



Universidad de Buenos Aires  
Facultad de ciencias exactas y naturales  
Departamento de Computacion

Ingenieria del Software I  
**Proyecto: Casino Online**  
Informe 1: Analisis de Requerimientos y especificación  
Numero de grupo: 2

**Autores**

Aquino, Isis	313/05	isisaquino@yahoo.com.ar
Alvarez, Maria de los Angeles	264/05	mdelosaalvarez@hotmail.com
Engler, Christian Alejandro	314/05	caeycae@gmail.com

**Resumen / Abstract**

Proyecto casino online

**Palabras clave / Keys**

UML, Objetivos, Agentes, Casos De Uso, Diagrama De Contexto, Modelo Conceptual, OCL, Diagrama de Actividades, FSM

1er Cuatrimestre 2008

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Objetivo del documento . . . . .	2
1.2. Convenciones de notación . . . . .	3
1.3. Documentos relacionados . . . . .	3
1.4. Organización del informe . . . . .	3
<b>2. Vista Modular</b>	<b>3</b>
2.1. Modulos Server . . . . .	3
2.2. Modulos Cliente y Administrador . . . . .	6
<b>3. Modulos</b>	<b>7</b>
3.1. Mensajero / Interpretador . . . . .	7
3.2. Interpretador / Parser / Servicios . . . . .	11
3.3. Casino . . . . .	12
<b>4. Funcionalidad Casino</b>	<b>16</b>
4.1. Registracion y Ingreso al casino online y modificacion de saldo . . . . .	16
4.1.1. Registracion . . . . .	16
4.1.2. Modificacion de saldo . . . . .	16
4.1.3. Ingreso y egreso del casino . . . . .	16
4.2. Administracion del Casino . . . . .	20
4.2.1. Apertura del casino . . . . .	20
4.2.2. Clausura del casino . . . . .	22
4.3. Invitado . . . . .	23
4.4. Modo Dirigido . . . . .	27
4.4.1. Inicio de Modo Dirigido . . . . .	28
4.4.2. Seteo de Jugadas . . . . .	29
4.4.3. Finalización de Modo Dirigido . . . . .	30
4.5. Seleccionador de Tipo de jugada . . . . .	30
4.6. Juego Tragamonedas . . . . .	32
4.6.1. Entrar en Tragamonedas . . . . .	32
4.6.2. Apostar en Tragamonedas . . . . .	34
4.6.3. Tirar en Tragamonedas . . . . .	36
4.6.4. Salir en Tragamonedas . . . . .	38
4.7. Juego Craps . . . . .	40
4.7.1. Entrar en Craps . . . . .	40
4.7.2. Apostar en Craps . . . . .	43
4.7.3. Tirar en Craps - Aceptado . . . . .	44
4.7.4. Salir en Craps . . . . .	48
4.8. Notificar Estado Casino . . . . .	50
4.9. Generacion de reportes . . . . .	51
4.9.1. Generacion de reporte: Ranking de Jugadores . . . . .	51
4.9.2. Generacion de reporte: Estado Actual . . . . .	52
<b>5. Anexo I - Conclusiones</b>	<b>53</b>

## 1. Introducción

### 1.1. Objetivo del documento

El objetivo de este documento es sentar las bases para la implementación del Casino On Line. Hemos tomado como punto de partida la especificacion aprobada por el departamento de calidad de software del grupo timbalista y a partir de ello construir el diseño utilizando principios que nos permitan asegurar la calidad del software a construir.

## 1.2. Convenciones de notación

## 1.3. Documentos relacionados

Los documentos relacionados con este proyecto son los que aquí se detallan:

- Arquitectura Conceptual de la Aplicación y Protocolo de comunicación del Casino OnLine y respectivas actualizaciones.

## 1.4. Organización del informe

**El Informe esta organizado de la siguiente manera:**

**Introduccion** Una introducción al mismo documento, y a los otros documentos relacionados para posibles consultas.

**Vista Modular** En esta seccion describimos la metodologia que adoptamos para efectuar el diseño. En la vista modular se pueden ver los distintos bloques que se describiran en el resto del presente informel. Aqui tambien se puede ver la conexion entre estos bloques.

**Modulos** Aqui se describe la comunicacion que existe entre los distintos modulos de la VistaModular, explicada con anterioridad

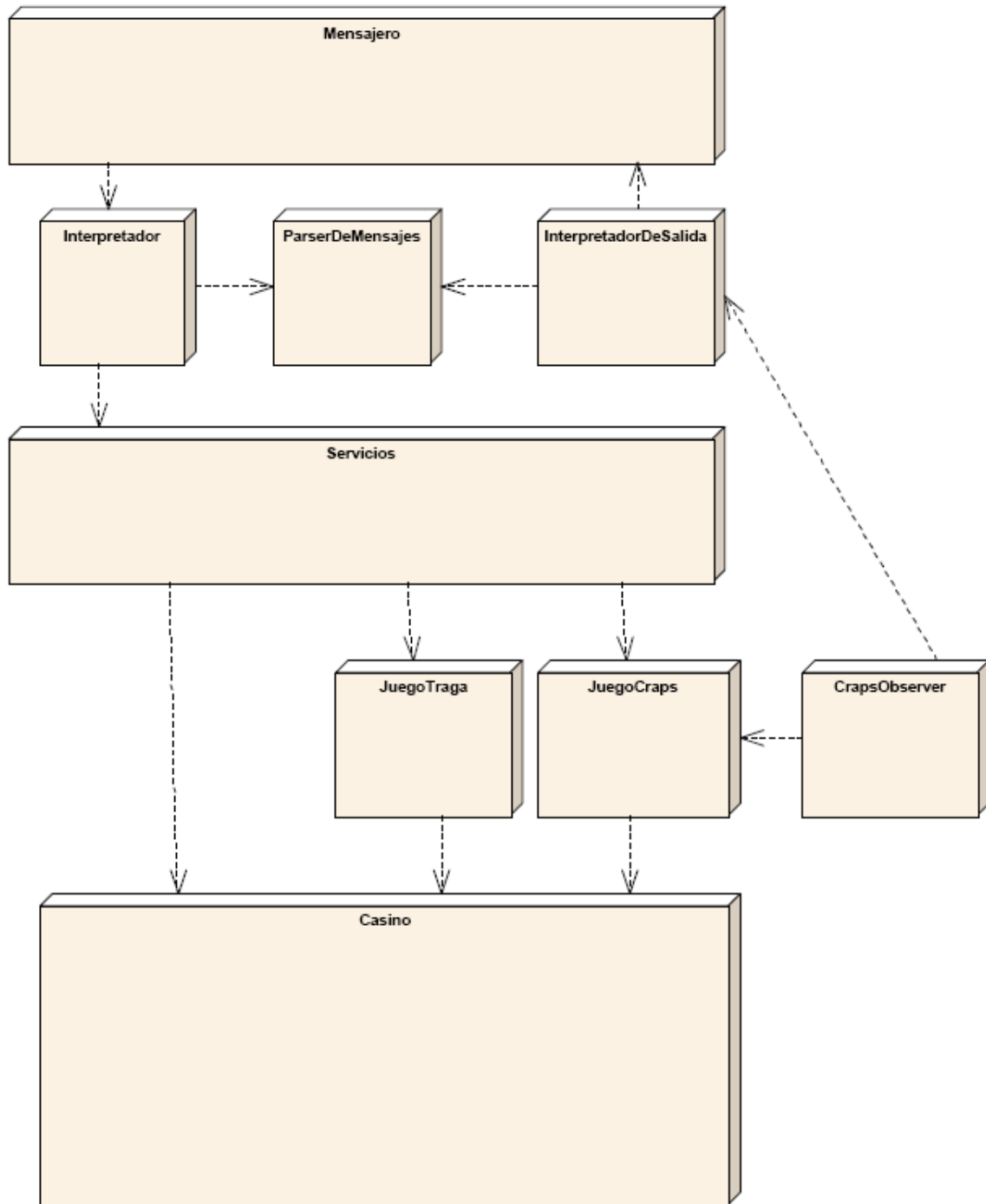
**Funcionalidad Casino** En esta seccion se explican cada una de las funcionalidad con diagramas de secuencia.

# 2. Vista Modular

Para organizar el diseño, dividimos el problema en varios problemas independientes con el fin de hacer el diseño mas flexible y con expectativas de ocultamiento por capas, de modo de que a cada capa se le pudiera cambiar la implementacion.

## 2.1. Modulos Server

Vista Modular Server



### Descripción de cada uno de los módulos

**Mensajero** El módulo mensajero se ocupa de recibir los mensajes que le envían los clientes al servidor del casino online. Sus obligaciones son “chequear” la llegada de un mensaje, empaquetarlo y entregárselo al siguiente módulo.

**Interprete** El módulo interprete se ocupa de tomar un mensaje e intentar transformarlo a un lenguaje entendible por los servicios del casino, en caso de que el mensaje no corresponda al formato esperado, deberá notificar al cliente. Para lograr esta transformación hace uso del módulo Parser.

**Parser** Este módulo convierte los datos del lenguaje de comunicación al lenguaje que maneja el casino y viceversa. Esto es para que la elección del lenguaje de cada una de estas partes sea independiente y no afecte al resto del sistema.

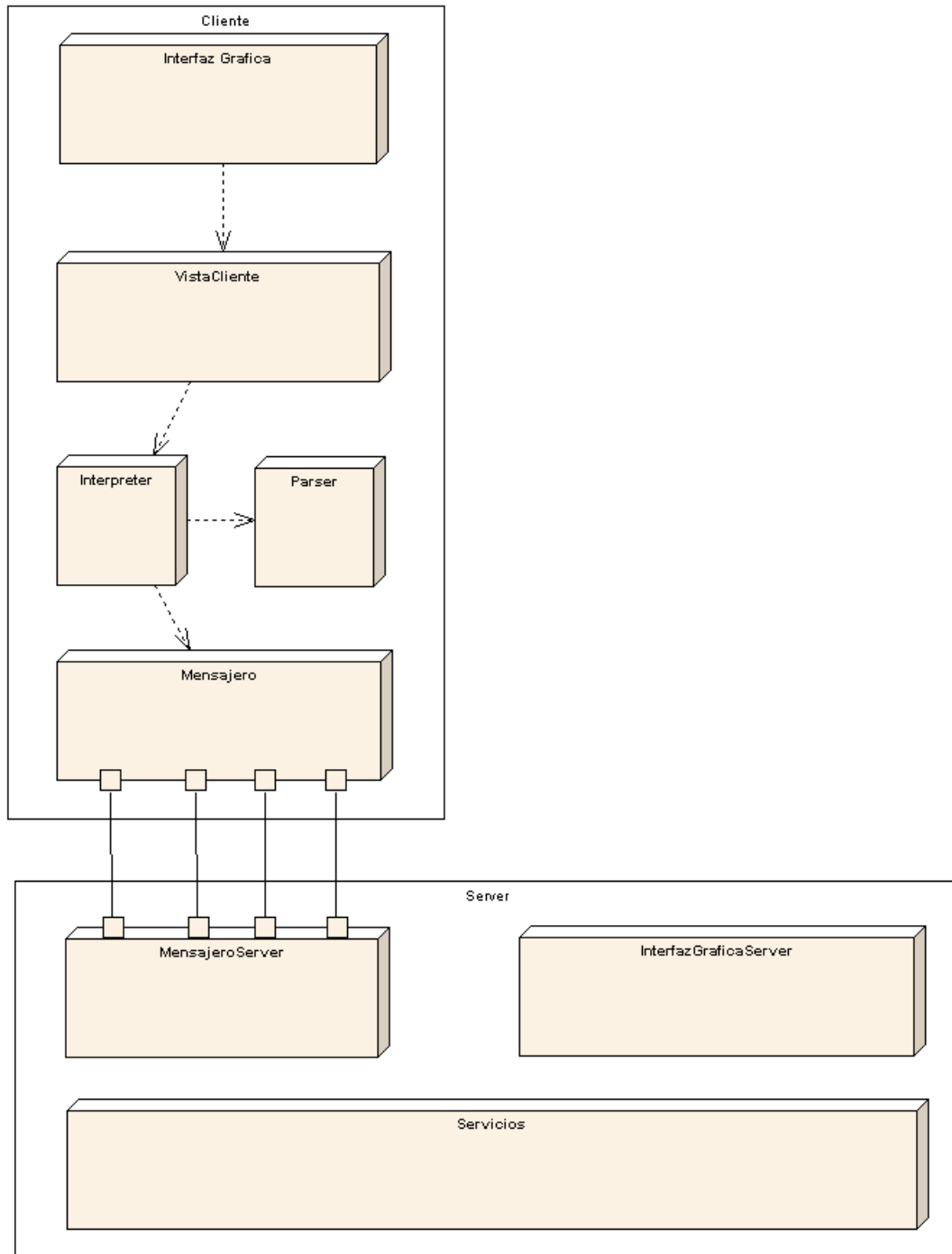
**Servicios** Este módulo nuclea todos los servicios que presenta el casino. El diseño del casino está

ocultado detras de este modulo para que la implementacion de estos servicios pueda ser modificada sin grandes efectos colaterales.

**Casino** Aqui esta el casino propiamente dicho, independiente a toda forma de comunicacion y protocolo. Este modulo se encarga de realizar todo lo especificado en el informe I.

## 2.2. Módulos Cliente y Administrador

### Vista Modular Cliente



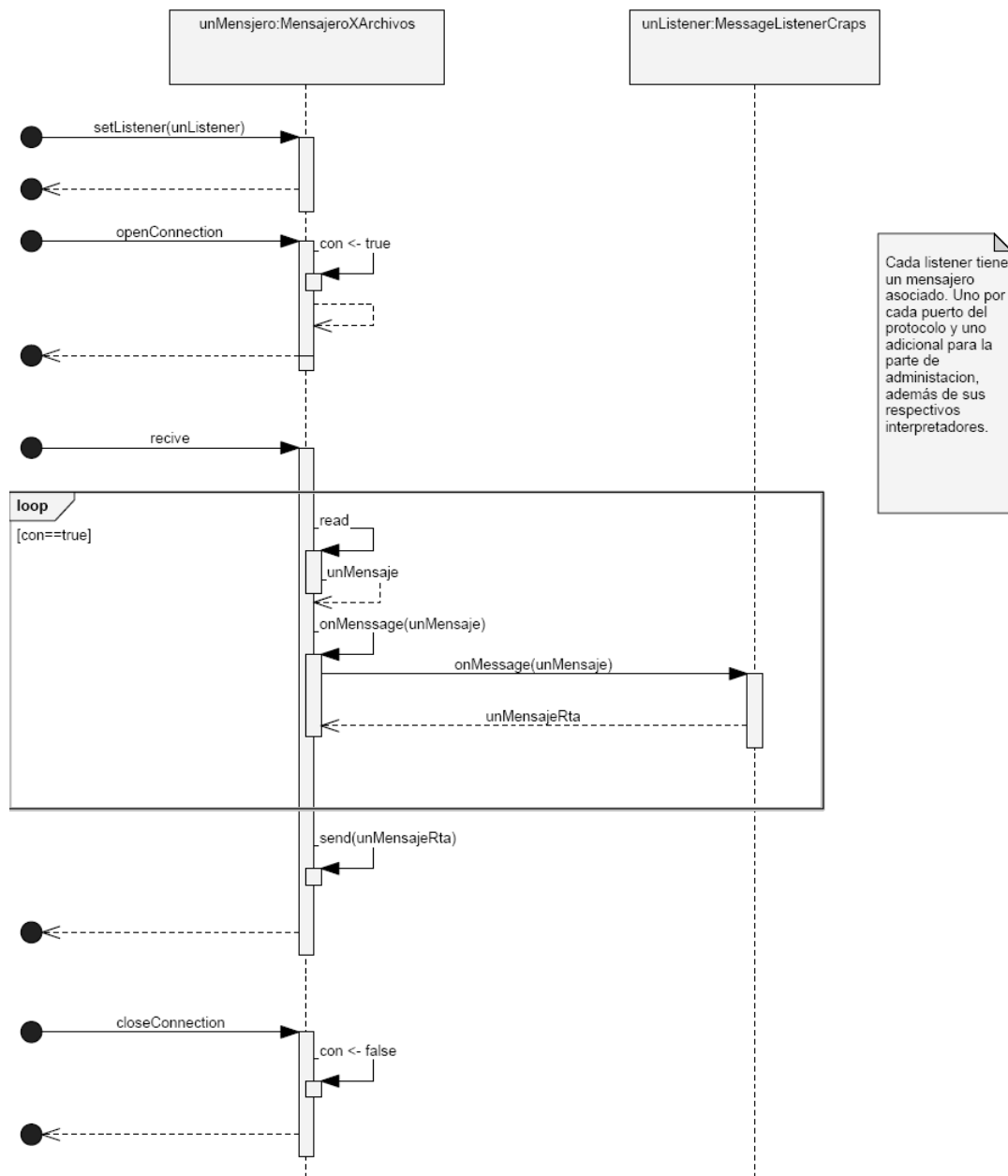
### 3. Módulos

#### 3.1. Mensajero / Interpretador

Diseñamos un servicio “core” de mensajería para que administre la recepción de mensajes independientemente del uso que se le de a ellos.

Presentamos aquí un diagrama de secuencia con un ejemplo de uso correcto del servicio de mensajería. No nos comprometemos a que la implementación del servicio sea “porArchivos”, pero respetará el modelo propuesto por la clase abstracta mensajero.

Diagrama de Secuencia: Mensajero



Secuencia valida para el uso de la api de mensajeria

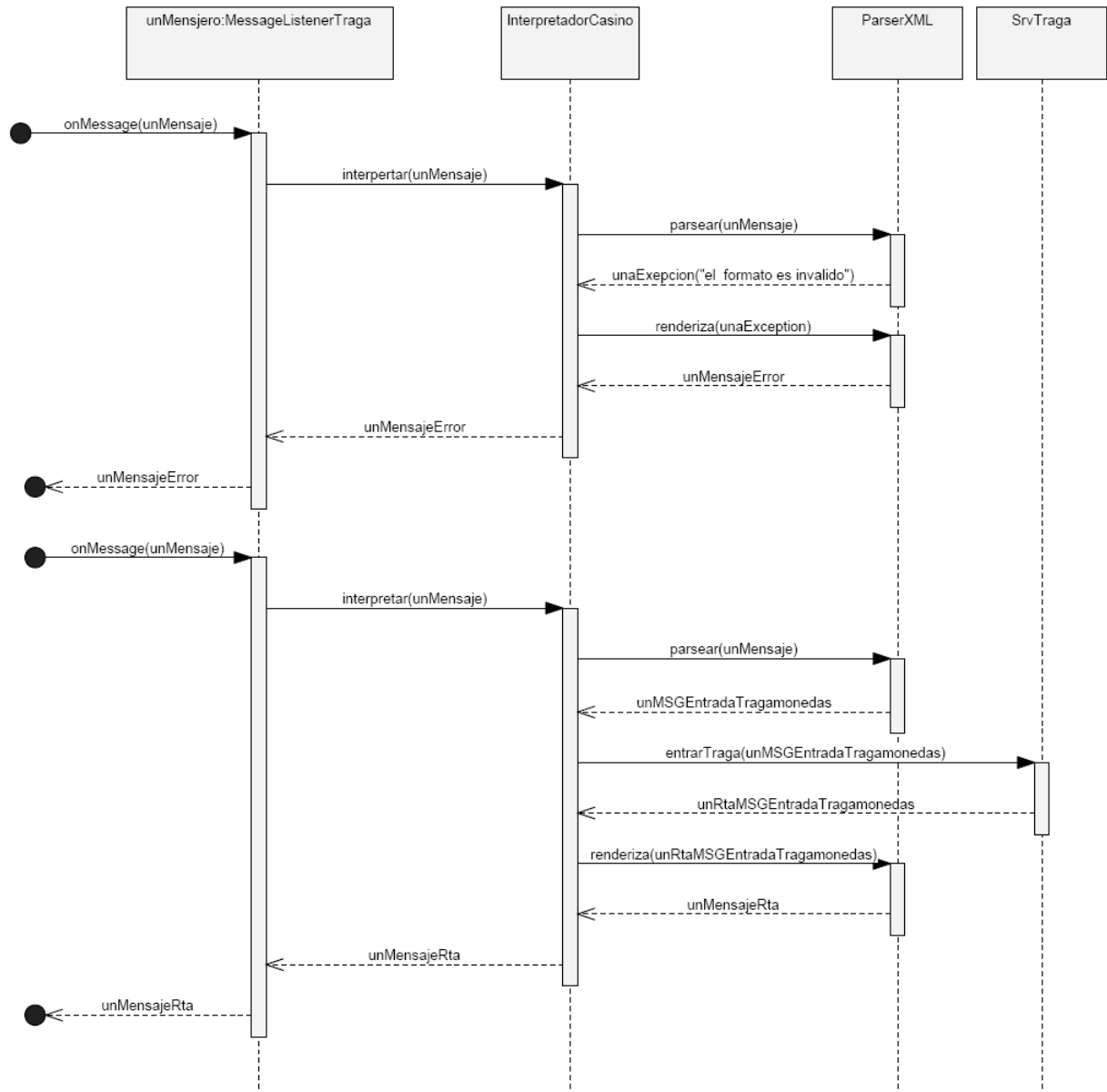
setListener: objeto que indicara que hacer ante el arribo de un message  
 openConnection: seteos iniciales de la coneccion  
 recive: se inicia el mensajero que recibirá mensajes hasta que se cierre la conecccion  
 closeConnection: finaliza la coneccion y le recepcion de mensajes

read: read bloqueante que espera hasta el arribo de un mensaje  
 onMessage: invoca al listener asociado para que realice la accion correspondiente



