

The background features a decorative graphic consisting of three blue circles of varying sizes, each composed of concentric rings. Two thin blue lines intersect at a point between the top and middle circles, extending towards the top-left and bottom-right corners of the page. A large blue circle is partially visible in the bottom-right corner.

GESTOR DE EVENTOS DEPORTIVOS

OLIMPIADAS2MIL

Francisco Huertas
Gonzalo Ortíz
Alicia Pérez
Laura Reyero
Gabriela Ruíz
Rubén Tarancón

CURSO 2009/2010

CONTENIDO

Introducción	5
Propósito	5
Ámbito del sistema.....	5
Definiciones, Acrónimos y Abreviaturas	6
Descripción general.....	8
Perspectiva del proyecto.....	8
Funciones del sistema	11
Características de los usuarios	11
Consideraciones de seguridad - Access control	13
Suposiciones y dependencias.....	15
Requisitos futuros	16
Requisitos específicos	18
Planificación	20
Casos de uso.....	23
Diagrama de clases.....	31
Observaciones	32
Restricciones ocl.....	33
Diagramas de estado.....	40
Eventos.....	40
Observaciones	40
Entradas	41
Observaciones	41
Diagramas de actividad	42
Eventos.....	42
Observaciones	42
Entradas	43
Observaciones	43
Buscar emplazamiento.....	44
Observaciones	44
Diagramas de secuencia	45
Obtener clasificación.....	45
Observaciones	45
Consultar horarios personalizados.....	46

Observaciones	46
Crear acta	47
Observaciones	47
Reservar entradas	48
Observaciones	48
Diagramas de componentes	49
Diagramas de Componentes	49
Base de datos	49
Gestores	50
Interfaces.....	52
Modelo de datos	53
Diagrama entidad-relación.....	53
Modelo relacional	54
Descripción de las tablas.....	54
Funciones de la base de datos	57
Patrones	63
Patrón singleton	63
Patrón DAO (Data Access Object)	63
Interfaz de usuario	64
Aplicación Local.....	64
Interfaz web	66
Investigación sobre los componentes.....	81
Propuesta para la interfaz.....	81
Propuesta para la base de datos.....	82
Propuesta para el blog	83
Propuesta para el componente de pago.....	83
Propuesta para la visualización de eventos	83
Cuestionario CMMI	85
Evolución del proyecto.....	92

INTRODUCCIÓN

PROPÓSITO

Queremos ofrecer una herramienta que facilite la gestión de eventos deportivos de grandes magnitudes.

El propósito es integrar todo lo que conlleva la planificación de un acontecimiento de tan grandes magnitudes en una herramienta. Que ésta sea capaz de fusionar la organización deportiva del mismo como la interacción con el usuario a través de una página web.

ÁMBITO DEL SISTEMA

Queremos ofrecer una herramienta que facilite la gestión de eventos deportivos de grandes magnitudes. Para ello elaboramos una aplicación que administra los eventos desde la adjudicación del proyecto hasta la finalización del mismo.

Una vez adjudicado el proyecto, se solicita al ayuntamiento o ayuntamientos de la ciudad o ciudades organizadoras una serie de datos relativos a sus municipios que nuestra aplicación utilizará para la gestión de los eventos

La entidad organizadora del evento deportivo nos proporcionará información relativa a las distintas competiciones que se disputarán (características deportivas, equipamiento necesario para los eventos, número de participantes, número de árbitros...) así como un listado de participantes, árbitros etc. asignados a cada competición (ver el apartado [Modelo de Datos](#)).

Una vez recopilada toda esta información, nuestra aplicación se encarga de organizar en el tiempo y el espacio todos los eventos relativos a cada competición, atendiendo a las limitaciones impuestas por cada deporte. Con esto nos referimos a decidir la ubicación y hora de los eventos, asignar los árbitros en función de los equipos que vayan a participar, encargarse de las entradas etc.

Así mismo, nuestra aplicación actúa de portal de noticias y de visualización de las características del evento deportivo (calendario, clasificación, etc.) según el perfil al que pertenezca el usuario.

También incluye funcionalidad adicional como el visor de eventos, con el que un usuario podrá ver los eventos deportivos en directo online.

DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

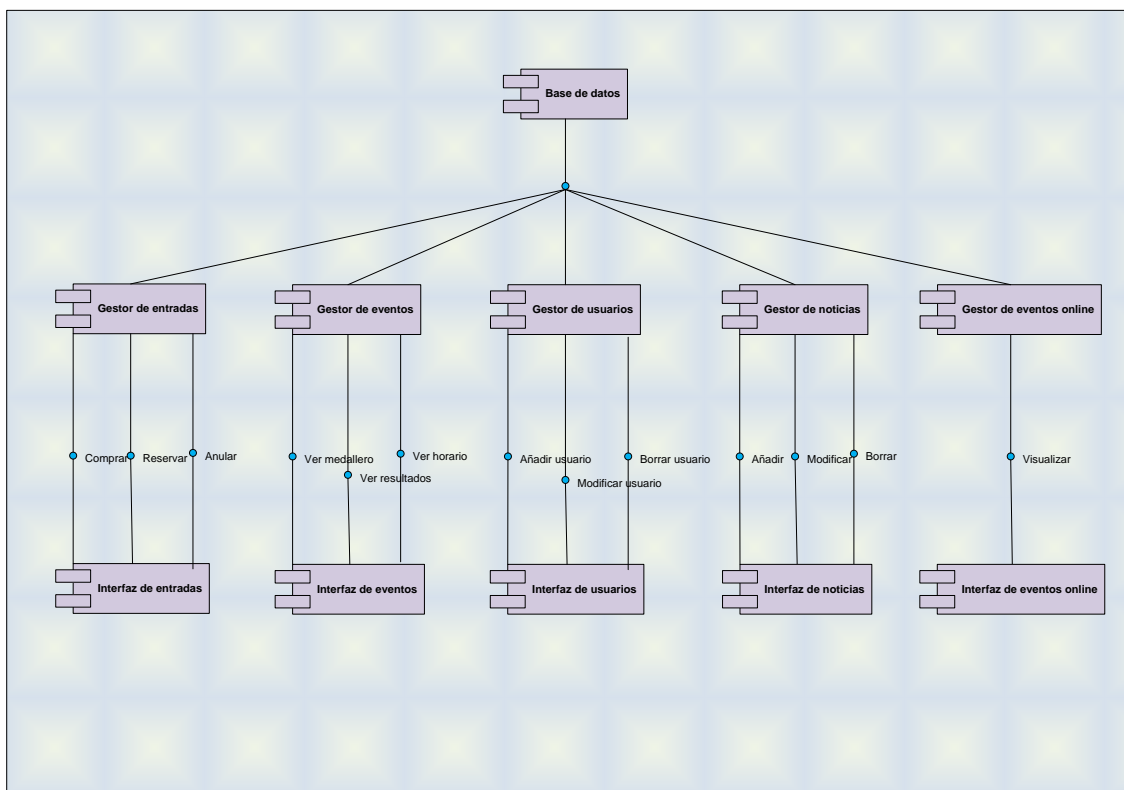
- **Competición:** En adelante competición es una categoría deportiva como podría ser el salto de altura o los cien metros lisos. Dada una competición puede saberse de manera unívoca sus normas y restricciones.
- **Evento:** Un evento es cada una de las pruebas que componen una competición.
- **Participante:** Persona que interviene activamente en un evento, bien sea para competir (en ese caso hablaremos de un deportista) o bien para moderar el evento (en ese caso árbitro).
- **Equipo:** Conjunto de deportistas que se asocian para participar en una competición. El equipo puede estar formado por uno o varios deportistas. Todos los integrantes son de la misma nacionalidad.
- **Emplazamiento:** Instalación en la que se van a desarrollar los eventos. Para que un evento se celebre en un emplazamiento éste tendrá que tener unas instalaciones acordes a las necesidades de cada competición.
- **Instalación:** Recursos de los que dispone un emplazamiento para los realizar los eventos.
- **Asiento:** Localidad que se encuentra en un emplazamiento susceptible de ser asociada a una entrada.
- **Evento final:** Ciertos tipos de competiciones de carácter eliminatorio (como todas las olímpicas) tendrán un evento final en la cual se decidirá el medallero.
- **Evento de consolación:** Ciertos tipos de competición de carácter eliminatorio tendrán una final de consolación en la que se competirá por el tercer y cuarto puesto. Típicamente ocurre en competiciones donde sólo juegan dos equipos por evento (como podría ser esgrima o fútbol).
- **COI:** Comité Olímpico Internacional. En principio la aplicación se plantea para organizar juegos olímpicos, aunque puede servir para organizar otros tipos de actividades deportivas. En este segundo caso, COI deja de referirse a este comité para referirse a la entidad gestora de la parte deportiva correspondiente.
- **Nacionalidad:** Federación nacional en la cual está federado el deportista o árbitro. Si la aplicación se usase para organizar competiciones no internacionales (como podría ser un torneo nacional), la nacionalidad de los participantes equivaldría al grupo o federación deportiva local en la que esté inscrito.
- **Medallero:** Cada competición tiene un medallero asignado que asignará la medalla de oro, plata y bronce al primer, segundo y tercer clasificado respectivamente.
- **Clasificación:** Ranking de los distintos equipos junto con su puntuación en una competición en un momento determinado.

- **Acta:** Conjunto de datos introducidos por el árbitro de un evento donde refleja los puntos más importantes de éste. Está compuesta por el resultado, las lesiones que se hayan producido y las sanciones aplicadas. También podrá introducir información adicional que sea relevante a nivel deportivo.
- **Sanción:** Penalización aplicada a un deportista en un evento concreto por una conducta antideportiva o antirreglamentaria. Se establece al introducir un acta.
- **Lesión:** Posible agravio sufrido por un deportista en un evento.
- **Resultado:** Será establecido al final de un evento y mostrará un listado de los participantes de un determinado evento junto a la puntuación obtenida en dicho evento. Esta puntuación puede ser el número de goles en fútbol, los segundos en carreras etc.
- **Entrada:** Billeto asociado al asiento de un emplazamiento para uno o varios eventos. La duración de la validez de la entrada será desde el inicio del primer evento hasta el final del último.
- **Entrada disponible:** Aquella entrada que no está ni comprada ni reservada por ningún usuario.
- **Entrada reservada:** Aquella entrada para la que un usuario ha manifestado al sistema su intención de compra en un plazo establecido
- **Entrada comprada:** Aquella entrada para la que un usuario ha efectuado el pago correspondiente al importe de esa entrada.
- **Tiempo de caducidad:** Tiempo máximo que puede encontrarse una entrada en estado reservado sin ser comprada. Pasado este tiempo, la entrada pasará a estado disponible.
- **Incidencia deportiva:** Con este término nos referiremos a sanciones o lesiones que afecten a un participante.
- **Núcleo de la aplicación:** Se refiere a la parte de la aplicación representada por el diagrama de clases, que suponiendo un entorno que realice las correctas llamadas a las funciones, funcionará correctamente con la interfaz propuesta o dentro de cualquier otro sistema.
- **Interfaz de la aplicación:** Se refiere a la parte de la aplicación que interactuará con el usuario. Se trata de un módulo adyacente al núcleo que recibirá peticiones de los usuarios, las comunicará al núcleo y mostrará al usuario los resultados.

DESCRIPCIÓN GENERAL

PERSPECTIVA DEL PROYECTO

DIAGRAMAS DE COMPONENTES



BASE DE DATOS

- La base de datos de nuestra aplicación constará de varias tablas para los diferentes usos del sistema. Habrá tablas para el registro de los usuarios, para la venta de entradas, equipos participantes, instalaciones, etc.
- Alguna de esas tablas nos la facilitará el COI (como la tabla equipos y la tabla competiciones) o el ayuntamiento de la ciudad organizadora, éstas no podrán ser modificadas. Otras tablas nos la dará el ayuntamiento de la ciudad organizadora (tabla instalaciones) y al igual que antes, nuestra aplicación no modificara esa tabla.
- El resto de tablas son las que mediante el sistema de gestión de bases de datos, abajo explicado, y los gestores harán que nuestra aplicación tenga una funcionalidad completa.
- En el apartado relativo a modelo de datos se explicara más detalladamente el formato de cada una de las tablas

SISTEMA DE GESTION DE BASE DE DATOS

- Es el componente que nos ayuda a gestionar la base de datos de la aplicación, el espacio físico y nos proporciona unas interfaces de acceso primitivas. Es el puente entre los gestores y las tablas de la base de datos.
- La interfaz consta de métodos de acceso a tablas (select), de creación de tablas (create), de borrado de tablas (drop) , borrado de información de una tabla (delete), modificación de información contenida en tablas (update) y de las propias tablas (alter) , y por ultimo un método para insertar información en las bases de datos (insert).

GESTOR DE ENTRADAS

- Es el componente a través del cual vamos a gestionar las entradas de los espectadores a las distintas disciplinas olímpicas.
- Va a ser el que va a recibir las ordenes de los compradores a través de la interfaz de entradas y el que va a tener que utilizar el SGDB para acceder y modificar las tablas. Gracias a él, los datos que los usuarios puedan introducir, serán coherentes con las tablas de la base de datos relativas a las entradas.
- Las operaciones que realizara sobre las tablas de la base de datos son:
 - Reservar entrada : añadirá una fila a la tabla de entradas reservadas
 - Comprar entrada: añadirá una fila a la tabla de entradas vendidas si la entrada que se intenta comprar pertenece a la tabla entradas reservadas y en caso afirmativo borrará la entrada de la tabla entradas reservadas.
 - Anular entrada : eliminará una fila de la tabla de entradas reservas

GESTOR DE USUARIOS

- Es el componente a través del cual vamos a gestionar los usuarios que van a poder disfrutar de las ventajas de esta aplicación.
- A través del formulario de registrarse de la interfaz el gestor va a recibir todos los datos necesarios para que el usuario sea registrado en la aplicación. A partir de ahí el gestor hará las operaciones necesarias para que en la tabla de usuario queden reflejados los cambios.
- Las operaciones que va a realizar el gestor son:
 - Registrar un usuario: añadir una fila a la tabla de usuarios.
 - Modificar usuario: modificar los datos en la fila de la tabla de usuarios.
 - Eliminar usuario: eliminar la fila de la tabla de usuarios.
 - Login: comprobar que los datos introducidos están en la tabla de usuarios.

GESTOR DE NOTICIAS, FOTOS

- Es el componente a través del cual vamos a gestionar las noticias y fotos que van aparecer en la herramienta.
- En la interfaz de noticias, el usuario podrá introducir el titular y el cuerpo de la noticia, o si lo desea, fotos. Las noticias y fotos se almacenaran en el servidor y el encargado será el gestor.
- Las operaciones que realiza son:
 - Añadir noticia
 - Modificar noticia
 - Añadir foto
 - Eliminar foto
 - Poner noticia en ultima hora
- En la parte relativa a componentes software se dará más información de este gestor y de cómo se va a integrar en nuestra aplicación

GESTOR DE EVENTO

- Es sin duda, el gestor más importante de nuestra aplicación a la vez que el más sofisticado.
- Utilizará las tablas de la base de datos de deportes, equipos e instalaciones que le vendrán dadas por el COI y el ayuntamiento para realizar un organigrama de toda la competición (en el diagrama de actividad Busca emplazamiento se detalla). Gracias a él se construirá el calendario completo de todos los eventos que se van a realizar en las olimpiadas y les podrá horario e instalaciones. También gestionará la tabla de resultados que después de cada partido se rellenará gracias a los actas y gracias a ello podremos visualizar inmediatamente las clasificaciones y el medallero. Por último el gestor tendrá operaciones más simples, pero no por ello menos importantes como pueden ser: modificar la información de un evento o modificar las actas de un partido.

INTERFACES

- Es el componente a través del cual el usuario va a poder acceder a todas las utilidades que ofrece el sistema, por lo tanto debe ser sencillo y muy intuitivo.
- Habrá una serie de interfaces, cada una la gestionará uno de los componentes arriba descritos.
- Más adelante se podrán ver con más detalle estas interfaces.

FUNCIONES DEL SISTEMA

Nuestro sistema gestiona principalmente aspectos de un proyecto deportivo:

- Gestión de competiciones: en este aspecto podemos encontrar las siguientes funciones:
 - Consultas: un usuario puede consultar información relativa al medallero, clasificación, participantes, calendario de eventos.
- Gestión de eventos:
 - Consultas: un usuario puede consultar información relativa a la fecha y la hora de un evento que aún no se haya celebrado, y consultar el resultado e incidencias de un evento celebrado.
 - Administración: un perfil administrador puede gestionar en qué lugar se va a celebrar un evento y a qué hora.
- Gestión de entradas:
 - El sistema se encarga de la venta, reserva y anulación de reserva de las entradas para los distintos eventos que estén planificados.
- Gestión de la información:
 - Consultas: cualquier usuario puede consultar el blog de noticias, puede visualizar eventos online tanto pasados como en tiempo real y puede recibir noticias en su correo o dispositivo móvil.
 - Administración: los perfiles administradores se encargarán de gestionar la información que se ofrece al usuario como las entradas del blog, etc.

CARACTERÍSTICAS DE LOS USUARIOS

USUARIO NO REGISTRADO

Otorgaremos este perfil a cualquier individuo que acceda a nuestra aplicación sin identificarse mediante una clave de usuario y una contraseña.

USUARIO REGISTRADO

Otorgaremos este perfil a cualquier individuo ajeno a la organización que acceda a nuestra aplicación identificándose mediante una clave de usuario y una contraseña.

ÁRBITRO

Otorgaremos este perfil a aquellos individuos identificados por la aplicación como árbitros.

COI

Otorgaremos este perfil a aquellos individuos identificados por la aplicación como miembros del Comité Olímpico Internacional.

AYUNTAMIENTO

Otorgaremos este perfil a aquellos individuos identificados por la aplicación como miembro del ayuntamiento de la localidad organizadora.

ADMINISTRACIÓN – GESTOR DE EVENTOS

Otorgaremos este perfil a aquellos usuarios con poderes de administración total en la aplicación.

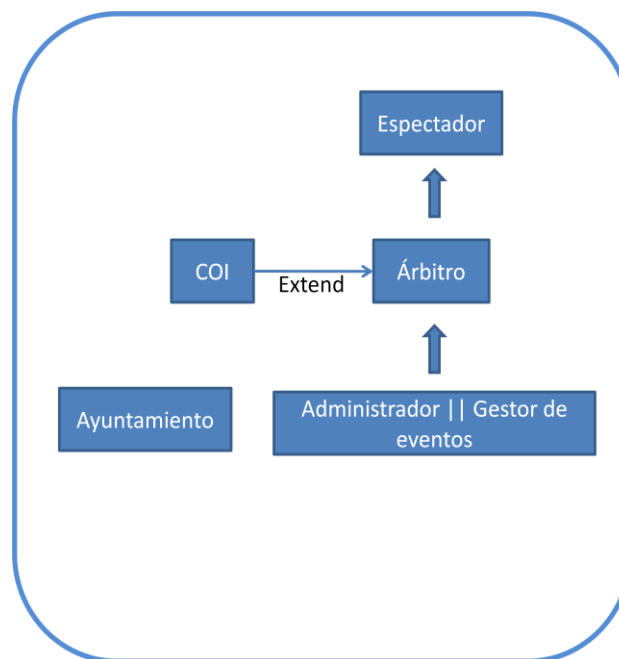


Figura: Diagrama de herencia entre los perfiles

CONSIDERACIONES DE SEGURIDAD - ACCESS CONTROL

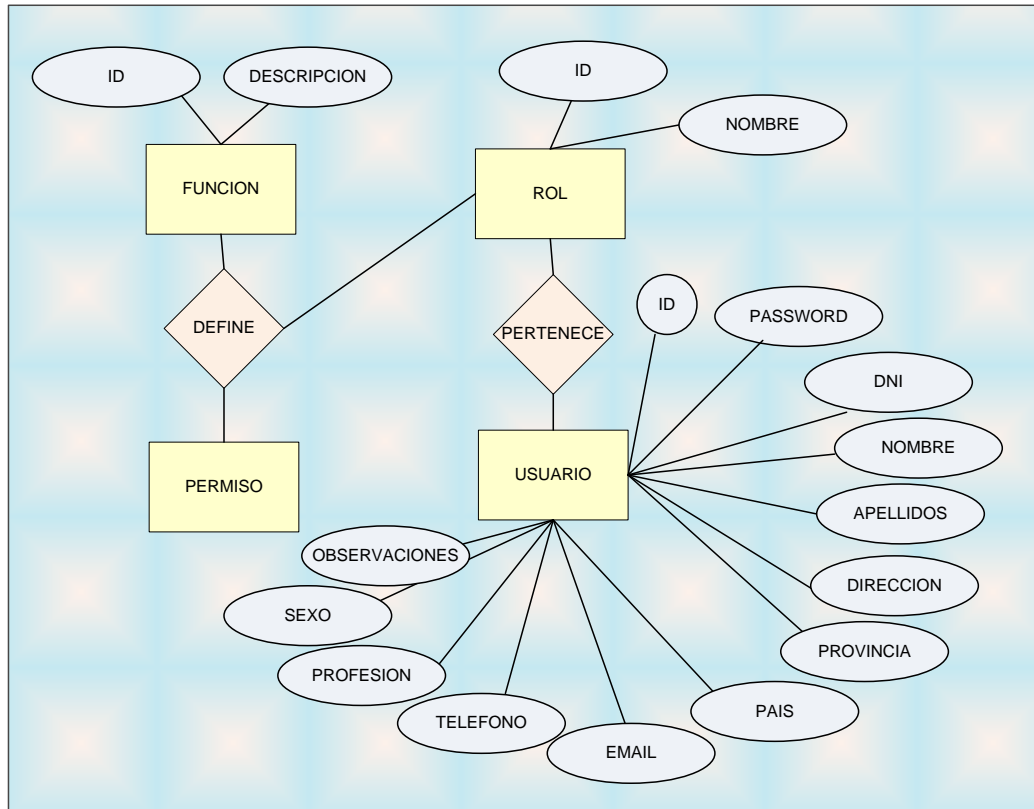
Como somos conscientes de la importancia de la protección de los datos de la aplicación, y pensamos que no se puede dejar en manos de un desarrollador, a continuación exponemos el sistema de seguridad que utilizaremos.

En la siguiente tabla se muestra una relación de las tablas a las que un usuario podrá acceder a través de la aplicación:

Usuario	Caso de uso	Tablas de la BBDD
COI	Crear evento	EVENTO
COI	Modificar evento	EVENTO
COI	Buscar emplazamiento	EMPLAZAMIENTO, LUGAR_EVENTO, EVENTO
COI	Reservar emplazamiento	EMPLAZAMIENTO, LUGAR_EVENTO, EVENTO
COI	Introducir árbitros	ARBITRO
COI	Introducir equipos	EQUIPO
COI	Introducir fecha y duración	EVENTO
Ayuntamiento	Introducir fotos	GESTOR DE BLOG, NO BBDD
Ayuntamiento	Introducir última hora	GESTOR DE BLOG, NO BBDD
Ayuntamiento	Introducir titulares	GESTOR DE BLOG, NO BBDD
Árbitro	Consultar acta	EVENTO_RESULTADO, RESULTADO, LESIÓN, SANCIÓN
Árbitro	Crear acta	EVENTO_RESULTADO, RESULTADO, LESIÓN, SANCIÓN
Árbitro	Modificar acta	EVENTO_RESULTADO, RESULTADO, LESIÓN, SANCIÓN
Usuario logueado	Logout	USUARIO
Usuario logueado	Borrarse	USUARIO
Usuario logueado	Modificar datos	USUARIO
Usuario logueado	Consultar entradas disponibles	ENTRADA
Usuario logueado	Reservar entrada	ENTRADA
Usuario logueado	Comprar entrada	ENTRADA
Usuario logueado	Anular entrada	ENTRADA
Usuario logueado	Ver evento online	GESTOR EVENTOS ONLINE
Usuario	Login	USUARIO
Usuario	Registro	USUARIO
Usuario	Consultar resultados	EVENTO_RESULTADO
Usuario	Consultar medallero	EVENTO_RESULTADO
Usuario	Consultar titulares	GESTOR BLOG
Usuario	Consultar última hora	GESTOR BLOG
Usuario	Consultar fotos	GESTOR BLOG
Usuario	Consultar disciplina	COMPETICIÓN
Usuario	Consultar calendario	LUGAR_EVENTO

El acceso a estas tablas se hará a través de la aplicación como ya hemos explicado anteriormente, pero dicha aplicación estará implementada de forma que la funcionalidad asignada a cada rol sea configurable por la base de datos.

