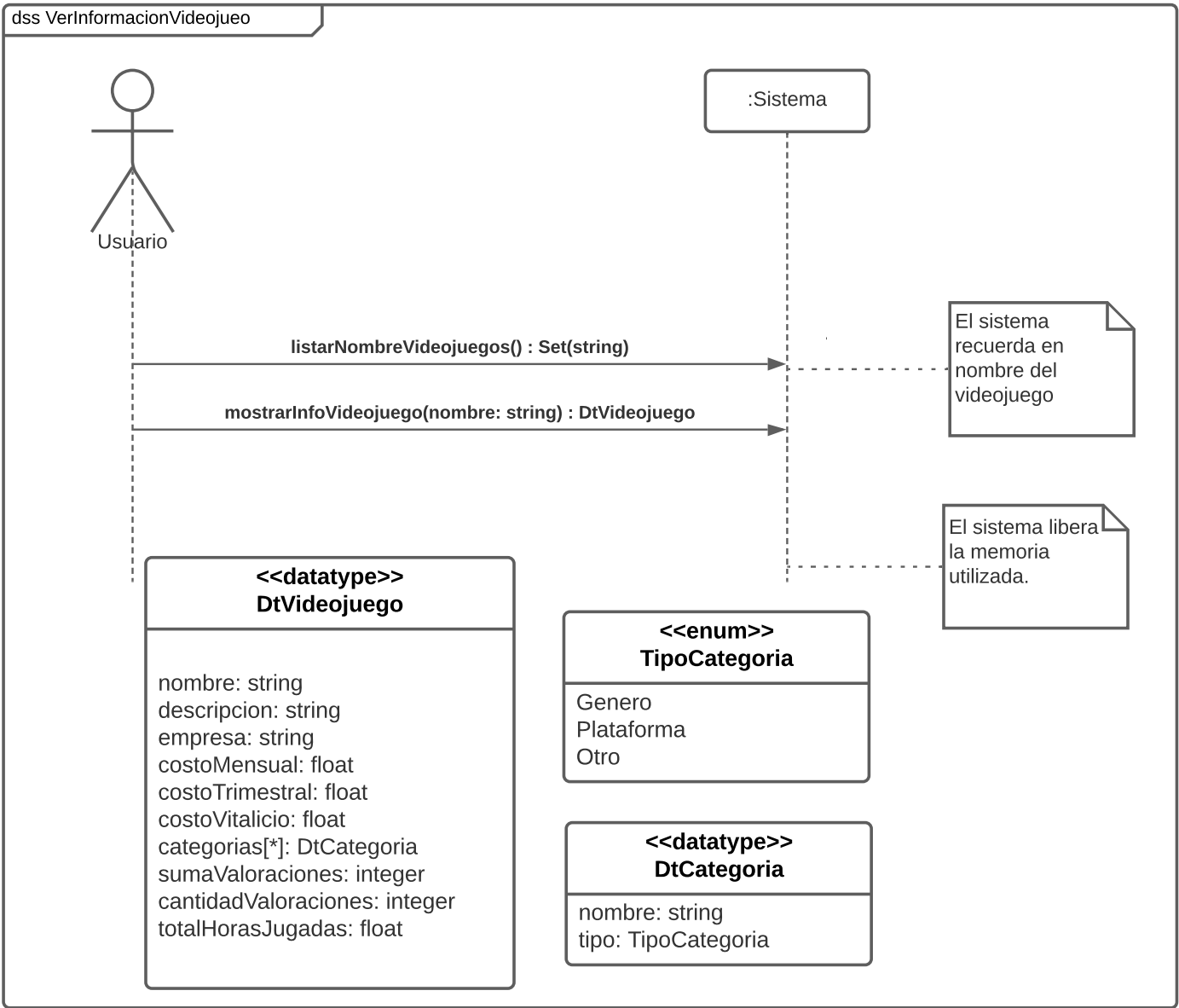
