**“Olympic games London 2012 - Football”**

Master en Desarrollo de Aplicaciones para Internet y Dispositivos Móviles

Francisco Caro Díaz

David Caviedes Márquez

## Índice General

## Descripción de la aplicación de ejemplo

* 1. Presentación proyecto máster para la aplicación web

Como punto de partido nuestro propósito era el desarrollo de un proyecto J2EE que nos brindara la posibilidad de simular a tiempo real todo lo relacionado con los deportes de las olimpiadas. Con vistas a que sería un proyecto fácilmente expandible para futuras propuestas y con expectativas de querer abarcar la casi totalidad de lo impartido en el curso nos decidimos por empezar esta aplicación web multi capa.

Se trata de una aplicación web donde diferenciamos las diferentes capas en las que podríamos trabajar en el proyecto:

* Un capa de presentación donde nos centramos en el uso de JSF como framework de desarrollo.
* Una capa de negocio implementada con componentes EJB, haciendo uso de JPA para mecanismo de interacción con la base de datos.
* Una capa de datos para la interacción con la base de datos oracle usando el patrón de diseño DAO.

Para la realización del proyecto nos decantamos por Netbeans como nuestro IDE de desarrollo haciendo uso del servidor de aplicaciones GlashFish que trae por defecto aunque también intentamos parte del desarrollo usar el servidor de aplicaciones de Oracle pero por falta de tiempo no llegamos a ello, y optamos por GlashFish.

Para el manejo de la base de datos, no usamos la predeterminado por Netbeans y nos adentramos en el uso del servidor Oracle Database 10g Express Edition al cual podríamos acceder también fácilmente desde nuestra aplicación escritorio realizada desde Jdeveloper o desde nuestra aplicación móvil realizada con Eclipse.

Tras instalar Jdeveloper, Netbeans y nuestro servidor Oracle empezamos con el diagrama de la base de datos sobre Jdeveloper, haciendo uso así de todo lo visto en clase. También usamos el programa SQLDeveloper para así abstenernos más en el manejo de datos de la base de datos y de paso hacer uso de otro programa y tener más conocimientos.

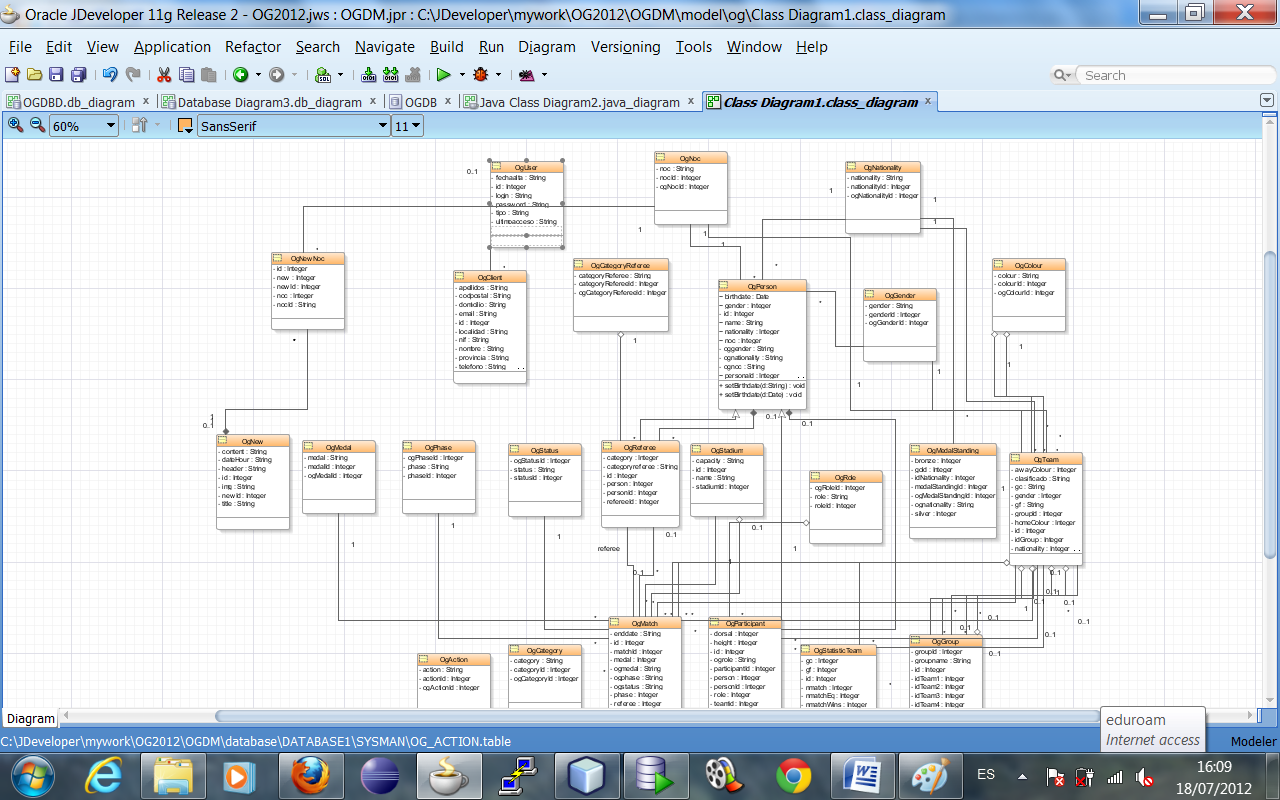
Tras hacer un estudio introductorio nos dimos cuenta que nuestra aplicación nos podría venir algo grande así que optamos por el desarrollo de solo un deporte, lo cual serviría como introducción para futuras expansiones.

## Capa de datos

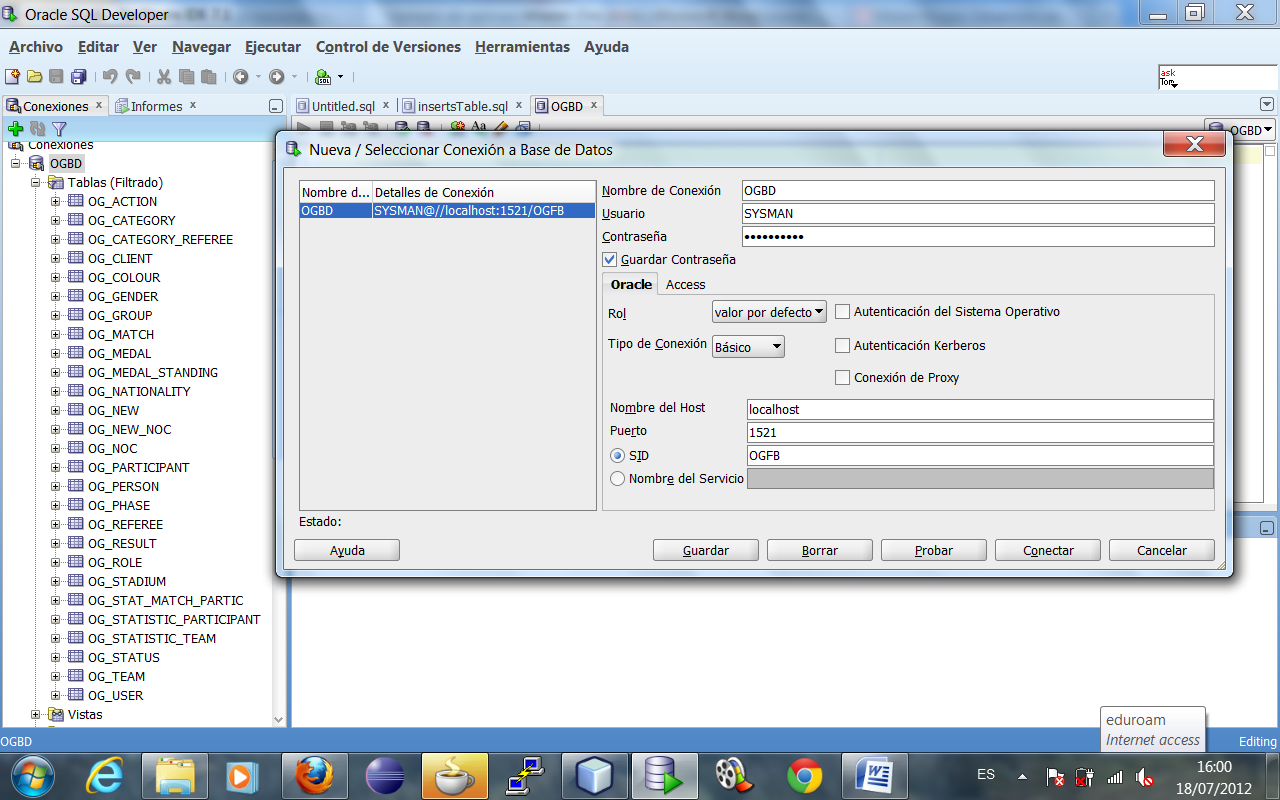
Inicialmente nuestro gran problema fue la gran cantidad de datos que íbamos a manejar ya que simplemente entre los datos de los atletas, los equipos y los usuarios nuestra base de datos tendría muchas tablas y sus correspondientes relaciones.

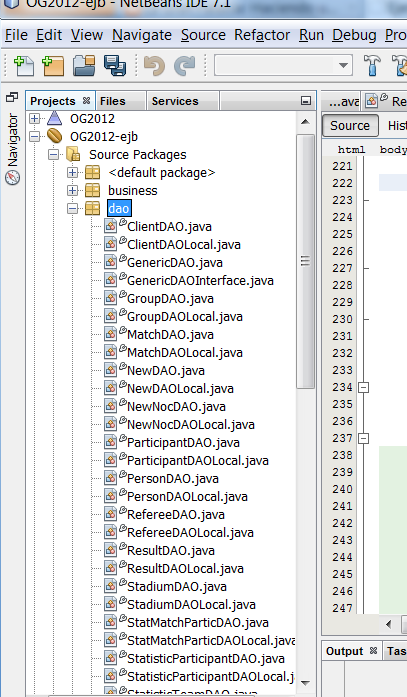
Tras ver cómo podíamos simplificar en algo esta parte nos que una base de datos bastante grande y con grandes cantidades de datos a manejar.

Este es nuestro esquema:



Desde Jdeveloper las vistas sería asi:



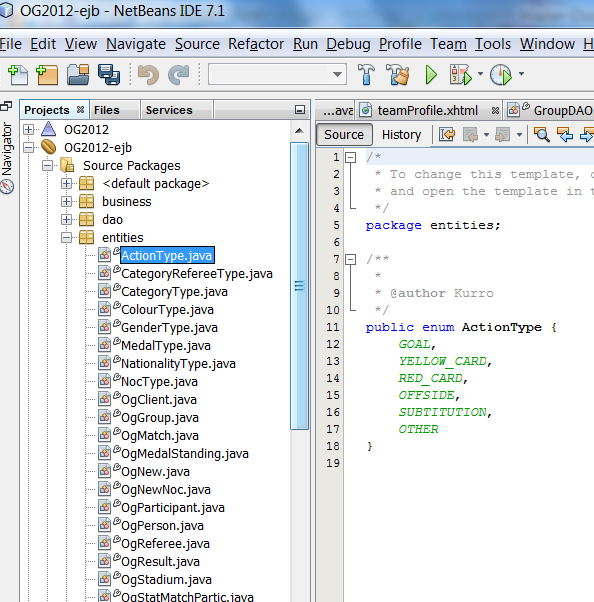
Los ficheros sql para generar, insertar y consultar dichas tablas están adjuntos en el email.

¿Por qué hemos usado DAO?

Hemos usado el patrón de diseño DAO porque es el que nos ofrecía la garantía de abstraernos de las tecnologías usadas y también del modo de acceso a los datos, dejando estás funciones delegadas en el Entity Manager creado con JPA para este fin. Al realizar el trabajo en grupo teníamos aquí además otra ventaja ya que los dos podíamos acceder a la base de datos sin tener en cuenta que teníamos en ella y que tecnología usábamos, además de poder darnos la facilidad de organizar el trabajo mejor ya que uno podría hacer funcionalidades de acceso a la base de datos mientras que el otro podría hacer otras y así abarcar más trabajo. Por este motivo decidimos también el crear un DAO genérico ya que usábamos funciones iguales para muchos casos.

Esta capa es la encargada en nuestra aplicación de dar soporte y funcionalidad a los EJB que están en nuestra capa de negocio.

En nuestro proyecto desarrollado en Netbeans nuestra organización de carpetas por capas nos dan la posibilidad de organizarnos bien, presentamos a continuación la carpeta donde tenemos todos los DAO los cuales accederán a la base de datos directamente y dan soporte a los EJB:

Para la realización de estos DAO hemos hecho el uso del patrón Facade ya que creamos un interfaz local que es por el cual accedemos desde nuestros EJB. Como se puede comprobar tenemos un dao y una interfaz por cada clase java y tabla de base de datos.

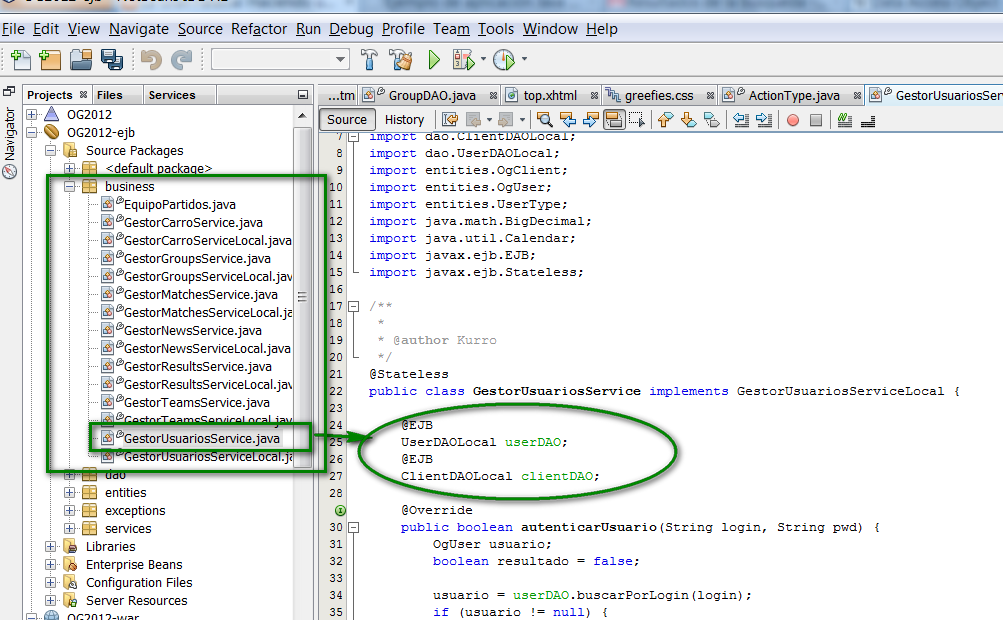
Támbien tenemos en nuestra aplicación una carpeta para nuestra entidades es donde relacionamos todas nuestras tablas con los objetos java.

Algunas tablas las hemos descrito como enumerados como se puede ver en la imagen.

## Capa de lógica de aplicación

Para las reglas de negocio hemos usado EJB 3.0. Los EJB se encargan de tomar el control del negocio y es el encargado de manejar las comunicaciones entre la capa de datos y la capa de presentación. A la vez estamos contribuyendo a que nuestra aplicación sea más simple, mas estructura, y más rápida ya que a la hora de pasar información de un servidor a otro será mucho menos pesada y por lo tanto más rápida. Esto lo pensamos sobre todo porque queríamos hacer uso de XSL a la hora de mostrar las pantallas de nuestra aplicación, siendo estas manejadas por información leida de base de datos. Está idea inicial la cambiamos por el uso de datos directamente leidos de base de datos y mostrados en las páginas JSF en lugar de usar XML y XSL. Aquí tuvimos bastante problemas sobre todo por la persistencia con la base de datos, ya que si partiamos de un esqueña de base de datos erróneo nuestra aplicación no funcionaría correctamente.

Se observa en la imagen nuestra carpeta con todos los Session Beans con sus interfaces:



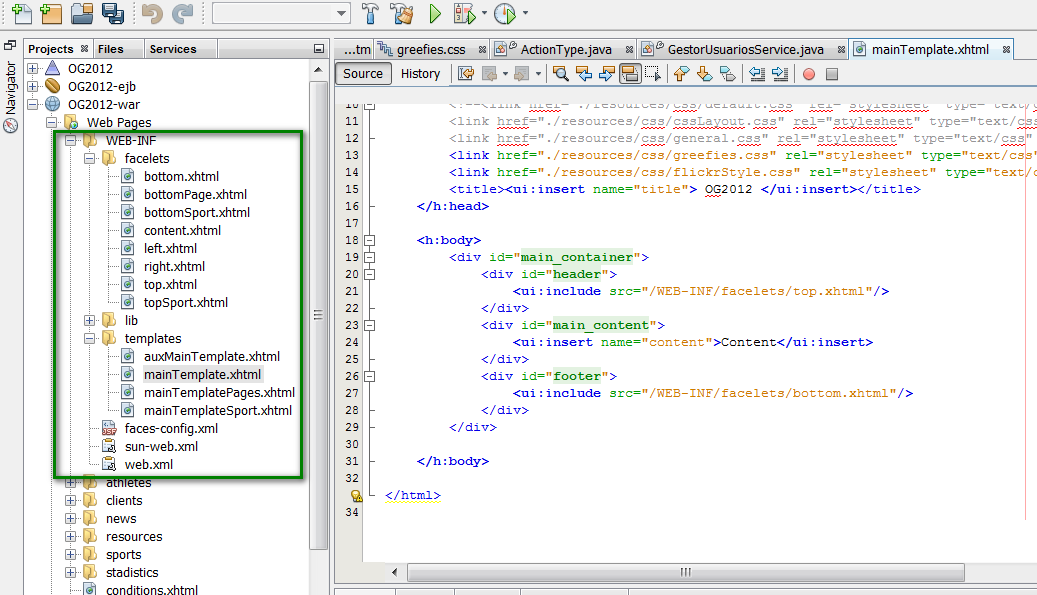
Estos sesión beans como vemos hacen uso de los servicios dados por la base de datos y a través de sus interfaces son los que darán funcionalidad a los managed bean que explicaremos a continuación.

Anotamos aquí que el único EJB de nuestra aplicación sin estado será el encargado de elegir un país favorito, ya que no nos hace falta guardar esta información aunque si la tendremos que mantener en la sesión abierta. (Es como una simulación de un carrito de la compra en el cual en lugar de ventras de productos ofertamos equipos olímpicos a seguir en nuestra aplicación, se ve este trato en la pantalla fb.xhtml).

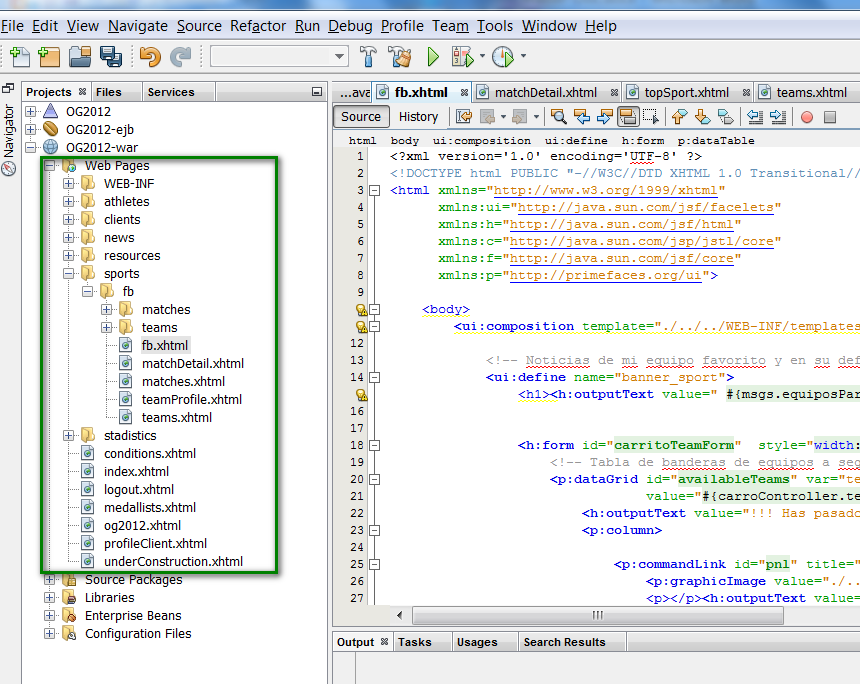
Los otros ejb son los encargados de dar otros servicios por ejemplo en el de los partidos, o los grupos, el manejo de la información de las noticias o la información del usuario logueado.