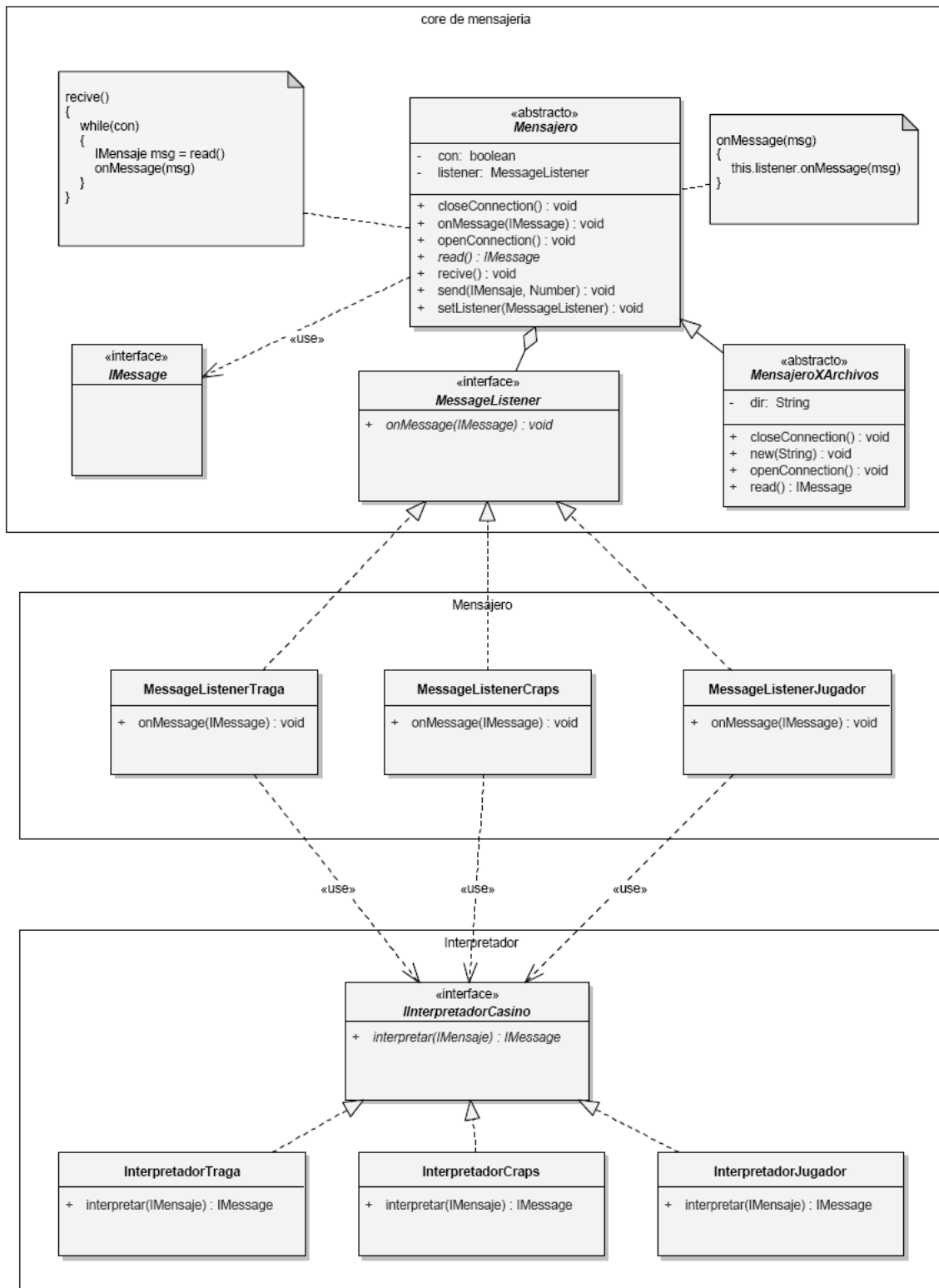
Ahora veamos mas en particular la operacion **OnMessage**. Esta operacion invoca al listener asociado al mensajero, dicha clase dira al mensajero que operacion realizar en el momento de recibir un mensaje.

### Diagrama de Secuencia: OnMessage



Para completar la explicacion del diseño de este modulo necesitamos el diagrama de clases.

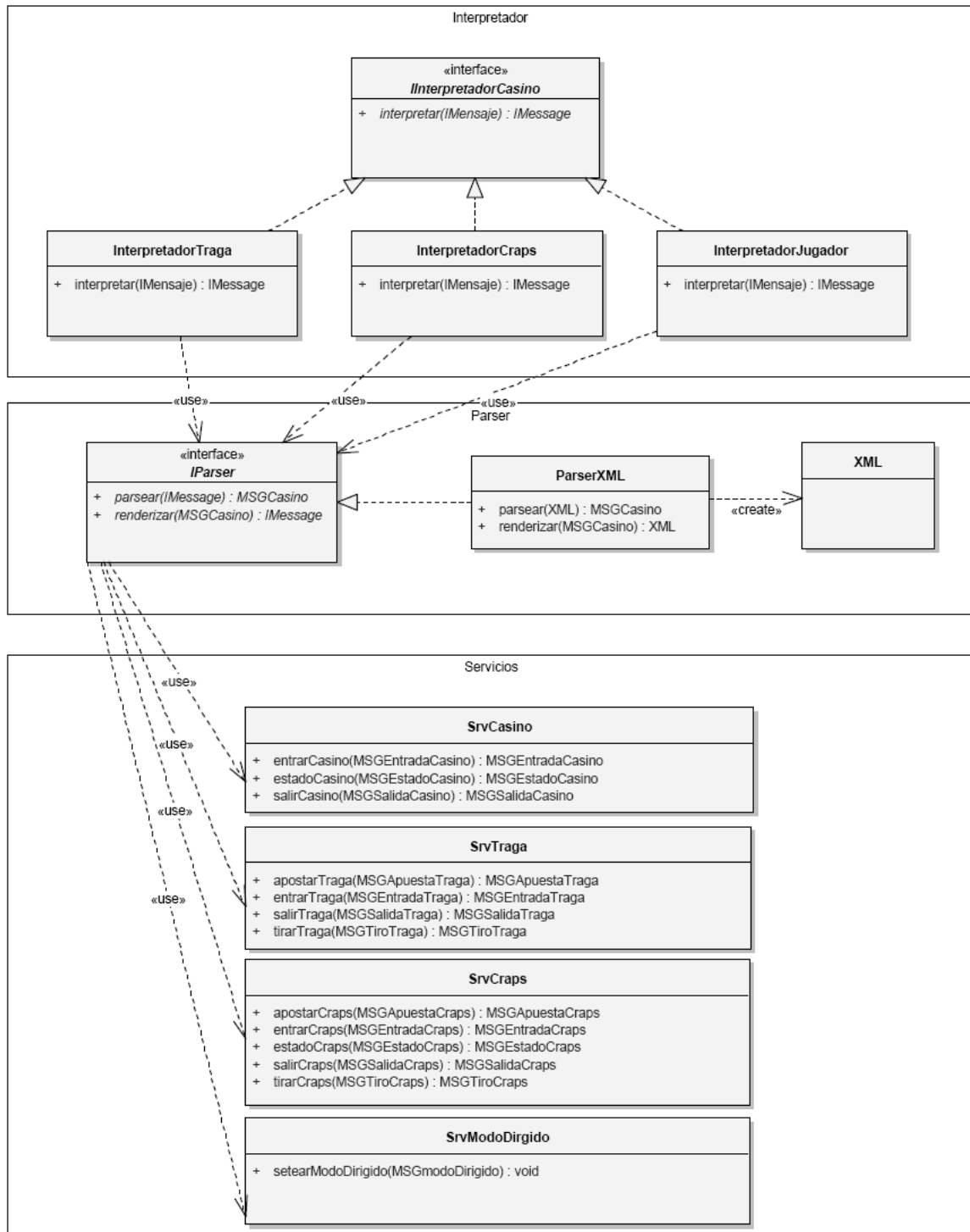
### Diagrama de Clases: Mensajero / Interpretador



### 3.2. Interpretador / Parser / Servicios

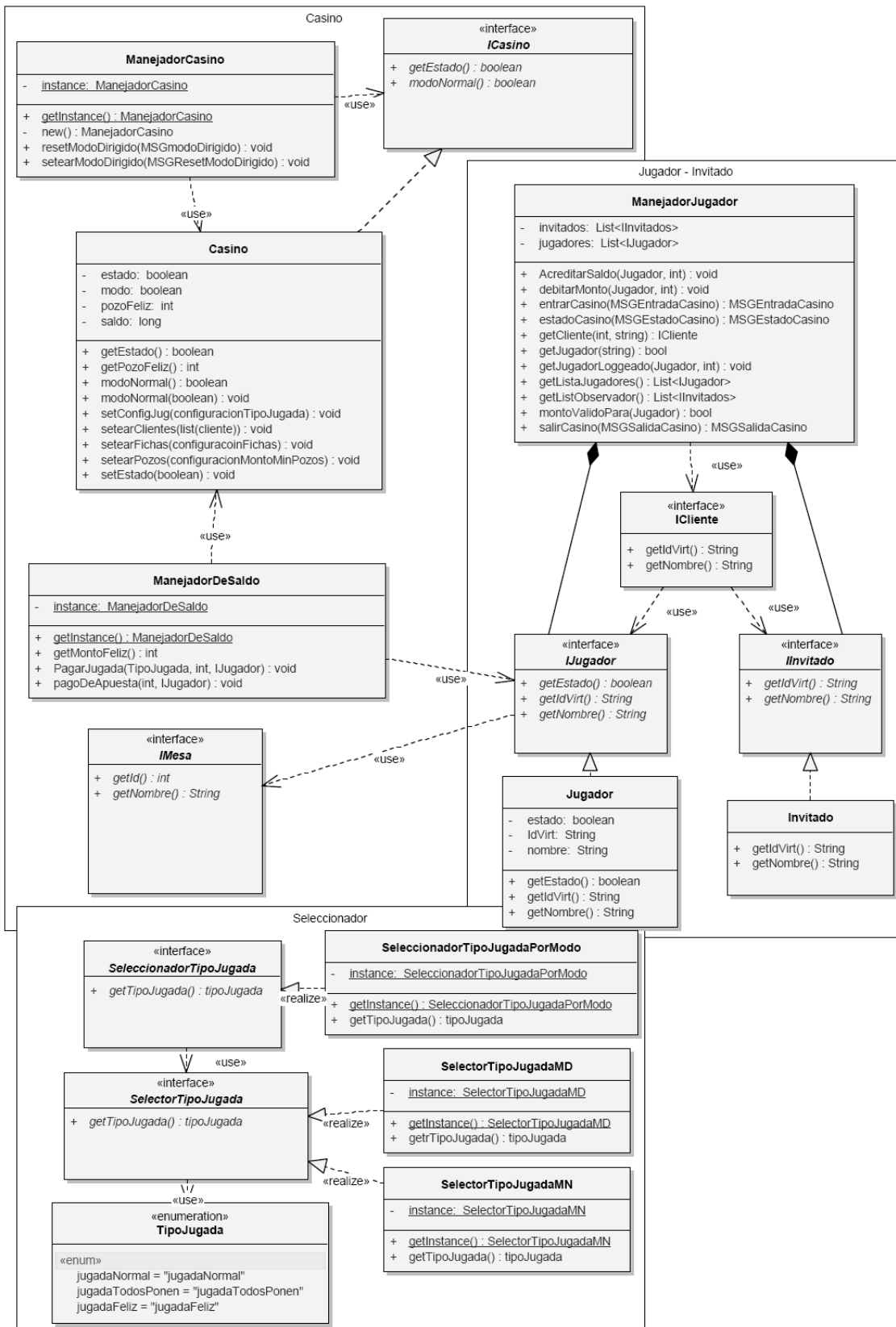
No hay mucha complejidad en estas interacciones y están suficientemente ilustradas en el diagrama de secuencia anterior. Presentamos el diagrama de clases de los módulos restantes.

Diagrama de Clases: Interpretador / Parser / Servicios

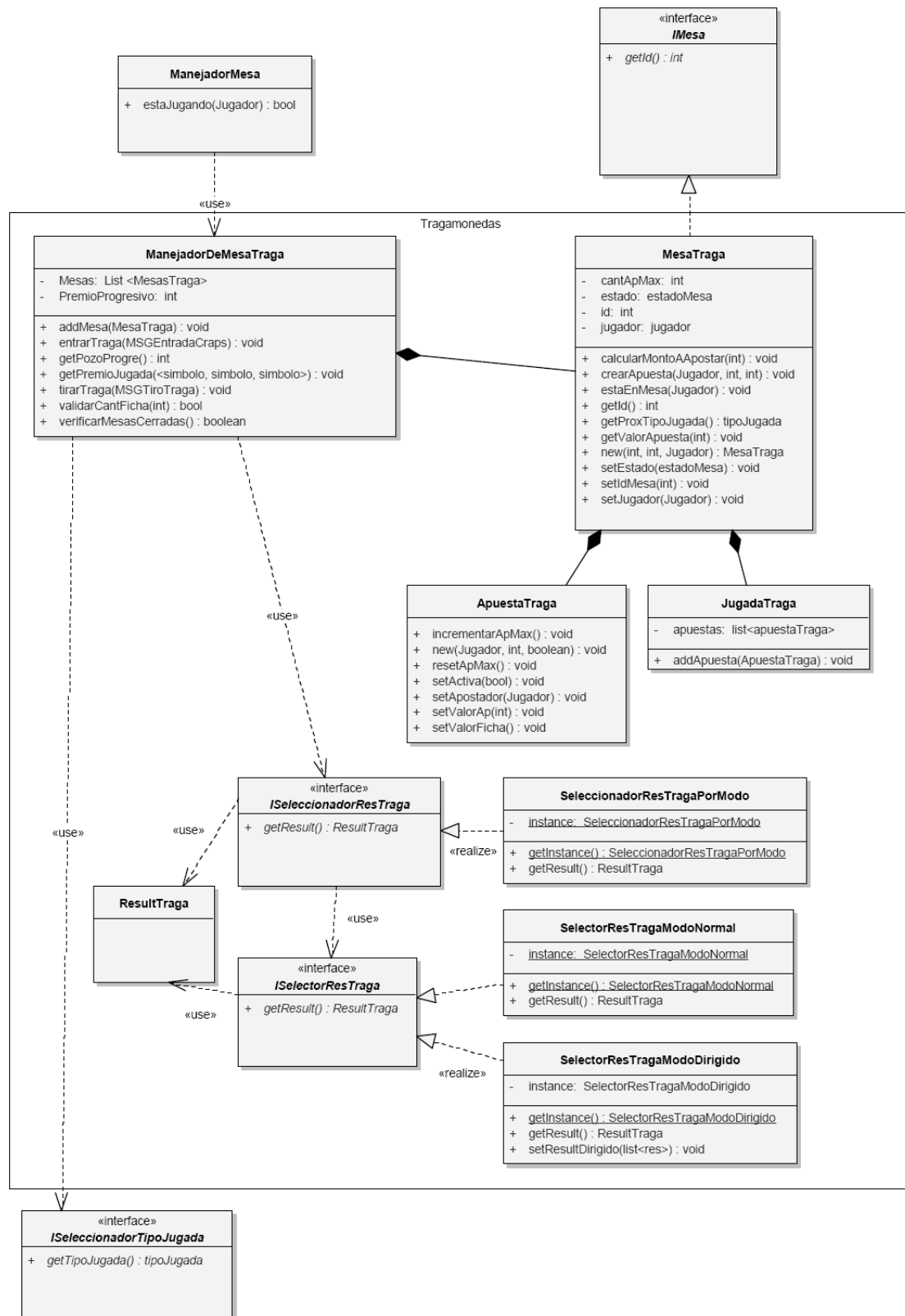


### 3.3. Casino

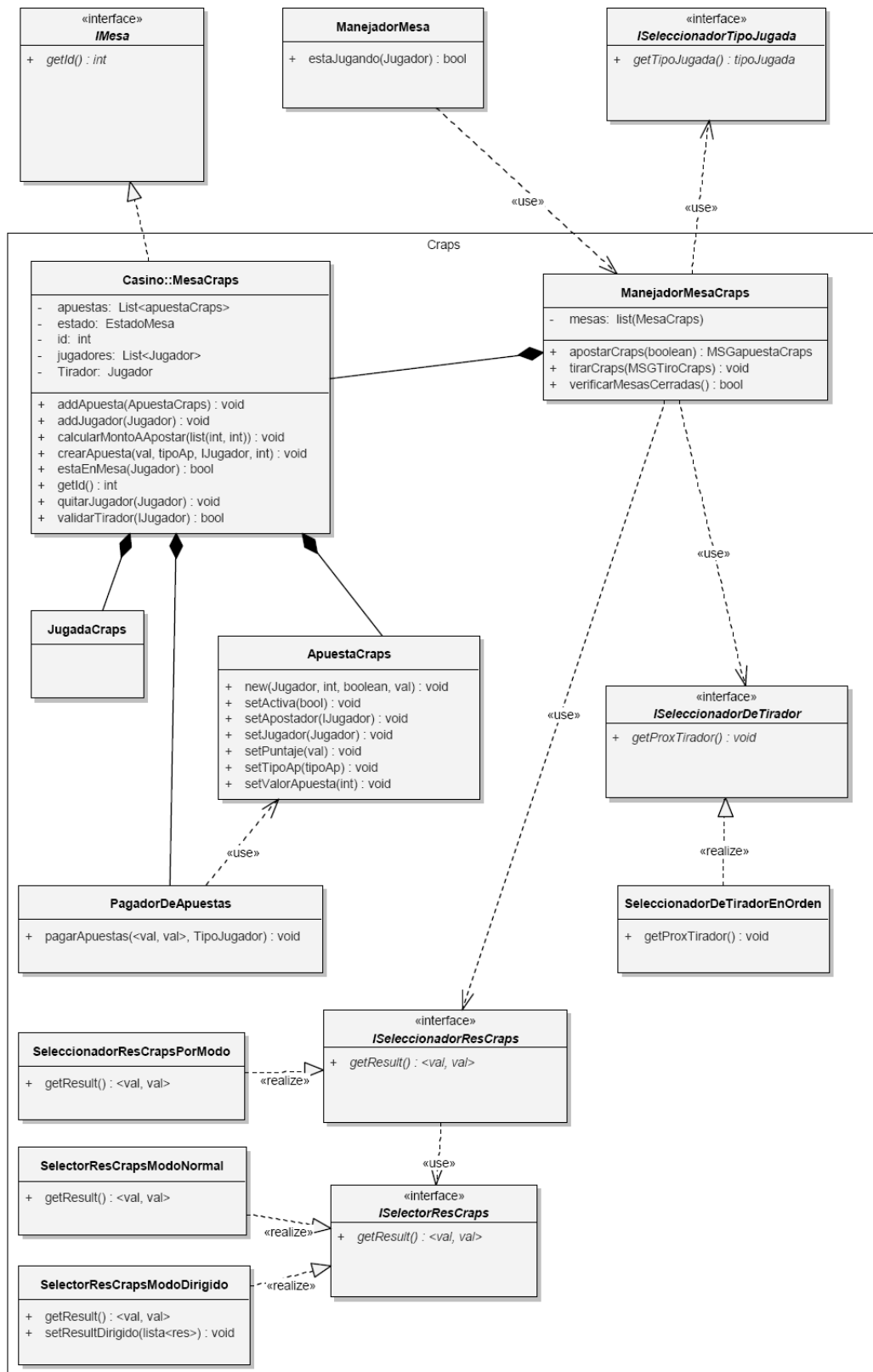
Diagrama de Clases: Casino / Jugador / Invitado / TipoJugada



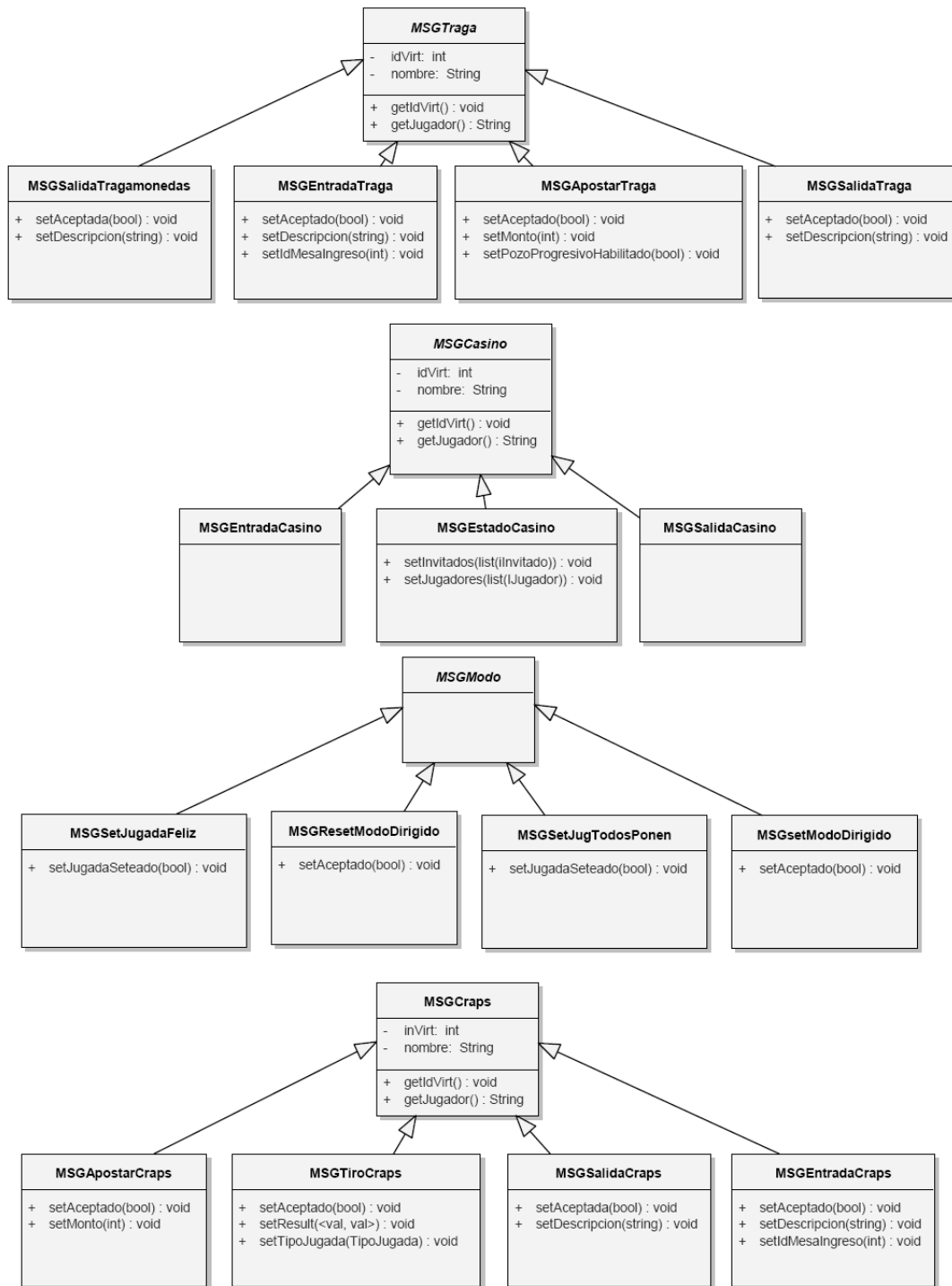
## Diagrama de Clases: Traga



## Diagrama de Clases: Craps



## Diagrama de Clases: Mensajes



## 4. Funcionalidad Casino

### 4.1. Registracion y Ingreso al casino online y modificacion de saldo

#### 4.1.1. Registracion

Segun lo acordado en la especificacion, la registracion de los usuarios se hace por fuera del sistema informatico del casino online.