Esto será posible gracias al siguiente modelo:



Con este modelo de datos el desarrollador no tiene que preocuparse de la funcionalidad según el perfil.

Al iniciar sesión, el usuario cargará de la base de datos toda la funcionalidad a la que tiene permisos. Al desarrollador lo único que le importará es, a la hora de mostrar u habilitar un componente del sistema, si el permiso para utilizarlo ha sido cargado al inicio. Si es así se mostrará/habilitará según el caso.

Las funciones serán las definidas en los casos de uso, y por defecto no se cargará ningún componente a no ser que se tenga el permiso correspondiente por temas de seguridad.

SUPOSICIONES Y DEPENDENCIAS

A continuación se enumeran todas aquellas cosas que se hayan dado por supuestas en la aplicación:

- En el núcleo
- Se ha puesto especial hincapié en que los métodos que relacionan eventos con emplazamientos y entradas cumplan las invariantes del sistema. Sin embargo se ha relajado el control de las invariantes que rigen la distribución de eventos, deportistas y jugadores delegando su control en la interfaz o, en su caso, en el usuario.
- Los requisitos que una competición pueda imponer a los emplazamientos donde se desarrollen sus eventos son tan variados como las propias competiciones. Por eso se ha simplificado el problema limitando las restricciones a tener piscinas, tener pistas de atletismo y tener circuitos de velocidad. Por ejemplo, la competición de salto triple y la de lanzamiento de peso tendrían, en nuestro sistema, los mismos requisitos: una pista de atletismo.
- Las lesiones y sanciones son meramente informativas. El sistema no impide que un jugador lesionado o sancionado compita en un evento.
- En el caso de los jugadores, si un jugador lesionado ha de participar en un evento es algo que se delega en las federaciones correspondientes y que un jugador sancionado participe en un evento es algo que ha de regular el árbitro de mismo.
- En el caso de los árbitros, el sistema no contempla que pueden ser sancionados pero sí contempla que se lesionen. El obligar a sustituir a un árbitro lesionado en los eventos en los que no pueda participar es una funcionalidad que se propone como futura.
- En la interfaz
- La interfaz es un anexo al núcleo. Por ello no se ha diseñado con tanto detalle como el núcleo. Por ejemplo, no hay diagrama de clases para la interfaz, sino que se describe su funcionamiento en un nivel más alto y abstracto.
- En las bases de datos
- Se presupone que las bases de datos ya vienen dadas por los distintos organismos: las competiciones, deportistas y árbitros así como los datos de cada uno de estos objetos viene dada por el COI o la organización deportiva correspondiente. La lista de emplazamientos con todos sus datos viene a su vez dada por la administración local.
- Se presupone la corrección de estas bases de datos dadas tanto con respecto a nuestros modelos de bases de datos como a las invariantes de núcleo.

REQUISITOS FUTUROS

A continuación se lista una serie de ampliaciones y mejoras propuestas:

- Núcleo:
- Control de los invariantes en la distribución de eventos, deportistas y jugadores. Esto no se ha realizado por requerir replantear desde un comienzo el método de inserción de estos datos. Si se añadiera podría enfocarse de varias maneras, entre las que se enumeran:
 - Suponiendo dada la distribución de eventos en el tiempo, limitar la distribución de los deportistas entre los equipos de tal manera que nunca un jugador pudiera verse en dos equipos que podrían acabar teniendo eventos que coincidieran en el tiempo. Este sistema tiene el inconveniente de que sería el COI quien tiene que amoldarse a la entidad local organizativa. Por ejemplo, un mal diseño evitaría que deportistas pudieran participar en las competencias de cien y doscientos metros lisos, algo bastante común, pero permitir a otro formar parte del equipo de fútbol y de baloncesto, algo realmente infrecuente.
 - Suponiendo dada la distribución de los jugadores en los equipos nacionales y los emplazamientos, organizar los eventos de tal manera que cumplan las invariantes. Este sistema delega en el COI la viabilidad o no del sistema. Por ejemplo, si el COI permite que un jugador esté inscrito en todos los deportes, es muy probable que dado el limitado número de emplazamientos, la olimpiada tuviera que durar muchísimo tiempo para poder cumplir que los eventos en los que potencialmente pueda estar ese jugador no se solapen.
 - Clasificando las competencias por categorías (atletismo velocidad, atletismo salto, fútbol, etc.) de tal manera que el COI se comprometa a que un deportista o un árbitro solo estará en competencias de una misma categoría, de tal manera que el sistema garantice que dada una categoría, los eventos de las competencias de estas categorías no se solapan. Este posiblemente fuera el mejor sistema, pero requeriría modificar más profundamente el diseño actual.
 - Gestión de incidencias deportivas: que un deportista lesionado o sancionado participe en un evento es algo que siempre habría que delegarse en el propio deportista y su federación. Sin embargo el sistema podría requerir al COI un sustituto para el participante lesionado o sancionado por los eventos que dure su incidencia si estuviera planificado que participase en eventos posteriores.
 - Mejorar el sistema de requisitos en los emplazamientos: El núcleo utiliza un campo para cada requisito que pueda imponer una competición a un emplazamiento. Para tres posibles restricciones como se tienen actualmente esto es aceptable, pero no si se éstas se extendiesen a las restricciones reales de tantas competencias como tendría una olimpiada. Para que el sistema fuera escalable debería usar un diseño de requisitos bastante más complejo.

- Otros:
- Eventos asignados automáticamente: Actualmente ha de ser un usuario el que asigne los eventos una fecha y un emplazamiento. Que esto fuera hecho automáticamente es algo que se quiso hacer desde un principio. Gracias a las funciones del núcleo se podría hacer una capa externa (al mismo nivel que la interfaz gráfica) que mediante algoritmos más o menos complejos realicen esta tarea automáticamente o al menos aconsejen al usuario dónde y cuándo asignar un evento. Sin embargo esto es algo tan dependiente de la implementación que en nuestro caso no tiene demasiado sentido hacer un diseño exhaustivo.
- Añadir al sistema una gestión del alojamiento, ya que teniendo la información relativa a los participantes de los eventos y suponiendo que al igual que los emplazamientos el ayuntamiento o los ayuntamientos organizadores nos facilitasen los datos relativos a los alojamientos de los municipios en cuestión y sus alrededores, podríamos calcular el número de habitaciones necesarias para alojar a los participantes (o incluso al personal organizador y de apoyo a los participantes como médicos, entrenadores etc.) y la forma de que el alojamiento fuese lo más óptimo para cada participante en función de los eventos previstos en los que fueran a participar.
- Añadir la gestión de transporte, si fuera en conjunto con gestión de alojamiento, suponiendo que el transporte como tal se delegaría en una empresa externa y que el sistema únicamente se encargaría de proporcionar las rutas para desplazar a los participantes o al personal indicado desde su alojamiento hasta el emplazamiento del evento que corresponda. Sin la funcionalidad del alojamiento también se podría desarrollar, suponiendo que conocemos el alojamiento del personal (aunque su distribución no sea la más óptima)

REQUISITOS ESPECÍFICOS

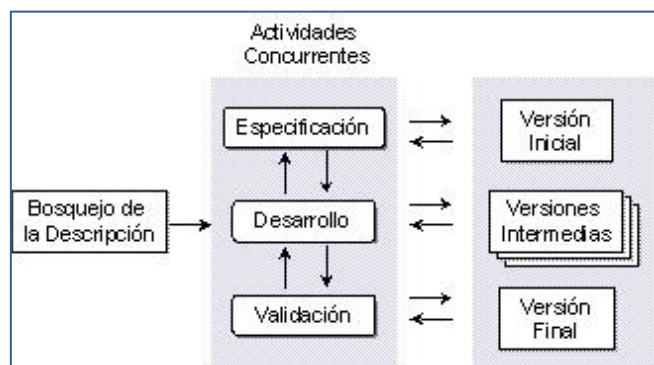
1. Todos los deportistas que forman parte de un mismo equipo, tiene la misma nacionalidad.
2. Para poder reservar una entrada es necesario que el usuario esté registrado en el sistema.
3. Para poder reservar una entrada es necesario que esa entrada esté disponible.
4. Para poder realizar la compra de una entrada, es necesario haberla reservado previamente.
5. Los organizadores determinarán el tiempo máximo que una entrada puede estar reservada sin comprar. Pasado este tiempo, la entrada volverá a estar disponible.
6. El software debe contar con un sistema de autenticación de usuarios a través de un identificador de usuario y una contraseña.
7. Para poder visualizar los eventos online, es necesario que el usuario esté registrado.
8. Cada comprador puede adquirir una o más entradas para asistir a un mismo evento.
9. Cada competición tendrá determinadas las necesidades del emplazamiento donde poder realizarse.
10. Cada competición determina el número de equipos y de árbitros necesarios para llevarla a cabo.
11. Una competición está formada por uno o varios eventos.
12. En competiciones eliminatorias donde sólo juegan dos equipos por evento (como podría ser esgrima, fútbol o tenis) existirá un evento de consolación que será el penúltimo evento de esa competición disputado
13. En competiciones eliminatorias, el ultimo evento será la final
14. Todos los emplazamientos adjuntan información sobre número y tipo de instalaciones que los componen.
15. Los emplazamientos tienen un número de asientos.
16. Un equipo no puede participar en dos eventos que se celebren a la vez.
17. Para cada evento deberá buscarse un emplazamiento atendiendo a las necesidades que impone la competición.
18. Tras la finalización de un evento el árbitro o árbitros encargados, deberán rellenar un acta con lo acontecido durante el mismo.

19. El acta del evento contendrá los diferentes resultados que han obtenido los equipos participantes, así como las lesiones o sanciones que se hayan podido producir durante el evento.
20. Toda la interacción del usuario con el sistema se realizará a través de un portal web.
21. En la aplicación web se exigirá estar registrado para poder tener acceso a información particular y para la venta de entradas.
22. Se podrá consultar la clasificación de las diferentes competiciones a través del portal web.
23. La organización puede publicar noticias y dar a conocer información sobre el estado del evento deportivo, la ciudad que lo alberga u otras noticias de interés general.
24. Se podrá visualizar en el portal un calendario con los próximos eventos que se celebrarán.
25. Un usuario registrado podrá visualizar un calendario personalizado de los eventos para los cuales haya adquirido entradas.
26. Un usuario registrado podrá ver el evento a través de la página web.

PLANIFICACIÓN

Para el desarrollo de nuestra aplicación, hemos decidido utilizar un desarrollo evolutivo. La idea detrás de este modelo es el desarrollo de una implantación del sistema inicial, exponerla a los comentarios del usuario y refinarla hasta que se desarrolle el sistema adecuado.

Una ventaja de este modelo es que se obtiene una rápida realimentación del usuario, ya que las actividades de especificación, desarrollo y pruebas se ejecutan en cada iteración.

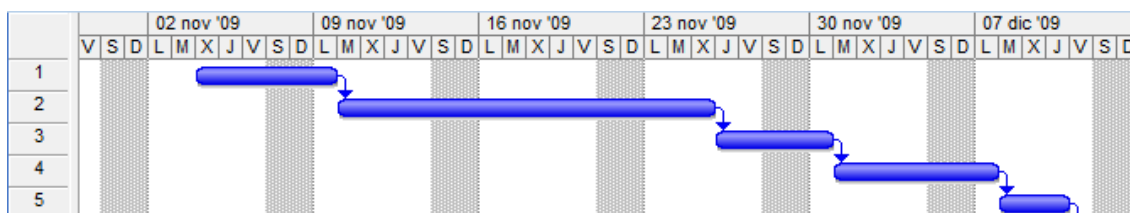


Nos hemos decantado por el Desarrollo Evolutivo Explorativo, cuyo objetivo es explorar con el usuario los requisitos hasta llegar a un sistema final. El desarrollo comienza con las partes que se tiene más claras. El sistema evoluciona conforme se añaden nuevas características propuestas por el usuario.

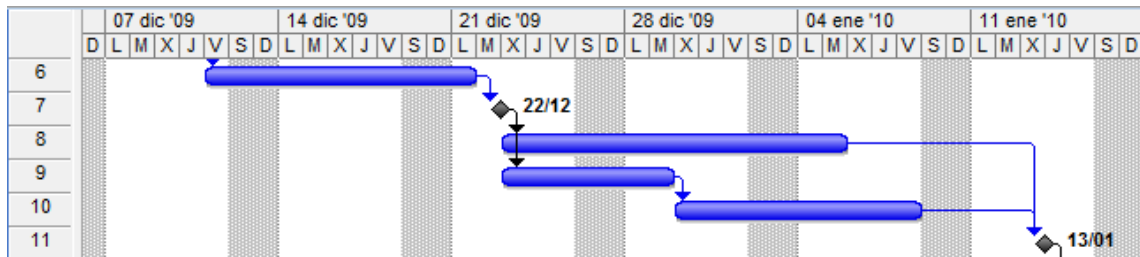
Para poder elaborar el sistema atendiendo a las necesidades del usuario, hemos planificado reuniones periódicas a lo largo del tiempo. En dichas reuniones, se expondrán los avances del proyecto y el usuario nos asesoraba sobre qué camino tomar.

A continuación mostramos una propuesta de planificación para las tareas que hemos adoptado en la elaboración del proyecto.

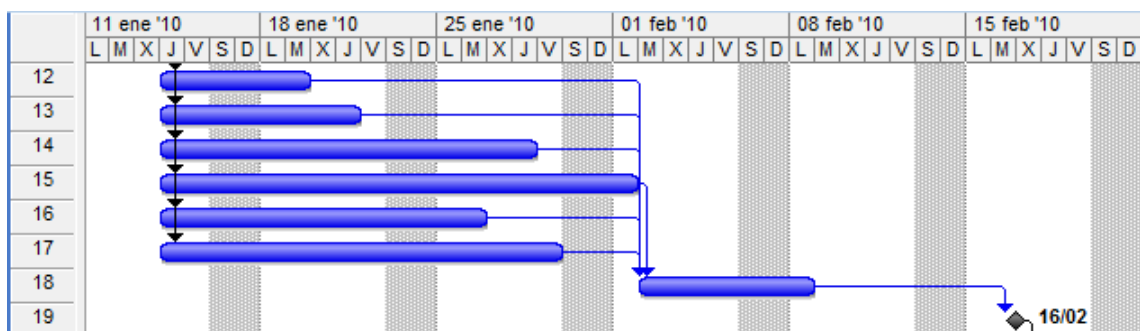
	Nombre de tarea	Duración	Comienzo	Fin
1	Reunión con el cliente	4 días	mié 04/11/09	lun 09/11/09
2	Realización de la primera versión de requisitos	12 días	mar 10/11/09	mié 25/11/09
3	Reunión con el cliente: presentación y aprobación/rechazo de los requisitos	3 días	jue 26/11/09	lun 30/11/09
4	Arreglo de los requisitos	5 días	mar 01/12/09	lun 07/12/09
5	Reunión con el cliente para la aprobación final de los requisitos	3 días	mar 08/12/09	jue 10/12/09



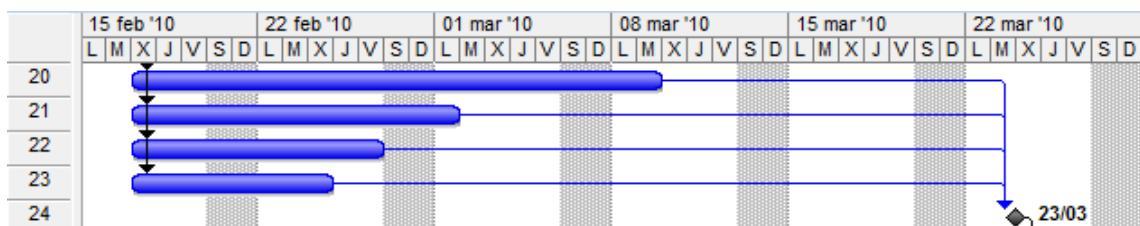
6	Planteamiento de los requisitos al equipo de trabajo y división de las tareas	7 días	vie 11/12/09	lun 21/12/09
7	Comienzo del desarrollo	1 día	mar 22/12/09	mar 22/12/09
8	Elaboración del diagrama de clases	10 días	mié 23/12/09	mar 05/01/10
9	Definición de los perfiles de usuario de la aplicación	5 días	mié 23/12/09	mar 29/12/09
10	Elaboración de los diagramas de casos de uso	8 días	mié 30/12/09	vie 08/01/10
11	Reunión con el cliente: presentación del trabajo desarrollado	3 días	lun 11/01/10	mié 13/01/10



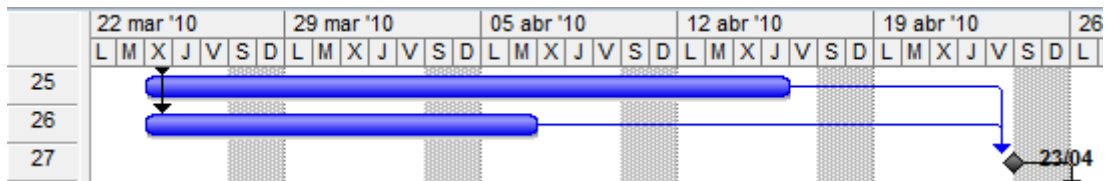
12	Modificaciones del diagrama de clases	4 días	jue 14/01/10	mar 19/01/10
13	Modificaciones de los diagramas de casos de uso	6 días	jue 14/01/10	jue 21/01/10
14	Elaboración de los diagramas de secuencia	11 días	jue 14/01/10	jue 28/01/10
15	Elaboración de los diagramas de actividad	13 días	jue 14/01/10	lun 01/02/10
16	Elaboración de los diagramas de estado	9 días	jue 14/01/10	mar 26/01/10
17	Elaboración de los diagramas de componentes	12 días	jue 14/01/10	vie 29/01/10
18	Preparación de la presentación al cliente	5 días	mar 02/02/10	lun 08/02/10
19	Reunión con el cliente: presentación del trabajo desarrollado	6 días	mar 09/02/10	mar 16/02/10



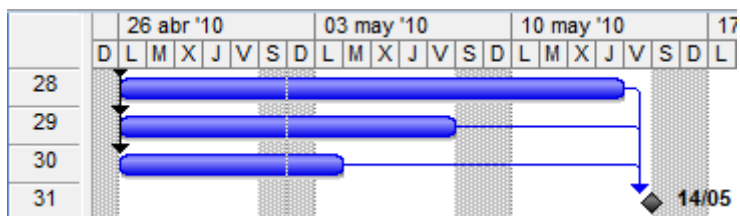
20	Elaboración de las modificaciones pertinentes en los diagramas	15 días	mié 17/02/10	mar 09/03/10
21	Elaboración del modelo de datos	9 días	mié 17/02/10	lun 01/03/10
22	Descripción de los patrones a utilizar	8 días	mié 17/02/10	vie 26/02/10
23	Planteamiento de una posible implementación	6 días	mié 17/02/10	mié 24/02/10
24	Reunión con el cliente: aprobación de los diagramas	10 días	mié 10/03/10	mar 23/03/10



25	Elaboración de un prototipo visual sin funcionalidad	17 días	mié 24/03/10	jue 15/04/10
26	Elaboración de la primera versión del documento oficial del proyecto	10 días	mié 24/03/10	mar 06/04/10
27	Reunión con el cliente: aprobación del prototipo y del documento	6 días	vie 16/04/10	vie 23/04/10



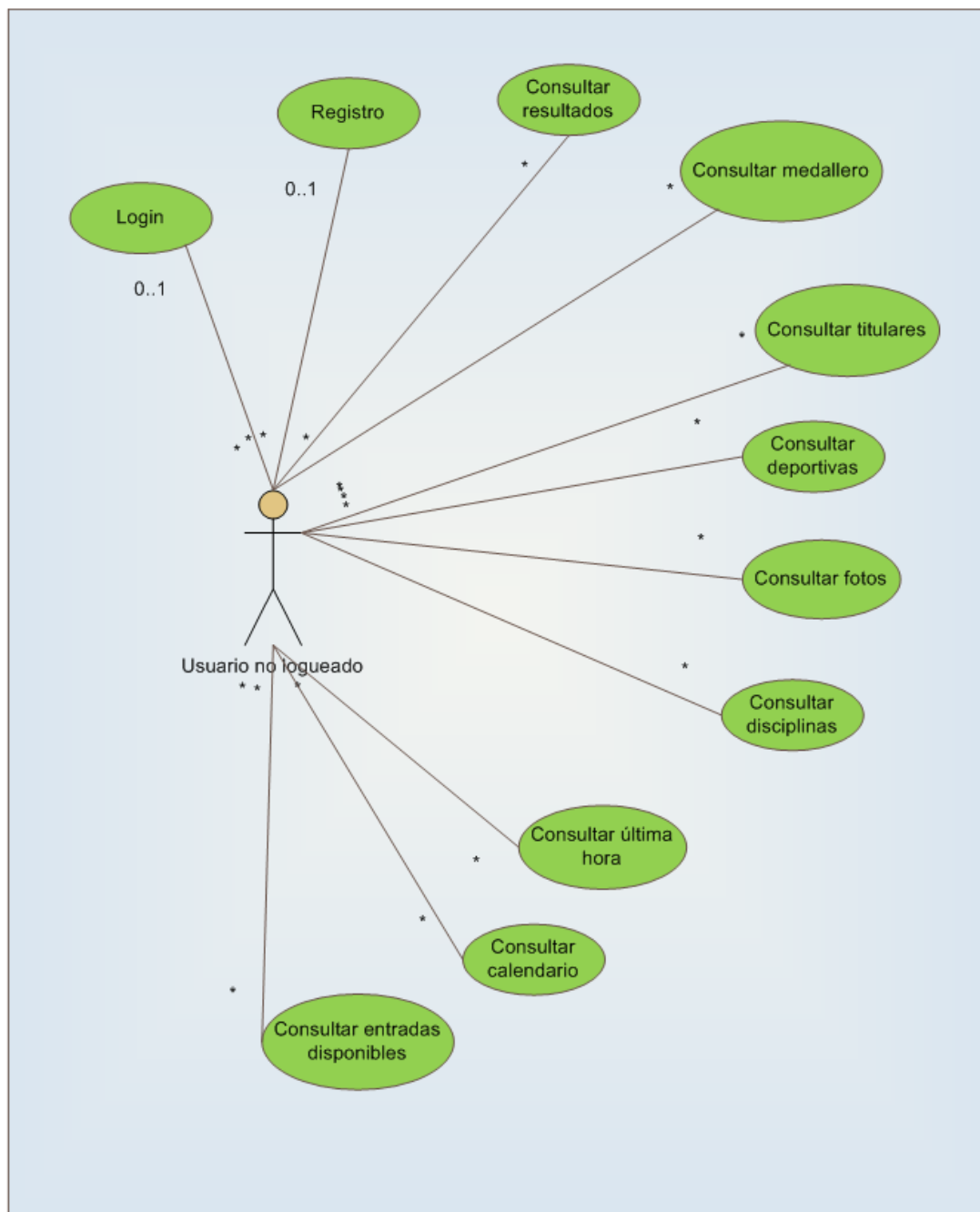
28	Realización de las mejoras especificadas	14 días	lun 26/04/10	jue 13/05/10
29	Determinación del equipo de desarrollo de la aplicación	10 días	lun 26/04/10	vie 07/05/10
30	Elaboración de la presentación final	6 días	lun 26/04/10	lun 03/05/10
31	Presentación y entrega final de la especificación del proyecto	1 día	vie 14/05/10	vie 14/05/10



CASOS DE USO

A continuación se detallan los casos de uso correspondiente a cada rol de la aplicación. Toda la herencia entre usuarios se explica en la definición de roles.

