

# 1 Introducción

## 1.1 Objetivo del documento

En el siguiente documento se tratarán de expresar los detalles de la estructura conceptual necesarias para el correcto funcionamiento del casino online, que fue solicitado por el grupo *Los Timbalistas*.

La orientación del texto es tal que beneficie al comprendimiento de las prioridades, mas sin perder de vista el detalle de lo que se busca del producto final. Así mismo, la presente documentación, pretende alcanzar la formalidad necesaria para poder ser utilizado como referencia, para la posterior etapa de diseño del programa.

Finalmente, cabe destacar que el lenguaje y nivel de abstracción utilizados, apuntan a lograr un compromiso entre la fácil legibilidad del documento, y una especificación clara de las características requeridas del software.

## 1.2 Destinatarios del documento

El presente documento está dirigido a los integrantes directivos del grupo *Los Timbalistas*, así como también, de haberlos, a los responsables de supervisión del proyecto y finalmente al personal del área de *Calidad de Software* asignados al seguimiento del mismo. Aún así, también puede ser de utilidad para los asesores del proyectos, y aquellos que vayan a tener contacto con el mismo; como sea la secretaria encargada de atender a los clientes, el administrador del sistema y los miembros del *Departamento de Marketing*.

## 1.3 Descripción del problema

Un grupo denominado *Los Timbalistas* decide realizar un emprendimiento con el único fin de juntar dinero para poder ir a Las Vegas. Para lograrlo comenzarán un proyecto para la puesta en funcionamiento de un casino on-line que en principio contendrá dos juegos: Tragamonedas y Craps.

## 1.4 Documentos relacionados

- Minuta Reunión Relevamiento [17-04-2008] - v1.6
- Reglas Craps [Documentos Relacionados]
- Reglas Tragamonedas [Documentos Relacionados]

## 1.5 Organización del informe

El siguiente informe cuenta con una introducción donde se explica el objetivo del este documento y sus destinatarios. Se describe brevemente el problema por el cual se está llevando a cabo el proyecto. Se describen en forma general los requerimientos y funciones del producto, sus restricciones, las suposiciones y dependencias del mismo, además de características de los usuarios a los cuales estará orientado dicho proyecto.

La tercera sección del informe contiene los requerimientos específicos, los objetivos del producto y cuenta con diagramas y explicaciones para todos los servicios que proveerá el sistema en forma general y también para cada juego en particular. Se muestran los estado posibles para cada juego en el sistema y se define cómo se comportará cada uno según los parámetros ingresados al inicializarse casino. Se muestran, además, algunos diagramas comunes a ambos juegos y otros correspondientes al funcionamiento general del casino.

El modelo de objetivos mostrará en forma amplia los requerimientos del proyecto y será la base sobre la cual se afirmarán otros modelos más específicos.

Cada sección del informe intentará mostrar de manera clara que todo lo que se desprende de los deseos del grupo "Los Timbalistas" está contemplado hasta el mínimo detalle y, en particular, en la sección 3 se intentó detallar cada punto de cada requerimiento necesario para el correcto funcionamiento del casino. Ésto se reflejó con algún modelo que, según nuestro criterio, sería la herramienta más correcta para modelar dicho aspecto. Finalmente, cada uno de los modelos se relacionan y se funden en uno solo a partir de breves explicaciones en lenguaje natural que intentan no confundir al lector, sino hacer de los distintos modelos, un único modelo consistente con los requerimientos del proyecto.

## 2 Descripción General

### 2.1 Perspectiva del producto

Este producto es independiente y completo en si mismo, puede funcionar completamente en computadoras capaces de ejecutar código Java o C#, sin necesidad de otros productos. Sin embargo se podrá posteriormente agregar módulos extra (nuevos juegos), que si dependerán de él para funcionar, en forma sencilla.

### 2.2 Funciones principales del producto

Este producto proveerá todo lo necesario para montar un casino virtual para múltiples usuarios. Presenta diversas funciones para sus distintos tipos de usuarios:

1. Administradores
  - i. Configuración de diversas características
    - las probabilidades de ganancia de los tragamonedas
    - los valores de ficha válidos para el "día" corriente
    - valor mínimo para el pozo progresivo
  - ii. Impresión de listados
    - Ranking de jugadores: jugadores que más dinero ganaron o perdieron en el "día"
    - Estado actual: informe del estado (saldo) del casino y los clientes
    - Detalle movimientos por jugador: apuestas, premios ganados, monto ganado, históricamente
  - iii. Modo dirigido
    - Ingresar o salir de modo dirigido
    - Controlar resultados de juego seleccionado
    - Controlar "jugada feliz" y "jugada todos ponen"
  - iv. Cierre del casino: el software solo lo permitirá cuando no haya mesas abiertas
  - v. Apertura del casino: sólo cuando el casino este cerrado
2. Clientes
  - i. Creación de mesas de juego
  - ii. Unirse a mesa de juego ya existente

- iii. Jugar y apostar a juego disponible (craps y tragamonedas) visualizando montos de pozos especiales
  - iv. Dejar mesa de juego
- 3. Invitados
  - i. Unirse a mesa de juego ya existente
  - ii. Ver desarrollo de juego disponible
  - iii. Dejar mesa de juego
- 4. Secretaria o empleada administrativa
  - i. Ingreso de adiciones y subtracciones al pozo principal
- 5. Departamento de Marketing
  - i. Altas y modificación de datos y saldo de clientes
- 6. Automáticas
  - i. Administrar saldo y llevar registro de ganancias y pérdidas de cada cliente
  - ii. Selección automática de jugada feliz y jugada todos ponen
  - iii. Registro de eventos relevantes
  - iv. Administración de ganancias y perdidas del casino
  - v. Cierre de mesas sin jugadores

## 2.3 Características de los usuarios

El producto esta orientado para el usuario promedio de programas de computadora, personas con conocimientos básicos de informática que les permitan abrir o cerrar un programa y sepan usar una interfaz gráfica básica.

Se asume que los jugadores conocen las reglas de los juegos así como la dinámica de los pozos y características especiales del casino.

Se asume que los administradores, departamento de marketing y empleada administrativa tendrán conocimientos normales de informática que les permita abrir el programa y usar la interfaz de configuración, altas y modificación de los datos de los clientes, o impresión de estados, así como las diversas opciones del sistema. También se asume que los administradores, secretaria y dpto. de marketing sabrán cuales son las opciones que les brindará el software y cómo trabajar con ellas.

## 2.4 Restricciones

Los usuarios disponen de computadoras "normales" (entendiendose por normal, una computadora que cumpla los requerimientos mínimos de los programas popularmente usados en sus versiones actuales, como pueden ser Windows Vista, etc). El programa deberá poder ejecutarse en un ordenador normal.

La aplicación deberá corre en red, al diseñarse la intercomunicación entre los clientes y el servidor deberá tenerse esto en cuenta.

El casino puede tener un gran numero de usuarios conectados, el sistema debe permitir esto. El sistema debe poder aceptar múltiples conexiones desde una misma terminal, sin permitir el ingreso múltiple por parte del mismo usuario.

El Departamento de Quality Assurance ha decidido que el sistema debe realizar en Java o C# así que se deberá usar uno de estos dos lenguajes para el desarrollo del software.

El sistema está orientado al usuario promedio, las interfases no deberán ser demasiado complicadas.

## 2.5 Supuestos y dependencias

El saldo inicial, depósitos y retiros son manejados por la secretaria por afuera del sistema.

El sistema no trabaja con tarjetas de crédito o débito, las maneja la secretaria por fuera del sistema.

La lista de clientes con sus saldos iniciales y todo ingreso / egreso de clientes será proporcionado en algún formato por el Departamento de Marketing y se cargará al abrir el casino.

El valor de las fichas válidas, los mínimos de los pozos y las tablas para los premios de los juegos son proporcionados al abrir el casino.

