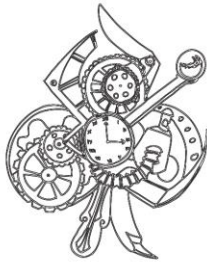


SISTEMA DE ADMINISTRACIÓN DE CASINOS

- Especificación de Requisitos del Software -



Pablo Esteban Deltell

Miguel Sánchez-Brunete Álvarez

Jonathan Carrero Aranda

Ingeniería del Software

Facultad de Informática

Universidad Complutense de Madrid



ÍNDICE

1.	INTRODUCCIÓN.....	2
1.1.	Propósito	2
1.2.	Alcance.....	2
1.3.	Definiciones, acrónimos y abreviaturas.....	2
1.4.	Referencias	3
1.5.	Resumen	3
2.	DESCRIPCIÓN GENERAL	4
2.1.	Perspectiva del producto.....	4
2.2.	Funciones del producto	4
2.3.	Características del usuario.....	5
2.4.	Restricciones.....	5
2.5.	Supuestos y dependencias	5
2.6.	Requisitos futuros	5
3.	REQUISITOS ESPECÍFICOS.....	7
3.1.	Interfaces externos.....	7
3.2.	Funciones.....	7
3.2.1.	Gerencia.....	7
3.2.2.	Mesas.....	iError! Marcador no definido.
3.2.3.	Juego	iError! Marcador no definido.
3.2.4.	Apuestas.....	iError! Marcador no definido.
a.	Requisitos de rendimiento	14
b.	Requisitos lógicos de la base de datos	15
c.	Restricciones de diseño	15
d.	Atributos del sistema software	15
2.	APÉNDICES.....	17

1. INTRODUCCIÓN

1.1. Propósito

Este documento contiene la especificación de requisitos del software (SRS) del proyecto SAC (Sistema de Administración de Casinos) siguiendo el estándar IEEE 830-1998. Dicho documento pretende ofrecer los requisitos que tendrán el software y aquellas funciones que, en cambio, no llegará a ofrecer.

Los aspectos destacados son aquellos que deberán tenerse en cuenta en el desarrollo del software, no obstante, en este documento también aparecen las limitaciones y restricciones de la aplicación.

El documento va dirigido tanto a los clientes como a los desarrolladores. Por este motivo hemos intentado utilizar un lenguaje poco rico en tecnicismos. En cualquier caso, puede consultarse el **punto 1.3**, donde se recogen algunas abreviaciones y jergas del ámbito informático.

1.2. Alcance

Nuestro proyecto consiste, principalmente, en la gestión de un casino a nivel de qué es lo necesario para que un jugador pueda interaccionar dentro del propio casino con sus distintos juegos.

Entre los servicios que estableceremos tanto a usuarios registrados como invitados será el acceso a la página web (en la cual se proporciona un registro gratuito para comenzar a jugar dentro del casino). Los juegos disponibles que aparecerán una vez identificado serán: ruleta, blackjack y póker.

Para el control de los juegos, existirá un usuario que es el que llevara la gestión de todo el casino (esta parte es invisible para los jugadores registrados en la web). Este usuario será llamado el *superusuario* (gerente). Su labor es la administración de los clientes y de los juegos, es decir, administrará toda la aplicación web.

1.3. Definiciones, acrónimos y abreviaturas

SAC: Sistema de Administración de Casinos.

Casino: instalación que alberga y tiene capacidad para ciertas actividades de distintos tipos de juegos de azar.

Crupier: empleado de los casinos, salas de juegos o centros de ocio en los que se organizan juegos de azar o juegos de mesa. Su función consiste en controlar las apuestas del juego.

Mesa: lugar en el que se desarrollan las partidas referentes a los distintos juegos que puede llegar a ofrecer el casino.

Superusuario: la cuenta de *superusuario* (también llamado *root*) existe en cualquier aplicación web para facilitar la administración del sistema. A diferencia de las demás cuentas de usuario, el *superusuario* puede operar sin límites dentro del sistema, obteniendo así el máximo de privilegios posibles a nivel de configuración.

1.4. Referencias

- IEEE Std. 830-1998.
- Documento del Plan de proyecto.
- Casos de uso.