CASOS DE USO DE USUARIO NO LOGUEADO



DESCRIPCIÓN DE LOS CASOS DE USO

CasoDeUso	
Nombre	Login
Actores	Usuario
Descripción	Cambiar su estado de usuario común a usuario logueado (o identificado), aumentando el número de operaciones disponibles
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Usuario registrado <i>Post-condiciones:</i> <ol style="list-style-type: none"> 1. Conversión de usuario común (o no identificado) a usuario logueado (o identificado) 2. Habilitar operaciones únicamente disponibles para usuarios identificados.
Objetivo	Identificación de un usuario ya registrado

CasoDeUso	
Nombre	Registro
Actores	Usuario
Descripción	Registro de una nueva cuenta de usuario, que permitirá al mismo identificarse en la aplicación web.
Restricciones	<i>Post-condiciones:</i> <ol style="list-style-type: none"> 1. Usuario identificado. 2. Habilitar operaciones únicamente disponibles para usuarios identificados.
Objetivo	Creación de una nueva cuenta de usuario

CasoDeUso	
Nombre	Consultar resultados
Actores	Usuario
Descripción	Visualización de los resultados de los eventos disputados, y actualizados en caso de que el acta correspondiente haya sido ampliada
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Haber seleccionado una disciplina y evento ya disputado. 2. Existencia de un acta y/o adjuntos referida al evento seleccionado.
Objetivo	Visualización de los resultados de los eventos disputados

CasoDeUso	
Nombre	Consultar titulares
Actores	Usuario
Descripción	Visualización de los titulares acerca de los eventos deportivos, suministrados por el comité organizador (Ayuntamiento y COI)
Restricciones	No aplicable
Objetivo	Visualización de los titulares acerca de los eventos deportivos

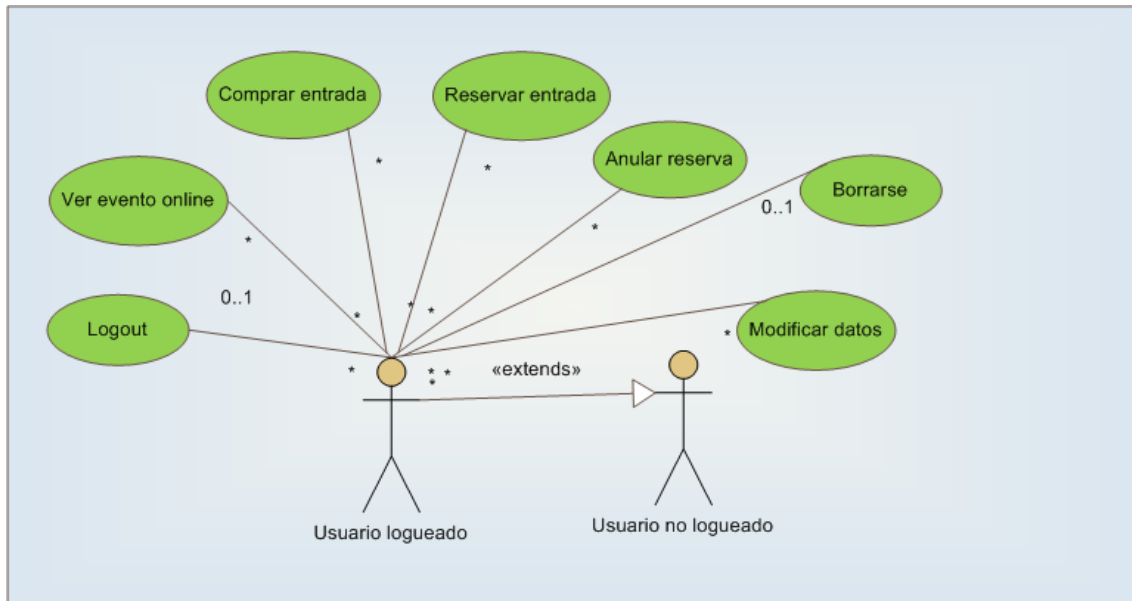
CasoDeUso	
Nombre	Consultar fotos
Actores	Usuario
Descripción	Visualización de las fotos de los eventos deportivos, suministradas por el comité organizador (Ayuntamiento y COI)
Restricciones	No aplicable
Objetivo	Visualización de las fotos de los eventos deportivos

CasoDeUso	
Nombre	Consultar disciplinas
Actores	Usuario
Descripción	Definición de las disciplinas existentes en los eventos, suministradas por el comité organizador (Ayuntamiento y COI)
Restricciones	No aplicable
Objetivo	Definición de las disciplinas existentes en los eventos

CasoDeUso	
Nombre	Consultar última hora
Actores	Usuario
Descripción	Visualización de las noticias de última hora referidas a eventos deportivos, suministradas por el comité organizador (Ayuntamiento y COI)
Restricciones	No aplicable
Objetivo	Visualización de las noticias de última hora referidas a eventos deportivos

CasoDeUso	
Nombre	Consultar calendario
Actores	Usuario
Descripción	Mostrar el calendario de eventos deportivos programados automáticamente por el cuadro de competencias de la aplicación y con posibles actualizaciones debido a modificaciones en los diferentes eventos
Restricciones	No aplicable
Objetivo	Mostrar el calendario de eventos deportivos programados

CASOS DE USO DE USUARIO LOGUEADO



DESCRIPCIÓN DE LOS CASOS DE USO

CasoDeUso	
Nombre	Borrarse
Actores	Usuario logueado
Descripción	Eliminar su cuenta como usuario de la aplicación
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Usuario registrado e identificado 2. Sus pagos han sido realizados satisfactoriamente
Objetivo	Eliminar su cuenta

CasoDeUso	
Nombre	Modificar datos
Actores	Usuario logueado
Descripción	Modificar los datos de su cuenta de usuario de la aplicación
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Usuario registrado e identificado
Objetivo	Modificar los datos de su cuenta

CasoDeUso	
Nombre	Anular reserva
Actores	Usuario logueado
Descripción	Cancelar la reserva de un evento seleccionado
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. La fecha del evento es posterior a la actual. 2. La fecha de reserva no ha superado el límite máximo tiempo de reserva. <i>Post-condiciones:</i> <ol style="list-style-type: none"> 1. Aumentar el número de entradas disponibles para un evento, en relación a la anulación.
Objetivo	Cancelar la reserva de un evento

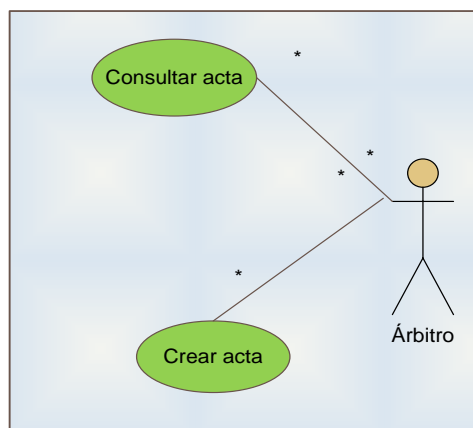
CasoDeUso	
Nombre	Comprar entrada
Actores	Usuario logueado
Descripción	Compra de entradas para un evento determinado
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. La fecha del evento es posterior a la fecha actual. 2. La fecha de reserva no ha superado el límite máximo tiempo de reserva. <i>Post-condiciones:</i> <ol style="list-style-type: none"> 1. Aumentar el número de entradas vendidas para el evento concreto. 2. Realizar el cobro mediante el servicio contratado
Objetivo	Comprar una reserva

CasoDeUso	
Nombre	Reservar entrada
Actores	Usuario logueado
Descripción	Reserva de un evento concreto para su posible futura compra
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Existe una o más entradas disponibles para el evento concreto. <i>Post-condiciones:</i> <ol style="list-style-type: none"> 1. Disminuir el número de entradas disponibles para el evento concreto
Objetivo	Reserva de entrada

CasoDeUso	
Nombre	Consultar entradas disponibles
Actores	Usuario logueado
Descripción	Mostrar las entradas disponibles, y sus características, de un evento seleccionado, de una disciplina específica.
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Haber seleccionado una disciplina y un evento
Objetivo	Mostrar las entradas disponibles de un evento seleccionado

CasoDeUso	
Nombre	Logout
Actores	Usuario logueado
Descripción	Cambiar su estado de usuario logueado a usuario común, restringiendo el número de operaciones permitidas
Restricciones	<i>Post-Condiciones:</i> <ol style="list-style-type: none"> 1. Convertir en usuario común (o no identificado) 2. Deshabilitar operaciones únicamente disponibles para usuarios identificados
Objetivo	Dejar de estar identificado en la aplicación web

CASOS DE USO DE ÁRBITRO

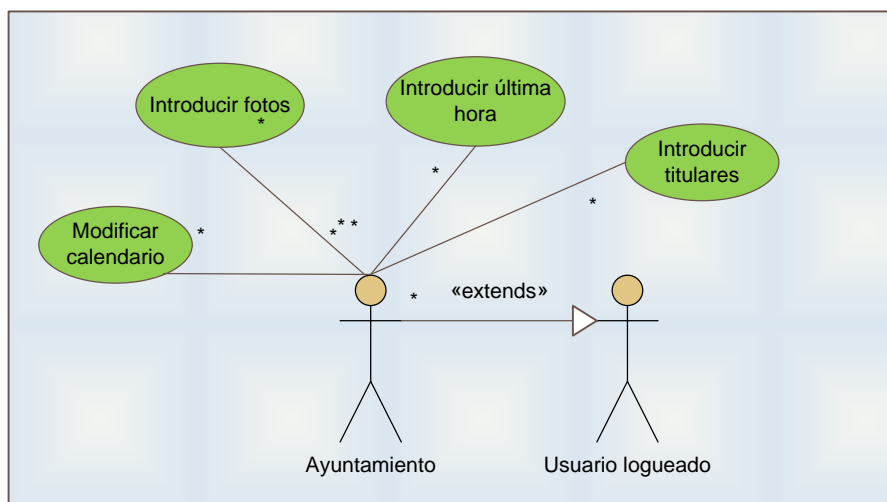


DESCRIPCIÓN DE LOS CASOS DE USO

CasoDeUso	
Nombre	Crear acta
Actores	Árbitro
Descripción	Genera un acta perteneciente a un evento concreto, en el que ha participado el árbitro que la crea; y que contiene todos los hechos deportivos sucedidos durante el evento así como el resultado del mismo.
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. El evento al que se refiere el acta ha sido disputado con anterioridad 2. El árbitro, que genera el acta, ha participado en dicho evento <i>Post-condiciones:</i> <ol style="list-style-type: none"> 1. Acta creada (no modificable)
Objetivo	Genera una nueva acta sobre un evento sucedido

CasoDeUso	
Nombre	Consultar acta
Actores	Árbitro
Descripción	Muestra el contenido de un acta perteneciente a un evento concreto y sucedido con anterioridad.
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Debe existir un acta relacionado con el evento concreto
Objetivo	Consulta el contenido de un acta sobre un evento sucedido

CASOS DE USO DE AYUNTAMIENTO



DESCRIPCIÓN DE LOS CASOS DE USO

CasoDeUso	
Nombre	Introducir fotos
Actores	Ayuntamiento
Descripción	Agregar a la interfaz web fotos concernientes a los eventos deportivos.
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Foto relativa a los eventos deportivos. 2. Foto no existente.
Objetivo	Aumentar el número de fotos existentes en la aplicación web.

CasoDeUso	
Nombre	Introducir ultima hora
Actores	Ayuntamiento
Descripción	Agregar, en la interfaz web, noticias de última hora relacionadas con los eventos deportivos gestionados
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Titular concerniente a los eventos deportivos 2. Titular no existente.
Objetivo	Nuevo titular de última hora en la aplicación

CasoDeUso	
Nombre	Introducir titulares
Actores	Ayuntamiento
Descripción	Agregar, en la interfaz web, las noticias relacionadas con los eventos deportivos gestionados
Restricciones	<i>Pre-condiciones:</i> <ol style="list-style-type: none"> 1. Noticia concerniente a los eventos deportivos 2. Noticia no existente
Objetivo	Nueva noticia en la aplicación

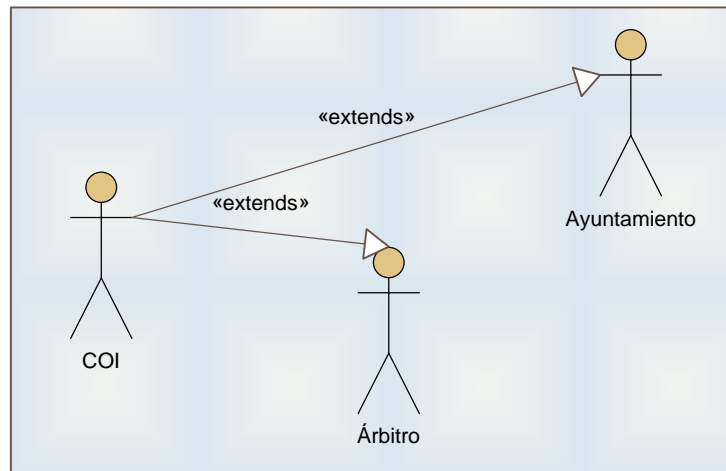
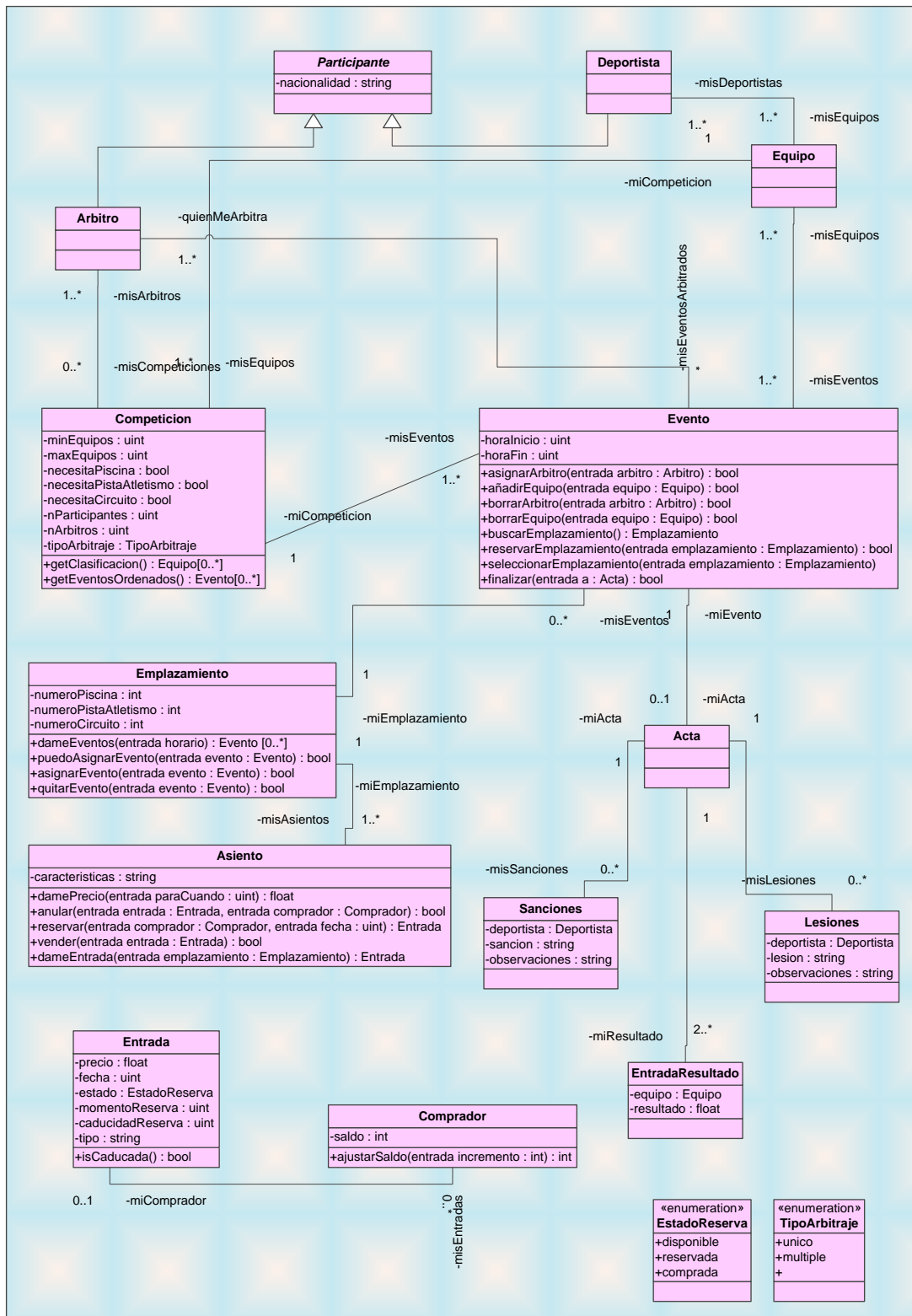


DIAGRAMA DE CLASES



OBSERVACIONES

Los equipos están formados por uno o más deportistas, todos de la misma nacionalidad. Y son los equipos los que participan en las diferentes competiciones.

El emplazamiento, lugar físico donde tendrá lugar el evento, posee información sobre sus instalaciones, para la posterior asignación. En función de esas instalaciones se podrá asignar el evento cuya competición tenga esos requerimientos.

Cada emplazamiento dispone de unos asientos para los espectadores en función de su aforo. Cada asiento puede tener asignada o no una entrada para asistir al evento.

Desde la clase evento, se gestionan todos los detalles necesarios para la organización del mismo. Para cada competición se determinan los eventos que van a tener lugar para ultimar la misma. Desde esta misma clase se gestiona la búsqueda de un emplazamiento, que es donde tendrá lugar, atendiendo a las necesidades de la competición. Una vez encontrado un emplazamiento acorde a las restricciones de la competición, será necesario reservarlo, para la posterior venta de entradas.

Para que el evento tenga lugar, será también necesario determinar los equipos que competirán por alcanzar el triunfo y los árbitros que lo regularán.

Para poder asignar a los árbitros al evento, será necesario que su nacionalidad no coincida con la de ninguno de los equipos participantes.

El número de equipos y árbitros que tendrá el evento vendrá impuesto por su competición.

Tras la finalización del evento, el árbitro o árbitros serán los encargados de elaborar un acta. Dicho acta está formada por al menos dos Entradas Resultado, cada una de las cuales posee el resultado de cada equipo que ha participado en el evento. El acta también reúne las posibles lesiones que hayan podido sufrir alguno de los deportistas de los equipos, así como las sanciones que se le hayan podido aplicar.

Para cada emplazamiento en el que tendrá lugar un evento, se ponen a la venta una serie de entradas hasta completar el aforo del emplazamiento. Una persona identificada como comprador puede reservar una o varias entradas que se encuentren en estado disponible. Tras lo cual, el comprador puede decidir proceder con la compra de las mismas.

La organización será la encargada de determinar el tiempo de caducidad de la reserva de las entradas. Si pasado este tiempo, no se ha procedido a la compra de la misma, la entrada pasará a estar disponible para una futura reserva.

RESTRICCIONES OCL

FUNCIONES AUXILIARES

```
Context Evento::solaparHorarios(tiempo: Integer): Boolean
derive: self.horaIni < tiempo and self.horaFin > tiempo
```

```
Context Evento::solapaCon(otro:Evento): Boolean
derive: result = self.solaparHorarios(otro.horaIni)
```

```
Context Emplazamiento::dameMaximosSolapantes(eventos : Collection(Evento)) : Integer
derive:
    result = eventos->asSet()->iterate(
        e1 : Evento      ;
        max : Integer = 0 |
        let
            newMax : Integer = eventos->select(e2 |
                e1.solapaCon(e2))
        in
            if(newMax > max)
                max = newMax
            else
                max = max
            endif
    )
```

```
Context Emplazamiento::damePiscinasLibres(e1 : Evento) :: Integer
derive:
    let eventos : Collection(Eventos) =
        self.misEventos->select(e2 |
            e2.solapaCon(e1)
            and
            e2.miCompeticion.requierePiscina)
    in
        result = numPiscinas -
            dameMaximosSolapantes(eventos)
```

```
Context Emplazamiento::damePistasAtletismoLibres(e1 : Evento) :: Integer
derive:
    let eventos : Collection(Eventos) =
        self.misEventos->select(e2 |
            e2.solapaCon(e1)                                and
            e2.miCompeticion.requierePistaAtletismo
        )
    in
        result = numPistasAtletismo - dameMaximosSolapantes(eventos)
```

```
Context Emplazamiento::dameCircuitosLibres(e1 : Evento) :: Integer
derive:
    let eventos : Collection(Eventos) =
        self.misEventos->select(e2 |
            e2.solapaCon(e1)                                and
            e2.miCompeticion.requiereCircuito
        )
    in
        result = numCircuitos - dameMaximosSolapantes(eventos)
```

INVARIANTES

- El número de jugadores de un equipo estará comprendido entre un mínimo y un máximo

```
context Equipo inv:
  self.misDeportistas->size() >= self.miCompeticion.minJugadoresPorEquipo and
  self.misDeportistas->size() <= self.miCompeticion.maxJugadoresPorEquipo
```

- Todos los miembros de un equipo tienen que ser de la misma nacionalidad

```
Context Equipo inv:
  Self.misDeportistas->forall(d1,d2 | d1.nacionalidad = d2.nacionalidad)
```

- Si una competición tiene un único árbitro la nacionalidad del mismo debe ser distinta de la de los equipos participantes

```
Context Evento inv:
  let nacionArbitro : Collection(String) = self.misArbitros.nacionalidad in
  Self.miCompeticion.nArbitrosPorEvento = 1 implies
  self.misEquipos->allInstances(e | e.misDeportistas.nacionalidad-
  >intersection(nacionArbitro)->isEmpty())
```

- Si una competición tiene varios árbitros, éstos no pueden compartir nacionalidad entre ellos (esto implica que para cada equipo, a lo sumo solo hay un árbitro con la misma nacionalidad)

```
Context Evento inv:
  Self.miCompeticion.tipoArbitraje = self.nArbitros > 1 implies
  self.misArbitros->forall(a1,a2 | a1.nacionalidad <> a2.nacionalidad)
```

- Los equipos de un evento tienen que estar inscritos en la competición a la que pertenece el evento

```
Context Evento inv:
  Self.misEquipos->forall(e | e.miCompeticion = self.miCompeticion)
```

- Los árbitros de un evento tienen que pertenecer a la competición a la que pertenece el evento

```
Context Evento inv:
  Self.misArbitros->forall(a | a.misCompeticiones->includes(self.miCompeticion))
```

- Un mismo equipo no puede estar en dos eventos que se solapan

```
Context Equipo inv:
  Self.misEventos->forall(e1, e2 | e1.solapaCon(e2) = false)
```

- Un mismo árbitro no puede estar en dos eventos que se solapan

```
Context Arbitros inv:
  Self.misEventos->forAll(e1, e2 | e1.solapaCon(e2) = false)
```

- Un jugador no puede estar en dos equipos con eventos que se solapen

```
Context Deportista inv:
  Self.misEquipos->forAll(eq1,eq2|eq1.misEventos->forAll(ev1 | eq2.misEventos-
>forAll(ev2 | ev1.solapaCon(ev2) = false)))
```

- El emplazamiento de todo evento cumple las restricciones de la competición

```
Context Evento inv:
  Self.miCompeticion.necesitaPiscina implies
    self.miEmplazamiento.numeroPiscina > 0
Context Evento inv:
  Self.miCompeticion.necesitaPistaAtletismo implies
    self.miEmplazamiento.numeroPistaAtletismo > 0
Context Evento inv:
  Self.miCompeticion.necesitaCircuito implies
    self.miEmplazamiento.numeroCircuito > 0
```

- Dado un emplazamiento no hay mas eventos asignados que necesiten recurso que recursos tiene el emplazamiento

```
Context evento inv:
  self.necesitaPiscina implies self.miEmplazamiento.misEventos->select(e|
    self.solapaCon(e) and
    e.necesitaPiscina
  )->size() <= self.miEmplazamiento.numeroPiscinas

Context evento inv:
  self.necesitaPistaAtletismo implies self.miEmplazamiento.misEventos-
>select(e|
    self.solapaCon(e) and
    e.necesitaPistaAtletismo
  )->size() <= self.miEmplazamiento.numeroPistaAtletismo

Context evento inv:
  self.necesitaCircuito implies self.miEmplazamiento.misEventos->select(e|
    self.solapaCon(e) and
    e.necesitaCircuito
  )->size() <= self.miEmplazamiento.numeroCircuito

Context Entrada
  inv entradaTieneEvento:
    self.asiento.miEmplazamiento.misEventos->exist(e|
      e.solapaCon(self.instante))

  inv entradaDisponibleSiYSoloSiNoTieneComprador
    self.estado = EstadoReserva::disponible implies self.miComprador->isEmpty()
    and
    self.miComprador->isEmpty() implies self.estado = EstadoReserva::disponible
```

