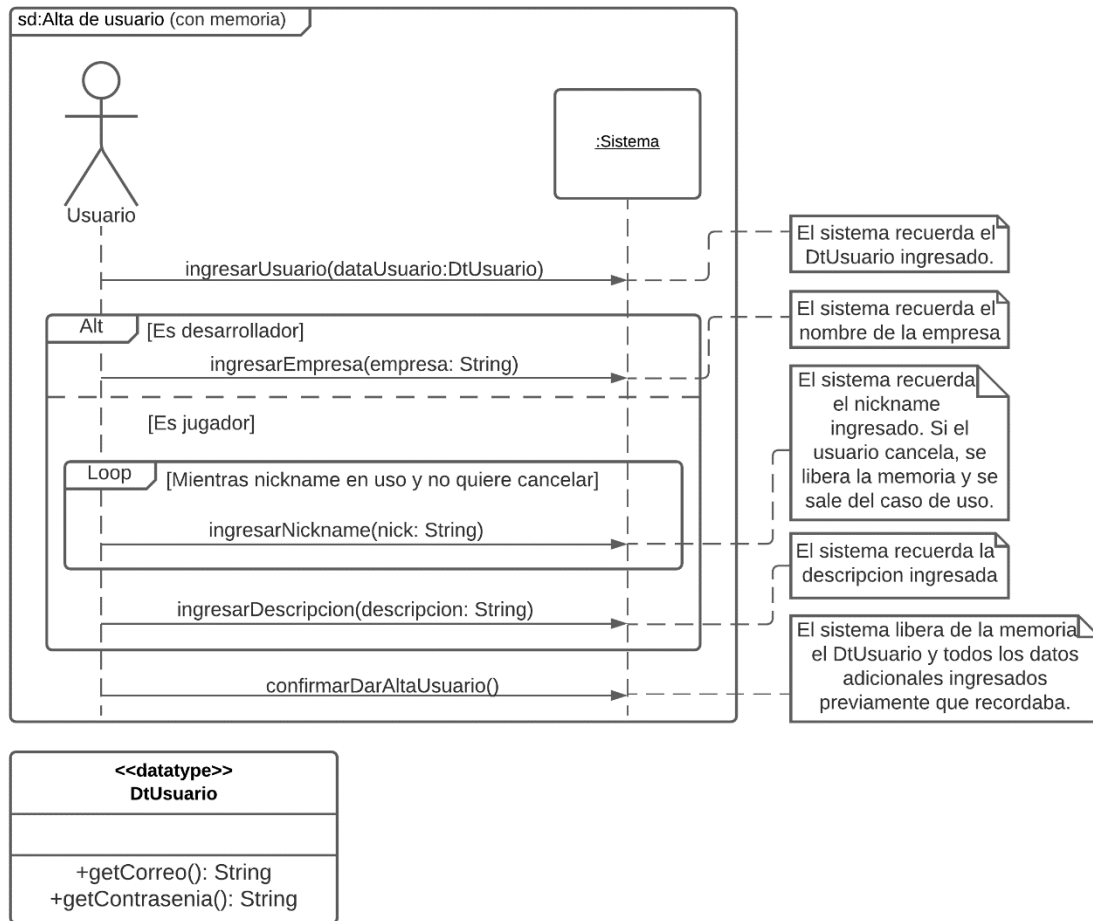


Alta de usuario:

Diagramas de Secuencia del Sistema



Contratos

Nombre	Ingresar usuario.
Operación	<code>ingresarUsuario(dataUsuario: DtUsuario)</code>
Entrada	dataUsuario representa la información asociada a DtUsuario que contiene al correo y contraseña del usuario, además del nickname y descripción en caso de ser jugador o la empresa en caso de ser desarrollador.
Salida	No hay.
Descripción	El usuario ingresa al sistema su información para darse de alta.

Precondiciones y Postcondiciones

Pre: No existe una instancia de Usuario en el sistema tal que el atributo `correo==dataUsuario.correo`.

Post: El sistema almacena en la memoria la instancia de DtUsuario ingresada, con los atributos de tipo String usuario y contraseña.

Nombre	Ingresar el nombre de la empresa.
Operación	<code>ingresarEmpresa(empresa: String)</code>
Entrada	"empresa" representa el nombre de la empresa del desarrollador a dar de alta.
Salida	No hay.
Descripción	El usuario ingresa el nombre de su empresa.

Precondiciones y Postcondiciones

Post: El sistema almacena en la memoria una instancia de String con el nombre de la empresa ingresada.

Nombre	Ingresar un nickname.
Operación	<code>ingresarNickname(nick: String)</code>
Entrada	"nickname" es el nickname del usuario jugador a dar de alta.
Salida	No hay.
Descripción	El jugador ingresa un nickname para su usuario.

Precondiciones y Postcondiciones

Pre: No existe una instancia de jugador en el sistema tal que el atributo `nickname == nick`.

Post: El sistema almacena en la memoria una instancia de String con el nickname ingresado.

Nombre	Ingresar una descripción.
Operación	<code>ingresarDescripcion(descripción: String)</code>
Entrada	"descripción" es la descripción del usuario del jugador a dar de alta.
Salida	No hay.
Descripción	El jugador ingresa la descripción para su usuario.

Precondiciones y Postcondiciones

Post: El sistema almacena en la memoria una instancia de String con la descripción ingresada.

Nombre	Finalizar creación de usuario.
Operación	<code>confirmarDarAltaUsuario()</code>
Entrada	No hay.
Salida	No hay.
Descripción	Se confirma la creación del usuario devolviendo la información ingresada previamente.

Precondiciones y Postcondiciones

Pre1: Existen en el sistema los datos necesarios de tipo String email y contraseña, para instanciar al Usuario.

Pre2: En el caso de dar de alta un Jugador existen en el sistema los datos necesarios de tipo String nickname y descripción.

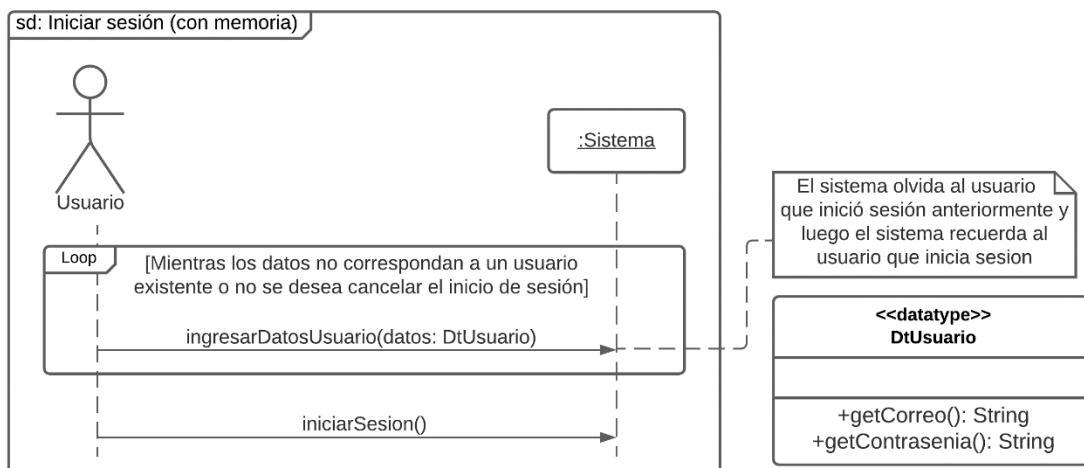
Pre3: En el caso de dar de alta un Desarrollador existe en el sistema una instancia de tipo String con el nombre de la empresa.