1.5. Resumen

El presente documento respeta la estructura recomendada por el estándar “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830-1998”.

- Sección 1: Introducción. Una visión general de los objetivos de la aplicación.
- Sección 2: Descripción del producto. Factores que afectan al producto, funciones, características, restricciones y requisitos futuros.
- Sección 3: Requisitos específicos. Interfaces, rendimiento, restricciones de diseño y atributos del Software.

2. DESCRIPCIÓN GENERAL

2.1. Perspectiva del producto

SAC es una aplicación web cuya función principal es llevar la administración de un casino. La aplicación ofrecerá información de todas las salas de juego y horarios de sus mesas. Apoyándose en una gran BBDD relacional, los usuarios, al registrarse, tendrán la posibilidad de elegir juegos y disfrutarán de contenido personalizado basado en el que podrán consultar su historial de juegos.

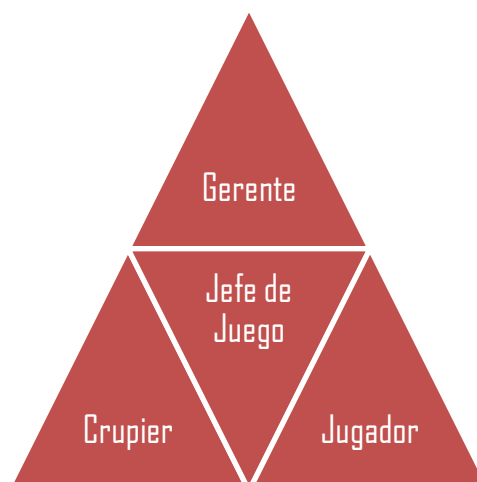
La aplicación será totalmente abierta, se podrá consultar cualquier juego sin necesidad de estar registrado, aunque sí hará falta para comenzar a jugar.

Por otro parte, estará gestionada por el gerente (*superusuario*) que podrá dar de alta y baja a los empleados que están a cargo de los juegos y a la vez ellos podrán dar de alta o baja a los crupieres de cada juego.

En lo relativo a la interfaz de usuario, se tratará de una interfaz gráfica que mostrará en pantalla una ventana con diversos botones y las diversas funcionalidades que hemos descrito en este documento. Cuando seleccionemos una de estas opciones, se abrirá una ventana nueva donde se pedirá al usuario los datos de entrada necesarios para realizar esa acción.

2.2. Funciones del producto

La función principal de este software es la de gestionar y administrar los juegos que posee un casino. Tanto a nivel de apuestas y clientes (usuarios que participan), como a nivel de empleados y sus responsabilidades (jefes de juego que controlan a los crupieres que tienen a su cargo).



Dentro de la aplicación web habrá diferentes jerarquías en función del usuario que acceda a ella. Concretamente habrá cuatro tipos de usuarios registrados: jugadores,

crupier, jefes de juego y gerente, teniendo cada uno de estos unos permisos y responsabilidades diferentes al resto. Así pues, podríamos definir las funciones del producto desde estos cuatro perfiles.

2.3. Características del usuario

SAC está destinada a ofrecer servicio a todo aquel que quiera jugar en casinos y que le interese este tipo de ocio. El manejo de la aplicación web es muy sencillo, pudiendo ser utilizada para cualquier tipo de usuario. Por motivos legales, el uso de la aplicación queda reservado para personas mayores de edad (pues un menor no puede apostar en este tipo de juegos).

2.4. Restricciones

Nuestro software va a ser una aplicación web, esto quiere decir en que no se requiere de un hardware demasiado potente ya que el proceso de la aplicación la llevara el servidor. Haría falta tener las versiones nuevas de los navegadores web (por la compatibilidad de HTML 5 mayoritariamente).

