

Alta de Usuario

Con Memoria



Usuario

:Sistema

ingresarDatosUsuario(datosUsuario: DtUsuario)

indicarTipoUsuario(tipo: bool)

alt

Si el usuario es un jugador (indicarTipoUsuario = 1)

existe = verificarNickname(nickname: string): bool

loop

Mientras el nickname ingresado ya este en uso / o no cancele

alt

El usuario quiere probar otros nicknames

ingresarNickname(nickname: string)

El usuario quiere cancelar el registro

cancelarAltaUsuario();

ingresarDescripcionDeJugador(descripcion: string)

Si el usuario es un desarrollador (indicarTipoUsuario = 0)

ingresarNombreEmpresa(nombre: string)

cancelar = decisionFinal(): bool

alt

Si cancelar es verdadero

cancelarAltaUsuario();

Si cancelar es falso

confirmarAltaUsuario();

El sistema recuerda los datos del usuario ingresados

DtUsuario

email: string

contrasenia: string



DtJugador

nickname: string

descripcion: string

DtDesarrollador

nombreEmpresa: string

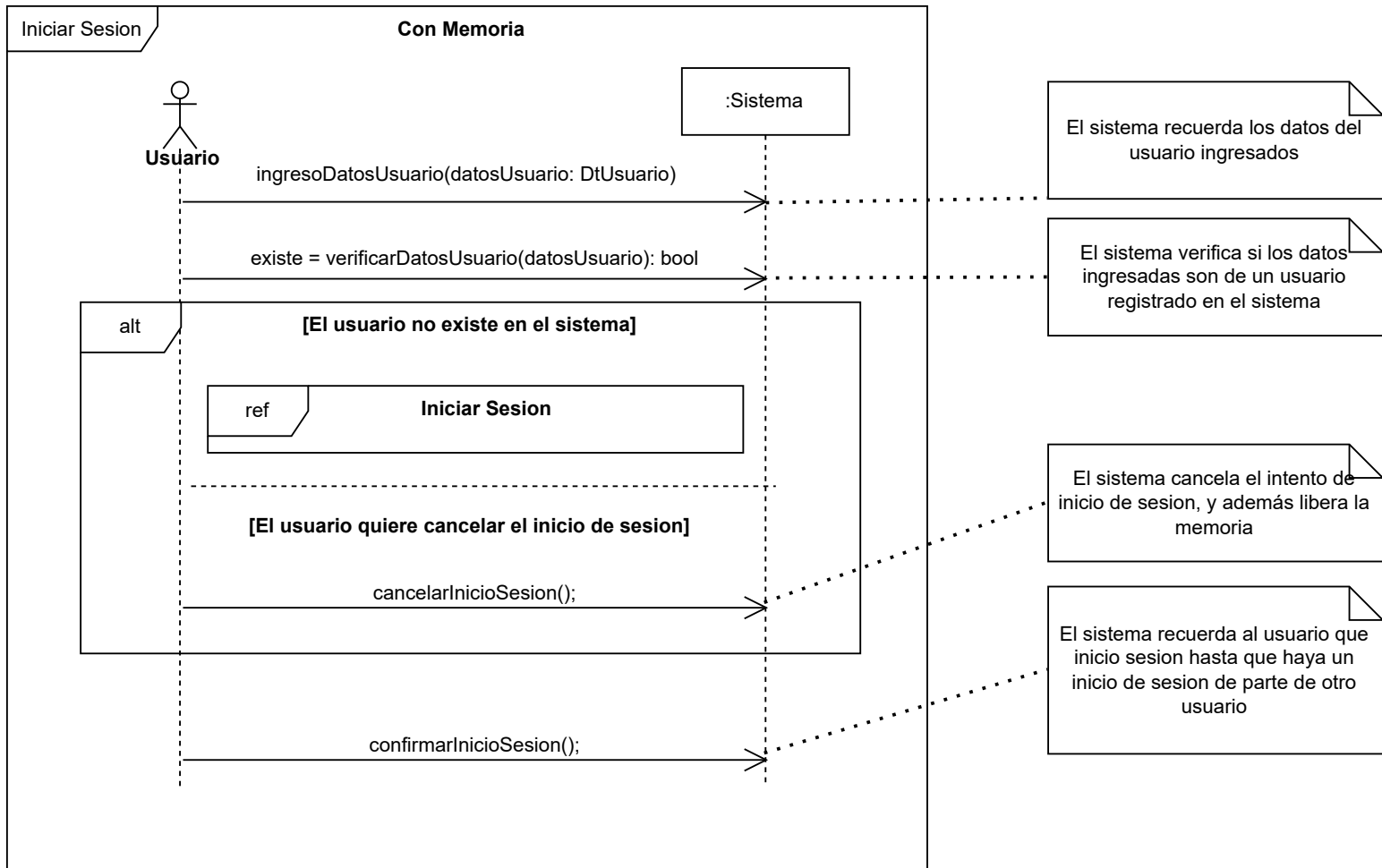
El sistema verifica si el nickname ingresado por el jugador ya existe