Post1: Se crea una instancia de Usuario (de tipo Jugador o Desarrollador dependiendo de la selección del usuario) con sus respectivos datos ingresados anteriormente que fueron recordados por el sistema.

Post2: Se libera todas las instancias almacenadas en la memoria del sistema para construir la respectiva instancia de jugador o desarrollador.

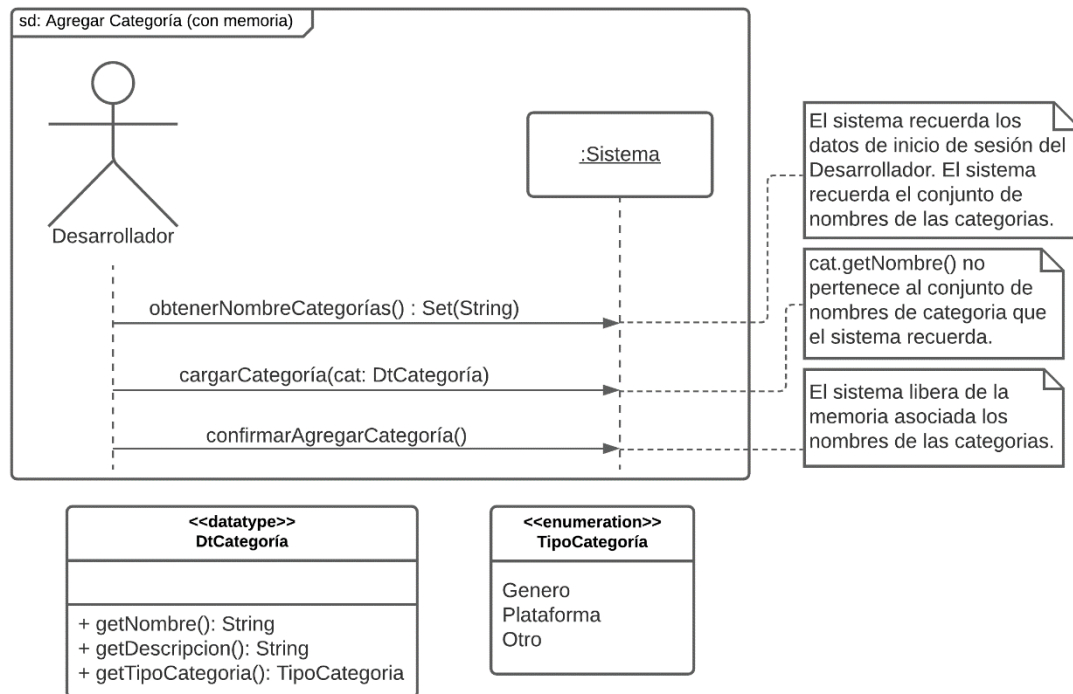
Iniciar sesión:

Diagramas de Secuencia del Sistema



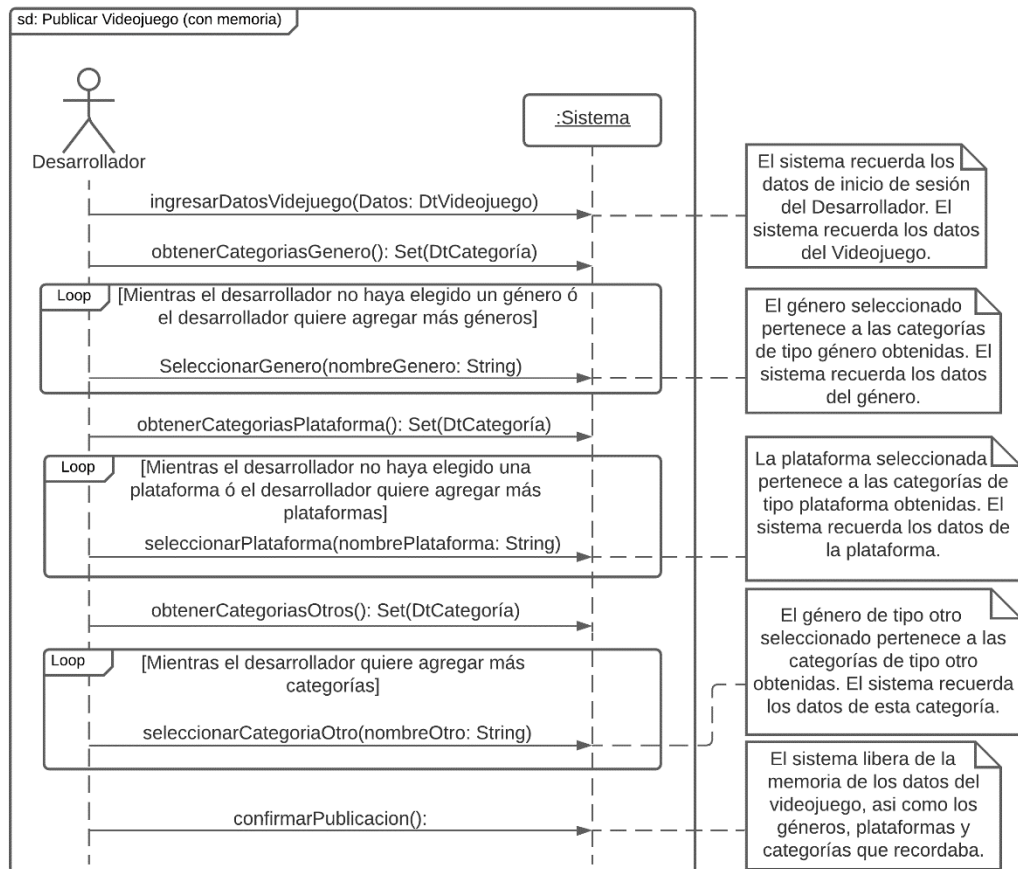
Agregar categoría:

Diagramas de Secuencia del Sistema



Publicar videojuego:

Diagramas de Secuencia del Sistema



<<datatype>> DtVideojuego
+getNombre(): String +getDescripcion(): String +getPrecios(): DtPrecios

<<datatype>> DtPrecios
+getMensual(): float +getTrimestral(): float +getAnual(): float +getVitalicia(): float

<<datatype>> DtCategoría
+getNombre(): String +getDescripcion(): String +getTipoCategoría(): TipoCategoría

<<enumeration>> TipoCategoría
Genero Plataforma Otro

Contratos

Nombre	Ingresar datos del videojuego.
Operación	<code>ingresarDatosVideojuego(Datos: DtVideojuego)</code>
Entrada	Datos representa la información asociada a DtVideojuego que contiene el nombre, descripción y el costo para cada tipo de suscripción (mensual, trimestral, anual y vitalicia).
Salida	No hay.
Descripción	El desarrollador ingresa al sistema los datos del videojuego a publicar.