- Ningún asiento tiene dos reservas en el mismo momento

```
Context Asiento inv:
    self.misReservas->forall(r1,r2|r1 <> r2 and r1.instante <>r2.instante)
```

- Para cada evento hay tantos equipos participantes como requiere la competición

```
Context Evento inv:
    self.misEquipos->size() = self.miCompeticion.nEquiposPorEvento
```

- Para cada evento hay tantos árbitros como requiere la competición

```
Context Evento inv:
    self.misArbitros->size() = self.miCompeticion.nArbitrosPorEvento
```

- Un evento está finalizado si y solo si tiene un acta asignada

```
Context Evento inv:
    self.miActa->notEmpty() implies self.isFinalizado() and
    self.isFinalizado() implies self.miActa->notEmpty()
```

- El saldo de un comprador no puede ser negativo

```
Context Comprador inv:
    self.saldo >= 0
```

- Los equipos que aparecen en los resultados de un acta son los mismos que los que los equipos del evento de acta

```
Context Acta inv:
    self.miEvento->notEmpty() implies self.resultado.equipos =
    self.miEvento.misEquipos
```

- Las lesiones que aparecen en un acta pueden hablar de un deportista o de un árbitro del evento

```
Context Lesión inv:
    self.miEvento->notEmpty() implies (
        self.miActa.miEvento.misEquipos.misDeportistas->includes(self.persona) or
        self.miActa.miEvento.misArbitros->includes(self.persona)
    )
```

- Las sanciones que aparecen en un acta pueden hablar de un deportista del evento

```
Context Sanción inv:
    self.miEvento->notEmpty() implies (
        self.miActa.miEvento.misEquipos.misDeportistas->includes(self.deportista)
    )
```

- Las competiciones de dos equipos por evento tienen una final y una de consolación donde se juegan el tercer y cuarto puesto. En estos dos eventos los equipos participantes han de ser distintos

```

Context Competicion inv
  let
    final      : Evento = self.getEventosOrdenados()->last(),
    consolacion : Evento = self.getEventosOrdenados()-
  >at(self.getEventosOrdenados()->size()-2)
  in
    Competicion.nEquiposPorEvento = 2 implies
      final.misEquipos->intersection(consolacion->misEquipos)->isEmpty()

```

PRE Y POST CONDICIONES

```

context Asiento::reservar(c : Comprador, fecha : Integer) : Entrada
pre:
  self.misEntradas->select(e | a.fecha = fecha)->isEmpty()
  precio >= 0
post:
  result.oclIsNew()                                and
  result.asiento      = self                        and
  result.fecha        = fecha                      and
  result.precio       = damePrecio(fecha)          and
  result.estado       = EstadoReserva::reservada   and
  result.miComprador->notEmpty()                   and
  self.misEntradas->includes(result)

```

```

context Asiento::vender(e: Entrada, c:Comprador) : Boolean
post:
  if
    e.estado@pre = EstadoReserva::reservada      and
    c = e.miComprador                            and
    c.saldo@pre > e.precio                       and
    self.includes(e)
  then
    result      = true                          and
    e.estado    = EstadoReserva::comprada        and
    c^ajustarSaldo(-e.precio)
  else
    result      = false
  endif

```

```

Context Asiento::anular(e: Entrada, c : Comprador) : Boolean
post:
  if
    e.estado@pre = EstadoReserva::reservada      and
    c = e.miComprador                            and
    self.includes(e)
  then
    result      = true                          and
    e.estado    = EstadoReserva::disponible      and
    e.miComprador->isEmpty()                     and
    self.misEntradas->excludes(e)
  else
    result      = false
  endif

```

```

context Comprador::ajustarSaldo(incremento :: Integer) : Integer
pre:
    self.saldo + incremento >= 0
post:
    self.saldo = self.saldo@pre + incremento

```

```

context Entrada::isCaducada() : Boolean
body:
    if self.estado = EstadoEntrada::reservada
    then time() < momentoReserva + caducidadReserva
    else false
    endif

```

```

Context Emplazamiento::quitarEvento(e : Evento) : Boolean
post:
    self.misEventos = self.misEventos@pre->excluding(e)

```

```

Context Emplazamiento::puedoAsignarEvento(e : Evento) : Boolean
post:
    if
        self.misEventos->includes(e)
    then
        result = true
    else
        result =
            e.miCompeticion.requierePiscinas implies
                self.damePiscinasLibres(e) > 1 and
            e.miCompeticion.requierePiscinas implies
                self.damePistasAtletismoLibres(e) > 1 and
            e.miCompeticion.requierePiscinas implies
                self.dameCircuitosLibres(e) > 1
    endif

```

```

Context Emplazamiento::asignarEvento(e : Evento) : Boolean
post:
    if self.puedoAsignarEvento(e)
    then
        self.misEventos = self.misEventos@pre->including(e)
        result = true
    else
        result = false
    endif

```

```

Context Evento::reservarEmplazamiento(e : Emplazamiento) : Boolean
post:
    if e = self.miEmplazamiento@pre then result = true
    else
        if e.puedoAsignarEvento(self)
        then
            (self.miEmplazamiento@pre->notEmpty() implies
                self.miEmplazamiento@pre^quitarEvento(self)) and
            self.miEmplazamiento = e and
            e^asignarEvento(self) and
            result = true
        else
            result = false
        endif
    endif
endif

```

```

Context Acta::getResultado(e:Equipo) : Integer
pre:
  self.miEvento.misEquipos->includes(e)
post:
  result = self.resultado->at(self.miEvento.misEquipos->indexOf(e))

```

```

Context Evento::getClasificacion() : OrderedSet(Equipo)
pre:
  self.finalizado()
post:
  if self.miCompeticion.ganaElMayorResultado
  then
    result = self.miActa.resultado->sortedBy(resultado).equipo
  else
    result = self.miActa.resultado->sortedBy(-resultado).equipo
  endif

```

```

Context Competicion::getEventosOrdenados() : OrderedSet(Evento)
derive:
  self.misEventos->asOrderedSet()->sortBy(horaInicio)

```

```

Context Competicion::getMedallero() : OrderedSet(Equipo)
pre:
  self.misEventos->select(e | not e.finalizado())->isEmpty()
post:
  let eventosOrdenados : OrderedSet(Evento) = self.getEventosOrdenados() in
  if eventosOrdenados->last().getClasificacion()->size() >= 3 or misEquipos->size()
  < 3
  then
    eventosOrdenados->last().getClasificacion()
  else
    eventosOrdenados->last().getClasificacion().prepend(
      eventosOrdenados->at(eventosOrdenados->size()-2).getClasificacion()
      ->first())
  endif

```

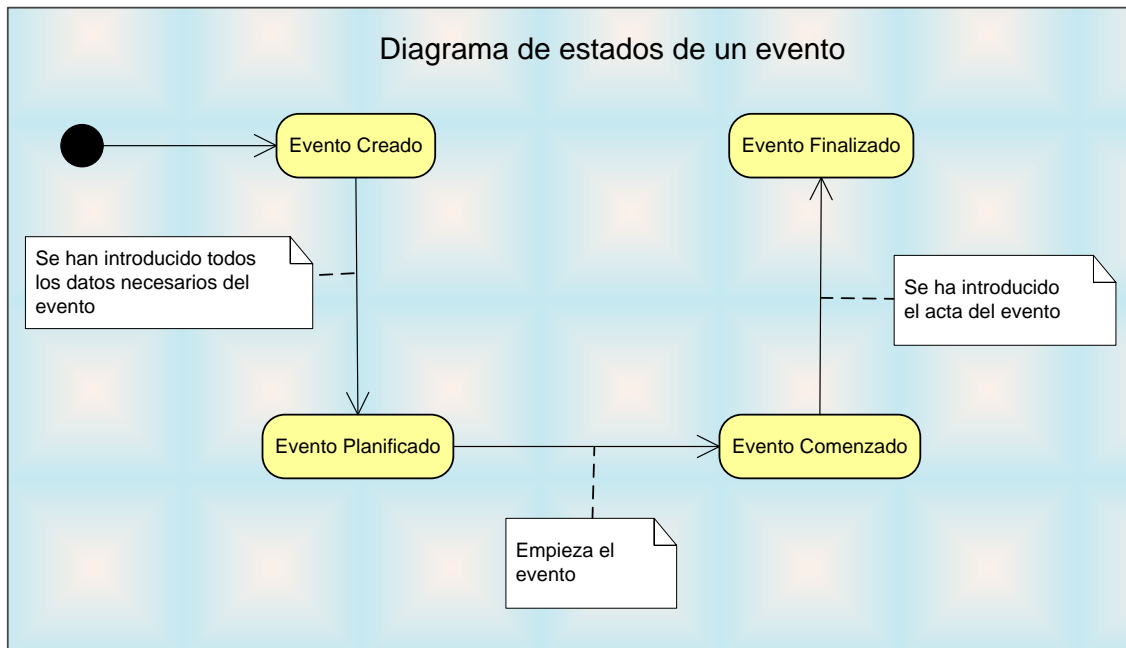
```

Context Evento::finalizar(a : Acta) : Boolean
post:
  if self.miActa->isEmpty() then
    self.miActa = a                                and
    a.miEvento = self                              and
    result = true                                   and
  else
    result = false
  endif

```

DIAGRAMAS DE ESTADO

EVENTOS

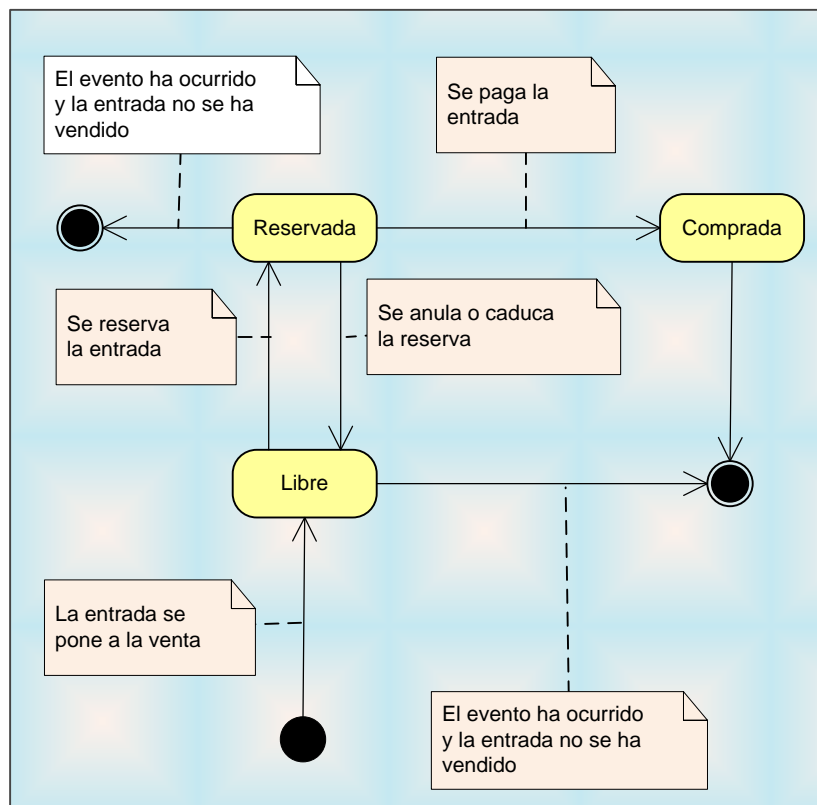


OBSERVACIONES

Representa todos los estados donde puede estar un evento.

- **Evento creado:** Estado inicial. Representa el hecho de que un evento ha sido dado de alta en el sistema, pero faltan datos. Este es el caso, por ejemplo, de una final de una competición, en la cual en un principio se conoce el lugar y la hora del evento pero no sus participantes. Este estado transita a "Evento planificado" en el momento en que se han introducido todos los datos necesarios para la ejecución del evento.
- **Evento Planificado:** Representa al evento que tiene todos los datos necesarios para celebrarse. Los datos de dicho evento pueden modificarse pero no eliminarse, por lo tanto siempre estará disponible para comenzar. Transita a "Evento Comenzado" cuando llega la hora planificada para el evento.
- **Evento Comenzado:** Representa el estado en que el evento se está celebrando físicamente pero aún no ha sido introducido el acta. Cuando esto ocurre, el estado transita a "Evento Finalizado".
- **Evento Finalizado:** Representa el momento en que el acta ha sido introducida en el sistema. En este momento no se puede realizar ninguna modificación en los datos del evento. Este estado no transita a ningún otro, es decir, es un estado final.

ENTRADAS



OBSERVACIONES

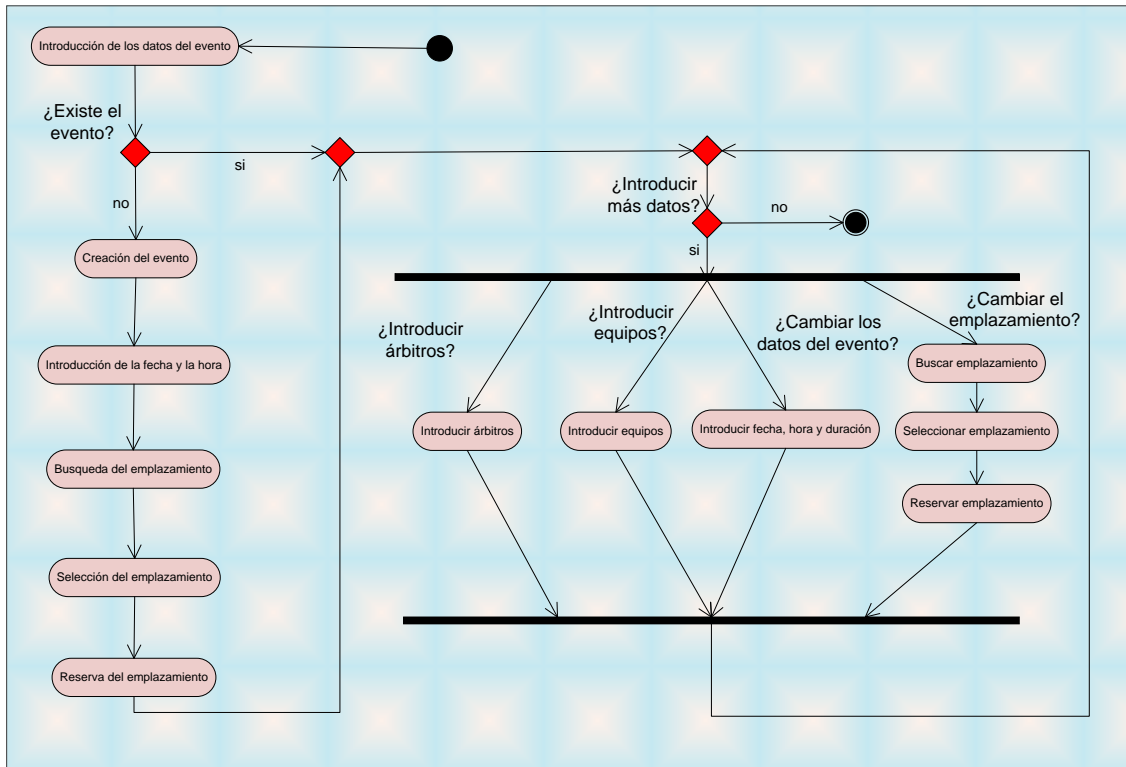
Los tres estados entre los que puede transitar una entrada son:

- Libre: Representa el estado inicial y al que se transita después de que una entrada haya caducado desde el estado de "Reservada". Desde este estado se puede transitar a "Reservada" en caso de producirse una reserva.
- Reservada: Cuando un usuario ha manifestado frente al sistema su intención de comprar una entrada pero aún no ha sido pagada. Se transita a este estado desde el estado de "Libre" al realizar la reserva y de él se puede transitar a "Libre" en caso de que la reserva caduque o "Comprada" en caso de que realice la compra.
- Comprada: Este estado representa a una entrada que ha sido adquirida por un usuario y se transita a este estado desde el estado "Reservada". Este estado no tiene transiciones posibles.

En caso de celebrarse el evento la entrada pasa a ser eliminada del sistema.

DIAGRAMAS DE ACTIVIDAD

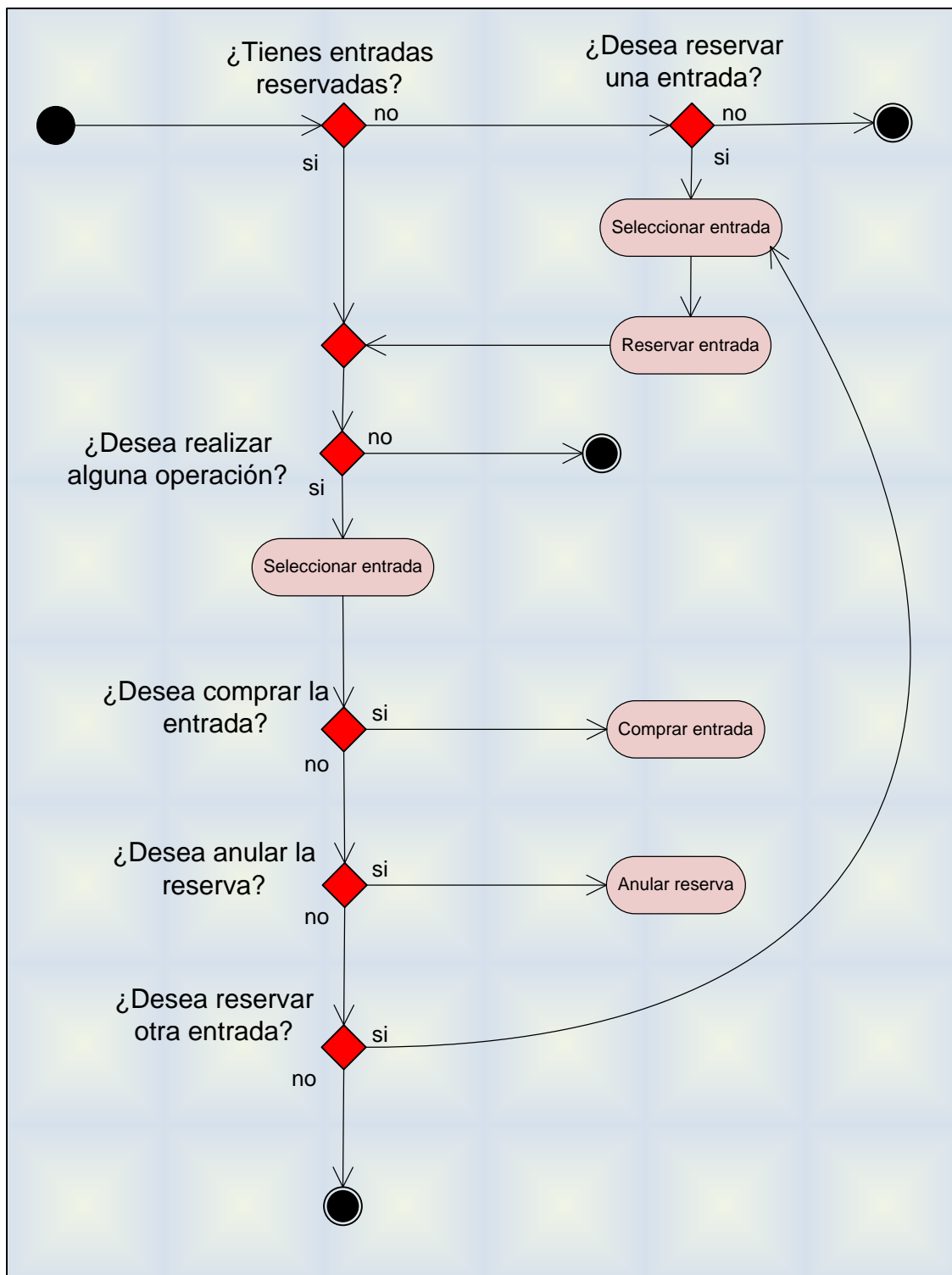
EVENTOS



OBSERVACIONES

Cuando creas un evento es obligatorio introducir una serie de datos obligatorios, tales como la fecha, emplazamiento y reserva de dicho emplazamiento. Una vez introducido estos datos, mientras el evento no se encuentre comenzado o finalizado, pueden modificarse los equipos, árbitros, emplazamiento y otros datos del evento. Estas modificaciones pueden hacerse de manera individual cada vez que se quiera modificar un evento.

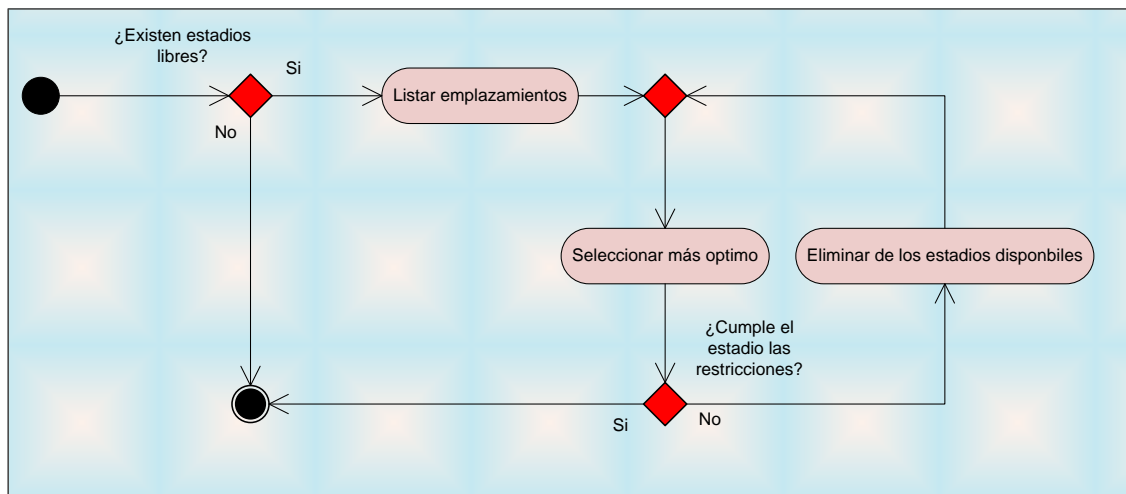
No existe la posibilidad de celebrar un evento sin que todos los datos estén introducidos.



OBSERVACIONES

Una entrada reservada no puede modificarse, puede anularse y reservar una entrada nueva.