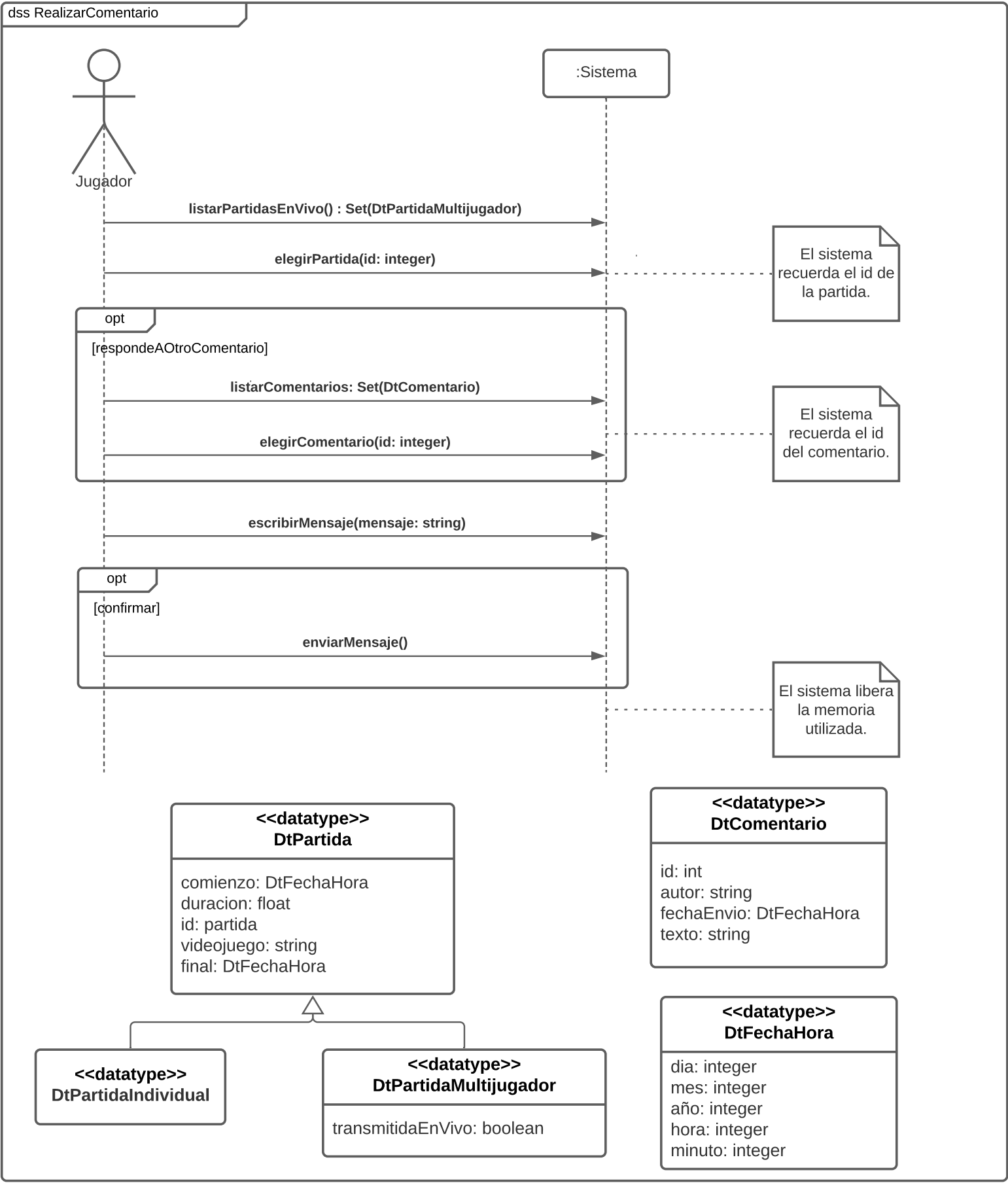