Precondiciones y Postcondiciones	
<p>Pre1: Existe en la memoria del sistema una instancia del Desarrollador que inicio sesión.</p> <p>Pre2: No existe en la memoria del sistema una instancia de Videojuego tal que el atributo <code>nombre==DtVideojuego.getNombre()</code>.</p> <p>Post: El sistema almacena en la memoria la instancia de DtVideojuego ingresada, con los datos de nombre, descripción y costos de suscripción del videojuego.</p>	

Nombre	Obtener las categorías de tipo género.
Operación	<code>obtenerCategoriasGenero(): Set(DtCategoría)</code>
Entrada	No hay.
Salida	Un conjunto de instancias de tipo DtCategoría que representa los datos de las categorías de tipo género.
Descripción	Retorna un conjunto de DtCategoría con la información de las categorías de tipo género.

Precondiciones y Postcondiciones	
<p>Post: Se construye y retorna un conjunto de instancias de categorías de tipo género con la información de los nombres de los géneros registrados en el sistema. Este conjunto de Géneros se almacena en la memoria del sistema.</p>	

Nombre	Seleccionar género.
Operación	<code>seleccionarGenero(nombreGenero: String)</code>
Entrada	nombreGenero representa el nombre que identifica al género en el sistema.
Salida	No hay.
Descripción	Selecciona una categoría de tipo género para el videojuego.

Precondiciones y Postcondiciones	
Pre: Existe en la memoria del sistema un género==nombreGenero perteneciente al conjunto de DtCategoría retornado por <code>mostrarCategoriaGenero()</code> .	
Post: Se almacena en la memoria del sistema la instancia nombreGenero de tipo String.	

Nombre	Obtener las categorías de tipo plataforma.
Operación	<code>obtenerCategoriasPlataforma(): Set(DtCategoría)</code>
Entrada	No hay.
Salida	Un conjunto de instancias de tipo DtCategoría que representa los datos de las categorías de tipo plataforma.
Descripción	Retorna un conjunto de DtCategoría con la información de las categorías de tipo plataforma.

Precondiciones y Postcondiciones	
Post: Se construye y retorna un conjunto de instancias de categorías de tipo plataforma con la información de los nombres de las plataformas registradas en el sistema. Este conjunto de Plataformas se almacena en la memoria del sistema.	

Nombre	Seleccionar plataforma.
Operación	<code>seleccionarPlataforma(nombrePlataforma: String)</code>
Entrada	nombrePlataforma representa el nombre que identifica la plataforma en el sistema.
Salida	No hay.
Descripción	Selecciona una categoría de tipo plataforma para el videojuego.

Precondiciones y Postcondiciones	
Pre: Existe en la memoria del sistema una plataforma==nombrePlataforma perteneciente al conjunto de DtCategoría retornado por <code>mostrarCategoriaPlataforma()</code> .	
Post: Se almacena en la memoria del sistema la instancia nombrePlataforma de tipo String.	

Nombre	Obtener las categorías de tipo otros.
Operación	<code>obtenerCategoriasOtros(): Set(DtCategoría)</code>
Entrada	No hay.
Salida	Un conjunto de instancias de tipo DtCategoría que representa los nombres de las categorías de tipo otro.
Descripción	Retorna un conjunto de DtCategoría con la información de las categorías de tipo otro.

Precondiciones y Postcondiciones

Post: Se construye y retorna un conjunto de instancias de categorías de tipo otro con la información de los nombres de las categorías de tipo otro registrados en el sistema. Este conjunto de Strings se almacena en la memoria del sistema.

Nombre	Seleccionar categoría otro.
Operación	<code>seleccionarCategoríaOtro(nombreOtro: String)</code>
Entrada	nombreOtro representa el nombre que identifica la categoría de tipo Otro en el sistema.
Salida	No hay.
Descripción	Selecciona una categoría de tipo otros para el videojuego.

Precondiciones y Postcondiciones

Pre: Existe en la memoria del sistema una CategoríaOtro==nombreOtro perteneciente al conjunto de DtCategoría retornado por `mostrarCategoríaOtros()`.

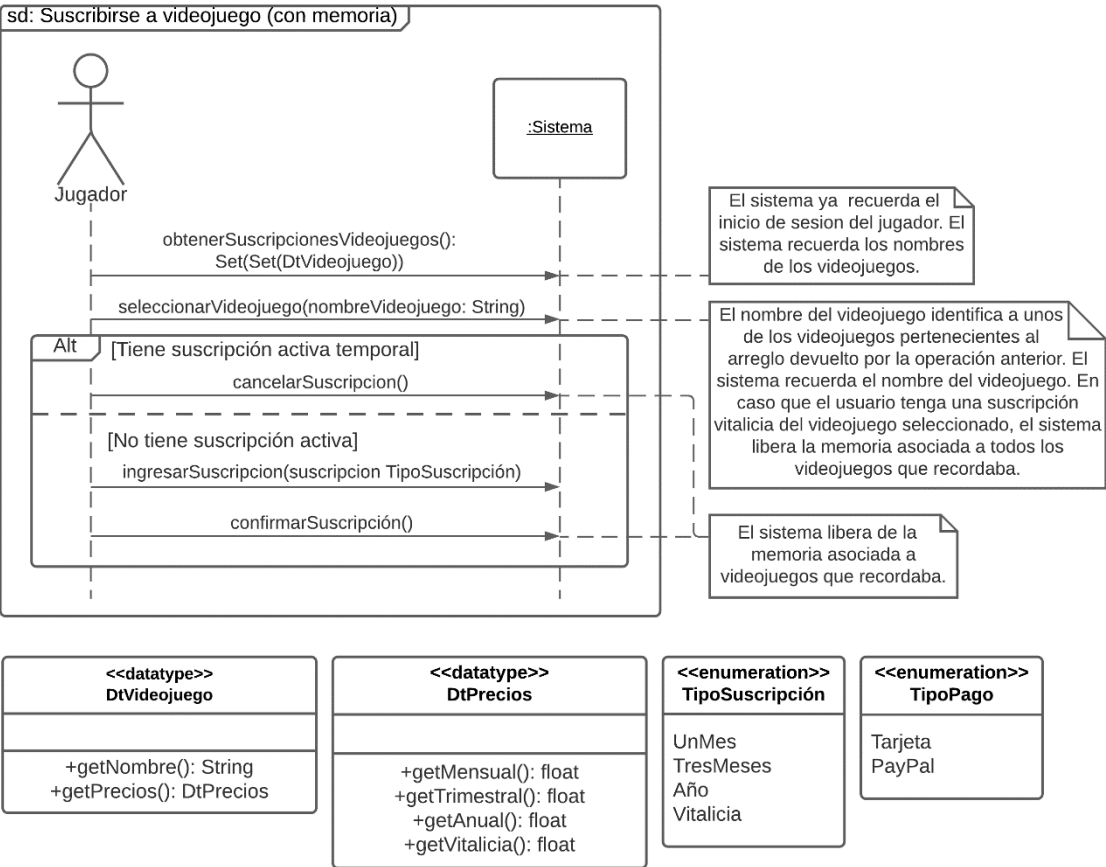
Post: Se almacena en la memoria del sistema la instancia nombreOtro de tipo String.