BUSCAR EMPLAZAMIENTO

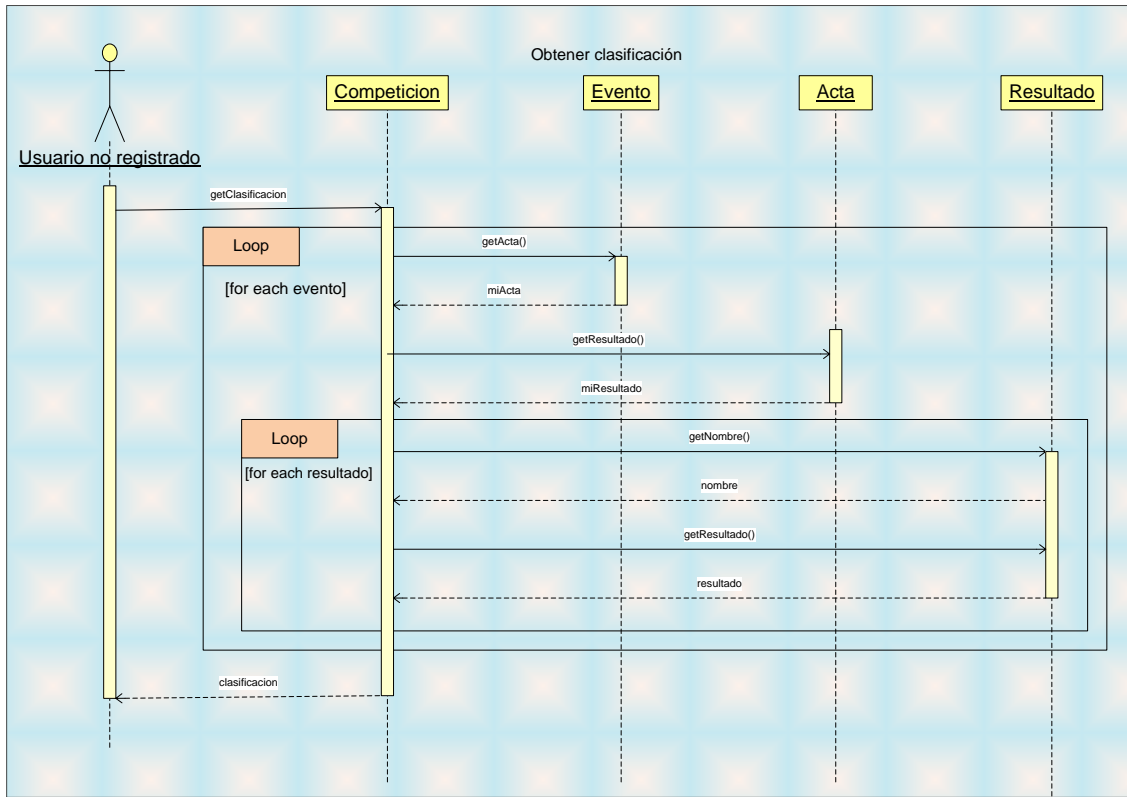


OBSERVACIONES

Para buscar un emplazamiento listamos todos los emplazamientos libres, acto seguido buscamos el más optimo los criterios de búsqueda son, el número de plazas, la calidad de las instalaciones y la comunicación. Una vez seleccionado el más optimo, si cumple las restricciones de disponibilidad, este emplazamiento es seleccionado, en caso contrario se elimina de la lista y se selecciona otro emplazamiento

DIAGRAMAS DE SECUENCIA

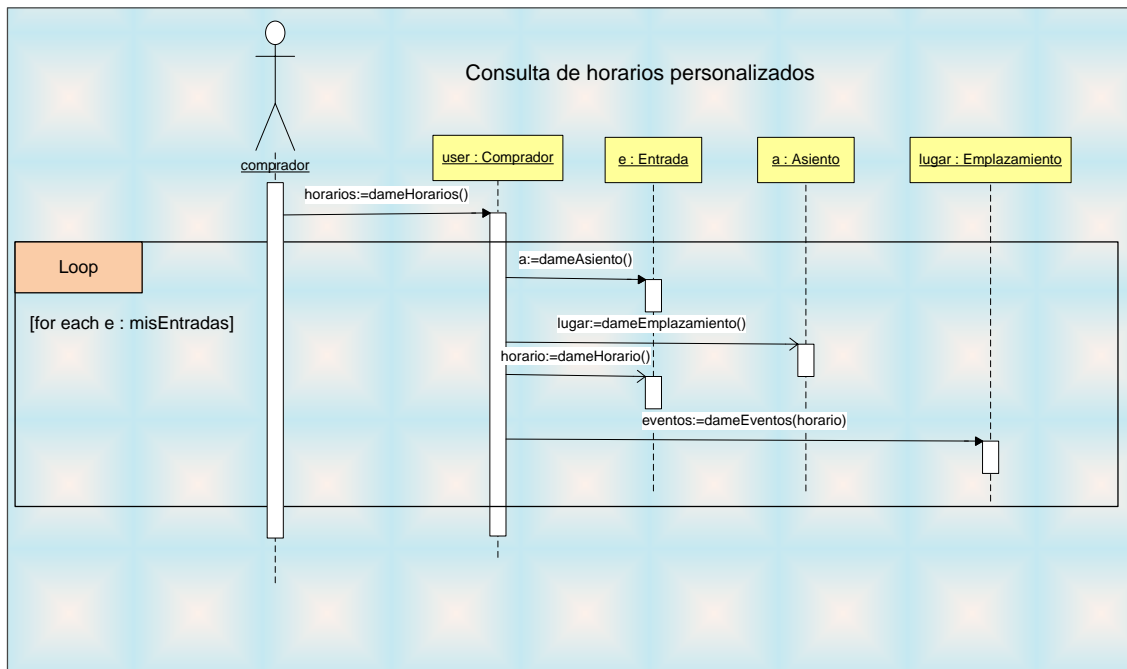
OBTENER CLASIFICACIÓN



OBSERVACIONES

Este diagrama como un usuario puede consultar el estado de una competición. Esto incluye resultados de partidos anteriores, próximos eventos, actas de los equipos etc.

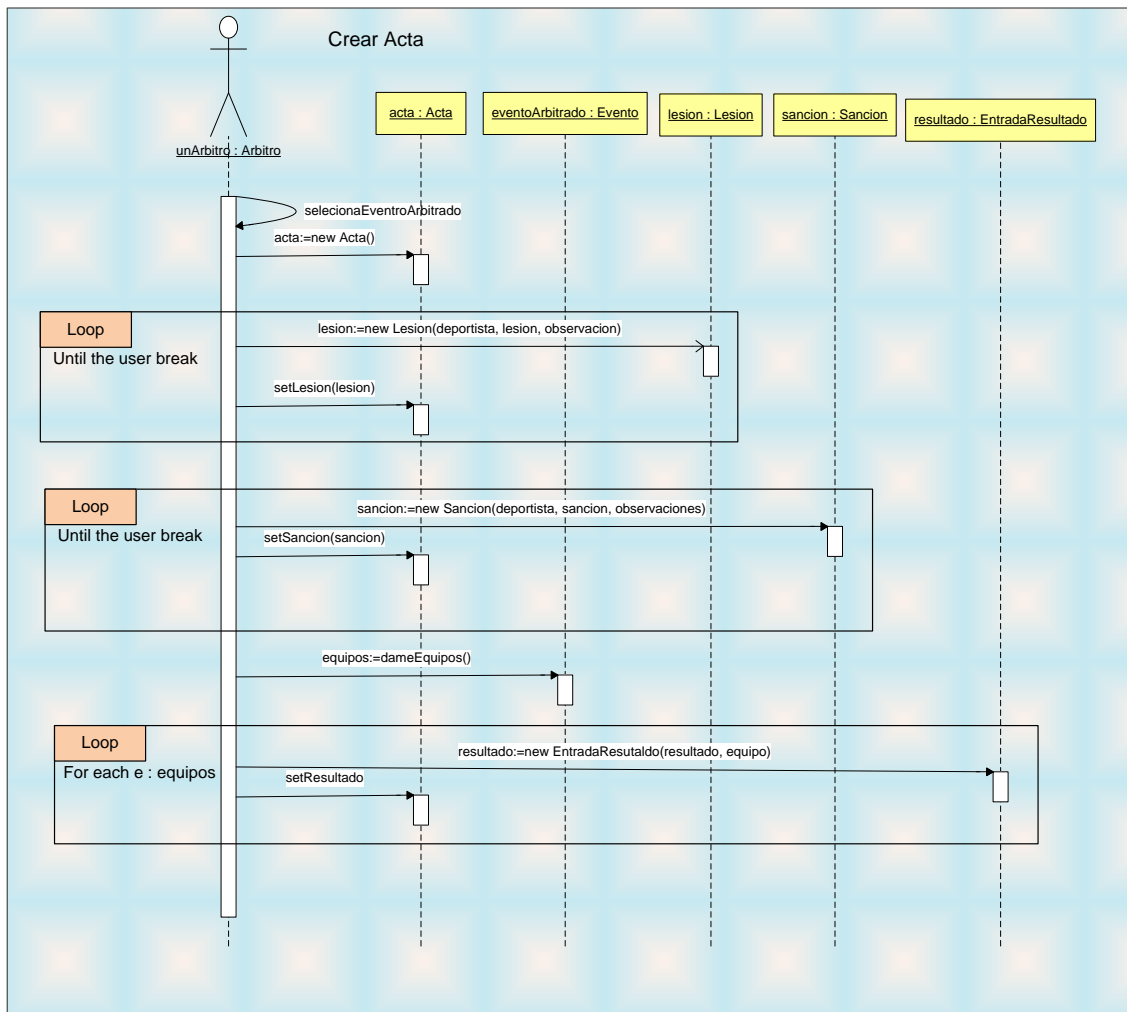
CONSULTAR HORARIOS PERSONALIZADOS



OBSERVACIONES

Este diagrama de secuencia representa cómo se genera un horario personalizado dependiendo de las entradas. El usuario solicita los horarios que le serán representados en la interfaz web. Esto se realiza con el objeto comprador, que será el que consultando todas las entradas y sus respectivos asientos y emplazamientos, dará una lista detallada de todo el horario.

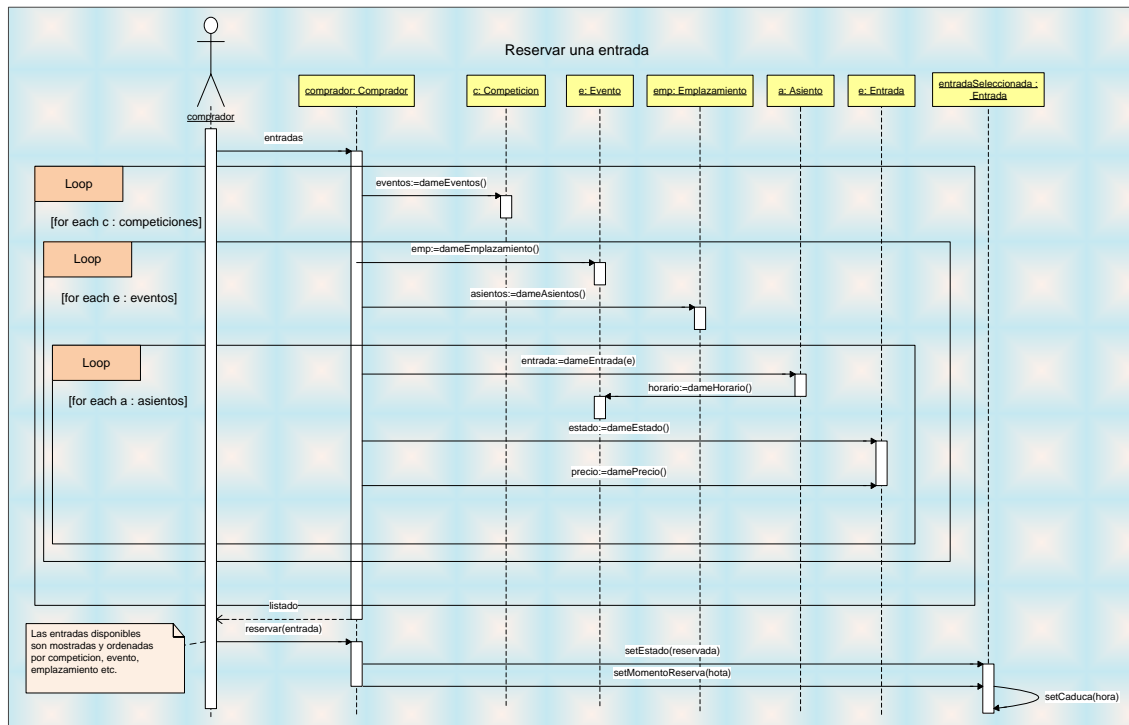
CREAR ACTA



OBSERVACIONES

Este diagrama de secuencia muestra como se crea un acta. Las actas una vez creadas no pueden ser modificadas y quedan almacenadas en el sistema. Al introducir el acta hay que especificar los atributos del acta y al almacenarla se actualizan, en caso de existir, los siguientes eventos de la competición.

RESERVAR ENTRADAS

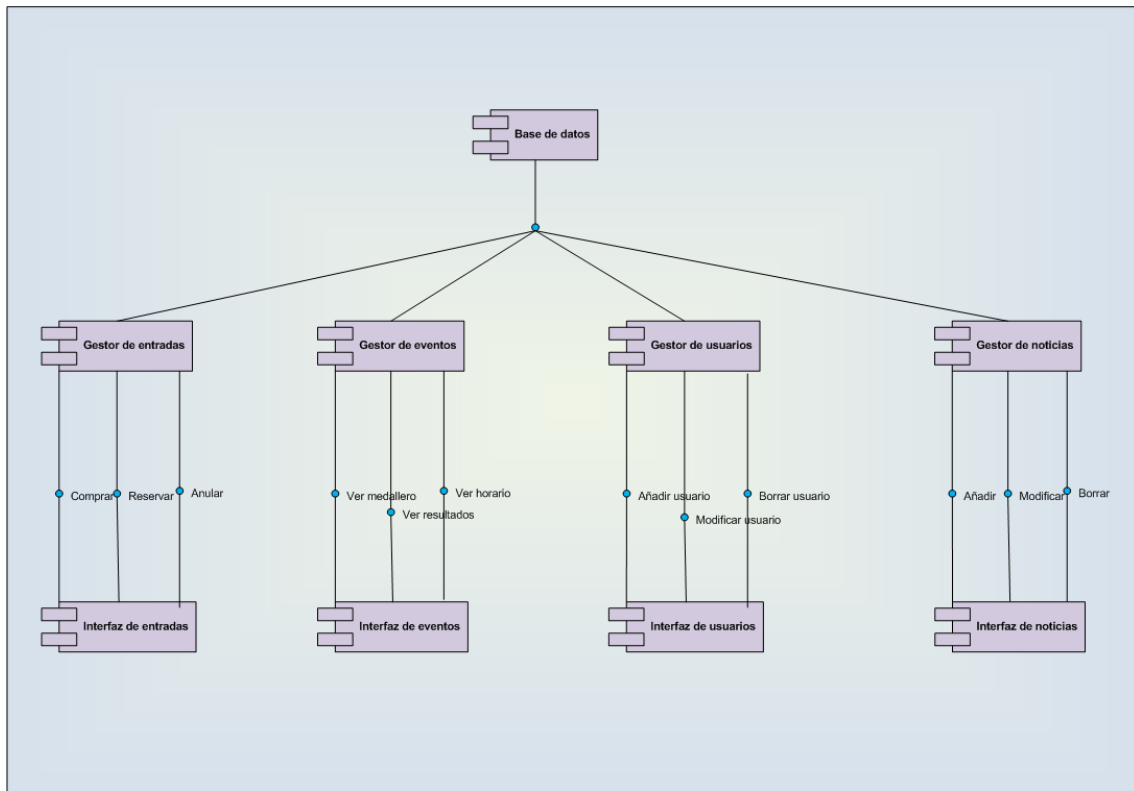


OBSERVACIONES

Para realizar una reserva lo primero que se hace es comprobar el estado de las entradas disponibles, haciendo un listado con todas las entradas disponibles y seleccionables. En el caso de existir entradas de varios tipos, si la entrada no se ajusta al perfil de comprador (ya sea porque no está disponible para él o porque ese tipo de entrada es específico para determinado tipo de personas) no aparecerán en el listado. Una vez realizada la petición de reserva, ésta es realizada.

DIAGRAMAS DE COMPONENTES

DIAGRAMAS DE COMPONENTES



BASE DE DATOS

- La base de datos de nuestra aplicación constará de varias tablas para los diferentes usos del sistema. Habrá tablas para el registro de los usuarios, para la venta de entradas, equipos participantes, instalaciones, etc.
- Alguna de esas tablas nos la enviará el COI (como la tabla equipos y la tabla competiciones) y en nuestra aplicación solo se podrán utilizar, nunca modificar. Otras tablas nos la dará el ayuntamiento de la ciudad organizadora (tabla instalaciones) y al igual que antes, nuestra aplicación no modificara esa tabla.
- El resto de tablas son las que mediante el sistema de gestión de bases de datos, abajo explicado, y los gestores harán que nuestra aplicación tenga una funcionalidad completa.
- En el apartado relativo a modelo de datos se explicara más detalladamente el formato de cada una de las tablas

SISTEMA DE GESTION DE BASE DE DATOS

- Es el componente que nos ayuda a gestionar la base de datos de la aplicación, el espacio físico y nos proporciona unas interfaces de acceso primitivas. Es el puente entre los gestores y las tablas de la base de datos.
- La interfaz consta de métodos de acceso a tablas (select), de creación de tablas (create), de borrado de tablas (drop) , borrado de información de una tabla (delete), modificación de información contenida en tablas (update) y de las propias tablas (alter) , y por ultimo un método para insertar información en las bases de datos (insert).

GESTORES

GESTOR DE ENTRADAS

- Es el componente a través del cual vamos a gestionar las entradas de los espectadores a las distintas disciplinas olímpicas.
- Va a ser el que va a recibir las ordenes de los compradores a través de la interfaz de entradas y el que va a tener que utilizar el SGDB para acceder y modificar las tablas. Gracias a él, los datos que los usuarios puedan introducir, serán coherentes con las tablas de la base de datos relativas a las entradas.
- Las operaciones que realizara sobre las tablas de la base de datos son:
 - Reservar entrada : añadirá una fila a la tabla de entradas reservadas
 - Comprar entrada: añadirá una fila a la tabla de entradas vendidas si la entrada que se intenta comprar pertenece a la tabla entradas reservadas y en caso afirmativo borrará la entrada de la tabla entradas reservadas.
 - Anular entrada : eliminará una fila de la tabla de entradas reservas

GESTOR DE USUARIOS

- Es el componente a través del cual vamos a gestionar los usuarios que van a poder disfrutar de las ventajas de esta aplicación.
- A través del formulario de registrarse de la interfaz el gestor va a recibir todos los datos necesarios para que el usuario sea registrado en la aplicación. A partir de ahí el gestor hará las operaciones necesarias para que en la tabla de usuario queden reflejados los cambios.
- Las operaciones que va a realizar el gestor son:
 - Registrar un usuario: añadir una fila a la tabla de usuarios.
 - Modificar usuario: modificar los datos en la fila de la tabla de usuarios.

- Eliminar usuario: eliminar la fila de la tabla de usuarios.
- Login: comprobar que los datos introducidos están en la tabla de usuarios.

GESTOR DE NOTICIAS, FOTOS

- Es el componente a través del cual vamos a gestionar las noticias y fotos que van aparecer en la herramienta.
- En la interfaz de noticias, el usuario podrá introducir el titular y el cuerpo de la noticia, o si lo desea, fotos. Las noticias y fotos se almacenaran en el servidor y el encargado será el gestor.
- Las operaciones que realiza son:
 - Añadir noticia
 - Modificar noticia
 - Añadir foto
 - Eliminar foto
 - Poner noticia en ultima hora
- En la parte relativa a componentes software se dará más información de este gestor y de cómo se va a integrar en nuestra aplicación

GESTOR DE EVENTO

- Es sin duda, el gestor más importante de nuestra aplicación a la vez que el más sofisticado.
- Utilizará las tablas de la base de datos de deportes, equipos e instalaciones que le vendrán dadas por el COI y el ayuntamiento para realizar un organigrama de toda la competición (en el diagrama de actividad Busca emplazamiento se detalla). Gracias a él se construirá el calendario completo de todos los eventos que se van a realizar en las olimpiadas y les podrá horario e instalaciones. También gestionará la tabla de resultados que después de cada partido se rellenará gracias a los actas y gracias a ello podremos visualizar inmediatamente las clasificaciones y el medallero. Por último el gestor tendrá operaciones más simples, pero no por ello menos importantes como pueden ser: modificar la información de un evento o modificar las actas de un partido.

GESTOR DE EVENTOS ONLINE

Es el componente a través del cual se va a poder ver las olimpiadas en directo en nuestra aplicación. En la interfaz de eventos online se podrá elegir que disciplina y evento deseas visualizar. Además podrás elegir la forma de pago.

Las operaciones que realiza son:

- Comprar evento online
- Ver evento Online

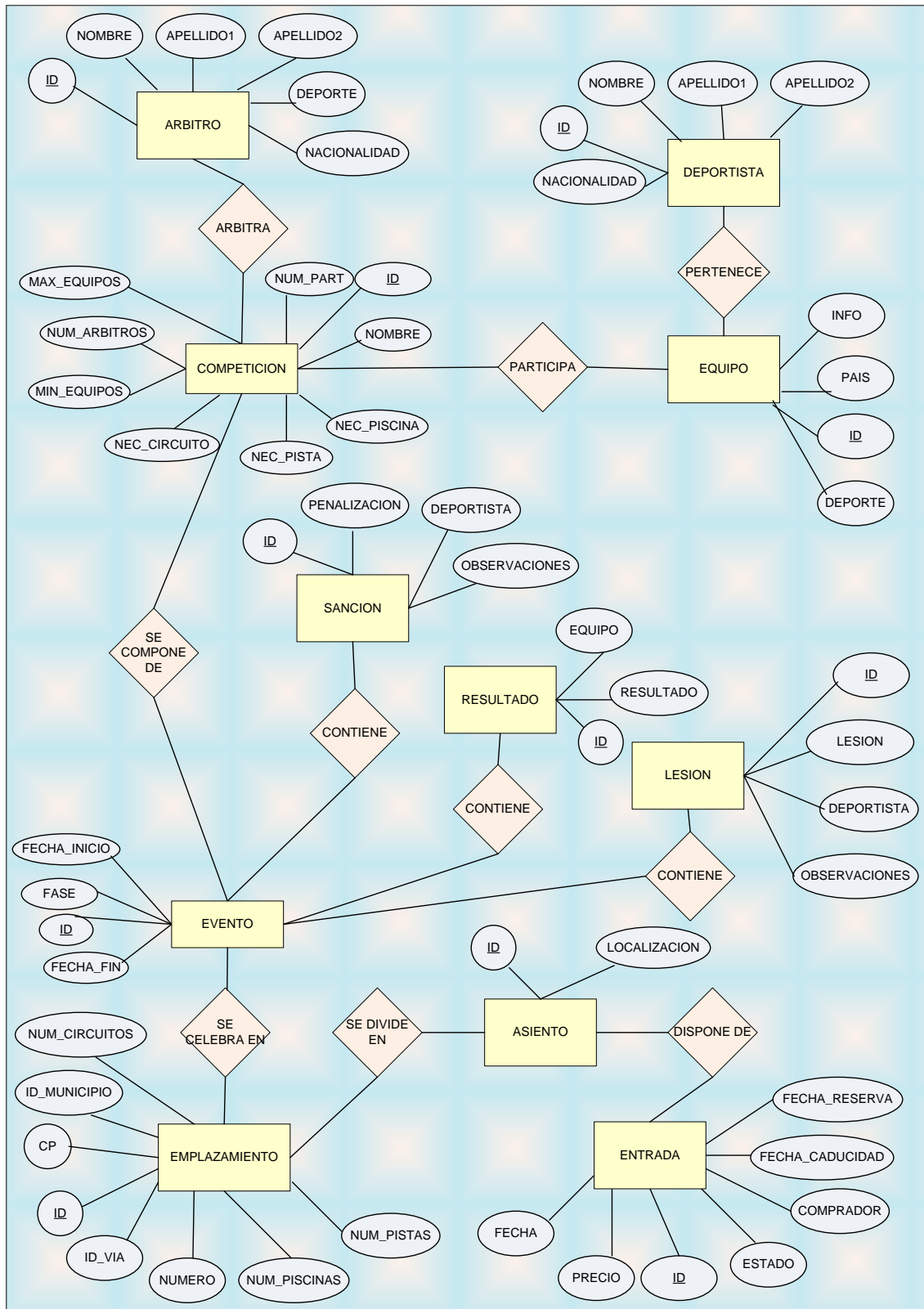
En la parte relativa a componentes software se dará más información de este gestor y como se va a integrar en nuestra aplicación.

INTERFACES

- Es el componente a través del cual el usuario va a poder acceder a todas las utilidades que ofrece el sistema, por lo tanto debe ser sencillo y muy intuitivo.
- Habrá una serie de interfaces, cada una la gestionará uno de los componentes arriba descritos.
- Más adelante se podrán ver con más detalle estas interfaces.