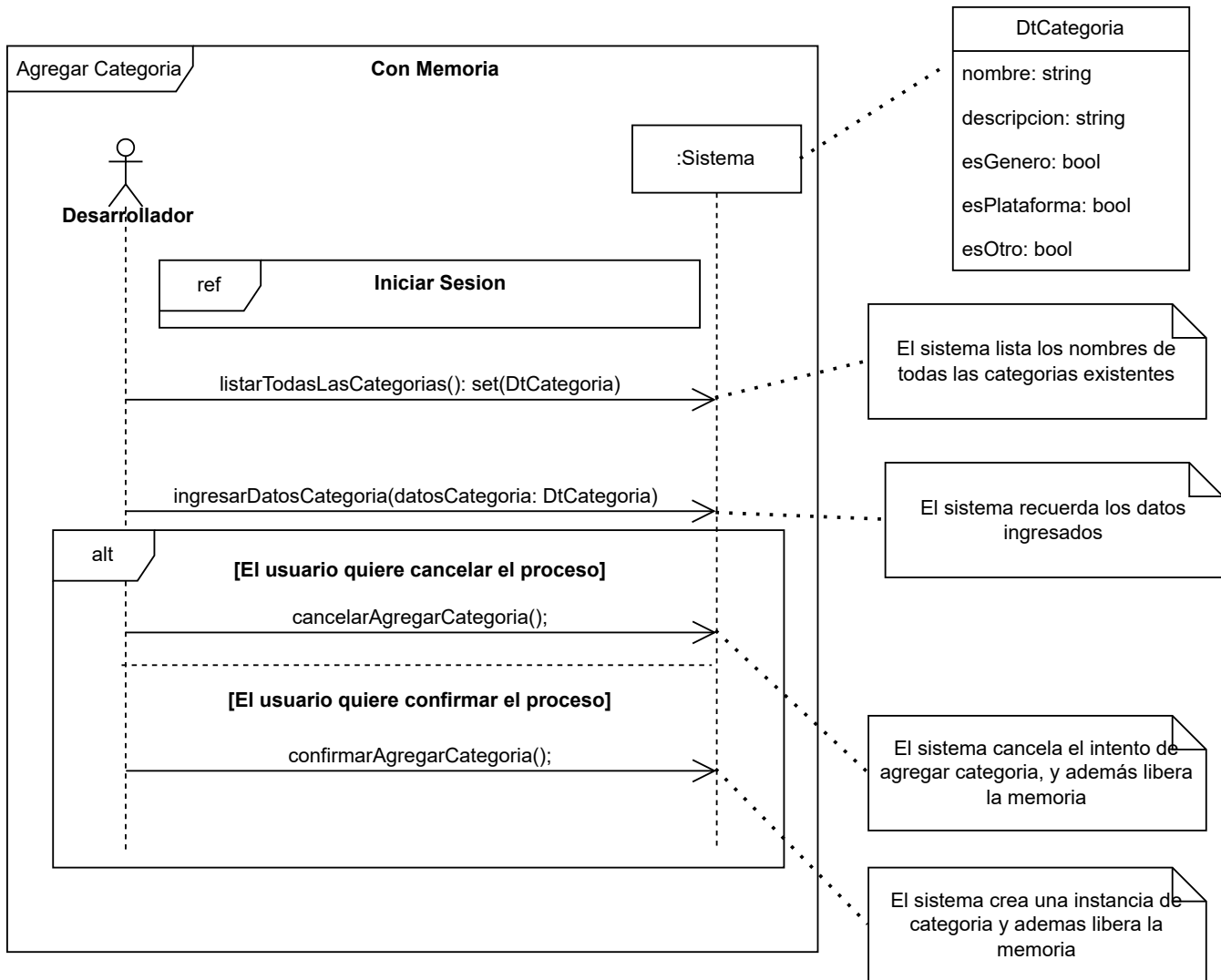
El usuario puede continuar ingresando nicknames hasta hallar uno en desuso