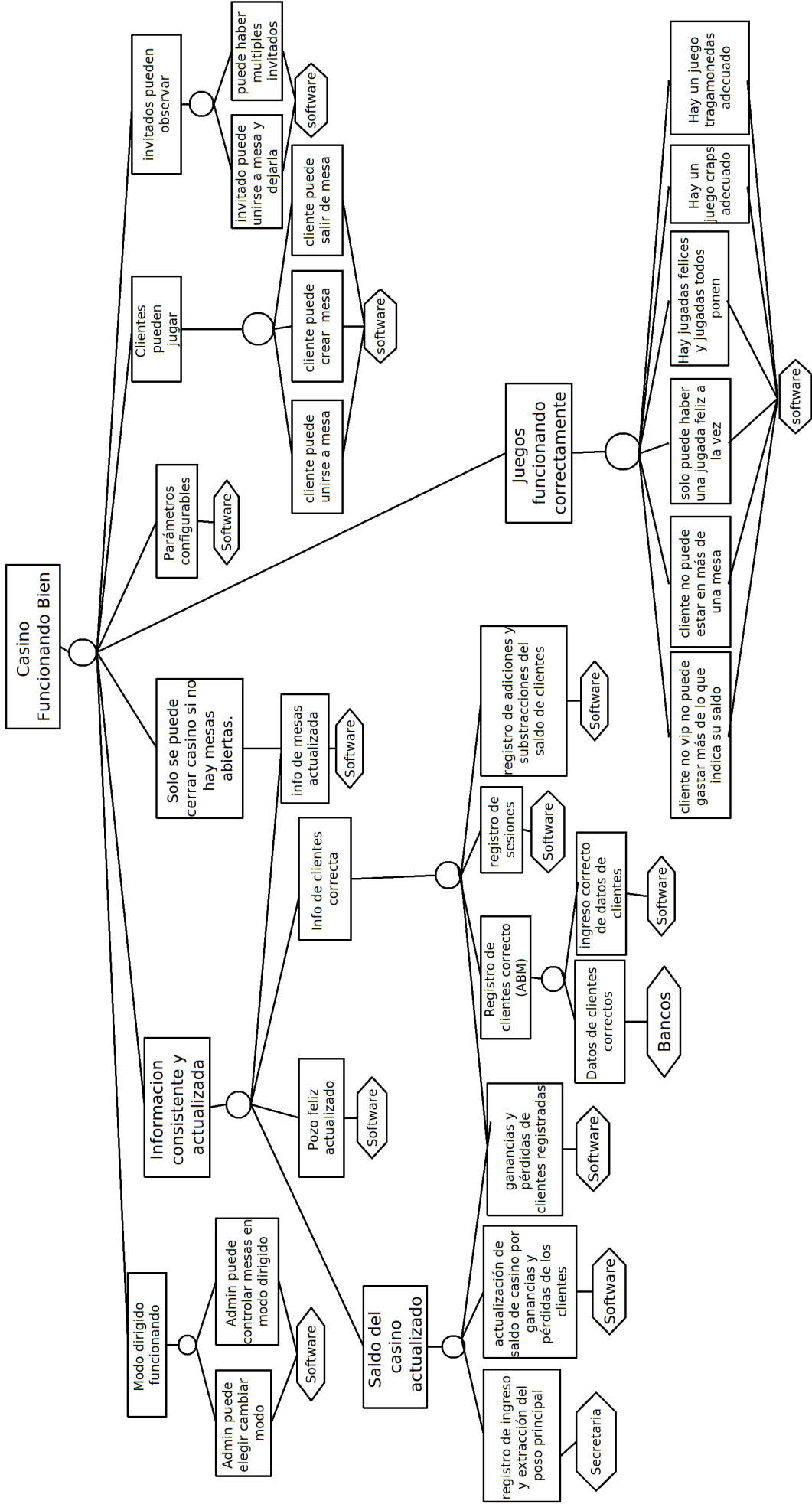
Las computadoras de los clientes son computadoras personales estandard, que pueden ejecutar programas Java o C#.

Suponemos, en esta versión, de lo siguiente:

- La secretaria, Dpto. de Marketing, Administrador, etc. estarán disponible para efectuar sus tareas.
- La red y computadoras de los usuarios siempre funcionan bien.
- Nuestras redes y computadoras usadas para los servidores también funcionan siempre correctamente, el disco rígido no fallará al intentar grabar el log de jugadas desde que se abre el casino. En otras palabras: nada relativo a las computadoras o conexiones de red, que no sea específicamente algo de nuestro sistema, fallará. Ya sea de hardware o software no programado por nosotros (como Windows o el sistema operativo elegido, drivers para el hardware del sistema, etc).

## 3 Requerimientos específicos

Como diagrama principal, y vinculante central entre los distintos diagramas de la especificación utilizaremos el siguiente diagrama de objetivos, en el que se muestran que es lo que se pretende lograr con el programa y como los distintos requerimientos se vinculan:



Del presente gráfico se desglosan los siguientes requerimientos que deberá cumplir el producto final:

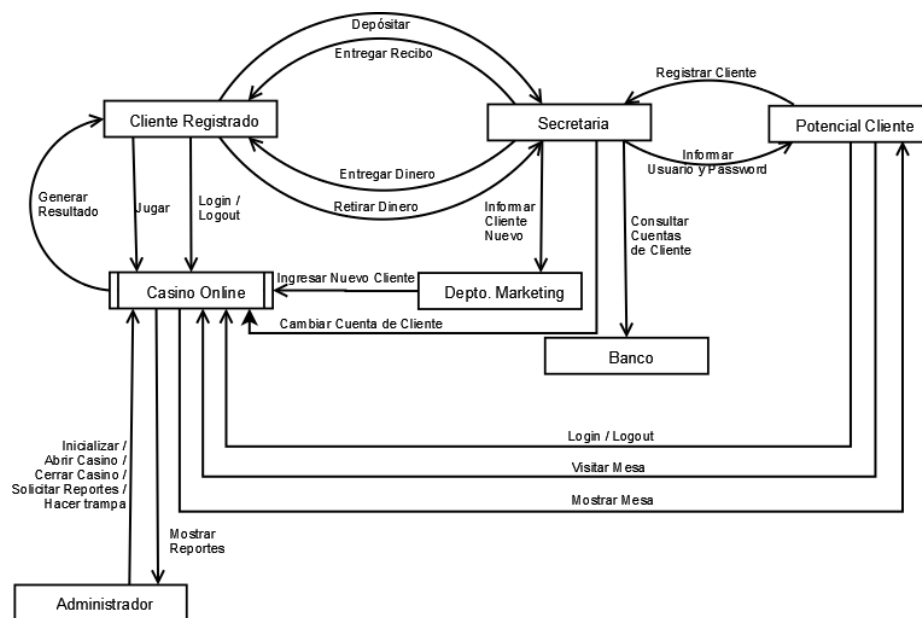
- R01.** El administrador puede cambiar de modo: esto significa que el administrador puede determinar cuando en el casino se activa el modo dirigido. *Requerimiento funcional importante*
- R02.** El administrador puede controlar las mesas en modo dirigido: significa que el administrador es capaz de controlar el resultado de los eventos aleatorios de las mesas de juego, cuando el casino este en modo dirigido. *Requerimiento funcional importante*
- R03.** Actualización de saldo del casino por ganancias y pérdidas de los clientes: este requerimientos se refiere a la necesidad de mantener el saldo del casino como resultado de la resolución de las distintas apuestas realizadas en las mesas de juego. *Requerimiento funcional esencial*
- R04.** Ganancias y pérdidas de clientes registradas: se deberá llevar cuenta de las alteraciones en el estado de cuenta de los clientes causadas por el desarrollo de los juegos en los que estos participen. *Requerimiento funcional esencial*
- R05.** Pozo feliz actualizado: esto significa que habrá que mantener el monto del pozo de jugadas felices, el cual se verá alterado tanto en las jugadas felices como en las jugadas todosponen. *Requerimiento funcional esencial*
- R06.** Registro de sesiones: se llevará un registro de las actividades de las actividades de cada cliente dentro del casino. *Requerimiento funcional importante*
- R07.** Información de mesas actualizada: el software deberá llevar cuenta de las mesas existentes durante su ejecución. *Requerimiento funcional esencial*
- R08.** Cliente no-VIP no puede gastar más de lo que indica su saldo: no debe permitirse que los clientes hagan apuestas mayores al crédito que tengan disponible. *Requerimiento funcional esencial*
- R09.** Cliente no puede estar en más de una mesa: no se debe permitir que los clientes que se encuentren en una mesa de juego, entren a otra sin antes salir de la que se encuentran. *Requerimiento funcional importante*
- R10.** Solo puede haber una jugada feliz a la vez: el software no deberá generar una jugada feliz en una mesa, si existe otra mesa que se encuentre en jugada feliz. *Requerimiento funcional esencial*
- R11.** Hay jugadas felices y jugadas todos ponen: el software deberá generar jugadas felices y jugadas todosponen en forma aleatoria. *Requerimiento funcional esencial*
- R12.** Hay un juego craps adecuado: se deberá implementar un juego de Craps que funcione adecuadamente con el resto del sistema. *Requerimiento funcional importante*
- R13.** Hay un juego tragamonedas adecuado: deberá implementarse un juego de Tragamo-

nedas que funcione adecuadamente con el resto del sistema. *Requerimiento funcional importante*