Hay que tener necesariamente ancho de banda, si es posible que sea de fibra para un mejor debido a un buen rendimiento en los juegos. Nuestro servidor dispondrá de un número elevado de accesos simultáneos. Para implementar la aplicación web haremos uso de HTML5, CSS3, Java y MySQL. Estará programado que el servidor haga copias de seguridad (en discos duros externos) cada “x” tiempo, para asegurar que no se pierda ningún dato. El servidor dispondrá de un sistema SAI (Sistema de Alimentación Ininterrumpida), administrado mediante una red NUT (Network UPS Tools). Las comunicaciones entre los usuarios y el servidor se realizaran a través de HTTPS (protocolo web de cifrado), con un certificado SSL, aprobado por una Autoridad de Certificación Española.

2.5. Supuestos y dependencias

A lo largo del desarrollo de este documento, se han mantenido reuniones con el profesor de la asignatura, que actúa como Jugador (cliente), para establecer los requisitos del software, de modo que han quedado ya cerrados con escasa probabilidad de ser modificados en el futuro. Estas reuniones, además, han tenido como objetivo eliminar los requisitos cuya implementación resulte más compleja, ya que este proyecto se centra sobre todo en la parte del uso de técnicas de Ingeniería del Software.

2.6. Requisitos futuros

Creemos que uno de los requisitos más importantes con vistas hacia el futuro, sería el implementar la opción de crear un nuevo juego (además de los tres que tenemos por defecto). Esta implementación estaría situada al nivel de gerente del casino, pues solo él tendría permisos para acceder a ella.

Además de crear nuevos juegos, se podrían dar de alta nuevas mesas en dichos juegos y contratar nuevos crupieres para controlar las funcionalidades de dichas mesas. Por descontado que cada usuario registrado en el casino podría acceder a estos nuevos juegos y apostar en ellos.

3. REQUISITOS ESPECÍFICOS

3.1. Interfaces externos

En esta sección especificaremos aquellos requisitos que intervienen en el proceso de desarrollo del software como por ejemplo qué interfaz le gustaría al usuario, la interfaz del hardware y qué entorno necesita el sistema.

La Interfaces de Usuario la trataremos de realizar muy parecidas a la Windows, para que de esta forma el usuario se sienta familiarizado, ya que la mayoría de las personas hemos trabajado o conocemos la interfaz gráfica de este sistema operativo y nos parece demasiado amigable y fácil de utilizar.

La interfaz del hardware no se desarrollará porque la aplicación no tiene relación con otros sistemas operativos.

Por otro lado, se ha identificado la información que almacenará cada módulo:

- **M. Gerencia:** Identificador del gerente (nombre, apellidos, DNI) e historial de modificaciones realizadas (cambios en los juegos y administración de los juegos y mesas).
- **M. Jefe de Juego:** Identificador de los jefes de juego (nombre, apellidos, DNI), cantidad de apuestas a poder realizar (aforo de las apuestas), código de cada mesa y su identificador de juego.
- **M. Crupier:** Identificador del crupier (nombre, apellidos, DNI) y horario disponible de juego.
- **M. Jugador:** Identificador del jugador (nombre, apellidos, DNI), cantidad de dinero disponible por parte del jugador, es decir, recursos de los que dispone e historial de apuestas realizadas (con los campos de pérdidas o ganancias de dichas apuestas).

3.2. Funciones

3.2.1. Gerencia

1. Dar de alta un juego

Prioridad: Alta.

Estabilidad: Alta.

Descripción: Permite crear un nuevo juego en el casino.

Entradas: Identificador del juego.

Salidas: Confirmación de la creación del juego.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Añade el juego (con sus campos necesarios) y le asigna un ID en su campo clave.

Requiere: Base de datos de los juegos.

Precondición: Juego no existe.

Postcondición: Juego existe y está disponible para los jugadores.

2. Dar de baja un juego

Prioridad: Media.

Estabilidad: Alta.

Descripción: Permite eliminar un nuevo juego en el casino.

Entradas: Identificador del juego.

Salidas: Confirmación de la eliminación del juego.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Elimina el juego y libera su ID.

Requiere: Base de datos de los juegos.

Precondición: Juego existe.

Postcondición: Juego dado de baja.

3. Mostrar las cuentas generales (pérdidas o ganancias) del casino

Prioridad: Muy alta.

Estabilidad: Muy alta.

Descripción: Muestra la cantidad de recursos disponibles en todo el casino.