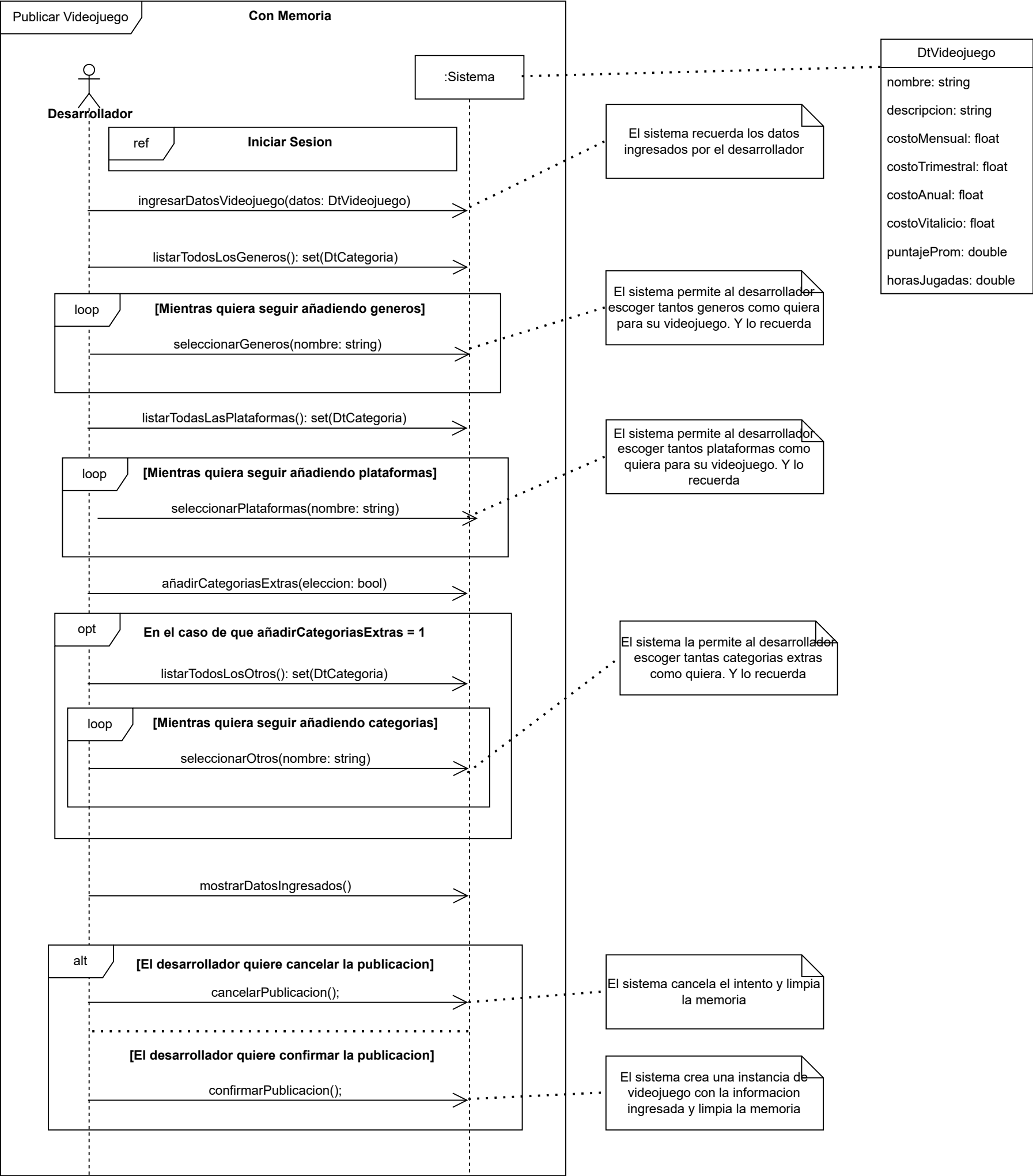
El sistema cancela el registro en proceso, y además libera la memoria