- R14.** Cliente puede unirse a mesa: aquellos clientes que no se encuentren en ninguna mesa de juego, deben poder unirse a mesas ya existentes que permitan un usuario mas. *Requerimiento funcional esencial*
- R15.** Cliente puede crear mesa: un cliente que no este en ninguna mesa, podrá crear una mesa nueva del juego que desee, con las características que desee en caso de ser permitido por el juego elegido. *Requerimiento funcional esencial*
- R16.** Cliente puede salir de mesa: los clientes que se encuentren en una mesa de juego, deben poder salir de ésta. *Requerimiento funcional esencial*
- R17.** Invitado puede unirse a mesa y dejarla: aquellas personas sin una cuenta en el sistema, que ingresen como invitados deben poseer la misma capacidad para entrar y salir de mesas que los clientes. *Requerimiento funcional deseable*
- R18.** Pueden haber múltiples invitados: el software debe permitir que ingresen una cantidad indefinida de invitados. *Requerimiento funcional deseable*
- R19.** Se puede abrir y cerrar el casino: el software debe permitir la apertura de casino y el cierre del mismo. *Requerimiento funcional importante*
- R20.** Se pueden cargar datos de los cliente: debe existir en el software alguna forma de ingresar los datos de los clientes que deseen registrarse en el sistema. *Requerimiento funcional esencial*
- R21.** Cerrar mesas sin clientes: se deben cerrar aquellas mesas que queden sin clientes. *Requerimiento funcional deseable*
- R22.** Extensibilidad del software en cantidad de juegos: el software debe ser logrado de forma que permita la adición de más juegos, sin que esto represente una alteración importante en el resto del sistema. *Requerimiento no funcional importante*
- R23.** Cliente VIP puede gastar más de lo que indica su saldo. *Requerimiento funcional importante*
- R24.** Se pueden obtener reportes: el software permite la generación y visualización de reportes que indiquen el ranking de jugadores, el estado del casino, y el detalle de los movimientos de los jugadores. *Requerimiento funcional importante*
- R25.** Los clientes pueden conectarse al sistema: debe permitirse a los clientes ingresar al sistema para jugar. *Requerimiento funcional esencial*
- R26.** En craps, proximo tirador según orden de ingreso: el sistema debe seleccionar el próximo tirador basándose en el orden en el que los clientes entraron a la mesa. *Requerimiento funcional deseable*

Cabe destacar que en los montos mencionados a lo largo de ésta especificación (asi como los valores de las apuestas), se utiliza como unidad la ficha; es decir, todos los montos se consideran en cantidad de fichas.

Secundariamente, utilizaremos como guía de los principales eventos generados en el entorno del sistema, el siguiente diagrama de contexto. En él se pueden ver los principales agentes que se relacionan con el software *Casino Online*, y entre ellos para lograr el funcionamiento esperado del sistema en su completitud.



**Figura 2.** Diagrama de Contexto

Los requerimientos "El administrador puede cambiar de modo", "El administrador puede controlar las mesas en modo dirigido", "Cliente puede crear mesa", "Cliente puede salir de mesa", "Invitado puede unirse a mesa y dejarla" y algunos otros detalles pertinentes a la administración de la información del programa se detallan en el siguiente diagrama de casos de uso:



**Figura 3.** Casos de Uso generales



<b>CASO DE USO: Creando mesa «req. R15»</b>
<p>PRE: <i>Cliente registrado</i> no está en una mesa.</p> <p>ACTOR PRIMARIO: <i>Cliente registrado</i>.</p> <ol style="list-style-type: none"> <li>1) El <i>Cliente registrado</i> elige el juego para la mesa a crear y las opciones requeridas por el mismo.</li> <li>2) Se hace entrar al <i>Cliente registrado</i> en la mesa creada. USA: <b>Uniéndose a mesa</b>.</li> </ol> <p>FIN DEL CASO DE USO.</p> <p>POS: Se creó una mesa del juego elegido, con un único <i>Cliente</i> en ella (el que la creó).</p>

<b>CASO DE USO: Uniéndose a mesa «req. R14»</b>
<p>PRE: <i>Cliente</i> no está en una mesa.</p> <p>PRE: La mesa elegida existe.</p> <p>ACTOR PRIMARIO: <i>Cliente</i>.</p> <ol style="list-style-type: none"> <li>1) Se actualiza el estado de la mesa elegida de forma correspondiente para tenerse en cuenta al nuevo integrante.</li> <li>2) Se registra el ingreso del jugador a la mesa.</li> <li>3) Se informa al <i>Cliente</i> del éxito del ingreso.</li> </ol> <p>FIN DEL CASO DE USO.</p> <p>POS: El <i>Cliente</i> se encuentra en la mesa elegida.</p>

<b>CASO DE USO: Pidiendo reporte «req. R24»</b>
<p>PRE: Verdadero.</p> <p>ACTOR PRIMARIO: <i>Administrador</i>.</p> <ol style="list-style-type: none"> <li>1) El <i>Administrador</i> elige el reporte deseado entre, <b>Ranking de jugadores, Estado actual y Detalle movimientos por jugador</b>.</li> <li>2) El sistema a partir del registro generado por la actividad de los <i>Cientes registrados</i>, y el estado de cuenta de cada uno de ellos, filtrará la información de forma que se obtengan solo los datos pedidos.</li> <li>3) El sistema dará un formato adecuado al tipo de reporte, y lo mostrará al <i>Administrador</i>.</li> </ol> <p>FIN DEL CASO DE USO.</p> <p>POS: El <i>Administrador</i> posee el reporte pedido.</p>

CASO DE USO: Ingresando al casino «req. R25 y R18»	
PRE: El casino esta abierto.	
ACTOR PRIMARIO: <i>Cliente</i> .	
EJECUCIÓN NORMAL	EJECUCIÓN ALTERNATIVA
1) El <i>Cliente</i> ingresa su nombre de usuario y contraseña en caso de tratarse de un <i>Cliente registrado</i> .	
2) El <i>Cliente</i> ingresa <b>invitado</b> como nombre de usuario y ninguna contraseña si se tratase de un <i>Cliente invitado</i> .	
3) Se informa al <i>Cliente</i> del ingreso satisfactorio.	3.1) En caso de que la contraseña sea incorrecta o el <i>Cliente registrado</i> que intenta ingresar ya esta en el casino, se rechaza el ingreso y se informa al <i>Cliente</i> de ello.
FIN DEL CASO DE USO.	FIN DEL CASO DE USO.
POS: El <i>Cliente</i> esta en el casino.	

<b>CASO DE USO: Inicializando el casino «req. R19»</b>	
PRE: El casino se encuentra cerrado.	
ACTOR PRIMARIO: <i>Administrador</i> .	
EJECUCIÓN NORMAL	EJECUCIÓN ALTERNATIVA
1) El <i>Administrador</i> ingresa los parámetros del casino.	
2) Se verifica la corrección de los parámetros ingresados y se carga la lista de clientes y sus datos, y el estado de la cuenta del casino.	2.1) Se informa al <i>Administrador</i> de los datos incorrectos y se aborta la ejecución.
3) Se establece el pozo acumulativo de las tragamonedas a su valor mínimo.	FIN DE CASO DE USO.
4) Se comienza a aceptar conexiones.	
FIN DEL CASO DE USO.	
POS: El casino está preparado para recibir la conexión de <i>Clientes</i> .	

<b>CASO DE USO: Activando modo dirigido «req. R01 y R02»</b>	
PRE: Modo dirigido desactivado.	
ACTOR PRIMARIO: <i>Administrador</i> .	
EJECUCIÓN NORMAL	EJECUCIÓN ALTERNATIVA
1) El <i>Administrador</i> elige el tipo de modo dirigido que desea activar, ya sea <b>Jugada Feliz a una mesa a elección</b> , <b>Jugada Todosponen a todas las mesas</b> o <b>Forzar resultado en</b> .	
2) Si se eligió la primer opción, se generará de forma forzada, una jugada feliz en la mesa elegida.	
3) Si se eligió la segunda opción, se generará una jugada todosponen en todas las mesas del casino.	
4) Si se eligió la última opción, el próximo resultado de todas las mesas del juego elegido, resultarán en el número especificado por el <i>Administrador</i> .	
5) Se verifica la corrección de los parámetros ingresados y se establece el modo dirigido, que estará activo hasta justo luego de la resolución del primer resultado de todas las mesas afectadas.	5.1) Se informa al <i>Administrador</i> de los parámetros incorrectos y se aborta la ejecución.
FIN DEL CASO DE USO.	FIN DEL CASO DE USO.
POS: El casino está en modo dirigido.	