Entradas: Recursos obtenidos en cada una de las mesas y, a su vez, en cada uno de los juegos.

Salidas: Nada.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Mostrar una tabla con los recursos.

Requiere: Base de datos de las apuestas (en términos monetarios).

Precondición: Ninguna.

Postcondición: Datos mostrados. Si no se han realizado apuestas, igualmente mostrará los recursos de los que dispone el casino en ese instante.

3.2.2. Jefe de Juego

1. Dar de alta un jefe de juego

Prioridad: Alta.

Estabilidad: Alta.

Descripción: Da de alta un jefe de juego.

Entradas: Datos sobre la persona (con campo clave DNI).

Salidas: Confirmación de alta.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Crea un nuevo jefe de juego y le asigna el ID del juego sobre el que es responsable.

Requiere: Base de datos de los jefes de juego.

Precondición: El DNI del jefe de juego no existe y tampoco hay ningún jefe de juego asignado a ese juego.

Postcondición: Jefe de juego queda asignado a un determinado juego.

2. Dar de baja un jefe de juego

Prioridad: Media.

Estabilidad: Alta.

Descripción: Da de baja un jefe de juego.

Entradas: Datos sobre la persona (con campo clave DNI).

Salidas: Confirmación de baja.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Elimina un jefe de juego y le asigna el ID del juego sobre el que es responsable.

Requiere: Base de datos de los jefes de juego.

Precondición: El DNI del jefe de juego existe.

Postcondición: Jefe de juego queda dado de baja.

3. Dar de alta una mesa

Prioridad: Alta.

Estabilidad: Alta.

Descripción: Da de alta una nueva mesa.

Entradas: Identificador de la mesa.

Salidas: Confirmación de alta.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Crea una nueva mesa que será asignada al juego en el que sea responsable el jefe de juego.

Requiere: Base de datos de los juegos y de las mesas.

Precondición: El identificador de la mesa no existe.

Postcondición: Mesa existe y está disponible para realizar apuestas.

4. Dar de baja una mesa

Prioridad: Media.

Estabilidad: Alta.

Descripción: Da de baja una mesa.

Entradas: Identificador de la mesa.

Salidas: Confirmación de baja.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Elimina una mesa).

Requiere: Base de datos de los juegos y de las mesas.

Precondición: El identificador de la mesa existe.

Postcondición: Mesa deja de existir (y, por lo tanto, libera al crupier que estuviera trabajando en dicha mesa).

5. Mostrar las cuentas (pérdidas o ganancias) del juego sobre el que sea responsable

Prioridad: Muy alta.

Estabilidad: Muy alta.

Descripción: Muestra la cantidad de recursos disponibles en un determinado juego.

Entradas: Recursos obtenidos en cada una de las mesas.

Salidas: Nada.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Mostrar una tabla con los recursos.

Requiere: Base de datos de las apuestas (en términos monetarios).

Precondición: Ninguna.

Postcondición: Datos mostrados. Si no se han realizado apuestas, igualmente mostrará los recursos de los que dispone el casino en ese instante.

6. Modificar los horarios de las mesas bajo su responsabilidad

Prioridad: Muy alta.

Estabilidad: Media.

Descripción: Permite modificar los horarios de apuestas.

Entradas: Identificador de las mesas.

Salidas: Confirmación de las modificaciones.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Establece nuevos horarios de apuestas, es decir, permite modificarle los horarios de trabajo a los crupieres.

Requiere: Base de datos de las mesas.

Precondición: Ya hay un horario establecido.

Postcondición: Nuevos horarios de apuestas.

3.2.3. Crupier

1. *Dar de alta un crupier*

Prioridad: Alta.

Estabilidad: Alta.

Descripción: Da de alta un crupier.

Entradas: Datos sobre la persona (con campo clave DNI).

Salidas: Confirmación de alta.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Crea un nuevo crupier y le asigna un ID de la mesa sobre la que trabajará.

Requiere: Base de datos de los crupieres y de las mesas.

Precondición: El DNI crupier no existe y tampoco hay ningún crupier asignado a esa mesa.

Postcondición: Crupier queda asignado a una mesa.

2. *Dar de baja un crupier*

Prioridad: Media.