El sistema cancela el registro en proceso, y ademas libera la memoria

El sistema crea una instancia de usuario y además libera la memoria







Asignar Puntaje a Videojuego

Sin Memoria

Jugador

:Sistema

ref

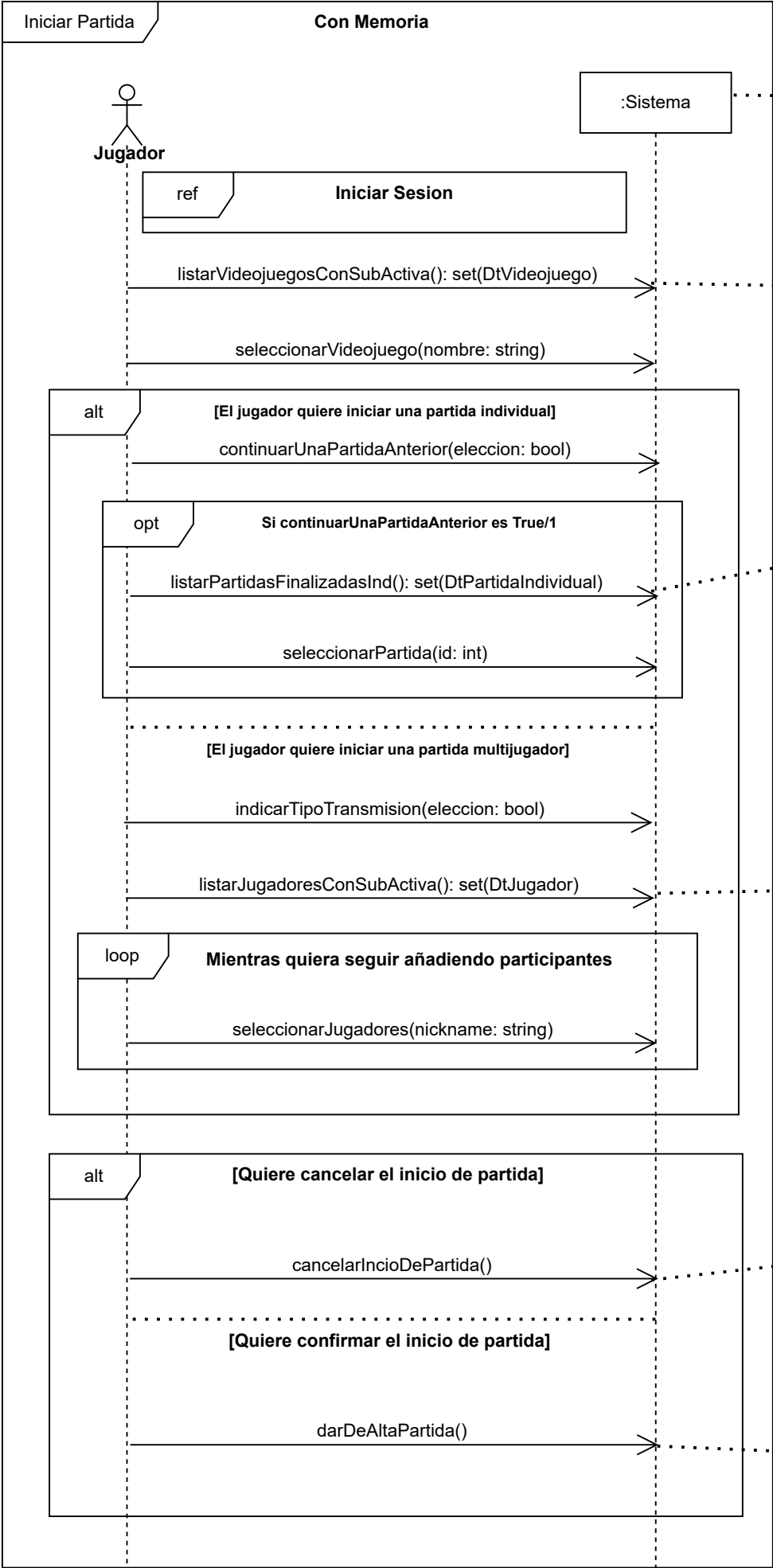
Iniciar Sesion

listarVideojuegos(): set(Videojuego)

puntuar(nombre: string, puntaje: int);