<b>CASO DE USO: Registrando cliente «req. R20»</b>	
PRE: Verdadero.	
ACTOR PRIMARIO: <i>Dpto. de Marketing</i> .	
EJECUCIÓN NORMAL	EJECUCIÓN ALTERNATIVA
1) El <i>Dpto. de Marketing</i> ingresa los datos del nuevo cliente.	
2) Se verifica la corrección de los datos ingresados.	2.1) Se informa al <i>Dpto. de Marketing</i> de los datos incorrectos y se aborta la ejecución.
FIN DEL CASO DE USO.	FIN DE CASO DE USO.
POS: El registro del cliente se encuentra listo para ser cargado en la próxima inicialización del casino.	

<b>CASO DE USO: Actualizando datos del casino «req. R03»</b>	
PRE: El casino se encuentra cerrado.	
ACTOR PRIMARIO: <i>Dpto. de Marketing</i> .	
EJECUCIÓN NORMAL	EJECUCIÓN ALTERNATIVA
1) El <i>Dpto. de Marketing</i> ingresa los datos de cuenta actualizados.	
2) Se verifica la corrección de los datos ingresados.	2.1) Se informa al <i>Dpto. de Marketing</i> de los datos incorrectos y se aborta la ejecución.
FIN DEL CASO DE USO.	FIN DE CASO DE USO.
POS: El casino posee la información actualizada de su estado de cuenta.	

<b>CASO DE USO: Actualizando datos de cliente «req. R03»</b>	
PRE: El casino se encuentra cerrado.	
ACTOR PRIMARIO: <i>Dpto. de Marketing</i> .	
EJECUCIÓN NORMAL	EJECUCIÓN ALTERNATIVA
1) El <i>Dpto. de Marketing</i> ingresa los datos del cliente actualizados.	
2) Se verifica la corrección de los datos ingresados.	2.1) Se informa al <i>Dpto. de Marketing</i> de los datos incorrectos y se aborta la ejecución.
FIN DEL CASO DE USO.	FIN DE CASO DE USO.
POS: El casino posee la información actualizada del cliente.	

La siguiente máquina de estados finita, detalla la restricción requerida en "Sólo puede haber una jugada feliz a la vez" (req. R10) y a su vez muestra cómo un *Administrador* puede forzar una jugada feliz en una mesa de juego determinada (req. R02).

La máquina de estados finita esta definida como la composición *Asignador de jugada feliz* || *Mesa 1* || ... || *Mesa n*, con  $n$  el numero de mesas existentes.

#### VARIABLES

jugada\_feliz: [1..n]

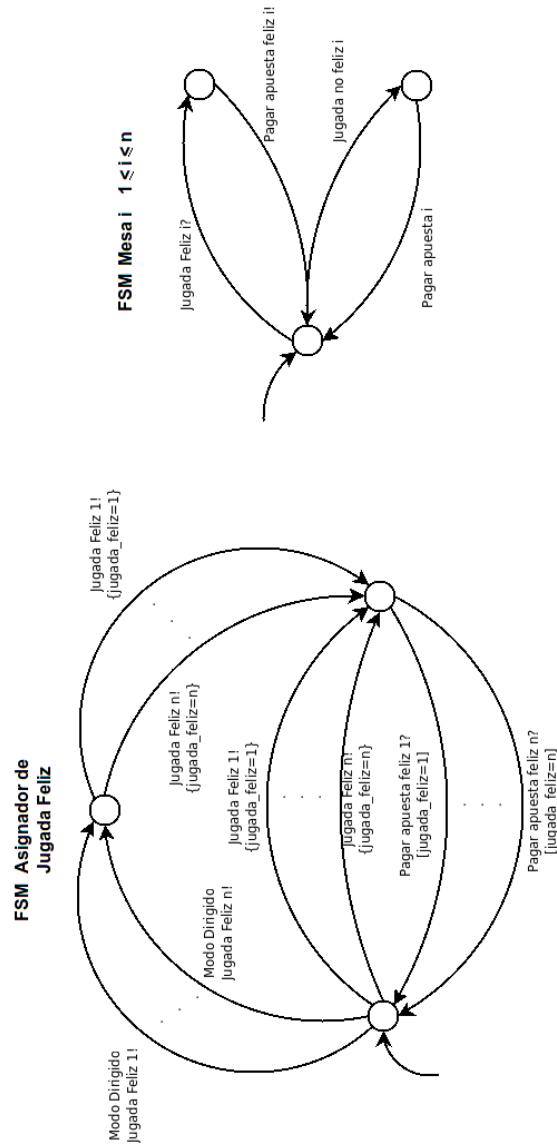


Figura 4. FSM de Jugada Feliz

El requerimiento "Que haya un juego craps adecuado" (**req. R12**) se refleja en los casos de uso "apostando", "Tirando Dados", "Determinar resultados de tirada" y "Pagar apuestas" y en el FSM Craps. Cabe destacar que en este FSM se modela el comportamiento de una sola mesa de Craps en forma independiente al resto de las mesas del casino.

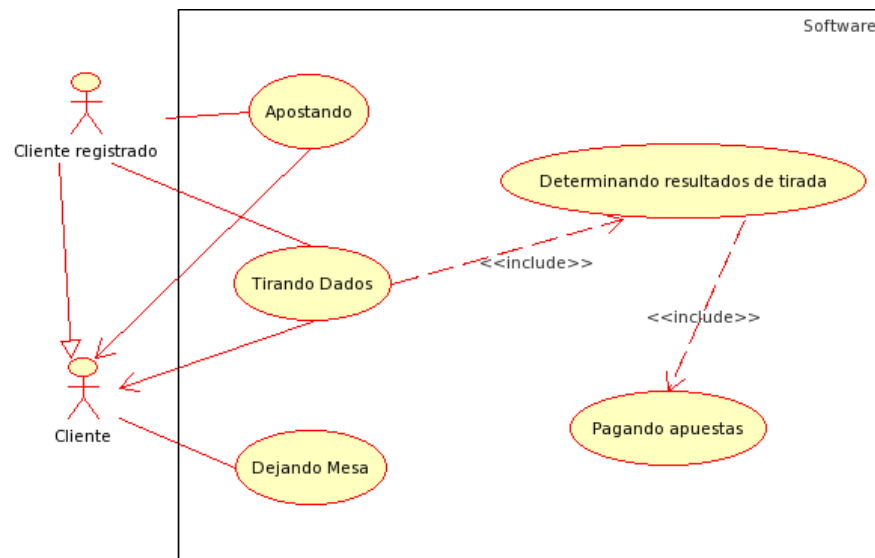
FSM Craps: Es composición de FSM Tirador || FSM Selector || FSM Jugador1 || ... || FSM JugadorN. Dónde Jugador1 a JugadorN son los Clientes registrados que estén en la mesa dada.

El actor principal asociado a este requerimiento, es el "Cliente registrado" que sería en el juego un jugador o tirador de dados.

Un jugador en una mesa de craps puede ser "tirador" de dados (solo uno a la vez por mesa) o "jugador".

El juego tendrá 2 estados: "inicio de ronda" (estados a la izquierda de la línea vertical en FSM jugador) y "no inicio de ronda" (estados a la derecha de la línea vertical en FSM jugador).

Durante el período de apuestas el "Cliente Registrado" tiene la posibilidad de apostar (C.U. "apostando"). Las apuestas válidas serán las indicadas en el C.U..



**Figura 5.** Diagrama de Casos de Uso de Craps

<b>CASO DE USO: Apostando «req. R12, R06 y R23»</b>	
PRE: <i>Cliente registrado</i> en la mesa y apostar habilitado. ACTOR PRIMARIO: <i>Cliente registrado</i> . ACTOR SECUNDARIO: <i>Cliente</i> .	
EJECUCIÓN NORMAL 1) Si el estado de la mesa es inicio de ronda, el <i>Cliente registrado</i> puede elegir apostar a <b>PassLine</b> , <b>DontPassLine</b> , <b>Field</b> o modificar la apuesta en <b>Place</b> . 2) Si el estado de la mesa no es inicio de ronda, el <i>Cliente registrado</i> puede elegir apostar a <b>Come</b> , <b>DontCome</b> , <b>Field</b> o modificar la apuesta en <b>Place</b> . 3) El <i>Cliente registrado</i> elige la suma a apostar y esta se resta de su crédito en caso de no tratarse de una apuesta en <b>Place</b> . 4) El <i>Cliente registrado</i> elige si desea incrementar o decrementar la apuesta en <b>Place</b> y en que monto.  5) Si el <i>Cliente registrado</i> lo desea puede iniciar otra apuesta (Vuelve a 1). 6) Se informa de la apuesta a los demás <i>Cientes</i> de la mesa. 7) Se registra la operación en los registros del sistema. FIN DEL CASO DE USO.	EJECUCIÓN ALTERNATIVA  3.1) Si ( <b>apuesta &gt; credito disponible</b> ) y <i>Cliente registrado</i> no es <i>Cliente VIP</i> , se rechaza la apuesta. FIN DE CASO DE USO. 4.1) Si ( <b>incremento &gt; credito disponible</b> ) y <i>Cliente registrado</i> no es <i>Cliente VIP</i> , o ( <b>decremento &lt; apuesta en Field</b> ), se rechaza la apuesta. FIN DE CASO DE USO.
POS: La/s apuesta/s especificadas se encuentran en la mesa de juego.	