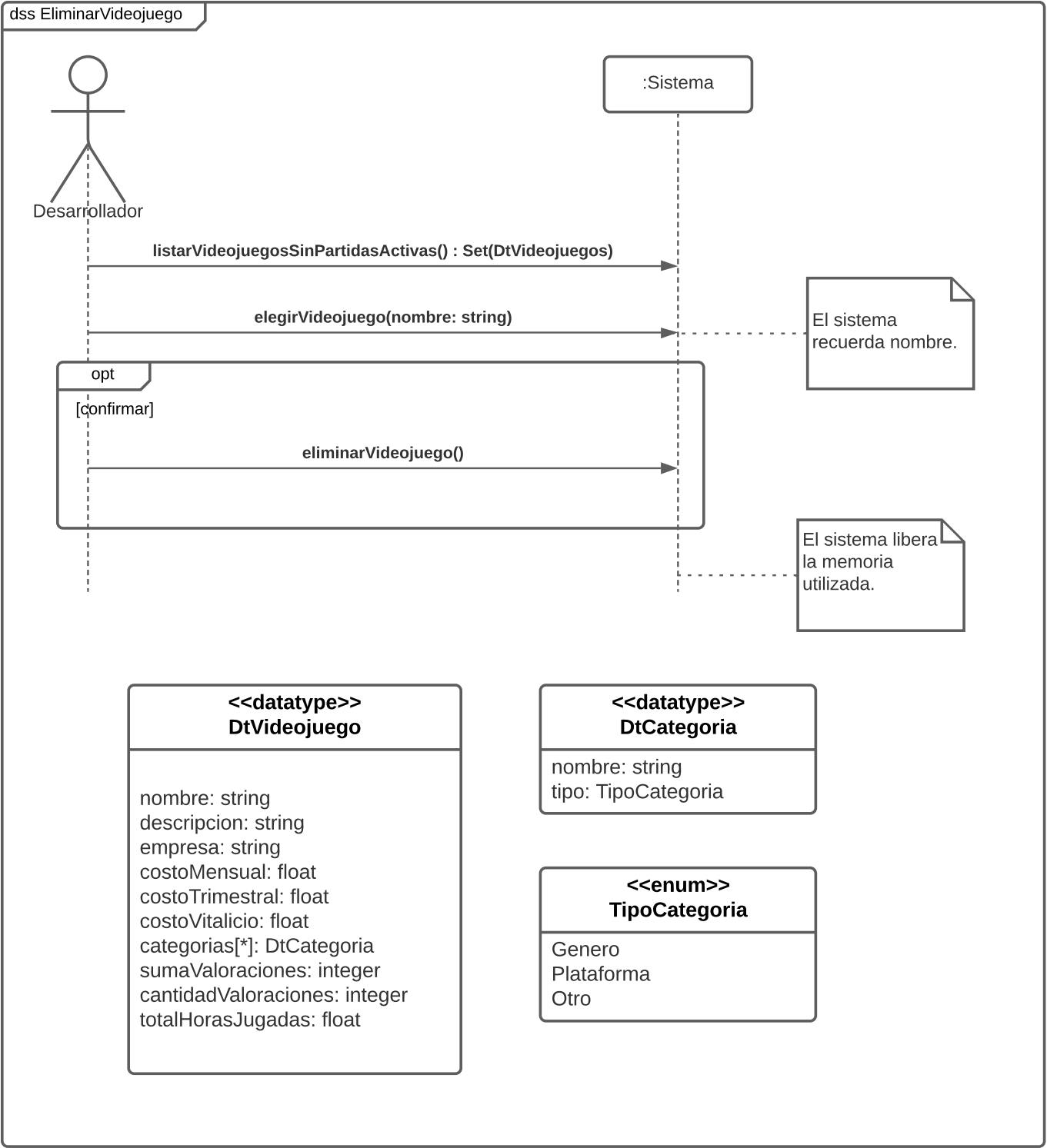
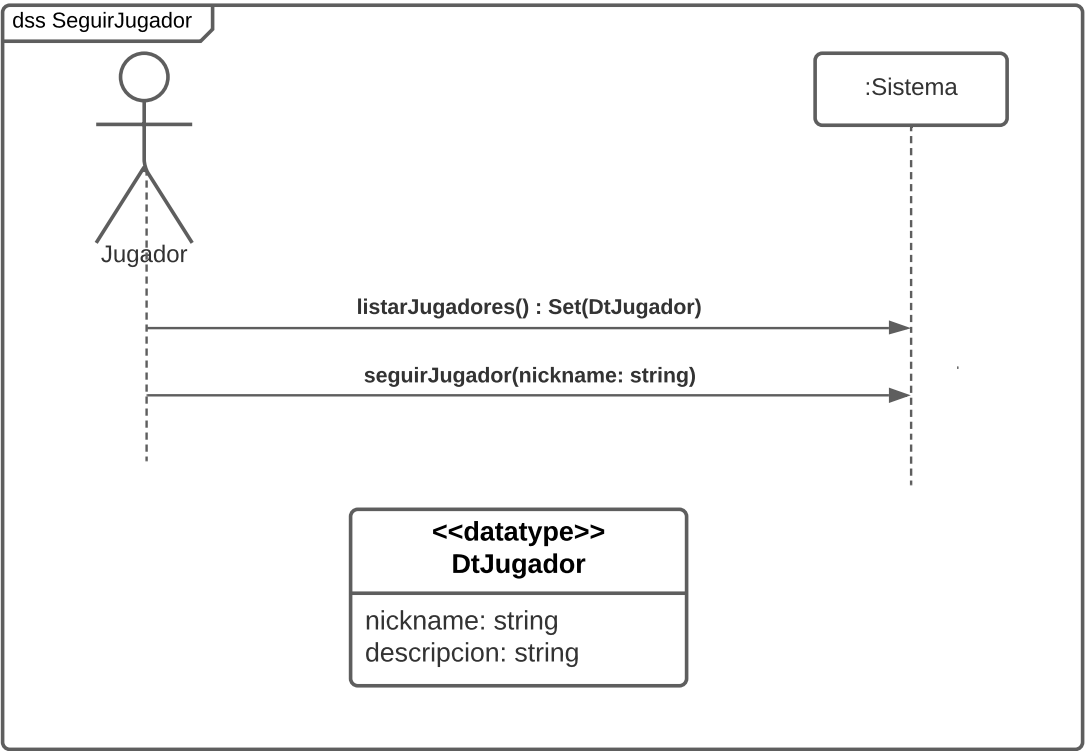