El sistema lista el nombre y
descripcion de cada videojuego en el
sistema

El sistema crea una instancia de
puntuacion con los datos, y además
libera la memoria



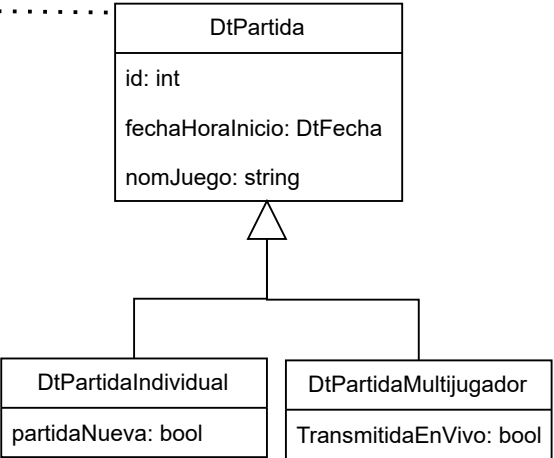
El sistema lista los nombres de los videojuegos para los que el jugador cuenta con una sub. activa

El sistema lista el historial de partidas indivduales finalizadas del jugador cronologicamente

El sistema lista el resto de jugadores en el sistema con una sub. activa para dicho juego

El sistema cancela el proceso y limpia la memoria

El sistema da de alta una nueva partida con los datos ingresados y ademas limpia la memoria



Abandonar Partida Multijugador

Con Memoria

Jugador

:Sistema

ref

Iniciar Sesión

listarPartidasActivas(idJugador: int): set(DtPartidaMultijugador)

elegirPartida(idPartida: int)

registrarAbandono(idJugador, idPartida, fechaHora: DtFecha)

El sistema lista todas las partidas multijugador cuyo id de jugador se encuentre como parte de los participantes y no tenga hora de finalizacion,

El sistema registra la fecha y hora en la que el jugador se retira de dicha partida, y limpia la memoria