En el momento de abrir el casino el sistema contará con un archivo XML (en el futuro podria ser de otra manera)

**Dicho archivo tendrá la siguiente sintaxis:**

```
<jugadores>
  <jugador nombre="Nombre" saldo="valor numerico del saldo" vip="true | false">
</jugadores>
```

#### 4.1.2. Modificacion de saldo

Las modificaciones de saldo real solo se pueden realizar mientras el casino permanece cerrado. Mas alla de la operatoria, deberá modificarse el archivo definido en el punto anterior con el monto correspondiente antes de la proxima apertura del casino.

#### 4.1.3. Ingreso y egreso del casino

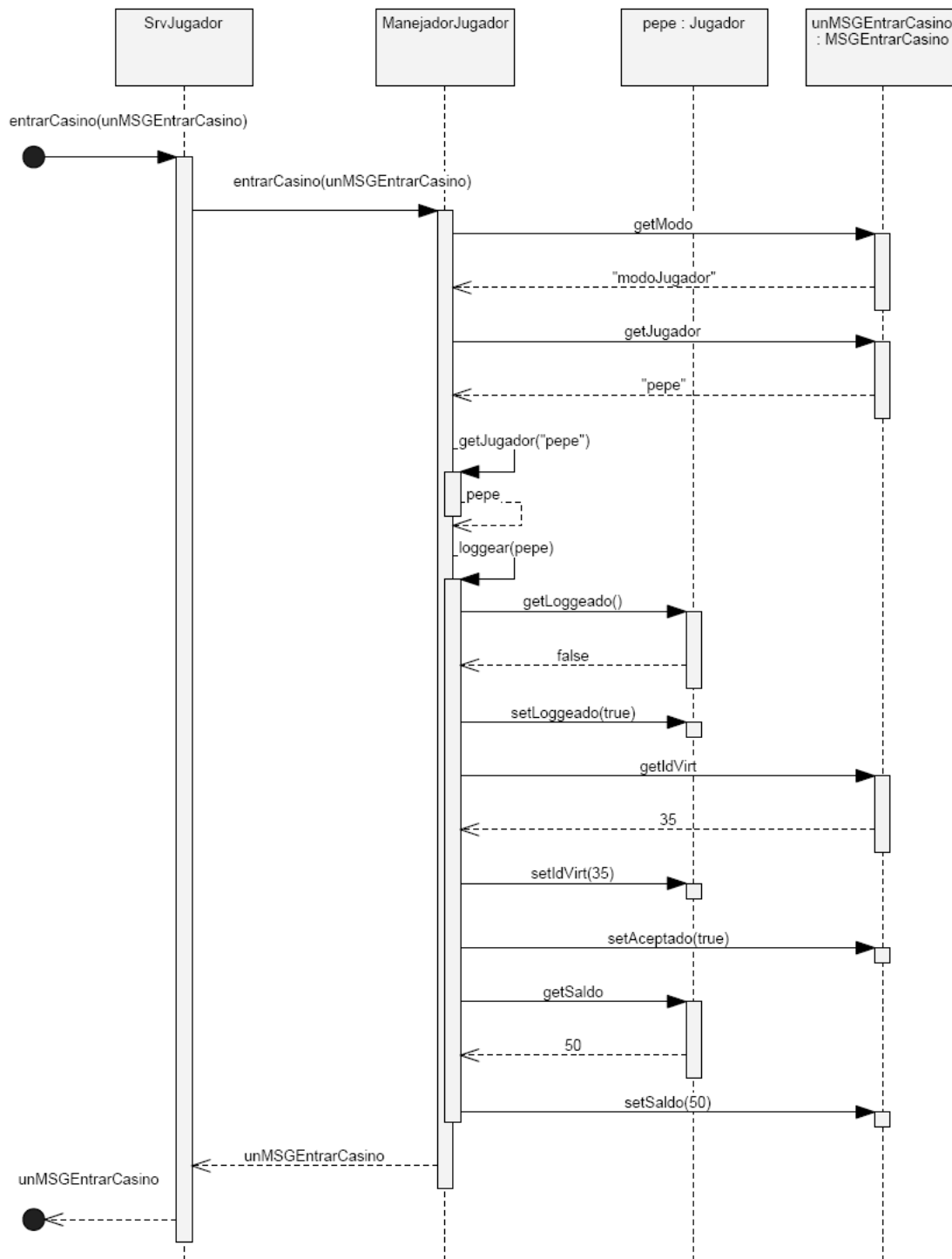
En esta seccion estamos apuntando al ingreso de jugadores al casino (no al ingreso de jugadores observadores, dicha interaccion con el casino esta expuesta en la seccion Invitado)

Presentamos aqui diagramas de secuencia del correcto ingreso al Casino Online.



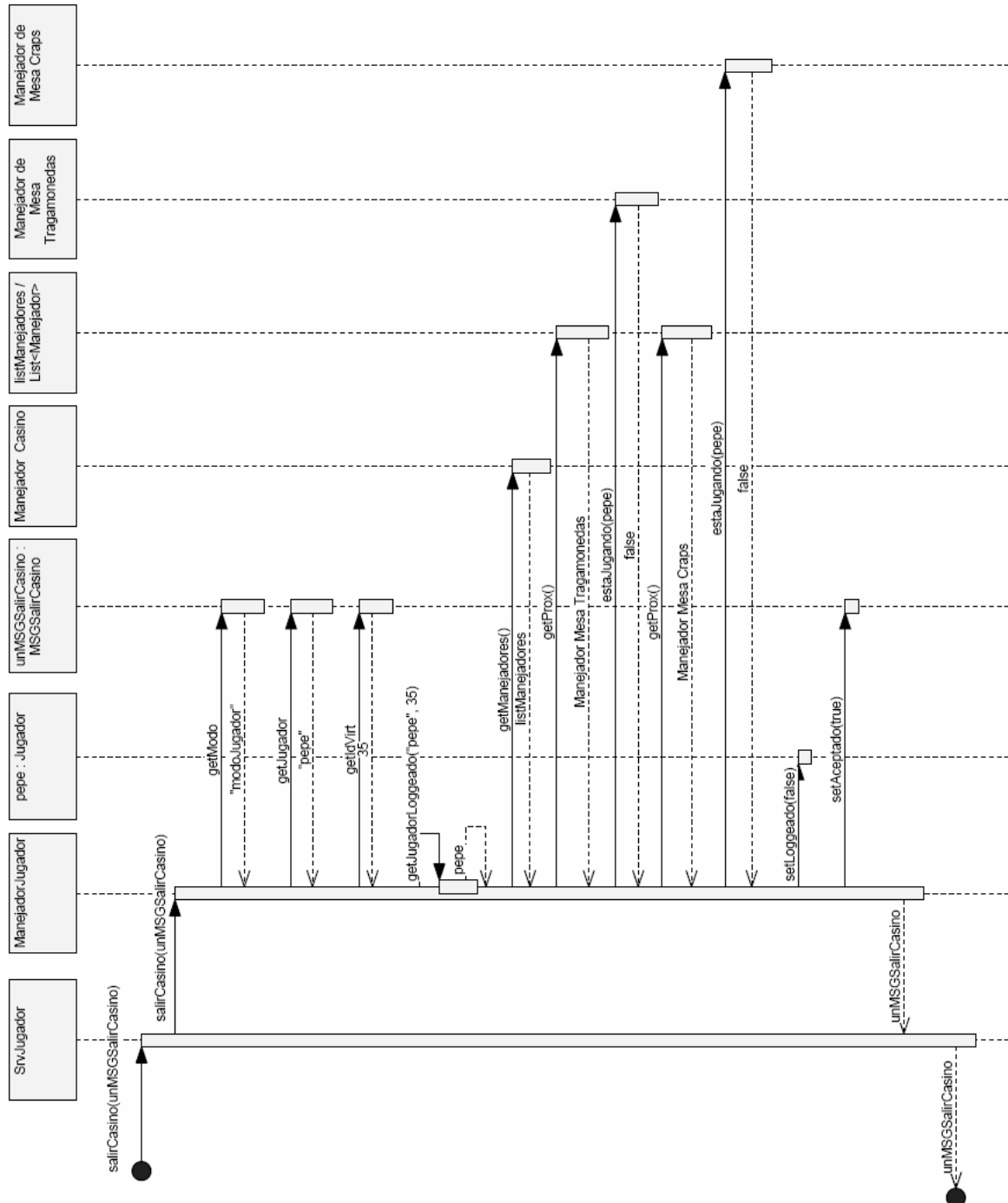
En este diagrama, observamos el mensaje del jugador “Pepe” que intenta loguearse en el casino online como jugador desde que llega al modulo de servicios y es resuelto por el modulo casino. El casino realiza todas las comprobaciones pertinentes y al estar todo “en regla” loggea al jugador y responde que la operacion pudo ser realizada.

### Jugador entrando al casino aceptado

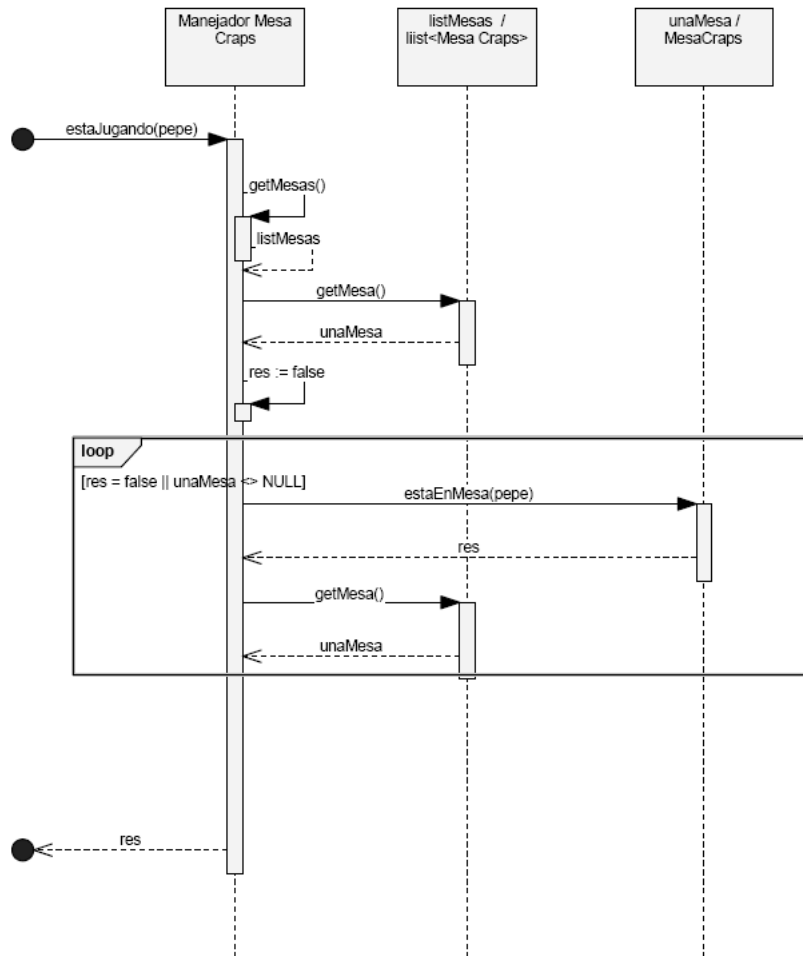


En este diagrama, observamos el mensaje del jugador “Pepe” que intenta desloguearse en el casino online como jugador desde que llega al modulo de servicios y es resuelto por el modulo casino. El casino realiza todas las comprobaciones pertinentes y al estar todo “en regla” desloggea al jugador y responde que la operacion pudo ser realizada.

### Jugador saliendo del casino aceptado



## Esta Jugando



Si así lo desea, un jugador puede solicitarle al casino su estado. Ver 4.8

## 4.2. Administracion del Casino

### 4.2.1. Apertura del casino

La apertura del casino cuenta con algunos archivos de configuración. Los archivos vendran con la implementacion del casino y podran ser modificados por el administrador antes de abrir el casino.

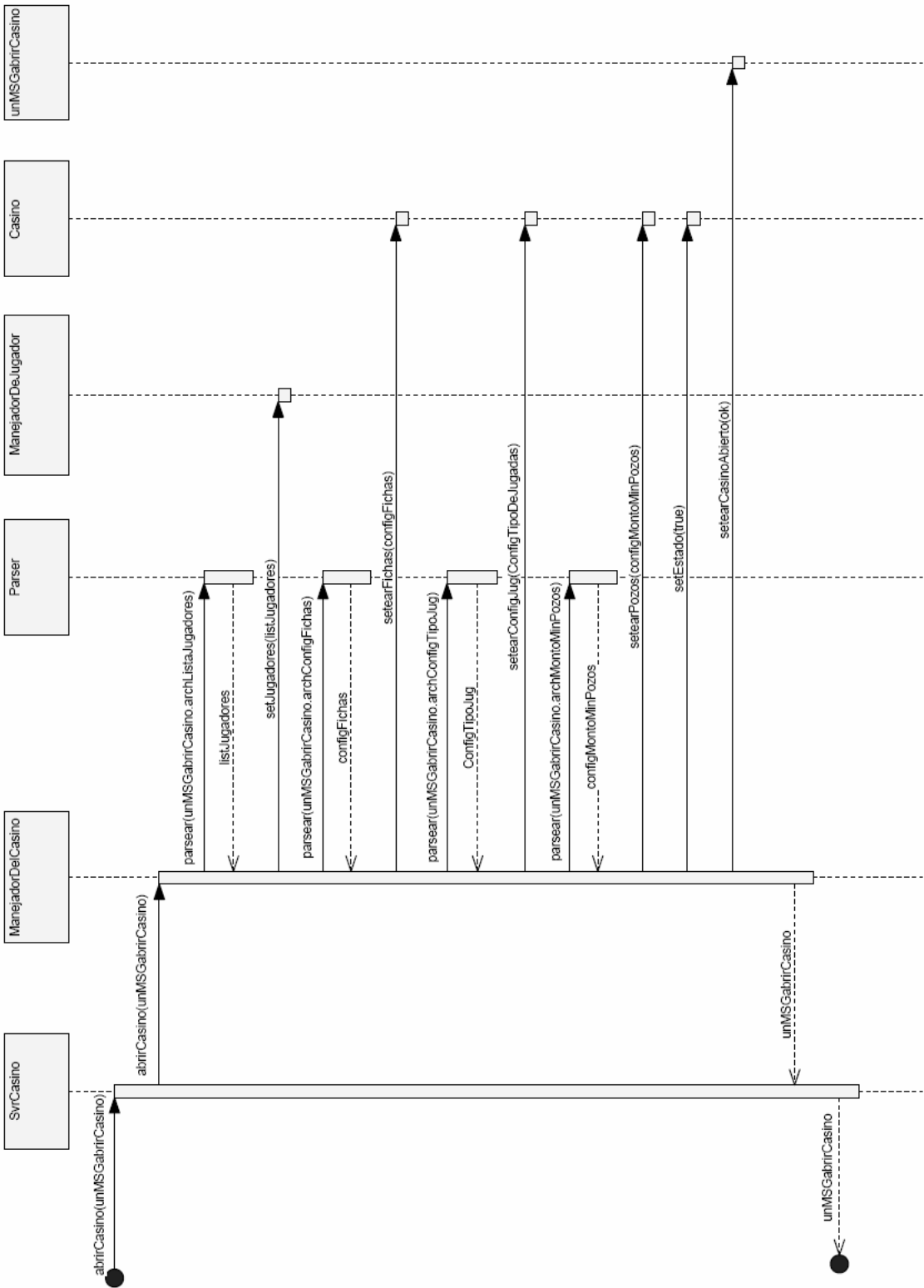
Estos archivos de momento tendrán formato XML. Este es un ejemplo de uno de estos archivos.

```
<valorFichas>
    <valorFicha valor="" />
</valorFichas>
<probablilidades>
    <jugadaFeliz proba="" montoMinimo="" />
    <jugadaTodosPonen proba="" />
    <jugadaNormal proba="" />
    <craps>
        <valorDado numero="1" proba="" />
        .
        .
        <valorDado numero="6" proba="" />
    </craps>
    <traga>
        <gordoProgresivo montoMinimo="" descuento=""/>
        <combinaciones>
            <simbolo1= "" proba=""/>
            .
            .
            <simboloN= "" proba=""/>

        </combinaciones>
    </traga>
</probablilidades>
```

Veamos un diagrama donde se realiza la apertura del casino y se muestra la carga de estos archivos.

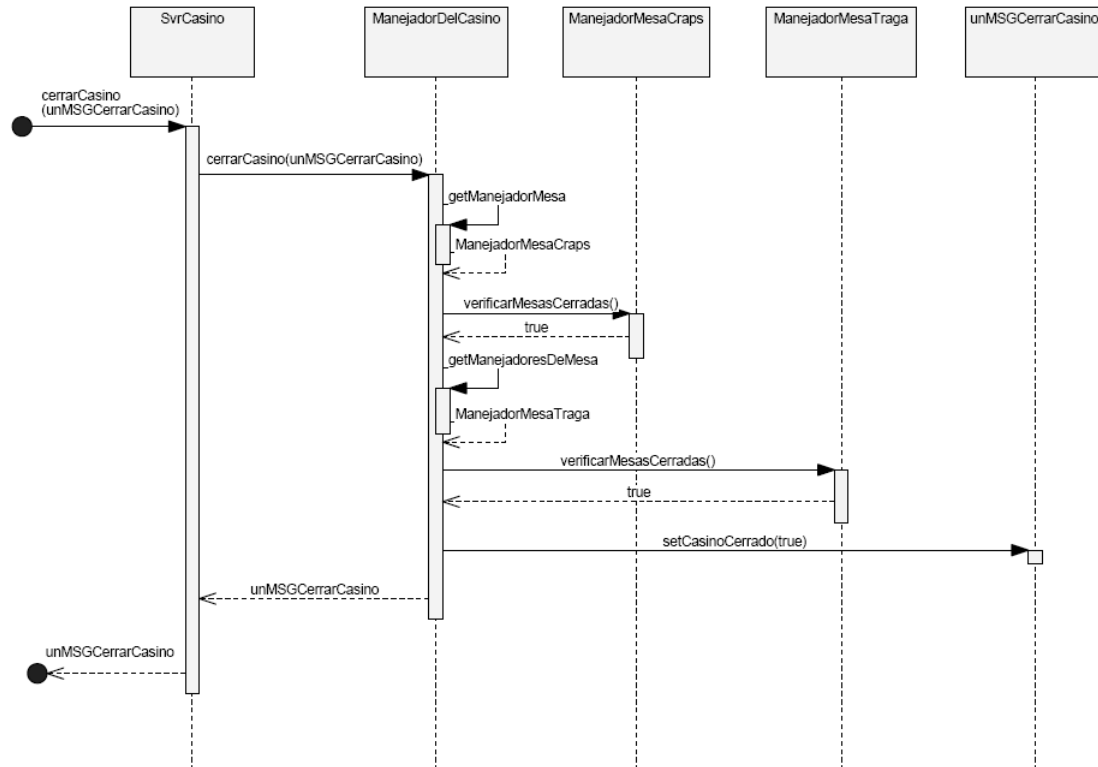
Abrir Casino



#### 4.2.2. Clausura del casino

Veamos también un diagrama en el que el administrador procede a cerrar el casino.