El jugador que sea tirador, puede elegir tirar los dados, o esta tirada se realizará en forma automática (ver FSM). Al tirarse los dados, se cierra el período de apuestas y se inicia el C.U. "tirando dados", ocurriendo todo lo indicado en el mismo.

<b>CASO DE USO: Tirando Dados «req. R12»</b>
PRE: <i>Cliente registrado</i> en la mesa, es tirador y el tiempo para hacer apuestas terminó. ACTOR PRIMARIO: <i>Cliente registrado</i> . ACTOR SECUNDARIO: <i>Cliente</i> .
EJECUCIÓN 1) El <i>Cliente registrado</i> tira los dados. 2) Se informa a los <i>Cientes</i> de la mesa, si ésta se encuentra en una jugada regular, una jugada feliz o una jugada todosponen. 3) Si el <b>Modo dirigido</b> está activado, sale el resultado especificado por el <i>Administrador</i> . 4) Si el <b>Modo dirigido</b> no está activado, el resultado se decide aleatoriamente. 5) El sistema informa a los <i>Cientes registrados</i> en la mesa el resultado. 6) El sistema paga las apuestas debidamente. USA: DETERMINANDO RESULTADOS DE TIRADA 7) Si el <i>Cliente registrado</i> tirador gana la ronda, elige si sigue de tirador o no. 8) Si el <i>Cliente registrado</i> tirador perdió la ronda, o en el paso anterior eligió no seguir tirando, el sistema elige un nuevo tirador basándose en el orden de ingreso a la mesa, entre los <i>Cientes registrados</i> en la mesa. FIN DEL CASO DE USO.
POS: Se completó un ciclo de tirada.

El caso de uso "Determinando resultados de tirada", refleja cálculos internos del sistema y usa el C.U. "Pagando apuestas", en el cual en base a las apuestas que haya, se paga a cada jugador lo que ganó, acorde a las reglas del juego (contribuyendo al req. R12) y se notifica al jugador de esta ganancia actualizando su saldo y de los resultados de las apuestas retirando las apuestas que corresponda según las reglas del juego.

CASO DE USO: Determinando resultados de tirada «req. R12 y R06»
<p>PRE: Verdadero.</p> <p>ACTOR PRIMARIO: <i>Crupier «clase abstracta»</i></p> <p>EJECUCIÓN</p> <ol style="list-style-type: none"> <li>1) El <i>Crupier</i> calcula cuanto pagar a cada <i>Cliente registrado</i>, multiplicando el valor de sus cada una de sus apuestas ganadoras (determinando cuales son ganadoras por el resultado de los dados y el estado del juego) por lo que se indica en la tabla de premios que corresponda (según cuál sea la apuesta y el estado del juego).</li> <li>USA: PAGANDO APUESTAS</li> <li>2) El <i>Crupier</i> pone el <b>Puck</b>, donde corresponda según la ultima tirada de dados. Si el estado es inicio de ronda y sale 4,5,6,8,9 o 10. el puck se pone en el nro que salió (en ON). Si el estado no es inicio de ronda y sale 7 o el numero en el que el puck estaba, se desplaza el puck (marcado OFF) a la esquina superior izquierda de la mesa.</li> <li>3) Si es jugada feliz, se agrega a la cantidad a pagar calculada para cada <i>Cliente registrado</i>, una parte del pozo de jugada feliz proporcional a lo que se había calculado previamente como ganancia para ese cliente respecto de todas las ganancias de la mesa para esa jugada; y se setea el pozo de jugada feliz a 0.</li> <li>4) Si es jugada todosponen, se resta a lo que cada <i>Cliente registrado</i> ganó la proporción configurada y se la suma al pozo de jugada feliz.</li> <li>5) Se aumenta el crédito de cada <i>Cliente registrado</i> en un monto igual al calculado en los pasos anteriores, y se informa al <i>Cliente registrado</i> de esto.</li> <li>6) Se registran los detalles de la operación en los registros del sistema.</li> </ol> <p>FIN DEL CASO DE USO.</p> <p>POS: Se pagaron las apuestas correctamente.</p>

CASO DE USO: Pagando apuestas «req. R12»
<p>PRE: Verdadero.</p> <p>ACTOR PRIMARIO: <i>Crupier «clase abstracta»</i></p> <p>EJECUCIÓN</p> <ol style="list-style-type: none"> <li>1) Si salió en los dados 2, 3 o 12, se pagan las apuestas de DontCome y DontPassBar según la tabla de premios y se retiran las apuestas de PassBar y Come.</li> <li>2) Si se está en inicio de ronda, y salió en los dados 7 u 11, se pagan las apuestas de PassBar según la tabla de premios y la cantidad apostada, y se retiran las apuestas de apuestas de DontPassBar.</li> <li>3) Si no se está en inicio de ronda, y salió en los dados 7 u 11, se pagan las apuestas de Come según la tabla de premios y la cantidad apostada y se retiran las apuestas de apuestas de DontCome.</li> <li>4) Si salió un 7, se pagan las apuestas de PlaceBetLose y DontComePoint según la tabla de premios y la cantidad apostada, y se retiran las apuestas de PlaceBetWin y ComePoint.</li> <li>5) Según el resultado de la tirada, se pagan las apuestas del PlaceBetWin o ComePoint según la tabla de premios, y se retiran las apuestas del PlaceBetLose o DontComePoint correspondiente.</li> <li>6) Si salió en los dados 2, 3, 4, 9, 10, 11 o 12, se pagan las apuestas de FieldBet según la tabla de premios y la cantidad apostada; o si el resultado es 5, 6, 7 u 8, se retiran las apuestas de FieldBet.</li> </ol> <p>FIN DEL CASO DE USO.</p> <p>POS: Se pagaron las apuestas correctamente.</p>

El C.U. Dejando mesa describe el proceso por el cuál un jugador puede dejar una mesa y cuándo puede hacerlo.

<b>CASO DE USO: Dejando Mesa «req. R16, R21 y R26»</b>
PRE: <i>Cliente</i> en la mesa, y <i>Cliente</i> no tiene apuestas en la mesa.
ACTOR PRIMARIO: <i>Cliente</i> .
1) Si el <i>Cliente</i> no es el tirador, deja la mesa
2) Si el <i>Cliente</i> es tirador, el sistema elige el próximo tirador basándose en el orden de ingreso a la mesa, entre los <i>Clientes registrados</i> en la mesa.
3) Si no quedan jugadores en la mesa, esta se cierra.
FIN DEL CASO DE USO.
POS: El <i>Cliente</i> no se encuentra mas en la mesa.

En el FSM Craps, se muestra e indica el funcionamiento y dinámica del juego CRAPS así como la sincronización entre los jugadores y el tirador de dados. Este FSM hace referencia al req. R12 "que haya un juego Craps adecuado".

En el FSM varias transiciones tienen marcas con nros del 0 al 8, estas hacen referencia a casos de uso o líneas dentro del caso de uso "tirando dados".