Nombre	Confirmar la publicación del videojuego.
Operación	<code>confirmarPublicacion()</code>
Entrada	No hay.
Salida	No hay.
Descripción	Confirma el alta de la publicación del videojuego, creando la instancia de partida correspondiente.

Precondiciones y Postcondiciones
<p>Pre1: El sistema tiene almacenado en su memoria los datos del desarrollador que lo publica.</p> <p>Pre2: El sistema tiene almacenado en la memoria el nombre, la descripción y los costos de cada tipo de suscripción del videojuego.</p> <p>Post1: En caso de que se deseó confirmar la publicación, se crea una instancia de Videojuego asociada con el Desarrollador que lo publicó y con cada una de las distintas Categorías seleccionadas (Género, Plataforma y Otro). Adicionalmente, a los atributos <code>totalHorasJugadas</code> y <code>puntaje</code> se les asigna el valor predeterminado '0' de tipo float (para ambos casos).</p> <p>Post2: Se libera todas las instancias almacenadas en la memoria del sistema para construir la respectiva instancia de videojuego.</p>

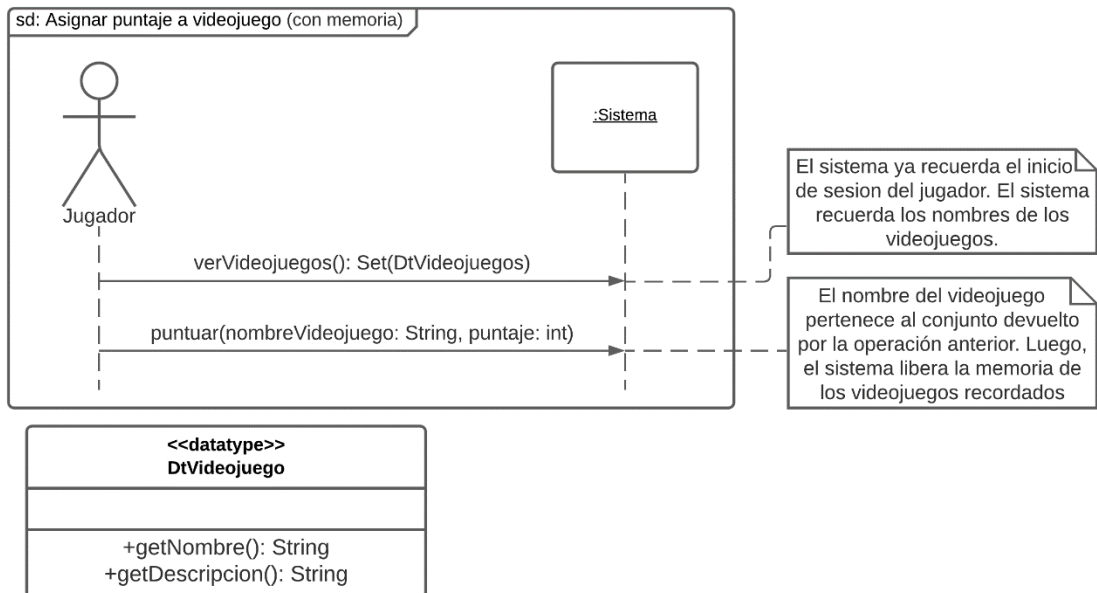
Suscribirse a videojuego:

Diagramas de Secuencia del Sistema



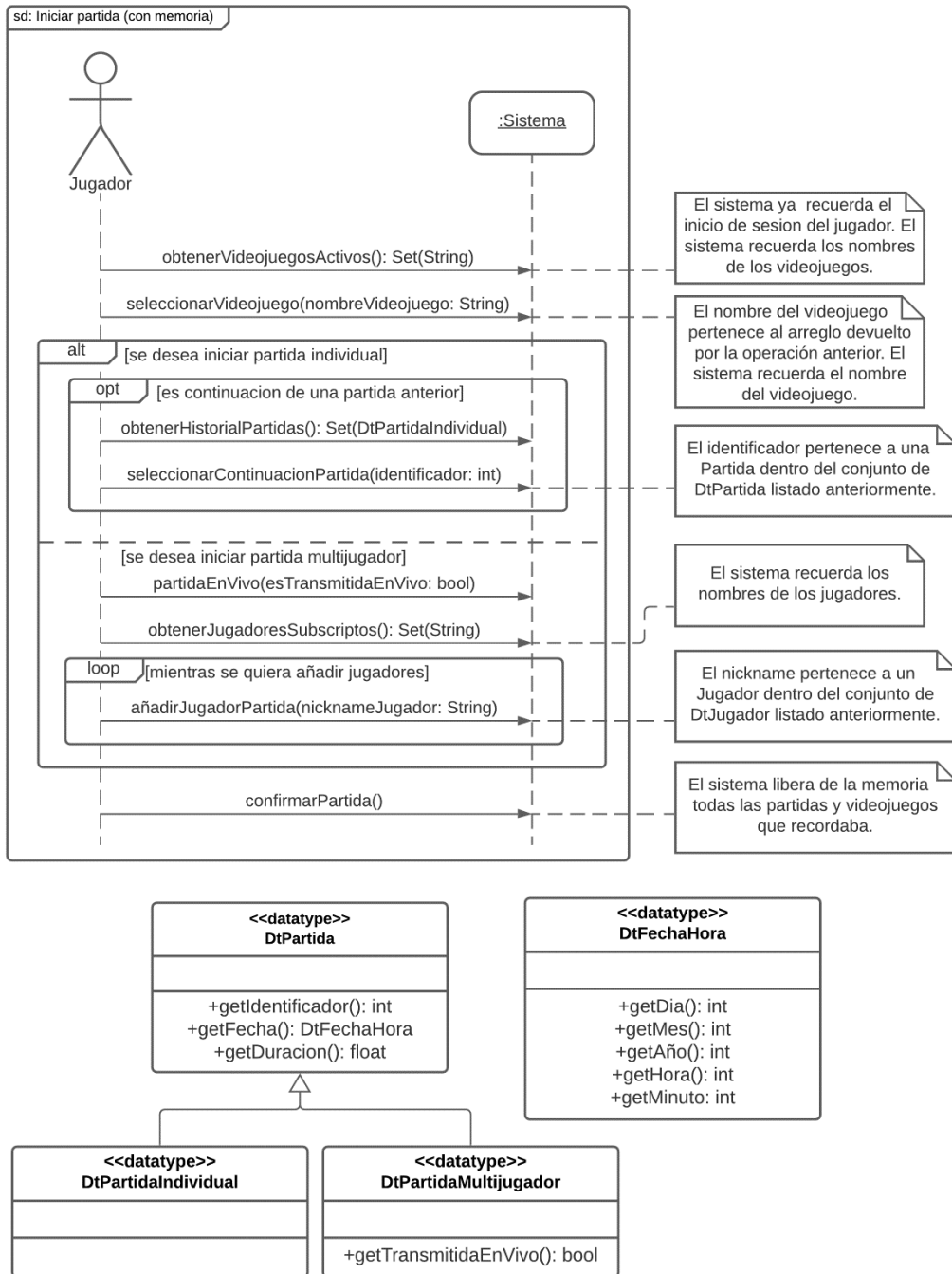
Asignar puntaje a videojuego:

Diagramas de Secuencia del Sistema



Iniciar partida:

Diagramas de Secuencia del Sistema



Contratos

Nombre	Obtener videojuegos con suscripción activa.
Operación	<code>obtenerVideojuegosActivos(): Set(String)</code>
Entrada	No hay.
Salida	Un conjunto de instancias de tipo String que representa los nombres de los videojuegos para los cuales el Jugador tiene una suscripción activa.
Descripción	Retorna un conjunto de Strings con los nombres de los videojuegos que presentan una suscripción activa para el Jugador.

Precondiciones y Postcondiciones	
Pre: Existe en la memoria del sistema una instancia del Jugador que inicio sesión.	
Post: Se construye y retorna un conjunto de instancias del tipo String con la información de los nombres de los Videojuegos, para los cuales el Jugador presenta una suscripción activa, existentes en el sistema. Este conjunto de Strings se almacena en la memoria del sistema.	

Nombre	Seleccionar videojuego.
Operación	<code>seleccionarVideojuego(nombreVideojuego: String)</code>
Entrada	nombreVideojuego representa el nombre que identifica al videojuego, en el sistema, al que se le inicia una partida.
Salida	No hay.
Descripción	Selecciona un videojuego para iniciar una partida.

Precondiciones y Postcondiciones	
Pre: Existe en la memoria del sistema un String==nombreVideojuego perteneciente al conjunto de Strings retornado por <code>obtenerVideojuegosActivos()</code> .	
Post: Se almacena en la memoria del sistema la instancia nombreVideojuego de tipo String.	

Nombre	Obtener historial de partidas del videojuego.
Operación	<code>obtenerHistorialPartidas(): Set(DtPartidaIndividual)</code>
Entrada	No hay.
Salida	Un conjunto de instancias de <code>DtPartidaIndividual</code> que representa las partidas iniciadas del videojuego anteriormente seleccionado.
Descripción	Retorna el conjunto de partidas iniciadas del videojuego seleccionado.

Precondiciones y Postcondiciones	
<p>Pre: Existe en la memoria del sistema un <code>String</code> con el nombre del videojuego seleccionado que se quiere obtener sus partidas.</p> <p>Post: Se construye y retorna un conjunto de instancias de <code>DtPartidaIndividual</code> con la información de las instancias de las partidas existentes en el sistema, pertenecientes al videojuego seleccionado que se encuentra en la memoria. Este conjunto de instancia de <code>DtPartidaIndividual</code> se almacena en la memoria del sistema.</p>	

Nombre	Seleccionar partida a continuar.
Operación	<code>seleccionarContinuacionPartida(identificador: int)</code>
Entrada	Identificador representa el nombre que identifica a la partida que se quiere seleccionar.
Salida	No hay.
Descripción	Selecciona una partida individual, tal que la partida a iniciar es continuación de ésta.

Precondiciones y Postcondiciones	
<p>Pre: Existe una instancia de <code>DtPartidaIndividual</code> perteneciente al conjunto de Partidas Individuales retornadas anteriormente tal que su atributo <code>id==identificador</code>.</p> <p>Post: Se almacena en la memoria del sistema la instancia <code>DtPartidaIndividual</code> cuyo <code>id==identificador</code>.</p>	

Nombre	Partida transmitida en vivo.
Operación	<code>partidaEnVivo(esTransmitidaEnVivo: bool)</code>
Entrada	<code>esTransmitidaEnVivo</code> indica que la partida a iniciar se transmitirá en vivo.
Salida	No hay.
Descripción	Indica si la partida a iniciar es transmitida en vivo, o no.

Precondiciones y Postcondiciones

Post: Se almacena en la memoria del sistema un booleano que indica si la partida se desea transmitir en vivo o no.

Nombre	Obtener jugadores suscriptos al videojuego.
Operación	<code>obtenerJugadoresSubscriptos(): Set(String)</code>
Entrada	No hay.
Salida	Un conjunto de instancias de tipo <code>String</code> que representan los nicknames de los jugadores suscriptos al videojuego seleccionado.
Descripción	Retorna un conjunto de <code>Strings</code> con los nombres de los jugadores que presentan una suscripción activa al videojuego seleccionado.

Precondiciones y Postcondiciones

Post: Se construye y retorna un conjunto de instancias de tipo `String` con la información de los nicknames de todos los Jugadores que presentan una suscripción activa al videojuego seleccionado. Este conjunto de `Strings` se almacena en la memoria del sistema.

Nombre	Añadir jugador a la partida.
Operación	<code>añadirJugadorPartida(nicknameJugador: String)</code>
Entrada	<code>nicknameJugador</code> representa el nickname que identifica al jugador, en el sistema, el cual se añadirá a la partida.
Salida	No hay.
Descripción	Añade un jugador a la partida multijugador a iniciar.