Contratos

Alta de Usuarios

Nombre	Ingresar datos usuario.
Operación	datosUsuario(email: string, contraseña: string)
Entrada	email y contraseña representan datos para registrar a un usuario en el sistema.
Salida	No hay.
Descripción	Almacena en memoria los datos del nuevo usuario que se registrará en el sistema.
Excepciones	Si no se cumple Pre1 se lanza la excepción ExUsuarioYaExiste.

Precondiciones y Postcondiciones	
Pre1:	No existe en el sistema una instancia de Usuario con su atributo email igual al email ingresado.
Post1:	Se almacenó en memoria del sistema el email y contraseña ingresados.

Nombre	Ingresar datos jugador.
Operación	datosJugador(nickname: string, descripción: string): boolean
Entrada	nickname y descripción representan datos para registrar un jugador en el sistema.
Salida	Retorna un boolean que representa si existe una instancia de jugador con ese nickname en el sistema.
Descripción	Almacena en memoria los datos del nuevo jugador que se ingresará en el sistema.
Excepciones	No hay.

Precondiciones y Postcondiciones
Post1: Se almacenó en memoria del sistema el nickname y descripción ingresados.

Nombre	Ingresar datos desarrollador.
Operación	datosDesarrollador(empresa: string)
Entrada	empresa representa un dato para registrar un desarrollador.
Salida	No hay.
Descripción	Almacena en memoria los datos del nuevo desarrollador que se ingresará en el sistema.
Excepciones	No hay.

Precondiciones y Postcondiciones
Post1: Se almacenó en memoria del sistema la empresa ingresada.

Nombre	Crear usuario.
Operación	crearUsuario()
Entrada	No hay.
Salida	No hay.
Descripción	Confirma el alta de usuario recordado por el sistema.
Excepciones	No hay.

Precondiciones y Postcondiciones
<p>Pre1: El sistema tiene almacenado en memoria los datos para registrar un Usuario (email y contraseña).</p> <p>Pre2: El sistema tiene almacenado en memoria los datos para registrar un Jugador (nickname y descripción), o Desarrollador (empresa).</p> <p>Post1: Se creó en el sistema una instancia de Jugador o Desarrollador (según corresponda) con sus atributos correspondientes que estaban almacenados en memoria.</p> <p>Post2: Se liberó la memoria del sistema.</p>

Iniciar partida

Nombre	Listar videojuegos con suscripcion activa.
Operación	listarVideojuegosConSuscripcionActiva():Set(DtVideojuego)
Entrada	No hay.
Salida	Set de instancias de la clase DtVideojuego.
Descripción	Lista los nombres de los videojuegos para los cuales el jugador tiene una suscripción activa.
Excepciones	Si no se cumple Pre1 se lanza excepción NoHaySesionIniciada.