DtFecha

anio: int

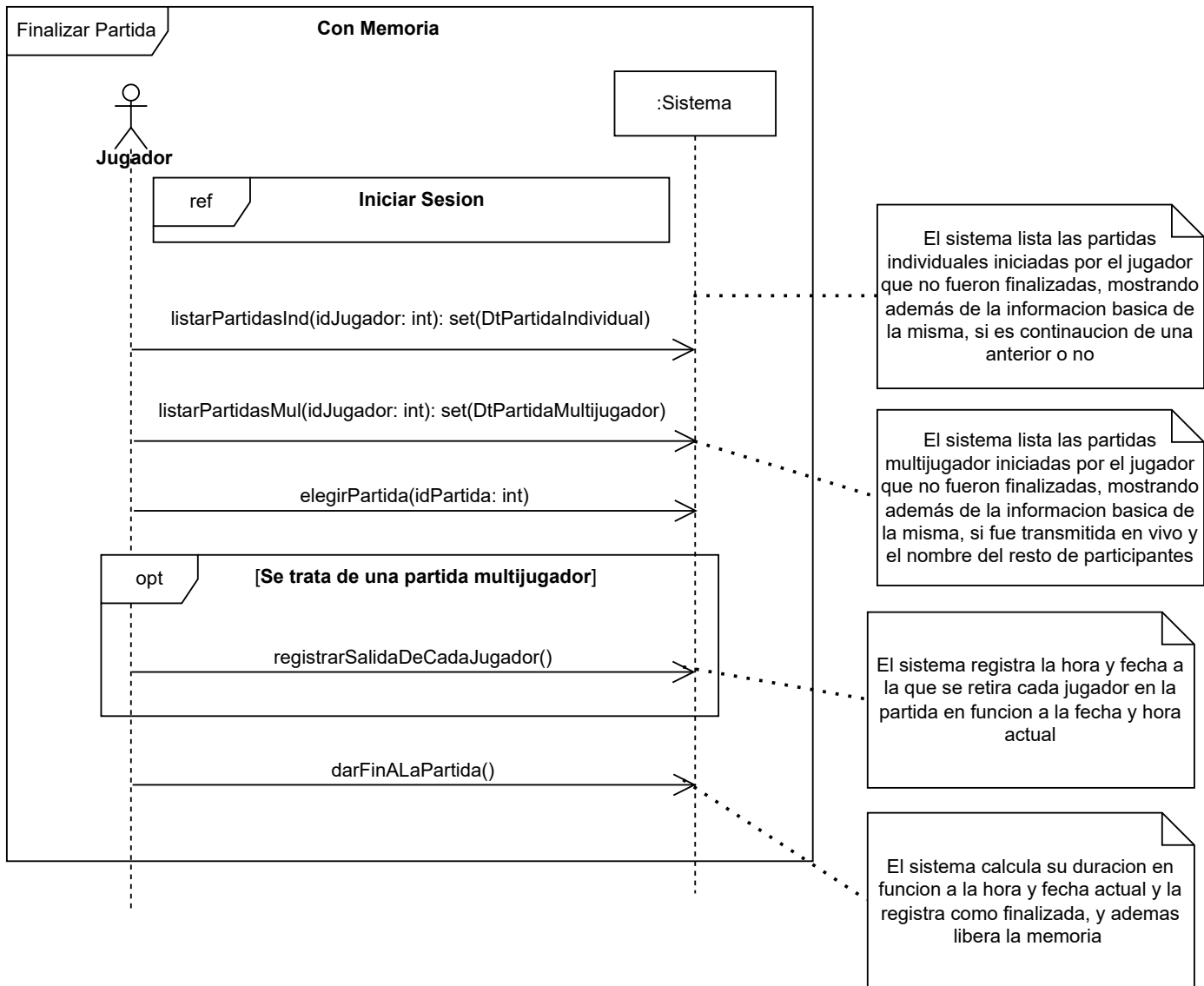
mes: int

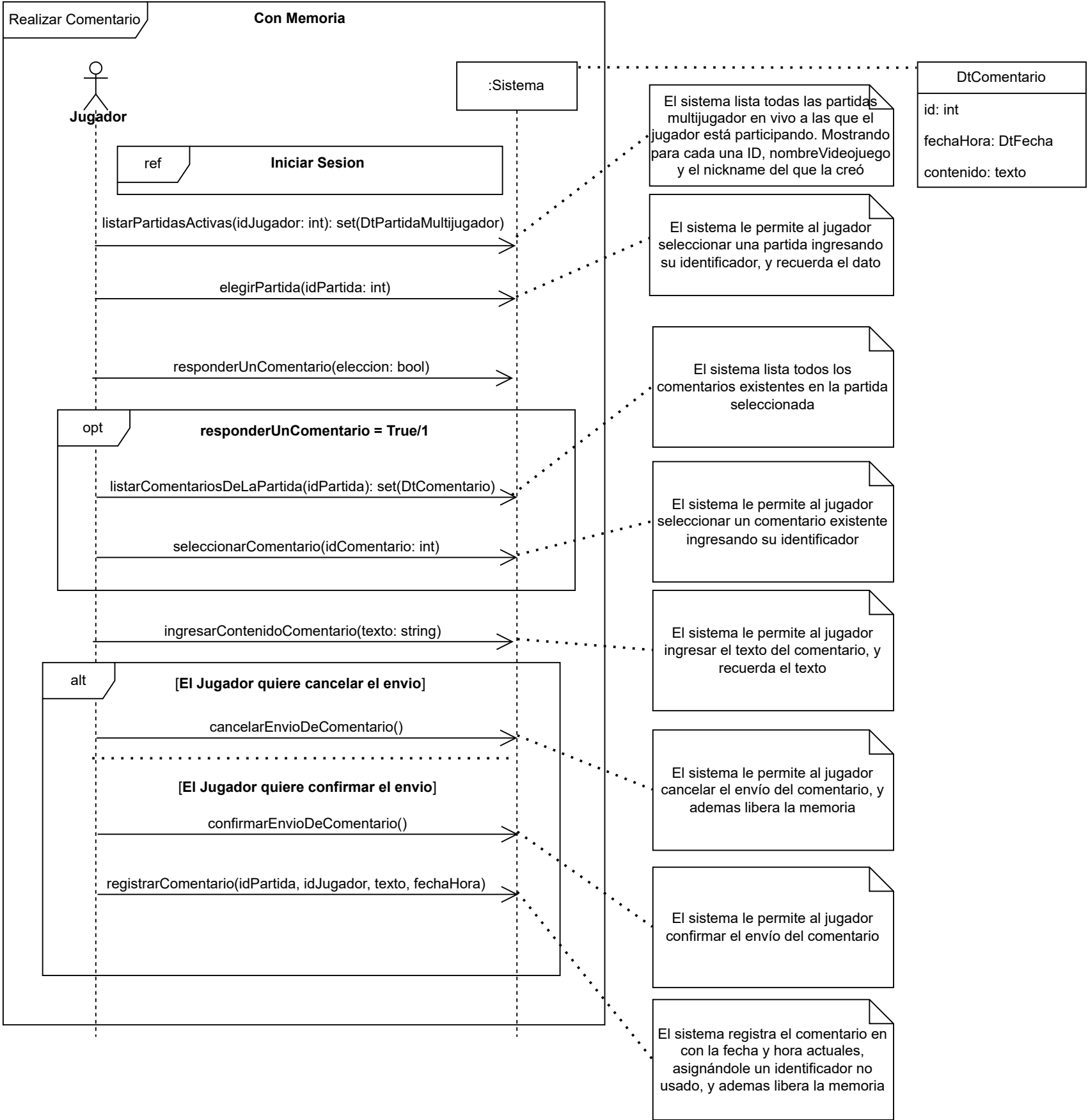
dia: int

dia: int

hora: int

minuto: int





Usuario

:Sistema

ref

Iniciar Sesión

listarTodosLosVideojuegos(): set(DtVideojuego)

seleccionarJuego(nombre: string)

mostrarDetallesVideojuego()