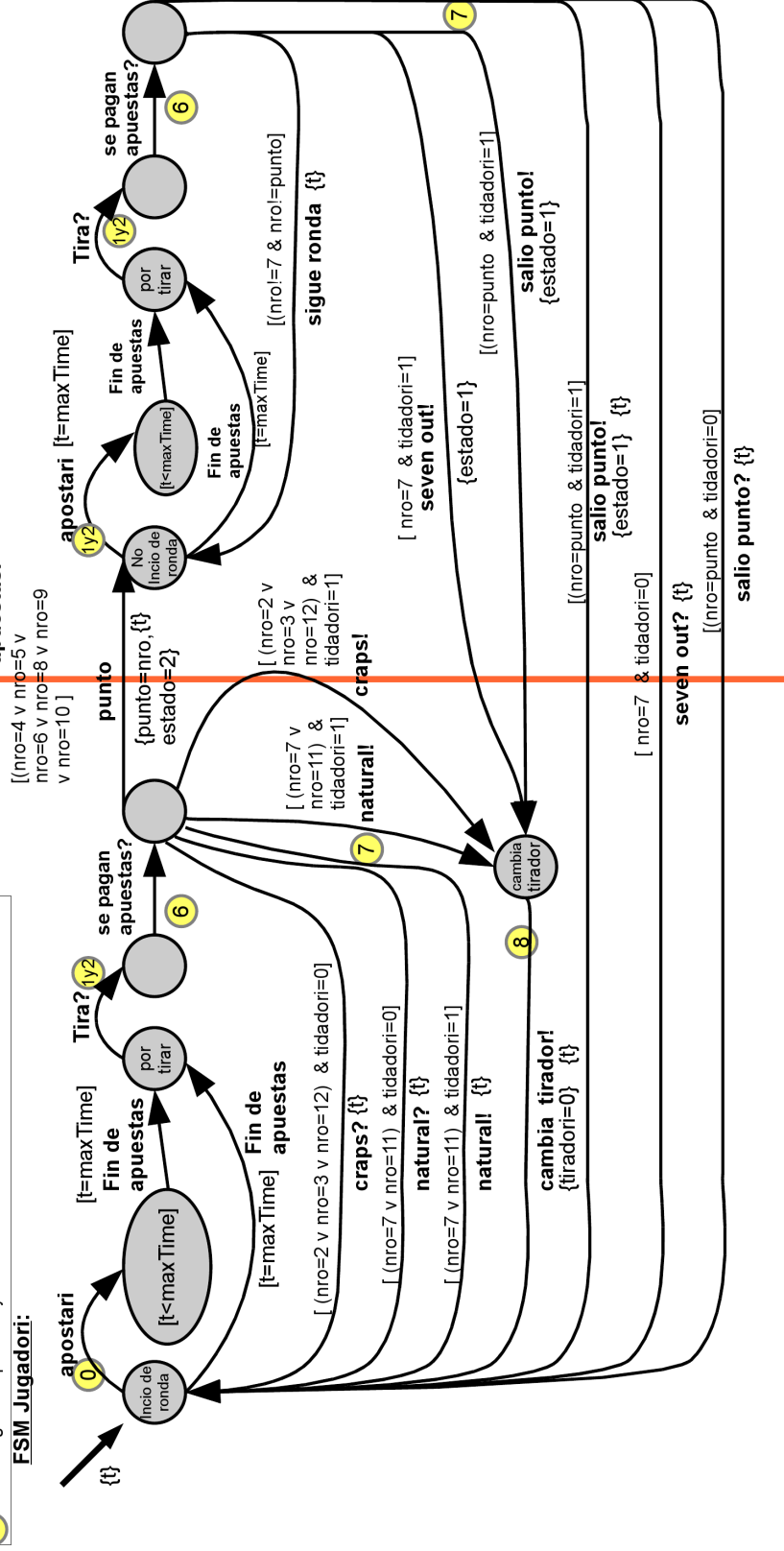
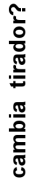
La selección de tirador la hace el sistema en el caso de uso "tirando dados" eligiendo a los jugadores en forma circular por orden de llegada a la mesa y sin elegir a jugadores que hayan dejado la mesa. Esto es representado por las transiciones Tira1 a TiraN del FSM Selector.

En el FSM Craps se puede observar como funcionan la rondas del juego, qué ocurre dados los resultados de los dados y cuando debe o puede cambiar el tirador (quien puede decidir dejar de ser el tirador en caso de haber ganado una ronda). También se puede ver que eventos pueden ocurrir en el juego dado las trazas válidas de este FSM.

De este FSM se puede observar que las apuestas solo pueden ocurrir en ciertos períodos desde que se pagaron las apuestas anteriores (o desde el inicio del juego) hasta que el tirador tira los dados (para lo cual cuenta con un tiempo determinado, tras el cual se realiza la tirada automáticamente) y que nadie puede tirar los dados desde que un tirador deja de ser tirador hasta que se elija un nuevo tirador.



- 6** CU: Pagando Apuestas y Determinando resultados de tirada



A continuación damos un ejemplo de las tablas nombradas en los casos de uso:

#### En cualquier estado de la mesa

a favor	Sitio 4	Sitio 5	Sitio SEIS	Sitio 8	Sitio NUEVE	Sitio 10	Field
2	9:5	7:5	7:6	pierde	pierde	pierde	2:1
3							1:1
4							1:1
5							pierde
6							pierde
7	pierde	pierde	pierde	pierde	pierde	pierde	pierde
8	7:6	7:5	7:6	7:6	7:5	9:5	pierde
9							1:1
10							1:1
11							1:1
12							2:1
en contra	Sitio 4	Sitio 5	Sitio SEIS	Sitio 8	Sitio NUEVE	Sitio 10	
2	pierde	pierde	pierde	4:5	5:8	5:11	
3							
4							
5							
6							
7	5:11	5:8	4:5	4:5	5:8	5:11	
8	pierde	pierde	pierde	pierde	pierde	pierde	
9							
10							
11							
12							

#### Si el estado de la mesa es inicio de ronda

apuesta	PassBar	NoPassBar	ComePointN	DontComePointN
2	pierde	1:1	2=N->1:1	2=pierde
3	pierde	1:1	3=N->1:1	3=pierde
4	1:1	pierde	4=N->1:1	4=pierde
5			5=N->1:1	5=pierde
6			6=N->1:1	6=pierde
7			pierde	1:1
8			8=N->1:1	8=pierde
9			9=N->1:1	9=pierde
10			10=N->1:1	10=pierde
11	1:1	pierde	11=N->1:1	11=pierde
12	pierde	devolución	12=N->1:1	12=pierde

#### Si el estado de la mesa no es inicio de ronda

apuesta	ComeBet	Don'tComeBet	ComePointN	DontComePointN	PointN
2	pierde	1:1	2=N->1:1	2=pierde	2=N->1:1
3	pierde	1:1	3=N->1:1	3=pierde	3=N->1:1
4	1:1	pierde	4=N->1:1	4=pierde	4=N->1:1
5			5=N->1:1	5=pierde	5=N->1:1
6			6=N->1:1	6=pierde	6=N->1:1
7			pierde	1:1	pierde
8			8=N->1:1	8=pierde	8=N->1:1
9			9=N->1:1	9=pierde	9=N->1:1
10			10=N->1:1	10=pierde	10=N->1:1
11	1:1	pierde	11=N->1:1	11=pierde	11=N->1:1
12	pierde	devolución	12=N->1:1	12=pierde	12=N->1:1

El requerimiento "Hay un juego tragamonedas adecuado" se refleja en los casos de uso "Apostando", "Apretando Botón", "Generando Resultado" y "Saliendo de Mesa" y en el FSM Juego Tragamonedas. Donde FSM Juego Tragamonedas: Es composición de FSM Tragamonedas || FSM Jugada || FSM Generador de Premio || FSM Trampa y Azar. (A continuación se detallan las variables de las 4 partes del FSM)

**Variables:**

var %jugadaFeliz: [0, 100]

var %premioProg: [0, 100]

var apuesta: [1, 3]

var credito: [-MAXCREDITO, MAXCREDITO]

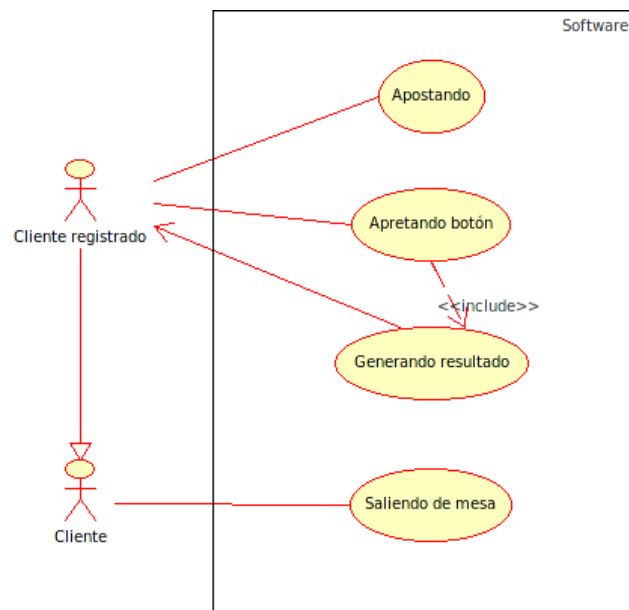
var creditoCasino: [-MAXCREDITO, MAXCREDITO]

var premioProgresivo: [MINPP, MAXPP]

var premio: [0, MAXSUPERMEGAPREMIO]

var jugadaFeliz: [MINJF, MAXJF]

var TP: [0, 1]



**Figura 7.** Casos de Uso de Tragamonedas

El actor principal asociado a este requerimiento, es el "Cliente registrado" que representa un jugador en juego y que tiene la posibilidad de apostar (Caso de Uso "Apostando") de 1 a 3 fichas de acuerdo con el valor predeterminado de la moneda.