Precondiciones y Postcondiciones	
Pre1: Existe en el sistema una instancia de Partida con su atributo id == id, final == NULL y que esté linkeado con la instancia de jugador con nickname igual al nickname del jugador logueado.	
Post1: Para la instancia de Partida guardada en el sistema con id == id, se setea su atributo final en la fecha y hora actuales del sistema, y se calcula y se setea su atributo duración.	

Nombre	Elegir videojuego.
Operación	elegirVideojuego(nombre: string)
Entrada	nombre de tipo string.
Salida	No hay.
Descripción	Almacena en memoria el nombre del videojuego del cual se iniciara partida.
Excepciones	Si no se cumple Pre1 se lanza excepción NoExisteVideojuego. Si no se cumple Pre2 se lanza excepción NoHaySuscripcionActivaParaVideojuego.

Precondiciones y Postcondiciones	
Pre1:	Debe existir una instancia de Videojuego cuyo atributo nombre == nombre.
Pre2:	Debe existir una asociación entre el jugador que inicio sesión y el videojuego cuyo atributo nombre == nombre. Dicha asociación de tipo Suscripción tiene que estar activa (suscripción.activa == true).
Post1:	Se almaceno en memoria del sistema el nombre del videojuego elegido.

Nombre	Listar partidas.
Operación	listarPartidas() : Set(DtPartida)
Entrada	No hay.
Salida	Set de instancias de la clase DtPartida.
Descripción	Lista en orden cronológico el historial de partidas individuales del jugador, mostrando para cada una su identificador, fecha y hora en la que se realizó y la duración.
Excepciones	Si no se cumple Pre1 se lanza excepción NoHaySesionIniciada.

Precondiciones y Postcondiciones	
Pre1:	El jugador debe haber iniciado sesión.
Post1:	Se retornó un conjunto de instancias de DtPartida asociados a la instancia de jugador con nickname == usuarioLogeado.nickname y donde el atributo final != NULL.

Nombre	Elegir partida.
Operación	elegirPartida(id: integer)
Entrada	id de tipo integer.
Salida	No hay.
Descripción	Almacena en memoria la selección de la partida que el jugador desea continuar.
Excepciones	Si no se cumple Pre1 se lanza excepción NoExistePartida.

Precondiciones y Postcondiciones	
Pre1:	Debe existir una instancia de la clase Partida cuyo atributo id == id.
Post1:	Se almaceno en memoria del sistema el id para la partida elegida.

Nombre	Transmitida en vivo.
Operación	transmitidaEnVivo(estado: boolean)
Entrada	estado de tipo boolean.
Salida	No hay.
Descripción	Almacena en memoria si la partida debe de ser transmitida en vivo o no.
Excepciones	No hay.

Precondiciones y Postcondiciones	
Post1: Se almaceno en memoria del sistema si la partida a iniciar debe de ser transmitida en vivo o no.	

Nombre	Listar nicknames de jugadores.
Operación	listarNicknames() : Set(string)
Entrada	No hay.
Salida	Set de string donde cada uno representa el nickname de un jugador que cuenta con una suscripción activa al videojuego elegido.
Descripción	Lista los nicknames de los demás jugadores que cuentan con una suscripción activa al videojuego elegido.
Excepciones	No hay.

Precondiciones y Postcondiciones	
Post1: Se retornó un conjunto de strings que representan los nicknames de todos aquellos jugadores asociaciados al videojuego cuyo nombre == nombre (memoria) y donde esta asociación Suscripción tiene el atributo activa == true.	

Nombre	Ingresar nicknames de jugadores.
Operación	ingresarNicknames(nicknames : Set(string))
Entrada	Set de elementos de tipo string donde cada uno representa un nickname de un jugador.
Salida	No hay.
Descripción	Almacena en memoria los nicknames de los jugadores que se van unir a la partida a ser creada.
Excepciones	Si no se cumple Pre1 se lanza excepción ExistenNicknamesSinJugadorAsociado. Si no se cumple Pre2 se lanza excepción ExistenJugadoresSinSuscripcionActiva.