Precondiciones y Postcondiciones

Pre: Existe en la memoria del sistema un `String==nicknameJugador` perteneciente al conjunto de `Strings` retornado por `obtenerJugadoresSubscriptos()`.

Post: Se almacena en la memoria del sistema el `String nicknameJugador`

Nombre	Confirmar partida.
Operación	<code>confirmarPartida()</code>
Entrada	No hay.
Salida	No hay.
Descripción	Confirma el alta de la partida, creando la instancia de partida correspondiente.

Precondiciones y Postcondiciones

Pre1: El sistema tiene almacenado en su memoria: el videojuego al que pertenece y el jugador que la inicia.

Pre2: En caso de que se deseó confirmar una partida individual, el sistema tiene almacenado en la memoria el nombre de la partida de la cual es continuación.

Pre3: En caso de que se deseó confirmar una partida multijugador, el sistema tiene almacenado en la memoria: el booleano que indica si la partida es transmitida en vivo y los nombres de los jugadores que se desean añadir.

Post1: Se crea una instancia de Partida (de tipo individual o multijugador, dependiendo de la opción elegida por el usuario) asociada con el Jugador que inició sesión y el Videojuego seleccionado. Adicionalmente, a los atributos id y fecha se les asigna un identificador numérico autogenerated internamente por el sistema y la fecha y hora actual del mismo respectivamente.

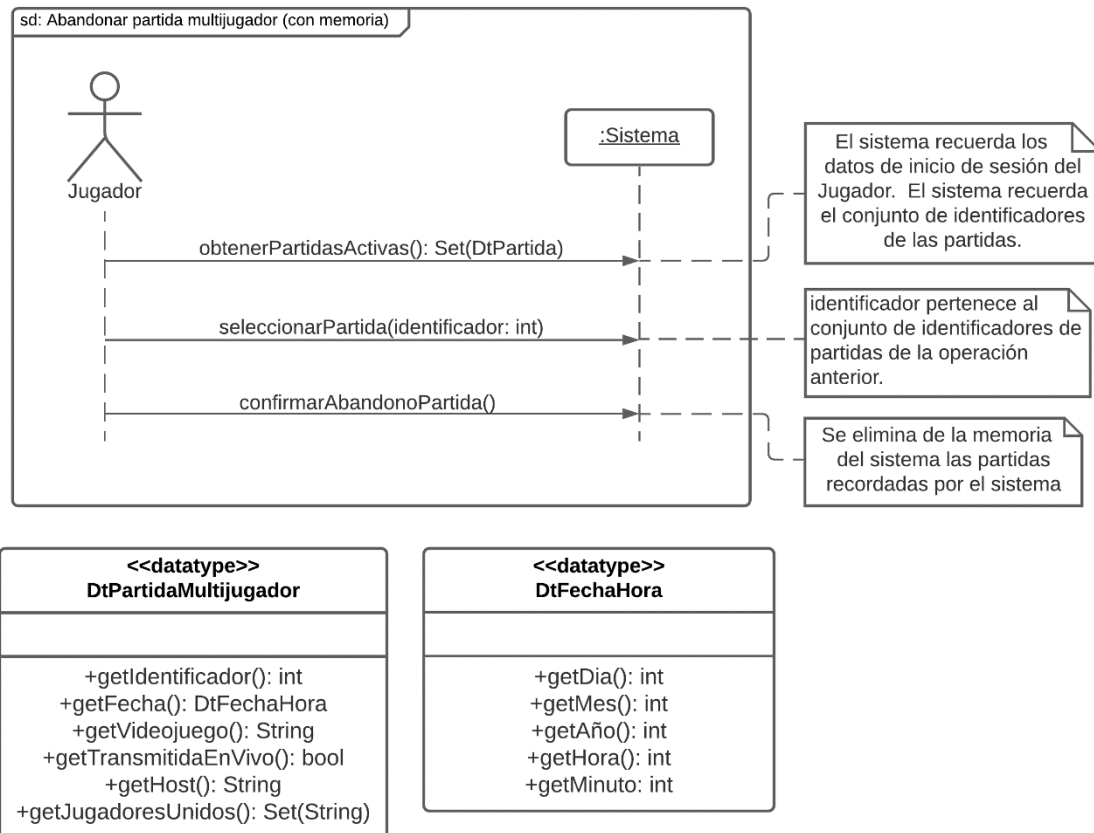
Post2: En caso de ser una partida individual que es continuación de otra partida, se asocia la misma con la partida anterior seleccionada.

Post3: En caso de ser una partida multijugador, el atributo `transmitidaEnVivo` tiene el valor del booleano que indica si la partida es transmitida en vivo. Además, se asocia la misma con todos los jugadores que se desean añadir.

Post4: Se libera la memoria del sistema.

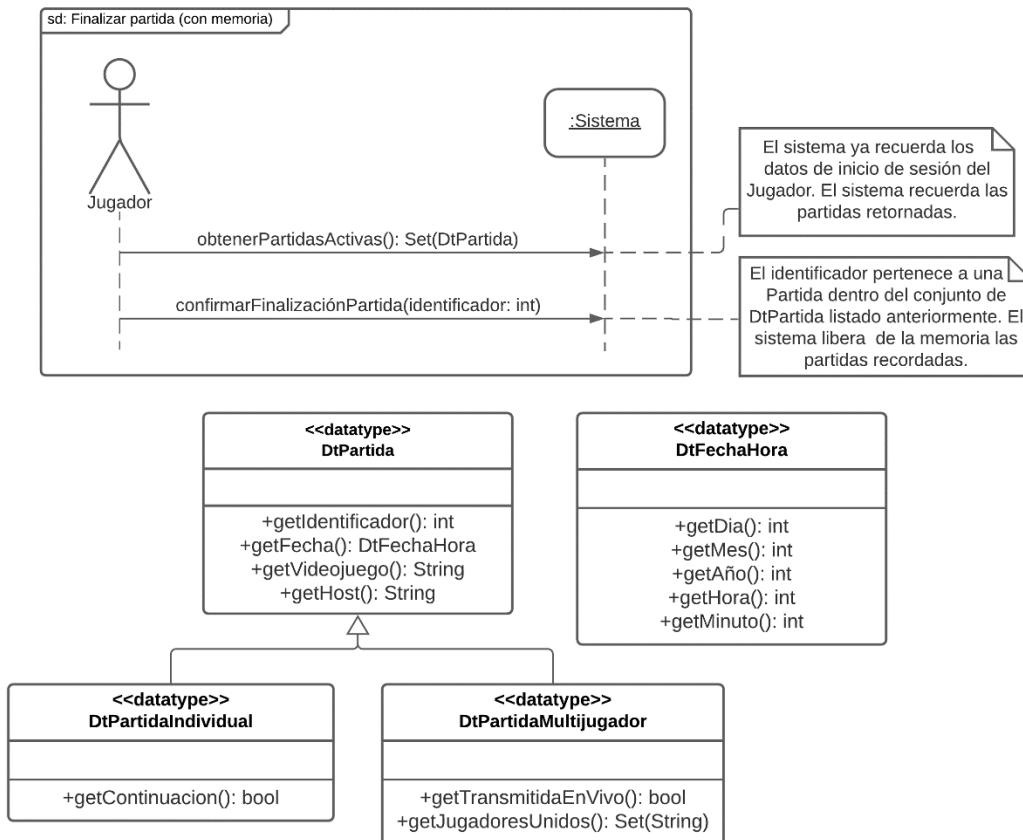
Abandonar partida multijugador:

Diagramas de Secuencia del Sistema



Finalizar partida:

Diagramas de Secuencia del Sistema



Contratos

Nombre	Obtener Partidas activas de un jugador.
Operación	<code>obtenerPartidasActivas(): Set(DtPartida)</code>
Entrada	No hay.
Salida	El sistema muestra las partidas activas del jugador.
Descripción	Retorna el conjunto de partidas activas iniciadas por un jugador.

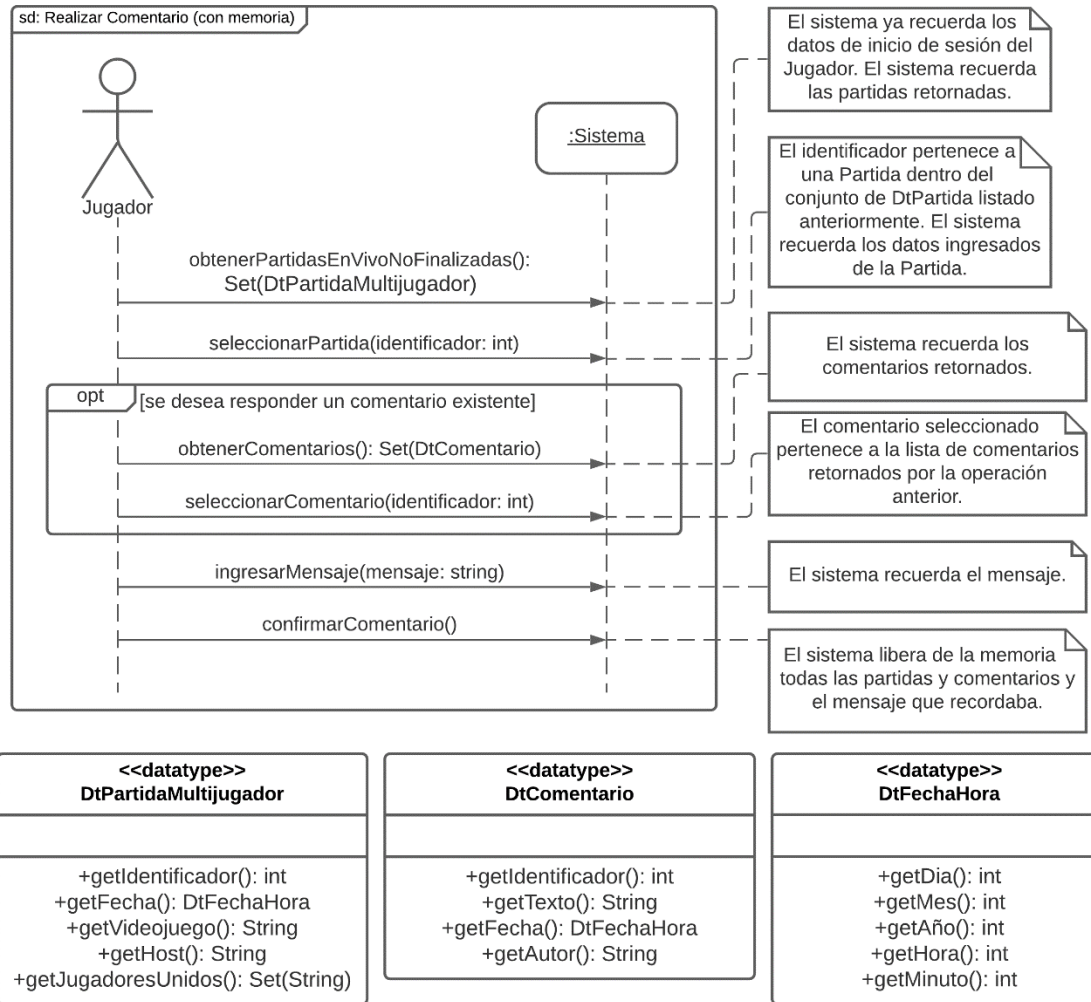
Precondiciones y Postcondiciones	
Pre: Existe en la memoria del sistema una instancia del Jugador que inicio sesión.	
Post: Se construye y retorna un conjunto de instancias de tipo <code>DtPartida</code> con la siguiente información: Identificador, fecha de inicio, videojuego correspondiente, si es una continuación de una partida (En el caso de ser de tipo individual), si es transmitida en vivo y los jugadores que se unieron a ella (Estas dos últimas entradas en el caso de ser partida multijugador). Las partidas retornadas son aquellas que fueron iniciadas por el jugador que inició sesión y, a su vez, se encuentran activas. Este conjunto de partidas se almacena en la memoria del sistema.	

Nombre	Finaliza una partida activa.
Operación	<code>confirmarFinalizaciónPartida(identificador: int)</code>
Entrada	Identificador representa el nombre que identifica a la partida que se quiere seleccionar.
Salida	No hay.
Descripción	Selecciona un videojuego para iniciar una partida.

Precondiciones y Postcondiciones	
Pre: Existe en la memoria del sistema una instancia de <code>DtPartida</code> tal que <code>id=identificador</code> perteneciente al conjunto de <code>DtPartidas</code> retornado por <code>obtenerPartidasActivas()</code> .	
Post1: La partida seleccionada por el identificador pasa a estar en estado "finalizada" (<code>DtPartida.getActiva()==false</code>).	
Post2: Se libera de la memoria del sistema las partidas recordadas.	