<b>CASO DE USO: Apostando «req. R13, R06 y R23»</b>	
PRE: <i>Cliente registrado</i> está en una mesa.	
ACTOR PRIMARIO: <i>Cliente registrado</i> .	
EJECUCIÓN NORMAL	EJECUCIÓN ALTERNATIVA
1) El <i>Cliente registrado</i> efectúa una apuesta de 1 a 3 monedas.	1.1) Si el crédito no le alcanza para pagar la apuesta y el <i>Cliente registrado</i> no es vip, impedir que éste apueste esa cantidad de monedas.
2) Se resta el valor de la apuesta al crédito del cliente. Luego se suma dicho valor al saldo del casino.	
3) Se registra la operación en el registro del sistema.	
FIN DEL CASO DE USO.	
POS: El <i>Cliente registrado</i> apostó.	

Notar que, para no llenar de información a la FSM Jugada, decidimos expresar en el Caso de Uso "Apostando" el hecho de que si el cliente registrado no es Vip ni tiene crédito suficiente para jugar, no podrá hacerlo.

Luego de apostar, el Cliente registrado tiene la opción de "tocar un botón" para que el sistema calcule el resultado de la jugada (Caso de Uso "Apretando Botón").

<b>CASO DE USO: Apretando botón «req. R13 y R06»</b>
PRE: <i>Cliente registrado</i> está en una mesa; y <i>Cliente registrado</i> apostó en la mesa.
ACTOR PRIMARIO: <i>Cliente registrado</i> .
1) El cliente aprieta el botón para generar el resultado de la máquina tragamonedas.
2) La máquina genera el resultado. USA: <b>Generando resultado</b> .
FIN DEL CASO DE USO.
POS: El <i>Cliente registrado</i> apretó el botón para generar el resultado.

En el Caso de Uso "Generando Resultado" se explica con más detalle la forma en la cual se calcula el premio para una apuesta y se hace referencia a una tabla de premios posibles según los resultados obtenidos en la jugada del tragamonedas.

<b>CASO DE USO: Generando resultado «req. R13 y R06»</b>
PRE: <i>Cliente registrado</i> apretó el botón.
ACTOR PRIMARIO: <i>Tragamonedas «clase abstracta»</i> .
1) El <i>Tragamonedas</i> calcula el resultado a pagar según la tabla de premios de acuerdo con la apuesta ingresada previamente.
2) Se agrega el porcentaje sobre la apuesta correspondiente al pozo progresivo a dicho pozo. Dicho valor se descuenta del crédito del casino.
3) Si la cantidad de apuestas máximas progresivas (3 monedas) es mayor o igual al número de apuestas necesarias para ganar el premio progresivo y el premio ganado es el máximo (salen los 3 dinosaurios), se sumará el premio progresivo a dicho premio y el pozo progresivo se inicializará.
4) Si la jugada es TodosPonen se restará el porcentaje correspondiente para el pozo feliz sobre el crédito del cliente y dicho monto se agregará al pozo.
5) Si la jugada es JugadaFeliz, se sumará el pozo feliz al premio. El pozo feliz se inicializará.
6) El <i>Tragamonedas</i> sumará el premio calculado al crédito del cliente y se restará del crédito del casino.
7) Se registra la operación en el registro del sistema.
FIN DEL CASO DE USO.
POS: Se pagaron las apuestas.

Resultado			Premio x1 ficha	Premio x2 fichas	Premio x3 fichas
Slot 1	Slot 2	Slot 2			
Dinosaurio	Dinosaurio	Dinosaurio	1000	2000	5000
Cereza	Cereza	Cereza	160	320	480
BAR Triple	BAR Triple	BAR Triple	80	160	240
BAR Doble	BAR Doble	BAR Doble	40	80	120
BAR Simple	BAR Simple	BAR Simple	20	40	60
BAR Simple	BAR Doble	BAR Triple	10	20	30
Cerezas	Cerezas	Dinosaurio	5	10	15
Cereza	Dinosaurio	BAR Simple	2	4	6

Figura 8. Ejemplo de tabla de premios de Tragamonedas

El Cliente registrado podrá salir de la máquina en la que se encuentra aún si ha realizado una apuesta (Caso de Uso "Saliendo de Mesa") y luego se registrará en el sistema que el cliente se retiró de ésta.

<b>CASO DE USO: Saliendo de mesa «req. R16 y R21»</b>
PRE: <i>Cliente</i> está en una mesa. ACTOR PRIMARIO: <i>Cliente</i> . 1) El <i>Cliente</i> aprieta el botón para salir de la máquina tragamonedas. 2) Se sumará el monto apostado al crédito del cliente y se restará del saldo del casino. 3) Se cierra la mesa. FIN DEL CASO DE USO. POS: El <i>Cliente</i> no está en ninguna mesa.

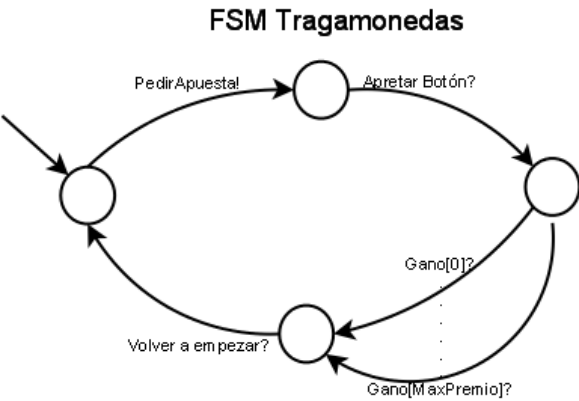


Figura 9.



También, como en el Caso de Uso "Apretando Botón" el Cliente registrado tiene la opción de "tocar un botón" para que el sistema calcule el resultado de la jugada, lo que también puede ser observado en el FSM Jugada. Además, la transición "Apretar Botón" también se encuentra en la FSM Generador de Premio que calculará el premio según el tipo de jugada actual, la apuesta realizada y las anteriores jugadas para determinar si se ha ganado o no el premio progresivo.

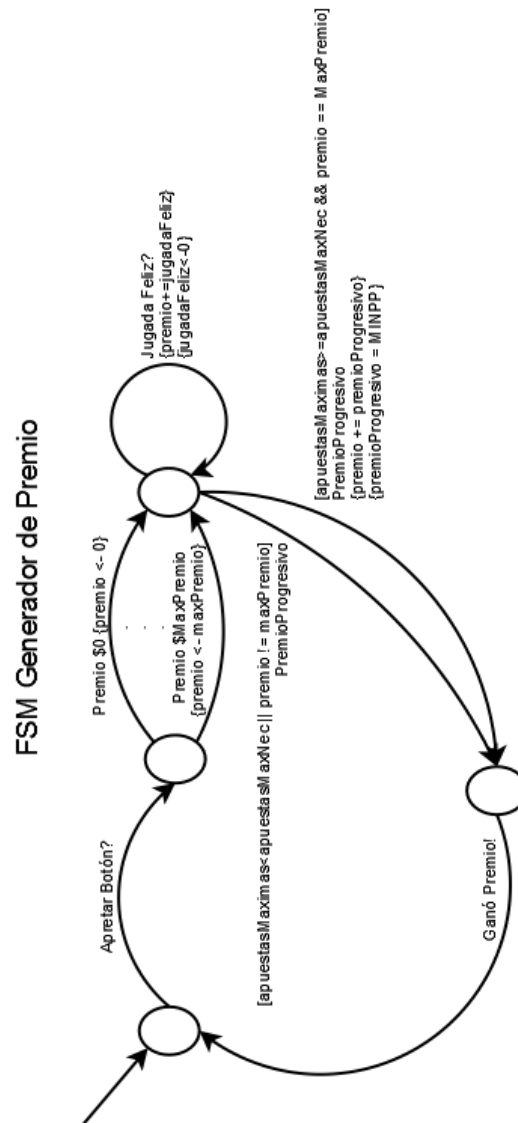


Figura 11.

Finalmente se pagarán las apuestas y se registrará la operación en el sistema.

También podremos observar que para una jugada dada se determinará el tipo de jugada según el FSM Trampa y Azar (Jugada Feliz, Todos Ponen o común).