Precondiciones y Postcondiciones	
Pre1: Para cada uno de los nicknames ingresados en el set debe existir una instancia de jugador cuyo atributo nickname coincida con el nickname ingresado en el set. Pre2: Cada jugador cuyo nickname coincide con el nickname ingresado, debe de estar asociado al videjuego cuyo atributo nombre == nombre (memoria) y donde la asociación Suscripción tiene el atributo activa == true.	
Post1: Se almaceno en memoria del sistema los nicknames de los jugadores que van a unirse a la partida a ser creada.	

Nombre	Confirmar creación de partida.
Operación	confirmarInicioPartida()
Entrada	No hay.
Salida	No hay.
Descripción	Confirma la creación de la partida con los datos almacenados en memoria del sistema.
Excepciones	No hay.

Precondiciones y Postcondiciones
<p>Pre1: El sistema tiene almacenado en memoria los datos para iniciar una partida individual o multijugador.</p> <p>Post1: Se inicio en el sistema una instancia de Partida con sus atributos correspondientes que estaban almacenados en memoria.</p> <p>Post2: Se creo un link entre la instancia creada de Partida y la instancia de jugador cuyo atributo nickname == usuarioLogeado.nickname.</p> <p>Post3: Se liberara la memoria del sistema asociada a los datos de la partida.</p>

Publicar Videojuego

Nombre	Ingresar datos del Videjuego a publicar.
Operación	ingresarDatosVideojuego(nombre: string, descripcion: string, costoMensual: float, costoTrimestral: float, costoAnual: float, costoVitalicio: float)
Entrada	nombre, descripción y el costo para cada tipo de suscripción (mensual, trimestral, anual y vitalicia).
Salida	No hay.
Descripción	Ingresa los datos del nuevo videojuego que se ingresará al sistema.
Excepciones	Si no se cumple Pre1 se lanza la excepción ExElDesarroladorNoInicioSesion. Si el no se cumple Pre2 se lanza la excepcion NobreYaExiste.

Precondiciones y Postcondiciones	
Pre1:	El desarrollador debe haber iniciado sesión.
Pre2:	El nombre ingresado para el videojuego no debe existir en el sistema.
Post1	Se almacena en la memoria del sistema los datos ingresados.

Nombre	Listar categoría de los videojuegos.
Operación	listarCategorias() : Set(DtCategoria)
Entrada	No hay.
Salida	Set de instancias de la clase DtCategoria.
Descripción	Lista las categorías de videojuegos(mostrando su nombre y descripción).
Excepciones	No hay.

Precondiciones y Postcondiciones	
Post1:	Se construyo y retorno conjunto de instancias de DtCategorias con la información de las categorías existentes en el sistema.

Nombre	Seleccionar loa géneros del videojuego.
Operación	agregarGenero(nombre: string)
Entrada	nombre del genero a asignarle al videojuego, el genero debe ser uno de los que se encuentra en las categorías del sistema .
Salida	No hay.
Descripción	Asocia el nombre ingresado como genero a los datos del videojuego.
Excepciones	Si no se cumple Pre1 lanza la excepción ExNombreNoValido.

Precondiciones y Postcondiciones	
Pre1:	Existe en el sistema una instancia de DtCategoria con su atributo nombre == nombre y TipoCategoria == Genero.
Post1:	El sistema almacena en memoria nombre del genero elegido.

Nombre	Seleccionar el plataforma del videojuego.
Operación	agregarPlataforma(nombre: string)
Entrada	nombre del plataforma a asignarle al videojuego algunas, de las que existe en el sistema.
Salida	No hay.
Descripción	Asocia el nombre ingresado como plataforma a los datos del videojuego.
Excepciones	Si no se cumple Pre1 lanza la excepcion NombreNoValido.

Precondiciones y Postcondiciones	
Pre1: Existe en el sistema una instancia de DtCategoria con su atributo nombre == nombre y TipoCategoria == Plataforma.	
Post1: El sistema almacena en memoria nombre del la plataforma elegida.	

Nombre	Seleccionar otras categorías para el videojuego.
Operación	agregarCategoria(nombre: string)
Entrada	nombre de la categoría a asignarle al videojuego, algunas de las que existe en el sistema.
Salida	No hay.
Descripción	Asocia un plataforma del sistema al los datos del videojuego en la memoria del sistema.
Excepciones	No hay.