MODELO DE DATOS

DIAGRAMA ENTIDAD-RELACIÓN



MODELO RELACIONAL

Equipo(idEquipo, país, info, idCompeticion)

Deportista(idDeportista, nacionalidad, nombre, apellido1, apellido2)

Pertenece_Deportista(idEquipo, idDeportista)

Competicion(idCompeticion, nombre, numPart, numArbitros, maxEquipos, minEquipos, necPiscina, necPista, necCircuito)

Evento(idEvento, fase, fecha_inicio, fecha_fin)

Resultado(idResultado, idEquipo, resultado)

Lesion(idLesion, idEvento, lesión, idDeportista, observaciones)

Sancion(idSancion, idEvento, idDeportista, penalización, observaciones)

Evento_Resultado(idEvento, idResultado)

Arbitro(idArbitro, nombre, apellido1, apellido2, nación)

ArbitrosEvento(idArbitro, idEvento)

Emplazamiento(idEmplazamiento, idMunicipio, idVia, cp, numero, numCircuitos, numPistas, numPiscinas)

Lugar_Evento(idEmplazamiento, idEvento, fecha)

Asiento(idAsiento, localizador, idEmplazamiento)

Entrada(idEntrada, idAsiento, idEmplazamiento, fecha, estado, precio, comprador, fechaReserva, fechaCaducidad)

DESCRIPCIÓN DE LAS TABLAS

TABLA DE EQUIPO

- El COI nos facilitará los equipos que van a participar en los diferentes deportes olímpicos. Cada equipo será identificado por “idEquipo”, y llevara información como el deporte al que pertenece, la nacionalidad del equipo e información adicional del equipo, como puede ser el nombre de los participantes del equipo.

TABLA DE DEPORTISTA

- La tabla de deportistas, al igual que la anterior, nos vendrá dada por el COI. Los únicos atributos relevantes para nuestra aplicación son el nombre, los apellidos y la nacionalidad.

TABLA DE PERTENECE_DEPORTISTA

- Esta tabla se encarga de relacionar los equipos con los deportistas que van a formar cada equipo.

TABLA DE COMPETICION

- En esta tabla se alberga la información principal de cada competición. Los datos nos los facilitará el COI.

TABLA DE EVENTO

- La tabla de eventos va a recoger la información de cada uno de los partidos que se van a realizar en la olimpiada. Nuestro sistema, a partir de las tablas de deportes, instalaciones y equipos gestionará esta tabla automáticamente. El identificador de cada evento va a ser “idEvento” y va a llevar información básica, como puede ser los equipos participantes, los árbitros, el deporte, la instalación, la fase de competición o la fecha

TABLA DE RESULTADO

- Esta tabla va a ser utilizada por los árbitros al introducir las actas de los eventos. Se guardará el resultado en tuplas de equipo – resultado. Esto es así para facilitarnos el almacenamiento de resultados de eventos donde participen más de un equipo, ya que, por ejemplo, en atletismo se pueden almacenar el corredor y su tiempo, en fútbol el equipo junto a su resultado etc.

TABLA DE LESION

- Esta tabla va a ser utilizada por los árbitros al introducir las actas de los eventos. Se guardará la lesión y el deportista que la sufre. En el campo observaciones se puede añadir información adicional como la duración estimada de la lesión o la gravedad.

TABLA DE SANCION

- Esta tabla va a ser utilizada por los árbitros al introducir las actas de los eventos. Se guardará la sanción, el deportista que ha sido sancionado y en qué deporte. También contiene un campo observaciones donde introducir información adicional como una descripción de la acción sancionada.

TABLA DE EVENTO_RESULTADO

- La misión de esta tabla es relacionar los eventos con el resultado de cada uno de los equipos que ha participado en el evento

TABLA DE ÁRBITRO

- Es otra tabla que nos dará el COI y simplemente se guardará el nombre y los apellidos del árbitro, el deporte que va a arbitrar y la nacionalidad de éste, muy importante a la hora de que no arbitre partidos de su propia nación.

TABLA DE ARBITROS_EVENTO

- En esta tabla habrá una relación entre los árbitros y los eventos que van a arbitrar.

TABLA DE EMPLAZAMIENTO

- En este caso la tabla del emplazamiento nos la dará el ayuntamiento de la ciudad organizadora y contendrá la información de las instalaciones que posee la ciudad para albergar las olimpiadas. Cada instalación viene identificada por “id” y lleva información como restricciones que tienen que ser compatibles con el atributo restricciones del deporte que se va a realizar en esta instalación. También tiene la capacidad de espectadores.

TABLA DE LUGAR_EVENTO

- En esta tabla se relacionara el evento con el emplazamiento que lo va a albergar.

TABLA DE ASIENTOS

- Cada estadio tendrá una tabla de asientos en la que cada asiento estará designado por un identificador “idAsiento”, que pertenecerá a un estadio concreto, “idEstadio”. Todos sabemos que en cada instalación hay asientos mejores y asientos peores, por eso hay que distinguirlo y como es obvio si el asiento es mejor tendrá un precio más caro. También es lógico que un asiento sea más costoso en una fase de competición que en otra, por ello esto también debemos de almacenarlo.

TABLA DE ENTRADA

- Existirá una tabla de entradas por cada instalación de las olimpiadas. Inicialmente estará vacía y a medida que se va reservando se va añadiendo una fila. Recordemos que para que una entrada sea vendida primero ha de ser reservada, por lo que las entradas vendidas son las que están en la tabla con el atributo “esReservada” a false. Por último es necesario añadir el identificador del usuario que compra la entrada y la fecha en la que lo hace, ya que hay que recordar que las entradas se compra por instalación y por día.

TABLA DE EQUIPOS

- El Access Control de esta tabla lo va tener el COI, las operaciones que va a poder realizar son:

```
añadirEquipo( idE, id, pais, inf)
pre:
  not(Equipo.filas -> exists(f1: fila |
    f1.idEquipo=idE and f1.idCompeticion=idC))
post:
  Equipo.filas -> exists(f1:fila |
    f1.idEquipo = idE and
    f1.idCompeticion = idC and
    f1.pais = pais and
    f1.info = inf)
```

```
eliminarEquipo(idE, idC)
pre:
  Equipo.filas -> exists(f1:fila |
    f1.idEquipo=idE and f1.idCompeticion=idC)
post:
  not(Equipo.filas -> exists(f1:fila |
    f1.idEquipo=idE and f1.idCompeticion=idC))
```

```
modificarDeporte(idE, idC, pais, inf)
pre:
  Equipo.filas -> exists(f1:fila |
    f1.idEquipo=idE and f1.idCompeticion=idC)
post:
  Equipo.filas -> exists(f1:fila |
    f1.idEquipo = idE and
    f1.idCompeticion = idC and
    f1.pais = pais and
    f1.info = inf)
```

TABLA DE DEPORTISTA

- El Access control en este caso también lo va a tener el COI, las operaciones que va a realizar son:

```
añadirDeportista(idD,nacion,nomb,apel1,ape2)
pre:
  not(Deportista.filas -> exists(f1: fila |
    f1.idDeportista=idE))
post:
  Deportista.filas -> exists(f1:fila |
    f1.idEquipo = idD and
    f1.nacionalidad = nacion and
    f1.nombre = nomb and
    f1.apellido1 = apel1
    f1.apellido2 = ape2)
```

```

eliminarEquipo(idD)
pre:
  Deportista.filas -> exists(f1:fila |
    f1.idDeportista=idD)
post:
  not(Deportista.filas -> exists(f1:fila |
    f1.idDeportista=idD))

```

```

modificarDeporte(idD,nacion,nomb,apel,ape2)
pre:
  Deportista.filas -> exists(f1:fila |
    f1.idDeportista=idD)
post:
  Deportista.filas -> exists(f1:fila |
    f1.idEquipo = idD and
    f1.nacionalidad = nacion and
    f1.nombre = nomb and
    f1.apellido1 = apel
    f1.apellido2 = ape2)

```

TABLA DE PERTENECE_DEPORTISTAS

- Esta tabla al relacionar los deportistas con el equipo al que pertenecen, el Access control es también del COI. Las operaciones que se pueden utilizar son:

añadirPertenece_Deportista(idE,idD)
 eliminarPertenece_Deportista(idE,idD)
 modificarPertenece_Deportista(idE,idD)

- Todas las operaciones anteriores son muy similares a añadirDeporte, modificarDeporte y eliminarDeporte, respectivamente. En esta tabla también van a ser importantes las operaciones de consulta como:

```

Equipo_de_deportista(idD)
pre:
  Pertenece_Deportista.filas->exists( f1:filas |
    f1.idDeportista = idD)
post:
  return Pertenece_Deportista.f1.idEquipo

```

```

Deportistas_de_Equipo(idE)
pre:
  Pertenece_Deportista.filas->exists( f1:filas |
    f1.idEquipo = idE)
post:
  return Pertenece_Deportista.f1.idDeportista

```

TABLA DE COMPETICION

- Esta tabla contiene toda la información relativa a la competición por lo que va a ser gestionada por el COI. Las operaciones son muy a las de la tabla deportista:

añadirCompeticion(idCompeticion, nombre, numPart, nArbitros, maxEquipos, minEquipos,
necPiscina,necPista, necCircuito)
modificarCompeticion(idCompeticion, nombre, numPart, nArbitros, maxEquipos, minEquipos,
necPiscina,necPista, necCircuito)
eliminarCompeticion(idCompeticion)

TABLA DE EVENTO

- Las operaciones que se pueden realizar son:

crearEvento(idEvento, equipos, árbitros, idDeporte, Emplazamiento, faseCompeticion, fecha)
modificarEvento(idEvento, equipos, árbitros, idDeporte, id Emplazamiento, faseCompeticion,
fecha)
EliminarEvento(idEvento)

TABLA DE RESULTADO

- Los árbitros emplean las siguientes consultas para poner los resultados:

```
crearActa(idE, res, obs)
pre:
  not(Resultados.filas->exits(f1:filas |
    f1.idEvento = idE)
post:
  Resutados.filas->exits(f1:filas |
    f1.idEvento = idE and
    f1.resultado = res and
    f1.observaciones = obs)
```

```
modificarActa(idE, res, obs)
pre:
  Resutados.filas->exits(f1:filas |
    f1.idEvento = idE)
post:
  Resutados.filas->exits(f1:filas |
    f1.idEvento = idE and
    f1.resultado = res and
    f1.observaciones = obs)
```

```
eliminarActa(idE)
pre:
  Resutados.filas->exits(f1:filas |
    f1.idEvento = idE)
post:
  not(Resultados.filas->exits(f1:filas |
    f1.idEvento = idE)
```

TABLA DE LESION

- El Access control de esta tabla pertenece al árbitro, las operaciones que puede realizar son:

añadirLesion(idLesion, idEvento, lesión, idDeportista, observaciones)
modificarLesion(idLesion, idEvento, lesión, idDeportista, observaciones)
eliminarLesion(idLesion, idEvento)

TABLA DE SANCION

- El Access control de esta tabla pertenece al árbitro, las operaciones que puede realizar son similares a las de la tabla lesión:

añadirSancion(idSancion, idEvento, idDeportista, penalización, observaciones)
modificarSancion(idSancion, idEvento, idDeportista, penalización, observaciones)
eliminarSancion(idSancion, idEvento)

TABLA DE EVENTO_RESULTADO

- La tabla está controlada también por el árbitro. Las operaciones que se pueden utilizar son:

añadirEvento_Resultado(idE,idR)
eliminarEvento_Resultado(idE,idR)
modificarEvento_Resultado(idE,idR)

- Todas las operaciones anteriores son muy similares a añadirDeporte, modificarDeporte y eliminarDeporte, respectivamente. En esta tabla también van a ser importantes las operaciones de consulta como:

```
Resultado_Evento(idE)
pre:
  Evento_Resultado.filas->exits( f1:filas |
    f1.idEvento = idE
post:
  return Evento_Resultado.f1.idResultado
```

TABLA DE ÁRBITRO

- El Access Control de esta tabla es del COI. Las operaciones que se van a poder realizar son:

añadirArbitro(idArbitro, nombre, apellidos, idDeporte)
eliminarArbitro(idArbitro)
modificarArbitro(idArbitro, nombre, apellidos, idDeporte)

- Estas operaciones son muy similares a añadirDeporte, modificarDeporte y eliminarDeporte respectivamente.

TABLA DE ARBITROS_EVENTO

- El Access control de la tabla es de Ayuntamiento. Las operaciones que se pueden utilizar son:

añadirArbitros_Evento(idA,idE)
eliminarArbitroEvento(idA,idE)
modificarArbitro_Evento(idA,idE)

- Todas las operaciones anteriores son muy similares a añadirDeporte, modificarDeporte y eliminarDeporte, respectivamente. En esta tabla también van a ser importantes las operaciones de consulta como:

```
Arbitros_Evento(idE)
pre:
    Arbitros_Evento.filas->exists( f1:filas |
        f1.idEvento = idE
post:
    return Arbitros_Evento.f1.idArbitro
```

TABLA DE EMPLAZAMIENTO

- La tabla relativa al emplazamiento nos la proporcionara el ayuntamiento de la ciudad organizadora, por lo que el Access control es del Ayuntamiento. Nosotros solo vamos a poder acceder a la tabla para hacer consultas. Las operaciones que se van a poder realizar son:

añadirEmplazamiento(idE, idD, rest, cap)
eliminarEmplazamiento(idE, idD)
modificarEmplazamiento(idE, idD, rest, cap)

- Estas operaciones son muy similares a añadirDeporte, modificarDeporte eliminarDeporte respectivamente.

```
exiteEmplazamiento(rest)
pre:
    not( Emplazamiento.filas->isEmpty() )
post:
    return Emplazamiento.filas->exists(f1:fila |
        f1.restriccion=rest)
```

TABLA DE LUGAR_EVENTO

- El Access control de la tabla es de Gestor de Eventos. Las operaciones que se pueden utilizar son:

añadirLugar_Evento(idEmplazamiento, idE, fecha)
eliminarLugar_Evento(idEmplazamiento, idE, fecha)
modificarLugar_Evento(idEmplazamiento, idE, fecha)

- Todas las operaciones anteriores son muy similares a añadirDeporte, modificarDeporte y eliminarDeporte, respectivamente. En esta tabla también van a ser importantes las operaciones de consulta como:

```
Lugar_Evento(idE, fecha)
pre:
  Lugar_Evento.filas->exits( f1:filas |
    f1.idEvento = idE and f1.fecha = fecha)
post:
  return Lugar_Evento.f1.idEmplazamiento
```

TABLA DE ASIENTOS

- El Access control pertenece al gestor de Entradas. Las operaciones que se puede realizar en esta tabla son:

```
insertarAsiento(idAsiento, localizador, idEmplazamiento)
modificarAsiento(idAsiento, localizador, idEmplazamiento)
eliminarAsiento(idAsiento)
```

TABLA DE ENTRADA

- El Access Control es el gestor de entradas y las operaciones que puede realizar son:

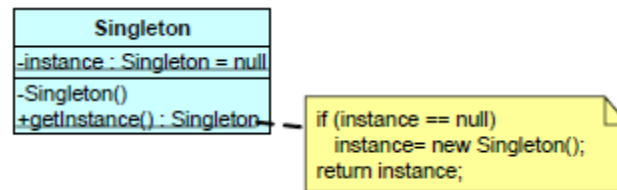
```
insertarEntrada(idEntrada, idAsiento, idEmplazamiento, fecha, estado, precio, comprador,
  fechaReserva, fechaCaducidad)
modificarEntrada(idEntrada, idAsiento, idEmplazamiento, fecha, estado, precio, comprador,
  fechaReserva, fechaCaducidad)
eliminarEntrada(idEntrada, idAsiento, idEmplazamiento)
```

```
verPrecio(idE, idA, idEmpl)
pre:
  Entrada.filas->exits( f1:filas |
    f1.idEntrada = idE and f1.idEmplazamiento = idEmpl
    and f1.idAsiento = idA)
post:
  return Entrada.f1.precio
```

PATRONES

PATRÓN SINGLETON

El objeto de este patrón es garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.



Usaremos este patrón cuando deba haber exactamente una instancia de una clase y deba ser accesible a los clientes desde un punto de acceso conocido. Por ejemplo:

- Acceso al archivo de configuración de una aplicación
- Gestor de impresión
- Una fábrica de objetos

PATRÓN DAO (DATA ACCESS OBJECT)

Los Objetos de Acceso a Datos son un Patrón de Diseño Core J2EE y considerados una buena práctica.

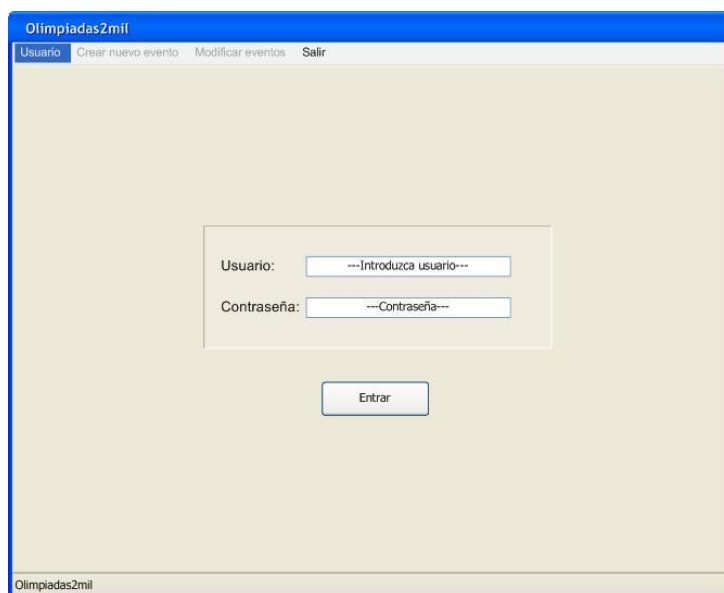
La ventaja de usar objetos de acceso a datos es que cualquier objeto de negocio (aquel que contiene detalles específicos de operación o aplicación) no requiere conocimiento directo del destino final de la información que manipula.

Consiste en utilizar un objeto de acceso a datos para abstraer y encapsular todos los accesos a la fuente de datos. El DAO maneja la conexión con la fuente de datos para obtener y almacenar datos.

INTERFAZ DE USUARIO

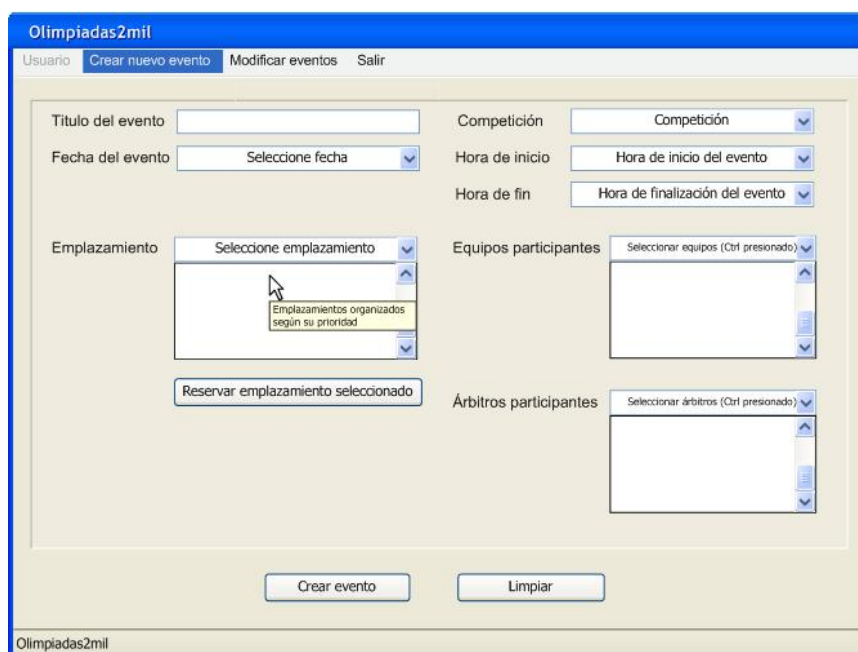
APLICACIÓN LOCAL

IDENTIFICACIÓN DE UN USUARIO:



Para acceder a las funcionalidades de la aplicación local es necesaria la identificación del usuario. Los usuarios de esta aplicación serán miembros del COI o del Ayuntamiento.

CREACIÓN DE UN NUEVO EVENTO:



El cuadro de competencias se crea automáticamente, cuando se inicia el software y se rellena a medida que se avanza en el mismo. Sin embargo, se permite la creación de eventos que no estén contemplados en el cuadro de competencias inicial.

MODIFICACIÓN DE LOS DATOS DE UN EVENTO:

The screenshot shows the 'Olimpiadas2mil' application window. At the top, there is a menu bar with 'Usuario', 'Crear nuevo evento', 'Modificar eventos' (which is highlighted), and 'Salir'. The main area is divided into two panels. The left panel contains two dropdown menus: 'Competición' and 'Evento', both currently showing 'Competición' and 'Evento' respectively. Below these is a 'Modificar evento' button. The right panel contains several fields: 'Fecha' (dropdown), 'Hora de inicio' (dropdown), 'Hora de fin' (dropdown), 'Emplazamiento' (dropdown with 'Seleccione emplazamiento'), a list box for 'Emplazamiento' with up/down arrows, a 'Reservar emplazamiento seleccionado' button, 'Equipos participantes' (dropdown with 'Seleccionar equipos (Ctrl presionado)'), a list box for 'Equipos participantes' with a tooltip that says 'Equipos de la competición que han superado etapas anteriores.', and 'Árbitros participantes' (dropdown with 'Seleccionar árbitros (Ctrl presionado)'). At the bottom left of the window, the text 'Olimpiadas2mil' is visible.

Los eventos se distribuyen automáticamente según los beneficios y mejoras que se puedan obtener en el orden de distribución, pero pueden ser modificados todos y cada uno de sus datos, con ciertas limitaciones:

- Solo se podrá cambiar el emplazamiento, a otro que esté disponible y que cumpla con las restricciones de la competición.

GESTIÓN DE USUARIOS

REGISTRO

Usuarios
Registro
Login

Noticias
Titulares
Última hora
Eventos online
Galería de fotos

Olimpiadas2020
Disciplinas
Distribución de eventos en el calendario
Resultados
Medallas

Entradas
Calendario
Ubicación de instalaciones
Gestión de entradas

Alta de usuario

ID Usuario	<input type="text"/>	País	<input type="text"/>
Contraseña	<input type="password"/>	Email	<input type="text"/>
Repetir contraseña	<input type="password"/>	Móvil	<input type="text"/>
DNI/NIF/...	<input type="text"/>	Profesión	<input type="text"/>
Nombre	<input type="text"/>	Sexo	<input type="text" value="M/F"/>
Apellidos	<input type="text"/>	Observaciones	<input type="text"/>
Dirección	<input type="text"/>		
Provincia	<input type="text"/>		

☒ He leído y acepto las condiciones de privacidad. [Leer](#)

☒ Deseo recibir en mi correo las últimas noticias.

El registro de usuarios se hace obligatorio para determinadas funcionalidades del software (gestión de entradas y visualización de eventos), facilitando al usuario la interacción en la plataforma, y de forma opcional para el resto de opciones.



La confirmación de dicho registro se hace automáticamente en la misma plataforma, ejecutando instantáneamente, y sin interacción con el usuario, su identificación; permitiendo desde ese momento, que se modifiquen los datos proporcionados.

IDENTIFICACIÓN DE USUARIO

The image shows a web interface for user identification. On the left is a sidebar menu with four sections: 'Usuarios' (containing 'Registro' and 'Login'), 'Noticias' (containing 'Titulares', 'Última hora', 'Eventos online', and 'Galería de fotos'), 'Olimpiadas2020' (containing 'Disciplinas', 'Distribución de eventos en el calendario', 'Resultados', and 'Medallas'), and 'Entradas' (containing 'Calendario', 'Ubicación de instalaciones', and 'Gestión de entradas'). The 'Login' option is highlighted. The main area features a large, faint background image of the Olympic rings. Overlaid on this is a light blue rectangular box containing the login form. The form has two input fields labeled 'ID Usuario' and 'Contraseña', followed by a blue 'Enviar' button. Below the button are two links: 'Recuperar contraseña' and 'Registrarse'.

Usuarios
Registro
Login

Noticias
Titulares
Última hora
Eventos online
Galería de fotos

Olimpiadas2020
Disciplinas
Distribución de eventos en el calendario
Resultados
Medallas

Entradas
Calendario
Ubicación de instalaciones
Gestión de entradas

ID Usuario

Contraseña

Enviar

[Recuperar contraseña](#)
[Registrarse](#)

Cada vez que el usuario entre en la interfaz web, tiene la posibilidad de identificarse, y la obligación de hacerlo si desea realizar las operaciones de visualización de eventos o gestionar entradas, con los datos que proporcione en su registro.