##### Cerrar Casino

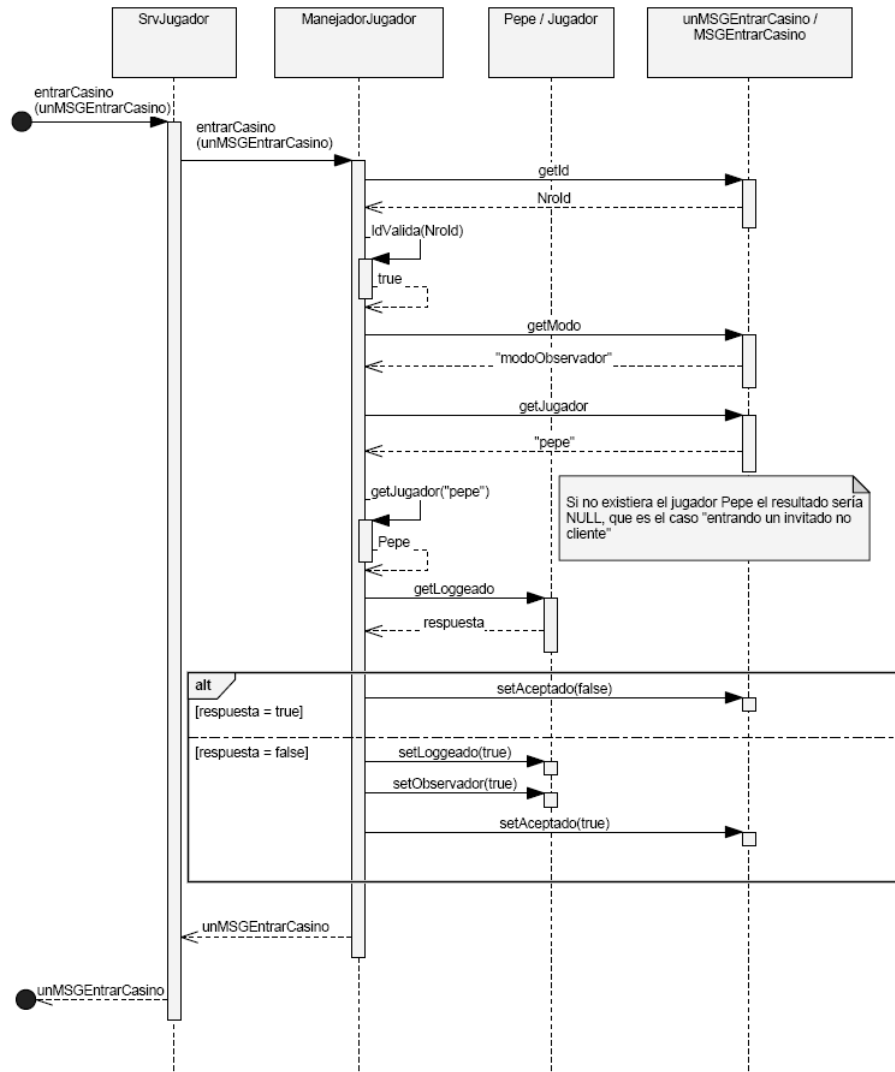


### 4.3. Invitado

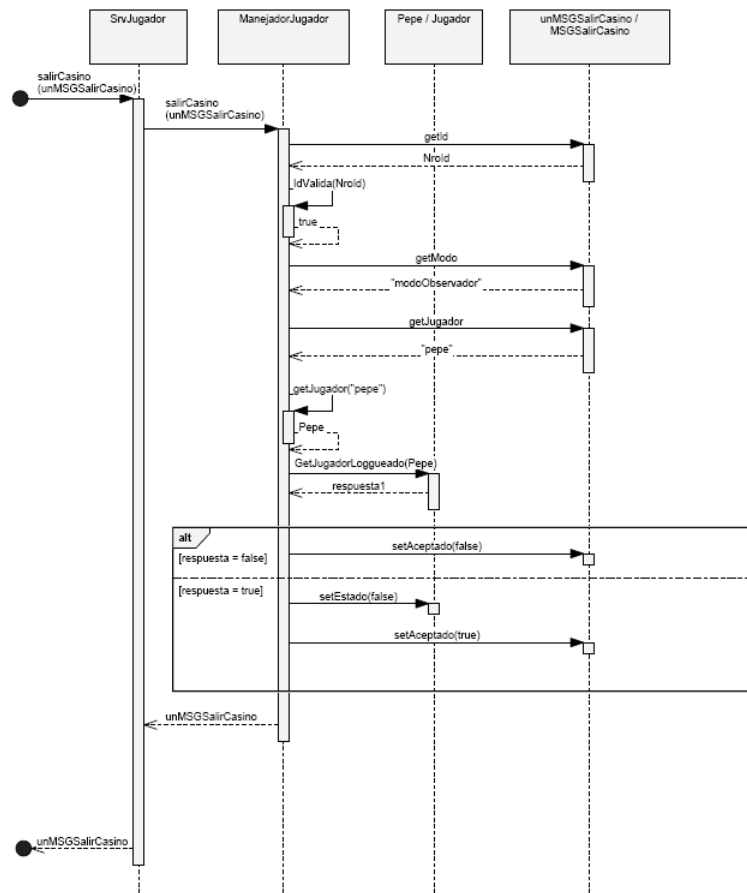
En esta sección estamos apuntando al ingreso de invitados al casino.

Presentamos aquí diagramas de secuencia del correcto ingreso y egreso de un invitado al casino online.

Cliente del casino ingresando al mismo como invitado, aceptado

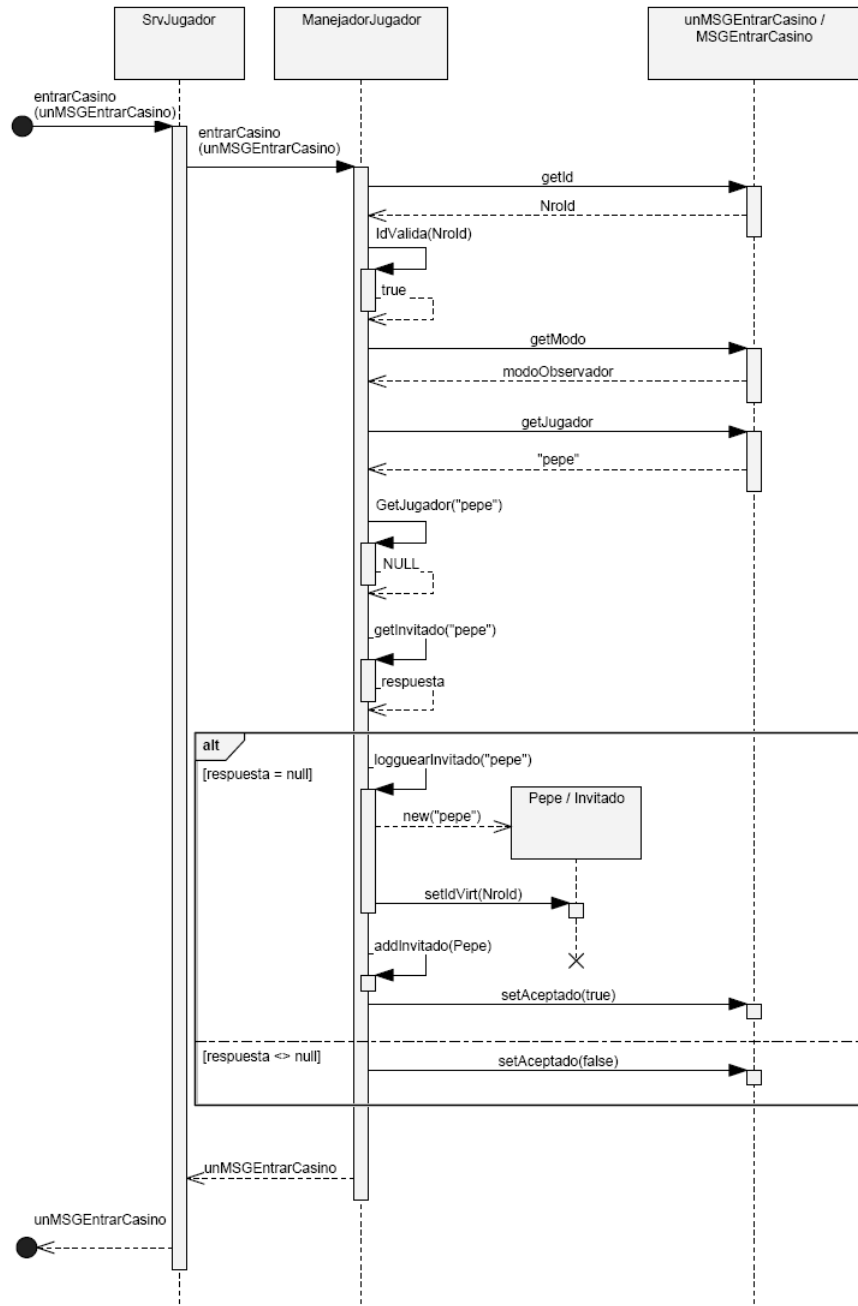


Cliente del casino saliendo del mismo como invitado, aceptado

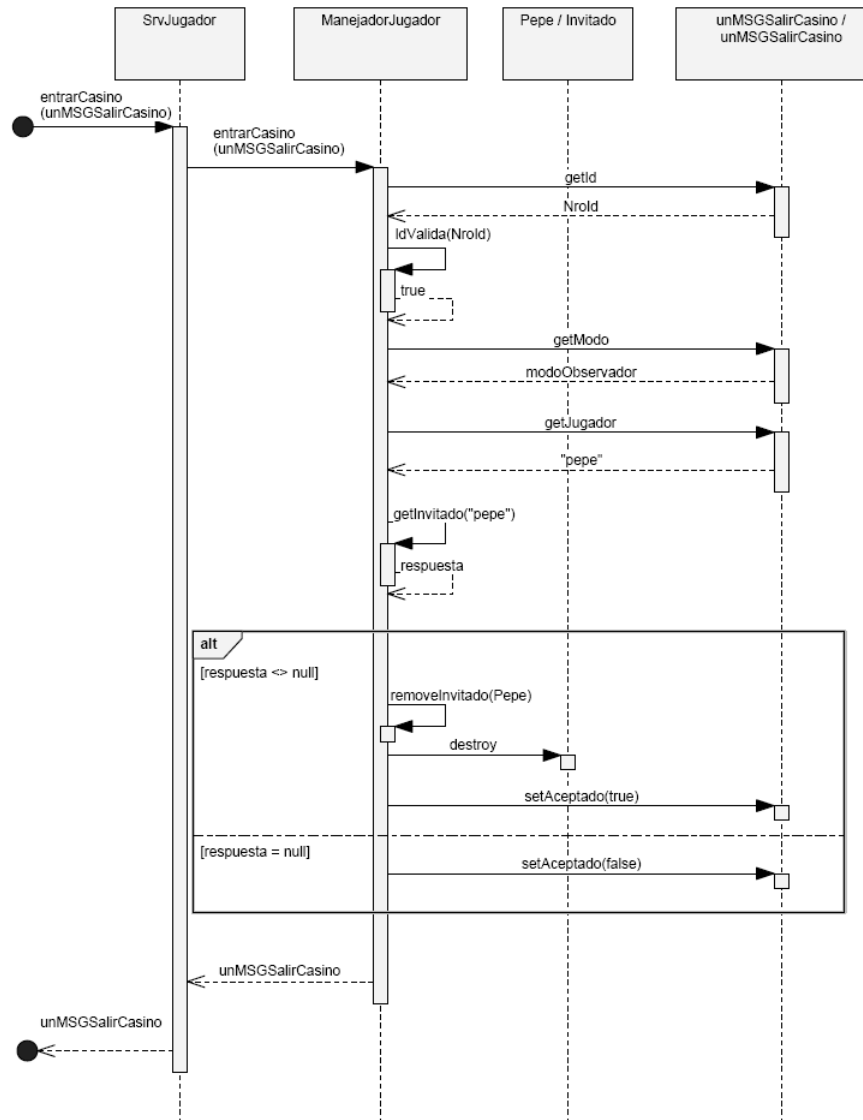




## Jugador ingresando al casino como invitado, aceptado



## Jugador del casino saliendo del casino como invitado, aceptado



Si así lo requiere, el invitado puede pedir el estado del Casino. Ver 4.8

#### 4.4. Modo Dirigido

El casino cuenta con algunos archivos de configuración para establecer el modo en que estará. Los archivos vendrán con la implementación del casino y podrán ser modificados por el administrador. Estos archivos de momento tendrán formato XML. Estos son ejemplos de algunos de estos archivos.

```
<setearModo>
  <modo>dirigido</modo>
  <resultados>
<resultadosTragamonedas>
"dino,dino,dino"
"cereza,cereza,bar"
"bar,bar,bar"
"doble bar,bar,doble bar"
...
</resultadosTragamonedas>
<resultadosCraps>
"1,7,3,6,12,11,3,6,3..."
</resultadosCraps>
...
  </resultados>
  <aceptado></aceptado>
</setearModo>
```

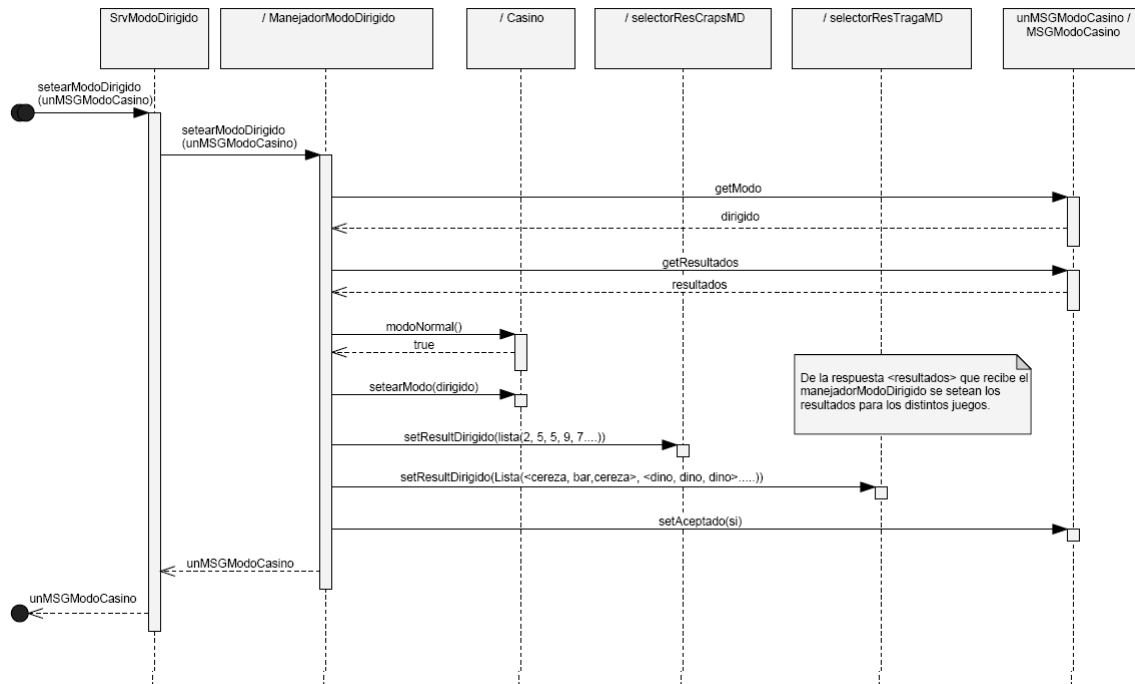
```
<setearModo>
  <modo>normal</modo>
  <resultados></resultados>
  <aceptado>ok </aceptado>
</setearModo>
```

```
<setearJugada>
  <tipo de jugada="normal, feliz, todosPonen">
  </tipo de jugada>
  <juego="nombre de juego">
  </juego>
  <mesa="id de la mesa">
  </mesa>
  <aceptado>si o no</aceptado>
  <descripcion></descripcion>
</setearJugada>
```

#### 4.4.1. Inicio de Modo Dirigido

En el momento de entrar en modo dirigido, se deberán configurar el resultado que obtendrán los juegos y qué jugadas deberán ir saliendo, ya que a partir de ese momento el azar no intervendrá en estos eventos.

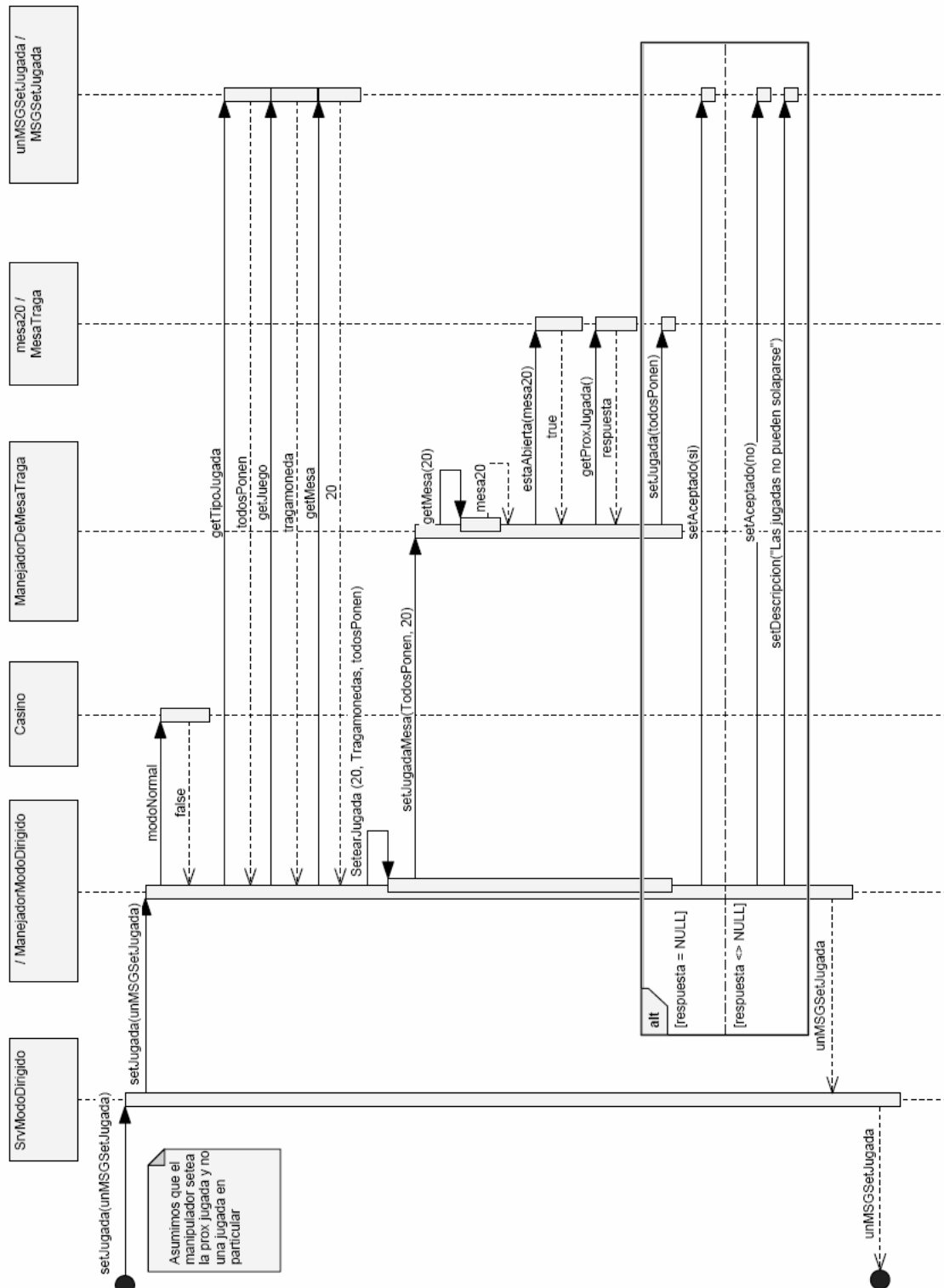
##### Inicio de Modo Dirigido



#### 4.4.2. Seteo de Jugadas

Una vez que el modo dirigido esta habilitado, se podra decidir la ocurrencia de jugadas felices y todos ponen para un juego y mesa particular.

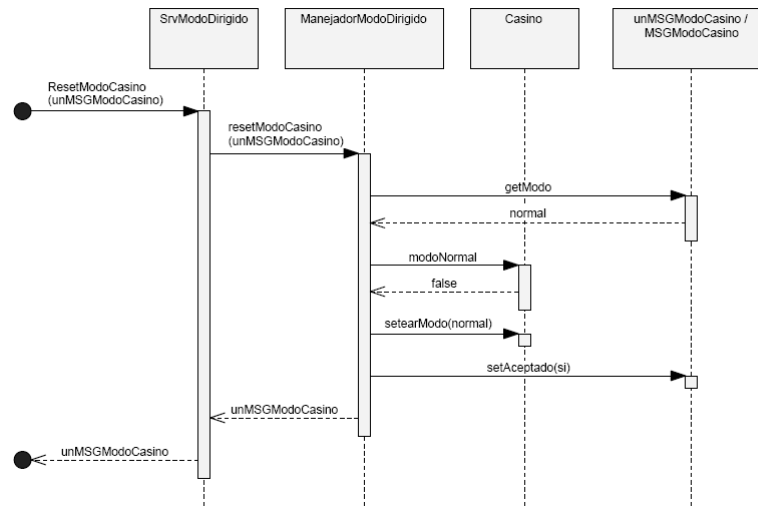
##### Seteo de Jugada



#### 4.4.3. Finalización de Modo Dirigido

Escenario donde se deshabilita el modo dirigido para volver al modo normal. Así se habilitaran nuevamente los resultados y seteo de Tipo de jugada azarosos.