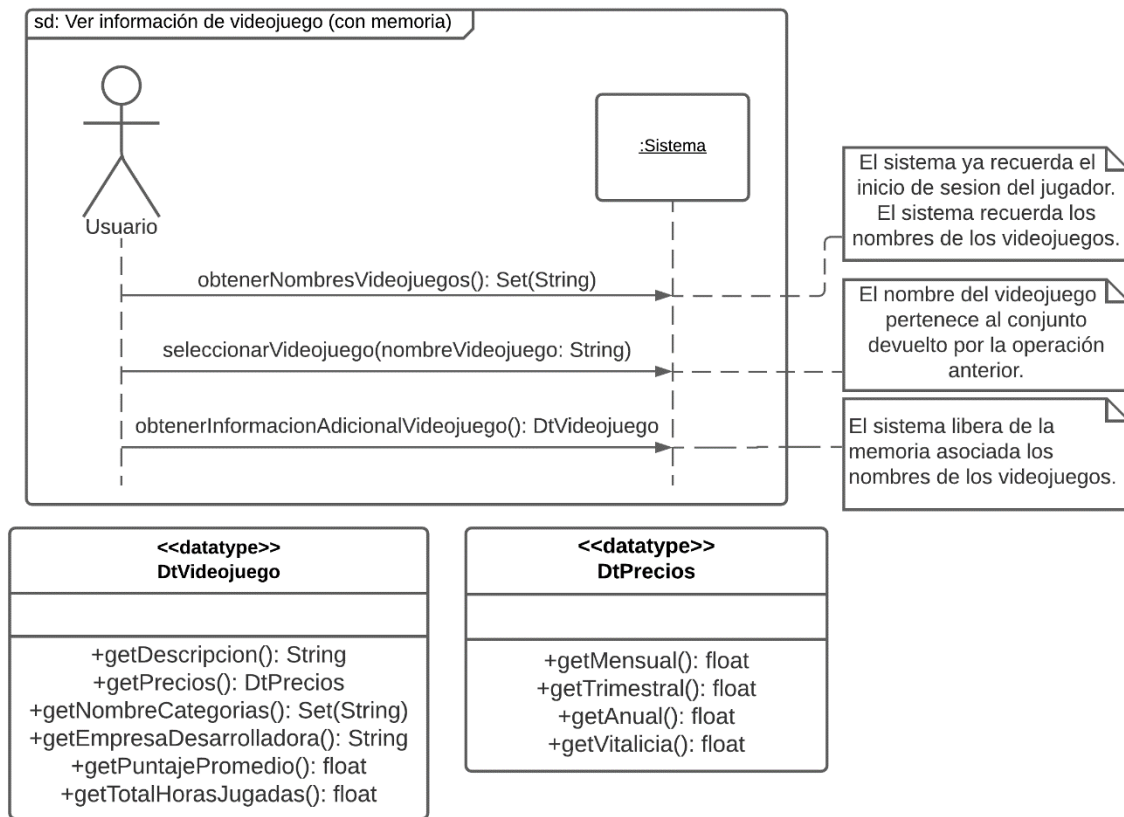
Realizar comentario:

Diagramas de Secuencia del Sistema



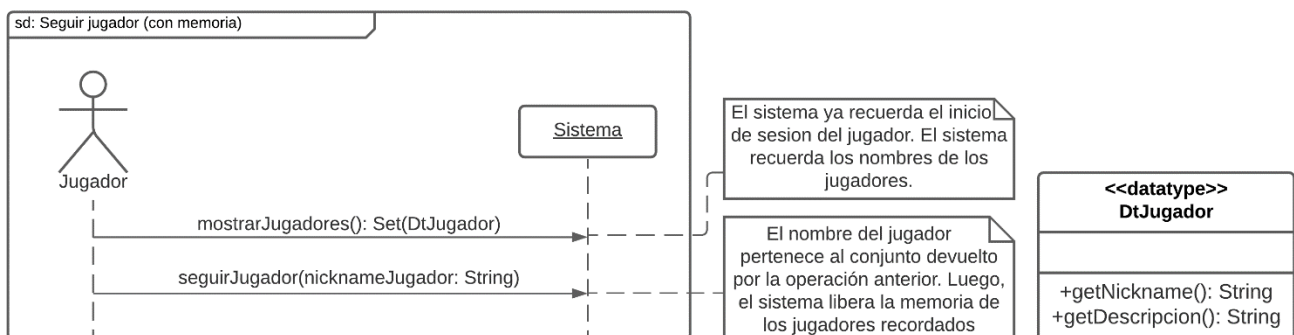
Ver información de videojuego:

Diagramas de Secuencia del Sistema



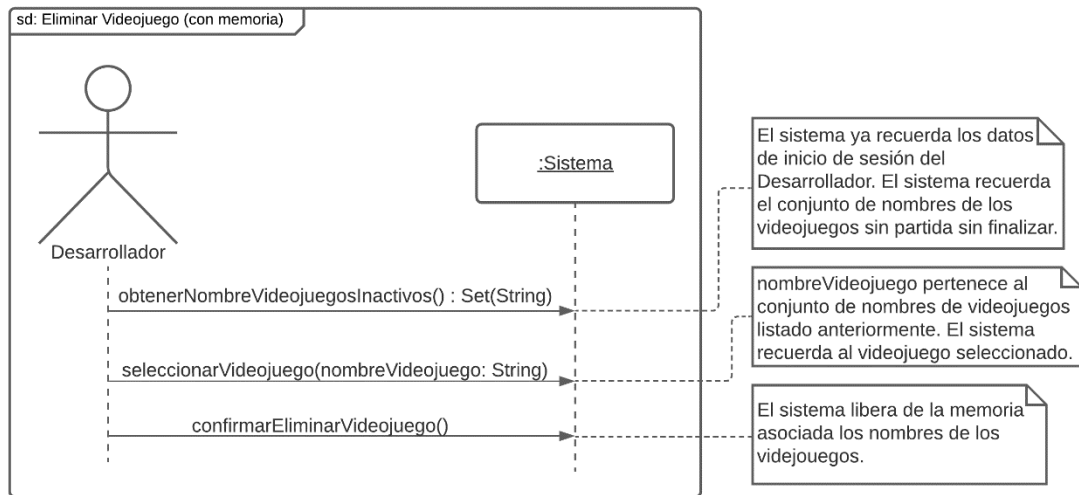
Seguir jugador:

Diagramas de Secuencia del Sistema



Eliminar videojuego:

Diagramas de Secuencia del Sistema



Anexo

Cambios en el modelo de dominio

Tras analizar los casos de uso detectamos que el precio de las suscripciones debía estar incluido como un atributo dentro del concepto videojuego. En nuestro modelo de dominio, el precio estaba contenido dentro del concepto suscripción. Determinamos que esto traería problemas ya que una suscripción necesariamente está asociada con un jugador. Por lo que, al realizar la primera suscripción de un videojuego, no habría ninguna suscripción de donde obtener el precio de la suscripción a instanciar. La solución es mover el atributo dentro del concepto videojuego.

Adicionalmente, ahora que el precio se encuentra dentro del concepto videojuego necesitamos identificar las distintas variantes según el tipo de suscripción. Por lo que construimos el datatype dtPrecios que contiene los precios de los 4 tipos de suscripciones. El nuevo atributo costoSuscripciones es del tipo descrito anteriormente.

La siguiente imagen ilustra la sección actualizada del modelo de dominio:

