

Olimpiadas

2mil

olimpiadas2020@googlegroups.com

Ingeniería de Software

Curso 2009 – 2010

Francisco Huertas

Gonzalo Ortiz

Alicia Pérez

Laura Reyero

Gabriela Ruiz

Rubén Tarancón

Introducción.....	5
Propósito	5
Análisis de requisitos	6
Especificación de cada componente	8
Diagramas	9
Diagrama de Casos de Uso	9
Observaciones	9
Diagrama de Casos de Uso de Evento	10
Observaciones	10
Diagrama de Clases.....	12
Observaciones	12
Diagrama de Secuencia Crear Acta	14
Observaciones	14
Diagrama de Secuencia Clasificación	15
Observaciones	15
Diagrama de Secuencia Consulta Horarios Personalizados	16
Observaciones	16
Diagrama de Secuencia Reserva	17
Observaciones	17
Diagrama de estado de las entradas.....	18
Observaciones	18
Diagrama de estado de los eventos	19
Observaciones	19
Diagrama de Actividad de Evento.....	20
Diagrama de Actividad de Entrada	21
Diagramas de Componentes	22
Observaciones	22

<i>Modelo de Datos</i>	<i>25</i>
DIAGRAMA ENTIDAD - RELACIÓN	25
DESCRIPCIÓN DE LAS TABLAS	25
Funciones de la base de datos.....	27
<i>Investigación de componentes utilizados.....</i>	<i>31</i>
COMPONENTE PARA LA INTERFAZ	31
COMPONENTE DE BASES DE DATOS	32
COMPONENTE DE BLOG.....	32
COMPONENTE DE PAGO	33
<i>Restricciones de Integridad: OCLPatrones.....</i>	<i>34</i>
<i>Patrones</i>	<i>41</i>
Patrones Creacionales: Patrón Singleton	41
Patrones Estructurales: Patrón Facade	41
Patrones de Comportamiento: Patrón Observer.....	42
Patrones de la Capa de Acceso a Datos.....	42
<i>Definición de Perfiles y Roles en la Aplicación.....</i>	<i>43</i>
Espectador	43
Árbitro	43
COI.....	43
Ayuntamiento	43
Administración Gestor de eventos	43
<i>Definición de las interfaces</i>	<i>45</i>
Aplicación Local.....	45
Interfaz de usuario.....	49
<i>Realización del Plan de Calidad.....</i>	<i>63</i>
Para qué.....	63
Cuándo.....	63
Cómo	63

No se debe	64
Importante.....	64
Flujo de actividades	64
“Log”	65
CUESTIONARIO OLIMPIADAS 2MIL (CMMI):.....	65
Evolución del proyecto	71

INTRODUCCIÓN

PROPÓSITO

El proyecto es una herramienta que ofrece una aplicación global para la gestión y administración de eventos deportivos de gran magnitud, como en este caso, las olimpiadas adjudicadas a su ciudad.

Esta solución engloba la gestión de todas las competiciones deportivas (instalaciones deportivas, horarios, participantes, la propia gestión de competición...), reserva y venta de entradas para los eventos programados y un portal que recoja la información deportiva.

ANÁLISIS DE REQUISITOS

- Un evento es un suceso programado para un deporte en el que participan dos o más equipos, en un lugar determinado. Por ello se hace necesario definir:
 - El emplazamiento en el que tendrá lugar, atendiendo a las restricciones impuestas por cada deporte (por ejemplo: Natación necesita un emplazamiento donde haya una piscina, Atletismo necesita uno con una pista de atletismo...)
 - El horario también juega un papel importante en el evento, ya que los eventos no se pueden solapar en el tiempo; ya sea porque un participante no pueda, como ser humano, encontrarse en dos lugares diferentes al mismo tiempo; como el hecho de que en una misma pista no se puedan estar realizando dos competiciones diferentes simultáneamente.
 - Las personas o equipos que compitan en el evento, deberán ser de la misma disciplina.
 - Los árbitros que regulen la competición del evento deberán ser de diferente nacionalidad a la de los deportistas que participan en ella.
 - Para cada evento, se pondrán a la venta un número variable de entradas para asistir como público, que dependerá del aforo del emplazamiento en el que va a tener lugar el evento. La venta de entradas se gestionará a través del portal web.
- Una competición es una rivalidad de quienes se disputan una cosa o la pretenden. Por ello se hace necesario definir:
 - Un cuadro de competición que regule cómo se van a distribuir los eventos atendiendo a la forma de juego de cada uno. Además contendrá información de los equipos rivales.
 - Un acceso para los responsables de cada equipo para poder inscribir a los suyos en las diferentes disciplinas.
- Para que un acontecimiento deportivo de tan gran magnitud tenga sentido, será necesario disponer de un portal web deportivo donde poder ir publicando la organización de los distintos eventos en el espacio y el tiempo, la clasificación de las diferentes disciplinas y el resultado de cada evento particular.
- La venta de entradas será llevada a cabo a través de Internet, donde, cada usuario podrá visualizar los próximos eventos, tanto el emplazamiento como el horario, así como el aforo disponible para cada uno.

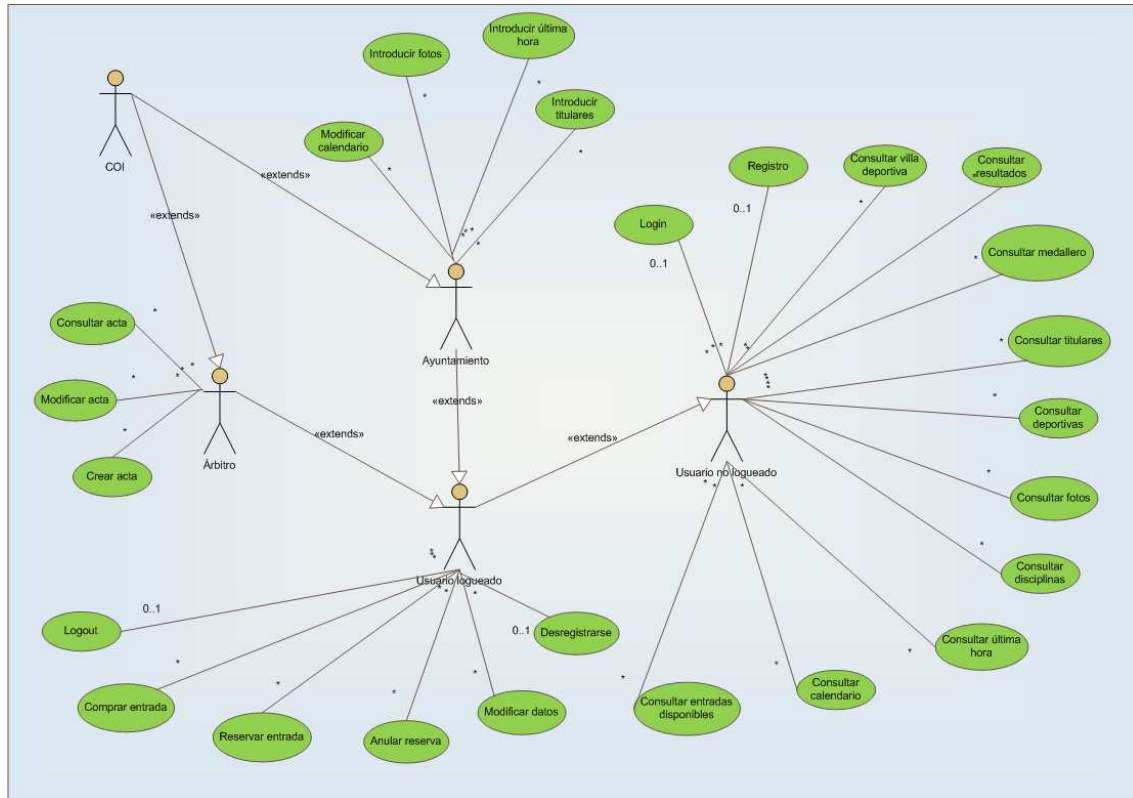
- Un equipo es cada uno grupo de personas que se disputan el triunfo.
- Todas las personas que formen parte del mismo equipo tienen que tener la misma nacionalidad.
- Todas las personas del equipo deberán participar en el mismo deporte.

ESPECIFICACIÓN DE CADA COMPONENTE

- Persona: ser humano que participa activamente en un evento. Pueden ser de dos tipos: deportista o árbitro.
- Deportista: persona que participa en el deporte que se realiza durante un evento. Pertenece a un equipo.
- Equipo: conjunto de personas que se asocian para participar en un deporte concreto.
- Arbitro: persona que regula el evento y que a posteriori elabora un acta de dicho evento.
- Competición: disputa entre dos o más equipos para conseguir la victoria, regulada por un árbitro.
- Comprador: usuario que desea asistir a un evento, adquiriendo una entrada.
- Entrada: billete asociado a los asientos de un emplazamiento.
- Evento: acontecimiento que tiene lugar en un emplazamiento y que acoge una o varias competiciones.
- Emplazamiento: instalación en la que se van a desarrollar los eventos. Tendrá que tener unas características acordes a las restricciones impuestas por cada deporte.
- Asiento: lugar físico que ocupara un comprador en el emplazamiento.
- Acta: memoria de todo lo acontecido durante un evento.
- Sanciones: posibles inhabilitaciones que aparecerán en el acta del evento. Son particulares para cada deportista.
- Resultado: desenlace del evento. Se refiere al equipo
- Lesiones: posibles agravios sufridos. Particular de cada deportista.

DIAGRAMAS

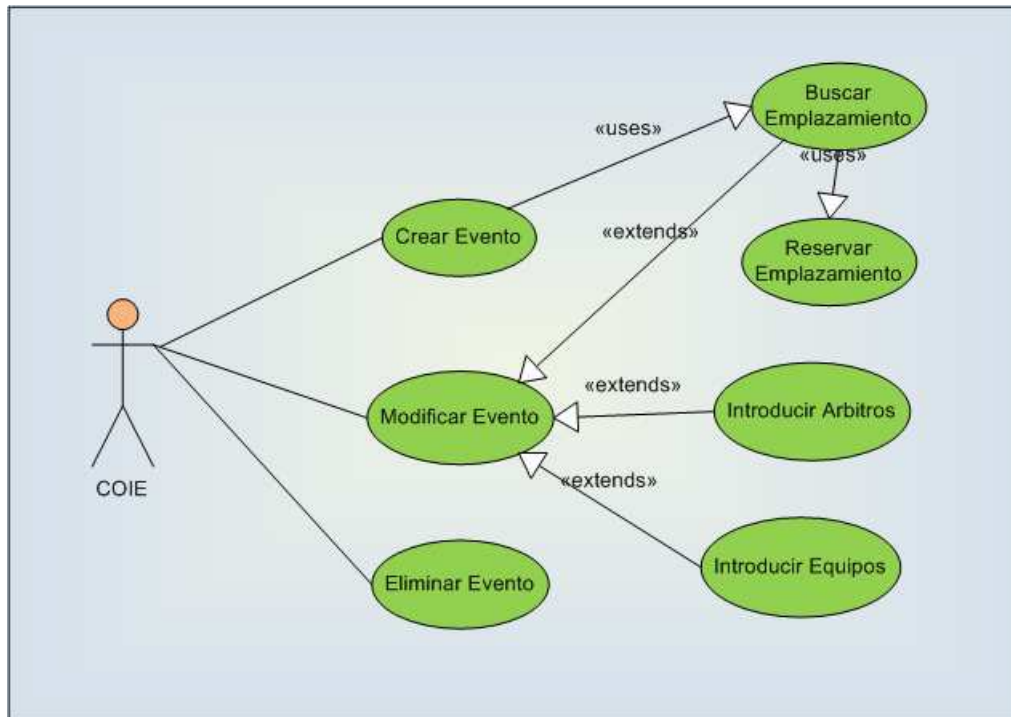
DIAGRAMA DE CASOS DE USO



OBSERVACIONES

Toda la herencia entre usuarios se explica en la definición de roles.

DIAGRAMA DE CASOS DE USO DE EVENTO



OBSERVACIONES

CasoDeUso 001	
Nombre	Buscar emplazamiento
Actores	COIE
Tipo	Primario y esencial
Descripción	Explora la BBDD de los emplazamientos, seleccionando alguno que cumpla las restricciones propias de la competición que se va a desarrollar en el evento.
Restricciones	Para iniciar la búsqueda del emplazamiento será necesario haber introducido los datos del evento
Objetivo	Seleccionar el emplazamiento asociado al evento.

CasoDeUso 002	
Nombre	Reservar emplazamiento
Actores	COIE
Tipo	Primario y esencial
Descripción	Bloqueo de las instalaciones para la realización del evento actual.
Restricciones	Para poder realizar una reserva es necesario encontrar, previamente, un emplazamiento
Objetivo	Restringir, a otros eventos, el uso de las instalaciones bloqueadas.

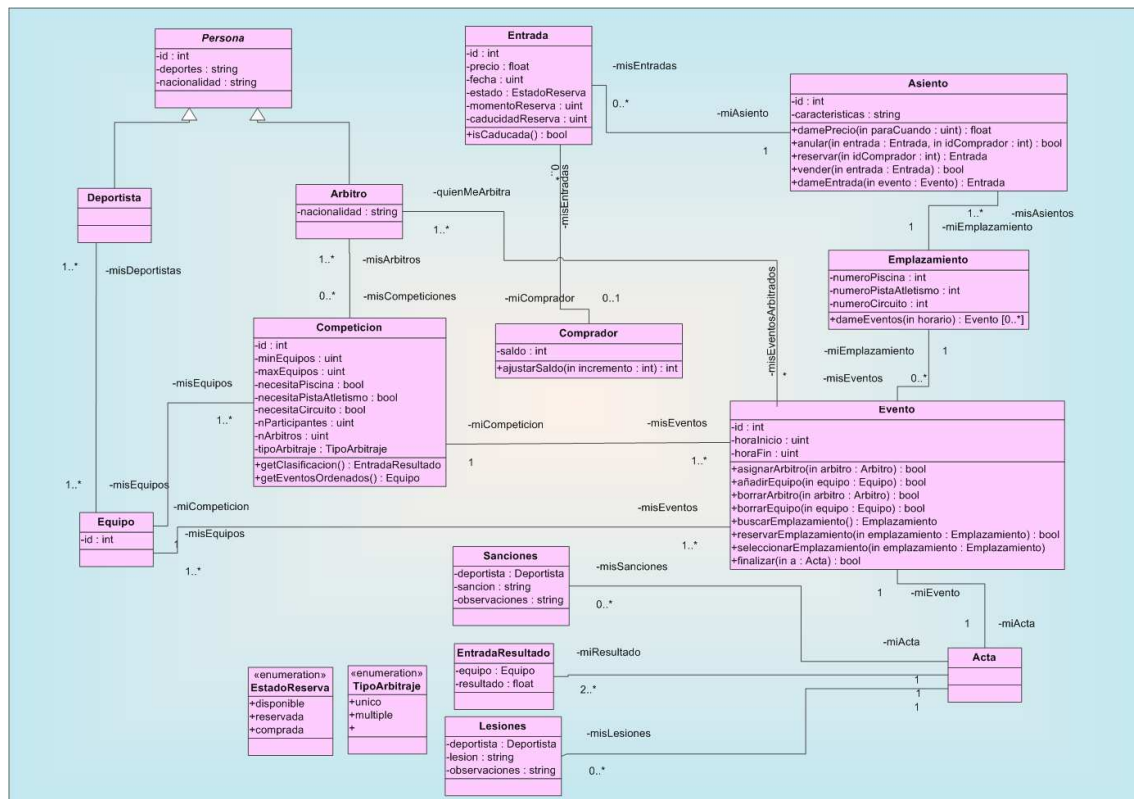
CasoDeUso 003	
Nombre	Introducir árbitros
Actores	COIE
Tipo	Secundario
Descripción	Seleccionar el número de árbitros necesarios al evento acorde a las restricciones de la actividad y de la nacionalidad.
Restricciones	Para poder asignar árbitros a un evento será necesaria su existencia
Objetivo	Asignación de árbitros al evento

CasoDeUso 004	
Nombre	Introducir equipos
Actores	COIE
Tipo	Secundario
Descripción	Seleccionar los equipos que participan en el evento tras la finalización del evento anterior según el cuadro de competición.
Restricciones	Para poder asignar los equipos a un evento será necesaria su existencia
Objetivo	Asignación de equipos al evento

CasoDeUso 005	
Nombre	Introducir fecha, horario y duración
Actores	COIE
Tipo	Primario y esencial
Descripción	Asignar un espacio de tiempo al evento, cumpliendo el cuadro de competición del deporte.
Restricciones	Para definir el espacio de tiempo del evento, es necesario haberlo identificado previamente
Objetivo	Definir la posición en el calendario

CasoDeUso 006	
Nombre	Introducir datos
Actores	COIE
Tipo	Primario y esencial
Descripción	Generar la identificación única del evento y dotar al mismo de las restricciones impuestas por la competición a la que pertenezca el evento.
Restricciones	N/A
Objetivo	Datos mínimos para la identificación del evento.

DIAGRAMA DE CLASES

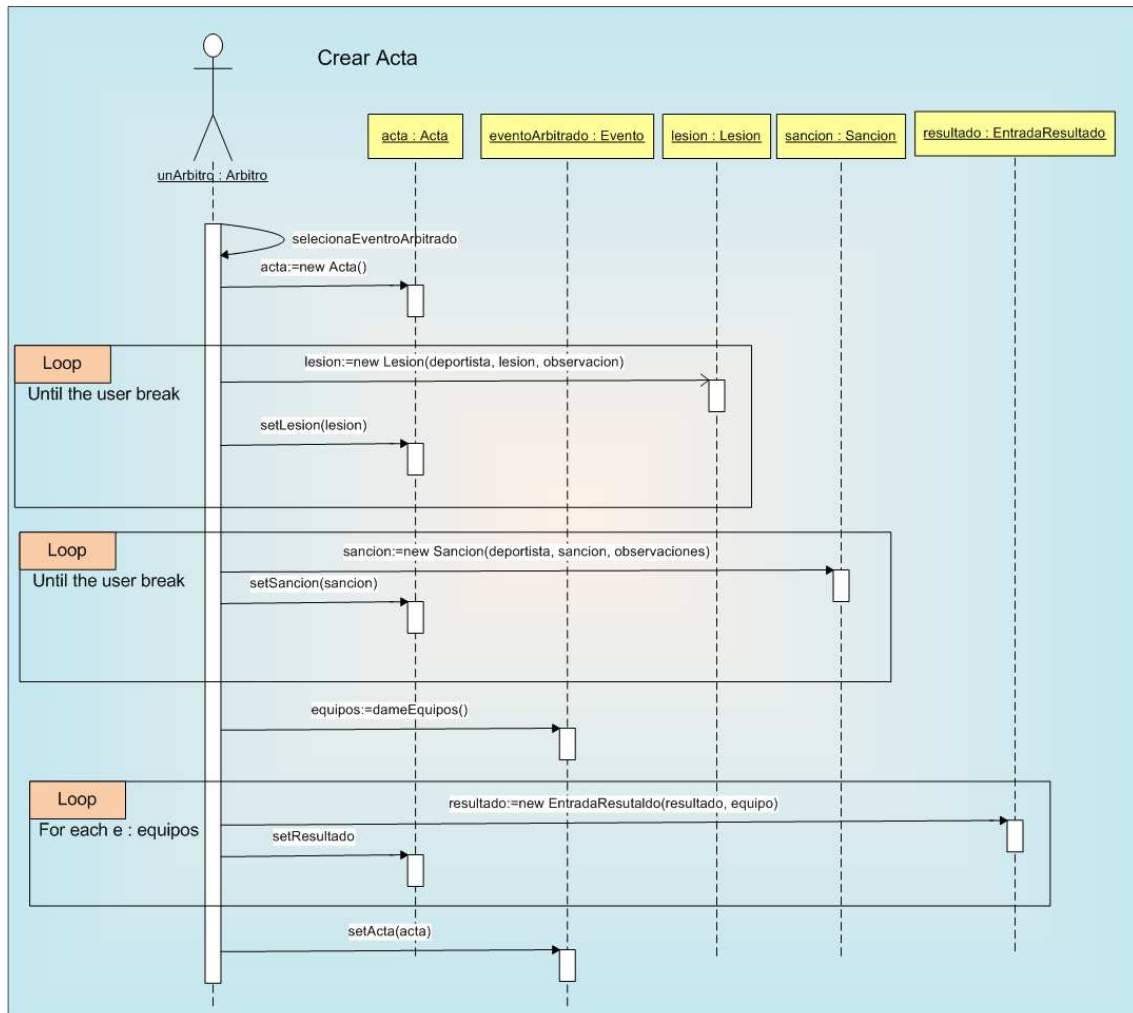


OBSERVACIONES

- Persona: ser humano que participa activamente en un evento. Pueden ser de dos tipos: deportista o árbitro.
- Deportista: persona que participa en el deporte que se realiza durante un evento. Pertenecer a un equipo.
- Equipo: conjunto de personas que se asocian para participar en un deporte concreto.
- Arbitro: persona que regula el evento y que a posteriori elabora un acta de dicho evento.
- Competición: disputa entre dos o más equipos para conseguir la victoria, regulada por un árbitro.
- Comprador: usuario que desea asistir a un evento, adquiriendo una entrada.
- Entrada: billete asociado a los asientos de un emplazamiento.

- Evento: acontecimiento que tiene lugar en un emplazamiento y que acoge una o varias competiciones.
- Emplazamiento: instalación en la que se van a desarrollar los eventos. Tendrá que tener unas características acordes a las restricciones impuestas por cada deporte.
- Asiento: lugar físico que ocupara un comprador en el emplazamiento.
- Acta: memoria de todo lo acontecido durante un evento.
- Sanciones: posibles inhabilitaciones que aparecerán en el acta del evento. Son particulares para cada deportista.
- Resultado: desenlace del evento. Se refiere al equipo
- Lesiones: posibles agravios sufridos. Particular de cada deportista.

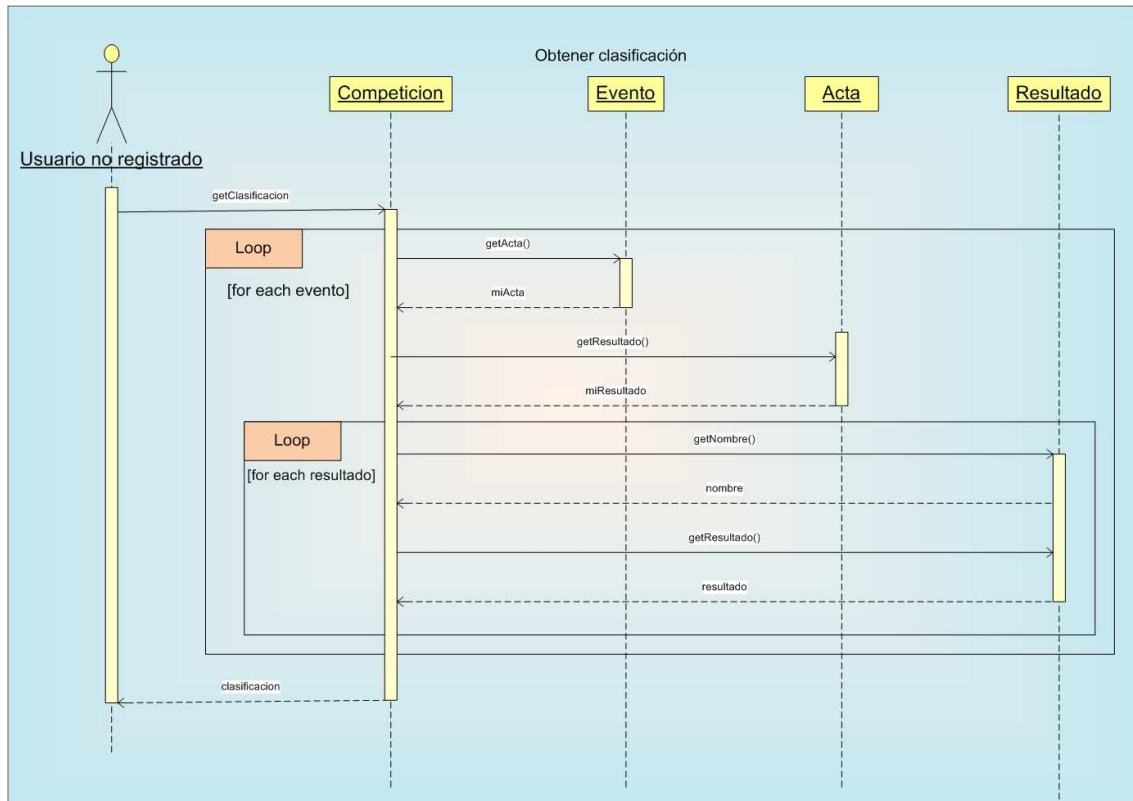
DIAGRAMA DE SECUENCIA CREAR ACTA



OBSERVACIONES

La creación del acta lleva implícito introducir todos los datos del acta, lesiones, sanciones y resultados; que una vez introducidos no se pueden modificar.

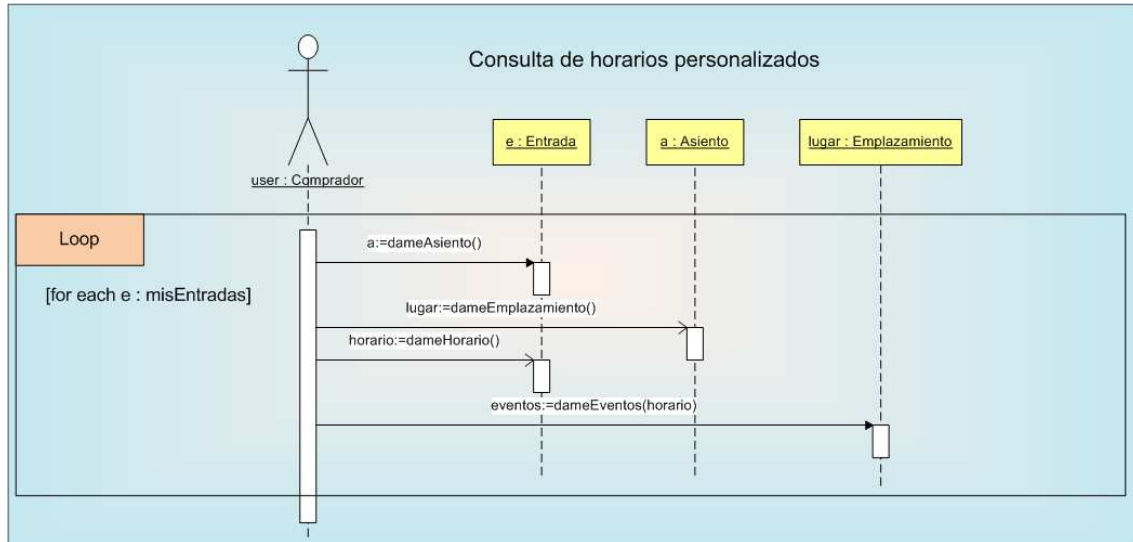
DIAGRAMA DE SECUENCIA CLASIFICACIÓN



OBSERVACIONES

Representa obtener todos los resultados de una competición. Se recorren todos los eventos de una clasificación, y analizando todas las actas, se obtienen todos los resultados y se clasifican, el usuario obtiene una lista de resultados ordenados.

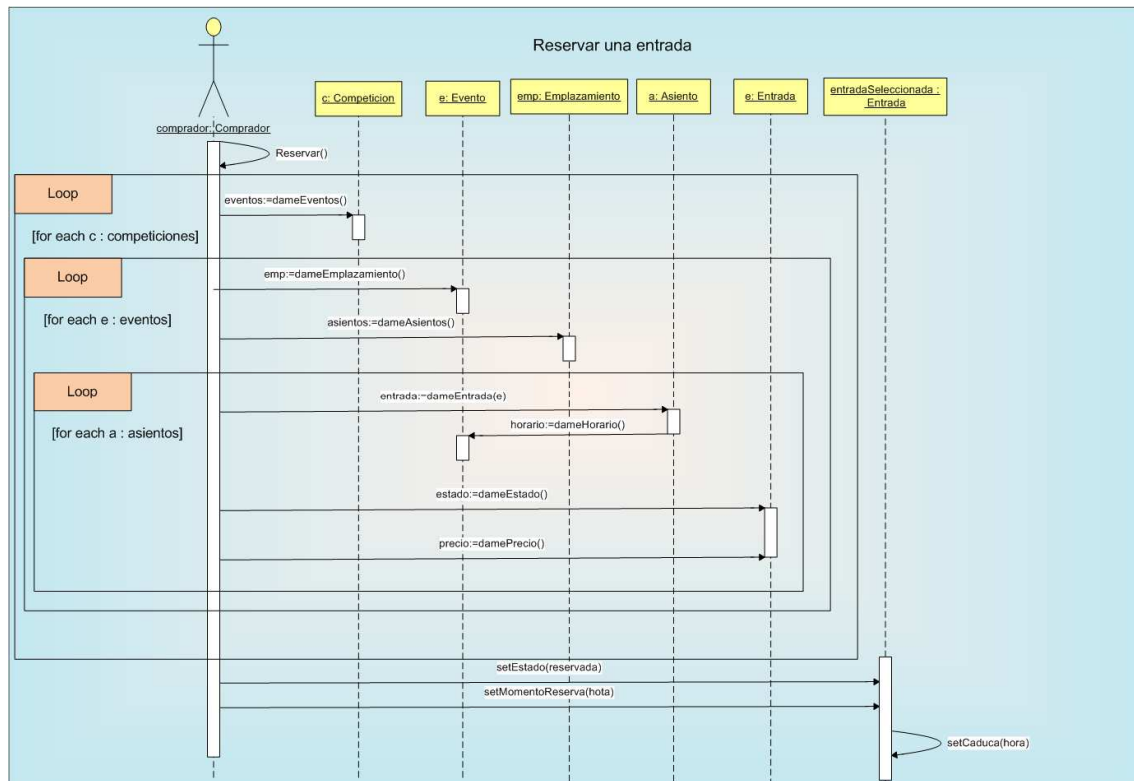
DIAGRAMA DE SECUENCIA CONSULTA HORARIOS PERSONALIZADOS



OBSERVACIONES

En la consulta del horario personalizado, después de seleccionar el asiento del cual quieres realizar la consulta, obtienes el emplazamiento. Tras ello consultas el horario de ese lugar. Se devuelve un calendario de los eventos.

DIAGRAMA DE SECUENCIA RESERVA



OBSERVACIONES

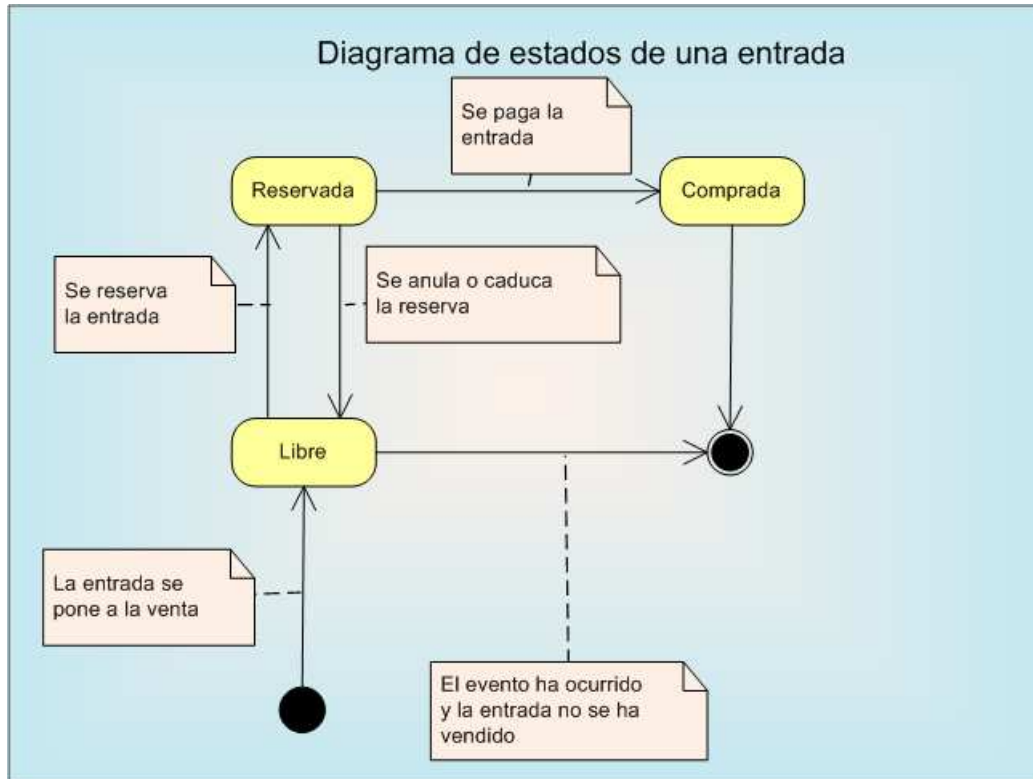
Este es el diagrama representativo del proceso de reservar una entrada

Primero es necesario obtener los eventos de cada competición así como el emplazamiento. Para cada emplazamiento el estado de ese asiento. En esta búsqueda habrá competiciones que repitan emplazamiento, y ese aspecto podría ser gestionado por el hilo principal.

Una vez que tenemos todos los Eventos, Emplazamiento y asientos, el usuario podrá seleccionar una entrada y esa entrada pasara cambiara a estado reservado.

Los diagramas de secuencia de anular entrada y comprar entrada, son triviales ya que solo se ven afectados dos componentes y una llamada entre ellos.

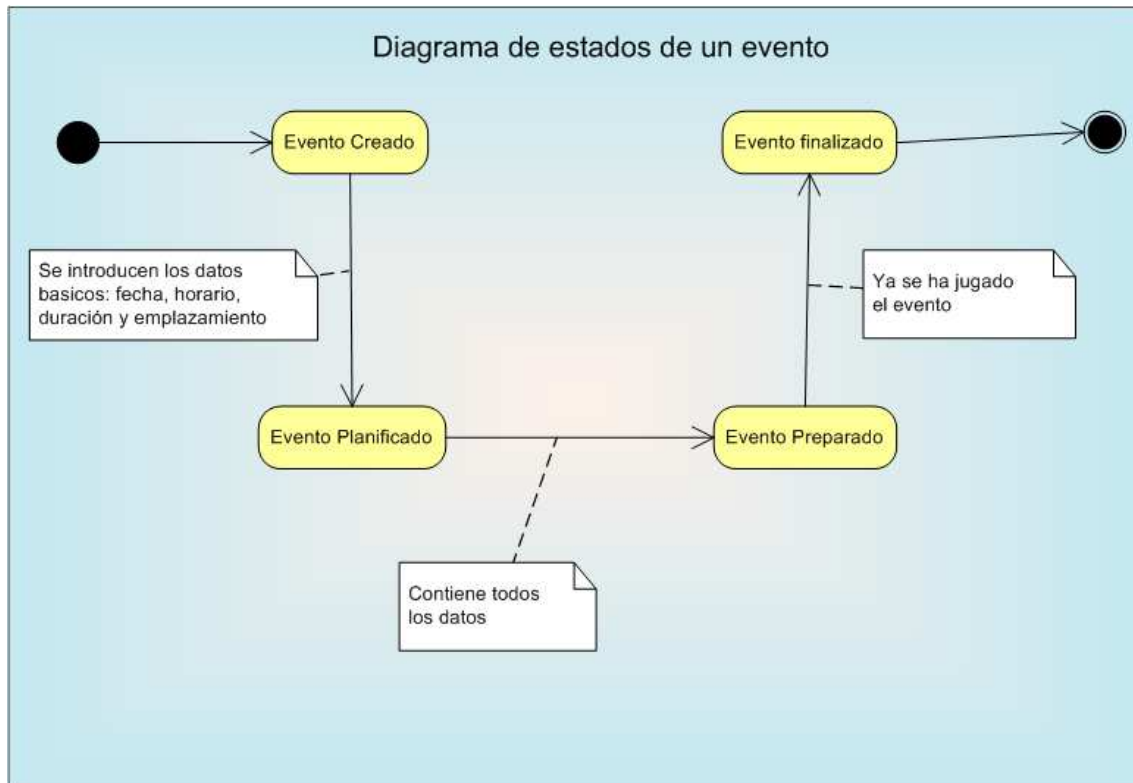
DIAGRAMA DE ESTADO DE LAS ENTRADAS



Observaciones

Representa los 3 estados en los que puede estar. Notar que no puede pasar de libre a comprada, si no que tiene que pasar antes por reservada. La transición de reservada a libre, puede ser por anulación o porque caduque la reserva, esta segunda transición es realizada por el sistema automáticamente.

DIAGRAMA DE ESTADO DE LOS EVENTOS



Observaciones

Los eventos pasan de un estado a otro de forma lineal, de tal manera que según se vayan efectuando las acciones requeridas, se realiza una transición; un evento, por tanto, nunca retrocede en los estados.

DIAGRAMA DE ACTIVIDAD DE EVENTO

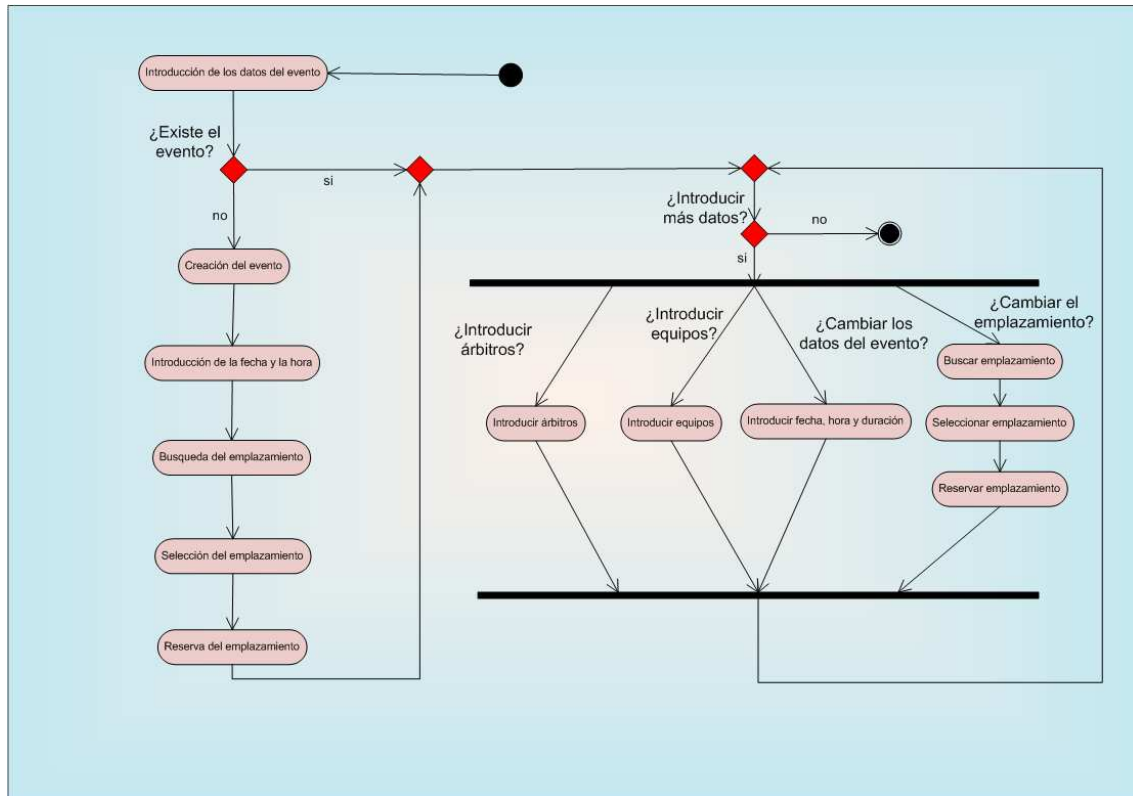
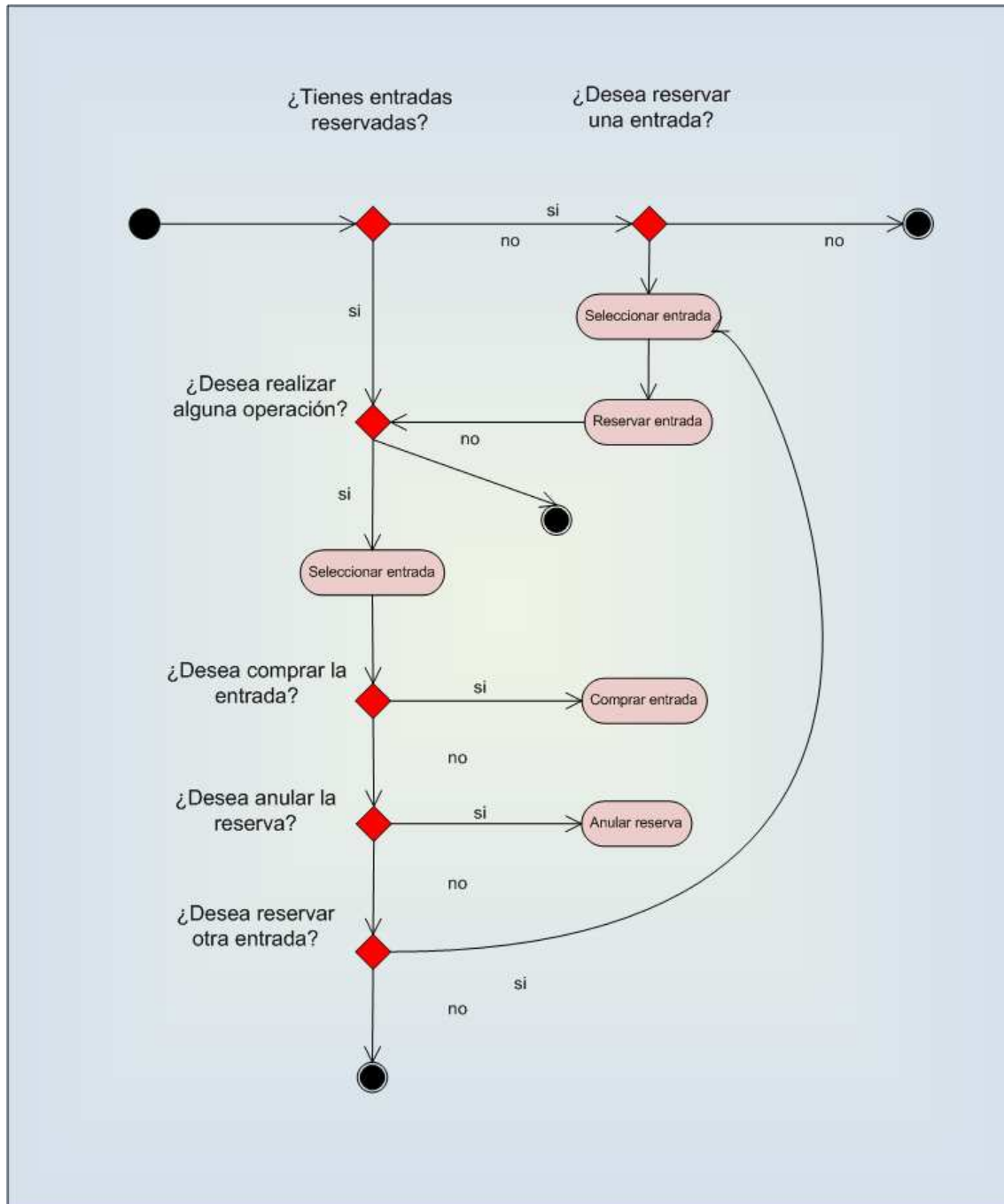
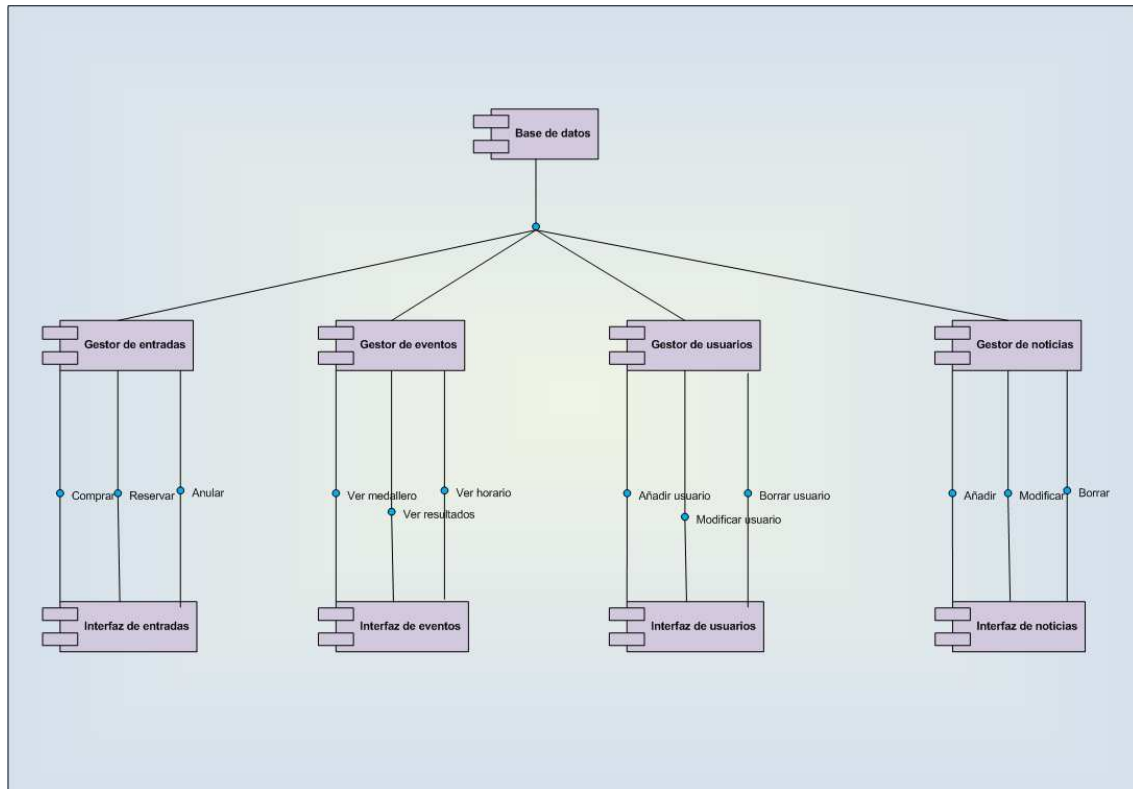


DIAGRAMA DE ACTIVIDAD DE ENTRADA



DIAGRAMAS DE COMPONENTES



OBSERVACIONES

■ 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.

■ 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 BBDD 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

■ GESTOR DE EVENTO

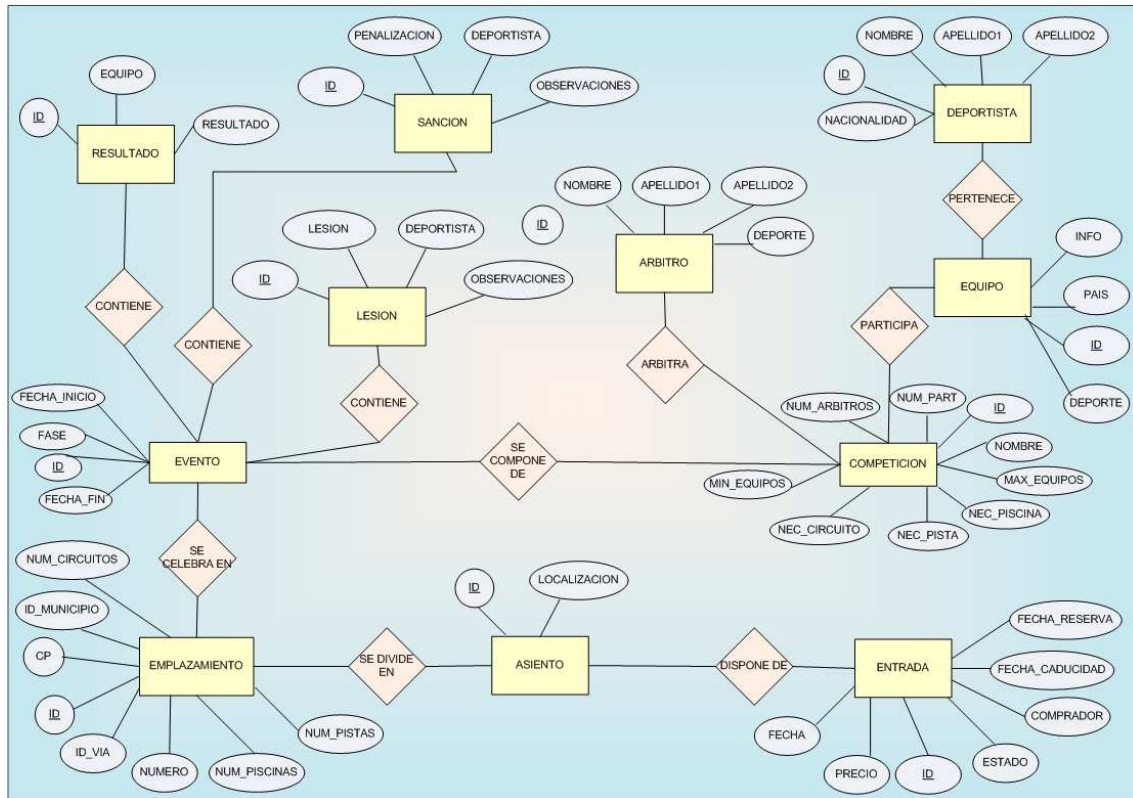
- Es sin duda, el gestor más importante de nuestra aplicación a la vez que el más sofisticado.
- Utilizará las tablas de la base de datos de deportes, equipos e instalaciones que le vendrán dadas por el COI y el ayuntamiento para realizar un organigrama de toda la competición. Gracias a él se construirá 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 actos 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.

MODELO DE DATOS

DIAGRAMA ENTIDAD - RELACIÓN



DESCRIPCIÓN DE LAS TABLAS

TABLA DE EQUIPO

- La tabla de equipos también nos vendrá impuesta por el COI 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 ÁRBITRO

- Es otra tabla que nos dará el COI y simplemente se guardará el nombre y los apellidos del árbitro y el deporte que va 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 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. 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 EVENTO

- La tabla de eventos va a recoger la información de cada uno de los partidos que se van a realizar en la olimpiada. Esta tabla nuestro sistema, a partir de las tablas de deportes, instalaciones y equipos la va a construir 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 para introducir las actas de los eventos. Se guardara el resultado y las incidencias que han ocurrido en el partido.

■ TABLA DE LESION

- Esta tabla va a ser utilizada por los árbitros para introducir en las actas de los eventos. Se guardara la lesión y el deportista que la sufre.

■ TABLA DE SANCION

- Esta tabla va a ser utilizada por los árbitros para introducir en las actas de los eventos. Se guardara la sanción y el deportista que ha sido sancionado y en que deporte.

FUNCIONES DE LA BASE DE DATOS

TABLA DE EQUIPOS

- Al igual que la tabla anterior, solo puede modificar los datos el COI, las operaciones que va a poder realizar son:

```
añadirEquipo( idE, idD, pais, inf)
pre:
  not(Equipo.filas -> exists(f1: fila |
    f1.idEquipo=idE and f1.idDeporte=idD))
post:
  Deporte.filas -> exists(f1:fila |
    f1.idEquipo = idE and
    f1.idDeporte = idD and
    f1.pais = pais and
    f1.informacion = inf)
```

```
eliminarEquipo(idE, idD)
pre:
  Deporte.filas -> exists(f1:fila |
    f1.idEquipo=idE and f1.idDeporte=idD)
post:
  not(Deporte.filas -> exists(f1:fila |
    f1.idEquipo=idE and f1.idDeporte=idD))
```

```
modificarDeporte(idE, idD, pais, inf)
pre:
  Deporte.filas -> exists(f1:fila |
    f1.idEquipo=idE and f1.idDeporte=idD)
post:
  Deporte.filas -> exists(f1:fila |
    f1.idEquipo = idE and
    f1.idDeporte = idD and
    f1.pais = pais and
    f1.informacion = inf)
```

TABLA DE ARBITRO

- Es la última de las tablas que va a estar gestionada por el 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 EMPLAZAMIENTO

- La tabla relativa al emplazamiento nos la proporcionara el ayuntamiento de la ciudad organizadora. 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->exits(f1:fila |
        f1.restriccion=rest)
```

■ TABLA DE ASIENTO

- Las operaciones que se puede realizar en esta tabla son:

```
verPrecio(idA, idI, faseC)
pre:
    Asiento.filas->exits( f1:filas |
        f1.idAsiento = idA and f1.idEmplazamiento = idI
post:
    return Asiento.f1.precio
```

■ TABLA DE ENTRADA

- Las operaciones de entradas son:

```
reservarEntrada(idA, idE, fecha, idU)
pre:
    not (Entradas.filas->exits(f1:filas |
        f1.idAsiento = idA and
        f1.idEmplazamiento = IdE and
        f1.fecha=fecha))
post
    Entradas.filas->exits(f1:filas |
        f1.idAsiento = idA and
        f1.idEmplazamiento = IdE and
        f1.fecha=fecha and
        f1.esReservada=true and
        f1.idUsuario = idU)
```

```

comprarEntrada(idA,idE,fecha)
pre:
  Entradas.filas->exits(f1:filas |
    f1.idAsiento = idA and
    f1.idEmplazamiento = IdE and
    f1.fecha=fecha and
    f1.esReservada=true)
post:
  Entradas.filas->exits(f1:filas |
    f1.idAsiento = idA and
    f1. idEmplazamiento = IdE and
    f1.fecha=fecha and
    f1.esReservada=false)

```

```

anularEntrada(idA,idI,fecha)
pre:
  Entradas.filas->exits(f1:filas |
    f1.idAsiento = idA and
    f1. idEmplazamiento = IdE and
    f1.fecha=fecha)
post:
  not(Entradas.filas->exits(f1:filas |
    f1.idAsiento = idA and
    f1. idEmplazamiento = IdE and
    f1.fecha=fecha))

```

■ 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)

```

```

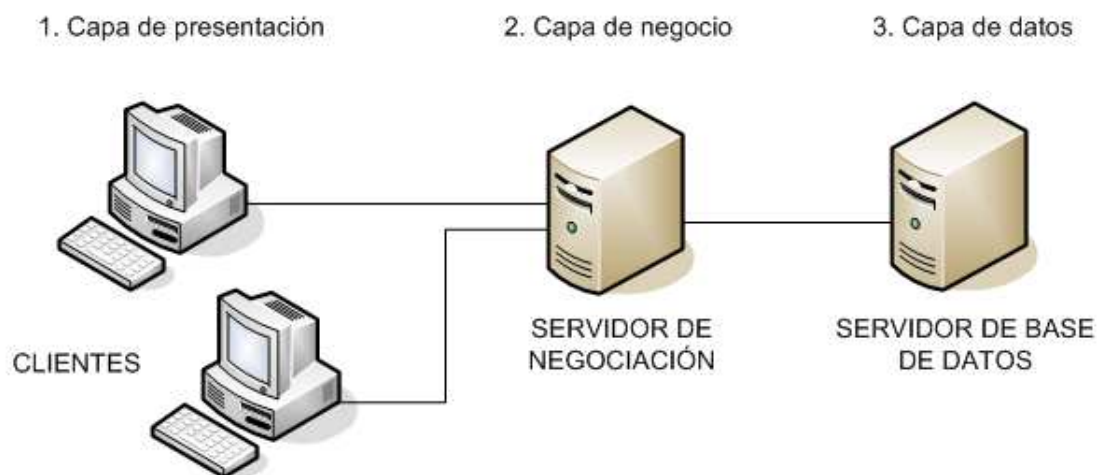
modificarActa(idE, res, obs)
pre:
  Resutados.filas->exits(f1:filas |
    f1.idEvento = idE)
post:
  Resutados.filas->exits(f1:filas |
    f1.idEvento = idE and
    f1.resultado = res and
    f1.observaciones = obs)

```

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

INVESTIGACIÓN DE COMPONENTES UTILIZADOS

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.



COMPONENTE 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/>

COMPONENTE DE BASES 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:

- Oracle es la BD más utilizada por las corporaciones más grandes del mundo, por su robustez y por la seguridad.
- Oracle tiene múltiples versiones para cada sistema operativo, entre ellos te puedo nombrar: UNIX, LINUX, NT, Win 2000- 98 - 95, Novel, y otros.
- Oracle ofrece soporte mundial a través de sus centros de soporte y sus sitios Web donde podrás encontrar desde scripts hasta documentos de instalación.
- Proporcionan 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 Dato Oracle. Oracle es más que un manejador de BD. Oracle corporation ofrece otras soluciones a la plataforma de Negocio, EBusiness, Ecommerce, Etc.
- Puedes programar con las últimas herramientas del mercado y utilizando como BD Oracle.
- Estas asegurado con el soporte.

COMPONENTE DE BLOG

Nuestra página web posee una parte de noticias, donde se actualizarán las novedades que vayan surgiendo.

Para esta parte nos hemos decantado por wordpress, ya que solo con instalarlo tienes acceso a los blogs. Para instalar el componente blog en nuestra página web lo hacemos incorporando wordpress a nuestra página en php. La forma de hacerlo es bastante sencilla. Se puede ver un video en el siguiente enlace:

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

En el propio blog vamos a incluir 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/>

COMPONENTE DE PAGO

PayPal es un sistema que permite a cualquier persona que tenga una dirección de e-mail 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>

RESTRICCIONES DE INTEGRIDAD: OCL

INVARIANTES

El número de jugadores de un equipo estará comprendido entre un nº mínimo y un nº máximo

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

Si una competición tiene un único árbitro la nacionalidad del arbitro debe ser distinta de la de alguno 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, no pueden compartir nacionalidad entre ellos y por tanto a lo sumo solo hay un árbitro de la misma nacionalidad que algunos de los equipos

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

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)
```

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 pueden estar en dos eventos que se solapen

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

Un mismo árbitro no pueden estar en dos eventos que se solapen

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

Un jugador no puede estar en dos equipos que se solapen

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

Para todo evento su emplazamiento cumple las restricciones de la competicion

```
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
```

Ningun asiento tiene dos reservas en el mismo momento

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

```
// Hay tantos participantes y no mas de los que requiere la competición
Context Evento inv:
  self.misEquipos->size() = self.miCompeticion.nEquiposPorEvento
```

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

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

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

```
Context Acta inv:
  self.resultado.equipos = self.miEvento.misEquipos
```

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

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

Las competencias de dos equipos por evento tienen una final y una de consolacion (respectivamente el ultimo y penultimo evento de esa competicion). 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()
```

METODOS

```
context Asiento::reservar(idComprador : Integer, 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::disponible  and
    result.miComprador->notEmpty()                   and
    self.misEntradas->includes(result)
```

```
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 Entrada::reservar(c:Comprador) : Boolean
post:
    if self.estado@pre = EstadoReserva::disponible
    then
        self.estado = EstadoReserva::reservada    and
        self.miComprador = c                      and
        result = true
    else
        result = false
```

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

```

Context Entrada::anular(c : Comprador) : Boolean
post:
  if
    self.estado@pre = EstadoReserva::reservada      and
    c = self.miComprador
  then
    result          = true                          and
    self.estado     = EstadoReserva::disponible     and
    self.miComprador->isEmpty()                      and
  else
    result          = 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^reservarEmplazamiento(self)                       and
      result = true
    else
      result = false
    endif
  endif
endif

```

```

Context Acta::getResultado(e:Evento) : 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
```

FUCIONES 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)
```


PATRONES

PATRONES CREACIONALES: PATRÓN SINGLETON

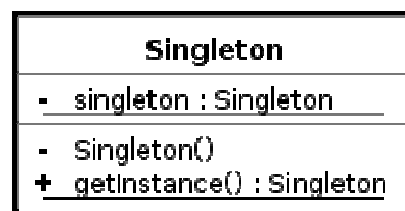
El objeto de este patrón es garantizar que solo existe una instancia de una clase y proporcionar un punto de acceso global a esta instancia.

La implementación es muy sencilla, se compone de una variable del mismo tipo que la clase y un método que crea una instancia si dicha variable es nula.

Las situaciones más habituales de aplicación de este patrón son aquellas en las que dicha clase controla el acceso a un recurso físico único (como puede ser el ratón o un archivo abierto en modo exclusivo) o cuando cierto tipo de datos debe estar disponible para todos los demás objetos de la aplicación.

En las partes multihilo hay que tener especial cuidado y proporcionar exclusión mutua en la creación de la instancia.

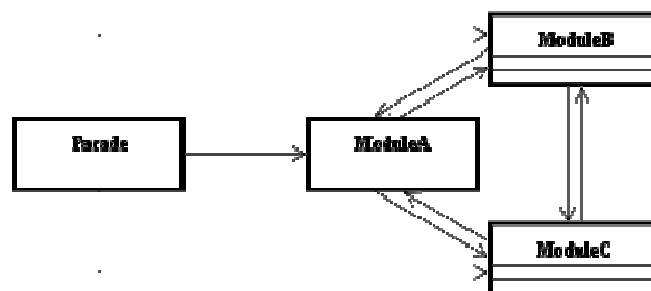
<http://es.wikipedia.org/wiki/Singleton>



PATRONES ESTRUCTURALES: PATRÓN FACADE

Es un patrón de diseño que provee de una interfaz entre el cliente y una interfaz o conjunto de interfaces más complejas. Puede ser útil a la hora de comunicar la capa de negocio con la capa de acceso a datos (DAOs)

[http://es.wikipedia.org/wiki/Facade_\(patrón_de_diseño\)](http://es.wikipedia.org/wiki/Facade_(patrón_de_diseño))



PATRONES DE COMPORTAMIENTO: PATRÓN OBSERVER

El flujo de trabajo en la interfaz irá controlado por el patrón observer.

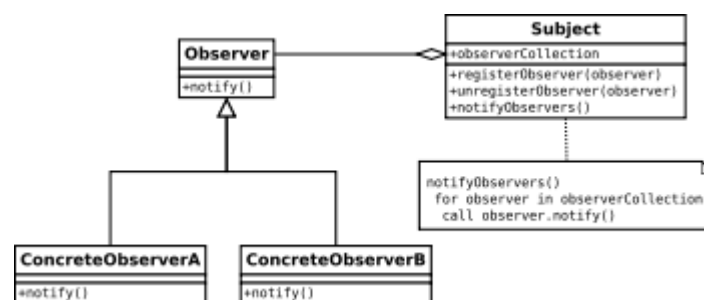
El objetivo de este patrón es desacoplar la clase de los objetos clientes del objeto, aumentando la modularidad del lenguaje, así como evitar bucles de actualización.

La clase "observada" define una serie de eventos que lanzará cuando ocurra determinada acción. Cualquier clase se podrá suscribir como observadora, implementando los eventos que desee observar, de modo que cuando una acción dispare el evento en la clase observada la clase observadora ejecutará su método.

Este patrón suele observarse en los frameworks de interfaces gráficas orientados a objetos, en los que la forma de capturar los eventos es suscribir 'listeners' a los objetos que pueden disparar eventos.

Más información:

[http://es.wikipedia.org/wiki/Observer_\(patrón_de_diseño\)](http://es.wikipedia.org/wiki/Observer_(patrón_de_diseño))



PATRONES DE LA CAPA DE ACCESO A DATOS

Un Data Access Object (DAO, Objeto de Acceso a Datos) es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.

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.

Más información:

http://es.wikipedia.org/wiki/Data_Access_Object

DEFINICIÓN DE PERFILES Y ROLES EN LA APLICACIÓN

ESPECTADOR

■ Funciones:

Suponemos que existe disponible para todos los usuarios una web donde se visualiza los datos de los eventos (participantes, emplazamiento, horario,...) y a su vez, en esta web existe una sección de venta de entradas

- Visualización de la información de los eventos.
- Compra, reserva y anulación de entradas

Este perfil tiene incorporado el perfil de Deportistas que participan en los eventos, ya que requieren el mismo grado de información.

ÁRBITRO

■ Funciones:

- Crea y modifica las actas de los eventos que arbitre.

COI

■ Funciones:

Suponemos que este perfil es el encargado de proveer, a la administración del evento, los participantes (COI). Así como también, puede realizar las operaciones propias de un árbitro.

AYUNTAMIENTO

■ Funciones:

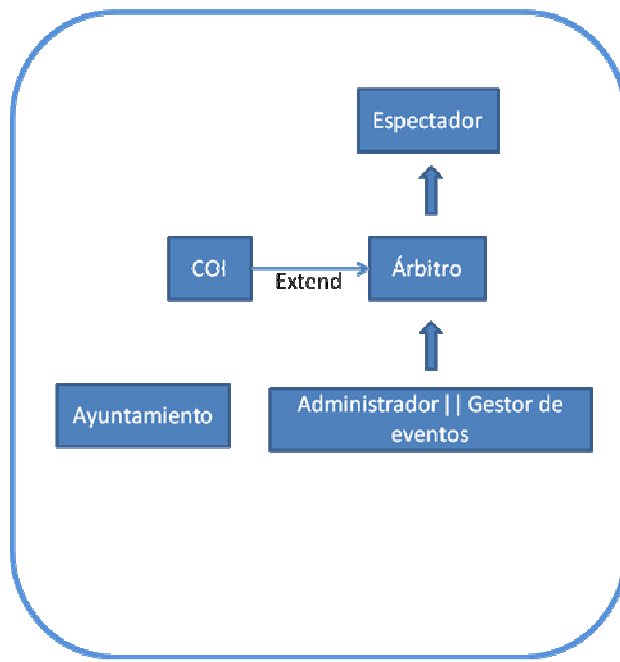
Suponemos que este perfil es el encargado de proveer, a la administración del evento, los datos de los emplazamientos (Ayuntamientos).

ADMINISTRACIÓN || GESTOR DE EVENTOS

■ Funciones:

- Crea, modifica y elimina eventos.

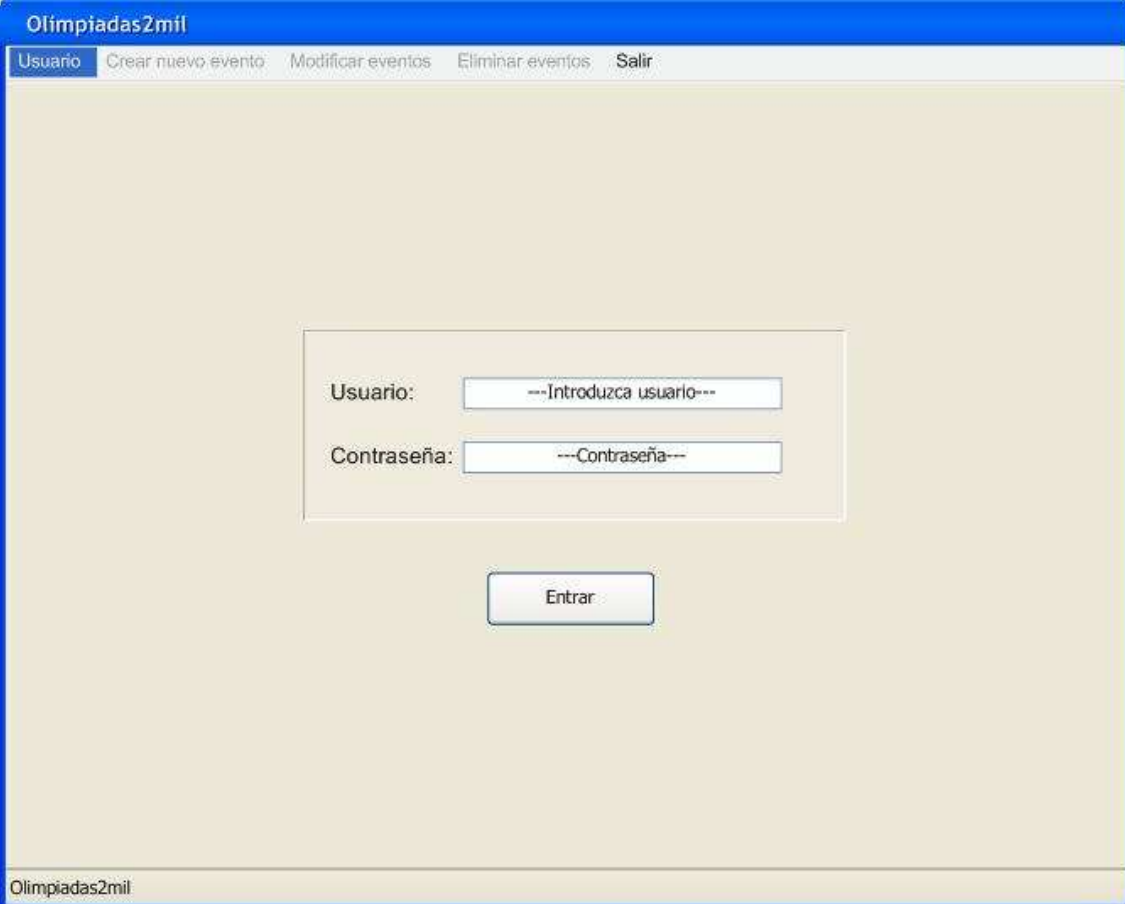
Todos los perfiles incorporan las funcionalidades de los perfiles inferiores.



DEFINICIÓN DE LAS INTERFACES

APLICACIÓN LOCAL

- Registro del usuario:



The screenshot displays the 'Olimpiadas2mil' application window. The title bar is blue and contains the text 'Olimpiadas2mil'. Below the title bar is a menu bar with five items: 'Usuario' (highlighted in blue), 'Crear nuevo evento', 'Modificar eventos', 'Eliminar eventos', and 'Salir'. The main area of the window has a light beige background. In the center, there is a login form with two input fields: 'Usuario:' with a dropdown menu showing '---Introduzca usuario---' and 'Contraseña:' with a dropdown menu showing '---Contraseña---'. Below these fields is a button labeled 'Entrar'. At the bottom left of the window, the text 'Olimpiadas2mil' is displayed.

■ Creación de un nuevo evento:

Olimpiadas2mil

Usuario **Crear nuevo evento** Modificar eventos Eliminar eventos Salir

Titulo del evento	<input type="text"/>	Competición	<input type="text" value="Competición"/>
Fecha del evento	<input type="text" value="Seleccione fecha"/>	Hora de inicio	<input type="text" value="Hora de inicio del evento"/>
		Hora de fin	<input type="text" value="Hora de finalización del evento"/>
Emplazamiento	<input type="text" value="Seleccione emplazamiento"/> <div><div>Emplazamientos organizados según su prioridad</div></div>	Equipos participantes	<input type="text" value="Seleccionar equipos (Ctrl presionado)"/> <div></div>
	<input type="button" value="Reservar emplazamiento seleccionado"/>	Árbitros participantes	<input type="text" value="Seleccionar árbitros (Ctrl presionado)"/> <div></div>

Olimpiadas2mil

- Modificación de los datos de un evento:

The screenshot shows the 'Olimpiadas2mil' application window. The title bar is blue with the text 'Olimpiadas2mil'. Below the title bar is a menu bar with the following options: 'Usuario', 'Crear nuevo evento', 'Modificar eventos' (highlighted in blue), 'Eliminar eventos', and 'Salir'. The main content area is divided into two sections. The left section contains two dropdown menus: 'Competición' (with 'Competición' selected) and 'Evento' (with 'Evento' selected). Below these is a button labeled 'Modificar evento'. The right section contains several fields and buttons: 'Fecha' (with a dropdown menu showing 'Fecha'), 'Hora de inicio' (with a dropdown menu showing 'Hora de inicio'), and 'Hora de fin' (with a dropdown menu showing 'Hora de finalización'). Below these is a section for 'Emplazamiento' with a dropdown menu showing 'Seleccione emplazamiento' and a list box. Below the list box is a button labeled 'Reservar emplazamiento seleccionado'. The next section is 'Equipos participantes' with a dropdown menu showing 'Seleccionar equipos (Ctrl presionado)' and a list box. Below the list box is a button labeled 'Reservar equipos seleccionados'. The final section is 'Árbitros participantes' with a dropdown menu showing 'Seleccionar árbitros (Ctrl presionado)' and a list box. The bottom of the window has a status bar with the text 'Olimpiadas2mil'.

Olimpiadas2mil

Usuario Crear nuevo evento **Modificar eventos** Eliminar eventos Salir

Competición Competición

Evento Evento

Modificar evento

Fecha Fecha

Hora de inicio Hora de inicio

Hora de fin Hora de finalización

Emplazamiento Seleccione emplazamiento

Reservar emplazamiento seleccionado

Equipos participantes Seleccionar equipos (Ctrl presionado)

Árbitros participantes Seleccionar árbitros (Ctrl presionado)

Olimpiadas2mil

■ Eliminación de un evento:

The screenshot shows the 'Olimpiadas2mil' application window. The title bar is blue with the text 'Olimpiadas2mil'. Below the title bar is a menu bar with the following options: 'Usuario', 'Crear nuevo evento', 'Modificar eventos', 'Eliminar eventos' (which is highlighted in blue), and 'Salir'. The main area of the window is divided into two sections. On the left, there is a box containing two dropdown menus: 'Competición' (with 'Competición' selected) and 'Evento' (with 'Evento' selected). Below these is a button labeled 'Eliminar evento'. On the right, there are several input fields and dropdown menus: 'Fecha' (with a 'Fecha' dropdown), 'Hora de inicio' (with a 'Hora de inicio' dropdown), 'Hora de fin' (with a 'Hora de finalización' dropdown), 'Emplazamiento' (with a 'Seleccione emplazamiento:' dropdown), 'Equipos participantes' (with a 'Seleccionar equipos (Ctrl presionado)' dropdown), and 'Árbitros participantes' (with a 'Seleccionar árbitros (Ctrl presionado)' dropdown). A mouse cursor is pointing at the 'Árbitros participantes' dropdown, which has a tooltip that reads: 'Árbitros que no participen En otro evento con las mismas especificaciones'. The bottom of the window has a status bar with the text 'Olimpiadas2mil'.

INTERFAZ DE USUARIO

■ Gestión de usuarios

Gracias a la interfaz conseguimos que los usuarios que lo deseen sean registrados en nuestra aplicación de una forma sencilla e intuitiva.

Si todavía no eres usuario registrado y quieres acceder a las ventajas de la aplicación debes pinchar en "registro" en el menú de la izquierda, rellenar los datos necesarios, aceptar las condiciones de privacidad y pinchar el botón "enviar".

The image shows a web application interface for user registration. On the left is a sidebar menu with four sections: 'Usuarios' (containing 'Registro' and 'Login'), 'Noticias' (containing 'Titulares', 'Última hora', 'Deportivas', and 'Galería de fotos'), 'Olimpiadas2020' (containing 'Disciplinas', 'Calendario', 'Resultados', 'Medallas', and 'Distribución de la Villa deportiva'), and 'Entradas' (containing 'Calendario', 'Ubicación de instalaciones', and 'Gestión de entradas'). The 'Registro' option is highlighted. The main area is titled 'Alta de usuario' and contains a registration form with the following fields: ID Usuario, Contraseña, Repetir contraseña, DNI/NIF/..., Nombre, Apellidos, Dirección, Provincia, País, Email, Movil, Profesion, Sexo (with a dropdown showing 'M/F'), and Observaciones. Below the form are two checkboxes: 'He leído y acepto las condiciones de privacidad. Leer' and 'Deseo recibir en mi correo las últimas noticias.' At the bottom are two buttons: 'Limpiar' and 'Enviar'.

Alta de usuario			
ID Usuario	<input type="text"/>	País	<input type="text"/>
Contraseña	<input type="text"/>	Email	<input type="text"/>
Repetir contraseña	<input type="text"/>	Movil	<input type="text"/>
DNI/NIF/...	<input type="text"/>	Profesion	<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"/>		

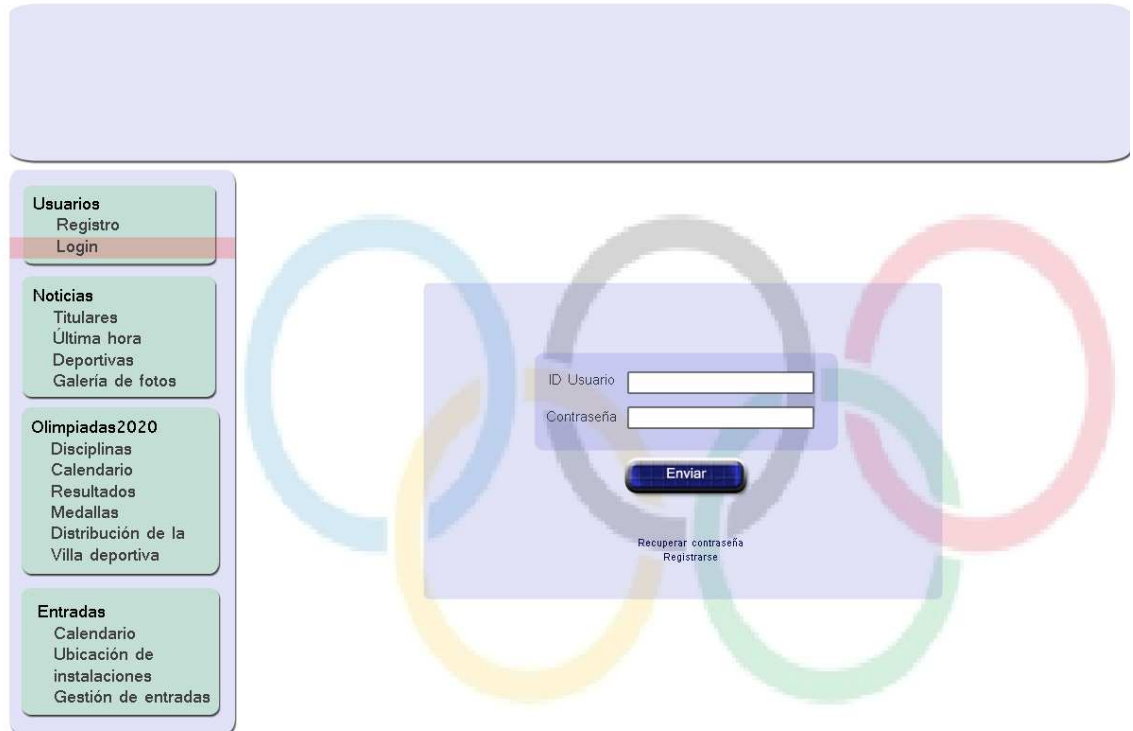
☒ He leído y acepto las condiciones de privacidad. [Leer](#)

☒ Deseo recibir en mi correo las últimas noticias.

Si todo está conforme:

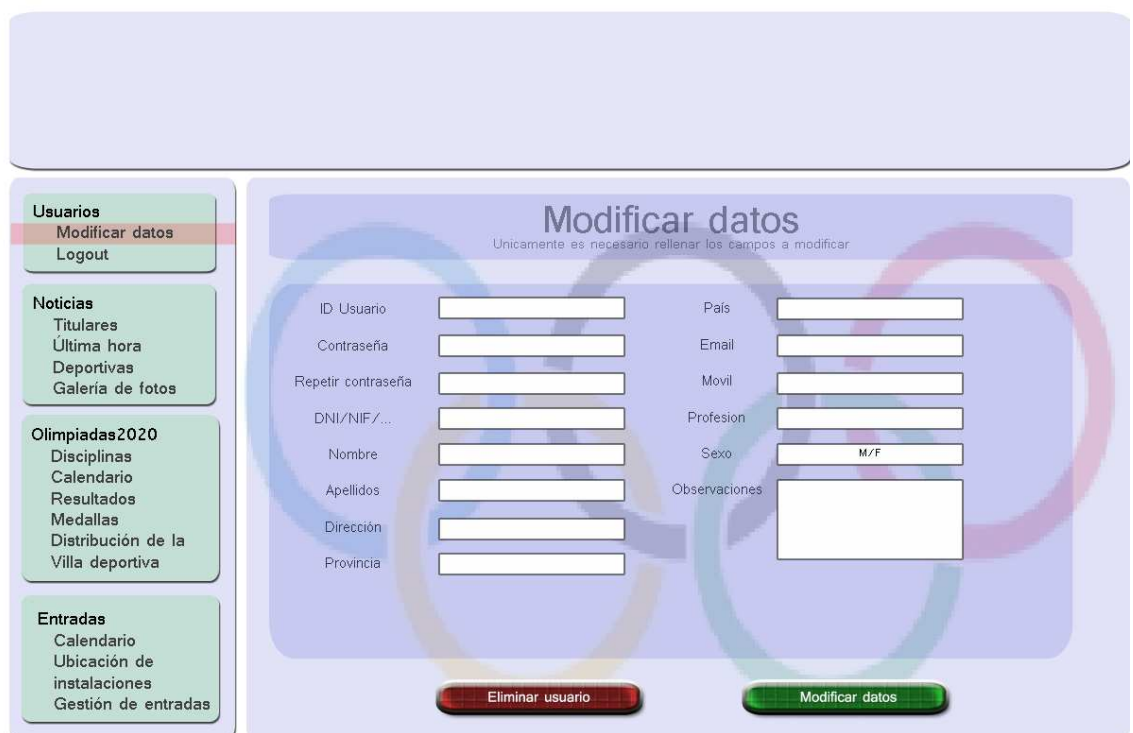


Si por el contrario ya eres usuario registrado simplemente tendrás que pinchar “login” del menú de la izquierda, a continuación rellenas el “usuario” y la “contraseña” y pulsas el boton “Enviar”. A partir de entonces podras acceder a las ventajas del sistema, cuanto mayor sea tu importancia en la aplicación, de mayores utilidades dispones.



The image shows a login form interface. On the left, there is a sidebar menu with four categories: 'Usuarios' (containing 'Registro' and 'Login'), 'Noticias' (containing 'Titulares', 'Última hora', 'Deportivas', and 'Galería de fotos'), 'Olimpiadas2020' (containing 'Disciplinas', 'Calendario', 'Resultados', 'Medallas', and 'Distribución de la Villa deportiva'), and 'Entradas' (containing 'Calendario', 'Ubicación de instalaciones', and 'Gestión de entradas'). The 'Login' option is highlighted. The main content area features a large, faint background image of the Olympic rings. Overlaid on this is a login form with two input fields: 'ID Usuario' and 'Contraseña'. Below these fields is a blue 'Enviar' button. At the bottom of the form, there are two links: 'Recuperar contraseña' and 'Registrarse'.

Si se quieren modificar algunos datos:



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

Una vez que has accedido al sistema identificandote con tu usuario y contraseña, en cualquier momento puedes desconectarte pinchando "logout" en la parte izquierda de la pantalla.



■ Gestión de eventos

Como se puede apreciar en la imagen inferior la interfaz de eventos tiene varias opciones a las que puedes acceder. Una de ellas es visualizar las disciplinas, para ello hay que hacer click en "disciplinas" y pinchar en el deporte olímpico que desees en la parte de arriba.

Usuarios

Registro

Login

Noticias

Titulares

Última hora

Deportivas

Galería de fotos

Olimpiadas2020

Disciplinas

Calendario

Resultados

Medallas

Distribución de la Villa deportiva

Entradas

Calendario

Ubicación de instalaciones

Gestión de entradas

Atletismo

Baloncesto

Boxeo

Ciclismo

Erguimento

Gimnasia

Hockey

Lucha libre

Pentatlón

Softbol

Tenis

Tiro con arco

Vela

Badminton

Basketbol

Canotaje

Equitación

Futbol

Gimnasia artística

Judo

Natación

Reno

Tae Kwon Do

Tennis mesa

Triatlon

Voleibol

ATLETISMO

Atletismo - Deporte Olímpico desde 1896

El Atletismo es de alguna forma, la personificación del Lema Olímpico "Citius, Altius, Fortius", el cual quiere decir; más alto, más rápido, más fuerte.

El atletismo (campo y pista) se trata de correr más rápido, saltar alto y lanzar objetos más lejos, que tu competidor. Los juegos en la antigua Grecia comenzaron con una simple carrera a campo traviesa, que posteriormente fueron evolucionando hasta lo que es hoy en día.

10000m masculino	100m masculino	110m obstáculos masculino	1500m masculino
200m femenino	20km caminata masculino	3000m con obstáculo masculino	400m masculino
400m obstáculos masculino	4x100m relevo masculino	4x400m relevo masculino	5000m masculino
50km caminata masculino	800m masculino	decatlón masculino	lanzamiento de disco masculino
lanzamiento de martillo masculino	salto de altura masculino	lanzamiento de jabalina masculino	salto largo masculino
maratón masculino	salto con espiga masculino	lanzamiento de bala masculino	salto triple masculino
10000m femenino	100m femenino	100m obstáculos femenino	1500m femenino
200m femenino	20km caminata femenino	3000m con obstáculo femenino	400m femenino
400m obstáculos femenino	4x100m relevo femenino	4x400m relevo femenino	5000m femenino
800m femenino	lanzamiento de disco femenino	lanzamiento de martillo femenino	heptatlón femenino
salto alto femenino	lanzamiento de jabalina femenino	salto largo femenino	maratón femenino
salto con espiga femenino	lanzamiento de bala femenino	salto triple femenino	

Otra opción es visualizar el calendario, para ello hay que pulsar “calendario” en el menú de la izquierda y a continuación elegir el deporte del que se esté interesado.

Usuarios

Registro

Login

Noticias

Titulares

Última hora

Deportivas

Galería de fotos

Olimpiadas2020

Disciplinas

Calendario

Resultados

Medallas

Distribución de la Villa deportiva

Entradas

Calendario

Ubicación de instalaciones

Gestión de entradas

Atletismo

Badminton

Balónmano

Basketbol

Boxeo

Canotaje

Ciclismo

Equitación

Esgrima

Futbol

Ginmasia

Gimnasia

Halterofilia

Hockey

Judo

Lucha

Lucha libre

Natación

Pentatlón

Reno

Softbol

TaeKondo

Tenis

Tenis de mesa

Tiro con arco

Tirritation

Vela

Voleibol

Legenda:

Fecha de apertura

Fecha de clausura

Competición

Finales

Apertura

Competición

Competición

Competición

Competición

Competición

Competición

Competición

Final

Final

Clausura

8

10

11

12

13

14

15

16

17

24

Si queremos visualizar los eventos hay que seleccionar “resultados” del menú, a continuación el deporte y luego la fase de competición en la que ha tenido lugar el partido.



Si nos registramos como árbitros, al entrar en los resultados y seleccionar el deporte y la fase de la competición no aparece un botón, “generar nueva acta”, que precisamente sirve para eso, para poner un nuevo resultado en la aplicación.

La siguiente secuencia de imágenes describe muy bien la acción. Al rellenar el acta tienes que añadir los equipos participantes, el resultado, las incidencias, las lesiones, las sanciones y las observaciones.

Visualización:

Usuarios
Modificar datos
Logout

Noticias
Titulares
Última hora
Deportivas
Galería de fotos

Olimpiadas2020
Disciplinas
Calendario
Resultados
Medallas
Distribución de la
Villa deportiva

Entradas
Calendario
Ubicación de
instalaciones
Gestión de entradas

AtletismoBalonmanoBoxeoCiclismoEsgrimaGimnasiaHockeyLuchaLibrePentatlónSoftbolTennisTiro con arcoVelaBadmintonBasquetbolCanotajeEquitaciónFútbolHalterofiliaJudoNataciónRemoTaekwondoTenis de mesaTriatlónVoleibol

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	Melzer, Jurgen Borrar Acta	
	Djokovic, Novak Leer acta	2 - 0 Modificar Acta	Monfils, Gael Borrar Acta	

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

Creación acta:

Usuarios
Modificar datos
Logout

Noticias
Titulares
Última hora
Deportivas
Galería de fotos

Olimpiadas2020
Disciplinas
Calendario
Resultados
Medallas
Distribución de la
Villa deportiva

Entradas
Calendario
Ubicación de
instalaciones
Gestión de entradas

AtletismoBalonmanoBoxeoCiclismoEsgrimaGimnasiaHockeyLuchaLibrePentatlónSoftbolTennisTiro con arcoVelaBadmintonBasquetbolCanotajeEquitaciónFútbolHalterofiliaJudoNataciónRemoTaekwondoTenis de mesaTriatlónVoleibol

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	Melzer, Jurgen Borrar Acta	
	Djokovic, Novak Leer acta	2 - 0 Modificar Acta	Monfils, Gael Borrar Acta	