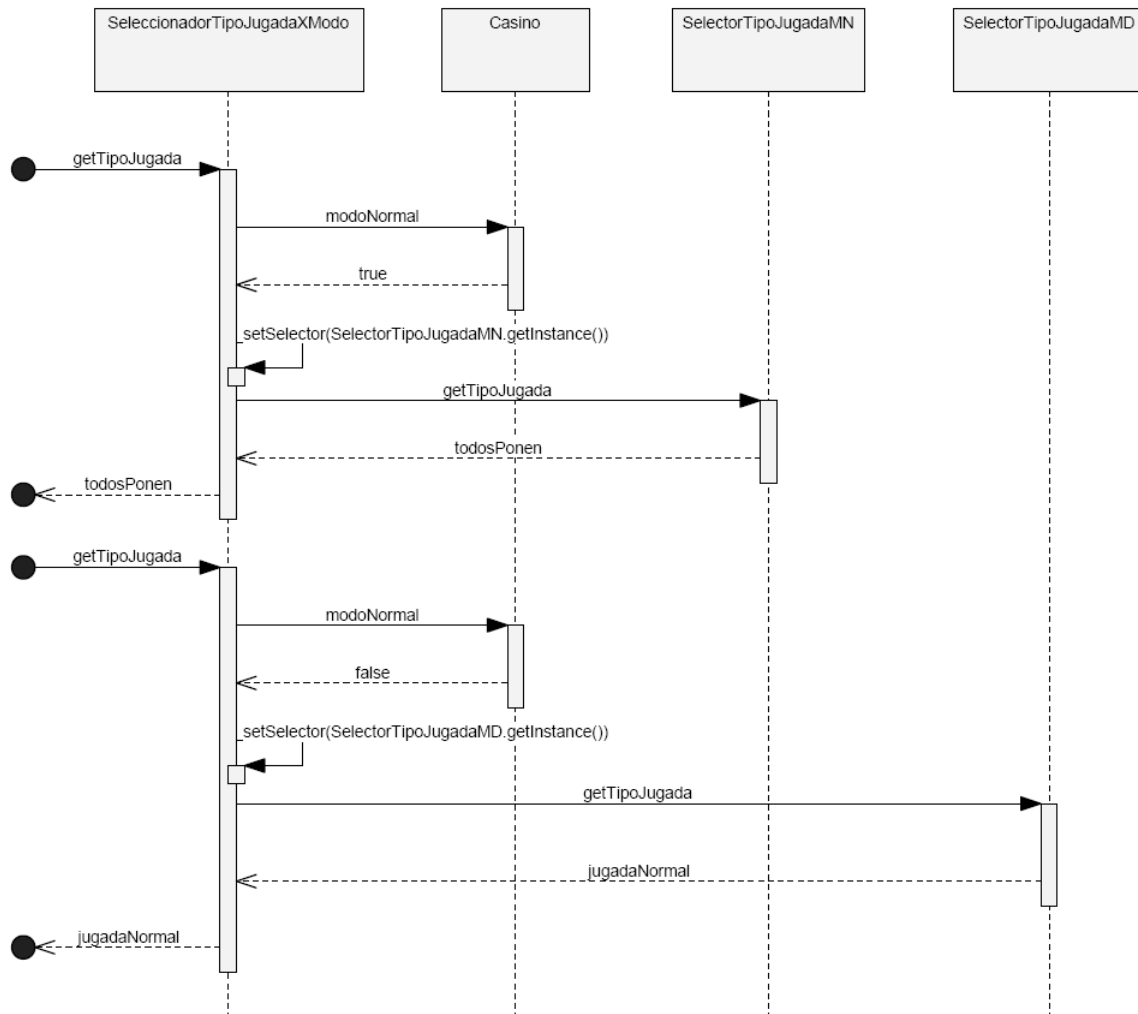
##### Finalización de Modo Dirigido



#### 4.5. Seleccionador de Tipo de jugada

El seleccionador de Tipo de jugada se encarga de, en un determinado momento, seleccionar el tipo de jugada para una jugada de una mesa. En el momento de la creación de una jugada, se invoca al seleccionador que según el criterio de selección (por ej: el modo en que se encuentra el casino), seleccionará a un selector de tipo de jugada (por ej: Selector aleatorio - Selector probabilístico) que elige un tipo de jugada.

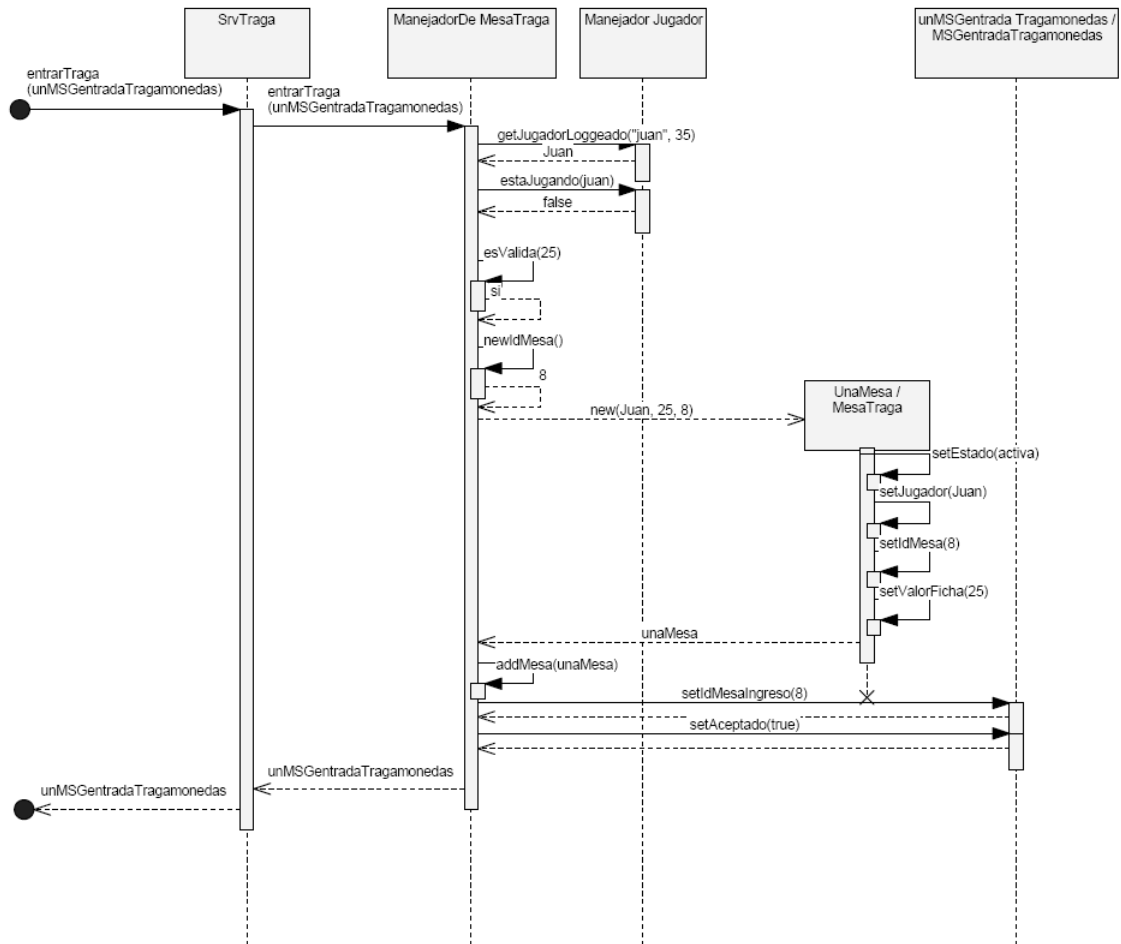
## Seleccionador de Tipo de Jugada



## 4.6. Juego Tragamonedas

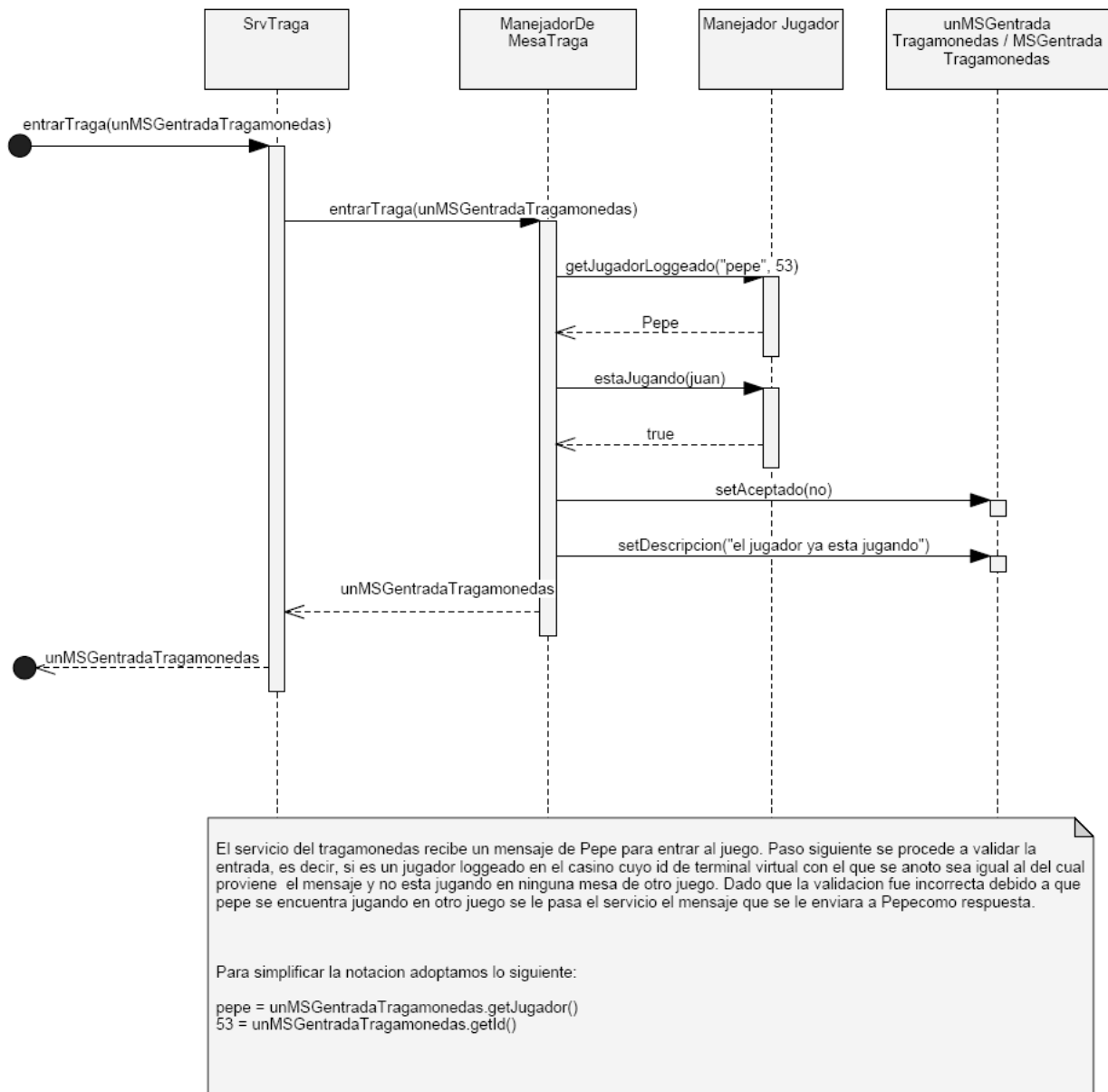
### 4.6.1. Entrar en Tragamonedas

Entrar en Traga Aceptado





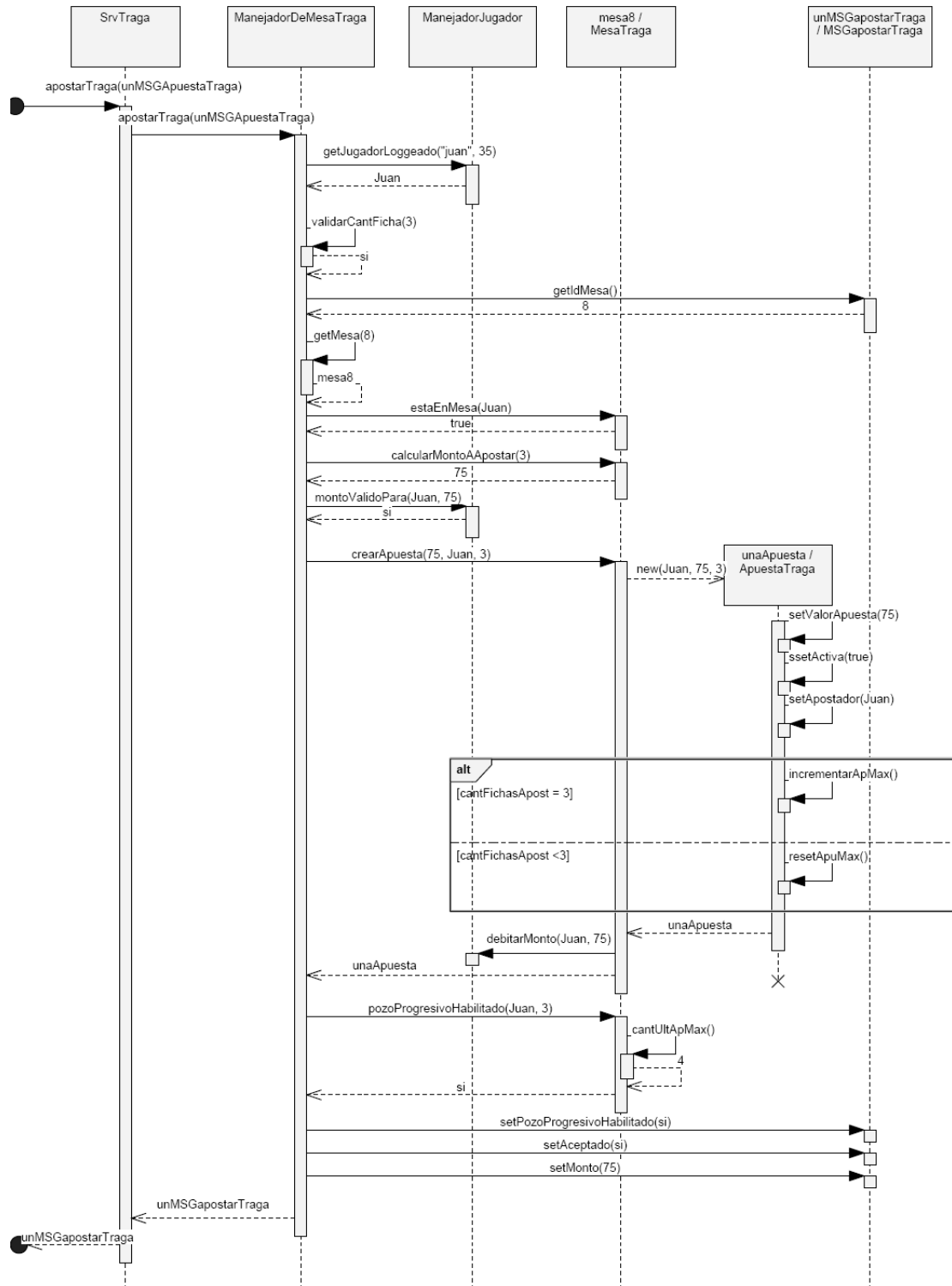
## Entrar en Traga Denegado



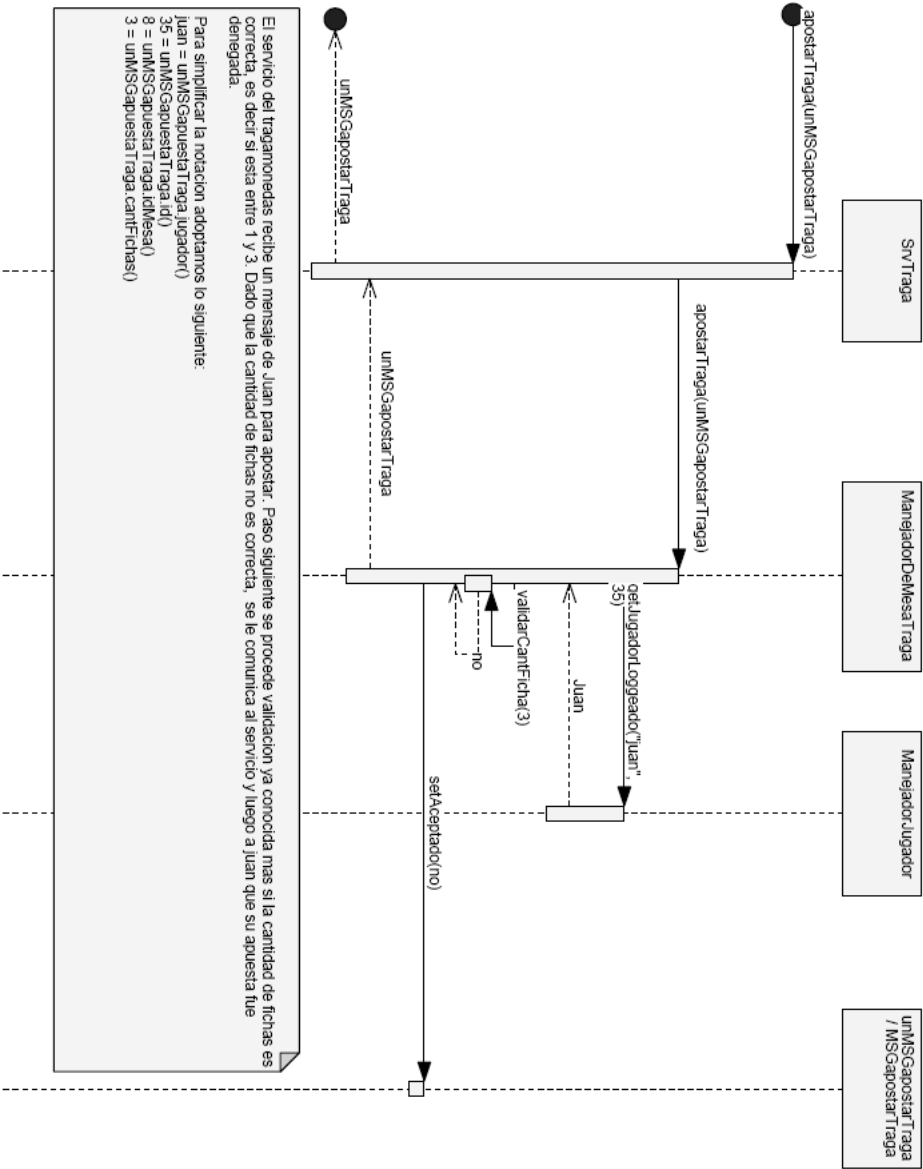
La funcionalidad EstaJugando se puede ver en 4.7.1