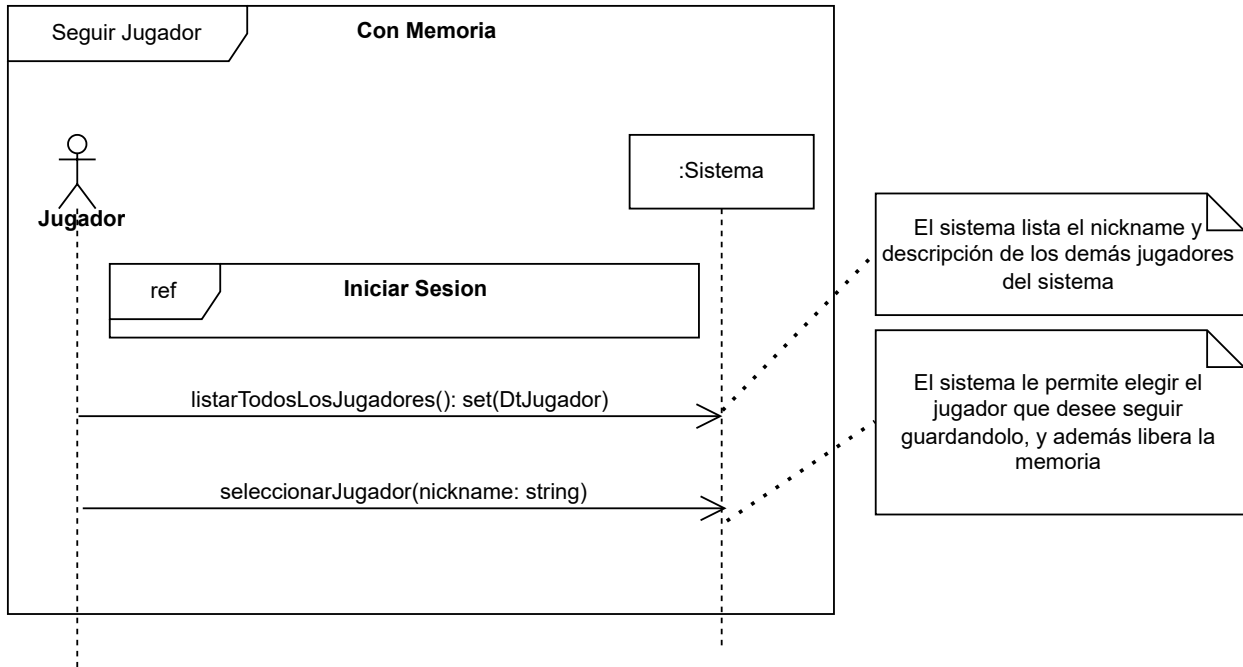
opt

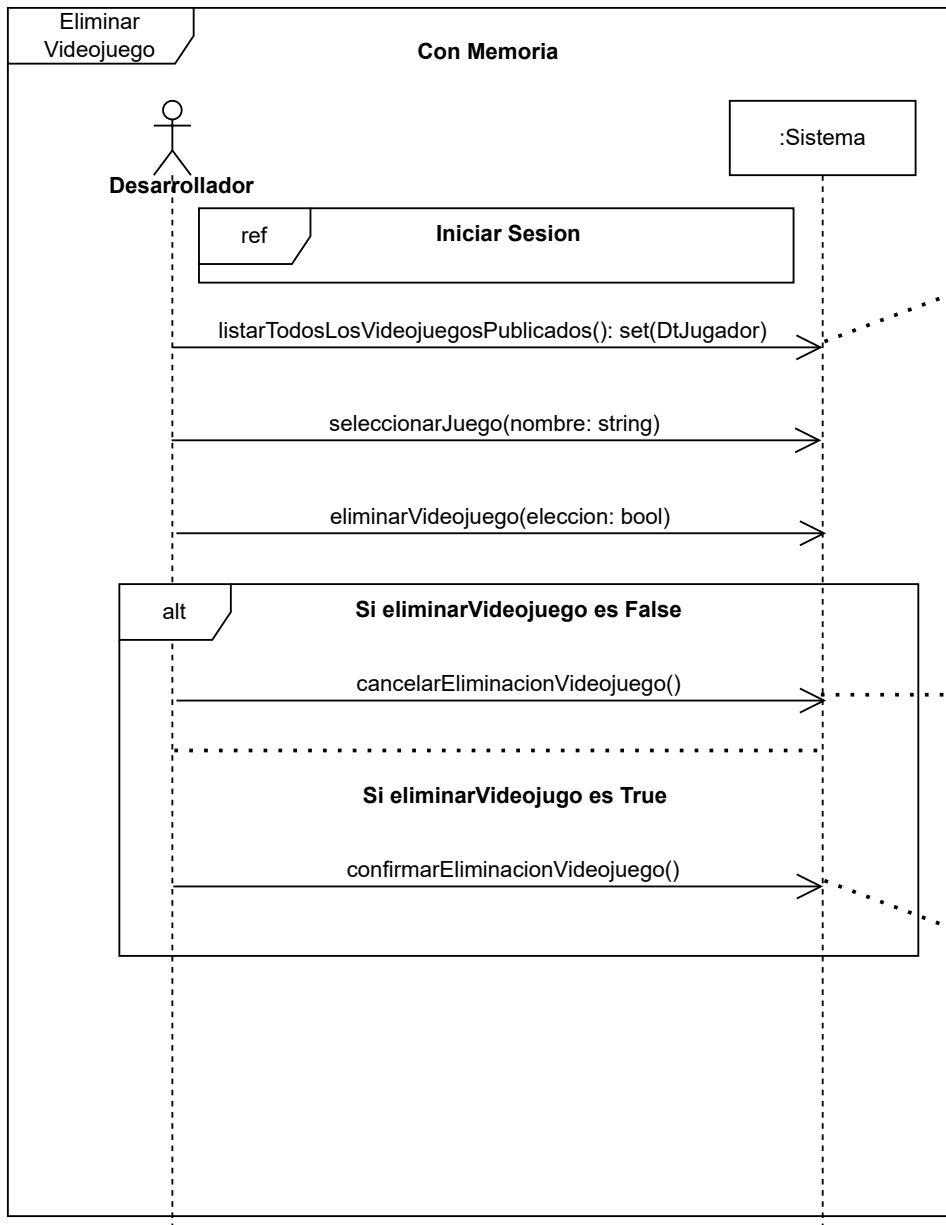
[Es desarrollador]

mostrarHorasTotalesJugadas()

El sistema entonces muestra la descripción, costos por tipo de suscripción, nombres de las categorías a la que pertenece, nombre de la empresa de su desarrollador y su puntaje promedio

En caso de ser desarrollador, muestra además la suma total de horas jugadas en función de todos los jugadores





El sistema lista el nombre de todos los juegos publicados por el desarrollador que no tengan partidas sin finalizar

El sistema cancela el proceso, limpiando además la memoria

El sistema elimina el videojuego junto a toda información asociada al mismo, es decir, suscripciones, partidas, comentarios, puntajes, etc. En el proceso, limpia la memoria