SALIDA



Para conseguir un mayor nivel de seguridad, el usuario deberá salir de la aplicación cada vez que finalice sus operaciones mediante la operación Logout.

MODIFICACIÓN DE DATOS DEL USUARIO

The screenshot shows the 'Modificar datos' (Modify data) form within the application. The sidebar menu is identical to the previous screenshot, with 'Usuarios' selected. The main content area has a title 'Modificar datos' and a subtitle 'Únicamente es necesario rellenar los campos a modificar'. The form consists of two columns of input fields. The left column includes: 'ID Usuario', 'Contraseña', 'Repetir contraseña', 'DNI/NIF/...', 'Nombre', 'Apellidos', 'Dirección', and 'Provincia'. The right column includes: 'País', 'Email', 'Movil', 'Profesión', 'Sexo' (with a dropdown menu showing 'M/F'), and 'Observaciones' (a text area). At the bottom of the form, there are two buttons: a red button labeled 'Eliminar usuario' and a green button labeled 'Modificar datos'.

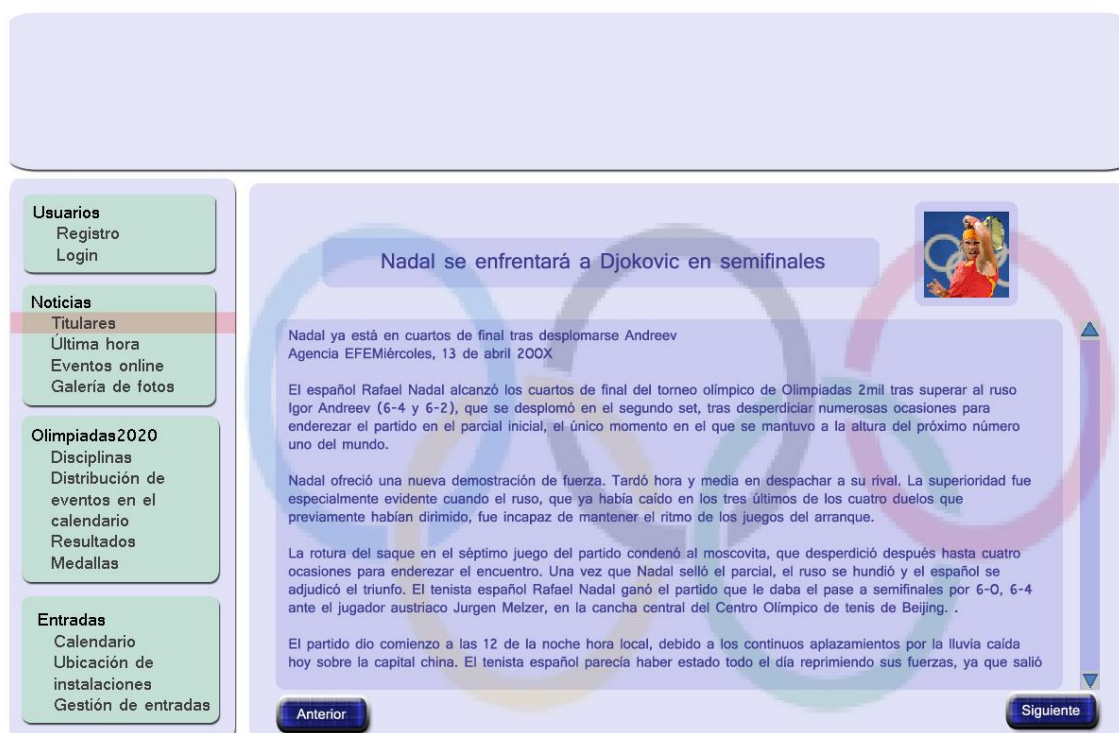
Todo usuario registrado e identificado tiene la posibilidad de modificar los datos que ha proporcionado, así como de borrarlos.

Para modificar los datos, el usuario debe rellenar, en el formulario, sólo los datos que desea modificar y confirmar mediante el botón “Modificar datos” dichas modificaciones.

Para eliminar su usuario, solo debe confirmar dicho deseo a través del botón “Eliminar usuario”, suprimiendo automáticamente al usuario de la aplicación.

GESTIÓN DE NOTICIAS

NOTICIAS – ULTIMA HORA – GALERÍA DE FOTOS



La página inicial de la aplicación web es Noticias. Interfaz común a todas las disciplinas olímpicas, organizada de forma inversa al suceso de la noticia.

Olimpiadas2mil proporciona herramientas de información de tipo “blog” al que tienen acceso todos los usuarios (con o sin identificación), así como también un servicio de correo mediante el cual se envían dicha información (únicamente a los usuarios registrados).

Esta herramienta proporciona noticias, ultima hora y galería de fotos, mantenido por el comité de organización de los eventos (COI y Ayuntamiento).

EVENTOS ONLINE SIN REGISTRO

Alta de usuario
Debe registrarse antes de realizar compras y/o reservas

ID Usuario:

Contraseña:

Repetir contraseña:

DNI/NIF/...:

Nombre:

Apellidos:

Dirección:

Provincia:

País:

Email:

Movil:

Profesion:

Sexo:

Observaciones:

☒ He leído y acepto las condiciones de privacidad. [Leer](#)

☒ Deseo recibir en mi correo las últimas noticias.

La visualización de eventos es una operación exclusiva para usuarios registrados, por lo que si un usuario no identificado intenta acceder a ella, es redirigido al formulario de registro. Si el usuario se ha registrado previamente, debe realizar el “Login” (o identificación).

EVENTOS ONLINE [SELECCIÓN → VISUALIZACIÓN]

Si el usuario se ha identificado previamente, puede seleccionar el evento a visualizar.

Eventos disponibles online

Boxeo - Cuarto de final - 19 Agosto Evento realizado

Boxeo - Cuarto de final - 20 Agosto Evento realizado

Boxeo - Semifinal - 21 Agosto

Boxeo - Semifinal - 22 Agosto

Boxeo - Final - 23 Agosto Aun no disponible

Boxeo - Final - 24 Agosto Aun no disponible

País	Equipo	Equipo	País	Detalles
	LAFFITA Andris	BALAKSHIN Georgy		Boxeo Semifinal 1 51 Kg. Ver online
	JONGJOHOR Somjit	PICARDI Vincenzo		Boxeo Semifinal 2 51 Kg. Ver online
	LOMACHENKO Vasyi	KILIC Yakup		Boxeo Semifinal 1 57 Kg. Ver online
	DJELKHIR Khedafi	IMRANOV Shahin		Boxeo Semifinal 2 57 Kg. Ver online
	BOONJUMNON Manus	IGLESIAS Roniel		Boxeo Semifinal 1 64 Kg. Ver online
	DIAZ Felix	VASTINE Alexis		Boxeo Semifinal 2 64 Kg. Ver online

El usuario debe seleccionar la disciplina y la competición que desea visualizar, automáticamente aparecerá un listado de los eventos disponibles para su visualización online.

The screenshot shows the 'Visualización de eventos' (Event Visualization) page. On the left, there is a sidebar with navigation links: 'Usuarios' (Modificar datos, Logout), 'Noticias' (Titulares, Última hora, **Eventos online**, Galería de fotos), 'Olimpiadas2020' (Disciplinas, Distribución de eventos en el calendario, Resultados, Medallas), and 'Entradas' (Calendario, Ubicación de instalaciones, Gestión de entradas). The main content area is titled 'Visualización de eventos' and features a grid of sports icons at the top. Below the icons, a list of events is displayed, including 'Boxeo - Cuarto de final - 19 Agosto', 'Boxeo - Cuarto de final - 20 Agosto', 'Boxeo - Semifinal - 21 Agosto', 'Boxeo - Semifinal - 22 Agosto', 'Boxeo - Final - 23 Agosto', and 'Boxeo - Final - 24 Agosto'. A table shows the participants for the 'Boxeo Semifinal 1 64 Kg.' event, with columns for 'País', 'Equipo', and 'Detalles'. The table lists two teams: 'BOONJUMNON Manus' and 'IGLESIAS Roniel'. A 'Ver online' button is visible, along with a checkbox for 'He leído y acepto las condiciones de privacidad' and a 'Leer' link. Navigation buttons 'Anterior' and 'Posterior' are at the bottom.

Una vez el usuario ha seleccionado el evento que desea ver, podrá visualizarlo

The screenshot shows the 'Visualización de eventos' (Event Visualization) page, similar to the previous one, but with a video player interface. The sidebar and event list are the same. The main content area now displays a large video player with a green play button in the center, indicating that the selected event is ready to be watched online. The 'Ver online' button is no longer visible, and the 'Anterior' and 'Posterior' navigation buttons remain at the bottom.

AMPLIACIÓN Y MODIFICACIÓN DE NOTICIAS – ULTIMA HORA – GALERÍA DE FOTOS



Todos los usuarios identificados como miembros del COI o del Ayuntamiento tienen la posibilidad de añadir, modificar e incluso, eliminar componentes en la herramienta de información; noticias, última hora o fotos para la galería.

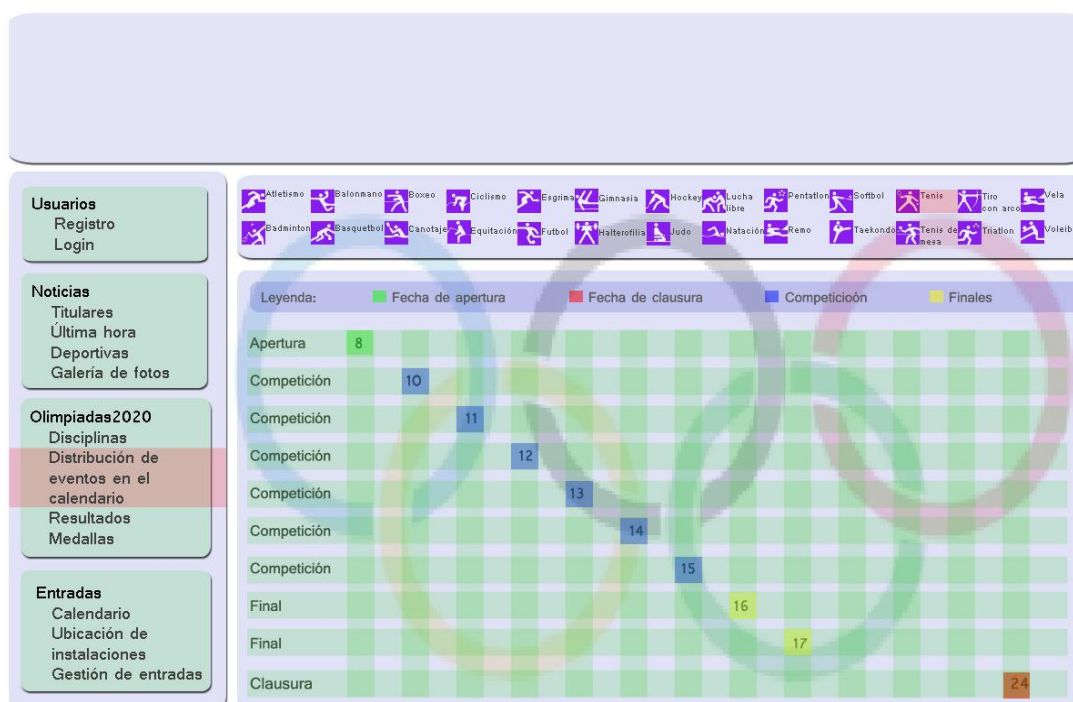
OLIMPIADAS2MIL - INFORMACIÓN

DISCIPLINAS



Muestra la descripción de las disciplinas que se disputaran, a todos los usuarios

CALENDARIO



Calendario de eventos deportivos, que se modifica automáticamente con las modificaciones de los eventos.

RESULTADOS DE LOS EVENTOS REALIZADOS

Usuarios
Registro
Login

Noticias
Titulares
Última hora
Eventos online
Galería de fotos

Olimpiadas2020
Disciplinas
Distribución de eventos en el calendario
Resultados
Medallas

Entradas
Calendario
Ubicación de instalaciones
Gestión de entradas

Individual - Femenino | 1a. ronda
Individual - Femenino | 2a. ronda
Individual - Masculino | 1a. ronda
Individual - Masculino | 2a. ronda
Dobles - Femenino | 1a. ronda
Dobles - Femenino | 2a. ronda
Dobles - Masculino | 1a. ronda
Dobles - Masculino | 2a. ronda

Pais	Equipo/Atleta	Resultado	Equipo/Atleta	Pais
	Federer, Roger	0 - 2	Blake, James	
	Nadal, Rafael	2 - 0	Meizer, Jurgan	
	Djokovic, Novak	2 - 0	Monfils, Gael	

Anterior Siguiente

AÑADIR RESULTADO

Únicamente los árbitros que hayan moderado un evento, pueden crear el acta de resultado de dicho evento, la cual solo puede ser ampliada por el mismo u otro arbitro participante del mismo evento, o autoridades del comité organizativo (COI).

Usuarios
Modificar datos
Logout

Noticias
Titulares
Última hora
Eventos online
Galería de fotos

Olimpiadas2020
Disciplinas
Distribución de eventos en el calendario
Resultados
Medallas

Entradas
Calendario
Ubicación de instalaciones
Gestión de entradas

Individual - Femenino | 1a. ronda
Individual - Femenino | 2a. ronda
Individual - Masculino | 1a. ronda
Individual - Masculino | 2a. ronda
Dobles - Femenino | 1a. ronda
Dobles - Femenino | 2a. ronda
Dobles - Masculino | 1a. ronda
Dobles - Masculino | 2a. ronda

Pais	Equipo/Atleta	Resultado	Equipo/Atleta	Pais
	Federer, Roger Leer acta	0 - 2 Modificar Acta	Blake, James Borrar Acta	
	Nadal, Rafael Leer acta	2 - 0 Modificar Acta	Meizer, Jurgen Borrar Acta	
	Djokovic, Novak Leer acta	2 - 0 Modificar Acta	Monfils, Gael Borrar Acta	

[Anterior](#) [Siguiente](#) [Anterior](#) [Generar nueva acta](#) [Siguiente](#)

Para agregar una nueva acta, el árbitro debe seleccionar el evento correspondiente y rellenar el formulario con los hechos sucedidos en ese evento.

Usuarios
Modificar datos
Logout

Noticias
Titulares
Última hora
Eventos online
Galería de fotos

Olimpiadas2020
Disciplinas
Distribución de eventos en el calendario
Resultados
Medallas

Entradas
Calendario
Ubicación de instalaciones
Gestión de entradas

Individual - Femenino | 1a. ronda
Individual - Femenino | 2a. ronda
Individual - Masculino | 1a. ronda
Individual - Masculino | 2a. ronda
Dobles - Femenino | 1a. ronda
Dobles - Femenino | 2a. ronda
Dobles - Masculino | 1a. ronda
Dobles - Masculino | 2a. ronda

Equipo participante

Resultado

Incidencias

Lesiones

Sancciones

Observaciones

[Anterior](#) [Siguiente](#) [Guardar](#) [Cancelar](#)

Dicha acta, será almacenada y el resultado del evento será mostrado en el cuadro de resultados (visible para todos los usuarios).

AÑADIR ADJUNTO A RESULTADO

Usuarios
 Modificar datos
 Logout

Noticias
 Titulares
 Última hora
 Eventos online
 Galería de fotos

Olimpiadas2020
 Disciplinas
 Distribución de eventos en el calendario
 Resultados
 Medallas

Entradas
 Calendario
 Ubicación de instalaciones
 Gestión de entradas

Individual - Femenino | 1a. ronda
 Individual - Femenino | 2a. ronda
 Individual - Masculino | 1a. ronda
 Individual - Masculino | 2a. ronda
 Dobles - Femenino | 1a. ronda
 Dobles - Femenino | 2a. ronda
 Dobles - Masculino | 1a. ronda
 Dobles - Masculino | 2a. ronda

Pais	Equipo/Atleta	Resultado	Equipo/Atleta	Pais
	Federer, Roger Leer acta	0 - 2 Agregar adjunto	Blake, James Borrar Acta	
	Nadal, Rafael Leer acta	2 - 0 Agregar adjunto	Meizer, Jorgen Borrar Acta	
	Djokovic, Novak Leer acta	2 - 0 Agregar adjunto	Monflis, Gael Borrar Acta	

Anterior Siguiente Anterior Generar nueva acta Siguiente

La modificación de actas de un evento no está permitida. Cualquier corrección que se desea aplicar al acta de un evento, se mostrará como un adjunto del acta original.

Al igual que el acta oficial del evento, los únicos usuarios con posibilidad de realizar esta operación son los árbitros participantes del evento y las autoridades del comité (COI).

MEDALLERO

Usuarios
 Registro
 Login

Noticias
 Titulares
 Última hora
 Eventos online
 Galería de fotos

Olimpiadas2020
 Disciplinas
 Distribución de eventos en el calendario
 Resultados
 Medallas

Entradas
 Calendario
 Ubicación de instalaciones
 Gestión de entradas

Federación				Federación			

El medallero muestra un listado (ordenado de forma descendente) de las medallas obtenidas por cada federación durante el campeonato, olimpiadas,...

GESTION DE ENTRADAS

CALENDARIO



ENTRADAS

Al igual que en el caso de “Eventos online”, si un usuario no identificado, desea acceder a la “Gestión de entradas”, este es redirigido al apartado de “Registro”. En el caso, de ser un usuario registrado, debe identificarse para poder acceder a esta sección.

RESERVAS Y ANULACIONES



The screenshot shows the 'Gestión de entradas' (Ticket Management) page. On the left is a sidebar with navigation links: Usuarios (Modificar datos, Logout), Noticias (Titulares, Última hora, Eventos online, Galería de fotos), Olimpiadas2020 (Disciplinas, Distribución de eventos en el calendario, Resultados, Medallas), and Entradas (Calendario, Ubicación de instalaciones, Gestión de entradas). The main content area has a header with 'Nueva entrada' and 'Reservas, anulaciones y compras'. Below this is a table with three rows of reservations:

Evento	Detalles del evento	Detalles de la entrada	Comprar	Anular
TIRO CON ARCO Individual Masculino Semifinal	15 de Agosto de 20XX 12:00 hh	Asiento preferente 16...	Comprar	Anular
NATACIÓN 400 metros Femenino Final	17 de Agosto de 20XX 10:00 hh	Asiento no-preferente 144...	Comprar	Anular
BASQUETBOL Final Masculino	24 de Agosto de 20XX 09:00 hh	Asiento VIP 4...	Comprar	Anular

At the bottom of the table are 'Anterior' and 'Siguiente' navigation buttons.

Página principal de la sección “Gestión de entradas”, donde se puede visualizar (en el caso de que existan) las reservas de un usuario identificado.



This screenshot shows the same 'Gestión de entradas' page, but instead of a table of reservations, a large blue box in the center displays the message: 'No tiene reservas que pueda anular o comprar' (You have no reservations that you can cancel or buy). The sidebar and header are identical to the previous screenshot.

NUEVA ENTRADA

Usuarios
Modificar datos
Logout

Noticias
Titulares
Última hora
Eventos online
Galería de fotos

Olimpiadas2020
Disciplinas
Distribución de eventos en el calendario
Resultados
Medallas

Entradas
Calendario
Ubicación de instalaciones
Gestión de entradas

Nueva entrada

Reservas, anulaciones y compras

Boxeo - Cuarto de final - 19 Agosto

Evento realizado

Boxeo - Cuarto de final - 20 Agosto

Evento realizado

Boxeo - Semifinal - 21 Agosto

Boxeo - Semifinal - 22 Agosto

Boxeo - Final - 23 Agosto

Aun no disponible

Boxeo - Final - 24 Agosto

Aun no disponible

Pais	Equipo	Equipo	Pais	Detalles	
	LAFFITA Andris	BALAKSHIN Georgy		Boxeo Semifinal 1 51 Kg.	Reservar
	JONGJOHOR Somjit	PICARDI Vincenzo		Boxeo Semifinal 2 51 Kg.	Reservar
	LOMACHENKO Vasyi	KILIC Yakup		Boxeo Semifinal 1 57 Kg.	Reservar
	DJELKHIR Khedafi	IMRANOV Shahin		Boxeo Semifinal 2 57 Kg.	Reservar
	BOONJUMNON Manus	IGLESIAS Roniel		Boxeo Semifinal 1 64 Kg.	Reservar
	DIAZ Felix	VASTINE Alexis		Boxeo Semifinal 2 64 Kg.	Reservar

Anterior
Posterior
Atras
Adelante

Para realizar la compra o reserva de una nueva entrada, el usuario identificado, selecciona la disciplina y nivel de competición de la misma, lo que conlleva a la presentación de un listado de eventos, para cada uno de los cuales se pueden realizar la reserva de una entrada del evento seleccionado.

Usuarios
Modificar datos
Logout

Noticias
Titulares
Última hora
Deportivas
Galería de fotos

Olimpiadas2020
Disciplinas
Distribución de eventos en el calendario
Resultados
Medallas

Entradas
Calendario
Ubicación de instalaciones
Gestión de entradas

Nueva entrada

Reservas, anulaciones y compras

Boxeo - Cuarto de final - 19 Agosto

Evento realizado

Boxeo - Cuarto de final - 20 Agosto

Evento realizado

Boxeo - Semifinal - 21 Agosto

Boxeo - Semifinal - 22 Agosto

Boxeo - Final - 23 Agosto

Aun no disponible

Boxeo - Final - 24 Agosto

Aun no disponible

Pais	Equipo	Equipo	Pais	Detalles	
	BOONJUMNON Manus	IGLESIAS Roniel		Boxeo Semifinal 1 64 Kg.	Reservar

Modificar características

Forma de pago

☒ He leído y acepto las condiciones de privacidad. [Leer](#)

Reservar

Comprar

Anterior
Posterior

Una vez, el usuario ha seleccionado la reserva de un evento puede realizar la compra inmediatamente, rellorando las características de la reserva y la forma de pago, o únicamente su reserva.

Ésta es la página de la interfaz que aparece de forma posterior a la selección de la operación de compra de una reserva.

En el caso de realizar la reserva, se mostrará en el listado de “Reservas, anulaciones y compras” de la sección “Gestión de entradas”.

INVESTIGACIÓN SOBRE LOS COMPONENTES

Primeramente exponer que vamos a utilizar una forma de programación en 3 capas. La capa de presentación, la de negocios y la de datos. La capa de presentación va a ser el componente de la interfaz, la capa de negocios va a ser la propia aplicación y la capa de datos va a ser la base de datos.

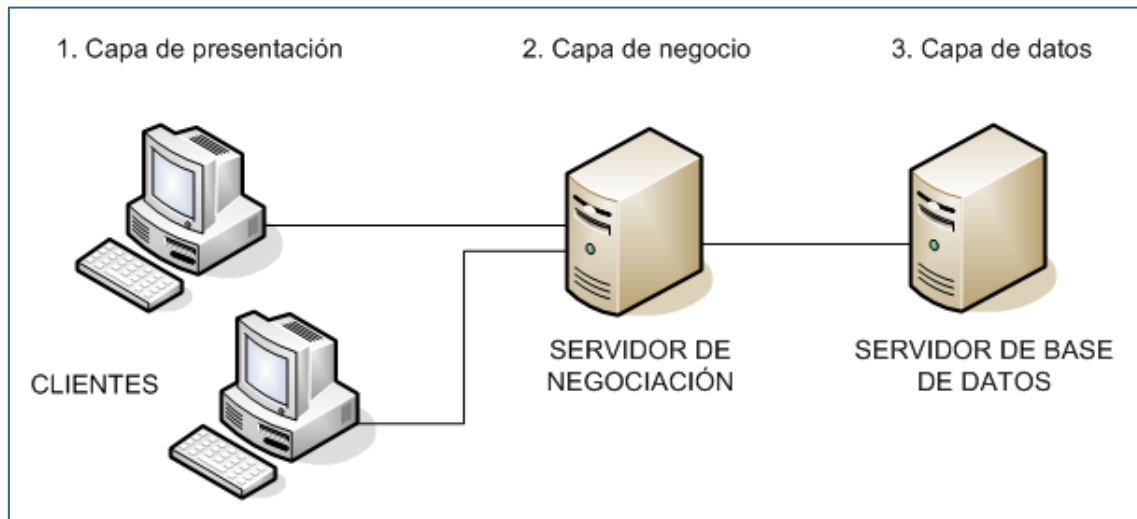


Figura: Modelo en tres capas

PROPUESTA PARA LA INTERFAZ

La interfaz se va a desarrollar en lenguaje PHP. Algunas de las ventajas que hemos visto en utilizar este lenguaje para desarrollar nuestra interfaz son:

- Es un lenguaje multiplataforma.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

La interfaz va a consistir básicamente en la parte visual, todos los accesos a la base de datos se van hacer a través de la capa de negocio, la interfaz solo va a llamar a las funciones de esa capa para que se ejecuten. El creador de la interfaz no tiene porque saber el funcionamiento exacto del sistema.