## 4.6.2. Apostar en Tragamonedas

## Apostar en Traga Aceptado

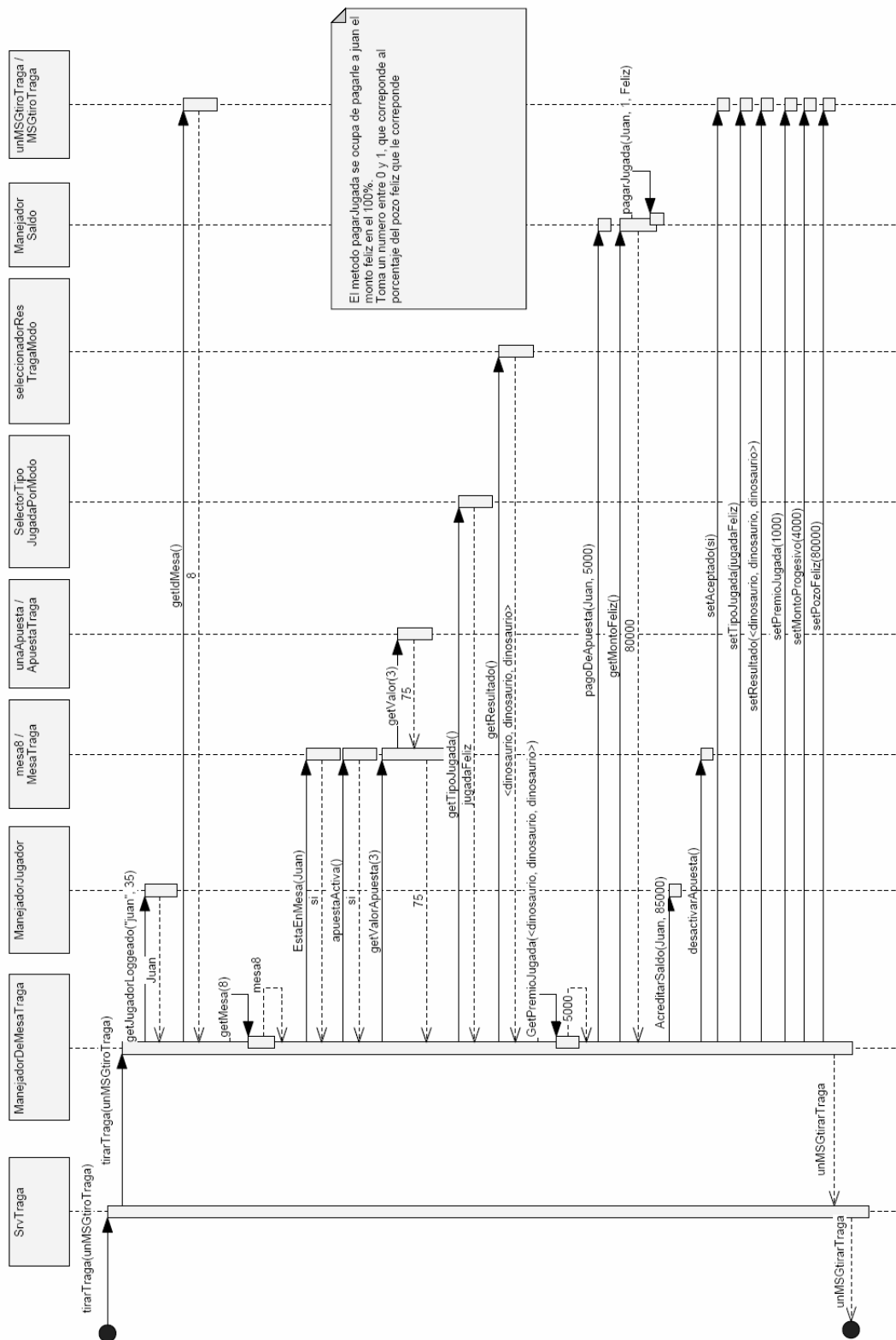


Apostar en Traga Denegado

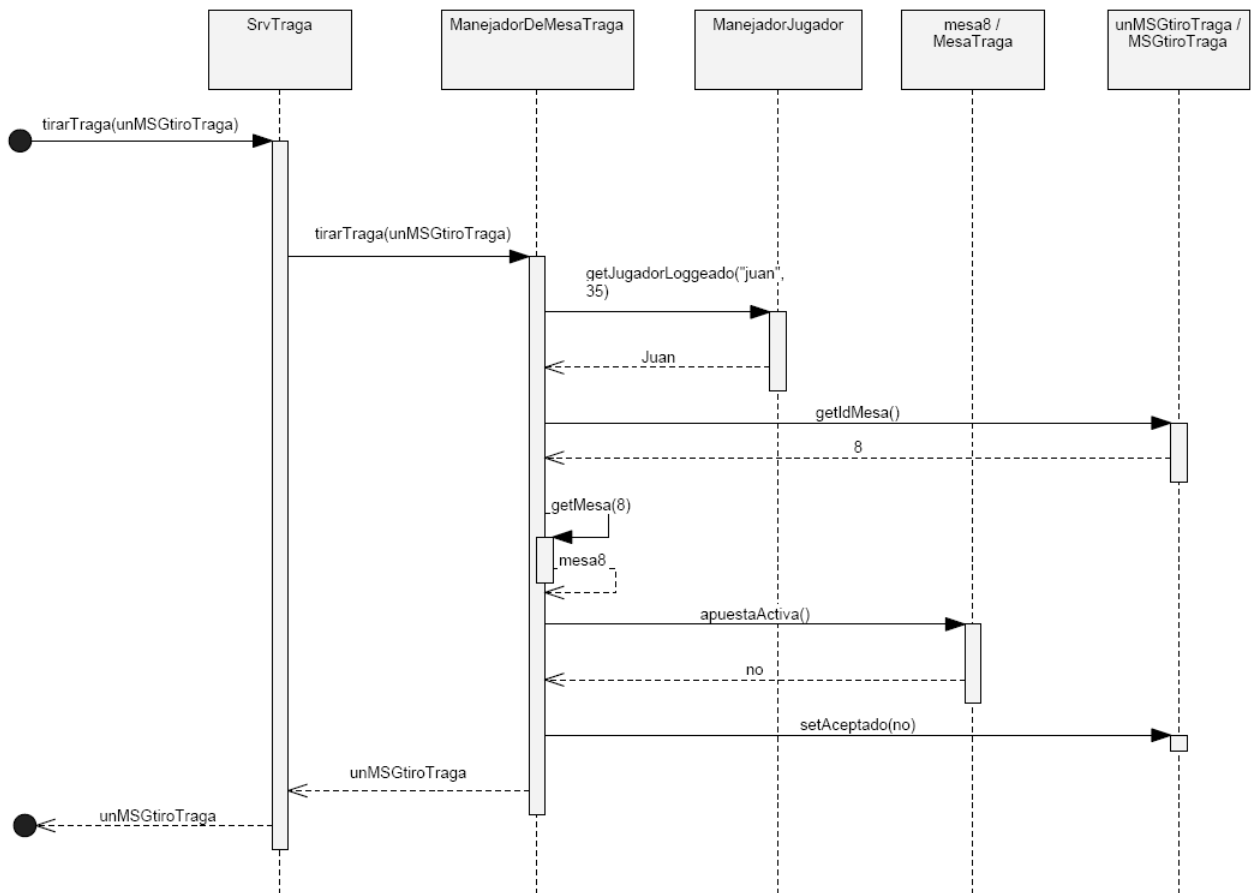


4.6.3. Tirar en Tragamonedas

Tirar en Traga Aceptado



## Tirar en Traga Denegado



El servicio del tragamonedas recibe un mensaje de Juan para tirar. Dado que Juan no hizo ninguna apuesta en la mesa, no se puede efectuar el tiro

Para simplificar la notación adoptamos lo siguiente:

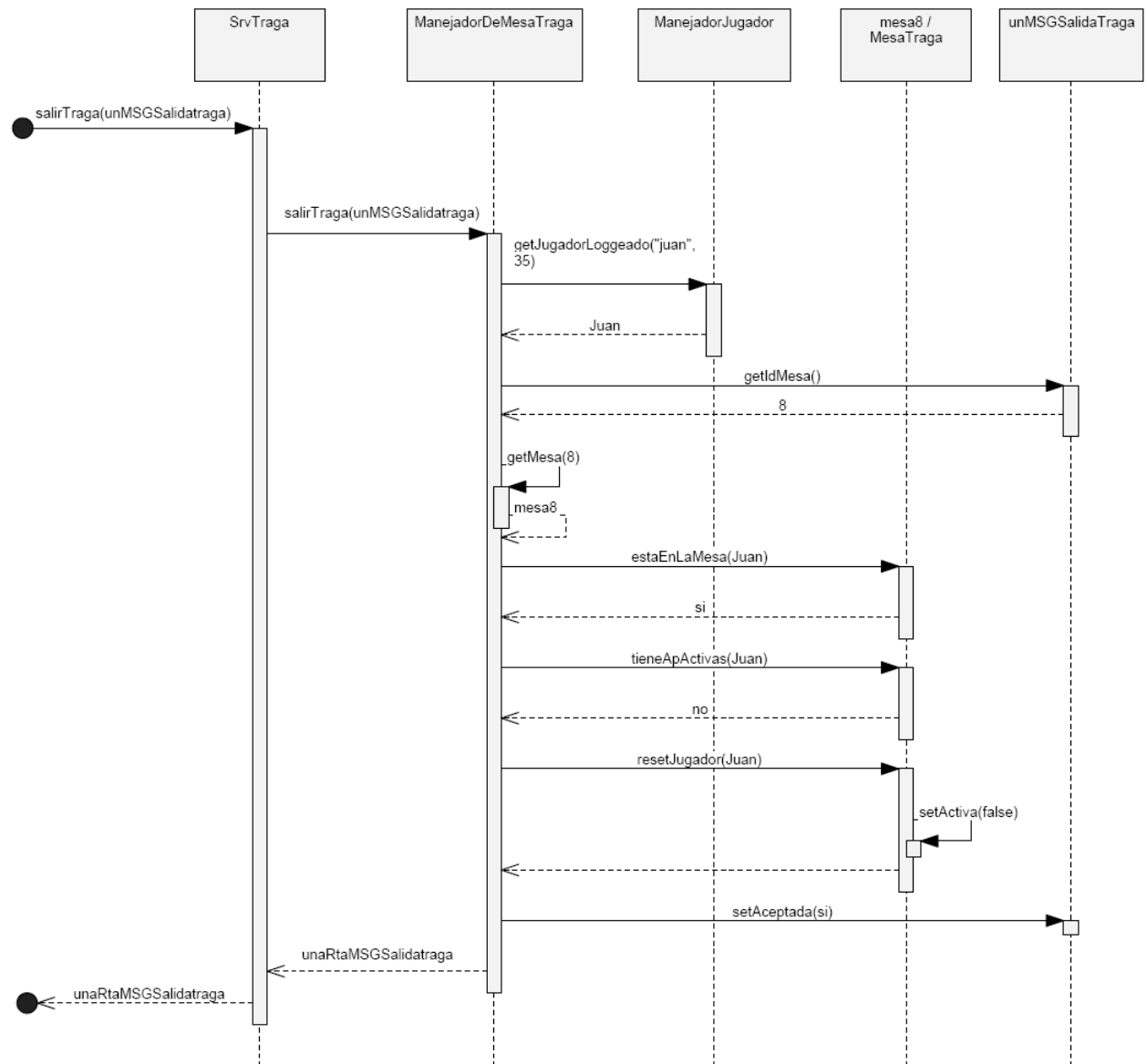
juan = unMSGapuestaTraga.jugador()

35 = unMSGapuestaTraga.id()

8 = unMSGapuestaTraga.idMesa()

## 4.6.4. Salir en Tragamonedas

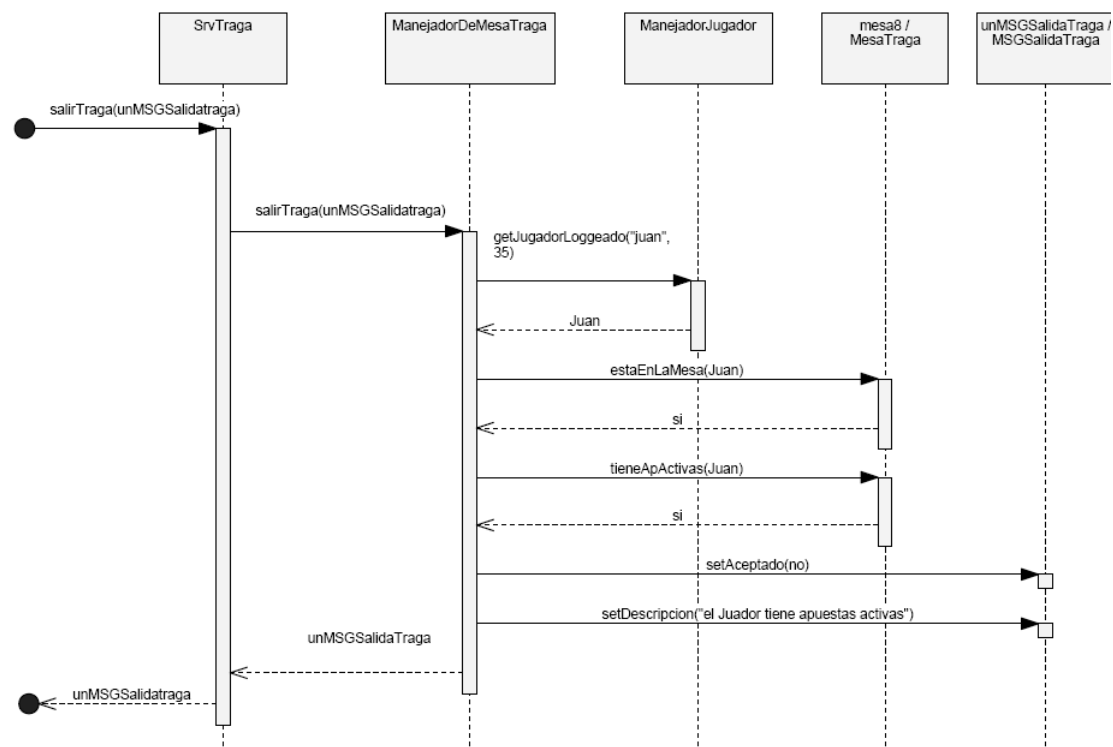
## Salir en Traga Aceptado



El servicio del tragamonedas recibe un mensaje de Juan para salir del juego. Paso siguiente se procede a validar la salida, es decir, si es esta loggeado en el casino, la id de terminal virtual con el que se anoto sea igual al del cual proviene el mensaje, estejugando en la mesa 8. pero no tenga apuestas activas. Dado que la validacion fue correcta, se procede a quitar el jugador de la mesa 8 y generar el mensaje de salida que se le pasa al servicio para enviaraselo a Juan como respuesta.

Para simplificar la notacion adoptamos lo siguiente:  
 juan = unMSGsalidaTraga.jugador()  
 35 = unMSGsalidaTraga.id()  
 8 = unMSGsalidaTraga.idMesa()

Salir en Traga Denegado



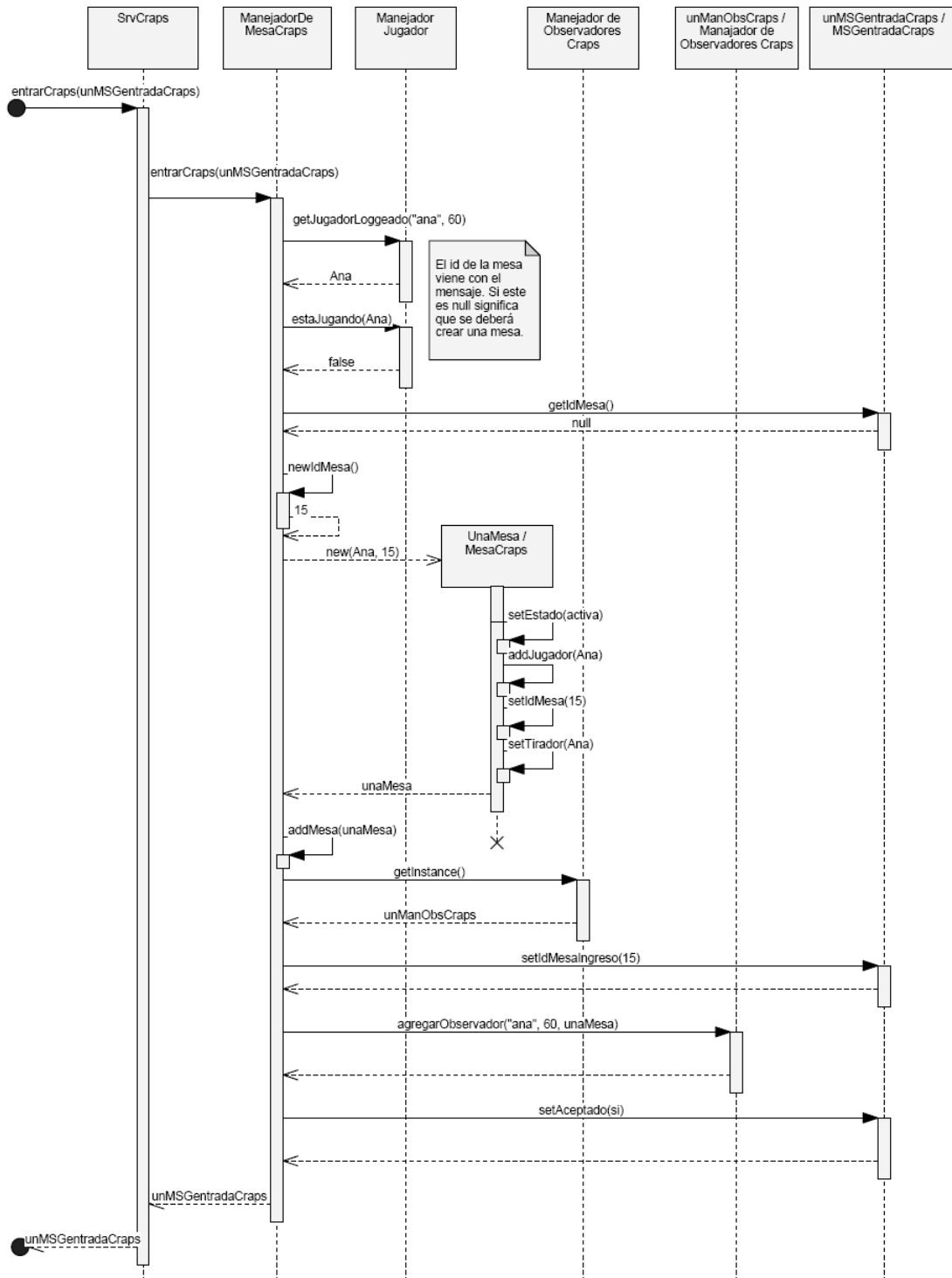
El servicio del tragamonedas recibe un mensaje de Juan para salir del juego. Paso siguiente se procede a validar la salida, es decir, si es esta loggeado en el casino, la id de terminal virtual con el que se anoto sea igual al del cual proviene el mensaje , este jugando en la mesa 8, pero no tenga apuestas activas. Dado que la validacion no fue correcta debido a que el jugador tiene apuestas activas en la mesa, se genera una excepcion con el mensaje de error. Este se le pasa al servicio para enviaraselo a Juan como respuesta.

Para simplificar la notacion adoptamos lo siguiente:  
juan = unMSGsalidaTraga.jugador()  
35 = unMSGsalidaTraga.id()  
8 = unMSGsalidaTraga.idMesa()