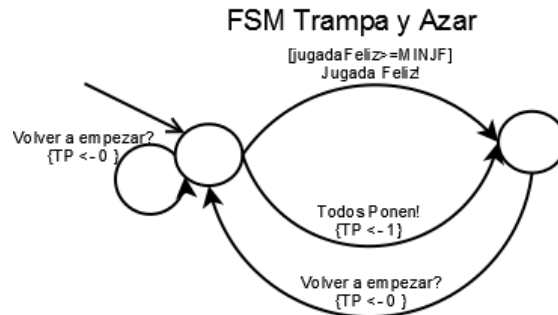


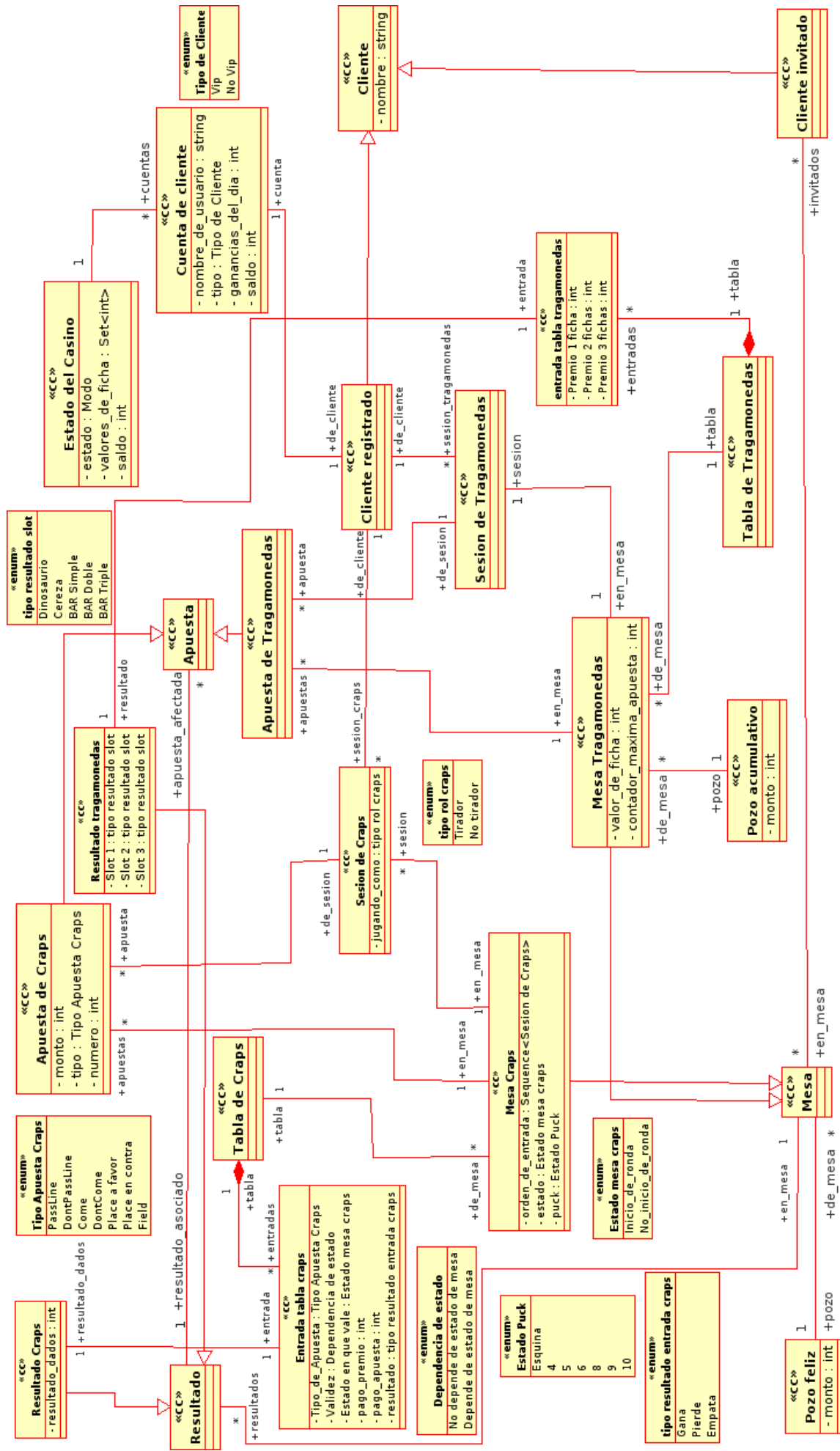
Figura 12.

Como se ha visto y veremos a lo largo del informe, se ha intentado simplificar el modelo dejando el modo de juego (dirigido o azar) así como el tipo de jugada (feliz, todos ponen o común) fuera del modelo.

Finalmente, en el siguiente diagrama de clases conceptuales, se detallan cuestiones como la relación entre las cantidades *Clientes* por mesa, restricciones sobre las apuestas en relación a la mesa de juego y su dueño. Este diagrama es de alguna forma, una visión diferente e independiente de la relación entre las entidades del software durante su ejecución. También cabe destacar que no reflejan las actividades del *Administrador* y el *Departamento de Marketing* con el software, ya que eso quedó suficientemente especificado en sus casos de uso.

Para el cumplimiento del requerimiento **R22**, se puede observar que en el caso de necesitar agregar un nuevo juego, sólo habría que agregar una clase derivada de **Mesa**, una derivada de **Apuesta**, otra derivada de **Resultado** y una nueva clase de sesión asociada con **Cliente registrado** y asociarlas entre ellas de manera analoga a las ya existentes, pero teniendo en cuenta los posibles cambios debido a la mecánica de juego.





Se definieron además algunas restricciones en OCL para el modelo anterior:

```
inv: Pozo_feliz.AllInstances()->size() = 1
inv: Pozo_acumulativo.AllInstances()->size() = 1
inv: Tabla_de_Craps.AllInstances()->size() = 1
inv: Tabla_de_Tragamonedas.AllInstances()->size() = 1
inv: Estado_del_casino.AllInstances()->size() = 1

context Cuenta_de_Cliente
inv: (self.tipo=Vip) implies (self.saldo >= 0)

context Apuesta_de_Craps
inv: self.monto > 1
inv: (self.tipo = place_a_favor or self.tipo = place_en_contra) implies
(self.numero = 4 or self.numero = 5 or self.numero = 6 or self.numero = 8 or
self.numero = 9 or self.numero = 10)
inv: self.de_sesion.en_mesa = self.en_mesa
inv: self.resultado_asociado.en_mesa = self.en_mesa

context Apuesta_de_Tragamonedas
inv: self.de_sesion.en_mesa = self.en_mesa
inv: self.resultado_asociado.en_mesa = self.en_mesa

context Pozo_Acumulativo
inv: self.monto >= 0

context Pozo_feliz
inv: self.monto >= 0
```

```

context Mesa_Craps
inv: self.sesion->IncludesAll(this.orden_de_entrada)
inv: self.sesion->select(S|S.Jugando_como=Tirador)->size() = 1
inv: self.estado=inicio_de_ronda implies self.puck=Esquina
inv: self.estado=No_inicio_de_ronda implies self.puck!=Esquina

context Resultado_Craps
inv: self.apuesta_afectada.OCLisKindOf(Apuesta_de_Craps)

context Resultado_Tragamonedas
inv: self.apuesta_afectada.OCLisKindOf(Apuesta_de_Tragamonedas)

context Tabla_de_craps
inv: self.entradas.forAll(e f | (e.Tipo_de_Apuesta = f.Tipo_de_Apuesta and
e.resultado_dados.resultado_dados = f.resultado_dados.resultado_dados and
e.Validez = No_depende_de_estado_de_mesa implies e = f )

inv: self.entradas.forAll(e f | e.Tipo_de_Apuesta = f.Tipo_de_Apuesta and
e.resultado_dados.resultado_dados = f.resultado_dados.resultado_dados and
e.Validez = f.Validez and (e.Validez = Depende_de_estado_de_mesa and
e.estado_en_que_vale = f.estado_en_que_vale)) implies e = f )

context Tabla_de_tragamonedas
inv: self.entradas.forAll(e f | (e.resultado.Slot_1 = f.resultado.Slot_1 and
e.resultado.Slot_2 = f.resultado.Slot_2 and e.resultado.Slot_3 =
f.resultado.Slot_3 implies e = f )

```

## 4 Glosario

**Cliente registrado.** cliente con una cuenta en el sistema, se lo puede ver como *jugador* con capacidad de jugar.

**CU.** Caso/s de Uso.

**FSM.** Máquina de Estados Finita.

**Inicio de ronda.** en Craps, el período entre el comienzo de una ronda y la primer tirada de la misma.

**Log.** sinónimo de registro.

**No inicio de ronda.** en Craps, el período restante de la ronda, luego del inicio de ronda.

## Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Objetivo del documento	1
1.2	Destinatarios del documento	1
1.3	Descripción del problema	1
1.4	Documentos relacionados	1
1.5	Organización del informe	1
<b>2</b>	<b>Descripción General</b>	<b>2</b>
2.1	Perspectiva del producto	2
2.2	Funciones principales del producto	2
2.3	Características de los usuarios	3
2.4	Restricciones	3
2.5	Supuestos y dependencias	4
<b>3</b>	<b>Requerimientos específicos</b>	<b>4</b>
<b>4</b>	<b>Glosario</b>	<b>?</b>
<b>5</b>	<b>Anexo I - Conclusiones del TP</b>	<b>?</b>