Precondiciones y Postcondiciones	
Pre1: Existe en la memoria nombre, descripción, costoMensual, costoTrimestral, costoAnual, costoVitalicio.	
Pre2: Existe en la memoria el nombre de un plataforma.	

Nombre	Mostrar los datos del videojuego a registrar
Operación	MostrarDatosVideojuego()
Entrada	No hay.
Salida	No hay.
Descripción	Muestra la datos ingresados para el videojuego.
Excepciones	No hay.

Precondiciones y Postcondiciones	
Post1: Se presenta los datos almacenados y asociados a videojuego.	

Nombre	Crear Videojuego.
Operación	CrearVideojuego()
Entrada	No hay.
Salida	No hay.
Descripción	Crea el videojuego con los datos almacenado por el por el sistema.
Excepciones	Si no se cumple la Pre1 o Pre2 lanza la excepción DatosNoValidos.

Precondiciones y Postcondiciones	
Pre1:	El sistema tiene almacenado en la memoria nombre, descripción, costoMensual, costoTrimestral, costoAnual y costoVitalicio.
Pre2:	Existe en memoria al menos una categoría de genero y de plataforma asociadas al videojuego a crear.
Post1:	Se crea una instancia de Videojuego con los valores en su tipo y sus atributos correspondientes a los almacenados en memoria.
Post2:	Se creo un link entre la instancia creada de Partida y la instancia de jugador cuyo atributo nickname == usuarioLogeado.nickname.
Post3:	Para una partida multijugador, se crearon links entre la instancia de Partida creada y las instancias de jugador que estaban asociadas a los nicknames almacenados en el sistema.
Post4:	Se liberó la memoria del sistema.

Finalizar Partida

Nombre	Listar partidas no finalizadas.
Operación	listarPartidasNoFinalizadas() : Set(DtPartida)
Entrada	No hay.
Salida	Un conjunto de instancias de DtPartida que representa las partidas no finalizadas por el jugador.
Descripción	Retorna las partidas no finalizadas iniciadas por un determinado jugador.
Excepciones	No hay.

Precondiciones y Postcondiciones	
Pre1:	El usuario debe haber iniciado sesión.
Post1:	Se retornó un conjunto de instancias de DtPartida asociados a la instancia de jugador con nickname == usuarioLogeado.nickname y cuyo atributo final == NULL.

Nombre	Finalizar partida
Operación	finalizarPartida(id: integer)
Entrada	id representa el identificador de una de las partidas que existen en el sistema
Salida	No hay
Descripción	Marca como finalizada la partida con identificador id existente en el sistema y registra la duración de la misma
Excepciones	Si no se cumple Pre1 se lanza la excepción ExPartidaNoValida

Precondiciones y Postcondiciones
<p>Pre1: Existe en el sistema una instancia de Partida con su atributo id == id, final == NULL y que esté linkeado con la instancia de jugador con nickname igual al nickname del jugador logueado.</p> <p>Post1: Para la instancia de Partida guardada en el sistema con id == id, se setea su atributo final en la fecha y hora actuales del sistema, y se calcula y se setea su atributo duración.</p>

Cambios realizados en el Modelo Dominio.

I Atributo "id" en Partida.

Se agrega el atributo "id" al concepto Partida, ya que en varios casos de uso se hace referencia a que existe un identificador numérico para cada partida.

II Atributo "duración" y "final" pasan de PartidaMultijugador a Partida.

Los atributos "final" y "duración" que eran específicos para el concepto PartidaMultijugador, pasan a estar en Partida, ya que se hace referencia a que también importa saber la duración para las instancias de PartidaIndividual.

III Atributos "costoTrimestral" y "costoAnual" en Videojuego.

Se agregan en el concepto Videojuego los atributos "costoTrimestral" y "costoAnual". La razón de este cambio es por un tema de interpretación, ya que en el **Laboratorio 1**, no estaba totalmente claro cuál era el costo para las suscripciones temporales trimestrales y anuales, y se asumió que estas llevarían un cargo mensual con el costo del atributo "costoMensual".

Diagrama de modelo de dominio