## 4.7. Juego Craps

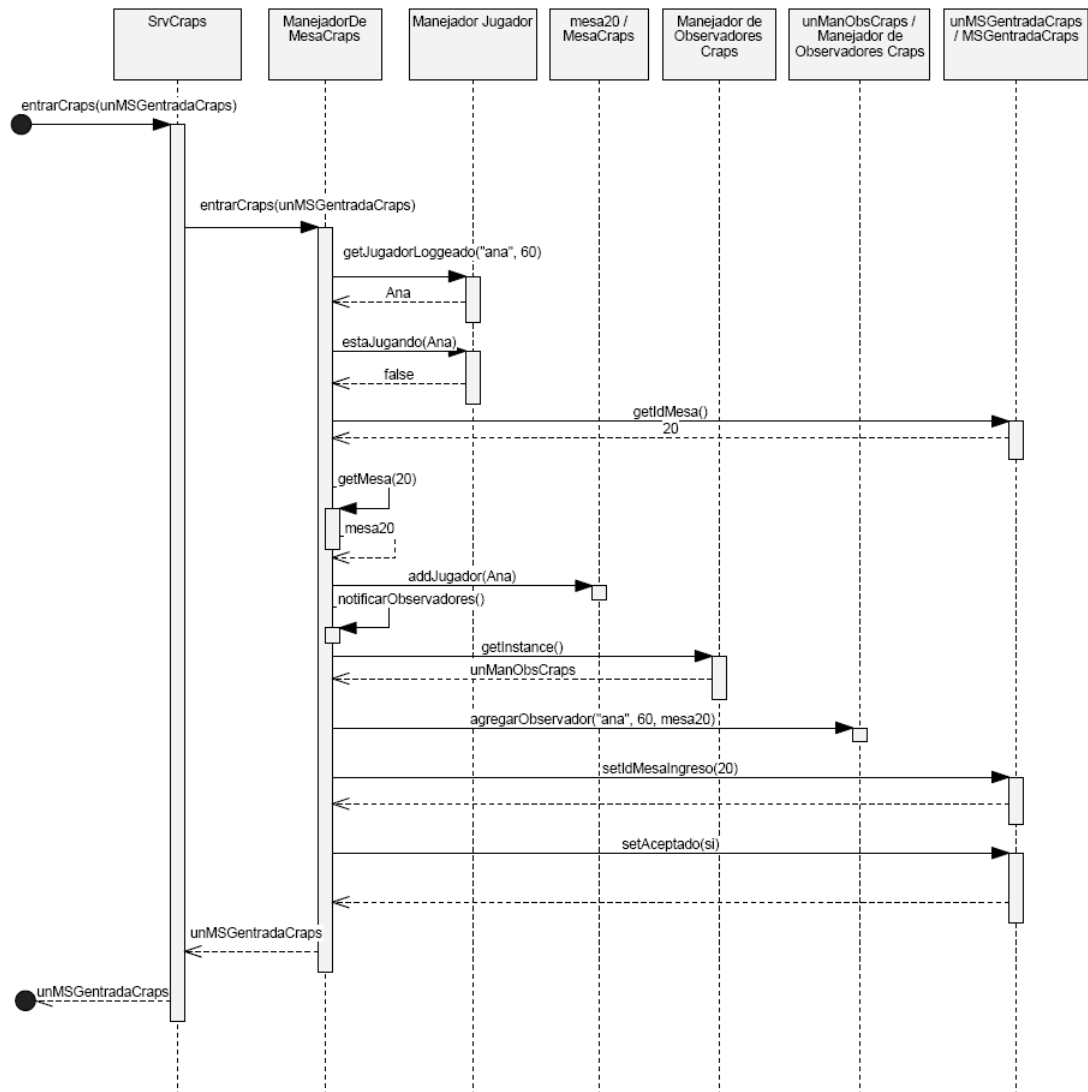
### 4.7.1. Entrar en Craps

Entrar (Abrir mesa) en Craps Aceptado

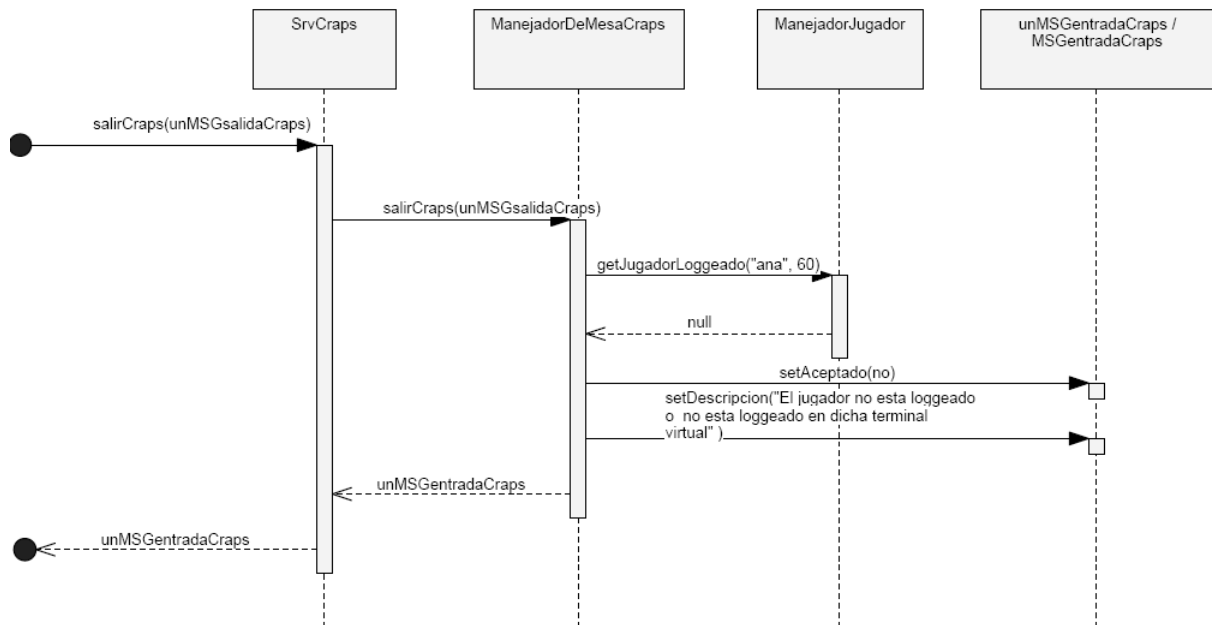




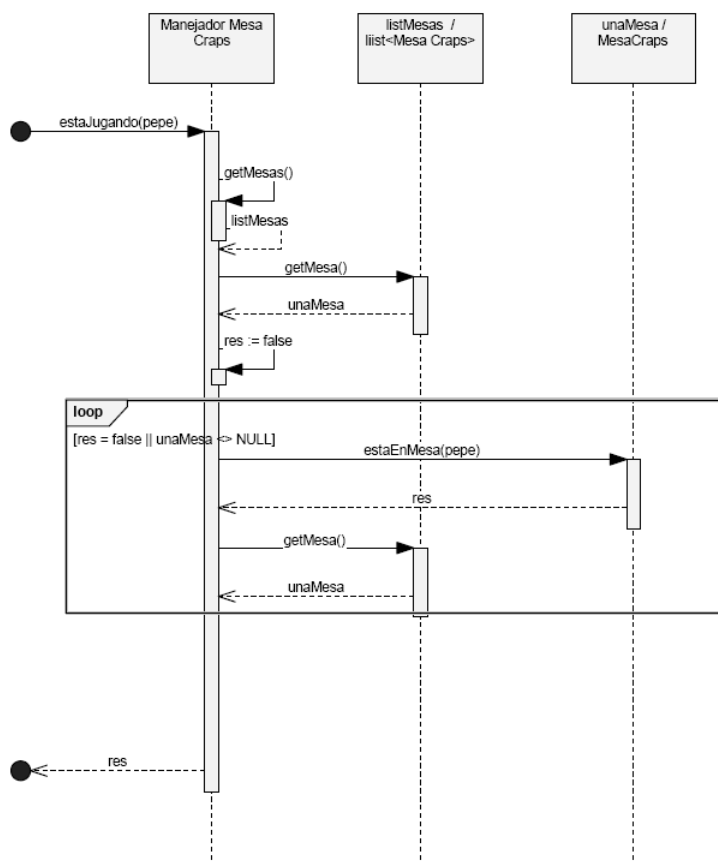
## Entrar (Unirse) en Craps Aceptado



## Entrar en Craps Denegado



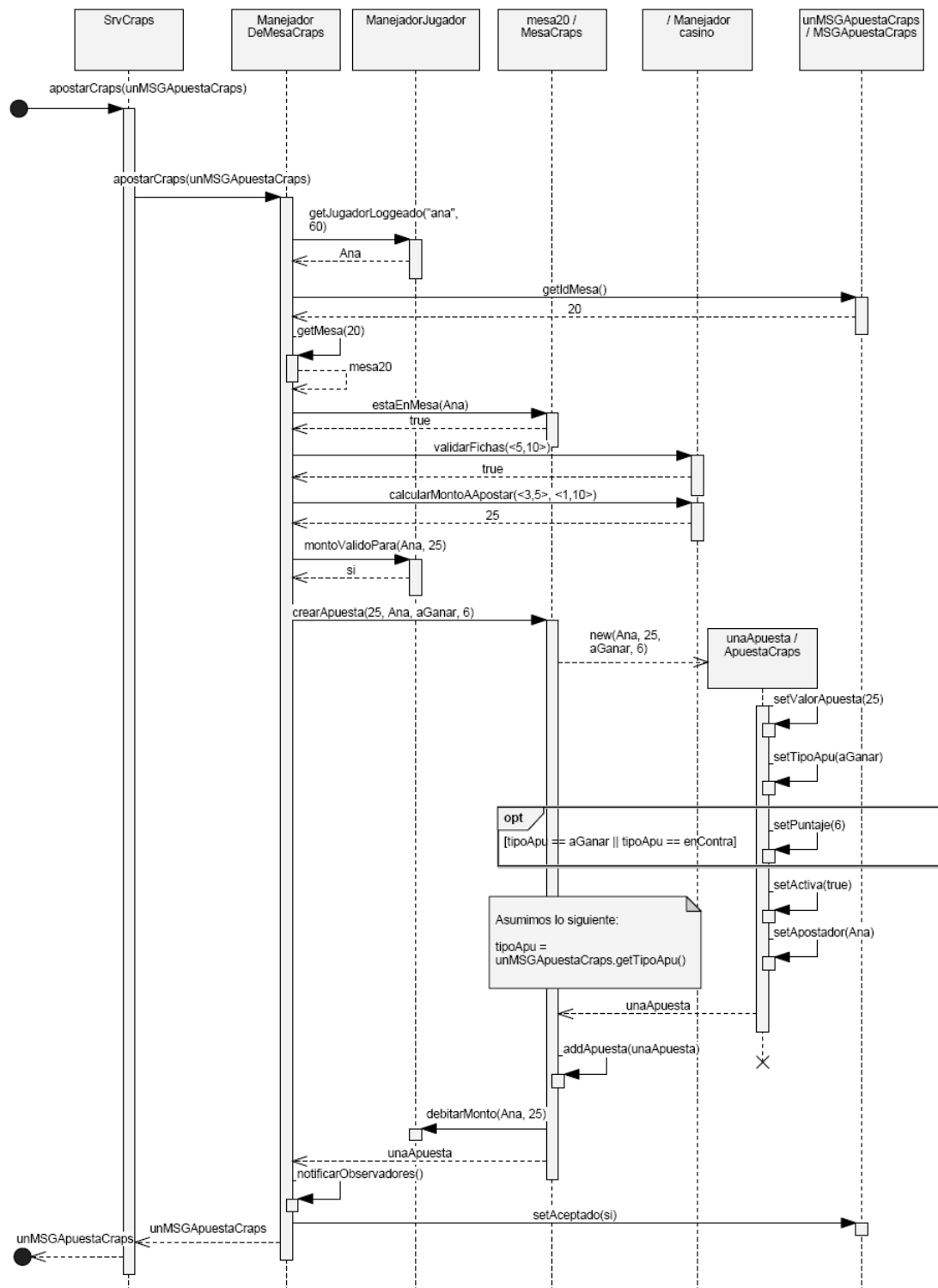
## Esta Jugando



El diagrama anterior describe la funcionalidad que recorre las mesas de craps para averiguar si un jugador dado se encuentra jugando en alguna de las mesas. De manera analoga se procede con las mesas del tragamonedas