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

Introducción de datos:

Usuarios
 Modificar datos
 Logout

Noticias
 Titulares
 Última hora
 Deportivas
 Galería de fotos

Olimpiadas2020
 Disciplinas
 Calendario
Resultados
 Medallas
 Distribución de la Villa deportiva

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

Sanciones

Observaciones

Anterior
Siguiente
Guardar
Cancelar

Al pinchar en “medallas” nos saldrá un listado ordenado de las federaciones con el número de medallas de cada tipo, ordenado de mayor a menor.

The image shows a web application interface for the 2020 Olympics. The interface is divided into a left sidebar with navigation links and a main content area with a table of medalists.

Left Sidebar Navigation:

- Usuarios**
 - Registro
 - Login
- Noticias**
 - Titulares
 - Última hora
 - Deportivas
 - Galería de fotos
- Olimpiadas2020**
 - Disciplinas
 - Calendario
 - Resultados
 - Medallas** (highlighted)
 - Distribución de la Villa deportiva
- Entradas**
 - Calendario
 - Ubicación de instalaciones
 - Gestión de entradas

Main Content Area:

The main content area displays a table of medalists. The table has a header row with the following columns: **Federación**, **Medalla de Oro** (Gold Medal), **Medalla de Plata** (Silver Medal), **Medalla de Bronce** (Bronze Medal), **Federación**, **Medalla de Oro**, **Medalla de Plata**, and **Medalla de Bronce**. The table body is currently empty, showing only the header row.

■ Gestión de las entradas

La interfaz para gestionar la venta de entradas también tiene varias opciones. La primera y para tener una visión global de la competición, es un calendario, como se observa en la siguiente imagen:



Si queremos comprar una entrada, pinchamos en “gestión de entradas”. En el caso de que no estemos registrados, saldrá un formulario, al igual que si seleccionamos “registro”.

Usuarios
[Registro](#)
[Login](#)

Noticias
[Titulares](#)
[Última hora](#)
[Deportivas](#)
[Galería de fotos](#)

Olimpiadas2020
[Disciplinas](#)
[Calendario](#)
[Resultados](#)
[Medallas](#)
[Distribución de la Villa deportiva](#)

Entradas
[Calendario](#)
[Ubicación de instalaciones](#)
[Gestión de entradas](#)

Alta de usuario

Debe registrarse antes de realizar compras y/o reservas

ID Usuario	<input type="text"/>	País	<input type="text"/>
Contraseña	<input type="password"/>	Email	<input type="text"/>
Repetir contraseña	<input type="password"/>	Movil	<input type="text"/>
DNI/NIF/...	<input type="text"/>	Profesion	<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.

Si por el contrario ya estamos registrados, nos saldrá una ventana donde podremos comprar una nueva entrada o ver/modificar alguna entrada ya comprada.

Al seleccionar nueva entrada, veremos un historial con las entradas compradas, las cuales las podremos modificar y anular pinchando los botones “modificar” y “anular” que hay a la izquierda de cada entrada.

Usuarios
[Modificar datos](#)
[Logout](#)

Noticias
[Titulares](#)
[Última hora](#)
[Deportivas](#)
[Galería de fotos](#)

Olimpiadas2020
[Disciplinas](#)
[Calendario](#)
[Resultados](#)
[Medallas](#)
[Distribución de la Villa deportiva](#)

Entradas
[Calendario](#)
[Ubicación de instalaciones](#)
[Gestión de entradas](#)

[Nueva entrada](#)
[Reservas, anulaciones y compras](#)

No tiene reservas que pueda anular o comprar



Al seleccionar una nueva entrada, pincharemos en “nueva entrada”, seleccionaremos el deporte, la fase de competición y por último reservaremos una entrada. Esta entrada permanecerá reservada un tiempo, mientras llevamos a cabo las gestiones de pago.



Usuarios

Modificar datos

Logout

Noticias

Titulares

Última hora

Deportivas

Galería de fotos

Olimpiadas2020

Disciplinas

Calendario

Resultados

Medallas

Distribución de la Villa deportiva

Entradas

Calendario

Ubicación de instalaciones

Gestión de entradas

Nueva entrada

Reservas, anulaciones y compras

Atletismo

Balonmano

Boxeo

Ciclismo

Esgrima

Gimnasia

Hockey

Lucha libre

Pentatlón

Softbol

Tenis

Tiro con arco

Vela

Badminton

Barquetbol

Canotaje

Equitación

Fútbol

Halterofilia

Judo

Natación

Remo

Taekwondo

Tenis de mesa

Triatlón

Voleibol

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
	BODNUNNON Manus	IGLESIAS Roniel		Boxeo Semifinal 1 64 Kg. Estadio Olímpico Nacional 16:00 hh

Modificar características

Forma de pago

☒ He leído y acepto las condiciones de privacidad. Leer

Reservar

Comprar

Anterior

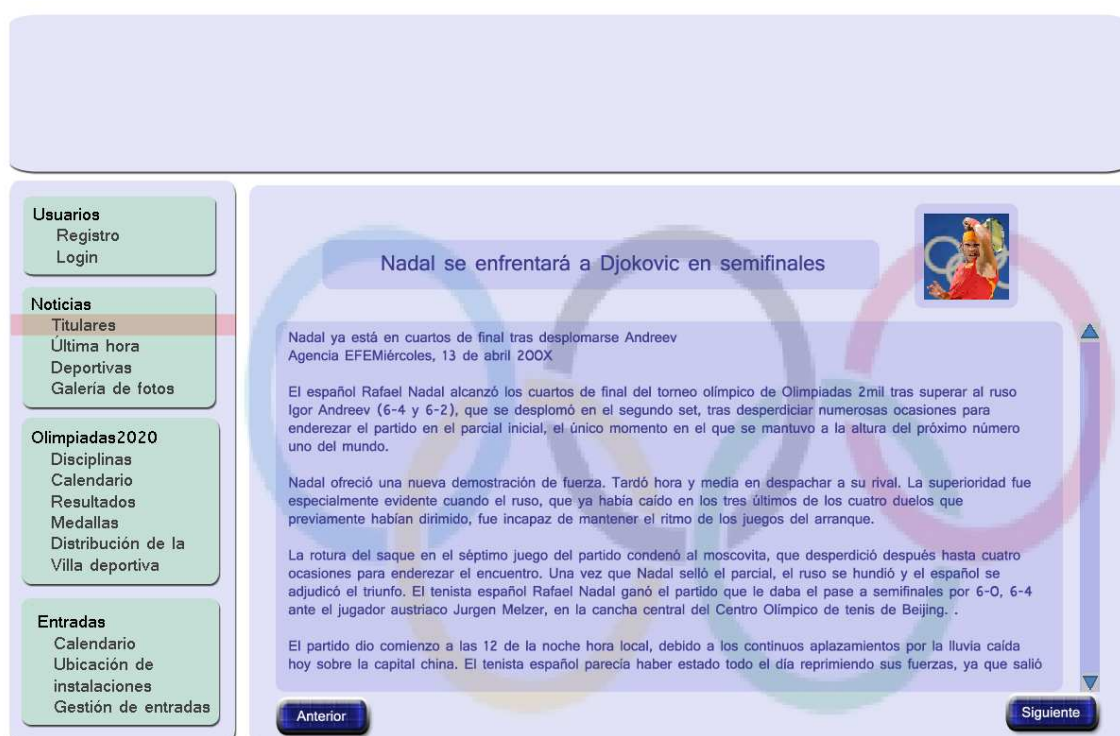
Posterior

■ Interfaz de noticias

La página principal de la aplicación web es la que se muestra a continuación, que coincide con la parte de noticias. Todo el mundo que entre a la aplicación podrá disfrutar de esta interfaz.

En la parte superior de todas las interfaces aparecerá un hueco, en este hueco se colocará el logo de cada olimpiada.

Habrà una única interfaz de noticias que será común a todas las disciplinas olímpicas, apareciendo en orden inverso al suceso de la noticias, es decir, las noticias más recientes serán las primeras en aparecer. Habrá dos botones de anterior y siguiente para moverte de una noticia a otra.



Si te registras como COI, al lado del enlace "noticias", situado en el menú de la izquierda, aparece la opción de editar que servirá para crear una nueva noticia, modificar alguna anterior o eliminarla.

En la siguiente imagen se puede ver claramente que para crear una noticia hay que pinchar el botón "nuevo" situado en la parte superior y para modificar o eliminar hay que pinchar en los botones "modificar" y "eliminar" situados a la derecha de cada una de las noticias.



REALIZACIÓN DEL PLAN DE CALIDAD

PARA QUÉ

Para planificar las tareas a abordar en el marco del proyecto que aseguren la calidad de los productos que se entregarán al cliente, y el futuro mantenimiento o evolución de los mismos.

CUÁNDO

Una vez la Oferta esté enviada al Cliente y la Solicitud de Inicio de Proyecto aprobada, y el hito metodológico de Fin de Fase de Viabilidad aparezca en SAP.

Como muy tarde en un plazo prudencial después de la aceptación de la Oferta y siempre antes de tramitar entregas al Cliente.

CÓMO

El Jefe/Coordinador Técnico elabora el Plan de Calidad a través de la Aplicación Web de Plan de Calidad. En él:

- Identifica el ciclo de vida del proyecto.
- Incluye la información del proyecto relativa a las actividades de soporte, los proyectos relacionados, etc.
- Identifica las fases y actividades técnicas, de gestión y de aseguramiento de calidad aplicables al proyecto, justificando cualquier desviación
- Asigna las entregas comprometidas en las fases pertinentes.
- Envía el Plan de Calidad (provisional) a la División de Metodología e Ingeniería de SW.

La División de Metodología e Ingeniería de SW lo revisa y aprueba. Si detecta errores se lo devuelve al Jefe de Proyecto para que los corrija. Una vez aprobado el Plan de Calidad, los hitos y actividades planificados se registran en el Sistema de Información.

La División de Calidad de Producto programará sus actividades de seguimiento y control, de acuerdo con lo recogido en el Plan de Calidad (Ver proceso de Aceptación de Fines de Fase).

NO SE DEBE

Eliminar actividades porque no las haya pedido el Cliente.

El Plan de Calidad es un compromiso formal del jefe/coordinador técnico de proyecto por lo que no es conveniente incluir actividades que no vayan a realizarse.

IMPORTANTE

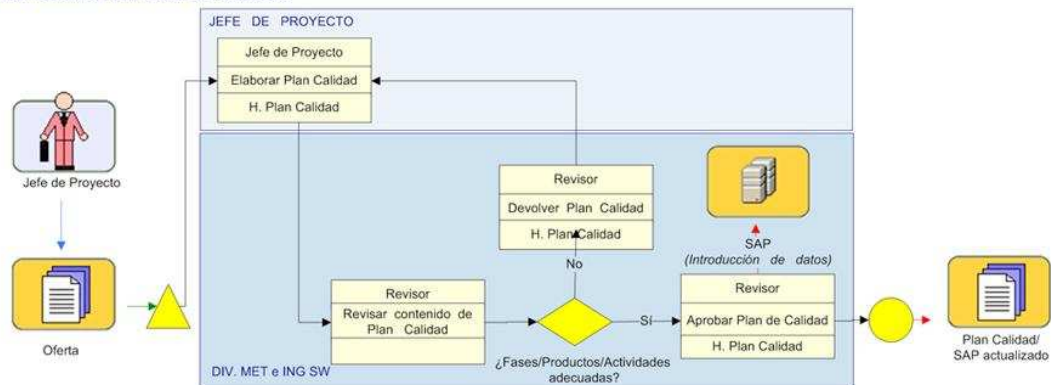
Existen proyectos para los cuales en principio no se considera necesario realizar un Plan de Calidad:

- Proyectos Europeos o Nacionales
- Aquellos en cuya Oferta no se compromete la entrega de productos al cliente (éstos suelen corresponder a proyectos de tipo Evolución y Soporte, etc.)

Hay proyectos a los que se les exige inicialmente la elaboración del Plan de Calidad, pero si el Jefe/Coordinador Técnico del Proyecto considera que no es aplicable la realización de una planificación inicial de actividades de calidad, deberá solicitar la no necesidad de Plan de Calidad a la División de Metodología e Ingeniería Software.

FLUJO DE ACTIVIDADES

REALIZACIÓN DE UN PLAN DE CALIDAD



“LOG”

CUESTIONARIO OLIMPIADAS 2MIL (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 sumía 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)

■ **NIVEL 4 (Gestionado)**

● **¿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 hablar con Marina, 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 participantes, árbitros y personas 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, villa olímpica, 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.