El lenguaje que vamos a utilizar para la aplicación es Java, por lo que tenemos que llamar desde PHP a java, el siguiente enlace nos muestra como se puede llevar a cabo:

<http://gonetil.wordpress.com/2008/07/28/instalar-phpjava-bridge-en-linux/>

PROPUESTA PARA LA BASE DE DATOS

Para desarrollar la nuestra base de datos nos hemos decantado por la última versión estable de Oracle, la Database 11g Enterprise Edition, ya que Oracle tiene el record mundial en rendimiento y en la relación calidad/precio, lo cual nos viene muy bien para nuestro sistema.

Las características más importantes que posee Oracle son:

- Es el gestor de bases de datos más utilizado por las compañías más grandes del mundo, por su robustez y por la seguridad.
- Tiene múltiples versiones para cada sistema operativo, entre ellos UNIX, LINUX, NT, Win 2000- 98 - 95, Novel etc.
- Ofrece soporte mundial a través de sus centros de soporte y sus sitios Web donde se puede encontrar desde scripts hasta documentos de instalación.
- Proporciona actualización de versiones gratis al cliente; siempre y cuando la licencia de soporte lo cubra.
- El 80% o más de los sitios Web en Internet tienen bases de datos Oracle.
- Oracle es más que un gestor de bases de datos. Oracle Corporation ofrece otras soluciones a la plataforma de Negocio, EBusiness, Ecommerce etc.
- Puedes programar con las últimas herramientas del mercado.

PROPUESTA PARA EL BLOG

Nuestra página web posee una parte de noticias, donde se actualizarán las novedades que vayan surgiendo y que gestionaremos por medio de un blog. Para esta parte nos hemos decantado por Wordpress, ya que sólo con instalarlo tienes acceso a los blogs.

Para instalar el componente blog en nuestra página web únicamente hemos incorporando Wordpress a nuestra página en PHP. La forma de hacerlo es bastante sencilla tal y como se muestra en el siguiente enlace:

<http://ezequino.com.ar/?p=28>

Al registrarse en nuestra aplicación, damos opción al usuario de suscribirse a un servicio de noticias y actualizaciones para que las pueda consultar en su correo o en su teléfono móvil. Para ello vamos a incluir en nuestro blog el protocolo RSS, que es una herramienta para publicar o visualizar artículos de una sola fuente en diferentes medios al mismo tiempo, generalmente al instante de publicarlos. Muchos medios en línea ponen a disposición su feed (dirección única del RSS del sitio) para mostrar los últimos titulares.

En el siguiente enlace se puede ver cómo hacerlo:

<http://www.vidadigital.net/blog/2008/04/21/rss-y-tu-blog-en-wordpresscom/>

PROPUESTA PARA EL COMPONENTE DE PAGO

PayPal es un sistema que permite a cualquier persona que tenga una dirección de correo electrónico enviar o recibir dinero online utilizando su tarjeta de crédito de manera totalmente segura.

Hemos pensado que sería una forma amena y sencilla de incorporar pagos online para nuestra aplicación, ya que dispone de herramientas guiadas y sencillas para la integración del paquete en la aplicación web. Además, es gratuita.

La forma de integrarlo se ve en la siguiente página:

<http://www.elwebmaster.com/articulos/un-script-php-para-integrar-pagos-con-paypal-a-tu-sitio-web>

PROPUESTA PARA LA VISUALIZACIÓN DE EVENTOS

Para utilizar este componente debemos poseer un servidor streaming en directo.

El [streaming](#) (flujo de datos) permite reproducir (escuchar o visualizar) contenidos multimedia procedentes de la red sin tener que esperar a que el archivo se haya descargado por completo, ya que la secuencia de audio/vídeo se reproduce mientras se descarga.

En el streaming en directo por Internet, el contenido multimedia es reproducido a iniciativa del servidor y la señal se retransmite y se visualiza en tiempo real. Intervienen tres componentes, gráficamente:

- Fuente multimedia: mezcla varias entradas (micrófono, webcam, archivos .mp3, CD, etc.), transforma la señal a un formato que entienda el servidor (.ogg o .mp3), empaqueta la señal y la convierte en un flujo de datos.
- Servidor streaming: recibe el flujo de datos (stream) y lo envía a la red.
- Cliente streaming: visualiza la secuencia de audio/vídeo mientras la descarga.

Hay varios servidores de streaming para Linux, pero el más utilizado es [IceCast](http://icecast.org) (icecast.org, paquete icecast2). Sus principales ventajas son:

- Puede manejar varios streams diferentes a la vez, cada uno con una URL.
- La fuente del stream y el servidor IceCast pueden estar en las distintas máquinas, algo que puede venir bien si donde está la fuente no hay una conexión con suficiente subida para soportar muchos clientes.
- Si queremos poner un enlace en nuestra web a una fuente IceCast la URL será (por ejemplo): http://nuestro_dominio:8000/freej.ogg.m3u

CUESTIONARIO CMMI

NIVEL 2 (REPETIBLE)

A lo largo del desarrollo del proyecto, ¿la forma de trabajo ha sido clara y controlada? ¿Por qué? ¿Se ha seguido algún método, procedimiento y/o entrenamiento?

Se distinguen dos líneas de opinión muy diferenciadas.

- La forma de trabajo no ha sido clara y controlada desde el principio, y al obtener una visión de cómo y cuánto se avanzaba, se eligió como metodología de trabajo divide y vencerás, es decir, la repartición del trabajo de forma equitativa y con ciertas limitaciones, evitando así tener que consensuar todas las cuestiones que se presentasen (relevantes o irrelevantes), comunicando únicamente las cuestiones de importancia global y con repercusiones.
- La forma de trabajo ha sido clara pero descontrolada, ya que no se eligió ningún método de trabajo determinado y ni se realizó entrenamiento previo a la realización de tareas, lo que conlleva a múltiples suposiciones erróneas y por tanto, a un retroceso (rehacer) en el trabajo.

¿El proyecto ha tenido una documentación correcta y coherente en cada fase? Argumenta tu respuesta

Se manifiesta la existencia de opiniones contrapuestas:

- El proyecto no ha tenido una documentación correcta y coherente, ya que los cambios realizados y requerimientos nuevos se comunicaban en las reuniones y no se reflejaban, así como sus motivos y soluciones, en ningún documento escrito. Esta forma de trabajo tuvo como resultado la necesidad de retomar debates ya concluidos.
- El proyecto ha tenido una buena documentación, acorde al desarrollo de las tareas. Se mantuvo una documentación visual que facilitaba la necesidad de otras explicaciones, y se controlaron las versiones de la documentación mediante el repositorio donde se alojó el proyecto durante su desarrollo.

En todo momento, ¿se ha conocido la documentación a presentar en la entrega final?

Se distinguen dos líneas de opinión muy diferenciadas.

- Durante el desarrollo del proyecto no ha sido conocida la documentación a presentar, hasta el final del proceso.
- La documentación a presentar ha sido conocida pero de forma inexacta, es decir, no en su totalidad; ya que se desconocía, de forma exacta, el alcance y nivel de profundidad que se deseaba obtener, porque siempre iba en aumento y la principal fuente de conocimiento a lo que se accedió fueron proyectos de años anteriores.

¿La documentación final ha sido elaborada desde el principio del proyecto?

Se manifiesta la existencia de opiniones contrapuestas:

- La documentación ha sido elaborada desde el principio, sentando las bases más sólidas posibles para obtener la coherencia y consistencia del documento, que ha sido modificado en numerosas ocasiones según aumentaban los requisitos de la entrega así como las modificaciones propias del proyecto, en cuanto a su definición y su alcance.
- La documentación final se ha relegado al último paso en el desarrollo, ya que se entiende como un documento recopilatorio de todo el trabajo realizado, que brinda la oportunidad de realizar una revisión total de todo el proyecto

¿El tiempo ha sido distribuido de una forma correcta y necesaria en las distintas fases del proyecto?

La distribución del tiempo no ha sido la idónea, por diversos factores:

- El método de trabajo al principio no era el adecuado, ya que todos los miembros no podían dedicarse a la misma tarea. Por esto motivo, se optó por la repartición de trabajo de forma equitativa, asumiendo una cantidad de tiempo desperdiciada por la mala organización
- La fase inicial (Especificación de Requisitos) se dio por concluida estando en un estado inexacta y demasiado ambicioso, lo que conllevó a múltiples debates acerca de las funcionalidades a desarrollar y el rediseño que obligaba a realizar los cambios decididos en los debates mencionados.

¿La planificación del trabajo ha sido la adecuada?

La planificación no ha sido la adecuada, por diversos motivos;

- La falta de un método de trabajo. El intento de que todos los miembros abarcasen todas las tareas conllevó al retraso y a tomar la iniciativa de dividir el trabajo, estrategia que debió emplearse desde el inicio del proyecto.
- Los innumerables debates sobre cualquier cambio. Se debería debatir solo lo imprescindible, es decir, aquellos cambios que tengan una repercusión real y significativa; y comunicar a todos los miembros en reuniones cortas los avances realizados por cada subgrupo.
- Falta de especificación en la idea a desarrollar. Se debería dedicar más tiempo en la especificación de requisitos dejando aclarado en su totalidad, y en medida de lo posible, sin lugar a ambigüedades, las funcionalidades y repercusiones de cada requisito, haciendo consciente al equipo del nivel de ambición del proyecto y el tiempo necesario para desarrollarlo, como también evitando la necesidad de rediseñar.
- Mayor conocimiento del trabajo a realizar desde el inicio del proyecto, facilitando de esta forma la metodología de repartición de tareas.

- Diseñar una estructura o jerarquía con roles bien definidos, existiendo una figura encargada de organizar al resto del equipo según las necesidades del proyecto.

¿La especificación de requisitos ha sido bien definida desde el inicio?

La especificación de requisitos no ha estado bien definida, debido a la falta de detalle en su definición y el exceso de ambición debido a las funcionalidades que se planteaban, lo que generó la necesidad de reducir dichas funcionalidad a medida que avanzaba el proyecto, y los innumerables debates causados por las ambigüedades o falta de exactitud en las definiciones; debiendo realizar reestructuraciones y rediseños.

Porcentaje de requisitos modificados

Los requisitos han sido modificados en torno al 60%, siendo una gran parte de estas modificaciones eliminaciones de requisitos. Las modificaciones realizadas han supuesto una gran pérdida de tiempo y esfuerzo, dando lugar a replanteamientos y rediseños, pasando por extensos debates sobre cada modificación.

NIVEL 3 (DEFINIDO Y ESTRUCTURADO)

¿La forma de trabajo ha funcionado bien?

Se distinguen dos líneas de opinión muy diferenciadas.

- La forma de trabajo ha funcionado bien al final, después de la evolución del grupo hasta conseguir su total integración.
- La forma de trabajo no ha funcionado bien porque no ha sido la adecuada, el proyecto se desarrolló bajo un grupo con una metodología desordenada, en la que todos los miembros trabajaban y opinaban sobre todo generando grandes pérdidas de tiempo y exceso de modificaciones

¿La definición de cada componente del proyecto ha sido la adecuada?

Cada componente ha tenido una definición correcta en un punto cercano a la versión definitiva. El principal obstáculo para obtener la definición adecuada eran las distintas modificaciones que se realizaban sobre la misma, ya que cada miembro del grupo tenía una visión diferente, lo que sumó al equipo en deliberaciones hasta obtener un punto en común (olvidado y posteriormente retomado en diversas ocasiones por la falta de documentación)

¿Las herramientas utilizadas en cada parte del desarrollo del proyecto han sido adecuadas?

Las herramientas han tenido que reemplazarse en varias ocasiones ya que no abarcaban todas las tareas que se requerían. Entre las limitaciones de las herramientas destacan:

- Incapacidad intrínseca para dividir el trabajo con Subversion (Trabajo en paralelo).
- Incapacidad de integración con otros modelos (OCLs)
- Ausencia de corrección sintáctica (OCLs)

¿El estado y avance del proyecto ha sido conocido, en todo momento, por cada miembro del equipo?

Cada miembro tenía a su disposición, en todo momento y en exactitud, conocer el estado y modificaciones realizadas en el proyecto mediante el envío de correos y actualizaciones del repositorio

Aunque el equipo es unánime en cuanto a la disponibilidad para conocer el estado del proyecto, se distinguen opiniones opuestas basadas en la dificultad para conseguirlo:

Todas las modificaciones eran consensuadas y en su mayoría realizadas en las horas lectivas de la asignatura (laboratorios) con la presencia de la mayoría (o totalidad) de los miembros del equipo.

Las actualizaciones en el repositorio se realizaban sin documentar (comentarios explicativos) y de forma autoritaria o no consensuada (ya que se suponía el concilio de opiniones entre todos los miembros), generando discusiones y una gran complejidad por desconocer los motivos respecto a modificaciones hechas.

¿El método de trabajo ha sido centralizado (reuniones periódicas)?

El método de trabajo ha sido centralizado ya que las horas lectivas en el laboratorio facilitan esta tarea. Se realizaban reuniones 2 veces por semana de 2 horas de duración para la puesta en común y nueva división de tareas.

¿Qué porcentaje (%) de tiempo se ha invertido en el aprendizaje necesario para desarrollar el proyecto?

El aprendizaje necesario se aproxima a un 20 %, ya que gran parte de esta tarea se realizaba en paralelo con las tareas a las que se destinaba el aprendizaje.

En oposición a esta opinión:

- No se invirtió tiempo en aprendizaje ya que lo explicado en las horas lectivas de teoría era suficiente, en suma con otras asignaturas anteriores.

¿Existe coordinación entre las diferentes tareas de los distintos miembros?

Existe coordinación ya que al tener un método de trabajo centralizado, todos los miembros conocen las tareas que se desarrollan en cada etapa así como el subgrupo o miembro encargado de dicha tarea.

¿Qué herramientas han favorecido la coordinación entre los miembros?

- Repositorio GoogleCode: Olimpiadas2mil
- Grupo de correo (GoogleGroups): Olimpiadas2020
- Eclipse (herramienta de modelado gratuita que interactúa con Subversion)

¿Qué porcentaje (%) de tiempo dedicado al proyecto han sido fructífero?

La mayoría del equipo estima que aproximadamente un 60% del tiempo dedicado ha sido fructífero, mientras que existe una estimación que considera un 30%.

¿Qué porcentaje (%) de tiempo se ha dedicado a cada fase del proyecto? ¿Ha sido el adecuado?

- Especificación de requisitos: 20%
- Diseño (Diagramas, invariantes, precondiciones, postcondiciones,...): 60%
- Memoria y documentación: 20%

No ha sido la distribución adecuada, por no generar un resultado óptimo. La escasa dedicación en la especificación de requisitos conllevó a gran número de debates y forzó a múltiples etapas de rediseño por cambios realizados.

¿Qué tareas han significado una pérdida de tiempo?

- Cambios de herramientas (rehacer lo que ya estaba hecho)

Las modificaciones y correcciones en los requisitos han significado un retraso significativo y las discusiones que generaban, un aumento en gran medida de este tiempo desperdiciado, ya que cada debate significaba el replanteamiento de alguno o varios elementos y su correspondiente funcionamiento.

- Diseños realizados (inconclusos) desechados.

¿El tiempo para realizar una tarea ha podido ser estimado/predicho?

La estimación de una tarea dependía en gran parte de las complicaciones que generará, debido a la realización de algún cambio y las correspondientes modificaciones que se deben realizar; por este motivo, el equipo se divide en dos posturas:

- Subgrupo que considera que podrían estimarse las tareas (aproximadamente)
- Subgrupo que considera que el tiempo dedicado a realizar una tarea no es predecible.

¿En cada parte del proyecto, se ha distinguido la figura de un líder?

No se distingue ninguna figura líder, cada miembro era consciente y responsable de la tarea que había elegido a realizar. Existían miembros más conscientes de las tareas a realizar que servían de guía para el desarrollo del proyecto.

¿Se han distinguido especialistas en el proyecto?

No existen especialistas declarados. Durante gran parte del desarrollo del proyecto el trabajo se ha dividido de forma equitativa en subgrupos, permitiendo que todos los miembros pudiesen participar en distintas tareas en las diversas fases.

¿Existen diferentes puntos de vista sobre la gestión del proyecto?

Existen tantos puntos de vista como miembros. Al no tener una estructura jerarquizada ni una especificación de requisitos totalmente clara y definida, cada miembro intentaba desarrollar el proyecto bajo su visión personal lo que generaba enfrentamientos (grandes pérdidas de tiempo). A medida que el proyecto tomaba forma y consistencia los debates disminuían, ya que toda la idea pasaba a tener claridad, consistencia y precisión.

¿El progreso en el proyecto ha sido lineal? Argumenta tu respuesta

Se divide en dos subgrupos que difieren en:

- El progreso ha sido lineal.
- El progreso del proyecto ha sido por ráfagas, siendo estas positivas (avances significativos en periodos cortos) o negativos (replanteamiento de requisitos, rediseño,... lo que generaba retrasos)

Aunque las ideas de ambos subgrupos se armonizan en la opinión de que en los últimos meses, sujetos a la presión de realizar una entrega, la progresión ha sido exponencial.

Nivel de madurez obtenido 1,4

MODIFICACIONES A APLICAR PARA MEJORAR EL NIVEL DE MADUREZ ALCANZADO

- Elegir un modelo de desarrollo.
- Revisar cada fase antes de darla por concluida e intentar esclarecer lo más posible cada punto o tarea, para evitar posibles confusiones, discusiones y la necesidad de realizar cambios en tareas concluidas.
- Elección de herramientas, basadas en documentación y experiencia.
- Establecer una jerarquía (elegir un líder), formar especialistas y generar una planificación global y otra dedicada a cada fase.

- Elaborar el documento final desde el principio, para hacernos más conscientes del trabajo ya realizado y las tareas que quedan por hacer.

EVOLUCIÓN DEL PROYECTO

En un principio pensábamos gestionar:

Gestión de las competiciones:

- Creación del cuadro de la competición de cada disciplina.
- Inscripción de los deportistas en las competiciones.

Gestión del emplazamiento:

- Que cumpla con los requisitos del deporte (natación – piscina, tenis – pista de tenis....)
- Que esté bien comunicado con el alojamiento
- Que tenga el aforo adecuado
- Que todos los emplazamientos de la misma disciplina estén lo más cerca posible.

Gestión del horario:

- Que no se solape con una competición de las mismas características
- Cercano a los horarios de máxima audiencia

Gestión de entradas

- Venta de entradas a espectadores
- Pases de prensa para los medios
- Invitaciones para organizadores

Gestión de los participantes:

- Gestión de alojamiento, de forma que el alojamiento este lo más cerca posible o lo mejor comunicado posible de la competición y que este lo más cerca posible del resto de participantes de tu país en otras competiciones.
- Gestión de transporte, para que desde los hoteles a las pistas deportivas haya buena comunicación.

Gestión de los árbitros:

- Gestión de alojamiento: deberían de estar en una zona de la ciudad, aislados de los participantes y todos juntos.
- Gestión del transporte: tendrían un transporte privado para ellos desde su lugar de hospedaje hasta la competición deportiva.

Gestión del personal contratado y voluntario.

- Definición de sueldos.
- Asignación tareas y horarios así como el organigrama de la organización.

Gestión de patrocinadores:

- Asignación de recursos publicitarios en base al grado de colaboración, estos recursos serán gestionados por terceros para la administración de publicidad.
- Gestión de los espacios de publicidad, que incluye el control del cumplimiento de las empresas encargadas de la gestión.
- Acuerdos de colaboración con empresas de transportes para reforzar sus servicios.

Venta de entradas por Internet.

Portal deportivo:

- Mantiene la información de los resultados, medallero, clasificaciones, así como el horario y lugar de las competiciones.

Íbamos a tener los siguientes roles y perfiles:

- ROL: Administración: Posee el control total de la aplicación.
 - PERFIL: Administrador del Comité de la Sede
- ROL: Control parcial: Solo en algunos aspectos de la aplicación
 - PERFIL: Autoridad pública (policías, bomberos...)
 - Ejemplo: Hay un accidente grave en la nacional de acceso al evento 3 horas antes. El policía introduce la incidencia y el sistema alerta a la organización, participantes y espectadores que hayan adquirido una entrada por mail o sms. Además en la consulta de rutas de acceso al evento muestra una ruta alternativa para llegar al estadio
 - PERFIL: Federación
 - Ejemplo: Se lesiona 3 horas antes un participante. La aplicación gestiona esa baja e informa a la administración.
 - PERFIL: Espectador
 - Ejemplo: Al comprar una entrada se modifican las localidades libres, se incrementa la recaudación del evento...
 - PERFIL: Medio de comunicación

- Ejemplo: TVE informa que sus 2 pases de prensa los van a ocupar Pepito Pérez y Juan Rodríguez.

- ROL: Consulta

- Cualquier perfil tiene como mínimo el perfil de consulta.
- Sector servicios: transporte, alojamiento, restaurantes, cines, teatros...
 - Deportivos: participantes, clasificaciones, horarios...

Tras una reunión con personal ajeno a nuestro equipo de desarrollo, decidimos centrarnos en gestionar los eventos y las entradas para estos eventos, y dejar de lado todo lo relacionado con el transporte y alojamiento de los deportistas, árbitros y participantes en general.

También eliminamos la gestión de federaciones (inscripciones de participantes en competiciones etc.)

Eliminamos la publicidad y los pases de prensa.

También eliminamos la gestión de personal porque suponemos que todos esos servicios los subcontratamos.

Añadimos el concepto de ACTA como un informe que el árbitro rellenará al final de cada partido y que proporcionará información relativa a cada evento y que servirá para obtener la información de las competiciones.

En cuanto a los ROLES:

- El COI mantendría un rol de administración
- Hemos eliminado los perfiles de Prensa, Deportista y Autoridad pública, por lo que solo tendría un rol de “Control parcial” el usuario registrado (en la gestión de entradas) y el árbitro (en la gestión de actas)

En cuanto a la funcionalidad:

- Al implementar la interfaz hemos añadido opciones de consulta (noticias, última hora, fotos, etc.)

Conflictos encontrados:

- Antes de eliminar la gestión de transporte hubo discusiones acerca de introducir una clase gestora del transporte que realizara las tareas de reserva de recursos, rutas etc. Pero se descartó porque no se adaptaba a la filosofía de la orientación a objetos.

- En la gestión de entradas pasó algo parecido, teníamos un gestor de entradas que se dedicaba a vender, reservar etc. Decidimos delegar esa funcionalidad en el asiento y esto también generó polémica, pero decidimos dejarlo así.
- En la mitad del proyecto nos dimos cuenta de que estábamos abordando demasiados ámbitos y no nos iba a permitir realizar un buen trabajo con todo, así que decidimos quitar funcionalidades. Sopesamos la situación y decidimos quedarnos únicamente con las entradas, eventos y competiciones.
- En un principio trabajábamos con Bouml, pero daba problemas al trabajar con el SVN porque almacenaba todos los diagramas en un mismo fichero y esto daba conflictos al reconciliar las versiones. Probamos con ArgoUml pero el resultado era el mismo, así que al final nos decantamos por Eclipse, que al guardar cada diagrama en un fichero no nos daba ese problema. Aun así, no sabemos hacer los diagramas de secuencia con el Eclipse y los hemos hecho con ArgoUML.