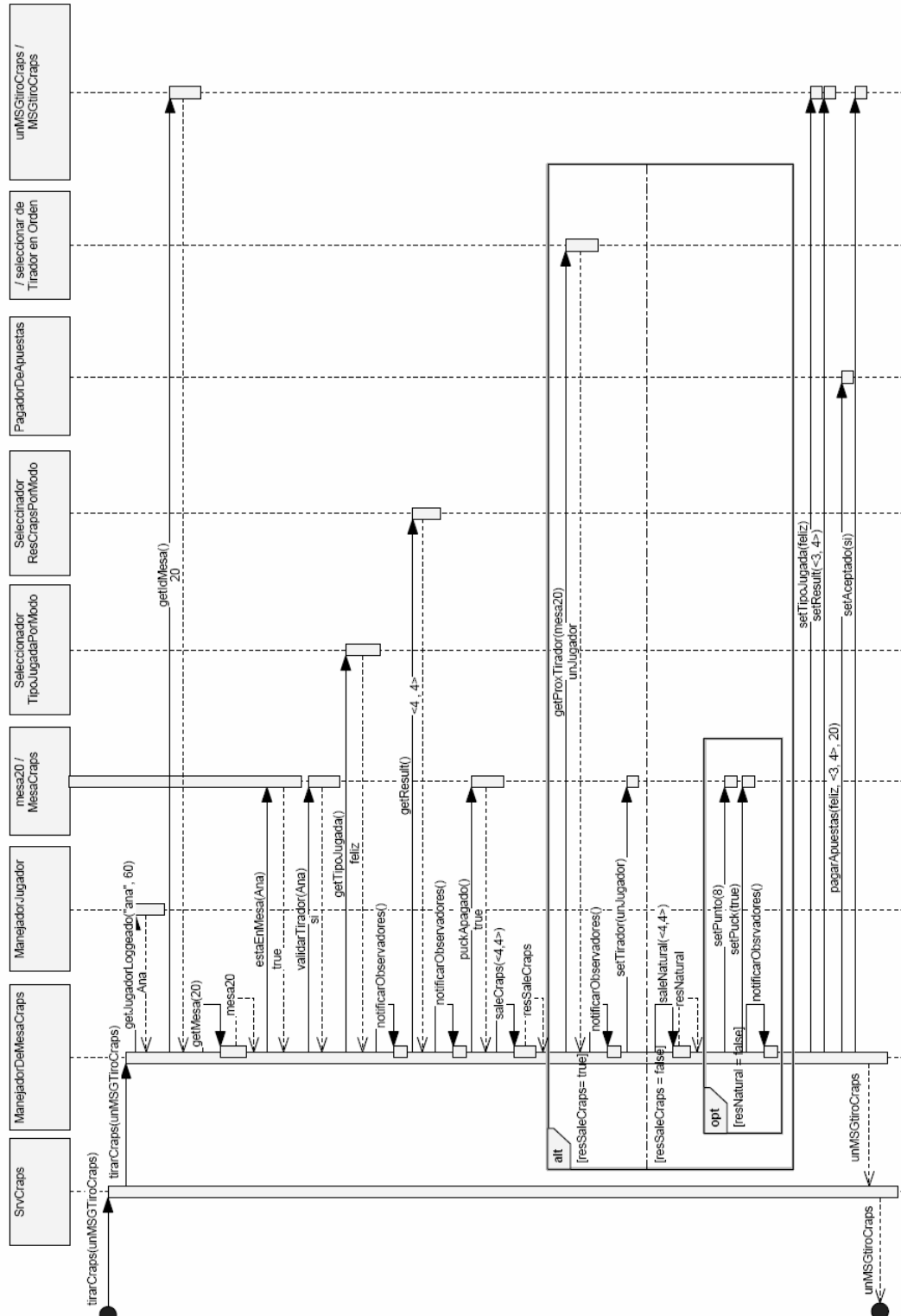
#### 4.7.2. Apostar en Craps

##### Apostar en Craps Aceptado

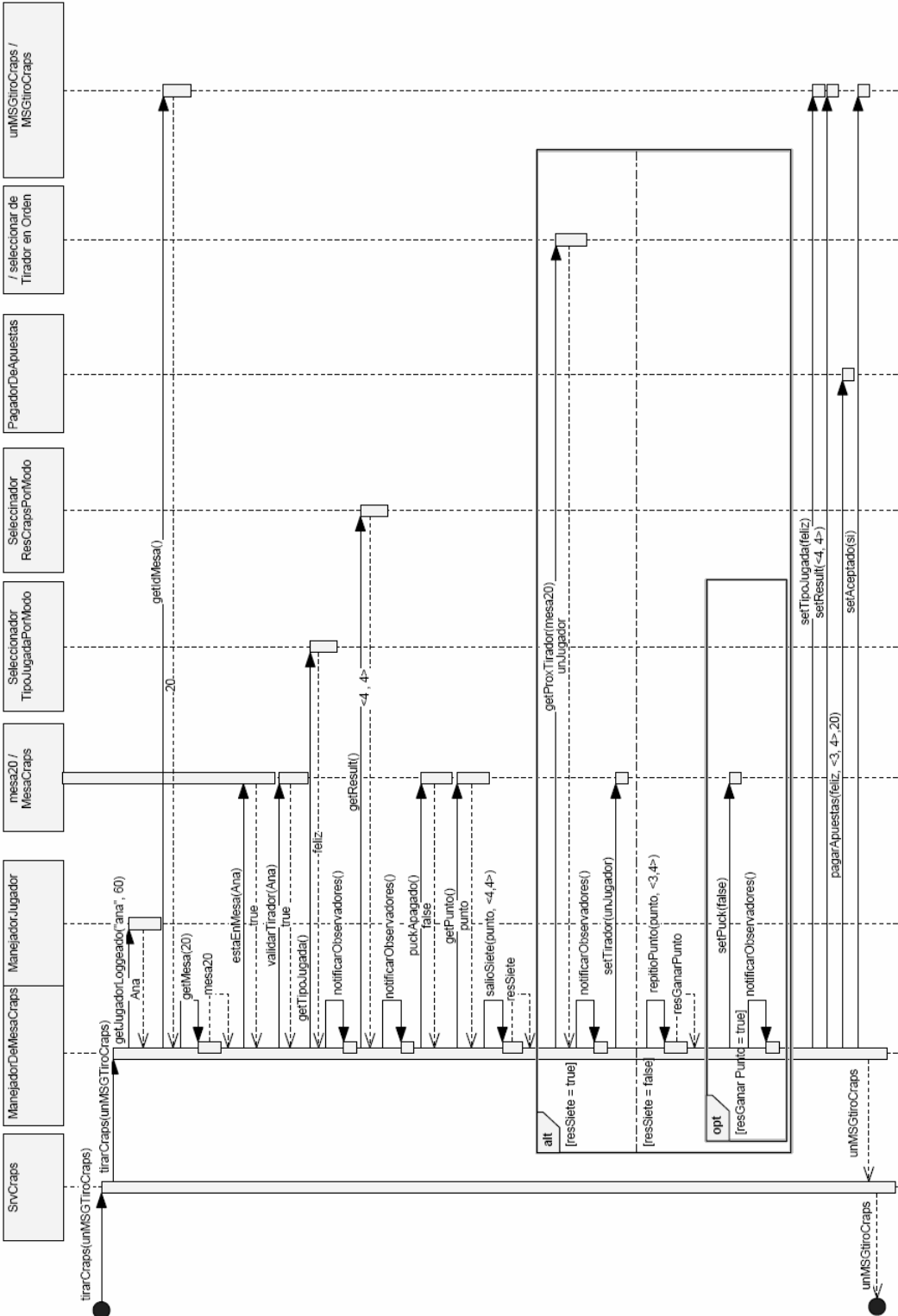


## 4.7.3. Tirar en Craps - Aceptado

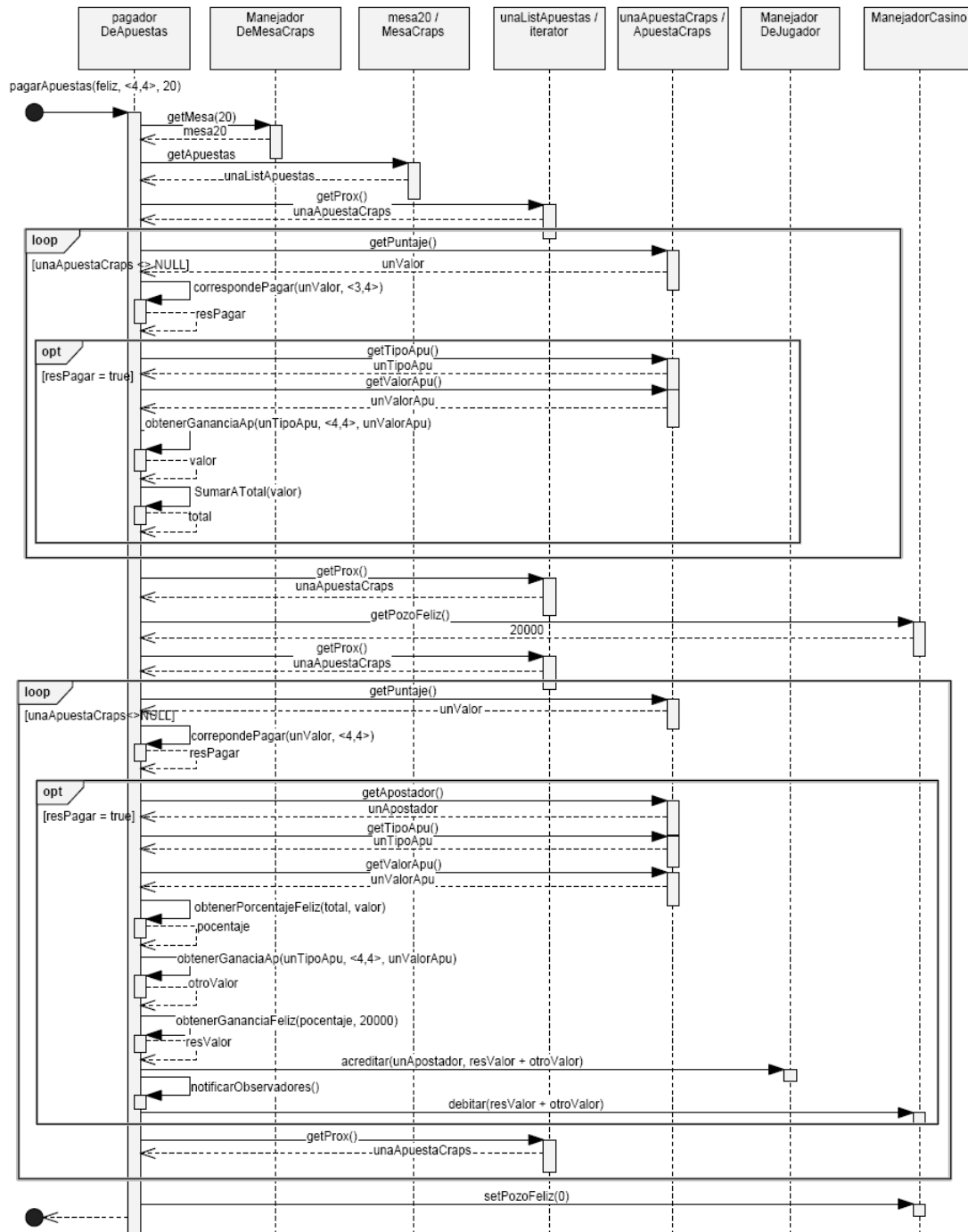
## Tiro de Salida



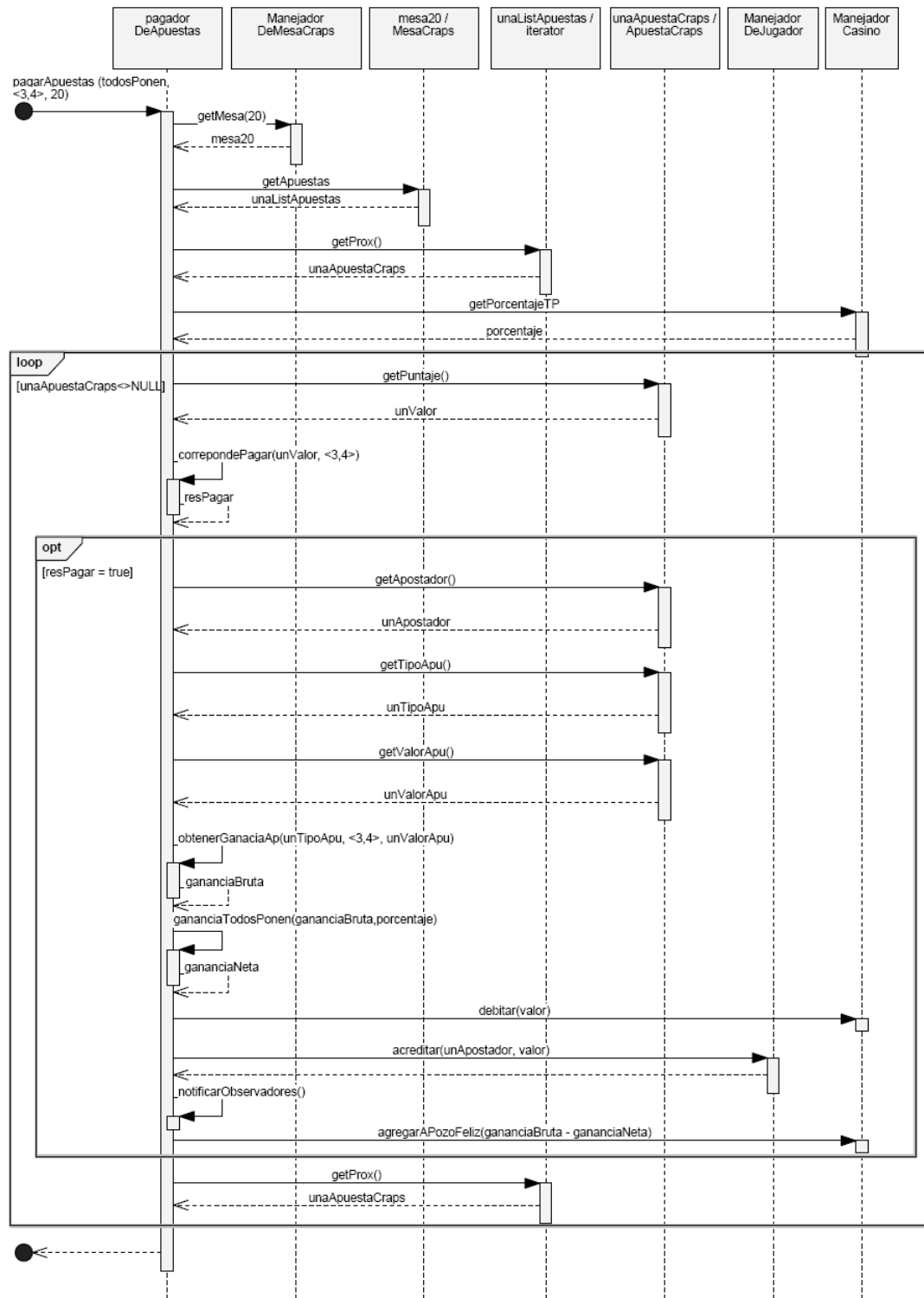
Tiro luego de establecido el punto



## PagarApuestas - Feliz

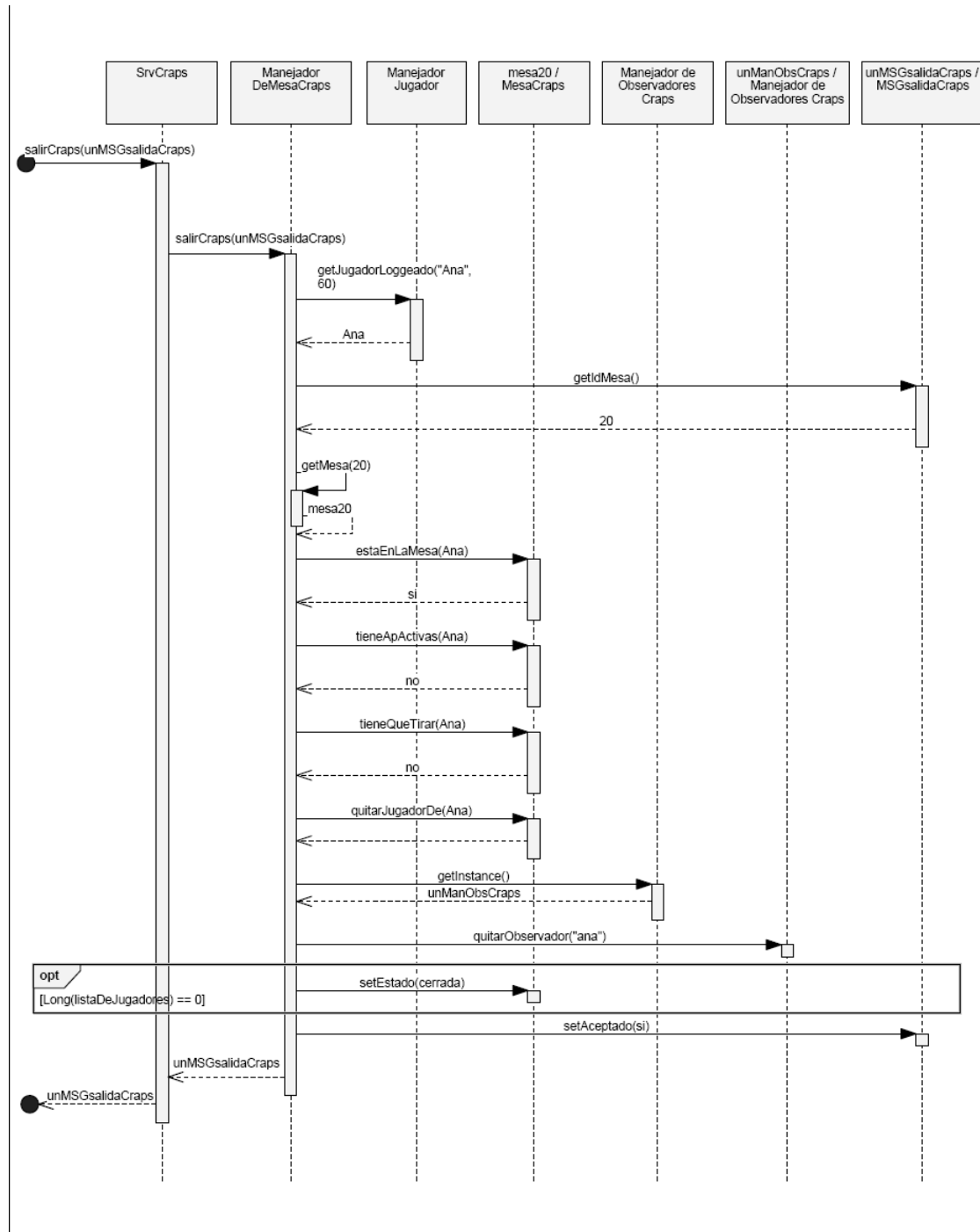


## PagarApuestas - TodosPonen



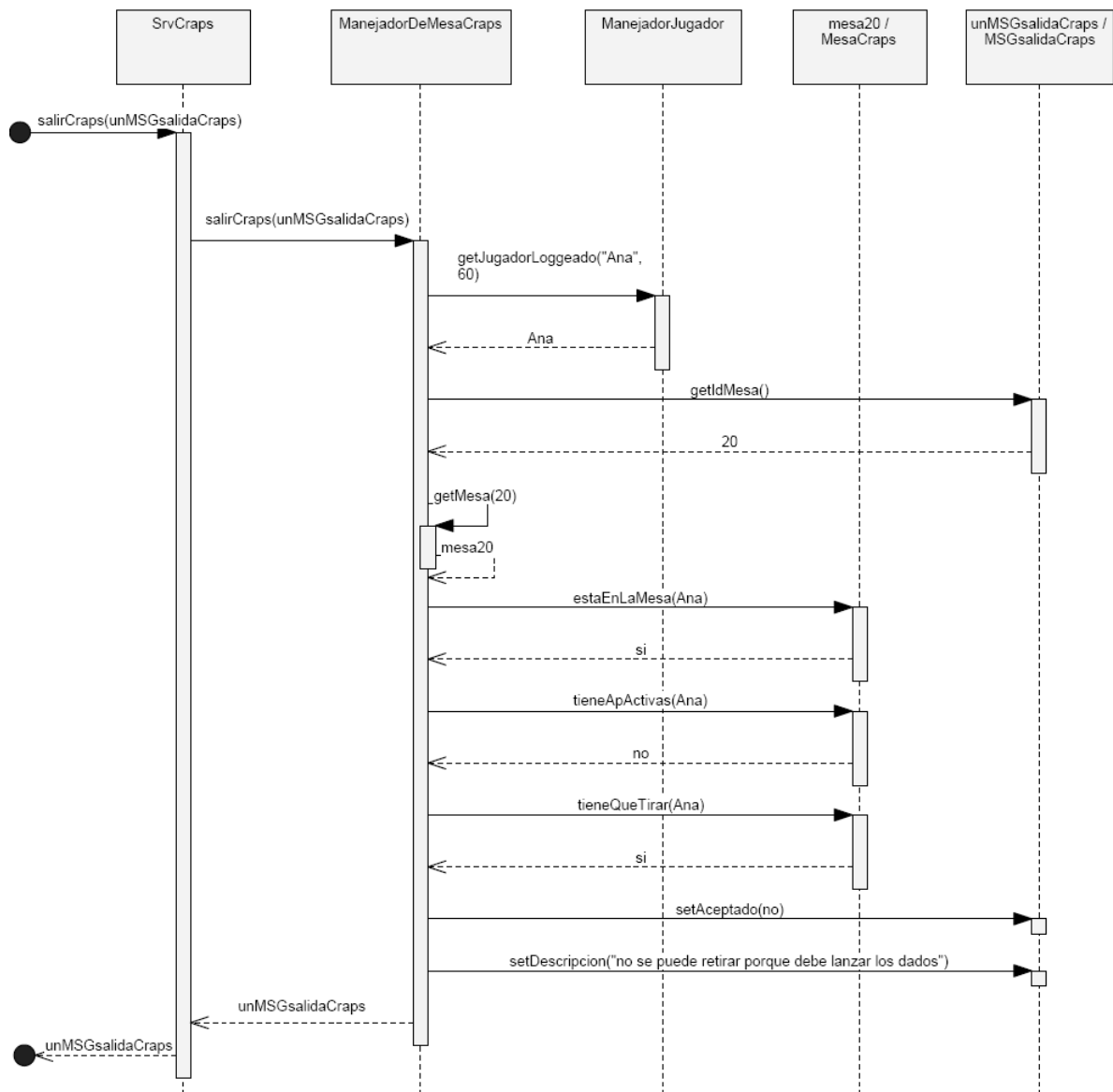
## 4.7.4. Salir en Craps

## Salir en Craps Aceptado





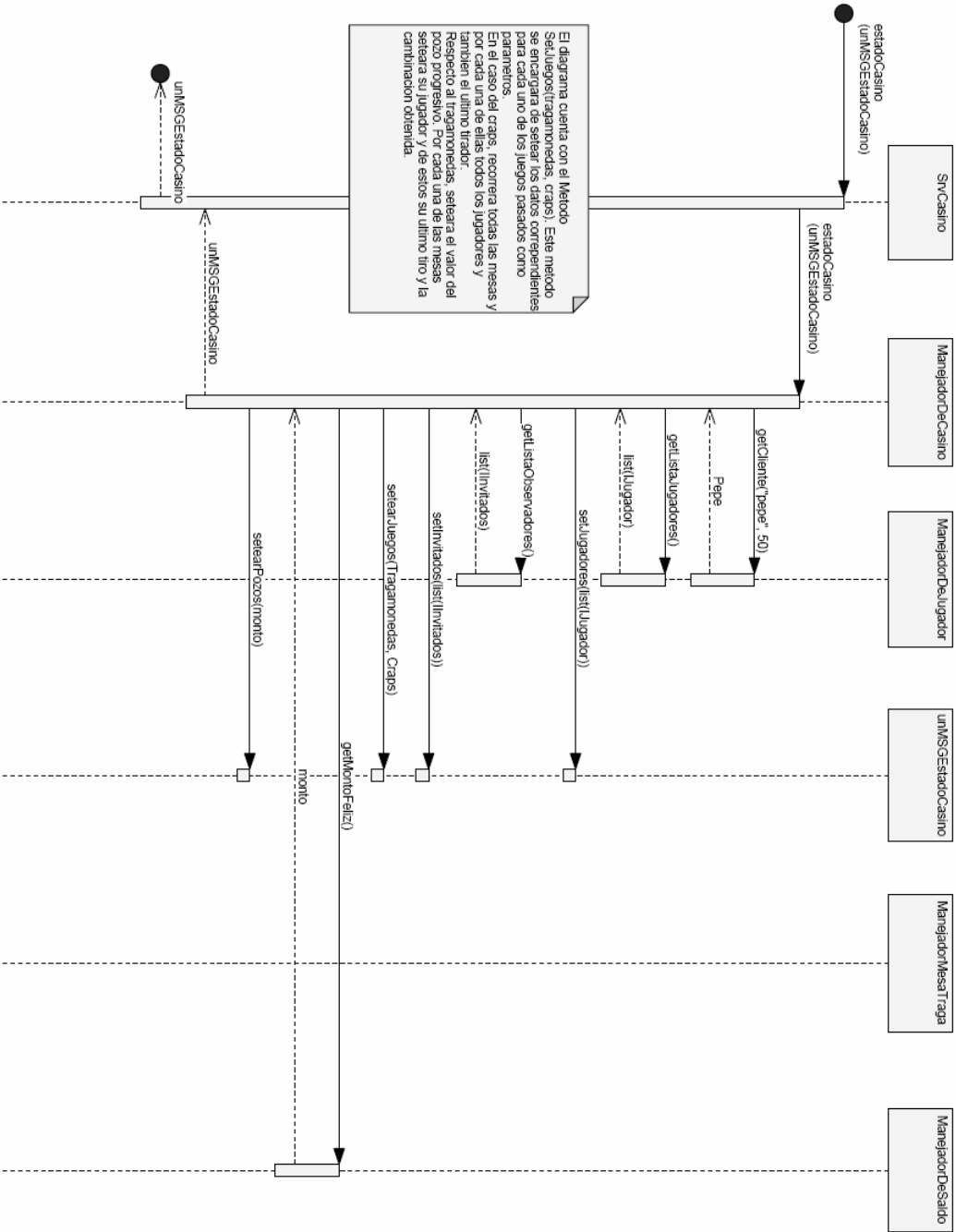
## Salir en Craps Denegado



4.8. Notificar Estado Casino

Tanto el invitado como un jugador pueden querer observar el estado del casino, preguntandoselo al mismo. El diagrama que sigue describe como el casino obtiene los datos para luego notificarselos a quien lo solicite

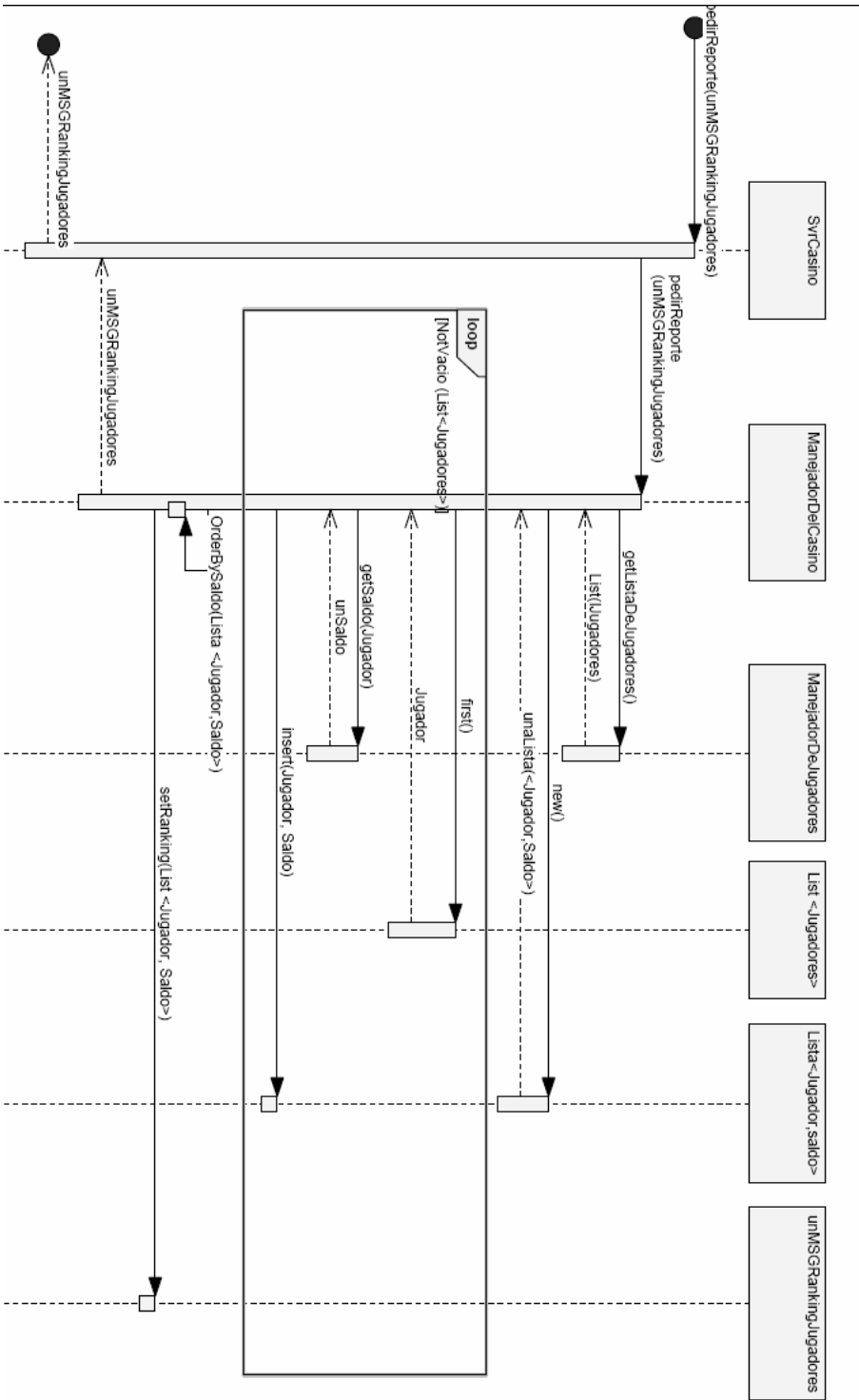
Notificar Estado Casino



4.9. Generacion de reportes

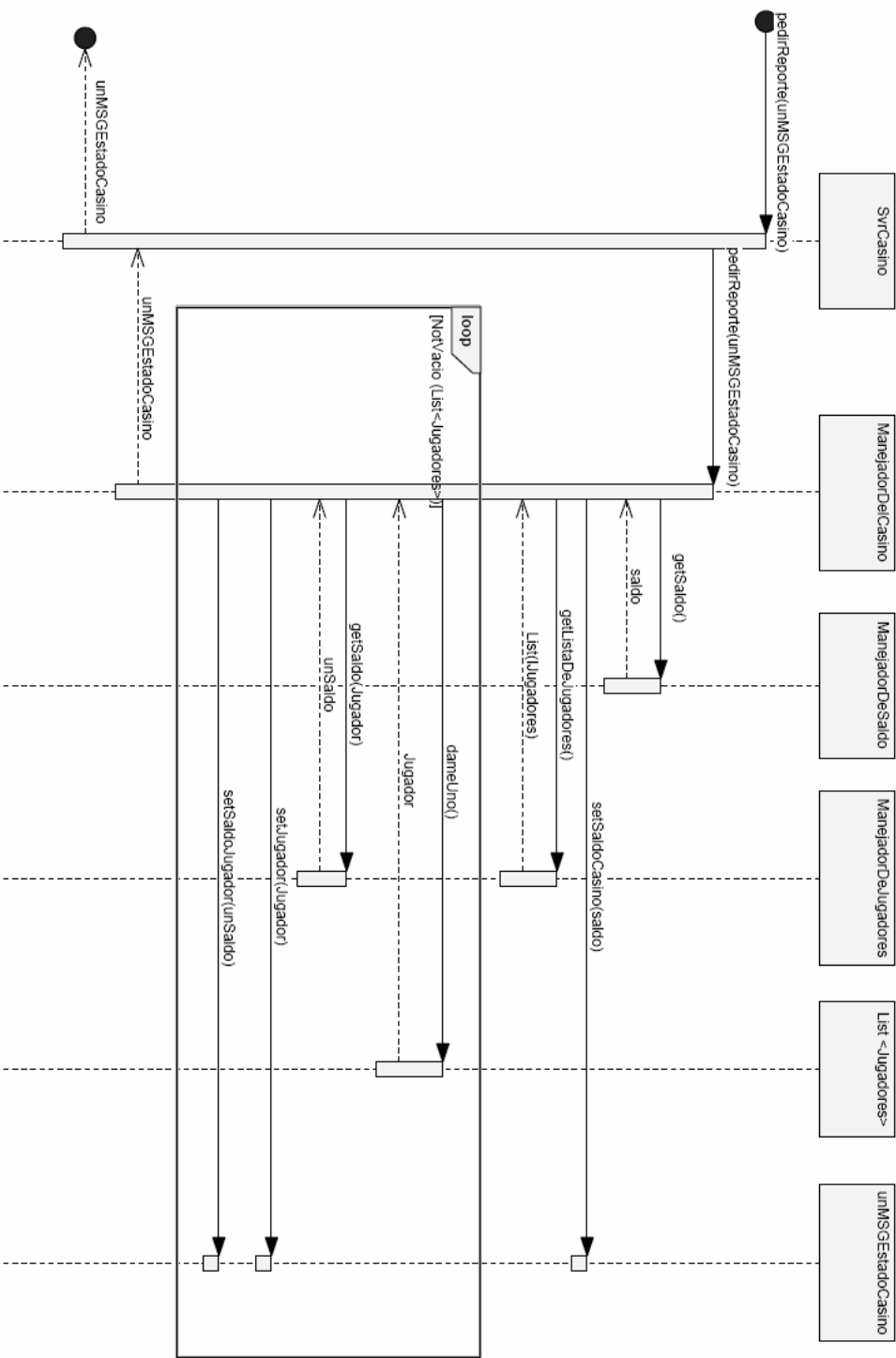
4.9.1. Generacion de reporte: Ranking de Jugadores

Ranking de Jugadores



4.9.2. Generacion de reporte: Estado Actual

Estado Actual



## 5. Anexo I - Conclusiones

Para la conclusión de este segundo informe hemos decidido hacer un breve resumen de la metodología que hemos utilizado para desarrollarlo, para demostrar que este es un trabajo no de avance lineal sino, un espiral donde constantemente se está consultando los documentos base, es decir, toda la especificación presentada en el Informe 1, y retocando el trabajo actual.

En primer lugar decidimos consultar nuestro modelo conceptual. Aunque nos sirvió para refrescar las ideas, no dimos cuenta que no aplicaba mucho al diseño pero sí para darnos una idea general. Empezamos a investigar un poco mas los principios de diseño y encontramos algunos que nos fueron de gran utilidad.

Empezamos a usar interfaces en las partes que vimos convenientes, como en Casino, Jugador e Invitado, así en un futuro podrían ampliar sus funciones, como muestra el Dependency Inversion Principle. Y en algunos casos usamos interfaces de interfaces para poder controlar otras dependencias (Interface Segregation Principle), como es el caso de los seleccionadores de resultados del tragamonedas y del craps, ya que en un futuro pueden agregarse otras formas de seleccionar los resultados aparte del modo normal (por azar) y el dirigido.

Al momento de realizar el diagrama de secuencias volvimos a consultar el informe 1, especialmente la parte de los Casos de Uso y las FSM. De estos diagramas de secuencias surgieron la mayor parte de los métodos.

Luego apareció el protocolo y sus actualizaciones, por lo que vimos necesario modificar y retocar en varias oportunidades nuestro diagrama de clases y por lo tanto también el de secuencias.

También estudiamos los distintos patrones. Un patrón que nos fue de utilidad fue el Singleton, para los Manejadores especialmente, ya que así podrían comunicarse entre sí. Otro tipo de patrones (como el Facade para el seleccionador de resultado o de tipo de jugada) sencillamente aparecieron, no fue necesario forzar el diseño.

En fin, el diseño requirió de pensar a futuro, en el 'cómo lo vamos a implementar y los cambios que pudieran surgir, aprovechando las herramientas y principios provistos para lograrlo.