## Capa de presentación Web

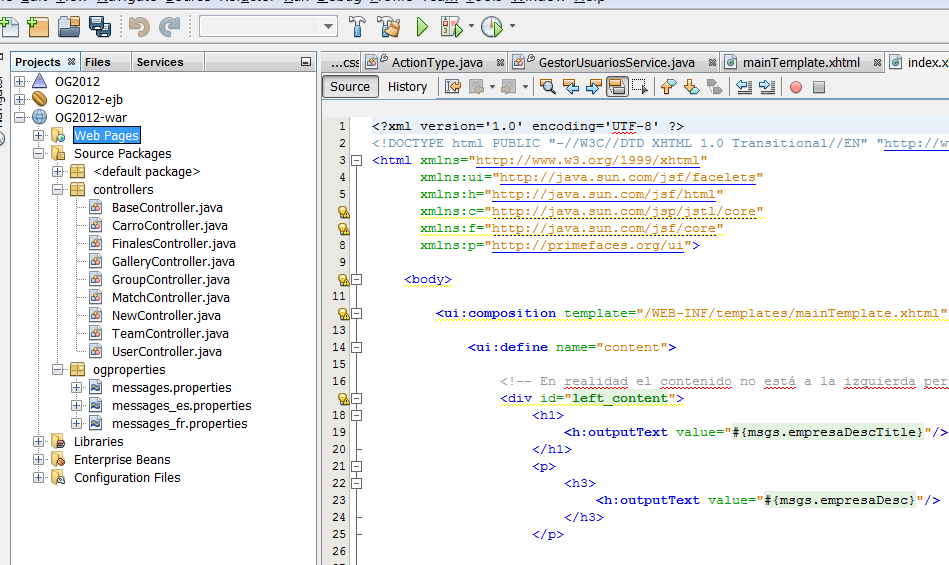
La capa de presentación de la aplicación se ha desarrollado con el framwork JSF haciendo uso de los Facelets para tener una definición de las vistas en lugar de escribir html, jsp u otra tecnología simplemente. En nuestra aplicación está presente en la carpeta del proyecto war:



Las plantillas nos han dado la posibilidad de poder reutilizar códigos común para varias páginas , además haciendo uso también las páginas JSF , usando los managed bean y el fichero faces-config.xml que otorga el framework las reglas de navegación y el flujo de una página a otra se hizo más fácil. Las páginas JSF las tenemos divididas por dominio como se presenta en la imagen siguiente:



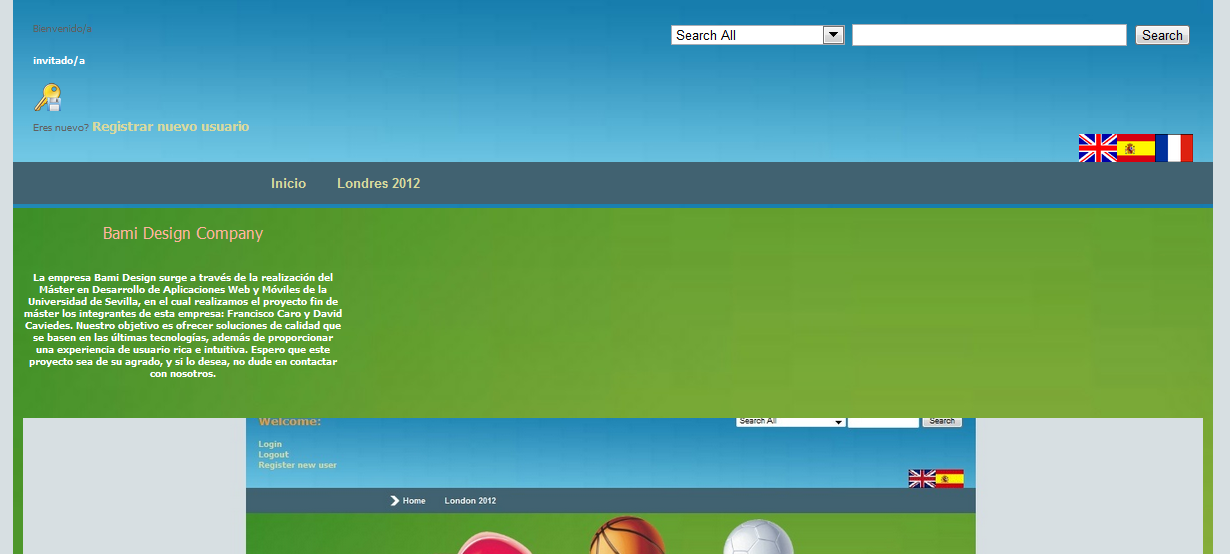
Los Managed bean que hemos creados son los que se encargan del manejo de las interacciones entre los cliente y nuestra aplicación. Entre ellos tenemos los encargados de nuestros usuarios, la competición o el manejo de la información de un equipo. Estos beans están en la carpeta fuente de más abajo:



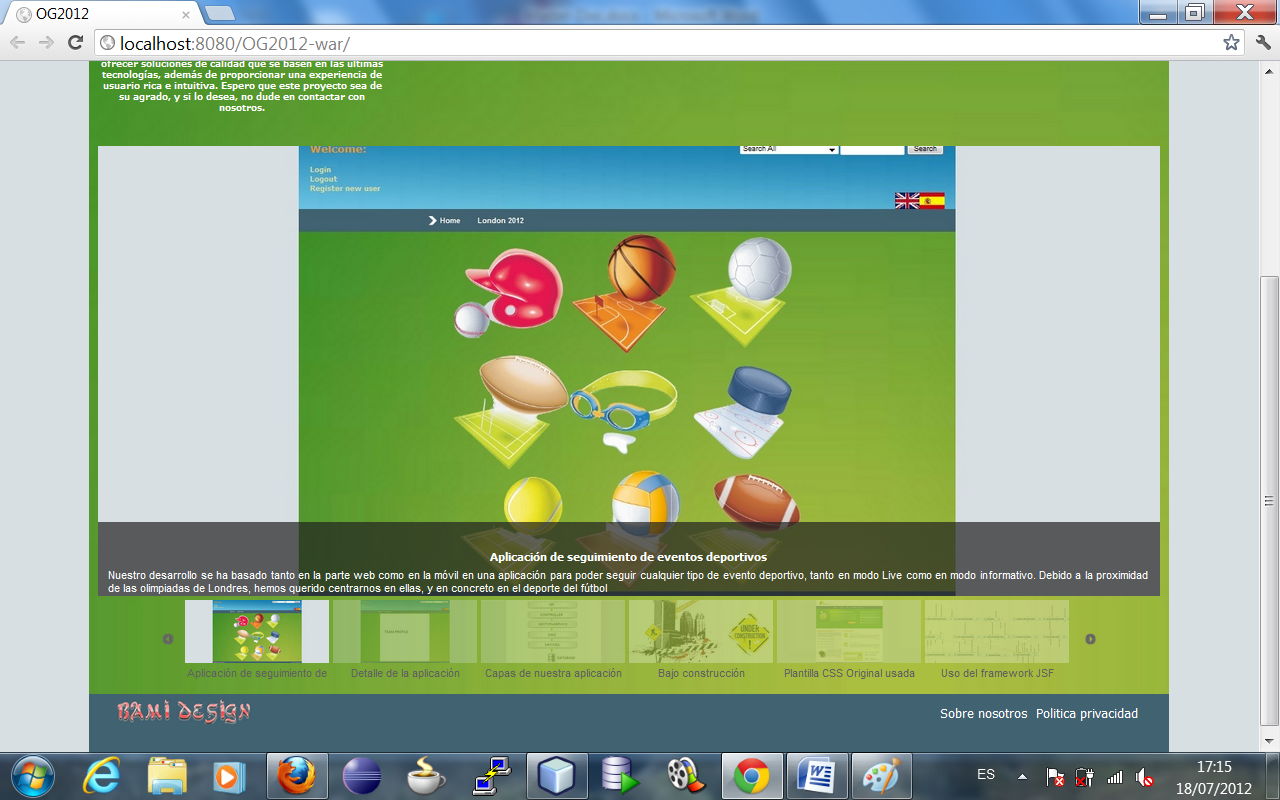
Más abajo tenemos los ficheros properties que dotan a nuestra aplicación de la internacionalización.

Además del uso de funciones javascript, hemos hecho uso de AJAX junto con JSF. Además a destacar es el uso de la librería de Primefaces el cual nos ha dotado a la aplicación de más calidad en la presentación en nuestras pantallas.

Si arrancamos nuestra aplicación en el indez ya podemos ver el uso que hacemos de estas librerías:



Uso de JQUERY:



Otro aspecto importante es la utilización de librerías CSS por defecto ya desarrolladas de las cuales hemos hecho uso en gran parte de las pantallas. Hemos adaptados esta plantilla Greefies a nuestras necesidades, cambiando fuentes, colores, posiciones de los objetos, fondos, definiendo nuevos estilos.

El desarrollo de la aplicación nos ha servido para refrescar nuestros conocimiento, cimentar muchos de ellos además de investigar en algunos casos y así motivarnos para futuros avances en la misma aplicación ampliando por ejemplo el espectro de deportes tratados. Debido a la configuración y desarrollo de la misma solo se tendría que realizar las pantallas JSF definiendo también las reglas de navegación mismas.