Estabilidad: Alta.

Descripción: Da de baja un crupier.

Entradas: Datos sobre la persona (con campo clave DNI).

Salidas: Confirmación de baja.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Elimina un crupier.

Requiere: Base de datos de los crupieres y de las mesas.

Precondición: El DNI crupier existe.

Postcondición: Crupier queda dado de baja.

3. *Mostrar las cuentas (pérdidas o ganancias) de la mesa sobre la que sea responsable*

Prioridad: Muy alta.

Estabilidad: Muy alta.

Descripción: Muestra la cantidad de recursos disponibles en una determinada mesa.

Entradas: Recursos obtenidos en cada una de las apuestas.

Salidas: Nada.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Mostrar una tabla con los recursos.

Requiere: Base de datos de las apuesta (en términos monetarios).

Precondición: Nada.

Postcondición: Datos mostrados. Si no se han hecho apuestas, mostrará los fondos de los que dispone esa mesa en concreto.

4. Consultar jugadores que haya en su mesa

Prioridad: Muy alta.

Estabilidad: Media.

Descripción: Muestra los jugadores que haya apostando.

Entradas: DNI de cada jugador.

Salidas: Nada.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Permite conocer cuántos jugadores hay apostando en un momento dado e identificarlos.

Requiere: Base de datos de los jugadores.

Precondición: Debe haber, al menos, un jugador apostando.

Postcondición: El crupier conoce a cada uno de los jugadores contra los que se 'enfrenta'.

5. Apostar en la mesa en la cual trabaje

Prioridad: Muy alta.

Estabilidad: Alta.

Descripción: El crupier apuesta con cada jugador.

Entradas: Datos sobre el funcionamiento del juego (aunque esto se ejecute de forma transparente) y apuestas realizadas.

Salidas: Nada.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Permite conocer cuántos jugadores hay apostando en un momento dado e identificarlos.

Requiere: Base de datos de los jugadores y de las apuestas.

Precondición: Debe haber, al menos, un jugador apostando.

Postcondición: Se conoce el resultado de la apuesta realizada.

3.2.4. Jugador

1. Dar de alta un jugador

Prioridad: Alta.

Estabilidad: Alta.

Descripción: Da de alta un jugador.

Entradas: Datos sobre la persona (con campo clave DNI).

Salidas: Confirmación de alta.

Origen: Operador consola.
Destino: El sistema en su conjunto.
Acción: Crea un nuevo jugador.
Requiere: Base de datos de los jugadores.
Precondición: El DNI del jugador no existe.
Postcondición: Se crea un nuevo perfil para un jugador.

2. Dar de baja un jugador

Prioridad: Alta.
Estabilidad: Alta.
Descripción: Da de baja un jugador.
Entradas: Datos sobre la persona (con campo clave DNI).
Salidas: Confirmación de baja.
Origen: Operador consola.
Destino: El sistema en su conjunto.
Acción: Crea un nuevo jugador.
Requiere: Base de datos de los jugadores.
Precondición: El DNI del jugador existe.
Postcondición: Se elimina el perfil del jugador.

3. Mostrar sus propias cuentas

Prioridad: Muy alta.
Estabilidad: Muy alta.
Descripción: Muestra los recursos disponibles del jugador en cuestión.
Entradas: Recursos obtenidos en cada una de las apuestas.
Salidas: Nada.
Origen: Operador consola.
Destino: El sistema en su conjunto.
Acción: Mostrar una tabla con los recursos.
Requiere: Base de datos de las apuestas del jugador e historial de apuestas.
Precondición: Nada.
Postcondición: Datos mostrados. Si no se han hecho apuestas, mostrará los fondos de los que dispone el jugador.

4. Mostrar su historial de apuestas

Prioridad: Muy alta.
Estabilidad: Muy alta.
Descripción: Muestra cada una de las apuestas.
Entradas: La apuesta que realiza en cada momento.
Salidas: Nada.
Origen: Operador consola.
Destino: El sistema en su conjunto.

Acción: Mostrar el historial (mesa donde ha jugado, juego al que ha jugado y resultado de la apuesta).

Requiere: Base de datos de las apuestas del jugador.

Precondición: Se ha debido realizar alguna apuesta.

Postcondición: El propio historial.

5. Apostar en una determinada mesa

Prioridad: Muy alta.

Estabilidad: Alta.

Descripción: El jugador elige la mesa en la que apostar.

Entradas: ID del juego e ID de la mesa.

Salidas: Confirmación de la apuesta.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: El jugador elige una mesa (dentro de un juego) en la cual puede realizar apuestas.

Requiere: Base de datos de los juegos y de las mesas

Precondición: Debe haber, al menos, un juego y una mesa existentes.

Postcondición: La apuesta del jugador se realiza.

6. Consultar jugadores que haya en su mesa

Prioridad: Muy alta.

Estabilidad: Alta.

Descripción: Muestra los datos del jugador.

Entradas: DNI del jugador.

Salidas: Nada.

Origen: Operador consola.

Destino: El sistema en su conjunto.

Acción: Permite conocer todos los campos asociados al jugador (nombre, apellidos, edad...).

Requiere: Base de datos de los jugadores.

Precondición: El perfil debe existir.

Postcondición: El jugador conoce su información detalladamente.

3.3. Requisitos de rendimiento

El único requisito esencial que necesitaría nuestro software, aparte de tener java, es una buena conexión de ancho de banda. Además, sería conveniente contar con un buen hosting y nombres de dominio.

3.4. Requisitos lógicos de la base de datos

Será necesario establecer un modelo E/R (diagrama de entidad relación) para el modelado de la base de datos. Dicho modelo nos permitirá representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

Con los diagramas de entidad relación, observamos que, por ejemplo, un determinado Jugador tiene una relación 1:N con los distintos juegos, donde N es la cantidad de juegos que podrá elegir dentro del casino. Los crupieres, juegos, mesas y jugadores, se verán relacionados mediante los diagramas. Cada campo de la base de datos tendrá un ID que lo identificará de manera única (el DNI podría ser un ID para un determinado Jugador).

3.5. Restricciones de diseño

Hemos llegado a la conclusión de que al tratarse de un proyecto en el que en principio no se persigue que nuestra aplicación web sea la más eficiente entre todas y de que estamos hablando de un proyecto académico, quizá nos resulta complejo el llegar a una gran profundidad (hablando a niveles profesionales) en cuanto al lenguaje de programación o lenguajes que utilicemos en su desarrollo. Más en cuanto a la funcionabilidad que en cuanto al propio diseño de la interfaz (pues sí tenemos bastantes conocimientos en lenguajes de diseño como HTML5, CSS3 y Javascript).

3.6. Atributos del sistema software

El atributo por excelencia que definiría nuestra aplicación es la portabilidad, porque de este modo se podrá utilizar o usar desde cualquier dispositivo (teléfonos móviles, tabletas, PC, Mac, etc.) que tenga conexión a internet. También la usabilidad será otro componente importantísimo para atraer clientes, intentando así que sea facilísima de utilizar, rápida de entender y simple.

Conforme pase el tiempo, se irá optimizando dependiendo de los resultados obtenidos. Se diseñará pensando en la accesibilidad para aquellas personas con algún tipo de discapacidad o con avanzada edad. El mantenimiento de la aplicación será constante para que los clientes tengan la mejor experiencia posible; hay que recordar que solo el *superusuario* (Gerente) es quien tiene todos los permisos sobre la aplicación, los demás no.

La seguridad es otro pilar fundamental en nuestro sistema por eso dispondrá de la Ley Orgánica 15/1999 de protección de datos de carácter personal, así mismo la integridad de todos nuestros datos está protegida con sistemas hardware para evitar pérdidas o dañar la BBDD (a través del SAI), además de copias de seguridad establecidas.

4. APÉNDICES

Información sobre Decretos referentes a leyes que legislan las apuestas a nivel nacional en España:

- Decreto 58/2006 para la Comunidad de Madrid:

<http://www.madrid.org/ICMdownload/EVDA.pdf>

- Regulaciones para Juegos Online en apuestas:

http://www.elderecho.com/actualidad/dossier/Regulados-juego-apuestas-casinos-online_0_229875161.html